



# Cisco APIC Container Plug-in Release Notes, Release 5.2(7)

## Introduction

This document describes the features, bugs, and limitations for the Cisco Application Policy Infrastructure Controller (APIC) Container Plug-in.

The Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) Plug-in provides network services to Kubernetes, Red Hat OpenShift, and Rancher RKE clusters on a Cisco ACI fabric. It allows the cluster pods to be treated as fabric end points in the fabric integrated overlay, as well as providing IP Address Management (IPAM), security, and load balancing services.

Release Notes are sometimes updated with new information about restrictions and bugs. See the following website for the most recent version of this document:

<https://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>

For more information about this product, see "Related Content."

**Note:** The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Date	Description
September 6, 2023	Added CSCwf64555 to the list of Resolved Issues.
July 9, 2023	Release 5.2(7) became available.

## New Software Features

Feature	Description
Rancher Kubernetes Engine (RKE) 1.3.21, 1.4.6	Cisco ACI supports RKE 1.3.21, 1.4.6 installed cluster integrated with ACI CNI.
OpenShift 4.12 on OpenStack 16.2	Cisco ACI supports Red Hat OpenShift 4.12 nested in Red Hat (OSP) 16.2. To enable this support, Cisco ACI provides customized Ansible modules to complement the upstream OpenShift installer.
OpenShift 4.12 on Bare Metal	Cisco ACI supports Red Hat OpenShift 4.12 on Bare metal with User Provisioned Infrastructure (UPI) method of installation. Cisco ACI provides customized Python script to complement the upstream OpenShift installer for integration with the ACI CNI.
OpenShift 4.12 on VMware vSphere	Cisco ACI supports Red Hat OpenShift 4.12 nested in VMware vSphere 7. Cisco ACI provides customized Ansible modules as reference to complement the upstream OpenShift installer for integration with the ACI CNI.

The following features are being released as technology preview to gather feedback and should not be used in production:

Feature	Description
IPv4/IPv6 dual-stack	IPv4/IPv6 dual-stack networking enables the allocation of both IPv4 and IPv6 addresses to pods and services with the ACI CNI.
Firewall Insertion	Insertion of a user managed firewall is supported with the ACI CNI by allowing insertion in the ACI service graph which models the ingress path of the Kubernetes external loadbalancer service.
ACI Multi-Pod	Live migration of virtual machines between different ACI Pods in the ACI Multi-Pod setup is supported with ACI CNI.

## Cisco ACI Virtualization Compatibility Matrix

For information about all Cisco ACI-supported (currently-supported) Container products along with the supported ecosystem releases (Kubernetes, OpenShift, OpenStack, Rancher, and vSphere), see the Cisco ACI Virtualization Compatibility Matrix at the following URL:

<https://www.cisco.com/c/dam/en/us/td/docs/Website/datacenter/aci/virtualization/matrix/virtmatrix.html>

## Software

### Installers:

OpenShift installer source scripts for OCP 4.10, 4.11, and 4.12 on OpenStack 16.2 are provided as releases artifacts.

For OpenShift on Baremetal, OpenStack and vSphere please refer to the [Install Guides](#).

### Configuration:

This release uses the acc-provision upstream open source project release [5.2.7.1](#) for ACI-CNI or Calico deployment manifest generation and ACI fabric configuration.

Download acc-provision from pypi:

[https://files.pythonhosted.org/packages/06/03/ed5fc793664f9b33acb72e5e3519ab34ab8f90a8d6e57bff49a68be4b77f/acc\\_provision-5.2.7.1.tar.gz](https://files.pythonhosted.org/packages/06/03/ed5fc793664f9b33acb72e5e3519ab34ab8f90a8d6e57bff49a68be4b77f/acc_provision-5.2.7.1.tar.gz)

Verify digest:

Algorithm	Hash digest
SHA256	419378061f5781bf1ebf30bab40b33a8601a4f1d9a2cc0427c7877e7e4e106e1
MD5	4af7e49d2dc48d2e97a98b7088ea17b4
BLAKE2b-256	0603ed5fc793664f9b33acb72e5e3519ab34ab8f90a8d6e57bff49a68be4b77f

Installation (needs Python 3.9):

```
python3 -m pip install ./acc_provision-5.2.7.1.tar.gz
```

Usage:

The release artifacts, `aci-cni-docker-images.yaml` or the `aci-cni-quay-images.yaml` should be used as part of the `acc-provision` input configuration.

