Cisco Container Platform 1.1 API Guide

First Published: June 22, 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 1.1 API Guide gives information on Cisco Container Platform APIs and development features.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco Container Platform 1.1 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 the Cisco Container Platform 1.x release, 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

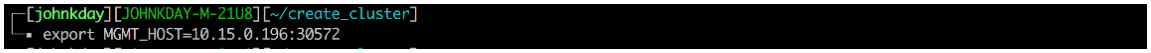
Before you Begin

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

Procedure

1. Export web IP:Port to MGMT_HOST environment variable

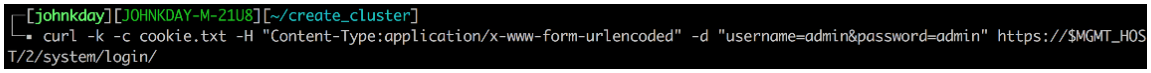
   export MGMT_HOST=<Web UI IP:PORT>

   ```
   ┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
   └▪ export MGMT_HOST=10.15.0.196:30572
   ```

2. Obtain a cookie using default “admin/admin” username and password for Cisco Container Platform

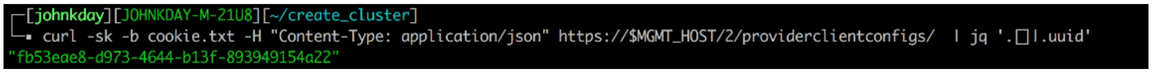
curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOST/2/system/login/



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOS
T/2/system/login/
```

3. Get list of Provider Client Configs

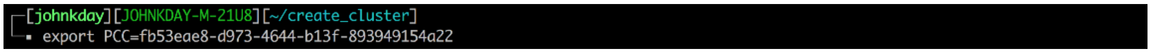
   curl -sk -b cookie.txt -H "Content-Type: application/json" https://$MGMT_HOST/2/providerclientconfigs/ | jq '.[]|.uuid'



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt -H "Content-Type: application/json" https://$MGMT_HOST/2/providerclientconfigs/  | jq '.□|.uuid'
"fb53eae8-d973-4644-b13f-893949154a22"
```

4. export PCC=<selected Provider Client Config>

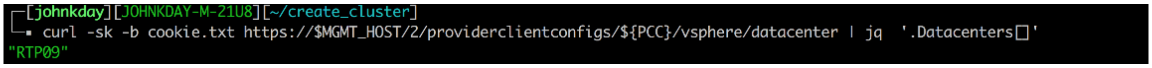
   export PCC=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export PCC=fb53eae8-d973-4644-b13f-893949154a22
```

5. Get the list of Datacenters

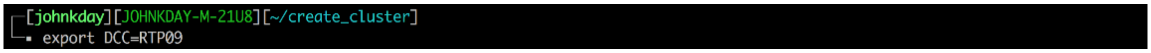
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter | jq '.Datacenters[]'



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter | jq  '.Datacenters□'
"RTP09"
```

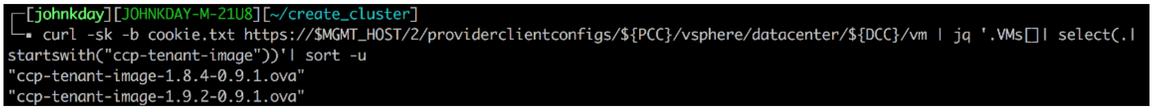
6. export DCC=<from list of DataCenters>

   export DCC=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export DCC=RTP09
```

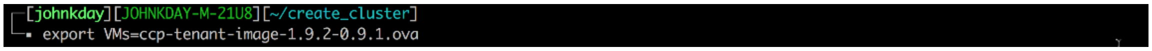
7. Get the list of tenant image VMs
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/vm | jq '.VMs[]| select(.| startswith("ccp-tenant-image"))'| sort -u



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/vm | jq '.VMs□| select(.|
startswith("ccp-tenant-image"))'| sort -u
"ccp-tenant-image-1.8.4-0.9.1.ova"
"ccp-tenant-image-1.9.2-0.9.1.ova"
```

8. export VMs=<from list of VMs>

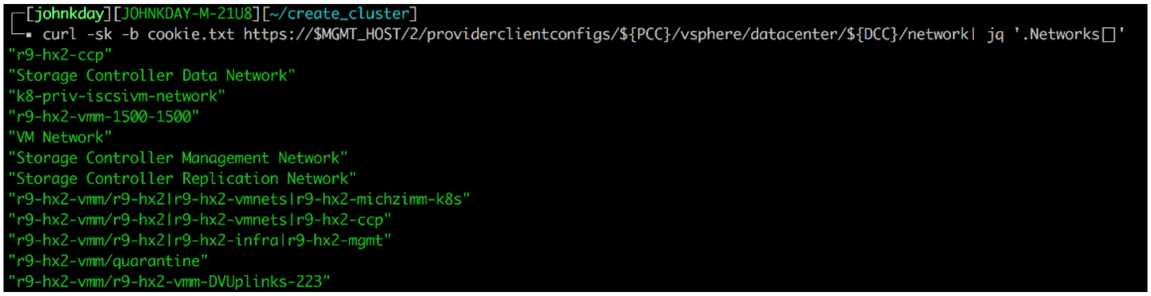
   export VMs=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export VMs=ccp-tenant-image-1.9.2-0.9.1.ova
```

9. Get the list of networks

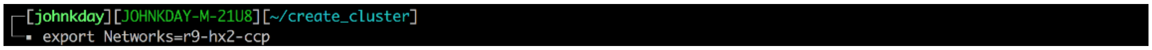
   curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/network| jq '.Networks[]'



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ 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"
"r9-hx2-vmm-1500-1500"
"VM Network"
"Storage Controller Management Network"
"Storage Controller Replication Network"
"r9-hx2-vmm/r9-hx2|r9-hx2-vmnets|r9-hx2-michzimm-k8s"
"r9-hx2-vmm/r9-hx2|r9-hx2-vmnets|r9-hx2-ccp"
"r9-hx2-vmm/r9-hx2|r9-hx2-infra|r9-hx2-mgmt"
"r9-hx2-vmm/quarantine"
"r9-hx2-vmm/r9-hx2-vmm-DVUplinks-223"
```

10. export Networks=<from list of Networks>

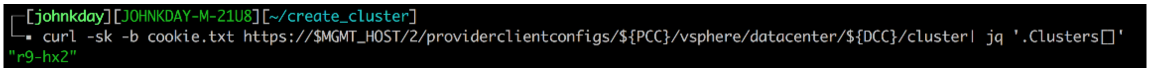
    export Networks=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export Networks=r9-hx2-ccp
```

11. Get the list of clusters

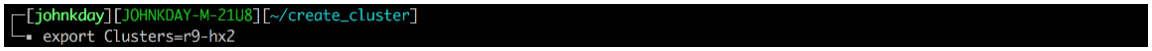
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster| jq '.Clusters[]'



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster| jq '.Clusters[]'
"r9-hx2"
```

