Cisco Container Platform 2.1.0 API Guide

First Published: November 09, 2018

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Abstract
The Cisco Container Platform 2.1.0 API Guide gives information on Cisco Container Platform APIs and development features.

Cisco Container Platform 2.1.0 API Guide

Contents

1 Overview

Cisco Container Platform API provides REST API as a language-agnostic programmatic interface for applications to send requests to a Cisco Container Platform deployment.

An API conforms to the RESTful conventions and is defined by using resource and methods. A resource is a collection of information that is identified by a Uniform Resource Identifier (URI). For example, `providerclientconfig` is a resource that is used to represent configuration information to connect to an infrastructure provider such as vCenter. Methods are HTTP methods that are exposed for a resource. The commonly used HTTP methods are POST, GET, PATCH, PUT and DELETE.

You can access the Swagger UI for Cisco Container Platform API using the following URL:
`https://<CCP IP>:<Port>/2/swaggerapi`

Where, `<CCP IP>` is the node-port IP address or the master VIP, and `<Port>` is the port number used by the Cisco Container Platform control plane.

2 Authentication

The Cisco Container Platform REST API server uses basic authentication with a local or Active Directory to authenticate API requests.

3 Authorization

In Cisco Container Platform, there are two kinds of **roles**, namely, administrators and users. Administrators can create, read, update, and delete all resources. Users have limited capabilities and can only read a subset of the objects.

After a client is authenticated, it has permission to all or a subset of the API based on the type of user used for authentication. For example, the client may see a subset of the data when authenticated as a non-administrator user.

4 Key Concepts

Cisco Container Platform automates the creation and lifecycle operations for Kubernetes Clusters. Each Kubernetes Cluster corresponds to a cluster resource type. It is identified by name for POST and GET methods allowing you to poll the status of a Kubernetes cluster before it is created. All other methods on a cluster object identify the cluster by its UUID in the URI.

For example:
- To query a cluster information use:
  `https://CCPIP:Port/2/clusters/<Cluster Name>`
- To connect a kubeconfig file to a cluster use:
  `https://CCPIP:Port/2/clusters/<Cluster UUID>/env`

Cisco Container Platform allows you to connect to infrastructure providers such as vSphere to create Kubernetes clusters on them. Information to connect to the provider, such as the vCenter URL, username and password are part of a `providerclientconfig` resource. Cisco Container Platform Installer configures a default `providerclientconfig` resource named vsphere which can be used to create clusters.

A Kubernetes cluster in Cisco Container Platform consists of a master node and multiple worker nodes. The master node has a fixed Virtual IP that is allocated from a Virtual IP Pool provided by the user. The Cisco Container Platform Installer uses a Virtual IP.

5 Use Cases Examples of Cisco Container Platform APIs

5.1 Creating a Tenant Cluster

Before you Begin

Ensure that curl and jq are installed on your client machine.

Procedure

1. Export web IP:Port to the MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Web UI IP:PORT>

   Example
   `export MGMT_HOST=10.20.30.40:32443`

2. Obtain a cookie using the username and password for your Cisco Container Platform instance.

   Command
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/

   Example
   `curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/`

3. Get list of Provider Client Configurations.

Command
curl -sk -b cookie.txt -H "Content-Type: application/json" https://$MGMT_HOST/2/providerclientconfigs/ | jq '.[].uuid'

Example
```
curl -sk -b cookie.txt -H "Content-Type: application/json"
https://$MGMT_HOST/2/providerclientconfigs/  | jq '.[].uuid'
"fb53eae8-d973-4644-b13f-893949154a22"
```

4. Configure the provider client that you want to use.

   Command
   export PCC=<Selected Provider Client Configuration>

   Example
   ```
   export PCC=fb53eae8-d973-4644-b13f-893949154a22
   ```

5. Get the list of datacenters.

   Command
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter | jq '.Datacenters[]'

   Example
   ```
   curl -sk -b cookie.txt
   https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datac
   enter | jq  '.Datacenters[]'
   "RTP09"
   ```

6. Configure the datacenter that you want to use.

   Command
   export DCC=<from list of DataCenters>

   Example
   ```
   export DCC=RTP09
   ```

7. Get the list of tenant image VMs.

   Command
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/vm | jq '.VMs[]| select(.| startswith("ccp-tenant-image"))'| sort –u

   Example
   ```
   curl -sk -b cookie.txt
   https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datac
   enter/${DCC}/vm | jq '.VMs[]| select(.| startswith("ccp-tenant-
   image"))'| sort -u

   "ccp-tenant-image-1.10.1-2.1.0.ova"

   "ccp-tenant-image-1.11.3-2.1.0.ova"
   ```

8. Configure the name of the VM image that you want to use.

   Command
   export VM=<from list of VMs>

   Example
   ```
   export VM=ccp-tenant-image-1.11.3-2.1.0.ova
   ```

9. Get the list of networks.

   Command
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/network| jq '.Networks[]'

   Example
   ```
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/network| jq '.Networks[]'

   "r9-hx2-ccp"
   "Storage Controller Data Network"
   "k8-priv-iscsivm-network"
   ```

10. Configure the network that you want to use.

    Command
    export NETWORK=<From list of Networks>

    Example
    ```
    export NETWORK=r9-hx2-ccp
    ```

11. Get the list of clusters.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster| jq '.Clusters[]'

    Example
    ```
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster| jq '.Clusters[]'

    "r9-hx2"
    ```

12. Configure the name of the cluster you want to use.

    Command
    export CLUSTER=<from list of clusters>

    Example
    ```
    export CLUSTER=r9-hx2
    ```

13. Get the list of pools.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster/${CLUSTER}/pool| jq ".Pools[]"

    Example
    ```
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster/${CLUSTER}/pool| jq ".Pools[]"
    "Resources"
    "Resources/Infrastructure"
    ```

14. Configure the vSphere resource pool you want to use.

Command
export POOL=<from list of Pools>

Example
```
export POOL=Resources
```

15. Get the list of datastores.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/datastore | jq -r '.Datastores[]| select(.| startswith("SpringpathDS")|not)'

    Example
    ```
    curl -sk -b cookie.txt
    https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datac
    enter/${DCC}/datastore | jq -r '.Datastores[]| select(.|
    startswith("SpringpathDS")|not)'

    ds1
    ISOs
    Hxdump
    r9-hx2-datastore-1
    ```

16. Configure the datastore that you want to use.

    Command
    export DATASTORE=<from list of datastores>

    Example
    ```
    export DATASTORE=r9-hx2-datastore-1
    ```

17. Configure a name for the tenant cluster.

    Note: The cluster name must start with an alphanumeric character (a-z, A-Z, 0-9). It can contain a combination of hyphen (-) symbols and alphanumeric characters (a-z, A-Z, 0-9). The maximum length of the cluster name is 46 characters.

    Command
    export NAME=<Name of cluster>

    Example
    ```
    export NAME=tc4
    ```

18. Configure a username to remotely access cluster nodes with a given sshkey.

    Command
    export USER=<Username>

    Example
    ```
    export USER=ccpuser
    ```

19. Configure the ssh public key for remote access.

    Command
    export SSHKEY=<Selected ssh public key for remote access>

    Example
    ```
    export SSHKEY=`head -1 ~/.ssh/id_rsa.pub`
    ```

20. Get the list of subnets.

Command
curl -sk -b cookies.txt -H "Content-Type: application/json" https://$MGMT_HOST/2/network_service/subnets/ | jq -r '.[0].uuid'

Example
```
curl -sk -b cookies.txt -H "Content-Type: application/json"
https://10.10.1.2:32442/2/network_service/subnets/ | jq -r
'.[0].uuid'
```

```
"842e4baf-4877-4330-a3e3-
4249983922a4"
```

21. Configure the subnet for the cluster.

    Command
    export SUBNET=<From the list of subnets>

    Example
    ```
    export SUBNET=842e4baf-4877-4330-a3e3-4249983922a4
    ```

22. Get the list of VIP pools in the subnet that you have chosen.

    Command
    curl -sk -b cookies.txt -H "Content-Type: application/json" https://$MGMT_HOST/2/network_service/subnets/${SUBNET}/pools| jq -r '.[0].uuid'

    Example
    ```
    curl -sk -b cookies.txt -H "Content-Type: application/json"
    https://10.10.1.2:32442/2/network_service/subnets/${SUBNET}/poo
    ls| jq -r '.[0].uuid'
    ```

    ```
    "fef830ce-dc92-46fe-8acb-01eaa539dc46"
    ```

23. Select the appropriate VIP pool if there are multiple options.

    Command
    export VIP_POOL=<From the list of pools>

    Example
    ```
    export VIP_POOL=fef830ce-dc92-46fe-8acb-01eaa539dc46
    ```

24. Copy and paste the following code to create a cluster json payload.

```
#————————————————————————
cat <<EOF > cluster_create.json
{
  "provider_client_config_uuid": "${PCC}",
  "type": 1,
  "cluster": "${CLUSTER}",
  "name": "${NAME}",
  "description": "",
  "workers": 2,
  "masters": 1,
  "vcpus": 2,
  "memory": 8192,
  "datacenter": "${DCC}",
  "datastore": "${DATASTORE}",
  "networks": [
    "${NETWORK}"
  ],
   "ingress_vip_pool_id": ${VIP_POOL},
     "resource_pool": "${CLUSTER}/${POOL}",
     "template": "${VM}",
     "ssh_user": "${USER}",
```

```
    "ssh_key": "${SSHKEY}",
    "deployer_type": "kubeadm",
    "kubernetes_version": "1.11.3",
    "deployer": {
      "provider_type": "vsphere",
      "provider": {
        "vsphere_datacenter": "${DCC}",
        "vsphere_datastore": "${DATASTORE}",
        "vsphere_client_config_uuid": "${PCC}",
        "vsphere_working_dir": "\/${DCC}\/vm"
      }
    }
  }
  EOF

  #————————————————————————————
```

25. Edit the `cluster_create.json` file to modify the number of workers, CPUs, memory, Kubernetes version, or description as needed.
26. Create a tenant cluster.