For details on how to use `acc-provision` refer to:

[Provisioning Cisco ACI to Work with Kubernetes](#)

[Generating an Updated Cisco ACI CNI Configuration](#)

### Container Images:

`acc-provision` generates deployment manifests which reference ACI-CNI container images built in upstream open source projects [aci-containers](#), [opflex](#) and [acc-provision-operator](#) with tag 5.2.7.1. Package and vulnerability details of these images are documented in the `acc-provision` project [release notes](#).

### Supported Scale

For the verified scalability limits (except for CLI limits), see the Verified Scalability Guide for this release. For Kubernetes-based Integrations (including Docker, OpenShift, and Rancher), and OpenStack Platform Scale Limits, see the following table.

**Note:** The scalability information in the following table applies to Kubernetes or OpenStack resources integrated with OpFlex into the Cisco ACI fabric. It does not apply to Microsoft SCVMM hosts or Cisco ACI Virtual Edge instances.

Limit Type	Maximum Supported
Number of OpFlex hosts per leaf <sup>1</sup>	120
Number of OpFlex hosts per port	20
Number of vPC links per leaf	40
Number of endpoints per leaf	10,000
Number of endpoints per host	400
Number of virtual endpoints per leaf	40,000

<sup>1</sup> The indicated scale value is for Cisco ACI version 5.0(1) and later. If the ACI version is less than 5.0(1), the number of supported OpFlex hosts are 40.

### Notes:

- For containers, an endpoint corresponds to a pod’s network interface. The number of pods that can be run on each node is however constrained by other system configuration and Kubernetes distribution specified limits. For kubeadm installed upstream Kubernetes its 110 pods per node, and for OpenShift its 250 pods per node.
- For OpFlex hosts per port — a port is either a physical port or a vPC. One vPC equals one port. The number of member ports in a vPC is inconsequential.
- For the CLI verified scalability limits, see the *Cisco NX-OS Style Command-Line Interface Configuration Guide* for this release.

---

## Known Limitations

- The NodePort service statistics exported to Prometheus get accounted under ClusterIp service statistics in on-premise deployments.
- A pod selector has to be always provided to map a port name to the port number, and an empty pod selector is not supported in the ingress direction.
- The Cisco ACI CNI Plug-in is not integrated with the Multi-Site Orchestrator. When deploying to a Multi-Site deployment, the Cisco ACI configurations implemented by the plug-in must not be affected by the Multi-Site Orchestrator.
- SNAT policy configuration is not applicable to traffic within the same cluster.
- An SNAT policy which goes into the *Failed* state (for example, on account of reusing an already used SNAT IP), cannot be updated or reused. A failed SNAT policy needs to be deleted and a new one created.
- Due to Python 3 dependencies that are currently available only on RHEL8, acc-provision tool is supported on RHEL8 operating system, but not on RHEL7 operating system.
- The file `openvswitch/db.sock` sometimes becomes a directory after node reload due to a race-condition between the `openvswitch` installed on the node and `openvswitch` installed by ACI-CNI. The work around is to delete the `/var/run/openvswitch/db.sock` directory, and restart the `aci-containers-openvswitch` pod. For more details, see *Red Hat Case 03299085*.

## Usage Guidelines

- Note that upgrading a Cisco ACI CNI cluster requires running `acc-provision` with the `"--upgrade"` option.
- Optimizations to mapping of Kubernetes Network Policy to ACI Host Protection Policies can be turned ON with the following configuration:

```
kube_config:
...
    hpp_optimization: True
```

This, and all other configuration changes should be performed using the `acc-provision` tool, and will take effect after the new manifests generated by `acc-provision` are applied. This configuration will be enabled by default in future releases.

- For running more than 250 pods per node, the following configuration needs to be added:

```
kube_config:
...
    opflex_agent_ovs_asyncjson_enabled: "true"
```

This is a preview feature. Note the configuration value is a string in quotation marks.

- The `aci-containers-operator` uses the Ansible Operator SDK. If another Kubernetes Operator which uses the Ansible Operator SDK is deployed on the same node, the health-check ports of the two Operators will conflict. There is currently no way to override these default ports either. To overcome this issue, the `aci-containers-operator` pod has node affinity rule for `"preferredDuringSchedulingIgnoredDuringExecutionfor"` with key `"preferred-node"` and value `"aci-`

---

containers-operator-2577247291". You can ensure that the aci-containers-operator is scheduled on a particular node by adding the following label to the node:

```
preferred-node=aci-containers-operator-2577247291
```

A similar affinity scheme should be applied to other conflicting pods to ensure that they do not get scheduled on the above node.

