CISCO CONFIDENTIAL

Cloud Service Assurance for VMDC 3.0 Design and Implementation Guide

SDU

Version 1.0

December 2012

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

CONTENTS

FIGURES

TABLES

CHAPTER 1

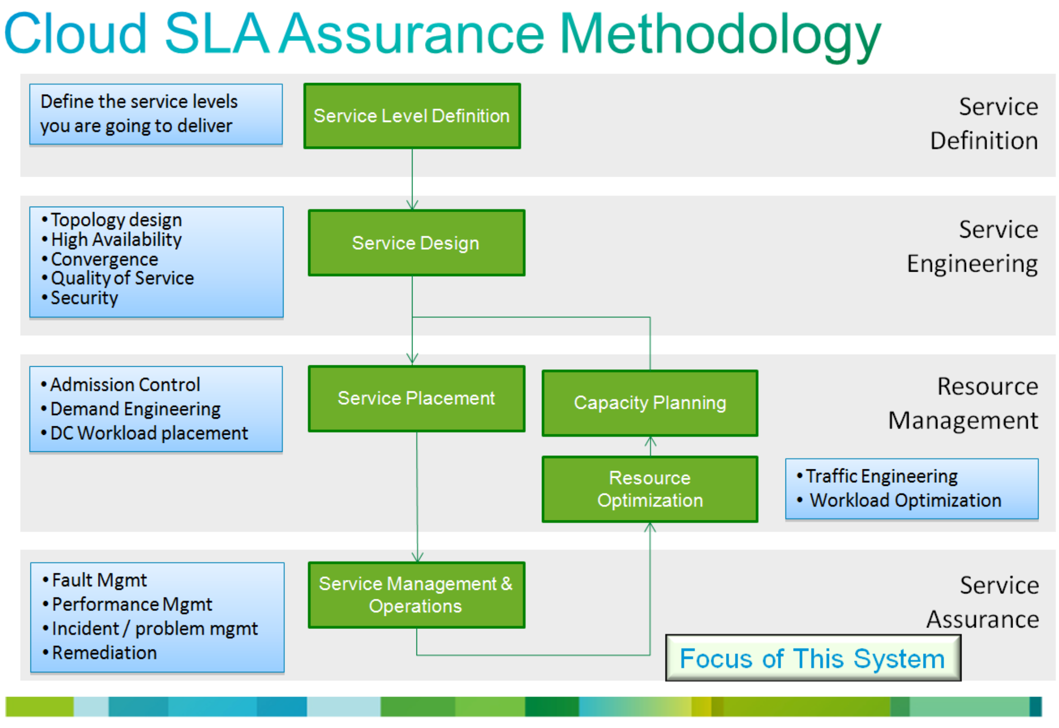
Introduction

In recent years, there has been a race by both traditional Service Providers (SPs) and public cloud providers such as Amazon to capture the cloud services market. SPs have identified the capability to offer Service Level Agreements (SLAs) as their key differentiator in the race for the cloud. In response, SPs are deploying virtual private cloud services accessed by Enterprises (cloud consumers) over the SP's IP/MPLS VPN network infrastructure. In addition, lack of trust had been identified as one of the key barriers for Enterprises to purchase cloud services. To gain end customer trust of cloud services, it is important that a cloud provider offer customers visibility in the performance of their applications hosted in the cloud.

SPs have to take measures both in engineering the service and in operating the service to offer their customers the SLAs necessary to realize the potential of virtual private cloud differentiation. The term "service assurance" is commonly used to refer to performance management and fault management, i.e., monitoring and reporting that the service levels are met and identifying/resolving service impacting faults. More generally, assurance means providing a high level of confidence that a commitment can be met; this encompasses more than just operation and management aspects, but also includes service engineering aspects.

The broader SLA assurance framework with all necessary functions to offer SLAs is illustrated in Figure 1-1. This framework includes service assurance as one of its building blocks, which is the focus of this system and this document. In addition to the virtual private cloud opportunity, service assurance also plays a role in Enterprise private clouds to enable efficient Day 2 operations and gain visibility necessary to optimize resources utilization.

Figure 1-1. Cloud SLA Assurance Methodology



Cloud SLA Assurance Methodology

Define the service levels you are going to deliver

Service Level Definition

Service Definition

- Topology design
- High Availability
- Convergence
- Quality of Service
- Security

Service Design

Service Engineering

- Admission Control
- Demand Engineering
- DC Workload placement

Service Placement

Capacity Planning

Resource Management

Resource Optimization

- Traffic Engineering
- Workload Optimization

- Fault Mgmt
- Performance Mgmt
- Incident / problem mgmt
- Remediation

Service Management & Operations

Service Assurance

Focus of This System

Both Infrastructure as a Service (IaaS) and Software as a Service (SaaS) private and virtual private cloud services can be offered on top of the Virtualized Multiservice Data Center (VMDC) architecture. The Cloud Service Assurance for VMDC (CLSA-VMDC) system provides service assurance capabilities for VMDC, as well as private and virtual private cloud IaaS. This system can also be leveraged as a building block of application-based cloud services such as Cisco Hosted Collaboration Solution (HCS), Cisco Virtualization Experience Infrastructure (VXI), and SP TelePresence.

This chapter presents the following topics:

- Section 1.1 System Purpose
- Section 1.2 System Objectives
- Section 1.3 Key Benefits of Cloud Service Assurance
- Section 1.4 CLSA-VMDC 3.0 Summary of Changes

1.1 System Purpose

This document describes design and implementation guidelines for Cloud Service Assurance for VMDC 3.0 (CLSA-VMDC 3.0), which is the second release of CLSA-VMDC and IaaS-based assurance offers. This version of the system supports VMDC 3.0, VMDC 2.2, VMDC 2.1, and earlier infrastructure architectures. CLSA-VMDC 3.0 is based on Zenoss Cloud Service Assurance (CSA), which was built from the ground up for cloud technology management. Zenoss CSA is a service impact model-based system that allows for rapid new service introduction, tenant-based service assurance, consolidated monitoring of the VMDC infrastructure, and simple customizations that can be deployed without service down time via plugins called ZenPacks.

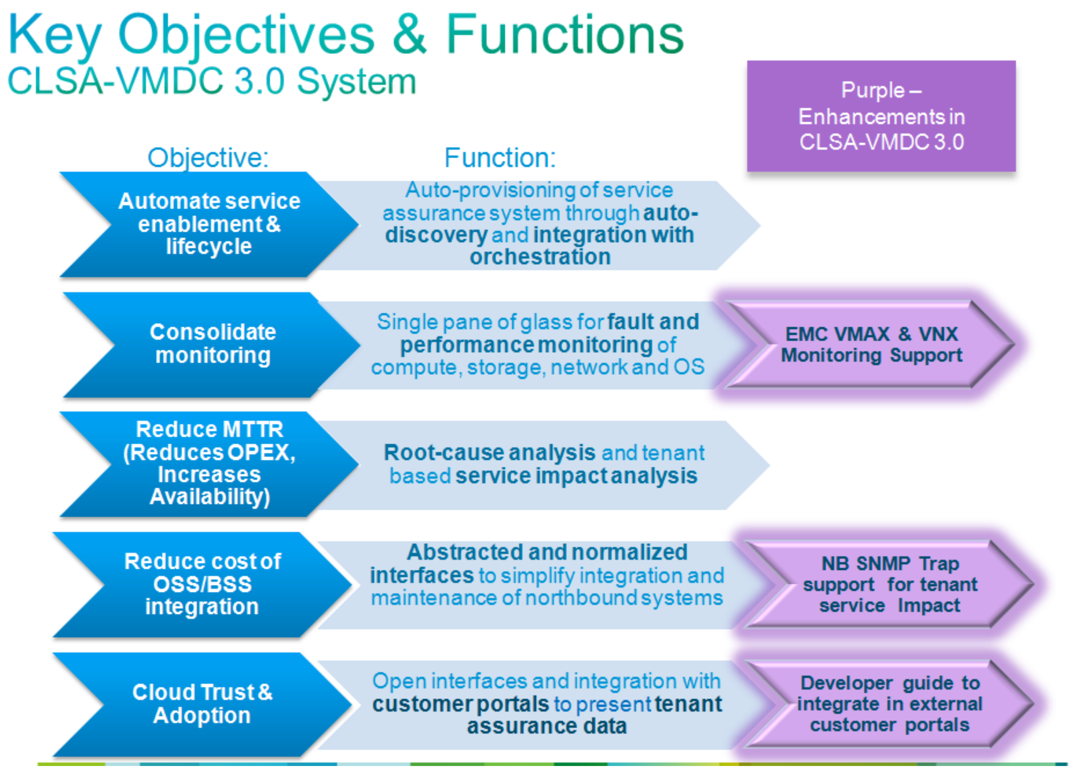
Note While this CLSA-VMDC Design and Implementation Guide (DIG) references the VMDC 3.0 system, previous versions of the VMDC system are also supported. The CLSA-VMDC system also supports other Data Center (DC) designs, as well as the VCE Vblock and NetApp FlexPod stacks.

Zenoss CSA is a multiservice system that offers real time aggregated dashboards as well as reporting capabilities. The system can be deployed both in centralized and distributed architecture and allows for incremental deployment growth. While it offers rich functionality for IaaS domains, the solution is lightweight and has open interfaces to allow for simple integration into existing Operations Support System (OSS) and ticketing systems with minimal cost. As such, this solution is positioned not as a replacement, but as a complement to existing Manager-of-Manager (MOM) systems (e.g., IBM Netcool), ticketing systems (e.g., BMC Remedy), and so on.

1.2 System Objectives

The key business objectives of the CLSA-VMDC 3.0 system and the respective technical functions that realize these benefits are illustrated in Figure 1-2 and discussed throughout this document.

Figure 1-2. Key Objectives and Functions of CLSA-VMDC 3.0



Key Objectives & Functions
CLSA-VMDC 3.0 System

Purple – Enhancements in CLSA-VMDC 3.0

Objective:
Function:

Automate service enablement & lifecycle
Auto-provisioning of service assurance system through **auto-discovery** and **integration with orchestration**

Consolidate monitoring
Single pane of glass for **fault and performance monitoring** of compute, storage, network and OS
EMC VMAX & VNX Monitoring Support

Reduce MTTR (Reduces OPEX, Increases Availability)
Root-cause analysis and tenant based **service impact analysis**

Reduce cost of OSS/BSS integration
Abstracted and normalized interfaces to simplify integration and maintenance of northbound systems
NB SNMP Trap support for tenant service Impact

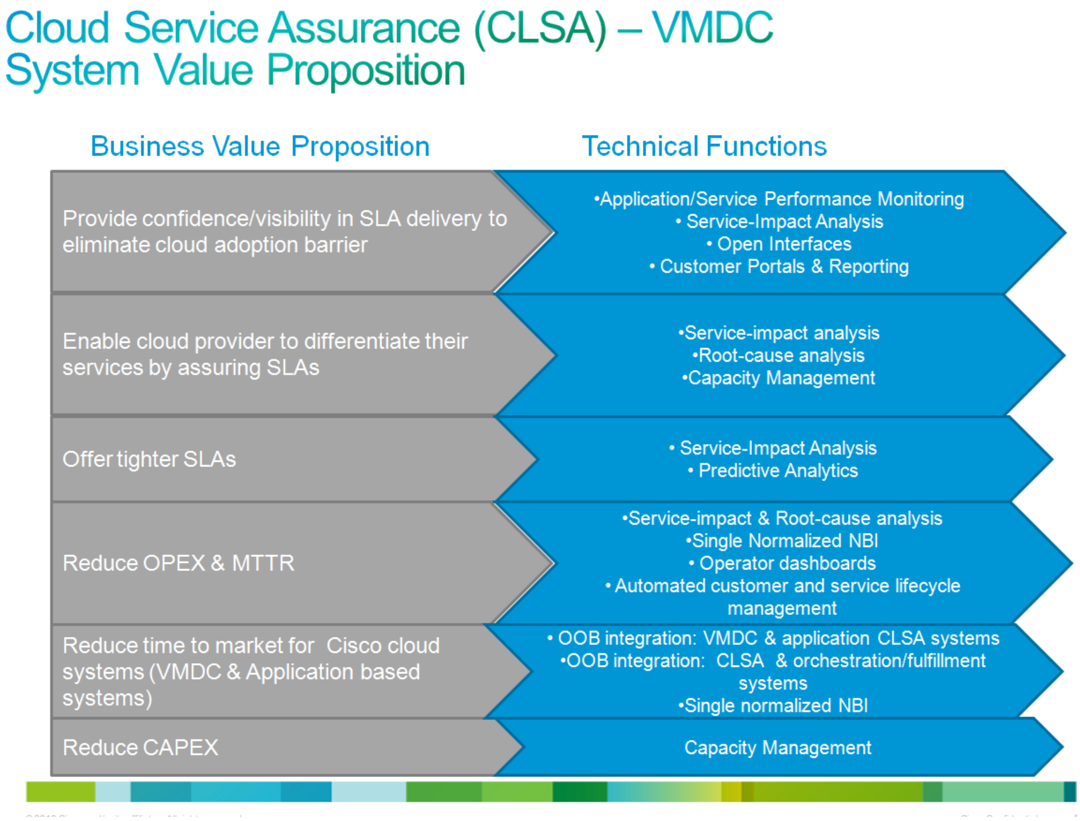
Cloud Trust & Adoption
Open interfaces and integration with **customer portals** to present **tenant assurance data**
Developer guide to integrate in external customer portals

Section 1.3 Key Benefits of Cloud Service Assurance provides more in-depth discussion on the benefits of cloud service assurance.

1.3 Key Benefits of Cloud Service Assurance

Figure 1-3 outlines the key business value propositions of cloud service assurance and the technical functions that help realize these value propositions.

Figure 1-3. Key Benefits of Cloud Service Assurance



Cloud Service Assurance (CLSA) – VMDC System Value Proposition

Business Value Proposition	Technical Functions
Provide confidence/visibility in SLA delivery to eliminate cloud adoption barrier	•Application/Service Performance Monitoring • Service-Impact Analysis • Open Interfaces • Customer Portals & Reporting
Enable cloud provider to differentiate their services by assuring SLAs	•Service-impact analysis •Root-cause analysis •Capacity Management
Offer tighter SLAs	• Service-Impact Analysis • Predictive Analytics
Reduce OPEX & MTTR	•Service-impact & Root-cause analysis •Single Normalized NBI • Operator dashboards • Automated customer and service lifecycle management
Reduce time to market for Cisco cloud systems (VMDC & Application based systems)	• OOB integration: VMDC & application CLSA systems •OOB integration: CLSA & orchestration/fulfillment systems •Single normalized NBI
Reduce CAPEX	Capacity Management

Cloud service assurance focuses on solving the following four key customer problem statements:

- Automating service enablement
- Consolidated monitoring
- Reducing Mean Time to Repair (MTTR)
- Northbound OSS/BSS integration

1.3.1 Automate Service Enablement

As previously noted, assurance services are a key component of the overall cloud service offering. In order to enable and manage the lifecycle of assurance services, a significant amount of manual configuration may be required. In cloud environments that call for self-service and large scale, automatic enablement of service assurance is required. Automatic enablement of service assurance can be achieved in a couple of different ways. Fundamentally, the following approaches can be taken to automate service enablement and life cycle:

1. Reduce necessary amount of configuration (by using technology that is self learning (e.g., self learning thresholds))
2. Automatic discovery (by assurance system)
3. Programmatic orchestrated provisioning (via integration with orchestration system)

CLSA-VMDC utilizes all of the above methods to automate service enablement with specific emphasis on automatic discovery.

The following types of objects are automatically discovered in CLSA-VMDC:

- Monitored devices (e.g., UCS, Nexus 7000, MDS 9000, etc.)
- Sub-components of devices and their relationships (e.g., UCS chassis, blades, fabric interconnect, etc.)
- Tenant-based Service Impact Analysis (SIA) models for the compute (e.g., tenant Virtual Machine (VM) mapping to service impacting dedicated and shared vCenter and UCSM managed resources)

1.3.2 Consolidated Monitoring

Due to the large number of components and technologies in many of the SP and IT systems, operations staff are typically segmented and specialized, and they utilize a number of customized tools. This operations staff division of labor results in a monitoring approach that involves observing multiple screens and interaction between a number of organizations when trying to solve even the simplest problems. For example, there are storage operations that are responsible for storage only using their favorite tool, and similarly, there are compute operations with their staff and tools, network operations, and applications operations, and so on. This approach not only increases Mean Time to Repair (MTTR), and thus customer dissatisfaction, but it will also be unmanageable for cloud systems that are extremely dynamic and deployed at extreme scale. While there will always be a need to have specialized staff with focused expertise, there must be some consolidation of monitoring products to provide a single pane of glass that will simplify Tier 1 and 2 operations.

In addition, in order to fully automate some of operations tasks through value add assurance functions such as Root Cause Analysis (RCA) and SIA, assurance products need to have visibility of all of the components that work together to deliver the service. While segmented visibility will always exist and present challenges in the cloud environment due to business and ownership boundaries, the effort needs to be made to provide as much visibility as possible. More visibility means more value add from the assurance system.

In order to solve visibility challenges, consolidated monitoring and data collection is one of the fundamental functions of any cloud service assurance system. Consolidated monitoring and data collection needs to be done in the following ways:

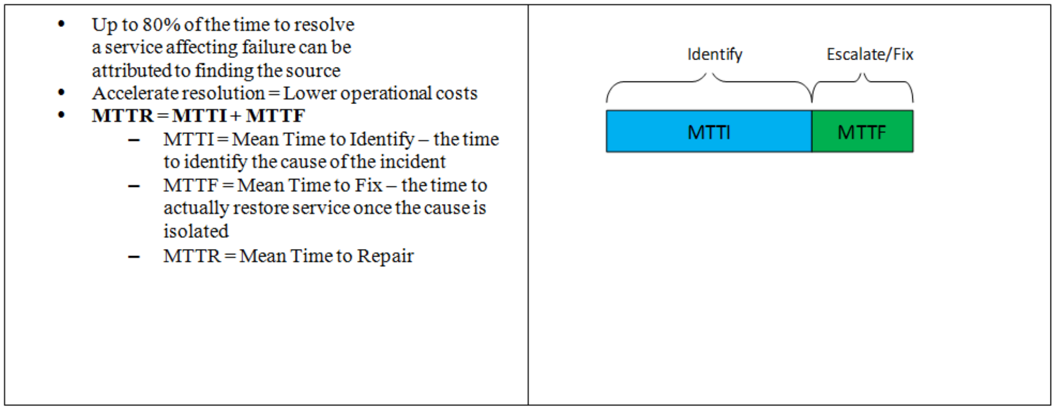
- **Various domains (applications, compute, storage, network).** The cloud assurance system needs to provide a single pane of glass to monitor components from various domains.
- **Fault and performance data.** The cloud assurance system needs to consolidate fault and performance data and leverage both for all of its higher order functions like RCA and SIA.
- **Various data sources, interfaces, and protocols.** The cloud assurance system needs to collect data from multiple data sources and protocols and consolidate this data into unified device and service models. Some examples of different data sources and protocols are SNMP, syslog, WS API, Netflow, customer opened tickets, and so on.

Consolidated monitoring provides the visibility necessary to enable the assurance system to provide more value add, while it can still achieve segmentation of operations through Role-based Access Control (RBAC) and flexible and configurable filtering capabilities.

1.3.3 Reducing Mean Time to Repair

In high pressure Network Operations Center (NOC) environments, operators handle various types of faults, isolate the issues, troubleshoot the problems, or escalate the problem to experts. To reduce the end-customer impact, it is very important to continuously improve MTTR. In traditional systems, general guidance for MTTR is less than 30 minutes from problem detection to problem resolution. For the cloud system, there is no generally accepted criteria, but expectations are that it will perform at least no worse than traditional systems.

Figure 1-4. Reducing Mean Time to Repair



- Up to 80% of the time to resolve a service affecting failure can be attributed to finding the source
- Accelerate resolution = Lower operational costs
- **MTTR = MTTI + MTTF**
  - MTTI = Mean Time to Identify – the time to identify the cause of the incident
  - MTTF = Mean Time to Fix – the time to actually restore service once the cause is isolated
  - MTTR = Mean Time to Repair

The VMDC system consists of multiple technologies and components such as compute, storage, network, and network services components. The VMDC system is integrated to leverage these multiple technologies to create a platform for SPs and Enterprises to offer cloud services. Due to the interdependence of the components in the VMDC system, fault and performance issues in these components impact the services offered. The large number of components and technologies necessary to deliver cloud services increases the challenge of identifying the root cause and normalizing and correlating the faults that are generated by each of the individual components.

System scale plays a key role in creating the need for specific notifications about system failures and a reduced set of faults on the NOC operator dashboard. For example, due to the large size of a VMDC system that serves multiple end-customers, the assurance system can potentially generate thousands of events/faults on the NOC dashboard. If the NOC operator has to look at every fault generated by each domain manager, then the NOC operator may become overwhelmed. This can result in a time-consuming task for the NOC operator, who has to review hundreds of events/faults to identify the actionable events and then escalate those to the experts. This fault isolation time period results in higher mean-time-to-investigate/identify, and hence longer MTTR. This all equates to longer downtimes and unsatisfied end customers.

To reduce the MTTR, it is very important that the NOC operators receive specific notifications identifying the root cause of a failure. To achieve this, CLSA-VMDC provides fault processing capabilities across components and domain managers and improves the correlation within the components and domains. CLSA-VMDC refers to RCA that spans across multiple domains as X-domain RCA.

1.3.4 Northbound OSS and BSS Integration

Almost every SP and many large Enterprises have existing OSS/Business Support Systems (BSS) deployed and operational (e.g., ticketing systems, MoM systems, problem and incident management systems, etc.). The SP staff and processes are generally aligned with the existing OSS/BSS workflows. VMDC is a new solution for SPs, however, SPs expect the VMDC assurance solution to integrate with their existing OSS/BSS.

The individual VMDC system components do offer interfaces to integrate with the OSS systems via SNMP Traps, syslogs, and emails. However, since each device and domain manager is an independent application, the integration interfaces are not consistent, and the number of integration points would be large (on the order of dozens of interfaces for VMDC system). Although the assurance domain manager integration northbound with the SP OSS is a one-time task, it needs ongoing maintenance due to:

- Need for ongoing fine-tuning
- Changes in the underlying system and interfaces (e.g., API changes on southbound devices and domain managers)
- Deployment of additional instances of domain managers
- Addition of new components and domain managers in future service assurance enhancements

In order to ease the integration of the VMDC system in existing OSS/BSS systems, and thus SP adoption of the VMDC system, the number of integration points between VMDC and the SP's OSS/BSS needs to be reduced. The SP needs to be shielded from all maintenance and changes in the underlying VMDC system and interfaces unless the change is introducing significant new functionality to the SP. This can be achieved by providing single normalized interfaces from CLSA-VMDC.

1.4 CLSA-VMDC 3.0 Summary of Changes

CLSA-VMDC 3.0 extends the VMDC assurance solution to provide support for several advanced features and to expand coverage of VMDC device discovery and monitoring. The list below identifies the major new features supported in this release:

- Zenoss High Availability Support
- New Zenoss Northbound Service Impact Trap
- New device support for both EMC VMAX and VNX block storage
- Cisco VMDC device families support extended (Nexus, ASA, UCS)
- New Zenoss Sample Tenant Portal

Note CLSA-VMDC version numbering is closely tied to VMDC IaaS releases. As new devices are added to the VMDC infrastructure, CLSA-VMDC will include new device support for discovery and monitoring in follow-on releases. Subsequent CLSA-VMDC releases will also continue to enhance support for SIA and RCA, expanding coverage out-of-the-box for network infrastructure.

Table 1-1 below lists the document updates associated with these enhancements for ease of reference.

Table 1-1. CLSA-VMDC 3.0 Summary of DIG Updates

Section Number	Section Title	Section Description
1.4	CLSA-VMDC 3.0 Summary of Changes	Identifies CLSA-VMDC 3.0 DIG updates (this section)
2	VMDC System Overview	Updated overview of VMDC System to include VMDC 3.0
3.5.5.4	Zenoss Impact SNMP Trap	New section providing details for the Zenoss Service Impact Trap

Section Number	Section Title	Section Description
NA	NA	NA
NA	NA	NA
NA	NA	NA
9.1	EMC	Introduction of EMC VMAX and VNX block devices via SMI-S Provider domain manager
9.2	ASASM	Additional support for ASA family to include ASASM
9.3	Nexus 3000	Addional support for Nexus 3000 family
9.4	FabricPath Line Card	Additional support for VMDC 3.0 Fabric Path Line Card
9.5	UCS 6200 Fabric Interconnect	Additional support for UCS Fabric Interconnect 6200 model
9.6	Nexus 1010	Introducing support for Nexus 1010 and interconnection to Nexus 1000V
9.7	Zenoss Sample Tenant Portal	Discussion of Zenoss CSA JSON API and Zenoss provided sample portal
Appendix A	Best Practices	Updated section
Appendix B	Caveats	Updated section
Appendix C	Key Performance Indicators for VMDC	New section listing KPI statistics by device category
Appendix D	Key Faults for VMDC	New section listing key faults by device category
Appendix E	Linux IPTables	New section providing IPTable configuration when using Linux firewall
Appendix F	Related Documentation	Updated with latest reference documents

CHAPTER 2

VMDC System Overview

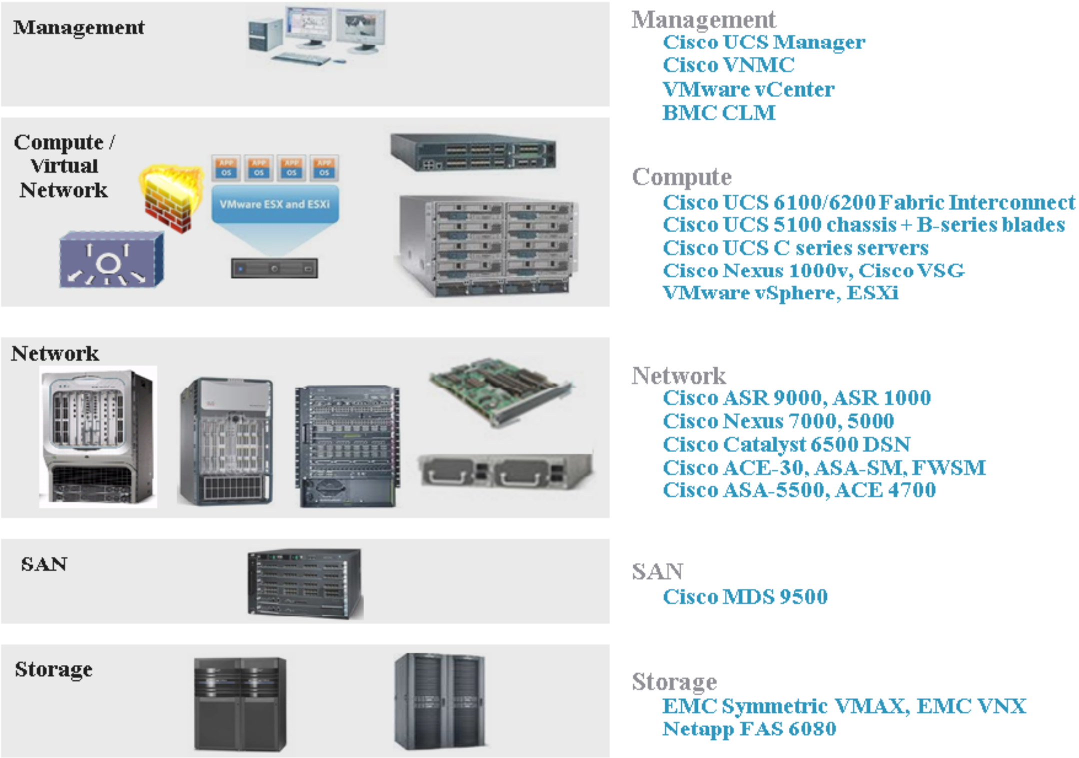
Cloud Service Assurance for VMDC (CLSA-VMDC) is the service assurance system used to monitor Cisco VMDC-based cloud deployments. This chapter provides a brief overview of the VMDC system and its components.

The VMDC system is the Cisco reference architecture for Infrastructure as a Service (IaaS) cloud deployments. This Cisco IaaS cloud architecture is designed around a set of modular Data Center (DC) components consisting of building blocks of resources called PoDs. A PoD, or Point of Delivery, comprises the Cisco Unified Computing System (UCS), SAN and NAS storage arrays, access (switching) layers, aggregation (switching and routing) layers connecting into the Data Center Service Node (DSN)-based services layer, and multiple 10 GE fabric using highly scalable Cisco network switches and routers.

The VMDC system is built around the UCS, Nexus 1000V, Nexus 5000 and Nexus 7000 switches, Multilayer Director Switch (MDS), Aggregation Services Router (ASR) 9000, ASR 1000, Adaptive Security Appliance (ASA) 5585-X or Adaptive Security Appliance Services Module (ASASM), Catalyst 6500 DSN, Application Control Engine (ACE), Nexus 1000V, Virtual Security Gateway (VSG), VMware vSphere, EMC VMAX/VNX, and NetApp FAS storage arrays. Cloud service orchestration is currently provided by the BMC Cloud Lifecycle Management (CLM) suite, and in the future, by Cisco Intelligent Automation for Cloud (CIAC).

Figure 2-1 provides a synopsis of the functional infrastructure components comprising the VMDC system.

Figure 2-1. VMDC Infrastructure Components



This chapter presents the following topics:

- Section 2.1 VMDC Modular Components
- Section 2.2 VMDC System Architecture

2.1 VMDC Modular Components

The VMDC system architecture provides a scalable solution that can address the needs of Enterprise and Service Provider cloud data centers. This architecture enables customers to select the design that best suits their immediate needs while providing a solution that can scale to meet future needs without retooling or redesigning the DC. This scalability is achieved using a hierarchical design with two different modular building blocks, PoD and Integrated Compute Stack (ICS).

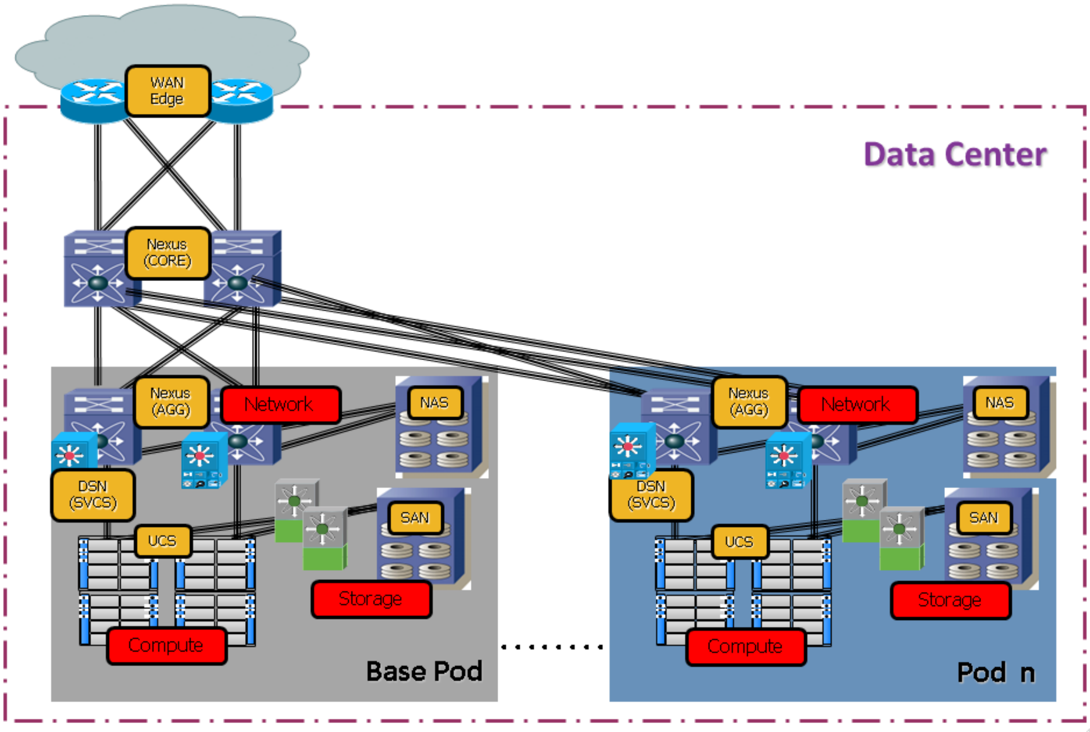
Point of Delivery (PoD)

The modular DC design starts with a basic infrastructure module called a PoD, which is a logical repeatable construct with predictable infrastructure characteristics and deterministic functions. A PoD identifies a modular unit of DC components and enables customers to add network, compute, and storage resources incrementally. This modular architecture provides a predictable set of resource characteristics (network, compute, and storage resource pools and power and space consumption) per unit that are added repeatedly as needed.

In this design, the aggregation layer switch pair, services layer nodes, and one or more integrated compute stacks are contained within a PoD. The PoD connects to the core layer devices in the DC. To scale a DC, additional PoDs can be deployed and connected to the core layer devices.

Figure 2-2 illustrates how PoDs can be used to scale compute, network, and storage in predictable increments within the DC.

Figure 2-2. VMDC PoDs for Scaling the Data Center

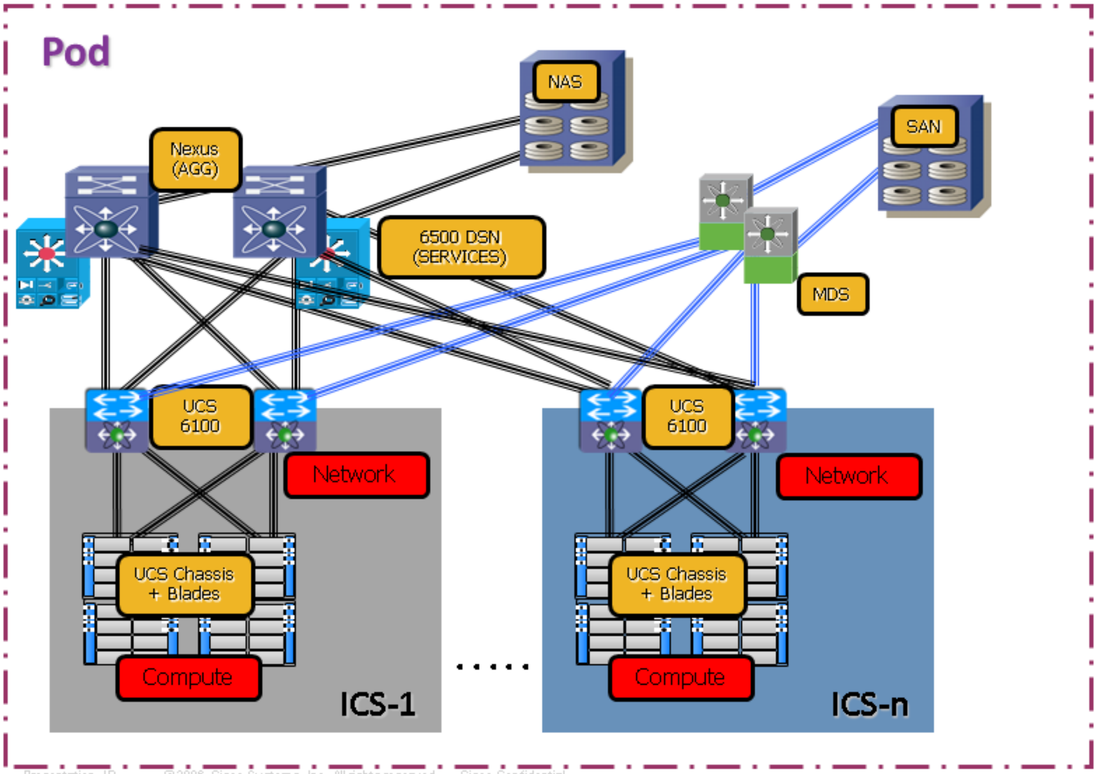


Integrated Compute Stack (ICS)

The second modular building block used is a generic ICS based on existing models, such as the VCE Vblock or NetApp FlexPod infrastructure packages. The VMDC architecture is not limited to a specific ICS definition, but can be extended to include other compute and storage stacks. An ICS can include network, compute, and storage resources in a repeatable unit. In this document, the access layer switch pair, storage, and compute resources are contained within an ICS. To scale a PoD, providers can add additional integrated compute stacks and can continue to scale in this manner until the resources reach the PoD design limit.

Figure 2-3 illustrates how integrated compute stacks can be used to scale the PoD.

Figure 2-3. VMDC ICS for Scaling the Data Center



2.2 VMDC System Architecture

The VMDC system utilizes a hierarchical network design for High Availability (HA) and scalability. The hierarchical or layered DC design uses redundant switches at each layer of the network topology for device-level failover that creates a highly available transport between end nodes using the network. DC networks often require additional services beyond basic packet forwarding, such as Server Load Balancing (SLB), firewall, and intrusion prevention. These services might be introduced as modules populating a slot of one of the switching nodes in the network or as standalone appliance devices. Each service approach also supports the deployment of redundant hardware to preserve High Availability (HA) standards set by the network topology. This layered approach is the basic foundation of the VMDC design to provide scalability, performance, flexibility, resiliency, and service assurance. VLANs and Virtual Routing and Forwarding (VRF) instances are used to provide tenant isolation within the DC architecture, and routing protocols within the VRF instances are utilized to interconnect the different networking and service devices.

The VMDC 2.2 and 3.0 releases are the latest released versions of this architecture. This section provides a brief synopsis of the VMDC 2.2 and 3.0 systems.

Note For detailed information on the VMDC 2.2 system architecture, refer to the following documents:

- VMDC 2.2 Design Guide
- VMDC 2.2 Implementation Guide

For detailed information on the VMDC 3.0 system architecture, refer to the following documents:

- VMDC 3.0 Design Guide
- VMDC 3.0 Implementation Guide

Information on previous VMDC system releases can be found at VMDC System Releases.

Note While this CLSA-VMDC Design and Implementation Guide (DIG) references the VMDC 2.2 and 3.0 systems, previous versions of the VMDC system are also supported. The CLSA-VMDC system also supports other DC designs, as well as the VCE Vblock and NetApp FlexPod stacks.

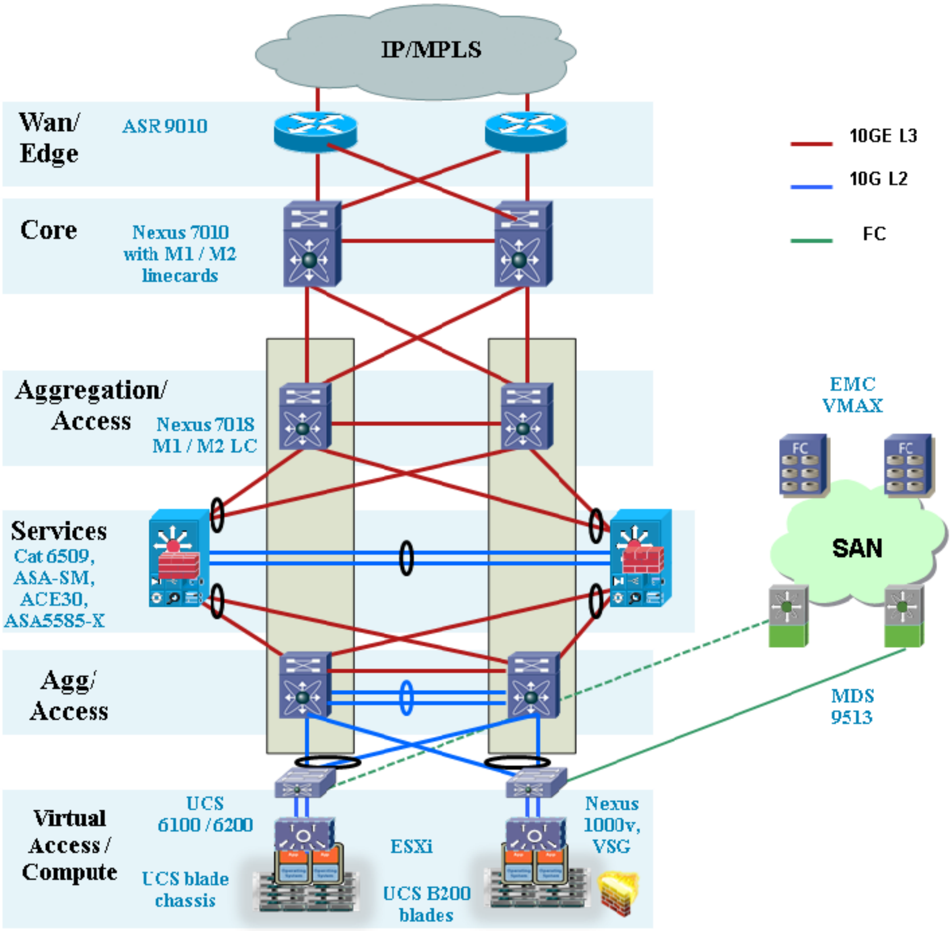
Both the VMDC 2.2 and 3.0 systems utilize a hierarchical multi-tenant DC architecture based on VRF-Lite, with VRF instances and VLANs to provide secure separation between tenants. Besides scalability, platform, and tenancy model differences, the VMDC 2.2 and 3.0 systems also differ in the Layer 2 (L2) technologies utilized within the PoD to provide redundancy and multi-pathing capabilities.

VMDC 2.2

The VMDC 2.2 architecture utilizes a Virtual Port-Channel (vPC) on the Nexus 7000 and Nexus 5000 switches to provide link and chassis redundancy capabilities. Downstream switches (like the UCS 6100/6200 Fabric Interconnect and the Catalyst 6500 DSN) dual connect to a pair of Nexus 7000 aggregation switches, and the individual cross links across the chassis are bundled into a vPC link. The vPC across the chassis protects against any individual link or chassis failures and also provides L2 multi-pathing across the link members to provide higher aggregated bandwidths. In this design, the Nexus 7000 is utilized as the aggregation switch, while the Nexus 5000 and UCS 6100/6200 act as access switches. Only M1 (or M2) linecards are needed on the Nexus 7000 switches in this design.

This multi-layered VMDC architecture is comprised of core, aggregation, services, and access layers. This architecture allows for DC modules to be added as demand and load increases. It also provides the flexibility to create different logical topologies utilizing device virtualization, the insertion of service devices, and traditional Layer 3 (L3) and L2 network configurations. Figure 2-4 provides a logical representation of the VMDC 2.2 architecture, with the services layer comprised of the Catalyst 6500 DSN, ACE30, and ASASM (or ASA 5585-X).

Figure 2-4. VMDC 2.2 System Architecture



VMDC 3.0

The VMDC 3.0 design introduces FabricPath into the VMDC system architecture. Instead of using a vPC, the VMDC 2.0 architecture utilizes FabricPath on the Nexus 7000 and Nexus 5000 switches to provide link and chassis redundancy. FabricPath uses Intermediate System to Intermediate System (IS-IS) as the underlying control plane for MAC learning, and also provides much higher link capacity utilization through 16x equal cost multi-pathing (ECMP). FabricPath provides a larger, flatter L2 domain, with the capability for "Any VLAN Anywhere" across the DC. FabricPath can be used to extend the server VLANs within the PoD, or across PoDs in the DC. In this design, the Nexus 5000 (and/or Nexus 7000) switches are used as FabricPath Leaf (Access) nodes, while Nexus 7000 switches are used as FabricPath Spine (Aggregation) nodes in the FabricPath domain. F1 (or F2) linecards are used on the Nexus 7000 switches for FabricPath downstream L2 connectivity, while M1 (or M2) linecards are utilized on the Nexus 7000 for upstream L3 connectivity.

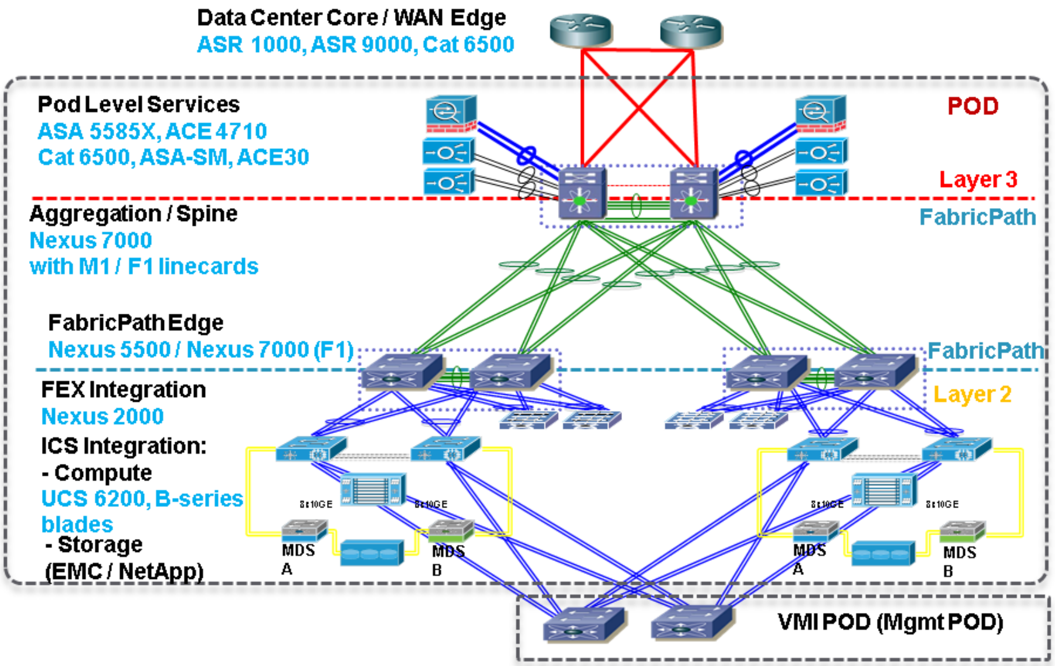
Cisco FabricPath provides the following benefits to the VMDC 3.0 solution:

- Replaces Spanning Tree with a mature link state protocol (IS-IS)
- Single control protocol used for unicast/multicast forwarding, and VLAN pruning
- Expansion of the L2 domain - Any VLAN Anywhere (within PoD and across PoDs)
- Improved link capacity usage through 16-way ECMP

- Improved convergence time
- Easy expansion - add additional access or spine nodes in plug-n-play manner

Figure 2-5 provides a logical representation of the VMDC 3.0 typical DC architecture with FabricPath utilized within the PoD, and the services layer comprised of the ACE 4710 and ASA 5585 appliances (or Catalyst 6500 DSN, ACE30, and ASASM).

Figure 2-5. VMDC 3.0 System Architecture



CHAPTER 3

CLSA-VMDC System Architecture

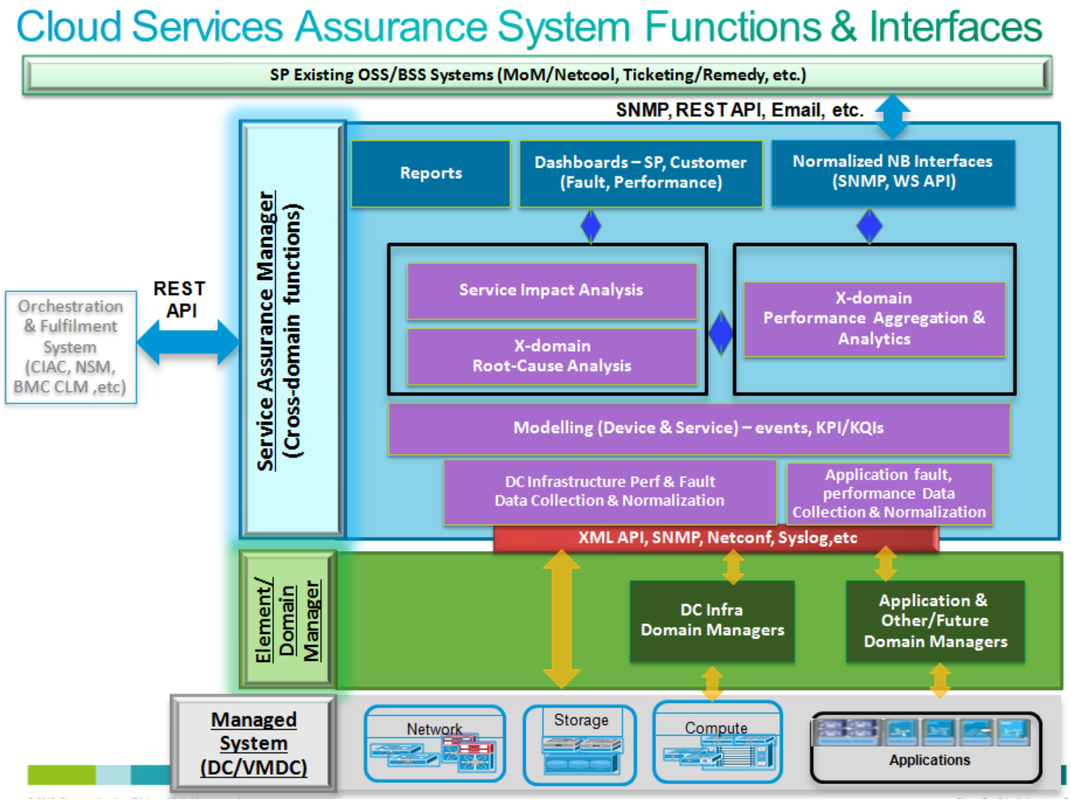
This chapter provides an overview of the Cloud Service Assurance for VMDC (CLSA-VMDC) system architecture.

- Section 3.1 Functional View and Section 3.2 Component View provide the functional and component views of the CLSA-VMDC system architecture.
- Section 3.3 System Components defines the components and interfaces used to deliver the system functions.
- Section 3.4 Monitored Components and Services lists the VMDC devices that are monitored by CLSA-VMDC.
- Section 3.5 Key Functions defines the functions of the new architecture.

3.1 Functional View

Figure 3-1 illustrates the functional framework for CLSA-VMDC. This functionality is delivered with one or more of the integrated products/components. In CLSA-VMDC, only a subset of this functionality is available. This section defines the functional layers of this architecture and identifies the layers that are available in CLSA-VMDC.

Figure 3-1. Functional View of CLSA-VMDC Architecture


Cloud Services Assurance System Functions & Interfaces
SP Existing OSS/BSS Systems (MoM/Netcool, Ticketing/Remedy, etc.)
SNMP, REST API, Email, etc.
Service Assurance Manager (Cross-domain functions)
Reports
Dashboards – SP, Customer (Fault, Performance)
Normalized NB Interfaces (SNMP, WS API)
Service Impact Analysis
X-domain Root-Cause Analysis
X-domain Performance Aggregation & Analytics
Modelling (Device & Service) – events, KPI/KQIs
DC Infrastructure Perf & Fault Data Collection & Normalization
Application fault, performance Data Collection & Normalization
XML API, SNMP, Netconf, Syslog,etc
Orchestration & Fulfilment System (CIAC, NSM, BMC CLM ,etc)
REST API
Element/ Domain Manager
DC Infra Domain Managers
Application & Other/Future Domain Managers
Managed System (DC/VMDC)
Network
Storage
Compute
Applications

The **Managed Device Layer** consists of Data Center (DC) infrastructure including compute, storage, and network components with instrumentation for inventory, fault, and performance data collection. The instrumentation used in this system includes Simple Network Management Protocol (SNMP), syslog, XML Application Programming Interface (API), NETCONF, vSphere API, and so on. Details of interfaces used per VMDC component are included in Section 3.4 Monitored Components and Services.

The **Domain/Element Management Layer** includes the UCS Manager (UCSM) and vCenter. They provide intra-domain inventory, fault, and performance monitoring for UCS and VMware hosts and VMs. These domain managers offer northbound interfaces APIs as well as SNMP and syslog interfaces. CLSA-VMDC utilizes UCS XML API and vSphere API interfaces. CLSA-VMDC 3.0 also introduces the Storage Management Initiative Specification (SMI-S) Provider domain manager to incorporate EMC VMAX and VNX inventory, fault, and performance monitoring.

The **Service Assurance Manager (SAM) Layer** provides all inter-domain functions and a single pane of glass to monitor all VMDC domains including compute, storage, and network. The high-level functions of each of the SAM layers are as follows:

- **Data Collection Layer.** The collection layer leverages domain managers, third-party tools, and so on to obtain performance, availability, and event data for the end-to-end multi-domain system via a range of open protocols such as SNMP, SSL, WMI, and so on. The collection layer is responsible for normalizing this data into a consistent format and persisting data. Collected data includes inventory, fault, and performance type of information.
- **Modeling Layer.** The modeling layer performs discovery, classification, and modeling to determine component dependencies and service dependency graphs. Both performance and fault data should be included in device and service models.

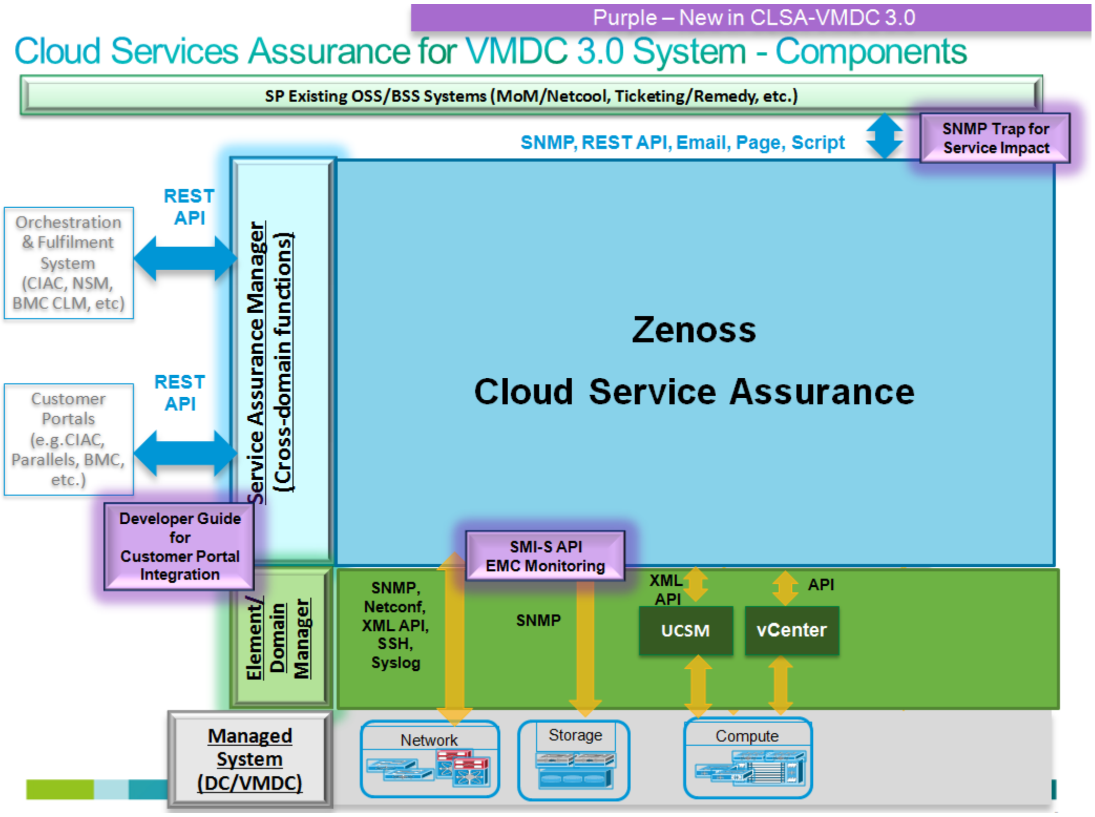
- **Service Model-based Technology.** CLSA-VMDC uses service model-based technology which is described in more detail in Section 3.5.4 Root Cause Analysis and Service Impact Analysis and Section 4 Zenoss Cloud Service Assurance Overview.
- **Root Cause Analysis (RCA).** Leverages the dependency graph or analytics algorithms to determine which events are the probable root cause of the problem and which ones are just consequences that create noise. Therefore, RCA reduces Mean Time to Repair (MTTR). There are a number of different approaches to RCA, but most of them can be classified in one of the following technologies:
  1. Event correlation rules-based
  2. Topology and service model-based
  3. Analytics based
- **Service-Impact Analysis (SIA).** Leverages the dependency graph or analytics algorithms and collects fault and performance data to do the following:
  - Answer who is impacted by the failures
  - Prioritize urgency of failure tickets based on business relevance
  - Determine whether redundancy protected the service
  - Identify failure impacted customers/tenants
  - Prevent future failures by identifying potential service impacting technical risks before they impact service
  - Provide data for SLA measurements and reporting
- **Performance Aggregation Layer.** This layer aggregates performance data from multiple domains (e.g, storage, network, compute for VMDC), normalizes it in the same format and units, provides threshold crossing alerts to the fault management part of the SAM, trends the data over time, and in some cases, performs additional analysis of the data.
- **Presentation Layer.** The presentation layer provides a single view to do both fault and performance monitoring for the entire system. Presentation is done both via dashboards and reports. CLSA-VMDC includes SP dashboards for both fault and performance.
- **Northbound Interface.** The Northbound Interface (NBI) is a special form of the presentation layer where normalized and enriched data is presented to northbound OSS/BSS systems via open interfaces such as WS API, SNMP, and email.

3.2 Component View

Section 3.1 Functional View defines the functions of the CLSA-VMDC architecture. This section defines the components used to deliver those functions, as well as their interfaces. The key component of the architecture for CLSA-VMDC is Zenoss Cloud Service Assurance (CSA), which plays the role of the SAM. In addition, several domain managers are utilized - UCS Manager (UCSM) for UCS hardware monitoring, VMware vCenter for monitoring the virtualized infrastructure, and SMI-S Provider for EMC VMAX and VNX monitoring.

Figure 3-2 illustrates the components and interfaces used to deliver the functional layers of the CLSA-VMDC architecture.

Figure 3-2. Component View of CLSA-VMDC Architecture


Purple – New in CLSA-VMDC 3.0
Cloud Services Assurance for VMDC 3.0 System - Components
SP Existing OSS/BSS Systems (MoM/Netcool, Ticketing/Remedy, etc.)
SNMP, REST API, Email, Page, Script
SNMP Trap for Service Impact
REST API
Orchestration & Fulfilment System (CIAC, NSM, BMC CLM, etc)
REST API
Customer Portals (e.g.CIAC, Parallels, BMC, etc.)
Developer Guide for Customer Portal Integration
Service Assurance Manager (Cross-domain functions)
Zenoss
Cloud Service Assurance
SMI-S API EMC Monitoring
Element/ Domain Manager
SNMP, Netconf, XML API, SSH, Syslog
SNMP
XML API
API
UCSM
vCenter
Managed System (DC/VMDC)
Network
Storage
Compute

Key system interfaces include:

- Southbound interface instrumentation to collect data from managed system devices.
- Northbound interface to integrate with OSS/BSS systems such Manager-of-Managers (MoM) (e.g., IBM Netcool), ticketing systems (e.g., Remedy) and so on. The interfaces available from CLSA-VMDC are SNMP, JSON API, email, page, commands, and Advanced Message Queuing Protocol (AMQP).
- CLSA-VMDC offers the JSON API interface for integration with orchestration and fulfillment systems.

3.3 System Components

Table 3-1 lists the Cisco and third-party components used in CLSA-VMDC.

Table 3-1. Cisco and Third-Party Components Used in CLSA-VMDC

Vendor	Model	Description
Zenoss	Resource Manager 4.2.3	Zenoss CSA software module that performs resource discovery, monitoring, and modeling.

Vendor	Model	Description
Zenoss	Impact 4.2.3	Zenoss CSA software module that performs service impact discovery and analysis.
Zenoss	Analytics 4.2.3	Zenoss CSA software module that performs long term data trending, processing, and reporting.
vCenter	vCenter 5.0	Domain manager for VMware based virtualization
Cisco	UCSM 2.0	Domain manager for UCS platform
EMC	SMI-S Provider 4.4.0.1	Domain manager for EMC VMAX and VNX platforms

Note The Zenoss software modules are packaged together as Zenoss CSA 4.2.3.

3.4 Monitored Components and Services

Table 3-2 lists the VMDC 3.0 devices that are monitored by the CLSA-VMDC system out-of-the-box and the instrumentation (interfaces) utilized by Zenoss CSA to collect data.

Table 3-2. VMDC 3.0 Components Monitored by CLSA-VMDC

Managed Component	Interfaces Utilized in CLSA-VMDC 3.0
Compute Components	
UCS 5108; B-series blades	ICMP, UCSM XML API
UCS 6100, 6200	ICMP, UCSM XML API
VMware ESX and ESXi Hypervisors	ICMP, vSphere API
VMware Virtual Machines	ICMP, vSphere API
Storage Components	
MDS 9000	ICMP, SNMP
EMC VMAX [1]	ICMP, SMI-S API
EMC VNX [1]	ICMP, SMI-S API
FAS6080, FAS3000	ICMP, SNMP, SSH
Network Components	
UCS 6100, 6200	ICMP, UCSM XML API
Nexus 7000 (e.g., 7018, 7010, 7009 including M1 and F1 cards) [2]	ICMP, NETCONF, SNMP
Nexus 5000 (e.g., 5548, 5596, and 5020)	ICMP, NETCONF, SNMP

Managed Component	**Interfaces Utilized in CLSA-VMDC 3.0**
Nexus 3000 [1]	ICMP, NETCONF, SNMP
Nexus 2000 (e.g., 2248 and 2232)	ICMP, NETCONF, SNMP
Nexus 1000V / Nexus 1010 [1]	ICMP, NETCONF, SNMP
ASR 9000	ICMP, SNMP, SSH
ASR 1000	ICMP, SNMP
Network Services Components	
Catalyst 6500 VSS	ICMP, SNMP, SSH
ACE (e.g., ACE20, ACE30, ACE4700)	ICMP, SNMP, ACE XML API
FWSM	ICMP, SNMP
ASASM [1]	ICMP, SNMP
ASA 5580-40	ICMP, SNMP
ASA 5585-40	ICMP, SNMP
Virtual Security Gateway	ICMP, SNMP, NETCONF, SSH

Note

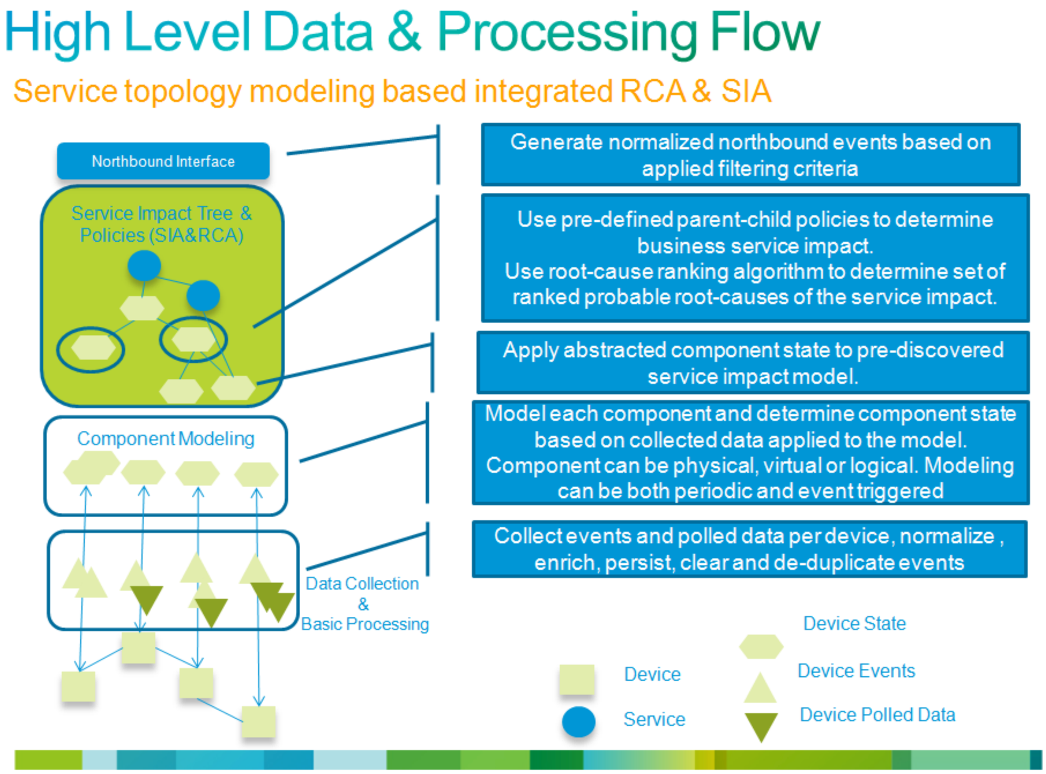
- [1] Denotes new enhancement for CLSA-VMDC 3.0.
- [2] FabricPath F1 cards are added to Nexus 7000 devices.

3.5 Key Functions

This section describes the key functions of CLSA-VMDC.

In the overall lifecycle of assurance services, the first task that has to be completed is enablement of service assurance services. Section 1.3.1 Automate Service Enablement provides details about enabling service assurance, including provisioning and automatic discovery. Once assurance services are enabled, they can be used for Day 2 operations. Figure 3-3 illustrates and explains the high-level, end-to-end data flow through the fault and problem management part of CLSA-VMDC.

Figure 3-3. End-to-End Fault and Problem Management Data and Processing Flow



High Level Data & Processing Flow

Service topology modeling based integrated RCA & SIA

Generate normalized northbound events based on applied filtering criteria

Use pre-defined parent-child policies to determine business service impact.
Use root-cause ranking algorithm to determine set of ranked probable root-causes of the service impact.

Apply abstracted component state to pre-discovered service impact model.

Model each component and determine component state based on collected data applied to the model. Component can be physical, virtual or logical. Modeling can be both periodic and event triggered

Collect events and polled data per device, normalize, enrich, persist, clear and de-duplicate events

The following sections discuss each of the stages and functions in this sample data flow:

- Data collection and device modeling
- Basic event processing
- RCA and SIA
- Northbound interface

This section also discusses the following additional functions related to the overall platform and its use:

- Performance Management
- Dashboards
- Reporting
- Multiservices

3.5.1 Automatic Enablement of Service Assurance

Automatic enablement of service assurance can be achieved in a couple of different ways. Fundamentally, the following are approaches that can be taken to automate service enablement and life cycle:

1. Reduce necessary amount of configuration (by using technology that is self learning (e.g., self learning thresholds))
2. Automatic discovery (by assurance system)

3. Programmatic orchestrated provisioning (via integration with orchestration system)

CLSA-VMDC focuses on automatic discovery. CLSA-VMDC also provide APIs for programmatic orchestrated provisioning, but they are not integrated or validated with any particular orchestration system. Automatic discovery and APIs are discussed in the following sections.

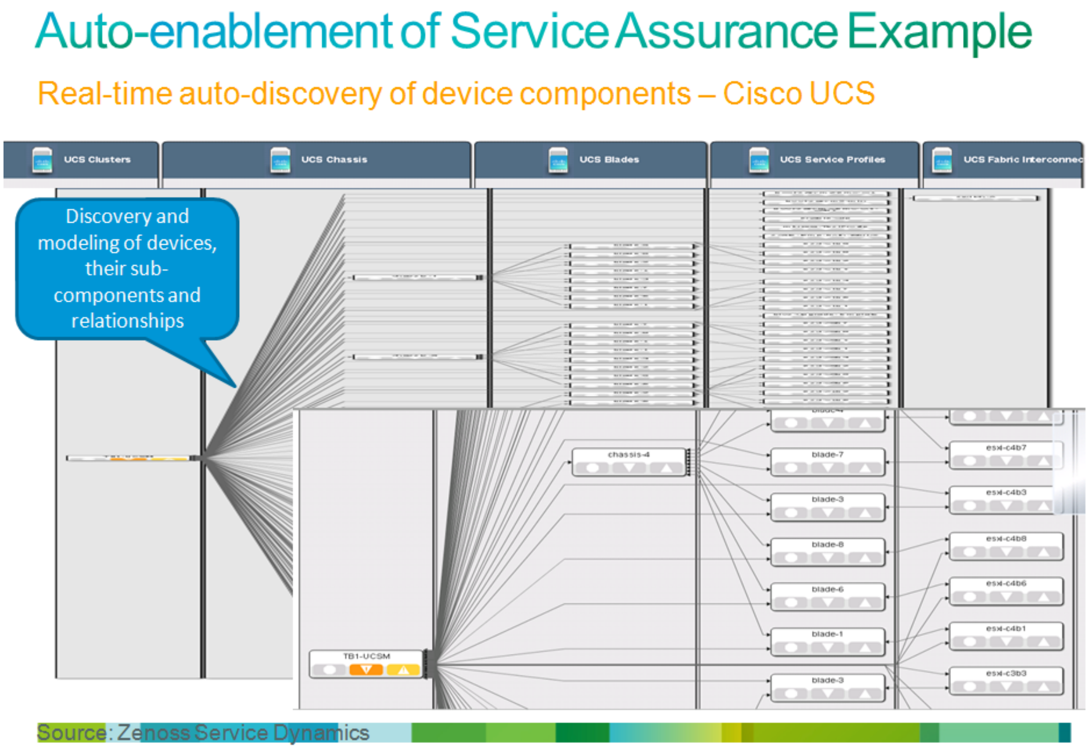
3.5.1.1 Automatic Discovery

The following types of objects are automatically discovered in CLSA-VMDC:

- Monitored devices (e.g., UCS, Nexus 7000, MDS 9000, etc.)
- Sub-components of devices and their relationships (e.g., UCS chassis, blades, fabric interconnect, etc.)
- Tenant-based Service Impact Analysis (SIA) model for the compute (e.g., tenant Virtual Machine (VM) mapping to service impacting resources, both dedicated and shared vCenter and UCSM managed resources). The exception is tenant name and its link to the service, which cannot be discovered, but relies on orchestrated provisioning. In this release, tenant name and mapping to the VM are provisioned manually, but the API is provided.

Figure 3-4 and Figure 3-5 illustrate examples of automatic enablement of service assurance.

Figure 3-4. Real-time Automatic Discovery of Device Components - Cisco UCS



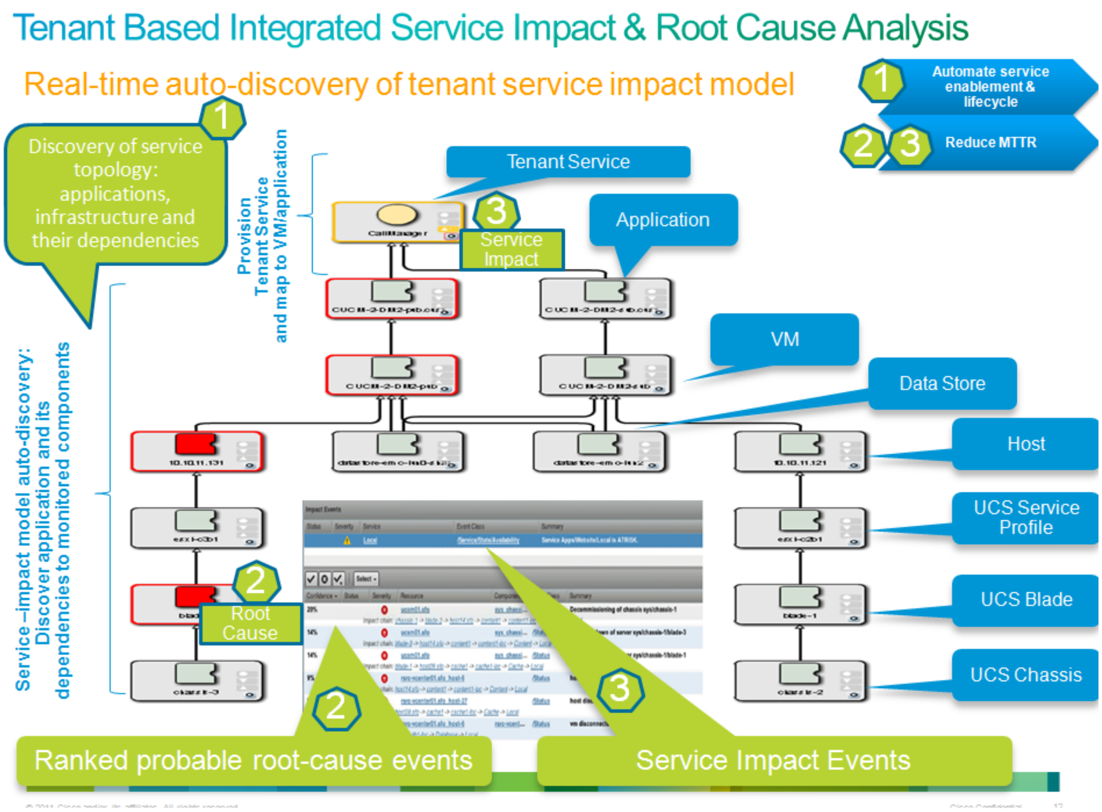
Auto-enablement of Service Assurance Example

Real-time auto-discovery of device components – Cisco UCS

Discovery and modeling of devices, their sub-components and relationships

Source: Zenoss Service Dynamics

Figure 3-5. Real-time Automatic Discovery of Tenant Service Impact Model


Tenant Based Integrated Service Impact & Root Cause Analysis
Real-time auto-discovery of tenant service impact model
1 Automate service enablement & lifecycle
2 3 Reduce MTTR
1 Discovery of service topology: applications, infrastructure and their dependencies
Provision Tenant Service and map to VM/application
Tenant Service
3 Service Impact
Application
VM
Data Store
Host
UCS Service Profile
UCS Blade
UCS Chassis
2 Root Cause
Service –impact model auto-discovery: Discover application and its dependencies to monitored components
2 Ranked probable root-cause events
3 Service Impact Events

3.5.1.2 Zenoss APIs for Programmatic Provisioning

CLSA-VMDC offers APIs to programmatically provision the following components in the service impact tree:

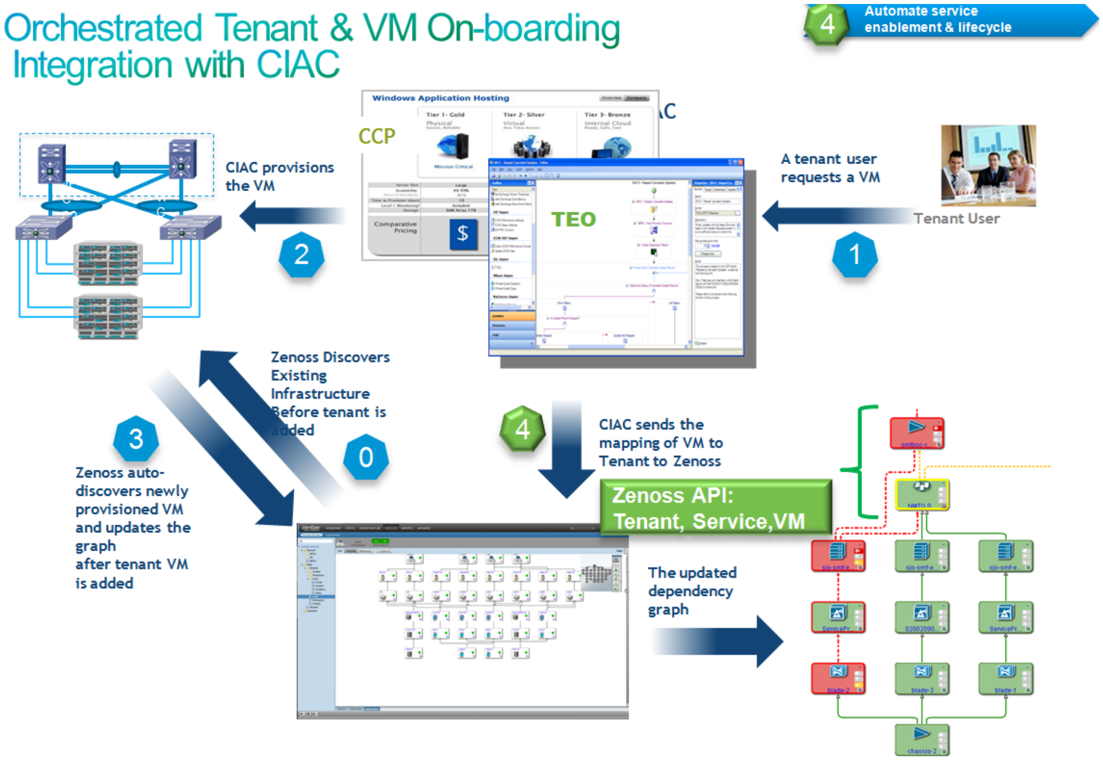
- Tenant Name
- Tenant ID
- Service Name
- Service ID
- VM Name
- VM ID

This enables automatic onboarding of the tenant and tenant compute service, which maps them to the already automatically discovered VM and its relationships to shared hardware.

Note Proof of Concept (PoC) of this functionality integrated with the Cisco Intelligent Automation for Cloud (CIAC) orchestration stack has been performed by Cisco Advanced Services; however, it was not validated as part of the CLSA-VMDC system. If this functionality is desired in the field before it is included as part of the Systems Development Unit (SDU) system release, then Cisco Advanced Services can perform integration with the desired orchestration stack using the provided API.

Figure 3-6 illustrates the high-level workflow that provisions the tenant and tenant service and then maps the workflow to the automatically discovered VM and the rest of the automatically discovered infrastructure.

Figure 3-6. Zenoss Tenant Provisioning Using CIAC Orchestration


Orchestrated Tenant & VM On-boarding
Integration with CIAC
4 Automate service enablement & lifecycle
CCP
Windows Application Hosting
TEO
A tenant user requests a VM
Tenant User
1
CIAC provisions the VM
2
Zenoss Discovers Existing Infrastructure Before tenant is added
0
3
Zenoss auto-discovers newly provisioned VM and updates the graph after tenant VM is added
4
CIAC sends the mapping of VM to Tenant to Zenoss
Zenoss API: Tenant, Service,VM
The updated dependency graph

3.5.2 Fault Performance, Configuration Data Collection, and Device Modeling

Consolidated monitoring and data collection at the SAM layer is one of the fundamental functions of CLSA-VMDC. Consolidated monitoring and data collection is characterized by the following attributes:

- **Various domains (applications, compute, storage, network).** The cloud assurance system needs to provide a single pane of glass to monitor components from various domains.
- **Fault and performance data.** The cloud assurance system needs to consolidate fault and performance data and leverage both for all of its higher order functions like RCA and SIA.
- **Various data sources, interfaces, and protocols.** The cloud assurance system needs to collect data from multiple data sources and protocols and consolidate this data in unified device and service models. Some examples of different data sources and protocols are SNMP, syslog, WS API, Netflow, customer opened tickets, and so on.

Zenoss Data Collection

Zenoss CSA offers consolidated monitoring for VMDC, including consolidation of domains (i.e., support for OS, compute, storage, and network), consolidation of performance and fault data (i.e., takes into consideration both polled performance data, asynchronous events it receives, as well as synthetic events it generates for both performance and availability), and consolidation of data sources (i.e., device monitoring models utilize multiple data sources such as SNMP, syslog, API, and consolidate it within unified device model).

Zenoss CSA uses an agentless data collection approach, which is critical for the type of scale expected in cloud systems. Instead of installing an agent on monitored devices, Zenoss supports a rich set of protocols to enable data collection. A list of protocols used for data collection from VMDC devices is included in Section 3.4 Monitored Components and Services. The following is a more comprehensive list of data collection interfaces that the Zenoss CSA platform supports:

Event input:

- SNMP
- Syslog
- XML Remote Procedure Call (RPC)
- JavaScript Object Notation (JSON)/API
- AMQP
- Windows Event Log

Easily configurable protocol usage:

- Secure Shell (SSH)
- Java Management Extensions (JMX)
- Windows Management Instrumentation (WMI)
- Perfmon
- Any script that returns data in a known format (such as Nagios)

Other collection mechanisms (model/performance/event data):

- Internet Control Message Protocol (ICMP)
- Telnet
- JMX
- Hypertext Transfer Protocol (HTTP) - Web Transactions
- Oracle
- Structured Query Language (SQL) Server
- MySQL
- Apache (mod_status)
- memcache
- Splunk Queries
- Simple Mail Transfer Protocol (SMTP)
- Post Office Protocol (POP)
- UCSM XML API
- vSphere Simple Object Access Protocol (SOAP) API
- vCloud Director
- Amazon EC2 and CloudWatch
- Cisco CallManager (AXL)

- Domain Name System (DNS)
- Lightweight Directory Access Protocol (LDAP)
- Network Time Protocol (NTP)
- File Transfer Protocol (FTP)
- Internet Relay Chat (IRC)
- Extensible Messaging and Presence Protocol (XMPP)
- Remote Procedure Call (RPC)
- Network News Transfer Protocol (NNTP)

Zenoss Device Modeling

Device modeling in Zenoss goes beyond traditional device discovery; it also uses standard Management Information Bases (MIBs) to discover interesting aspects of the device and automatically defines models for that device type. Once modeled, these learned attributes can be inherited as part of the model when a new device of the same type is discovered again. The information below describes various attributes of the Zenoss device modeling process.

Initial Zenoss Model (plugins):

- Interfaces to access device and objects of interest (KPI statistics, events, thresholds, etc.) are statically defined
- Models are assigned to a device class

Device Modeling:

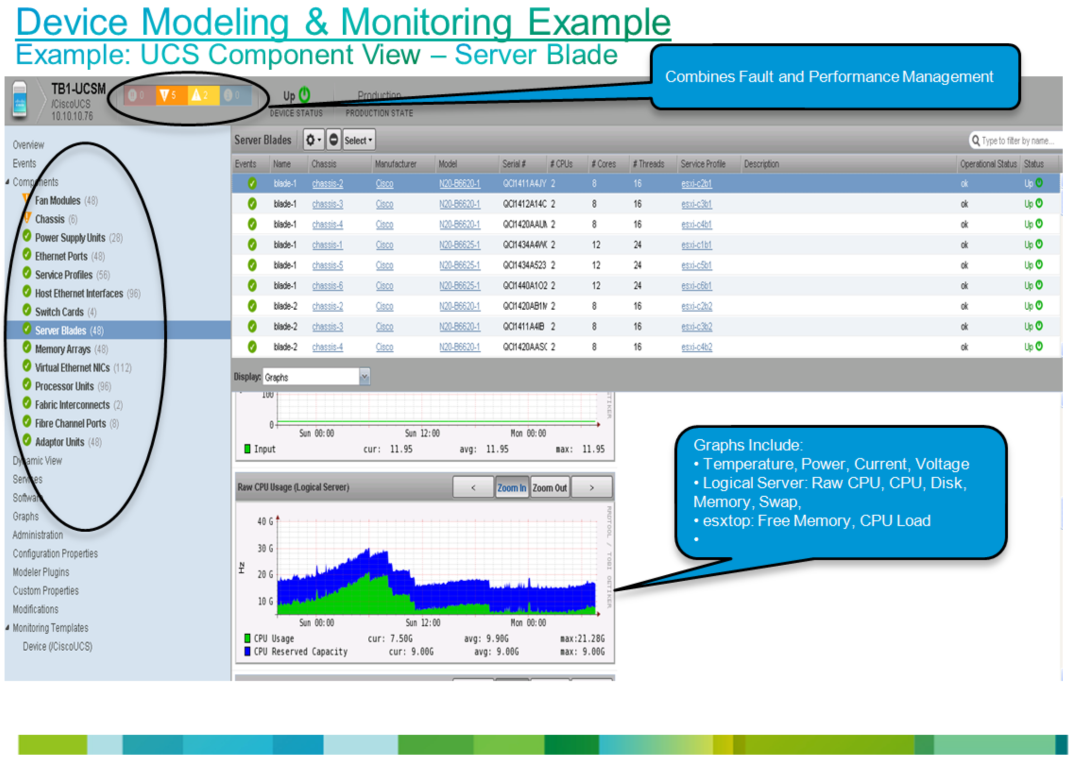
- During individual device discovery, all modeler plug-ins for the device class are automatically considered, and a model per instance of the device is created.
- After discovery modeling, monitoring and event processing automatically starts.

Device Remodeling:

- Model per device instance can dynamically change in response to events (e.g., blade removed, etc.)
- ZenModelerDeamon - per collector configuration happens every 12 hours
- ZenVMwareDeamon (exception for VMware and remodels every 4 hours)
- List of events that trigger remodeling is configurable (default set exists)

An example of unified monitoring using Zenoss CSA is illustrated in Figure 3-7.

