



The bridge to possible

Infrastructure as Code (IaC)

Ansible 활용 및 소개

허재프로, Technical Solution Specialist
Korea Cloud Architecture SE Team

Index

- **laC란?**

Infrastructure as Code (laC) 의 필요성

- **laC 활용 사례 및 데모**

CSV(엑셀)를 이용한 ACI 배포

Ansible 를 활용한 ACI, Nexus 스위치 모니터링

- **요약**

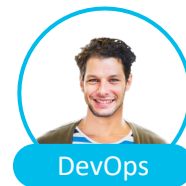
대규모 하이브리드 클라우드 데이터센터 고객 요구사항



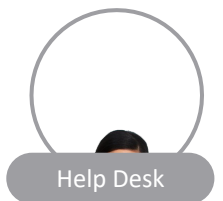
어떻게 신속하고 정확한
어플리케이션 배포를 위한 대규모
IT 운영을 할 수 있을까요?

다양한 CI/CD Tool들을 통합하여 사용할
수 있어야 합니다.

Terraform, Jenkins, Github, Ansible, etc



DevOps



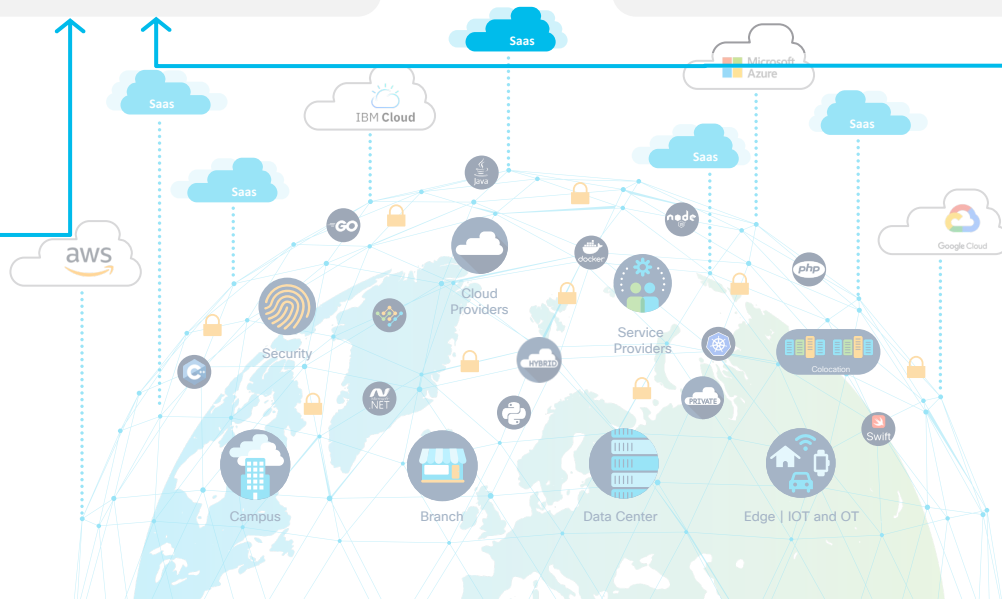
Help Desk

특정 도메인 벤더의
전문가 없이도 인프라
자동화구성이
필요합니다.



LoBs

IT 티켓 대응 및 관리 이력
추적을 위한 셀프서비스
대쉬보드 및 모니터링이
필요합니다.

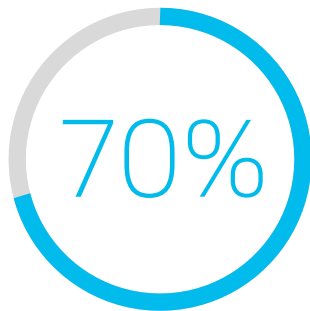


신속한 고객 어플리케이션 배포를 위한 인프라의 Day 0 ZTP 플랫폼이 필요!

Legacy 인프라 운영팀 요구사항



95%의 네트워크
변경을 수동 작업



Human Error로 인한
정책 위반 발생



네트워크 모니터링 &
트러블슈팅에 대한 OpEx 증가
비율

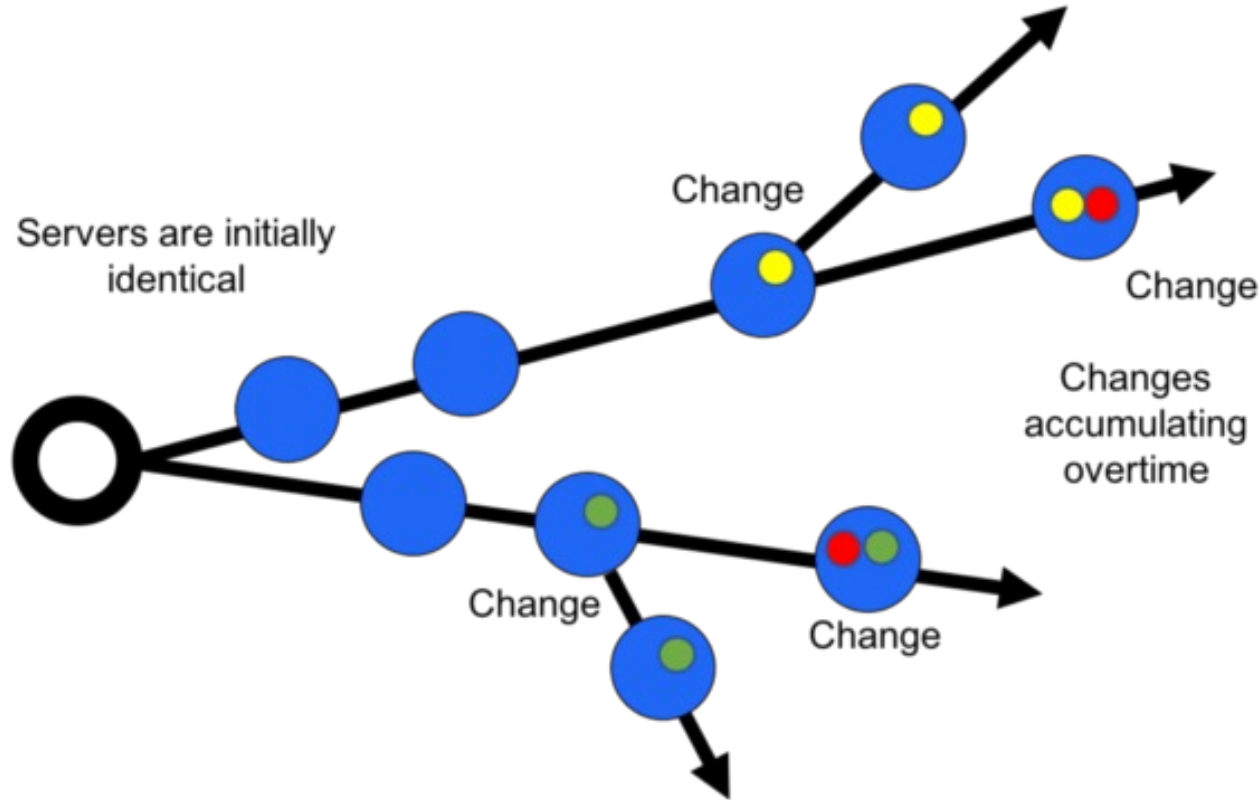


정책 배포, 모니터링을 동시에 할 수 있는 효율적인 방법이 필요!

laC 소개 및 Use-Case



Why IaC? - Configuration Drift



Why Infrastructure-as-Code (IaC)?

