Cisco Container Platform 8.0.0 Release Notes

First Published: 2020-12-18

Introduction

Cisco Container Platform is a fully curated, lightweight container management platform for production-grade environments, powered by Kubernetes, and delivered with Cisco enterprise-class support. It reduces the complexity of configuring, deploying, securing, scaling, and managing containers using automation along with Cisco's best practices for security and networking. Cisco Container Platform is built with an open architecture using open source components.

Features

Feature	Description
Kubernetes Lifecycle Management	Enables you to deploy Kubernetes clusters, add or removed nodes, and upgrade Kubernetes clusters to latest versions.
Persistent Storage	Allows you to persist data for containerized applications between upgrades and updates through HyperFlex storage driver.
Monitoring and Logging	Provides dashboards, alerts, and indexing to monitor resource usage and behavior of platform components through Elasticsearch, Fluentd, and Kibana (EFK) stack and Prometheus.
Container Networking	Provides container to container and container to non-containerized application layers communication with security policies.
Load Balancing	Offers software Ingress load balancing through NGINX and node port functionality of Kubernetes for containerized applications.
Role Based Access Control	Integrates with Active Directory and offers permission-based rules.
Multi-GPU Support	Optimized for AI/ML workloads with multi-GPU support.

Revision History

Release	Date	Description
1.0	May 22, 2018	First release
1.0.1	May 25, 2018	Updated the **Fixed Issues**and **Know Issues** sections
1.1.0	June 29, 2018	Added the **What's New** and **Upgrading Cisco Container Platform** sections Updated the **Fixed Issues**and **Know Issues** sections
1.4.0	July 31, 2018	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
1.4.1	August 6, 2018	Added the **Fixed Issues, 1.4.1** section
1.5.0	September 6, 2018	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
2.0.1	October 15, 2018	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
2.1.0	November 1, 2018	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
2.1.1	December 6, 2018	Added the **Fixed Issues, 2.1.1** section
2.2.2	December 13, 2018	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
3.0.0	February 7, 2019	Updated the **System Requirements**, **Fixed Issues**, **Known Issues**, and **What's New** sections
3.0.1	February 21, 2019	Updated the **Fixed Issues** section
3.1.0	March 20, 2019	Updated the **System Requirements**, **Fixed Issues**, **Known Issues**, and **What's New** sections
3.2.0	April 29, 2019	Updated the **System Requirements**, **Fixed Issues**, **Known Issues**, and **What's New** sections

Release	Date	Description
3.2.1	May 6, 2019	Updated the **Fixed Issues** and **Known Issues** sections
4.0.0	June 11, 2019	Updated the **System Requirements**, **What's New**, **Fixed Issues**, and **Known Issues** sections
4.0.1	July 3, 2019	Updated the **What's New** and **Known Issues** sections
4.1.0	July 25, 2019	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
4.2.0	August 29, 2019	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
4.2.1	September 27, 2019	Updated the **What's New** and **Fixed Issues** sections
5.0.0	October 22, 2019	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections Included a new section on **Feature Compatibility Matrix**
5.1.0	December 16, 2019	Updated the **System Requirements**, **What's New**, **Feature Compatibility Matrix**, **Fixed Issues**, and **Known Issues** sections
6.0.0	March 10, 2020	Updated the **Feature Compatibility Matrix**, **What's New**, **Fixed Issues**, and **Known Issues** sections
6.1.0	April 24, 2020	Updated the **Feature Compatibility Matrix**, **What's New**, and **Known Issues** sections
6.1.1	July 1, 2020	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
7.0.0	September 11, 2020	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections
8.0.0	December 18, 2020	Updated the **What's New**, **Fixed Issues**, and **Known Issues** sections

System Requirements

- Cisco Container Platform installer OVA
- Latest two versions of the tenant OVA
- vCenter 6.7 Update 2 or later
- vCenter cluster with High Availability (HA) and Distributed Resource Scheduler (DRS) enabled
- A DHCP server that provides IP addresses to the Cisco Container Platform installer VMs
- A shared datastore that is mounted on all the ESXi hosts in the cluster
- Cisco Container Platform Control Plane VMs need to have network access to the vCenter appliance API
- Kubectl version within one minor version of target Kubernetes cluster
- HyperFlex version 4.0+ is required to use the HyperFlex Container Storage Interface (CSI) storage plugin

Feature Compatibility Matrix

Function	Component	Validated Version
Container runtime	Docker	19.03.13
Operating system	Ubuntu	18.04
Orchestration	Kubernetes (on-prem)	1.17.14, 1.18.12
Orchestration	Kubernetes (EKS)	1.15, 1.16
Orchestration	Kubernetes (AKS)	1.16.15, 1.17.13
Orchestration	Kubernetes (GKE)	Latest stable version in the release channel of GKE
IaaS (pre-req)	vSphere	6.7
Infrastructure	HyperFlex UCS	HyperFlex: 4.0.1a+ UCS: Any version that supports VMware
CNI	ACI, Calico	ACI: 4.2.2.2 Calico: 3.16.1
SDN	ACI	4.1+
Container storage	CSI Driver	1.0.1
L7 load balancing	Nginx (community) Ingress	0.33.0
Monitoring	Prometheus Grafana	Prometheus: 2.17.1 Grafana: 6.7.4

Function	Component	Validated Version
Logging	EFK	6.8.8
L3 load balancing	MetalLB	0.8.3
Service mesh	Istio/Envoy	1.6.9
Registry	Harbor	2.1.0
Machine Learning	Kubeflow	1.0.0

What's New

- Support for Kubernetes 1.18 and 1.17
- Support for Kubeflow 1.0.0
- Support for specification of overrides when creating and patching add-ons
- Deprecated support for Istio in v2 tenant clusters

Installing Cisco Container Platform

For step by step instructions on installing Cisco Container Platform, refer to the *Cisco Container Platform Installation Guide*.