Figure 3-7. Unified Monitoring Using Zenoss CSA



Device Modeling & Monitoring Example

Example: UCS Component View – Server Blade

Combines Fault and Performance Management

Events	Name	Chassis	Manufacturer	Model	Serial #	# CPUs	# Cores	# Threads	Service Profile	Description	Operational Status	Status
✓	blade-1	chassis-2	Cisco	N20-B6620-1	QCI1411A4JY	2	8	16	esxi-c2b1		ok	Up
✓	blade-1	chassis-3	Cisco	N20-B6620-1	QCI1412A14C	2	8	16	esxi-c3b1		ok	Up
✓	blade-1	chassis-4	Cisco	N20-B6620-1	QCI1420AAUN	2	8	16	esxi-c4b1		ok	Up
✓	blade-1	chassis-1	Cisco	N20-B6625-1	QCI1434A4WK	2	12	24	esxi-c1b1		ok	Up
✓	blade-1	chassis-5	Cisco	N20-B6625-1	QCI1434A523	2	12	24	esxi-c5b1		ok	Up
✓	blade-1	chassis-6	Cisco	N20-B6625-1	QCI1440A1O2	2	12	24	esxi-c6b1		ok	Up
✓	blade-2	chassis-2	Cisco	N20-B6620-1	QCI1420AB1W	2	8	16	esxi-c2b2		ok	Up
✓	blade-2	chassis-3	Cisco	N20-B6620-1	QCI1411A4IB	2	8	16	esxi-c3b2		ok	Up
✓	blade-2	chassis-4	Cisco	N20-B6620-1	QCI1420AASC	2	8	16	esxi-c4b2		ok	Up

Display: Graphs
100
0
Sun 00:00
Sun 12:00
Mon 00:00
Input cur: 11.95 avg: 11.95 max: 11.95
Raw CPU Usage (Logical Server)
<
Zoom In
Zoom Out
>
40 G
30 G
20 G
10 G
Hz
Sun 00:00
Sun 12:00
Mon 00:00
CPU Usage cur: 7.50G avg: 9.90G max:21.28G
CPU Reserved Capacity cur: 9.00G avg: 9.00G max: 9.00G

Graphs Include:
- Temperature, Power, Current, Voltage
- Logical Server: Raw CPU, CPU, Disk, Memory, Swap,
- esxtop: Free Memory, CPU Load
-

3.5.3 Event Processing

In CLSA-VMDC, event processing is divided into two categories:

- Basic event processing
- Event processing that is part of RCA and SIA

This section only describes basic event processing functions, while RCA and SIA are discussed in the following sections. The basic event processing functions included in this system are event classification, normalization, de-duplication, enrichment, persistence, and clearing.

Event classification groups similar events in event classes, so that some of the more complex processing may be simplified by looking at event classes rather than each individual event.

Event normalization translates various formats of the raw collected data into a single format that is used by the SAM. Often, the same format or subset of the fields of normalized format can be sent to northbound systems. This function allows simplified integration of northbound systems since they have to deal with a single event format for multiple device types and instrumentation protocols.

Event de-duplication eliminates multiple events that have the exact same content with the exception of the time stamp. After de-duplication, a single event is kept, and typically a counter indicating the number of occurrences of the event is added, as well as a timestamp indicating the first and last occurrence of the duplicate event.

Event persistence archives all events to be used for forensic analysis. In some systems, persistence exists only on post-processed events, while in others, for raw events as well.

Event clearing is used to indicate when the original condition for which the event was raised is removed. Explicit event clearing is done by generating clearing events with the field within the clearing event, which points to the ID of the event that it is clearing. For example, if an interface down event for a specific interface had an ID of ID1, when the interface goes up again, an event with ID2 should be raised, which includes as one of its fields a reference to event ID1. Explicit event clearing is recommended. In addition to explicit clearing, time-based clearing can be utilized as well. Time-based clearing clears the event after a specific time interval elapses from the time that the original event was received.

3.5.4 Root Cause Analysis and Service Impact Analysis

One of the key functions of CLSA-VMDC is Root Cause Analysis (RCA) and tenant-based Service Impact Analysis (SIA).

The objective of RCA is to reduce MTTR by determining which events are probable root causes of the problem and which events are just consequences that create noise.

The following are the objectives of tenant-based SIA:

- To prioritize the urgency of failure tickets based on business relevance
- To determine whether redundancy protected the service
- To identify failure impacted customers/tenants
- To prevent future failures by identifying potential service impacting technical risks before they impact service
- To enable Service Level Agreement (SLA) measurements and reporting

3.5.4.1 Zenoss SIA and RCA

Zenoss CSA uses model-based SIA, which produces a set of ranked probable root causes as a by-product of SIA. This service impact-based approach to RCA is a fundamentally different approach from legacy rule-based systems:

- **Bottom-up.** What services are impacted by conditions below (Zenoss) vs.
- **Top-down.** What is the cause of problem at service level (legacy products)

Zenoss does not determine a single root cause, but instead identifies multiple related events (probable root cause events) and presents the following:

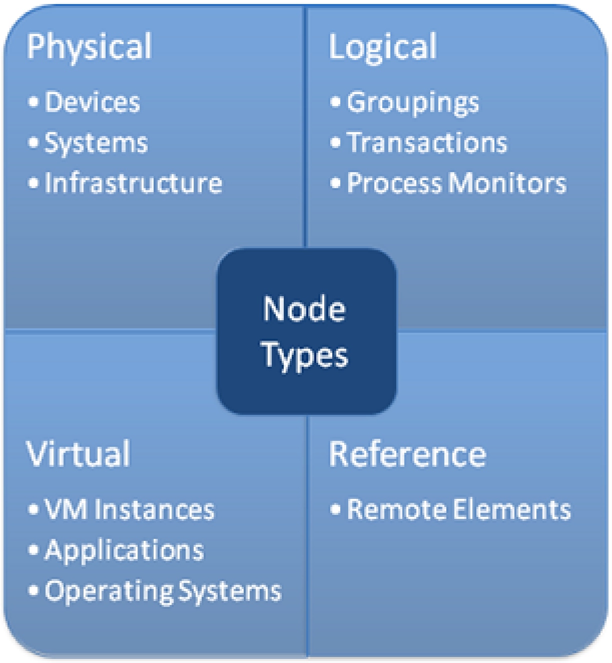
- A root cause ranking algorithm is utilized to rank probable root cause events in order of confidence that the event is the actual root cause event. This algorithm ranks impact events based on a variety of criteria, including the severity of the event, service graph depth, and the number of graph branches affected by an event.
- Hierarchical service dependency graphs provide a visual indication of probable root causes leading to a service impact.

Events flow through the graph referencing molecular node policies to determine whether they should be passed, filtered, aggregated, or masked. There are a few key elements of RCA and SIA in Zenoss CSA. Each assurance service within Zenoss is modeled with a service impact tree that consists of a set of nodes, policies applied to the nodes, and the relationships between the nodes:

- The service can be arbitrarily defined and can be a very abstract service that consists of other sub-services, or on other extreme, one can even define a single physical interface as a service. This provides a very flexible framework for service definition.

- Model nodes represent arbitrary components such as physical, logical, or virtual resource. For example, nodes can represent an end-to-end service such as voice, a virtual resource such as a VM, or a physical resource such as a chassis or physical interface. The following four types of nodes are currently supported, as illustrated in Figure 3-8:
  - **Physical.** Systems, infrastructure, and network devices that a service relies on.
  - **Virtual.** Software components that make up a service.
  - **Logical.** Aspects of a service that must be measured or evaluated as a set to determine state (facilitates extension of an impact graph by providing a hook to incorporate arbitrary events into impact analysis).
  - **Reference (future release).** Provide a link to dependencies managed by an external instance of Zenoss or other management system capable of propagating state information to Zenoss.

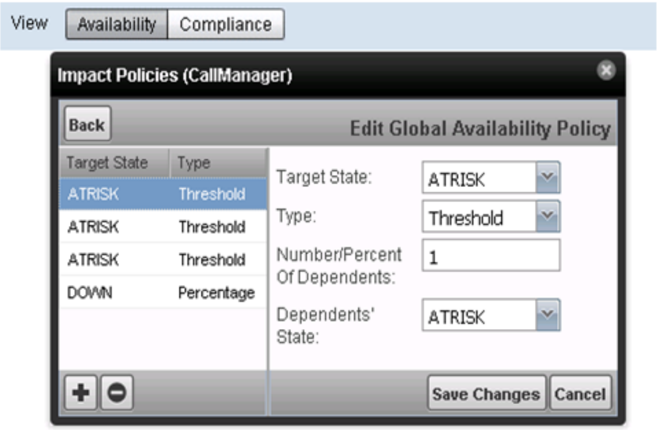
Figure 3-8. Node Types



- Policy is defined per node, which allows it to move as the resources move, which is a critical characteristic for the cloud environment. Zenoss refers to this policy as a molecular policy since it is defined per node. Zenoss utilizes a very simple policy that can define the state of the node solely as a function of the state of its children nodes, which allows for service impact "rules" decoupling from device events resulting in the following:
  - "Rules" defined in a single place for any given device or service: device events processing in event processing software modules, service impact processing in service impact graphs (i.e., device events do not need to be considered in service level rules)
  - Simplified development and maintenance of cross-domain service impact and RCA customizations: do not have to correlate device events from multiple devices to determine cross-domain service impact and possible root causes

- Note that whenever desired, device events can be used as part of service impact "rules" via use of logical nodes whose rules define how to interpret the service impact of specific events based on its type and severity.
- Policy can be global or contextual:
  - Global policy applies to device/service type in any service graph.
  - Contextual policy applies only to device/service in the particular service graph.
- Each node has a default policy applied, which reduces the need for custom configuration. The default policy is often sufficient, but can be modified where required via GUI or API. Figure 3-9 illustrates a sample node policy.

Figure 3-9. Sample Node Policy


View
Availability
Compliance
Impact Policies (CallManager)
Back
Edit Global Availability Policy
Save Changes
Cancel

Target State	Type
ATRISK	Threshold
ATRISK	Threshold
ATRISK	Threshold
DOWN	Percentage

Target State: ATRISK
Type: Threshold
Number/Percent Of Dependents: 1
Dependents' State: ATRISK

- Each node has a state which is determined using node policy. Currently, there are four service and node availability states supported (UP, DOWN, DEGRADED, AT RISK), and two service performance states supported (IN/OUT). Software development is required to increase the number of states.

Creation of the service model in Zenoss CSA can be done in the following ways:

- Automatically discover service models predefined by Zenoss. Examples are service models for OS, VMware, and UCS domains.
- A GUI service model can be modified or created via a simple GUI approach.
- REST APIs allow for programmatic provisioning of the service models.

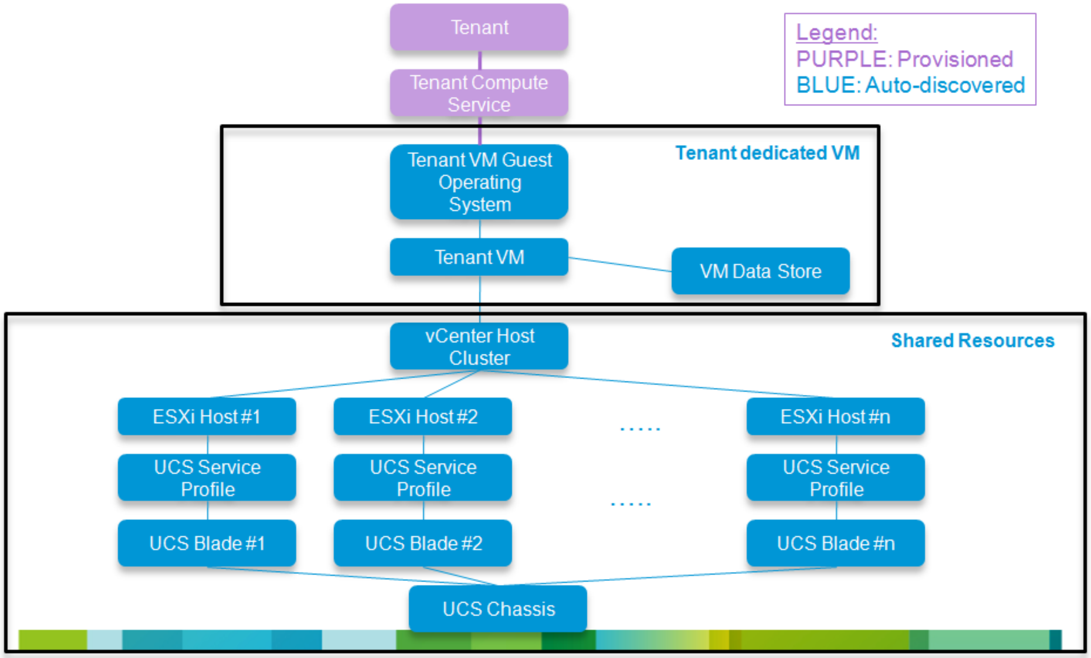
3.5.4.2 VMDC Assurance Service Models

In order to perform SIA, CLSA-VMDC uses service models with polled and asynchronous data to perform SIA and RCA. CLSA-VMDC offers an out-of-the-box tenant service model for compute. In future releases, CLSA-VMDC will expand the library of out-of-the-box service models that will be validated and delivered as part of this system. However, note that users can easily customize service models as well as create new ones.

Tenant Compute Assurance Service

Figure 3-10 defines the out-of-the-box tenant compute service model to be delivered as part of CLSA-VMDC. More details are provided about this service model in Section 6.4 Tenant SIA and RCA.

Figure 3-10. Tenant Compute Assurance Service Model - Generic Application



Service Model Policy

Each node (referred to as the parent node) in the service model has a policy defined that calculates the state of that node based on the state of its children and any explicit events associated with the parent node.

For the particular service model illustrated in Figure 3-10, the specific policies listed in Table 3-3 should be applied.

Table 3-3. Service Model Policy Decisions

Node	Node State	If Child Node State
Tenant Compute Service	UP/DOWN/AT RISK	UP/DOWN/AT RISK
Tenant Guest OS	UP/DOWN/AT RISK	UP/DOWN/AT RISK
Tenant VM	UP/DOWN/AT RISK	UP/DOWN/AT RISK
ESXi Cluster	• UP/DOWN • AT RISK	• All Children UP/DOWN • At Least One Child DOWN/AT RISK
ESXi Host	UP/DOWN/AT RISK	UP/DOWN/AT RISK

Node	Node State	If Child Node State
UCS Blade	UP/DOWN/AT RISK	UP/DOWN/AT RISK

Out-of-the-box, all nodes use the default policy where the worst impact wins. The one exception is the VMware cluster, which is DOWN if all children are DOWN and DEGRADED if any nodes are DOWN or DEGRADED.

In addition to considering the parent/child policy, the explicit state of the nodes is determined by both availability and events for components the node represents. For VMware and UCS nodes, the explicit node impact status is determined mainly by modeled properties. As modeling occurs or various events are received, Zenoss reassesses the impact state by querying the Zenoss model. For example, when a VM power off event is received, the model is updated and the VM status is reassessed and updated.

Service Model Variations

Note that the model defined in this section illustrates a single-tier application with a single VM. Variation of this service model would be models for the following:

- Multi-tier application, where there would be multiple "tenant dedicated VM" blocks tied to the tenant compute service. The tenant compute service default policy may need to be customized.
- Single-tier application that supports application level redundancy via clustering (e.g., Cisco UC applications such as CUCM). In this case, the model would be modified to include multiples of "tenant dedicated VM" blocks. The default policy used for the "tenant compute service" should be applicable. An example of this service model is illustrated in Figure 3-10.

Service Model Enablement

Most of this model is automatically discovered, while the top node of the service model needs to be provisioned. Typically, provisioning would be done in an automated way when the tenant and VM get onboarded. In CLSA-VMDC, there is no integration with the orchestration stack, and as such, the top node of the service model is manually provisioned. Note that in real deployments, per-tenant manual provisioning is not an option, in which case either an available Zenoss API can be used by the orchestration platform of choice, or if not provisioned, the tenant service impact is still possible but results are given in the VM context rather than tenant service context. For example, there would be no automatic mapping between tenant name, tenant service name, and VM ID.

In future CLSA-VMDC releases, integration with VMDC orchestration stacks will be implemented and validated. In addition to automatic discovery of the service model from VM down, if operating systems such as Windows or Linux are deployed, they should also be automatically discovered.

Mobility Handling

The host to VM relationship is given by VMware during modeling stage. Whenever VMware generates an event that indicates VM movement, Zenoss reacts and remodels the source and target hosts to update its model. Depending on the event collection interval specified in the Zenoss configuration, the model change can take anywhere from 15 seconds to 3 minutes. With the out-of-the-box configuration, the average time would be about 1 minute.

Redundancy Implications

A service model with three service states accounts for redundancy. The AT RISK state is used to indicate conditions where the service or service model node is still functioning despite a failure of one of its children because redundancy protected the service. For the particular service model shown in Figure 3-10, redundancy capabilities that are accounted for include the following:

- If one of the blades/hosts fails, and the vCenter cluster that VM belongs to has multiple blades/ hosts, then the VM node is marked AT RISK as opposed to DOWN based on the status of its children. Note that explicit VM related state and events can result in the state of the VM node being down even though the state of its children alone would result in an AT RISK state

- In a case where there is application level redundancy and thus more than one VM and application deployed for single tier applications, there is also service model redundancy built in on the application/VM level. For example, a service is AT RISK if one of the application nodes/VMs is DOWN because the remaining application/VM nodes provides redundancy for the failed application/VM node.

3.5.4.3 VMDC RCA and SIA Use Cases

Once service impact models are defined, the data is applied to service impact models to maintain real-time state of the service availability and performance, as well as to determine probable root cause of any failures that may happen. This section provides a list of failure scenarios (use cases) validated as part of the CLSA-VMDC test effort, for which the out-of-the-box compute service model can determine correct probable root cause and service state for previously defined services. All of the use cases are validated in an environment where VMware High Availability (HA) is deployed.

See Section 6.4 Tenant Service Impact Analysis and Root Cause Analysis for an example workflow illustrating a UCS switch failure event, including screenshots.

Use Case Name (Fault):

- VM Failure
- VM vNIC failure
- VM vMotion - VM vMotion is not a true fault event, since the VM stays up, however, the impact graph does track the VM's host swap.
- ESXi host failure
- UCS Blade failure
- UCS chassis failure
- UCS P/S failure
- UCS FEX failure
- UCS 6100 chassis failure
- UCS 6100 interfaces to UCS 5100 failure
- VM CPU degradation (Threshold Crossing Alert (TCA))
- VM Memory degradation (TCA)
- Host CPU degradation (TCA)
- Host Memory degradation (TCA)

3.5.5 Northbound Interface

One of the key, new functions of CLSA-VMDC architecture is a single, normalized Northbound Interface (NBI) provided by the SAM.

The key objectives of the single, normalized interface are:

- **To simplify and reduce the cost of integrating providers existing northbound system with the CLSA-VMDC system.** The provider needs to integrate and maintain just one interface rather than multiple dozens of interfaces towards individual devices and/or domain managers. CLSA-VMDC is responsible for absorbing updates related to any relevant changes in the underlying system and devices.

- **To enable CLSA-VMDC to be inserted in various business and operational deployment environments.** This is achieved by offering a variety of interface protocols, rich filtering capabilities, and notifications with tenant awareness.
- **To enable CLSA-VMDC to simplify service assurance of overlaid application based systems that are deployed on top of VMDC infrastructure.** An example of this type of system is the Hosted Collaboration Solution (HCS). This is achieved by providing tenant service level notifications rather than device level notifications, which enables a service overlay (or multi-tier SIA) to be implemented by HCS, and as such, Cloud Service Assurance-HCS (CLSA-HCS) would have to deal with the state of only a handful of services coming from CLSA-VMDC, rather than thousands of events coming from individual VMDC devices.

Zenoss northbound integration is supported via:

- JavaScript Object Notation (JSON)/Representational State Transfer Application Programming Interface (ReST API)
- SNMP Traps (ZENOSS-MIB.txt and ZENOSS-IMPACT-MIB.txt)
- Syslog
- Event queues (AMQP and Java/Python wrappers) and event commands (command line call with event context)
- SMTP email

Configurable filtering capabilities are offered to provide different data to different northbound consumers. The following sections describe the interfaces, data, and filtering capabilities in more detail.

3.5.5.1 SNMP Northbound Interface

One of the key requirements for CLSA-VMDC is to offer asynchronous notifications via SNMP. These notifications are consumed either by the provider's existing northbound systems such as MoM, ticketing, and SLA management systems, or by other Cisco systems deployed on VMDC architecture such as HCS.

Regardless of the source or type of the event, all events should be sent using the same normalized format. However, as discussed in this chapter, there may be differences in the values of the populated fields based on the type of events (e.g., service impact events contain information about service name and state, while device level events do not).

3.5.5.2 Zenoss SNMP Notification Content

Zenoss CSA uses custom Zenoss MIB implementations for northbound notifications. The original SNMP MIB addresses the resource manager part of the product, but not the service impact part. MIB extensions have been designed to address service impact events and related probable root cause events as a part of the this phase of CLSA-VMDC. For a discussion of the new service impact trap, see Section 3.5.5.4 Zenoss Service Impact SNMP Trap.

Events associated with devices use ZENOSS-MIB for notifications. The ZENOSS-MIB.txt file is located in the following Zenoss directory: $ZENHOME/share/mibs/site. Device level SNMP notifications can be sent to multiple destinations. Refer to the Zenoss Cloud Service Assurance Installation and Administration Guide for more information regarding notifications.

Table 3-4 maps the fields of Zenoss MIBs to the SAM requirements.

Table 3-4. Zenoss MIB Fields

Zenoss MIB Field Name	Description
evtId	Unique identifier ID of the event
evtDedupid	De-duplication ID of the event
evtDevice	Device associated with event
evtComponent	Device component associated with event
evtClass	Event classification
evtKey	Event key used for refining event granularity beyond device and component. Used in de-duplication, automatic clearing.
evtSummary	Event message truncated to 128 characters
evtSeverity	Event severity number: 0=clear(normal), 1=debug, 2=info, 3=warning,4=error, 5=critical
evtState	Event state number: 0=new, 1=acknowledged, 2=suppressed
evtClassKey	Class key for rule processing often matches component
evtGroup	Logical grouping of event sources
evtStateChange	Last time event changed through administrative activity
evtFirstTime	First time an event was received
evtLastTime	Last time an event was received
evtCount	Number of times this event has been seen
evtProdState	Production state of the device or component associated with this event
evtAgent	Collector process that received or created this event
evtDeviceClass	Class of device that this event is associated with
evtLocation	Location of device that this event is associated with
evtSystems	Systems containing the device that this event is associated with
evtDeviceGroup	Groups containing the device that this event is associated with
evtIpAddress	IP address that this event was generated or sent from
evtFacility	Syslog facility if the event was initially sent as a syslog

Zenoss MIB Field Name	Description
evtPriority	Syslog priority if the event was initially sent as a syslog
evtNtEvId	Windows NT_EVENT_ID if the event was initially received from Windows event log
evtOwnerId	User that acknowledged this event
evtClearId	evtId that cleared this event
evtDevicePriority	Priority of the device that this event is associated with
evtClassMapping	Name of the event class mapping that matched this event

3.5.5.3 Zenoss Notification Filtering

Filtering capabilities using Zenoss Triggers can be used to customize notifications based on the needs of different northbound consumers:

- Multiple subscribers/receivers may receive notifications.
- Each notification subscriber/receiver may apply a different filter: one receiver may subscribe to service events, another may subscribe to compute events, and a third may subscribe to network events.
- Each system user should be able to apply different filters.

For more information regarding Triggers, refer to the Zenoss Cloud Service Assurance Installation and Administration Guide.

3.5.5.4 Zenoss Service Impact SNMP Trap

This section defines the SNMP notification for Zenoss Impact, which is new for CLSA-VMDC 3.0. The following data is available internally within Zenoss Impact for service related events. This data was used by the notification script in CLSA-VMDC 2.2.

- Service Name
- Severity
- Timestamp
- Service state
- URLs to EventDetail, page to acknowledge and close events, device events
- All events in the impact chain. Each event in impact chain includes:
  - Device
  - Component
  - Device Class
  - Event Class
  - Severity
  - Timestamp
  - Message

  - URLs to EventDetail, page to acknowledge and close events, device events

Zenoss Impact provides a flexible framework to define arbitrary services, including support for hierarchical service nesting. In such environments, the question arises for which nodes and/or levels of hierarchy notifications should be sent. Services are collected under Service Organizers. A Service Organizer consists of multiple folders and in each folder there is set of services. In Zenoss Impact, the notification trigger criteria is configured for Service Organizer folders and its services and not based on individual nodes and their hierarchy level in the impact tree. This approach provides good balance between flexibility to select notification trigger criteria and simplicity of implementation.

In order for CLSA-VMDC to send notifications per service instance state change, the appropriate structure must be created to organize the services. The following sections discuss the folders and the structure used for the services defined in CLSA-VMDC 3.0.

Service Organizers

Service Organizers are located on the left tab in the Impact GUI.

The Shared Services folder includes:

- Service Name X (e.g., Network Aggregation service, Network Core service, etc.)
- Service Name Y

The Customer Name folder includes:

- Tenant Service 1 (e.g., Tenant Compute service, Tenant Network service, etc.)
- Tenant Service 2

Notification Triggers

The user is able to select services and or/folders for which to send notifications. This action is available both in the GUI, as well as via the REST API so that the orchestration system at the time of onboarding the tenant service can select whether or not to enable notification for the service.

The notification policy should be selectable both per folder or per service instance. This enables support for the following use cases:

- Where a single operator or NB system manages and/or provides visibility to all services of single tenant/customer (since one folder is defined per tenant).
- Where different services of the same tenant are managed by different operators/NB systems, e.g., notification for IaaS services are sent to the IaaS operator while notifications for Unified Communications as a Service (UCaaS) services are sent to the UC operator.

Notification Timing

This section defines the guidelines for service impact notification triggers and timing. An attempt is made to balance any delay in notifications indicating change with excessive noise in events sent due to transient state during service impact analysis. In order to have the capability to delay some service impact notifications, there is a timer that can be configured (value range 0-10 minutes with default of three minutes).

Service impact notifications are triggered when the following events occur:

- If the service state changes (top-level service in each folder):
  - The notification indicating a service state change should always be sent immediately, regardless of the value of the notification delay timer. This enables the northbound system to immediately detect the change. Also, for northbound systems that are using service state

notifications to measure service availability and SLA, this immediate notification enables more accurate service availability measurements.

  - When the service state changes back to UP, the event should serve as a clearing event for the previous service state change event. As such, the ID of the service event that it is clearing must be included.

- If the service state does not change, but most a probable root-cause event changes (i.e., root cause ranked with highest confidence % changes):
  - This trigger honors the notification delay timer, and as such, it is sent only if the event is generated after the notification timer expires.

The following example shows the use of the notification delay timer and the two notification types listed above. Assume that the following conditions exist:

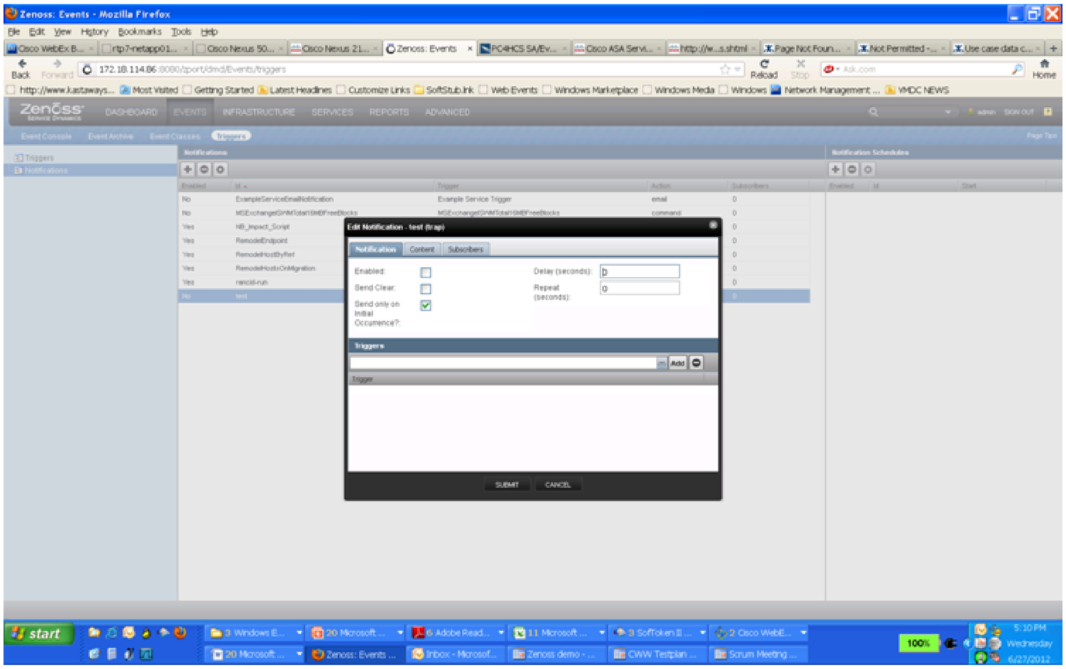
1. The notification delay timer is set to three minutes.
2. The root cause is a UCS blade failure, and the final service state for the IaaS service is AT RISK.

If these conditions exist, the following occurs:

1. At zero seconds, event E1 (VMware event for the VM) arrives. The root cause at that time is RC1= VM Failure.
2. A service impact event is sent northbound indicating that the IaaS state = AT RISK, RC=VM.
3. At one minute, event E2 (VMware event for the host) arrives. The root cause at that time is RC2= Host Failure. Since the notification delay timer is set to three minutes, there are no events sent northbound due to the change of root-cause events. Only one minute has passed since the service state change time.
4. At four minutes, event E3 (UCSM event for blade) arrives. The root cause at that time is RC3=UCS blade failure. A service impact event is sent northbound indicating that the IaaS state = AT RISK, RC= Blade.

Figure 3-11 shows the existing capability that Zenoss has to delay notifications and also to send the clearing events. The same capabilities would be extended to the service impact events.

Figure 3-11. Edit Notification Timer



Edit Notification - test (trap)

Notification | Content | Subscribers

Enabled: □

Send Clear: □

Send only on Initial Occurrence?: ☑

Delay (seconds): 0

Repeat (seconds): 0

Triggers

Trigger

Notification Content

The following fields should be included in the northbound service impact notification:

- **Folder Name (one up only).** The customer name would typically be placed here, but keeping the field generic allows flexibility to use folders in any way desired (e.g., to represent shared infrastructure services, reseller, etc.). The operator can include the option to have a full folder path.
- **Folder Type.** The folder type indicates what the folder represent, e.g., for folders representing the customer name, the folder type would have value the value "customer."
- **Service Instance Name and systemwide unique ID**
- **Service Type.** This field can be used to filter notifications by type of service that the northbound consumer is interested in, even though each instance of the service may be in different folders which are representing different customers.
- **Service State.** The service state is UP, DOWN, AT RISK, or DEGRADED.
- **URLs to Service Impact EventDetail.** This page acknowledges and closes events and device events.
- **Timestamp**
- **Event clearing ID.** The ID of the event that is being cleared by this event.
- **Probable root-cause event name and systemwide unique ID (event with highest confidence level)**
- **Probable root-cause confidence level**
- **Probable root-cause device, component, and severity**

- **Impact chain and ID to events in impact chain.** The ID can be used to retrieve the impact chain via REST API upon receipt of the notification.
- **URLs to probable root-cause EventDetail.** This page acknowledges and closes events and device events.

Note In CLSA-VMDC 3.0, the following fields are not supported: Folder Type, Service Type, and URLs to probable root-cause event detail. In addition, theEvent Clearing ID is implemented slightly differently than proposed above. The Service Instance Name & system wide unique ID is implemented in a field called zenImpactUUID. The initial and clearing events have the same zenImpactUUID, however they have states new and cleared.

Root-cause Event Notification

In addition to sending probable root-cause events as part of service impact notification, there is also a need to be able to send only probable root-cause events. For example, in cases of more catastrophic failures where a single root-cause event impacts a larger number of services, northbound systems that are not service focused may prefer to receive only one notification representing the root-cause event and not receive multiple service impacting notifications.

Even in this case, it is desirable to provide the relationship between the root-cause event and the services it impacted. This can be done by including a list of services impacted by the same root-cause event in the root-cause event notification URL or ID.

Root-cause notification is not a separate notification in CLSA-VMDC 3.0; instead, the root-cause event is communicated as a field via the service impact notification.

3.5.5.5 WS or ReST API

The JSON API can be used to obtain the following:

- Device model and attributes data
- Performance data
- Event data
- Service data

Most of the information visible via the GUI can also be obtained via the JSON API.

In addition to retrieving data, the JSON API can also be used for the following:

- Managing events (acknowledge, clear, close)
- Adding devices to be monitored
- Setting production state
- Initiating discovery and modeling of devices
- Managing thresholds
- Managing reports
- Other configurations

More information on the JSON API can be found at the following URL:

http://community.zenoss.org/community/documentation/official_documentation/api

3.5.5.6 Northbound Integration Use Case Examples

This section includes typical use cases that illustrate the rich filtering capabilities of the NBI.

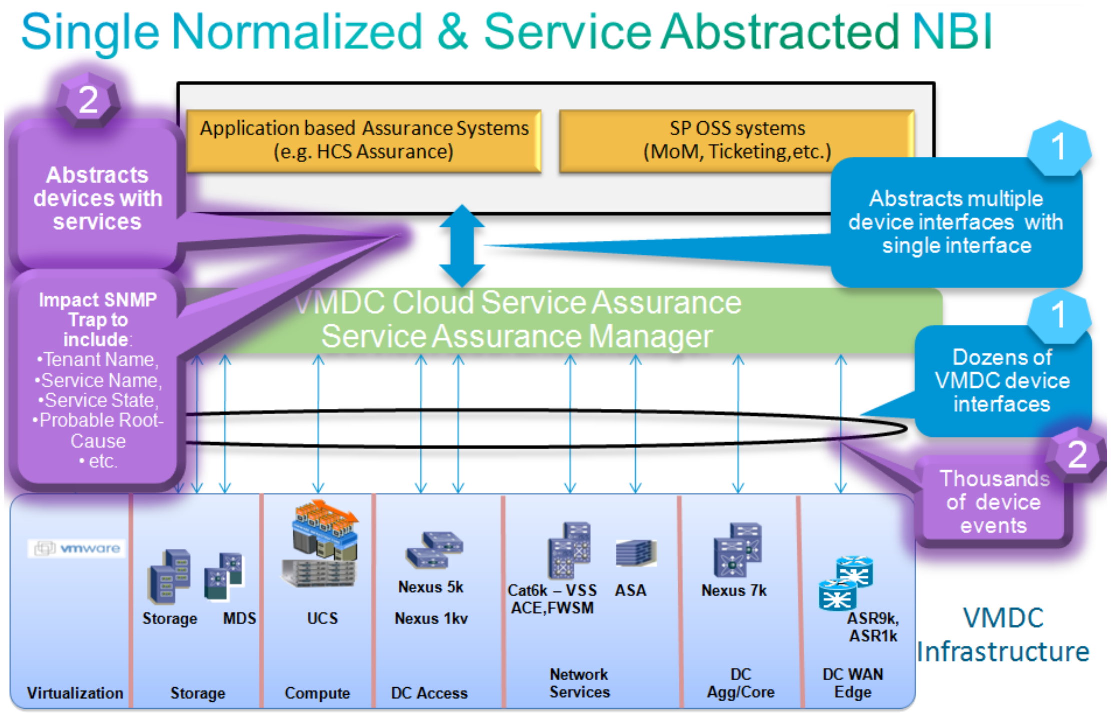
3.5.5.6.1 Abstraction via Single Interface

One of the key functions of the SAM layer as defined in CLSA-VMDC architecture is the capability to provide a single, normalized NBI that is consistent regardless of the formats of data used by the underlying VMDC components. This allows simplified integration and ongoing interface maintenance with providers existing OSS systems as:

- There is only one integration point as opposed to the number of integration points being proportional to the number of VMDC devices and domain managers.
- Changes in any of the underlying interfaces on managed devices are absorbed by the SAM as opposed to the provider having to update OSS systems every time there is a change in one of the managed components.

Figure 3-12 illustrates how the VMDC system is abstracted via a single interface to the provider's existing OSS system. The purple areas represent the enhancements for CLSA-VMDC 3.0.

Figure 3-12. Single Normalized and Service Abstraction NBI


Single Normalized & Service Abstracted NBI
2
Abstracts devices with services
Application based Assurance Systems (e.g. HCS Assurance)
SP OSS systems (MoM, Ticketing,etc.)
1
Abstracts multiple device interfaces with single interface
Impact SNMP Trap to include:
•Tenant Name,
•Service Name,
•Service State,
•Probable Root-Cause
• etc.
VMDC Cloud Service Assurance
Service Assurance Manager
1
Dozens of VMDC device interfaces
2
Thousands of device events
vmware
Storage
MDS
UCS
Nexus 5k
Nexus 1kv
Cat6k – VSS ACE,FWSM
ASA
Nexus 7k
ASR9k, ASR1k
Virtualization
Storage
Compute
DC Access
Network Services
DC Agg/Core
DC WAN Edge
VMDC Infrastructure

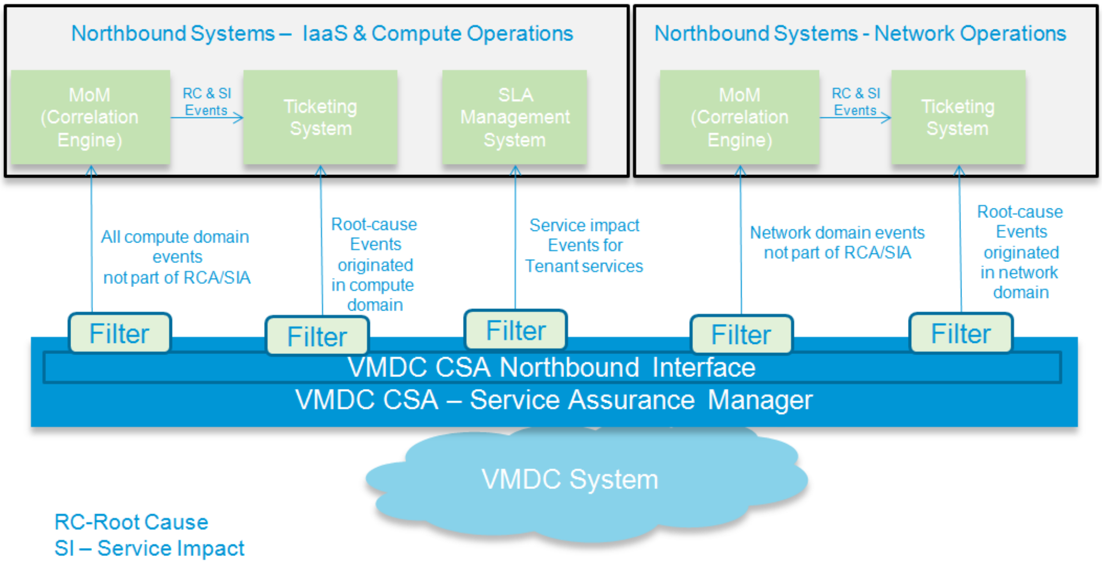
3.5.5.6.2 Integration With Multiple Northbound Systems

This use case example illustrates the need for different types of notifications and northbound filtering capabilities.

Figure 3-13. VMDC CSA Integration in Northbound OSS Systems

VMDC CSA Integration in Northbound OSS Systems

Example of filtering use case



Northbound Systems – IaaS & Compute Operations

MoM (Correlation Engine)

RC & SI Events

Ticketing System

SLA Management System

Northbound Systems - Network Operations

MoM (Correlation Engine)

RC & SI Events

Ticketing System

All compute domain events not part of RCA/SIA

Root-cause Events originated in compute domain

Service impact Events for Tenant services

Network domain events not part of RCA/SIA

Root-cause Events originated in network domain

Filter

Filter

Filter

Filter

Filter

VMDC CSA Northbound Interface

VMDC CSA – Service Assurance Manager

VMDC System

RC-Root Cause
SI – Service Impact

In this use case example, there is an IT department with two operations teams: one for network operations and one for server/compute operations. In addition, each one of the teams has a ticketing system and a MoM capable of further event processing and RCA. Assume also that the server/compute operations team has an SLA management system used to measure and manage SLA compliance.

Using the extensive filtering capabilities of the northbound notifications, the needs of both of these operations teams and their various northbound systems can be satisfied with a single instance of the service assurance system. In this example, five northbound notification destinations are configured, each with a different filter (also known as a notification trigger) as follows:

- All root cause events originated by vCenter or UCSM are sent to the Compute Operations ticketing system.
- All service-impact events originated by vCenter or UCSM are sent to the Compute Operations SLA management system.
- All other compute events that may require additional analysis are sent to the Compute Operations MoM.
- All root cause events originated by network devices are sent to the Network Operations ticketing system.
- All other compute events that may require additional analysis are sent to the Network Operations MoM.

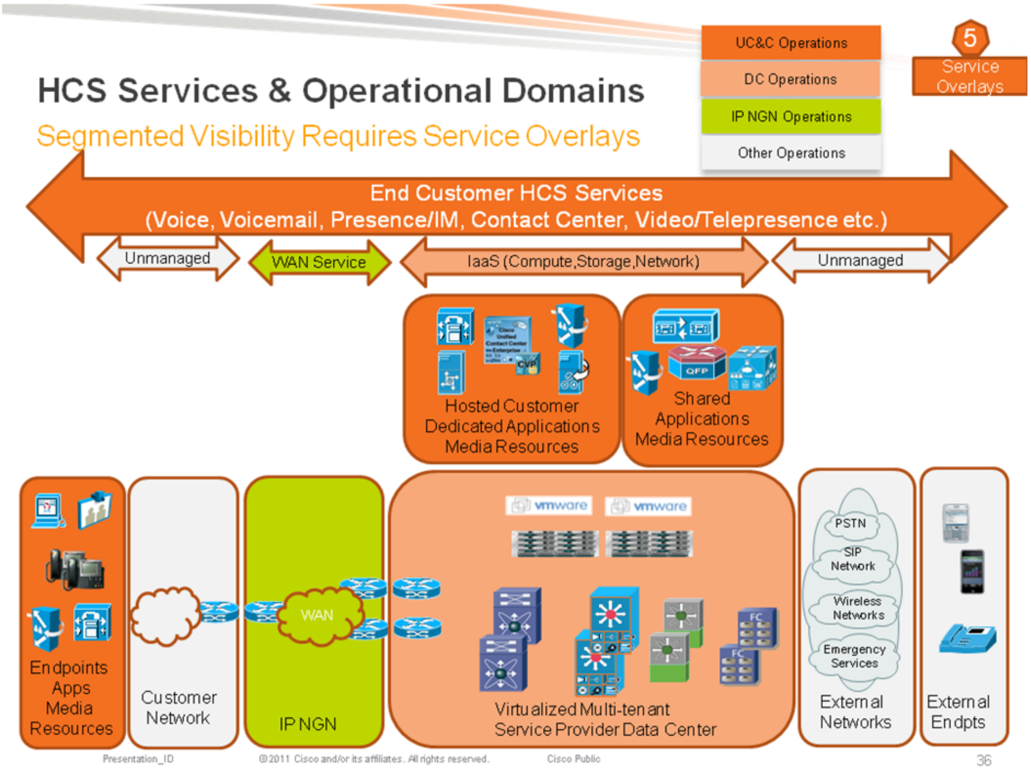
3.5.5.6.3 Abstraction Through Service Overlays

This use case illustrates the need for service impact notifications from CLSA-VMDC. This use case is a prerequisite for integrating CLSA-VMDC into CLSA-HCS. To deliver HCS services (voice, voicemail, etc.) to the end customer/tenant, multiple services need to be provided to the customer, which are referred to as service overlays. In a scenario for top-level service such as HCS, there are a number of benefits to only processing abstracted events related to a few underlying services:

- Complexity of its fault management system can be reduced significantly if it is only receiving events related to few underlying services (IaaS, MPLS VPN WAN service, etc.) rather than having to deal with device level events from tens of underlying components.
- More flexibility to support various business and operational deployment models that vary in which domains and services are owned and operated by the provider offering top-level (e.g., HCS) services.

Figure 3-14 and Figure 3-15 illustrate the service overlay approach for application-based services such as HCS, and the need for service level abstraction from the underlying infrastructure system.

Figure 3-14. HCS Services and Operational Domains



HCS Services & Operational Domains

Segmented Visibility Requires Service Overlays

UC&C Operations

DC Operations

IP NGN Operations

Other Operations

5

Service Overlays

End Customer HCS Services

(Voice, Voicemail, Presence/IM, Contact Center, Video/Telepresence etc.)

Unmanaged

WAN Service

IaaS (Compute,Storage,Network)

Unmanaged

Hosted Customer Dedicated Applications Media Resources

Shared Applications Media Resources

Endpoints Apps Media Resources

Customer Network

WAN

IP NGN

vmware

vmware

Virtualized Multi-tenant Service Provider Data Center

PSTN

SIP Network

Wireless Networks

Emergency Services

External Networks

External Endpts

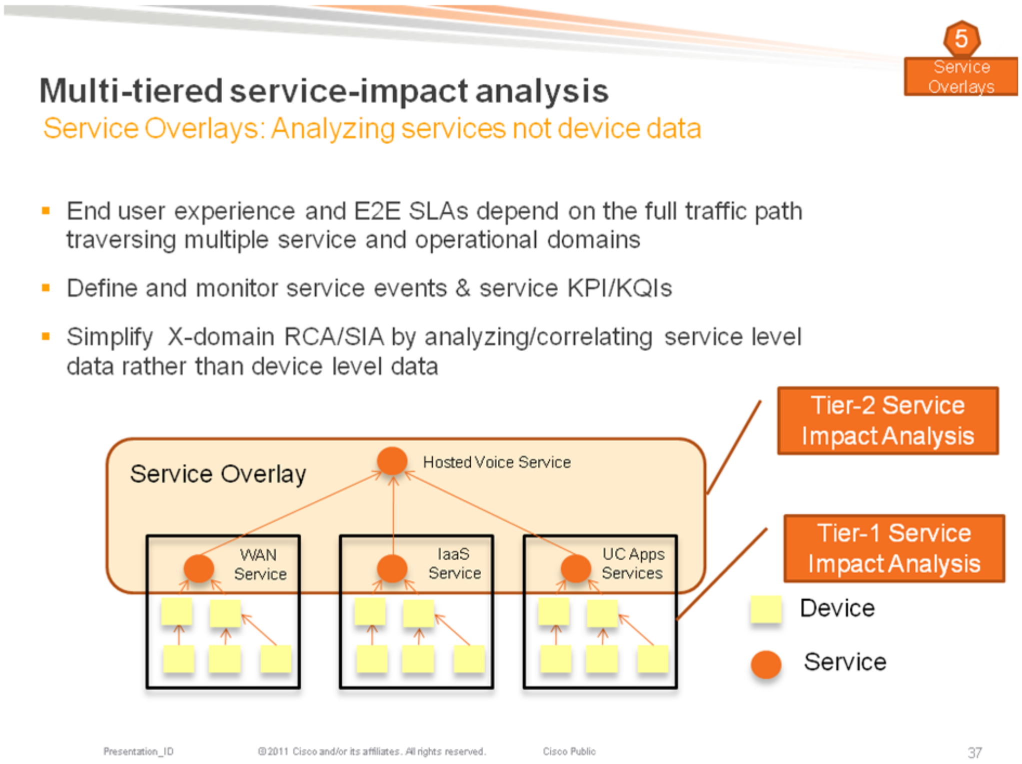
Presentation_ID

©2011 Cisco and/or its affiliates. All rights reserved.

Cisco Public

36

Figure 3-15. Multi-tiered SIA



Multi-tiered service-impact analysis

Service Overlays: Analyzing services not device data

- End user experience and E2E SLAs depend on the full traffic path traversing multiple service and operational domains
- Define and monitor service events & service KPI/KQIs
- Simplify X-domain RCA/SIA by analyzing/correlating service level data rather than device level data

Presentation_ID ©2011 Cisco and/or its affiliates. All rights reserved. Cisco Public 37

3.5.6 Performance Management

The following performance management capabilities are provided out-of-the-box in CLSA-VMDC:

- KPI statistics resource monitoring and trending:
  - Resource monitoring is partially validated as part of CLSA-VMDC.
- Performance service impact models for compute and storage:
  - TCAs utilized as part of SIA
  - Validated as part of CLSA-VMDC
- Application response time measurements:
  - Not validated as part of CLSA-VMDC
  - For details, refer to product documentation on www.zenoss.com.
- Performance reporting:
  - Not validated as part of CLSA-VMDC
  - For details, refer to product documentation on www.zenoss.com.

3.5.7 Dashboards

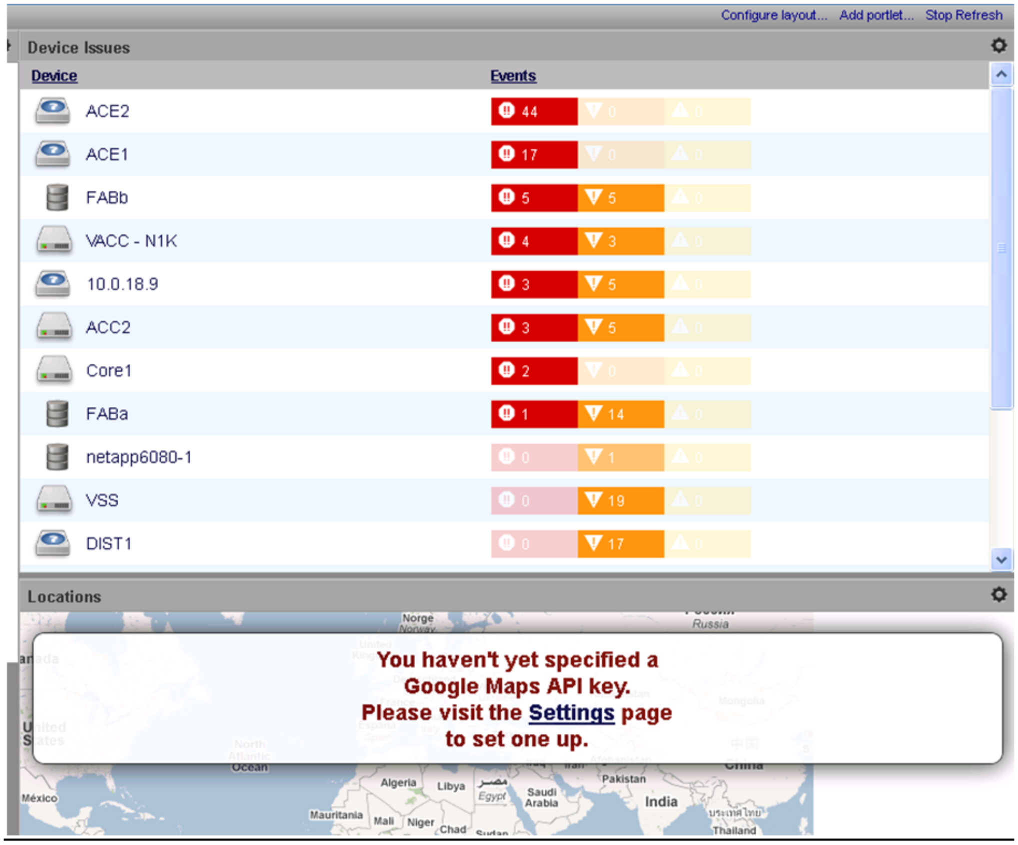
CLSA-VMDC features aggregated SP dashboards, as well as both device level and service level dashboards that operators can use to obtain more details. The following are the key dashboard categories for Zenoss CSA:

- Aggregated systemwide resources status dashboards
- Service inventory and status dashboards
- Infrastructure/resource monitoring dashboards
- Event dashboards

Aggregated Systemwide Resources Status Dashboards

These dashboards list all devices with events systemwide, sorted by number of highest priority events.

Figure 3-16. Aggregated Systemwide Resources Status Dashboard



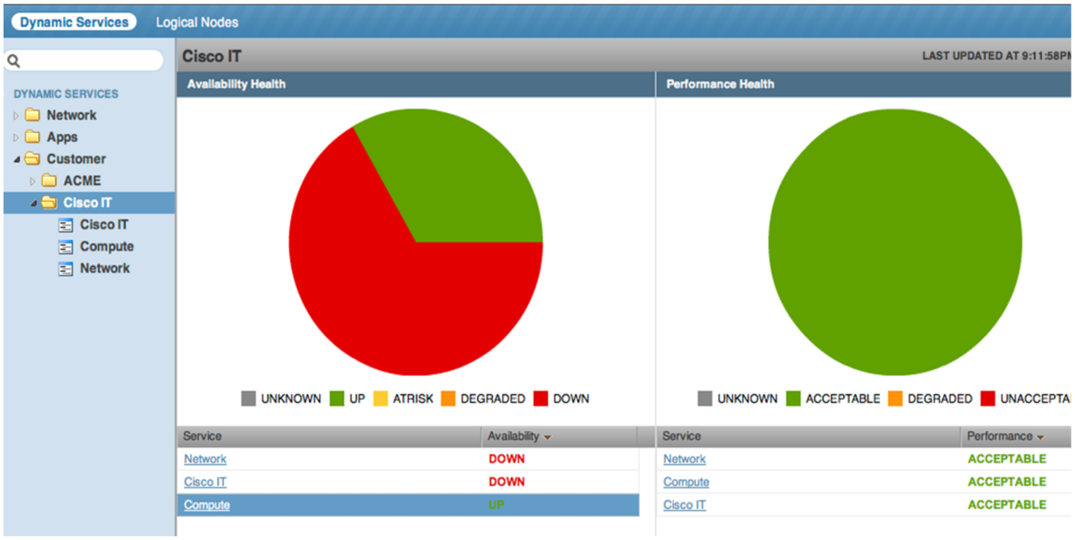
Device	Events		
ACE2	44	0	0
ACE1	17	0	0
FABb	5	5	0
VACC - N1K	4	3	0
10.0.18.9	3	5	0
ACC2	3	5	0
Core1	2	0	0
FABa	1	14	0
netapp6080-1	0	1	0
VSS	0	19	0
DIST1	0	17	0

You haven't yet specified a Google Maps API key. Please visit the Settings page to set one up.

Service Inventory and Status Dashboards

These dashboards show the availability and performance state of all services in the system.

Figure 3-17. Service Inventory and Status Dashboard

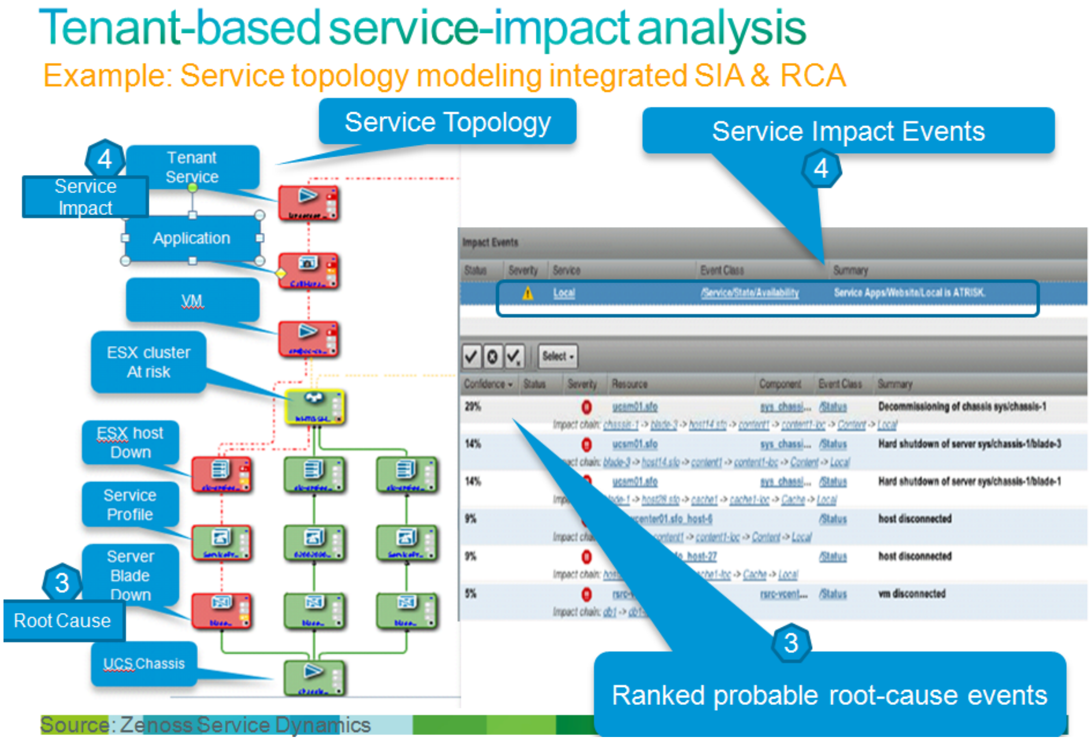


Service	Availability
Network	DOWN
Cisco IT	DOWN
Compute	UP

Service	Performance
Network	ACCEPTABLE
Compute	ACCEPTABLE
Cisco IT	ACCEPTABLE

Figure 3-18 shows a per-service detailed dashboard, which lists service impact events and related probable root cause events, as well as a visualization of the service model tree.

Figure 3-18. Per-service Detailed Dashboard



Impact Events

Status	Severity	Service	Event Class	Summary
	⚠	Local	/Service/State/Availability	Service Apps/Website/Local is ATRISK.

✓ ⊗ ✓ Select ▾

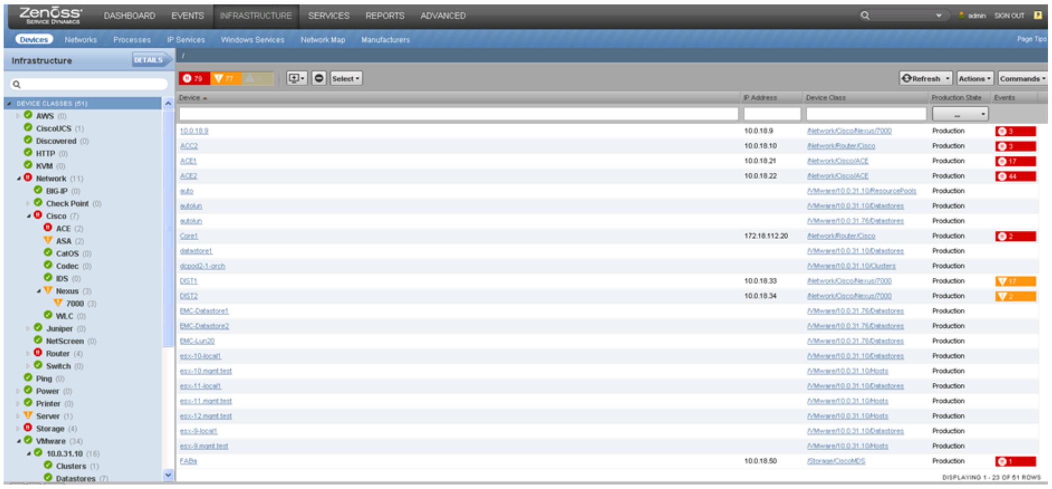
Confidence ▾	Status	Severity	Resource	Component	Event Class	Summary
29%		●	ucsm01.sfo	sys_chassi...	/Status	Decommissioning of chassis sys/chassis-1
			Impact chain: chassis-1 -> blade-3 -> host14.sfo -> content1 -> content1-loc -> Content -> Local			
14%		●	ucsm01.sfo	sys_chassi...	/Status	Hard shutdown of server sys/chassis-1/blade-3
			act chain: blade-3 -> host14.sfo -> content1 -> content1-loc -> Content -> Local			
14%			ucsm01.sfo	sys_chassi...	/Status	Hard shutdown of server sys/chassis-1/blade-1
			Imp ade-1 -> host28.sfo -> cache1 -> cache1-loc -> Cache -> Local			
9%			center01.sfo_host-6		/Status	host disconnected
			Impact cha content1 -> content1-loc -> Content -> Local			
9%		●	o_host-27		/Status	host disconnected
			Impact chain: hos che1-loc -> Cache -> Local			
5%		●	rsrc-	rsrc-vcent...	/Status	vm disconnected
			Impact chain: db1 -> db1			

Source: Zenoss Service Dynamics

Infrastructure/Resource Monitoring Dashboards

These dashboards list the inventory of all devices and their status.

Figure 3-19. Infrastructure Dashboard



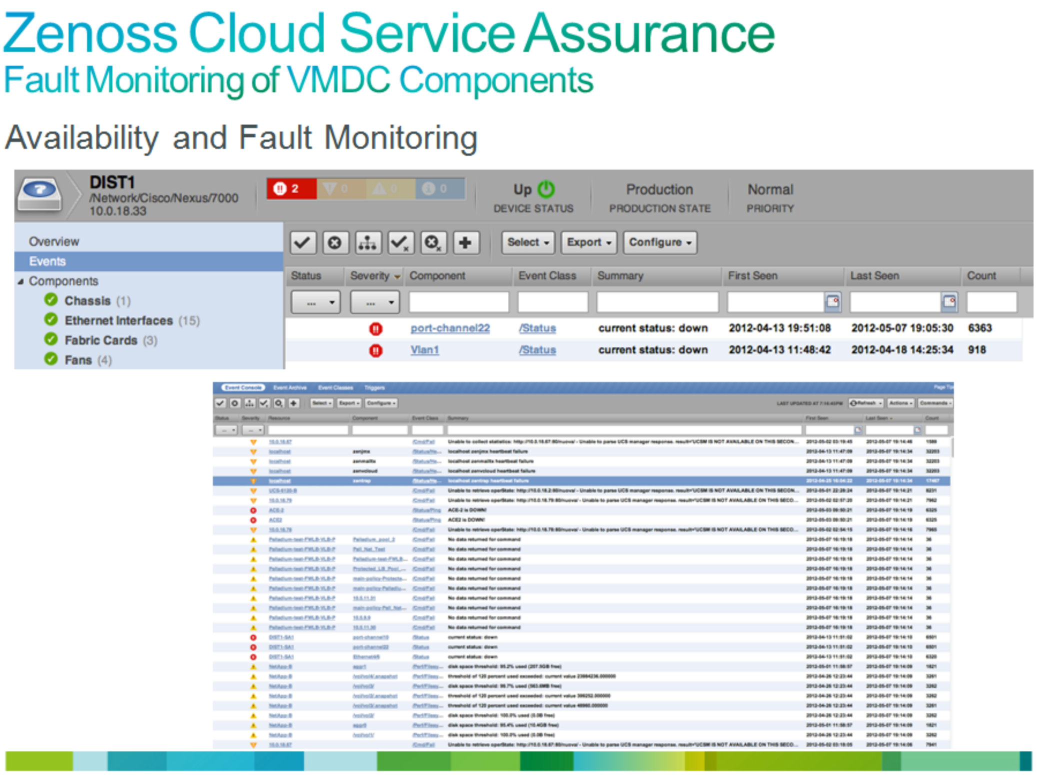
DEVICE CLASSES (51)
- AWS (0)
- CiscoUCS (1)
- Discovered (0)
- HTTP (0)
- KVM (0)
- Network (11)
  - BIG-IP (0)
  - Check Point (0)
  - Cisco (7)
    - ACE (2)
    - ASA (2)
    - CatOS (0)
    - Codec (0)
    - IDS (0)
    - Nexus (3)
      - 7000 (3)
    - WLC (0)
  - Juniper (0)
  - NetScreen (0)
  - Router (4)
  - Switch (0)
- Ping (0)
- Power (0)
- Printer (0)
- Server (1)
- Storage (4)
- VMware (34)
  - 10.0.31.10 (16)
    - Clusters (1)
    - Datastores (7)

Device ▲	IP Address	Device Class	Production State	Events
10.0.18.9	10.0.18.9	/Network/Cisco/Nexus/7000	Production	3
ACC2	10.0.18.10	/Network/Router/Cisco	Production	3
ACE1	10.0.18.21	/Network/Cisco/ACE	Production	17
ACE2	10.0.18.22	/Network/Cisco/ACE	Production	44
auto		/VMware/10.0.31.10/ResourcePools	Production	
autolun		/VMware/10.0.31.10/Datastores	Production	
autolun		/VMware/10.0.31.76/Datastores	Production	
Core1	172.18.112.20	/Network/Router/Cisco	Production	2
datastore1		/VMware/10.0.31.10/Datastores	Production	
dcpod2-1-orch		/VMware/10.0.31.10/Clusters	Production	
DIST1	10.0.18.33	/Network/Cisco/Nexus/7000	Production	17
DIST2	10.0.18.34	/Network/Cisco/Nexus/7000	Production	2
EMC-Datastore1		/VMware/10.0.31.76/Datastores	Production	
EMC-Datastore2		/VMware/10.0.31.76/Datastores	Production	
EMC-Lun20		/VMware/10.0.31.76/Datastores	Production	
esx-10-local1		/VMware/10.0.31.10/Datastores	Production	
esx-10.mgmt.test		/VMware/10.0.31.10/Hosts	Production	
esx-11-local1		/VMware/10.0.31.10/Datastores	Production	
esx-11.mgmt.test		/VMware/10.0.31.10/Hosts	Production	
esx-12.mgmt.test		/VMware/10.0.31.10/Hosts	Production	
esx-9-local1		/VMware/10.0.31.10/Datastores	Production	
esx-9.mgmt.test		/VMware/10.0.31.10/Hosts	Production	
FABa	10.0.18.50	/Storage/Cisco/MDS	Production	1

DISPLAYING 1 - 23 OF 51 ROWS

Figure 3-20 and Figure 3-21 show a detailed component dashboard and graphical view (example UCS server blade).

Figure 3-20. Detailed Component Dashboard



Zenoss Cloud Service Assurance

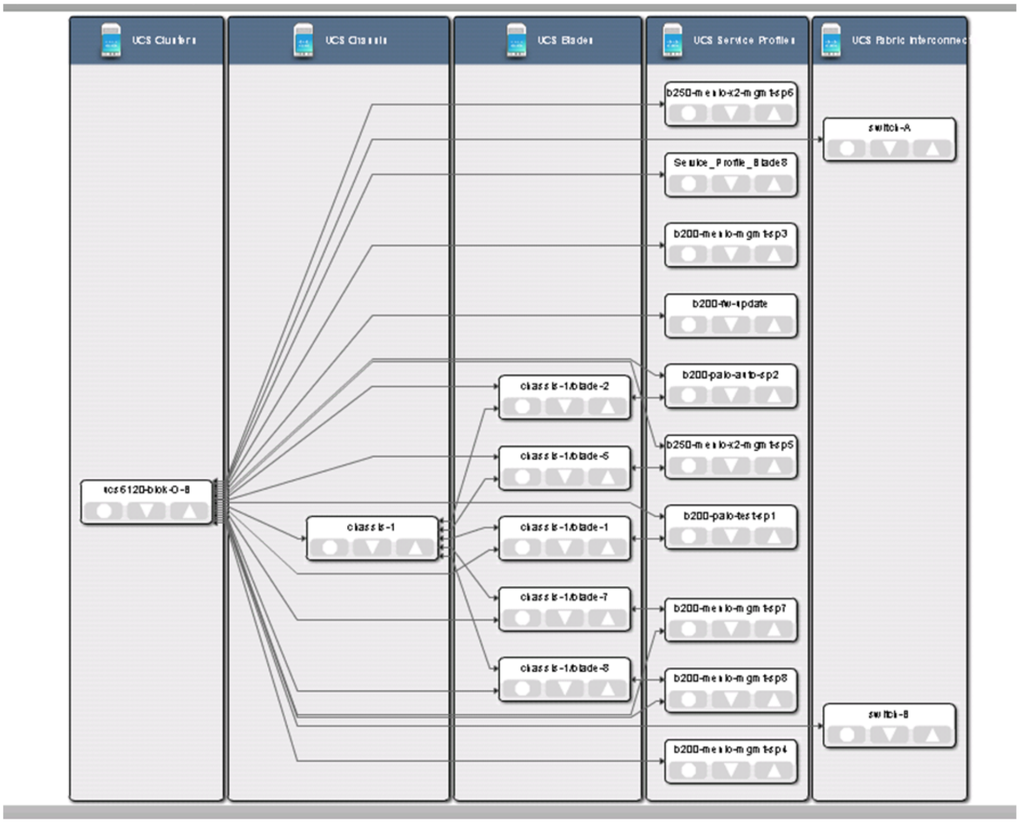
Fault Monitoring of VMDC Components

Availability and Fault Monitoring

Status	Severity	Component	Event Class	Summary	First Seen	Last Seen	Count
	‼	port-channel22	/Status	current status: down	2012-04-13 19:51:08	2012-05-07 19:05:30	6363
	‼	Vlan1	/Status	current status: down	2012-04-13 11:48:42	2012-04-18 14:25:34	918

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
	▼	10.0.18.67		/Cmd/Fail	Unable to collect statistics: http://10.0.18.67:80/nuova/ - Unable to parse UCS manager response. result="UCSM IS NOT AVAILABLE ON THIS SECON...	2012-05-02 03:19:45	2012-05-07 19:14:46	1589
	▼	localhost	zenjmx	/Status/He...	localhost zenjmx heartbeat failure	2012-04-13 11:47:09	2012-05-07 19:14:34	32203
	▼	localhost	zenmailtx	/Status/He...	localhost zenmailtx heartbeat failure	2012-04-13 11:47:09	2012-05-07 19:14:34	32203
	▼	localhost	zenvcloud	/Status/He...	localhost zenvcloud heartbeat failure	2012-04-13 11:47:09	2012-05-07 19:14:34	32203
	▼	localhost	zentrap	/Status/He...	localhost zentrap heartbeat failure	2012-04-25 16:04:22	2012-05-07 19:14:34	17467
	▼	UCS-6120-B		/Cmd/Fail	Unable to retrieve operState: http://10.0.18.2:80/nuova/ - Unable to parse UCS manager response. result="UCSM IS NOT AVAILABLE ON THIS SECON...	2012-05-01 22:28:24	2012-05-07 19:14:21	8231
	▼	10.0.18.79		/Cmd/Fail	Unable to retrieve operState: http://10.0.18.79:80/nuova/ - Unable to parse UCS manager response. result="UCSM IS NOT AVAILABLE ON THIS SECO...	2012-05-02 02:57:20	2012-05-07 19:14:21	7962
	ⓧ	ACE-2		/Status/Ping	ACE-2 is DOWN!	2012-05-03 09:50:21	2012-05-07 19:14:19	6325
	ⓧ	ACE2		/Status/Ping	ACE2 is DOWN!	2012-05-03 09:50:21	2012-05-07 19:14:19	6325
	▼	10.0.18.78		/Cmd/Fail	Unable to retrieve operState: http://10.0.18.78:80/nuova/ - Unable to parse UCS manager response. result="UCSM IS NOT AVAILABLE ON THIS SECO...	2012-05-02 02:54:15	2012-05-07 19:14:16	7965
	⚠	Palladium-test-FWLB-VLB-P	Palladium_pool_2	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	Pall_Nat_Test	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	Palladium-test-FWLB...	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	Protected_LB_Pool_...	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	main-policy-Protecte...	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	main-policy-Palladiu...	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	10.5.11.31	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	main-policy-Pall_Nat...	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	10.5.9.9	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	⚠	Palladium-test-FWLB-VLB-P	10.5.11.30	/Cmd/Fail	No data returned for command	2012-05-07 16:19:18	2012-05-07 19:14:14	36
	ⓧ	DIST1-SA1	port-channel10	/Status	current status: down	2012-04-13 11:51:02	2012-05-07 19:14:10	6501
	ⓧ	DIST1-SA1	port-channel22	/Status	current status: down	2012-04-13 11:51:02	2012-05-07 19:14:10	6501
	ⓧ	DIST1-SA1	Ethernet4/5	/Status	current status: down	2012-04-13 11:51:02	2012-05-07 19:14:10	6320
	⚠	NetApp-B	aggr1	/Perf/Filesy...	disk space threshold: 95.2% used (207.5GB free)	2012-05-01 11:58:57	2012-05-07 19:14:09	1821
	⚠	NetApp-B	/vol/vol4/.snapshot	/Perf/Filesy...	threshold of 120 percent used exceeded: current value 23984236.000000	2012-04-26 12:23:44	2012-05-07 19:14:09	3261
	⚠	NetApp-B	/vol/vol3/	/Perf/Filesy...	disk space threshold: 99.7% used (563.6MB free)	2012-04-26 12:23:44	2012-05-07 19:14:09	3262
	⚠	NetApp-B	/vol/vol2/.snapshot	/Perf/Filesy...	threshold of 120 percent used exceeded: current value 399252.000000	2012-04-26 12:23:44	2012-05-07 19:14:09	3262
	⚠	NetApp-B	/vol/vol3/.snapshot	/Perf/Filesy...	threshold of 120 percent used exceeded: current value 48960.000000	2012-04-26 12:23:44	2012-05-07 19:14:09	3261
	⚠	NetApp-B	/vol/vol2/	/Perf/Filesy...	disk space threshold: 100.0% used (0.0B free)	2012-04-26 12:23:44	2012-05-07 19:14:09	3262
	⚠	NetApp-B	aggr0	/Perf/Filesy...	disk space threshold: 95.4% used (10.4GB free)	2012-05-01 11:58:57	2012-05-07 19:14:09	1821
	⚠	NetApp-B	/vol/vol1/	/Perf/Filesy...	disk space threshold: 100.0% used (0.0B free)	2012-04-26 12:23:44	2012-05-07 19:14:09	3262
	▼	10.0.18.87		/Cmd/Fail	Unable to retrieve operState: http://10.0.18.87:80/nuova/ - Unable to parse UCS manager response. result="UCSM IS NOT AVAILABLE ON THIS SECO...	2012-05-02 03:18:05	2012-05-07 19:14:06	7941

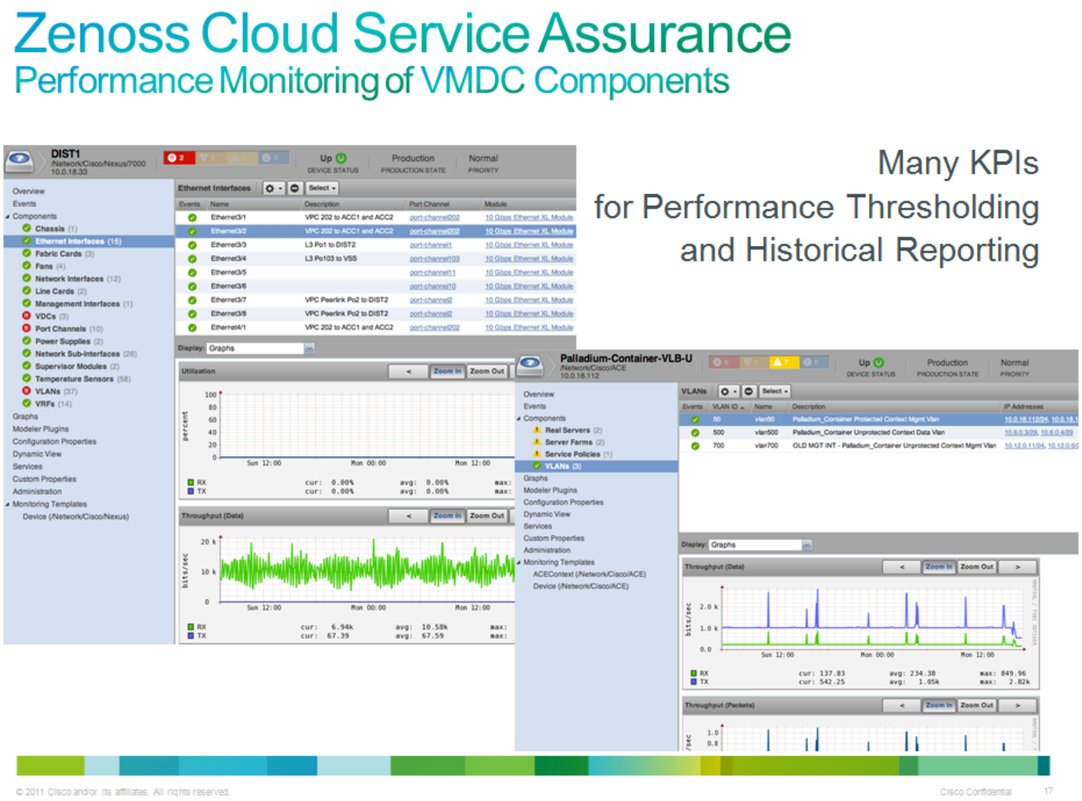
Figure 3-21. UCS Server Blade Graphical View



Event Dashboards

These dashboards show all events in the console (similar consoles exist per component as well).

Figure 3-22. Event Dashboard



Zenoss Cloud Service Assurance
Performance Monitoring of VMDC Components

Many KPIs
for Performance Thresholding
and Historical Reporting

DIST1
/Network/Cisco/Nexus/7000
10.0.18.33

Ethernet Interfaces

Events	Name	Description	Port Channel	Module
✓	Ethernet3/1	VPC 202 to ACC1 and ACC2	port-channel202	10 Gbps Ethernet XL Module
✓	Ethernet3/2	VPC 202 to ACC1 and ACC2	port-channel202	10 Gbps Ethernet XL Module
✓	Ethernet3/3	L3 Po1 to DIST2	port-channel1	10 Gbps Ethernet XL Module
✓	Ethernet3/4	L3 Po103 to VSS	port-channel103	10 Gbps Ethernet XL Module
✓	Ethernet3/5		port-channel11	10 Gbps Ethernet XL Module
✓	Ethernet3/6		port-channel10	10 Gbps Ethernet XL Module
✓	Ethernet3/7	VPC Peerlink Po2 to DIST2	port-channel2	10 Gbps Ethernet XL Module
✓	Ethernet3/8	VPC Peerlink Po2 to DIST2	port-channel2	10 Gbps Ethernet XL Module
✓	Ethernet4/1	VPC 202 to ACC1 and ACC2	port-channel202	10 Gbps Ethernet XL Module

Display: Graphs

Palladium-Container-VLB-U
/Network/Cisco/ACE
10.0.18.112

VLANs

Events	VLAN ID	Name	Description	IP Addresses
✓	50	vlan50	Palladium_Container Protected Context Mgmt Vlan	10.0.18.112/24, 10.0.18.1
✓	500	vlan500	Palladium_Container Unprotected Context Data Vlan	10.6.0.3/29, 10.6.0.4/29
✓	700	vlan700	OLD MGT INT - Palladium_Container Unprotected Context Mgmt Vlan	10.12.0.11/24, 10.12.0.6

Display: Graphs

Cisco Confidential 17

See Section 6.2 Dashboard Monitoring for a use case example of dashboard monitoring.

3.5.8 Reporting

CLSA-VMDC provides a range of defined and custom report options, including the following:

- Device reports
- Event reports
- Performance reports
- Graph reports
- Multi-graph reports
- Custom device reports

Reports can be exported to external files and systems or can be viewed locally. Reports can also be generated ad hoc or scheduled. See the Zenoss Cloud Service Assurance Installation and Administration Guide for more information.

The following is a list of reports supported out-of-the-box for CLSA-VMDC:

- Device Reports (9)
  - All Devices
  - All Monitored Components
  - Device Changes

  - MAC Addresses
  - Model Collection Age
  - New Devices
  - Ping Status Issues
  - SNMP Status Issues
  - Software Inventory
- Custom Device Reports
- Graph Reports
- Multi-graph Reports
- Event Reports (3)
  - All EventClasses
  - All EventMappings
  - All Heartbeats
- Performance Reports (7)
  - Aggregate Reports
  - Availability Report
  - CPU Utilization
  - Filesystem Util Report
  - Interface Utilization
  - Memory Utilization
  - Threshold Summary
- Storage (3)
  - Clients
  - Licenses
  - Disk Firmware
- Enterprise Reports (17)
  - Organizer Graphs
  - 95th Percentile
  - Defined Thresholds
  - Interface Volume
  - Network Topology
  - Customized Performance Templates
  - User Event Activity
  - Notifications and Triggers by Recipient
  - Datapoints by Collector
  - Organizer Availability
  - Maintenance Windows

  - Interface Utilization
  - Event Time to Resolution
  - Data Sources in Use
  - Users Group Membership
  - Cisco Inventory
  - Guest to Datapools
- MSExchange (1)
  - MSExchangeAvailability
- VMware (5)
  - ESXs
  - VMware Utilization
  - VMs
  - Datastores
  - Clusters
- Cisco UCS Reports (2)
  - Hardware Inventory
  - Free Slots

3.5.9 Multi-tenancy

This section discusses the CLSA-VMDC approach to multi-tenancy. VMDC architecture supports multi-tenant delivery, and CLSA-VMDC must therefore support an assurance window into these tenant services to equip cloud providers with the ability to assure logically distinct customer services. A related topic, Role-Based Access Control (RBAC), is also presented in this section.

CLSA-VMDC multi-tenancy

VMDC provides a multi-tenancy cloud infrastructure by logically separating tenant services that are implemented on a shared physical infrastructure. Tenants consume a portion of network, storage, and compute resources that have been allocated from the larger pool represented by the cloud. CLSA-VMDC delivers cloud provider assurance of shared infrastructure devices and their sub-components. In addition, CLSA-VMDC supports the multi-tenancy aspect of the VMDC architecture through the use of defined Tenant Services.

Zenoss CSA enables an administrator to stitch together service element nodes which taken as a whole comprise a specific tenant service. A CLSA-VMDC tenant service begins with creation of the topmost element node named for the tenant. To this tenant node, underlying VMware vSphere and UCS shared infrastructure elements can be discovered and attached. See Section 3.5.4.2 VMDC Assurance Service Models for more details regarding tenant services.

Using the tenant service modeling feature of CLSA-VMDC, cloud customers' services can be assured independently. Elements of the tenant service that are unique to that tenant customer such as specific VM's are visible only to the cloud provider or the service owner. Elements of the service that belong to shared infrastructure, such as a UCS chassis or a storage device are visible across multiple tenant services, as would be expected. In fact, if a shared device experiences a fault condition, all services associated with that device should be impacted. However, any fault condition associated with unique elements of a tenant service are not visible to other tenants.

Note This phase of CLSA-VMDC only supports cloud provider visibility into dashboards and service impact trees. Tenant customer visibility into service impact trees via customer portals will be supported in future releases.

RBAC Implementation

As a cloud providers' infrastructure increases in scale, it becomes important to provide a segmentation of operations capability to implement a division of responsibility. CLSA-VMDC fulfills this need with its RBAC implementation. Beyond this division of responsibility capability, RBAC can also be used to support groups of users with limited visibility into specific tenant services. Table 3-5 lists the out-of-the-box roles that may be used to segment cloud provider operations responsibilities and access.

Table 3-5. Global User Role Definitions

Role	Definition
ZenUser	Provides global read-only access to system objects.
ZenManager	Provides global read-write access to system objects.
Manager	Provides global read-write access to system objects and read-write access to the Zope object database (which includes all devices, users, and event mappings)
ZenOperator	Provides users the ability to manage events, i.e. acknowledge, move to archive, etc.

These predefined roles are global in scope, such that an operator may access all cloud objects, but only be allowed certain operations. In addition to these global user roles, users may be defined that are more limited in scope. These user roles may be assigned to organizational groups to manage a subset of the entire infrastructure or even specific tenant services.

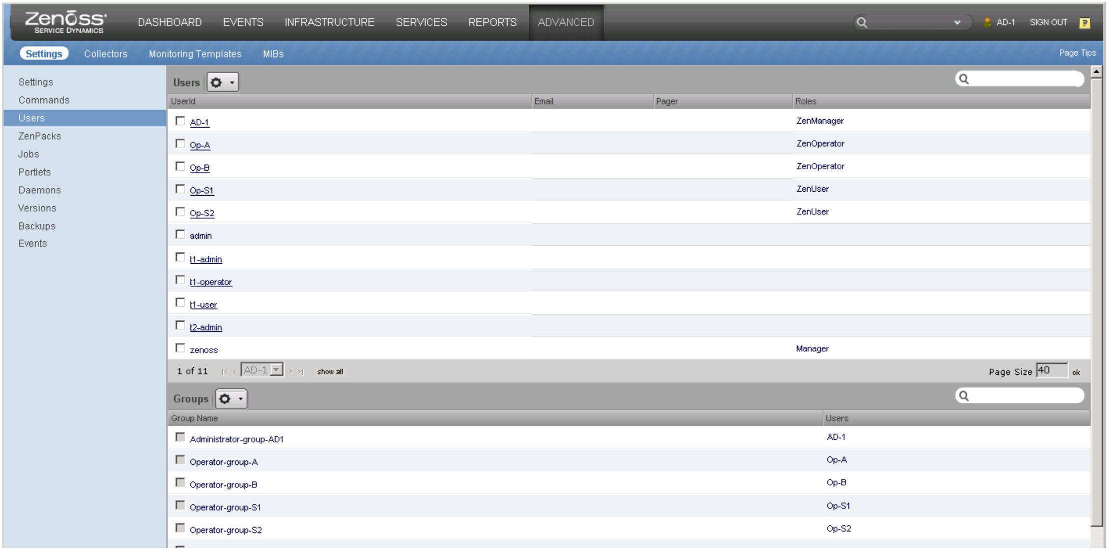
Organizational groups are used to collect subsets of infrastructure and/or services into logical categories for segmented operations. Table 3-6 lists the broad group categories and suggested uses for each group type. Groups can be devised for each tenant such that a customer's tenant services can be assigned to a tenant group.

