Back up and Restore Cisco Container Platform

This chapter contains the following topics:

Back up Cisco Container Platform

You can back up the Cisco Container Platform application data that pertains to the following components:

- Application users
- Virtualization providers
- Tenant clusters

Note The logging or monitoring data from Prometheus, Grafana, and the EFK stack is not included in the backup archive.

You must ensure that you use an up-to-date backup archive for a restore operation. Tasks such as creating, deleting, upgrading, or scaling tenant clusters or altering the number of Load Balancer Virtual IP addresses will create changes in the data that will not be present in the backup. If you perform such tasks after a backup and use an outdated backup archive to restore your Cisco Container Platform environment, unexpected IP address conflicts, unmanageable tenant clusters, or an unsuccessful restore may occur.

Before you begin

Ensure that at least 6 consecutive IP addresses are available in the same pool where the Cisco Container Platform Control Plane is deployed.

When the target for a restore is a new cluster, you must ensure that additional free IP addresses are available to avoid conflicts with the IP addresses that are currently in use.

For more information on the requirement for additional free IP addresses, refer to the *Managing Networks* section of the *Cisco Container Platform User Guide*.

Step 1 Log in to the console of the master node of the Cisco Container Platform Control Plane.

Note
- Note down the IP addresses assigned to the VMs of the control plane.
- Note down the IP address of the ingress-nginx-controller service.

Step 2 Run the **backup-k8s-artifacts.sh** script to create a backup of the Kubernetes artifacts.

backup-k8s-aritfacts.sh

```
./ccp_related_files/backup-k8s-artifacts.sh
    backing up provider secrets
    backing up kubeconfig secrets
    backing up network resources
    backing up cluster resources
```

Run the following command:

```
./ccp_related_files/backup-k8s-artifacts.sh
```

Step 3 Copy the tar file generated in Step 2 to a secure location outside of the current master node of the control plane.

Step 4 Run the **percona-backup.sh** script to create a backup of the percona database that contains data related to ccp-api, ccp-networks, and ccp-appdata. This data is used when restoring the control plane data.

percona-backup.sh

```
/ccp_related_files/percona-backup.sh ccp-percona-db-backup.tar
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
tar: Removing leading `/' from member names
tar: Removing leading `/' from member names

Retrieving allocated CCP Control Cluster IPs
Retrieving IP pools for control plane subnets
Retrieving all allocated IPs within the control subnets
Determining IPs in pools
Determining available IP ranges

------------------------------------------------
IP Range 10.10.96.126 to 10.10.96.140 has 15 free IPs.     # <==

Backup is valid only if control plane operations are not performed
. No cluster create, upgrade, delete, scale out or in, change in
number of LoadBalancer VIPs, etc. Restoring a backup after changes
can result in IP collisions.

Percona DB backup saved to percona-db-backup/percona-db.tar
CCP Backup saved to ccp-percona-db-backup.tar


ATTENTION

Part of the backup includes a decryption key needed to restore the CPP Control Cluster.
Store this key securely and separately from the backup data.

The key is below:


8b2de348aca874342a1288c77fe821d6567fb619ab60a72e13af5f8f9dcbe3d21d669335d3651ec54dd6dc5dae8729b8a27f0cdcbfd9b302a23f2bb35a939be5
```

Run the following command:

```
./ccp_related_files/percona-backup.sh ccp-percona-db-backup.tar
```

Step 5 After running the backup script, note down the following information from the console:

- The valid IP pool ranges with enough free IPs to create a replacement Cisco Container Platform Control Plane.

  You must save the IP ranges for future use while specifying the **IP Address Range** on the **Network Settings** screen during an install, see Deploying Cisco Container Platform.

- The encryption key is needed to decrypt the backup data encryption key that is stored on your disk.

  You must save the encryption key for future use while restoring the database to a new Cisco Container Platform Control Plane. For more information, see Restore Cisco Container Platform, on page 3. Losing the encryption key will prevent the restoration of the Cisco Container Platform Control Plane cluster.

Step 6 Use the **scp** utility to copy the **ccp-percona-db-backup.tar** file from Step 4 to a secure location outside of the master node control plane.

Note You must ensure that the backup archive is maintained securely as anyone with access to it has administrative capabilities on all tenant clusters.

Restore Cisco Container Platform

You can restore a valid backup to a new Cisco Container Platform Control Plane instance of the same version that has control over all the existing Cisco Container Platform settings and tenant clusters. Restoration of the backup to the same Cisco Container Platform Control Plane instance is not supported.