    Command
    curl -sk -X POST -b cookie.txt -H "Content-Type: application/json" -d @cluster_create.json https://$MGMT_HOST/2/clusters | tee output.txt | jq '.name,.uuid,.state'

    Example

    ```
    curl -sk -X POST -b cookie.txt -H "Content-Type:
    application/json" -d @cluster_create.json
    https://$MGMT_HOST/2/clusters | tee output.txt | jq
    '.name,.uuid,.state'

    "tc4"

    "8ccaa3a1-8a11-4996-9224-5723b7ecfdfd"

    "READY"
    ```

27. Configure the tenant cluster UUID.

    Command
    #export TC=<UUID of the selected tenant cluster>

    Example

    ```
    export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
    ```

28. Download the KUBECONFIG environment file.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env

    Example

    ```
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env
    -o ${TC}.env
    ```

29. Export the config file to KUBECONFIG environment variable.

    Command
    export KUBECONFIG=./${TC}.env

    Example

    ```
    export KUBECONFIG=./${TC}.env
    ```

30. View nodes on a tenant cluster.

Command

kubectl get nodes -o wide

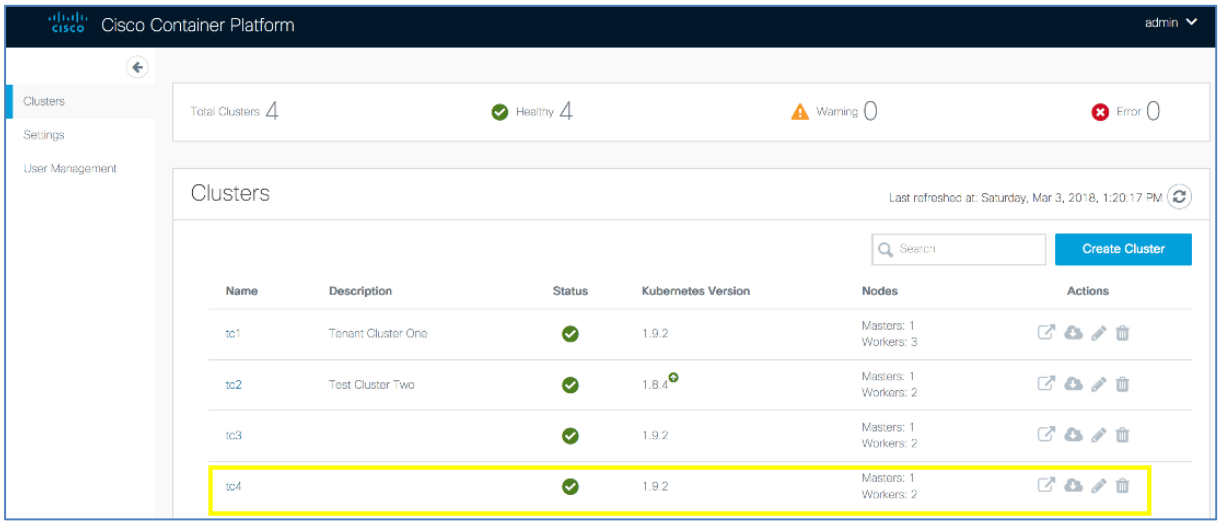
Example

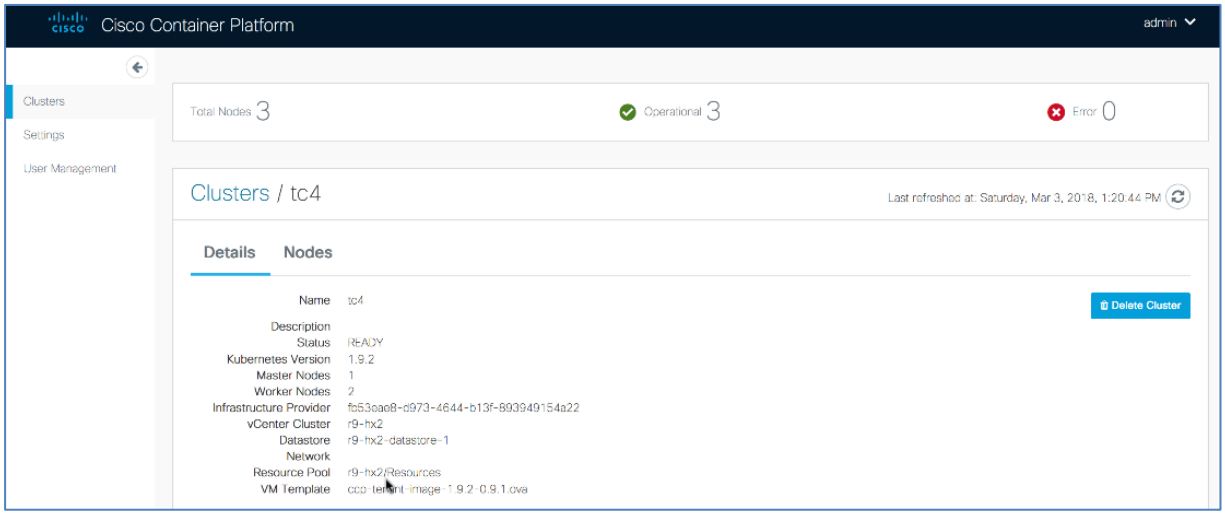
```
kubectl get nodes -o wide
```

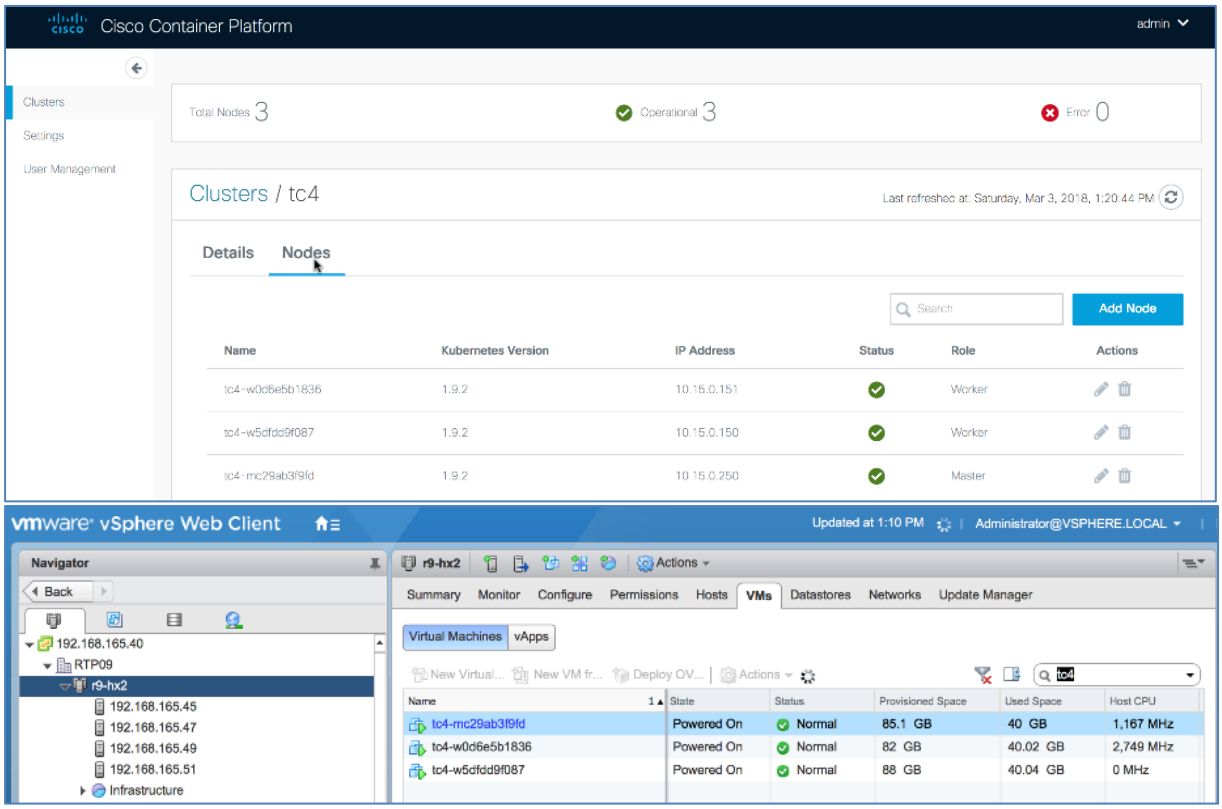
```
NAME              STATUS   ROLES    AGE   VERSION   EXTERNAL-IP     OS-IMAGE             KERNEL VERSION          CONTAINER RUNTIME
tc4-mc29ab3f9fd   Ready    master   3m    v1.9.2    10.15.0.250     Ubuntu 16.04.3 LTS   4.4.0-104-generic       Docker://1.13.1

tc4-w0d6e5b1836   Ready    <none>   2m    v1.9.2    10.15.0.151     Ubuntu 16.04.3 LTS   4.4.0-104-generic       Docker://1.13.1

Tc4-w5dfdd9f087   Ready    <none>   2m    v1.9.2    10.15.0.150     Ubuntu 16.04.3 LTS   4.4.0-104-generic       Docker://1.13.1
```







5.2 Deleting a Tenant Cluster

Before you Begin

Ensure that curl and jq are installed on your client machine.

