Cisco Container Platform 4.2.0 API Guide

First Published: August 30, 2019

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 4.2.0 API Guide gives information on Cisco Container Platform APIs and development features.

Cisco Container Platform 4.2.0 API Guide

Contents

1 Overview

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

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

2 Accessing Cisco Container Platform API

You can access the Cisco Container Platform APIs using the following URL:
`https://<CCP IP>/2/swaggerapi`

Where, `<CCP IP>` is the virtual IP address that you provided during the installation of Cisco Container Platform. It is the Ingress Controller LoadBalancer IP address.

3 Key Concepts

3.1 Provider Client Configuration

Cisco Container Platform connects to infrastructure providers such as vCenter to create and manage Virtual Machines that are used for Kubernetes Clusters. The configuration information to connect to the infrastructure provider is represented by a `providerclientconfig` resource.

3.2 Cluster

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

3.3 User Management and Authorization

3.3.1 LDAP and Local Users

Cisco Container Platform supports Active Directory users and local users. Active directory configuration and authorization correspond to the ldap resource type in Cisco Container Platform. Local User management and authorizations correspond to the localusers resource type.

3.4 Subnets and Virtual IP Address Pools

Cisco Container Platform enables you to select an existing network, create a subnet in that network, and then create a Cisco Container Platform Virtual IP Address (VIP) pool within that subnet.
VIP pools are reserved ranges of IP addresses that are assigned as virtual IP addresses within the Cisco Container Platform clusters. Subnets correspond to network_service/subnets resource and VIP pools are a sub-resource of subnets of the type pools.

4 Examples of API Use Cases for vSphere v2 Clusters

4.1 Creating vSphere Tenant Clusters

Before you Begin

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

Procedure

1. Export Cisco Container Platform Virtual IP to the MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Control Plane VIP>

   Example
   ```
   export MGMT_HOST=10.20.30.40
   ```

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

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

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

3. Get list of Provider Client Configurations.

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

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

   Response
   ```
   "fb53eae8-d973-4644-b13f-893949154a22"
   ```

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

   Command
   export PCC=<Selected Provider Client Configuration>

Example

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

5. Get the list of datacenters.

   Command

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

   Example

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

   Response

   ```
   "RTP09"
   ```

6. Configure the datacenter that you want to use.

   Command

   export DCC=<from list of DataCenters>

   Example

   ```
   export DCC=RTP09
   ```

7. Get the list of tenant image VMs.

   Command

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

   Example

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

   Response

   ```
   "ccp-tenant-image-1.13.5-4.1.0.ova"

   "ccp-tenant-image-1.12.7-4.1.0.ova"
   ```

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

   Command

   export VM=<from list of VMs>

   Example

   ```
   export VM=ccp-tenant-image-1.12.3-3.1.0.ova
   ```

9. Get the list of networks.

   Command

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

   Example

   ```
   curl -sk -b cookie.txt
   ```

```
https://$MGMT_HOST/2/providerclientconfigs/${PCC}/vsphere/datac
enter/${DCC}/network| jq '.Networks[]'
```

Response

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

10. Configure the network that you want to use.

    Command
    export NETWORK=<From list of Networks>

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

11. Get the list of clusters.

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

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

    Response

    ```
    "r9-hx2"
    ```

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

    Command
    export CLUSTER=<from list of clusters>

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

13. Get the list of pools.

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

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

    Response

    ```
    "Resources"
    "Resources/Infrastructure"
    ```

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

Command
export POOL=<from list of Pools>

Example
```
export POOL=Resources
```

15. Get the list of datastores.

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

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

    Response

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

16. Configure the datastore that you want to use.

    Command
    export DATASTORE=<from list of datastores>

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

17. Configure a name for the tenant cluster.

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

    Command
    export NAME=<Name of cluster>

    Example
    ```
    export NAME=tc4
    ```

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

    Command
    export USER=<Username>

    Example
    ```
    export USER=ccpuser
    ```

19. Configure the ssh public key for remote access.

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

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

Note: If there is no `public key file, please run ssh-keygen to create a key pair.`

20. Get the list of subnets.

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

    Example
    ```
    curl -sk -b cookie.txt -H "Content-Type: application/json" https://10.20.30.40:32442/2/network_service/subnets/ | jq -r '.[0].uuid'
    ```

    Response

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

21. Configure the subnet for the cluster.

    Command
    export SUBNET=<From the list of subnets>

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

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

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

    Example
    ```
    curl -sk -b cookie.txt -H "Content-Type: application/json" https://10.20.30.40:32442/2/network_service/subnets/${SUBNET}/pools| jq -r '.[0].uuid'
    ```