Upgrading Cisco Container Platform

- Upgrading Cisco Container Platform control plane is only supported from the 5.0.0 release for deployments using Calico or ACI CNI.
  - Upgrading tenant clusters from Cisco Container Platform 4.0.0 that use Kubernetes 1.13 to tenant clusters in Cisco Container Platform 6.0 that use Kubernetes 1.15 is supported.
  - Upgrading tenant clusters from Cisco Container Platform 4.0+ or 5.x that use Kubernetes 1.14 to tenant clusters in Cisco Container Platform 6.0 that use Kubernetes 1.16 is supported.
  - Upgrading Kubernetes 1.13 to 1.16 is not supported.
- If an existing deployment uses Contiv for CNI, then upgrades to the current version are not supported.
- When you upgrade Cisco Container Platform, the cert-manager API will be migrated from `certmanager.k8s.io/v1alpha1` to `cert-manager.io/v1`. Due to the namespace change, existing cert-manager resources will be migrated to a new version. You can find a backup of the previous cert-manager resources on the first control plane node that is upgraded, in the `/opt/ccp/manifests` directory.

Backing Up and Restoring Cisco Container Platform Data

The IP addresses required for the new Control Plane must be from the original IP address pool range of the Control Plane that was created during installation. If this is not possible, you must open a support case for assistance in creating a complete backup.

Fixed Issues

- Fixed Cisco Container Platform web interface bugs
- Included security updates for the containers
- Improved logging of events in Cisco Container Platform
- Improved user experience for session timeout in Cisco Container Platform

Known Issues

- During the upgrade of a multi-master vSphere or Openstack tenant cluster, the upgrade fails because the etcd leader is reported missing.

  The following error is displayed in the ****cloud-init**** logs of one of the master nodes:

  ```
  Error: etcdserver: leader changed
  ```

  ****Workaround****

  Before upgrading a multi-master tenant cluster, you need to ensure that the etcd leader is available. For more information on the scripts to be run to resolve this issue, see Multi-master vSphere or Openstack Tenant Cluster Fails to Upgrade.

- Cisco Container Platform installation fails if you configure the 172.17.0.0/16 subnet as your pod CIDR.

  ****Workaround****

  This subnet is not available because Docker already uses this subnet. You must configure Cisco Container Platform to use a different subnet as your pod CIDR.

- Deployment of the `vsphere-csi-controller` pod fails with a **CrashLoopBackOff** status because of special characters in the `csi-vsphere.conf` file.

  Workaround

  1. View the logs for the vsphere-csi-controller pod.

     ```
     kubectl logs vsphere-csi-controller-577f56865c-sx28h -c vsphere-csi-controller
     -n kube-system
     ```

  2. Check for a log record with the following details:

     ```
     {"level":"error","time":"2020-12-17T03:15:58.681282472Z","caller":"config/config.go:301","msg":"error
      while reading config file: 8:27: unknown escape
     sequence","TraceId":"0d9ed568-a202-40b2-80ad-173febdde0d4",...
     ```

     Note This log record indicates that the `vsphere-csi-controller` pod has encountered an error in the `csi-vsphere.conf` file, which is stored in the `kube-system/vsphere-config-secret` file. You must verify the contents of the file and ensure that none of the assigned values have escape characters or delimiters as part of the string values. The character `"` is a delimiter for string values and character `\` is the escape sequence character. If these characters are used for any values in the `csi-vsphere.conf` file, the `vsphere-csi-controller` pod will fail when reading the file.

3. Retrieve the contents of the `csi-vsphere.conf` file.

```
    $ kubectl get secrets vsphere-config-secret -n kube-system -o json | jq -r
'.data."csi-vsphere.conf"' | base64 -d > csi-vsphere.conf
```

Sample of an invalid csi-vsphere.conf file

```
[Global]
cluster-id = "cluster001"
insecure-flag = true
[VirtualCenter "vcenter01.example.com"]
port = 443
datacenters = DC01
user = "user@vsphere.local"
password = "admin"pass"\!23" # <-- invalid value
```

In this example, the password field contains invalid characters `"` and `\`. Although this password is valid in vCenter, `vsphere-csi-controller` cannot interpret these values correctly and the deployment of `vsphere-csi-controller` will fail.

4. Update the `csi-vsphere.conf` file.

Example of a valid csi-vsphere.conf file

```
[Global]
cluster-id = "cluster001"
insecure-flag = true
[VirtualCenter "vcenter01.example.com"]
port = 443
datacenters = DC01
user = "user@vsphere.local"
password = "admin\"pass\"\\!23" # <-- corrected password
```

5. Update the secret `kube-system/vsphere-config-secret`.

```
    $ kubectl delete secret vsphere-config-secret -n kube-system
    $ kubectl create secret generic vsphere-config-secret -n kube-system
--from-file=csi-vsphere.conf=csi-vsphere.conf
    secret/vsphere-config-secret created
```

6. After the secret is successfully created, restart the vsphere-csi-controller pods to use the latest values from the secret.

```
    $ kubectl delete pods -l app=vsphere-csi-controller -n kube-system
```

7. Wait until the pods are deleted completely. The new pods will be automatically restarted.

- If you have EKS clusters in Cisco Container Platform 6.1.1, upgrading Cisco Container Platform from 6.1.1 to 8.0.0 fails because of the change in the Kubernetes versions that are supported.

  Cisco Container Platform 6.1.1 supports Kubernetes versions 1.13 and 1.14, while Cisco Container Platform 8.0.0 supports Kubernetes versions 1.15 and 1.16. EKS upstream does not allow a direct upgrade from Kubernetes version 1.14 to 1.16.