Table 3-6. Device and Service Group Categories

Group Categories	Group Purpose
Group	Can be used for collecting similar devices, e.g. all switches group, all compute devices group
Systems	Can be used to collect all equipment with a specific data center, e.g. data center A, data center B
Locations	Can be used to collect devices by geographic boundaries, e.g. city, state, or even specific device rack

Figure 3-23 illustrates a list of users with either global or customized group role assignments. Users with the global user roles would belong to the cloud provider.

Figure 3-23. Custom User Groups



UserId	Email	Pager	Roles
□ AD-1			ZenManager
□ Op-A			ZenOperator
□ Op-B			ZenOperator
□ Op-S1			ZenUser
□ Op-S2			ZenUser
□ admin			
□ t1-admin			
□ t1-operator			
□ t1-user			
□ t2-admin			
□ zenoss			Manager

1 of 11 AD-1 show all Page Size 40 ok

Group Name	Users
□ Administrator-group-AD1	AD-1
□ Operator-group-A	Op-A
□ Operator-group-B	Op-B
□ Operator-group-S1	Op-S1
□ Operator-group-S2	Op-S2

Users with non-global roles can belong to either the cloud provider or even a cloud customer. While RBAC is only supported for the cloud provider in this phase, it could be the mechanism to deliver a cloud customer, or multiservices, assurance portal in future phases.

CHAPTER 4

Zenoss Cloud Service Assurance Overview

Zenoss Cloud Service Assurance (CSA) is used as the Service Assurance Manager (SAM) for Cloud Service Assurance for Virtualized Multiservice Data Center (CLSA-VMDC). Zenoss CSA consists of the following two elements:

- Core assurance platform
- VMDC ZenPacks that include VMDC specific device plugins that provide out-of-the-box support for VMDC components on the core assurance platform

This chapter discusses the product architecture and provides an overview of the capabilities of the core assurance platform. This chapter presents the following topics:

- Section 4.1 Zenoss Cloud Service Assurance Functional Overview
- Section 4.2 Zenoss Cloud Service Assurance Architecture Highlights

4.1 Zenoss CSA Functional Overview

Zenoss CSA focuses on the assurance and optimization category of cloud operations. The console provides a unified view of assurance and operations. Cloud operations consoles meet a combination of Enterprise role-based security and Service Provider (SP) multiservice needs. Enterprises have traditionally defined multiple organizational roles limiting the actions a user in the role can perform. SPs have traditionally provided each customer a view of just their given resources as a paid service. In cloud-enabled organizations, both needs must be met simultaneously.

As far as customers are concerned, the ability to deliver against the service level they chose from the Service Catalog is paramount. Whether characterizing this mutual understanding as an expectation or a formal agreement, there is a need to track and report against the agreed to metrics in a manner that is easy to understand. This means providing separate views for each customer and workload, and operating as if there are multiple distinct tenants using common cloud resources.

To provide the service level metrics and meet the expectations, the workloads and the supporting infrastructure must be monitored to detect issues that may impact the customer. The Impact and Event and Management collects device, application, and infrastructure information that identifies potential issues and evaluates it to determine which issues are critical. Performance Monitoring collects vital performance statistics at every layer of the infrastructure, evaluates it to determine whether anything should be fed into the Impact Management process, and stores it for long term analytics. These two functions provide reactive management to application, device, and infrastructure issues.

For some issues such as a sudden failure of a power supply, there is no advanced warning. For other issues, there are trends that can be collectively understood and provide early warning of an impending issue. Predictive Analytics attempts to provide proactive analysis of data, searching for identification and remediation of issues before they reach a critical state.

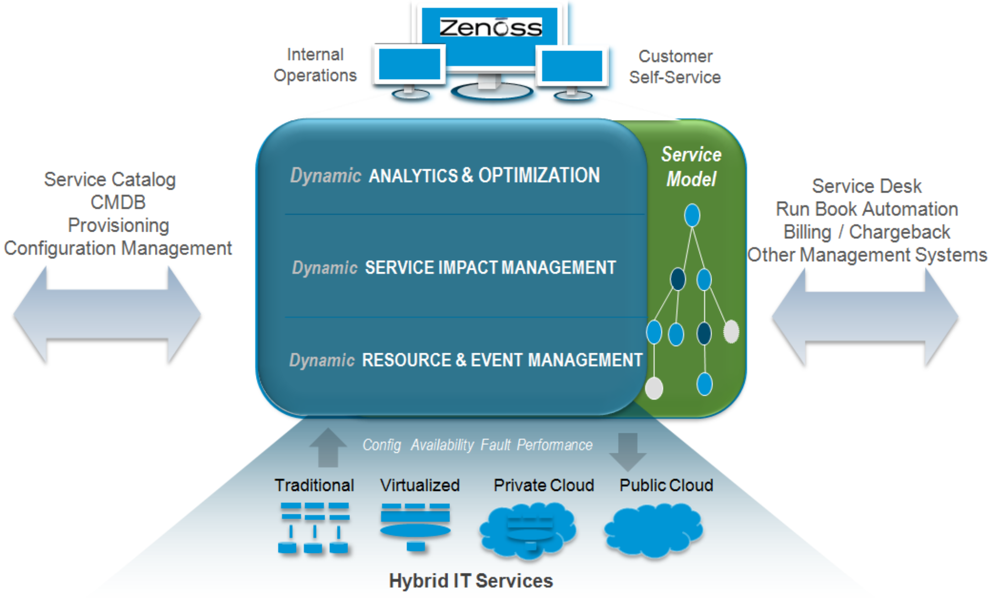
In cloud operations, the discipline of Capacity Analytics changes roles from an assessment of workload-driven resource requirements against the fixed amount available from a dedicated hardware platform. Reacting to the assessment can take weeks as new hardware is provisioned. Within the cloud Data Center (DC), resource allocations can be altered in minutes, and the reaction to changing workload needs should be just as fast. Administrators need to communicate to customers the needs of their workloads and get confirmation that they are willing to bear any increased costs, or allow the resource needs to go unmet and application performance to suffer. Administrators also need to provision enough resources for the overall DC to ensure that peak load commitments can be met, which means understanding capacity at an aggregated level.

Figure 4-1 shows the key functions that are available within Zenoss CSA. There are three key categories of the functions implemented in three software sub-components:

1. Resource Management, which provides discovery, data collection, performance, and event monitoring and notification capabilities.
2. Impact and Event Management, which provides full event management lifecycle, Root Cause Analysis (RCA), and Service Impact Analysis (SIA).
3. Analytics and Optimization, which provides a data warehouse for long term data trending, analytics engine, and reporting capabilities.

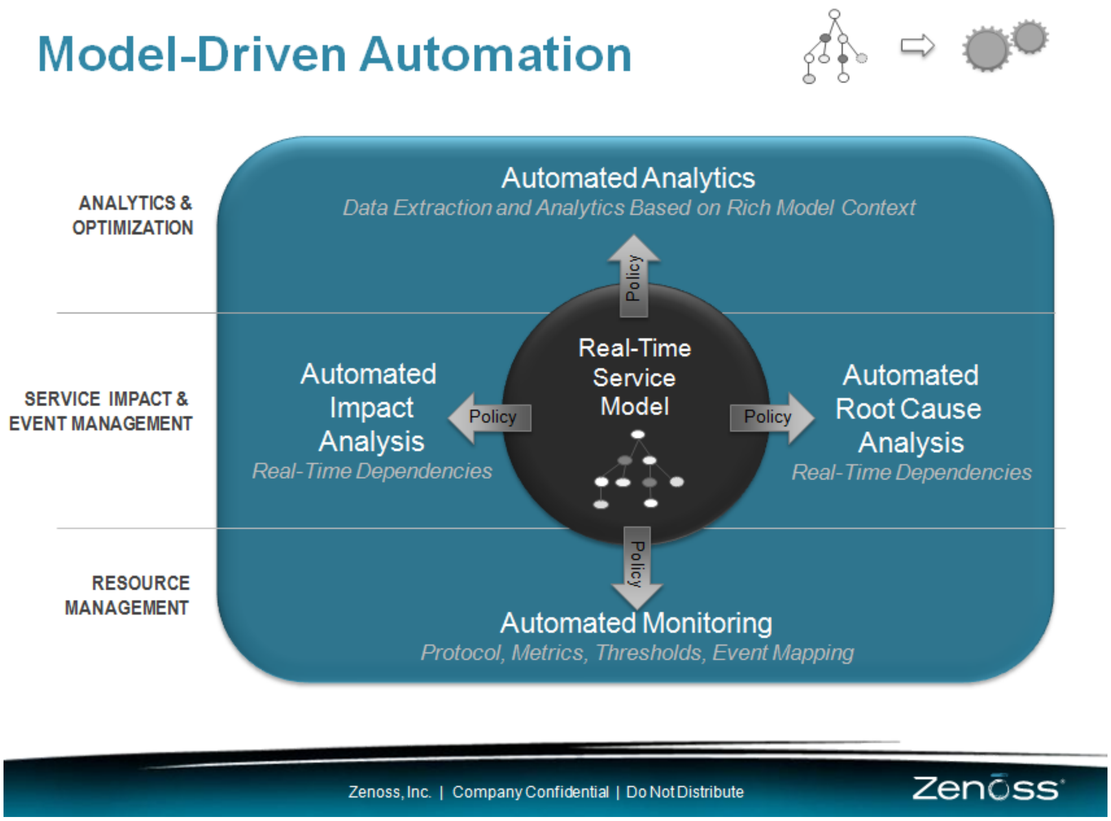
Note Resource Management and Impact and Event Management are the focus of CLSA-VMDC. The Analytics and Optimization functions were only evaluated on a best effort basis, but are not fully customized nor validated for VMDC environments.

Figure 4-1. Key Functions of Zenoss CSA



The core issue in the cloud DC is keeping up with its rapid rate of change. When a customer's order for a new IT service is automatically fulfilled and placed into production, there is no time to manually update tools or components. At the heart of Zenoss CSA is a unified, real-time understanding of the entire IT environment, including resources, services, relationships, dependencies, state, and configuration. With this understanding, Zenoss is able to simplify the service assurance process with template-driven resource monitoring and automated impact and RCA. Unlike traditional systems that rely on configuration databases that are updated in batch mode, the Zenoss model is maintained in near real time through a series of discovery and modeling techniques that tap into the stream of configuration changes as they happen across the physical, virtual, and cloud-based infrastructure. Like every other aspect of the product, this model can be extended through its open API.

Figure 4-2. Model-Driven Automation



Model-Driven Automation

ANALYTICS & OPTIMIZATION

SERVICE IMPACT & EVENT MANAGEMENT

RESOURCE MANAGEMENT

Automated Analytics
Data Extraction and Analytics Based on Rich Model Context

Automated Impact Analysis
Real-Time Dependencies

Real-Time Service Model

Automated Root Cause Analysis
Real-Time Dependencies

Automated Monitoring
Protocol, Metrics, Thresholds, Event Mapping

Zenoss, Inc. | Company Confidential | Do Not Distribute

The following sections provide more details on the key functions of Zenoss CSA.

4.1.1 Dynamic Resource Management

The foundation of service assurance in the hybrid cloud DC is unified, cross-domain resource monitoring and control that brings together configuration, performance, availability, fault, event and log information across the physical, virtual and cloud-based infrastructure and applications, and enables automated actions to be performed at the resource level. Zenoss CSA delivers this capability on a scalable, open platform that is easy to extend, and is able to track dynamic elements and relationships as they evolve in near real time.

Key feature areas include the following:

- **Discovery and modeling.** Automatically maintain real-time inventory and configuration details for the entire IT environment; includes real-time relationship tracking of dynamic relationships common in virtualized and cloud-based infrastructures.

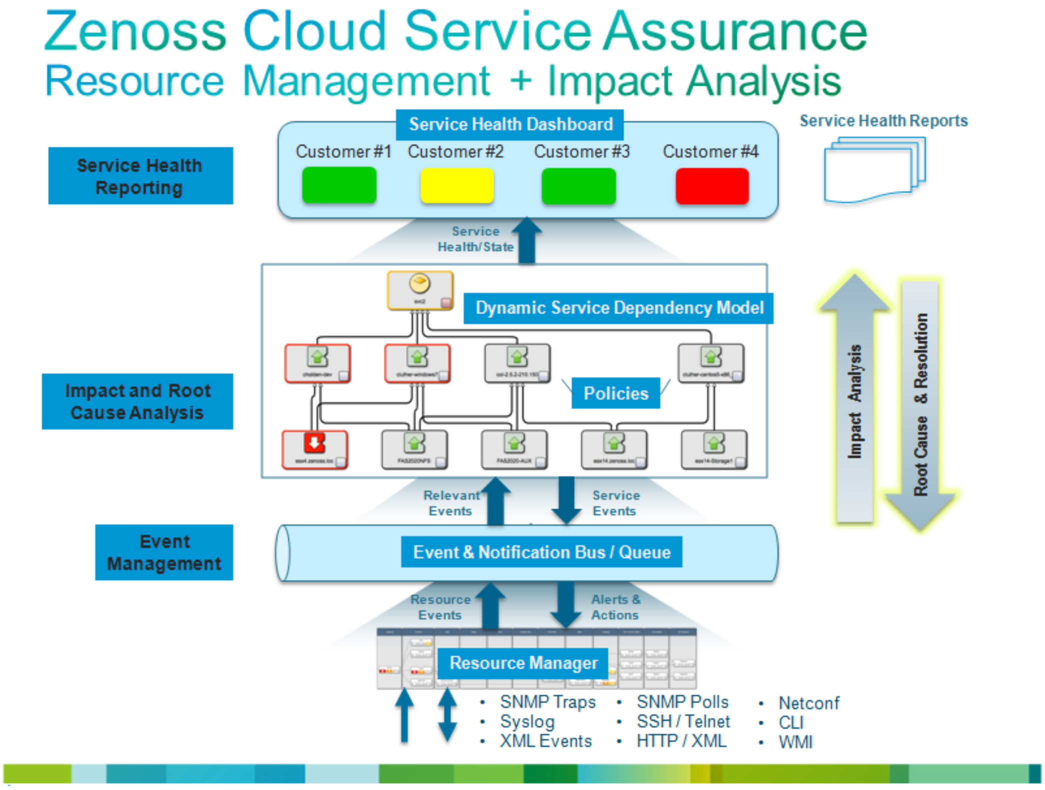
- **Full-stack monitoring.** Unify and automate performance, availability, and event monitoring of networks, servers, storage and applications across physical, virtual, and cloud-based environments with a single, model-driven, horizontally-scalable, extensible collection platform.
- **Notification and control.** Rich alerting and remediation framework allows the user to be notified via email, text, or pager based on user-specified policies, or to take direct automated action to address a problem in real time.

4.1.2 Dynamic Impact and Event Management

Maintaining a real-time perspective on service health, and linking health issues to the underlying infrastructure in a reliable, simple, and cost-effective way can be challenging in hybrid data centers. In particular, legacy approaches to impact management and RCA simply break down due to the shared and dynamic nature of virtualized and cloud-based infrastructures.

Figure 4-3 illustrates how Zenoss CSA transforms state information from the resource manager into a stream of events that are processed in near real time by its SIA and RCA engine, leveraging the real-time service model. This processing feeds a service health dashboard and generates service events that are used for alerting, troubleshooting, and to initiate real-time automation and service remediation. The result is real-time service level awareness, rapid triage, and closed-loop automation that thrives in the dynamic, hybrid cloud environment.

Figure 4-3. Service Health Dashboard



Dynamic Service Modeling

Zenoss CSA maintains a real-time service model, and automatically discovers infrastructure dependencies. Service constructs can be easily defined based on logical business constructs to define the infrastructure groupings supporting a specific application service. For example, a Customer Relationship Management (CRM) application service might require e-mail, web, and database services to be present in order to operate. These logical definitions define the collection of services required for the CRM service to be considered functional. Once this logical service hierarchy is defined, the application or OS instances delivering the service functions are associated and Zenoss CSA takes care of the rest. Using advanced dependency modeling capabilities, Zenoss CSA pulls in all relevant infrastructure elements. Virtual Machine (VM) partitions, blades, chassis, storage, network interfaces, and a wide variety of device components are all automatically discovered and mapped into the relevant service dependency graphs.

Dynamic Impact Analysis

Dynamic Impact Analysis identifies which services are affected by conditions in supporting components or infrastructure. For example, a failing fan in a Cisco UCS chassis might result in dozens of virtual machines being moved to new virtual hosts. With Zenoss impact analysis, IT operations can determine which business services will be affected by the fan failure and can plan corrective actions to minimize service level disruptions.

Dynamic RCA

Dynamic RCA allows IT operators to quickly identify the specific events most likely to be the cause of a service impacting condition. In complex IT environments, it is not uncommon for a single component failure to cause a cascade of failures, resulting in an event storm totaling thousands of individual events. Zenoss CSA includes a proprietary **Confidence Ranking Engine** built on top of Impact Analysis to quickly triage these events and identify where IT resources should be applied to correct these types of situations. This algorithm filters impact events based on a variety of criteria including severity of the event, service graph depth, and the number of graph branches affected by an event. This ranking algorithm allows IT operators to target resources to address events deemed the most likely cause of a service failure or degradation. Real world deployments of Zenoss CSA have validated the effectiveness of the service impact framework by demonstrating significant event reduction and highly accurate identification of root cause events.

Simple, Modular Policy - "Policy Gates"

Traditional impact managers require complex, top down rule sets to be defined, which require either a static IT infrastructure or a detailed understanding of all possible infrastructure configurations to identify service impact or determine root cause. This approach fails in dynamic virtualized or cloud data centers due to the need to maintain hard dependencies on named infrastructure elements. In contrast, Zenoss CSA uses Policy Gates to define impact rules on an element-by-element basis, and rolls up impact results through the current service model to reach conclusions. The design premise of this system is that the state of any given element in a service graph is determined by analyzing the state of the immediate children of that element. A change in the state of a given element is propagated to the parents of that element, causing the parents to evaluate their own state using their own Policy Gate configurations. The net result of this approach is that functions such as event aggregation, filtering, de-duplication and masking are provided automatically, eliminating the need for highly specialized skills to write impact rules and dramatically reducing human effort in event processing.

Unified, Scale Event Management

Aggregate and manage events for an entire IT environment with a next generation event management system that provides automated event normalization and enrichment, and is easily extended, integrated, and scaled through an embedded message bus. Zenoss CSA is capable of processing in excess of 1,500 events per second with a single event processor. Field deployments have shown that the system is capable of quickly parsing through event storms scaling to thousands of events in seconds, resulting in just a handful of events after processing through the Service Impact and Confidence Ranking.

4.1.3 Dynamic Analytics and Optimization

The final step of the service assurance lifecycle is historical analysis and planning. Deep, cross-domain analytics are needed to perform capacity planning and drive optimization of the environment. Zenoss

CSA enables this through its integrated analytics capabilities that directly leverage the real-time service model and all state information from the Resource, Impact, and Event Management modules. Leveraging a powerful, open business intelligence engine, the Zenoss analytics capability provides a scalable and rich analytics platform that addresses tenant reporting needs, management dashboards, capacity planning, and insight for optimization. Specific capabilities include:

- **Turnkey operations data warehouse.** Automatically aggregates and normalizes configuration, performance, and event history for the entire IT environment across physical, virtual, and cloud-based infrastructure and applications.
- **Unified historical analytics.** Understand utilization and health trends across an IT's entire infrastructure including tenant-based consumption and availability reporting; gain deep, timely insight through drag-and-drop dashboards, out-of-the-box reports, and powerful ad-hoc analytics all available through a multiservice web portal. The Zenoss Analytics software module is not validated or included as part of CLSA-VMDC. However, this software module can be obtained directly from Zenoss, Inc.
- **Predictive analytics.** Forecast capacity needs and anticipate availability problems through predictive trending that allows for visualization and proactive management of upcoming operational issues and infrastructure requirements

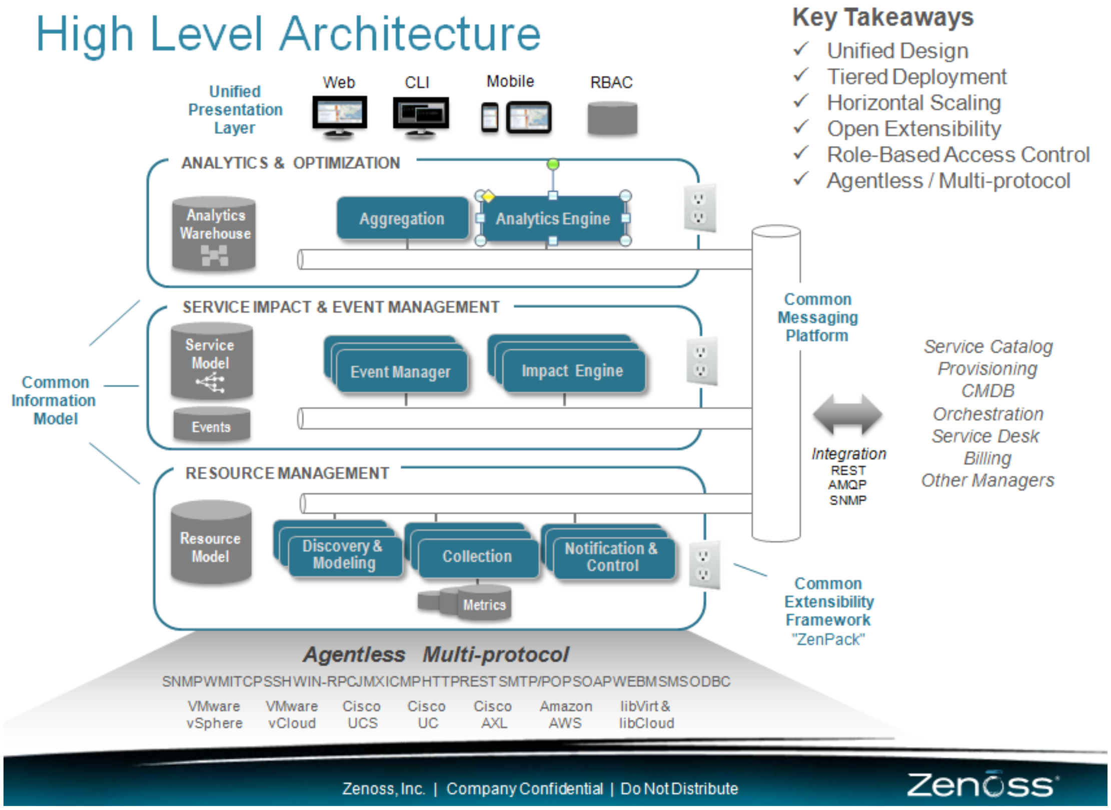
4.2 Zenoss CSA Highlights

This section highlights the key aspects of the Zenoss CSA architecture that distinguish it from other platforms. Some of the key architecture characteristics are listed below.

- **Unified design.** End-to-end service assurance and analytics capability designed from the ground up as one product on a common architecture.
- **Horizontal scaling.** Scale the deployment to manage hundreds of nodes from a single server to 100K nodes in a globally distributed configuration, leveraging low-cost hardware. Scale as needed to manage elastic infrastructure.
- **Agentless, multi-protocol.** Agentless collection and control platform that leverages a suite of secure access methods, management APIs, and synthetic transactions to instrument the full stack at scale without the need for proprietary agents.
- **Open integration and extensibility framework.** Rapidly extend, customize, and integrate with other management tools, leveraging open architecture and "ZenPack" plug-in framework. Leverage a global community of extension developers and partners.
- **Integrated RCA and SIA.** RCA is performed as a side product of the SIA, as opposed to traditional systems, which perform two functions using two different sets of rules, models, or even products. This simplifies development of customizations, as well as provides direct relationship between root cause events and service impact events caused by the root cause events.

Figure 4-4 and Figure 4-5 highlight the key aspects of the Zenoss CSA architecture.

Figure 4-4. High-Level Architecture

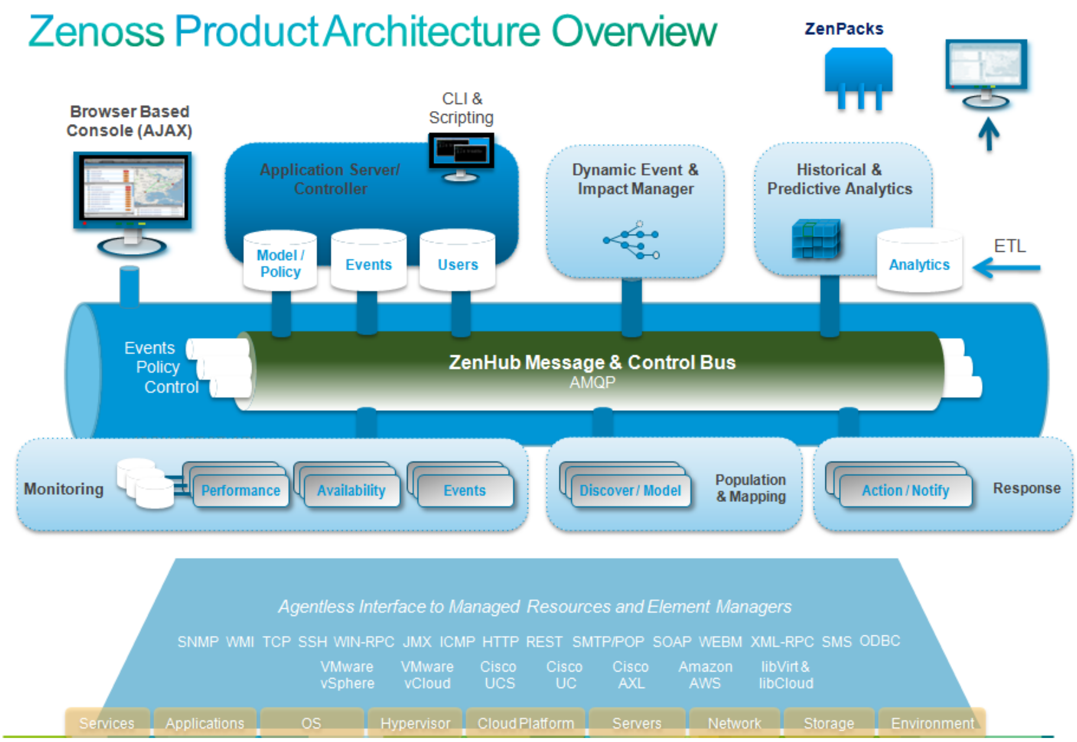


High Level Architecture

Key Takeaways

- ✓ Unified Design
- ✓ Tiered Deployment
- ✓ Horizontal Scaling
- ✓ Open Extensibility
- ✓ Role-Based Access Control
- ✓ Agentless / Multi-protocol

Figure 4-5. Zenoss Product Architecture Overview



CHAPTER 5

System Design Constraints

This chapter discusses the system design constraints of Cloud Service Assurance for Virtualized Multiservice Data Center (CLSA-VMDC) and describes the specific capabilities of the products used. This chapter presents the following topics:

- Section 5.1 System Redundancy and Availability
- Section 5.2 System Sizing and Scalability
- Section 5.3 System Deployment Models
- Section 5.4 System Maintenance and Serviceability

5.1 System Redundancy and Availability

Various redundancy deployment models and trade-offs for CLSA-VMDC are described in Section 5.3 System Deployment Models. This section describes the specific capabilities of the products used.

Zenoss Redundancy Capabilities

Zenoss Service Dynamic supports the following levels of redundancy and Disaster Recovery (DR) capabilities:

- VMware High Availability (HA) within a single Data Center (DC)
- Distributed Replicated Block Device (DRBD) and Linux active-standby HA - for latencies less than 20 msec between Zenoss components
- DRBD Proxy for latencies above 20 msec between Zenoss components

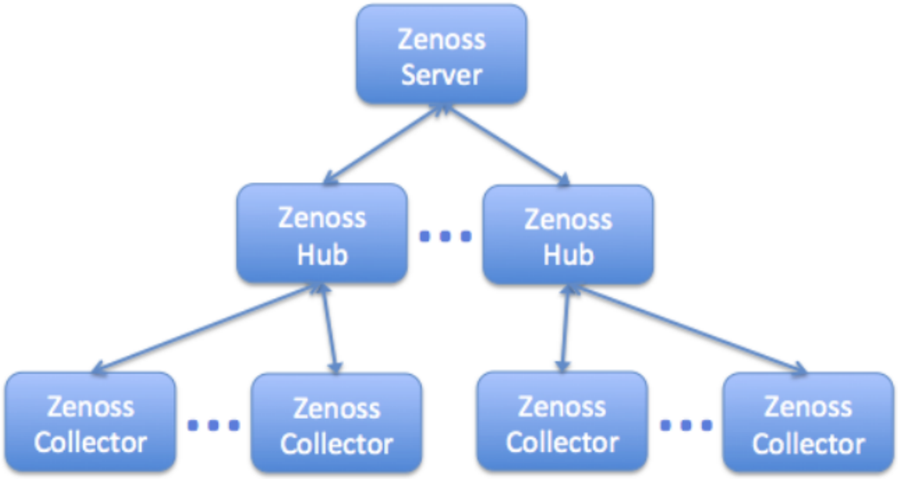
As noted in earlier sections, Zenoss Cloud Service Assurance (CSA) subcomponents can be deployed on a single Virtual Machine (VM) or they can be distributed. The following are three subcomponents that can be deployed in a distributed fashion:

- The **Zenoss Collector** is responsible for collecting data (event and performance) from managed devices.
- The **Zenoss Hub** provides an aggregation point for multiple collectors (up to 16 collectors can be deployed per hub).
- The **Zenoss Server or Master** aggregates hubs, performs analysis of data, and provides the User Interface (UI).

A distributed architecture allows for both scale growth, as well as flexibility in addressing different deployment models based on availability, security, cost, and latency requirements. Customers have complete choice on how distributed they want their Zenoss deployment to be. A minimal installation includes a Zenoss server, a hub, and a collector running on a single server or VM.

Figure 5-1 illustrates distributed deployments.

Figure 5-1. Distributed Deployment



5.1.1 High Availability Deployment Options

When planning an HA solution, implementation and operational costs must be balanced against the service level requirements of the business. How long can the system be down and how much will it cost? The officially supported HA implementation from Zenoss is Linux-based clustering using Pacemaker, Corosync, and DRBD. Cisco has verified Zenoss v4.2.3 in conjunction with Linux HA deployments, but there are other options that should be considered due to the complexity in deploying a Linux HA system. Alternative HA installation scenarios, including VMware HA, are briefly outlined in this section in order to provide a more informed view of the many available options for HA. Each HA scenario is presented along with its advantages, disadvantages, and expected recovery time.

Operator-Based Recovery

An operator-based recovery is a completely manual procedure for monitoring the health and recovery of a given system. Operators are given documented procedures to execute based on a set of guidelines, which outline the current status of the monitored system. Undocumented situations are automatically referred to second-level support staff for resolution.

Advantages

- Avoids more complex HA configuration and management.

Disadvantages

- Requires operational staff to identify system or application problems prior to executing any recovery procedures.
- Non-deterministic availability or fulfillment of more aggressive Service Level Agreement (SLA) requirements.

- Does not protect against data corruption if using shared storage.
- Would require manual data replication if not using shared storage.

VMware HA

VMware HA protects against VM hardware and Operating System (OS) level failures by supporting VM restarts in a clustered ESXi host configuration. Once failure of the active VM is detected, a new VM instance is powered on elsewhere in the cluster. For more detailed information regarding VMware HA, refer to the vSphere Availability Guide.

Advantages

- Meets more aggressive SLA requirements as the failover requires only the time to boot the VM.
- Implementation is potentially complex, but limited to VMware configuration with no third-party software.
- Recovery procedures for VM failure are automatic.

Disadvantages

- Zenoss does not support the VMware application API for application monitoring.
- Does not protect against data corruption.
- Requires operational staff to execute any recovery procedures when encountering partial VM or application failures.
- Only supported within a local data center unless implementing VMware stretched cluster configurations.

VMware FT

VMware Fault Tolerance (FT) protects against VM hardware failure by maintaining a fully-synchronized secondary VM. In the event of a primary VM failure, the secondary VM is brought online immediately, without a required boot-up or switchover requirement. For more detailed information regarding VMware FT, refer to the vSphere Availability Guide.

Advantages

- Meets more aggressive SLA requirements with a minimal downtime switchover in the event of full VM failure.
- Implementation is potentially complex, but limited to VMware configuration with no third-party software.

Disadvantages

- Zenoss does not support the VMware application API for application monitoring.
- Does not protect against data corruption.
- Requires operational staff to execute any recovery procedures when encountering partial VM or application failures.
- Application level errors are propagated to secondary device.
- Limited to a single Virtual CPU (vCPU), which could potentially limit use in larger VMDC installations.
- Only supported within a local data center.

Linux Clustering

Linux clustering uses Corosync, Pacemaker, and DRBD to maintain and monitor hardware and application health, while maintaining one or more online standby nodes. In the event of hardware or application failure, applications are relocated and started on available standby nodes. Linux clustering with DRBD is recommended for locations with connections less than 20 msec of delay between the cluster nodes.

Advantages

- The failover procedure is automated.
- Covers both application and system level failures using Corosync and Pacemaker.
- Supported deployment model for Zenoss.
- Can be used to support both VM and physical machine deployments.
- Applicable to both local data center and WAN based remote sites.

Disadvantages

- Implementation is complex.
- Requires additional third-party software and licensing.
- Does not protect against data corruption.

DRBD Proxy

DRBD Proxy allows multiple sites to maintain synchronized filesystems across network connections greater than 20 msec.

Advantages

- Supported deployment model for Zenoss over WAN connections.
- Provides synchronized data replication across WAN connections.

Disadvantages

- Requires additional third-party software and licensing.
- Does not protect against data corruption.
- Does not protect against application failures.
- Requires operational staff to execute any recovery procedures when encountering any failures.

Recovery Time Comparison

Deployment Type	Full System Recovery	Application Recovery	Tested	Local/Remote DC
Operator Based	5-10 minutes (manual)	5-10 minutes (manual)	No	Local
VMware HA	<5 min (automatic)	<5 min (manual)	Yes	Local
VMware FT	<1 min (automatic)	<1 min (manual)	No	Local
Linux Clustering	<5 min (automatic)	<5 min (automatic)	Yes	Local

Deployment Type	Full System Recovery	Application Recovery	Tested	Local/Remote DC
DRBD Proxy	5-10 minutes (manual)	5-10 minutes (manual)	Yes	Remote

Note The manual process noted in the full system and application recovery columns only covers the amount of time to physically recover the system in the event of a failure. The time from failure to recovery is actually made up of the time required for an operator to acknowledge the failure, identify the problem type, and execute any recovery procedures. Failure acknowledgement and problem identification are both variable times based on operator coverage and procedures.

5.1.2 Disaster Recovery Deployment

Disaster Recovery (DR) can be addressed in two ways. First, HA clusters can be distributed across geographically remote data centers when network latency between data centers does not exceed about 20 ms. Above that latency, instead of the DRBD continuous replication, it is recommended to use DRBD Proxy, which queues updates based on network loads. With the default Zenoss configuration, DRBD Proxy stores up to 1 Gb of updates in the event of a network outage to the remote site. Storage queue sizes may be adjusted to match expected monitoring loads.

Distributed collectors can operate independently of the Zenoss master. When a collector starts up, it requests its configuration from its hub, which means that no collector configuration is required initially other than to assign a collector to a hub. The performance data collected by a collector remains local to the collector until requested by a user for viewing in the Zenoss UI. Events are queued up within a collector for a configurable time whenever the connection to the Zenoss master is lost.

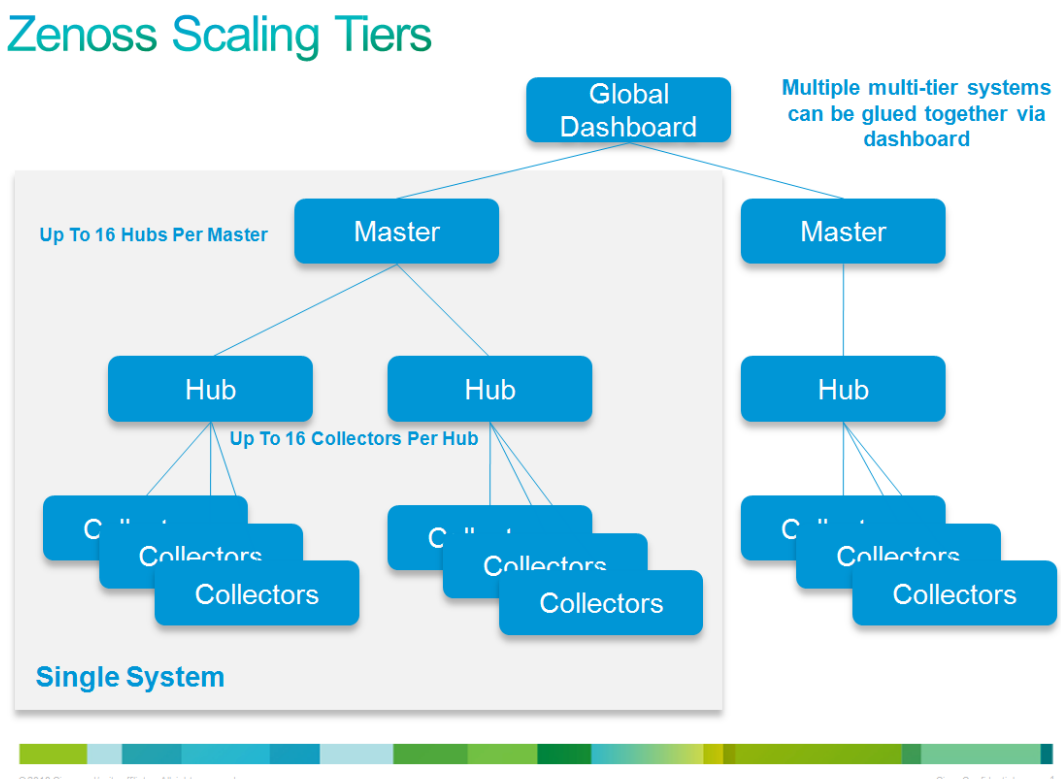
5.2 System Sizing and Scalability

Note Scalability and performance testing have not been performed by Cisco as part of this phase; all of the data in this section is based on product testing and deployment data provided by Zenoss.

Zenoss CSA can be deployed both in a centralized manner on a single VM, as well as in a distributed manner for larger scale deployments. Referring to Figure 5-2, the Zenoss CSA modular architecture allows for small initial deployments and scaling expansion at two levels:

- Within a single system via distributed collectors and hubs
- Putting multiple systems together via a global dashboard

Figure 5-2. Zenoss Scaling Tiers



Zenoss CSA recommends sizing increments for lab, Compact Pod, and Large Pod installations, in either a SP IaaS environment or an Enterprise Private Cloud environment. The sizing increments are proportional to the number of large VMDC PoDs deployed to align Zenoss CSA deployment with the VMDC growth model.

In the SP IaaS environment, a cloud provider delivers and manages at the virtual machine layer, while the cloud tenant retains responsibility for the workloads running inside the virtual machine. This is the typical mode supported by SPs and allows cloud tenants to set a security boundary where the SP has no access to tenant data and applications.

Table 5-1 provides the sizing recommendations for public cloud deployments.

Table 5-1. Zenoss Sizing Recommendations for Public Cloud Deployments

VMDC 2.2 IaaS Environment	Capacity Overview	Number of VMDC PoDs	Zenoss Master Server	Zenoss Database	Zenoss Hub	Zenoss Collector
Lab	8 servers, 180 VMs, 25 VLANs, 6 firewall contexts, 16 load balancer contexts	<1	1 VM, 8 CPU cores, 16 GB RAM, 2 NICs, 200 GB storage, Centos 6.2	—	—	1 VM, 2 CPU cores, 10 GB RAM, 2 NICs, 150 GB storage, Centos 6.2

VMDC 2.2 IaaS Environment	Capacity Overview	Number of VMDC PoDs	Zenoss Master Server	Zenoss Database	Zenoss Hub	Zenoss Collector
PoD (non-scaled)	64 servers, 1440 VMs, 180 VLANs, 6 firewall contexts, 16 load balancer contexts	1	1 VM, 8 CPU cores, 32 GB RAM, 2 NICs, 180 GB storage, Centos 6.2	1 VM, 2 CPU cores, 32 GB RAM, 2 NICs, 165 GB storage, Centos 6.2	1 VM, 8 CPU cores, 32 GB RAM, 2 NICs, 150 GB storage, Centos 6.2	2 VMs, each 2 CPU cores, 20 GB memory, 2 NICs, 150 GB storage
PoD (scaled)	512 servers, 11520 VMs, 520 VLANs, 8 firewall contexts, 24 load balancer contexts	1	1 VM, 16 CPU cores, 64 GB RAM, 2 NICs, 215 GB storage, Centos 6.2	2 VMs, each 12 CPU cores, 32 GB RAM, 2 NICs, 250 GB storage, Centos 6.2	2 VMs, each 16 CPU cores, 64 GB RAM, 2 NICs, 185 GB storage, Centos 6.2	7 VMs, each 2 CPU cores, 20 GB RAM, 2 NICs, 150 GB storage

In the Enterprise private cloud environment, a cloud provider delivers and manages at the workload level, providing management of both application workloads and the cloud infrastructure components. This is the typical mode supported in Enterprise organizations with IT sharing responsibility for application service levels with the application owner. Taking application workload management into consideration, the private cloud environment places additional load on the service assurance system and alters the recommended configuration size.

Table 5-2 highlights Zenoss sizing recommendations for private cloud deployments.

Table 5-2. Zenoss Sizing Recommendations for Private Cloud Deployments

VMDC 2.2 Installation Size	Capacity Overview	Number of VMDC PoDs	Zenoss Master Server	Zenoss Database	Zenoss Hub	Zenoss Collector
Lab	8 servers, 180 VMs, 25 VLANs, 6 firewall contexts, 16 load balancer contexts	<1	1 VM, 8 CPU cores, 16 GB RAM, 2 NICs, 200 GB storage, Centos 6.2	—	—	1 VM, 4 CPU cores, 12 GB RAM, 2 NICs, 150 GB storage, Centos 6.2
PoD (non-scaled)	64 servers, 1440 VMs, 180 VLANs, 6 firewall contexts, 16 load	1	1 VM, 8 CPU cores, 32 GB RAM, 2 NICs, 180 GB storage, Centos 6.2	1 VM, 2 CPU cores, 32 GB RAM, 2 NICs, 170 GB storage, Centos 6.2	1 VM, 8 CPU cores, 32 GB RAM, 2 NICs, 150 GB storage, Centos 6.2	2 VMs, each 4 CPU cores, 20 GB memory, 2 NICs, 150 GB storage

VMDC 2.2 Installation Size	Capacity Overview	Number of VMDC PoDs	Zenoss Master Server	Zenoss Database	Zenoss Hub	Zenoss Collector
	balancer contexts					
PoD (Scaled) Private Cloud	512 servers, 11520 VMs, 11520 Windows & Linux OS, 520 VLANs, 8 firewall contexts, 24 load balancer contexts	1	1 VM, 24 CPU cores, 64 GB RAM, 2 NICs, 215 GB storage, Centos 6.2	2 VMs, each 16 CPU cores, 32 GB RAM, 2 NICs, 300 GB storage, Centos 6.2	2 VMs, each 16 CPU cores, 64 GB RAM, 2 NICs, 185 GB storage, Centos 6.2	14 VMs, each 4 CPU cores, 20 GB RAM, 2 NICs, 150 GB storage

Zenoss CSA can be configured to support larger installations. Detailed configuration architecture planning services, in consultation with Zenoss, are strongly recommended for all production installations.

5.3 System Deployment Models

This section discusses the various deployment models of CLSA-VMDC. Section 5.3.1 Business and Operational Deployment Models lists various business and operational deployment models possible by CLSA-VMDC, while Section 5.3.2 CLSA-VMDC Deployment Overview provides an overview of the location, IP, and security deployment model used for CLSA-VMDC.

5.3.1 Business and Operational Deployment Models

The following are the targeted business and operation deployment model options supported by CLSA-VMDC:

- Private clouds owned and managed by Enterprises
- Virtual private cloud where some business entity (SP) owns and manages customer/tenant, service, and infrastructure:
  - Assumption is that the IP Next Generation Network (NGN) network is managed by a different assurance system
  - Same operator for all VMDC infrastructure layers (compute, storage, and network)
  - Different operators for:
    - Service
    - Compute
    - Storage
    - Network

5.3.2 CLSA-VMDC Deployment Model

Some of the key benefits of CLSA-VMDC, as well as the Hosted Collaboration Solution (HCS) system in which CLSA-VMDC is targeted to be integrated, are:

- Support for deployment of tenant's applications in geo-redundant data centers to support application clustering. This implies that a single tenant may have VMs in multiple data centers, and these VMs are part of the same application Active/Active clusters.
- Support for various levels of redundancy, including:
  - **VMware HA (Simplex SA deployment)** is typically used for intra-DC failures. VMware HA results in assurance service outage, and for maximum service availability, application level HA is required in addition to VMware HA. However, for deployments where cost is the key concern, VMware HA without application HA may be the appropriate choice.
  - **Disaster Recovery (DR)** is used for periodic backups of the service assurance system, so that the system can be recovered in case of complete DC outages. Typically, recovery is not in real time, so some data loss and assurance service outage should be expected.
  - **SA application HA** provides real-time service assurance system recovery and assurance service continuity. When deployed in geo-redundant date centers, latency constraints are typically more stringent than in case of DR.
- Capability to offer maximum visibility by segregating the managed system from the management system. This is especially important for assurance systems in order to detect outages and to correctly measure system availability and Service Level Agreement (SLA). If the monitoring system is not segregated from the managed system, then it is not able to detect certain failures of the managed system as the assurance system could experience outages at the same time. Segregation can be done on multiple levels:
  - **DC level (will refer to as remote assurance system)**. The assurance system is located in a different DC from the VMDC components that it is monitoring. This can be implemented by placing the assurance system in the dedicated management DC, or can be implemented using assurance system application level redundancy with a node in DC1 monitoring components in DC2, and the assurance node in DC2 monitoring VMDC components in DC1; both assurance systems have complete service level visibility. Segregation on the DC level ensures full visibility in any managed system failure, including complete DC failure, and allows SLA measurements for DC outages. This is the preferred deployment model for many Service Providers (SPs), especially those offering more traditional services such as voice where availability expectations are typically very high. For example, the HCS system has a requirement to support this model. In this model, latency constraints between the managed system and the assurance system need to be considered. Note that in the case of the remote assurance deployment model, one may split up components of the assurance system such that the data collector layer is placed in the same DC as the managed system, and the rest of the assurance system is remote, which typically reduces latency constraints.
  - **PoD level (will refer to as colocated assurance system).** The assurance system is located in an access PoD that is dedicated to management applications. In this model, some managed system failures in core and aggregation layers may not be detected as the management system itself may temporarily lose connectivity. In this deployment model, in order to maintain maximum possible visibility, even in the face of failures that could affect the assurance system itself, the assurance system needs to be deployed with application level redundancy (clustering for example).
  - The **Optional remote deployment model** can be extended with geo-redundancy support for VMDC assurance components, where Service Assurance Manager (SAM) redundancy is extended between management PoDs in geo-graphically distributed data centers. In

this case, any latency constraints between components of the assurance system must be considered. There are two options for geo-redundant deployment of the assurance system:

- **Colocated.** The assurance system is placed in the same data centers as the managed system. For example, a service assurance system that monitors two data centers (A and B) has two service assurance application nodes placed one in each DC, A and B.
- **Remote.** The assurance system is placed in two data centers that are different than data centers where the managed system resides. These data centers are sometimes referred to as Network Operations Center (NOC) data centers.

- **Secure network communication between tenant dedicated components that are being monitored and the shared management assurance system.** This is typically implemented by deploying extranet VPNs in which the assurance system would reside and firewalls between assurance products and tenant VPNs.
- **Support for monitoring in NAT-ed environment.** Tenant components in their private VPN could have overlapping IPs, and in fact, in some systems such as HCS, it is a preferred deployment model to use the same IP address for given application of each tenant. In such deployments, there are two common configurations:
  - The use of a dedicated management interface/address for each monitored component. For example, second vNIC on each VM or a management loopback interface on routers.
  - The use of static NAT between monitored devices and the assurance system. While it is recommended to use a dedicated management interface since some of the assurance functions can be lost when crossing the NAT boundary, in some deployments, use of NAT cannot be avoided. For example, some applications such as Cisco UCS do not support the use of a second vNIC.

In addition to meeting the above availability and visibility related requirements, there are additional constraints related to cost and latency sensitivity of individual products that influences the design for each specific deployment. Table 5-3 defines a few potential deployment models based on availability, cost, visibility, and latency requirements and constraints. All of these deployment models are supported, but not all are validated as part of CLSA-VMDC.

Note For more information on specific SAM product redundancy capabilities, see Section 5.4 System Maintenance and Serviceability.

Table 5-3. Potential Deployment Models

Deployment Model Name	Availability	Cost	Visibility	Latency Tolerance
Colocated Simplex	Lower	Lower	Medium	Higher
Colocated HA	Medium	Higher	Medium	Higher
Colocated geo-redundant DR	Medium	Higher	Medium	Medium
Colocated geo-redundant HA	Higher	Higher	Medium	Lower
Remote simplex	Lower	Lower	Higher	Higher
Remote HA	Medium	Higher	Medium	Higher

Deployment Model Name	Availability	Cost	Visibility	Latency Tolerance
Remote geo-redundant DR	Medium	Higher	Higher	Medium
Remote geo-redundant HA	Higher	Higher	Higher	Lower

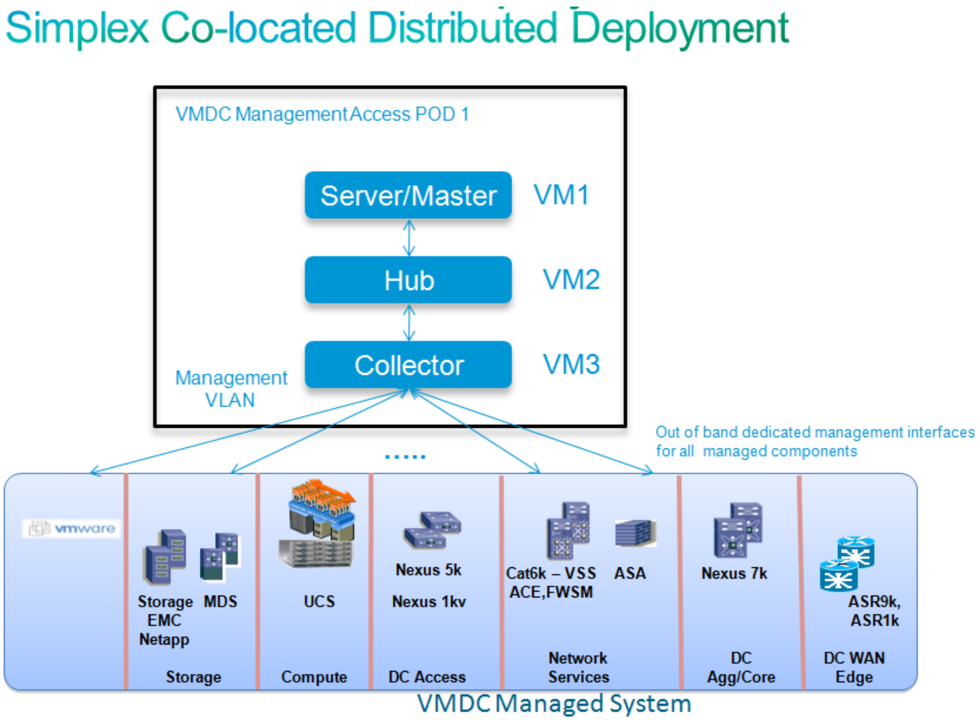
The following deployment models are validated for CLSA-VMDC:

- **Colocated simplex deployment model with dedicated management interfaces per VM.** Dedicated management interface per VM with assurance system deployed in dedicated management access PoD. No application level redundancy is deployed, and VMware HA is only redundancy leveraged for the assurance system. This is the model similar to the current implementation for other management applications in VMDC Systems.
- **Colocated HA using Linux HA clustering and DRBD.** Dedicated management interface per VM with assurance system deployed in dedicated management access PoD. Both system level and application level redundancy is deployed within this system.
- **Colocated geo-redundant DR using DRBD Proxy for remote DR site synchronization.** Dedicated management access for assurance system. No dedicated interfaces between local deployment and remote DR site. No application level redundancy is deployed.
- **The use of static NAT** between monitored devices and the assurance system. Limited validation was performed against some networking and storage components via a NAT interface.

5.3.3 Zenoss CSA Deployment Model

Figure 5-3 illustrates the Zenoss deployment model for CLSA-VMDC.

Figure 5-3. Zenoss Deployment Model for CLSA-VMDC



Section 5.4 System Maintenance and Serviceability provides more information on the HA capabilities of Zenoss CSA. This section only outlines one validated deployment model.

For functional testing, it is sufficient to deploy Zenoss in a centralized fashion on a single VM. However, Zenoss should be deployed in distributed fashion with separate VMs for collector, hub, and master in order to allow for growth by adding more collectors as more devices are added. All three VMs should be deployed in a single, dedicated VLAN and placed in the management access PoD. Each managed component, including VMs, should use a dedicated out-of-band management interface. Optionally, a few Zenoss VMs with a single interface and NAT between them could be deployed. Single VM (master, hub, collector) and split server (VM with master/hub and VM with a collector) deployments were validated by Cisco SDU.

5.4 System Maintenance and Serviceability

This section discusses the system maintenance and serviceability constraints for CLSA-VMDC.

Upgrades:

- The user must install/upgrade ZenPacks using the OS command line interface. Refer to Installing ZenPacks and Removing ZenPacks for more information.
- When the Zenoss master is configured in an HA cluster configuration with network-level volume replication (DRBD), upgrades are not service impacting.
- If Zenoss servers are virtualized, then snapshots can be used to limit the service impact of upgrades.

Backup and Restore:

- zenbackup and zenrestore tools perform consolidated backups of the MySQL databases, RRD files, and Neo4j database. Refer to the Zenoss Cloud Service Assurance Installation and Administration Guide for more information.
- Backup and restore backs up the following data:
  - Events database
  - Zope database, which includes all devices, users, and event mappings
  - $ZENHOME/etc directory, which contains configuration files for the system daemons
  - $ZENHOME/perf directory, which contains performance data
- Full Zenoss system restore requires a re-installation of the Zenoss software and a zenrestore from the backup file created by zenbackup. This includes code customizations done as part of ZenPacks.
- VMware snapshots can also be used for backup and restore.
- See Backup and Recovery for more information on backup and restore procedures.

CHAPTER 6

Operational Use Cases

Cloud infrastructures present a challenging environment for both cloud Service Providers and Enterprises to assure both highly available and high performance services. By design, the Cisco Virtualized Multiservice Data Center (VMDC) is a physically and logically redundant, virtualized cloud infrastructure. This design avoids single points of failure and provides High Availability (HA). If a fault should occur on a VMDC component, then it is expected that services using that component will continue to operate seamlessly or recover quickly by using an alternate network path or compute host. However, until the fault is repaired, the VMDC service is now "at risk" of a longer duration outage after any subsequent fault. Cloud Service Assurance for VMDC (CLSA-VMDC) allows cloud operators to increase their awareness and resolution of this "at risk" condition to reduce preventable outages and reduce Mean Time to Repair (MTTR).

CLSA-VMDC provides cloud service personnel with different responsibilities the tools to perform timely service assurance tasks. Assurance administrative personnel are concerned with ensuring the correct operation of the assurance system and with adding and removing new or at-fault devices to maintain an accurate representation of the cloud infrastructure. Assurance operations personnel are tasked with the daily operations of monitoring, fault identification and notification. CLSA-VMDC enables both sets of activities for cloud service personnel.

This chapter illustrates the following six cloud service personnel use cases:

1. Section 6.1 Discovering Data Center (DC) Devices and Components. Once CLSA-VMDC is installed, the next administrative task is discovering all of the VMDC components. While discovery is an initial system deployment task, this use case is included here because discovery is an ongoing activity as the cloud infrastructure changes over time. Assurance administrative personnel need to discover new devices following replacement, upgrades and VMDC cloud infrastructure expansion.
2. Section 6.2 Dashboard Monitoring. Cloud operations personnel daily monitor the cloud by observation of the aggregated dashboard. This proverbial "single pane of glass" serves as the summary window for the entire cloud service status. The use case demonstrates that the system wide dashboard is both customizable and navigable. When a device fault occurs, cloud operators can quickly drill down from the dashboard to investigate the issue. This use case also highlights several other dashboards and their features.
3. Section 6.3 Consolidated Resource Monitoring of VMDC Components. Once discovered, each cloud infrastructure device accumulates a history of Key Performance Indicator (KPI) statistics. Cloud operators can browse these statistics to assess the performance of devices and their components. The graphical presentation also allows the operator to visually identify troublesome trends or cyclic behavior to proactively manage resources. This use case illustrates that as events occur, these KPI graphs can be consulted to identify and replace those resources at risk of failure, thus maintaining a high level of service availability.

4. Section 6.4 Tenant Service Impact Analysis (SIA) and Root Cause Analysis (RCA). One of the challenges of cloud service assurance is to manage the presentation of information to assurance personnel that represents the complexity of the environment without information overload. Consider a potential event storm when a fault occurs in just the right spot in the cloud. This use case illustrates an approach to service assurance that presents tenant compute services graphically. The state of each node in the service graph is dependent on its own state and the state of its children. The state of the entire Service Impact graph is derived by the "bubbling up" of the state calculations of each node in the graph. Event storms need not be deciphered by operators because these tenant compute service graphs simplify the problem domain.
5. Section 6.5 Identifying Availability Impact of Shared Resource Faults on Tenant Services. Cloud infrastructure faults may or may not impact the services riding over the network and compute resources. Service availability impact depends on the severity, proximity and confluence of fault events. Some events such as a redundant link failure should have no discernible impact on tenant services. Other more severe fault events such as a sudden Cisco Unified Computing System (UCS) chassis failure would impact service availability. With compute cluster redundancy, the availability impact of this more severe type of fault can be minimized. This use case discusses various shared infrastructure faults in general and then walks the reader through a chassis failure fault.
6. Section 6.6 Identifying Performance Problems Before They Impact Tenants. In addition to service availability impact, service assurance must also be concerned with service quality. Dependent on the competing demands of tenant services, components of the cloud infrastructure may experience higher than average loads on individual resources such as CPU, memory and storage. Cloud operators can manage these kinds of quality issues by monitoring tenant service performance impact. This final use case highlights service performance health monitoring by using KPI thresholds. Device KPI threshold configuration is described and two scenarios are shown for both Virtual Machine (VM) and ESXi host CPU degradation.

6.1 Discovering DC Devices and Components

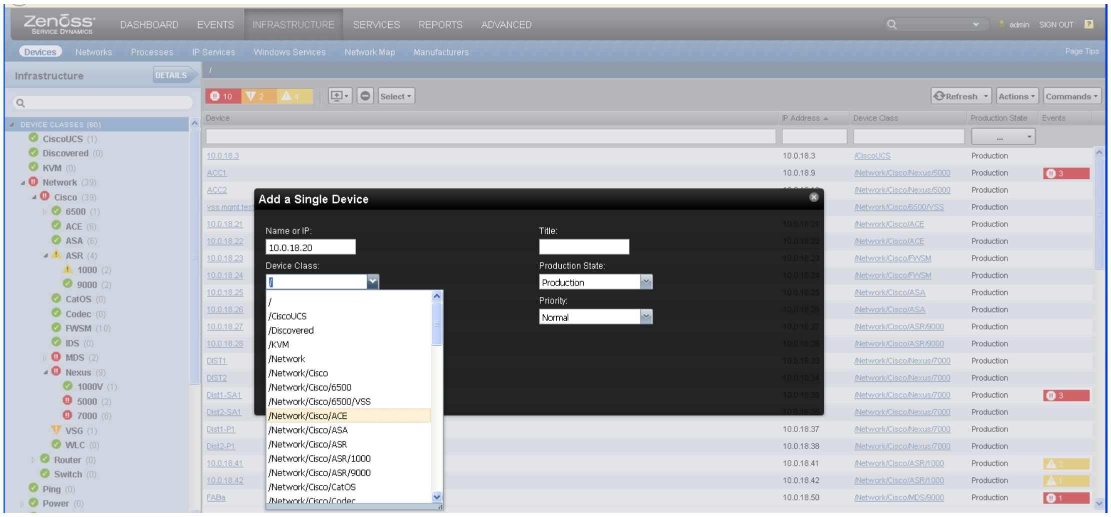
This use case illustrates multiple ways to use Zenoss CSA to discover individual VMDC components. Successful device discovery will first verify device reachability and then autoclassify the device into its proper device class. The first two sections show how to discover a single device via IP address and how to discover multiple devices by supplying a subnet IP address and a subnet mask. The next section shows a brief illustration of multiple device discovery using a text file approach and the *zenbatchload* utility. The final two sections are special cases illustrating how to discover a Cisco UCS compute infrastructure and a VMware vSphere virtual environment. EMC VMAX and VNX storage are introduced with this release, and CLSA-VMDC and EMC discovery is presented separately in Section 7.1.3 Adding and Monitoring EMC Arrays Within Zenoss .

Note See Section # Zenoss CSA Southbound Integration for the VMDC managed device configuration requirements.

6.1.1 Add a Single Device

The following use case describes how to manually discover a single VMDC device with Zenoss CSA. To add a single device, highlight the topmost device class **Devices** under the Infrastructure management tab. Click the + icon and choose **Add a Single Device...** From this popup window, the user can specify the device IP address or DNS name and the device name to be used within Zenoss. Select the device class. Click **Add.** After the device is modeled, the device appears under the device class selected.

Figure 6-1. Adding Single Device



Add a Single Device

Name or IP: 10.0.18.20

Device Class: /

- /
- /CiscoUCS
- /Discovered
- /KVM
- /Network
- /Network/Cisco
- /Network/Cisco/6500
- /Network/Cisco/6500/VSS
- /Network/Cisco/ACE
- /Network/Cisco/ASA
- /Network/Cisco/ASR
- /Network/Cisco/ASR/1000
- /Network/Cisco/ASR/9000
- /Network/Cisco/CatOS
- /Network/Cisco/Codec

Title:

Production State: Production

Priority: Normal

Device	IP Address	Device Class	Production State	Events
10.0.18.3	10.0.18.3	/CiscoUCS	Production	
ACC1	10.0.18.9	/Network/Cisco/Nexus/5000	Production	3
ACC2		/Network/Cisco/Nexus/5000	Production	
vss.mgmt.test		/Network/Cisco/6500/VSS	Production	
10.0.18.21	10.0.18.21	/Network/Cisco/ACE	Production	
10.0.18.22	10.0.18.22	/Network/Cisco/ACE	Production	
10.0.18.23	10.0.18.23	/Network/Cisco/FWSM	Production	
10.0.18.24	10.0.18.24	/Network/Cisco/FWSM	Production	
10.0.18.25	10.0.18.25	/Network/Cisco/ASA	Production	
10.0.18.26	10.0.18.26	/Network/Cisco/ASA	Production	
10.0.18.27	10.0.18.27	/Network/Cisco/ASR/9000	Production	
10.0.18.28	10.0.18.28	/Network/Cisco/ASR/9000	Production	
DIST1	10.0.18.33	/Network/Cisco/Nexus/7000	Production	
DIST2	10.0.18.34	/Network/Cisco/Nexus/7000	Production	
Dist1-SA1	10.0.18.35	/Network/Cisco/Nexus/7000	Production	3
Dist2-SA1	10.0.18.36	/Network/Cisco/Nexus/7000	Production	
Dist1-P1	10.0.18.37	/Network/Cisco/Nexus/7000	Production	
Dist2-P1	10.0.18.38	/Network/Cisco/Nexus/7000	Production	
10.0.18.41	10.0.18.41	/Network/Cisco/ASR/1000	Production	⚠ 2
10.0.18.42	10.0.18.42	/Network/Cisco/ASR/1000	Production	⚠ 1
FABa	10.0.18.50	/Network/Cisco/MDS/9000	Production	1

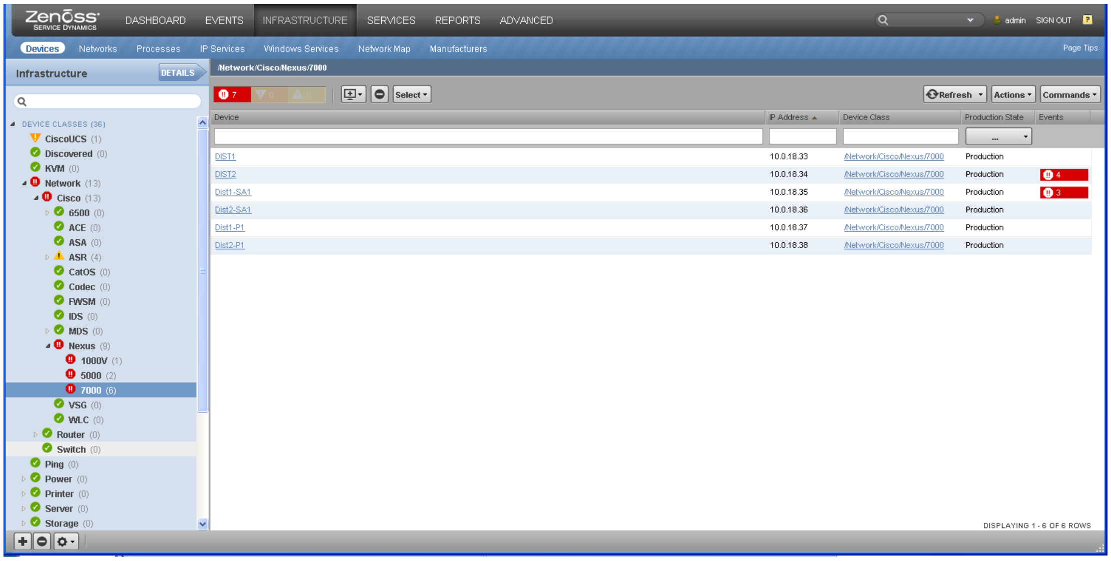
Adding a Single Device with Multiple Contexts

Some VMDC devices include virtual contexts and must be treated differently. A context for a given device is an independent device being maintained within another independent device. For example, a Nexus 7000 may have several Virtual Device Contexts (VDC). Devices that have multiple virtual contexts need to have all of the contexts added as separate devices. This would require that every context have a management IP address accessible by the Zenoss server, plus any SNMP configurations as stated earlier.

Example - Adding a Nexus 7000 with Multiple Virtual Device Contexts

To discover a Nexus 7000 with VDCs, ensure that all of the contexts have an IP address reachable from the Zenoss server. All of the management IPs need to be resolvable either from DNS or from a local hosts file. Each context needs to be added separately. After everything is discovered, the VDCs show up under the admin context view. From this view, the user can access all of the VDCs. The VDCs also appear as separate devices under the Infrastructure tab. For example, if there were two Nexus 7000s configured with three VDCs, they would appear as six Nexus 7000s under the Infrastructure tab.

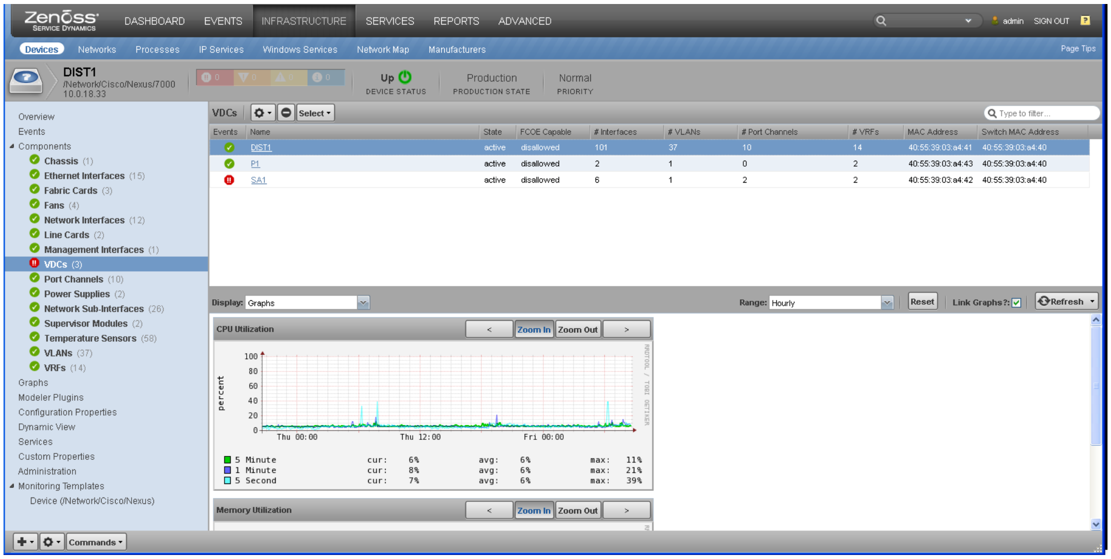
Figure 6-2. Two Nexus 7000s with Three VDCs



Device	IP Address	Device Class	Production State	Events
DIST1	10.0.18.33	/Network/Cisco/Nexus/7000	Production	
DIST2	10.0.18.34	/Network/Cisco/Nexus/7000	Production	4
Dist1-SA1	10.0.18.35	/Network/Cisco/Nexus/7000	Production	3
Dist2-SA1	10.0.18.36	/Network/Cisco/Nexus/7000	Production	
Dist1-P1	10.0.18.37	/Network/Cisco/Nexus/7000	Production	
Dist2-P1	10.0.18.38	/Network/Cisco/Nexus/7000	Production	

Clicking on one of the admin contexts opens a view where all of the VDCs can be accessed.

Figure 6-3. Nexus 7000 Admin Context

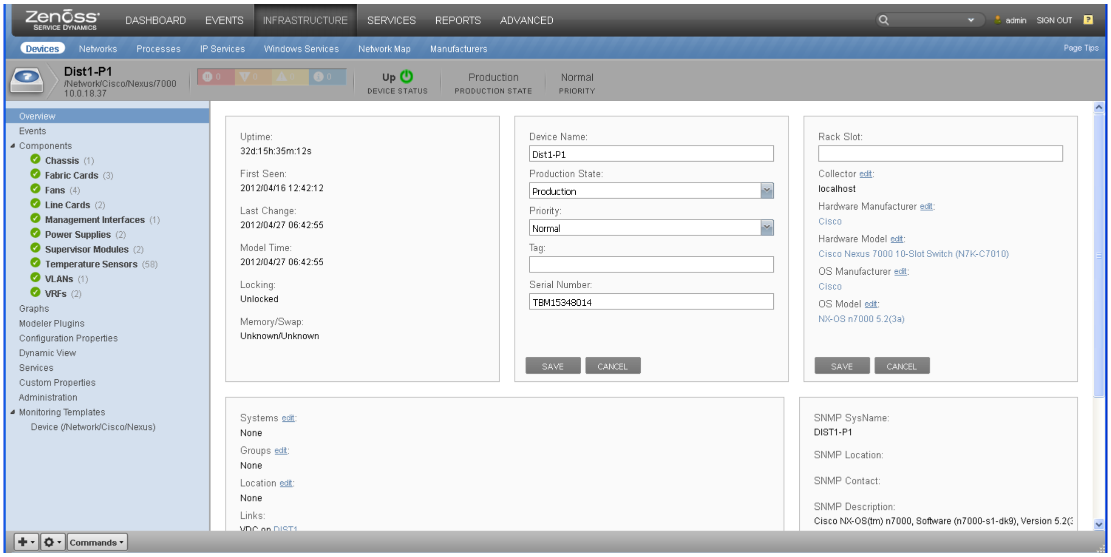


Events	Name	State	FCOE Capable	# Interfaces	# VLANs	# Port Channels	# VRFs	MAC Address	Switch MAC Address
✓	DIST1	active	disallowed	101	37	10	14	40:55:39:03:a4:41	40:55:39:03:a4:40
✓	P1	active	disallowed	2	1	0	2	40:55:39:03:a4:43	40:55:39:03:a4:40
!	SA1	active	disallowed	6	1	2	2	40:55:39:03:a4:42	40:55:39:03:a4:40

```
5 Minute    cur:   6%    avg:   6%    max:  11%
1 Minute    cur:   8%    avg:   6%    max:  21%
5 Second    cur:   7%    avg:   6%    max:  39%
```

Clicking on one of the VDCs opens the VDC's component view.

Figure 6-4. Nexus 7000 VDC Component View



Dist1-P1
/Network/Cisco/Nexus/7000
10.0.18.37

Up DEVICE STATUS
Production PRODUCTION STATE
Normal PRIORITY

Uptime: 32d:15h:35m:12s
First Seen: 2012/04/16 12:42:12
Last Change: 2012/04/27 06:42:55
Model Time: 2012/04/27 06:42:55
Locking: Unlocked
Memory/Swap: Unknown/Unknown

Device Name: Dist1-P1
Production State: Production
Priority: Normal
Tag:
Serial Number: TBM15348014

Rack Slot:
Collector edit: localhost
Hardware Manufacturer edit: Cisco
Hardware Model edit: Cisco Nexus 7000 10-Slot Switch (N7K-C7010)
OS Manufacturer edit: Cisco
OS Model edit: NX-OS n7000 5.2(3a)

Systems edit: None
Groups edit: None
Location edit: None
Links: VDC on DIST1

SNMP SysName: DIST1-P1
SNMP Location:
SNMP Contact:
SNMP Description: Cisco NX-OS(tm) n7000, Software (n7000-s1-dk9), Version 5.2(3

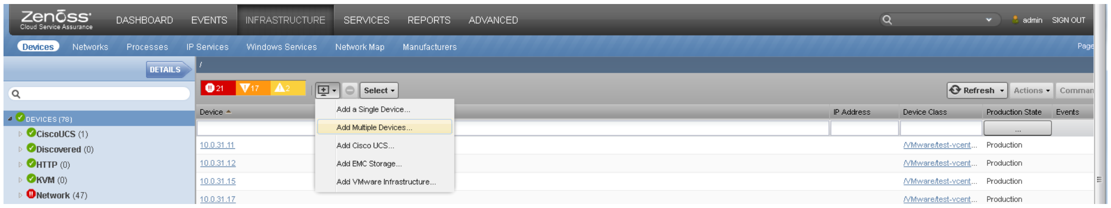
6.1.2 Add Multiple Devices

The following use case describes how to add multiple devices manually or through automatic discovery.

Adding Multiple Devices Manually

Under the Infrastructure management tab in the tool, click the **plus sign** icon and choose **Add Multiple Devices...**

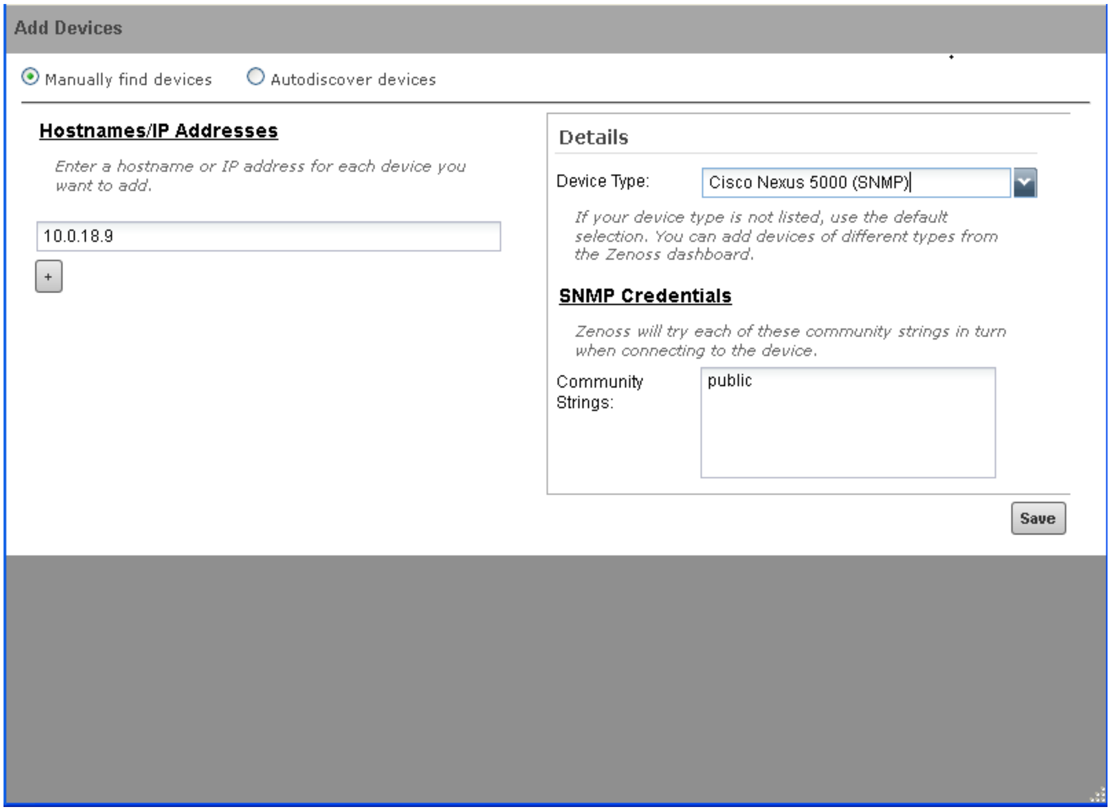
Figure 6-5. Add Multiple Devices



Device	IP Address	Device Class	Production State	Events
10.0.31.11		/VMware/test-vcent...	Production	
10.0.31.12		/VMware/test-vcent...	Production	
10.0.31.15		/VMware/test-vcent...	Production	
10.0.31.17		/VMware/test-vcent...	Production	

From this popup window, add the first device by entering the device IP address or DNS name and add the details such as device class.

Figure 6-6. Add First Device



Add Devices

● Manually find devices ○ Autodiscover devices

Hostnames/IP Addresses

Enter a hostname or IP address for each device you want to add.

10.0.18.9

Details

Device Type: Cisco Nexus 5000 (SNMP)

If your device type is not listed, use the default selection. You can add devices of different types from the Zenoss dashboard.

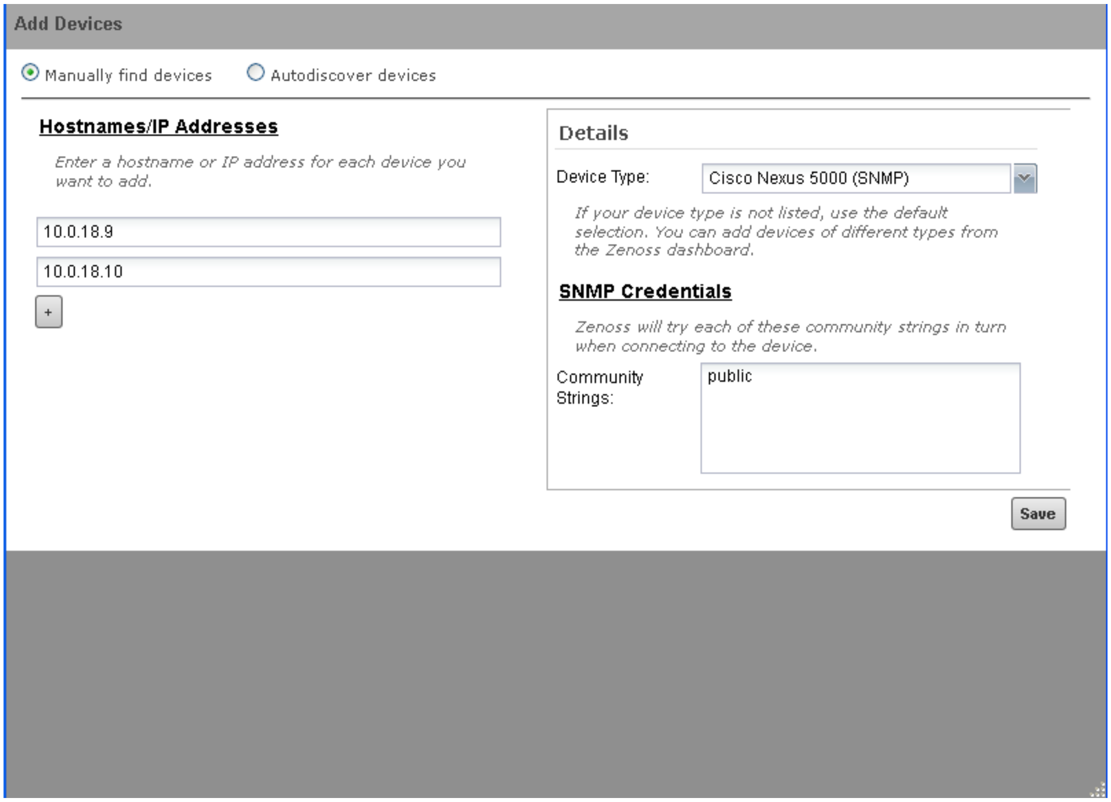
SNMP Credentials

Zenoss will try each of these community strings in turn when connecting to the device.

Community Strings: public

Click the **plus sign** icon under the IP address or DNS name entered to add another device. Notice that the details need to be filled in again for this device. This allows for the entering of multiple different types of devices.

Figure 6-7. Add Second Device



Add Devices

● Manually find devices ○ Autodiscover devices

Hostnames/IP Addresses

Enter a hostname or IP address for each device you want to add.

10.0.18.9

10.0.18.10

Details

Device Type: Cisco Nexus 5000 (SNMP)

If your device type is not listed, use the default selection. You can add devices of different types from the Zenoss dashboard.

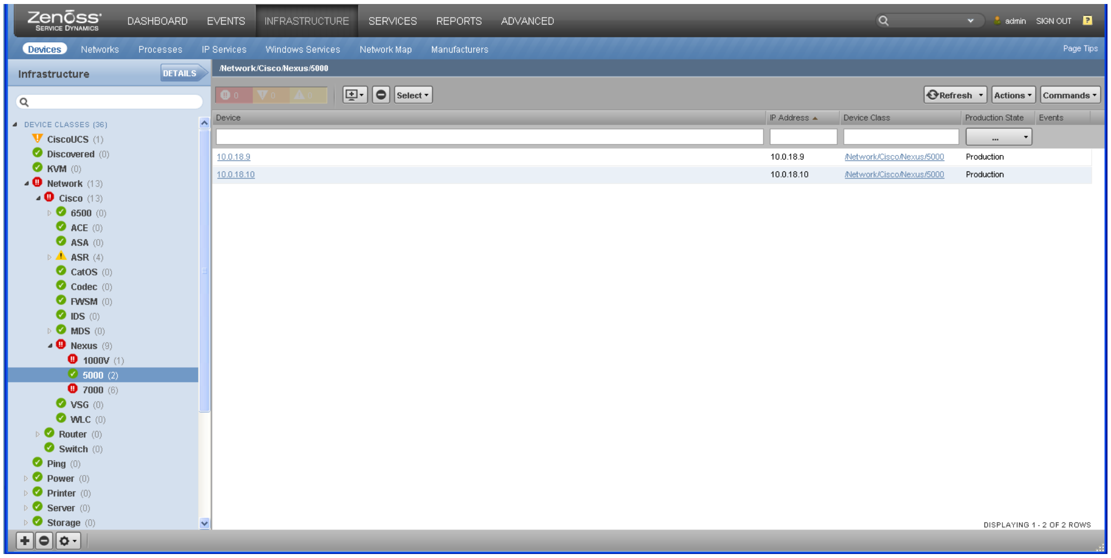
SNMP Credentials

Zenoss will try each of these community strings in turn when connecting to the device.

Community Strings: public

When finished, click the **Save** button. Both devices are now added.