Note that if no node with the above label exists, then, the aci-containers-operator will still get scheduled on some node.

- The size of each log file collected in the cluster report can be optionally set using the following acc-provision input configuration (default is 10 MB):

```
logging:
  size <size-in-bytes>
```

Note that the truncation happens at the beginning and the latest content of the log file is collected.

- Sometimes it takes longer for service endpoints to be ready but since they are configured are successfully configured as endpoints of that service, traffic will start get loadbalanced to these endpoints and may get temporarily blackholed. To avoid this, a delay along with the details of the services of type Loadbalancer can be specified in the acc-provision input file, such that the ACI service graph will be programmed with a delay. The following example shows a delay of 30 seconds being introduced for ingress-service (belonging to openshift-ingress) and a delay pf 60 seconds for monitoring-service (belonging to openshift-monitoring):

```
kube_config:
...
  service_graph_endpoint_add_delay:
    delay: 30
  services:
    - name: ingress-service
      namespace: openshift-ingress
    - name: monitoring-service
      namespace: openshift-monitoring
      delay:60          #override delay of 30
```

Note that endpoints are added to the service graph only after the pod goes into Ready state.

- To enable drop logging, perform the following configuration in the acc-provision input file:

```
drop_log_config:
  enable: True
```

For more information, see [Enabling the OpFlex Drop Log Feature](#).

- The scope of the SNAT service graph contract can be configured by the user in the acc-provision input file as follows:

```
kube_config:
  snat_operator:
    contract_scope: <scope name>
```

---

Valid values (as allowed by Cisco APIC) are "global", "tenant" and "context". The default is set to "global".

- The subnets listed under `extern_static` and `extern_dynamic` can be automatically added to `rdconfig` usersubnets by setting the following configuration in `acc-provision` input file:

```
kube_config:
  add_external_subnets_to_rdconfig: True
```

Note that if the initial value of `add_external_subnets_to_rdconfig` is `true` but later modified to `false`, the usersubnets automatically will not be removed and the `rdconfig` CR will have to be updated manually to remove them. Each entry in the `rdconfig` results in a new OVS flow regardless of whether the subnets overlap or not.

- The `aci-containers-controller` pod subscribes for notifications on certain objects to the Cisco APIC. There is a timeout associated with this subscription. A shorter timeout requires more frequent subscription renewals. The timeout is set to 900 seconds, and can be changed by configuring the `acc-provision` input file:

```
aci_config:
  apic_refresh_time: 1200
```

**Note:** The subscription timeout is configurable only in Cisco APIC 4.x or later.

- To ensure that the subscription renewal happens in time before the subscription timeout expires on the APIC side, the `aci-containers-controller` pod starts the renewal process a little earlier. By default, it starts 150 seconds before the subscription expiry. If the system is heavily loaded and you notice subscriptions are not renewed in time (this requires examining the `aci-containers-controller` and Nginx APIC logs), this period can be altered by adjusting the following configuration in the `acc-provision` input file:

```
aci_config:
  apic_refresh_ticker_adjust: 150
```

- The memory request and limit for the Open vSwitch container is set to 128Mi and 1Gi respectively. It can be changed by configuring the `acc-provision` input file as follows:

```
kube_config:
  ovs_memory_limit: "5Gi"
  ovs_memory_request: "512Mi"
```

- The Multus CNI deployment can be enabled in the OpenShift installation by performing the following configuration in the `acc-provision` input file:

```
multus:
  disable: False
```

- The memory request and limit for the Open vSwitch container is set to 128Mi and 1Gi respectively. It can be changed by configuring the `acc-provision` input file as follows:

```
kube_config:
  ovs_memory_limit: "5Gi"
  ovs_memory_request: "512Mi"
```

- Default memory request and limit for `aci-containers-system` namespace pods is set to 128Mi and 3Gi respectively and can be changed by configuring the `acc-provision` input file as follows.

```
kube_config:
  aci_containers_memory_request: "512Mi"
  aci_containers_memory_limit: "5Gi"
```

**Note:** This namespace wide memory resource setting is not applied to openvswitch container. As stated earlier, please use `ovs_memory_request`, `ovs_memory_limit` to change openvswitch container memory request and limit.