  Workaround

  You need to first upgrade Cisco Container Platform 6.1.1 to 7.0.0, where all EKS clusters are upgraded to Kubernetes version 1.15, and then upgrade from Cisco Container Platform 7.0.0 to 8.0.0, where all the EKS clusters are upgraded to Kubernetes version 1.16.

- Istio is no longer supported in v2 tenant clusters. If Istio is installed in a v2 cluster and you have upgraded to the latest version of Cisco Container Platform, it is required to manually delete Istio from the v2 tenant cluster using helm.
- Cluster upgrades may fail due to a failed Kubeflow 0.7 installation.

  Workaround

  Before upgrading, ensure that you uninstall Kubeflow webhooks from your cluster:

  ```
  $ kubectl delete MutatingWebhookConfiguration inferenceservice.serving.kubeflow.org

  $ kubectl delete ValidatingWebhookConfiguration inferenceservice.serving.kubeflow.org
  ```

- Cluster creation fails for AKS clusters because of an invalid Kubernetes version.

  When you deploy Cisco Container Platform, your deployment is configured to use certain AKS Kubernetes versions. If AKS decommissions support for a Kubernetes version that is used in a Cisco Container Platform deployment, then cluster creation fails.

  Workaround

  1. Determine the supported Kubernetes versions from the following AKS website:

     https://aka.ms/supported-version-list

  2. Create an SSH connection to the master VM of the Cisco Container Platform Control Plane.
  3. Take a backup of the `ccp-k8s-version` ConfigMap.

     ```
     kubectl get cofigmap ccp-k8s-version -n ccp -o yaml > ccp-k8s-versions.yaml
     ```

  4. Patch ConfigMap to allow the creation of new AKS clusters for the newly supported Kubernetes versions.

     ```
     kubectl patch configmap ccp-k8s-versions -n ccp -p '{"data": {"AKS_K8S_VERSIONS":
     "1.16.15,1.17.11"}}'
     ```

  5. Patch ConfigMap to include a valid upgrade path for the existing AKS clusters.

     ```
     kubectl patch configmap ccp-k8s-versions -n ccp -p '{"data":
     {"AKS_K8S_UPGRADE_MAP": '{"1.15.8":["1.16.15"],"1.16.15":["1.17.11"]}'}}'
     ```

     For example, let us say that your Cisco Container Platform deployment uses an AKS cluster of version 1.15.8. After you patch the ConfigMap, you will have the support to upgrade the AKS cluster version from 1.15.8 to 1.16.15.

  6. Restart the pods to ensure that the changes in the ConfigMap are registered.

     ```
     kubectl delete pods -n ccp --selector=app=ccp-aks-operator
     kubectl delete pods -n ccp --selector=app=api
     kubectl delete pods -n ccp --selector=app=kaas-dashboard
     ```

     The changes will take effect after the pods are in the **Running** state.

- HyperFlex Data Platform HX Connect UI fails to display v3 clusters.

  Workaround

  Log in to Cisco Container Platform to manage your v3 clusters.

- In Cisco Container Platform versions earlier than 7.0.0, scaling to more than seven v3 clusters can cause new tenant creation or other tenant cluster management requests to fail.

Workaround

Edit the ccp-net-tinker kubernetes resource on a Cisco Container Platform master node with kubectl.

Run `kubectl edit deploy ccp-net-tinker`, updating the following settings:

```
* .spec.template.spec.containers.0.resources.limits.cpu: 600m
* .spec.template.spec.containers.0.resources.limits.memory: 600Mi
* .spec.template.spec.containers.1.resources.limits.cpu: 500m
* .spec.template.spec.containers.1.resources.limits.memory: 512Mi
```

The reconciliation of tenant clusters should complete.

- Cluster provisioning fails when you select an invalid combination of GKE zone and machine type.

  Workaround

  Choose a machine type that is available in your zone of choice. For more information, see Available Regions and zones.

- Logging add-ons do not work after upgrading a tenant cluster created on Cisco Container Platform 6.0.0.

  Workaround

  Uninstall and reinstall the add-ons.

- Installation of Kubeflow 0.7 as an add-on to Cisco Container Platform fails because of some issues with the upstream Kubeflow version 0.7.

  Workaround

  You can manually install Kubeflow 1.0 on a tenant cluster.

  Follow these steps:

  1. Install Kubeflow 1.0 on the tenant master node.

     ```
     export KF_APP="kf-app"
     export
KFDEF_URL="https://raw.githubusercontent.com/kubeflow/manifests/v1.0.0/kfdef/kfctl_k8s_istio.v1.0.0.yaml"

     export
KFCTL_URL="https://github.com/kubeflow/kfctl/releases/download/v1.0/kfctl_v1.0-0-g94c35cf_linux.tar.gz"

     mkdir -p ${KF_APP}
     cd ${KF_APP}
     wget -O kfctl.tar.gz ${KFCTL_URL}
     tar -zxvf kfctl.tar.gz
     chmod +x kfctl
     wget -O kfctl_k8s_istio.yaml ${KFDEF_URL}
     ./kfctl apply -V -f kfctl_k8s_istio.yaml
     ```

  2. Verify that all Kubeflow pods in the `kubeflow`, `knative-serving` and `istio-system` namespaces are up and running.

     ```
     kubectl get pods -A
     ```

  3. Get the Kubeflow URL.

     ```
     export INGRESS_PORT=$(kubectl -n istio-system get service istio-ingressgateway
 -o jsonpath='{.spec.ports[?(@.name=="http2")].nodePort}')
     export INGRESS_HOST=$(kubectl get po -l istio=ingressgateway -n istio-system
-o jsonpath={.items[0].status.hostIP})
     echo "http://$INGRESS_HOST:$INGRESS_PORT"
     ```

- If you are using the Hyperflex Container Storage Interface (CSI), and you reboot the Kubernetes nodes, the provisioned Persistent Volume (PV) may be unusable after the reboot.