Procedure

1. Export web IP:Port to MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Web UI IP:PORT>

   Example
   ```
   export MGMT_HOST=10.20.30.40:32443
   ```

2. Obtain a cookie using the username and password for your Cisco Container Platform instance.

   Command
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/

   Example
   ```
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/
   ```

3. List tenant clusters.

   Command
   curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r '.[].name, .uuid'

Example

```
curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r
'.[].name, .uuid'
tc1
aef65a35-c013-4d91-9edb-e2ef8359f95b
tc2
8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
tc3
a523fce7-b71e-444a-9626-871e17fe1fcd
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

4. Export the tenant cluster.

   Command

   export TC=<selected cluster from list>

   Example

   ```
   export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
   ```

5. Delete the tenant cluster.

   Command

   curl -sk -b cookie.txt -X DELETE https://$MGMT_HOST/2/clusters/${TC}

   Example

   ```
   curl -sk -b cookie.txt -X DELETE
   https://$MGMT_HOST/2/clusters/${TC}
   ```

5.3 Configuring Windows AD Service Account for Authentication

Before you Begin

Ensure that curl and jq are installed on your client machine.

Procedure

1. Export web IP:Port to MGMT_HOST environment variable.

   Command

   export MGMT_HOST=<Web UI IP:PORT>

   Example

   ```
   export MGMT_HOST=10.20.30.40: 32443
   ```

2. Obtain a cookie using the username and password for your Cisco Container Platform instance.

   Command

   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/

   Example

   ```
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-
   urlencoded" -d "username=admin&password=<Password from the
   installer>" https://$MGMT_HOST/2/system/login/
   ```

3. Query Windows AD server to verify the Service Account connection and members of the Cisco Container Platform accounts.

Command

ldapsearch -x -h <AD Server> -D "<Bind Distinguished Name>" -w '<Password>' -b "<Base Distinguished Name>" -s "<Scope>"

Example

```
ldapsearch -x -h 192.0.2.1 -D "CN=Adam A.
Arkanis,CN=Users,DC=r9-hx,DC=local" -w 'Password' -b "dc=r9-
hx,dc=local" -s sub "(cn=CCP*)" member cn

# extended LDIF
#
# LDAPv3
# base <dc=r9-hx,dc=local> with scope subtree
# filter: (cn=CCP*)
# requesting: member cn
#

# CCPAdmins, Users, r9-hx.local
dn: CN=CCPAdmins,CN=Users,DC=r9-hx,DC=local
cn:  CCPAdmins
member: CN=Andrew A.  Andres,CN=Users,DC=r9-hx,DC=local
member: CN=Adam A.  Arkanis,CN=Users,DC=r9-hx,DC=local

# CCPDevOps, Users, r9-hx.local
dn: CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local
cn: CCPDevOps
member: CN=Bob B. Bondurant,CN=Users,DC=r9-hx,DC=local
member: CN=Becky B. Bartholemew,CN=Users,DC=r9-hx,DC=local
```

4. Create json payload file for creating AD service account in Cisco Container Platform.

Command

```
cat << EOF > ldap_serviceaccount.json
{
 "Server": ” <AD Server>”,
 "Port": 3268,
 "ServiceAccountDN": "<Bind Distinguished Name>",
 "ServiceAccountPassword": "<Password>",
 "StartTLS": false,
 "InsecureSkipVerify": true
}
EOF
```

Example

```
cat << EOF > ldap_serviceaccount.json
{
  "Server": ” 192.0.2.1”,
  "Port": 3268,
  "ServiceAccountDN": "CN=Adam A. Arkanis,CN=Users,DC=r9-
hx,DC=local",
  "ServiceAccountPassword": "Password",
  "StartTLS": false,
  "InsecureSkipVerify": true
}
EOF
```

5. Create the service account for Cisco Container Platform.

   Command
   curl -sk -b cookie.txt -X PUT -H "Content-Type: application/json" -d @ldap_serviceaccount.json https://$MGMT_HOST/2/ldap/setup

   Example
   ```
   curl -sk -b cookie.txt -X PUT -H "Content-Type: application/json" -d @ldap_serviceaccount.json https://$MGMT_HOST/2/ldap/setup
   {
       "Server": " 192.0.2.1",
       "Port": 3268,
       "BaseDN": "DC=r9-hx,DC=local",
       "ServiceAccountDN": "CN=Adam A. Arkanis,CN=Users,DC=r9-hx,DC=local",
       "ServiceAccountPassword": "",
       "StartTLS": false,
       "InsecureSkipVerify": true
   }
   ```

6. Confirm service account configuration.

   Command
   curl -k -b cookie.txt https://$MGMT_HOST/2/ldap/setup

   Example
   ```
   curl -k -b cookie.txt https://$MGMT_HOST/2/ldap/setup
   {
        "Server": " 192.0.2.1",
        "Port": 3268,
        "BaseDN": "DC=r9-hx,DC=local",
        "ServiceAccountDN": "CN=Adam A. Arkanis,CN=Users,DC=r9-hx,DC=local",
        "ServiceAccountPassword":  "",
        "StartTLS":  false,
        "InsecureSkipVerify":  true
   }
   ```

5.4 Assigning and Removing Windows AD Group to Tenant Cluster

Before you Begin

Ensure that curl and jq are installed on your client machine.

Procedure

1. Export web IP:Port to MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Web UI IP:PORT>

   Example
   ```
   export MGMT_HOST=10.20.30.40:32443
   ```

2. Obtain a cookie using the username and password for your Cisco Container Platform instance.

Command
curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/

Example
```
curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=<Password from the installer>" https://$MGMT_HOST/2/system/login/
```

3. Create json payload file for assigning an AD group to a SysAdmin or DevOps role.

```
cat << EOF > ldap_devops_group.json
{
  "LdapDN": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "Role": "DevOps"
}
EOF
```

4. Create an LDAP group.
An error message is displayed, if an LDAP group already exists and can continue with script.

Command
curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d @ldap_devops_group.json https://$MGMT_HOST/2/ldap/groups

Example
```
curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d @ldap_devops_group.json https://$MGMT_HOST/2/ldap/groups
```

```
{
  “LdapDN”: “CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local”,
  “Role”: “DevOps”
}
```

5. Get list of configured AD groups in Cisco Container Platform.

Command
curl -sk -b cookie.txt https://$MGMT_HOST/2/ldap/groups

Example
```
curl -sk -b cookie.txt https://$MGMT_HOST/2/ldap/groups
```

```
[
 {
  “LdapDN”: “CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local”,
  “Role”: “DevOps”
 }
]

#Return list of clusters to assign AD group to
```

6. Get list of clusters for which you want to assign an AD group.

Command
curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'

Example
```
curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r
```

```
'.[]|.name, .uuid'
tc1
aef65a35-c013-4d91-9edb-e2ef8359f95b
tc2
8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
tc3
a523fce7-b71e-444a-9626-871e17fe1fcd
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

7. Export the selected tenant cluster.

   Command
   export TC=<Selected tenant cluster>

   Example
   ```
   export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
   ```

8. Create a json payload for assigning AD group to a tenant cluster.
   ```
   cat << EOF > ldap_authz.json
   {
     "name": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
     "local": false
    }
   EOF
   ```

9. Authorize group access to the selected tenant cluster.

   Command
   curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d @ldap_authz.json https://$MGMT_HOST/2/clusters/${TC}/authz

   Example
   ```
   curl -sk -b cookie.txt -X POST -H "Content-Type:
   application/json" -d @ldap_authz.json
   https://$MGMT_HOST/2/clusters/${TC}/authz

   {
     “AuthID”: “743e54da-037e-4386-99a7-a3da36e51936”,
     “Name”: “CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local”,
     “Local”: false
   }
   ```

10. Verify authorization of AD group to the tenant cluster.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/authz

    Example
    ```
    curl -sk -b cookie.txt
    https://$MGMT_HOST/2/clusters/${TC}/authz

    {
      “AuthList”:[
        {
         “AuthID”: “743e54da-037e-4386-99a7-a3da36e51936”,
         “Name”: “CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local”,
         “Local”: false
    ```

```
      }
    ]
}
```

11. Authenticate as a user from an AD DevOps group.

    Command
    curl -sk -c cookie_user.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=<AD User>&password=<Password>" https://$MGMT_HOST/2/system/login/

    Example
    ```
    curl -sk -c cookie_user.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=BobBB&password=Password" https://$MGMT_HOST/2/system/login/
    ```