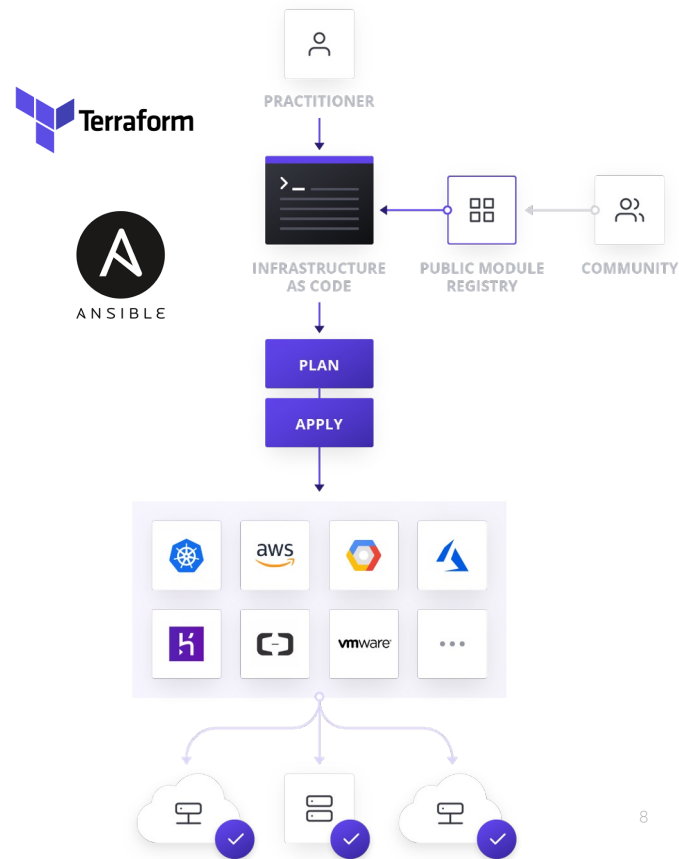
- Automation
 - 매뉴얼 운영으로 오래 걸리는 설정을 자동화
 - 자동화된 프로세스를 손쉽게 여러 환경에 적용
- Multi-solution Orchestration
 - 오픈 생태계를 통해 멀티 벤더, 멀티 솔루션 모듈 제공
- Infra History Tracking & Rollback
 - SVC(Git) 내 소스 관리를 통해 인프라 변경사항을 저장 및 조회 가능
 - 이전 인프라 상태로 쉽게 롤백 가능 (Terraform)
 - 코드를 통한 인프라 상태 모니터링 (Ansible)
- Auto provisioning
 - CI/CD 파이프라인 구성을 통해 애플리케이션 배포 시 인프라 자동 구성
-> Ansible AWX 및 Terraform Enterprise/Cloud 를 통해 구현 가능

- 확장성 (동일 형상 재구축)
- 안정성 (Human-Error 감소)
- 민첩성

→ ROI & TCO 감소

IaC (Infrastructure as a Code)의 효과

- 버전 관리 시스템을 통한 휴먼에러 방지 및 이력 관리추적
- 오픈소스 Provider/ Module을 통한 빠르고 쉬운 인프라 관리 모듈 생성
- 다양한 벤더 인프라와의 API를 통한 인프라 관리
- 시스코의 10개 이상의 Provider와 28개의 공식 Github Repository 제공
- 인프라의 자동화 및 코드화
ex) 서버 프로파일 배포, VM 배포 및 스냅샷 관리, 다수의 네트워크 장비 설정 자동화, 클라우드 환경 배포 자동화

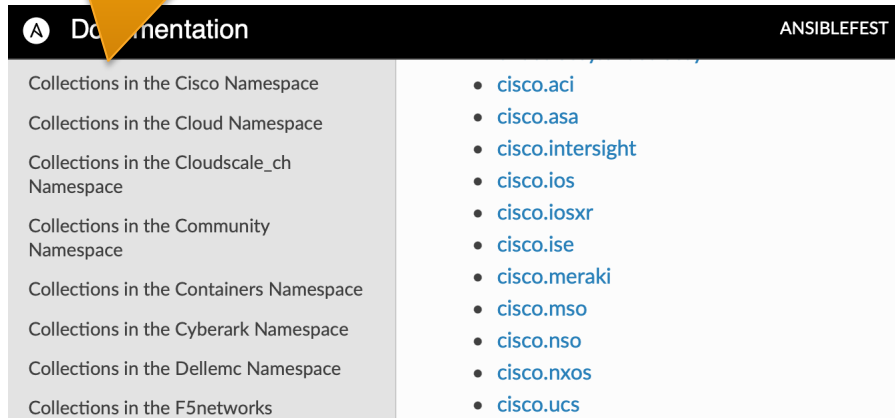


Cisco 공식 Terraform & Ansible 플러그인 제공

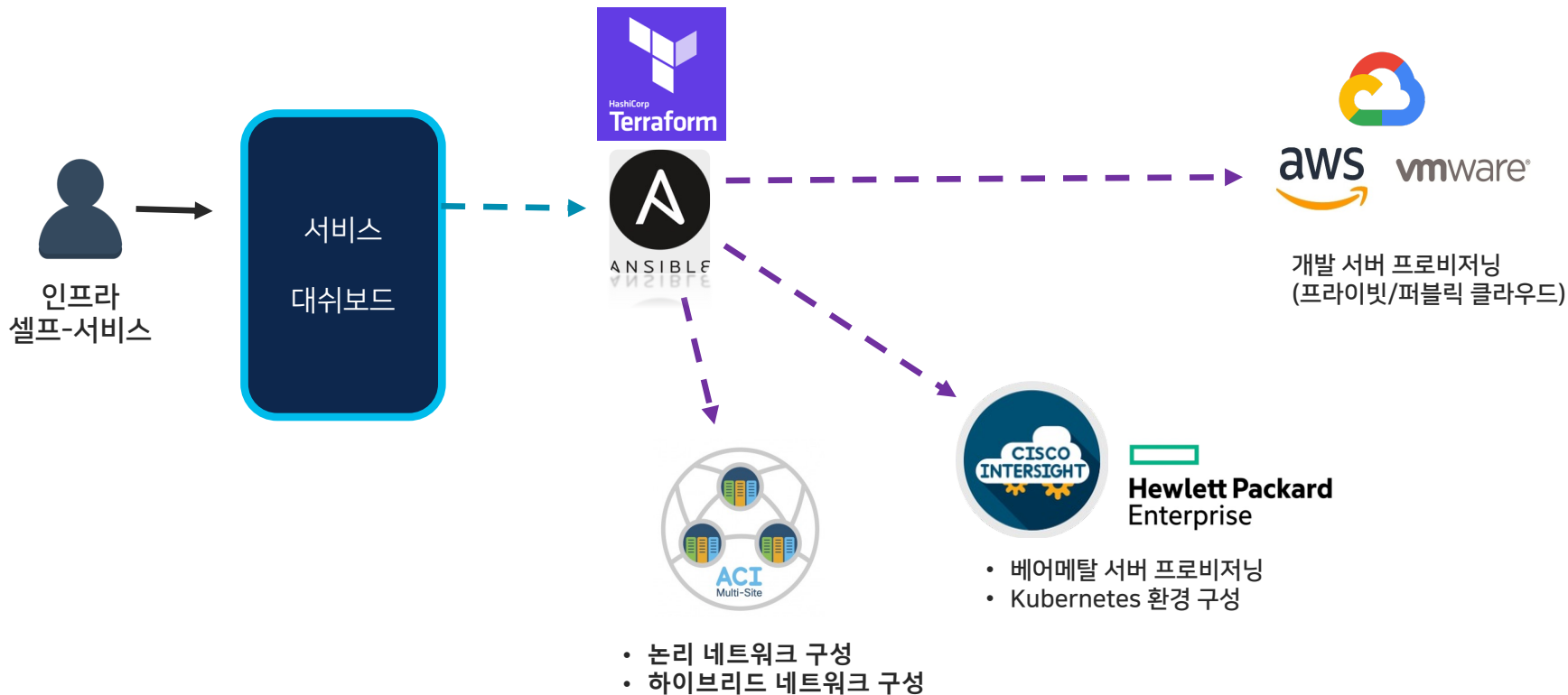
10개 이상의
테라폼 Provider 제공



10개 이상의
Ansible 모듈 제공



IaC 활용 Automation – 하이브리드 클라우드 Use Case



What is Ansible?