Figure 6-8. Two Devices Added



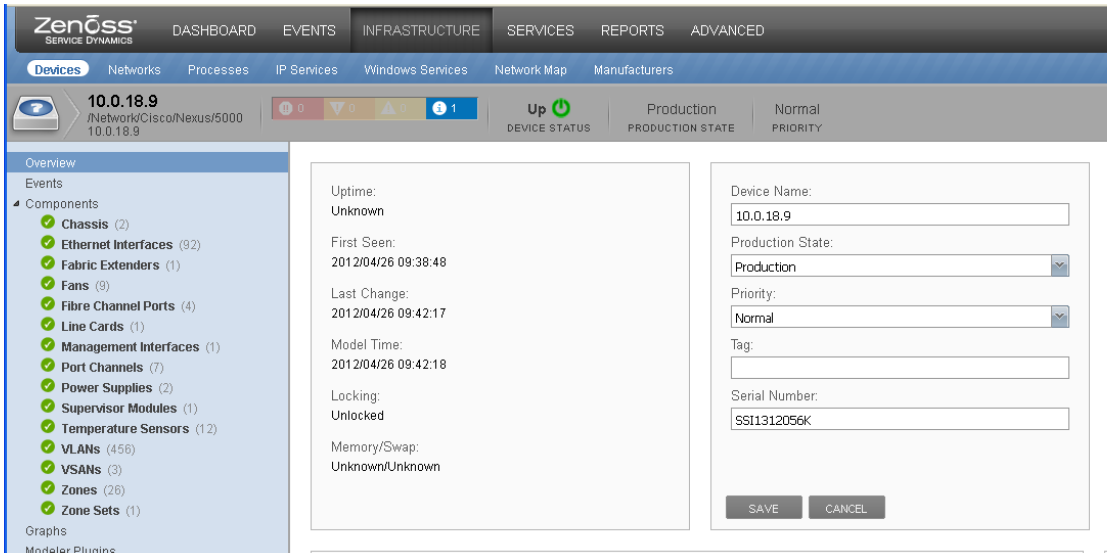
/Network/Cisco/Nexus/5000

Device	IP Address	Device Class	Production State	Events
10.0.18.9	10.0.18.9	/Network/Cisco/Nexus/5000	Production	
10.0.18.10	10.0.18.10	/Network/Cisco/Nexus/5000	Production	

DISPLAYING 1 - 2 OF 2 ROWS

The device name can be changed later if an IP address was used to discover the device.

Figure 6-9. Change Name



10.0.18.9
/Network/Cisco/Nexus/5000
10.0.18.9

0 0 0 1

Up DEVICE STATUS

Production PRODUCTION STATE

Normal PRIORITY

Uptime:
Unknown

First Seen:
2012/04/26 09:38:48

Last Change:
2012/04/26 09:42:17

Model Time:
2012/04/26 09:42:18

Locking:
Unlocked

Memory/Swap:
Unknown/Unknown

Device Name: 10.0.18.9

Production State: Production

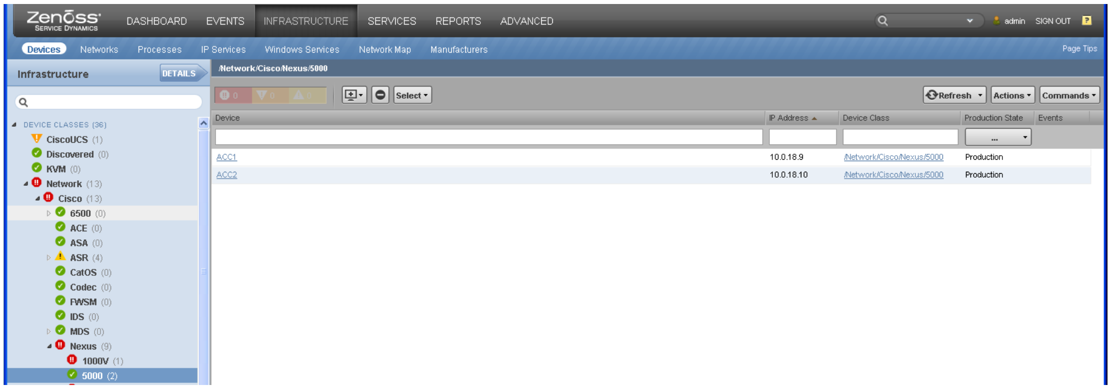
Priority: Normal

Tag:

Serial Number: SSI1312056K

The name is changed on both devices.

Figure 6-10. Name Changed



/Network/Cisco/Nexus/5000

Device	IP Address	Device Class	Production State	Events
ACC1	10.0.18.9	/Network/Cisco/Nexus/5000	Production	
ACC2	10.0.18.10	/Network/Cisco/Nexus/5000	Production	

Adding Multiple Devices Through Automatic Discovery

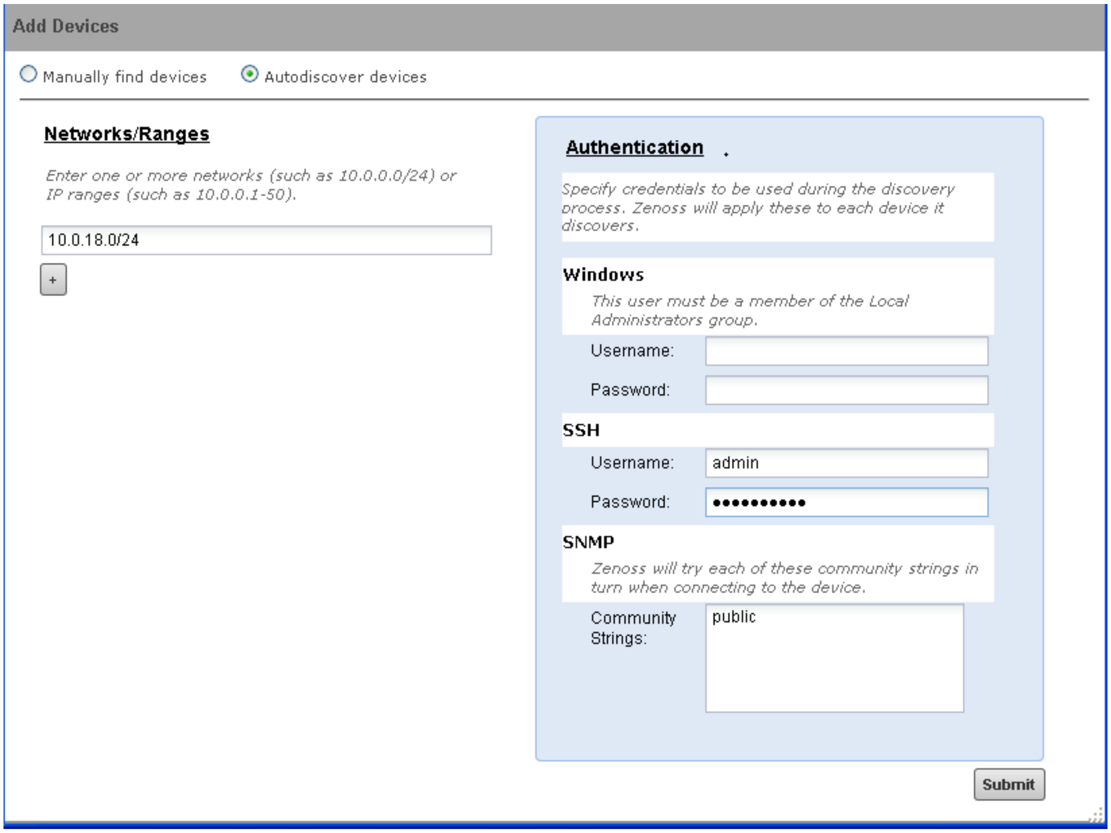
The automatic discovery process attempts to discover devices on the network using either a network subnet or an IP range. During this process, Zenoss also attempts to automatically classify devices found on the network.

Note Using this process on a subnet with more than a few devices is not recommended. Based on the range of the network being searched and the number of active devices on the searched network, this process can take several hours to complete. See the zenbatchload process for more efficient bulk configuration of Zenoss.

To ensure that Zenoss properly discovers and classifies devices, the devices need to be configured for remote access.

Under the Infrastructure management tab in the tool, click the **plus sign** icon and choose **Add Multiple Devices...** In this popup window, there are two options: manually find devices and autodiscover devices. Select the **Autodiscovery** option. Enter the network IP address range for the devices to be discovered and add any additional information needed. If the Global Configuration Properties have been edited, then additional information is not needed. Additional network subnets can be autodiscovered by clicking on the **plus sign** icon below the Network/Address ranges. When finished, click the **Submit** button.

Figure 6-11. Autodiscover Devices



Add Devices

○ Manually find devices ● Autodiscover devices

Networks/Ranges

Enter one or more networks (such as 10.0.0.0/24) or IP ranges (such as 10.0.0.1-50).

10.0.18.0/24

Authentication

Specify credentials to be used during the discovery process. Zenoss will apply these to each device it discovers.

Windows

This user must be a member of the Local Administrators group.

Username:

Password:

SSH

Username: admin

Password: ●●●●●●●●●●

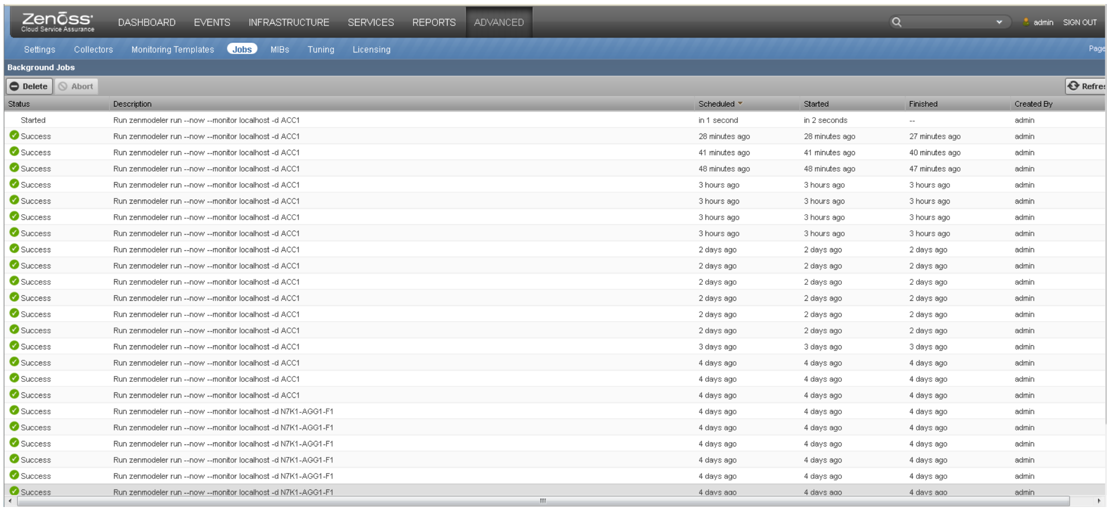
SNMP

Zenoss will try each of these community strings in turn when connecting to the device.

Community Strings: public

As the devices are being discovered, they start to populate the Discovered device class and are displayed in the default window. To view the status of the discovery process, click the **Advanced** tab and select **Jobs** from the row on the top. The initial discovery job should be seen either in the running state or finished. After a device has been identified, Zenoss launches another process to classify the device discovered.

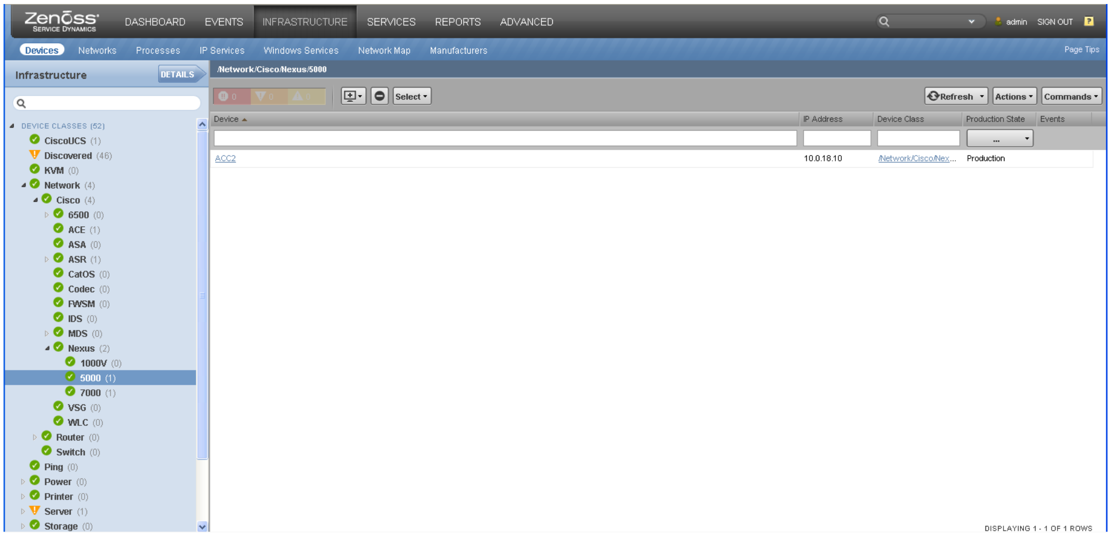
Figure 6-12. Discovery Status



Status	Description	Scheduled	Started	Finished	Created By
Started	Run zenmodeler run --now --monitor localhost -d ACC1	in 1 second	in 2 seconds	--	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	28 minutes ago	28 minutes ago	27 minutes ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	41 minutes ago	41 minutes ago	40 minutes ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	48 minutes ago	48 minutes ago	47 minutes ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	3 hours ago	3 hours ago	3 hours ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	3 hours ago	3 hours ago	3 hours ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	3 hours ago	3 hours ago	3 hours ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	3 hours ago	3 hours ago	3 hours ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	2 days ago	2 days ago	2 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	2 days ago	2 days ago	2 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	2 days ago	2 days ago	2 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	2 days ago	2 days ago	2 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	2 days ago	2 days ago	2 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	2 days ago	2 days ago	2 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	3 days ago	3 days ago	3 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d ACC1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d N7K1-AGG1-F1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d N7K1-AGG1-F1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d N7K1-AGG1-F1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d N7K1-AGG1-F1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d N7K1-AGG1-F1	4 days ago	4 days ago	4 days ago	admin
✓ Success	Run zenmodeler run --now --monitor localhost -d N7K1-AGG1-F1	4 days ago	4 days ago	4 days ago	admin

As devices are discovered and classified, they are placed under the proper device class.

Figure 6-13. As Devices are Discovered

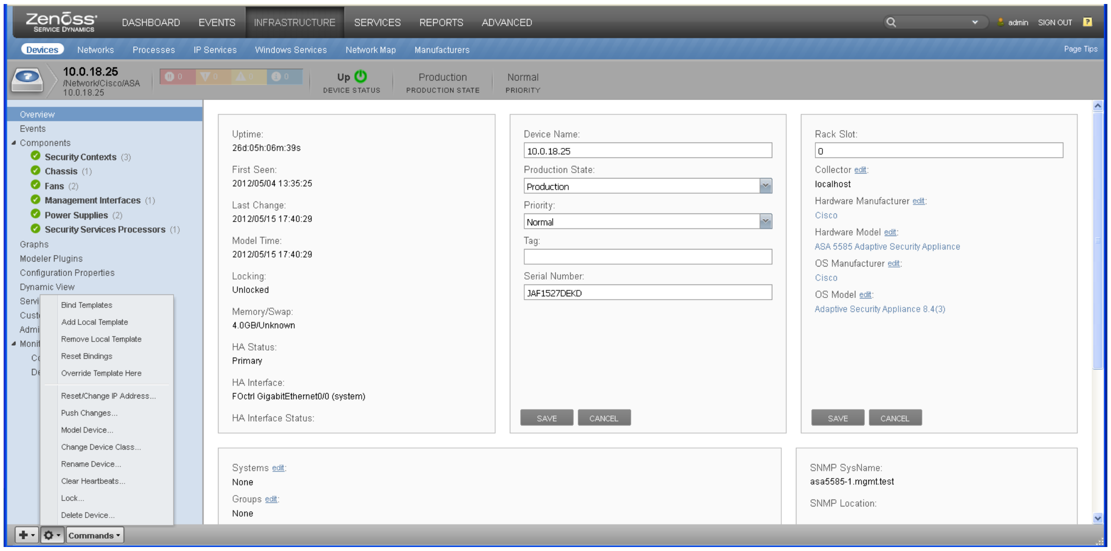


Device	IP Address	Device Class	Production State	Events
ACC2	10.0.18.10	/Network/Cisco/Nex...	Production	

The time required to complete the discovery process depends on the number of devices. To determine if the discovery process has completed, check to ensure that all of the jobs under the Advanced tab have succeeded and that there are no processes still pending. There could still be devices in the Discovered device class. Some of the devices do not need to be monitored. On devices that do not need to be monitored, select all of the devices that are not needed. Select the **-** icon from the top of the window. Click **Remove** when prompted to remove devices.

On devices that are discovered and classified, ensure that the devices have the necessary configuration. On some devices like the ASA and FWSM, autodiscovery fails if the Zenoss server is not added as an allowable snmp-server. After editing the device configuration to allow access for the Zenoss server, drag the device from the Discovered window into the correct device class. Select all of the IPs associated with the device if it was a device with multiple contexts. Click **OK** when prompted to move the device. Click **OK** on the Remodeled Required warning. Go to the device class of the device that was moved and select one of the devices. In the case of a device with multiple contexts, select the IP address of the the admin context. On the bottom left of the screen, click the sprocket icon and select **Model Device.**

Figure 6-14. Model Device



10.0.18.25
/Network/Cisco/ASA
10.0.18.25

Up DEVICE STATUS
Production PRODUCTION STATE
Normal PRIORITY

Uptime:
26d:05h:06m:39s

First Seen:
2012/05/04 13:35:25

Last Change:
2012/05/15 17:40:29

Model Time:
2012/05/15 17:40:29

Locking:
Unlocked

Memory/Swap:
4.0GB/Unknown

HA Status:
Primary

HA Interface:
FOctrl GigabitEthernet0/0 (system)

HA Interface Status:

Device Name:
10.0.18.25

Production State:
Production

Priority:
Normal

Tag:

Serial Number:
JAF1527DEKD

Rack Slot:
0

Collector edit:
localhost

Hardware Manufacturer edit:
Cisco

Hardware Model edit:
ASA 5585 Adaptive Security Appliance

OS Manufacturer edit:
Cisco

OS Model edit:
Adaptive Security Appliance 8.4(3)

Systems edit:
None

Groups edit:
None

SNMP SysName:
asa5585-1.mgmt.test

SNMP Location:

Virtual IP Address Mapping

For devices such as the UCS or Catalyst 6500 Service Modules like the FWSM and ASASM, Virtual IP (VIP) addresses are recommended to be used as the monitoring point. During Zenoss autodiscovery of a subnet that contains a device with a VIP address, the associated physical addresses are also discovered and modeled, leading to duplicate entries for the same device. These duplicate device entries can be manually removed later. One method to prevent this condition from occurring is to configure an ACL at the most optimal location to block SNMP communication between the Zenoss server and the physical IP addresses. This ACL only allows communication with VIP addresses and hence, unique device discovery.

6.1.3 Add Multiple Devices with the zenbatchload Utility

The *zenbatchload* utility is the preferred method for adding multiple devices into Zenoss CSA. The *zenbatchload* utility adds devices specified in a text file. The following description provides the basic steps for using *zenbatchload*:

1. Log in to the Zenoss CSA server as the administrative user zenoss.
2. Create a configuration text file to describe the devices to be discovered.
3. Execute the *zenbatchload* command against that configuration file.

Table 6-1 lists some of the most useful *zenbatchload* commands.

Table 6-1. List of zenbatchload Commands

Command	Purpose
zenbatchload mydevicelist.txt	Executes zenbatchload against a device configuration file to begin device discovery

Command	Purpose
zenbatchload --help	Shows a list of zenbatchload commands
zenbatchload --sample_configs	Shows an example device configuration file
zenbatchload --nocommit mydevicelist.txt	Verifies configuration file syntax only

A sample configuration file is provided below.

```
# VMware vSphere
/Devices/VMware loader='vmware', loader_arg_keys=['id', 'host', 'username',
 'password', 'useSsl', 'collector']
mgmt-vcenter id='mgmt-vcenter', host='mgmt-vcenter', username='Administrator',
 password='device-passwd', useSsl=True, collector='localhost'
prod-vcenter id='prod-vcenter', host='prod-vcenter', username='Administrator',
 password='device-passwd', useSsl=True, collector='localhost'

# UCS Managers
/Devices/CiscoUCS loader='ciscoucs', loader_arg_keys=['host', 'username', 'password',
 'port', 'useSsl', 'collector']
UCS host='UCS', username='admin-user', password='device-passwd', port=80,
 useSsl=False, collector='localhost'

# Catalyst 6500
/Devices/Network/Cisco/6500 zSnmpCommunity='public', zCommandUsername='admin-user',
 zCommandPassword='device-passwd'
6500-1

# Catalyst 6500 VSS
/Devices/Network/Cisco/6500/VSS zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
VSS

# ACE
/Devices/Network/Cisco/ACE zSnmpCommunity='public', zCommandUsername='admin-user',
 zCommandPassword='device-passwd'
ACE-1
ACE-2
cfcf510a60c1400eab004ec9149e6a3d
8e66ef78fa574f538001ef3b8a7e7482

# FWSM
/Devices/Network/Cisco/FWSM zSnmpCommunity='public', zCommandUsername='admin-user',
 zCommandPassword='device-passwd'
FWSM-Slot-4-A
FWSM-Slot-4-B
Pall-CreateContainer-VFW-A
Pall-CreateContainer-VFW-B
Palladium-Container-VFW-A
Palladium-Container-VFW-B
Palladium-Tarheels-VFW-A
Palladium-Tarheels-VFW-B
Palladium-test-FWLB-VFW-A
Palladium-test-FWLB-VFW-B

# ASA & ASASM
/Devices/Network/Cisco/ASA zSnmpCommunity='public', zCommandUsername='admin-user',
 zCommandPassword='device-passwd'
asa5585-1-A
asa5585-1-B
cust-1-A
cust-1-B
cust-2-A
cust-2-B
ASA-SLOT-5-A
ASA-SLOT-5-B
b7a5a6268bf44cbab5c4d63f7e168ffe-A
```

```
b7a5a6268bf44cbab5c4d63f7e168ffe-B

# Nexus 7000
/Devices/Network/Cisco/Nexus/7000 zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
DIST1
DIST2
DIST1-SA1
DIST2-SA1
DIST1-P1
DIST2-P1
N7K1
N7K1-AGG1-M1
N7K1-AGG1-F1

# Nexus 5000/2000
/Devices/Network/Cisco/Nexus/5000 zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
ACC1
ACC2

# Nexus 3000
/Devices/Network/Cisco/Nexus/3000  zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
N3k-1

# Nexus 1000V
/Devices/Network/Cisco/Nexus/1000V zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
N1kv-Orch

# VSG
/Devices/Network/Cisco/VSG zSnmpCommunity='public', zCommandUsername='admin-user',
 zCommandPassword='device-passwd'
rowg-vsg2

# ASR 9000
/Devices/Network/Cisco/ASR/9000 zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
ASR9K-PE3
ASR9K-PE4

# ASR 1000
/Devices/Network/Cisco/ASR/1000 zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
ASR1K-PE1
ASR1K-PE2

# MDS 9000
/Devices/Network/Cisco/MDS/9000 zSnmpCommunity='public', zCommandUsername='admin-
user', zCommandPassword='device-passwd'
FABa
FABb

# SMI-S Provider serving EMC VNX storage arrays
/Devices/Storage/EMC/VNX smi-s-vnx zWBEMPassword='smis-
passwd',zWBEMPort='5990',zWBEMUseSSL='False',zWBEMUsername='admin-user'

# SMI-S Provider serving EMC VMAX storage arrays
/Devices/Storage/EMC/VMAX smi-s-vmax zWBEMPassword='smis-
passwd',zWBEMPort='15988',zWBEMUseSSL='False',zWBEMUsername='admin-user'
```

Note The SNMP default version is SNMPv2c. If SNMPv3 is preferred, then other zProperties may be added to the device class templates in the example above. For example zSnmpVer='v3', zSnmpAuthPassword='authpw', zSnmpAuthType='authtype', zSnmpPrivPassword='privpasswd', zSnmpPrivType='privtype', zSNMPSecurityName='secname'.

6.1.4 Add a UCS Domain

The following use case describes how to configure a read-only user in UCS and how to add a UCS domain to Zenoss CSA. A read-only user can be added by either creating a local user with a read-only role, or by using a Lightweight Directory Access Protocol (LDAP) or a Terminal Access Controller Access Control System (TACACS) to set the correct role. The read-only role is a predefined role on the UCS.

Note If no role is defined for a user, then it is automatically put in the read-only role.

Creating a Local Read-Only User

Below is an example of how to set up a local user with read-only privileges via the UCS CLI.

```
UCS-A# scope security
UCS-A#/security # create local-user zenossuser
UCS-A#/security/local-user *# set password clsapassword
UCS-A#/security # exit
UCS-A#/security # commit
```

Note A reference on how to set up authentication on UCS can be found in the UCS Configuring Role Based Access Control Guide. There are also configuration guides for setting up LDAP and TACACS authentication. In addition, the Cisco Support Community UCS Forum is a good reference to find helpful tips on setting up authentication on UCS.

Zenoss uses XML API to access information from UCS. XML API is enabled by default on a UCS Domain. To verify that XML API works, log in to `https://<ucs manager-ip>/visore.html`. Log in using the Zenoss credentials. If no error is seen, then Zenoss should not have any problems monitoring the UCS.

Discovering a UCS Domain

To add a UCS domain to the discovered Devices section, select the **Add Cisco UCS...** option from the Action menu.

Enter the UCS login information. Zenoss CSA discovers a UCS domain with or without SSL. Selecting the SSL check box ensures that the connection uses SSL. Without SSL enabled, the default connection port is 80. With SSL enabled, the default connection port number is 443. Connection ports can be changed if needed.

Figure 6-15. Add UCS Option

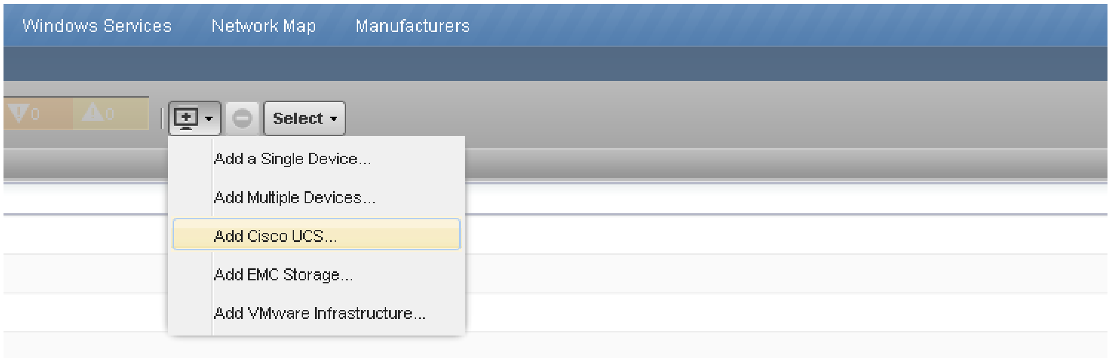
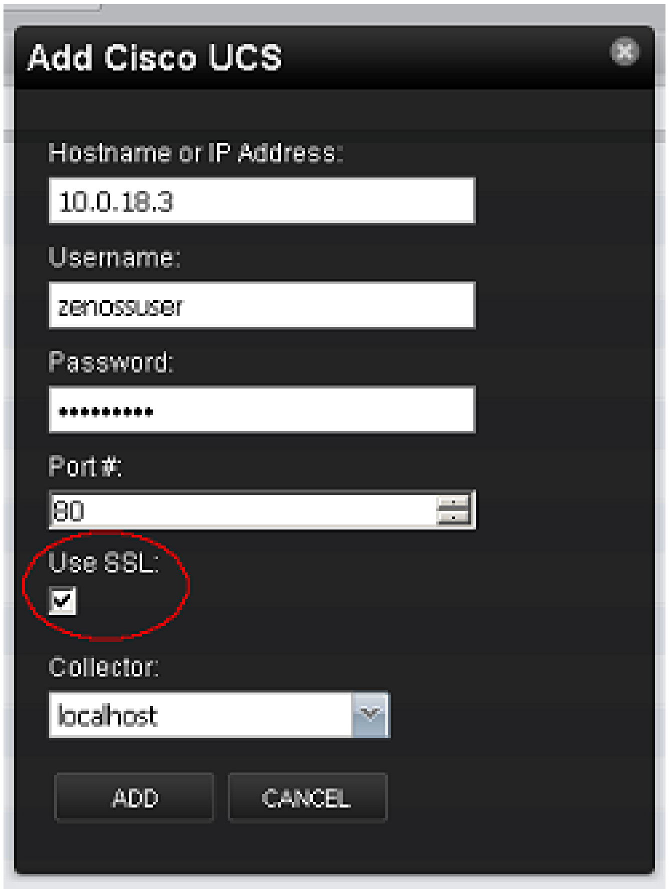


Figure 6-16. Enter UCS Login Information



Add Cisco UCS

Hostname or IP Address: 10.0.18.3

Username: zenossuser

Password: *********

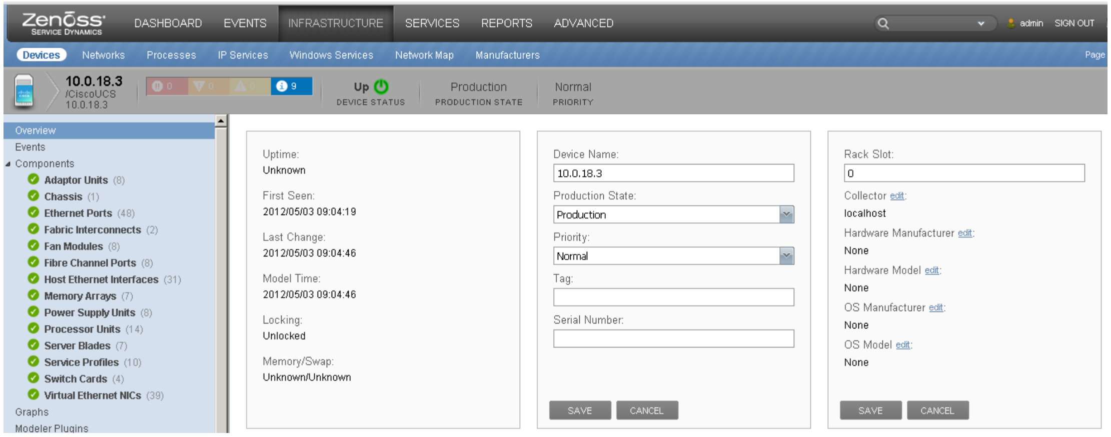
Port #: 80

Use SSL: ☑

Collector: localhost

Figure 6-17 provides a view of the UCS components discovered and monitored by Zenoss.

Figure 6-17. UCS Component View



10.0.18.3
/CiscoUCS
10.0.18.3

Up
DEVICE STATUS

Production
PRODUCTION STATE

Normal
PRIORITY

Overview
Events
Components
- Adaptor Units (8)
- Chassis (1)
- Ethernet Ports (48)
- Fabric Interconnects (2)
- Fan Modules (8)
- Fibre Channel Ports (8)
- Host Ethernet Interfaces (31)
- Memory Arrays (7)
- Power Supply Units (8)
- Processor Units (14)
- Server Blades (7)
- Service Profiles (10)
- Switch Cards (4)
- Virtual Ethernet NICs (39)

Graphs
Modeler Plugins

Uptime:
Unknown

First Seen:
2012/05/03 09:04:19

Last Change:
2012/05/03 09:04:46

Model Time:
2012/05/03 09:04:46

Locking:
Unlocked

Memory/Swap:
Unknown/Unknown

Device Name:
10.0.18.3

Production State:
Production

Priority:
Normal

Tag:

Serial Number:

Rack Slot:
0

Collector edit:
localhost

Hardware Manufacturer edit:
None

Hardware Model edit:
None

OS Manufacturer edit:
None

OS Model edit:
None

6.1.5 Add VMware vSphere Management Domain

The following use case describes how to configure a read-only vSphere user and how to add a vSphere domain to Zenoss CSA.

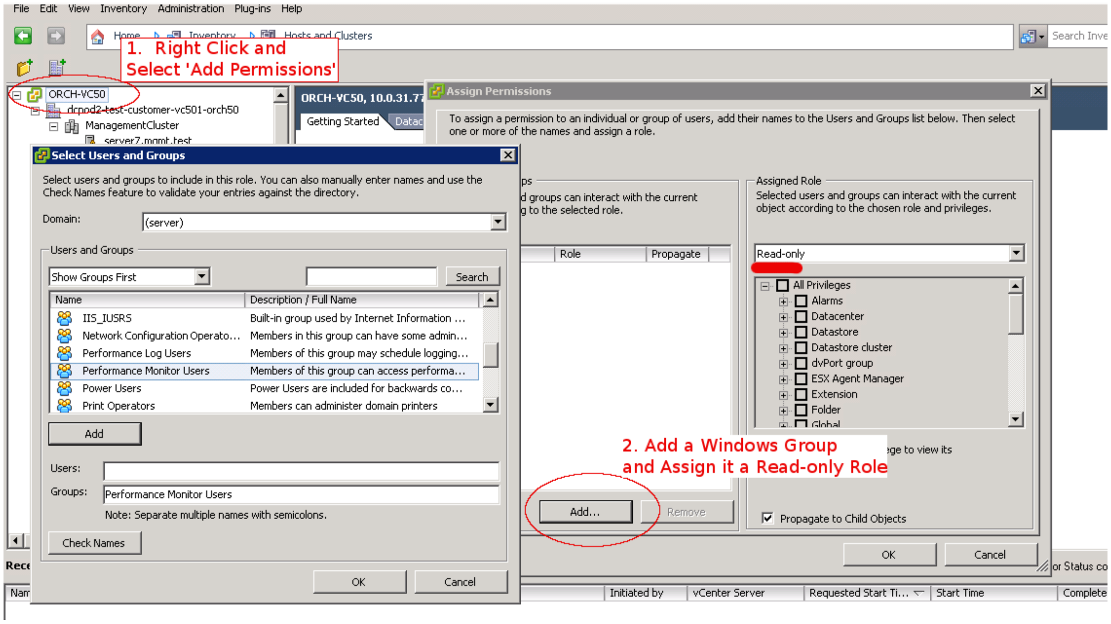
Enabling a vSphere Read-Only User

vSphere domain discovery requires a vSphere user with a read-only role, similar to the UCS domain discovery process. To add a Windows group or user with read-only privileges to the vSphere data center, perform the following steps:

1. Select the top most node to apply the authorization to in the vSphere host and cluster view.
2. Right-click the node and select **Add Permissions.** By default, the read-only role is selected.
3. Click **Add** to apply a Windows user or group to the read-only role.

In Figure 6-18, the Performance User group is added to the read-only role on an entire vSphere management domain.

Figure 6-18. Create a vSphere Read-only Group or User



1. Right Click and Select 'Add Permissions'

Select Users and Groups

Select users and groups to include in this role. You can also manually enter names and use the Check Names feature to validate your entries against the directory.

Domain: (server)

Users and Groups

Show Groups First   Search

Name	Description / Full Name
IIS_IUSRS	Built-in group used by Internet Information ...
Network Configuration Operato...	Members in this group can have some admin...
Performance Log Users	Members of this group may schedule logging...
Performance Monitor Users	Members of this group can access performa...
Power Users	Power Users are included for backwards co...
Print Operators	Members can administer domain printers

Add

Users:

Groups: Performance Monitor Users

Note: Separate multiple names with semicolons.

Check Names   OK   Cancel

Assign Permissions

To assign a permission to an individual or group of users, add their names to the Users and Groups list below. Then select one or more of the names and assign a role.

ps

d groups can interact with the current g to the selected role.

Role   Propagate

Add...   Remove

Assigned Role

Selected users and groups can interact with the current object according to the chosen role and privileges.

Read-only

- □ All Privileges
  - □ Alarms
  - □ Datacenter
  - □ Datastore
  - □ Datastore cluster
  - □ dvPort group
  - □ ESX Agent Manager
  - □ Extension
  - □ Folder
  - □ Global

ge to view its

☑ Propagate to Child Objects

OK   Cancel

2. Add a Windows Group and Assign it a Read-only Role

Discovering a vSphere Domain

Once a read-only user or group is created on vSphere, add a vSphere management domain to the Zenoss Devices section. Select the **Add VMware Infrastructure** option. Enter the hostname, username, and password information to manage the vSphere server.

Figure 6-19. Add VMware Infrastructure

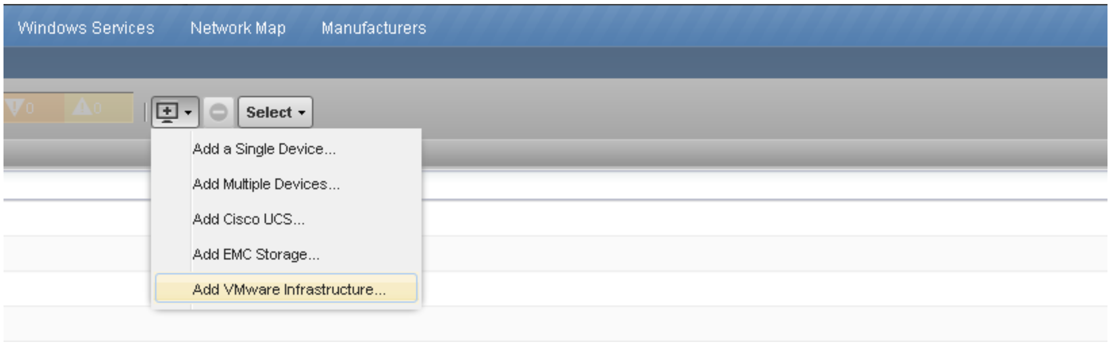
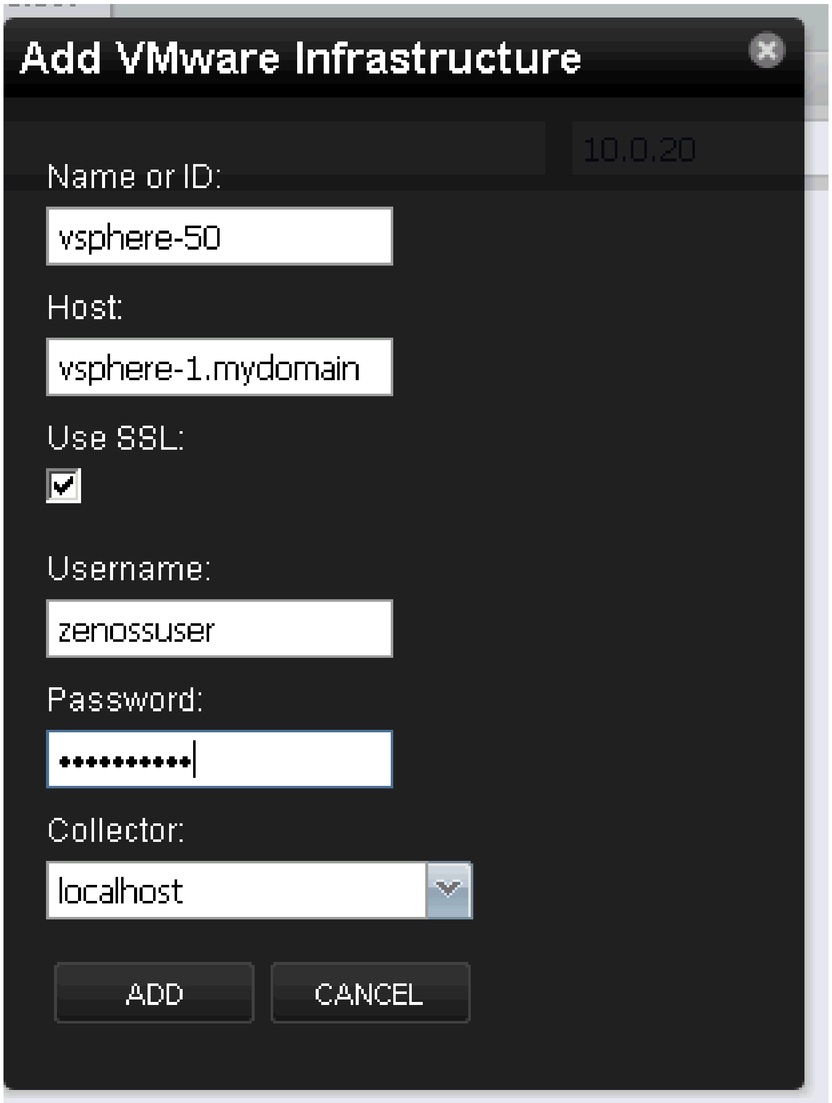


Figure 6-20. Add VMware Infrastructure Details



Add VMware Infrastructure

Name or ID: vsphere-50

Host: vsphere-1.mydomain

Use SSL: ☑

Username: zenossuser

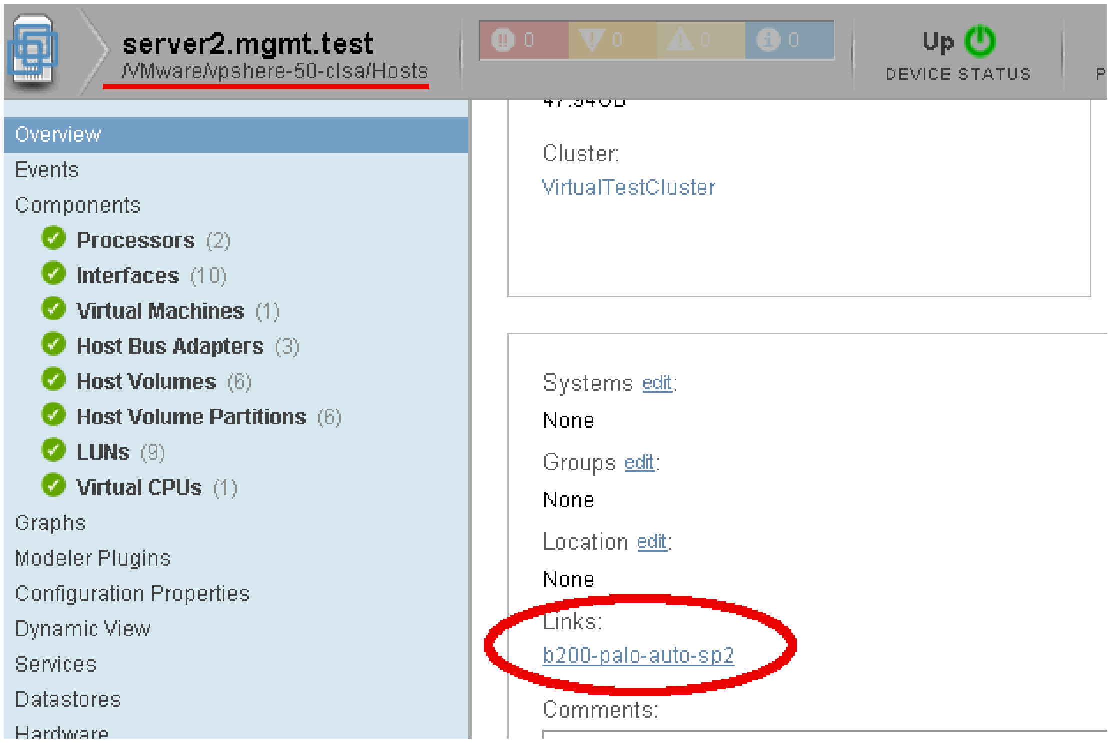
Password: •••••••••••

Collector: localhost

It takes a few minutes for Zenoss to discover and populate the VMware device class views. To view the progress, select **Advanced > Jobs.**

In addition to showing the standard components of a vSphere domain, Zenoss can link the UCS service profile with the associated vSphere ESXi host. To enable this functionality, ensure that the UCS is added to the Zenoss Device infrastructure before the vSphere domain is added.

Figure 6-21. Link Between vSphere ESXi Host and UCS Profile in Zenoss View



server2.mgmt.test
/VMware/vpshere-50-clsa/Hosts

Cluster:
VirtualTestCluster

Systems edit:
None

Groups edit:
None

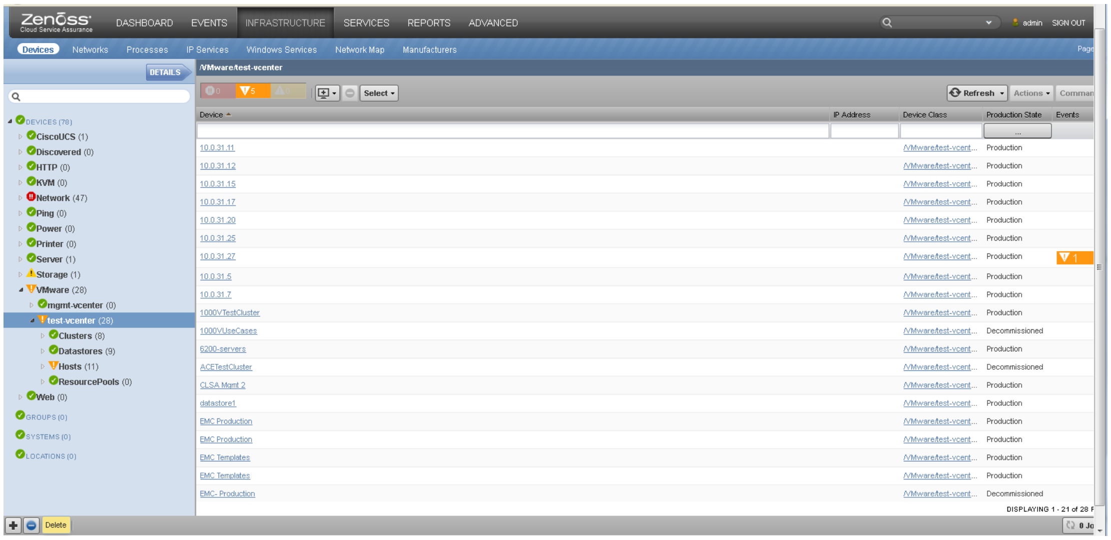
Location edit:
None

Links:
b200-palo-auto-sp2

Comments:

To delete the VMware vSphere management domain and of all its components, click the **Delete** option in the bottom left corner of the screen.

Figure 6-22. Delete VMware Domain Option



/VMware/test-vcenter

Device	IP Address	Device Class	Production State	Events
10.0.31.11		/VMware/test-vcent...	Production	
10.0.31.12		/VMware/test-vcent...	Production	
10.0.31.15		/VMware/test-vcent...	Production	
10.0.31.17		/VMware/test-vcent...	Production	
10.0.31.20		/VMware/test-vcent...	Production	
10.0.31.25		/VMware/test-vcent...	Production	
10.0.31.27		/VMware/test-vcent...	Production	1
10.0.31.5		/VMware/test-vcent...	Production	
10.0.31.7		/VMware/test-vcent...	Production	
1000VTestCluster		/VMware/test-vcent...	Production	
1000VUseCases		/VMware/test-vcent...	Decommissioned	
6200-servers		/VMware/test-vcent...	Production	
ACETestCluster		/VMware/test-vcent...	Decommissioned	
CLSA Mgmt 2		/VMware/test-vcent...	Production	
datastore1		/VMware/test-vcent...	Production	
EMC Production		/VMware/test-vcent...	Production	
EMC Production		/VMware/test-vcent...	Production	
EMC Templates		/VMware/test-vcent...	Production	
EMC Templates		/VMware/test-vcent...	Production	
EMC- Production		/VMware/test-vcent...	Decommissioned	

DISPLAYING 1 - 21 of 28 F

6.1.6 Device Discovery Guidance

As described previously in this chapter, device discovery can be accomplished using several different methods. These methods enable discovery of individual devices, specific domain mangers, and sets of devices from just a few up to an entire cloud. Each of these methods is summarized in Table 6-2 for reference. The remainder of this section highlights the unique management attributes of VMDC devices with respect to assurance discovery, and provides general device discovery guidance for the various management implementations.

Table 6-2. Device Discovery Methods

Device Type	Method	Requirements
Single Device	Zenoss CSA GUI	IP Address
UCSM	Zenoss CSA GUI	UCSM Credentials
vSphere	Zenoss CSA GUI	vSphere Credentials
EMC	Zenoss CSA GUI	EMC SMI-S Provider Credentials
Multiple Devices *	Zenoss CSA GUI	List, subnet, or range of IP addresses
Multiple Devices	Zenoss CSA CLI	Zenbatchload Configuration file

Note * When adding a lengthy list of devices, administrators should avoid the GUI option and use the *zenbatchload* utility method instead.

VMDC Devices Management Interface Attributes

Each VMDC device has unique characteristics and these attributes need to be taken into account when choosing a device discovery method. Some devices have a single management IP address, and others have a primary management address along with one or more Virtual IP (VIP) addresses. In addition, three "devices" under the VMDC reference architecture, UCSM, vSphere, and EMC (SMI-S Provider), can more accurately be described as domain managers for VMDC compute and storage sub-components. Table 6-3 summarizes these VMDC attributes.

Table 6-3. VMDC Devices and Their Unique Discovery Attributes

Device	Management IP	Management VIP	Domain Manager
ASR 9000	X		
ASR 1000	X		
6500 DSN	X		
Nexus 5000	X		
Nexus 3000	X		
Nexus 1000	X		
VSG	X		
MDS 9500	X		
NetApp	X		
Nexus 7000	X	X	

Device	Management IP	Management VIP	Domain Manager
ACE	X	X	
FWSM	X	X	
ASA/ASASM	X	X	
UCSM [1]	X	X	X
vSphere [2]	X		X
EMC SMI-S Provider [3]	X		X

Note

- [1] The UCS compute system management implementation is similar to a Hot Standby Router Protocol (HSRP) setup. The UCSM application runs on a pair of 6100s each with a unique management IP. The UCSM is typically accessed using a VIP. The 6100 and 5100 compute stack would be managed through the UCSM and thus do not have separate management IP addresses for assurance monitoring.
- [2] The vSphere infrastructure system is typically managed using a vSphere vCenter server. All ESXi hosts and their VMs are discovered when the vSphere vCenter is added to the Zenoss CSA device list.
- [3] An EMC storage device for both VMAX and VNX can only be accessed through an SMI-S Provider. EMC devices are thus added to the Zenoss CSA device list by adding an SMI-S Provider domain manager.

Guidance for Domain Managers

Domain managers, UCSM, vSphere, and EMC SMI-S should not be autodiscovered. They should be discovered using their unique device specific methods. They may also be discovered using the *zenbatchload* utility.

Guidance for Devices with Single Management IPs

Devices with single management IP addresses can be discovered individually, as a list of devices or possibly via subnet autodiscovery. The subnet discovery approach may be more efficient if these types of devices are segregated onto their own smaller subnet segments. If a few unwanted devices are discovered, they can easily be removed manually.

Devices with Both Management and Virtual IPs

Devices with a combination of management IP and VIP addresses need to be discovered intentionally and not by autodiscovery. It is recommended that both the IP and VIP address for these devices be discovered using the GUI-based add a single device method. Alternatively, they may also be discovered using the *zenbatchload* utility. Again, if duplicate devices are unintentionally discovered, they can easily be removed manually.

A Final Word on Discovery

The primary benefit of Zenoss CSA autodiscovery is not so much the initial discovery of a device's management IP, but instead the autodiscovery of that devices' subcomponents and the relationships behind the management address. Once a device is initially discovered, it is then classified by device type. Classification allows that device to be modeled and its subcomponents discovered.

The *zenbatchload* utility is the preferred method of discovering more than a few devices and is recommended for discovering an entirely new cloud or data center.

6.2 Dashboard Monitoring

Zenoss CSA provides multiple types of operational dashboards that are useful for monitoring cloud resource status. The following types of dashboards are illustrated in this section:

- **Aggregated Systemwide Dashboard.** This dashboard provides a customizable desktop that allows the operator to view the aggregate status of a list of devices.
- **Services Dashboard.** This dashboard provides the status of logically separated tenant services enabled on the VMDC cloud infrastructure.
- **Infrastructure Dashboard.** This dashboard summarizes the event status for device classes, individual devices, or their components.
- **Events Dashboard.** This dashboard aggregates the events of all devices into a single presentation.

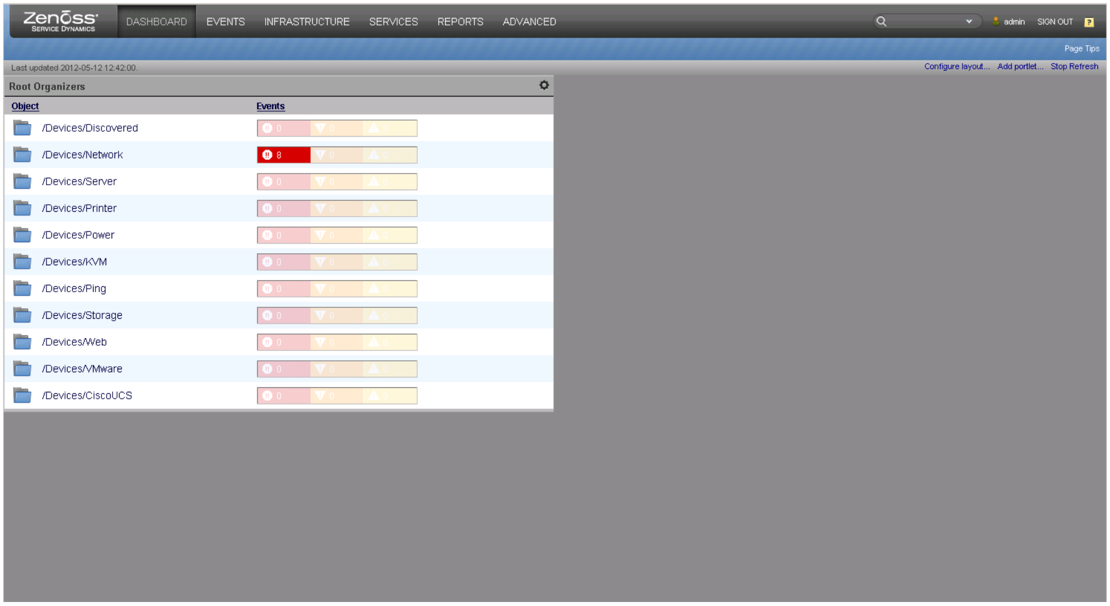
6.2.1 Aggregated Systemwide Dashboard

This section discusses how to customize and use the Aggregated Systemwide dashboard.

Customizing the Dashboard

VMDC resources can be monitored at an aggregate level using the CLSA-VMDC main dashboard. This dashboard is customizable in both format and content by cloud operator personnel. Figure 6-23 illustrates a sample customized dashboard emphasizing a device class listing. The operator can immediately see that there are eight severe events mapped to the network device class.

Figure 6-23. VMDC Service Assurance Dashboard



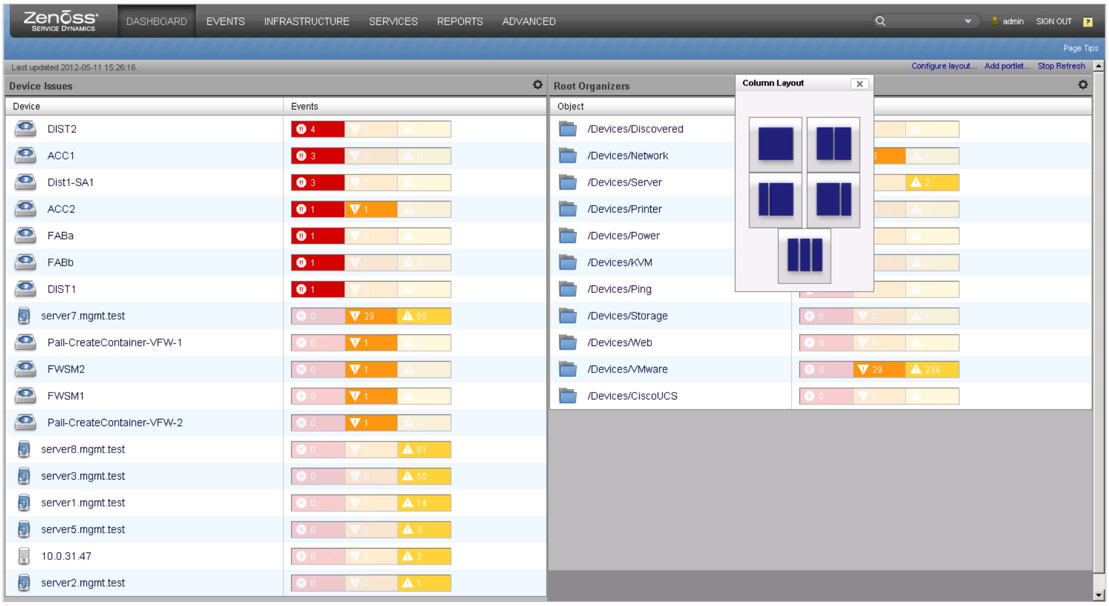
Last updated 2012-05-12 12:42:00.

Root Organizers

Object	Events		
/Devices/Discovered	0	0	0
/Devices/Network	8	0	0
/Devices/Server	0	0	0
/Devices/Printer	0	0	0
/Devices/Power	0	0	0
/Devices/KVM	0	0	0
/Devices/Ping	0	0	0
/Devices/Storage	0	0	0
/Devices/Web	0	0	0
/Devices/VMware	0	0	0
/Devices/CiscoUCS	0	0	0

The dashboard presentation format can be set to one, two, or three columns, as shown in Figure 6-24. This format can be changed later and the existing content adjusts to the current layout selection.

Figure 6-24. VMDC Service Assurance Dashboard Column Layout Selection



Last updated 2012-05-11 15:26:16.

Device Issues

Device	Events		
DIST2	4	0	0
ACC1	3	0	0
Dist1-SA1	3	0	0
ACC2	1	1	0
FABa	1	0	0
FABb	1	0	0
DIST1	1	0	0
server7.mgmt.test	0	29	85
Pall-CreateContainer-VFW-1	0	1	0
FWSM2	0	1	0
FWSM1	0	1	0
Pall-CreateContainer-VFW-2	0	1	0
server8.mgmt.test	0	0	81
server3.mgmt.test	0	0	50
server1.mgmt.test	0	0	14
server5.mgmt.test	0	0	3
10.0.31.47	0	0	2
server2.mgmt.test	0	0	1

Root Organizers

Object			
/Devices/Discovered		0	0
/Devices/Network		5	0
/Devices/Server		0	2
/Devices/Printer		0	0
/Devices/Power		0	0
/Devices/KVM		0	0
/Devices/Ping		0	0
/Devices/Storage	0	0	0
/Devices/Web	0	0	0
/Devices/VMware	0	29	234
/Devices/CiscoUCS	0	0	0

Within a dashboard column, an operator may apply a specific monitoring template from a list of nine portlet selections. A dashboard portlet collects and summarizes fault events according to a logical filter. For example, the dashboard can be customized to display fault information categorized by device classes or by individual devices. Table 6-4 provides a brief description of the portlet choices that can be placed on the dashboard.

Table 6-4. Dashboard Portlet Content Choices

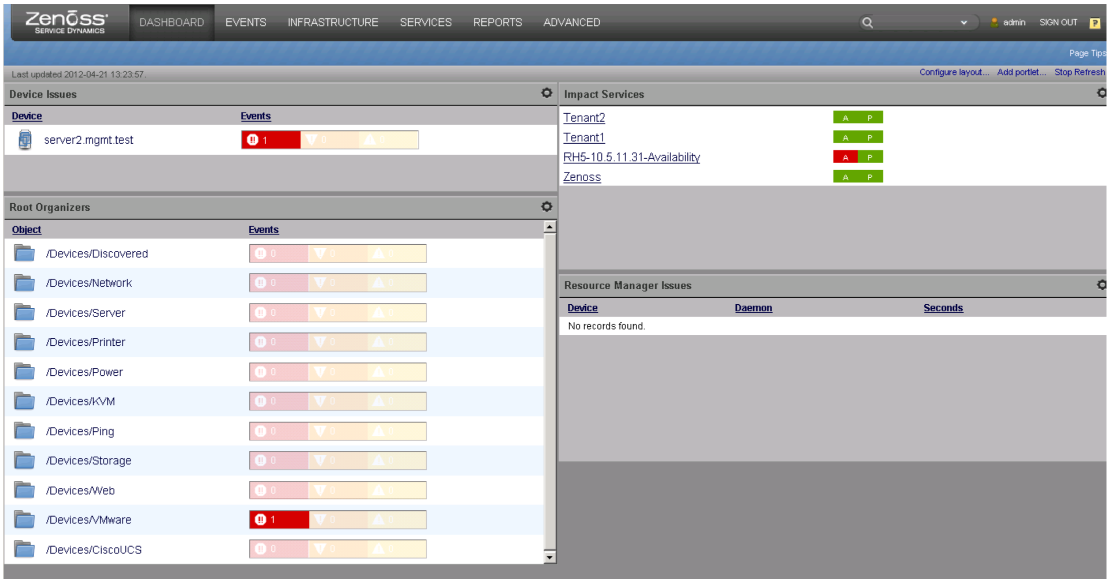
Portlet	Portlet Usage
Device Issues	Summarizes events based on the list of all devices monitored
Google Maps	Shows configured locations and configured network connections
Daemon Processes Down	Contains assurance system self-monitoring information about daemons and servers
Impact Services	Lists VMDC tenant service status
Production States	Shows devices assigned to a particular production state such as production, pre-production, test, or maintenance
Site Window	Provides convenient hot links to Zenoss product documentation and training information. This can be customized to link to any web site useful to an operator
Top-level Organizers	Organizes resources and events by device class, group, location, or system
Messages	Displays assurance system messages

Portlet	Portlet Usage
Watch List	Presents event summary for an operator chosen list of devices

Using the Dashboard

The dashboard not only provides a simple and obvious representation of faults, it also provides the operator several ways to drill down into an issue. The dashboard example shown in Figure 6-25 highlights several error conditions in red. At the upper left, a single device called server2 has a severe fault. Below the device list, the device class /Devices/VMware also has a severe fault. Since there is only one device and one device class exhibiting a fault, there is high probability that server2 is the cause of the service availability issue shown at the upper right of the dashboard.

Figure 6-25. Dashboard Reports a Serious Issue



Last updated 2012-04-21 13:23:57.

Device Issues

Device	Events
server2.mgmt.test	1 0 0

Root Organizers

Object	Events
/Devices/Discovered	0 0 0
/Devices/Network	0 0 0
/Devices/Server	0 0 0
/Devices/Printer	0 0 0
/Devices/Power	0 0 0
/Devices/KVM	0 0 0
/Devices/Ping	0 0 0
/Devices/Storage	0 0 0
/Devices/Web	0 0 0
/Devices/VMware	1 0 0
/Devices/CiscoUCS	0 0 0

Impact Services

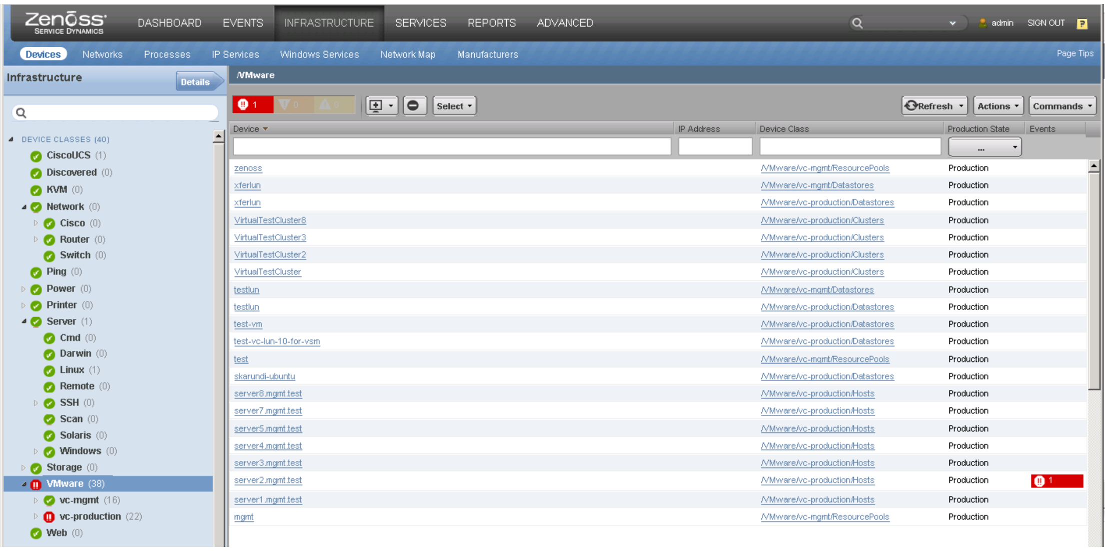
Tenant2	A P
Tenant1	A P
RH5-10.5.11.31-Availability	A P
Zenoss	A P

Resource Manager Issues

Device	Daemon	Seconds
No records found.		

From the dashboard, an operator can jump to subsequent displays that provide more context around the fault. By selecting the device class /Devices/VMware, the dashboard takes the operator to the VMware infrastructure devices page seen in Figure 6-26. Here, the issue can be associated with a specific vCenter called vc-production.

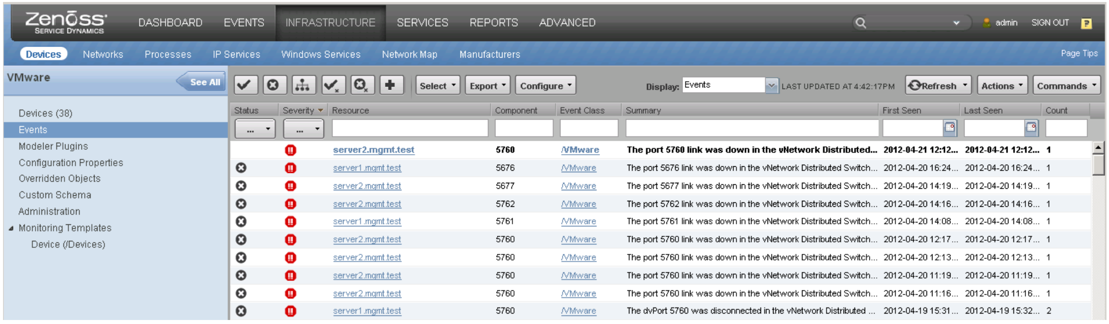
Figure 6-26. Navigating to Device Class /Devices/VMware



Device	IP Address	Device Class	Production State	Events
zenoss		/VMware/vc-mgmt/ResourcePools	Production	
xferlun		/VMware/vc-mgmt/Datastores	Production	
xferlun		/VMware/vc-production/Datastores	Production	
VirtualTestCluster8		/VMware/vc-production/Clusters	Production	
VirtualTestCluster3		/VMware/vc-production/Clusters	Production	
VirtualTestCluster2		/VMware/vc-production/Clusters	Production	
VirtualTestCluster		/VMware/vc-production/Clusters	Production	
testlun		/VMware/vc-mgmt/Datastores	Production	
testlun		/VMware/vc-production/Datastores	Production	
test-vm		/VMware/vc-production/Datastores	Production	
test-vc-lun-10-for-vsm		/VMware/vc-production/Datastores	Production	
test		/VMware/vc-mgmt/ResourcePools	Production	
skarundi-ubuntu		/VMware/vc-production/Datastores	Production	
server8.mgmt.test		/VMware/vc-production/Hosts	Production	
server7.mgmt.test		/VMware/vc-production/Hosts	Production	
server5.mgmt.test		/VMware/vc-production/Hosts	Production	
server4.mgmt.test		/VMware/vc-production/Hosts	Production	
server3.mgmt.test		/VMware/vc-production/Hosts	Production	
server2.mgmt.test		/VMware/vc-production/Hosts	Production	1
server1.mgmt.test		/VMware/vc-production/Hosts	Production	
mgmt		/VMware/vc-mgmt/ResourcePools	Production	

If instead, the operator selects the /Device/VMware red fault event, the display jumps to the VMware infrastructure events page, as seen in Figure 6-27. This page lists events for all VMware components. Here, the operator learns that the event for server2 under investigation is associated with a port 5760 link down.

Figure 6-27. Navigating to VMware Infrastructure Events Listing

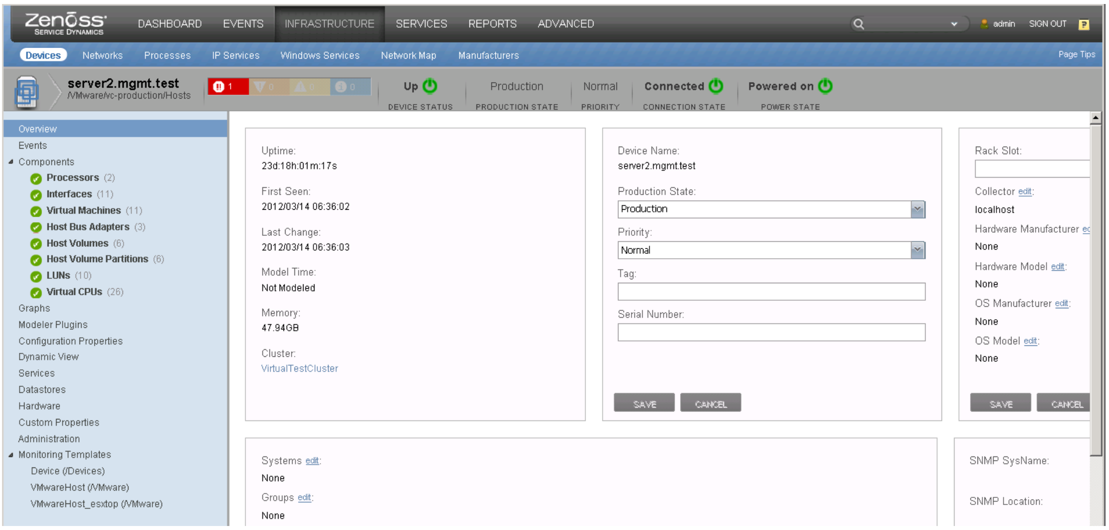


Display: Events LAST UPDATED AT 4:42:17PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
	●	**server2.mgmt.test**	**5760**	**/VMware**	**The port 5760 link was down in the vNetwork Distributed...**	**2012-04-21 12:12...**	**2012-04-21 12:12...**	**1**
☒	●	server1.mgmt.test	5676	/VMware	The port 5676 link was down in the vNetwork Distributed Switch...	2012-04-20 16:24...	2012-04-20 16:24...	1
☒	●	server2.mgmt.test	5677	/VMware	The port 5677 link was down in the vNetwork Distributed Switch...	2012-04-20 14:19...	2012-04-20 14:19...	1
☒	●	server2.mgmt.test	5762	/VMware	The port 5762 link was down in the vNetwork Distributed Switch...	2012-04-20 14:16...	2012-04-20 14:16...	1
☒	●	server1.mgmt.test	5761	/VMware	The port 5761 link was down in the vNetwork Distributed Switch...	2012-04-20 14:08...	2012-04-20 14:08...	1
☒	●	server2.mgmt.test	5760	/VMware	The port 5760 link was down in the vNetwork Distributed Switch...	2012-04-20 12:17...	2012-04-20 12:17...	1
☒	●	server2.mgmt.test	5760	/VMware	The port 5760 link was down in the vNetwork Distributed Switch...	2012-04-20 12:13...	2012-04-20 12:13...	1
☒	●	server2.mgmt.test	5760	/VMware	The port 5760 link was down in the vNetwork Distributed Switch...	2012-04-20 11:19...	2012-04-20 11:19...	1
☒	●	server2.mgmt.test	5760	/VMware	The port 5760 link was down in the vNetwork Distributed Switch...	2012-04-20 11:16...	2012-04-20 11:16...	1
☒	●	server1.mgmt.test	5760	/VMware	The dvPort 5760 was disconnected in the vNetwork Distributed ...	2012-04-19 15:31...	2012-04-19 15:32...	2

If the operator returns to the main dashboard and selects the specific device server2, then the dashboard jumps to the /VMware/vc-production/Hosts overview under the infrastructure view. Referring to Figure 6-28, information about server2 can be identified that assists the operator in troubleshooting the fault.

Figure 6-28. Navigating to servers2 Hosts Overview



server2.mgmt.test
/VMware/vc-production/Hosts

1 0 0 0

Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY
Connected — CONNECTION STATE
Powered on — POWER STATE

Uptime:
23d:18h:01m:17s

First Seen:
2012/03/14 06:36:02

Last Change:
2012/03/14 06:36:03

Model Time:
Not Modeled

Memory:
47.94GB

Cluster:
VirtualTestCluster

Device Name:
server2.mgmt.test

Production State:
Production

Priority:
Normal

Tag:

Serial Number:

Rack Slot:

Collector edit:
localhost

Hardware Manufacturer ed

None

Hardware Model edit:
None

OS Manufacturer edit:
None

OS Model edit:
None

Systems edit:
None

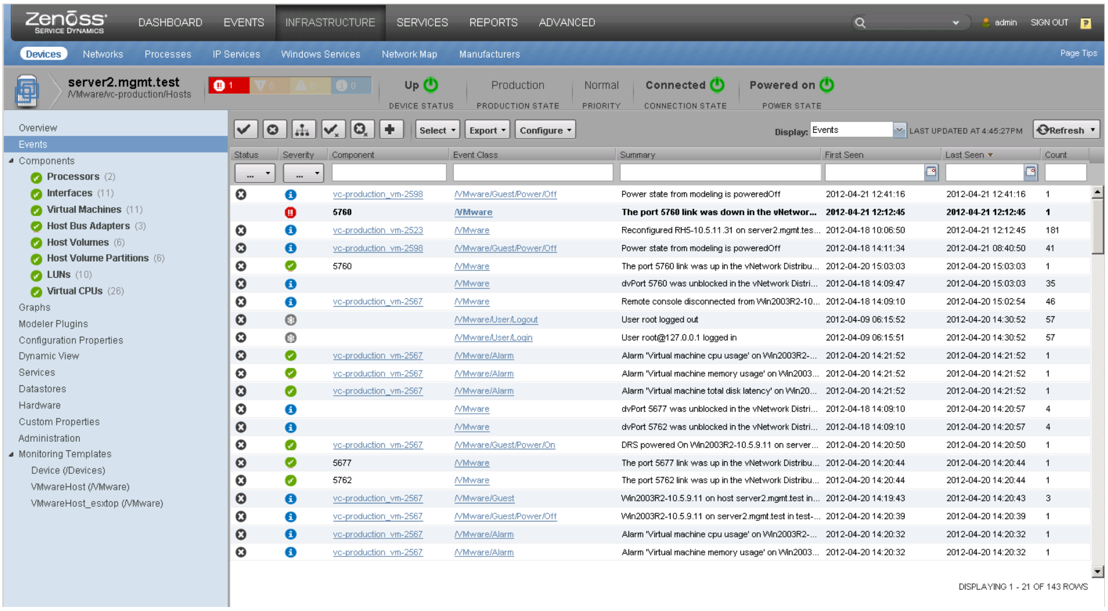
Groups edit:
None

SNMP SysName:

SNMP Location:

Returning to the dashboard again, the operator can select the server2 red event to jump directly to the server2 only events page. While providing confirmation that the event is associated with port 5760 link down, it also provides a historical view to prior events specific to this device.

Figure 6-29. Navigating to Specific server2 Hosts Events View



server2.mgmt.test
/VMware/vc-production/Hosts

1 0 0 0

Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY
Connected — CONNECTION STATE
Powered on — POWER STATE

Display: Events LAST UPDATED AT 4:45:27PM

Status	Severity	Component	Event Class	Summary	First Seen	Last Seen	Count
⊗	ⓘ	vc-production_vm-2598	/VMware/Guest/Power/Off	Power state from modeling is poweredOff	2012-04-21 12:41:16	2012-04-21 12:41:16	1
	‼	**5760**	**/VMware**	**The port 5760 link was down in the vNetwor...**	**2012-04-21 12:12:45**	**2012-04-21 12:12:45**	**1**
⊗	ⓘ	vc-production_vm-2523	/VMware	Reconfigured RH5-10.5.11.31 on server2.mgmt.tes...	2012-04-18 10:06:50	2012-04-21 12:12:45	181
⊗	ⓘ	vc-production_vm-2598	/VMware/Guest/Power/Off	Power state from modeling is poweredOff	2012-04-18 14:11:34	2012-04-21 08:40:50	41
⊗	✓	5760	/VMware	The port 5760 link was up in the vNetwork Distribu...	2012-04-20 15:03:03	2012-04-20 15:03:03	1
⊗	ⓘ		/VMware	dvPort 5760 was unblocked in the vNetwork Distri...	2012-04-18 14:09:47	2012-04-20 15:03:03	35
⊗	ⓘ	vc-production_vm-2567	/VMware	Remote console disconnected from Win2003R2-10...	2012-04-18 14:09:10	2012-04-20 15:02:54	46
⊗	⚙		/VMware/User/Logout	User root logged out	2012-04-09 06:15:52	2012-04-20 14:30:52	57
⊗	⚙		/VMware/User/Login	User root@127.0.0.1 logged in	2012-04-09 06:15:51	2012-04-20 14:30:52	57
⊗	✓	vc-production_vm-2567	/VMware/Alarm	Alarm 'Virtual machine cpu usage' on Win2003R2-...	2012-04-20 14:21:52	2012-04-20 14:21:52	1
⊗	✓	vc-production_vm-2567	/VMware/Alarm	Alarm 'Virtual machine memory usage' on Win2003...	2012-04-20 14:21:52	2012-04-20 14:21:52	1
⊗	✓	vc-production_vm-2567	/VMware/Alarm	Alarm 'Virtual machine total disk latency' on Win20...	2012-04-20 14:21:52	2012-04-20 14:21:52	1
⊗	ⓘ		/VMware	dvPort 5677 was unblocked in the vNetwork Distri...	2012-04-18 14:09:10	2012-04-20 14:20:57	4
⊗	ⓘ		/VMware	dvPort 5762 was unblocked in the vNetwork Distri...	2012-04-18 14:09:10	2012-04-20 14:20:57	4
⊗	✓	vc-production_vm-2567	/VMware/Guest/Power/On	DRS powered On Win2003R2-10.5.9.11 on server...	2012-04-20 14:20:50	2012-04-20 14:20:50	1
⊗	✓	5677	/VMware	The port 5677 link was up in the vNetwork Distribu...	2012-04-20 14:20:44	2012-04-20 14:20:44	1
⊗	✓	5762	/VMware	The port 5762 link was up in the vNetwork Distribu...	2012-04-20 14:20:44	2012-04-20 14:20:44	1
⊗	ⓘ	vc-production_vm-2567	/VMware/Guest	Win2003R2-10.5.9.11 on host server2.mgmt.test in...	2012-04-20 14:19:43	2012-04-20 14:20:43	3
⊗	ⓘ	vc-production_vm-2567	/VMware/Guest/Power/Off	Win2003R2-10.5.9.11 on server2.mgmt.test in test-...	2012-04-20 14:20:39	2012-04-20 14:20:39	1
⊗	ⓘ	vc-production_vm-2567	/VMware/Alarm	Alarm 'Virtual machine cpu usage' on Win2003R2-...	2012-04-20 14:20:32	2012-04-20 14:20:32	1
⊗	ⓘ	vc-production_vm-2567	/VMware/Alarm	Alarm 'Virtual machine memory usage' on Win2003...	2012-04-20 14:20:32	2012-04-20 14:20:32	1

DISPLAYING 1 - 21 OF 143 ROWS

The Aggregated Systemwide dashboard provides the cloud operator a customizable tool to identify faults quickly. Additionally, once faults are identified, the dashboard enables convenient navigation features for the operator to drill down for more detailed information while troubleshooting.

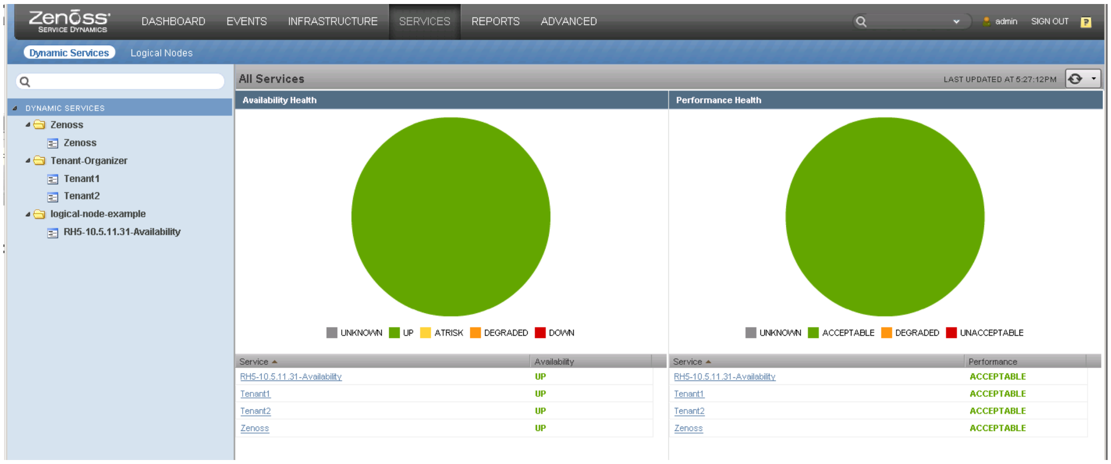
6.2.2 Services Dashboard

The Services dashboard is available to help cloud operators monitor the health of tenant services. The Services dashboard is split into two, side-by-side views - tenant services availability health on the left and performance health on the right. The top of the dashboard represents all services using a pie chart. The bottom half of the dashboard lists each specific service name and its availability and performance health status.

The Services dashboard is a useful tool to visually assess the status of VMDC tenant services rapidly. If the operator notices that a service displays a status other than green, then the table below the pie chart can be quickly correlated to the service of interest by both status name and color.

Figure 6-30 illustrates a clean dashboard where all services are reporting available and acceptable status.

Figure 6-30. VMDC Services Dashboard



Service ▲	Availability
RH5-10.5.11.31-Availability	UP
Tenant1	UP
Tenant2	UP
Zenoss	UP

Service ▲	Performance
RH5-10.5.11.31-Availability	ACCEPTABLE
Tenant1	ACCEPTABLE
Tenant2	ACCEPTABLE
Zenoss	ACCEPTABLE

A proportional percentage of the pie chart representing each service changes colors based on that service's status. The service status changes following reception of service impacting events and after calculation of the default, custom, or global service policies in effect. See Section 3.5.4 Root Cause Analysis and Service Impact Analysis for more information regarding tenant service policies. Table 6-5 and Table 6-6 indicate the colors used for service availability and performance state indicators.

Table 6-5. Service Availability Color Mapping

Availability States	Pie Chart Color Code
UNKNOWN	Grey
UP	Green
AT RISK	Yellow
DEGRADED	Orange
DOWN	Red

Table 6-6. Service Performance Color Mapping

Performance States	Pie Chart Color Code
UNKNOWN	Grey
ACCEPTABLE	Green

Performance States	Pie Chart Color Code
DEGRADED	Orange
UNACCEPTABLE	Red

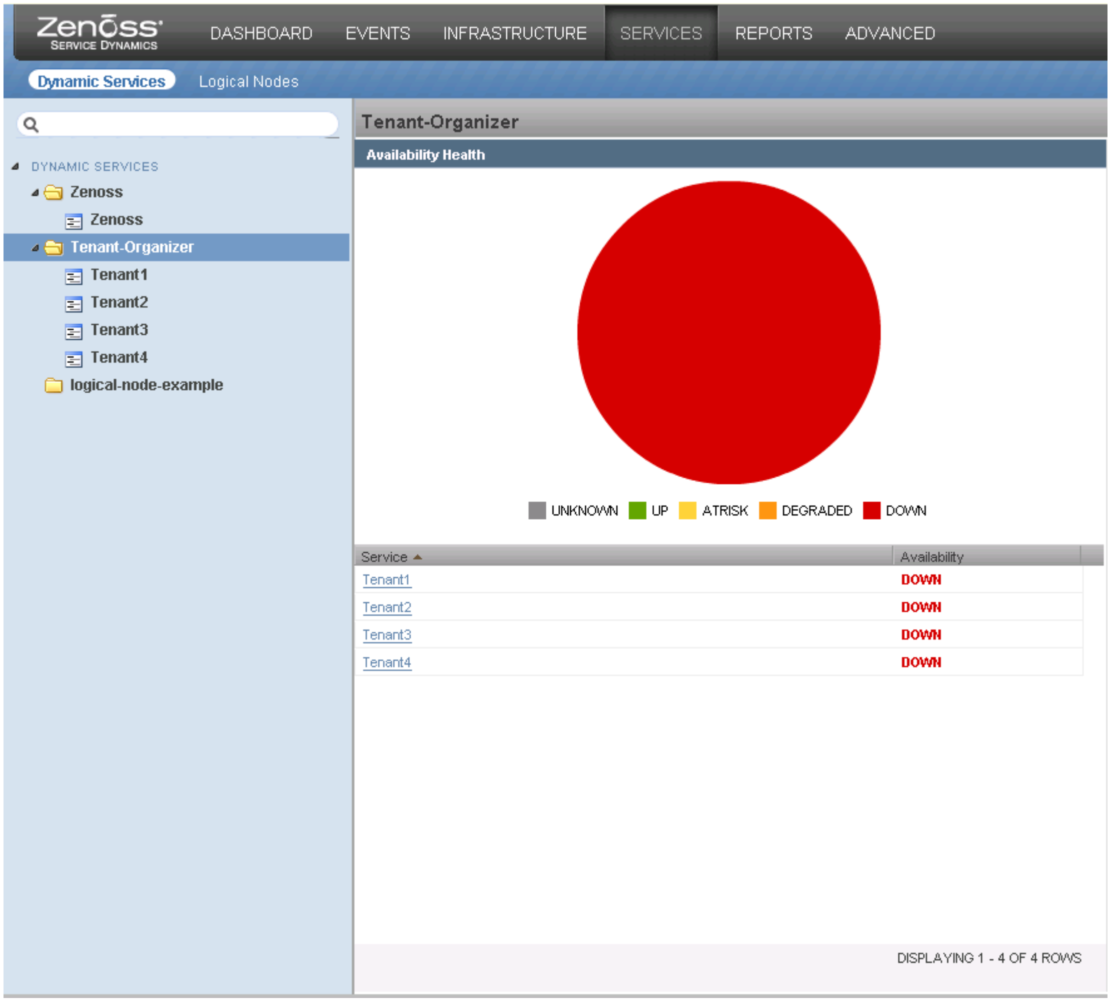
To illustrate the dashboard's usefulness, tenant service policies have been modified according to the cases in Table 6-7. Once a service availability impacting event occurs, the Service Availability dashboard reflects the policies in effect.