When restoring a backup archive to a new Control Plane, the Ingress, kube-apiserver, and node IP addresses will not be restored, they will remain the same as when the new Control Plane was created.

Before you begin

For versions 3.2 and later, you must have the Encryption key provided during the Back up Cisco Container Platform.

Step 1 Power off the VMs that belong to the previous Control Plane instance.

Step 2 Install a new Cisco Container Platform control plane instance using one of the following approaches:

- **Approach 1:** Restore the Cisco Container Platform control plane to the same IP address.

  When installing the Cisco Container Platform control plane, you can assign the IP address of the original control plane to the new control plane.

  During the installation of Cisco Container Platform, in the **Network Settings** step, enter the same Pool IP range as that used in the original instance of the Cisco Container Platform control plane.

  For example, if the original control plane was deployed using the IP address range 10.96.96.6 to 10.96.96.10, you can use the same IP pool range, so that the new control plane will be deployed with the same IP addresses.

- **Approach 2:** Restore to the Cisco Container Platform control plane within the same Pool IP range as that used in the original instance, but using a different IP address than the original control plane.

You must note down the IP address range suggested during the execution of the **percona-backup.sh script** in Back up Cisco Container Platform, on page 1.

During the installation of Cisco Container Platform, in the **Network Settings** step, enter a Pool IP address that falls within the suggested IP range.

For example, if the original control plane was deployed using the IP address range 10.96.96.6 to 10.96.96.10. The suggested IP range from the output of **percona-backup.sh** is 10.10.96.126 to 10.10.96.140, then you can provide any IP address range from the above range such as:

- Entire range: 10.10.96.126 - 10.10.96.140
- Partial IP address range: 10.10.96.135 - 10.10.96.140

Note You must ensure that the IP address range must have a minimum of six IP addresses available.

Step 3 After the new Cisco Container Platform control plane is up and running, copy the backup artifacts (.tar) files from your secure location to the master node of the control plane.

Step 4 Restore the Kubernetes artificates such as cluster providers, cluster kubeconfig, cluster objects, and network objects using the **restore-k8s-artifacts.sh** script.

restore-k8s-artifacts.sh

```
$ ./ccp_related_files/restore-k8s-artifacts.sh backup-1620255959.tar
 + echo 'extracting tar file'
 extracting tar file
 + tar -xf backup-1620255959.tar -C restore-1620258443/
 + echo 'restoring artifacts'
 restoring artifacts
 + echo 'restoring providers'
 restoring providers
 + kubectl apply -f restore-1620258443/k8s-provider.yaml
 secret/vsphere-provider-ec8eb851-6091-4812-9e4e-1668134b4001 created
 + echo 'restoring kubeconfig-secrets'
 restoring kubeconfig-secrets
 + kubectl apply -f restore-1620258443/kubeconfig-secrets.yaml
 secret/vsc-006-kubeconfig created
 + echo 'restoring network resources'
 restoring network resources
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/001-k8s-network.yaml
 ipallocator.net.ccp.cisco.com/vsc-006-ipallocator created
 + sleep 2
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/002-k8s-network.yaml
 clusternetwork.net.ccp.cisco.com/vsc-006 created
 + sleep 2
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/003-k8s-network.yaml
 metallb.net.ccp.cisco.com/vsc-006-lb created
 + sleep 2
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/004-k8s-network.yaml
 ipaddress.net.ccp.cisco.com/vsc-006-lb-qc47p created
 ipaddress.net.ccp.cisco.com/vsc-006-master-gro-aed3ea78b7-ens192 created
 ipaddress.net.ccp.cisco.com/vsc-006-node-group-5e6bde4243-ens192 created
 ipaddress.net.ccp.cisco.com/vsc-006-node-group-e295e46618-ens192 created
 ipaddress.net.ccp.cisco.com/vsc-006-vip created
 + sleep 2
 + for i in {1..7}
 + kubectl apply -f restore-1620258443/005-k8s-network.yaml
```

```
cni.net.ccp.cisco.com/vsc-006-cni created
+ sleep 2
+ for i in {1..7}
+ kubectl apply -f restore-1620258443/006-k8s-network.yaml
netconfig.net.ccp.cisco.com/vsc-006-master-gro-aed3ea78b7 created
netconfig.net.ccp.cisco.com/vsc-006-node-group-5e6bde4243 created
netconfig.net.ccp.cisco.com/vsc-006-node-group-e295e46618 created
+ sleep 2
+ for i in {1..7}
+ kubectl apply -f restore-1620258443/007-k8s-network.yaml
nginxingress.net.ccp.cisco.com/vsc-006-ingress created
+ sleep 2
+ echo 'restoring cluster resources'
restoring cluster resources
+ kubectl apply -f restore-1620258443/k8s-clusters-resources.yaml
cluster.tlc.ccp.cisco.com/vsc-006 created
vspherecluster.vsphere.ccp.cisco.com/vsc-006 created
+ echo 'restoring complete!'
restoring complete!
+ echo 'SUCCESS!'
SUCCESS!
```