12. export Clusters=<from list of clusters>

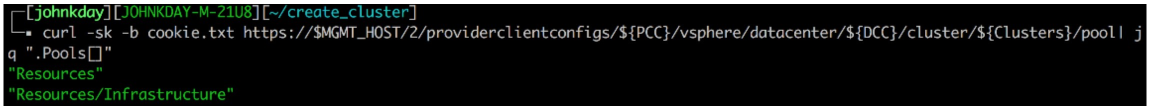
    export Clusters=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export Clusters=r9-hx2
```

13. Get the list of pools

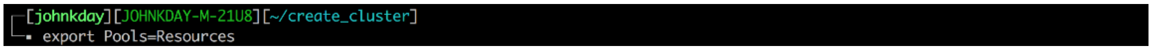
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster/${Clusters}/pool| jq ".Pools[]"



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/cluster/${Clusters}/pool| j
q ".Pools[]"
"Resources"
"Resources/Infrastructure"
```

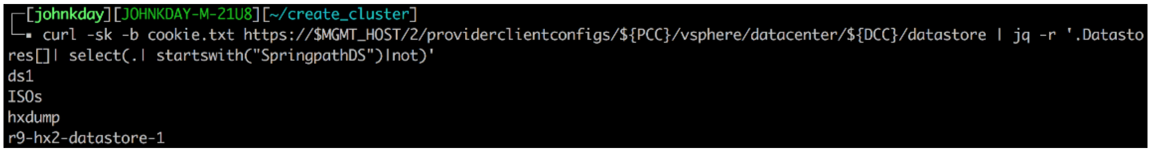
14. export Pools=<from list of Pools>

    export Pools=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export Pools=Resources
```

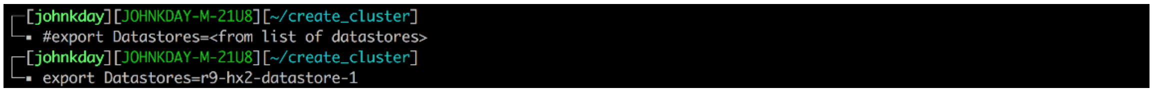
15. Get the list of datastores
    curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/datastore | jq -r '.Datastores[]| select(.| startswith("SpringpathDS")|not)'



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datacenter/${DCC}/datastore | jq -r '.Datasto
res[]| select(.| startswith("SpringpathDS")|not)'
ds1
ISOs
hxdump
r9-hx2-datastore-1
```

16. export Datastores=<from list of datastores>

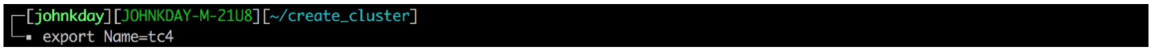
    export Datastores=



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ #export Datastores=<from list of datastores>
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export Datastores=r9-hx2-datastore-1
```

17. export Name=<give name to cluster> (lowercase and numbers, no spaces)

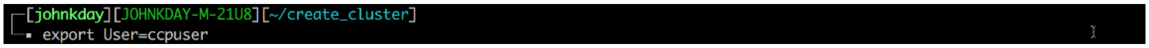
    export Name=tc4



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export Name=tc4
```

18. export User=<username to remotely access cluster nodes with given sshKey>

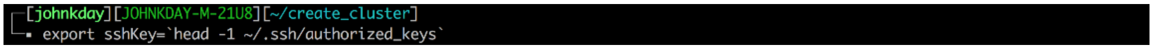
    export User=ccpuser



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export User=ccpuser
```

19. export sshKey=<selected ssh public key for remote access>

    export sshKey=`head -1 ~/.ssh/authorized_keys`

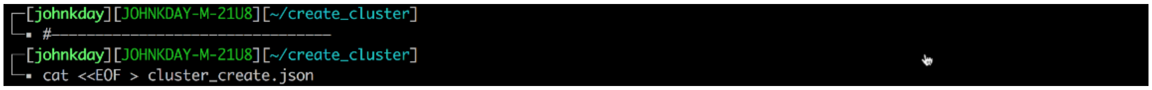


```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export sshKey=`head -1 ~/.ssh/authorized_keys`
```

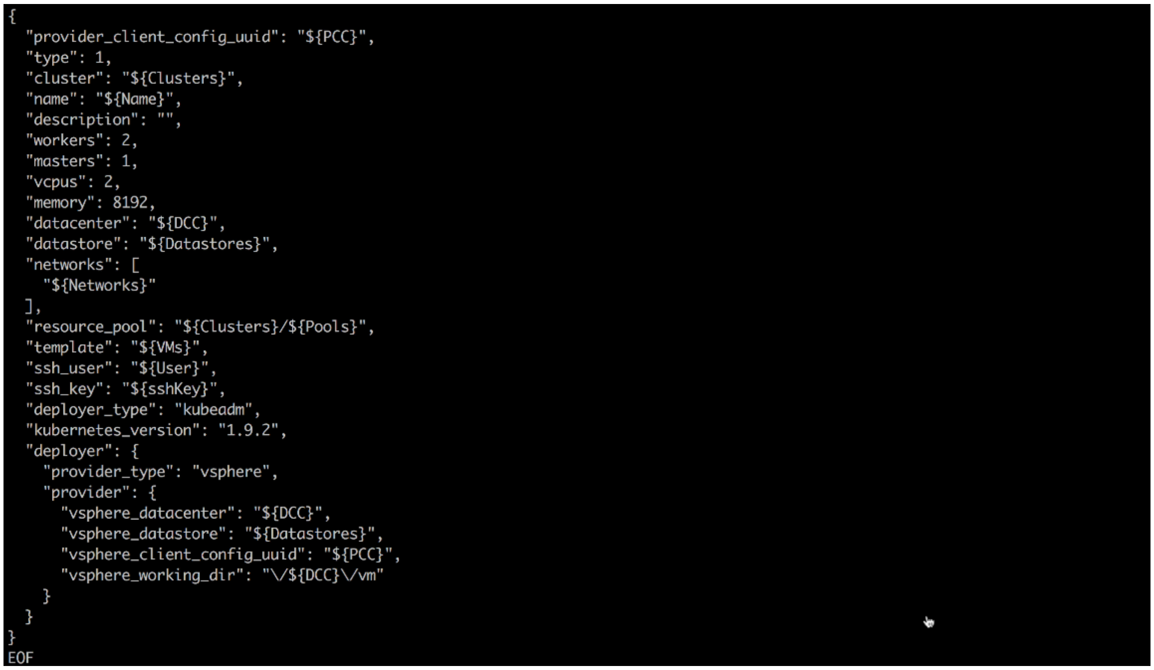
20. copy and paste to create cluster json payload

```
#————————————————————————————————————————————
cat <<EOF > cluster_create.json
{
  "provider_client_config_uuid": "${PCC}",
  "type": 1,
  "cluster": "${Clusters}",
  "name": "${Name}",
  "description": "",
  "workers": 2,
  "masters": 1,
  "vcpus": 2,
  "memory": 8192,
  "datacenter": "${DCC}",
  "datastore": "${Datastores}",
  "networks": [
    "${Networks}"
  ],
  "resource_pool": "${Clusters}/${Pools}",
  "template": "${VMs}",
  "ssh_user": "${User}",
  "ssh_key": "${sshKey}",
  "deployer_type": "kubeadm",
  "kubernetes_version": "1.10.1",
  "deployer": {
    "provider_type": "vsphere",
    "provider": {
      "vsphere_datacenter": "${DCC}",
      "vsphere_datastore": "${Datastores}",
      "vsphere_client_config_uuid": "${PCC}",
    "vsphere_working_dir": "\/${DCC}\/vm"
   }
  }
}
EOF