12. Verify tenant cluster access list for an AD user.

    Command
    curl -sk -b cookie_user.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'

    Example
    ```
    curl -sk -b cookie_user.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'

    tc4

    8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
    ```

13. Export the selected tenant cluster.

    Command
    export TC=<Selected tenant cluster>

    Example
    ```
    export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
    ```

14. Download the KUBECONFIG environment file.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env

    Example
    ```
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env
    ```

15. Export the config file to KUBECONFIG environment variable.

    Command
    export KUBECONFIG=./${TC}.env

    Example
    ```
    export KUBECONFIG=./${TC}.env
    ```

16. View nodes on the tenant cluster.

    Command
    kubectl get nodes -o wide

    Example
    ```
    kubectl get nodes -o wide

    NAME            STATUS   ROLES   AGE   VERSION   EXTERNAL-IP  OS-IMAGE          KERNEL VERSION
    CONTAINER-RUNTIME
    ```

```
tc4-mc29ab3f9fd   Ready    master   1h    v1.9.2    10.20.30.250   Ubuntu 16.04.3 LTS   4.4.0-104-generic
docker://1.13.1

tc4-w0d6e5b1836   Ready    <none>   1h    v1.9.2    10.20.30.151   Ubuntu 16.04.3 LTS   4.4.0-104-generic
docker://1.13.1

tc4-w5dfdd9f087   Ready    <none>   1h    v1.9.2    10.20.30.150   Ubuntu 16.04.3 LTS   4.4.0-104-generic
docker://1.13.1
```

17. Remove AD group access.

    Command
    #curl -sk -b cookie.txt -X DELETE https://$MGMT_HOST/2/ldap/groups/<DN of Group>

    Example
    ```
    curl -sk -b cookie.txt -X DELETE
    https://$MGMT_HOST/2/ldap/groups/CN=CCPDevOps,CN=Users,DC=r9-
    hx,DC=local
    ```

18. Verify that authorization of AD group to tenant cluster is removed.

    Command
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/authz

    Example
    ```
    curl -sk -b cookie.txt
    https://$MGMT_HOST/2/clusters/${TC}/authz
    ```

    ```
    {
      "AuthList": []
    }
    ```

5.5 Download Tenant Cluster KUBECONFIG Environment File

Before you Begin

Ensure that curl and jq are installed on your client machine.

Procedure

1. Export web IP:Port to MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Web UI IP:PORT>

   Example
   ```
   export MGMT_HOST=10.20.30.40: 32443
   ```

2. Obtain a cookie using the username and password for your Cisco Container Platform instance.

   Command
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password==<Password from the installer>" https://$MGMT_HOST/2/system/login/

   Example
   ```
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-
   urlencoded" -d "username=admin&password=<Password from the
   installer>" https://$MGMT_HOST/2/system/login/
   ```

3. List tenant clusters.

   Command
   curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid

Example

```
curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r
'.[]|.name, .uuid'
tc1
aef65a35-c013-4d91-9edb-e2ef8359f9gb
tc2
8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
tc3
a523fce7-b71e-444a-9626-871e17fe1fcd
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

4. Export a tenant cluster.

   Command
   export TC=<selected cluster from list>

   Example

   ```
   export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
   ```

5. Download the KUBECONFIG environmental file.

   Command
   curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env

   Example

   ```
   curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env
   -o ${TC}.env
   ```

6. Export the config file to KUBECONFIG environment variable.

   Command
   export KUBECONFIG=./${TC}.env

   Example

   ```
   export KUBECONFIG=./${TC}.env
   ```

7. View nodes on the tenant cluster.
   Command
   kubectl get nodes -o wide

   Example

   ```
   kubectl get nodes -o wide
   ```

```
NAME             STATUS   ROLES    AGE   VERSION   EXTERNAL-IP   OS-IMAGE           KERNEL VERSION     CONTAINER-RUNTIME
tc4-mc29ab3f9fd   Ready    master   1h    v1.9.2    10.20.30.250  Ubuntu 16.04.3 LTS  4.4.0-104-generic  docker://1.13.1
tc4-w0d6e5b1836   Ready    <none>   1h    v1.9.2    10.20.30.151  Ubuntu 16.04.3 LTS  4.4.0-104-generic  docker://1.13.1
tc4-w5dfdd9f087   Ready    <none    1h    v1.9.2    10.20.30.150  Ubuntu 16.04.3 LTS  4.4.0-104-generic  docker://1.13.1
```

5.6 Obtaining TC Master and Ingress VIPs

FOR MASTER

```
`curl -sk -X GET -b temp/cookie.txt
https://$MGMT_HOST/2/clusters/<clustername> | jq  '.master_vip`
```

FOR INGRESS VIPS

```
`curl -sk -X GET -b temp/cookie.txt
https://$MGMT_HOST/2/clusters/<cluster> | jq  '.ingress_vips`
```

6 Cisco Container Platform API Reference

2/system : List of system endpoints

POST /2/system/login — Management server login

Parameters

Parameter	Value	Description	Parameter Type	Data Type
username		User Name	formData	string
password		Password	formData	string
token		JWT Token	formData	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		

default

Try it out!

GET /2/system/livenessHealth — Returns a string representing the health of the system

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		

default

Try it out!

GET /2/system/health — Returns the health of the system

Response Class (Status 200)

OK

Model | Example Value


```
{
  "TotalSystemHealth": "string",
  "CurrentNodes": 0,
  "ExpectedNodes": 0,
  "NodesStatus": [
    {
      "NodeName": "string",
      "NodeCondition": "string",
      "NodeStatus": "string",
      "LastTransitionTime": "string"
    }
  ],
  "PodStatusList": [
    {
      "PodName": "string",
      "PodCondition": "string",
      "PodStatus": "string",
      "LastTransitionTime": "string"
    }
  ]
}
```


Response Content Type application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/system/CorcHealth — Get corc health

Response Class (Status 200)

OK

Model | Example Value

```
{}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value {}

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default		Model | Example Value {}	

Try it out!

2/providerclientconfigs : List of provider client config endpoints

GET /2/providerclientconfigs — Get provider client configuration list

Parameters

Parameter	Value	Description	Parameter Type	Data Type
name		Provider Client Config Name	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Config found		
401	Unauthorized		
404	Config not found		
default			

Try it out!

POST /2/providerclientconfigs — Add provider client configuration

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

```
{
  "uuid": "string",
  "name": "string",
  "config": {}
}
```

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

HTTP Status Code	Reason	Response Model	Headers

201	Added config successfully	Model | Example Value
		{
		"uuid": "string",
		"name": "string",
		"config": {}
		}	

400	Bad request		
401	Unauthorized		

Try it out!

DELETE /2/providerclientconfigs/{clientconfigUUID} Delete provider client configuration

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204	Deleted config successfully		
400	Config in use		
401	Unauthorized		
404	Config not found		

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID} Get provider client configuration

Response Class (Status 200)

Config found

Model | Example Value


```
{
  "uuid": "string",
  "name": "string",
  "config": {}
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Config not found		
default			

Try it out!

PATCH /2/providerclientconfigs/{clientconfigUUID} Update provider client configuration

Response Class (Status 200)

Config found

Model | Example Value


```
{
  "uuid": "string",
  "name": "string",
  "config": {}
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)** Parameter content type: application/json ▼		body	Model | Example Value `{ "uuid": "string", "name": "string", "config": {} }`
clientconfigUUID	**(required)**	**Client Config UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Config not found		
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/clusters — Get list of clusters who are using providerclientconfig

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Clusters found		
401	Unauthorized		
404	Clusters not found		
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter — Gets the list of vSphere Data Centers.

Response Class (Status 200)

OK

Model | Example Value


```
{
  "Datacenters": [
    "string"
  ]
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/cluster

Gets the list of vSphere Clusters in a datacenter.

Response Class (Status 200)

OK

Model | Example Value


```
{
  "Clusters": [
    "string"
  ]
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string
datacenterName	**(required)**	**Datacenter Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/vm Gets the list of vSphere Virtual Machines.

Response Class (Status 200)

OK

Model | Example Value

```
{
  "VMs": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string
datacenterName	**(required)**	**Datacenter Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/network Gets the list of vSphere Networks.

Response Class (Status 200)

OK

Model | Example Value


```
{
  "Networks": [
    "string"
  ]
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string
datacenterName	**(required)**	**Datacenter Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/datastore Gets the list of vSphere Datastores.