Run the following command:

```
./ccp_related_files/restore-k8s-artifacts.sh k8s-artifacts-backup-1621282525.tar
```

Step 5 Restore the percona-db database in one of the following ways:

- To install the Cisco Container Platform control plane to the **same IP address**, use the following **perona-restore.sh** script:

  perona-restore.sh

```
$ ./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
Determining if restoring to same or new cluster
Obtaining appdata from secret
Restoring to the same CCP Control Cluster, no network data extraction required
Restoring database, ignore 'command terminated with exit code 137' messages
restart_mysql_pod
Restarting mysql (1/2)
pod "mysql-0" deleted
Waiting for mysql (1/12)
Restarting mysql (2/2)
pod "mysql-0" deleted
Waiting for mysql (1/12)
rotate_user_credentials
Rotating user credentials
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql: [Warning] Using a password on the command line interface can be insecure.
restart_pods
Restarting pods (by deleting)
pod "kaas-api-74d784fffc-h9zjf" deleted
pod "kaas-appdata-547d6ff889-kfwf5" deleted
pod "kaas-ccp-aks-operator-8559549855-l2sx8" deleted
pod "kaas-ccp-cluster-operator-bc4f85c9c-cqrbl" deleted
pod "kaas-ccp-eks-operator-5ff44786cd-tfcwk" deleted
pod "kaas-ccp-gke-operator-549888d77c-q6xzh" deleted
pod "kaas-ccp-vsphere-operator-757f9ff84c-xbzdv" deleted
pod "kaas-corc-5b8c797589-q8pfj" deleted
pod "kaas-cx-aes-key-job-3nnek-cgjzq" deleted
pod "kaas-dashboard-f5556f997-gj5jf" deleted
pod "kaas-network-77cd6df999-tvl68" deleted
```

```
pod "kaas-network-77cd6df999-zf8j2" deleted
pod "kaas-network-initdb-vuhu3-4tft9" deleted
pod "kaas-sddc-bfb47bd4-jk7cc" deleted
pod "kaas-slagent-6f5bc4dd8b-4f4xs" deleted
pod "kaas-slagent-q5lha-5wd2z" deleted
wait_for_pods
Wait for all kaas pods, especially CORC, appdata, and slagent
pod/kaas-api-74d784fffc-h57ss condition met
pod/kaas-appdata-547d6ff889-cdlhk condition met
pod/kaas-ccp-aks-operator-8559549855-q94sn condition met
pod/kaas-ccp-cluster-operator-bc4f85c9c-zhwvw condition met
pod/kaas-ccp-eks-operator-5ff44786cd-gmncj condition met
pod/kaas-ccp-gke-operator-549888d77c-f9x5j condition met
pod/kaas-ccp-vsphere-operator-757f9ff84c-n5q5h condition met
pod/kaas-corc-5b8c797589-dbx6j condition met
pod/kaas-dashboard-f5556f997-bhkrc condition met
pod/kaas-network-77cd6df999-frsk7 condition met
pod/kaas-network-77cd6df999-wkf9w condition met
pod/kaas-sddc-bfb47bd4-6wfkm condition met
pod/kaas-slagent-6f5bc4dd8b-fbqjq condition met
resore_appdata
Restoring existing appdata - in place of backup
Wrote output to /dev/null
Restoring to the same CCP Control Cluster, no network data alignment required
align_uuid
Aligning the network uuids of restored cluster
Retrieving current ccp-appdata contents
Creating a copy of current ccp-appdata contents
Retrieving current ccp-appdata contents
updating uuid of ipaddress associated with node ccpres-same-worker0182107096
updating uuid of ipaddress associated with node ccpres-same-workerdfb4de8f98
updating uuid of ipaddress associated with node ccpres-same-masterf00682a62b
updating uuid of ipaddress associated with node ccpres-same-workerfeac01c85e
updating uuid of loadbalancer ip 10.10.96.115
updating uuid of node_pool default-pool
updating uuid for ip 10.10.96.119
updating uuid for ip 10.10.96.118
updating uuid for ip 10.10.96.116
updating uuid for ip 10.10.96.117
Testing retrievel of CCP Control Cluster data
Wrote output to /dev/null
Testing retrievel of CCP Tenant data
Wrote output to /dev/null
secret "temp-appdata" deleted

SUCCESS!
```