    Response

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

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

    Command
    export VIP_POOL=<From the list of pools>

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

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

    ```
    #————————————————————————
    cat <<EOF > cluster_create.json
    {
      "provider_client_config_uuid": "${PCC}",
      "type": 1,
      "cluster": "${CLUSTER}",
      "name": "${NAME}",
      "description": "",
      "workers": 2,
      "masters": 1,
      "vcpus": 2,
      "memory": 8192,
      "datacenter": "${DCC}",
    ```

```
  "datastore": "${DATASTORE}",
  "networks": [
    "${NETWORK}"
  ],
   "ingress_vip_pool_id": "${SUBNET}",
   "load_balancer_ip_num": 1,
         "resource_pool": "${CLUSTER}/${POOL}",
      "template": "${VM}",
      "ssh_user": "${USER}",
      "ssh_key": "${SSHKEY}",
      "deployer_type": "kubeadm",
      "kubernetes_version": "1.11.3",
      "deployer": {
      "provider_type": "vsphere",
      "provider": {
        "vsphere_datacenter": "${DCC}",
        "vsphere_datastore": "${DATASTORE}",
        "vsphere_client_config_uuid": "${PCC}",
        "vsphere_working_dir": "\/${DCC}\/vm"
         }
      }
}
EOF

#————————————————————————————
```

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

26. Create a tenant cluster.

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

    Example

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

    Response

    ```
    "tc4"