  Workaround

  1. Login to the Hyperflex manager dashboard.

     The Hyperflex manager dashboard appears.

  2. Go to **Settings** > **Integrations**, and enable Kubernetes.

  3. SSH into each hyperflex storage controller and run the following commands:

     ```
     sed -ie "s/iscsiEnable=false/iscsiEnable=true/" /etc/init/scvmclient.conf
            initctl reload-configuration
            stop scvmclient; start scvmclient
            initctl emit --no-wait system-datastore-created
     ```

  4. Wait for 10 minutes, and then restart the pods.

     The HyperFlex CSI PV is usable now.

- Using the Kubeconfig file to access the Kubernetes dashboard fails.

  The method of using the Kubeconfig file to access the Kubernetes dashboard, as described in Accessing Kubernetes Clusters on vSphere, is not working for V3 vSphere and OpenStack tenant clusters.

  Workaround

  Use the Kubernetes default token to access the dashboard, as described in Accessing Kubernetes Clusters on vSphere.

- If you are using the EFK logging add-on, you may find the Fluentd pods consuming more than 30% of the CPU cycles on worker nodes, and the logs filling with trailing backslashes. This is a known issue with Fluentd [Issue 2545].

  Workaround

  You must change the following configurations on the Fluentd deployment:

  1. Edit the Fluentd Configmap.

     a. Run the following command on the tenant master node:

        ```
        $ kubectl edit cm -n ccp fluentd-es-config-v0.1.1
        ```

     b. Find path `/var/log/containers/*.log` in the source and add a line for exclude_path as shown in the following code snippet:

        ```
        <source>
        type tail
        path /var/log/containers/*.log
        pos_file /var/log/es-containers.log.pos
        time_format %Y-%m-%dT%H:%M:%S.%NZ
        tag kubernetes.*
       exclude_path "#{ENV['FLUENT_CONTAINER_TAIL_EXCLUDE_PATH'] || use_default}"
        # Add this line
        read_from_head true
        format multi_format
        <pattern>
            format json
            time_key time
            time_format %Y-%m-%dT%H:%M:%S.%NZ
        </pattern>
        ```

```
<pattern>
     format /^(?<time>.+) (?<stream>stdout|stderr) [^ ]* (?<log>.*)$/
     time_format %Y-%m-%dT%H:%M:%S.%N%:z
</pattern>
</source>
```

2. Edit the Fluentd DaemonSet.
   a. Run the following command on the tenant master node:

      ```
      $ kubectl edit ds -n ccp fluentd-es-v2.0.2
      ```

   b. Find the environment variables section in the source, and add `FLUENT_CONTAINER_TAIL_EXCLUDE_PATH` as shown in the following code snippet:

      ```
      spec:
      containers:
      - env:
          - name: FLUENTD_ARGS
            value: --no-supervisor -q
          - name: FLUENT_CONTAINER_TAIL_EXCLUDE_PATH            # Add this line

            value: /var/log/containers/fluentd*                 # Add this line

        image:
      registry.ci.ciscolabs.com/cpsg_base-apps/fluentd-elasticsearch/releases/fluentd-elasticsearch:v2.4.0-cisco2

        imagePullPolicy: IfNotPresent
        name: fluentd-es
        resources:
          limits:
              memory: 500Mi
          requests:
              cpu: 100m
              memory: 200Mi
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      ```

      After these changes are saved, the fluend-es pods will restart.

- Cisco Container Platform does not support AWS IAM authentication for vSphere clusters.
- In an air-gapped environment, you cannot connect to the Smart Licensing service using a proxy configured through the Cisco Container Platform web interface.

  Workaround

  1. Disable the proxy server.
  2. Log in to the Cisco Container Platform web interface.
  3. In the left pane, click **Licensing**.
  4. Click the **Licensing Status** tab, and then click the **View/Edit** link that appears under **TRANSPORT SETTINGS**.
  5. Click the **CONNECT USING HTTP/HTTPS PROXY** toggle button to disable using the proxy server.
  6. SSH to the control master node, and run the following command after replacing the IP and port with your proxy server's IP and port:

```
     kubectl patch deploy kaas-slagent --patch
'{"spec":{"template":{"spec":{"setHostnameAsFQDN":true,"containers":[{"name":"kaas-slagent"}],"containers":[{"env":[{"name":"AACPS","value":"Dcom.cisco.rsa.configfile=/config/agentconfig.json
 -Dhttp.proxyHost=<proxy_ip> -Dhttp.proxyPort=<proxy_port>
-Dhttps.proxyHost=<proxy_ip>
-Dhttps.proxyPort=<proxy_port>"}],"name":"kaas-slagent"}]}}}}'
```

Wait for a few minutes to ensure a new slagent pod comes up with the above patched environment variable.

7. In the Cisco Container Platform web interface, go to the **Smart Licensing** tab, and register the license using the token obtained from software.cisco.com. For more information, see the steps to configure Cisco Smart Software Licensing in the latest Cisco Container Platform User Guide.

- Use of Contiv CNI is no longer supported in Cisco Container Platform 6.1.0.
- v2 clusters are being deprecated from the 6.1.0 release.
- You cannot delete a user management group if its group name contains special characters.
- Cisco Container Platform does not support localization.
- Harbor add-on is not supported for Openstack IPV6 tenants.
- Harbor add-on is not supported on ACI clusters.
- When you use VMware with a non-English locale as the provider, tenant cluster installations of v2 and v3 clusters are not supported.
- On a tenant that uses ACI CNI, installation of kubeflow fails if an SNAT ACI policy is not configured.