Table 6-7. Availability Health Dashboard Examples

Case	Description
Case 1	All tenant service policies have been configured to resolve to down for a specific fault.
Case 2	Each tenant service has been configured to resolve to each of the possible states when a severe fault occurs.

Case 1 is shown in Figure 6-31. In this case, a catastrophic event has occurred to a common component underlying these specific tenant services, and the service dashboard clearly shows all services down in red.

Figure 6-31. All Tenant Services Shown Down

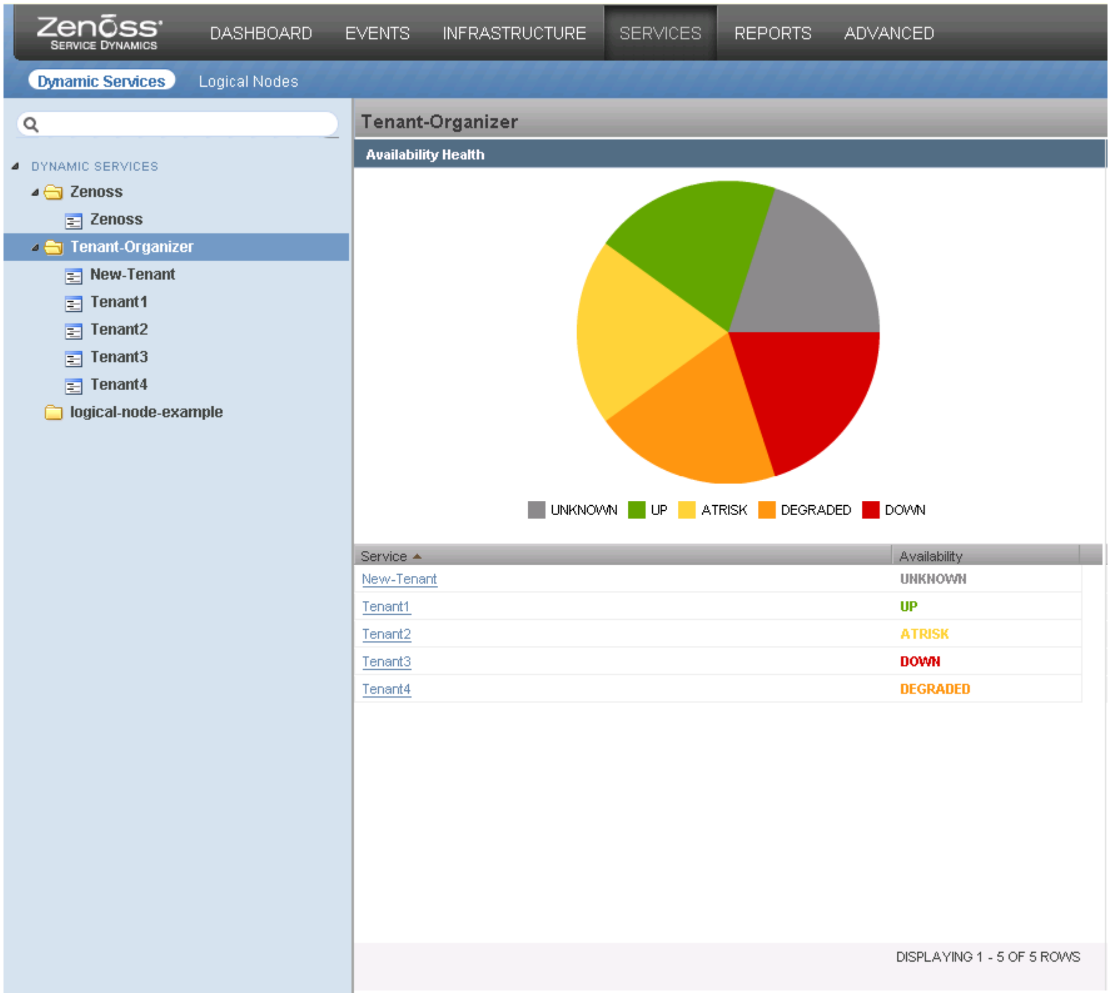


Service ▲	Availability
Tenant1	DOWN
Tenant2	DOWN
Tenant3	DOWN
Tenant4	DOWN

DISPLAYING 1 - 4 OF 4 ROWS

Case 2 is presented in Figure 6-32. Here, the service policies have been arbitrarily set to resolve each tenant service state to a different status merely for demonstration purposes. When a severe fault is again registered to the same shared infrastructure, the dashboard displays a different color for each tenant's service state.

Figure 6-32. Tenant Services Shown with Multiple States



Tenant-Organizer

Availability Health

Service ▲	Availability
New-Tenant	UNKNOWN
Tenant1	UP
Tenant2	ATRISK
Tenant3	DOWN
Tenant4	DEGRADED

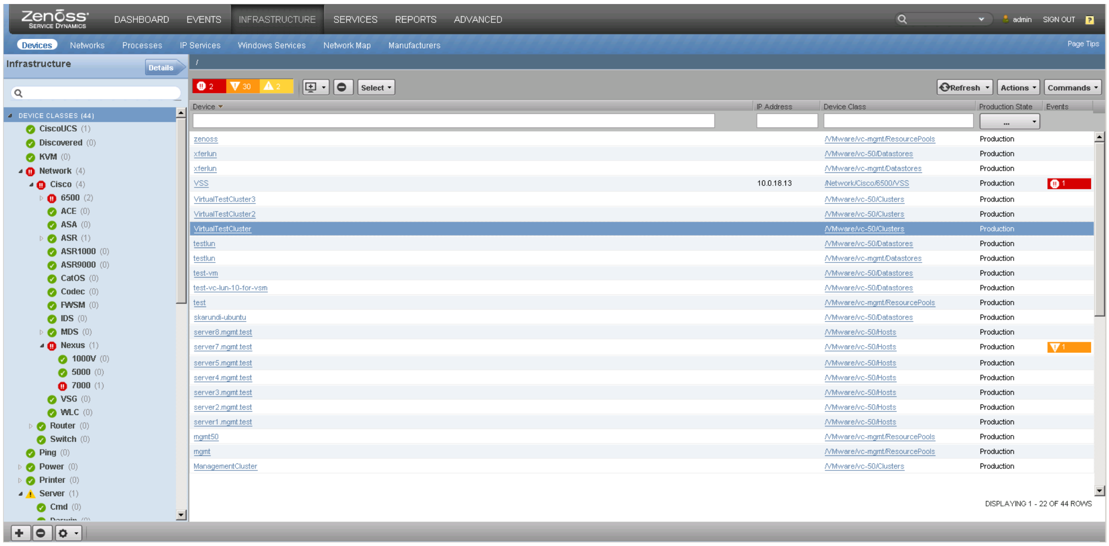
DISPLAYING 1 - 5 OF 5 ROWS

The Performance Health dashboard operates in a similar manner to the Availability Health dashboard. The distinction between the two dashboards is that availability is impacted by monitored resource fault events, and performance is impacted by monitored performance statistics and thresholds.

6.2.3 Infrastructure Dashboard

The Infrastructure dashboard enables a cloud operator to survey the overall status of all physical and virtual VMDC resources being monitored and to navigate down into areas of concern. Figure 6-33 shows a view of the entire dashboard.

Figure 6-33. Infrastructure Dashboard Default View



- DEVICE CLASSES (44)
  - CiscoUCS (1)
  - Discovered (0)
  - KVM (0)
  - Network (4)
    - Cisco (4)
      - 6500 (2)
      - ACE (0)
      - ASA (0)
      - ASR (1)
      - ASR1000 (0)
      - ASR9000 (0)
      - CatOS (0)
      - Codec (0)
      - FWSM (0)
      - IDS (0)
      - MDS (0)
      - Nexus (1)
        - 1000V (0)
        - 5000 (0)
        - 7000 (1)
      - VSG (0)
      - WLC (0)
    - Router (0)
    - Switch (0)
  - Ping (0)
  - Power (0)
  - Printer (0)
  - Server (1)
    - Cmd (0)

Device	IP Address	Device Class	Production State	Events
zenoss		/VMware/vc-mgmt/ResourcePools	Production	
xferlun		/VMware/vc-50/Datastores	Production	
xferlun		/VMware/vc-mgmt/Datastores	Production	
VSS	10.0.18.13	/Network/Cisco/6500/VSS	Production	1
VirtualTestCluster3		/VMware/vc-50/Clusters	Production	
VirtualTestCluster2		/VMware/vc-50/Clusters	Production	
VirtualTestCluster		/VMware/vc-50/Clusters	Production	
testlun		/VMware/vc-50/Datastores	Production	
testlun		/VMware/vc-mgmt/Datastores	Production	
test-vm		/VMware/vc-50/Datastores	Production	
test-vc-lun-10-for-vsm		/VMware/vc-50/Datastores	Production	
test		/VMware/vc-mgmt/ResourcePools	Production	
skarundi-ubuntu		/VMware/vc-50/Datastores	Production	
server8.mgmt.test		/VMware/vc-50/Hosts	Production	
server7.mgmt.test		/VMware/vc-50/Hosts	Production	1
server5.mgmt.test		/VMware/vc-50/Hosts	Production	
server4.mgmt.test		/VMware/vc-50/Hosts	Production	
server3.mgmt.test		/VMware/vc-50/Hosts	Production	
server2.mgmt.test		/VMware/vc-50/Hosts	Production	
server1.mgmt.test		/VMware/vc-50/Hosts	Production	
mgmt50		/VMware/vc-mgmt/ResourcePools	Production	
mgmt		/VMware/vc-mgmt/ResourcePools	Production	
ManagementCluster		/VMware/vc-50/Clusters	Production	

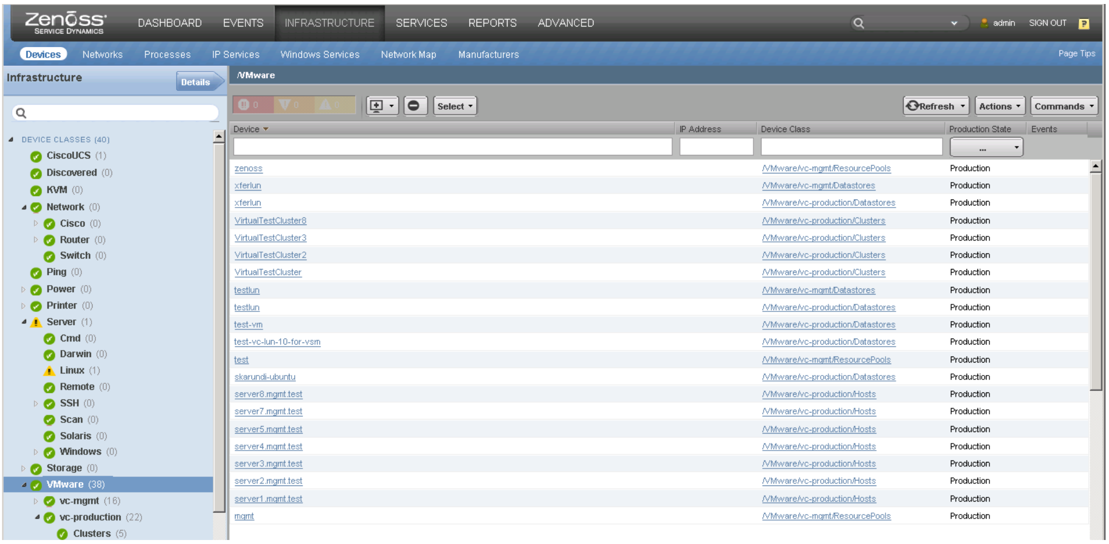
DISPLAYING 1 - 22 OF 44 ROWS

The dashboard is divided into three interconnected regions that feature a telescoping view of the infrastructure and device information. The initial view of the dashboard defaults to selecting all device classes in the navigation window on the left. Corresponding to that selection, a summary of the highest three event severity types is presented across the top of the dashboard. Below that, all infrastructure devices are listed in a scrolling window.

Note The navigation window's icons change to reflect the highest severity event for that device class or device.

The telescoping view allows the cloud operator the flexibility to navigate from an "all devices" view down to a "device class" view, and even further down to a "device specific" view. As these separate views are selected from the left, both the event summary bar and the device listing present filtered information limited to the view chosen. For example, highlighting the VMware device class on the left restricts the event summary bar to only those events associated with the VMware infrastructure. In addition, the device listing window only displays VMware related components. Figure 6-34 shows the VMware device class view.

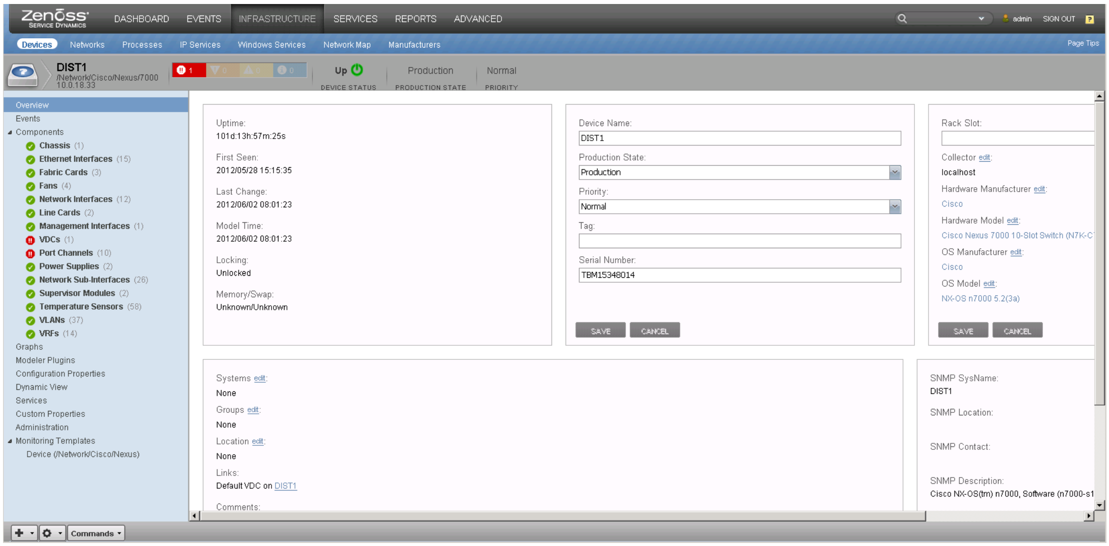
Figure 6-34. VMware Device Class View



Device	IP Address	Device Class	Production State	Events
zenoss		/VMware/vc-mgmt/ResourcePools	Production	
xferlun		/VMware/vc-mgmt/Datastores	Production	
xferlun		/VMware/vc-production/Datastores	Production	
VirtualTestCluster8		/VMware/vc-production/Clusters	Production	
VirtualTestCluster3		/VMware/vc-production/Clusters	Production	
VirtualTestCluster2		/VMware/vc-production/Clusters	Production	
VirtualTestCluster		/VMware/vc-production/Clusters	Production	
testlun		/VMware/vc-mgmt/Datastores	Production	
testlun		/VMware/vc-production/Datastores	Production	
test-vm		/VMware/vc-production/Datastores	Production	
test-vc-lun-10-for-vsm		/VMware/vc-production/Datastores	Production	
test		/VMware/vc-mgmt/ResourcePools	Production	
skarundi-ubuntu		/VMware/vc-production/Datastores	Production	
server8.mgmt.test		/VMware/vc-production/Hosts	Production	
server7.mgmt.test		/VMware/vc-production/Hosts	Production	
server5.mgmt.test		/VMware/vc-production/Hosts	Production	
server4.mgmt.test		/VMware/vc-production/Hosts	Production	
server3.mgmt.test		/VMware/vc-production/Hosts	Production	
server2.mgmt.test		/VMware/vc-production/Hosts	Production	
server1.mgmt.test		/VMware/vc-production/Hosts	Production	
mgmt		/VMware/vc-mgmt/ResourcePools	Production	

Navigating once more down to a specific device such as a Nexus 7000 shows only the information related to that device. Figure 6-35 details this type of view at a device level.

Figure 6-35. Example of a Nexus 7000 Device View



DIST1
/Network/Cisco/Nexus/7000
10.0.18.33

Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY

Uptime:
101d:13h:57m:25s

First Seen:
2012/05/28 15:15:35

Last Change:
2012/06/02 08:01:23

Model Time:
2012/06/02 08:01:23

Locking:
Unlocked

Memory/Swap:
Unknown/Unknown

Device Name:
DIST1

Production State:
Production

Priority:
Normal

Tag:

Serial Number:
TBM15348014

Rack Slot:

Collector edit:
localhost

Hardware Manufacturer edit:
Cisco

Hardware Model edit:
Cisco Nexus 7000 10-Slot Switch (N7K-C

OS Manufacturer edit:
Cisco

OS Model edit:
NX-OS n7000 5.2(3a)

Systems edit:
None

Groups edit:
None

Location edit:
None

Links:
Default VDC on DIST1

Comments:

SNMP SysName:
DIST1

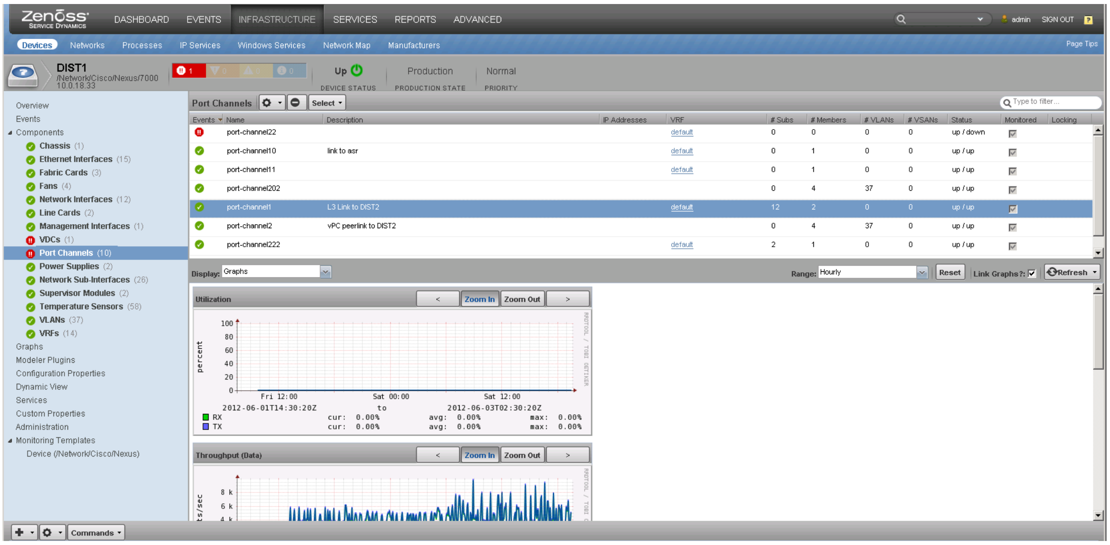
SNMP Location:

SNMP Contact:

SNMP Description:
Cisco NX-OS(tm) n7000, Software (n7000-s1

At this level, the dashboard provides access to detailed information about the device. A cloud operator can view the device event summary across the top and can select event detail in the navigation window. Other specific device sub-components can be explored as well. Figure 6-36 shows the port-channel components selected and highlights a fault event associated with port-channel22 in up/down state.

Figure 6-36. Example of a Nexus 7000 Port-Channel Sub-component View



Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY

Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
●	port-channel22			default	0	0	0	0	up / down	☑	
✓	port-channel10	link to asr		default	0	1	0	0	up / up	☑	
✓	port-channel11			default	0	1	0	0	up / up	☑	
✓	port-channel202				0	4	37	0	up / up	☑	
✓	port-channel1	L3 Link to DIST2		default	12	2	0	0	up / up	☑	
✓	port-channel2	vPC peerlink to DIST2			0	4	37	0	up / up	☑	
✓	port-channel222			default	2	1	0	0	up / up	☑	

Display: Graphs
Range: Hourly
Link Graphs?: ☑

```
2012-06-01T14:30:20Z           to           2012-06-03T02:30:20Z
RX                cur:  0.00%        avg:  0.00%        max:  0.00%
TX                cur:  0.00%        avg:  0.00%        max:  0.00%
```

The Infrastructure dashboard provides navigation and fault isolation capabilities to a VMDC cloud operator. In particular, the telescoping capability of this dashboard makes it useful for navigating from the aggregated view down to the specific status of individual VMDC infrastructure components.

6.2.4 Events Dashboard

The Events dashboard is perhaps the most straightforward of all of the dashboards. This dashboard does not provide device or service navigation as shown previously. This dashboard instead presents a configurable list of all events that have been received from the VMDC infrastructure. Because this list can be overwhelming in its length and level of detailed information, the dashboard provides useful event management capabilities to an operator. The two most notable of these capabilities are its sorting and filtering features.

Note Conveniently, the volume of information has already been de-duplicated by representing repetitive events on one line and showing an event count.

Event Sorting

While previous dashboards present only the highest three event severities, the Events dashboard provides access to all events and severities. Event severity is differentiated according to Table 6-8.

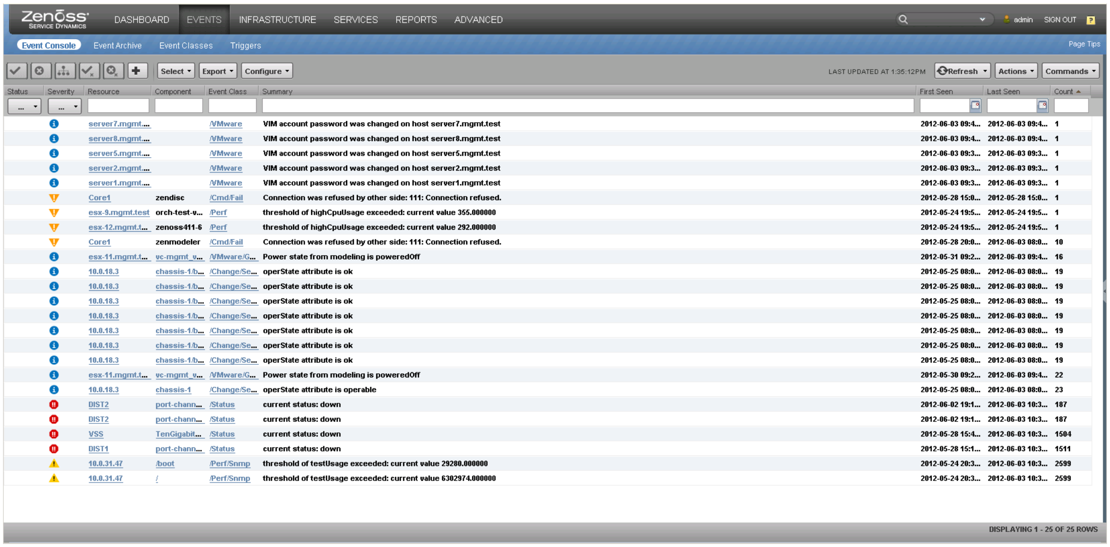
Table 6-8. Mapping of Event Severity, Naming, and Color Code

Severity	Name	Color
0	Clear	Green
1	Debug	Grey
2	Info	Blue
3	Warning	Yellow
4	Error	Orange

Severity	Name	Color
5	Critical	Red

Events may be sorted according to any of the attributes presented across the column heading bar to collect similar events together. For example, events may be sorted according to severity, resource (device), or number of occurrences. Figure 6-37 shows an initial view of the current event list.

Figure 6-37. Initial Event List



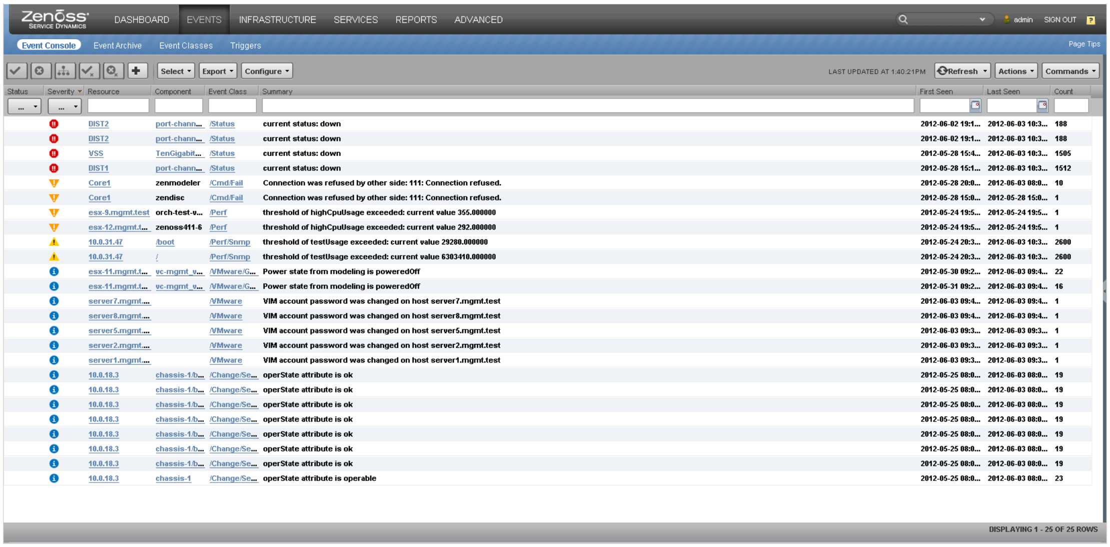
LAST UPDATED AT 1:35:12PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count ▲
	ⓘ	server7.mgmt....		/VMware	VIM account password was changed on host server7.mgmt.test	2012-06-03 09:4...	2012-06-03 09:4...	1
	ⓘ	server8.mgmt....		/VMware	VIM account password was changed on host server8.mgmt.test	2012-06-03 09:4...	2012-06-03 09:4...	1
	ⓘ	server5.mgmt....		/VMware	VIM account password was changed on host server5.mgmt.test	2012-06-03 09:3...	2012-06-03 09:3...	1
	ⓘ	server2.mgmt....		/VMware	VIM account password was changed on host server2.mgmt.test	2012-06-03 09:3...	2012-06-03 09:3...	1
	ⓘ	server1.mgmt....		/VMware	VIM account password was changed on host server1.mgmt.test	2012-06-03 09:3...	2012-06-03 09:3...	1
	▼	Core1	zendisc	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-28 15:0...	2012-05-28 15:0...	1
	▼	esx-9.mgmt.test	orch-test-v...	/Perf	threshold of highCpuUsage exceeded: current value 355.000000	2012-05-24 19:5...	2012-05-24 19:5...	1
	▼	esx-12.mgmt.t...	zenoss411-6	/Perf	threshold of highCpuUsage exceeded: current value 292.000000	2012-05-24 19:5...	2012-05-24 19:5...	1
	▼	Core1	zenmodeler	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-28 20:0...	2012-06-03 08:0...	10
	ⓘ	esx-11.mgmt.t...	vc-mgmt_v...	/VMware/G...	Power state from modeling is poweredOff	2012-05-31 09:2...	2012-06-03 09:4...	16
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	esx-11.mgmt.t...	vc-mgmt_v...	/VMware/G...	Power state from modeling is poweredOff	2012-05-30 09:2...	2012-06-03 09:4...	22
	ⓘ	10.0.18.3	chassis-1	/Change/Se...	operState attribute is operable	2012-05-25 08:0...	2012-06-03 08:0...	23
	‼	DIST2	port-chann...	/Status	current status: down	2012-06-02 19:1...	2012-06-03 10:3...	187
	‼	DIST2	port-chann...	/Status	current status: down	2012-06-02 19:1...	2012-06-03 10:3...	187
	‼	VSS	TenGigabit...	/Status	current status: down	2012-05-28 15:4...	2012-06-03 10:3...	1504
	‼	DIST1	port-chann...	/Status	current status: down	2012-05-28 15:1...	2012-06-03 10:3...	1511
	⚠	10.0.31.47	/boot	/Perf/Snmp	threshold of testUsage exceeded: current value 29280.000000	2012-05-24 20:3...	2012-06-03 10:3...	2599
	⚠	10.0.31.47	/	/Perf/Snmp	threshold of testUsage exceeded: current value 6302974.000000	2012-05-24 20:3...	2012-06-03 10:3...	2599

DISPLAYING 1 - 25 OF 25 ROWS

Figure 6-38 shows the same list sorted according to severity.

Figure 6-38. Event List Sorted Manually According to Severity



LAST UPDATED AT 1:40:21PM

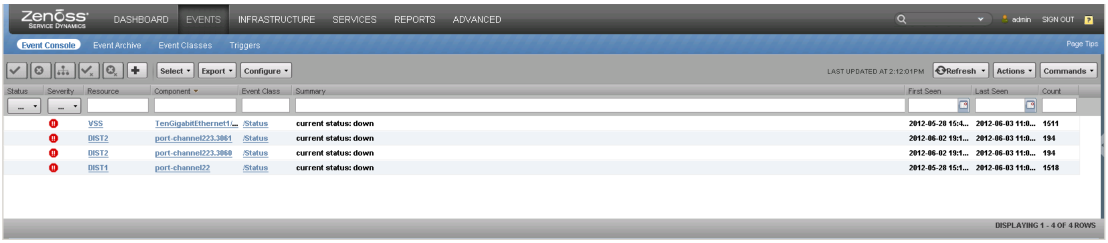
Status	Severity ▼	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
	‼	DIST2	port-chann...	/Status	current status: down	2012-06-02 19:1...	2012-06-03 10:3...	188
	‼	DIST2	port-chann...	/Status	current status: down	2012-06-02 19:1...	2012-06-03 10:3...	188
	‼	VSS	TenGigabit...	/Status	current status: down	2012-05-28 15:4...	2012-06-03 10:3...	1505
	‼	DIST1	port-chann...	/Status	current status: down	2012-05-28 15:1...	2012-06-03 10:3...	1512
	▼	Core1	zenmodeler	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-28 20:0...	2012-06-03 08:0...	10
	▼	Core1	zendisc	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-28 15:0...	2012-05-28 15:0...	1
	▼	esx-9.mgmt.test	orch-test-v...	/Perf	threshold of highCpuUsage exceeded: current value 355.000000	2012-05-24 19:5...	2012-05-24 19:5...	1
	▼	esx-12.mgmt.t...	zenoss411-6	/Perf	threshold of highCpuUsage exceeded: current value 292.000000	2012-05-24 19:5...	2012-05-24 19:5...	1
	⚠	10.0.31.47	/boot	/Perf/Snmp	threshold of testUsage exceeded: current value 29280.000000	2012-05-24 20:3...	2012-06-03 10:3...	2600
	⚠	10.0.31.47	/	/Perf/Snmp	threshold of testUsage exceeded: current value 6303410.000000	2012-05-24 20:3...	2012-06-03 10:3...	2600
	ⓘ	esx-11.mgmt.t...	vc-mgmt_v...	/VMware/G...	Power state from modeling is poweredOff	2012-05-30 09:2...	2012-06-03 09:4...	22
	ⓘ	esx-11.mgmt.t...	vc-mgmt_v...	/VMware/G...	Power state from modeling is poweredOff	2012-05-31 09:2...	2012-06-03 09:4...	16
	ⓘ	server7.mgmt....		/VMware	VIM account password was changed on host server7.mgmt.test	2012-06-03 09:4...	2012-06-03 09:4...	1
	ⓘ	server8.mgmt....		/VMware	VIM account password was changed on host server8.mgmt.test	2012-06-03 09:4...	2012-06-03 09:4...	1
	ⓘ	server5.mgmt....		/VMware	VIM account password was changed on host server5.mgmt.test	2012-06-03 09:3...	2012-06-03 09:3...	1
	ⓘ	server2.mgmt....		/VMware	VIM account password was changed on host server2.mgmt.test	2012-06-03 09:3...	2012-06-03 09:3...	1
	ⓘ	server1.mgmt....		/VMware	VIM account password was changed on host server1.mgmt.test	2012-06-03 09:3...	2012-06-03 09:3...	1
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1/b...	/Change/Se...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
	ⓘ	10.0.18.3	chassis-1	/Change/Se...	operState attribute is operable	2012-05-25 08:0...	2012-06-03 08:0...	23

DISPLAYING 1 - 25 OF 25 ROWS

Event Filtering

Underneath the column heading bar lies the event filtering bar. Since the status and severity columns have a finite list of attributes, they are filtered using a set of check boxes. The initial list of all events can thus easily be trimmed down to only the most severe event list. Figure 6-39 shows the reduced list of events by selecting only the critical severity check box.

Figure 6-39. Event List Filtered by Critical Severity



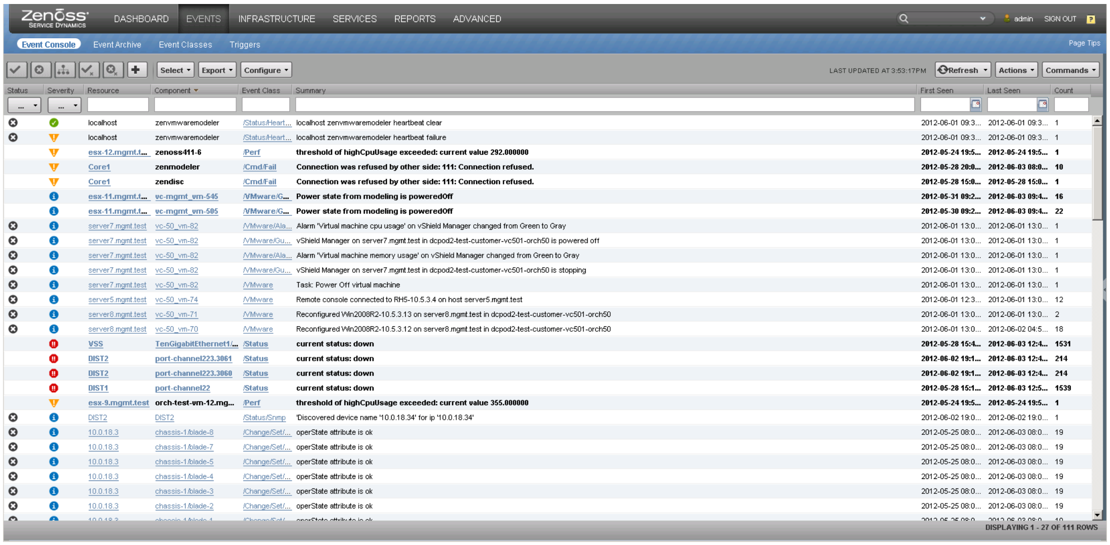
LAST UPDATED AT 2:12:01PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
	‼	VSS	TenGigabitEthernet1/...	/Status	current status: down	2012-05-28 15:4...	2012-06-03 11:0...	1511
	‼	DIST2	port-channel223.3061	/Status	current status: down	2012-06-02 19:1...	2012-06-03 11:0...	194
	‼	DIST2	port-channel223.3060	/Status	current status: down	2012-06-02 19:1...	2012-06-03 11:0...	194
	‼	DIST1	port-channel22	/Status	current status: down	2012-05-28 15:1...	2012-06-03 11:0...	1518

DISPLAYING 1 - 4 OF 4 ROWS

The remaining column filters are implemented with a text-based filtering capability. When entering text into these filtering fields, the dashboard immediately reduces the list of events according to the current text match. As the filtering text is typed, the dashboard reacts dynamically and the filter becomes increasingly more restrictive. Figure 6-40 shows a list of events with various severities and devices.

Figure 6-40. Pre-filtered Event List



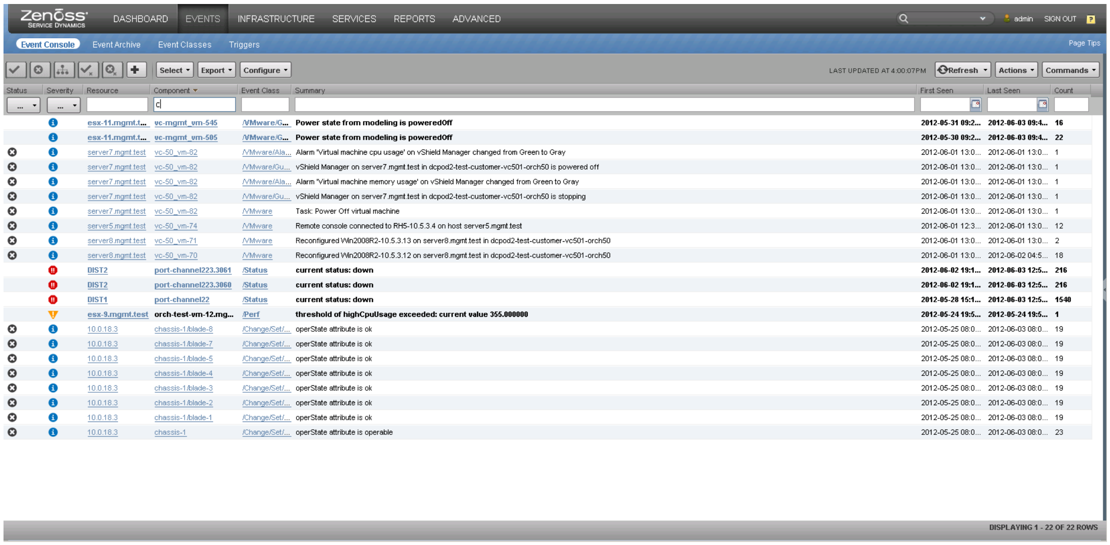
LAST UPDATED AT 3:53:17PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
⊗	✓	localhost	zenvmwaremodeler	/Status/Heart...	localhost zenvmwaremodeler heartbeat clear	2012-06-01 09:3...	2012-06-01 09:3...	1
⊗	!	localhost	zenvmwaremodeler	/Status/Heart...	localhost zenvmwaremodeler heartbeat failure	2012-06-01 09:3...	2012-06-01 09:3...	1
	!	esx-12.mgmt.t...	zenoss411-6	/Perf	threshold of highCpuUsage exceeded: current value 292.000000	2012-05-24 19:5...	2012-05-24 19:5...	1
	!	Core1	zenmodeler	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-28 20:0...	2012-06-03 08:0...	10
	!	Core1	zendisc	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-28 15:0...	2012-05-28 15:0...	1
	ℹ	esx-11.mgmt.t...	vc-mgmt_vm-545	/VMware/G...	Power state from modeling is poweredOff	2012-05-31 09:2...	2012-06-03 09:4...	16
	ℹ	esx-11.mgmt.t...	vc-mgmt_vm-505	/VMware/G...	Power state from modeling is poweredOff	2012-05-30 09:2...	2012-06-03 09:4...	22
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Ala...	Alarm 'Virtual machine cpu usage' on vShield Manager changed from Green to Gray	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Gu...	vShield Manager on server7.mgmt.test in dcpod2-test-customer-vc501-orch50 is powered off	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Ala...	Alarm 'Virtual machine memory usage' on vShield Manager changed from Green to Gray	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Gu...	vShield Manager on server7.mgmt.test in dcpod2-test-customer-vc501-orch50 is stopping	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware	Task: Power Off virtual machine	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server5.mgmt.test	vc-50_vm-74	/VMware	Remote console connected to RH5-10.5.3.4 on host server5.mgmt.test	2012-06-01 12:3...	2012-06-01 13:0...	12
⊗	ℹ	server8.mgmt.test	vc-50_vm-71	/VMware	Reconfigured Win2008R2-10.5.3.13 on server8.mgmt.test in dcpod2-test-customer-vc501-orch50	2012-06-01 13:0...	2012-06-01 13:0...	2
⊗	ℹ	server8.mgmt.test	vc-50_vm-70	/VMware	Reconfigured Win2008R2-10.5.3.12 on server8.mgmt.test in dcpod2-test-customer-vc501-orch50	2012-06-01 13:0...	2012-06-02 04:5...	18
	‼	VSS	TenGigabitEthernet1/...	/Status	current status: down	2012-05-28 15:4...	2012-06-03 12:4...	1531
	‼	DIST2	port-channel223.3061	/Status	current status: down	2012-06-02 19:1...	2012-06-03 12:4...	214
	‼	DIST2	port-channel223.3060	/Status	current status: down	2012-06-02 19:1...	2012-06-03 12:4...	214
	‼	DIST1	port-channel22	/Status	current status: down	2012-05-28 15:1...	2012-06-03 12:5...	1539
	!	esx-9.mgmt.test	orch-test-vm-12.mg...	/Perf	threshold of highCpuUsage exceeded: current value 355.000000	2012-05-24 19:5...	2012-05-24 19:5...	1
⊗	ℹ	DIST2	DIST2	/Status/Snmp	'Discovered device name '10.0.18.34' for ip '10.0.18.34'	2012-06-02 19:0...	2012-06-02 19:0...	1
⊗	ℹ	10.0.18.3	chassis-1/blade-8	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-7	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-5	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-4	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-3	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-2	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-1	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19

DISPLAYING 1 - 27 OF 111 ROWS

To demonstrate the subsequent filtering, Figure 6-41, Figure 6-42, and Figure 6-43 show the same event list filtered dynamically on the component column for text "c", "ch", and "chan" respectively.

Figure 6-41. Event List Filtered on Component "c"

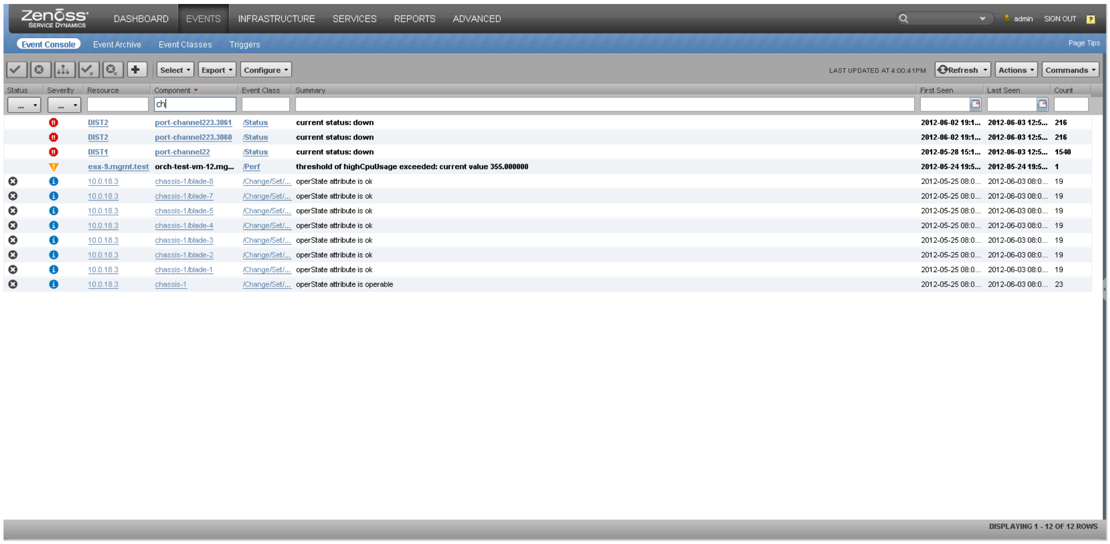


LAST UPDATED AT 4:00:07PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
...	...		c					
	ℹ	**esx-11.mgmt.t...**	**vc-mgmt_vm-545**	**/VMware/G...**	**Power state from modeling is poweredOff**	**2012-05-31 09:2...**	**2012-06-03 09:4...**	**16**
	ℹ	**esx-11.mgmt.t...**	**vc-mgmt_vm-505**	**/VMware/G...**	**Power state from modeling is poweredOff**	**2012-05-30 09:2...**	**2012-06-03 09:4...**	**22**
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Ala...	Alarm 'Virtual machine cpu usage' on vShield Manager changed from Green to Gray	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Gu...	vShield Manager on server7.mgmt.test in dcpod2-test-customer-vc501-orch50 is powered off	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Ala...	Alarm 'Virtual machine memory usage' on vShield Manager changed from Green to Gray	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware/Gu...	vShield Manager on server7.mgmt.test in dcpod2-test-customer-vc501-orch50 is stopping	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server7.mgmt.test	vc-50_vm-82	/VMware	Task: Power Off virtual machine	2012-06-01 13:0...	2012-06-01 13:0...	1
⊗	ℹ	server5.mgmt.test	vc-50_vm-74	/VMware	Remote console connected to RH5-10.5.3.4 on host server5.mgmt.test	2012-06-01 12:3...	2012-06-01 13:0...	12
⊗	ℹ	server8.mgmt.test	vc-50_vm-71	/VMware	Reconfigured Win2008R2-10.5.3.13 on server8.mgmt.test in dcpod2-test-customer-vc501-orch50	2012-06-01 13:0...	2012-06-01 13:0...	2
⊗	ℹ	server8.mgmt.test	vc-50_vm-70	/VMware	Reconfigured Win2008R2-10.5.3.12 on server8.mgmt.test in dcpod2-test-customer-vc501-orch50	2012-06-01 13:0...	2012-06-02 04:5...	18
	‼	**DIST2**	**port-channel223.3061**	**/Status**	**current status: down**	**2012-06-02 19:1...**	**2012-06-03 12:5...**	**216**
	‼	**DIST2**	**port-channel223.3060**	**/Status**	**current status: down**	**2012-06-02 19:1...**	**2012-06-03 12:5...**	**216**
	‼	**DIST1**	**port-channel22**	**/Status**	**current status: down**	**2012-05-28 15:1...**	**2012-06-03 12:5...**	**1540**
	⚠	**esx-9.mgmt.test**	**orch-test-vm-12.mg...**	**/Perf**	**threshold of highCpuUsage exceeded: current value 355.000000**	**2012-05-24 19:5...**	**2012-05-24 19:5...**	**1**
⊗	ℹ	10.0.18.3	chassis-1/blade-8	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-7	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-5	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-4	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-3	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-2	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-1	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1	/Change/Set/...	operState attribute is operable	2012-05-25 08:0...	2012-06-03 08:0...	23

DISPLAYING 1 - 22 OF 22 ROWS

Figure 6-42. Event List Filtered on Component "ch"

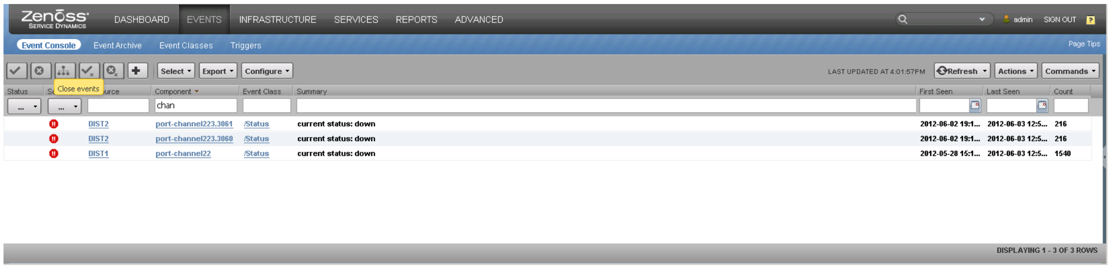


LAST UPDATED AT 4:00:41PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
...	...		ch					
	‼	**DIST2**	**port-channel223.3061**	**/Status**	**current status: down**	**2012-06-02 19:1...**	**2012-06-03 12:5...**	**216**
	‼	**DIST2**	**port-channel223.3060**	**/Status**	**current status: down**	**2012-06-02 19:1...**	**2012-06-03 12:5...**	**216**
	‼	**DIST1**	**port-channel22**	**/Status**	**current status: down**	**2012-05-28 15:1...**	**2012-06-03 12:5...**	**1540**
	⚠	**esx-9.mgmt.test**	**orch-test-vm-12.mg...**	**/Perf**	**threshold of highCpuUsage exceeded: current value 355.000000**	**2012-05-24 19:5...**	**2012-05-24 19:5...**	**1**
⊗	ℹ	10.0.18.3	chassis-1/blade-8	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-7	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-5	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-4	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-3	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-2	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1/blade-1	/Change/Set/...	operState attribute is ok	2012-05-25 08:0...	2012-06-03 08:0...	19
⊗	ℹ	10.0.18.3	chassis-1	/Change/Set/...	operState attribute is operable	2012-05-25 08:0...	2012-06-03 08:0...	23

DISPLAYING 1 - 12 OF 12 ROWS

Figure 6-43. Event List Filtered on Component "chan"



LAST UPDATED AT 4:01:57PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
...	...		chan					
	‼	**DIST2**	**port-channel223.3061**	**/Status**	**current status: down**	**2012-06-02 19:1...**	**2012-06-03 12:5...**	**216**
	‼	**DIST2**	**port-channel223.3060**	**/Status**	**current status: down**	**2012-06-02 19:1...**	**2012-06-03 12:5...**	**216**
	‼	**DIST1**	**port-channel22**	**/Status**	**current status: down**	**2012-05-28 15:1...**	**2012-06-03 12:5...**	**1540**

DISPLAYING 1 - 3 OF 3 ROWS

While the Events dashboard has the potential to present a deluge of event information, it also provides the necessary tools to a cloud operator to sort and filter that information quickly.

6.3 Consolidated Resource Monitoring of VMDC Components

CLSA-VMDC enables operator personnel to monitor VMDC network, service, and compute resources to ensure a highly available cloud service. Resources can be monitored at an aggregate level using several dashboards, or they can be monitored and inspected individually for more detailed information.

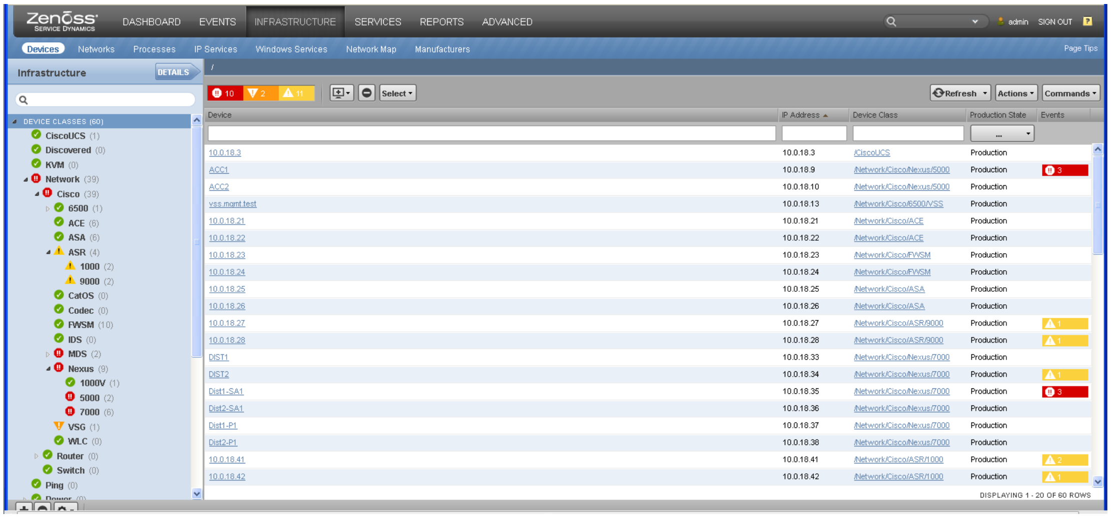
The following sections illustrate the more detailed monitoring views of VMDC network, services, compute, and storage components.

6.3.1 Network Resource Monitoring

To access a device, select the **Infrastructure** tab and then choose a device from the overall device listing in the main window, or drill down into a device class and select a particular device.

To access the Nexus 5000 device ACC2, select the **Infrastructure** tab and then choose the ACC2 device from the overall device listing in the main window.

Figure 6-44. List of All Devices

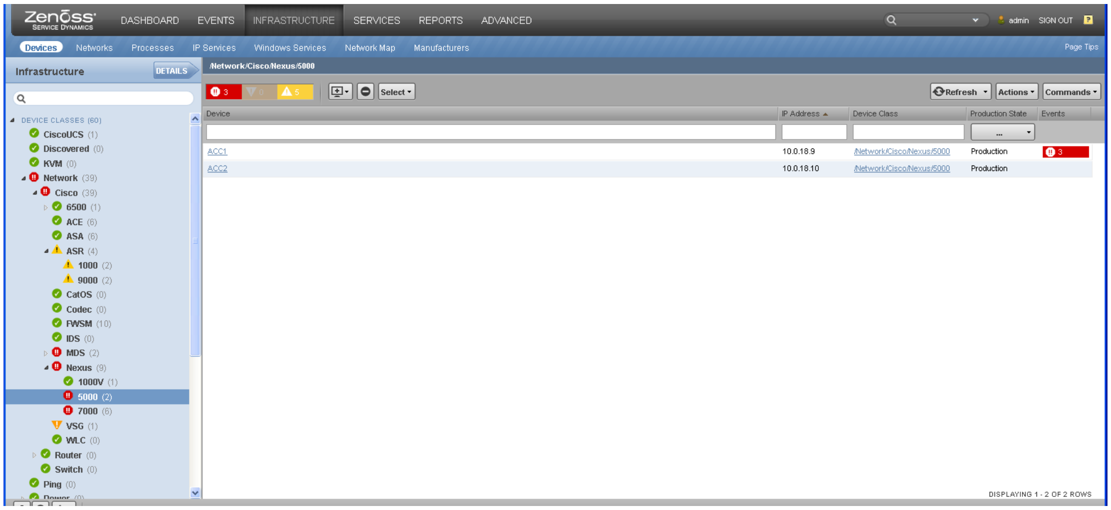


Device	IP Address ▲	Device Class	Production State	Events
10.0.18.3	10.0.18.3	/CiscoUCS	Production	
ACC1	10.0.18.9	/Network/Cisco/Nexus/5000	Production	‼ 3
ACC2	10.0.18.10	/Network/Cisco/Nexus/5000	Production	
vss.mgmt.test	10.0.18.13	/Network/Cisco/6500/VSS	Production	
10.0.18.21	10.0.18.21	/Network/Cisco/ACE	Production	
10.0.18.22	10.0.18.22	/Network/Cisco/ACE	Production	
10.0.18.23	10.0.18.23	/Network/Cisco/FWSM	Production	
10.0.18.24	10.0.18.24	/Network/Cisco/FWSM	Production	
10.0.18.25	10.0.18.25	/Network/Cisco/ASA	Production	
10.0.18.26	10.0.18.26	/Network/Cisco/ASA	Production	
10.0.18.27	10.0.18.27	/Network/Cisco/ASR/9000	Production	⚠ 1
10.0.18.28	10.0.18.28	/Network/Cisco/ASR/9000	Production	⚠ 1
DIST1	10.0.18.33	/Network/Cisco/Nexus/7000	Production	
DIST2	10.0.18.34	/Network/Cisco/Nexus/7000	Production	⚠ 1
Dist1-SA1	10.0.18.35	/Network/Cisco/Nexus/7000	Production	‼ 3
Dist2-SA1	10.0.18.36	/Network/Cisco/Nexus/7000	Production	
Dist1-P1	10.0.18.37	/Network/Cisco/Nexus/7000	Production	
Dist2-P1	10.0.18.38	/Network/Cisco/Nexus/7000	Production	
10.0.18.41	10.0.18.41	/Network/Cisco/ASR/1000	Production	⚠ 2
10.0.18.42	10.0.18.42	/Network/Cisco/ASR/1000	Production	⚠ 1

DISPLAYING 1 - 20 OF 60 ROWS

Or, drill down into the Nexus 5000 device class and select the device ACC2 from there.

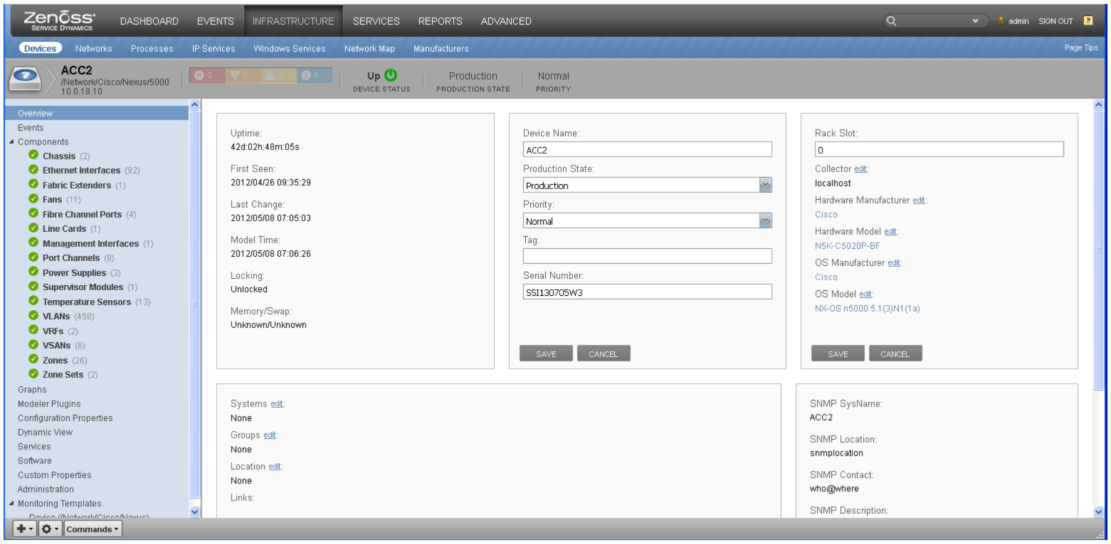
Figure 6-45. List of Nexus 5000s



Device	IP Address ▲	Device Class	Production State	Events
ACC1	10.0.18.9	/Network/Cisco/Nexus/5000	Production	3
ACC2	10.0.18.10	/Network/Cisco/Nexus/5000	Production	

The initial display shows all of the components that are monitored on the left side and the overview of the device in the main window. The components monitored vary from device to device. The overview contains some classification and status information. Some of these fields are editable such as Device Name.

Figure 6-46. Nexus 5000



ACC2
/Network/Cisco/Nexus/5000
10.0.18.10

Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

Uptime: 42d:02h:48m:05s
First Seen: 2012/04/26 09:35:29
Last Change: 2012/05/08 07:05:03
Model Time: 2012/05/08 07:06:26
Locking: Unlocked
Memory/Swap: Unknown/Unknown

Device Name: ACC2
Production State: Production
Priority: Normal
Tag:
Serial Number: SSI130705W3

Rack Slot: 0
Collector edit: localhost
Hardware Manufacturer edit: Cisco
Hardware Model edit: N5K-C5020P-BF
OS Manufacturer edit: Cisco
OS Model edit: NX-OS n5000 5.1(3)N1(1a)

Systems edit: None
Groups edit: None
Location edit: None
Links:

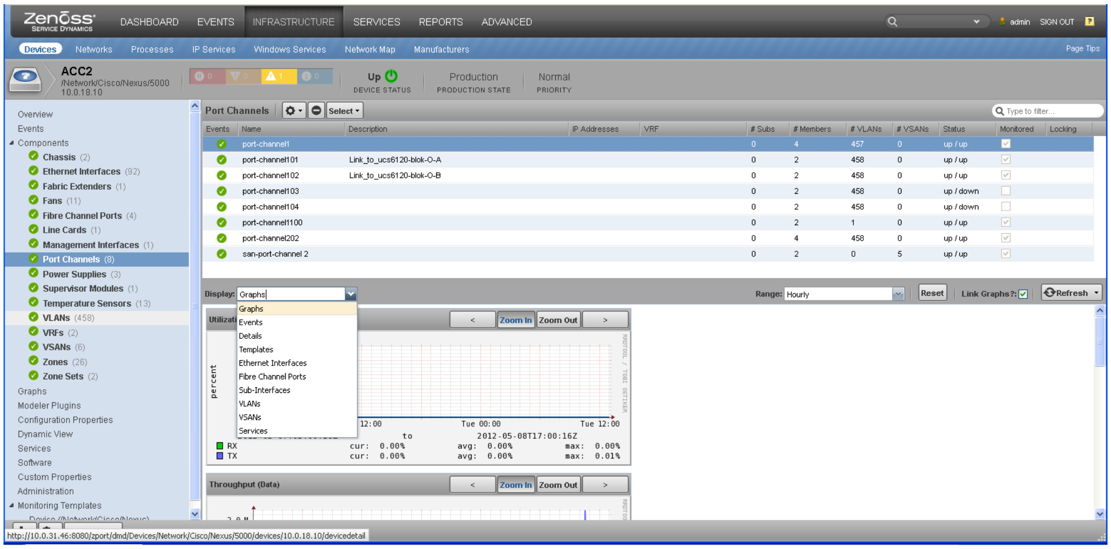
SNMP SysName: ACC2
SNMP Location: snmplocation
SNMP Contact: who@where
SNMP Description:

The status bar under the menu tabs contains the device name, device class, device IP address, event rainbow status display, overall device status indicator, production state, and priority. The current status of this device is Up and there are no events. Additional information for each component can be displayed by clicking on the component listing. Clicking **Port Channels** provides a listing of all of the port-channels configured on the device.

For each port-channel, there is additional information displayed that includes the Event status, Name, Description, IP Address, Virtual Routing and Forwarding (VRF) interface, #Subs (number of sub-interfaces), #Members (number of Ethernet interfaces), #VLANs, #VSANs, Status, Monitored, and Locking. By viewing port-channel1, the user can see that there are four Members and 457 VLANs associated with this port-channel. Also seen is that the Events information is green and the status is Up/Up. The status indicates that the port-channel is administrative up and there is an active link.

Beneath the port-channels listing, there is another display window. This window defaults to graphs. The graphs displayed vary depending on the component. The network interface components display traffic related graphs, utilization, throughput, and errors. Clicking the **down** arrow next to the graphs displays a list of other display options. Under the port-channels component, there is a listing of all sub-components that are related to port-channels, plus the standard display options.

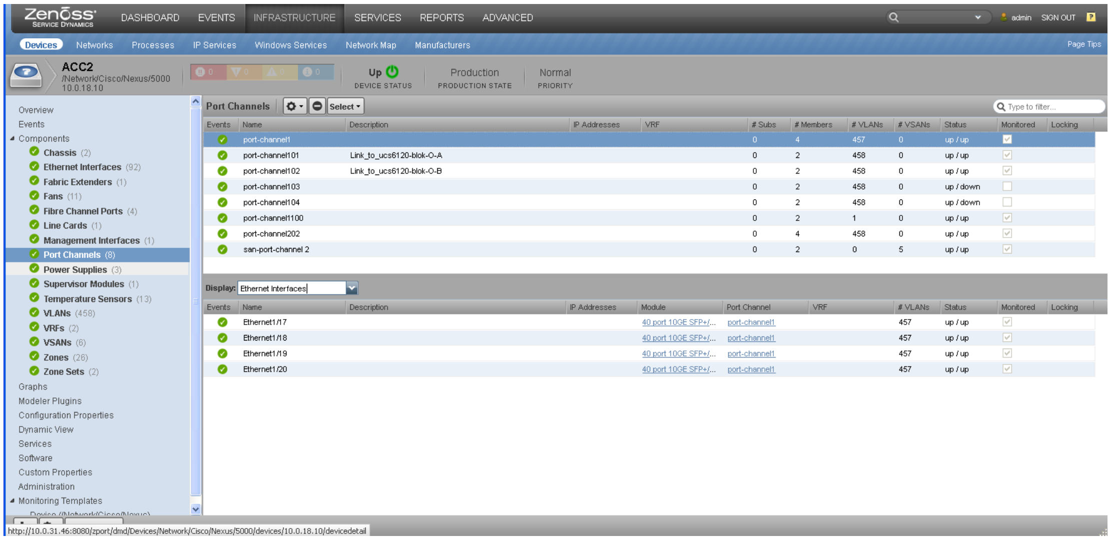
Figure 6-47. Nexus 5000 Port-channels



Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
✓	port-channel1				0	4	457	0	up / up	☑	
✓	port-channel101	Link_to_ucs6120-blok-O-A			0	2	458	0	up / up	☑	
✓	port-channel102	Link_to_ucs6120-blok-O-B			0	2	458	0	up / up	☑	
✓	port-channel103				0	2	458	0	up / down	□	
✓	port-channel104				0	2	458	0	up / down	□	
✓	port-channel1100				0	2	1	0	up / up	☑	
✓	port-channel202				0	4	458	0	up / up	☑	
✓	san-port-channel 2				0	2	0	5	up / up	☑	

Selecting **Ethernet Interfaces** displays all of the Ethernet interfaces related to port-channel1.

Figure 6-48. Related Ethernet Interfaces

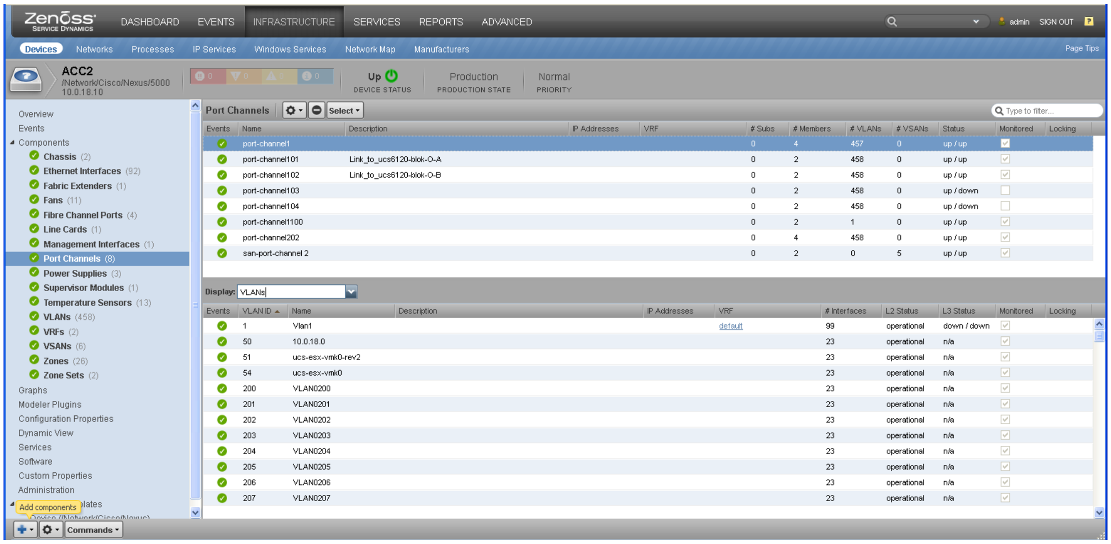


Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
✓	port-channel1				0	4	457	0	up / up	☑	
✓	port-channel101	Link_to_ucs6120-blok-O-A			0	2	458	0	up / up	☑	
✓	port-channel102	Link_to_ucs6120-blok-O-B			0	2	458	0	up / up	☑	
✓	port-channel103				0	2	458	0	up / down	□	
✓	port-channel104				0	2	458	0	up / down	□	
✓	port-channel1100				0	2	1	0	up / up	☑	
✓	port-channel202				0	4	458	0	up / up	☑	
✓	san-port-channel 2				0	2	0	5	up / up	☑	

Events	Name	Description	IP Addresses	Module	Port Channel	VRF	# VLANs	Status	Monitored	Locking
✓	Ethernet1/17			40 port 10GE SFP+/...	port-channel1		457	up / up	☑	
✓	Ethernet1/18			40 port 10GE SFP+/...	port-channel1		457	up / up	☑	
✓	Ethernet1/19			40 port 10GE SFP+/...	port-channel1		457	up / up	☑	
✓	Ethernet1/20			40 port 10GE SFP+/...	port-channel1		457	up / up	☑	

Selecting **VLANs** displays all of the VLANs associated with port-channel1.

Figure 6-49.   Related VLANs



ACC2
/Network/Cisco/Nexus/5000
10.0.18.10

Up — DEVICE STATUS | Production — PRODUCTION STATE | Normal — PRIORITY

Port Channels

Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
✓	port-channel1				0	4	457	0	up / up	☑	
✓	port-channel101	Link_to_ucs6120-blok-O-A			0	2	458	0	up / up	☑	
✓	port-channel102	Link_to_ucs6120-blok-O-B			0	2	458	0	up / up	☑	
✓	port-channel103				0	2	458	0	up / down	□	
✓	port-channel104				0	2	458	0	up / down	□	
✓	port-channel1100				0	2	1	0	up / up	☑	
✓	port-channel202				0	4	458	0	up / up	☑	
✓	san-port-channel 2				0	2	0	5	up / up	☑	

Display: VLANs

Events	VLAN ID ▲	Name	Description	IP Addresses	VRF	# Interfaces	L2 Status	L3 Status	Monitored	Locking
✓	1	Vlan1			default	99	operational	down / down	☑	
✓	50	10.0.18.0				23	operational	n/a	☑	
✓	51	ucs-esx-vmk0-rev2				23	operational	n/a	☑	
✓	54	ucs-esx-vmk0				23	operational	n/a	☑	
✓	200	VLAN0200				23	operational	n/a	☑	
✓	201	VLAN0201				23	operational	n/a	☑	
✓	202	VLAN0202				23	operational	n/a	☑	
✓	203	VLAN0203				23	operational	n/a	☑	
✓	204	VLAN0204				23	operational	n/a	☑	
✓	205	VLAN0205				23	operational	n/a	☑	
✓	206	VLAN0206				23	operational	n/a	☑	
✓	207	VLAN0207				23	operational	n/a	☑	

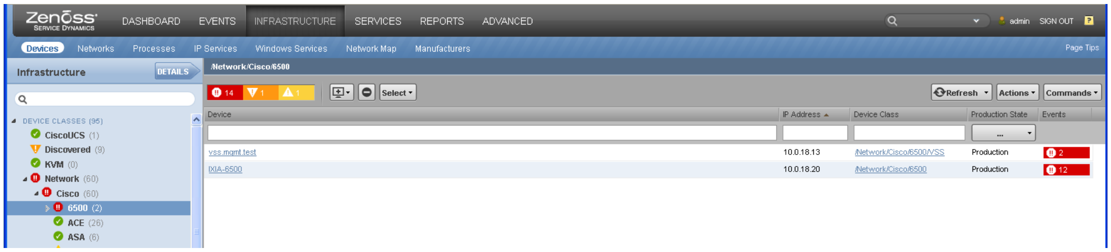
Fault Detection

Zenoss can be used for an SNMP Trap receiver to actively monitor the status of devices. All devices need to be configured to enable SNMP Traps and to send traps to the Zenoss server; what needs to be configured varies from device to device.

The Zenoss server processes the SNMP Traps sent from a monitored device and triggers an event. These events typically arrive marked as Critical (red), Error (orange), Warning (yellow), Info (blue), and Debug and Clear. By default, only the Critical, Error, Warning, and Info events are turned on and are represented in the rainbow bar seen under the Infrastructure and Devices tab. Event notification can be seen throughout the Zenoss application, including the main dashboard.

Only the unacknowledged events are displayed. Clicking on the event opens the Events view; the view depends on the window that is being viewed. For example, clicking the rainbow event bar under the 6500 device class directs the user to the Events under the 6500 and all events for all of the 6500s are displayed.

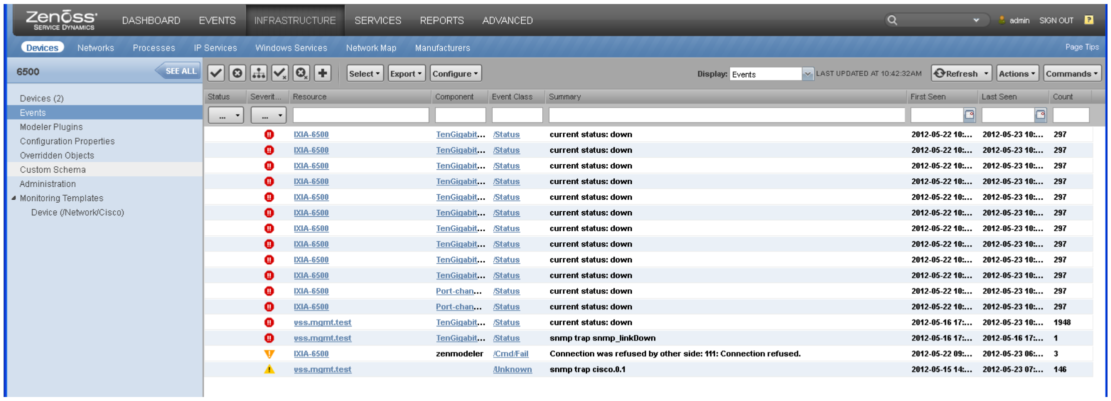
Figure 6-50.   Events Under Device Class Cisco 6500



/Network/Cisco/6500

Device	IP Address ▲	Device Class	Production State	Events
vss.mgmt.test	10.0.18.13	/Network/Cisco/6500/VSS	Production	2
IXIA-6500	10.0.18.20	/Network/Cisco/6500	Production	12

Figure 6-51. Every Event Under Device Class Cisco 6500

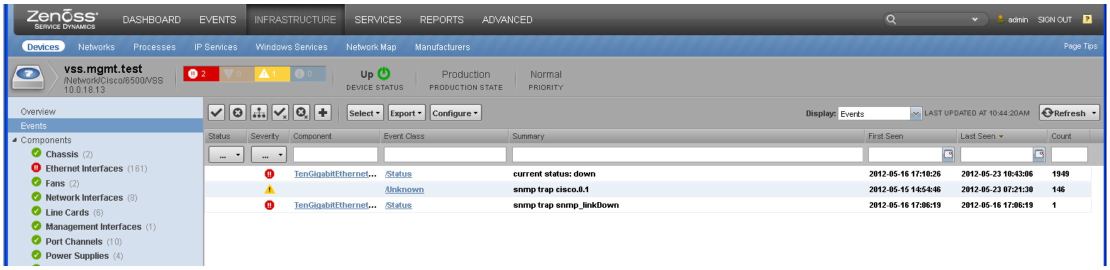


Display: Events LAST UPDATED AT 10:42:32AM

Status	Severit...	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	TenGigabit...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	Port-chan...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	IXIA-6500	Port-chan...	/Status	current status: down	2012-05-22 10:...	2012-05-23 10:...	297
	❗	vss.mgmt.test	TenGigabit...	/Status	current status: down	2012-05-16 17:...	2012-05-23 10:...	1948
	❗	vss.mgmt.test	TenGigabit...	/Status	snmp trap snmp_linkDown	2012-05-16 17:...	2012-05-16 17:...	1
	▼	IXIA-6500	zenmodeler	/Cmd/Fail	Connection was refused by other side: 111: Connection refused.	2012-05-22 09:...	2012-05-23 06:...	3
	⚠	vss.mgmt.test		/Unknown	snmp trap cisco.0.1	2012-05-15 14:...	2012-05-23 07:...	146

Clicking the events next to the vss.mgmt.test device directs the user to the Events under that device.

Figure 6-52. Events Under Single Device



vss.mgmt.test
/Network/Cisco/6500/VSS
10.0.18.13

❗ 2 ▼ 0 ⚠ 1 ⓘ 0

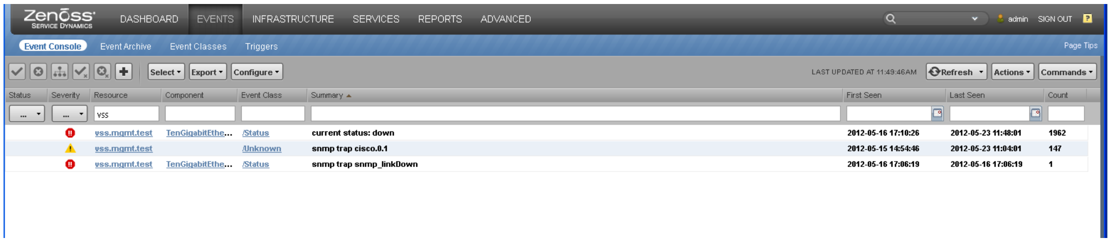
Up — DEVICE STATUS; Production — PRODUCTION STATE; Normal — PRIORITY

Display: Events LAST UPDATED AT 10:44:20AM

Status	Severity	Component	Event Class	Summary	First Seen	Last Seen	Count
	❗	TenGigabitEthernet...	/Status	current status: down	2012-05-16 17:10:26	2012-05-23 10:43:06	1949
	⚠		/Unknown	snmp trap cisco.0.1	2012-05-15 14:54:46	2012-05-23 07:21:30	146
	❗	TenGigabitEthernet...	/Status	snmp trap snmp_linkDown	2012-05-16 17:06:19	2012-05-16 17:06:19	1

Events for the entire Zenoss server can be seen by clicking the Events tab. Events can be sorted or filtered.

Figure 6-53. Events Sorted by Resource



LAST UPDATED AT 11:49:46AM

Resource filter: vss

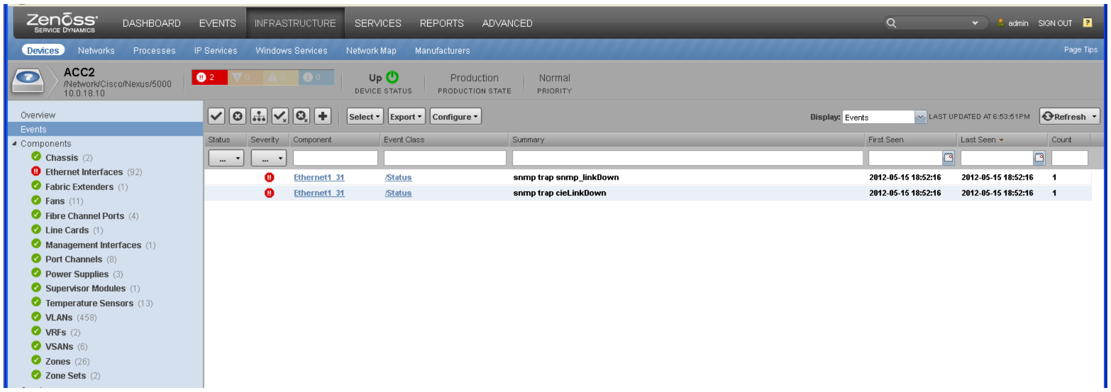
Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen	Count
	❗	vss.mgmt.test	TenGigabitEthe...	/Status	current status: down	2012-05-16 17:10:26	2012-05-23 11:48:01	1962
	⚠	vss.mgmt.test		/Unknown	snmp trap cisco.0.1	2012-05-15 14:54:46	2012-05-23 11:04:01	147
	❗	vss.mgmt.test	TenGigabitEthe...	/Status	snmp trap snmp_linkDown	2012-05-16 17:06:19	2012-05-16 17:06:19	1

When a link goes down, if properly configured, a Cisco device sends out SNMP Traps. These traps can be viewed locally on the monitored device.

```
2012 May 15 18:54:23 ACC2 %ETH_PORT_CHANNEL-5-PORT_DOWN: port-channel202: Ethernet1/31
 is down
2012 May 15 18:54:23 ACC2 %ETHPORT-5-IF_DOWN_LINK_FAILURE: Interface Ethernet1/31 is
 down (Link failure)
```

Upon reception of these two traps, Zenoss triggers an event for each trap. The affected component's status indicator turns red.

Figure 6-54. Link Down Event



ACC2
/Network/Cisco/Nexus/5000
10.0.18.10

2 0 0 0

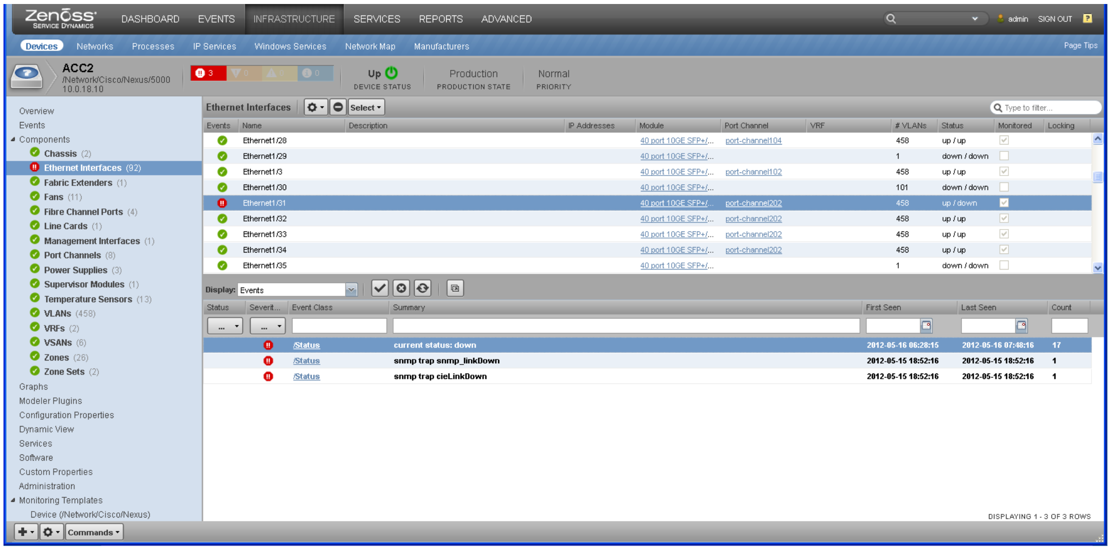
Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY

Display: Events LAST UPDATED AT 6:53:51PM

Status	Severity	Component	Event Class	Summary	First Seen	Last Seen	Count
	●	Ethernet1_31	/Status	snmp trap snmp_linkDown	2012-05-15 18:52:16	2012-05-15 18:52:16	1
	●	Ethernet1_31	/Status	snmp trap cieLinkDown	2012-05-15 18:52:16	2012-05-15 18:52:16	1

These same events can be seen under the Ethernet Interfaces component by selecting the interface name and displaying the events. The status of the individual link turns red, and the status now displays Up/Down.

Figure 6-55. Link Down Event Under Component



ACC2
/Network/Cisco/Nexus/5000
10.0.18.10

3 0 0 0

Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY

Ethernet Interfaces

Events	Name	Description	IP Addresses	Module	Port Channel	VRF	# VLANs	Status	Monitored	Locking
✓	Ethernet1/28			40 port 10GE SFP+/...	port-channel104		458	up / up	☑	
✓	Ethernet1/29			40 port 10GE SFP+/...			1	down / down	□	
✓	Ethernet1/3			40 port 10GE SFP+/...	port-channel102		458	up / up	☑	
✓	Ethernet1/30			40 port 10GE SFP+/...			101	down / down	□	
●	Ethernet1/31			40 port 10GE SFP+/...	port-channel202		458	up / down	☑	
✓	Ethernet1/32			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/33			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/34			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/35			40 port 10GE SFP+/...			1	down / down	□	

Display: Events

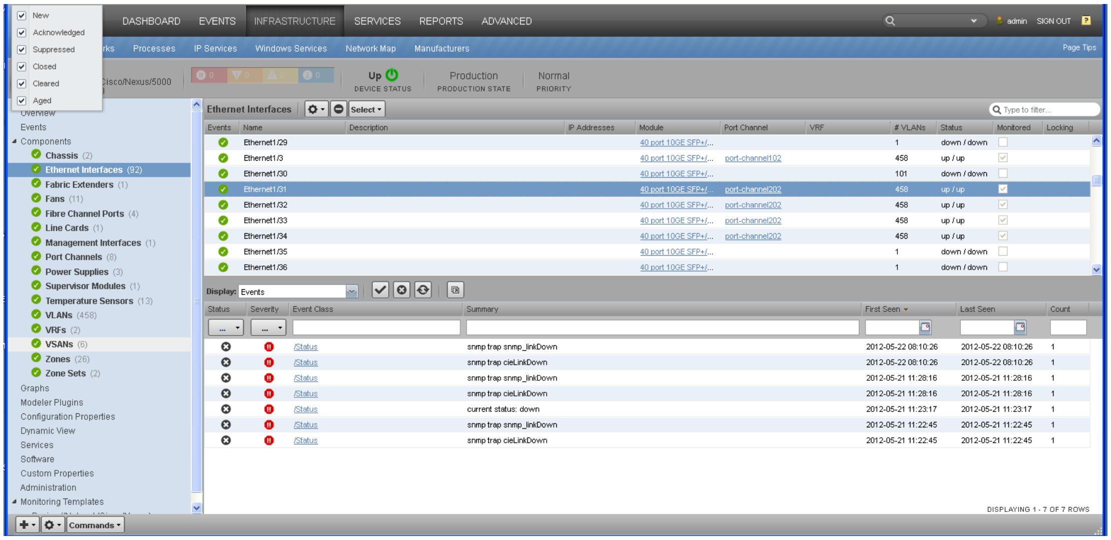
Status	Severit...	Event Class	Summary	First Seen	Last Seen	Count
	●	/Status	current status: down	2012-05-16 06:28:15	2012-05-16 07:48:16	17
	●	/Status	snmp trap snmp_linkDown	2012-05-15 18:52:16	2012-05-15 18:52:16	1
	●	/Status	snmp trap cieLinkDown	2012-05-15 18:52:16	2012-05-15 18:52:16	1

DISPLAYING 1 - 3 OF 3 ROWS

After resolving the issue and bringing the link up, the status indicator turns green and the events are cleared.