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

    "READY"
    ```

27. Configure the tenant cluster UUID.

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

    Example

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

28. Download the KUBECONFIG environment file.

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

Example

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

29. Export the config file to KUBECONFIG environment variable.

    Command

    export KUBECONFIG=./${TC}.env

    Example

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

30. View nodes on a tenant cluster.

    Command

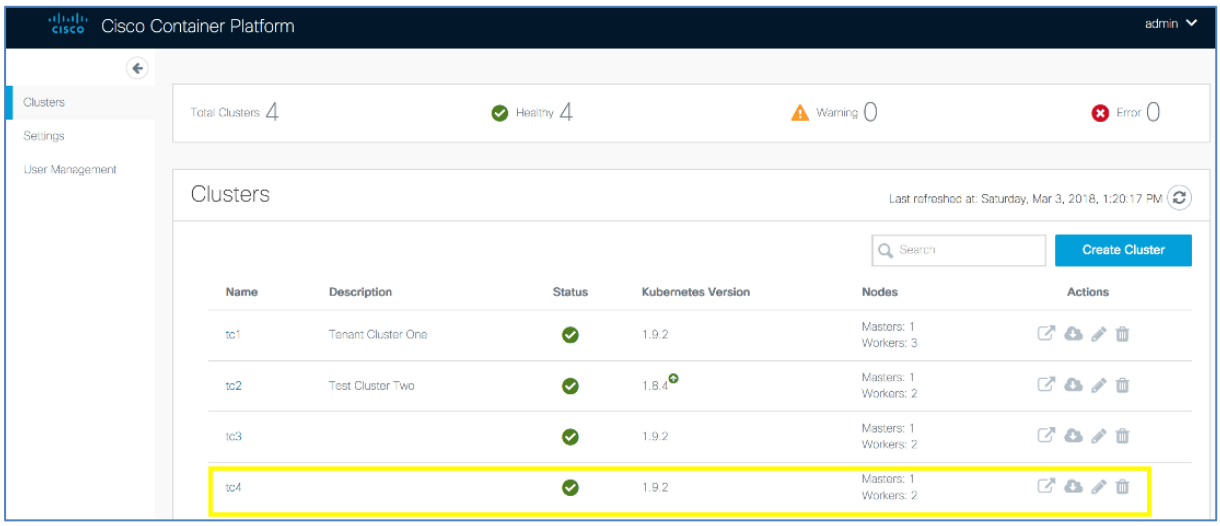
    kubectl get nodes -o wide

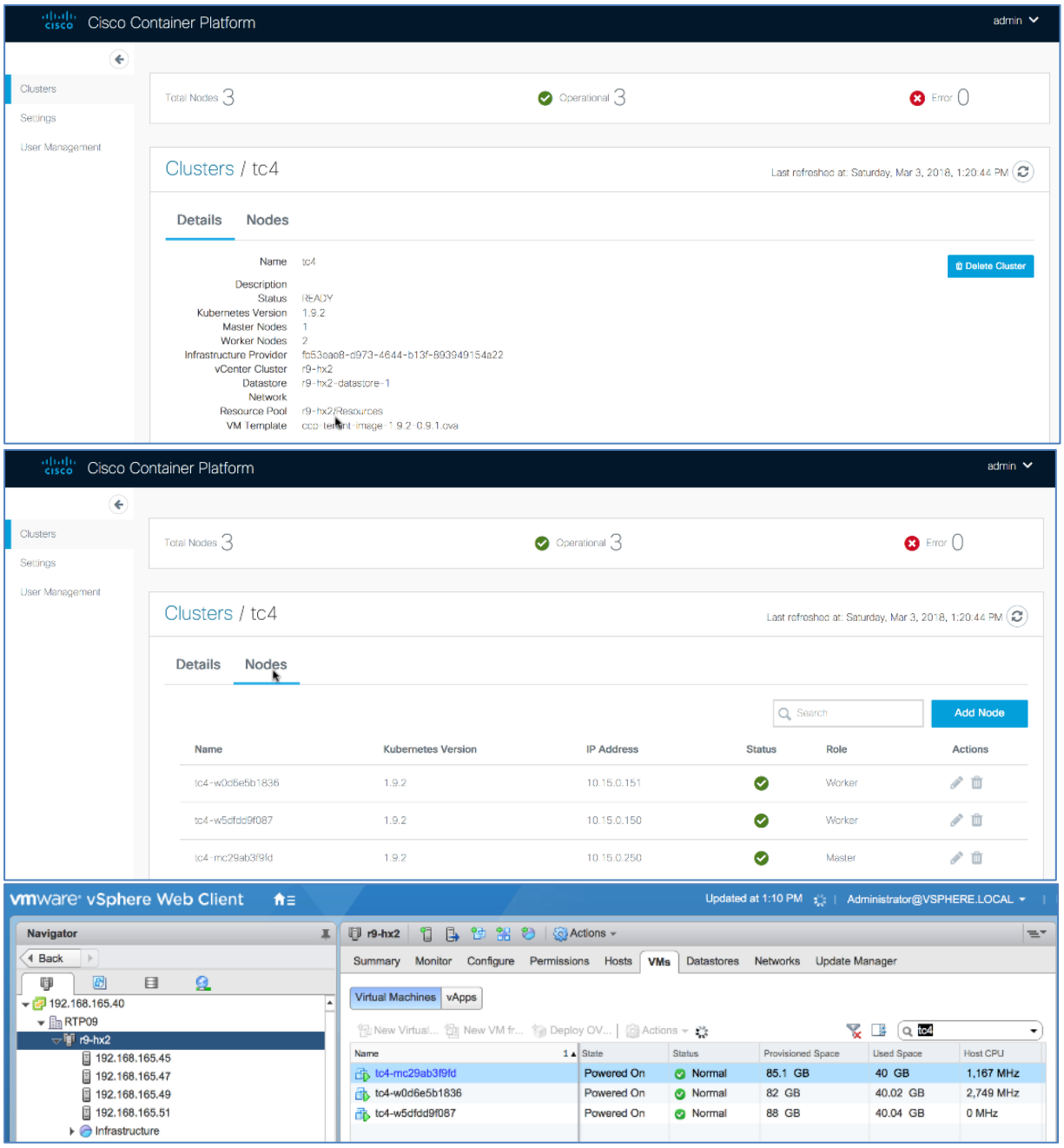
    Example

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

Response

```
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
```





4.2 Deleting vSphere Tenant Clusters

Before you Begin

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

Procedure

1. Export Cisco Container Platform Virtual IP to the MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Control Plane VIP>

   Example
   ```
   export MGMT_HOST=10.20.30.40
   ```

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

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

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

3. List tenant clusters.

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

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

   Response
   ```
   tc1
   aef65a35-c013-4d91-9edb-e2ef8359f95b
   tc2
   8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
   tc3
   a523fce7-b71e-444a-9626-871e17fe1fcd
   tc4
   8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
   ```

4. Export the tenant cluster.

   Command
   export TC=<selected cluster from list>

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

5. Delete the tenant cluster.

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

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

4.3 Configuring Windows AD Service Account for Authentication

Before you Begin

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

Procedure

1. Export Cisco Container Platform Virtual IP to the MGMT_HOST environment variable.

Command
export MGMT_HOST=<Control Plane VIP>

Example
```
export MGMT_HOST=10.20.30.40
```

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

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

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

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

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

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

   Response

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

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

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

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

   Command
   cat << EOF > ldap_serviceaccount.json
   {
    "Server": " <AD Server>",

```
 "Port": 3268,
 "ServiceAccountDN": "<Bind Distinguished Name>",
 "ServiceAccountPassword": "<Password>",
 "StartTLS": false,
 "InsecureSkipVerify": true
}
EOF
```