Apart from above option to set namespace wide memory request and limit, you can choose to configure container specific memory request and limit through `acc-provision` input file as follows:

```
kube_config:
  aci_containers_controller_memory_request: "256Mi"
  aci_containers_controller_memory_limit: "5Gi"
  aci_containers_host_memory_request: "256Mi"
  aci_containers_host_memory_limit: "5Gi"
  mcast_daemon_memory_request: "256Mi"
  mcast_daemon_memory_limit: "5Gi"
  opflex_agent_memory_request: "256Mi"
  opflex_agent_memory_limit: "5Gi"
  acc_provision_operator_memory_request: "256Mi"
  acc_provision_operator_memory_limit: "5Gi"
  aci_containers_operator_memory_request: "256Mi"
  aci_containers_operator_memory_limit: "5Gi"
```

This container specific configuration takes priority over the namespace wide configuration.

- ACI CNI pods are critical and to mark them node critical, `PriorityClass "system-node-critical"` can be set by configuring the `acc-provision` input file as follows:

```
kube_config:
  use_system_node_priority_class: True
```

- To support ACI multi-pod VM migration, the following configuration should be given in the `acc-provision` input file:

```
kube_config:
  aci_multipod: True
```

For an RKE-1 on Ubuntu the following additional configuration should be set in the `acc-provision` input file::

```
kube_config:
  aci_multipod_ubuntu: True
```



- After moving a VM across ACI pods, the DHCP needs to be renewed on the infra-vlan. The ACI-CNI initiates this DHCP release and request. To reliably obtain the DHCP, sometimes multiple attempts may be required. By default, this is attempted five times each after a delay of five seconds. These defaults can be tuned using the following configuration in the acc-provision input file:

```
kube_config:
  dhcp_renew_max_retry_count: 4
  dhcp_delay: 6
```

- Dualstack can be configured by providing the IPv4 and IPv6 in the acc-provision input configuration file as follows:

```
net_config:
  node_subnet:
    - <ipv4-subnet>
    - <ipv6-subnet>
  pod_subnet:
    - <ipv4-subnet>
    - <ipv6-subnet>
  extern_dynamic:
    - <ipv4-subnet>
    - <ipv6-subnet>
  extern_static:
    - <ipv4-subnet>
    - <ipv6-subnet >
```

- In cases of heavy load, the opflex-agent requests to the leaf switch may fail and the opflex-agent needs to retry after a randomized backoff. The upper bound on this backoff can be configured to adapt specific load conditions to avoid frequent retries:

```
kube_config:
  opflex_agent_policy_retry_delay_timer: 60 # default is 10 seconds
```

- For the VMware VDS integration, you can refer to the Enhanced Link Aggregation Group (eLAG) configured through the Cisco APIC by using the following configuration in the acc-provision input file:

```
nested_inside:
  type: vmware
...
  elag_name: <eLAG-name-used>
```

- User update to include a firewall node in the loadbalancer service graph template are now supported. The following annotation should be added to the corresponding Kubernetes loadbalancer type service:

```
opflex.cisco.com/service-graph-name: <some-value>
```

- Policy Based Routing (PBR) tracking can be enabled for the Cisco APIC service graph created for supporting the SNAT feature. More details on PBR tracking can be found in the chapter "Configuring Policy-Based Redirect" In the Cisco APIC Layer 4 to Layer 7 Services Deployment Guide, Release 5.2(x).

---

One HealthGroup for each node is created, and it is associated with the redirect policy of the SNAT service graph with the internet protocol service level agreement (IP SLA) interval set to 5 seconds. This interval is configurable through the acc- provision input file:

```
net_config:
    service_monitor_interval: 10
```

If the service\_monitor\_interval is set to zero, PBR tracking is disabled.

PBR tracking can be also be enabled for other Cisco APIC service graphs created for each Kubernetes external service, setting the following configuration in the acc-provision input file:

```
net_config:
    pbr_tracking_non_snat: true
```

If enabled, the service\_monitoring\_interval described earlier applies here as well.

**Note:** In a Cisco ACI CNI-based cluster, the same worker node is used to provide both the external Layer 4 load balancer and SNAT services. So if PBR tracking is enabled, and if the worker node reports unhealthy status for SNAT, a fault appears in the redirect policies associated with all other (non-SNAT) service graphs that have this node. However, this fault does not actually affect those other services and traffic from those services is still distributed to that node. The fault manifests for those other services only in the Cisco APIC GUI.