Run the following command:

```
./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
```

- To install the Cisco Container Platform control plane within the same IP address range configured for the original instance, but **using different IP addresses than the original control plane**, use the following **perona-restore.sh** script:

perona-restore.sh

```
$ ./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
Determining if restoring to same or new cluster
Obtaining appdata from secret

==================================
Enter Decryption Key presented during backup (key will not echo to screen):
Starting data extraction utility pod kube-system:ccp-backed-up-db
pod/ccp-backed-up-db created
```

```
      Waiting for data extraction utility pod to be Ready
      Waiting for pod
      Waiting for pod
      Configure data extraction utility pod
      Copying backup data from local into the utility pod
      Extract data from the backup database
      Unpacking data
      Loading data
      Configuring mysql
      Waiting for mysqld
      waiting for mysql...
      waiting for mysql...
      mysqld is alive
      Retrieve subnet data from backup
      Running subnet data query
      Copy network data extracted from backup down to local
      Delete utility pod
      pod "ccp-backed-up-db" deleted
      Retrieving new CCP Control Plane data
      Retrieving new IP allocations for CCP Control Plane
      Network checks pass, can continue with data restore
      Dump new CCP Control Cluster network data
      mysqldump: [Warning] Using a password on the command line interface can be insecure.
      Copy restore network data locally
      tar: Removing leading `/' from member names
      Retrieve new CPP Control Cluster subnet data
      Running subnet data query
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Copy new subnet data to local
      tar: Removing leading `/' from member names

      Retrieving encrypted aes key from backup
      Decrypting aes key
      *** WARNING : deprecated key derivation used.
      Using -iter or -pbkdf2 would be better.
      Updating Control Cluster aes key
      Flag --export has been deprecated, This flag is deprecated and will be removed in future.
      Warning: kubectl apply should be used on resource created by either kubectl create
--save-config or kubectl apply
      secret/cx-aes-key configured
      Restoring database, ignore 'command terminated with exit code 137' messages
      Restarting mysql (1/2)
      pod "mysql-0" deleted
      Waiting for mysql (1/12)
      Restarting mysql (2/2)
      pod "mysql-0" deleted
      Rotating user credentials
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Restarting pods (by deleting)
      pod "kaas-api-6c766d99c-thzlw" deleted
      pod "kaas-appdata-64b4656dd9-9mr62" deleted
      pod "kaas-ccp-aks-operator-5859d8f779-lb6qd" deleted
      pod "kaas-ccp-cluster-operator-b6d9cf9f7-zmnmv" deleted
      pod "kaas-ccp-eks-operator-64d46dcf5d-hr9ft" deleted
      pod "kaas-ccp-gke-operator-758ff99d65-b7vmp" deleted
      pod "kaas-ccp-vsphere-operator-7fb9b6768f-gr9q8" deleted
      pod "kaas-corc-64cd569468-wrbwn" deleted
      pod "kaas-cx-aes-key-job-ozb8s-9nnn6" deleted
      pod "kaas-dashboard-7f8fbbd7db-42bmc" deleted
      pod "kaas-network-7948cf457f-4dwrk" deleted
```

```
      pod "kaas-network-7948cf457f-krxkx" deleted
      pod "kaas-network-initdb-jaz9p-5gtcv" deleted
      pod "kaas-sddc-74978bfd56-zcdj5" deleted
      pod "kaas-slagent-5957db454d-bdz57" deleted
      pod "kaas-slagent-7vmip-lkh6l" deleted
      Wait for all kaas pods, especially CORC, appdata, and slagent
      pod/kaas-api-6c766d99c-r9dd7 condition met
      pod/kaas-appdata-64b4656dd9-vhchp condition met
      pod/kaas-ccp-aks-operator-5859d8f779-mjx58 condition met
      pod/kaas-ccp-cluster-operator-b6d9cf9f7-hkvdt condition met
      pod/kaas-ccp-eks-operator-64d46dcf5d-9g9pn condition met
      pod/kaas-ccp-gke-operator-758ff99d65-j5jqn condition met
      pod/kaas-ccp-vsphere-operator-7fb9b6768f-mm59p condition met
      pod/kaas-corc-64cd569468-kmwrt condition met
      pod/kaas-dashboard-7f8fbbd7db-crtsp condition met
      pod/kaas-network-7948cf457f-qzwp7 condition met
      pod/kaas-network-7948cf457f-tlz8d condition met
      pod/kaas-sddc-74978bfd56-g64lk condition met
      pod/kaas-slagent-5957db454d-6lgxp condition met
      Restoring existing appdata - in place of backup
      Wrote output to /dev/null
      Waiting for new CCP Control Cluster data to become available
      Wrote output to /dev/null
      Aligning subnet information in new CCP Control Cluster data to be compatable with tenant
data
      Retrieving new CCP Control Plane data
      Retrieving restored subnet configuration
      Updating appdata vip subnet identifier from 9a15ef9a-e2ff-482a-8a55-06a8bb0f224d to
b5c36f49-d449-42d2-8c95-6a1756a175c3
      Updating appdata vip subnet identifier from 9a15ef9a-e2ff-482a-8a55-06a8bb0f224d to
b5c36f49-d449-42d2-8c95-6a1756a175c3
      Done updating subnet data
      Copy the new CCP Control Cluster's network DB to the db pod
      Update the CCP Control cluster IPs to the new IPs
      Load copy of current ccp-network db
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysqldump: [Warning] Using a password on the command line interface can be insecure.
      Create copied db of restored ccp-network db
      Create new work database to combine data
      mysql: [Warning] Using a password on the command line interface can be insecure.
      mysql: [Warning] Using a password on the command line interface can be insecure.
      See if we can add the created_at column
      mysql: [Warning] Using a password on the command line interface can be insecure.
      ERROR 1067 (42000) at line 2: Invalid default value for 'created_at'
      Already had created_at column
      See if we can add the owner column
      mysql: [Warning] Using a password on the command line interface can be insecure.
      ERROR 1060 (42S21) at line 2: Duplicate column name 'owner'
      Already had owner column
      Copy system IPs into work db
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Checking that the correct number of IPs are present in work db
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Copy reconciled data back to current ccp-network db
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Restore new owner data if available
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Restore new created_at data if available
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Drop temporary databases
      mysql: [Warning] Using a password on the command line interface can be insecure.
      Updated ccp-network system IP's for control cluster to match restored subnets and pools
      Aligning the network uuids of restored cluster
```

```
Retrieving current ccp-appdata contents
Creating a copy of current ccp-appdata contents
Retrieving current ccp-appdata contents
updating uuid of ipaddress associated with node ccp610res-same-worker1a8e8814b1
updating uuid of ipaddress associated with node ccp610res-same-masterbb3eefc4d0
updating uuid of ipaddress associated with node ccp610res-same-workerd258545f91
updating uuid of ipaddress associated with node ccp610res-same-workereee85e4c64
updating uuid of loadbalancer ip 10.10.96.115
updating uuid of node_pool default-pool
updating uuid for ip 10.10.96.117
updating uuid for ip 10.10.96.116
updating uuid for ip 10.10.96.119
updating uuid for ip 10.10.96.118
Testing retrievel of CCP Control Cluster data
Wrote output to /dev/null
Testing retrievel of CCP Tenant data
Wrote output to /dev/null
secret "temp-appdata" deleted

SUCCESS!
```