Example

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

5. Create the service account for Cisco Container Platform.

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

   Example

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

   Response

```
{
    “Server”: ” 192.0.2.1”,
    “Port”: 3268,
    “BaseDN”: “DC=r9-hx,DC=local”,
    “ServiceAccountDN”: “CN=Adam A. Arkanis,CN=Users,DC=r9-
hx,DC=local”,
    “ServiceAccountPassword”: “”,
    “StartTLS”: false,
    “InsecureSkipVerify”: true
}
```

6. Confirm service account configuration.

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

   Example

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

   Response

```
{
     "Server": " 192.0.2.1",
     "Port": 3268,
     "BaseDN": "DC=r9-hx,DC=local",
     "ServiceAccountDN": "CN=Adam A. Arkanis,CN=Users,DC=r9-
hx,DC=local",
     "ServiceAccountPassword":  "",
     "StartTLS":  false,
     "InsecureSkipVerify":  true
}
```

4.4 Managing Windows AD Group Authorizations for Tenant Clusters

Before you Begin

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

Procedure

1. Export Cisco Container Platform Virtual IP to the MGMT_HOST environment variable.

   Command
   export MGMT_HOST=<Control Plane VIP>

   Example
   ```
   export MGMT_HOST=10.20.30.40
   ```

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

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

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

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

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

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

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

   Example
   ```
   curl -sk -b cookie.txt -X POST -H "Content-Type:
   ```

```
application/json" -d @ldap_devops_group.json
https://$MGMT_HOST/2/ldap/groups
```

Response

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

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

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

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

   Response

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

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

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

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

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

   Response

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

7. Export the selected tenant cluster.

   Command
   export TC=<Selected tenant cluster>

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

8. Create a json payload for assigning AD group to a tenant cluster.
   ```
   cat << EOF > ldap_authz.json
   ```

```
{
  "name": "CN=CCPDevOps,CN=Users,DC=r9-hx,DC=local",
  "local": false
 }
EOF
```

9. Authorize group access to the selected tenant cluster.

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

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

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

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

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

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

    Response

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

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

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

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

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

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

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

Response

```
tc4

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

13. Export the selected tenant cluster.

    Command
    export TC=<Selected tenant cluster>

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

14. Download the KUBECONFIG environment file.

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

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

15. Export the config file to KUBECONFIG environment variable.

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

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

16. View nodes on the tenant cluster.

    Command
    kubectl get nodes -o wide

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

    Response

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

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

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

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

17. Remove AD group access.

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

Example

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

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

    Command

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

    Example

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

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

4.5 Downloading Tenant Cluster KUBECONFIG Environment File

Before you Begin

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

Procedure

1. Export Cisco Container Platform Virtual IP to the MGMT_HOST environment variable.

   Command

   export MGMT_HOST=<Control Plane VIP>

   Example

   ```
   export MGMT_HOST=10.20.30.40
   ```

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

   Command

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

   Example

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

3. List tenant clusters.

   Command

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

   Example

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

Response

```
tc1
aef65a35-c013-4d91-9edb-e2ef8359f9gb
tc2
8dab31ef-3efa-4de6-9e0d-07e6ff68bc24
tc3
a523fce7-b71e-444a-9626-871e17fe1fcd
tc4
8ccaa3a1-8a11-4996-9224-5723b7ecfdfd
```

4. Export a tenant cluster.

   Command
   export TC=<selected cluster from list>

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

5. Download the KUBECONFIG environmental file.

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

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

6. Export the config file to KUBECONFIG environment variable.

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

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

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

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

   Response

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

4.6 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`
```

5 Examples of API Use Cases for AWS EKS Clusters

V3 API support providers and clusters for EKS. vSphere and AKS clusters. Sections 5 and 6 give examples for usage with EKS and vSphere respectively using v3 API.

5.1 Logging in to Cisco Container Platform

Command

curl -c cookies.txt -k -X POST -d "username=admin&password=<your_password>" -H "Content-Type:application/x-www-form-urlencoded" "https://<ccp_url>/2/system/login"

Example

1. Log in to Cisco Container Platform.

```
curl -c cookies.txt -k -X POST -d
"username=admin&password=my_password" -H "Content-
Type:application/x-www-form-urlencoded"
"https://10.20.30.40/2/system/login"
```

2. Retrieve the token from the cookies.txt file created as a result of the above command and then store it in an environment variable like this:

```
$ cat cookies.txt
# Netscape HTTP Cookie File
# https://curl.haxx.se/docs/http-cookies.html
# This file was generated by libcurl! Edit at your own
risk.

10.20.30.40   FALSE    /    FALSE    0    CXAccessToken
     eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJBTExfQ0xVU1RFUl
NfQVVUSCI6dHJ1ZSwiZXhwIjoxNTQ4NjM5MDMyLCJyb2xlIjoiQWRtaW5pc
3RyYXRvciJ9.ypjTZFKKmfuBvRxodu-MLedIkQROVNqHdqXgKKdAv7M
```