#———————————————————————————————————————————————
```



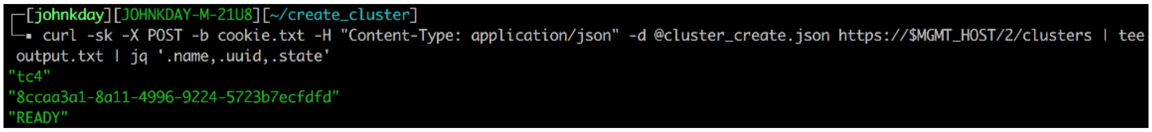
```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ #——————————————————————————————
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ cat <<EOF > cluster_create.json
```



```
{
  "provider_client_config_uuid": "${PCC}",
  "type": 1,
  "cluster": "${Clusters}",
  "name": "${Name}",
  "description": "",
  "workers": 2,
  "masters": 1,
  "vcpus": 2,
  "memory": 8192,
  "datacenter": "${DCC}",
  "datastore": "${Datastores}",
  "networks": [
    "${Networks}"
  ],
  "resource_pool": "${Clusters}/${Pools}",
  "template": "${VMs}",
  "ssh_user": "${User}",
  "ssh_key": "${sshKey}",
  "deployer_type": "kubeadm",
  "kubernetes_version": "1.9.2",
  "deployer": {
    "provider_type": "vsphere",
    "provider": {
      "vsphere_datacenter": "${DCC}",
      "vsphere_datastore": "${Datastores}",
      "vsphere_client_config_uuid": "${PCC}",
      "vsphere_working_dir": "\/${DCC}\/vm"
    }
  }
}
EOF
```

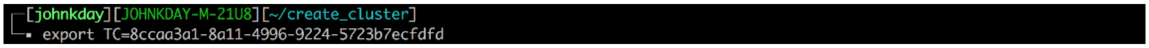
21. Edit cluster_create.json file to adjust number of workers, cpus, memory, kubernetes version or description if needed.

22. Create the Tenant Cluster
    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'



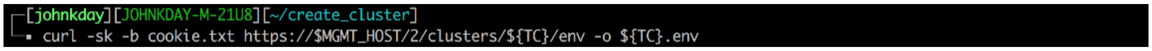
```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -X POST -b cookie.txt -H "Content-Type: application/json" -d @cluster_create.json https://$MGMT_HOST/2/clusters | tee
 output.txt | jq '.name,.uuid,.state'
"tc4"
"8ccaa3a1-8a11-4996-9224-5723b7ecfdfd"
"READY"
```

23. Export Tentant Cluster UUID
    #export TC=<Selected>
    export TC=



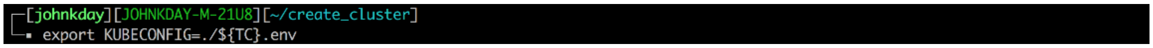
```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

24. Download KUBCONFIG env file
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env



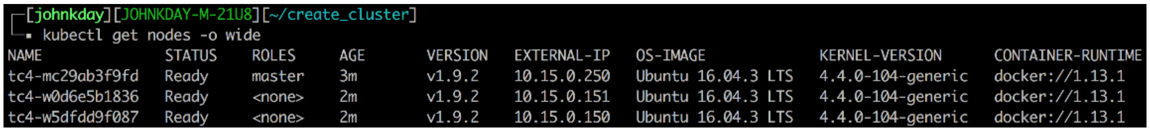
```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env
```

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

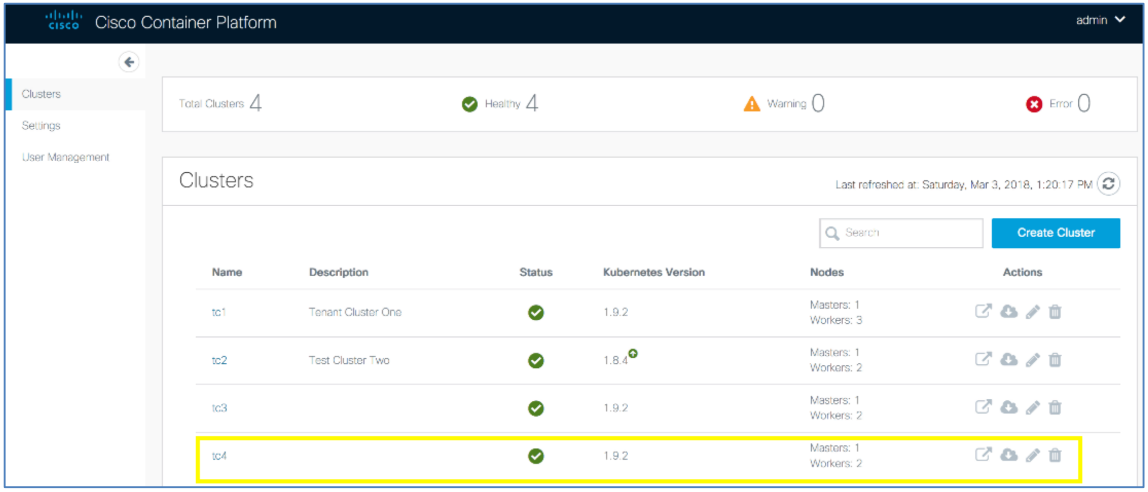


```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ export KUBECONFIG=./${TC}.env
```