Run the following command:

```
./ccp_related_files_new/percona-restore.sh ccp-percona-db-backup.tar
```

When prompted, enter the AES key that was generated during the creation of the percona-backup. For more information, see Back up Cisco Container Platform, on page 1.

Back Up Harbor Database

The database on Harbor tenant contains information such as user data and audit logs. This information can be backed up as a safety precaution before attempting a tenant upgrade on a Harbor tenant as the upgrade process may perform a database migration.

Note This backup process does not include docker images hosted on the Harbor registry.

Step 1 Log in to the console of the master node of Harbor tenant.

Step 2 Run the following command.

```
/opt/ccp/charts/harbor-db-backup.sh ./harbor_db_backup.sql default ccp-harbor
```

Step 3 Copy the `harbor_db_backup.sql` backup file to a secure location.

Restore Harbor Database

You can restore a valid Harbor database on a new or an existing Harbor tenant.

Step 1 Copy the backup from the secure location to Harbor tenant master.

```
scp ./harbor_db_backup.sql <harbor_tenant_master>:/tmp/harbor_db_backup.sql
```

Step 2 Log in to the console of the master node of Harbor tenant.

Step 3 Run the following command.

```
/opt/ccp/charts/harbor-db-restore.sh /tmp/harbor_db_backup.sql default ccp-harbor
```