Response Class (Status 200)

OK

Model | Example Value


```
{
  "Datastores": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Client Config UUID**	path	string
datacenterName	**(required)**	**Datacenter Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/cluster/{clusterName}/pool

Gets the list of vSphere Pools.

Response Class (Status 200)

OK

Model | Example Value

```
{
  "Pools": [
    "string"
  ]
}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
clientconfigUUID	**(required)**	**Datacenter Name**	path	string
datacenterName	**(required)**	**Datacenter Name**	path	string
clusterName	**(required)**	**Cluster Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

2/clusters : List of cluster endpoints

GET /2/clusters — Get all clusters

Response Class (Status 200)

Clusters found

Model | Example Value


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
```

```
"Infra": {},
"labels": [
  {
    "key": "string",
    "value": "string"
  }
],
"nodes": [
  {
    "uuid": "string",
    "name": "string",
    "ip_info": [
      {
        "IPInfo": {
          "id": 0,
          "uuid": "string",
          "ip": "string",
          "subnet": "string",
          "gateway": "string",
          "nameservers": [
            "string"
          ],
          "netmask": "string",
          "mtu": 0
        },
        "if_name": "string",
        "dhcp": true,
        "hflex_volume;omitempty": true
      }
    ],
    "public_ip": "string",
    "private_ip": "string",
    "is_master": true,
    "state": "string",
    "cloud_init_data": "string",
    "kubernetes_version": "string",
    "error_log": "string",
    "template": "string",
```

```
      "mac_addresses": [
        "string"
      ],
      "node_pool_type": "string",
      "node_pool_id": 0
    }
  ],
  "deployer": {
    "provider_type": "string",
    "provider": {
      "vsphere_datacenter": "string",
      "vsphere_datastore": "string",
      "vsphere_scsi_controller_type": "string",
      "vsphere_working_dir": "string",
      "vsphere_client_config_uuid": "string",
      "client_config;omitempty": {
        "ip": "string",
        "port": 0,
        "username": "string",
        "password": "string"
      }
    }
  },
  "kubernetes_version": "string",
  "cluster_env_url": "string",
  "cluster_dashboard_url": "string",
  "network_plugin": {
    "name": "string",
    "status": "string",
    "details": "string"
  },
  "node_ip_pool_uuid": "string",
  "ccp_private_ssh_key": "string",
  "ccp_public_ssh_key": "string",
  "ntp_pools": [
    "string"
  ],
  "ntp_servers": [
```

```
    "string"
  ],
  "is_control_cluster": true,
  "is_adopt": true,
  "registries_self_signed": [
    "string"
  ],
  "registries_insecure": [
    "string"
  ],
  "registries_root_ca": [
    "string"
  ],
  "ingress_vip_pool_id": "string",
  "ingress_vip_addr_id": "string",
  "ingress_vips": [
    "string"
  ],
  "helm_charts": [
    {
      "helmchart_uuid": "string",
      "cluster_UUID": "string",
      "chart_url": "string",
      "name": "string",
      "options": "string"
    }
  ],
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
```

```
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Content Type application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	No cluster found		
default			

Try it out!

POST /2/clusters

Create a cluster with the given specification

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)**		body	Model | Example Value

Parameter content type: application/json ▼


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
      "ip_info": [
        {
          "IPInfo": {
            "id": 0,
            "uuid": "string",
            "ip": "string",
            "subnet": "string",
            "gateway": "string",
            "nameservers": [
              "string"
```

Parameter	Value	Description	Parameter Type	Data Type


```
            ],
            "netmask": "string",
            "mtu": 0
          },
          "if_name": "string",
          "dhcp": true,
          "hflex_volume;omitempty": true
        }
      ],
      "public_ip": "string",
      "private_ip": "string",
      "is_master": true,
      "state": "string",
      "cloud_init_data": "string",
      "kubernetes_version": "string",
      "error_log": "string",
      "template": "string",
      "mac_addresses": [
        "string"
      ],
      "node_pool_type": "string",
      "node_pool_id": 0
    }
  ],
  "deployer": {
    "provider_type": "string",
    "provider": {
      "vsphere_datacenter": "string",
      "vsphere_datastore": "string",
      "vsphere_scsi_controller_type": "string",
      "vsphere_working_dir": "string",
      "vsphere_client_config_uuid": "string",
      "client_config;omitempty": {
        "ip": "string",
        "port": 0,
        "username": "string",
        "password": "string"
      }
    }
  },
  "kubernetes_version": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
"cluster_env_url": "string",
"cluster_dashboard_url": "string",
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
  "string"
],
"ingress_vip_pool_id": "string",
"ingress_vip_addr_id": "string",
"ingress_vips": [
  "string"
],
"helm_charts": [
  {
    "helmchart_uuid": "string",
    "cluster_UUID": "string",
    "chart_url": "string",
    "name": "string",
    "options": "string"
  }
],
```

Parameter	Value	Description	Parameter Type	Data Type


```
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201			

HTTP Status Code	Reason	Response Model	Headers
	Created cluster successfully		

Model | Example Value


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
      "ip_info": [
        {
          "IPInfo": {
            "id": 0,
            "uuid": "string",
            "ip": "string",
            "subnet": "string",
            "gateway": "string",
            "nameservers": [
              "string"
            ],
            "netmask": "string",
            "mtu": 0
          },
          "if_name": "string",
```

HTTP Status Code	Reason	Response Model	Headers


```
            "dhcp": true,
            "hflex_volume;omitempty": true
          }
        ],
        "public_ip": "string",
        "private_ip": "string",
        "is_master": true,
        "state": "string",
        "cloud_init_data": "string",
        "kubernetes_version": "string",
        "error_log": "string",
        "template": "string",
        "mac_addresses": [
          "string"
        ],
        "node_pool_type": "string",
        "node_pool_id": 0
      }
    ],
    "deployer": {
      "provider_type": "string",
      "provider": {
        "vsphere_datacenter": "string",
        "vsphere_datastore": "string",
        "vsphere_scsi_controller_type": "string",
        "vsphere_working_dir": "string",
        "vsphere_client_config_uuid": "string",
        "client_config;omitempty": {
          "ip": "string",
          "port": 0,
          "username": "string",
          "password": "string"
        }
      }
    },
    "kubernetes_version": "string",
    "cluster_env_url": "string",
    "cluster_dashboard_url": "string",
    "network_plugin": {
      "name": "string",
      "status": "string",
      "details": "string"
```

HTTP Status Code	Reason	Response Model	Headers


```
  },
  "node_ip_pool_uuid": "string",
  "ccp_private_ssh_key": "string",
  "ccp_public_ssh_key": "string",
  "ntp_pools": [
    "string"
  ],
  "ntp_servers": [
    "string"
  ],
  "is_control_cluster": true,
  "is_adopt": true,
  "registries_self_signed": [
    "string"
  ],
  "registries_insecure": [
    "string"
  ],
  "registries_root_ca": [
    "string"
  ],
  "ingress_vip_pool_id": "string",
  "ingress_vip_addr_id": "string",
  "ingress_vips": [
    "string"
  ],
  "helm_charts": [
    {
      "helmchart_uuid": "string",
      "cluster_UUID": "string",
      "chart_url": "string",
      "name": "string",
      "options": "string"
    }
  ],
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
```

HTTP Status Code	Reason	Response Model	Headers

		{
		"IPInfo": {
		"id": 0,
		"uuid": "string",
		"ip": "string",
		"subnet": "string",
		"gateway": "string",
		"nameservers": [
		"string"
		],
		"netmask": "string",
		"mtu": 0
		},
		"never_release": true
		}
		],
		"node_pools": [
		{}
		],
		"is_istio_enabled": true,
		"is_harbor_enabled": true,
		"harbor_registry_size": "string",
		"harbor_admin_server_password": "string",
		"master_node_pool": {},
		"worker_node_pool": {},
		"storage_class": "string"
		}	

400	Bad request		
401	Unauthorized		

Try it out!

DELETE /2/clusters/{clusterUUID} Delete a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204	Deleted cluster successfully		
401	Unauthorized		
404	Cluster not found		

Try it out!

PATCH /2/clusters/{clusterUUID} Patch a cluster

Response Class (Status 200)

Cluster patched successfully

Model | Example Value


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
```

```
"labels": [
  {
    "key": "string",
    "value": "string"
  }
],
"nodes": [
  {
    "uuid": "string",
    "name": "string",
    "ip_info": [
      {
        "IPInfo": {
          "id": 0,
          "uuid": "string",
          "ip": "string",
          "subnet": "string",
          "gateway": "string",
          "nameservers": [
            "string"
          ],
          "netmask": "string",
          "mtu": 0
        },
        "if_name": "string",
        "dhcp": true,
        "hflex_volume;omitempty": true
      }
    ],
    "public_ip": "string",
    "private_ip": "string",
    "is_master": true,
    "state": "string",
    "cloud_init_data": "string",
    "kubernetes_version": "string",
    "error_log": "string",
    "template": "string",
    "mac_addresses": [
```

```
      "string"
    ],
    "node_pool_type": "string",
    "node_pool_id": 0
  }
],
"deployer": {
  "provider_type": "string",
  "provider": {
    "vsphere_datacenter": "string",
    "vsphere_datastore": "string",
    "vsphere_scsi_controller_type": "string",
    "vsphere_working_dir": "string",
    "vsphere_client_config_uuid": "string",
    "client_config;omitempty": {
      "ip": "string",
      "port": 0,
      "username": "string",
      "password": "string"
    }
  }
},
"kubernetes_version": "string",
"cluster_env_url": "string",
"cluster_dashboard_url": "string",
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
```

```
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
  "string"
],
"ingress_vip_pool_id": "string",
"ingress_vip_addr_id": "string",
"ingress_vips": [
  "string"
],
"helm_charts": [
  {
    "helmchart_uuid": "string",
    "cluster_UUID": "string",
    "chart_url": "string",
    "name": "string",
    "options": "string"
  }
],
"master_vip_addr_id": "string",
"master_vip": "string",
"master_mac_addresses": [
  "string"
],
"load_balancer_ip_num": 0,
"load_balancer_ip_info_list": [
  {
    "IPInfo": {
      "id": 0,
      "uuid": "string",
      "ip": "string",
```

```
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)** Parameter content type: application/json ▼		body	Model | Example Value


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
```