  Workaround

  You must configure an SNAT ACI policy before installing kubeflow. For more information, see Configuring and Applying SNAT Policy.

- Istio is not supported when you upgrade a tenant cluster from the 5.0 version to a later version as the `ccp-ist-operator`and `ccp-ist-cr` Istio helm charts are deprecated in Cisco Container Platform 5.1.0.
- When Istio is installed in a Cisco Container Platform 5.1.0 tenant, the `grafana` pod in the `istio-system` namespace crashes at times.

  Workaround

  Run the following command to upgrade `grafana` in the `istio-system` namespace to a newer image.

  ```
  helm upgrade istio -f /opt/ccp/charts/istio.yaml --namespace istio-system \
  --set grafana.image.tag=6.3.5-cpsg2 /opt/ccp/charts/istio.tgz
  ```

- When a Cisco Container Platform tenant with Istio is upgraded, all the Istio custom resources (CRs) that you have created are lost due to upgrade issues in istio, helm, and helm's `crd-install` hook.

  Workaround

  1. Before upgrading the Cisco Container Platform tenant, run the following bash script to list all the Istio CRs in the tenant.

  ```
  #!/usr/bin/env bash

        while read -r istioCR
        do
            output=$(kubectl get "$istioCR" --all-namespaces -o yaml 2>/dev/null |
  grep '^items:')
  ```

```
            if ! grep -q "\[\]" <<< $output; then
                echo -e "\n======== istio's "$istioCR" ========\n"
                kubectl get "$istioCR" --all-namespaces
            fi
        done < <(kubectl get crd --all-namespaces | grep 'istio\.io' | awk '{print
$1}')
```

2. Back up the manifests of all the Istio CRs you have created.
3. Upgrade the Cisco Container Platform tenant.
4. After the tenant upgrade is complete, recreate your Istio CRs using the backed up manifests.

- The Swagger API page may fail to load.

  Workaround

  1. Open the swagger UI using the following URL format:

     https://<ccp_ui_ip>/2/swaggerapi

  2. Click the **three dots** icon at the upper right corner of the menu bar, and then choose **More Tools** > **Developer Tools**.

     The **Developer Tools** panel appears.

  3. Click **Sources** > **Overrides** > **Select folder for overrides**.
  4. Select a newly created empty folder to store the local overrides.
  5. In the notification that appears just below the URL bar, click **Allow** to give Developer Tools full access to the selected folder.
  6. Check the **Enable Local Overrides** checkbox.
  7. Click the **Page** tab and click on the `index` file.

     The `index` file appears in the right pane.

  8. Edit the `index` file to add the following line after the first line:

     ```
         '<meta http-equiv="Content-Security-Policy" content="default-src *; style-src
      'self' 'unsafe-inline'; script-src * 'unsafe-inline' 'unsafe-eval'">'
     ```

  9. Save the file using `CTRL+S`.
  10. Close the **Developer Tools** panel.

- Taints and labels for node groups are not supported.
- The v3 tenant clusters indicate a **READY** status before provisioning is fully completed.
- The **vSphere** tab for v3 clusters is not enabled for control planes that are configured to use contiv-vpp network plugin in tenant clusters.
- During the creation of a vSphere v3 cluster, master node group and worker node groups cannot have different Kubernetes versions.
- The Harbor add-on is only available for v2 clusters.
- The Istio operator add-on is only available for v3 clusters and must be installed before installing Istio.

- The Istio operator add-on must be installed before removing Istio on a v3 cluster.
- After installing Istio, pods such as istio-pilot and istio-egressgateway remain unavailable, and logging errors indicating Insufficient CPU occur.

  Workaround

  Follow one of these steps:

  - Increase tenant cluster worker node count to greater than 1
  - Increase tenant cluster worker node VCPU count to greater than 2

- An Nginx Ingress Controller on a tenant cluster may randomly pick a port that is needed by another application, causing port conflicts.

  The following example shows an Istio operator error due to a port conflict caused by the Nginx Ingress Controller.

```
    2019-10-09T23:08:59.319Z    ERROR    controller-runtime.controller    Reconciler
error    {"controller": "istio-application", "request": "ccp/ccp-istio", "error": "Failed
 to install istio helm chart, error: Error: release istio failed: Service
\"istio-ingressgateway\" is invalid: spec.ports[3].nodePort: Invalid value: 31400:
provided port is already allocated\n, exit status 1"}

    ccpuser@user-tlc-0-master-0:~$ kubectl get svc --all-namespaces
    NAMESPACE      NAME                              TYPE           CLUSTER-IP
EXTERNAL-IP    PORT(S)                       AGE
    ccp            nginx-ingress-controller        LoadBalancer   10.10.10.248
10.10.10.89   80:31400/TCP,443:31766/TCP   45m
    ...
```

  Workaround

  Follow these steps to recreate the Nginx Ingress Controller service on a different port:

  1. On the control plane, delete the NginxIngress CR. Net Tinker on the control plane subsequently deletes the Nginx Ingerss Controller on the tenant cluster.

```
   ccpuser@user-cp-master7cdedf6f97:~$ kubectl delete nginxingress user-tlc-ingress
nginxingress.net.ccp.cisco.com "user-tlc-ingress" deleted
```

  2. Net Tinker on the control plane automatically recreates the Nginx Ingress CR and subsequently recreates the other Nginx resources on the tenant cluster during the next reconciliation.

```
    ccpuser@user-cp-master7cdedf6f97:~$ kubectl get nginxingress
    NAME                STATE
    user-tlc-ingress
    ccpuser@user-cp-master7cdedf6f97:~$ kubectl get nginxingress
    NAME                STATE
    user-tlc-ingress   InProgress
    ccpuser@user-cp-master7cdedf6f97:~$ kubectl get nginxingress
    NAME                STATE
    user-tlc-ingress   InProgress
    ccpuser@user-cp-master7cdedf6f97:~$ kubectl get nginxingress
    NAME                STATE
    user-tlc-ingress   Ready
```

  3. Verify if the Nginx Ingress Controller is running on a new, unused pod.

```
    ccpuser@user-tlc-0-master-0:~$ kubectl get svc --all-namespaces
    NAMESPACE      NAME                              TYPE           CLUSTER-IP
EXTERNAL-IP    PORT(S)                       AGE
    ccp            nginx-ingress-controller        LoadBalancer   10.10.10.173
```

```
10.10.10.89   80:32305/TCP,443:30702/TCP   4s
    ...
```

- Cisco Container Platform 2.2.2, is exposed to the TTA-2019-001 Calico vulnerability. This vulnerability is addressed in Cisco Container Platform 3.0+.