By default, the events view only displays current events. This view can be changed by selecting the box under the status and checking the events that the user wishes to see. By checking all of the categories, the user can display a historical listing of events. Note that when a specific event's display settings are changed, this results in a global change for all event listings.

Figure 6-56. Historical Events



Events	Name	Description	IP Addresses	Module	Port Channel	VRF	# VLANs	Status	Monitored	Locking
✓	Ethernet1/29			40 port 10GE SFP+/...			1	down / down	□	
✓	Ethernet1/3			40 port 10GE SFP+/...	port-channel102		458	up / up	☑	
✓	Ethernet1/30			40 port 10GE SFP+/...			101	down / down	□	
✓	Ethernet1/31			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/32			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/33			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/34			40 port 10GE SFP+/...	port-channel202		458	up / up	☑	
✓	Ethernet1/35			40 port 10GE SFP+/...			1	down / down	□	
✓	Ethernet1/36			40 port 10GE SFP+/...			1	down / down	□	

Status	Severity	Event Class	Summary	First Seen	Last Seen	Count
☒	●	/Status	snmp trap snmp_linkDown	2012-05-22 08:10:26	2012-05-22 08:10:26	1
☒	●	/Status	snmp trap cieLinkDown	2012-05-22 08:10:26	2012-05-22 08:10:26	1
☒	●	/Status	snmp trap snmp_linkDown	2012-05-21 11:28:16	2012-05-21 11:28:16	1
☒	●	/Status	snmp trap cieLinkDown	2012-05-21 11:28:16	2012-05-21 11:28:16	1
☒	●	/Status	current status: down	2012-05-21 11:23:17	2012-05-21 11:23:17	1
☒	●	/Status	snmp trap snmp_linkDown	2012-05-21 11:22:45	2012-05-21 11:22:45	1
☒	●	/Status	snmp trap cieLinkDown	2012-05-21 11:22:45	2012-05-21 11:22:45	1

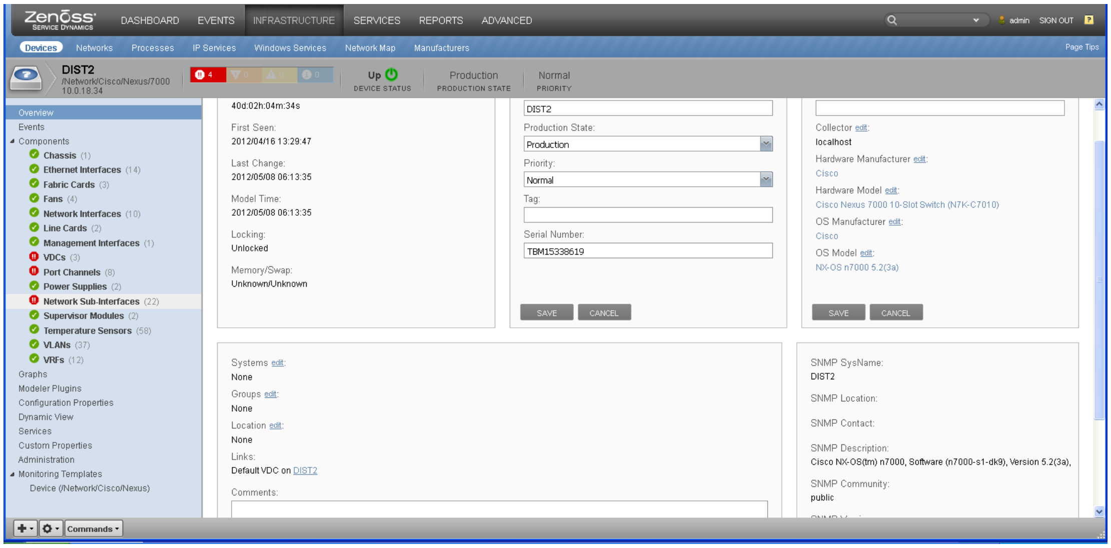
DISPLAYING 1 - 7 OF 7 ROWS

False Fault Detection

At times, there are critical events logged that are not necessarily events that need to be monitored. Some of these could be links that are no longer in use.

On this device, the event rainbow indicates that there are four critical status events and the device status is Up. The user can click one of the status indicators in the event rainbow to display the active events. This would be the same as clicking the Events tab.

Figure 6-57. False Events on Device



40d:02h:04m:34s

First Seen:
2012/04/16 13:29:47

Last Change:
2012/05/08 06:13:35

Model Time:
2012/05/08 06:13:35

Locking:
Unlocked

Memory/Swap:
Unknown/Unknown

DIST2

Production State:
Production

Priority:
Normal

Tag:

Serial Number:
TBM15338619

Collector edit:
localhost

Hardware Manufacturer edit:
Cisco

Hardware Model edit:
Cisco Nexus 7000 10-Slot Switch (N7K-C7010)

OS Manufacturer edit:
Cisco

OS Model edit:
NX-OS n7000 5.2(3a)

Systems edit:
None

Groups edit:
None

Location edit:
None

Links:
Default VDC on DIST2

Comments:

SNMP SysName:
DIST2

SNMP Location:

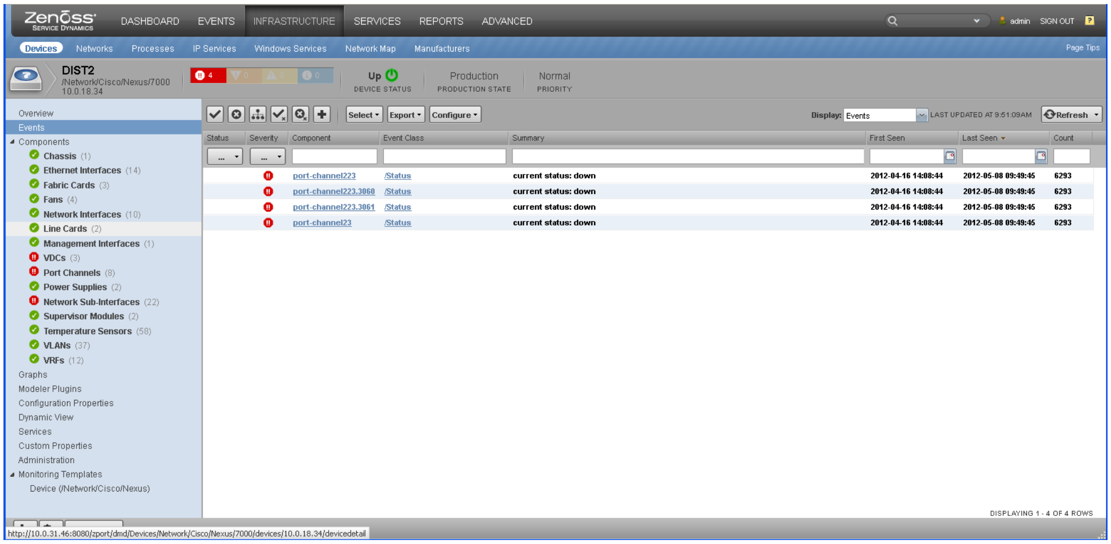
SNMP Contact:

SNMP Description:
Cisco NX-OS(tm) n7000, Software (n7000-s1-dk9), Version 5.2(3a),

SNMP Community:
public

The events displayed show that port-channels 23 and 223 are down and sub-interfaces 223.3060 and 223.6061 are also down. In the component listing on the left, the port-channels and Network Sub-Interfaces components do indicate that there are critical events. To get more details, click one of the events to go to the component section.

Figure 6-58. Port-channel Status Down



DIST2
/Network/Cisco/Nexus/7000
10.0.18.34

Up — DEVICE STATUS | Production — PRODUCTION STATE | Normal — PRIORITY

Display: Events   LAST UPDATED AT 9:51:09AM

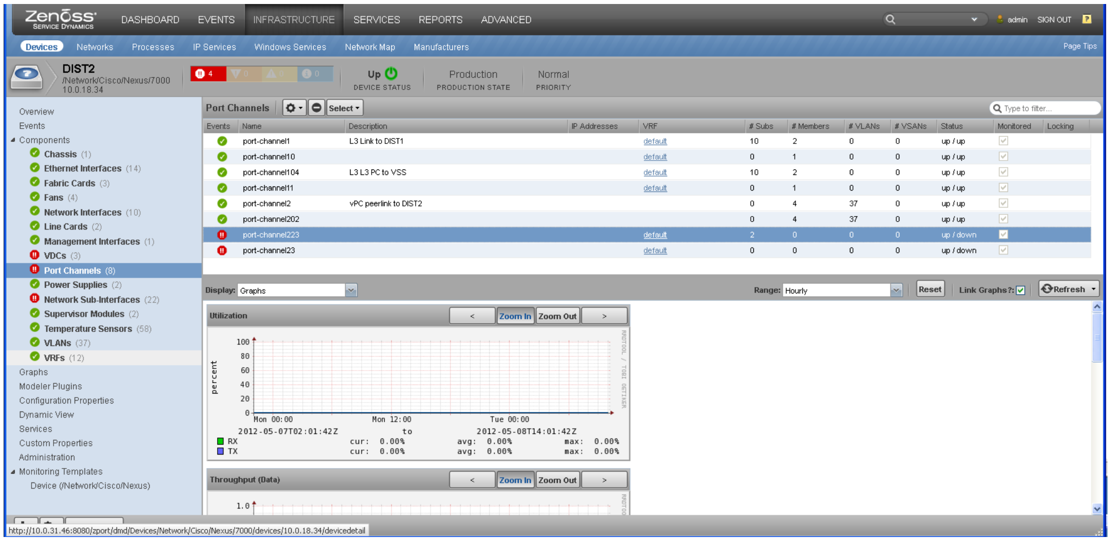
Status	Severity	Component	Event Class	Summary	First Seen	Last Seen	Count
	●	port-channel223	/Status	current status: down	2012-04-16 14:08:44	2012-05-08 09:49:45	6293
	●	port-channel223.3060	/Status	current status: down	2012-04-16 14:08:44	2012-05-08 09:49:45	6293
	●	port-channel223.3061	/Status	current status: down	2012-04-16 14:08:44	2012-05-08 09:49:45	6293
	●	port-channel23	/Status	current status: down	2012-04-16 14:08:44	2012-05-08 09:49:45	6293

DISPLAYING 1 - 4 OF 4 ROWS

http://10.0.31.46:8080/zport/dmd/Devices/Network/Cisco/Nexus/7000/devices/10.0.18.34/devicedetail

In this display, both port-channels are listed as Up/Down. The Up status indicates that the link is administratively up in the device configuration. The Down status indicates that the link is down.

Figure 6-59. Port-channels 23 and 223



DIST2
/Network/Cisco/Nexus/7000
10.0.18.34

Up — DEVICE STATUS | Production — PRODUCTION STATE | Normal — PRIORITY

Port Channels

Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
✓	port-channel1	L3 Link to DIST1		default	10	2	0	0	up / up	☑	
✓	port-channel10			default	0	1	0	0	up / up	☑	
✓	port-channel104	L3 L3 PC to VSS		default	10	2	0	0	up / up	☑	
✓	port-channel11			default	0	1	0	0	up / up	☑	
✓	port-channel2	vPC peerlink to DIST2			0	4	37	0	up / up	☑	
✓	port-channel202				0	4	37	0	up / up	☑	
●	port-channel223			default	2	0	0	0	up / down	☑	
●	port-channel23			default	0	0	0	0	up / down	☑	

Display: Graphs   Range: Hourly   Reset   Link Graphs?: ☑   Refresh

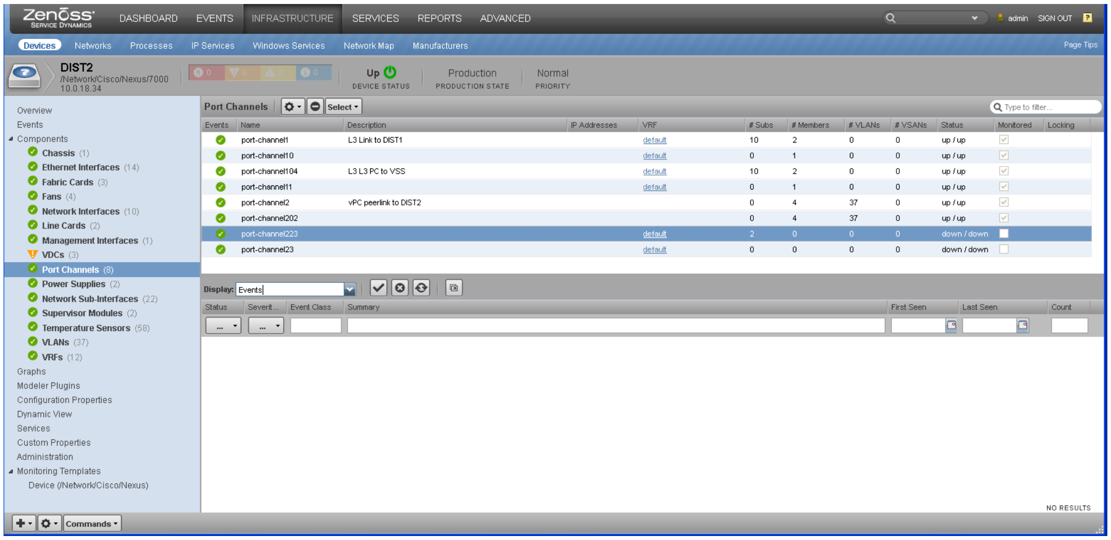
```
2012-05-07T02:01:42Z          to          2012-05-08T14:01:42Z
RX              cur:  0.00%        avg:  0.00%        max:  0.00%
TX              cur:  0.00%        avg:  0.00%        max:  0.00%
```

http://10.0.31.46:8080/zport/dmd/Devices/Network/Cisco/Nexus/7000/devices/10.0.18.34/devicedetail

In this case, the link is no longer used. If the link is not in use, then the link should have been put in the admin down state to prevent any false event notifications. If a device is not administratively up during modeling, then the link is not monitored.

After resolving the configuration issue by putting the port-channels in shutdown, the events can now be Acknowledged and Closed. The device configuration change should force a re-modeling of the device. If the browser is refreshed, then all critical events are cleared.

Figure 6-60. Events Cleared



DIST2
/Network/Cisco/Nexus/7000
10.0.18.34

0 0 0 0

Up DEVICE STATUS
Production PRODUCTION STATE
Normal PRIORITY

Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
✓	port-channel1	L3 Link to DIST1		default	10	2	0	0	up / up	☑	
✓	port-channel10			default	0	1	0	0	up / up	☑	
✓	port-channel104	L3 L3 PC to VSS		default	10	2	0	0	up / up	☑	
✓	port-channel11			default	0	1	0	0	up / up	☑	
✓	port-channel2	vPC peerlink to DIST2			0	4	37	0	up / up	☑	
✓	port-channel202				0	4	37	0	up / up	☑	
✓	port-channel223			default	2	0	0	0	down / down	□	
✓	port-channel23			default	0	0	0	0	down / down	□	

Display: Events

Status	Severit...	Event Class	Summary	First Seen	Last Seen	Count
...	...					

The monitoring of a link can also be turned off from Zenoss by unchecking the monitor box. This is added overhead since the monitor needs to be checked to start monitoring again.

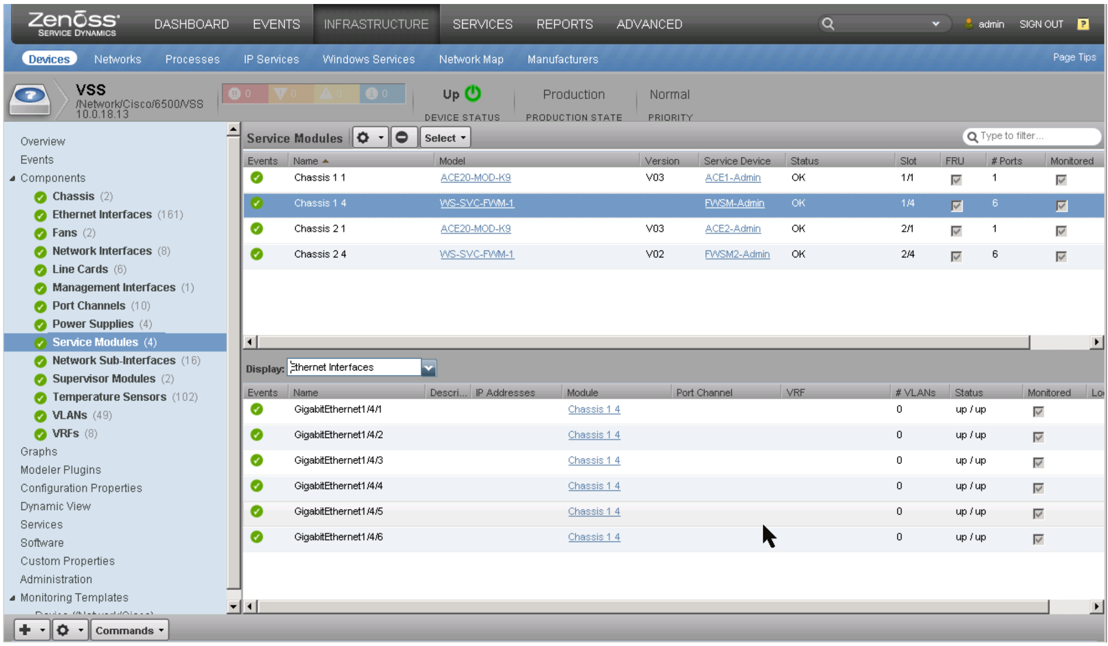
6.3.2 Services Resource Monitoring

The following VMDC services components are monitored by Zenoss CSA:

- VSS - Data Center Services Node (DSN)
- ACE - Application Control Engine
- FWSM - Firewall Services Module
- ASA - Adaptive Security Appliance
- VSG - Virtual Services Gateway

Figure 6-61 shows the service modules' view within the Cisco VSS DSN.

Figure 6-61. VSS DSN - Service Modules View



VSS
/Network/Cisco/6500/VSS
10.0.18.13

Up — DEVICE STATUS; Production — PRODUCTION STATE; Normal — PRIORITY

Service Modules

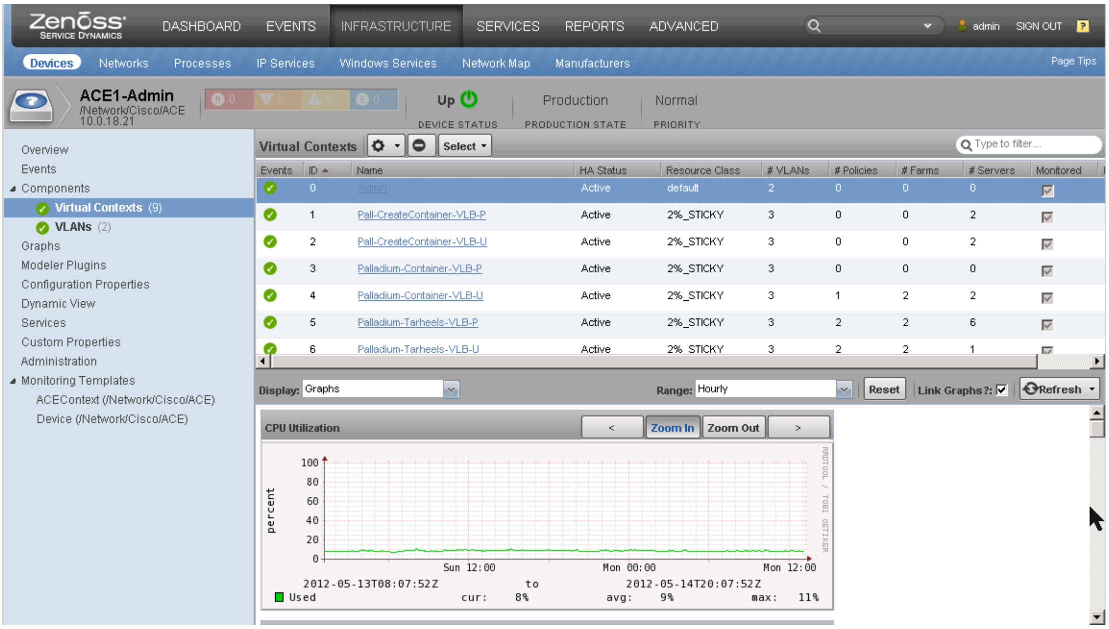
Events	Name	Model	Version	Service Device	Status	Slot	FRU	# Ports	Monitored
✓	Chassis 1 1	ACE20-MOD-K9	V03	ACE1-Admin	OK	1/1	☑	1	☑
✓	Chassis 1 4	WS-SVC-FWM-1		FWSM-Admin	OK	1/4	☑	6	☑
✓	Chassis 2 1	ACE20-MOD-K9	V03	ACE2-Admin	OK	2/1	☑	1	☑
✓	Chassis 2 4	WS-SVC-FWM-1	V02	FWSM2-Admin	OK	2/4	☑	6	☑

Display: Ethernet Interfaces

Events	Name	Descri...	IP Addresses	Module	Port Channel	VRF	# VLANs	Status	Monitored	Lo
✓	GigabitEthernet1/4/1			Chassis 1 4			0	up / up	☑	
✓	GigabitEthernet1/4/2			Chassis 1 4			0	up / up	☑	
✓	GigabitEthernet1/4/3			Chassis 1 4			0	up / up	☑	
✓	GigabitEthernet1/4/4			Chassis 1 4			0	up / up	☑	
✓	GigabitEthernet1/4/5			Chassis 1 4			0	up / up	☑	
✓	GigabitEthernet1/4/6			Chassis 1 4			0	up / up	☑	

From this view, all of the ACE and FWSM device specifics are displayed within this single view. Additional configuration details can be accessed from the individual Service Device by clicking on the device name in that column. For example, the ACE module in slot 1 would display the view shown in Figure 6-62. Figure 6-62 shows the ACE Module Virtual Contexts summary view, which includes status and allocation information for each context.

Figure 6-62. VSS DSN - ACE Module View



ACE1-Admin
/Network/Cisco/ACE
10.0.18.21

Up — DEVICE STATUS; Production — PRODUCTION STATE; Normal — PRIORITY

Virtual Contexts

Events	ID	Name	HA Status	Resource Class	# VLANs	# Policies	# Farms	# Servers	Monitored
✓	0	Admin	Active	default	2	0	0	0	☑
✓	1	Pall-CreateContainer-VLB-P	Active	2%_STICKY	3	0	0	2	☑
✓	2	Pall-CreateContainer-VLB-U	Active	2%_STICKY	3	0	0	2	☑
✓	3	Palladium-Container-VLB-P	Active	2%_STICKY	3	0	0	0	☑
✓	4	Palladium-Container-VLB-U	Active	2%_STICKY	3	1	2	2	☑
✓	5	Palladium-Tarheels-VLB-P	Active	2%_STICKY	3	2	2	6	☑
✓	6	Palladium-Tarheels-VLB-U	Active	2%_STICKY	3	2	2	1	☑

Display: Graphs   Range: Hourly   Link Graphs?: ☑

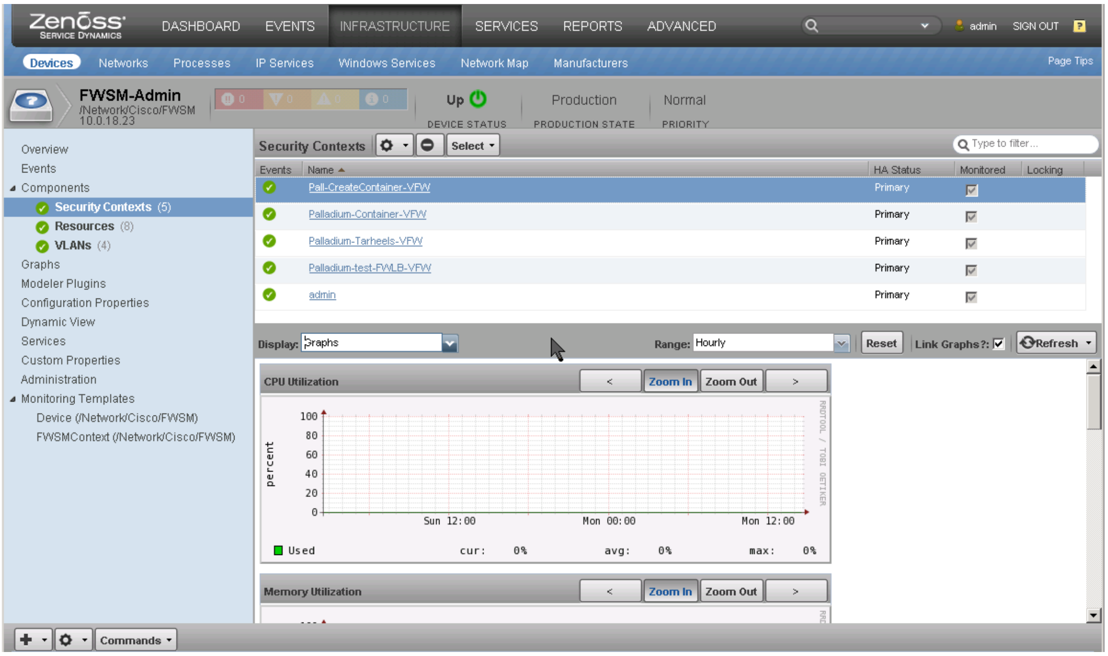
```
2012-05-13T08:07:52Z         to         2012-05-14T20:07:52Z
Used                   cur:     8%     avg:    9%        max:    11%
```

From this screen, an operator can select any of the individual VMDC tenant's contexts and use the graphs display to analyze a wide variety of performance metrics. In this example, CPU utilization for this context is displayed. In addition, the following graph selections are listed in the pull-down menu:

- Concurrent Connections
- Bandwidth
- Connection Rate
- Sticky Connections
- Translated Connections
- MAC Miss Rate

Figure 6-63 shows the FWSM Contexts summary view.

Figure 6-63. VSS DSN - FWSM View



FWSM-Admin
/Network/Cisco/FWSM
10.0.18.23

Up — DEVICE STATUS
Production — PRODUCTION STATE
Normal — PRIORITY

Security Contexts

Events	Name	HA Status	Monitored	Locking
✓	Pall-CreateContainer-VFW	Primary	☑	
✓	Palladium-Container-VFW	Primary	☑	
✓	Palladium-Tarheels-VFW	Primary	☑	
✓	Palladium-test-FWLB-VFW	Primary	☑	
✓	admin	Primary	☑	

Display: Graphs
Range: Hourly
Link Graphs?: ☑

CPU Utilization

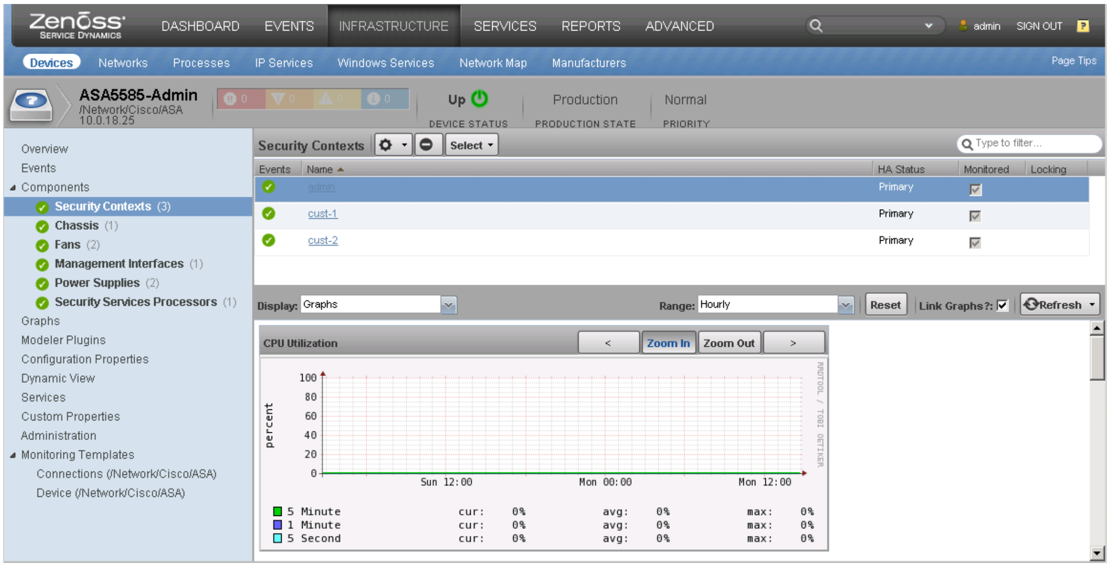
🟩 Used cur: 0% avg: 0% max: 0%

The following performance metrics are available in the graphs' pull-down menu:

- CPU Utilization
- Memory Utilization
- Concurrent Connections
- Concurrent Translations
- Translation Rate

Figure 6-64 shows the ASA summary view.

Figure 6-64. ASA - Device View



Events	Name	HA Status	Monitored	Locking
✓	admin	Primary	☑	
✓	cust-1	Primary	☑	
✓	cust-2	Primary	☑	

```
🟩 5 Minute    cur:  0%    avg:  0%    max:  0%
🟦 1 Minute    cur:  0%    avg:  0%    max:  0%
🟦 5 Second    cur:  0%    avg:  0%    max:  0%
```

Since this device is a standalone appliance, it is not linked under the VSS DSN; instead, it is available at the top-level Zenoss device navigation menu under **Network** > **Cisco** > **ASA category**. Selecting the Administrative context of the device yields the following device summary display:

- CPU Utilization
- Memory Utilization
- Concurrent Connections
- Concurrent Translations
- Translation Rate

Similar to the capabilities shown in Section 6.3.1 Network Resource Monitoring, Zenoss CSA provides cloud operators access to event details and KPI statistics for monitoring services components. Other device KPI information can be found in Appendix A Key Performance Indicators for VMDC.

6.3.3 Storage Resource Monitoring

The following VMDC storage components are monitored:

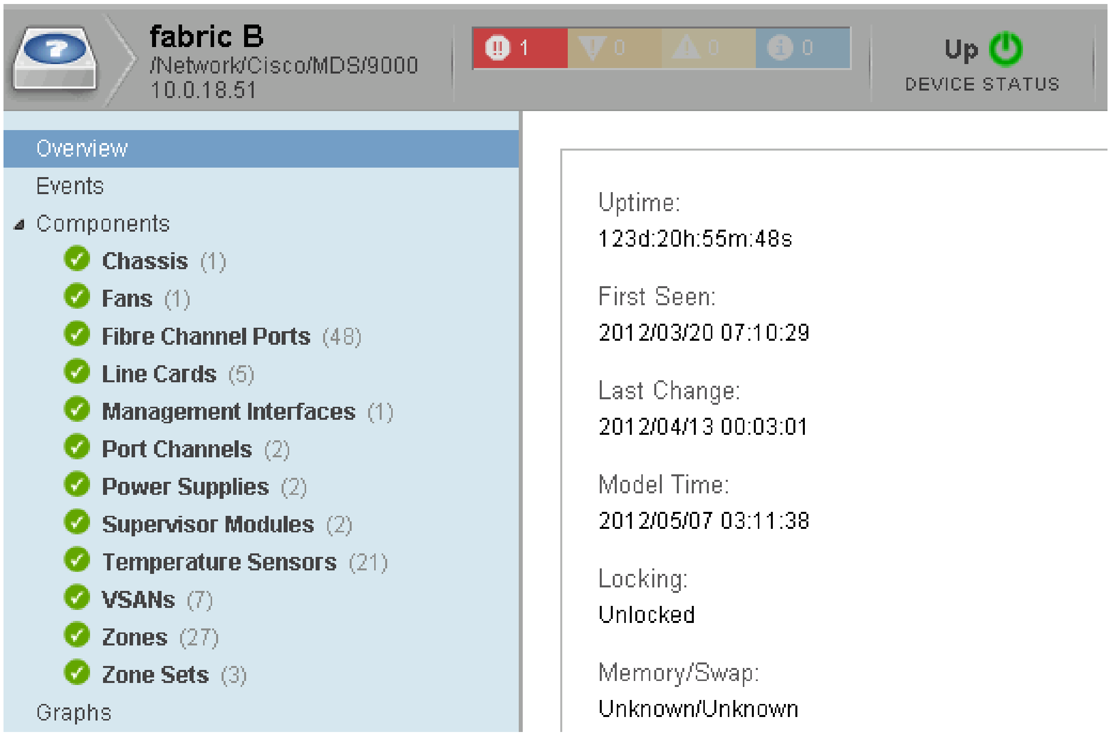
- Multilayer Director Switch (MDS)
- Nexus 5000
- NetApp
- EMC (new for Zenoss v4.2.3 - see new device support for Section 7.1 EMC for more information)

MDS

Zenoss monitors all major hardware and software components of the MDS. This includes the chassis and its sub-components, VSAN database, zone, and zoneset tables.

Note Note that Zenoss monitors only active zones and zonesets.

Figure 6-65. MDS Component View



Uptime:
123d:20h:55m:48s

First Seen:
2012/03/20 07:10:29

Last Change:
2012/04/13 00:03:01

Model Time:
2012/05/07 03:11:38

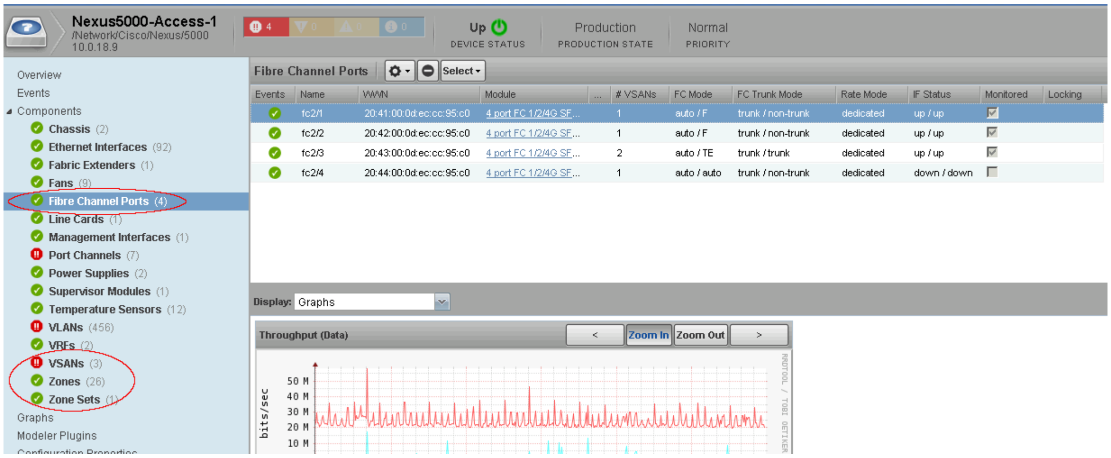
Locking:
Unlocked

Memory/Swap:
Unknown/Unknown

Nexus 5000

The Nexus 5000 Component view provides FiberChannel port information, VSAN, zone, and zoneset information.

Figure 6-66. Nexus 5000 Component View



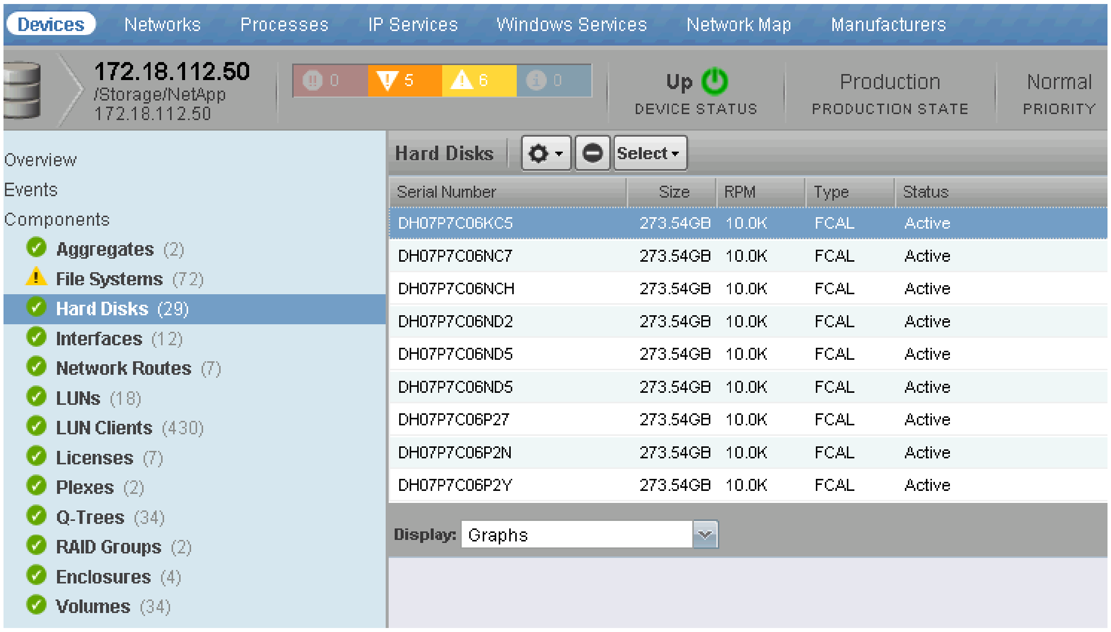
Events	Name	WWN	Module	...	# VSANs	FC Mode	FC Trunk Mode	Rate Mode	IF Status	Monitored	Locking
✓	fc2/1	20:41:00:0d:ec:cc:95:c0	4 port FC 1/2/4G SF...		1	auto / F	trunk / non-trunk	dedicated	up / up	☑	
✓	fc2/2	20:42:00:0d:ec:cc:95:c0	4 port FC 1/2/4G SF...		1	auto / F	trunk / non-trunk	dedicated	up / up	☑	
✓	fc2/3	20:43:00:0d:ec:cc:95:c0	4 port FC 1/2/4G SF...		2	auto / TE	trunk / trunk	dedicated	up / up	☑	
✓	fc2/4	20:44:00:0d:ec:cc:95:c0	4 port FC 1/2/4G SF...		1	auto / auto	trunk / non-trunk	dedicated	down / down	☐	

NetApp

Zenoss includes a ZenPack for monitoring NetApp Arrays. Figure 6-67 provides a view of the components monitored.

Note Note that ESXi data store to NetApp Logical Unit Number (LUN) associations are currently only possible if using Internet Small Computer System Interface (iSCSI) or Network File System (NFS). The current ZenPack (version 2.2.12) does not have the ability to monitor the FiberChannel ports on a NetApp array.

Figure 6-67. NetApp Component View



Serial Number	Size	RPM	Type	Status
DH07P7C06KC5	273.54GB	10.0K	FCAL	Active
DH07P7C06NC7	273.54GB	10.0K	FCAL	Active
DH07P7C06NCH	273.54GB	10.0K	FCAL	Active
DH07P7C06ND2	273.54GB	10.0K	FCAL	Active
DH07P7C06ND5	273.54GB	10.0K	FCAL	Active
DH07P7C06ND5	273.54GB	10.0K	FCAL	Active
DH07P7C06P27	273.54GB	10.0K	FCAL	Active
DH07P7C06P2N	273.54GB	10.0K	FCAL	Active
DH07P7C06P2Y	273.54GB	10.0K	FCAL	Active

Similar to the capabilities shown in Section 6.3.1 Network Resource Monitoring, Zenoss CSA provides cloud operators access to event details and KPI statistics for monitoring storage components.

6.3.4 Compute Resource Monitoring

The following VMDC compute components are monitored:

- UCS Infrastructure
- VMware vSphere

UCS

Zenoss places UCS infrastructure under the /CiscoUCS device class. Zenoss monitors key components of the UCS domain using the UCSM domain manager. Components monitored include all chassis subsystems like fans, power supplies, and server blades. Table 6-9 provides additional information about some of the components.

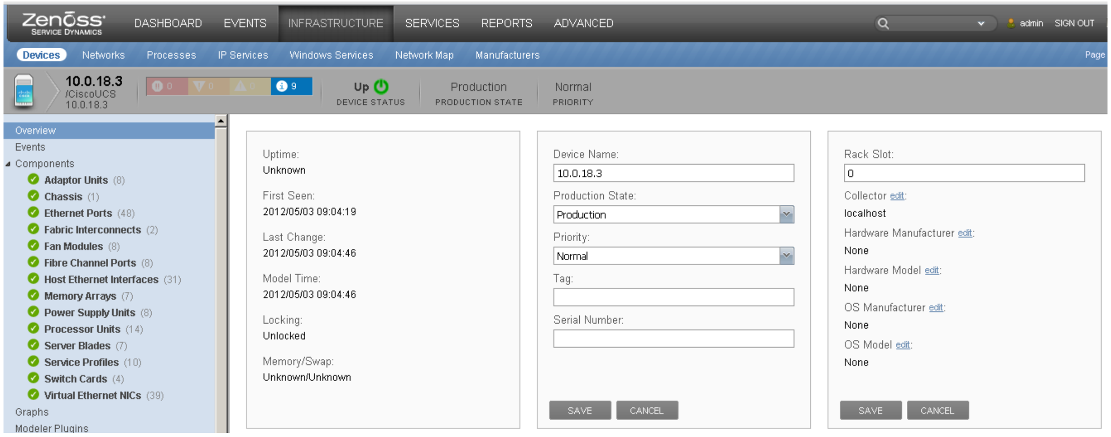
Table 6-9. UCS Components

Component	Notes
Adapter Units	List of all CNAs in the UCS domain

Component	Notes
Ethernet Ports	List of all physical Ethernet ports associated with this domain. These Ethernet ports reside only on the fabric interconnects.
Host Ethernet Interfaces	List of all server ports. Each listing describes the relationship between a server Ethernet port, the CNA (adapter) it is associated with, and the blade location.

Figure 6-68 illustrates the UCS component listing managed by a single UCSM.

Figure 6-68. UCS Component View



Uptime:
Unknown

First Seen:
2012/05/03 09:04:19

Last Change:
2012/05/03 09:04:46

Model Time:
2012/05/03 09:04:46

Locking:
Unlocked

Memory/Swap:
Unknown/Unknown

Device Name: 10.0.18.3

Production State: Production

Priority: Normal

Tag:

Serial Number:

Rack Slot: 0

Collector edit:
localhost

Hardware Manufacturer edit:
None

Hardware Model edit:
None

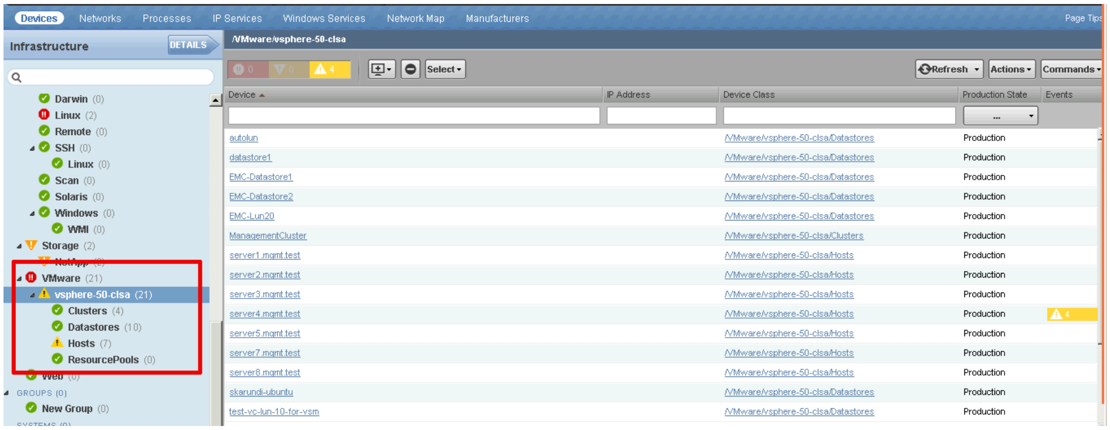
OS Manufacturer edit:
None

OS Model edit:
None

VMware

Zenoss places VMware vSphere infrastructure under the /VMware device class. Zenoss can monitor the data store, resource pools, ESXi hosts, and VMs associated with the ESXi hosts. Figure 6-69 shows the high-level component view for a vSphere management domain.

Figure 6-69. VMware Component View



/VMware/vsphere-50-clsa

Device	IP Address	Device Class	Production State	Events
autolun		/VMware/vsphere-50-clsa/Datastores	Production	
datastore1		/VMware/vsphere-50-clsa/Datastores	Production	
EMC-Datastore1		/VMware/vsphere-50-clsa/Datastores	Production	
EMC-Datastore2		/VMware/vsphere-50-clsa/Datastores	Production	
EMC-Lun20		/VMware/vsphere-50-clsa/Datastores	Production	
ManagementCluster		/VMware/vsphere-50-clsa/Clusters	Production	
server1.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	
server2.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	
server3.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	
server4.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	⚠ 4
server5.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	
server7.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	
server8.mgmt.test		/VMware/vsphere-50-clsa/Hosts	Production	
skarundi-ubuntu		/VMware/vsphere-50-clsa/Datastores	Production	
test-vc-lun-10-for-vsm		/VMware/vsphere-50-clsa/Datastores	Production	

Similar to the capabilities shown in Section 6.3.1 Network Resource Monitoring, Zenoss CSA provides cloud operators access to event details and KPI statistics for monitoring compute components.

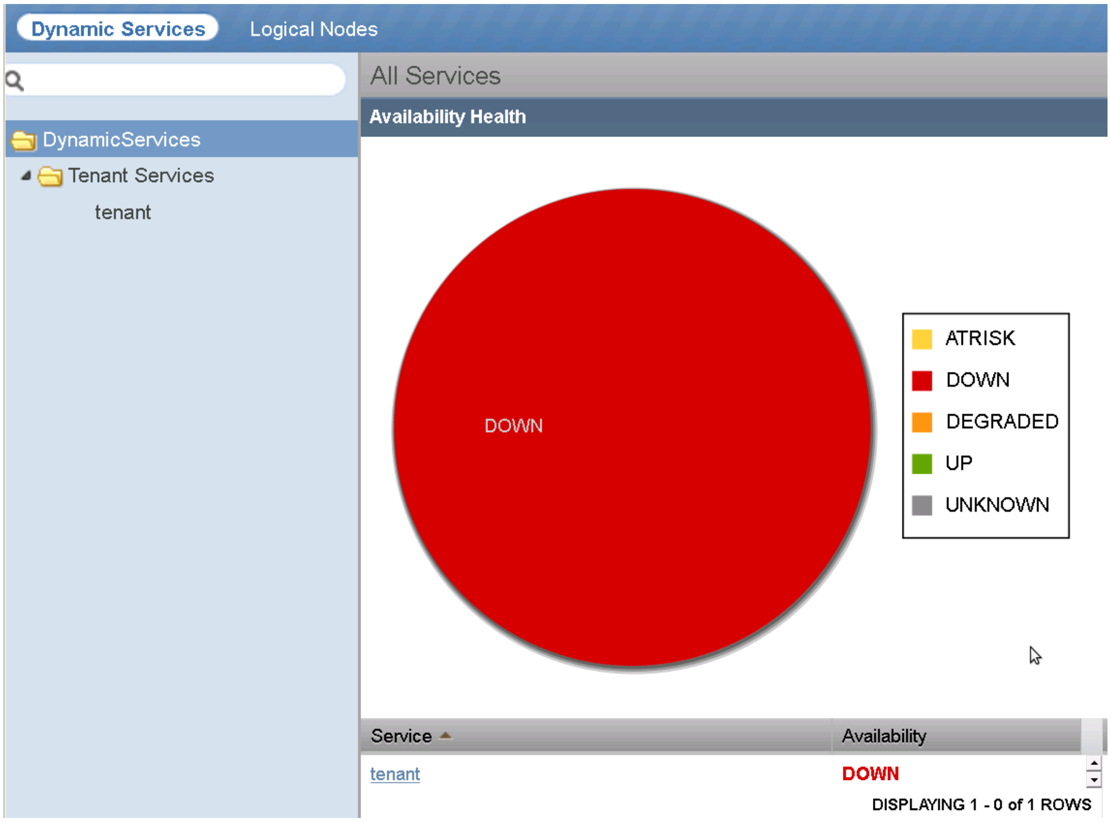
6.4 Tenant Service Impact Analysis and Root Cause Analysis

The following use case discusses how to determine the root cause of a service impacting fault. The example shows two VMs in the tenant, Windows-1 and Centos-1 that experience a failure due to complete access switch to UCS communication failure. This is a severe scenario where all ESXi hosts in the environment are unreachable.

For an overview of how an impact graph works, see Section 3.5.4 Root Cause Analysis and Service Impact Analysis.

When the failure occurs, the Service Impact Dashboard receives an error and the tenant service goes red.

Figure 6-70. Dashboard View of a Tenant Service Failure



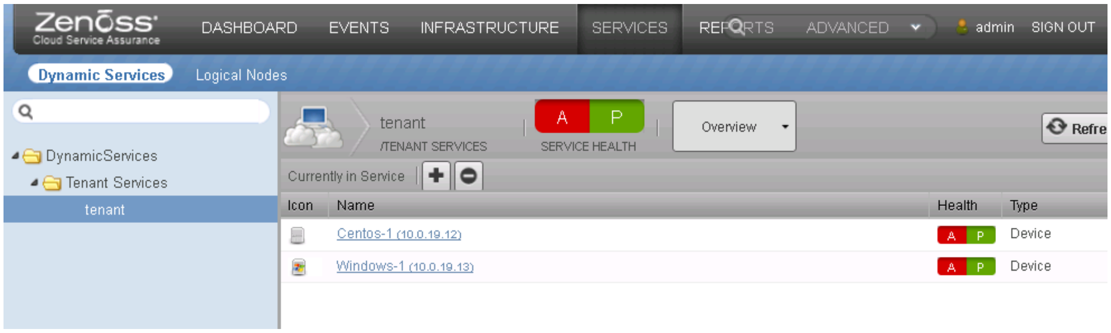
Service	Availability
tenant	DOWN

DISPLAYING 1 - 0 of 1 ROWS

Determining Root Cause - Operator Workflow

Click the tenant service to view the impact overview. The overview provides a summary of VMs that have failed. In this example, all VMs are down. Click the **Overview** button and switch to the Impact Event view to understand the errors associated with the failure.

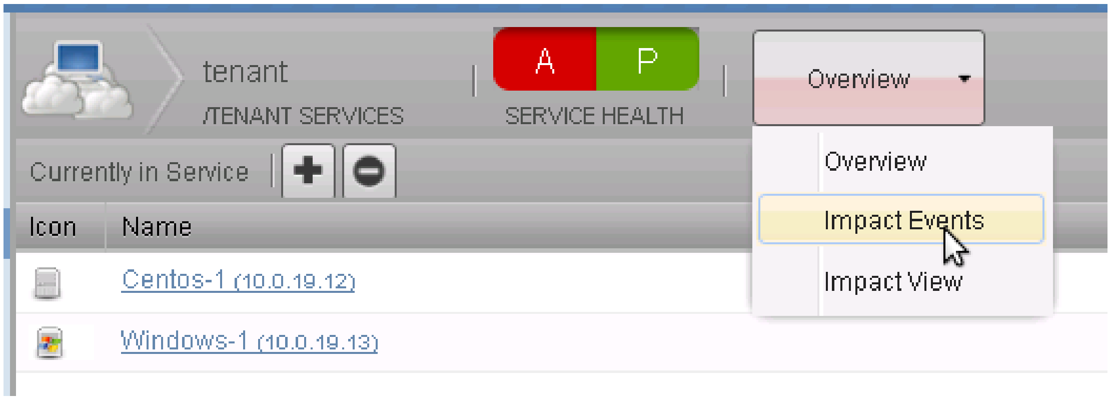
Figure 6-71. Impact Overview of Tenant Service Failure



Icon	Name	Health	Type
	Centos-1 (10.0.19.12)	A P	Device
	Windows-1 (10.0.19.13)	A P	Device

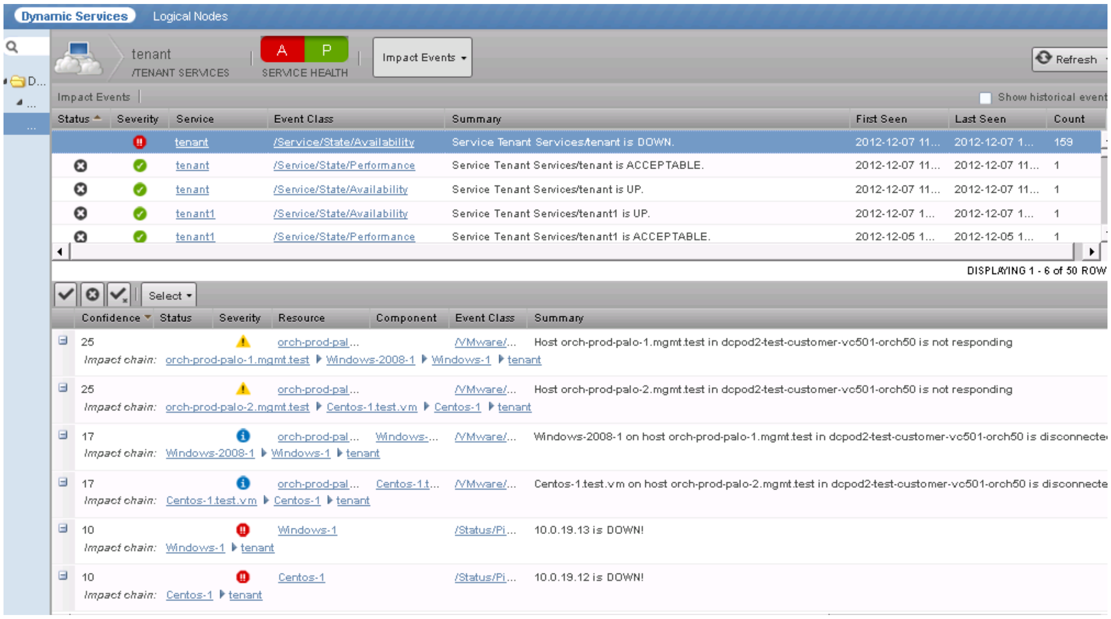
Switch to the Impact Event View to review the error messages associated with the failure.

Figure 6-72. Switch To Impact Event View from Impact Event Overview



On the Impact Event view, click the message that is not marked with an **X**. This indicates that the failure is active. These events should be listed at the top of the event list. In the example, there are five error messages listed in the event. Each is marked with a level of confidence. The higher the confidence value, the greater the likelihood that this is the root cause. In the example, the error message with the highest confidence is stating that two ESXi hosts are not responding.

Figure 6-73. Impact Event Errors



Impact Events

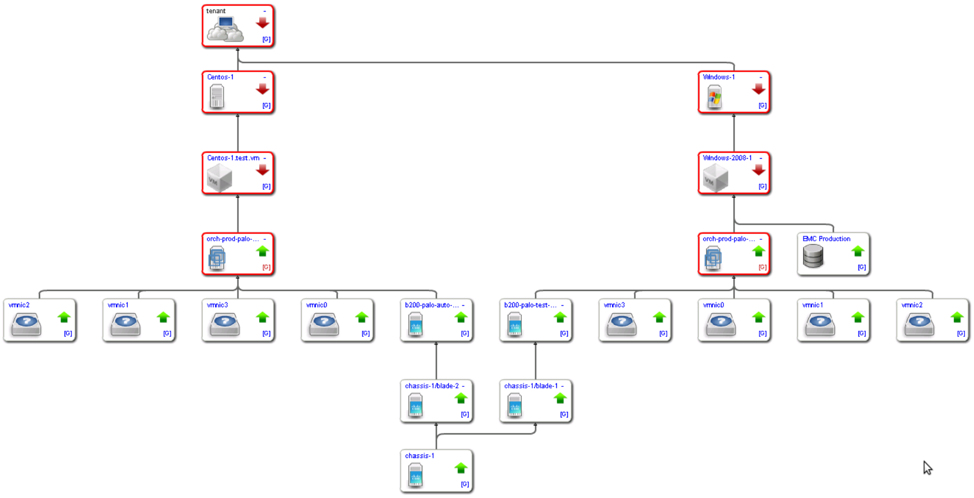
Status	Severity	Service	Event Class	Summary	First Seen	Last Seen	Count
	‼	tenant	/Service/State/Availability	Service Tenant Services/tenant is DOWN.	2012-12-07 11...	2012-12-07 1...	159
⊗	✓	tenant	/Service/State/Performance	Service Tenant Services/tenant is ACCEPTABLE.	2012-12-07 11...	2012-12-07 11...	1
⊗	✓	tenant	/Service/State/Availability	Service Tenant Services/tenant is UP.	2012-12-07 11...	2012-12-07 11...	1
⊗	✓	tenant1	/Service/State/Availability	Service Tenant Services/tenant1 is UP.	2012-12-07 1...	2012-12-07 1...	1
⊗	✓	tenant1	/Service/State/Performance	Service Tenant Services/tenant1 is ACCEPTABLE.	2012-12-05 1...	2012-12-05 1...	1

DISPLAYING 1 - 6 of 50 ROW

Confidence	Status	Severity	Resource	Component	Event Class	Summary
25		⚠	orch-prod-pal...		/VMware/...	Host orch-prod-palo-1.mgmt.test in dcpod2-test-customer-vc501-orch50 is not responding
Impact chain: orch-prod-palo-1.mgmt.test ▶ Windows-2008-1 ▶ Windows-1 ▶ tenant						
25		⚠	orch-prod-pal...		/VMware/...	Host orch-prod-palo-2.mgmt.test in dcpod2-test-customer-vc501-orch50 is not responding
Impact chain: orch-prod-palo-2.mgmt.test ▶ Centos-1.test.vm ▶ Centos-1 ▶ tenant						
17		ℹ	orch-prod-pal...	Windows-...	/VMware/...	Windows-2008-1 on host orch-prod-palo-1.mgmt.test in dcpod2-test-customer-vc501-orch50 is disconnecte
Impact chain: Windows-2008-1 ▶ Windows-1 ▶ tenant						
17		ℹ	orch-prod-pal...	Centos-1.t...	/VMware/...	Centos-1.test.vm on host orch-prod-palo-2.mgmt.test in dcpod2-test-customer-vc501-orch50 is disconnecte
Impact chain: Centos-1.test.vm ▶ Centos-1 ▶ tenant						
10		‼	Windows-1		/Status/Pi...	10.0.19.13 is DOWN!
Impact chain: Windows-1 ▶ tenant						
10		‼	Centos-1		/Status/Pi...	10.0.19.12 is DOWN!
Impact chain: Centos-1 ▶ tenant						

By switching views from the Impact Events to the Impact Graph, the operator can see that both ESXi hosts are down, and there is no impact to the UCS infrastructure. This means that there is no known issue with the UCS. All ESXi hosts are not responding. Connectivity to the ESXi host is the most likely problem and contacting network engineers would be the next step.

Figure 6-74. Expanded View of Tenant Service Failure



6.5 Identifying Availability Impact of Shared Resource Faults on Tenant Services

When faults occur on shared tenant resources, like the UCS chassis or an ESXi host, multiple tenant services are affected. Most faults on shared resources do not have an adverse affect on the performance or availability of a tenant. This is due to the redundant design of VMDC and the HA features built into the architecture.

Most faults, such as a failure of a UCS Input-Output Module (IOM) or high CPU utilization on ESXi host, trigger either an automatic failover, as in the case of a UCS IOM failure, or a vMotion of the VM from the affected host to another host in the cluster.

There are a few events, such as a UCS blade crash that could result in a tenant service outage due to a VM requiring a restart. A blade crash results in a short failure of the tenant service VM as the VM using VMware HA restarts on a different host. If an entire UCS chassis goes down, then this failure should have the same impact as a blade failure. The ESXi clusters should be designed in a way that at least two hosts in the ESXi clusters are on two blades on two different chassis.

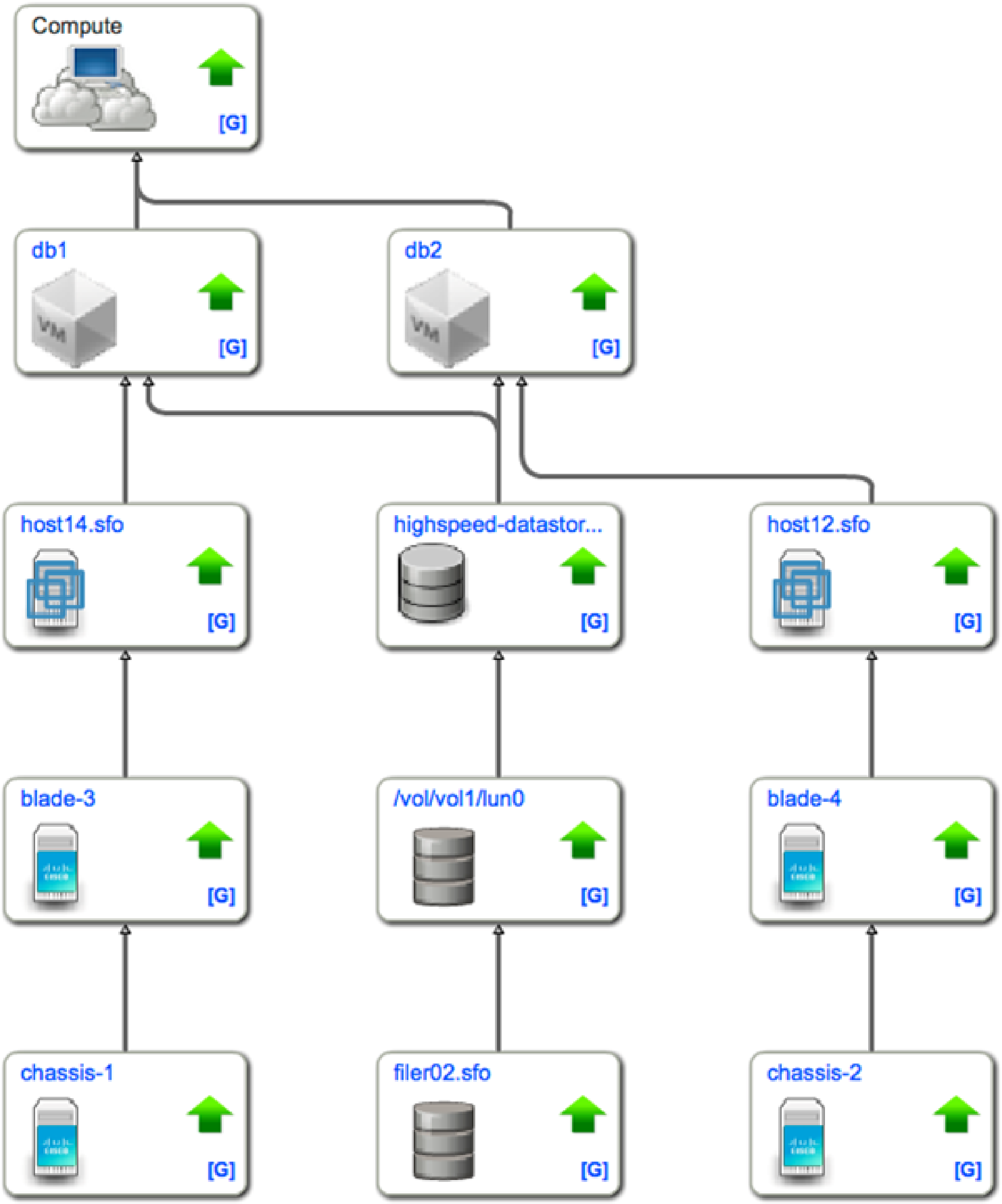
Chassis Failure Use Case

The following use case illustrates a severe fault, a UCS chassis failure, and its impact on tenant services. Figure 6-75 shows a generic compute tenant service defined with the resources listed here.

- One Compute service node
- Two database VMs
- One shared data store
- Two ESXi hosts
- Two UCS blades
- Two UCS chassis

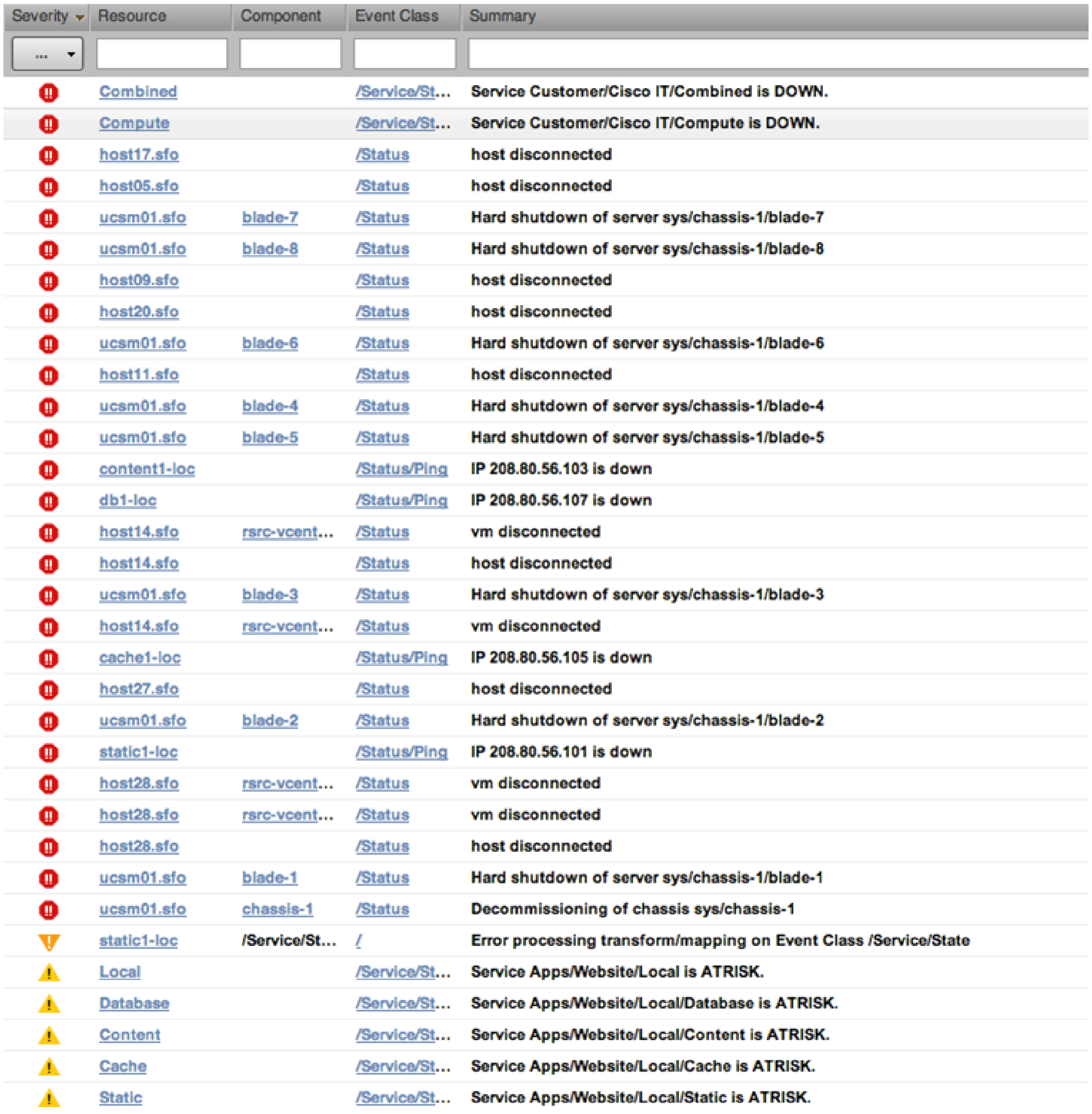
The graph is simplified for illustrative purposes. UCS service profiles and VMware cluster nodes have been omitted. Multi-blade and multi-chassis vSphere clusters are recommended in the compute stack.

Figure 6-75. Compute Service Impact Graph Before Chassis Failure



Assume UCS chassis-1 at the lower left experiences a complete outage. At this point, an event storm hits the monitoring tool event console. The challenge before an operator normally would be to determine the root cause by reading every event, hopefully not too fast to skip over the right one, but fast enough to solve the problem before a customer calls. Figure 6-76 illustrates an event console with a long list of attention getting red events.

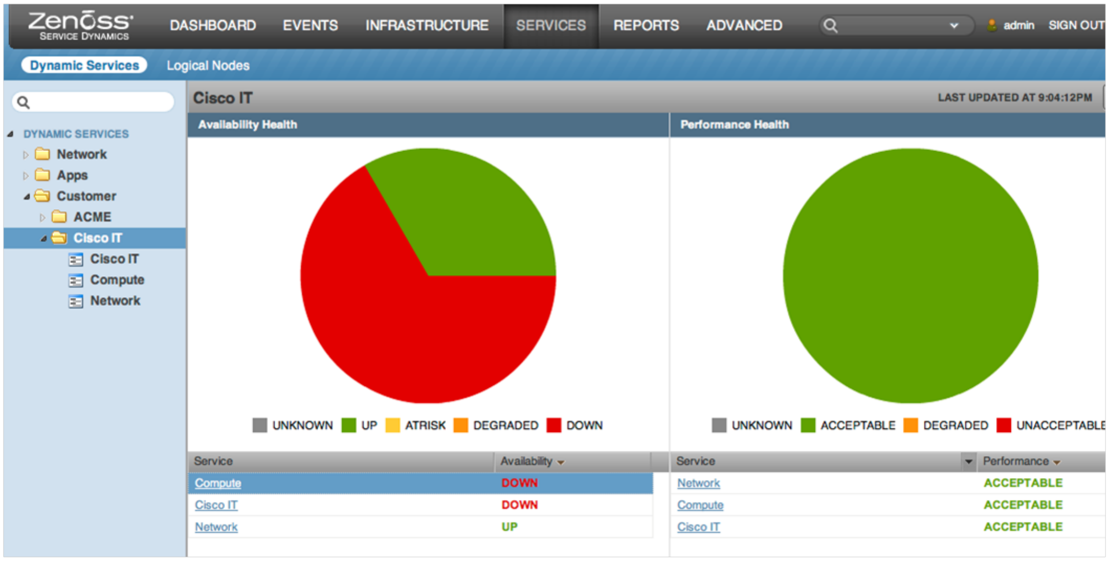
Figure 6-76. Event Console After Chassis Failure



Severity	Resource	Component	Event Class	Summary
●	Combined		/Service/St…	Service Customer/Cisco IT/Combined is DOWN.
●	Compute		/Service/St…	Service Customer/Cisco IT/Compute is DOWN.
●	host17.sfo		/Status	host disconnected
●	host05.sfo		/Status	host disconnected
●	ucsm01.sfo	blade-7	/Status	Hard shutdown of server sys/chassis-1/blade-7
●	ucsm01.sfo	blade-8	/Status	Hard shutdown of server sys/chassis-1/blade-8
●	host09.sfo		/Status	host disconnected
●	host20.sfo		/Status	host disconnected
●	ucsm01.sfo	blade-6	/Status	Hard shutdown of server sys/chassis-1/blade-6
●	host11.sfo		/Status	host disconnected
●	ucsm01.sfo	blade-4	/Status	Hard shutdown of server sys/chassis-1/blade-4
●	ucsm01.sfo	blade-5	/Status	Hard shutdown of server sys/chassis-1/blade-5
●	content1-loc		/Status/Ping	IP 208.80.56.103 is down
●	db1-loc		/Status/Ping	IP 208.80.56.107 is down
●	host14.sfo	rsrc-vcent…	/Status	vm disconnected
●	host14.sfo		/Status	host disconnected
●	ucsm01.sfo	blade-3	/Status	Hard shutdown of server sys/chassis-1/blade-3
●	host14.sfo	rsrc-vcent…	/Status	vm disconnected
●	cache1-loc		/Status/Ping	IP 208.80.56.105 is down
●	host27.sfo		/Status	host disconnected
●	ucsm01.sfo	blade-2	/Status	Hard shutdown of server sys/chassis-1/blade-2
●	static1-loc		/Status/Ping	IP 208.80.56.101 is down
●	host28.sfo	rsrc-vcent…	/Status	vm disconnected
●	host28.sfo	rsrc-vcent…	/Status	vm disconnected
●	host28.sfo		/Status	host disconnected
●	ucsm01.sfo	blade-1	/Status	Hard shutdown of server sys/chassis-1/blade-1
●	ucsm01.sfo	chassis-1	/Status	Decommissioning of chassis sys/chassis-1
▼	static1-loc	/Service/St…	/	Error processing transform/mapping on Event Class /Service/State
⚠	Local		/Service/St…	Service Apps/Website/Local is ATRISK.
⚠	Database		/Service/St…	Service Apps/Website/Local/Database is ATRISK.
⚠	Content		/Service/St…	Service Apps/Website/Local/Content is ATRISK.
⚠	Cache		/Service/St…	Service Apps/Website/Local/Cache is ATRISK.
⚠	Static		/Service/St…	Service Apps/Website/Local/Static is ATRISK.

An operator has the option to monitor the Services dashboard instead of the Events dashboard to more quickly determine if any services have been impacted. Figure 6-77 shows that some services have been impacted by a still to be determined fault.

Figure 6-77. Service Dashboard Indicating Tenant Service Impact

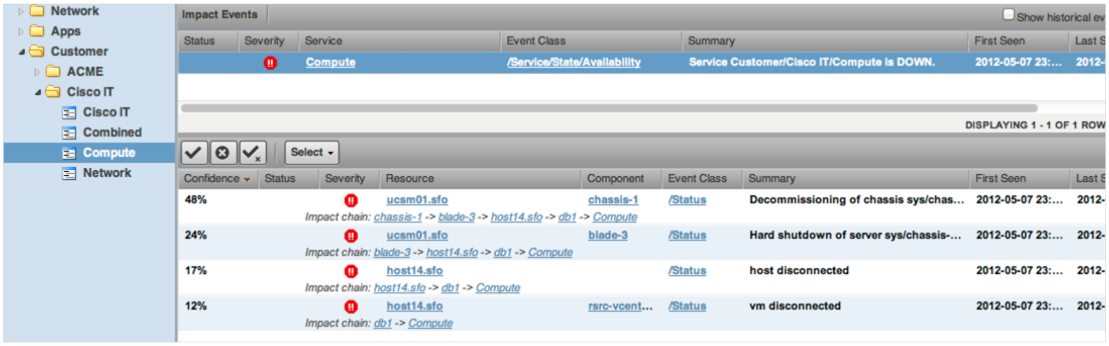


Service	Availability
Compute	DOWN
Cisco IT	DOWN
Network	UP

Service	Performance
Network	ACCEPTABLE
Compute	ACCEPTABLE
Cisco IT	ACCEPTABLE

The operator can then drill down from the Services dashboard into a specific service to explore the list of impact events and the prioritized list of root causes. Figure 6-78 provides a view of these impact and root cause events identifying chassi-1 failure as the primary root cause.

Figure 6-78. Service Impact and Root Cause Event Screen

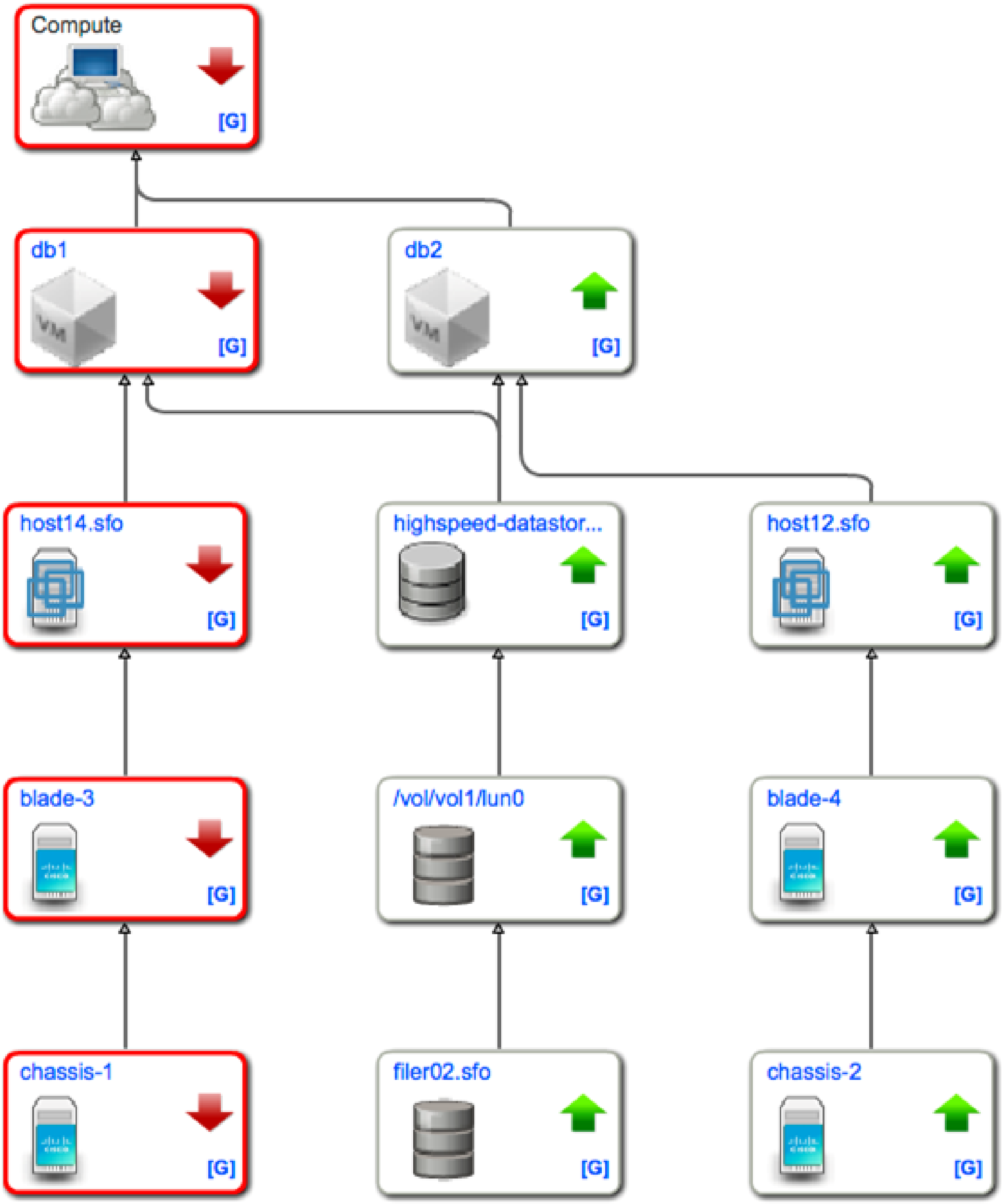


Status	Severity	Service	Event Class	Summary	First Seen	Last S
	‼	Compute	/Service/State/Availability	Service Customer/Cisco IT/Compute is DOWN.	2012-05-07 23:...	2012-

Confidence	Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last S
48%		‼	ucsm01.sfo	chassis-1	/Status	Decommissioning of chassis sys/chas...	2012-05-07 23:...	2012-
		Impact chain: chassis-1 -> blade-3 -> host14.sfo -> db1 -> Compute						
24%		‼	ucsm01.sfo	blade-3	/Status	Hard shutdown of server sys/chassis-...	2012-05-07 23:...	2012-
		Impact chain: blade-3 -> host14.sfo -> db1 -> Compute						
17%		‼	host14.sfo		/Status	host disconnected	2012-05-07 23:...	2012-
		Impact chain: host14.sfo -> db1 -> Compute						
12%		‼	host14.sfo	rsrc-vcent...	/Status	vm disconnected	2012-05-07 23:...	2012-
		Impact chain: db1 -> Compute						

If the operator switched views back to the Service Impact graph, then the Compute service would clearly be shown as down. Figure 6-79 details the Service Impact graph following a chassis failure.

Figure 6-79. Service Impact Graph After Chassis Failure



Operator familiarity with the Services tools including the Services dashboard, Service Impact graph, and Service Impact and Root Cause event screens should significantly reduce MTTR for cloud assurance operators. After these few troubleshooting steps, the root cause is easily identified, and the outage can be corrected.

6.6 Identifying Performance Problems Before They Impact Tenants

CLSA-VMDC reduces the administrative overhead of performance monitoring configuration by enabling performance monitoring centralization. Zenoss CSA performance monitoring is configured on the Zenoss server, and any pre-configured performance alerts from other domain managers such as VMware are treated as informational so that Zenoss maintains control of performance notification. For example, if vSphere sends an alert that the CPU alarm of a server has gone from green to red, this alert does not necessarily trigger a change in the tenant service. The performance thresholds on Zenoss CSA would also have to be crossed to impact a tenant service. The commonly monitored performance items on an ESXi host and its associated VMs are as follows:

- CPU
- Memory
- Disk usage

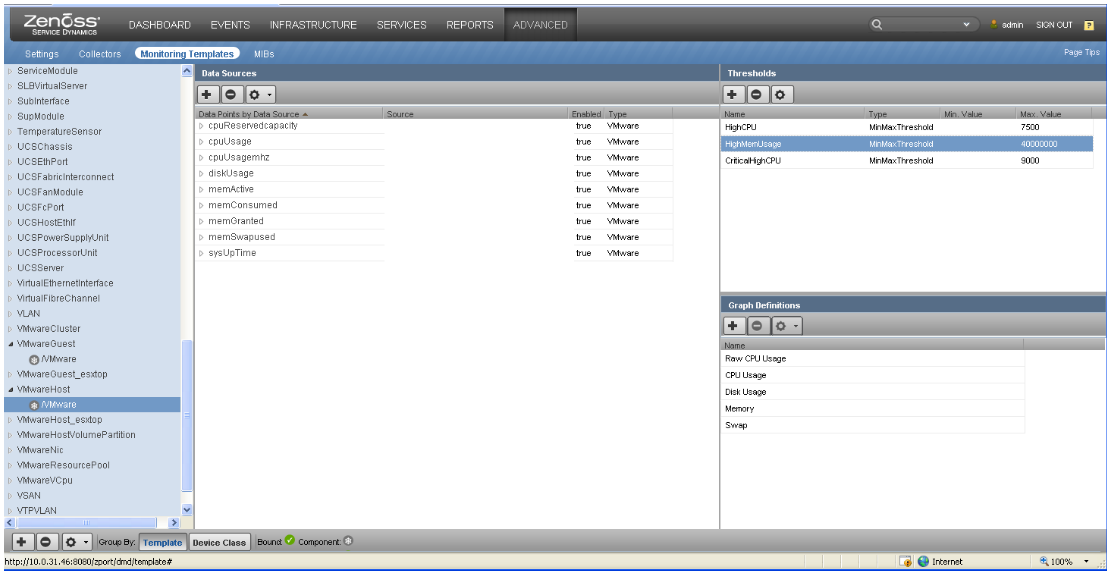
Note The general procedures used to configure Zenoss performance monitoring are found in the Zenoss Cloud Service Assurance Installation and Administration Guide.

The following use case discusses how to configure and identify performance problems before they impact tenants.

Performance Threshold Configuration

Thresholds can be added to any device to alert the operator that there could be an issue within a tenant or a group of compute devices. Some examples would be setting memory or CPU thresholds on the VMware ESXi servers or the VMs. Thresholds are set by editing the monitoring templates. From the Advanced tab, select **Monitoring Templates**. In this case, the VMware Host template is modified by adding a couple of CPU thresholds and a Memory usage threshold. There are no thresholds configured by default.