26. View nodes on Tenant Cluster
    kubectl get nodes -o wide



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ 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
```

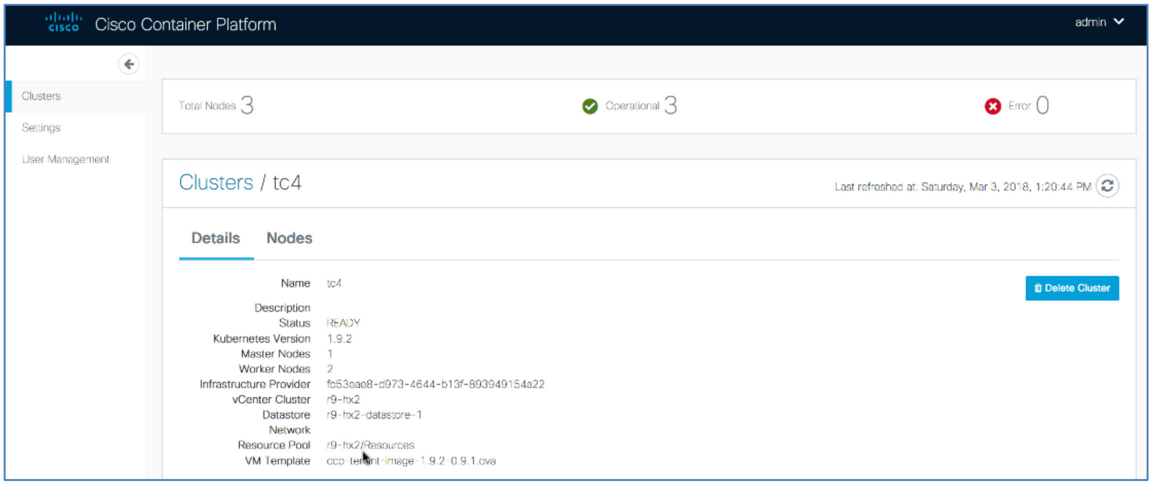


Total Clusters 4 ✓ Healthy 4 ⚠ Warning 0 ✖ Error 0

Clusters

Last refreshed at: Saturday, Mar 3, 2018, 1:20:17 PM

Name	Description	Status	Kubernetes Version	Nodes	Actions
tc1	Tenant Cluster One	✓	1.9.2	Masters: 1 Workers: 3	
tc2	Test Cluster Two	✓	1.8.4	Masters: 1 Workers: 2	
tc3		✓	1.9.2	Masters: 1 Workers: 2	
tc4		✓	1.9.2	Masters: 1 Workers: 2	

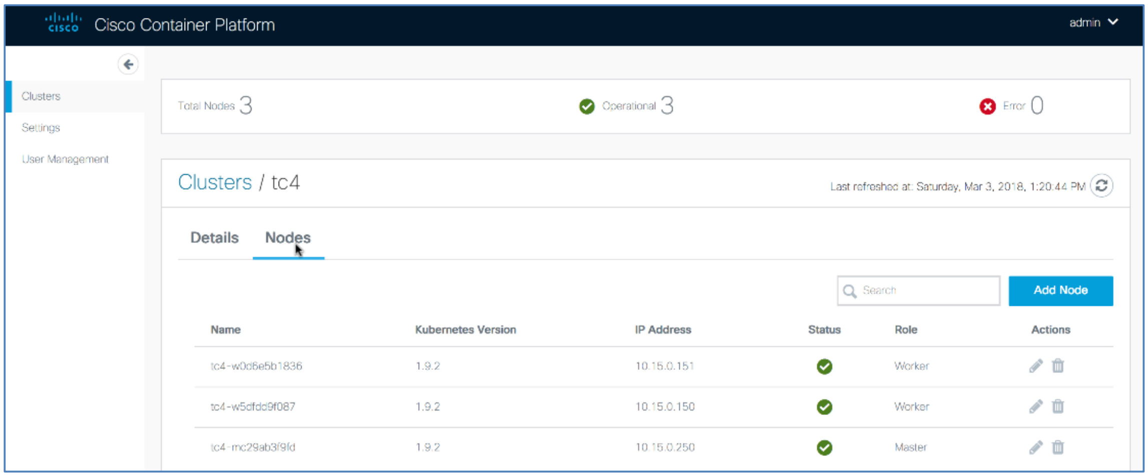


Total Nodes 3 ✓ Operational 3 ✖ Error 0

Clusters / tc4

Last refreshed at: Saturday, Mar 3, 2018, 1:20:44 PM

Name	tc4
Description	
Status	READY
Kubernetes Version	1.9.2
Master Nodes	1
Worker Nodes	2
Infrastructure Provider	fb53eae8-d973-4644-b13f-893949154a22
vCenter Cluster	r9-hx2
Datastore	r9-hx2-datastore-1
Network	
Resource Pool	r9-hx2/Resources
VM Template	ccp-tenant-image-1.9.2-0.9.1.ova

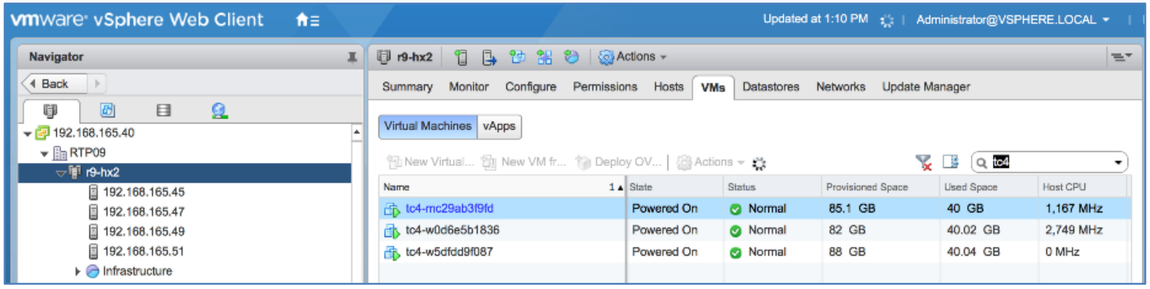


Total Nodes 3 | Operational 3 | Error 0

Clusters / tc4

Last refreshed at: Saturday, Mar 3, 2018, 1:20:44 PM

Name	Kubernetes Version	IP Address	Status	Role	Actions
tc4-w0d6e5b1836	1.9.2	10.15.0.151	✓	Worker	
tc4-w5dfdd9f087	1.9.2	10.15.0.150	✓	Worker	
tc4-mc29ab3f9fd	1.9.2	10.15.0.250	✓	Master	



Name	State	Status	Provisioned Space	Used Space	Host CPU
tc4-mc29ab3f9fd	Powered On	Normal	85.1 GB	40 GB	1,167 MHz
tc4-w0d6e5b1836	Powered On	Normal	82 GB	40.02 GB	2,749 MHz
tc4-w5dfdd9f087	Powered On	Normal	88 GB	40.04 GB	0 MHz

5.2 Deleting a Tenant Cluster

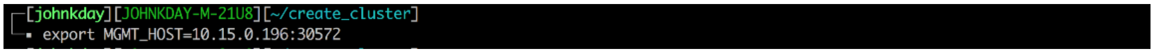
Before you Begin

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

Procedure

1. Export web IP:Port to MGMT_HOST environment variable

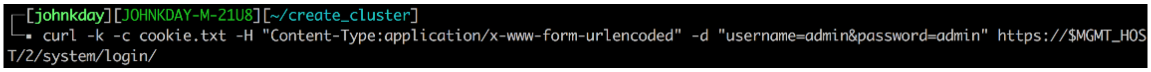
   export MGMT_HOST=<Web UI IP:PORT>



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ export MGMT_HOST=10.15.0.196:30572
```

2. Obtain a cookie using default “admin/admin” username and password for CCP

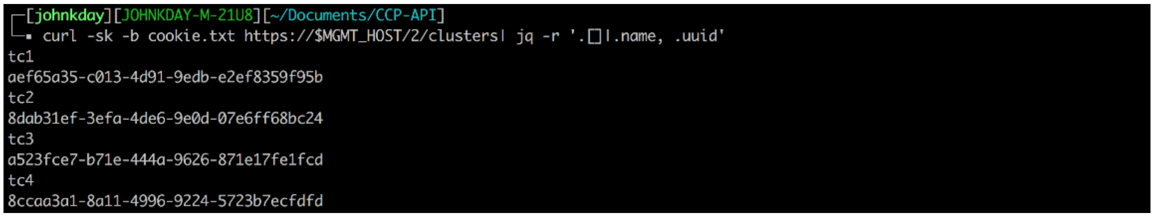
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOST/2/system/login/



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOS
T/2/system/login/
```

3. List Tenant Clusters

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

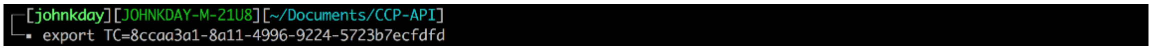


```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'
tc1
aef65a35-c013-4d91-9edb-e2ef8359f95b
tc2
8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
tc3
a523fce7-b71e-444a-9626-871e17fe1fcd
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

'