  Workaround

  For a tenant cluster that runs Cisco Container Platform 2.2.2, follow these steps to address the vulnerability:

  1. Modify the `calico-config` ConfigMap to set the `log_level` to `warning` or higher.

     ```
     kubectl edit configmap -n kube-system calico-config
     ```

  2. Modify the calico-node container in the calico-node daemonset to set the environment variable `FELIX_LOGSEVERITYSCREEN` to `info` or higher.

     ```
     kubectl patch ds -n kube-system calico-node --patch \ '{"spec": {"template":
{"spec": {"containers": [
    Unknown macro: {"name"}
    ]}}}}'`
     ```

  3. To generate a new secret, follow these steps in the `kube-system` namespace:

     a. Find the correct secret.

        ```
        kubectl get secrets -n kube-system | grep calico-node
        ```

     b. Delete the secret.

        ```
        kubectl delete secret -n kube-system
        ```

     A new secret is automatically generated by the token controller.

- When Cisco Container Platform is upgraded from 4.2.1 to 5.0.0, AKS cluster creation fails with a 500 Internal Server Error.

  Workaround

  Run the following command on the Cisco Container Platform control plane:

  ```
  kubectl delete validatingwebhookconfiguration validating-webhook-configuration
  ```

- Earlier versions of Cisco Container Platform 4.2.0, are exposed to the TTA-2019-002 Calico vulnerability. This vulnerability is addressed in Cisco Container Platform 4.2.0.

  Workaround

  For a tenant cluster that runs an earlier version of Cisco Container Platform 4.2.0, follow these steps to address the vulnerability:

  1. Download `calicoctl`.

     ```
     curl -O -L
https://github.com/projectcalico/calicoctl/releases/download/v3.7.4/calicoctl
     ```

  2. Ensure calicoctl is excutable.

     ```
     chmod +x calicoctl
     ```

  3. Apply the attached network policy.

     ```
     sudo DATASTORE_TYPE=kubernetes KUBECONFIG=/etc/kubernetes/admin.conf ./calicoctl
 apply -f <DENY_POD_IPIP>.yaml
     ```

4. Verify the iptable rule.

```
    -A cali-po-XXXXXX -s <POD CIDR> -p ipencap -m comment --comment "cali:XXXXXXX"
-j DROP
```

- In Cisco Container Platform 4.1.0, Azure Kubernetes Service (AKS) cluster creation fails if you use Kubernetes 1.13.5. The **Clusters** page displays an **UNKNOWN** status for the cluster.

  Workaround

  Upgrade to Cisco Container Platform 4.2.0, and then attempt to create Azure Kubernetes Service (AKS) clusters using Kubernetes 1.13.9.

- Azure Kubernetes Service (AKS) cluster creation may fail due to issues in the underlying Azure infrastructure components.

  Workaround

  Create a new AKS cluster.

- Even though the Cisco Container Platform web interface to create AKS clusters indicates that the Pod CIDR and Service CIDR are optional parameters, it is required to specify these parameters.

- When you create local user accounts, it is required to specify the **First Name** and **Last Name** fields.

- Kubernetes dashboard pod may be in a CrashLoopBackOff state on a cluster.

  Workaround

  Connect to the master node of a cluster using SSH and execute the following commands:

```
kubectl apply -f /opt/ccp/manifests/kubernetes-dashboard-role.yaml
kubectl apply -f /opt/ccp/manifests/kubernetes-dashboard-rolebinding.yaml
```

- Mounting volumes fail when using pods with Persistent Volumes provisioned using HyperFlex CSI or HyperFlex.

  As a result, the following errors may occur:

  - The pod remains in the `ContainerCreating` state.
  - Viewing pod details results in errors.

    For example:

```
kubectl describe pod <Pod name>
```

```
Warning FailedMount ... Unable to mount volumes for pod ...: timeout expired waiting
 for volumes to attach or mount for pod
Warning FailedMount ... MountVolume.MountDevice failed for volume ... : rpc error:
 code = DeadlineExceeded desc = context deadline exceeded
```

  - The log file on the HyperFlex controller VM contains errors.

    For example:

    Error found in `/var/log/springpath/debug-scvmclient.log`:

```
scvmclient[xxx:xxx]: USER: ALERT: ISTGT.ISTGT.GenericMessage: istgt_lu_disk_init:xxx:
 Retrying open for LU1: LUNxxx: retryCnt:1
scvmclient[xxx:xxx]: USER: ALERT: ISTGT.ISTGT.GenericMessage: istgt_lu_disk_init:xxx:
 LU1: LUNxxx: open error(errno=25000)
```

  Workaround

Follow these steps to reset the state of the HyperFlex FlexVolumes and CSI volumes:

1. Delete the Persistent Volume Claims and Persistent Volumes that are created using HyperFlex FlexVolume or CSI provisioners.