3. Set your env variable using the token value obtained from Step 2.

```
export
TOKEN=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJBTExfQ0xVU1RF
UlNfQVVUSCI6dHJ1ZSwiZXhwIjoxNTQ4NjM5MDMyLCJyb2xlIjoiQWRtaW5
pc3RyYXRvciJ9.ypjTZFKKmfuBvRxodu-MLedIkQROVNqHdqXgKKdAv7M
```

5.2 Creating Providers for EKS

Command

```
curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
        "type": "eks",
        "name":"name_of_your_eks_cluster",
        "role_arn": "you_aws_role_arn",
        "access_key_id": "your_AWS_access_key_id",
        "secret_access_key": "your_AWS_secret_access_key"
}' https:// <ccp-url>/v3/providers/
```

Example

```
curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
```

```
            "type": "eks",
            "name":"selvi-eks-provider",
            "role_arn":
"arn:aws:iam::123456789123:role/eksServiceRole",
            "access_key_id": "ABCDEFGHIJKLMNOPQRST",
            "secret_access_key":
"THISISNOTAREALSECRETKEYBUTLOOKSLIKEONE"
}' https://10.20.30.40/v3/providers/
```

5.3 Retrieving List of Providers for EKS

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN" https://<ccp-url>/v3/providers

Example

```
curl -k -X GET -H "X-Auth-Token":"$TOKEN"
https://10.20.30.40/v3/providers
```

5.4 Retrieving Specific Provider for EKS

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN" https:// <ccp-url>/v3/providers/<provider_uuid>/

Example

```
curl -k -X GET -H "X-Auth-Token":"$TOKEN" https://
10.20.30.40/v3/providers/17d7d949-cf95-4676-80a7-ae3d773dc3b0/
```

Response

```
[
    {
        "access_key_id": "ABCDEFGHIJKLMNOPQRST",
        "id": "7edd7790-a776-4a91-91f3-0938483dbf78",
        "name": "selvi-eks-provider",
        "role_arn": "arn:aws:iam::12345678912:role/ccp-eks-
7edd7790-a776-4a91-91f3-0938483dbf78",
        "type": "eks"
    }
]
```

5.5 Modifying Providers for EKS

You cannot update the provider details once it is created. This includes parameters such as the Role_ARN, Type, Access_Key_ID, and Secret_Access_Key.

5.6 Deleting Providers for EKS

Command

curl -k -X DELETE -H "x-auth-token: $TOKEN" https:// <ccp-url>/v3/providers/<provider_uuid>/

Example

```
curl -k -X DELETE -H "x-auth-token: $TOKEN"
https://10.20.30.40/v3/providers/7edd7790-a776-4a91-91f3-
0938483dbf78/
```

5.7 Creating EKS clusters

Command

```
curl -k -X POST  -H "x-auth-token: $TOKEN" -d \
'{
   "provider":"provider_uuid",
   "vpc_sizing": {
     "subnet":"<your_desired_subnet>",
     "public_subnets": ["<desired_pub_subnet1>", "<desired_pub_subnet2>", "
<desired_pub_subnet3>"],
     "private_subnets": ["<desired_priv_subnet1>", "<desired_priv_subnet2>",
"<desired_priv_subnet3>"]
},
   "region":"<aws_region_string>",
   "type":"eks",
   "ami": "<ami_id>",
   "instance_type": "<amazon_instance_type>",
   "worker_count": <number_of_workers_in_eks_cluster>,
   "access_role_arn":"<arn_of_role_in_your_aws_account>",
   "name": "<name_of_your_eks_cluster>",
   "ssh_keys": ["<your_ssh_key_to_be_able_to_access_your_workers>",
"<optionally_another_ssh_key>"]
}' https://<ccp_url>/v3/clusters/
```

Example

```
curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
    "provider":"17d7d949-cf95-4676-80a7-ae3d773dc3b0",
    "vpc_sizing": {
      "subnet":"10.20.0.0/16",
      "public_subnets": ["10.20.1.0/24", "10.20.2.0/24",
"10.20.3.0/24"],
      "private_subnets": ["10.20.4.0/24", "10.20.5.0/24",
"10.20.6.0/24"]
    },
    "region":"us-west-2",
    "type":"eks",
    "ami": "ami-09677889326e51ea1",
    "instance_type": "t2.small",
    "worker_count": 1,

"access_role_arn":"arn:aws:iam::123456789123:role/KubernetesAdmin
",
    "name": "selvi_eks_1",
    "ssh_keys": ["ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIHdSrKkWhwED6awk9sjegF0dgcKnotmyrealkey
selvik@SELVIK-M-C1DM", "another_dummy"]
}' https://10.20.30.40/v3/clusters/
```