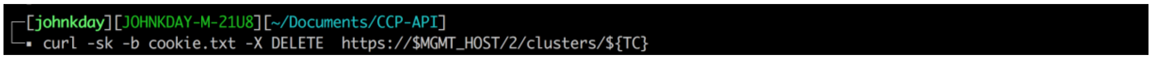
4. Export Tenant Cluster

   export TC=<selected cluster from list>



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

5. Delete the Tenant Cluster
   curl -sk -b cookie.txt -X DELETE https://$MGMT_HOST/2/clusters/${TC}



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt -X DELETE  https://$MGMT_HOST/2/clusters/${TC}
```

5.3 Configuring Windows AD Service Account for Authentication

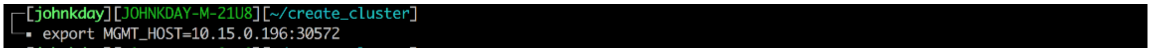
Before you Begin

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

Procedure

1. Export web IP:Port to MGMT_HOST environment variable

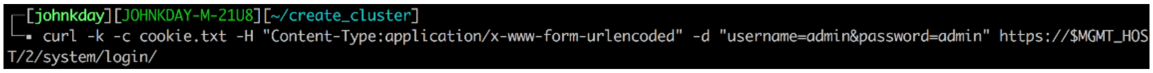
   export MGMT_HOST=<Web UI IP:PORT>



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ export MGMT_HOST=10.15.0.196:30572
```

2. Obtain a cookie using default “admin/admin” username and password for CCP

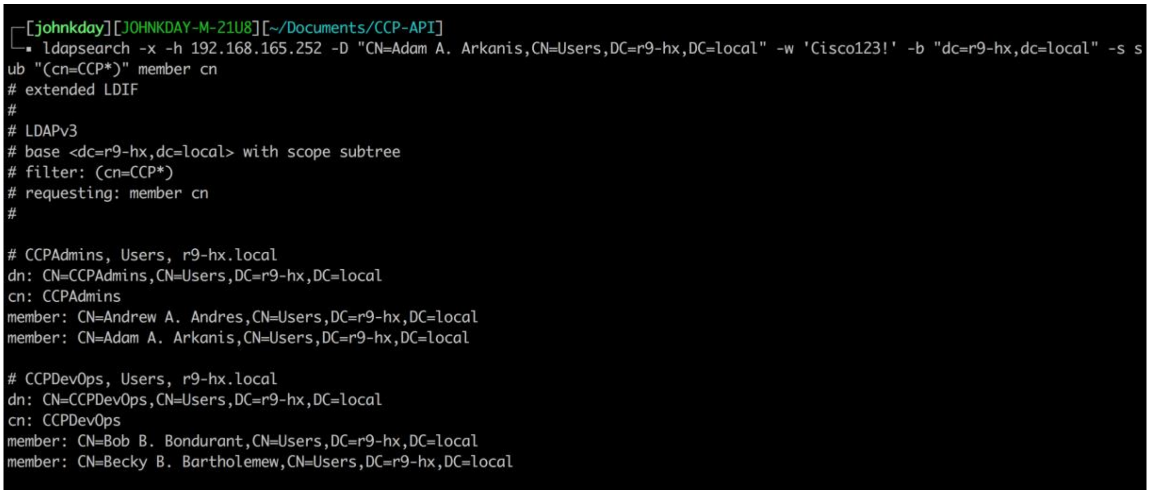
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOST/2/system/login/



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOS
T/2/system/login/
```

3. Query Windows AD server to verify Service Account connection and members of CCP accounts

   ldapsearch -x -h 192.168.165.252 -D "CN=Adam A. Arkanis,CN=Users,DC=r9-hx,DC=local" -w 'Cisco123!' -b "dc=r9-hx,dc=local" -s sub "(cn=CCP*)" member cn



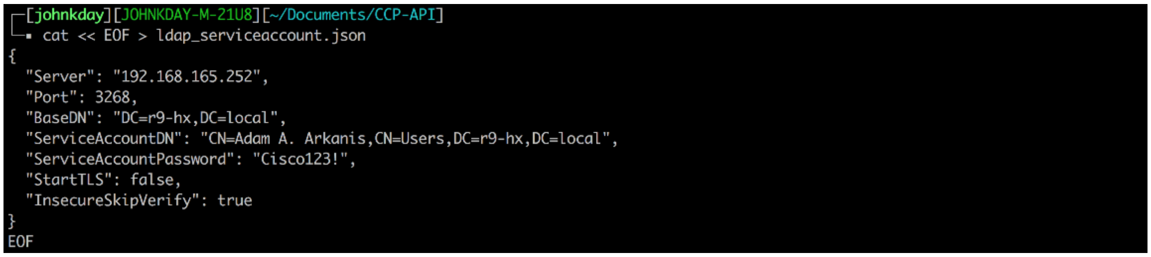
```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ ldapsearch -x -h 192.168.165.252 -D "CN=Adam A. Arkanis,CN=Users,DC=r9-hx,DC=local" -w 'Cisco123!' -b "dc=r9-hx,dc=local" -s s
ub "(cn=CCP*)" member cn
# extended LDIF
#
# LDAPv3
# base <dc=r9-hx,dc=local> with scope subtree
# filter: (cn=CCP*)
# 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 CCP

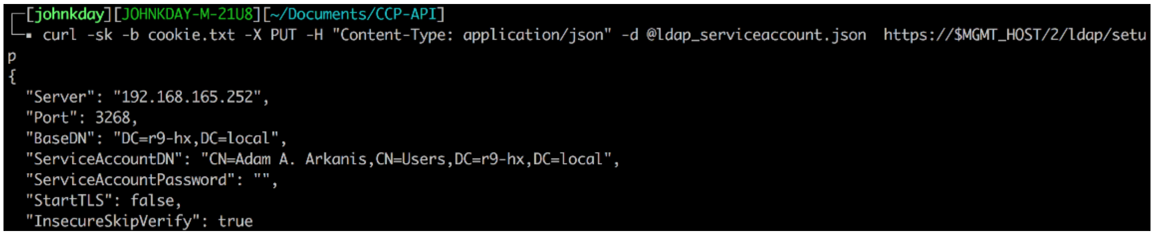
```
cat << EOF > ldap_serviceaccount.json
{
  "Server": "192.168.165.252",
  "Port": 3268,
  "ServiceAccountDN": "CN=Adam A. Arkanis,CN=Users,DC=r9-hx,DC=local",
"BaseDN": "DC=r9-hx,DC=local", "ServiceAccountPassword": "Cisco123!",
  "StartTLS": false,
  "InsecureSkipVerify": true
}
EOF
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ cat << EOF > ldap_serviceaccount.json
{
  "Server": "192.168.165.252",
  "Port": 3268,
  "BaseDN": "DC=r9-hx,DC=local",
  "ServiceAccountDN": "CN=Adam A. Arkanis,CN=Users,DC=r9-hx,DC=local",
  "ServiceAccountPassword": "Cisco123!",
  "StartTLS": false,
  "InsecureSkipVerify": true
}
EOF
```