- 오픈소스
- 자동화, 장비 개별 설정에 중심
- Version 2.10
 - ACI support - 2.4
- UNIX/Linux 지원
- 다양한 Cisco 장비 지원
 - ACI, MSO, IOS, NX-OS, IOS- XR
- Agentless
 - Push Model
- Idempotent
- YAML based
 - 읽기 쉬운 문법 구조
- APIC REST API interface
 - GUI와 동일
- no programming skills

Ansible 구조

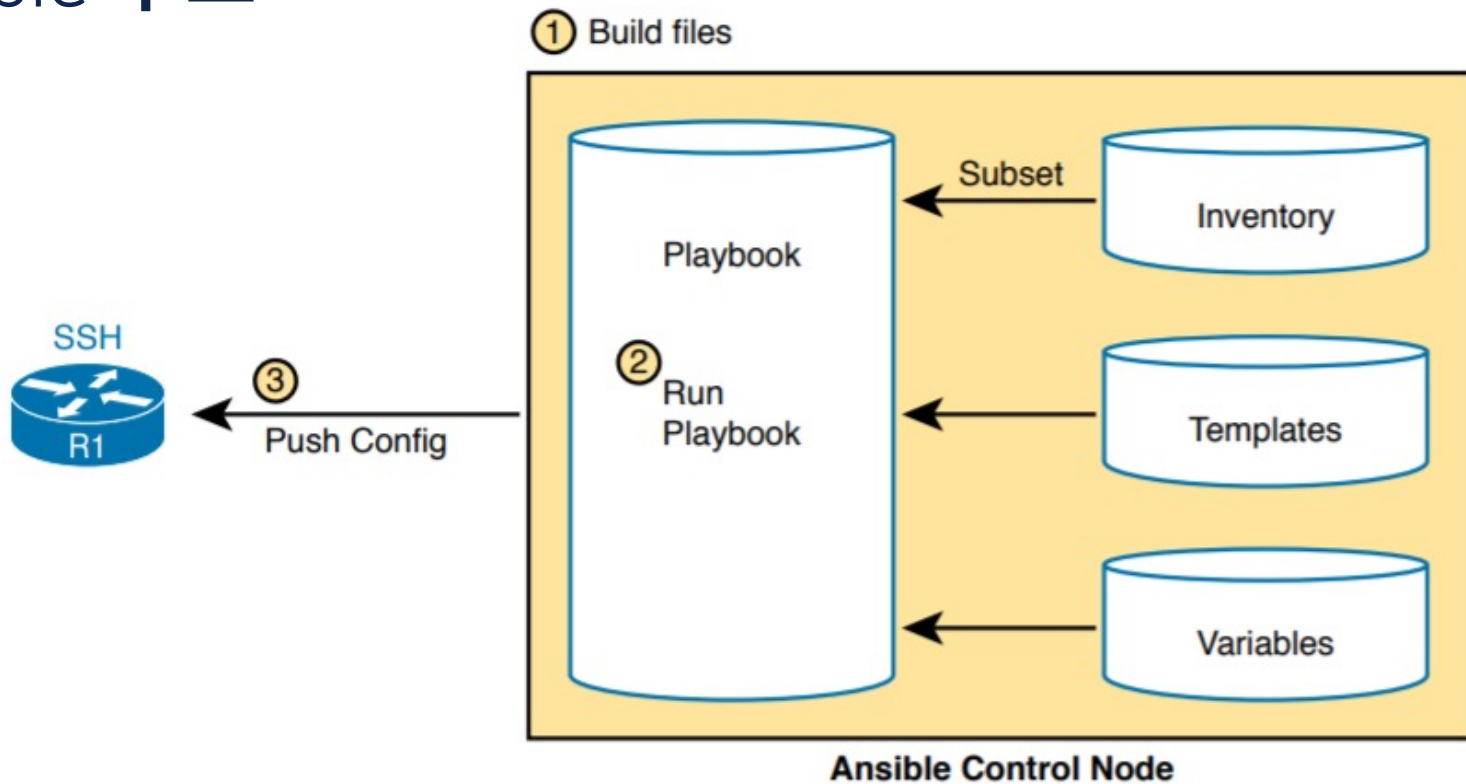
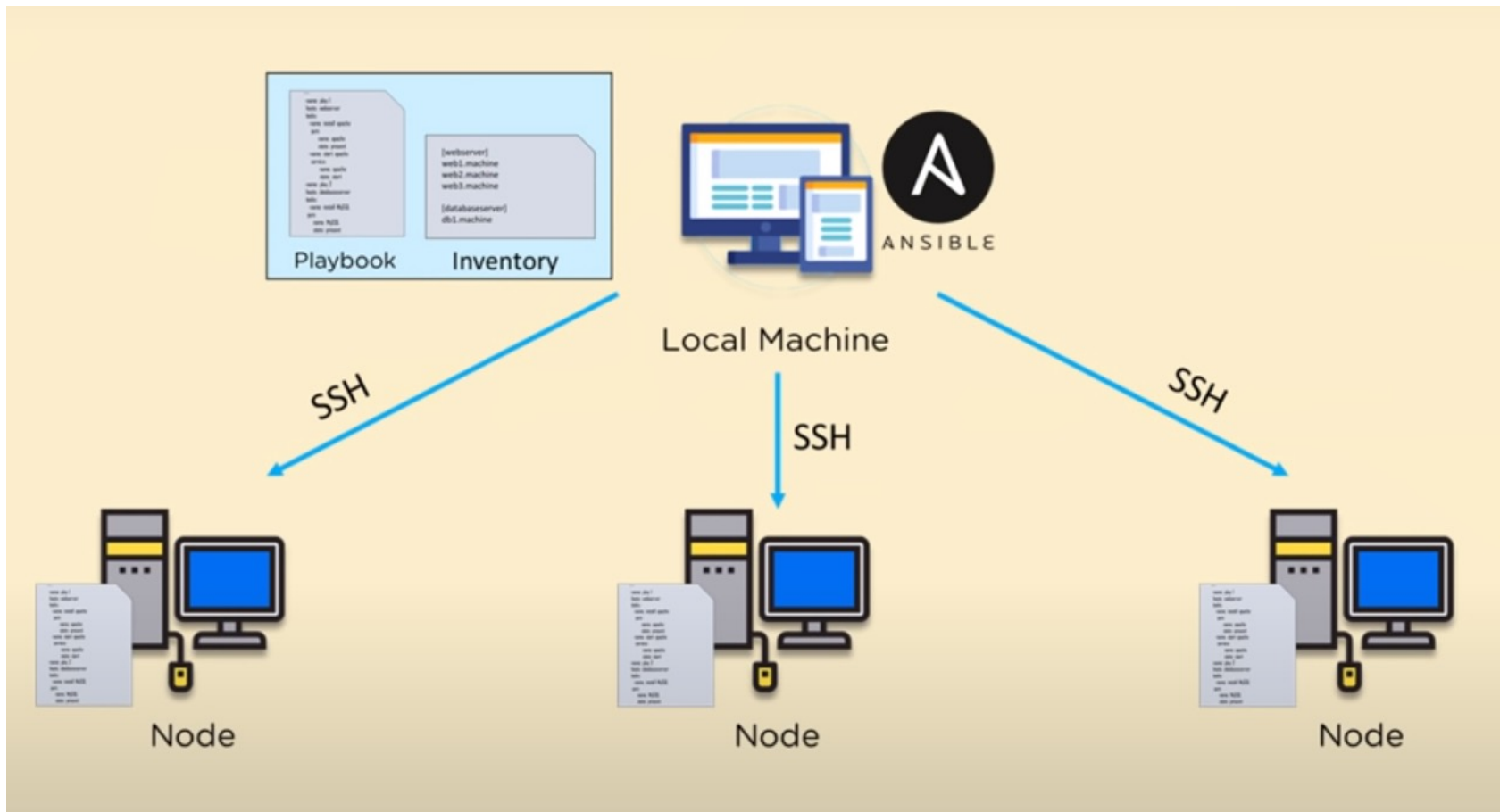
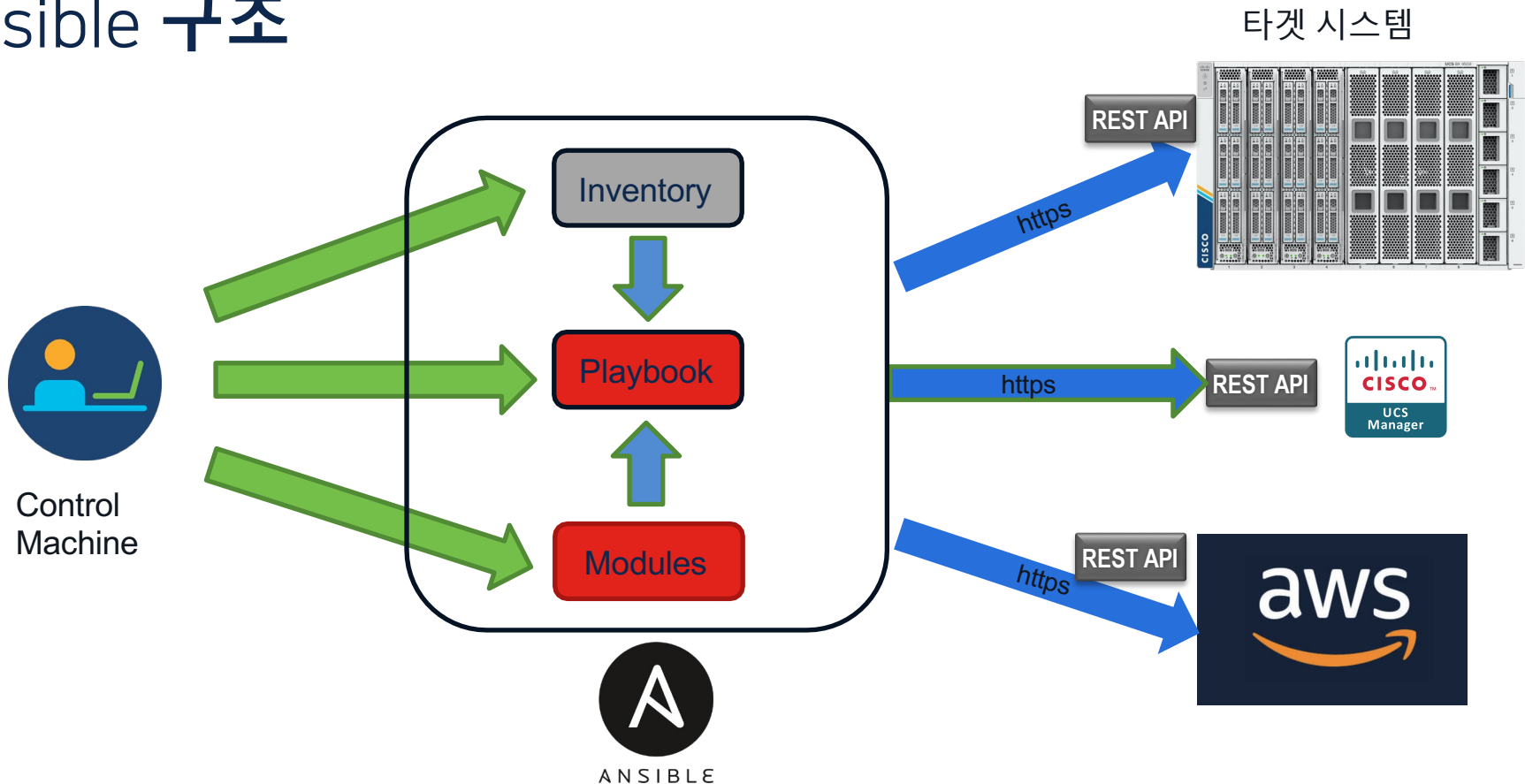