Response

```
{
   "id":"094c1544-58e5-46cf-8a3f-94de81f35574",
   "type":"eks",
   "name":" selvi_eks_1",
```

```
   "provider":"17d7d949-cf95-4676-80a7-ae3d773dc3b0",
   "region":"us-west-2",
   "status":"CREATING",
   "status_detail":null,
   "access_role_arn":"arn:aws:iam::123456789123:role/KubernetesAd
min",
   "kubeconfig":null,
   "vpc_sizing":{
      "subnet":"10.20.0.0/16",
      "public_subnets":[
         "10.20.1.0/24",
         "10.20.2.0/24",
         "10.20.3.0/24"
      ],
      "private_subnets":[
         "10.20.4.0/24",
         "10.20.5.0/24",
         "10.20.6.0/24"
      ]
   },
   "ami":"ami-09677889326e51ea1",
   "instance_type":"t2.small",
   "ssh_key_name":"",
   "worker_count":1,
   "vpc_id":null
}
```


Note: The API returns the values immediately and the status is indicated as *CREATING*.

5.8 Retrieving all EKS clusters

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN" https://<ccp_url>/v3/clusters

Example

```
curl -k -X GET -H "X-Auth-Token":"$TOKEN"
https://10.10.99.190/v3/clusters
```

Response


```
[
   {
      "id":"094c1544-58e5-46cf-8a3f-94de81f35574",
      "type":"eks",
      "name":"selvi_eks_1",
      "provider":"17d7d949-cf95-4676-80a7-ae3d773dc3b0",
      "region":"us-west-2",
      "status":"CREATING_MASTER",
      "status_detail":"",
      "access_role_arn":"arn:aws:iam::123456789123:role/Kubernete
sAdmin",
      "kubeconfig":null,
      "vpc_sizing":{
         "subnet":"10.20.0.0/16",
         "public_subnets":[
            "10.20.1.0/24",
            "10.20.2.0/24",
            "10.20.3.0/24"
         ],
         "private_subnets":[
            "10.20.4.0/24",
```

```
                "10.20.5.0/24",
                "10.20.6.0/24"
            ]
        },
        "ami":"ami-09677889326e51ea1",
        "instance_type":"t2.small",
        "ssh_key_name":"",
        "worker_count":1,
        "vpc_id":"vpc-thisis72e6cnotreal"
    }
]
```


5.9 Retrieving Specific EKS Clusters

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN"
https://<ccp_url>/v3/clusters/<your_cluster_uuid>/

Example

```
curl -k -X GET -H "X-Auth-Token":"$TOKEN"
https://10.10.99.190/v3/clusters/5a5f0db5-110c-4151-80e8-
9b78889d30bc/
```

Response


```
 [
    {
        "id":"094c1544-58e5-46cf-8a3f-94de81f35574",
        "type":"eks",
        "name":"selvi_eks_1",
        "provider":"17d7d949-cf95-4676-80a7-ae3d773dc3b0",
        "region":"us-west-2",
        "status":"CREATING_MASTER",
        "status_detail":"",
        "access_role_arn":"arn:aws:iam::123456789123:role/Kubernete
sAdmin",
        "kubeconfig":null,
        "vpc_sizing":{
            "subnet":"10.20.0.0/16",
            "public_subnets":[
                "10.20.1.0/24",
                "10.20.2.0/24",
                "10.20.3.0/24"
            ],
            "private_subnets":[
                "10.20.4.0/24",
                "10.20.5.0/24",
                "10.20.6.0/24"
            ]
        },
        "ami":"ami-09677889326e51ea1",
        "instance_type":"t2.small",
        "ssh_key_name":"",
        "worker_count":1,
        "vpc_id":"vpc-thisis72e6cnotreal"
    }
]
```

5.10 Modifying EKS clusters

Command

```
curl -k -X PATCH -H "x-auth-token: $TOKEN" -d \
'{
   "worker_count": 2
}' https://<ccp_url>/v3/clusters/<cluster_uuid>/
```

Example

```
curl -k -X PATCH -H "x-auth-token: $TOKEN" -d \
'{
    "worker_count": 2
}' https://10.20.99.190/v3/clusters/5a5f0db5-110c-4151-80e8-9b78889d30bc/
```