5. Create the service account for CCP

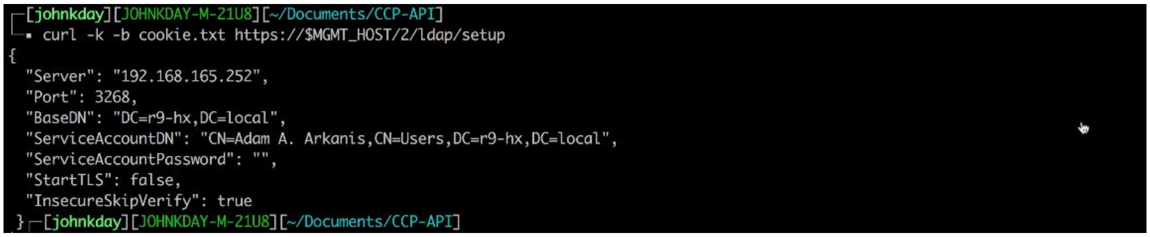
```
curl -sk -b cookie.txt -X PUT -H "Content-Type: application/json" -d
@ldap_serviceaccount.json  https://$MGMT_HOST/2/ldap/setup
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt -X PUT -H "Content-Type: application/json" -d @ldap_serviceaccount.json  https://$MGMT_HOST/2/ldap/setu
p
{
  "Server": "192.168.165.252",
  "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

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



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -k -b cookie.txt https://$MGMT_HOST/2/ldap/setup
{
  "Server": "192.168.165.252",
  "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
}┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
```

5.4 Assigning and Removing Windows AD Group to Tenant Cluster

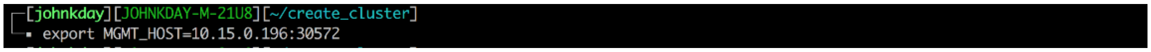
Before you Begin

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

Procedure

1. Export web IP:Port to MGMT_HOST environment variable

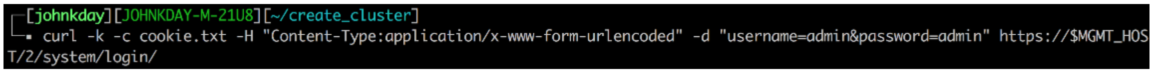
   export MGMT_HOST=<Web UI IP:PORT>



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ export MGMT_HOST=10.15.0.196:30572
```

2. Obtain a cookie using default “admin/admin” username and password for CCP

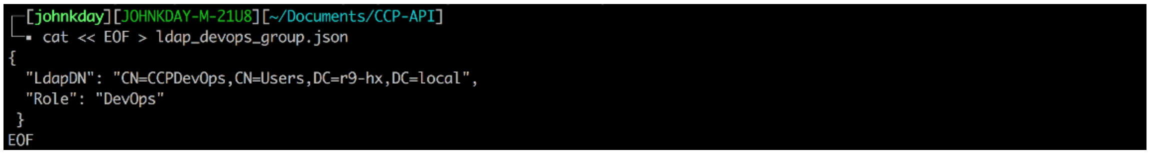
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOST/2/system/login/



```
┌─[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└─▪ curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOS
T/2/system/login/
```

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

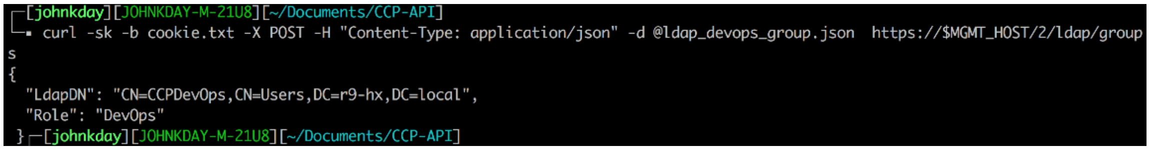
```
cat << EOF > ldap_devops_group.json
{
  "LdapDN": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "Role": "DevOps"
}
EOF
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ cat << EOF > ldap_devops_group.json
{
  "LdapDN": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "Role": "DevOps"
 }
EOF
```

4. Create LDAP Group (You will get an error message if one has already been created and can continue with script)

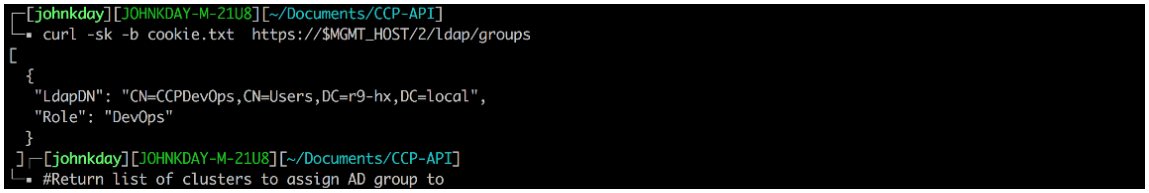
   curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d @ldap_devops_group.json https://$MGMT_HOST/2/ldap/groups



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d @ldap_devops_group.json  https://$MGMT_HOST/2/ldap/group
s
{
  "LdapDN": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "Role": "DevOps"
}┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
```

5. Return list of configured AD groups in CCP

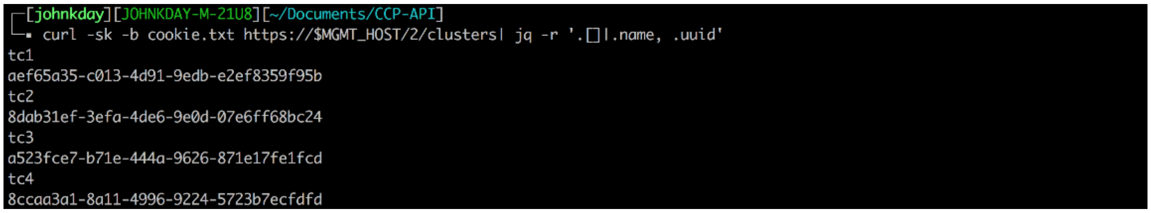
```
curl -sk -b cookie.txt  https://$MGMT_HOST/2/ldap/groups
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt  https://$MGMT_HOST/2/ldap/groups
[
  {
   "LdapDN": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
   "Role": "DevOps"
  }
 ]┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ #Return list of clusters to assign AD group to
```

6. Return list of clusters to assign AD group to

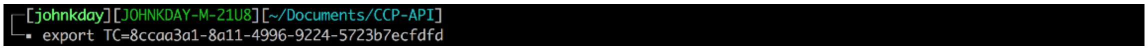
```
curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ 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 selected Tenant Cluster

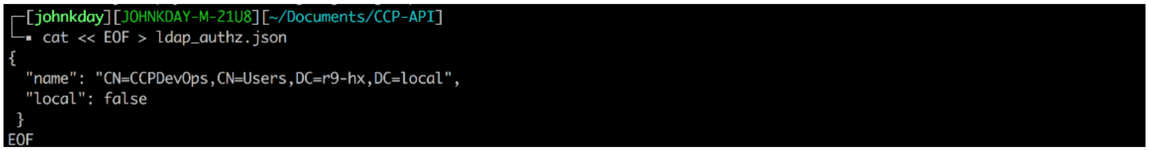
```
export TC=
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

8. Create json payload for assigning AD group to Tenant Cluster

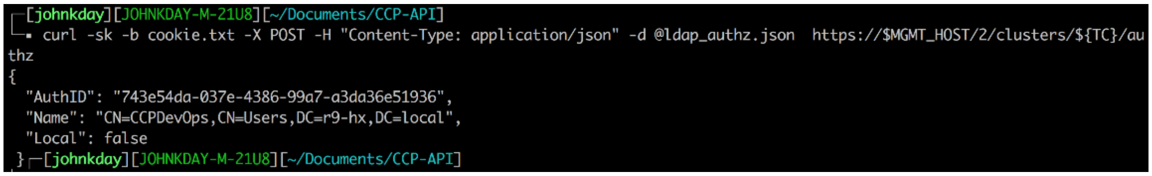
```
cat << EOF > ldap_authz.json
{
  "name": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "local": false
 }
EOF
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ cat << EOF > ldap_authz.json
{
  "name": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "local": false
 }
EOF
```