Figure 19-8 *Ansible Push Model*

Ansible 구조



Ansible 구조



ACI Ansible Inventory 예시

YAML inventory file

```
apic1:
  hosts:
    10.9.3.21:
  vars:
    username: admin
    password: CiscoAC1
```

INI inventory file

```
[apic1]
10.9.3.21 username=admin password=CiscoAC1
10.9.3.22 username=ansible privatekey=ansible.key
```

Ansible Playbook 파헤치기

```
---  
# Demo ACI Playbook  
- name: Configuring Example Tenant  
  hosts: apic1  
  connection: local  
  gather_facts: no  
  
  tasks:  
    - name: Create Tenant  
      aci_tenant:  
        hostname: "{{ inventory_hostname }}"  
        username: "{{ username }}"  
        private_key: ansible.key  
        tenant: "Cisco"  
        description: "Tenant configured by Ansible"  
        validate_certs: no  
        state: present
```

Start of YAML

Comment

Name of Playbook

Hosts from inventory

Connection is local to this host

Collects information about targets

Watch the Indentation!

Ansible Playbook 파헤치기

```
---
# Demo ACI Playbook
- name: Configuring Example Tenant
  hosts: apic1
  connection: local
  gather_facts: no

tasks:
- name: Create Tenant
  aci_tenant:
    hostname: "{{ inventory_hostname }}"
    username: "{{ username }}"
    private_key: ansible.key
    tenant: "Cisco"
    description: "Tenant configured by Ansible"
    validate_certs: no
    state: present
```

Task name

Module Name

Hostname

Authentication

Tenant

Description of task

Validate certs

Add if not already "present"

Ansible ACI Modules

```
(2.9) threnzy@THRENZY-M-F1G3 2.9 % ansible-doc -l | grep ^aci
aci_aaa_user                Manage AAA users (aaa:User)
aci_aaa_user_certificate    Manage AAA user certificates (aaa:UserCert)
aci_access_port_block_to_access_port  Manage port blocks of Fabric interface poli...
aci_access_port_to_interface_policy_leaf_profile  Manage Fabric interface policy leaf profile...
aci_access_sub_port_block_to_access_port  Manage sub port blocks of Fabric interface ...
aci_aep                    Manage attachable Access Entity Profile (AE...
aci_aep_to_domain          Bind AEPs to Physical or Virtual Domains (i...
aci_ap                    Manage top level Application Profile (AP) o...
aci_bd                    Manage Bridge Domains (BD) objects (fv:BD)
aci_bd_subnet             Manage Subnets (fv:Subnet)
aci_bd_to_l3out           Bind Bridge Domain to L3 Out (fv:RsBDToOut)
aci_config_rollback       Provides rollback and rollback preview func...
aci_config_snapshot       Manage Config Snapshots (config:Snapshot, c...
aci_contract              Manage contract resources (vz:BrCP)
aci_contract_subject       Manage initial Contract Subjects (vz:Subj)
aci_contract_subject_to_filter  Bind Contract Subjects to Filters (vz:RsSub...
aci_domain                Manage physical, virtual, bridged, routed o...
aci_domain_to_encap_pool  Bind Domain to Encap Pools (infra:RsVlanNs)
aci_domain_to_vlan_pool   Bind Domain to VLAN Pools (infra:RsVlanNs)
aci_encap_pool            Manage encap pools (fvns:VlanInstP, fvns:Vx...
aci_encap_pool_range      Manage encap ranges assigned to pools (fvns...
aci_epg                   Manage End Point Groups (EPG) objects (fv:A...
aci_epg_monitoring_policy Manage monitoring policies (mon:EPGPol)
aci_epg_to_contract       Bind EPGs to Contracts (fv:RsCons, fv:RsPro...
aci_epg_to_domain         Bind EPGs to Domains (fv:RsDomAtt)
aci_fabric_node           Manage Fabric Node Members (fabric:NodeIden...
aci_fabric_scheduler      This modules creates ACI schedulers
aci_filter                Manages top level filter objects (vz:Filter...
aci_filter_entry           Manage filter entries (vz:Entry)
```

Ansible ACI Modules – EPG Module Example

```
- name: Add a new EPG
  cisco.aci.aci_epg:
    host: apic
    username: admin
    password: SomeSecretPassword
    tenant: production
    ap: intranet
    epg: web_epg
    description: Web Intranet EPG
    bd: prod_bd
    monitoring_policy: default
    preferred_group: true
    state: present
    delegate_to: localhost
```

```
if state == "present":
    aci.payload(
        aci_class="fvAEPg",
        class_config=dict(
            name=epg,
            descr=description,
            prio=priority,
            pcEnfPref=intra_epg_isolation,
            fwdCtrl=fwd_control,
            prefGrMemb=preferred_group,
            nameAlias=name_alias,
            isAttrBasedEPg=useg,
        ),
        child_configs=child_configs,
    )

    aci.get_diff(aci_class="fvAEPg")

    aci.post_config()
```

ACI Tenant Playbook 예시

```
vim
# Demo ACI Playbook
- name: Configuring Example Tenant
  hosts: apic1
  connection: local
  gather_facts: no

  tasks:
  - name: Create a New Tenant
    cisco.aci.aci_tenant:
      hostname: "{{ inventory_hostname }}"
      username: "{{ username }}"
      password: "{{ password }}"
      tenant: "CiscoLive"
      description: "Tenant configured by Ansible"
      validate_certs: no
      state: present
```