Parameter	Value	Description	Parameter Type	Data Type


```
"state": "string",
"template": "string",
"ssh_user": "string",
"ssh_password": "string",
"ssh_key": "string",
"Infra": {},
"labels": [
  {
    "key": "string",
    "value": "string"
  }
],
"nodes": [
  {
    "uuid": "string",
    "name": "string",
    "ip_info": [
      {
        "IPInfo": {
          "id": 0,
          "uuid": "string",
          "ip": "string",
          "subnet": "string",
          "gateway": "string",
          "nameservers": [
            "string"
          ],
          "netmask": "string",
          "mtu": 0
        },
        "if_name": "string",
        "dhcp": true,
        "hflex_volume;omitempty": true
      }
    ],
    "public_ip": "string",
    "private_ip": "string",
    "is_master": true,
    "state": "string",
    "cloud_init_data": "string",
    "kubernetes_version": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
      "error_log": "string",
      "template": "string",
      "mac_addresses": [
        "string"
      ],
      "node_pool_type": "string",
      "node_pool_id": 0
    }
  ],
  "deployer": {
    "provider_type": "string",
    "provider": {
      "vsphere_datacenter": "string",
      "vsphere_datastore": "string",
      "vsphere_scsi_controller_type": "string",
      "vsphere_working_dir": "string",
      "vsphere_client_config_uuid": "string",
      "client_config;omitempty": {
        "ip": "string",
        "port": 0,
        "username": "string",
        "password": "string"
      }
    }
  },
  "kubernetes_version": "string",
  "cluster_env_url": "string",
  "cluster_dashboard_url": "string",
  "network_plugin": {
    "name": "string",
    "status": "string",
    "details": "string"
  },
  "node_ip_pool_uuid": "string",
  "ccp_private_ssh_key": "string",
  "ccp_public_ssh_key": "string",
  "ntp_pools": [
    "string"
  ],
  "ntp_servers": [
    "string"
```

Parameter	Value	Description	Parameter Type	Data Type


```
  ],
  "is_control_cluster": true,
  "is_adopt": true,
  "registries_self_signed": [
    "string"
  ],
  "registries_insecure": [
    "string"
  ],
  "registries_root_ca": [
    "string"
  ],
  "ingress_vip_pool_id": "string",
  "ingress_vip_addr_id": "string",
  "ingress_vips": [
    "string"
  ],
  "helm_charts": [
    {
      "helmchart_uuid": "string",
      "cluster_UUID": "string",
      "chart_url": "string",
      "name": "string",
      "options": "string"
    }
  ],
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
```

Parameter	Value	Description	Parameter Type	Data Type


```
            "string"
          ],
          "netmask": "string",
          "mtu": 0
        },
        "never_release": true
      }
    ],
    "node_pools": [
      {}
    ],
    "is_istio_enabled": true,
    "is_harbor_enabled": true,
    "harbor_registry_size": "string",
    "harbor_admin_server_password": "string",
    "master_node_pool": {},
    "worker_node_pool": {},
    "storage_class": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Cluster not found		
default			

Try it out!

PUT /2/clusters/{clusterUUID} Update a cluster

Response Class (Status 200)

Cluster updated successfully

Model | Example Value

```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
      "ip_info": [
        {
          "IPInfo": {
            "id": 0,
            "uuid": "string",
            "ip": "string",
            "subnet": "string",
            "gateway": "string",
            "nameservers": [
              "string"
            ],
            "netmask": "string",
            "mtu": 0
          },
```

```
        "if_name": "string",
        "dhcp": true,
        "hflex_volume;omitempty": true
      }
    ],
    "public_ip": "string",
    "private_ip": "string",
    "is_master": true,
    "state": "string",
    "cloud_init_data": "string",
    "kubernetes_version": "string",
    "error_log": "string",
    "template": "string",
    "mac_addresses": [
      "string"
    ],
    "node_pool_type": "string",
    "node_pool_id": 0
  }
],
"deployer": {
  "provider_type": "string",
  "provider": {
    "vsphere_datacenter": "string",
    "vsphere_datastore": "string",
    "vsphere_scsi_controller_type": "string",
    "vsphere_working_dir": "string",
    "vsphere_client_config_uuid": "string",
    "client_config;omitempty": {
      "ip": "string",
      "port": 0,
      "username": "string",
      "password": "string"
    }
  }
},
"kubernetes_version": "string",
"cluster_env_url": "string",
```

```
"cluster_dashboard_url": "string",
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
  "string"
],
"ingress_vip_pool_id": "string",
"ingress_vip_addr_id": "string",
"ingress_vips": [
  "string"
],
"helm_charts": [
  {
    "helmchart_uuid": "string",
    "cluster_UUID": "string",
    "chart_url": "string",
    "name": "string",
    "options": "string"
```

```
    }
  ],
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

Parameter content type:
application/json ▼


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
      "ip_info": [
        {
          "IPInfo": {
            "id": 0,
            "uuid": "string",
            "ip": "string",
            "subnet": "string",
            "gateway": "string",
            "nameservers": [
              "string"
            ],
            "netmask": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
            "mtu": 0
          },
          "if_name": "string",
          "dhcp": true,
          "hflex_volume;omitempty": true
        }
      ],
      "public_ip": "string",
      "private_ip": "string",
      "is_master": true,
      "state": "string",
      "cloud_init_data": "string",
      "kubernetes_version": "string",
      "error_log": "string",
      "template": "string",
      "mac_addresses": [
        "string"
      ],
      "node_pool_type": "string",
      "node_pool_id": 0
    }
  ],
  "deployer": {
    "provider_type": "string",
    "provider": {
      "vsphere_datacenter": "string",
      "vsphere_datastore": "string",
      "vsphere_scsi_controller_type": "string",
      "vsphere_working_dir": "string",
      "vsphere_client_config_uuid": "string",
      "client_config;omitempty": {
        "ip": "string",
        "port": 0,
        "username": "string",
        "password": "string"
      }
    }
  },
  "kubernetes_version": "string",
  "cluster_env_url": "string",
  "cluster_dashboard_url": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
  "string"
],
"ingress_vip_pool_id": "string",
"ingress_vip_addr_id": "string",
"ingress_vips": [
  "string"
],
"helm_charts": [
  {
    "helmchart_uuid": "string",
    "cluster_UUID": "string",
    "chart_url": "string",
    "name": "string",
    "options": "string"
  }
],
"master_vip_addr_id": "string",
"master_vip": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Cluster not found		

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/clusters/{clusterName} Get a cluster by name

Response Class (Status 200)

Cluster found

Model | Example Value


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
```

```
    "ip_info": [
      {
        "IPInfo": {
          "id": 0,
          "uuid": "string",
          "ip": "string",
          "subnet": "string",
          "gateway": "string",
          "nameservers": [
            "string"
          ],
          "netmask": "string",
          "mtu": 0
        },
        "if_name": "string",
        "dhcp": true,
        "hflex_volume;omitempty": true
      }
    ],
    "public_ip": "string",
    "private_ip": "string",
    "is_master": true,
    "state": "string",
    "cloud_init_data": "string",
    "kubernetes_version": "string",
    "error_log": "string",
    "template": "string",
    "mac_addresses": [
      "string"
    ],
    "node_pool_type": "string",
    "node_pool_id": 0
  }
],
"deployer": {
  "provider_type": "string",
  "provider": {
    "vsphere_datacenter": "string",
```

```
    "vsphere_datastore": "string",
    "vsphere_scsi_controller_type": "string",
    "vsphere_working_dir": "string",
    "vsphere_client_config_uuid": "string",
    "client_config;omitempty": {
      "ip": "string",
      "port": 0,
      "username": "string",
      "password": "string"
    }
  }
},
"kubernetes_version": "string",
"cluster_env_url": "string",
"cluster_dashboard_url": "string",
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
```

```
    "string"
  ],
  "ingress_vip_pool_id": "string",
  "ingress_vip_addr_id": "string",
  "ingress_vips": [
    "string"
  ],
  "helm_charts": [
    {
      "helmchart_uuid": "string",
      "cluster_UUID": "string",
      "chart_url": "string",
      "name": "string",
      "options": "string"
    }
  ],
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
```

```
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterName	**(required)**	**Cluster Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Cluster not found		
default			

Try it out!