Response


```
[
   {
      "id":"094c1544-58e5-46cf-8a3f-94de81f35574",
      "type":"eks",
      "name":"selvi_eks_1",
      "provider":"17d7d949-cf95-4676-80a7-ae3d773dc3b0",
      "region":"us-west-2",
      "status":"CREATING_MASTER",
      "status_detail":"",
      "access_role_arn":"arn:aws:iam::123456789123:role/KubernetesAdmin",
      "kubeconfig":null,
      "vpc_sizing":{
         "subnet":"10.20.0.0/16",
         "public_subnets":[
            "10.20.1.0/24",
            "10.20.2.0/24",
            "10.20.3.0/24"
         ],
         "private_subnets":[
            "10.20.4.0/24",
            "10.20.5.0/24",
            "10.20.6.0/24"
         ]
      },
      "ami":"ami-09677889326e51ea1",
      "instance_type":"t2.small",
      "ssh_key_name":"",
      "worker_count":1,
      "vpc_id":"vpc-thisis72e6cnotreal"
   }
]
```


5.11 Deleting EKS clusters

Command

```
curl -k -X DELETE -H "x-auth-token: $TOKEN"
https://<ccp_url>/v3/clusters/cluster_uuid/
```

Example

```
curl -k -X DELETE -H "x-auth-token: $TOKEN"
https://10.10.99.190/v3/clusters/5a5f0db5-110c-4151-80e8-
9b78889d30bc/
```

6 Examples of API Use Cases for vSphere v3 Clusters

6.1 Logging in to Cisco Container Platform

Commands

export CCP=https://<Cisco Container Platform URL>
export TOKEN=$(curl -v -k -X POST \
-H "Content-Type:application/x-www-form-urlencoded" \
-d "username=<CCP Username>&password=<CCP Password>" \
$CCP/v3/system/login 2> >(grep x-auth-token) | \
grep x-auth-token | awk -F ":" '{print $2}' | tr -d '\n\r')

Example

Log in to Cisco Container Platform and get the X-Auth-Token

```
export TOKEN=$(curl -v -k -X POST \
    -H "Content-Type:application/x-www-form-urlencoded" \
    -d "username=admin&password=password" \
    $CCP/v3/system/login  2> >(grep x-auth-token) | \
    grep x-auth-token | awk -F ":" '{print $2}' | tr -d '\n\r')
echo $TOKEN
```

6.2 Creating Providers for vSphere v3

Command

curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
"type": "vsphere",
"name":"name_of_vsphere_provider",
"address": "vCenter_url",
"username": "vCenter_username",
"password": "vCenter_password",
"port": "vCenter_port",
"insecure_skip_verify" : true_or_false
}' $CCP/v3/providers/

Example

```
curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
            "type": "vsphere",
            "name":"hx3",
            "address": "vcenter.domain.com",
            "username": "administrator@vsphere.local",
            "password": "password",
            "port": "443",
            "insecure_skip_verify": true
}' $CCP/v3/providers/
```

6.3 Retrieving List of Providers

Command
curl -k -X GET -H "X-Auth-Token: $TOKEN" $CCP/v3/providers/

Example
```
curl -k -X GET -H "x-auth-token: $TOKEN" $CCP/v3/providers/
```

6.4 Retrieving Specific Provider

Command
curl -k -X GET -H "X-Auth-Token:$TOKEN" $CCP/v3/providers/<provider_uuid>/

Example
```
curl -k -X GET -H "X-Auth-Token: $TOKEN"
$CCP/v3/providers/b54efda6-78c7-4418-9b89-955da6585984/
```

Response
```
{
    "id": "b54efda6-78c7-4418-9b89-955da6585984",
    "type": "vsphere",
    "name": "vcenter",
    "address": " vcenter.domain.com",
    "port": 443,
    "username": "administrator@vsphere.local",
    "insecure_skip_verify": true
}
```

6.5 Modifying Providers

Command
```
curl -k -X PATCH -H "x-auth-token: $TOKEN" -d  \
'{
        "type": "vsphere",
        "name":"name_of_vsphere_provider",
        "address": "vCenter_url",
        "username": "vCenter_username",
        "password": "vCenter_password",
        "port": "vCenter_port",
        "insecure_skip_verify" : true_or_false
}' $CCP/v3/providers/your_provider_id/
```

Example
```
curl -k -X PATCH -H "x-auth-token: $TOKEN" -d \
'{
            "type": "vsphere",
            "name":"vcenter-1",
            "address": "vcenter.domain.com",
            "username": "administrator@vsphere.local",
            "password": "password",
            "port": "443",
            "insecure_skip_verify": true
}' $CCP/v3/providers/b54efda6-78c7-4418-9b89-955da6585984/
```

Response
```
{
    "id": "b54efda6-78c7-4418-9b89-955da6585984",
    "type": "vsphere",
    "name": "vcenter-1",
    "address": " vcenter.domain.com",
    "port": 443,
```

```
    "username": "administrator@vsphere.local",
    "insecure_skip_verify": true
}
```