ACI Tenant Playbook 예시

```
(2.9) THRENY-M-F1G3:BRKACI-1619 threnzy$ ansible-playbook -i hosts ciscolive.yml

PLAY [Configuring Example Tenant] *****

TASK [Create a New Tenant] *****
changed: [10.95.33.231]

PLAY RECAP *****
10.95.33.231      : ok=1    changed=1    unreachable=0    failed=0    skipped=0
rescued=0        ignored=0

(2.9) THRENY-M-F1G3:BRKACI-1619 threnzy$
```

- Runs through each task.
- Let's you know how many tasks were OK, changed, failed, etc.
- To see more output use “-v”, “-vvv”, or “-vvvv”

ACI Tenant Playbook 예시

All Tenants

Name	Alias	Description	Bridge Domains	VRFs	EPGs	Health Score
CiscoLive		Tenant configured by Ansible	0	0	0	100
common			1	2	0	100
infra			2	2	2	100
mgmt			1	2	0	100

IaC 활용 ACI 자동화 - Use Case

프로비저닝 (Terraform)

- 모든 고객사 환경에 동일하게 필요한 설정 작업
 - Fabric Policy 설정
 - Common tenant 내 L3out 설정
- GUI 환경에서 오래 걸리는 설정 / 잦은 변경
 - Static path EPGbinding 설정
 - 애플리케이션 배포 시 필요한 네트워크 변경 사항
 - Rollback 및 CI/CD 형상 관리
 - 하이퍼바이저 Snapshot 관리 및 VM 배포
 - Drift 기능을 통한 인프라 변경 감지

모니터링 (Ansible)

- Health Score
- Interface 상태
- SYSLOG...
- 실시간 Webex / Email Alert
- Excel 형태의 데이터 추출

그런데..
단순 정책 배포만으론
부족합니다.

휴면에러를 줄이는
동시에 대량
배포를하는 방법은
없을까?



인프라 운영팀

Excel을 통한 ACI 데이터 IaC 대량배포



인프라 운영팀



L3Out

Static Port

EPG Binding

VRF, Routing



대량의 정책 생성 (스위치로 번갈)

엑셀을 통해 다수의 정책 작성

1	tenant	vrf	bd	bd_gateway	bd_mask	bd_scope	ap	epg1	epg2	epg3	contract1	contract2	filter1
2	csv	csv-vrf	csv-bd	10.1.1.1	24	private	csv_ap	web	app	db	web_to_app	app_to_db	web_to_app
3	cvs2	csv2-vrf	csv2-bd	10.2.2.1	24	public	csv2_ap	moe	larry	curly	web_to_app	app_to_db	web_to_app

Excel을 통한 ACI 데이터 IaC 대량배포

```
---
- name: Configuring Tenant with CSV file
  hosts: apic1
  connection: local
  gather_facts: false
  vars:
    aci_login: &aci_login
    hostname: '{{ inventory_hostname }}'
    username: '{{ username }}'
    password: '{{ password }}'
    validate_certs: False
    output_path: "check_json.json"
  tasks:
    # Read a CSV file and access Tenant csv
    - name: Read data from CSV file to be used to configure ACI
      read_csv:
        path: tenant.csv
        # key: tenant
      register: mytenant
```

외부 csv 파일을
읽고, mytenant
변수로 등록

```
- name: Create Tenants using CSV and loop
  aci_tenant:
    <<: *aci_login
    hostname: "{{ inventory_hostname }}"
    username: "{{ username }}"
    #private_key: "{{ private_key }}"
    tenant: "{{ item.tenant }}"
    description: "Tenant configured by Ansible"
    validate_certs: false
    state: present
  loop:
    - tenant: "{{ mytenant.list.0.tenant }}"
    - tenant: "{{ mytenant.list.1.tenant }}"
```

Mytenant 변수를
읽고, Tenant
이름으로 지정하여
Task 실행

인프라 모니터링

인프라 장애가 났을 때
Downtime을 줄이고,
Self-Recovery하는
방법은 없을까?



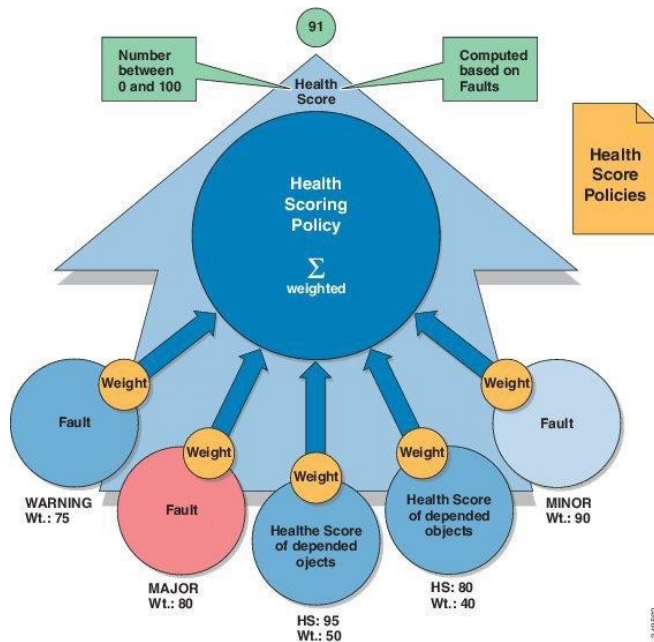
인프라 운영팀

Ansible AWX을 통한 ACI 인프라 모니터링



ACI 모니터링 대상

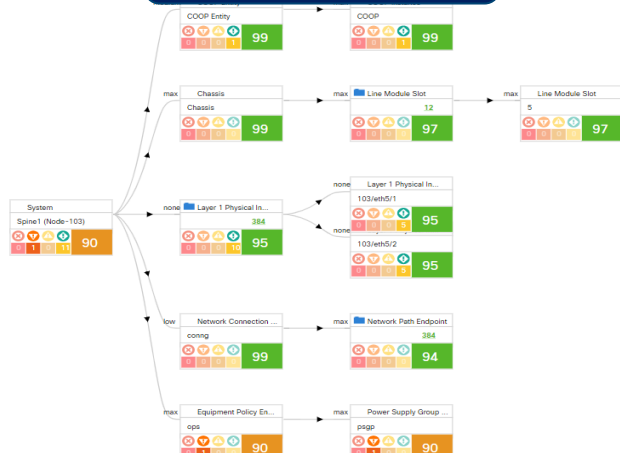
- APIC 대쉬보드 모니터링 제공대상:
 - 시스템 헬스 스코어
 - Pod 헬스 스코어
 - Tenant 헬스 스코어
 - Fault
 - ACI 제공 모든 API
- 헬스 스코어: 지정한 Object의 Fault 기반 점수. Fault 찾을 시, Severity Level이 높을 시 낮게 책정.
- Health Score는 ACI Policy에 따라 계산됨
- NXOS 스위치의 specific한 show 명령어



ACI 모니터링 대상

하드웨어 장애 탐지 및 헬스 스코어 제공

하드웨어 상태 분석



하드웨어 장애 탐지

System

Tenants

Fabric

Virtual Networking

L4-L7 Services

Admin

Operations

Apps

Integrations

QuickStart

Dashboard

Controllers

System Settings

Smart Licensing

Faults

Config Zones

Events

Audit Log

Active Sessions

Faults

▼ Severity

Domain

Type

Code

Count

Cause

Sample

Infra

Operational

F1201

1

span-provision-failed

This

Access

Operational

F1425

2

ip-provisioning-failed

This

Tenant

Operational

F0135

5

unsupported-operation

This

Infra

Operational

F0103

1

port-down

This

Infra

Operational

F3848

1

equipment-psu-down

This

External

Operational

F0299

2

protocol-bgp-adjacency-down

This

문제발견시 TS 제공

Cause	Sample	Sort
Interface-management-...	This fault occurs when a port is set to up but the operational state is down	Sort Ascending
Interface-physical-down	This fault occurs when a port is down and is in use for Infra and epg	Sort Descending
configuration-failed	This fault occurs when an End Point Group is incompletely or incorrectly configur...	Columns
protocol-isis-down	This event occurs when the oper state of a Mcast tree changes to inactive.	Filters