PATCH /2/clusters/{clusterUUID}/upgrade — Upgrade a cluster

Response Class (Status 200)

Cluster updated successfully

Model | Example Value

```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
      "ip_info": [
        {
          "IPInfo": {
            "id": 0,
            "uuid": "string",
            "ip": "string",
            "subnet": "string",
            "gateway": "string",
            "nameservers": [
              "string"
            ],
            "netmask": "string",
            "mtu": 0
          },
```

```
        "if_name": "string",
        "dhcp": true,
        "hflex_volume;omitempty": true
      }
    ],
    "public_ip": "string",
    "private_ip": "string",
    "is_master": true,
    "state": "string",
    "cloud_init_data": "string",
    "kubernetes_version": "string",
    "error_log": "string",
    "template": "string",
    "mac_addresses": [
      "string"
    ],
    "node_pool_type": "string",
    "node_pool_id": 0
  }
],
"deployer": {
  "provider_type": "string",
  "provider": {
    "vsphere_datacenter": "string",
    "vsphere_datastore": "string",
    "vsphere_scsi_controller_type": "string",
    "vsphere_working_dir": "string",
    "vsphere_client_config_uuid": "string",
    "client_config;omitempty": {
      "ip": "string",
      "port": 0,
      "username": "string",
      "password": "string"
    }
  }
},
"kubernetes_version": "string",
"cluster_env_url": "string",
```

```
"cluster_dashboard_url": "string",
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
  "string"
],
"ingress_vip_pool_id": "string",
"ingress_vip_addr_id": "string",
"ingress_vips": [
  "string"
],
"helm_charts": [
  {
    "helmchart_uuid": "string",
    "cluster_UUID": "string",
    "chart_url": "string",
    "name": "string",
    "options": "string"
```

```
    }
  ],
  "master_vip_addr_id": "string",
  "master_vip": "string",
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)** Parameter content type: application/json ▼		body	Model | Example Value


```
{
  "uuid": "string",
  "provider_client_config_uuid": "string",
  "aci_profile_uuid": "string",
  "name": "string",
  "description": "string",
  "workers": 0,
  "masters": 0,
  "state": "string",
  "template": "string",
  "ssh_user": "string",
  "ssh_password": "string",
  "ssh_key": "string",
  "Infra": {},
  "labels": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "nodes": [
    {
      "uuid": "string",
      "name": "string",
      "ip_info": [
        {
          "IPInfo": {
            "id": 0,
            "uuid": "string",
            "ip": "string",
            "subnet": "string",
            "gateway": "string",
            "nameservers": [
              "string"
            ],
            "netmask": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
            "mtu": 0
          },
          "if_name": "string",
          "dhcp": true,
          "hflex_volume;omitempty": true
        }
      ],
      "public_ip": "string",
      "private_ip": "string",
      "is_master": true,
      "state": "string",
      "cloud_init_data": "string",
      "kubernetes_version": "string",
      "error_log": "string",
      "template": "string",
      "mac_addresses": [
        "string"
      ],
      "node_pool_type": "string",
      "node_pool_id": 0
    }
  ],
  "deployer": {
    "provider_type": "string",
    "provider": {
      "vsphere_datacenter": "string",
      "vsphere_datastore": "string",
      "vsphere_scsi_controller_type": "string",
      "vsphere_working_dir": "string",
      "vsphere_client_config_uuid": "string",
      "client_config;omitempty": {
        "ip": "string",
        "port": 0,
        "username": "string",
        "password": "string"
      }
    }
  },
  "kubernetes_version": "string",
  "cluster_env_url": "string",
  "cluster_dashboard_url": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
"network_plugin": {
  "name": "string",
  "status": "string",
  "details": "string"
},
"node_ip_pool_uuid": "string",
"ccp_private_ssh_key": "string",
"ccp_public_ssh_key": "string",
"ntp_pools": [
  "string"
],
"ntp_servers": [
  "string"
],
"is_control_cluster": true,
"is_adopt": true,
"registries_self_signed": [
  "string"
],
"registries_insecure": [
  "string"
],
"registries_root_ca": [
  "string"
],
"ingress_vip_pool_id": "string",
"ingress_vip_addr_id": "string",
"ingress_vips": [
  "string"
],
"helm_charts": [
  {
    "helmchart_uuid": "string",
    "cluster_UUID": "string",
    "chart_url": "string",
    "name": "string",
    "options": "string"
  }
],
"master_vip_addr_id": "string",
"master_vip": "string",
```

Parameter	Value	Description	Parameter Type	Data Type


```
  "master_mac_addresses": [
    "string"
  ],
  "load_balancer_ip_num": 0,
  "load_balancer_ip_info_list": [
    {
      "IPInfo": {
        "id": 0,
        "uuid": "string",
        "ip": "string",
        "subnet": "string",
        "gateway": "string",
        "nameservers": [
          "string"
        ],
        "netmask": "string",
        "mtu": 0
      },
      "never_release": true
    }
  ],
  "node_pools": [
    {}
  ],
  "is_istio_enabled": true,
  "is_harbor_enabled": true,
  "harbor_registry_size": "string",
  "harbor_admin_server_password": "string",
  "master_node_pool": {},
  "worker_node_pool": {},
  "storage_class": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Cluster not found		

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/clusters/{clusterID}/authz — List authorizations for a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterID	**(required)**	**Cluster UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
401	Unauthorized		
404	Record not found		
default			

Try it out!

POST /2/clusters/{clusterID}/authz — Add authorization for a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterID	**(required)**	**Cluster UUID**	path	string

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

```
{
  "Name": "string",
  "Local": true
}
```

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201	OK	Model | Example Value	

```
{
  "AuthID": "string",
  "Name": "string",
  "Local": true
}
```

401	Unauthorized		
404	Record not found		

Try it out!

DELETE /2/clusters/{clusterID}/authz/{authID} Delete authorization for a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterID	(required)	**Cluster UUID**	path	string
authID	(required)	**Authorization UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204			
401	Unauthorized		
404	Record not found		

Try it out!

GET /2/clusters/{clusterUUID}/dashboard — Get dashboard

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
404	Record not found		
default			

Try it out!

GET /2/clusters/{clusterUUID}/env — Get cluster environment

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
404	Record not found		
default			

Try it out!

DELETE /2/clusters/{clusterUUID}/helmcharts/{HelmChartUUID} Delete helm chart for cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
HelmChartUUID	**(required)**	**HelmChartUUID**	path	string
clusterUUID	**(required)**	**Cluster UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204	Deleted helm chart successfully	Model | Example Value	


```
{
  "helmchart_uuid": "string",
  "cluster_UUID": "string",
  "chart_url": "string",
  "name": "string",
  "options": "string"
}
```

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	HelmChart not found		

Try it out!

GET /2/clusters/{clusterUUID}/helmcharts

Get HelmCharts object for a given cluster

Response Class (Status 200)

HelmCharts found

Model | Example Value


```
{
  "helmchart_uuid": "string",
  "cluster_UUID": "string",
  "chart_url": "string",
  "name": "string",
  "options": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUD	**(required)**	**Cluster UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	HelmCharts not found		
default			

Try it out!

POST /2/clusters/{clusterUUID}/helmcharts — Create a helmChart for cluster with the given specification

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string
body	**(required)**		body	Model | Example Value

Parameter content type: application/json ▼

```
{
  "helmchart_uuid": "string",
  "cluster_UUID": "string",
  "chart_url": "string",
  "name": "string",
  "options": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201	Created helmChart successfully	Model | Example Value	
400	Bad request		
401	Unauthorized		

```
{
  "helmchart_uuid": "string",
  "cluster_UUID": "string",
  "chart_url": "string",
  "name": "string",
  "options": "string"
}
```

Try it out!

POST /2/clusters/{clusterUUID}/nodepools — Create a node pool for a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string
body	**(required)**		body	Model | Example Value

Parameter content type: application/json ▼

```
{
  "name": "string",
  "size": 0,
  "labels": "string",
  "taints": "string",
  "subnet_uuid": "string",
  "vcpus": 0,
  "memory": 0,
  "template": "string"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201	Created nodepool successfully	Model | Example Value	
400	Bad request		
401	Unauthorized		
404	Cluster not found		

```
{
  "NodePool": {}
}
```

Try it out!

DELETE /2/clusters/{clusterUUID}/nodepools/{nodePoolID} Delete a node pool from a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string
nodePoolID	**(required)**	**Node Pool ID**	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204	Deleted nodepool successfully	Model | Example Value {}	
401	Unauthorized		
404	Cluster or NodePool not found		

Try it out!

PATCH /2/clusters/{clusterUUID}/nodepools/{nodePoolID} Update a node pool in a cluster

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	**(required)**	**Cluster UUID**	path	string
nodePoolID	**(required)**	**Node Pool ID**	path	integer

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

```
{
  "size": 0
}
```

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204	Updated nodepool successfully	Model | Example Value {}	
400	Bad request		
401	Unauthorized		
404	Cluster or NodePool not found		

Try it out!

2/ldap : List of ldap endpoints

GET /2/ldap/setup — Get LDAP parameters

Response Class (Status 200)

OK

Model | Example Value

```
{
  "Server": "string",
```

```
  "Port": 0,
  "BaseDN": "string",
  "ServiceAccountDN": "string",
  "ServiceAccountPassword": "string",
  "StartTLS": true,
  "InsecureSkipVerify": true
}
```


Response Content Type application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default		Model | Example Value	


```
{
  "Server": "string",
  "Port": 0,
  "BaseDN": "string",
  "ServiceAccountDN": "string",
  "ServiceAccountPassword": "string",
  "StartTLS": true,
  "InsecureSkipVerify": true
}
```


Try it out!

PUT /2/ldap/setup — Setup/update LDAP parameters

Response Class (Status 200)

OK

Model | Example Value


```
{
  "Server": "string",
  "Port": 0,
```

```
  "BaseDN": "string",
  "ServiceAccountDN": "string",
  "ServiceAccountPassword": "string",
  "StartTLS": true,
  "InsecureSkipVerify": true
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