- Starting with Cisco APIC Release 5.2(1), a fault (vmmClusterFaultInfo) is generated in ACI, if a Kubernetes namespace, deployment, or pod is annotated with an EPG name that does not resolve to an existing EPG. A log statement is added in the aci-containers-controller log to alert the user. The fault will be cleared upon the next correct annotation, or when the aci-containers-controller restarts, or when the annotated namespace, deployment, or pod is deleted.
- You should be familiar with installing and using Kubernetes or OpenShift. Cisco ACI does not provide the Kubernetes or OpenShift installer. Refer to the following documents on Cisco.com for details:
  - [Cisco ACI and Kubernetes Integration](#)
  - [OpenShift Install Guides](#)
  - [Cisco ACI CNI Plugin for Red Hat OpenShift Container Platform Architecture and Design Guide](#)
  - [Upgrading the Cisco ACI CNI Plug-in](#)
  - [Cisco ACI and Calico 3.23.2 Integration](#)
- OpenShift has a tighter security model by default, and many off-the-shelf Kubernetes applications, such as guestbook, may not run on OpenShift (if, for example, they run as root or open privileged ports like 80).
- Refer to the article *Getting any Docker image running in your own OpenShift cluster* on the Red Hat OpenShift website for details. The Cisco ACI CNI Plug-in is not aware of any configuration on OpenShift cluster or pods when it comes to working behind a proxy. Running OpenShift "oc new-app", for instance, may require access to Git Hub, and if the proxy settings on the OpenShift cluster are not correctly set, this access may fail. Ensure your proxy settings are correctly set.
- In this release, the maximum supported number of PBR based external services is 250 virtual IP addresses (VIPs). Scalability is expected to increase in upcoming releases.

**Note:** With OpenShift, master nodes and router nodes are tainted by default, and you might see lower scale than an upstream Kubernetes installation on the same hardware.

- Some deployments require installation of an "allow" entry in IP Tables for IGMP. This must be added to all hosts running an OpFlex agent and using VXLAN encapsulation to the leaf. The rule must be added using the following command:

```
$ iptables -A INPUT -p igmp -j ACCEPT
```

In order to make this change persistent across reboots, add the command either to `/etc/rc.d/rc.local` or to a cron job that runs after reboot.

- Both RHEL and Ubuntu distributions set `net.ipv4.igmp_max_memberships` set to 20 by default. This limits the number of end point groups (EPGs) that can be used in addition to the kube-default EPG for pod networking. If you anticipate using more than 20 EPGs, set the value to the desired number of EPGs on each node as follows:

```
$ sysctl net.ipv4.igmp_max_memberships=desired_number_of_epgs
```

## Open Issues

There are no open issues in this release.

## Resolved Issues

Click the bug ID to access the Bug Search tool and see additional information about the bug.

Bug ID	Description
<a href="#">CSCwa23407</a>	Openshift VM live migration between different Pods in the ACI Multi-Pod setup is not supported.
<a href="#">CSCwd57920</a>	Acc-provision sets annotation on fvTenant for pre-existing Tenant.
<a href="#">CSCwf64555</a>	Acc-provision does not set the port-group uplink to eLag when configured in nested VMware setup.

## Known Issues

Bug ID	Description
<a href="#">CSCwa36696</a>	Temporary loss of K8s Pods cause 30-60 seconds traffic drops due to ACI objects re-deploy.

## Related Content

See the [Cisco Application Policy Infrastructure Controller \(APIC\)](#) page for the documentation.

The documentation includes installation, upgrade, configuration, programming, and troubleshooting guides, technical references, release notes, and knowledge base (KB) articles, as well as other documentation. KB articles provide information about a specific use case or a specific topic.

By using the "Choose a topic" and "Choose a document type" fields of the APIC documentation website, you can narrow down the displayed documentation list to make it easier to find the desired document.

You can watch videos that demonstrate how to perform specific tasks in the Cisco APIC on the [Cisco Data Center Networking](#) YouTube channel.

---

## Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, send your comments to [apic-docfeedback@cisco.com](mailto:apic-docfeedback@cisco.com). We appreciate your feedback.

### Legal Information

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL:

<http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

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.

© 2023 Cisco Systems, Inc. All rights reserved.