ACI 모니터링 대상

Interface 별 Traffic 사용량 및 EPG 별 Traffic 사용량 정보 제공



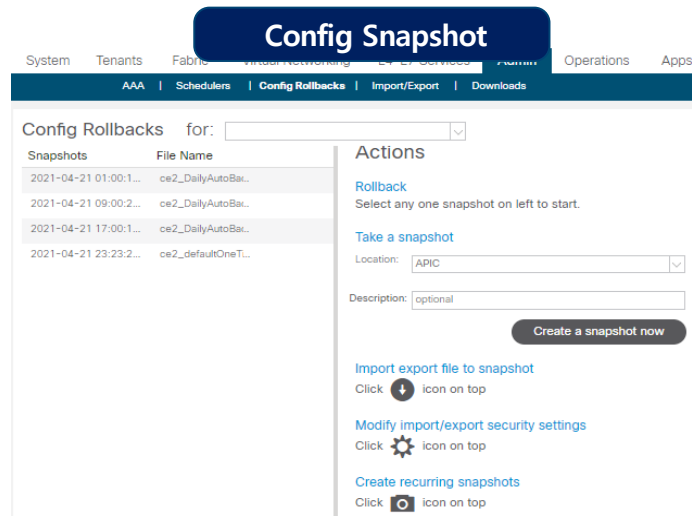
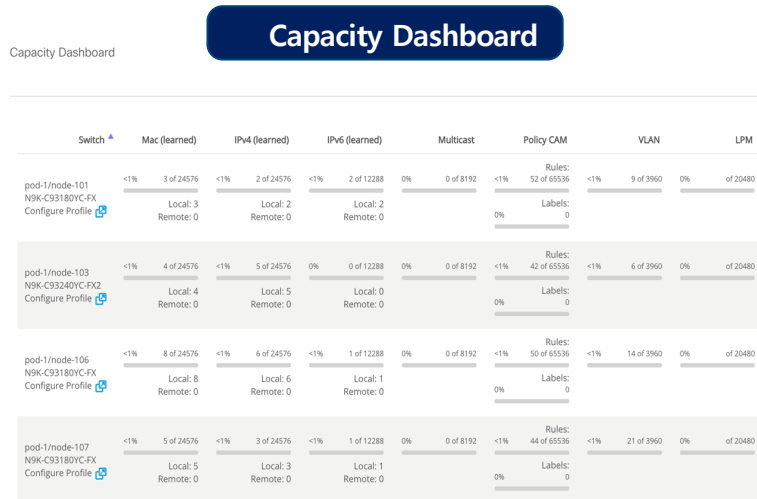
* Timeline 기능으로 과거의 Stats 도 조회가 가능합니다.



* Timeline 기능으로 과거의 Stats 도 조회가 가능합니다.

ACI 모니터링 대상

실시간 Resource 관리 및 Config Snapshot



* Config Snapshot 기능을 통해 쉬운 Rollback 제공

레퍼런스

Cisco Customer Success 팀 IaC를 통한 ACI 자동화 사례

제조 S사의 데이터센터 ACI 네트워크 구축

대상 환경

- ACI 기반 데이터센터 네트워크 구성
 - 제조/IT/R&D 인프라 (500여대)
- 신규 R&D HPC 구축
 - ACI 80여대, 서버 2000여대 규모
 - DC환경포함 구축기간 1개월 (ACI 1일)

자동화 적용

- ACI 신규 구축시 Terraform 오픈소스 도입
 - 서버포트 단위 설정 : 7000여개 작업
 - SDN 논리 구성 포트 매핑
 - 기획/작업자 대상 Terraform 지원

결과

- 인프라 작업 속도 개선
 - ACI 구축 수일 → 1시간내 완료
 - 테스트망 사전검증
 - 작업결과 실시간 확인 가능
- 인프라 자동화 확대
 - 전사 모범사례 선정
 - 경영진 지시로 인프라 자동화 전체 확산 (컴퓨팅 영역, Terraform)

CI/CD 소개

- CI - 지속적 통합 (Continuous Integration)
 - 모든 개발자의 변경사항을 한 곳의 공유 레포지토리로 통합해 빌드/관리하는 협업 도구
 - 소프트웨어 개발 시 코드 변경사항에 대한 관리를 통합해 빠르게 변경/검증하는 방법
 - 사용자가 코드작성 시 젠킨스는 사용자를 대신해 코드를 Pull/Execute/Validate
- CD - 지속적 배포 (Continuous Deployment)
 - 자동화된 배포 파이프라인을 통한 새로운 소프트웨어를 전달하는 방법
 - 변경사항 추적성은 지속적 통합 도구에 의존

젠킨스? CI/CD?



Jenkins

- 사용자의 서버에서 여러 CI/CD 매크로를 만들어주는 시종
- **1800개** 이상의 플러그인 제공으로 높은 연동성
- 온프레미스 설치형 Web GUI 소프트웨어