Parameter content type: application/json ▼


```
{
  "Server": "string",
  "Port": 0,
  "BaseDN": "string",
  "ServiceAccountDN": "string",
  "ServiceAccountPassword": "string",
  "StartTLS": true,
  "InsecureSkipVerify": true
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
default		Model | Example Value	


```
{
  "Server": "string",
  "Port": 0,
  "BaseDN": "string",
  "ServiceAccountDN": "string",
  "ServiceAccountPassword": "string",
  "StartTLS": true,
  "InsecureSkipVerify": true
}
```

Try it out!

GET /2/ldap/groups — Get CX LDAP Groups

Parameters

Parameter	Value	Description	Parameter Type	Data Type
dn		LDAP DN	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
default			

Try it out!

POST /2/ldap/groups — Create CX LDAP Group

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

```
{
  "LdapDN": "string",
  "Role": "string"
}
```

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

HTTP Status Code	Reason	Response Model	Headers
201		Model | Example Value	

```
{
  "LdapDN": "string",
  "Role": "string"
}
```

Try it out!

PUT /2/ldap/groups — Update a CX LDAP Group.

Response Class (Status 200)

Model | Example Value

```
{
  "LdapDN": "string",
  "Role": "string"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)**		body	Model | Example Value

```
{
  "LdapDN": "string",
  "Role": "string"
}
```

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

GET /2/ldap/groups/authz — Get CX the cluster authorizations for a CX LDAP group

Parameters

Parameter	Value	Description	Parameter Type	Data Type
dn		LDAP DN	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
default			

Try it out!

DELETE /2/ldap/groups/{ldapDN} — Delete CX LDAP Group specified by LDAP DN

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204			

Try it out!

2/license : List of licensing endpoints

DELETE /2/license/{resource} Refer to the smart licensing documentation

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

GET /2/license/{resource} Refer to the smart licensing documentation

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

DELETE /2/license/{resource}/{agentID} Refer to the smart licensing documentation

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

GET /2/license/{resource}/{agentID} Refer to the smart licensing documentation

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

POST /2/license/{resource}/{agentID} Refer to the smart licensing documentation

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

2/aci_profiles : List of ACI profile endpoints

GET /2/aci_profiles Get all ACI profiles

Response Class (Status 200)

ACI profiles found

Model | Example Value


```
{
  "uuid": "string",
  "name": "string",
  "apic_hosts": "string",
  "apic_username": "string",
  "apic_password": "string",
  "aci_vmm_domain_name": "string",
  "aci_infra_vlan_id": 0,
  "vrf_name": "string",
  "l3_outside_policy_name": "string",
  "l3_outside_network_name": "string",
  "aaep_name": "string",
  "nameservers": [
    "string"
  ],
```

```
  "aci_allocator": {
    "node_vlan_start": 0,
    "node_vlan_end": 0,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
  },
  "control_plane_contract_name": "string",
  "aci_tenant": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
name		search term for profile name	query	string
offset		Page start	query	long
limit		Page size	query	long

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	No ACI profiles found		
default			

Try it out!

POST /2/aci_profiles — Create an ACI profile with the given configuration

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

Parameter content type:
application/json ▼


```
{
  "uuid": "string",
  "name": "string",
  "apic_hosts": "string",
  "apic_username": "string",
  "apic_password": "string",
  "aci_vmm_domain_name": "string",
  "aci_infra_vlan_id": 0,
  "vrf_name": "string",
  "l3_outside_policy_name": "string",
  "l3_outside_network_name": "string",
  "aaep_name": "string",
  "nameservers": [
    "string"
  ],
  "aci_allocator": {
    "node_vlan_start": 0,
    "node_vlan_end": 0,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
  },
  "control_plane_contract_name": "string",
  "aci_tenant": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

HTTP Status Code	Reason	Response Model	Headers
201	Created ACI profile successfully	Model | Example Value	
400	Bad request		
401	Unauthorized		


```
{
  "uuid": "string",
  "name": "string",
  "apic_hosts": "string",
  "apic_username": "string",
  "apic_password": "string",
  "aci_vmm_domain_name": "string",
  "aci_infra_vlan_id": 0,
  "vrf_name": "string",
  "l3_outside_policy_name": "string",
  "l3_outside_network_name": "string",
  "aaep_name": "string",
  "nameservers": [
    "string"
  ],
  "aci_allocator": {
    "node_vlan_start": 0,
    "node_vlan_end": 0,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
  },
  "control_plane_contract_name": "string",
  "aci_tenant": "string"
}
```


Try it out!

DELETE /2/aci_profiles/{aciProfileUUID} Delete an ACI profile

Parameters

Parameter	Value	Description	Parameter Type	Data Type
aciProfileUUID	**(required)**	**ACI profile UUID**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204	Deleted ACI profile successfully		
401	Unauthorized		
404	ACI profile not found		

Try it out!

PATCH /2/aci_profiles/{aciProfileUUID} Update an ACI profile

Response Class (Status 200)

ACI profile updated successfully

Model | Example Value


```
{
  "uuid": "string",
  "name": "string",
  "apic_hosts": "string",
  "apic_username": "string",
  "apic_password": "string",
  "aci_vmm_domain_name": "string",
  "aci_infra_vlan_id": 0,
  "vrf_name": "string",
  "l3_outside_policy_name": "string",
  "l3_outside_network_name": "string",
  "aaep_name": "string",
  "nameservers": [
    "string"
```

```
  ],
  "aci_allocator": {
    "node_vlan_start": 0,
    "node_vlan_end": 0,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
  },
  "control_plane_contract_name": "string",
  "aci_tenant": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

Parameter content type:
application/json ▼


```
{
  "uuid": "string",
  "name": "string",
  "apic_hosts": "string",
  "apic_username": "string",
  "apic_password": "string",
  "aci_vmm_domain_name": "string",
  "aci_infra_vlan_id": 0,
  "vrf_name": "string",
  "l3_outside_policy_name": "string",
  "l3_outside_network_name": "string",
  "aaep_name": "string",
  "nameservers": [
    "string"
  ],
  "aci_allocator": {
    "node_vlan_start": 0,
    "node_vlan_end": 0,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
  },
  "control_plane_contract_name": "string",
  "aci_tenant": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	ACI profile not found		
default			

Try it out!

GET /2/aci_profiles/{aciProfileName} Get an ACI profile by name

Response Class (Status 200)

ACI profile found

Model | Example Value


```
{
  "uuid": "string",
  "name": "string",
  "apic_hosts": "string",
  "apic_username": "string",
  "apic_password": "string",
  "aci_vmm_domain_name": "string",
  "aci_infra_vlan_id": 0,
  "vrf_name": "string",
  "l3_outside_policy_name": "string",
  "l3_outside_network_name": "string",
  "aaep_name": "string",
  "nameservers": [
    "string"
  ],
  "aci_allocator": {
    "node_vlan_start": 0,
    "node_vlan_end": 0,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
  },
  "control_plane_contract_name": "string",
  "aci_tenant": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
aciProfileName	**(required)**	**ACI profile name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	ACI profile not found		
default			

Try it out!

2/keyvalues : List of endpoints for key values

GET /2/keyvalues/{key}

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

POST /2/keyvalues/{key}

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

2/aci_api : accessing ACI api

POST /2/aci_api/login — ACI login

Response Class (Status 200)

OK

Model | Example Value

```
{
  "token": "string"
}
```

Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)**		body	Model | Example Value

```
{
  "apic_ips": "string",
  "apic_username": "string",
  "apic_password": "string"
}
```

Parameter content type: application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

v3 : CCP v3 API (alpha)

GET /v3/{resource} forwards v3 API requests to the v3 API service

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

2/localusers

GET /2/localusers Get CX local users

Parameters

Parameter	Value	Description	Parameter Type	Data Type
name		User Name	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
default			

Try it out!

POST /2/localusers Create CX local user

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
body	(required)		body	Model | Example Value

Parameter content type: application/json ▼


```
{
  "Token": "string",
  "UserName": "string",
  "FirstName": "string",
  "LastName": "string",
  "Password": "string",
  "Disable": true,
  "Role": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201		Model | Example Value	


```
{}
```


Try it out!

DELETE /2/localusers/{username} — Delete a local user

Parameters

Parameter	Value	Description	Parameter Type	Data Type
username	**(required)**	**User Name**	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
204			

Try it out!

PATCH /2/localusers/{username} Update a local user. Can provide either or both parameters.

Response Class (Status 200)

OK

Model | Example Value


```
{
  "FirstName": "string",
  "LastName": "string",
  "Password": "string",
  "Disable": true,
  "Role": "string"
}
```


Response Content Type application/json ▼

Parameters

Parameter	Value	Description	Parameter Type	Data Type
username	**(required)**	**User Name**	path	string
body	**(required)**		body	Model | Example Value

Parameter content type: application/json ▼


```
{
  "FirstName": "string",
  "LastName": "string",
  "Password": "string",
  "Disable": true,
  "Role": "string"
}
```


Response Messages

HTTP Status Code	Reason	Response Model	Headers

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

2/rbac

GET /2/rbac — get the role of the current user

Response Class (Status 200)

OK

Model | Example Value

```
{
  "role": "string"
}
```

Response Content Type application/json ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Not Authorized		
404	Not Found		
default			

Try it out!

[BASE URL: /]

ERROR {...}