   ```
   kubectl delete pvc <Persistent volume claim name>
   kubectl delete pv <Persistent volume name>
   ```

2. Log in to the HyperFlex controller VM with the HyperFlex Management IP address.

   ```
   /usr/share/zookeeper/bin/zkCli.sh
   ```

3. On the zkCli console, run the following commands:

   ```
   rmr /hxVolumeInv
   exit
   ```

4. To clear the HyperFlex Persistent Volume state, run the following commands on each HyperFlex controller VM:

   ```
   rm /nfs/SYSTEM/istgt.conf
   restart scvmclient
   restart hxSvcMgr
   ```

5. Verify if the volume is mounted.

   ```
   ls /nfs/SYSTEM
   ```

   If the volume is not mounted, run the following command to mount it:

   ```
   initctl emit --no-wait system-datastore-created
   ```

- Deploying tenant clusters with GPU requires manual configuration.

  After cluster creation, run the following command to manually install the Nvidia Device Plugin on the tenant cluster:

  ```
  kubectl apply -f
  https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/master/nvidia-device-plugin.yml
  ```

- You cannot modify the size of the repository of a Harbor tenant cluster in the 3.2.0 and 3.2.1 version of Cisco Container Platform.

  Workaround

  Provision a Harbor tenant using Cisco Container Platform 4.0.0.

- When you upgrade from the Cisco Container Platform 3.0 version, the managed IP addresses that belong to the old clusters not properly released.
- In the Cisco Container Platform web interface, the EKS clusters are visible only to Admin users.
- Limitations when using vSphere with Cisco Container Platform:
  - You must not place the Cisco Container Platform VMs in a nested folder. It must be retained in the default installation location.
  - You must not configure Cisco Container Platform to use datastores located in folders or in a Storage DRS (SDRS).
- cert-manager is now deployed in tenant clusters. It is supported as Tech Preview.
- Cisco Container Platform upgrade from a version earlier than 2.2.2 fails when the cluster name contains uppercase letters.

Workaround

1. SSH to the Cisco Container Platform Control Plane master VM and change the cluster name to lowercase in the `ccp-appdata` table:

```
      sudo apt-get update
      sudo apt-get install -y jq
      kubectl exec -it mysql-0 -- mysql -p$(kubectl get secret mysql -o json | jq
-r '.data["mysql-root-password"]' | base64 -d) ccp-appdata -e "update
keyvalues_keyvalue set value = replace(value, 'CCP-CLUSTER-NAME',
lower('CCP-CLUSTER-NAME')) where instr(value, 'CCP-CLUSTER-NAME') > 0;"
      kubectl exec -it mysql-0 -- mysql -p$(kubectl get secret mysql -o json | jq
-r '.data["mysql-root-password"]' | base64 -d) ccp-appdata -e "select * from
keyvalues_keyvalue;"
```

2. If you are using a localized version of vSphere, follow these steps to rename the datastore folder for the cluster data:
   a. In the vSphere web client, click **vCenter**.
   b. Click the **Storage** tab.
   c. From the left pane, choose the datastore that is used to create the cluster.
   d. Select the folder with the cluster name that you want to change.

      For example: `CCP-CLUSTER-NAME`
   e. Rename the folder to the lowercase of the same name.

      For example: `ccp-cluster-name`

3. Follow these steps to ensure that any existing disk path uses lowercase names:
   a. Click the **Virtual Machines** tab, choose the VM named `ccp-cluster-name-masterxxxxx`, and then click **Edit settings**.
   b. Remove **Harddisk 2**.
   c. Click the **Manage Other Disks** tab and remove **Harddisk**.
   d. Click **Add Existing Hard Disk** and choose the disk from `datastore/<your cluster name>/etcd.disk`.
   e. Click **Add Existing Hard Disk** and choose the disk from `datastore/<your cluster name>/cert.disk`.

4. Start the upgrade of Cisco Container Platform using the same cluster name in lowercase.

- On Upgrading to HyperFlex 3.5.2, volume traffic disruption occurs.

  Note: This section is applicable only if you are using the HyperFlex Flex Volume plugin for Kubernetes.

  In HyperFlex 3.5.1 or earlier, the IP address used by the vSwitch on ESXi hosts was 169.254.1.1. The HyperFlex clusters whose **Storage Hypervisor Network** addresses are in the range 169.254.1.0/24 conflicted with 169.254.1.1. To work around this IP conflict issue, in HyperFlex 3.5.2, the default IP address is changed to 169.254.254.1. Due to this change, the Flex Volume configuration on the Kubernetes nodes will no longer be correct after an upgrade.

  Workarounds

Note: You must use **only one** of the following two options to workaround this issue.

Option 1: Change Configuration on HyperFlex Controller VMs

You can use this option when there are no existing HyperFlex clusters that use the 169.154.1.0/24 range on ESXi. This avoids the need to change the Kubernetes node configuration for these clusters.

After upgrading HyperFlex to 3.5.2, follow these steps to change the default IP address to 169.254.1.1:

1. Run the following command to find iscsiTargetAddress = "169.254.254.1" and replace it with iscsiTargetAddress = "169.254.1.1" in the application.conf file:

```
    sed -i -e 's/iscsiTargetAddress*169.254.1.1/iscsiTargetAddress*169.254.254.1/g'
 /opt/springpath/storfs-mgmt/hxSvcMgr-1.0/conf/application.conf
```

2. Run the following command to find istgtConfTargetAddress = "169.254.254.1" and replace it with istgtConfTargetAddress = "169.254.1.1" in the application.conf file:

```
     sed -i -e
's/istgtConfTargetAddress*169.254.254.1/istgtConfTargetAddress*169.254.1.1/g'
/opt/springpath/storfs-mgmt/hxSvcMgr-1.0/conf/application.conf
```

3. Run the following commands to restart the following services:

```
     restart hxSvcMgr
     restart stMgr
```

Option 2: Change Configuration on all Kubernetes VMs

You can use this option when there are existing HyperFlex clusters that use the 169.154.1.0/24 range on ESXi. After a Kubernetes cluster operation such as scale up or upgrade, this step must be repeated on the new VMs. For this reason, we recommend option 1 as the preferred solution.