Figure 6-80. Monitoring Templates



Data Sources

Data Points by Data Source	Source	Enabled	Type
cpuReservedcapacity		true	VMware
cpuUsage		true	VMware
cpuUsagemhz		true	VMware
diskUsage		true	VMware
memActive		true	VMware
memConsumed		true	VMware
memGranted		true	VMware
memSwapused		true	VMware
sysUpTime		true	VMware

Thresholds

Name	Type	Min. Value	Max. Value
HighCPU	MinMaxThreshold		7500
HighMemUsage	MinMaxThreshold		40000000
CriticalHighCPU	MinMaxThreshold		9000

Graph Definitions

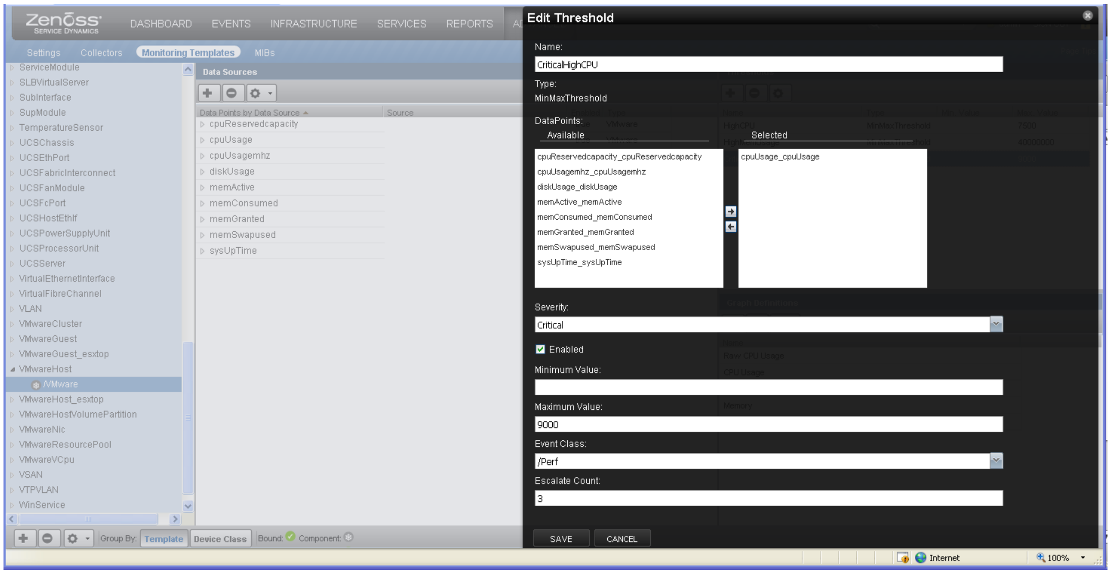
Name
Raw CPU Usage
CPU Usage
Disk Usage
Memory
Swap

The CriticalHighCPU has been configured to:

- Use the cpuUsage datapoint
- Be logged as a critical event
- Be Enabled
- Have a max value of 9000 (90%)
- Have an event class of /Perf
- Have an Escalation count of 3

If the cpuUsage goes over 90% and is seen three times, then a critical event under the performance class is generated.

Figure 6-81. Edit Threshold



Edit Threshold

Name: CriticalHighCPU

Type: MinMaxThreshold

DataPoints:

Available:
cpuReservedcapacity_cpuReservedcapacity
cpuUsagemhz_cpuUsagemhz
diskUsage_diskUsage
memActive_memActive
memConsumed_memConsumed
memGranted_memGranted
memSwapused_memSwapused
sysUpTime_sysUpTime

Selected:
cpuUsage_cpuUsage

Severity: Critical

☑ Enabled

Minimum Value:

Maximum Value: 9000

Event Class: /Perf

Escalate Count: 3

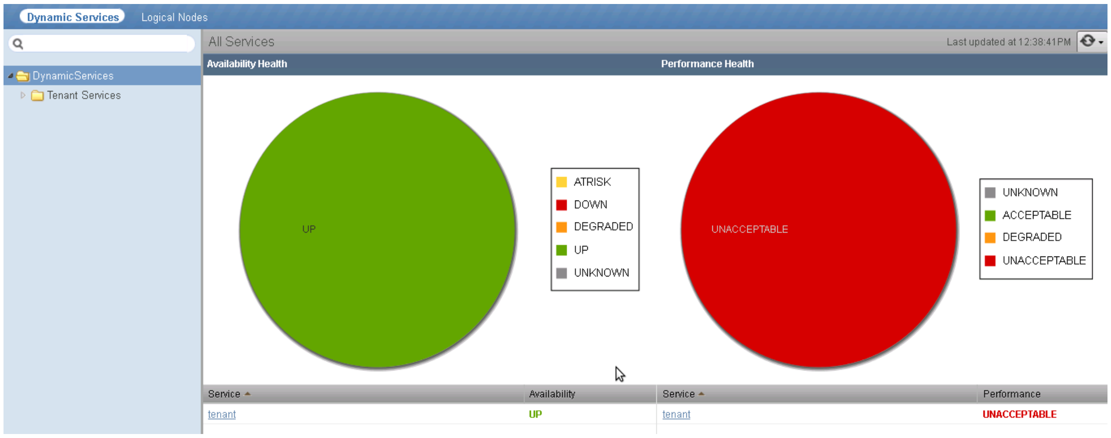
Performance Monitoring of an ESXi Host

It is recommended when issuing performance monitoring on a ESXi Host, to configure performance monitoring on a ESXi Guest VM as well. This means enabling Performance Monitoring on the VmwareHost and VmwareGuest Monitoring Templates as shown in Figure 6-81. It is useful to have VM Monitoring in addition to Host monitoring because VM monitoring helps isolates the VMs that may be root cause of the performance problem.

Alerts from Zenoss CPU monitoring of the VMware ESXi Host and VM can influence the state of the Service Impact graph. In Figure 6-82, a CPU threshold trigger has been crossed. The Services dashboard shows a Performance Health impact for the tenant service.

Note The severity level of an alert from VMware is not reported in Zenoss. Zenoss marks all VMware events as informational, even though these events may be severe or critical VMware events. The thresholds configured within Zenoss determine the potential impact of a given VMware event.

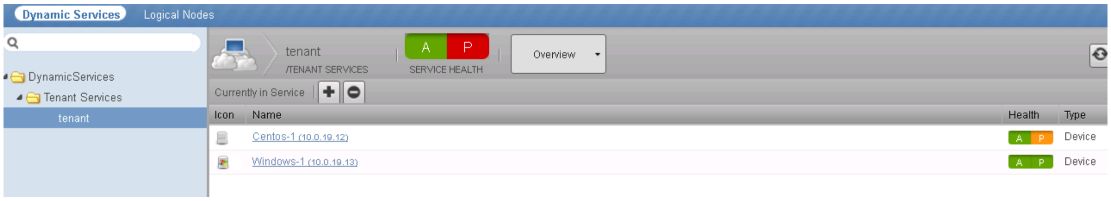
Figure 6-82. Performance Health Impact Graph



Service	Availability	Service	Performance
tenant	UP	tenant	UNACCEPTABLE

From the overview screen, the operator can detect which VMs may be the cause of the performance problem. In this case, a Linux server "centos-1" appears to have a performance problem. With the VmwareGuest Performance monitoring enabled, a VM encountering performance problems will detected. With just the VMwareHost Performance monitoring enabled, a general error will be detected, but exactly which VM is affected cannot be easily determined through the Performance Overview screen.

Figure 6-83. Performance Impact Analysis Dashboard

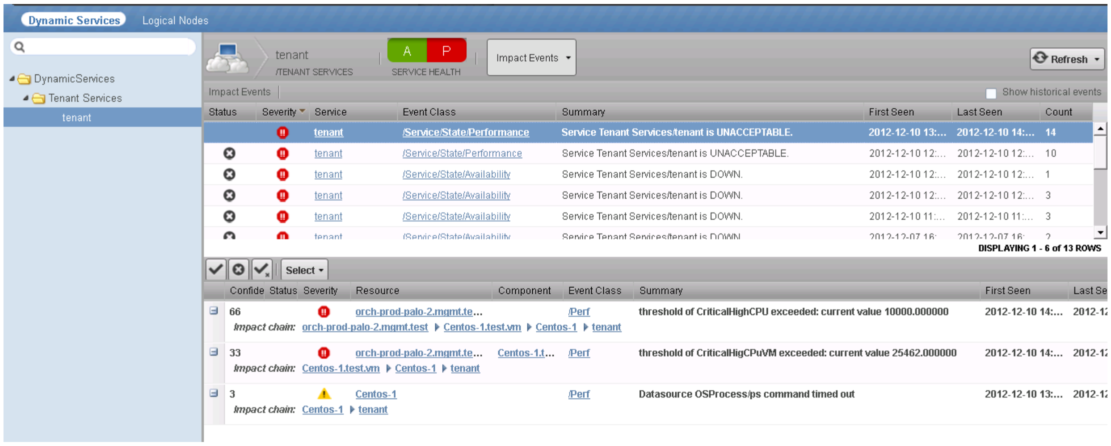


Icon	Name	Health	Type
	Centos-1 (10.0.19.12)	A P	Device
	Windows-1 (10.0.19.13)	A P	Device

The Performance Impact Event lists alerts with different confidence levels. The higher the confidence level the most likely the event is the root cause of the problem. In this example, there are three events generated - one from the VMwareHost Monitoring template, the second from the VMwareVM Monitoring template, and the third an error message from the OS itself. This is most likely a symptom of the high CPU utilization.

The events with the higher confidence level of 66% and 33% relate to VM/Host CPU utilization, which is correct.

Figure 6-84. Performance Impact Event



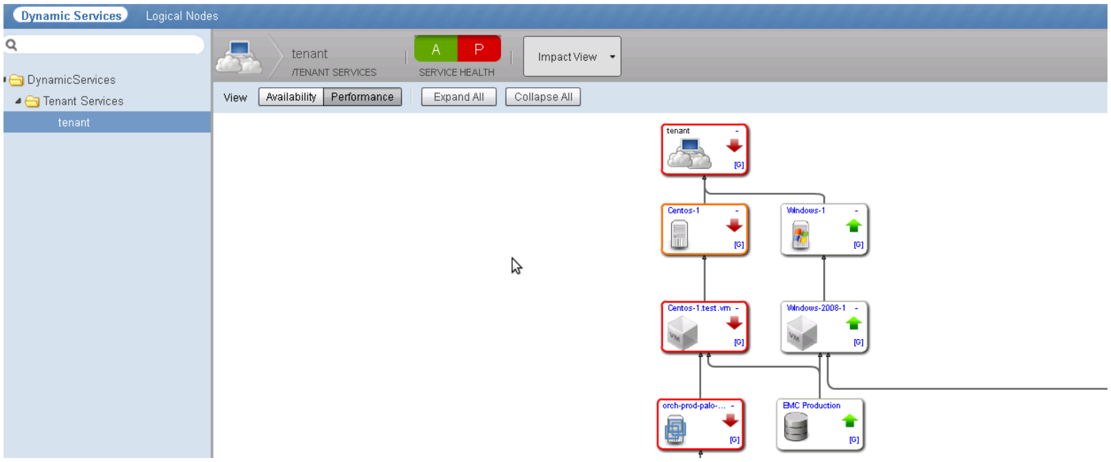
Status	Severity	Service	Event Class	Summary	First Seen	Last Seen	Count
	●	tenant	/Service/State/Performance	Service Tenant Services/tenant is UNACCEPTABLE.	2012-12-10 13:...	2012-12-10 14:...	14
☒	●	tenant	/Service/State/Performance	Service Tenant Services/tenant is UNACCEPTABLE.	2012-12-10 12:...	2012-12-10 12:...	10
☒	●	tenant	/Service/State/Availability	Service Tenant Services/tenant is DOWN.	2012-12-10 12:...	2012-12-10 12:...	1
☒	●	tenant	/Service/State/Availability	Service Tenant Services/tenant is DOWN.	2012-12-10 12:...	2012-12-10 12:...	3
☒	●	tenant	/Service/State/Availability	Service Tenant Services/tenant is DOWN.	2012-12-10 11:...	2012-12-10 11:...	3
☒	●	tenant	/Service/State/Availability	Service Tenant Services/tenant is DOWN.	2012-12-07 16:	2012-12-07 16:	2

DISPLAYING 1 - 6 of 13 ROWS

Confide	Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Se
66		●	orch-prod-palo-2.mgmt.te...		/Perf	threshold of CriticalHighCPU exceeded: current value 10000.000000	2012-12-10 14:...	2012-1
Impact chain: orch-prod-palo-2.mgmt.test ▸ Centos-1.test.vm ▸ Centos-1 ▸ tenant								
33		●	orch-prod-palo-2.mgmt.te...	Centos-1.t...	/Perf	threshold of CriticalHigCPuVM exceeded: current value 25462.000000	2012-12-10 14:...	2012-1
Impact chain: Centos-1.test.vm ▸ Centos-1 ▸ tenant								
3		⚠	Centos-1		/Perf	Datasource OSProcess/ps command timed out	2012-12-10 13:...	2012-1
Impact chain: Centos-1 ▸ tenant								

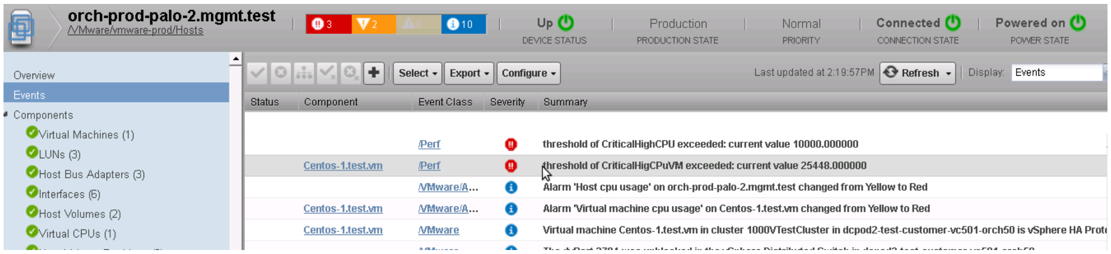
A graphical representation of the performance health can be seen in Figure 6-84 illustrating the Service Impact graph of an affected server.

Figure 6-85. Service Impact Graph Showing Performance Health



The resource manager can be used for further analysis of the problem, as seen in Figure 6-86, Figure 6-87, and Figure 6-88. An operator can see previous event messages associated with the high CPU threshold crossing alert. Also, historical performance related KPI statistics such as CPU can be viewed for both the ESXi host and the VM.

Figure 6-86. Event That Triggered the Performance Impact



Status	Component	Event Class	Severity	Summary
		/Perf	●	threshold of CriticalHighCPU exceeded: current value 10000.000000
	Centos-1.test.vm	/Perf	●	threshold of CriticalHigCPuVM exceeded: current value 25448.000000
		/VMware/A...	ⓘ	Alarm 'Host cpu usage' on orch-prod-palo-2.mgmt.test changed from Yellow to Red
	Centos-1.test.vm	/VMware/A...	ⓘ	Alarm 'Virtual machine cpu usage' on Centos-1.test.vm changed from Yellow to Red
	Centos-1.test.vm	/VMware	ⓘ	Virtual machine Centos-1.test.vm in cluster 1000VTestCluster in dcpod2-test-customer-vc501-orch50 is vSphere HA Prot

Figure 6-87. Historical CPU Utilization on ESXi Host

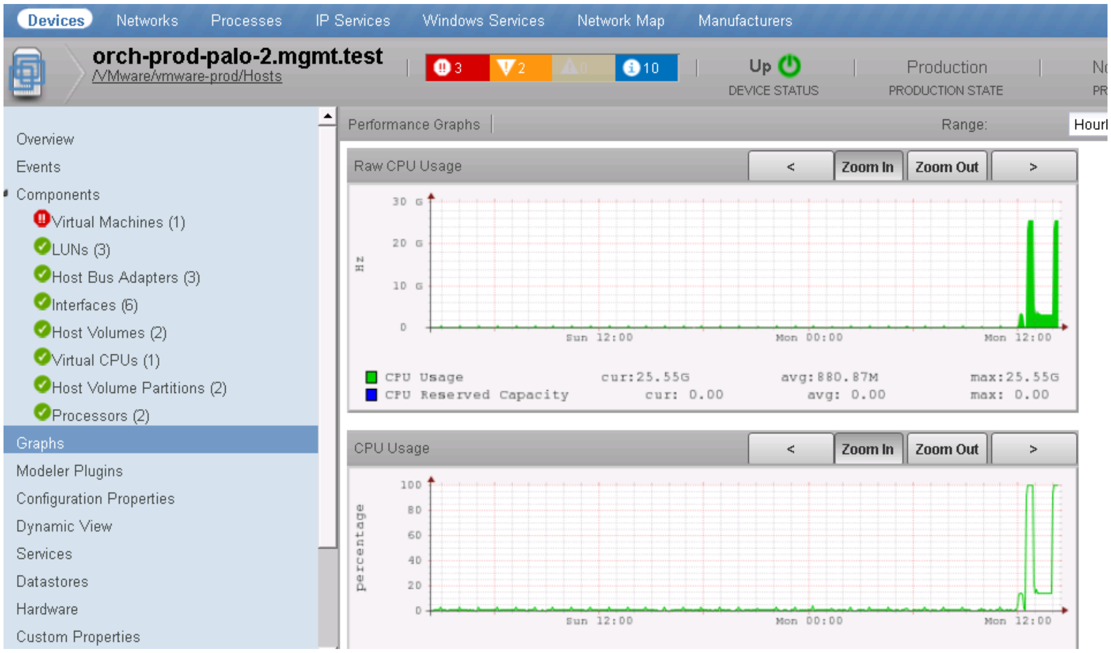
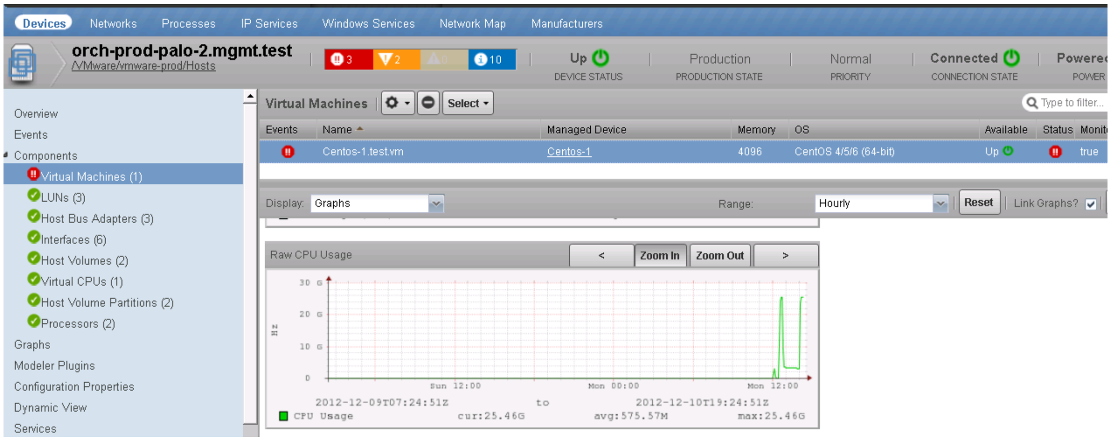


Figure 6-88. Historical CPU Utilization on Virtual Machine



Events	Name	Managed Device	Memory	OS	Available	Status	Monit
	Centos-1.test.vm	Centos-1	4096	CentOS 4/5/6 (64-bit)	Up		true

Display: Graphs Range: Hourly Reset Link Graphs? ☑

Raw CPU Usage
<
Zoom In
Zoom Out
>
Hz
30 G
20 G
10 G
0
Sun 12:00
Mon 00:00
Mon 12:00
2012-12-09T07:24:51Z to 2012-12-10T19:24:51Z
CPU Usage cur:25.46G avg:575.57M max:25.46G

CHAPTER 7

CLSA-VMDC 3.0 Enhancements

Cloud Service Assurance for Virtualized Multiservice Data Center (CLSA-VMDC) 3.0 includes expanded support for several new features and VMDC device models. This chapter presents the following topics:

- Section 7.1 EMC. Zenoss Cloud Service Assurance (CSA) adds support for EMC VMAX and VNX storage devices. EMC discovery and monitoring is achieved by using EMC's implementation of the Storage Networking Industry Association (SNIA) Storage Management Initiative Specification (SMI-S) Provider interface.
- Section 7.2 Adaptive Security Appliance Services Module (ASASM). Zenoss CSA extends support for VMDC load balancing components to include the ASASM.
- Section 7.3 Nexus 3000. While not specifically included in the VMDC Infrastructure as a Service (IaaS) architecture, Nexus 3000 support has been added to Zenoss CSA to meet Cisco field requests.
- Section 7.4 FabricPath Line Card (FPLC). The VMDC 3.0 design incorporates the Nexus switching product family FabricPath technology. Zenoss CSA adds support for the FPLC.
- Section 7.5 UCS 6200 Fabric Interconnect. The Unified Computing System (UCS) product line has been extended to support the UCS 6200 Fabric Interconnect model including discovery, monitoring, and compute service impact modeling.
- Section 7.6 Nexus 1010. The Nexus product family has been extended to support discovery and monitoring of the Nexus 1000V Virtual Supervisor Module (VSM) on the Nexus 1010.
- Section 7.7 Zenoss Sample Tenant Portal. An overview of the Zenoss sample tenant portal and the JavaScript Object Notation (JSON) Application Programming Interface (API) is presented.

7.1 EMC

With the release of Zenoss 4.2.3, EMC VNX and EMC VMAX storage arrays have been added to the list of supported storage devices. This section reviews the installation of the infrastructure required for EMC support, along with general use scenarios within Zenoss for monitoring EMC VNX and EMC VMAX arrays.

What is supported?

- EMC VNX Block
- EMC Symmetrix VMAX

What is not supported?

- EMC VNX File (limitation of the EMC Storage Management Initiative Specification (SMI-S) Provider)
- EMC VNX and EMC VMAX being served from the same EMC SMI-S Provider.
- EMC VMAX Connectivity Monitoring (limitation of the EMC SMI-S Provider for fiber connected VMAX arrays)
- EMC Simulated Arrays

The remainder of this section is split into the following subsections:

- Section 7.1.1 SMI-S Provider Installation and Configuration
- Section 7.1.2 Accessing EMC Arrays from the SMI-S Provider
- Section 7.1.3 Adding and Monitoring EMC Arrays Within Zenoss

7.1.1 SMI-S Provider Installation and Configuration

Zenoss uses the EMC Storage Management Initiative Specification (SMI-S) Provider to gain access to the EMC VNX and EMC VMAX arrays. This section covers the installation and configuration of the EMC SMI-S Provider. Prior to installing the EMC SMI-S Provider, review the official EMC SMI-S installation manual along with the release notes for the EMC SMI-S Provider, as noted in Appendix # Related Documentation.

Potential EMC SMI-S Provider Installation Conflicts

One resource issue outlined in the release notes includes a specific warning when installing the EMC SMI-S Provider on the same machine as either the EMC Control Center (ECC) or Unisphere application. The release notes outline a potential workaround for the conflict, otherwise, it is recommended that the EMC SMI-S Provider not be installed on the same machine with ECC or Unisphere.

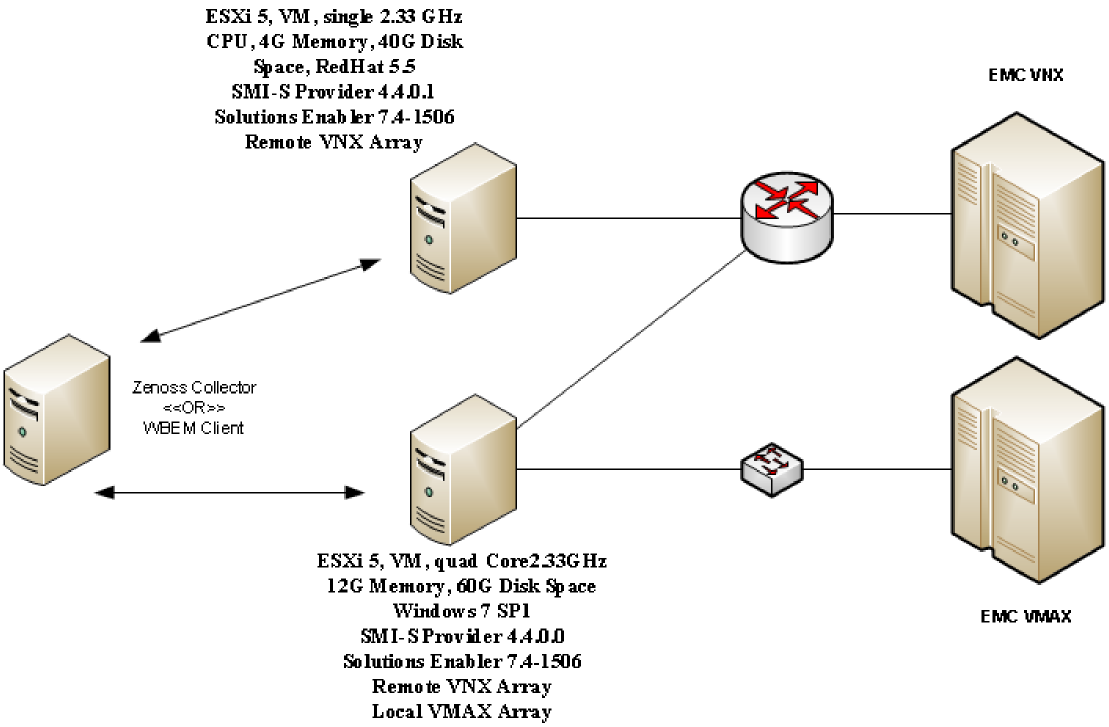
The default ports associated with the EMC SMI-S Provider are 5988 and 5989 for the EMC Common Object Module (ECOM) server and 5986 and 5986 for Web Services-Management (WS-MAN). Confirm that these ports are not active on the system prior to the installation using the **netstat -a** command to show the active ports. Refer to the "Restrict Active Ports" portion of this section for information on changing the default ports.

Cisco SDU Verification Network

For the purpose of this document, the network below was used to confirm the functionality of Zenoss with both the EMC VNX and EMC VMAX arrays. The configuration allowed various combinations to be tested, including the following:

- Microsoft Windows SMI-S Provider
- Linux SMI-S Provider
- Single Array SMI-S Deployments
- Multi-Array Homogeneous SMI-S Provider Deployments
- Heterogeneous SMI-S Provider Deployments

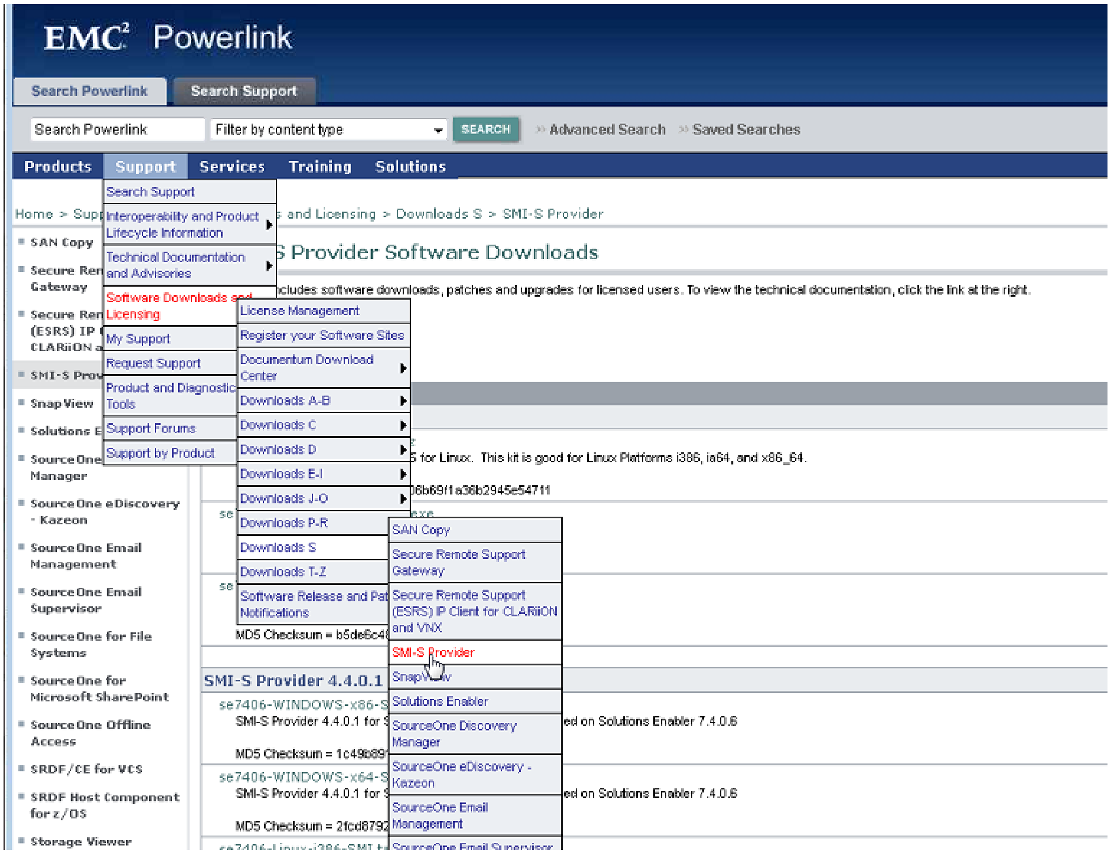
Figure 7-1. Zenoss EMC Verification Network



Download the EMC SMI-S Provider

The EMC SMI-S Provider should be downloaded from the EMC Powerlink support site and installed on either a Linux or Windows server.

Figure 7-2. EMC Powerlink Download Page


EMC² Powerlink
Search Powerlink
Search Support
Search Powerlink
Filter by content type
SEARCH
Advanced Search
Saved Searches
Products
Support
Services
Training
Solutions
Search Support
Interoperability and Product Lifecycle Information
Technical Documentation and Advisories
Software Downloads and Licensing
My Support
Request Support
Product and Diagnostic Tools
Support Forums
Support by Product
License Management
Register your Software Sites
Documentum Download Center
Downloads A-B
Downloads C
Downloads D
Downloads E-I
Downloads J-O
Downloads P-R
Downloads S
Downloads T-Z
Software Release and Pat Notifications
SAN Copy
Secure Remote Support Gateway
Secure Remote Support (ESRS) IP Client for CLARiiON and VNX
SMI-S Provider
SnapView
Solutions Enabler
SourceOne Discovery Manager
SourceOne eDiscovery - Kazeon
SourceOne Email Management
SourceOne Email Supervisor
Home > Sup s and Licensing > Downloads S > SMI-S Provider
SAN Copy
Secure Ren Gateway
Secure Ren (ESRS) IP CLARiiON a
SMI-S Prov
SnapView
Solutions E
SourceOne Manager
SourceOne eDiscovery - Kazeon
SourceOne Email Management
SourceOne Email Supervisor
SourceOne for File Systems
SourceOne for Microsoft SharePoint
SourceOne Offline Access
SRDF/CE for VCS
SRDF Host Component for z/OS
Storage Viewer
S Provider Software Downloads
SMI-S Provider 4.4.0.1

cludes software downloads, patches and upgrades for licensed users. To view the technical documentation, click the link at the right.

5 for Linux. This kit is good for Linux Platforms i386, ia64, and x86_64.

06b69f1a36b2945e54711

MD5 Checksum = b5de6c4

se7406-WINDOWS-x86-S
SMI-S Provider 4.4.0.1 for S ed on Solutions Enabler 7.4.0.6
MD5 Checksum = 1c49b89

se7406-WINDOWS-x64-S
SMI-S Provider 4.4.0.1 for S ed on Solutions Enabler 7.4.0.6
MD5 Checksum = 2fcd8792

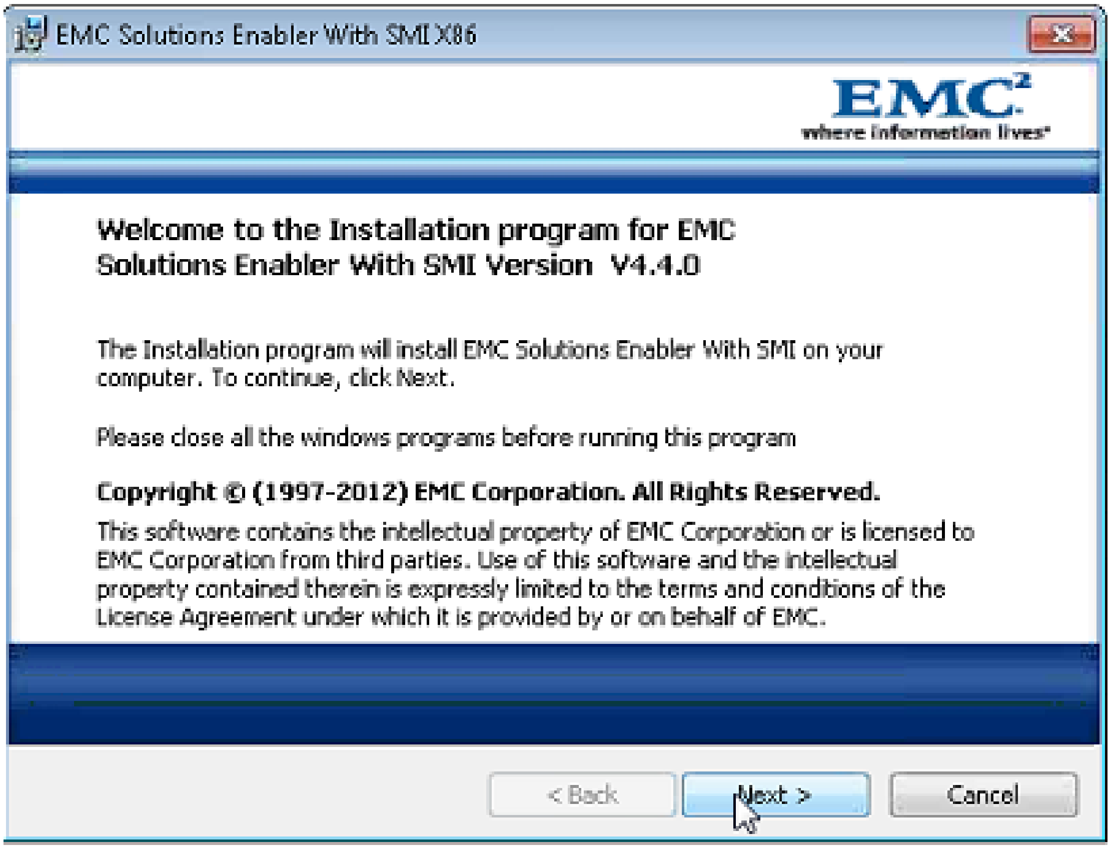
EMC Solutions Enabler with SMI Installation

The EMC SMI-S Provider is supported on both Windows and Linux-based systems. Installations examples included cover both Windows and Linux versions.

Windows Installation

With the appropriate Windows file downloaded from the EMC Powerlink site, select the executable installation file and follow the installation example below.

Figure 7-3. Initial Window Installation - Select Next



EMC Solutions Enabler With SMI X86

Welcome to the Installation program for EMC Solutions Enabler With SMI Version V4.4.0

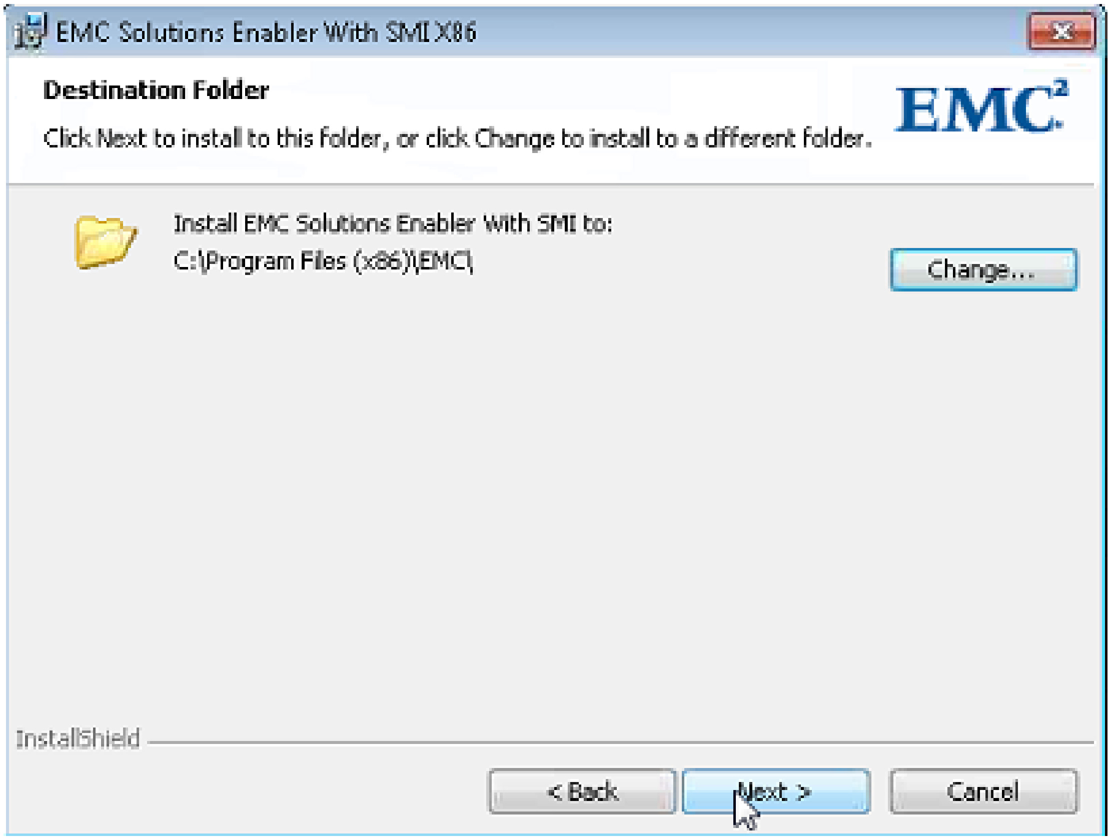
The Installation program will install EMC Solutions Enabler With SMI on your computer. To continue, click Next.

Please close all the windows programs before running this program

Copyright © (1997-2012) EMC Corporation. All Rights Reserved.

This software contains the intellectual property of EMC Corporation or is licensed to EMC Corporation from third parties. Use of this software and the intellectual property contained therein is expressly limited to the terms and conditions of the License Agreement under which it is provided by or on behalf of EMC.

Figure 7-4. Windows Installation Location - Select Next or Change



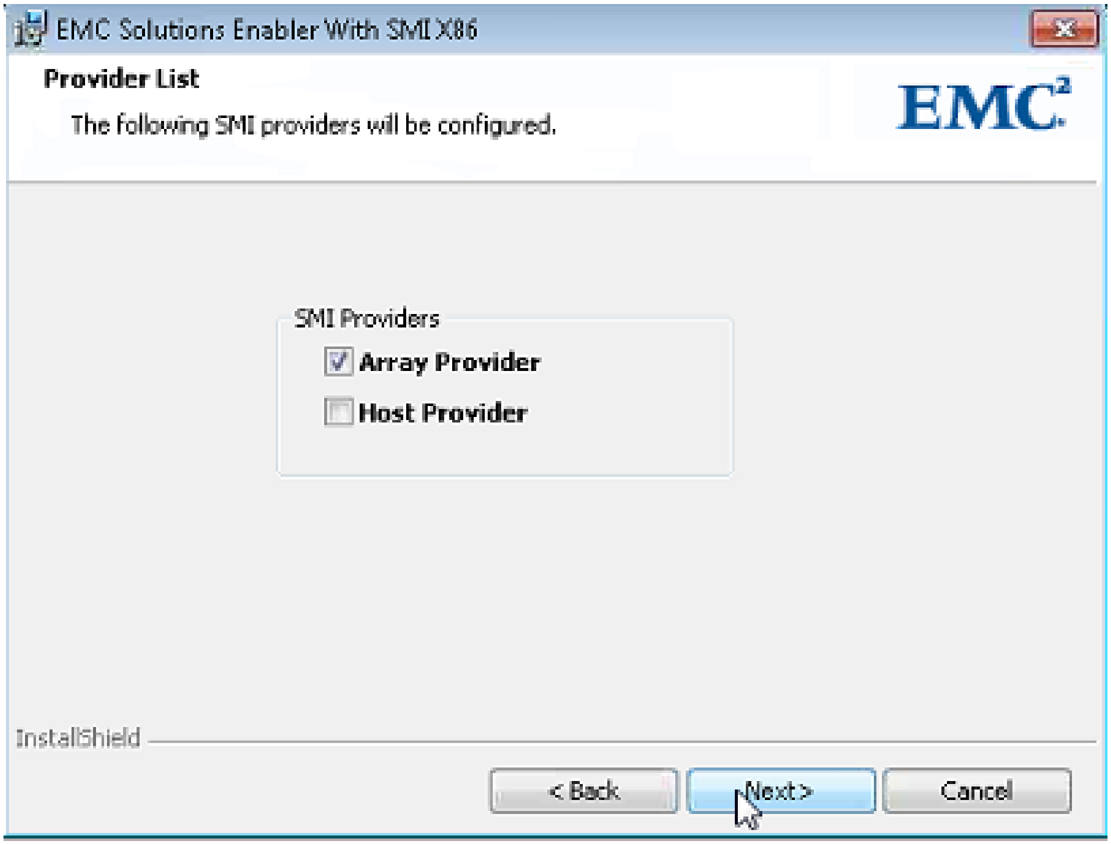
EMC Solutions Enabler With SMI X86

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install EMC Solutions Enabler With SMI to:
C:\Program Files (x86)\EMC\

Figure 7-5. SMI-S Provider Selection - Select Next



EMC Solutions Enabler With SMI X86

Provider List

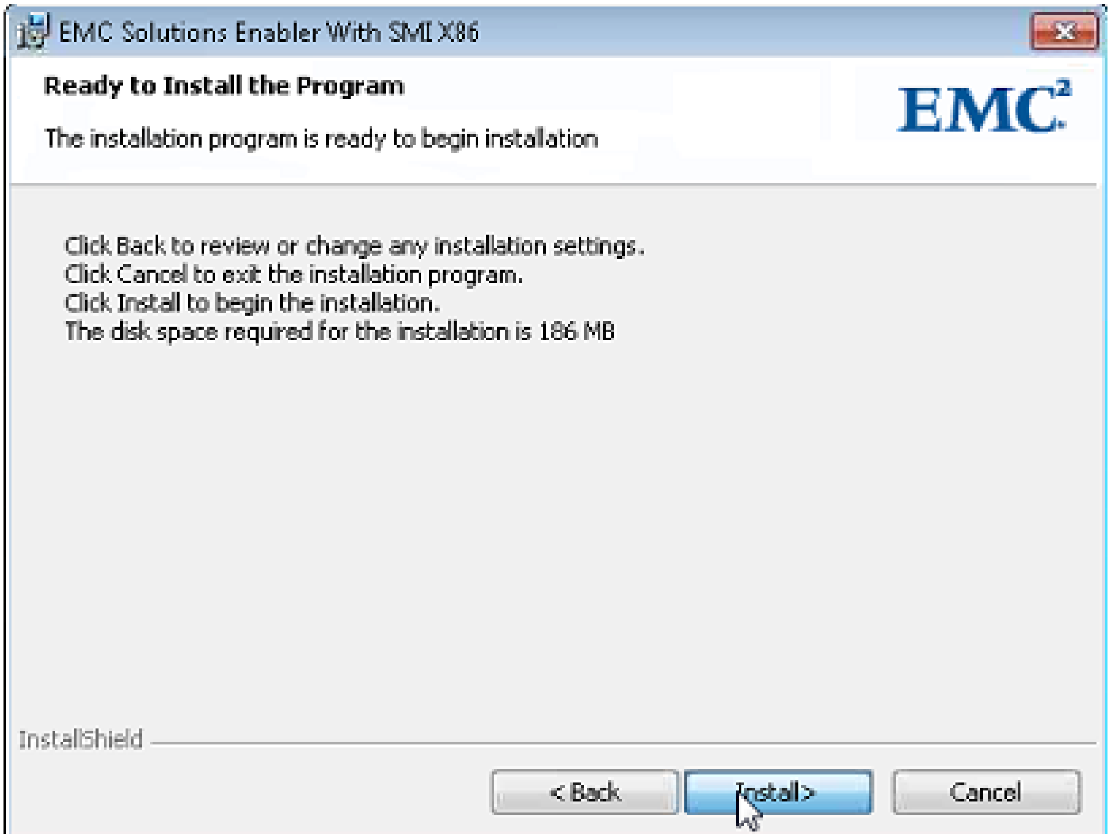
The following SMI providers will be configured.

SMI Providers

☑ **Array Provider**

□ **Host Provider**

Figure 7-6. Final Install Confirmation - Select Install



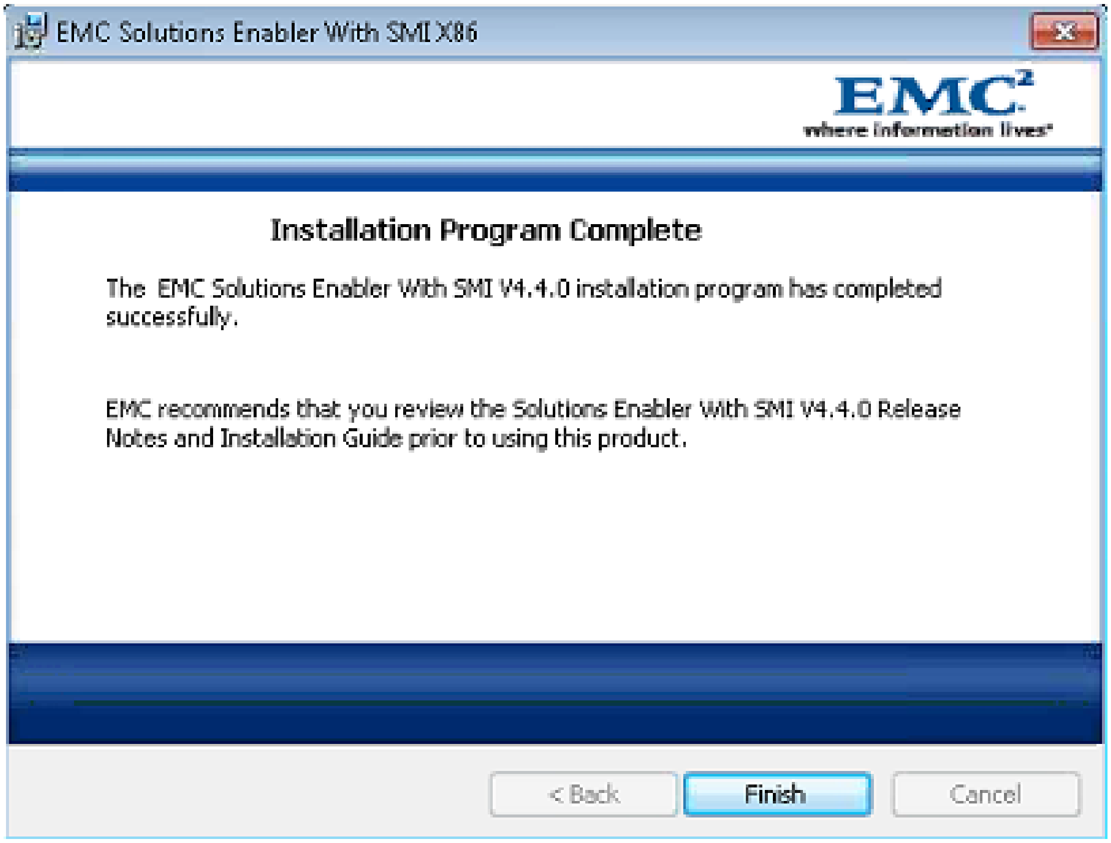
EMC Solutions Enabler With SMI X86

Ready to Install the Program

The installation program is ready to begin installation

Click Back to review or change any installation settings.
Click Cancel to exit the installation program.
Click Install to begin the installation.
The disk space required for the installation is 186 MB

Figure 7-7. Installation Complete - Select Finish



Installation Program Complete

The EMC Solutions Enabler With SMI V4.4.0 installation program has completed successfully.

EMC recommends that you review the Solutions Enabler With SMI V4.4.0 Release Notes and Installation Guide prior to using this product.

Installation of a Unix Based SMI-S Provider

The SMI-S Provider can be downloaded from the EMC Powerlink support site. The Linux bundles come in combined platform groups, so, i386 and x64 are combined into a single installation package even though they may be marked i386. The installation notes on the EMC Powerlink site provide additional information.

Linux Installation Process

With the downloaded gzipped tar file, complete the Linux installation using the following example:

```
[root@generic smi-s]# tar --ungzip -xvf se7406-Linux-i386-SMI.tar.gz
se7406_install.sh
se7406-Linux-i386-SMI.rpm
sepubkey.asc
emc_se_linux_response_file

[root@generic smi-s]\# ./se7406_install.sh -install

\#-------------------------------------------------------------------------\-
# EMC Installation Manager
\#-------------------------------------------------------------------------\-
Copyright (c) [1997-2012] EMC Corporation. All Rights Reserved.

This software contains the intellectual property of EMC Corporation or is
licensed to EMC Corporation from third parties. Use of this software and the
intellectual property contained therein is expressly limited to the terms and
conditions of the License Agreement under which it is provided by or on behalf
of EMC.
```

```
Do you want to import public key for verifying Digital Signatures? [Y]:

Solutions Enabler Native Installer [SMI] Kit Location : /i386

Checking for OS version compatibility......
Checking for previous installation of Solutions Enabler......

Checking for active processes.....

Checking for active SYMCLI components...

Checking for LIBGCC version compatibility......

Installing symcli-data-V7.4.0-6.i386.rpm.....

Installing symcli-thincore-V7.4.0-6.i386.rpm.....

Installing symcli-base-V7.4.0-6.i386.rpm.....

Installing symcli-symcli-V7.4.0-6.i386.rpm.....

Installing symcli-64bit-V7.4.0-6.x86_64.rpm.....

Installing symcli-symrecover-V7.4.0-6.i386.rpm.....

Installing symcli-smi64-V7.4.0-6.x86_64.rpm.....

Enabling stordaemon...

Installation Program Complete

The EMC Solutions Enabler SMI V4.4.0 installation program has completed successfully.
EMC recommends that you review the Solutions Enabler SMI V4.4.0 Release Notes and
Installation Guide prior to using this product.

\#-----------------------------------------------------------------------------\-
# The following HAS BEEN INSTALLED in /opt/emc via the rpm utility.
\#-----------------------------------------------------------------------------\-
ITEM  PRODUCT                                          VERSION
01    EMC Solutions Enabler                            V7.4.0.6
02    SMI  KIT                                         V4.4.0
\#-----------------------------------------------------------------------------\-
```

Starting, Stopping, and Status Checking the SMI-S Provider

The EMC SMI-S Provider includes several daemons. As part of the installation process, the emc_storapid service is installed and configured. The examples below cover the use of the emc_storapid service. The additional daemon associated with the EMC SMI-S Provider is the ECOM daemon. The EMC ECOM daemon is not fully configured during the installation of the EMC SMI-S Provider. Additional steps are outlined to finish the configuration of the ECOM daemon.

```
[root@generic bin]\# service emc_storapid status
#* #** Daemon storapid is not currently running

[root@generic bin]\# service emc_storapid start
Starting storapid
Waiting for daemon to start.  This may take several seconds.

[root@generic bin]\# service emc_storapid status

Daemon State                                     : Running
Daemon Start Time                                : Thu Nov  8 16:56:02 2012
Version                                          : V7.4-1506 (0.6)
Auto-Restart by Watchdog                         : Enabled
```

```
Total Number of Connections                     : 1
Number of Active Connections                    : 0
Total Number of Requests                        : 0

Gatekeeper Management                           :
Gatekeeper (GK) Management State                : Running
GK Management Total open GKs                    : 0
GK Management open GKs highwater                : 0
Allow PPath Native Devs as GKs                  : N/A
MPGK Selection                                  : conditional
Total File Descriptors                          : 20

Feature options                                 :
Parallel Discovery                            : Enabled
Parallel Inquiry                              : Enabled
Internode Lock Recovery                       : Disabled
Lock Information Export                       : Enabled
Background Audit Logging                      : Disabled
Background App Registration                   : Enabled

Config options                                  :
Resource Recovery Interval                    : 5 second(s)
Parallel Discovery Max Threads                : 6
Parallel Inquiry Max Threads                  : 6
Device Inquiry Timeout                        : 60 second(s)
Internode Lock Recovery Heartbeat             : N/A
Background App Reg Interval                   : 900 second(s)
Persistent DEL Recovery Interval              : 60 second(s)
SFS File Recovery Interval                    : 7200 second(s)
Maximum FDs allowed                           : 1024

[root@generic bin]\# service emc_storapid stop
Stopping storapid
storapid                      Told to shutdown
Waiting for daemon(s) to shutdown.  This may take several seconds.

[root@generic bin]\# service emc_storapid status
*** Daemon storapid is not currently running
```

The only documented method for stopping the EMC ECOM daemon is a manual kill or reboot. To provide an automated method for ECOM daemon starting and stopping, the sample script below can be used for the EMC ECOM daemon. Installation instructions are included as part of the script.

```
#!/bin/sh
#
# ecom          This shell script takes care of starting and stopping
#               emc ecom support
#
# chkconfig: 345 99 01
# description: EMC ECOM startup script
#
# Script installation:
#
#   copy the full script code into /etc/init.d/emc_ecom
#   chmod 755 /etc/init.d/emc_ecom
#   chown root:root /etc/init.d/emc_ecom
#   chkconfig --add emc_ecom
#   chkconfig --level 345 emc_ecom on
#

# Source function library.
. /etc/rc.d/init.d/functions

OPTIONS="-d"

# Source ECOM configuration.
if [ -e /etc/sysconfig/ecom ] ; then
```

```
        . /etc/sysconfig/ecom
fi

RETVAL=0
prog="ECOM"

start() {
        echo -n $"Starting $prog: "
        echo
        daemon /opt/emc/ECIM/ECOM/bin/ECOM $OPTIONS
        RETVAL=$?
        echo
        return $RETVAL
}

stop() {
        # Stop service.
        echo -n $"Shutting down $prog: "
        killproc $prog
        echo
        return $RETVAL
}

restart(){
        stop
        start
}

# See how we were called.
case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        status)
                status $prog
                ;;
        restart)
                stop
                start
                ;;
        *)
                echo $"Usage: $0 {start|stop|restart|status}"
                exit 1
esac

exit $RETVAL
```

Below are examples for starting, stopping, checking the status, and restarting the EMC ECOM daemon using the supplied sample script.

```
[root@generic init.d]# service emc_ecom start
Starting ECOM:
-I- Loading configuration files from /opt/emc/ECIM/ECOM/conf
-I- Base directory set to /opt/emc/ECIM/ECOM/
                                                           [  OK  ]

[root@generic init.d]# service emc_ecom status
ECOM (pid 21855) is running...

[root@generic init.d]# service emc_ecom restart
Shutting down ECOM:                                        [  OK  ]
Starting ECOM:
-I- Loading configuration files from /opt/emc/ECIM/ECOM/conf
-I- Base directory set to /opt/emc/ECIM/ECOM/
```

```
                                                           [  OK  ]

[root@generic init.d]# service emc_ecom stop
Shutting down ECOM:                                        [  OK  ]
```

Security Considerations for the EMC SMI-S Provider

Much like any other system, controlling access to the SMI-S Provider is important due to the fact that some write capabilities are possible within the SMI-S Provider. General recommendations apply for the EMC SMI-S Provider:

- Reduce active ports.
- Remove non-SSL ports.
- Change default ports.
- Configure a non-default user and password for the SMI-S Provider.
- Configure iptables or a Microsoft Firewall to restrict access to specific remote hosts. The configuration of iptables or a Microsoft Firewall is not specifically addressed beyond this notation. The use and configuration of iptables or a Microsoft Firewall is left to the reader to investigate.

Restrict Active Ports

When using the default installation for the EMC SMI-S Provider, up to four ports are enabled for both SSL and non-SSL connections within the TestSmiProvider UI. Executing the **slp** command shows all of the active ports on the system related to the SMI-S Provider.

```
(localhost:5985) ? slp
service:wbem:[https://172.18.114.7:5986]
service:wbem:[http://172.18.114.7:5985]
service:wbem:[https://172.18.114.7:5989]
service:wbem:[http://172.18.114.7:5988]
service:wbem:[https://172.18.114.7:5986]
service:wbem:[http://172.18.114.7:5985]
service:wbem:[https://172.18.114.7:5989]
service:wbem:[http://172.18.114.7:5988]
Please press enter key to continue...
```

To reduce the number of active ports, update the **<installDIR>/emc/ECIM/ECOM/conf/Port_settings.xml** file then restart the SMI-S processes.

Default Port Settings - Four Active Ports (Both SSL and Non-SSL)

```
<ECOMSettings>
<ECOMSetting Name="Port0">
<port>5988</port>
<secure>false</secure>
<slp>true</slp>
</ECOMSetting>

<ECOMSetting Name="Port1">
<port>5989</port>
<secure>true</secure>
<slp>true</slp>
</ECOMSetting>

<ECOMSetting Name="Port2">
<port>5985</port>
<secure>false</secure>
<slp>true</slp>
</ECOMSetting>
```

```
<ECOMSetting Name="Port3">
<port>5986</port>
<secure>true</secure>
<slp>true</slp>
</ECOMSetting>

</ECOMSettings>
```

Reduced Port Settings Using a Non-default Port - One Active SSL Port

```
<ECOMSettings>
<ECOMSetting Name="Port0">
<port>5990</port>
<secure>true</secure>
<slp>true</slp>
</ECOMSetting>
</ECOMSettings>
```

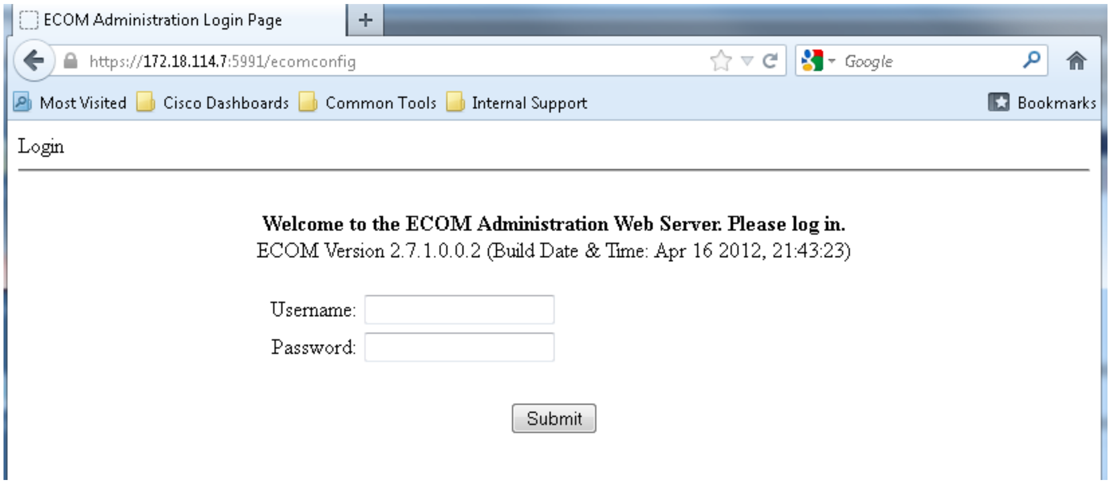
Reduced Active Ports

```
(localhost:5990) ? slp
service:wbem:[https://172.18.114.7:5990]
service:wbem:[https://172.18.114.7:5990]
Please press enter key to continue...
```

Configure a Non-default User and Password for the SMI-S Provider

Using the active running ports on the SMI-S Provider, access the ECOM Administration Login page and log in using the original default SMI-S username and password.

Figure 7-8. Login Screen for the ECOM Administration Login Page



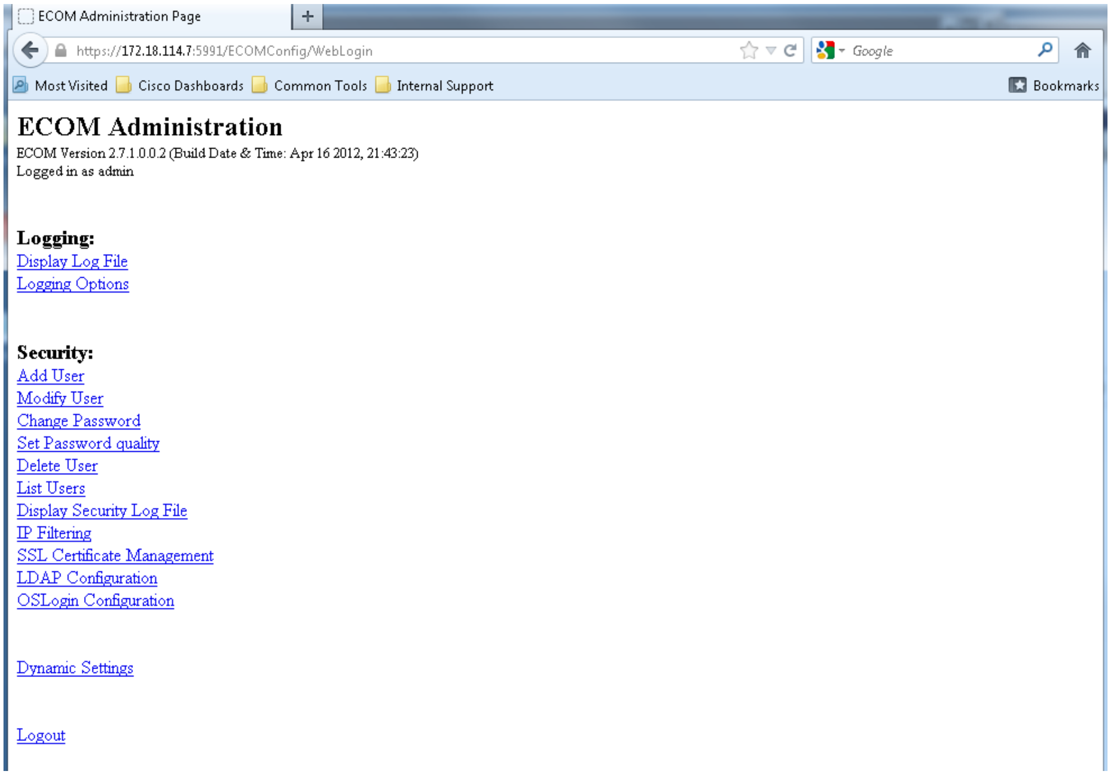
Login

Welcome to the ECOM Administration Web Server. Please log in.
ECOM Version 2.7.1.0.0.2 (Build Date & Time: Apr 16 2012, 21:43:23)

Username:
Password:

After logging in, the ECOM administration page is displayed. At this point, standard administrative capabilities are available. Refer to the SMI-S and ECOM user guides for more detailed usage of the administration page.

Figure 7-9. ECOM Administration Page



ECOM Administration

ECOM Version 2.7.1.0.0.2 (Build Date & Time: Apr 16 2012, 21:43:23)
Logged in as admin

Logging:
Display Log File
Logging Options

Security:
Add User
Modify User
Change Password
Set Password quality
Delete User
List Users
Display Security Log File
IP Filtering
SSL Certificate Management
LDAP Configuration
OSLogin Configuration

Dynamic Settings

Logout

7.1.2 Accessing EMC Arrays from the SMI-S Provider

The following section provides examples of tasks that are required within the SMI-S Provider in order to support the monitoring of EMC arrays from within Zenoss. Operations covered include adding arrays, deleting arrays, and displaying configured arrays within the SMI-S Provider.

Adding Remotely Connected EMC VNX Arrays

As noted earlier, mixing EMC VNX and EMC VMAX systems on the same SMI-S Provider is currently not supported by Zenoss. Multiple EMC arrays of the same type, either VNX or VMAX, is supported by Zenoss. As a general note, mixing EMC VNX and EMC VMAX systems is supported by EMC.

Use the **<INSTALLDIR>/emc/ECIM/ECOM/bin/TestSmiProvider addsys** command to add remote EMC VNX hosts to the SMI-S Provider. The user needs the IP address of both SP-A and SP-B, along with the appropriate user credentials. The user needs to have administrator roles in the system domain for a successful log in. The result of the addsys should be "OUTPUT : 0" if the system was successfully added to the SMI-S Provider.

```
(localhost:5990) ? addsys
Add System {y|n} [n]: y

ArrayType (1=Clar, 2=Symm) [1]:
One or more IP address or Hostname or Array ID

Elements for Addresses
IP address or hostname or array id 0 (blank to quit): 172.18.137.153
IP address or hostname or array id 1 (blank to quit): 172.18.137.154
IP address or hostname or array id 2 (blank to quit):
Address types corresponding to addresses specified above.
```

```
(1=URL, 2=IP/Nodename, 3=Array ID)
Address Type (0) [default=2]:
Address Type (1) [default=2]:
User [null]: ********
Password [null]: ********
++++ EMCAddSystem ++++
OUTPUT : 0
Legend:0=Success, 1=Not Supported, 2=Unknown, 3=Timeout, 4=Failed
       5=Invalid Parameter
       4096=Job Queued, 4097=Size Not Supported
Note: Not all above values apply to all methods - see MOF for the method.

System : //172.18.114.7/root/
emc:Clar_StorageSystem.CreationClassName="Clar_StorageSystem",Name="CLARiiON
+APM00111700778"

In 24.805600 Seconds

Please press enter key to continue...
```

Adding Locally Connected EMC VMAX Arrays

When adding EMC VMAX arrays to the SMI-S Provider, execute the **<INSTALLDIR>/emc/ECIM/ECOM/bin/TestSmiProvider discovery** command. For EMC VMAX arrays, the SMI-S Provider accesses the array via the fiber connection. Once added, the discovery process should report "OUTPUT : 0" if the process was successful.

```
(localhost:5990) ? disco

Filter {y|n} [n]:

New question

Flags {y|n} [n]:
++++ EMCDiscoverSystem ++++
OUTPUT : 0
Legend:0=Success, 1=Not Supported, 2=Unknown, 3=Timeout, 4=Failed
       5=Invalid Parameter
       4096=Job Queued, 4097=Size Not Supported
Note: Not all above values apply to all methods - see MOF for the method.

In 10.872351 Seconds

Please press enter key to continue...
```

Confirming Connected EMC Arrays

With both VNX and VMAX connections, the EMC SMI Provider Tester command **dv** allows users to display any connected EMC devices along with supplying all of the various version information related to the SMI-S Provider and the connected EMC devices.

```
(localhost:5990) ? dv
\+++\+ Display version information \+++\+

CIM ObjectManager Name: EMC:172.18.114.7

CIMOM Version: EMC CIM Server Version 2.7.1.0.0.2

SMI-S spec version: 1.6.0

SMI-S Provider version: V4.4.0.1

SMI-S Provider Location: Proxy

SMI-S Provider Server:
```

```
Linux generic-host 2.6.18-194.8.1.el5 #1 SMP Wed Jun 23 10:52:51 EDT 2010 x86_64 VM
 Guest OS (64bit Libraries)

Solutions Enabler version: V7.4-1506 0.6

Firmware version information:
(Local)  Symmetrix Array 000192601892 (VMAX-1) : 5876.82.57
(Remote) CLARiiON Array APM00111700778 (Rack Mounted VNX5700) : 05.31.000.5.704
(Remote) CLARiiON Array APM00112504463 (Rack Mounted VNX5700) : 05.31.000.5.720

Retrieve and Display data - 1 Iteration(s) In 0.611698 Seconds

Please press enter key to continue...
```

Note The output above is for example purposes only as it is an SMI-S Provider configured to be a heterogeneous server (serving both EMC VNX and EMC VMAX). Zenoss currently only supports SMI-S Providers configured as homogeneous servers.

For listing specific VNX or VMAX connections, users can enumerate instances for specific storage types to confirm active EMC connections. The data found using the SMI-S Provider Tester also corresponds to data required when systems need to be removed from the provider (covered in the next section).

Removing EMC VNX and EMC VMAX Arrays from the SMI-S Provider

Both the EMC VNX and EMC VMAX arrays can be successfully removed from the the SMI-S Provider by executing the **<INSTALLDIR>/emc/ECIM/ECOM/bin/TestSmiProvider** command remsys. Prior to running remsys, the full ObjectPath for the array is needed as part of the parameters for remsys. Using the TestSmiProvider command ei, output the ObjectPath for the EMC that is to be removed, then execute remsys to the complete the removal.

Display EMC VNX Object Path Information

The following code segment gives an example for finding the ObjectPath for an EMC VNX system. With this example, there are two VNX systems in the SMI-S Provider.

```
(localhost:5990) ? ei
Class: Clar_StorageSystem
DeepInheritance [true]:
LocalOnly [false]:
IncludeQualifiers [false]:
Property to include ('-' for all, '[empty]' for no properties)? [-]: ObjectPath
Property to include ('-' for all, '[empty]' for no properties)? [-]:
\+++\+ Testing EnumerateInstances: Clar_StorageSystem \+++\+
Instance 0:
ObjectPath : Clar_StorageSystem.CreationClassName="Clar_StorageSystem",Name="CLARiiON
+APM00111700778"   <<<<<< OBJECT PATH FOR VNX #1 >>>>>>
<INSTANCE  CLASSNAME="Clar_StorageSystem" >
</INSTANCE>

Number of instance qualifiers: 0
Number of instance properties: 0

Instance 1:
ObjectPath : Clar_StorageSystem.CreationClassName="Clar_StorageSystem",Name="CLARiiON
+APM00112504463"   <<<<<< OBJECT PATH FOR VNX #2 >>>>>>
<INSTANCE  CLASSNAME="Clar_StorageSystem" >
</INSTANCE>

Number of instance qualifiers: 0
Number of instance properties: 0
```

```
Enumerate 2 instances; repeat count 1;return data in 0.056013 seconds

Retrieve and Display data - 1 Iteration(s) In 0.072147 Seconds

Please press enter key to continue...
```

Display EMC VMAX Object Path Information

The following code segment gives an example for finding the ObjectPath for an EMC VMAX system. With this example, there is one EMC VMAX in the SMI-S Provider.

```
(localhost:5990) ? ei
Class: Symm_StorageSystem
DeepInheritance [true]:
LocalOnly [false]:
IncludeQualifiers [false]:
Property to include ('-' for all, '[empty]' for no properties)? [-]: ObjectPath
Property to include ('-' for all, '[empty]' for no properties)? [-]:
\+++\+ Testing EnumerateInstances: Symm_StorageSystem \+++\+
Instance 0:
ObjectPath : Symm_StorageSystem.CreationClassName="Symm_StorageSystem",Name="SYMMETRIX
+000192601892" <<<<<< OBJECT PATH FOR VMAX >>>>>>
<INSTANCE CLASSNAME="Symm_StorageSystem" >
</INSTANCE>

Number of instance qualifiers: 0
Number of instance properties: 0

Enumerate 1 instances; repeat count 1;return data in 0.506587 seconds

Retrieve and Display data - 1 Iteration(s) In 0.506807 Seconds

Please press enter key to continue...
```

Using the ObjectPath for the given array to be removed from the SMI-S Provider, execute the remsys command and confirm the command was successful. The code segment below shows an example of an EMC VNX being removed from the SMI-S Provider.

```
(localhost:5990) ? remsys
Remove System {y|n} [n]: y
System's ObjectPath[null]:
 Clar_StorageSystem.CreationClassName="Clar_StorageSystem",Name="CLARiiON
+APM00111700778"
About to delete system
 Clar_StorageSystem.CreationClassName="Clar_StorageSystem",Name="CLARiiON
+APM00111700778"
Are you sure {y|n} [n]: y
++++ EMCRemoveSystem ++++
OUTPUT : 0
Legend:0=Success, 1=Not Supported, 2=Unknown, 3=Timeout, 4=Failed
       5=Invalid Parameter
       4096=Job Queued, 4097=Size Not Supported
Note: Not all above values apply to all methods - see MOF for the method.

In 8.494349 Seconds

Please press enter key to continue...
```

Clearing the SMI-S Provider Database

If there are several EMC arrays attached to the SMI-S Provider and all of them are being removed, clearing the SMI-S Provider database is an alternative to manually removing all of the devices.

```
[root@generic db]# service emc_ecom stop
```

```
Shutting down ECOM:                                        [  OK  ]
[root@generic db]\# service emc_storapid stop
Stopping storapid
storapid                      Told to shutdown
Waiting for daemon(s) to shutdown.  This may take several seconds.
[root@generic db]\# rm \-f /var/symapi/db/*.bin
[root@generic db]\# rm \-f /opt/emc/ECIM/ECOM/log/\*
[root@generic db]\# service emc_storapid start
Starting storapid
Waiting for daemon to start.  This may take several seconds.
[root@generic db]# service emc_ecom start
Starting ECOM:
-I- Loading configuration files from /opt/emc/ECIM/ECOM/conf
-I- Base directory set to /opt/emc/ECIM/ECOM/
                                                           [  OK  ]
```

7.1.3 Adding and Monitoring EMC Arrays Within Zenoss

When adding new EMC arrays to Zenoss, it is important to remember that Zenoss does not communicate with EMC arrays directly, but instead through a storage domain manager, EMC's SMI-S Provider. SMI-S Providers are added and deleted as a whole. All EMC arrays attached to a given SMI-S Provider are included within the listing for the given SMI-S Provider. Adding and deleting individual EMC arrays from a Zenoss monitored SMI-S Provider requires the EMC array to be added or deleted from the associated SMI-S Provider.

Note When EMC arrays are added or deleted from monitored SMI-S Providers, a remodel (manual or scheduled) is required in order to update the inventory.

Adding an EMC VNX Array to Zenoss

Figure 7-10 and Figure 7-11 show the addition of an SMI-S Provider with two EMC VNX arrays attached. For this example, the less secure non-SSL based port has been selected. Adding a device can take several minutes, depending on the size and number of arrays attached to the SMI-S Provider. If the appropriate VNX or VMAX type selection is not correct when adding new EMC arrays, the modeling process does not work correctly.

Note Once assigned to either a VNX or VMAX class, changing the class for EMC arrays is not supported. The EMC array needs to be removed and added again with the correct type selection.

Figure 7-10. Adding EMC Arrays to Zenoss

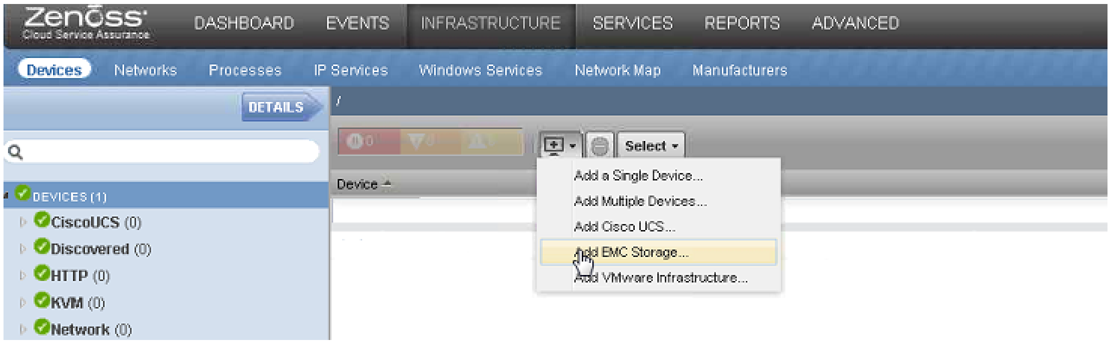
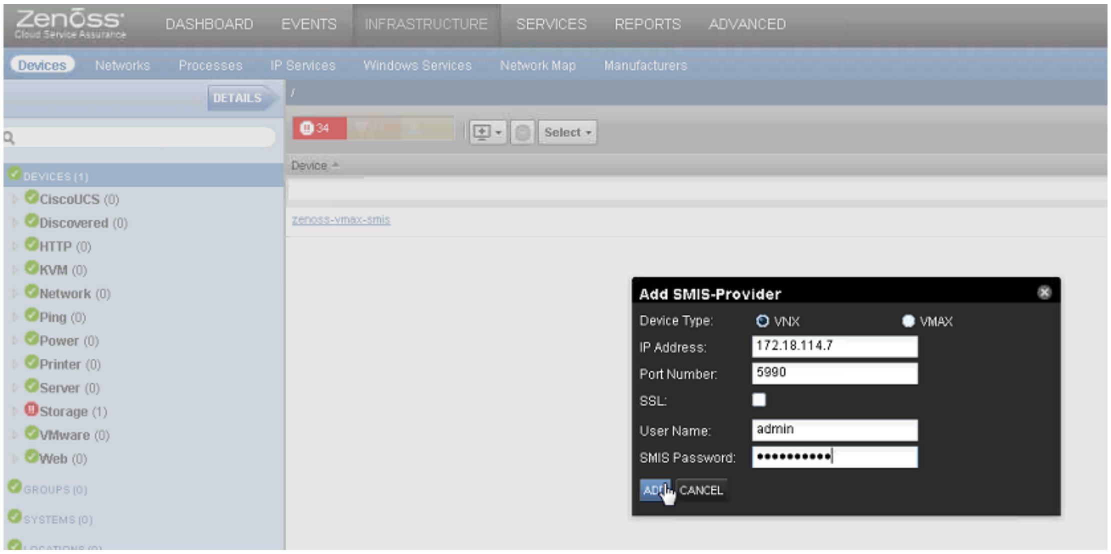


Figure 7-11 shows adding an EMC VNX array to Zenoss using a non-SSL based connection to the SMI-S Provider.

Figure 7-11. Adding EMC VNX Array to Zenoss



Add SMIS-Provider

Device Type: ◉ VNX ○ VMAX

IP Address: 172.18.114.7

Port Number: 5990

SSL: □

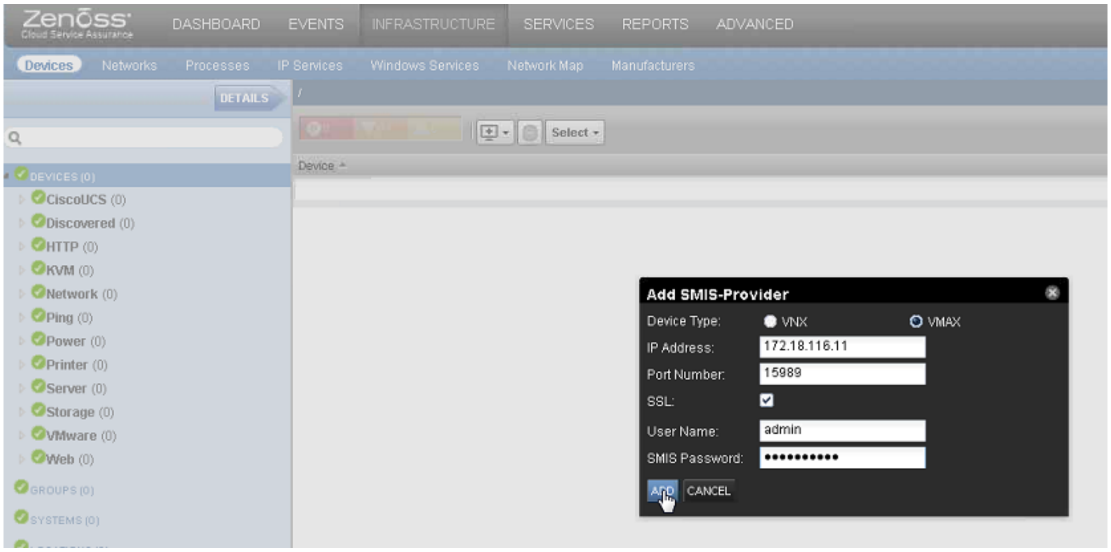
User Name: admin

SMIS Password: ••••••••••

Adding an EMC VMAX Array to Zenoss

Figure 7-12 shows the addition of an SMI-S Provider with a single EMC VMAX attached. For this example, the more secure SSL based port has been selected.

Figure 7-12. Adding EMC VMAX Array to Zenoss



Add SMIS-Provider

Device Type: ○ VNX ◉ VMAX

IP Address: 172.18.116.11

Port Number: 15989

SSL: ☑

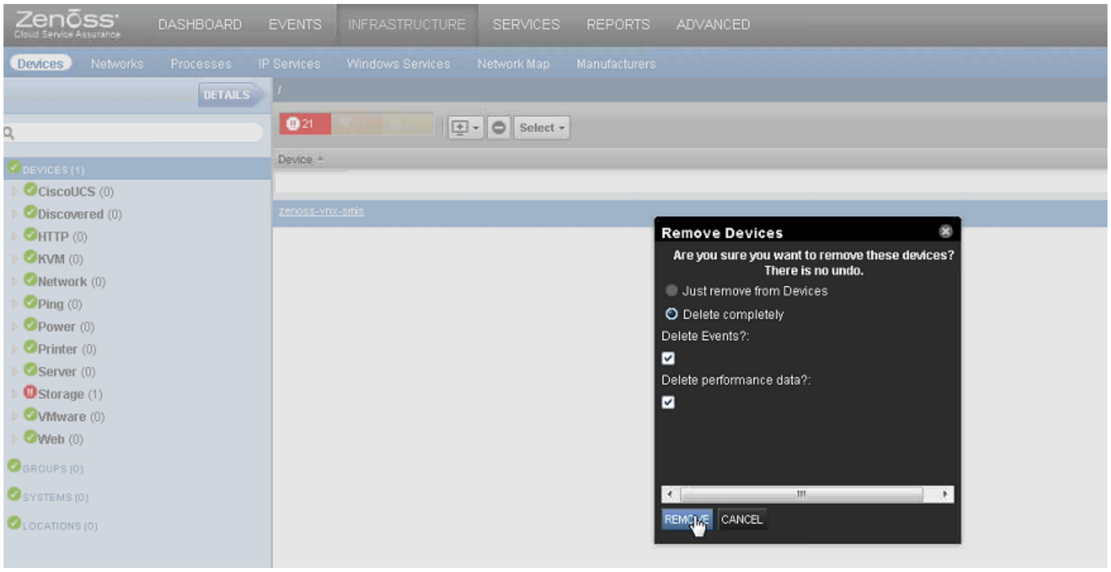
User Name: admin

SMIS Password: •••••••••••

Removing EMC Devices from Zenoss

Figure 7-13 shows the removal of the EMC VNX from Zenoss, however, all EMC devices are removed in the same manner. Attempting to delete an individual EMC array from a monitored device results in a failure banner, but the device is not removed. Removing devices can take several minutes to complete, depending on the number of devices contained in the SMI-S Provider. A browser refresh is required to remove some of the original elements in the current browser window.

Figure 7-13. Remove EMC Device from Zenoss



Remove Devices

Are you sure you want to remove these devices? There is no undo.

○ Just remove from Devices

○ Delete completely

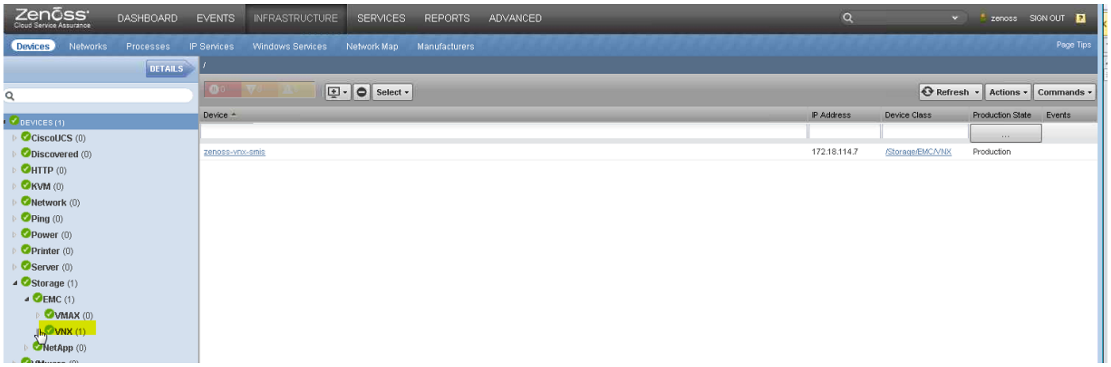
Delete Events?: ☑

Delete performance data?: ☑

Multiple EMC Array Organization

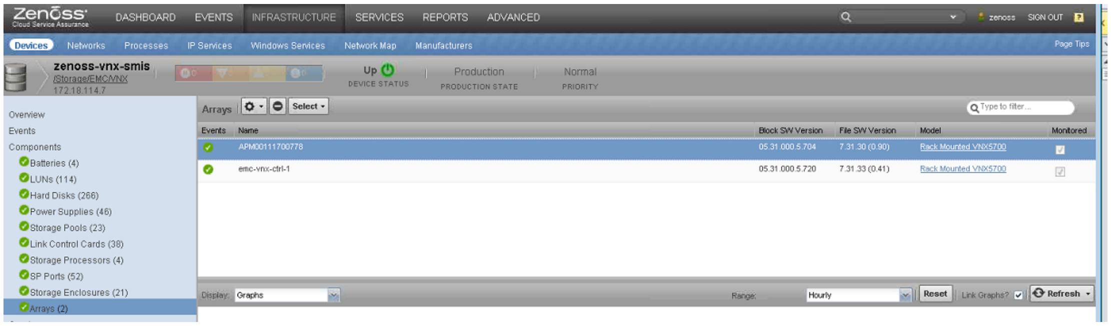
Due to the fact that Zenoss treats SMI-S Providers as EMC array containers, the number of storage devices reflected in the Zenoss display does not reflect the number of active arrays per SMI-S Provider. The only exception to this view of EMC arrays within Zenoss is when the maximum number of EMC arrays contained within an SMI-S Provider is one.