9. Authorize group access to selected Tenant Cluster

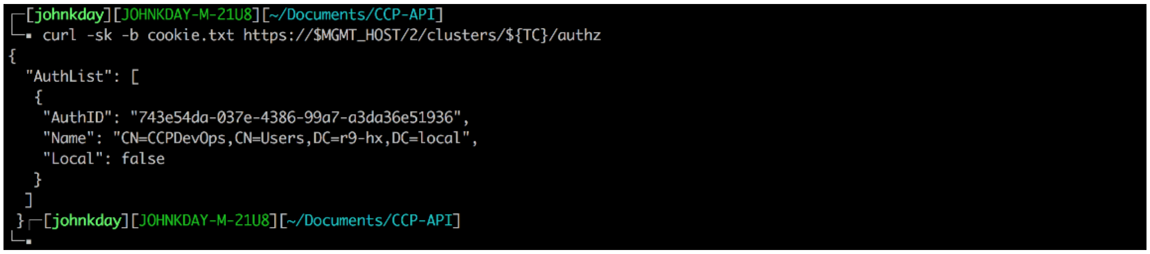
```
curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d
@ldap_authz.json https://$MGMT_HOST/2/clusters/${TC}/authz
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt -X POST -H "Content-Type: application/json" -d @ldap_authz.json  https://$MGMT_HOST/2/clusters/${TC}/au
thz
{
  "AuthID": "743e54da-037e-4386-99a7-a3da36e51936",
  "Name": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "Local": false
 }┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
```

10. Verify authorization of AD group to Tenant Cluster

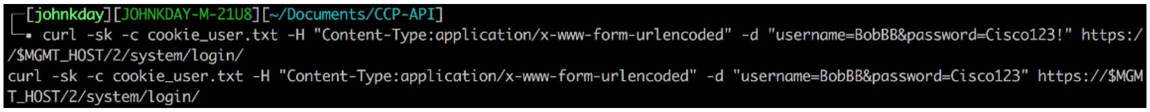
```
curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/authz
```



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ 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
   }
  ]
 }┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪
```

11. Authenticate as user from AD DevOps group

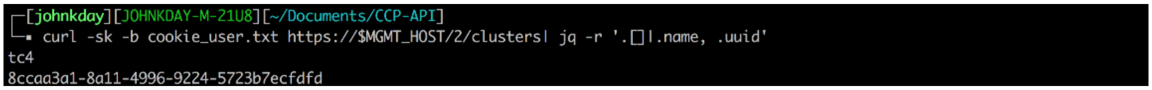
    curl -sk -c cookie_user.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=BobBB&password=Cisco123!" https://$MGMT_HOST/2/system/login/



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -c cookie_user.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=BobBB&password=Cisco123!" https:/
/$MGMT_HOST/2/system/login/
curl -sk -c cookie_user.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=BobBB&password=Cisco123" https://$MGM
T_HOST/2/system/login/
```

12. Verify Tenant Cluster access list for AD user

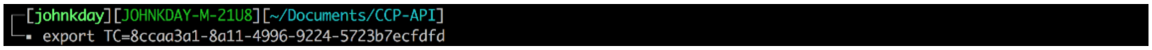
    curl -sk -b cookie_user.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie_user.txt https://$MGMT_HOST/2/clusters| jq -r '.[]|.name, .uuid'
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

13. Export selected Tenant Cluster

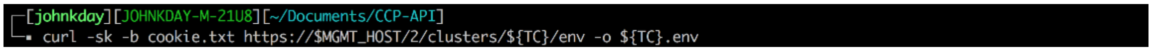
    export TC=



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

14. Download KUBCONFIG env file

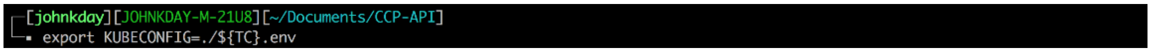
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env
```

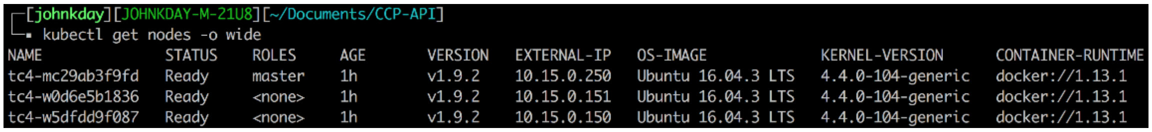
15. Export the config file to KUBECONFIG environment variable

    export KUBECONFIG=./${TC}.env



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ export KUBECONFIG=./${TC}.env
```

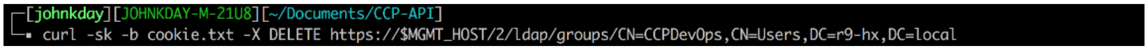
16. View nodes on Tenant Cluster
    kubectl get nodes -o wide



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ 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    10.15.0.250   Ubuntu 16.04.3 LTS   4.4.0-104-generic    docker://1.13.1
tc4-w0d6e5b1836   Ready    <none>   1h       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>   1h       v1.9.2    10.15.0.150   Ubuntu 16.04.3 LTS   4.4.0-104-generic    docker://1.13.1
```

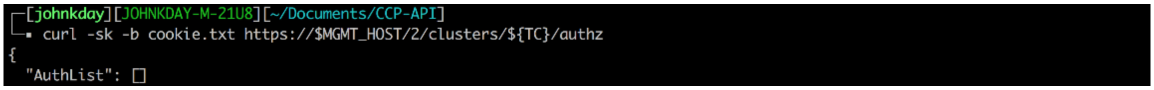
17. Remove AD group access

    #curl -sk -b cookie.txt -X DELETE https://$MGMT_HOST/2/ldap/groups/<DN of Group>
    curl -sk -b cookie.txt -X DELETE
    https://$MGMT_HOST/2/ldap/groups/CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local



```
┌[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└▪ curl -sk -b cookie.txt -X DELETE https://$MGMT_HOST/2/ldap/groups/CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local
```

18. Verify authorization of AD group to Tenant Cluster is removed
    curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/authz



```
┌[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/authz
{
  "AuthList": □
```

5.5 Download Tenant Cluster KUBECONFIG Environment File

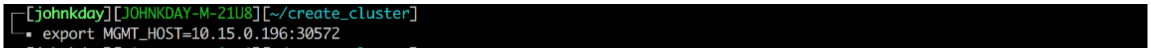
Before you Begin

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

Procedure

1. Export web IP:Port to MGMT_HOST environment variable

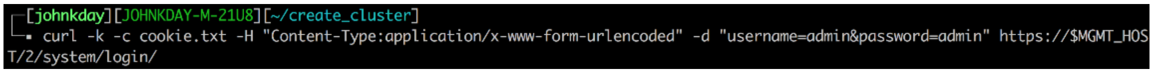
   export MGMT_HOST=<Web UI IP:PORT>



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ export MGMT_HOST=10.15.0.196:30572
```