CI/CD IaC 파이프라인 예시



Pipeline intersight_iac_demo_compute

Intersight IaC Demo Pipeline

edit description

Disable Project



Recent Changes

Stage View

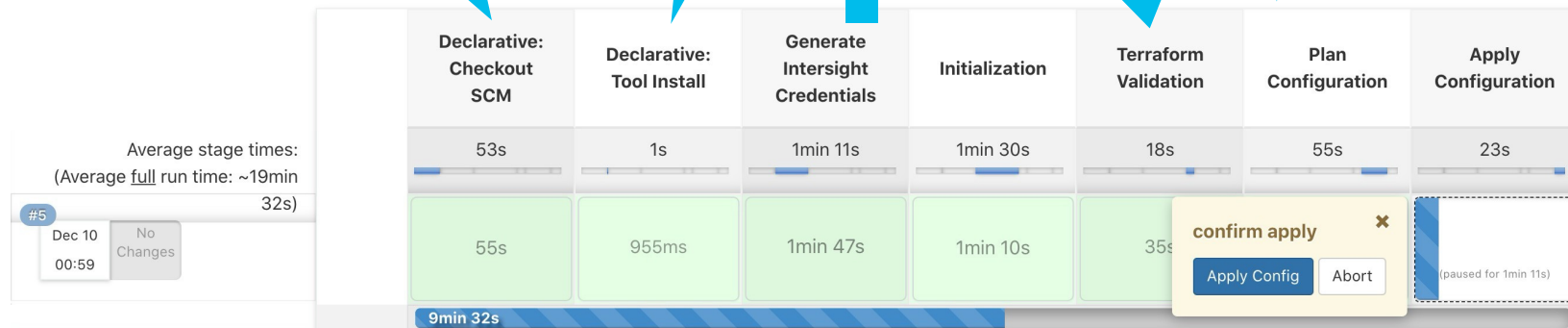
Github에서 인프라
코드를 Checkout &
Pull

빌드에 필요한 Tool
설치

Fetch Intersight
api_key를 암호관리
시스템인 Vault에서
가져오고,
인증서도 발급

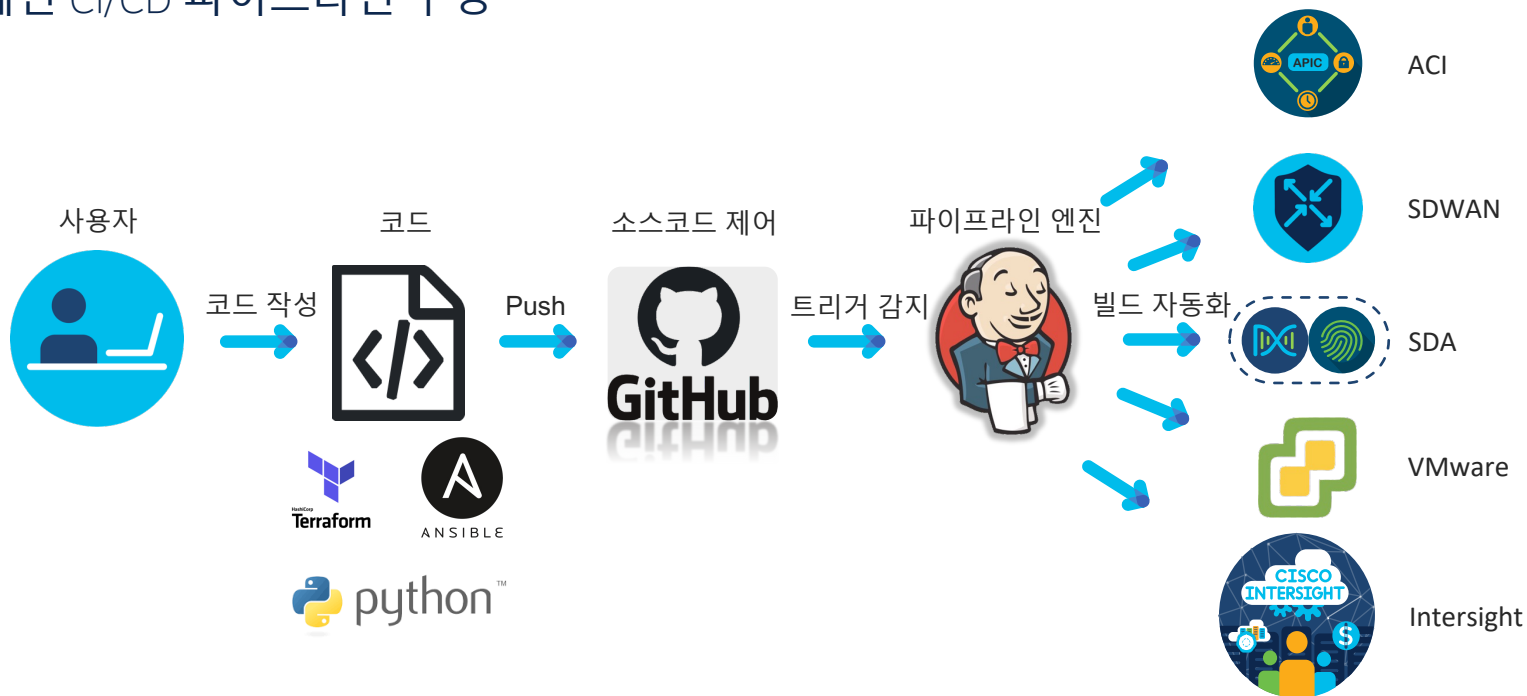
Terraform Plan

Terraform init

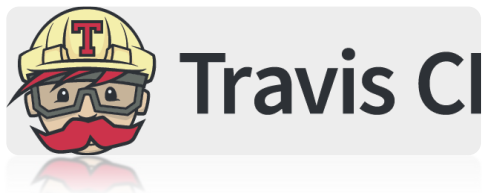


CI/CD 워크플로우

멀티도메인 CI/CD 파이프라인 구성



CI/CD의 다양한 Tools



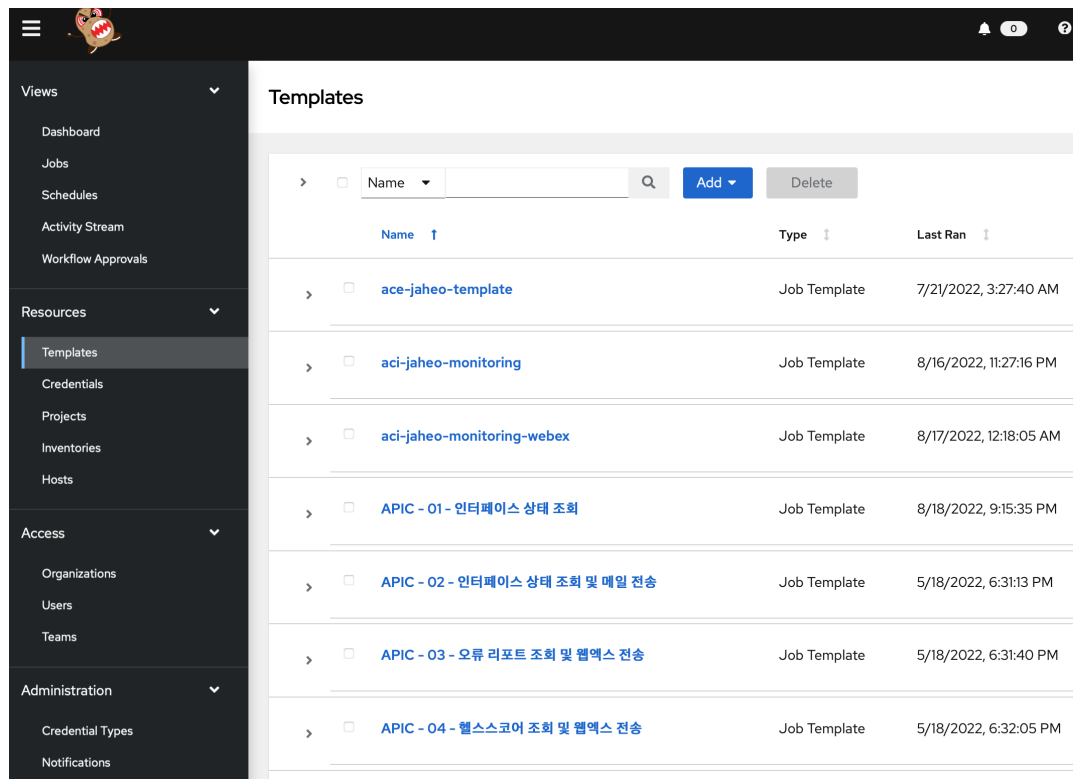


젠킨스 도입효과

- 지속적 통합 (Continuous Integration)
 - 모든 개발자의 변경사항을 한 곳의 공유 레포지토리로 통합해 빌드/관리하는 협업 도구
 - 소프트웨어 개발 시 코드 변경사항에 대한 관리를 통합해 빠르게 변경/검증하는 방법
 - 사용자가 코드작성 시 젠킨스는 사용자를 대신해 코드를 Pull/Execute/Validate
- 편리한 사용
 - 역할 기반 인증(RBAC)을 통해 사용자의 권한 관리 가능
 - Web 기반 GUI를 제공, 사용자가 스크립트에 쉽게 접속 가능
 - 알림/로깅 기능을 통해 작업 결과를 중앙화 관리 가능 및 사용자에게 알림

Ansible AWX란?

- Ansible Tower의 오픈소스 버전, 온프레미스 환경 구축
- RBAC 기반 안전하고 효율적인 Ansible 개발 Pipeline
 - 팀별, 유저별 권한제어 가능
 - 웹 UI/API 제공
- Ansible 자동화 및 관리, 모니터링 제공
- Ansible 작업 실행 기록 및 작업 예약 기능 제공



The screenshot displays the Ansible Tower web interface. On the left is a dark sidebar with a navigation menu. The main content area on the right is titled 'Templates' and shows a list of job templates. The sidebar menu includes sections for Views, Resources, Access, and Administration. The 'Resources' section has 'Templates' highlighted. The 'Templates' page includes a search bar, an 'Add' button, and a 'Delete' button. Below these is a table with columns for Name, Type, and Last Ran.

Name	Type	Last Ran
ace-jaheo-template	Job Template	7/21/2022, 3:27:40 AM
aci-jaheo-monitoring	Job Template	8/16/2022, 11:27:16 PM
aci-jaheo-monitoring-webex	Job Template	8/17/2022, 12:18:05 AM
APIC - 01 - 인터페이스 상태 조회	Job Template	8/18/2022, 9:15:35 PM
APIC - 02 - 인터페이스 상태 조회 및 매일 전송	Job Template	5/18/2022, 6:31:13 PM
APIC - 03 - 오류 리포트 조회 및 웹엑스 전송	Job Template	5/18/2022, 6:31:40 PM
APIC - 04 - 헬스스코어 조회 및 웹엑스 전송	Job Template	5/18/2022, 6:32:05 PM

Ansible AWX와 ACI, 스위치 모니터링

The screenshot shows the Ansible AWX web interface. The sidebar on the left contains the following sections:

- Views**
 - Dashboard
 - Jobs
 - Schedules
 - Activity Stream
 - Workflow Approvals
- Resources**
 - Templates** (highlighted)
 - Credentials
 - Projects
 - Inventories
 - Hosts
- Access**
 - Organizations
 - Users
 - Teams
- Administration**
 - Credential Types
 - Notifications