After upgrading HyperFlex to 3.5.2, run the following command for every Kubernetes VM to find "targetIp": "169.254.1.1" and replace it with "targetIp": "169.254.254.1" in the hxflexvolume.json file:

```
     ssh -l <ssh user> -i <private key file> <VM IP> -- sed -i -e
's/169.254.1.1/169.254.254.1/g' /etc/kubernetes/hxflexvolume.json
```

Note:

The `<ssh user>` must match the ssh user that you specified during cluster creation.

The `<private key file>` must correspond to the public key that you specified during cluster creation.

- During a Control Plane upgrade, if you change the **SUBNET CIDR** field on the **Verify Network** screen, the **IP ADDRESS RANGE** is updated.

Workaround

Note: You must use **only one** of the following two options to workaround this issue.

Option 1:

Go to the **Authenticate CCP** screen, enter the necessary data, and then click **NEXT**.

The original IP address range is restored.

Option 2:

In a Contiv or Calico deployment, find the original IP address range from the **Network Editing** screen.

Note:

  - In an ACI deployment earlier than the 2.2.x, the original start and end IP address is the existing Control Plane IP address.
  - In an ACI deployment 2.2.x and later, the original start and end IP address is the same as that which is configured during the Cisco Container Platform installation.

- During an ACI tenant upgrade, you can safely ignore the Subnet field.
- Cisco Container Platform must use the Kubernetes images that are associated with the current release. Older versions of the tenant base image are not supported.
- ACI tenant cluster does not work with a link-local interface with Kubernetes 1.11.3.
- You can use only the latest two versions of the tenant image that are associated with the current release. Use of older versions of the tenant image is not supported.
- You will get errors when you scale up tenant clusters or add new node pools to clusters that were created using an older version of **Cisco Container Platform**.

  Workaround

  You must upgrade Cisco Container Platform before attempting to scale up tenant clusters or add new node pools.

- When using HyperFlex as the dynamic provisioner, mounting volumes may fail with the following error message:

```
    MountVolume.SetUp failed for volume "xxxxx" : mount command failed, status: Failed
to mount volume xxxxx, reason:
```

  Workaround

  1. Restart the scvmclient on the esx server using the following command:

     ```
     /etc/init.d/scvmclient restart
     ```

  2. Ensure that the status is running.

- In an ACI environment, the link to a tenant cluster Kubernetes Dashboard from the Cisco Container Platform dashboard is not supported. To view the tenant cluster in the Kubernetes Dashboard, you need to obtain the Ingress IP of external IP address using `kubectl get svc`.
- The Cisco Container Platform web interface displays links to external pages such as Smart Licensing. You cannot launch these pages if you do not have access to them.
- Virtual IP address is not released when cluster creation fails.
- In a Contiv deployment, you should not use `matchExpressions` for a NetworkPolicy.
- In a Contiv deployment, network policy does not work with the hostnetwork pod.
- In a Contiv deployment, the pod CIDR must be at least a /14 network.
- In a Calico deployment:
  - The network policy matching on labels will not block hostnetwork access to pods or services.
  - Host IP change may impact pod networking. To resolve the issue, you need to restart the Calico pods.

- `istioctl` is not installed when you enable Istio.

  For more information on installing Istio, refer to the latest Istio documentation.
- When you upgrade tenant clusters, the Prometheus and EFK components are purged before installing the new versions. If you want to save history, a manual backup and migration are required before a tenant cluster upgrade.
- Taking a snapshot of the VMs managed by Cisco Container Platform is currently unsupported and results in failures during upgrades.
- ACI deployments are only supported in online mode.
- ACI deployments do not support Kubernetes security context.

Viewing Open and Resolved Bugs

The open and resolved bugs for this release are accessible through the Cisco Bug Search Tool. This web-based tool enables you to access the Cisco bug tracking system, which maintains information about bugs and vulnerabilities in this product and other Cisco hardware and software products. You can search for bugs using bug IDs or keywords.

Before you begin

Ensure that you have a Cisco username and password to log in to the Cisco Bug Search Tool. If you do not have a Cisco username and password, you can register for an account.

Procedure

Step 1 Log in to the Cisco Bug Search Tool with your Cisco username and password.

Step 2 To search for a specific bug, enter the bug ID in the **Search For** field and press the **`Enter`** key.

Step 3 To search for the bugs that belong to the current release, enter **`Cisco Container Platform 6.0.0`** in the **Search For** field, and then press the **`Enter`** key.

Note
- Once the search results are displayed, you can use the **Filter** options to easily find the bugs that are of interest to you.
- You can search for bugs by status, severity, modified date, and so on.

Step 4 To export the results to a spreadsheet, click the **Export Results to Excel** link.

For more information on the Cisco Bug Search Tool, refer to http://www.cisco.com/web/applicat/cbsshelp/help.html.

Related Documentation

The following table lists the documents that are available for Cisco Container Platform.

Document	Description
Cisco Container Platform Installation Guide	Describes installing Cisco Container Platform on your deployment environment.
Cisco Container Platform User Guide	Describes administering and managing Kubernetes clusters, and deploying applications on them.
Cisco Container Platform API Guide	Describes the Cisco Container Platform APIs.

These documents are available on cisco.com.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see What's New in Cisco Product Documentation.

What's New in Cisco Product Documentation lists all new and revised Cisco technical documentation. You can subscribe to it, and receive free RSS feed service directly to your desktop using a reader application.