2. Obtain a cookie using default “admin/admin” username and password for CCP

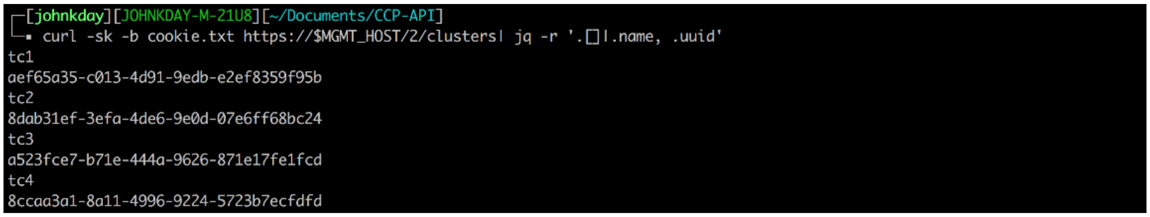
   curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOST/2/system/login/



```
┌[johnkday][JOHNKDAY-M-21U8][~/create_cluster]
└▪ curl -k -c cookie.txt -H "Content-Type:application/x-www-form-urlencoded" -d "username=admin&password=admin" https://$MGMT_HOS
T/2/system/login/
```

3. List Tenant Clusters

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

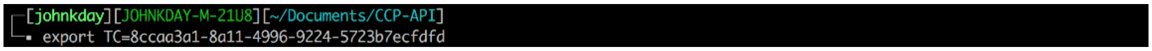


```
┌[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters| jq -r '.□|.name, .uuid'
tc1
aef65a35-c013-4d91-9edb-e2ef8359f95b
tc2
8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
tc3
a523fce7-b71e-444a-9626-871e17fe1fcd
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

'

4. Export Tenant Cluster

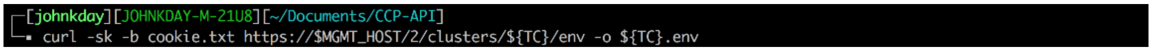
   export TC=<selected cluster from list>



```
┌[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└▪ export TC=8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

5. Download KUBCONFIG env file

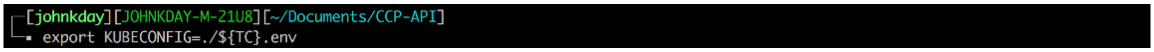
   curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env



```
┌[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└▪ curl -sk -b cookie.txt https://$MGMT_HOST/2/clusters/${TC}/env -o ${TC}.env
```

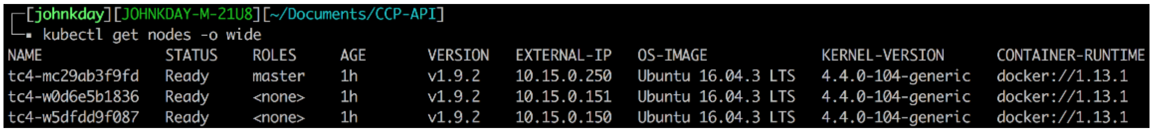
6. Export the config file to KUBECONFIG environment variable

   export KUBECONFIG=./${TC}.env



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ export KUBECONFIG=./${TC}.env
```

7. View nodes on Tenant Cluster
   kubectl get nodes -o wide



```
┌─[johnkday][JOHNKDAY-M-21U8][~/Documents/CCP-API]
└─▪ 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    10.15.0.250   Ubuntu 16.04.3 LTS  4.4.0-104-generic   docker://1.13.1
tc4-w0d6e5b1836   Ready     <none>    1h        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>    1h        v1.9.2    10.15.0.150   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		

HTTP Status Code	Reason	Response Model	Headers
default			

Try it out!

POST /2/providerclientconfigs Add provider client configuration

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	(required) Parameter content type: application/json		body	Model Example Value `{ "uuid": "string", "name": "string", "config": {} }`

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

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201	Added config successfully	Model Example Value `{ "uuid": "string", "name": "string", "config": {} }`	
400	Bad request		
401	Unauthorized		

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

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		

HTTP Status Code	Reason	Response Model	Headers
`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
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",
      "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"
      ]
    }
  ],
  "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

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) 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",
      "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"
      ]
    }
  ],
  "deployer": {
    "provider_type": "string",
    "provider": {
      "vsphere_datacenter": "string",
      "vsphere_datastore": "string",
```

Parameter	Value	Description	Parameter Type	Data Type

```
      "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

Response Messages

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

```
{
  "uuid": "string",
```

HTTP Status Code	Reason	Response Model	Headers

```
  "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",
      "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"
      ]
    }
  ],
  "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"
  },
  "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": [
```

HTTP Status Code	Reason	Response Model	Headers

```
      "string"
    ],
    "registries_insecure": [
      "string"
    ],
    "registries_root_ca": [
      "string"
    ],
    "ingress_vip_pool_id": "string",
    "ingress_vip_addr_id": "string",
    "ingress_vips": [
      "string"
    ],
    "keepalived_vrid": 0,
    "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"
    ]
  }
```

HTTP Status Code	Reason	Response Model	Headers
400	Bad request		
401	Unauthorized		

Try it out!

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

Parameters

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",
      "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"
      ]
    }
  ],
  "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

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",
```

Parameter	Value	Description	Parameter Type	Data Type

```
      "name": "string",
      "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"
      ]
    }
  ],
  "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

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",
      "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"
      ]
    }
  ],
  "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
```

```
  ]
}
```

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",
      "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"
      ]
    }
  ],
  "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"
```

Parameter	Value	Description	Parameter Type	Data Type

```
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
404	Cluster not found		
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",
      "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"
      ]
    }
  ],
  "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

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!

GET /2/clusters/{clusterUUID}/health Get health of cluster

Response Class (Status 200)

Cluster is healthy

Model | Example Value

```
{
  "cluster_health_status": "string"
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
clusterUUID	(required)	**Cluster UUID**	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",
      "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"
      ]
    }
  ],
  "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"
  },
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

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",
      "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"
      ]
    }
  ],
  "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"
  },
  "ccp_private_ssh_key": "string",
  "ccp_public_ssh_key": "string",
  "ntp_pools": [
    "string"
```

Parameter	Value	Description	Parameter Type	Data Type

```
  ],
  "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"
  ],
  "keepalived_vrid": 0,
  "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"
  ]
}
```

Response Messages

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

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

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
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" }`	
401	Unauthorized		
404	HelmChart not found		

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

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) Parameter content type: application/json		body	Model | Example Value `{ "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 `{ "helmchart_uuid": "string", "cluster_UUID": "string", "chart_url": "string", "name": "string", "options": "string" }`	

HTTP Status Code	Reason	Response Model	Headers
400	Bad request		
401	Unauthorized		

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)** Parameter content type: application/json		body	Model | Example Value

```
{
  "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

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)** Parameter content type: application/json		body	Model Example Value { "LdapDN": "string", "Role": "string" }

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
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)** Parameter content type: application/json		body	Model Example Value { "LdapDN": "string", "Role": "string" }

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"
}
```

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)** Parameter content type: application/json		body	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"
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			
201	Created ACI profile 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"
}
```

HTTP Status Code	Reason	Response Model	Headers
400	Bad request		
401	Unauthorized		

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"
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type

Parameter	Value	Description	Parameter Type	Data Type
body	**(required)** Parameter content type: application/json		body	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"
}
```

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"
}
```

Response Content Type application/json

Parameters

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)** Parameter content type: application/json		body	Model | Example Value `{ "apic_ips": "string", "apic_username": "string", "apic_password": "string" }`

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

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default			

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

```
{
  "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) Parameter content type: application/json		body	Model | Example Value { "FirstName": "string", "LastName": "string", "Password": "string", "Disable": true, "Role": "string" }

Response Messages

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 {...}