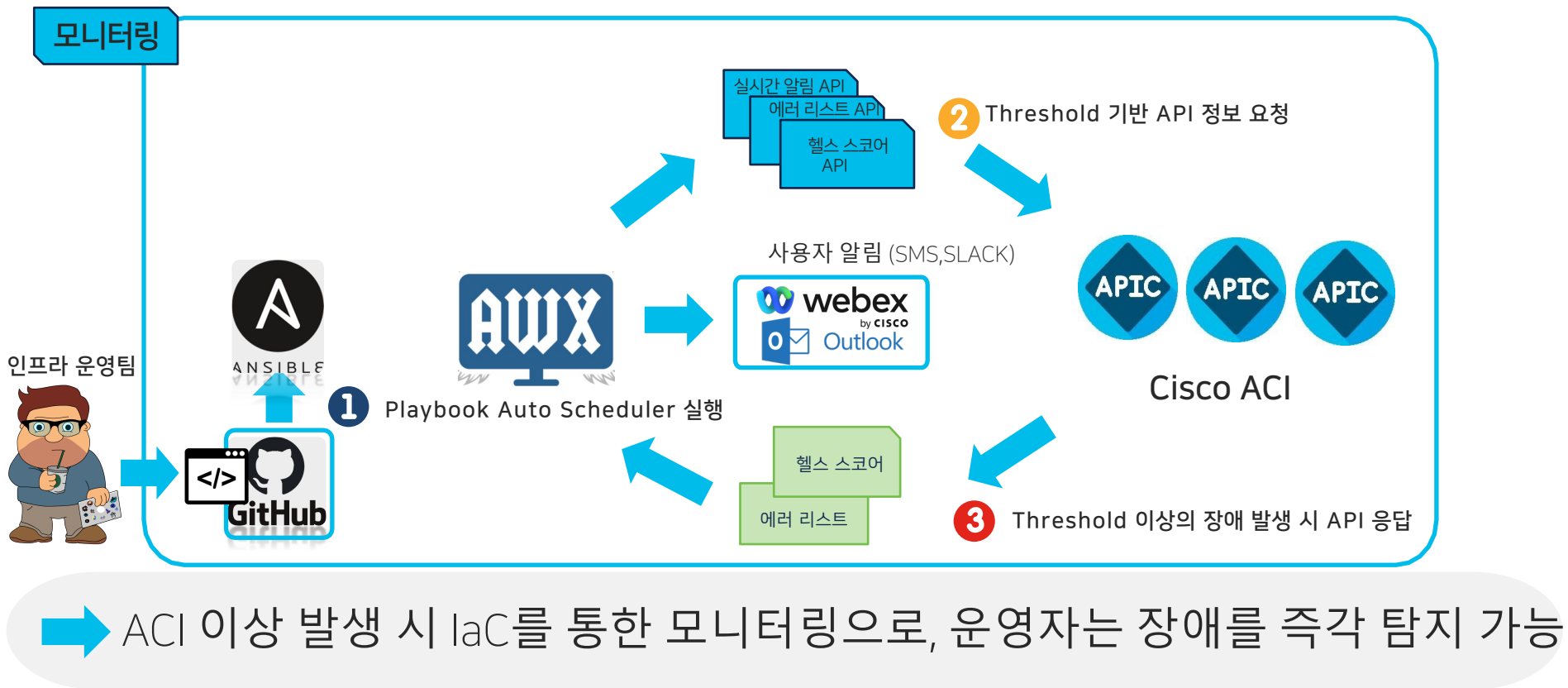
The main content area is titled "Templates" and displays a table of templates. The table has columns for Name, Type, and Last Ran. The templates listed are:

Name	Type	Last Ran
ace-jaheo-template		2022, 3:27:40 AM
aci-jaheo-monitoring		2022, 11:27:16 PM
aci-jaheo-monitoring-webex		2022, 12:18:05 AM
APIC - 01 - 인터페이스 상태 조회	Job Template	8/18/2022, 9:15:35 PM
APIC - 02 - 인터페이스 상태 조회 및 메일 전송		
APIC - 03 - 오류 리포트 조회 및 웹엑스 전송		
APIC - 04 - 헬스스코어 조회 및 웹엑스 전송	Job Template	5/18/2022, 6:32:05 PM

Two green callout boxes highlight specific templates:

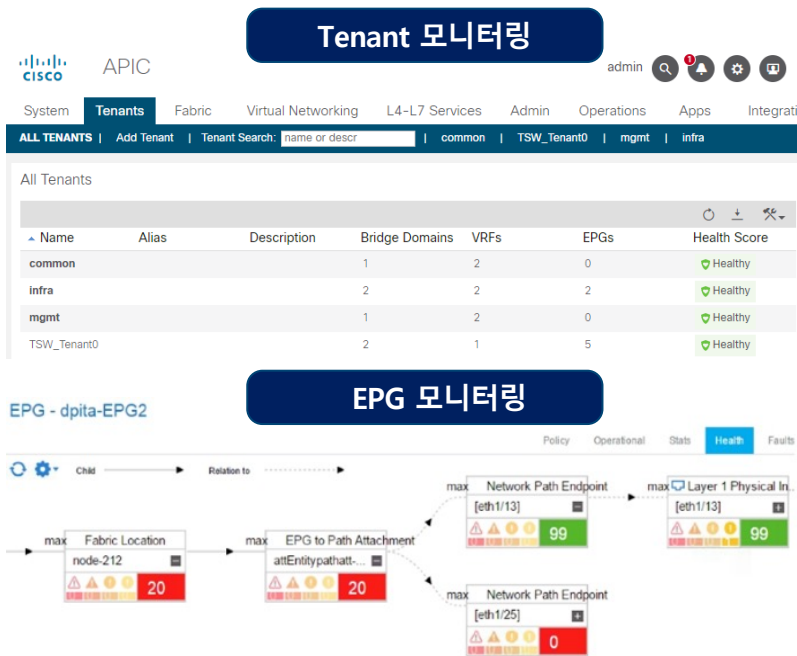
- Callout 1:** 인터페이스 상태 정보를 1분 마다 조회하도록 설정 (Interface status information is set to be queried every 1 minute).
- Callout 2:** 장애 발생 시 Webex 및 메일 전송 트리거 (Trigger for Webex and email transmission when a failure occurs).

laC를 활용한 ACI 자동화 Day 2 모니터링 운영 예시



ACI 모니터링 대상

ACI Fabric 전체 / Tenant별 장애 탐지 및 헬스 스코어 제공



요약

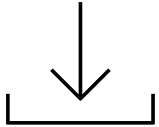
- 네트워킹 CI/CD 파이프라인은 네트워크 프로비저닝 프로세스를 일관되고 자동화된 방식으로 구현하는 데 필수적입니다
- 변경 작업으로 인해 발생하는 사고의 비율이 높습니다
따라서 변경 전/후 검증을 파이프라인에 포함하는 것이 매우 중요합니다
- 시스코 솔루션 및 IaC 도입을 통하여 네트워크 변경 전 및 변경 후 검증을 위해 파이프라인에 통합할 수 있습니다

요약

- IaC를 통하여 하이브리드 클라우드 환경의 인프라에 대해 통합 멀티 벤더 자동화가 가능합니다. 이 과정에서 Cisco NDFC를 함께 이용할 경우, 시스코 네트워크 인프라의 GUI Controller 기반의 ZTP와 자동화 모니터링을 함께 수행할 수 있습니다.
- IaC는 단순 배포 방식이 아닌, csv 파일 등 운영자에 더 편리하고 간편한 형식의 외부 데이터를 선언할 수 있습니다. 이는 운영환경의 휴먼 에러의 감소로 장애를 방지하고 효율적인 운영환경을 제공합니다.
- Ansible AWX 등의 소프트웨어를 통하여 데이터 센터 Fabric에 대해 운영자가 원하는 정보를 지속적으로 민첩하게 모니터링하여 장애 발생 시 Downtime을 줄일 수 있습니다.

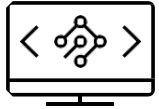
Next Steps

What to do after session



Cisco and Ansible Guide Link

[Code in GitHub](#)



Test the code in your lab (or use DevNet Sandbox)

[DevNet ACI Sandboxes](#)



Explore more about Cisco Live IaC Sessions

[DEVNET-1369](#) | [BRKDCN-2673](#)

감사합니다.