Figure 7-14. Single SMI-S Provider with Multiple VNX Arrays



Device	IP Address	Device Class	Production State	Events
zenoss-vnx-smis	172.18.114.7	/Storage/EMC/VNX	Production	

Figure 7-15. Expanded View Showing Multiple Arrays in One SMI-S Provider



zenoss-vnx-smis
/Storage/EMC/VNX
172.18.114.7

Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

Events	Name	Block SW Version	File SW Version	Model	Monitored
✓	APM00111700778	05.31.000.5.704	7.31.30 (0.90)	Rack Mounted VNX5700	☑
✓	emc-vnx-ctrl-1	05.31.000.5.720	7.31.33 (0.41)	Rack Mounted VNX5700	☑

Display: Graphs
Range: Hourly
Link Graphs? ☑

Zenoss EMC Model Definition

As users begin working with the Zenoss EMC interface, several differences can be noted between Zenoss CSA and the standard EMC management tools. With Zenoss's primary focus being operations level monitoring, some of the EMC array concepts have been collapsed in order to reduce the overall workspace being viewed by operators, but not to reduce fault coverage. Table 7-1 and Table 7-2 outline several of the Zenoss CSA to EMC mapping changes.

Table 7-1. EMC VNX to Zenoss Mapping Outline

EMC VNX	Zenoss CSA v4.2.3
LUN	Data Devices
Disks	Hard Disks
Disk Array Enclosures	Storage Enclosures

Table 7-2. EMC VMAX to Zenoss Mapping Outline

EMC VMAX	Zenoss CSA v4.2.3
Storage Groups	Storage Pools
Virtual Volumes	Data Devices
Disk Groups > Disks	Hard Disks
Back End Directors	Storage Processors
Port Group > Ports	SP Ports

7.2 ASASM

Zenoss monitors the ASASM via an SNMP interface. The ASASM supports network monitoring using SNMPv1, SNMPv2c, and SNMPv3 and supports the use of all three versions simultaneously. The ASASM supports SNMP read-only access through GET requests.

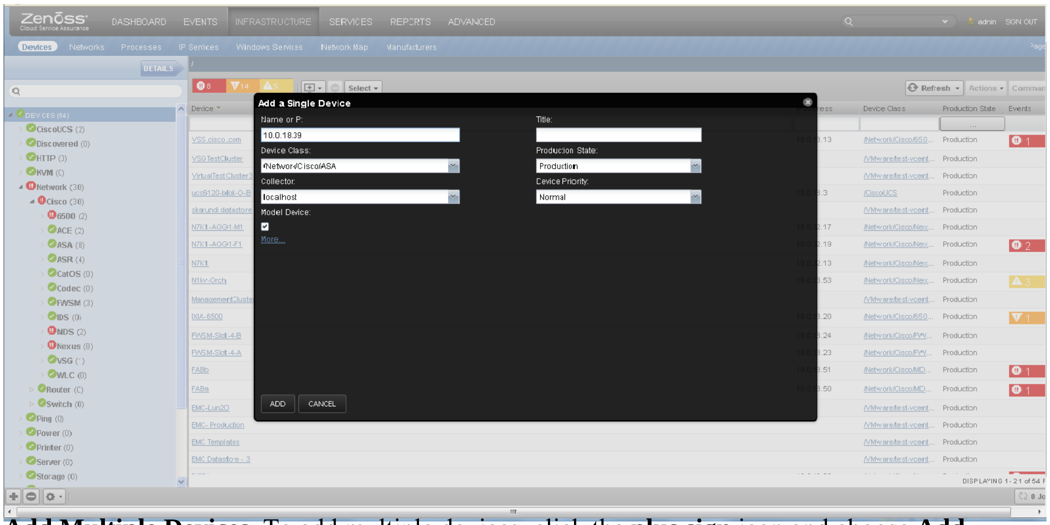
The ASASM operates within the Virtual Switch System (VSS) VMDC component in single mode/ multimode, as well as routed or transparent mode. The ASASM with multimode needs to have all of the security contexts added as separate devices. This would require that every context have a routable management IP address accessible to the Zenoss server, plus any SNMP configurations.

Discovering ASASM and ASASM Components

This section illustrates various methods to discover components by Zenoss.

1. **Add a Single Device.** To add a single device, click the **plus sign** icon under the Infrastructure Management tab and choose **Add a Single Device**. From the popup window, the user can specify the ASASM IP address or DNS name and the device name to be used within Zenoss. Select the **/Network/Cisco/ASA** device class. Click **Add**. After the device is modeled, the device appears under ASA device class. When all of the security contexts are added to Zenoss, model the admin context so that it can update its security context component.

Figure 7-16. Adding a Single Device



Add a Single Device

Name or P: 10.0.18.39

Title:

Device Class: /Network/Cisco/ASA

Production State: Production

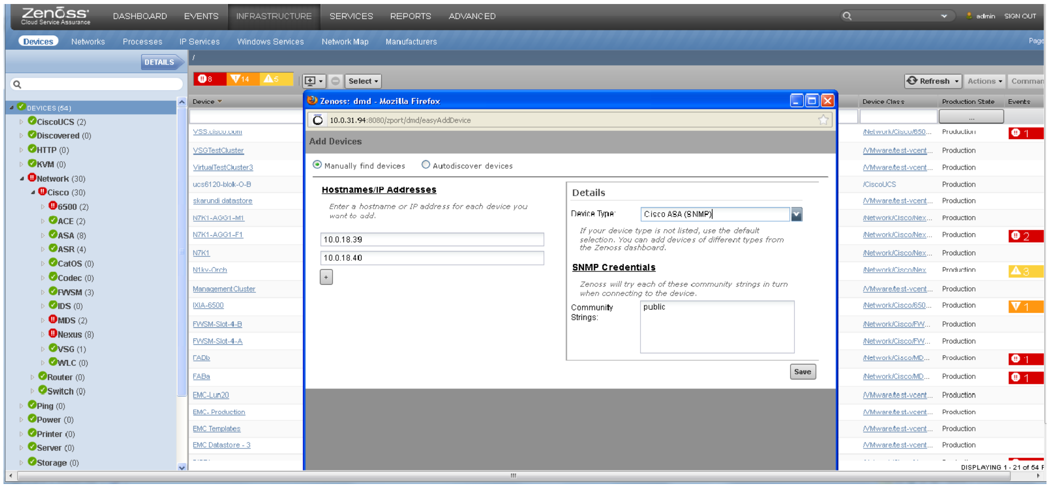
Collector: localhost

Device Priority: Normal

Model Device: ☑

2. **Add Multiple Devices.** To add multiple devices, click the **plus sign** icon and choose **Add Multiple Devices** under the Infrastructure Management tab. From the popup window, specify to manually find devices (default) and enter the ASSAM IP address or DNS name and add the details. Click the **plus sign** icon under the IP address or name entered to add another security context. Notice that the details need to be filled in again for this device. When finished, click the **Save** button. Both devices are now added.

Figure 7-17. Adding Multiple Devices



Zenoss: dmd - Mozilla Firefox

10.0.31.94:8080/zport/dmd/easyAddDevice

Add Devices

● Manually find devices ○ Autodiscover devices

Hostnames/IP Addresses

Enter a hostname or IP address for each device you want to add.

10.0.18.39

10.0.18.40

Details

Device Type: Cisco ASA (SNMP)

If your device type is not listed, use the default selection. You can add devices of different types from the Zenoss dashboard.

SNMP Credentials

Zenoss will try each of these community strings in turn when connecting to the device.

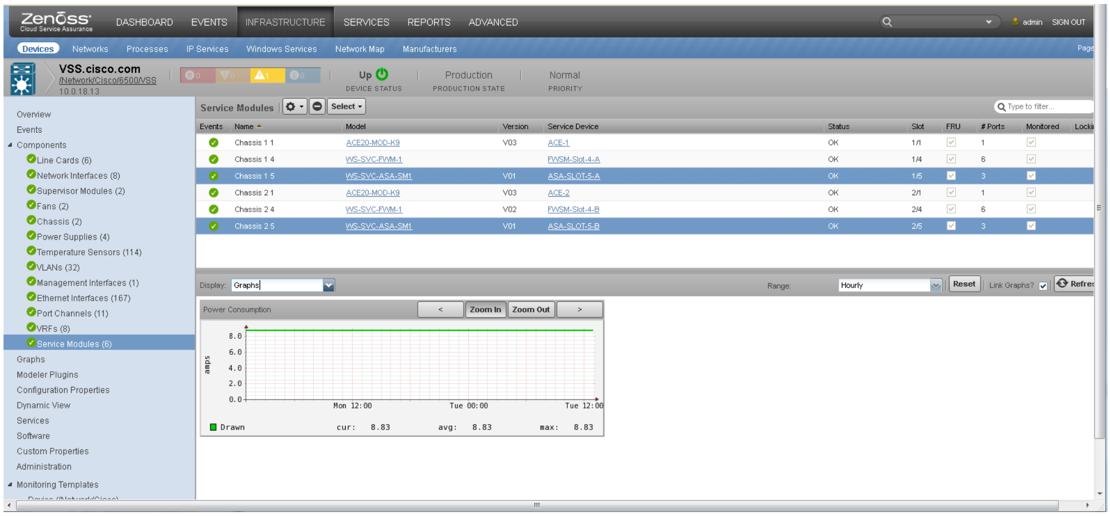
Community Strings: public

Save

Resource Monitoring

All of the KPI statistics supported by Zenoss for FWSM are also supported for ASASM. Figure 7-18 and Figure 7-19 illustrate typical KPI statistics such as VSS service module power consumption monitoring and ASASM VLAN monitoring.

Figure 7-18. Service Module Monitoring



VSS.cisco.com
/Network/Cisco/6500/VSS
10.0.18.13

Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

Service Modules

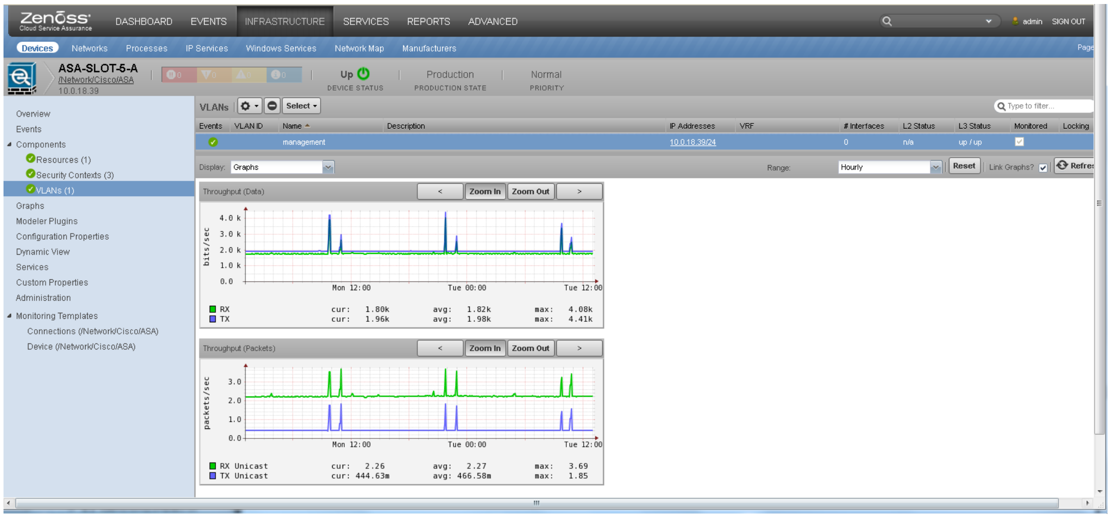
Events	Name	Model	Version	Service Device	Status	Slot	FRU	# Ports	Monitored
✓	Chassis 1 1	ACE20-MOD-K9	V03	ACE-1	OK	1/1	☑	1	☑
✓	Chassis 1 4	WS-SVC-FWM-1		FWSM-Slot-4-A	OK	1/4	☑	6	☑
✓	Chassis 1 5	WS-SVC-ASA-SM1	V01	ASA-SLOT-5-A	OK	1/5	☑	3	☑
✓	Chassis 2 1	ACE20-MOD-K9	V03	ACE-2	OK	2/1	☑	1	☑
✓	Chassis 2 4	WS-SVC-FWM-1	V02	FWSM-Slot-4-B	OK	2/4	☑	6	☑
✓	Chassis 2 5	WS-SVC-ASA-SM1	V01	ASA-SLOT-5-B	OK	2/5	☑	3	☑

Display: Graphs Range: Hourly Link Graphs? ☑

Power Consumption

Drawn cur: 8.83 avg: 8.83 max: 8.83

Figure 7-19. VLAN Monitoring



Events	VLAN ID	Name	Description	IP Addresses	VRF	# Interfaces	L2 Status	L3 Status	Monitored	Locking
✓		management		10.0.18.39/24		0	n/a	up / up	☑	

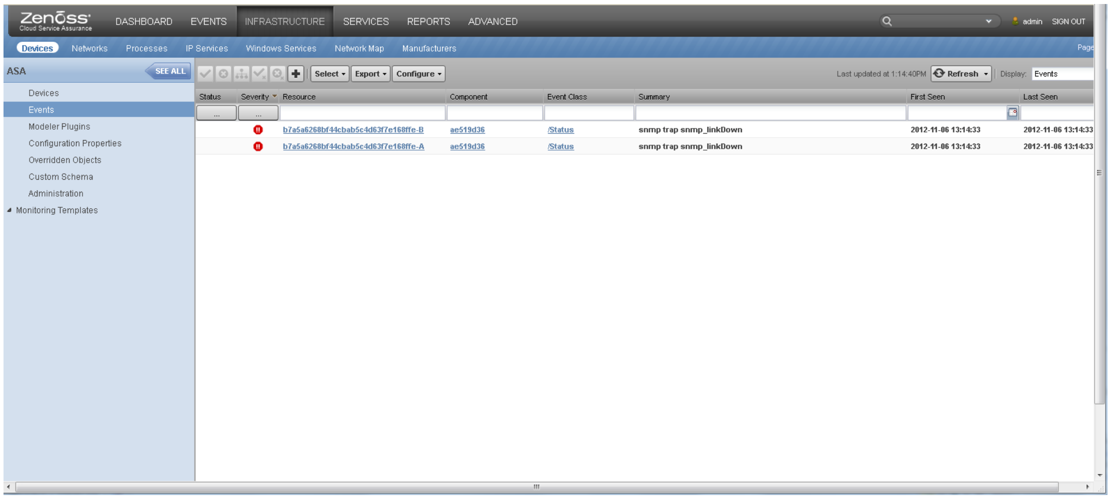
```
🟩 RX    cur:  1.80k    avg:  1.82k    max:  4.08k
🟦 TX    cur:  1.96k    avg:  1.98k    max:  4.41k
```

```
🟩 RX Unicast    cur:  2.26       avg:  2.27       max:  3.69
🟦 TX Unicast    cur:  444.63m    avg:  466.58m    max:  1.85
```

Event Monitoring

Zenoss also supports SNMP Trap event monitoring for the ASASM. Figure 7-20 shows an example trap for a link-down state under the ASASM Events view.

Figure 7-20. ASA Event Monitoring



Last updated at 1:14:40PM

Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen
	●	b7a5a6268bf44cbab5c4d63f7e168ffe-B	ae519d36	/Status	snmp trap snmp_linkDown	2012-11-06 13:14:33	2012-11-06 13:14:33
	●	b7a5a6268bf44cbab5c4d63f7e168ffe-A	ae519d36	/Status	snmp trap snmp_linkDown	2012-11-06 13:14:33	2012-11-06 13:14:33

7.3 Nexus 3000

The Nexus 3000 switching platform is not a standard device referenced by the VMDC architecture. This device has been included in CLSA-VMDC 3.0 based on field requests for Zenoss CSA support.

Features

Nexus 3000 switch platform support includes the following features:

- Discovery of the device and autodiscovery of its components
- Monitoring of KPI statistics

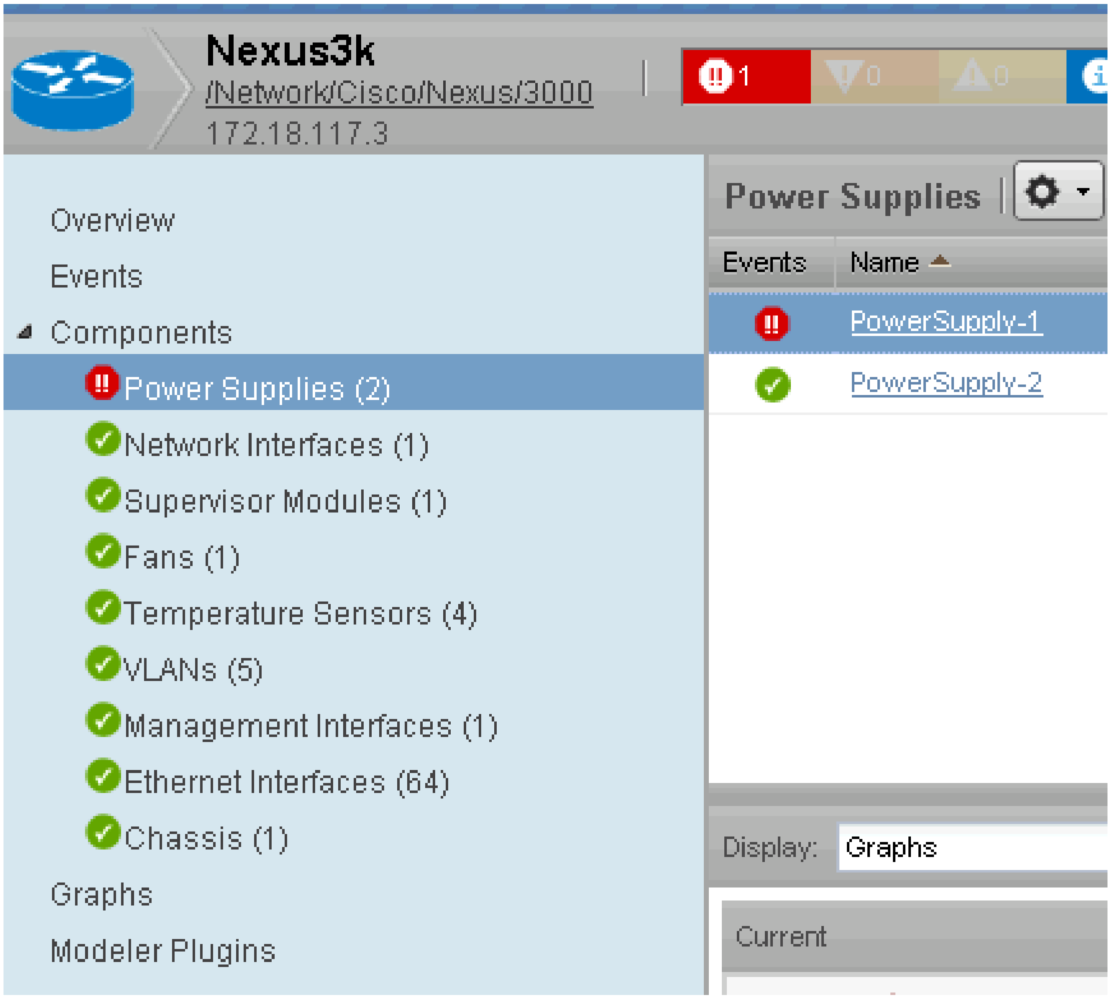
- Processing of SNMP Traps from the Nexus 3000

CLSA-VMDC 3.0 does not monitor the routing protocols configured on a Nexus 3000.

Configuration

The Nexus 3000 is monitored via SNMPv3 or SNMPv2c. Figure 7-21 shows the list of components autodiscovered from a Nexus 3548 switch.

Figure 7-21. Resource Manager Components for the Nexus 3000

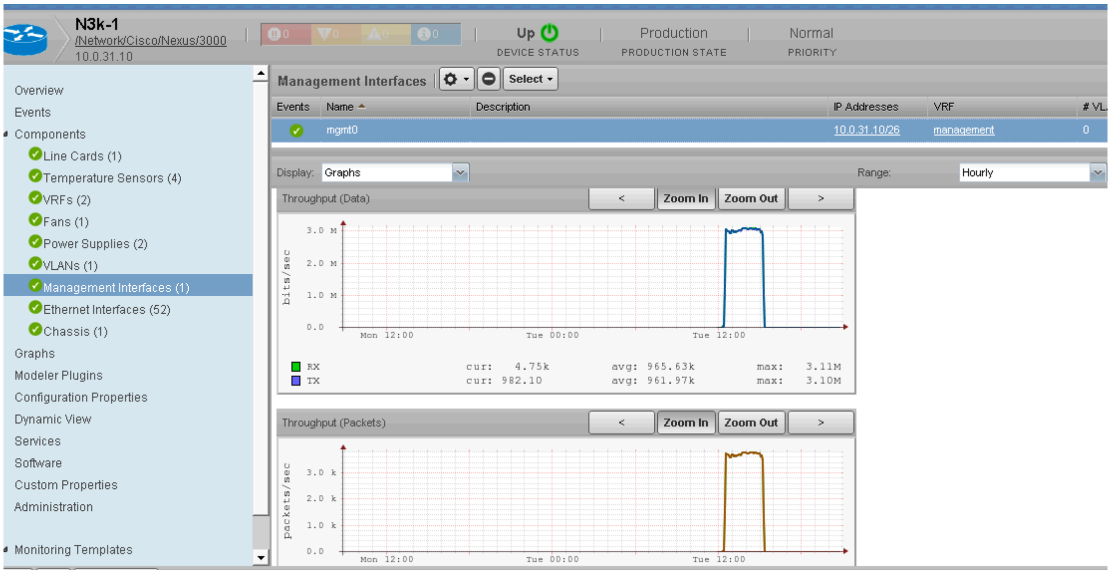


Nexus3k
/Network/Cisco/Nexus/3000
172.18.117.3

KPI Monitoring Component List

Zenoss CSA supports monitoring of KPI statistics for both the device and components. Figure 7-22 illustrates an example KPI. See Section A.1 Network Key Performance Indicators for more details on the supported KPI statistics for the Nexus 3000.

Figure 7-22. Resource Manager KPI Example for the Nexus 3000



Events	Name	Description	IP Addresses	VRF	# VL
✓	mgmt0		10.0.31.10/26	management	0

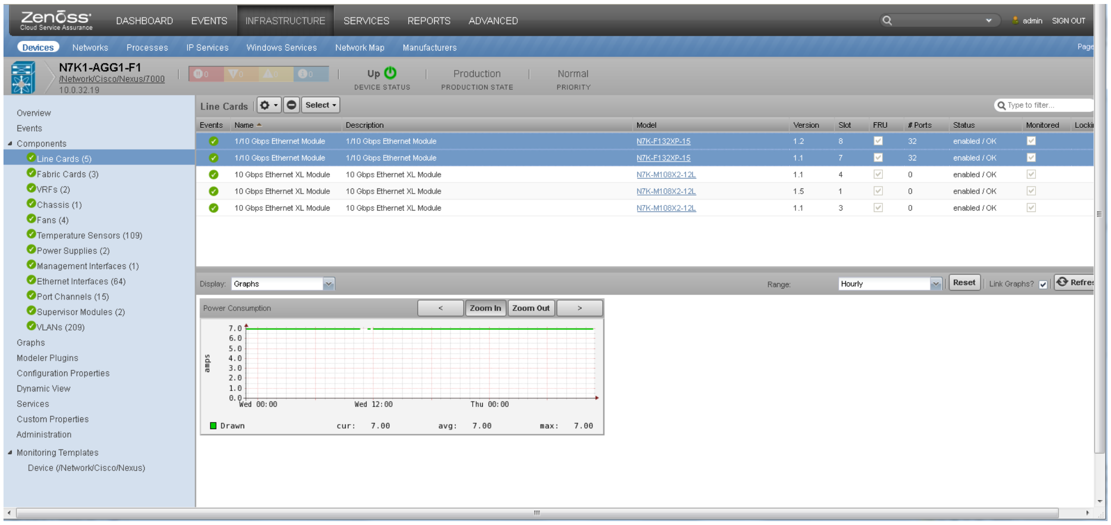
Display: Graphs
Range: Hourly

🟩 RX	cur:	4.75k	avg: 965.63k	max:	3.11M
🟦 TX	cur:	982.10	avg: 961.97k	max:	3.10M

7.4 FabricPath Line Card

The FabricPath Line Card (FPLC) is an interface line card for the Nexus product family. Zenoss CSA is expanding coverage of the Nexus 7000 to include the F1 FPLC. The FPLC is treated as a component of the Nexus platform and is autodiscovered along with all of the components previously supported in CLSA-VMDC 2.2. It does not require any special SNMP configuration at the Nexus 7000 device level. Figure 7-23 shows the F1 FPLC listed as a component of the Nexus 7000.

Figure 7-23. Line Card Discovery



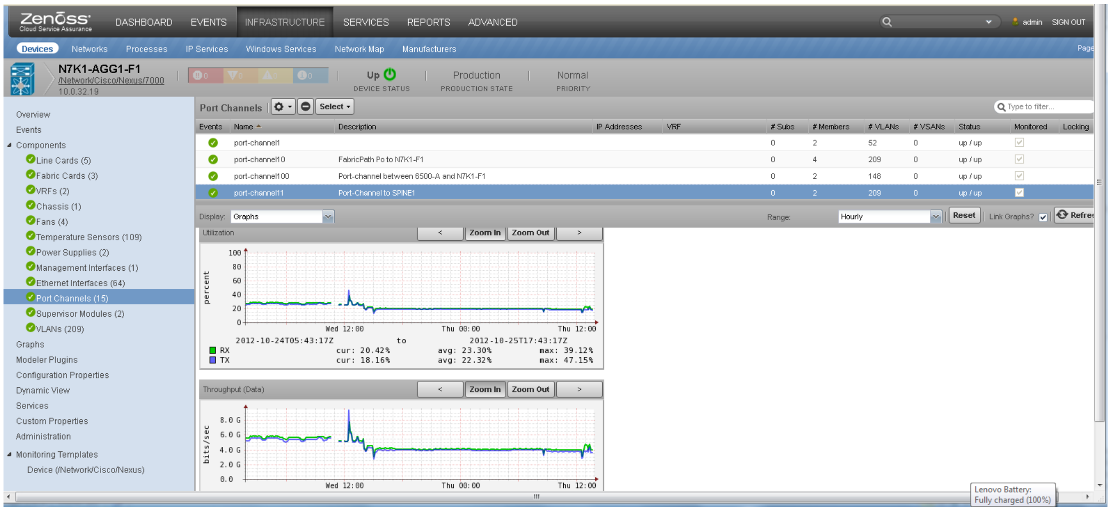
Events	Name	Description	Model	Version	Slot	FRU	# Ports	Status	Monitored
✓	1/10 Gbps Ethernet Module	1/10 Gbps Ethernet Module	N7K-F132XP-15	1.2	8	☑	32	enabled / OK	☑
✓	1/10 Gbps Ethernet Module	1/10 Gbps Ethernet Module	N7K-F132XP-15	1.1	7	☑	32	enabled / OK	☑
✓	10 Gbps Ethernet XL Module	10 Gbps Ethernet XL Module	N7K-M108X2-12L	1.1	4	☑	0	enabled / OK	☑
✓	10 Gbps Ethernet XL Module	10 Gbps Ethernet XL Module	N7K-M108X2-12L	1.5	1	☑	0	enabled / OK	☑
✓	10 Gbps Ethernet XL Module	10 Gbps Ethernet XL Module	N7K-M108X2-12L	1.1	3	☑	0	enabled / OK	☑

Display: Graphs
Range: Hourly
Link Graphs? ☑

🟩 Drawn cur: 7.00 avg: 7.00 max: 7.00

Standard KPI statistics are monitored by Zenoss CSA, similar to M1/M2 line cards on the Nexus 7000. Figure 7-24 and Figure 7-25 show sample KPI statistics monitored at the port-channel and interface levels. See Section A.1 Network Key Performance Indicators for more details on the supported line card KPI statistics for the FPLC.

Figure 7-24. Sample Port-channel KPI Statistics on Line Card 1



N7K1-AGG1-F1
/Network/Cisco/Nexus/7000
10.0.32.19

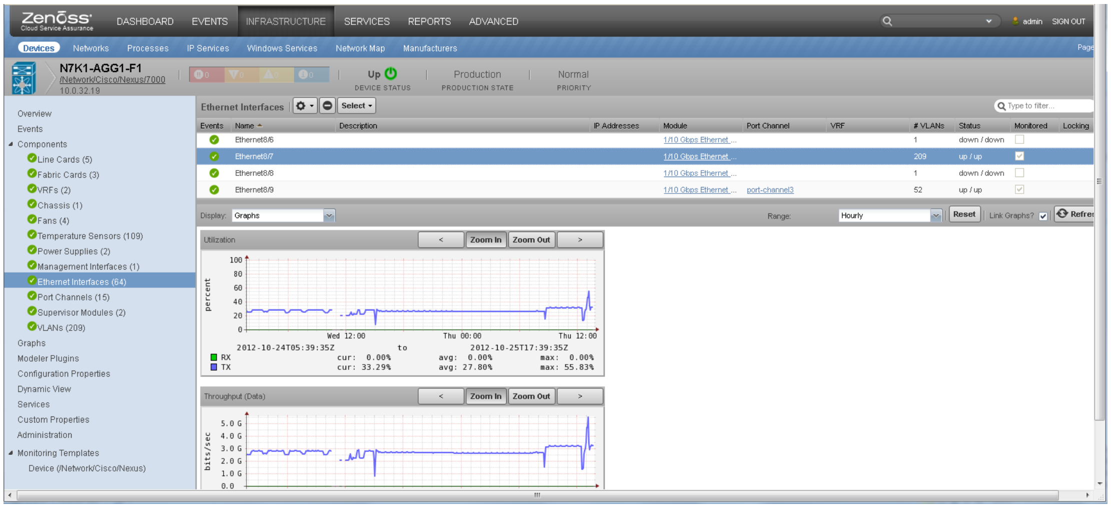
Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

Events	Name	Description	IP Addresses	VRF	# Subs	# Members	# VLANs	# VSANs	Status	Monitored	Locking
✓	port-channel1				0	2	52	0	up / up	☑	
✓	port-channel10	FabricPath Po to N7K1-F1			0	4	209	0	up / up	☑	
✓	port-channel100	Port-channel between 6500-A and N7K1-F1			0	2	148	0	up / up	☑	
✓	port-channel11	Port-Channel to SPINE1			0	2	209	0	up / up	☑	

Display: Graphs   Range: Hourly

```
2012-10-24T05:43:17Z          to          2012-10-25T17:43:17Z
RX              cur: 20.42%      avg: 23.30%      max: 39.12%
TX              cur: 18.16%      avg: 22.32%      max: 47.15%
```

Figure 7-25. Sample Interface KPI Statistics on Line Card 2



N7K1-AGG1-F1
/Network/Cisco/Nexus/7000
10.0.32.19

Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

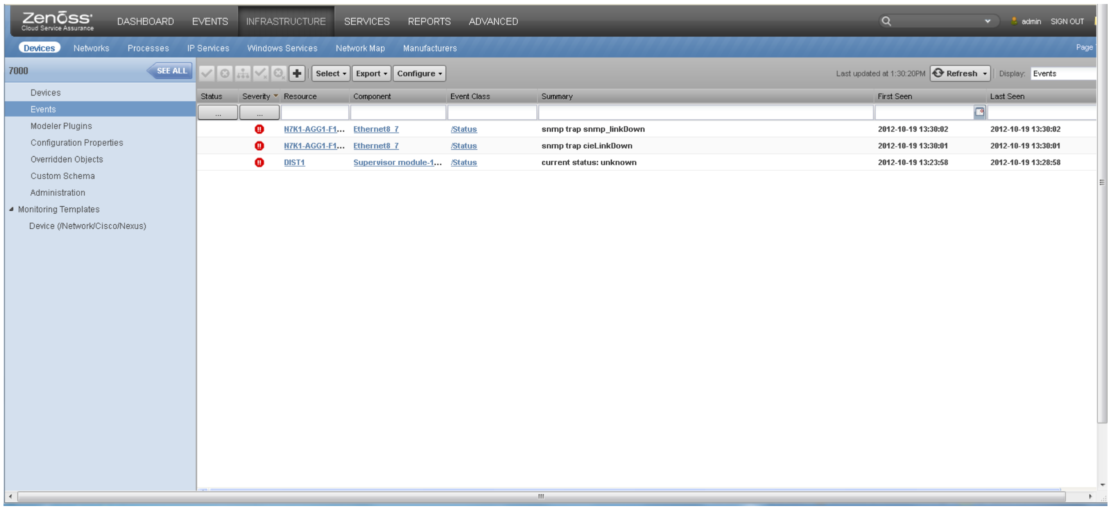
Events	Name	Description	IP Addresses	Module	Port Channel	VRF	# VLANs	Status	Monitored	Locking
✓	Ethernet8/6			1/10 Gbps Ethernet ...			1	down / down	□	
✓	Ethernet8/7			1/10 Gbps Ethernet ...			209	up / up	☑	
✓	Ethernet8/8			1/10 Gbps Ethernet ...			1	down / down	□	
✓	Ethernet8/9			1/10 Gbps Ethernet ...	port-channel3		52	up / up	☑	

Display: Graphs   Range: Hourly

```
2012-10-24T05:39:35Z          to          2012-10-25T17:39:35Z
RX              cur:  0.00%      avg:  0.00%      max:  0.00%
TX              cur: 33.29%      avg: 27.80%      max: 55.83%
```

All of the events at the line-card level are reported appropriately by Zenoss. Figure 7-26 shows the Nexus 7000 Events view where FPLC events would be located.

Figure 7-26. Events on the Line Card



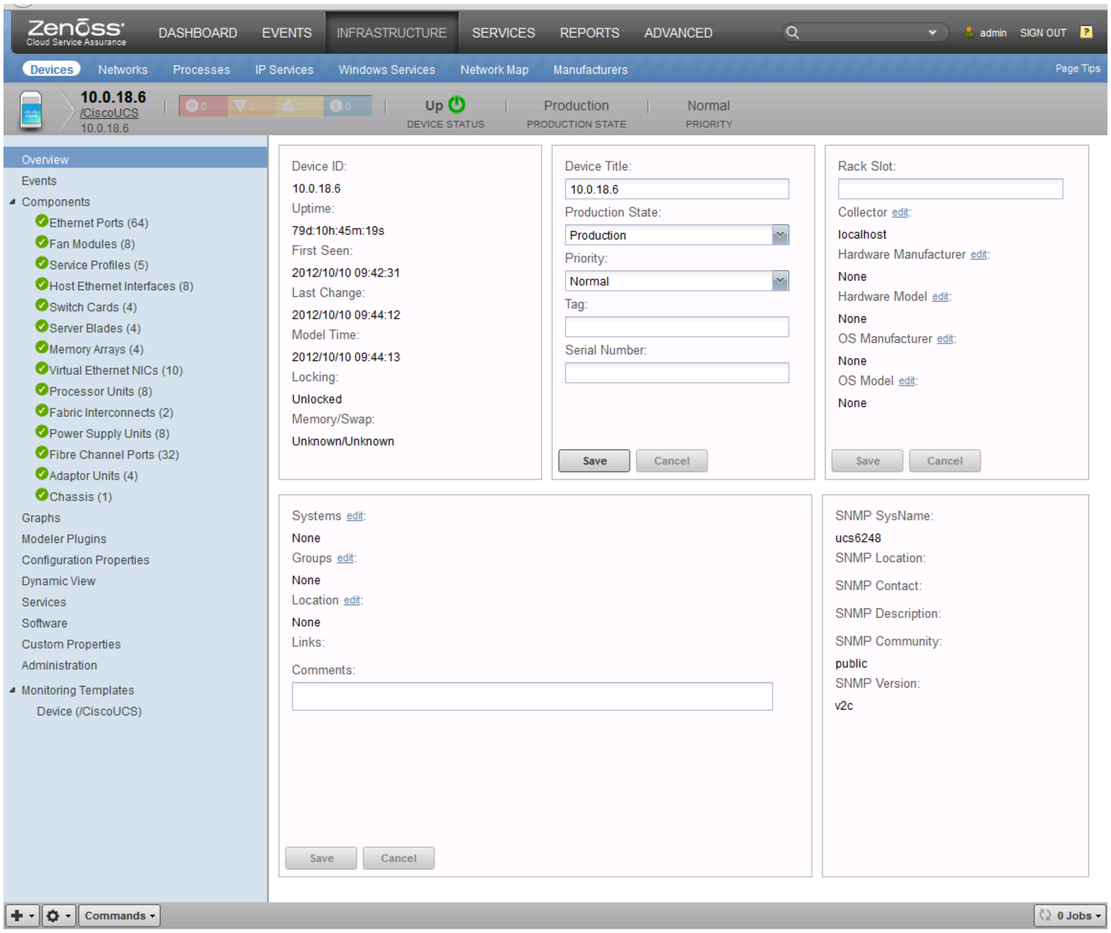
Status	Severity	Resource	Component	Event Class	Summary	First Seen	Last Seen
	●	N7K1-AGG1-F1...	Ethernet8_7	/Status	snmp trap snmp_linkDown	2012-10-19 13:30:02	2012-10-19 13:30:02
	●	N7K1-AGG1-F1...	Ethernet8_7	/Status	snmp trap cieLinkDown	2012-10-19 13:30:01	2012-10-19 13:30:01
	●	DIST1	Supervisor module-1...	/Status	current status: unknown	2012-10-19 13:23:58	2012-10-19 13:28:58

7.5 UCS 6200 Fabric Interconnect

UCS 6200 Discovery

CLSA-VMDC support for the UCS product family has been extended to cover 6200 Fabric Interconnect devices. These devices are correctly discovered and categorized using the UCSM VIP address. As seen in Figure 7-27, Zenoss CSA discovers all components that are supported on UCS 6100 devices.

Figure 7-27. UCS 6200 Discovery



10.0.18.6
/CiscoUCS
10.0.18.6

Up — DEVICE STATUS | Production — PRODUCTION STATE | Normal — PRIORITY

Components:
- Ethernet Ports (64)
- Fan Modules (8)
- Service Profiles (5)
- Host Ethernet Interfaces (8)
- Switch Cards (4)
- Server Blades (4)
- Memory Arrays (4)
- Virtual Ethernet NICs (10)
- Processor Units (8)
- Fabric Interconnects (2)
- Power Supply Units (8)
- Fibre Channel Ports (32)
- Adaptor Units (4)
- Chassis (1)

Device ID: 10.0.18.6
Uptime: 79d:10h:45m:19s
First Seen: 2012/10/10 09:42:31
Last Change: 2012/10/10 09:44:12
Model Time: 2012/10/10 09:44:13
Locking: Unlocked
Memory/Swap: Unknown/Unknown

Device Title: 10.0.18.6
Production State: Production
Priority: Normal
Tag:
Serial Number:

Rack Slot:
Collector edit: localhost
Hardware Manufacturer edit: None
Hardware Model edit: None
OS Manufacturer edit: None
OS Model edit: None

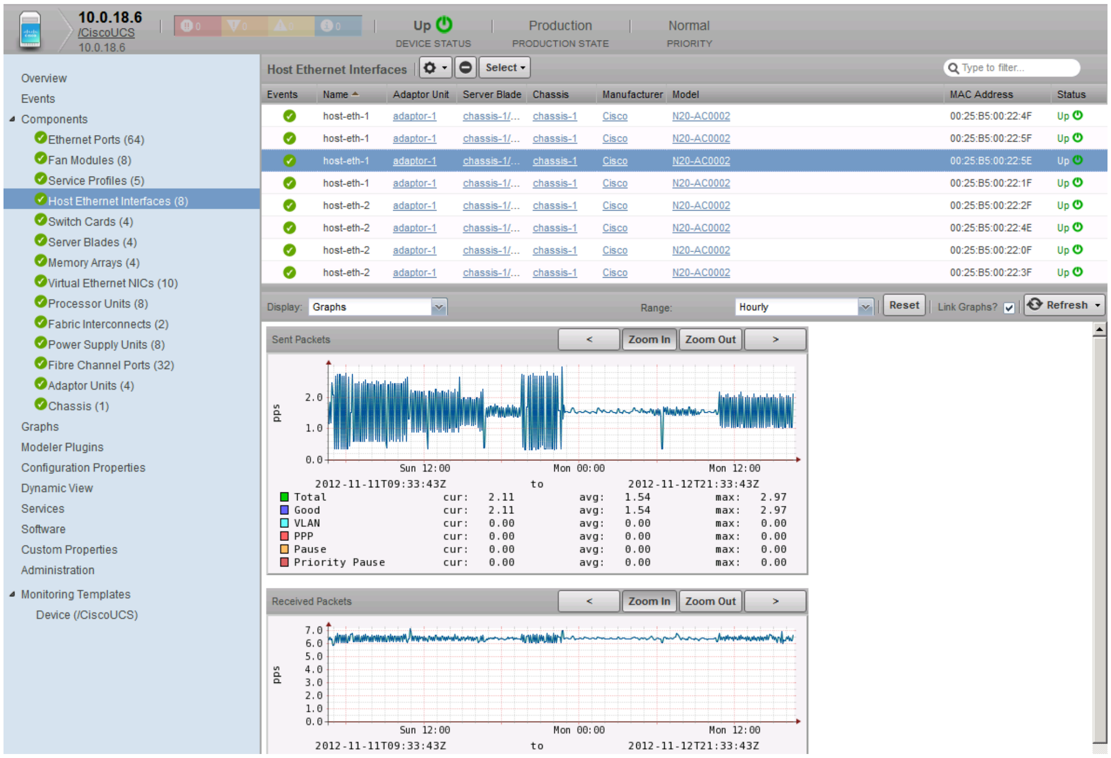
Systems edit: None
Groups edit: None
Location edit: None
Links:
Comments:

SNMP SysName: ucs6248
SNMP Location:
SNMP Contact:
SNMP Description:
SNMP Community: public
SNMP Version: v2c

UCS 6200 KPI Monitoring

Zenoss CSA provides KPI monitoring similar to the UCS 6100 devices. Figure 7-28 illustrates a sample KPI view for Ethernet interfaces.

Figure 7-28. UCS 6200 KPI Monitoring



Events	Name	Adaptor Unit	Server Blade	Chassis	Manufacturer	Model	MAC Address	Status
✓	host-eth-1	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:4F	Up
✓	host-eth-1	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:5F	Up
✓	host-eth-1	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:5E	Up
✓	host-eth-1	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:1F	Up
✓	host-eth-2	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:2F	Up
✓	host-eth-2	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:4E	Up
✓	host-eth-2	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:0F	Up
✓	host-eth-2	adaptor-1	chassis-1/...	chassis-1	Cisco	N20-AC0002	00:25:B5:00:22:3F	Up

Display: Graphs Range: Hourly Link Graphs? ☑

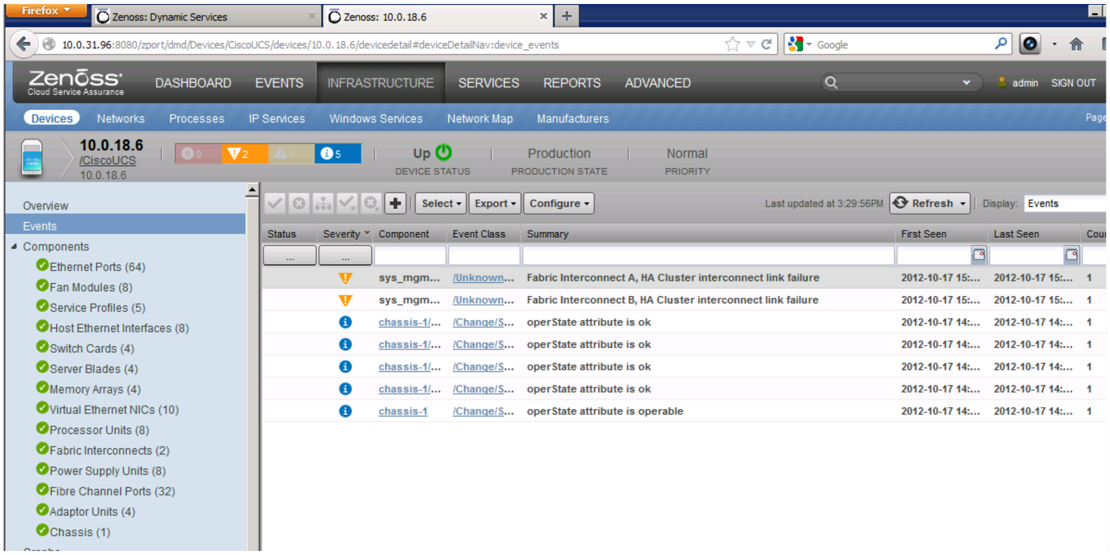
```
2012-11-11T09:33:43Z          to          2012-11-12T21:33:43Z
Total                cur:   2.11        avg:   1.54        max:   2.97
Good                 cur:   2.11        avg:   1.54        max:   2.97
VLAN                 cur:   0.00        avg:   0.00        max:   0.00
PPP                  cur:   0.00        avg:   0.00        max:   0.00
Pause                cur:   0.00        avg:   0.00        max:   0.00
Priority Pause       cur:   0.00        avg:   0.00        max:   0.00
```

```
2012-11-11T09:33:43Z          to          2012-11-12T21:33:43Z
```

UCS 6200 Event Monitoring

Zenoss CSA provides SNMP Trap event monitoring for the 6200. Fault events such as a UCS 6200 chassis failure and link failure between a pair of 6200s were captured by Zenoss. Figure 7-29 shows a sample event view with SNMP Trap generated events.

Figure 7-29. Sample SNMP Trap Event View



10.0.31.96:8080/zport/dmd/Devices/CiscoUCS/devices/10.0.18.6/devicedetail#deviceDetailNav:device_events

Last updated at 3:29:56PM Display: Events

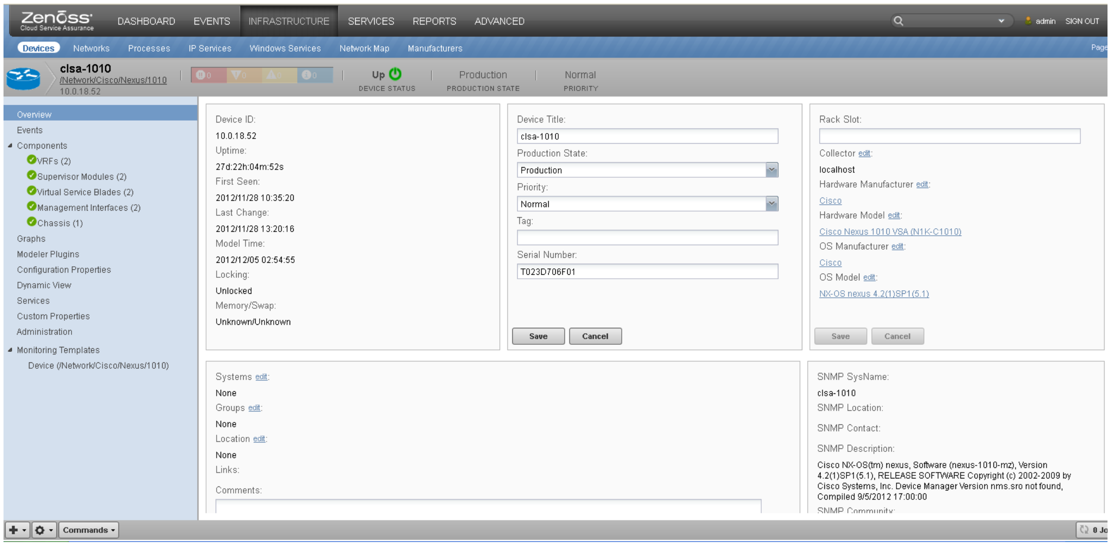
Status	Severity	Component	Event Class	Summary	First Seen	Last Seen	Cou
	!	sys_mgm...	/Unknown...	Fabric Interconnect A, HA Cluster interconnect link failure	2012-10-17 15:...	2012-10-17 15:...	1
	!	sys_mgm...	/Unknown...	Fabric Interconnect B, HA Cluster interconnect link failure	2012-10-17 15:...	2012-10-17 15:...	1
	i	chassis-1/...	/Change/S...	operState attribute is ok	2012-10-17 14:...	2012-10-17 14:...	1
	i	chassis-1/...	/Change/S...	operState attribute is ok	2012-10-17 14:...	2012-10-17 14:...	1
	i	chassis-1/...	/Change/S...	operState attribute is ok	2012-10-17 14:...	2012-10-17 14:...	1
	i	chassis-1/...	/Change/S...	operState attribute is ok	2012-10-17 14:...	2012-10-17 14:...	1
	i	chassis-1	/Change/S...	operState attribute is operable	2012-10-17 14:...	2012-10-17 14:...	1

7.6 Nexus 1010

Zenoss can monitor the Nexus 1010 using SNMP versions v1, v2c, and v3. SNMP configuration on the Nexus 1010 is similar to configuration on other Nexus devices.

Zenoss CSA autodiscovers the components shown in Figure 7-30.

Figure 7-30. Components on the Nexus 1010



clsa-1010
/Network/Cisco/Nexus/1010
10.0.18.52

Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

Device ID:
10.0.18.52
Uptime:
27d:22h:04m:52s
First Seen:
2012/11/28 10:35:20
Last Change:
2012/11/28 13:20:16
Model Time:
2012/12/05 02:54:55
Locking:
Unlocked
Memory/Swap:
Unknown/Unknown

Device Title:
clsa-1010
Production State:
Production
Priority:
Normal
Tag:

Serial Number:
T023D706F01

Rack Slot:

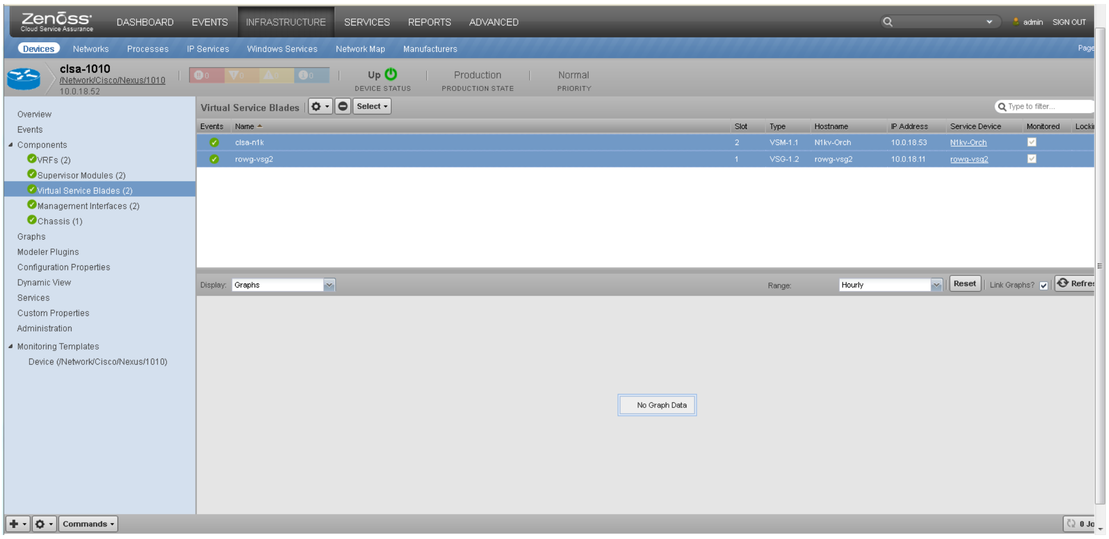
Collector edit:
localhost
Hardware Manufacturer edit:
Cisco
Hardware Model edit:
Cisco Nexus 1010 VSA (N1K-C1010)
OS Manufacturer edit:
Cisco
OS Model edit:
NX-OS nexus 4.2(1)SP1(5.1)

Systems edit:
None
Groups edit:
None
Location edit:
None
Links:
Comments:

SNMP SysName:
clsa-1010
SNMP Location:
SNMP Contact:
SNMP Description:
Cisco NX-OS(tm) nexus, Software (nexus-1010-mz), Version 4.2(1)SP1(5.1), RELEASE SOFTWARE Copyright (c) 2002-2009 by Cisco Systems, Inc. Device Manager Version nms.sro not found, Compiled 9/5/2012 17:00:00
SNMP Community:

If the VSG or Nexus 1000V VSM has been installed on the Nexus 1010, they are exposed under the virtual service blade component, as shown in Figure 7-31.

Figure 7-31. Virtual Service Blades Discovered on the Nexus 1010



clsa-1010
/Network/Cisco/Nexus/1010
10.0.18.52

Up DEVICE STATUS | Production PRODUCTION STATE | Normal PRIORITY

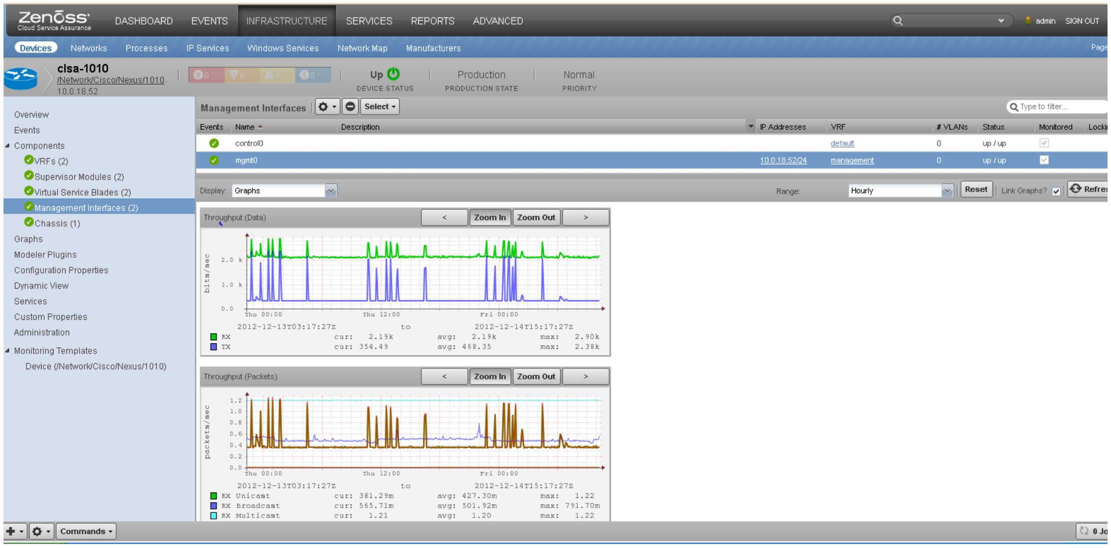
Virtual Service Blades

Events	Name	Slot	Type	Hostname	IP Address	Service Device	Monitored	Locki
✓	clsa-n1k	2	VSM-1.1	N1kv-Orch	10.0.18.53	N1kv-Orch	☑	
✓	rowg-vsg2	1	VSG-1.2	rowg-vsg2	10.0.18.11	rowg-vsg2	☑	

Display: Graphs
Range: Hourly

Zenoss CSA supports Nexus 1010 monitoring of KPI statistics, as shown in Figure 7-32.

Figure 7-32. Key Performance Indicator Statistics on the Nexus 1010



Events	Name	Description	IP Addresses	VRF	# VLANs	Status	Monitored	Locki
✓	control0			default	0	up / up	☑	
✓	mgmt0		10.0.18.52/24	management	0	up / up	☑	

Display: Graphs

Range: Hourly

Link Graphs? ☑

```
2012-12-13T03:17:27Z            to            2012-12-14T15:17:27Z
🟩 RX                  cur:   2.19k     avg:   2.19k     max:   2.90k
🟦 TX                  cur: 354.49      avg: 468.35      max:   2.38k
```

```
2012-12-13T03:17:27Z            to            2012-12-14T15:17:27Z
🟩 RX Unicast          cur: 381.29m     avg: 427.30m     max:   1.22
🟦 RX Broadcast        cur: 565.71m     avg: 501.92m     max: 791.70m
🟦 RX Multicast        cur:   1.21      avg:   1.20      max:   1.22
```

7.7 Zenoss Sample Tenant Portal

This section provides a brief overview of the Zenoss Sample Tenant Portal, which can be leveraged by cloud Service Providers and Enterprises to develop production-ready portals for their customers. For more detailed information about the sample portal, refer to Zenoss Portal Documentation Release 1.0.0.

7.7.1 Sample Tenant Portal

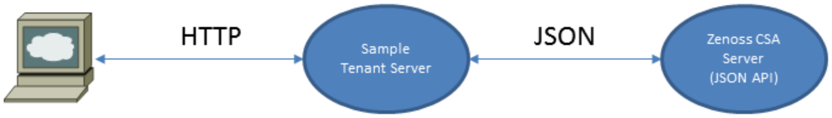
Zenoss CSA is typically installed with cloud administrative privileges in the Service Provider's management Network Operations Center (NOC). Customers or tenants accessing the resources of that cloud should be provided with an interface that allows a customer-focused view into that cloud. This customer-focused view would provide information about a customer's usage of cloud resources and their status. Cloud providers will want to separate their administrative and operational views of Zenoss CSA from their customers' read-only views. With this in mind, the Zenoss CSA 4.2.3 release includes a sample tenant portal based on the Zenoss JavaScript Object Notation (JSON) Application Programming Interface (API).

The sample portal provides an example of how a tenant portal might be constructed. The portal consists of a client-side web browser that requests information from a proxy server. The proxy server communicates with a JSON API included in the Zenoss CSA server, which provides responses back to the client.

Note The Zenoss sample tenant portal is not a fully functioning customer portal. It is provided as an example for SPs who wish to incorporate the Zenoss API into their existing tool set or who wish to develop a customer portal separately.

The sample tenant portal architecture is shown in Figure 7-33. A client-side browser must have IP connectivity to the sample portal server. For simplicity, the portal server can be installed on the same host as the Zenoss CSA server. However, if security is a concern, a separate host with IP connectivity to the Zenoss server can be used.

Figure 7-33. Sample Tenant Portal Architecture



7.7.2 Sample Tenant Portal Installation

Sample portal installation is relatively straightforward. The portal is implemented in two parts that includes a portal ZenPack and a standalone portal server. The portal ZenPack must be installed on the Zenoss CSA server, and the portal server is installed separately. First, ensure that the Zenoss CSA server has been updated with the two ZenPacks listed in Table 7-3.

Table 7-3. Sample Tenant Portal ZenPack Requirements

ZenPack	Version
Impact	ZenPacks.zenoss.Impact
PortalIntegration	ZenPacks.zenoss.PortalIntegration

Next, set up a Linux host with the requirements shown in Table 7-4. The portal server can be colocated with the Zenoss CSA server, or a second Linux host can be set up that is dedicated to the portal server.

Table 7-4. Sample Portal Installation Requirements

Requirement	Version
Operating System	Red Hat (CentOS) 5 or 6 (or any Linux with Python 2.6)
Python Language	Python 2.6
Virtualenv	Extra Packages for Enterprise Linux (EPEL) Repository

The following steps outline the portal installation procedure:

1. Install EPEL, Python 2.6, and virtualEnv into a Linux platform.
2. Extract the Zenoss portal from the Zenoss Sample Portal tar archive.
3. Activate a virtual Python environment.
4. Install the zenoss-portal in the virtual Python environment using develop mode.

Once the zenoss-portal is installed, a few configuration steps must be performed. On the portal host, configure the Zenoss CSA server address and credentials as environment variables. These may also be added to shell initialization scripts so they are persistent.

```
export ZENOSS_URL=http://zenoss.yourdomain:8080
export ZENOSS_USERNAME="<api_username>"
export ZENOSS_PASSWORD="<api_password>"
```

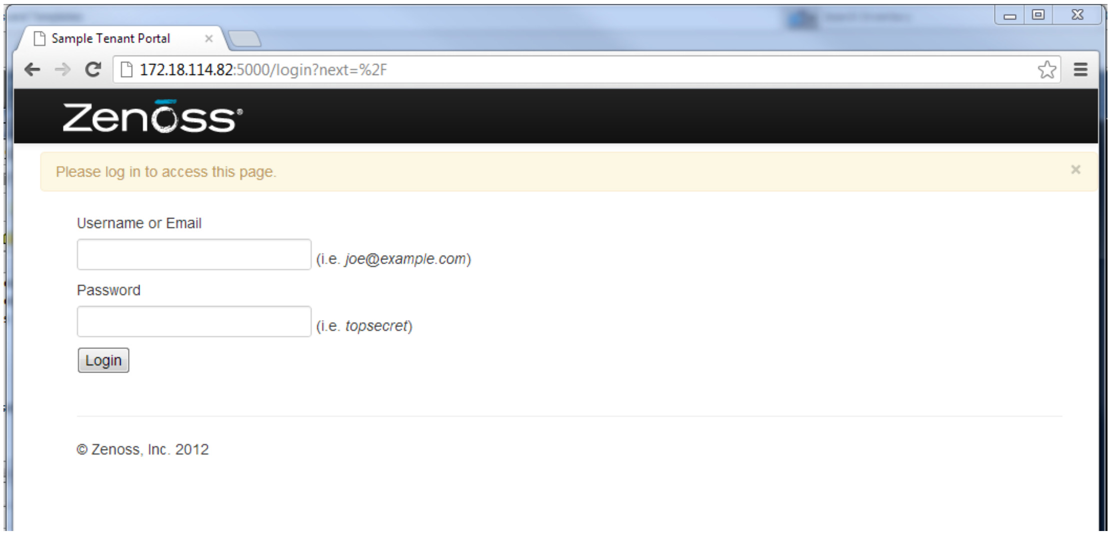
Finally, on the Zenoss CSA server, create a service organizer and a service for the "ACME" customer using the following steps:

1. Navigate to the Services menu in the Zenoss web interface.
2. Add a top-level service organizer named Customers.
3. Add a secondary service organizer underneath Customers named ACME.
4. Add one or more services into the Customers/ACME organizer.
5. Add resources such as VMs, VLANs, etc., into the /Customers/ACME services.

The portal comes with a pre-configured single user, joe@example.com, that is mapped to a customer named ACME. The portal can be expanded to include other example customers and users. The portal retrieves service and resource information for joe@example.com by looking for Zenoss services configured under /Customers/ACME.

Launch the portal and browse to the portal server's URL. Once the portal is installed, a browser session to the portal URL looks similar to the login screen shown in Figure 7-34.

Figure 7-34. Zenoss Sample Portal Login Screen



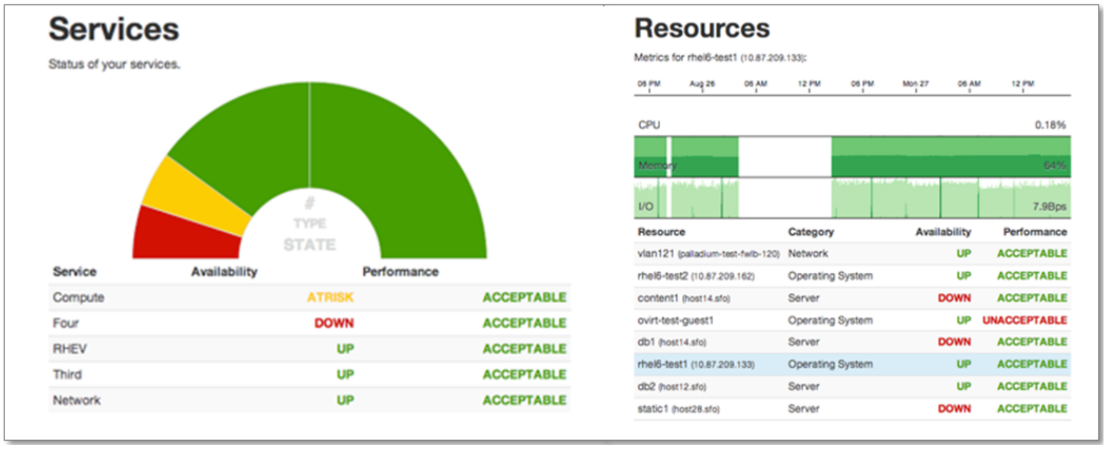
172.18.114.82:5000/login?next=%2F

Please log in to access this page.

© Zenoss, Inc. 2012

Logging in as **joe@example.com** with password **topsecret** presents a service and resources view for the ACME customer.

Figure 7-35. Zenoss Sample Portal Services and Resources Screen



Services

Status of your services.

Service	Availability	Performance
Compute	ATRISK	ACCEPTABLE
Four	DOWN	ACCEPTABLE
RHEV	UP	ACCEPTABLE
Third	UP	ACCEPTABLE
Network	UP	ACCEPTABLE

Resources

Metrics for rhel6-test1 (10.87.209.133):

Resource	Category	Availability	Performance
vlan121 (palladium-test-fwlb-120)	Network	UP	ACCEPTABLE
rhel6-test2 (10.87.209.162)	Operating System	UP	ACCEPTABLE
content1 (host14.sfo)	Server	DOWN	ACCEPTABLE
ovirt-test-guest1	Operating System	UP	UNACCEPTABLE
db1 (host14.sfo)	Server	DOWN	ACCEPTABLE
rhel6-test1 (10.87.209.133)	Operating System	UP	ACCEPTABLE
db2 (host12.sfo)	Server	UP	ACCEPTABLE
static1 (host28.sfo)	Server	DOWN	ACCEPTABLE

Note Refer to the Zenoss Sample Portal Documentation for more details regarding portal installation and configuration.

7.7.3 Zenoss JSON API

Zenoss CSA provides support for a JSON Python API that enables programmatic access to the Zenoss database. This section provides a brief overview of the API components and a few suggestions for API usage.

Zenoss API Overview

The Zenoss JSON API was introduced in Zenoss 3.0 to allow programmatic access to the Zenoss server and coincided with a redesign of the Zenoss user interface. For this reason, the API can be employed to perform scripted command and control functions on the Zenoss server, in addition to manual user operations. The API can be used for interconnecting the Zenoss server with other cloud Service Provider management tools or for providing customized views into the Zenoss database. Several potential API uses are listed below.

- **Automated orchestration system.** A cloud service orchestration system can be integrated with Zenoss to automatically add services or specific devices for discovery and monitoring.
- **Limited and focused operational views.** Special purpose views might be created to assist Service Provider operations personnel in daily monitoring tasks.
- **External management interface.** A providers' existing management tool set can be interfaced to control Zenoss and extract summary status information.
- **Custom tenant portal.** A limited scope, read-only view can be made available to a cloud Service Provider's customers.

The API is comprised of router object classes and methods for interacting with these classes. Table 7-5 lists the available routers.

Table 7-5. List of JSON API Router Object Classes

Router	Purpose
DetailNavRouter	Zenoss view navigation
DeviceRouter	Retrieving information about and interacting with devices discovered by Zenoss
EventsRouter	Used to manage/view events
ImpactRouter	Working with service impact
JobsRouter	Working with Zenoss jobs
MessagingRouter	Clearing and viewing browser user informational messages
MibRouter	Adding, deleting, retrieving MIB information
NetworkRouter	Working with devices related by specific network
Network6Router	Working with v6 devices related by specific network
PortalRouter	Working with the Sample Portal
ProcessRouter	Working with Zenoss processes
RelatedEventsRouter	Working with Zenoss events
ReportRouter	Working with Zenoss reports
ServiceRouter	Working with IP and Windows service definitions
SettingsRouter	Getting and setting user interface parameters
TemplateRouter	Working with Zenoss KPI graphs
TriggersRouter	Working with router based triggers
ZenPackRouter	Add objects to ZenPacks

Details about the API are documented at the following URL:

http://community.zenoss.org/community/documentation/official_documentation/api

API Usage Suggestions

Several of the routers have methods that would be more suited to an administrative or operational interface with a Zenoss CSA server. These routers have methods such as addDevice, getDeviceClasses, getEvents, etc. The router classes below have the following methods that might be useful for a customer tenant portal.

DeviceRouter

These three methods provide a list of devices organized by group, location, or system. A group can be used to collect devices of similar type or purpose together. A location can be used to collect devices in geographic proximity. A system can be used to collect devices all belonging to a logical system such as a data center. These methods would need filtering by customer authentication, to restrict a customer view into only their groups, locations, or systems. They may be useful to provide device status information for a customer, but it may be preferable to only provide a logical view of a customer's services and not the specific equipment which may be shared in a public cloud.

- getGroups - get a list of all groups
- getLocations - get a list of all locations
- getSystems - get a list of all systems

ImpactRouter

Much of this router class will have administrative methods not likely useful to a CLSA customer. The following methods may be helpful:

- getDependencies - gets every item currently impacting a dynamic service
- getInfo - retrieve information about a dynamic service
- getInfogetInfoFromGuid - get information about a dynamic service
- getInstances - get a list of dynamic services
- getServices - retrieves info objects for services
- getTree - returns dynamic service tree

RelatedEventsRouter

There are only two methods in this class, but they may be useful when using the customer tenant portal:

- Query - queries for events related to a service event
- Related_services - find services that a given event impacted

PortalRouter

These methods work with the sample portal.

APPENDIX A

Key Performance Indicators for VMDC

This appendix provides a listing of the Key Performance Indicator (KPI) statistics for Virtualized Multiservice Data Center (VMDC) components and subcomponents.

A.1 Network Key Performance Indicators

Table A-1 lists the network KPI statistics for CLSA-VMDC.

Table A-1. Network KPI Statistics for CLSA-VMDC

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
Nexus 7000/5000, 3000/2000	Environmental	Power Consumption	SNMP, NETCONF	SNMP, SSH
		Temperature	SNMP, NETCONF	SNMP, SSH
		Current	SNMP, NETCONF	SNMP, SSH
	Device Performance	CPU Utilization	SNMP, NETCONF	SNMP, SSH
		Memory Utilization	SNMP, NETCONF	SNMP, SSH
	Bandwidth Usage	Utilization	SNMP, NETCONF	SNMP, SSH
		Throughput	SNMP, NETCONF	SNMP, SSH
	Interface Errors	Discards RX/TX	SNMP, NETCONF	SNMP, SSH
		Errors RX/TX	SNMP, NETCONF	SNMP, SSH
Nexus 1000V	Environmental	Power Consumption (only if Nexus	SNMP, NETCONF	SNMP, SSH

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		1000V is on Nexus 1010)		
	Device Performance	CPU Utilization	SNMP, NETCONF	SNMP, SSH
		Memory Utilization	SNMP, NETCONF	SNMP, SSH
	Bandwidth Usage	Utilization	SNMP, NETCONF	SNMP, SSH
		Throughput	SNMP, NETCONF	SNMP, SSH
	Interface Errors	Discards RX/TX	SNMP, NETCONF	SNMP, SSH
		Errors RX/TX	SNMP, NETCONF	SNMP, SSH
ASR 9000	Environmental	Temperature	SNMP	SNMP
	Device Performance	CPU Utilization	SNMP	SNMP
		Memory Utilization	SNMP	SNMP
	Bandwidth Usage	Utilization	SNMP	SNMP
		Throughput	SNMP	SNMP
	Interface Errors	Discards RX/TX	SNMP	SNMP
		Errors RX/TX	SNMP	SNMP
ASR 1000	Environmental	Power Consumption	SNMP	SNMP
		Temperature	SNMP	SNMP
		Current	SNMP	SNMP
	Device Performance	CPU Utilization	SNMP	SNMP
		Memory Utilization	SNMP	SNMP
	Bandwidth Usage	Utilization	SNMP	SNMP
		Throughput	SNMP	SNMP
	Interface Errors	Discards RX/TX	SNMP	SNMP
		Errors RX/TX	SNMP	SNMP

A.2 Services Key Performance Indicators

Table A-2 lists the services KPI statistics for VMDC.

Table A-2. Services KPI Statistics for VMDC

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
Catalyst 6500 VSS	Environmental	Power Consumption	SNMP	SNMP
		Temperature	SNMP	SNMP
		Cooling Capacity	SNMP	SNMP
		Current	SNMP	SNMP
		CPU Utilization	SNMP	SNMP
		Memory Utilization	SNMP	SNMP
		Operational Port Count	SNMP	SNMP
	Bandwidth Usage	Utilization	SNMP	SNMP
		Throughput	SNMP	SNMP
	Interface Errors	Discards RX/TX	SNMP	SNMP
		Errors RX/TX	SNMP	SNMP
ACE Module	Device Performance	CPU Utilization	SNMP	SNMP
		ACE Context-Connection Rate	SNMP	SNMP
		ACE Context-SSL Connection Rate	SNMP	SNMP
		ACE Context-Inspection Rate	SNMP	SNMP
		ACE Context-MAC Miss Rate	SNMP	SNMP
		ACE Context-Bandwidth	SNMP	SNMP
		ACE Context-Concurrent Connections	SNMP	SNMP
		ACE Context-Proxy Connections	SNMP	SNMP
		ACE Context-Sticky Connections	SNMP	SNMP

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		ACE Context-Translated Connections	SNMP	SNMP
		Concurrent Connections (Virtual Context only)	SNMP	SNMP
		Connection Rate (Virtual Context only)	SNMP	SNMP
		Drop Rate (Virtual Context only)	SNMP	SNMP
		Hit Rate (Virtual Context only)	SNMP	SNMP
		Real Servers (Virtual Context only)	SNMP	SNMP
	Bandwidth Usage	Throughput	SNMP	SNMP
FWSM	Device Performance	CPU Utilization	SNMP	SNMP
		Memory Utilization	SNMP	SNMP
		Security Context-Concurrent Connections (Admin Context only)	SNMP	SNMP
		Security Context-Concurrent Translations (Admin Context only)	SNMP	SNMP
		Security Context-Translation Rate (Admin Context only)	SNMP	SNMP
		Concurrent Connections	SNMP	SNMP
		Hosts	SNMP	SNMP
		IPsec Sessions	SNMP	SNMP
		ASDM sessions	SNMP	SNMP
		SSH sessions	SNMP	SNMP

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		TELNET sessions	SNMP	SNMP
		MAC Addresses	SNMP	SNMP
		NAT Translations	SNMP	SNMP
	Bandwidth Usage	Throughput	SNMP	SNMP
ASA 5585	Device Performance	CPU Utilization	SNMP	SNMP
		Memory Utilization	SNMP	SNMP
		Connections (Admin Context only)	SNMP	SNMP
		Connection Rates (Admin Context only)	SNMP	SNMP
		URL Access Request Rates (Admin Context only)	SNMP	SNMP
		Concurrent Connections	SNMP	SNMP
	Bandwidth Usage	Throughput	SNMP	SNMP
		Utilization (only for management interfaces)	SNMP	SNMP
	Interface Errors	Discards RX/TX	SNMP	SNMP
		Errors RX/TX	SNMP	SNMP
ASASM	Device Performance	CPU Utilization	SNMP	SNMP
		Memory Utilization	SNMP	SNMP
		Connections (Admin Context only)	SNMP	SNMP
		Connection Rates (Admin Context only)	SNMP	SNMP
		URL Access Request Rates (Admin Context only)	SNMP	SNMP

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		Concurrent Connections	SNMP	SNMP
	Bandwidth Usage	Throughput	SNMP	SNMP
VSG			SNMP, NETCONF	SNMP,SSH

A.3 Storage Key Performance Indicators

Table A-3 lists the storage related KPI statistics for VMDC.

Table A-3. Storage Related KPI Statistics for VMDC

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
MDS 9000 and Nexus 5000 SAN Ports	Environmental	Power Consumption	SNMP, NETCONF	SNMP,SSH
		Temperature	SNMP, NETCONF	SNMP,SSH
		Current	SNMP, NETCONF	SNMP,SSH
	Device Performance	Supervisor CPU Utilization	SNMP, NETCONF	SNMP,SSH
		Supervisor Memory Utilization	SNMP, NETCONF	SNMP,SSH
	Bandwidth Usage	Physical Interface bandwidth utilization	SNMP, NETCONF	SNMP,SSH
		Throughput	SNMP, NETCONF	SNMP,SSH
	Interface Errors	Errors which include FBSY, FRJT	SNMP, NETCONF	SNMP,SSH
NetApp	File System	LUN utilization	SSH, SNMP	SSH, SNMP
EMC VNX	Storage Processors	Time Utilization	SMI-S Provider	SMI-S API
		Cache Dirtiness	SMI-S Provider	SMI-S API
		Flush Rate	SMI-S Provider	SMI-S API
		Flush Throughput	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		Data Throughput	SMI-S Provider	SMI-S API
	Arrays	Cache Efficiency	SMI-S Provider	SMI-S API
		Cache Dirtiness	SMI-S Provider	SMI-S API
		Flush Rate	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API
		Data Throughput	SMI-S Provider	SMI-S API
		Input Power	SMI-S Provider	SMI-S API
	Data Devices	Time Utilization	SMI-S Provider	SMI-S API
		Queueing	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API
		Data Throughput	SMI-S Provider	SMI-S API
	Hard Disks	Space Utilization	SMI-S Provider	SMI-S API
		Time Utilization	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API
		Data Throughput	SMI-S Provider	SMI-S API
	Storage Pools	Space Utilization	SMI-S Provider	SMI-S API
		Managed Space Usage	SMI-S Provider	SMI-S API
		Raw Space Usage	SMI-S Provider	SMI-S API
	Storace Enclosures	Input Power	SMI-S Provider	SMI-S API
		Inlet Temperature	SMI-S Provider	SMI-S API
EMC VMAX	Hard Disks	Space Utilization	SMI-S Provider	SMI-S API
		Time Utilization	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API
		Data Throughput	SMI-S Provider	SMI-S API
	Arrays	Space Utilization	SMI-S Provider	SMI-S API
		Managed Space Usage	SMI-S Provider	SMI-S API
		Raw Space Usage	SMI-S Provider	SMI-S API
		Cache Efficiency	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		Data Throughput	SMI-S Provider	SMI-S API
	Storage Processors	Time Utilization	SMI-S Provider	SMI-S API
		Cache Dirtiness	SMI-S Provider	SMI-S API
		Flush Rate	SMI-S Provider	SMI-S API
		Flush Throughput	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API
		Data Throughput	SMI-S Provider	SMI-S API
	Data Devices	Time Utilization	SMI-S Provider	SMI-S API
		Queueing	SMI-S Provider	SMI-S API
		Operation Throughput	SMI-S Provider	SMI-S API
		Data Throughput	SMI-S Provider	SMI-S API
	Storage Pools	Space Utilization	SMI-S Provider	SMI-S API
		Managed Space Usage	SMI-S Provider	SMI-S API
		Raw Space Usage	SMI-S Provider	SMI-S API

A.4 Compute Key Performance Indicators

Table A-4 lists the compute KPI statistics for VMDC.

Table A-4. Compute KPI Statistics for VMDC

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
UCS B-series	Chassis	Power Utilization	UCSM	XML API
		Temperature	UCSM	XML API
		Voltages	UCSM	XML API
		Current	UCSM	XML API
		Fan Speed	UCSM	XML API
	Fabric Interconnect	CPU Utilization	UCSM	XML API
		Memory Utilization	UCSM	XML API
		Ethernet Port Throughput	UCSM	XML API

VMDC System Component	KPI Category	KPI	Monitored by	Instrumentation
		Ethernet Port Send Packets	UCSM	XML API
		Ethernet Port Receive Packets	UCSM	XML API
		Ethernet Port Loss stats	UCSM	XML API
		Ethernet Port Pause stats	UCSM	XML API
		Ethernet Port errors RX/TX Deferred,Out Discard, Under size,Align,FCS,Int Mac RX/TX	UCSM	XML API
		Ethernet Port Send Packets	UCSM	XML API
		FC Port Throughput	UCSM	XML API
		FC Port errors RX/TX,Discard RX/TX,Too Long Rx,Too short RX,CRC RX,Link Failure,Signal Losses,Sync Losses	UCSM	XML API
VMware	Virtual Machine Performance	CPU utilization	vCenter	vSphere API
		Memory utilization	vCenter	vSphere API
		Disk utilization	vCenter	vSphere API
	ESXi Host Performance	CPU utilization	vCenter	vSphere API
		Memory utilization	vCenter	vSphere API
		Disk utilization	vCenter	vSphere API

GLOSSARY

A

ACE	Application Control Engine (Cisco)
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ASA	Adaptive Security Appliance (Cisco)
ASR	Aggregation Services Router (Cisco)
ASASM	Adaptive Security Appliance Services Module (Cisco)

B

BSS	Business Support System

C

CIAC	Cisco Intelligent Automation for Cloud
CLM	Cloud Lifecycle Management (BMC)
CLSA	Cloud Service Assurance (Cisco)
CLSA-HCS	Cloud Service Assurance for HCS
CLSA-VMDC	Cloud Service Assurance for VMDC
CNSM	Cisco Network Services Manager
CRM	Customer Relationship Management
CSA	Cloud Service Assurance (Zenoss)
CUCM	Cisco Unified Communications Manager

D

DDTS	Distributed Defect Tracking System
DIG	Design and Implementation Guide
DNS	Domain Name System
DR	Disaster Recovery
DRBD	Distributed Replicated Block Device
DSN	Data Center Services Node

E

ECC	EMC Control Center (EMC)
ECOM	EMC Common Object Module (EMC)

F

FC	Fibre Channel
FCoE	Fibre Channel over Ethernet
FLOGI	Fabric Login
FPLC	FabricPath Line Card
FT	Fault-Tolerant
FTP	File Transfer Protocol
FWSM	Firewall Services Module

H

HA	High Availability
HCS	Hosted Collaboration Solution (Cisco)
HSRP	Hot Standby Router Protocol

HT	Hyperthreading
HTTP	Hypertext Transfer Protocol

I

IaaS	Infrastructure as a Service
ICMP	Internet Control Message Protocol
ICS	Integrated Compute Stack
IOM	Input-Output Module
IRC	Internet Relay Chat
IS-IS	Intermediate System to Intermediate System
iSCSI	Internet Small Computer System Interface

J

JAR	Java Archive
JMX	Java Management Extensions
JSON	JavaScript Object Notation

K

KPI	Key Performance Indicator

L

L2	Layer 2
L3	Layer 3
L4	Layer 4
L7	Layer 7
LCMC	Linux Cluster Management Console

LDAP	Lightweight Directory Access Protocol
LUN	Logical Unit Number

M

MDS	Multilayer Director Switch
MIB	Management Information Base
MoM	Manager-of-Managers
MTTF	Mean Time to Fix
MTTI	Mean Time to Identify
MTTR	Mean Time to Repair

N

NAT	Network Address Translation
NBI	Northbound Interface
NFS	Network File System
NGN	Next Generation Network
NNTP	Network News Transfer Protocol
NOC	Network Operations Center
NTP	Network Time Protocol

O

OID	Object Identifier
OS	Operating System
OSS	Operations Support System

P

PoC	Proof of Concept
PoD	Point of Delivery. The PoD is a basic infrastructure module that is a logical repeatable construct with predictable infrastructure characteristics and deterministic functions. A PoD identifies a modular unit of data center components and enables customers to add network, compute, and storage resources incrementally.
POP	Post Office Protocol
PSU	Power Supply Unit

Q

QoS	Quality of Service

R

RAID	Redundant Array of Independent Disks
RBAC	Role-Based Access Control
RCA	Root Cause Analysis
RPC	Remote Procedure Call
RPM	RPM Package Manager (Red Hat)

S

SaaS	Software as a Service
SAM	Service Assurance Manager
SDU	Systems Development Unit (Cisco)
SIA	Service Impact Analysis
SLA	Service Level Agreement
SLB	Server Load Balancing
SMTP	Simple Mail Transfer Protocol
SMI-S	Storage Management Initiative Specification

SNIA	Storage Networking Industry Association
SNMP	Simple Network Management Protocol
SNMPv1	Simple Network Management Protocol version 1
SNMPv2c	Simple Network Management Protocol version 2c
SNMPv3	Simple Network Management Protocol version 3
SOAP	Simple Object Access Protocol
SP	Service Provider
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
STONITH	Shoot The Other Node In The Head
SVI	Switched Virtual Interface

T

TACACS	Terminal Access Controller Access Control System
TCA	Threshold Crossing Alert
TCP	Transmission Control Protocol

U

UCaaS	Unified Communications as a Service
UCS	Unified Computing System (Cisco)
UCSM	Unified Computing System Manager (Cisco)
UI	User Interface

V

vCPU	Virtual CPU (VMware)

VEM	Virtual Ethernet Module
VIP	Virtual IP
VM	Virtual Machine
VMDC	Virtualized Multiservice Data Center (Cisco)
vPC	Virtual Port-Channel
VNMC	Virtual Network Management Center
VPDC	Virtual Private Data Center
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VSG	Virtual Security Gateway (Cisco)
VSM	Virtual Supervisor Module
VSS	Virtual Switch System (Cisco)

W

WMI	Windows Management Instrumentation
WS-MAN	Web Services-Management

X

XMLRPC	XML Remote Procedure Call
XMPP	Extensible Messaging and Presence Protocol