6.6 Deleting Providers

Command

curl -k -X DELETE -H "x-auth-token: $TOKEN" https:// <ccp-url>/v3/providers/<provider_uuid>/

Example

```
curl -k -X DELETE -H "x-auth-token: $TOKEN"
$CCP/v3/providers/7edd7790-a776-4a91-91f3-0938483dbf78/
```

6.7 Creating vSphere V3 clusters

Command

curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
"provider":"provider_uuid",
"vpc_sizing": {
"subnet":"<your_desired_subnet>",
"public_subnets": ["<desired_pub_subnet1>", "<desired_pub_subnet2>", "
<desired_pub_subnet3>"],
"private_subnets": ["<desired_priv_subnet1>", "<desired_priv_subnet2>",
"<desired_priv_subnet3>"]
},
"region":"<aws_region_string>",
"type":"eks",
"ami": "<ami_id>",
"instance_type": "<amazon_instance_type>",
"worker_count": <number_of_workers_in_eks_cluster>,
"access_role_arn":"<arn_of_role_in_your_aws_account>",
"name": "<name_of_your_eks_cluster>",
"ssh_keys": ["<your_ssh_key_to_be_able_to_access_your_workers>",
"<optionally_another_ssh_key>"]
}' https://<ccp_url>/v3/clusters/

Example

```
curl -k -X POST -H "x-auth-token: $TOKEN" -d \
'{
  "provider": "17d7d949-cf95-4676-80a7-ae3d773dc3b0",
  "vpc_sizing": {
    "subnet": "10.20.0.0\/16",
    "public_subnets": [
      "10.20.1.0\/24",
      "10.20.2.0\/24",
      "10.20.3.0\/24"
    ],
    "private_subnets": [
      "10.20.4.0\/24",
      "10.20.5.0\/24",
      "10.20.6.0\/24"
    ]
  },
  "region": "us-west-2",
  "type": "eks",
```

```
  "ami": "ami-09677889326e51ea1",
  "instance_type": "t2.small",
  "worker_count": 1,
  "access_role_arn":
"arn:aws:iam::123456789123:role\/KubernetesAdmin",
  "name": "selvi_eks_1",
  "ssh_keys": [
    "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIHdSrKkWhwED6awk9sjegF0dgcKnotmyrealkey
selvik@SELVIK-M-C1DM",
    "another_dummy"
  ]
}' https://10.20.30.40/v3/clusters/
```

Response


```
{
  "id": "094c1544-58e5-46cf-8a3f-94de81f35574",
  "type": "eks",
  "name": " selvi_eks_1",
  "provider": "17d7d949-cf95-4676-80a7-ae3d773dc3b0",
  "region": "us-west-2",
  "status": "CREATING",
  "status_detail": null,
  "access_role_arn": "arn:aws:iam::123456789123:role\/KubernetesAdmin",
  "kubeconfig": null,
  "vpc_sizing": {
    "subnet": "10.20.0.0\/16",
    "public_subnets": [
      "10.20.1.0\/24",
      "10.20.2.0\/24",
      "10.20.3.0\/24"
    ],
    "private_subnets": [
      "10.20.4.0\/24",
      "10.20.5.0\/24",
      "10.20.6.0\/24"
    ]
  },
  "ami": "ami-09677889326e51ea1",
  "instance_type": "t2.small",
  "ssh_key_name": "",
  "worker_count": 1,
  "vpc_id": null
}
```


Note: The API returns the values immediately, and the status is indicated as *CREATING*.

6.8 Retrieving all clusters

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN" $CCP/v3/clusters

Example

```
curl -k -X GET -H "X-Auth-Token":"$TOKEN" $CCP/v3/clusters
```

Response


```
 [
```

```
    {
        "id": "35de61b9-5175-40d5-bea3-1b058fb22c45",
        "type": "vsphere",
        "name": "demo-cluster",
        "provider": "b54efda6-78c7-4418-9b89-955da6585984",
        "status": "READY",
        "spec": {
            "name": "demo-cluster",
            "type": "vsphere",
            "kubernetes_version": "1.13.5",
            "ip_allocation_method": "ccpnet",
            "master_vip": "",
            "load_balancer_num": 1,
            "subnet_id": "ea042d99-9c69-43f8-ac44-ab0b9c843dcf",
            "ntp_pools": [],
            "ntp_servers": [],
            "root_ca_registries": [],
            "self_signed_registries": {},
            "vsphere_infra": {
                "cluster": "HX3",
                "datacenter": "HX3",
                "datastore": "hx3-data",
                "folder": "",
                "guestOS": "",
                "hostSystem": "",
                "networks": [
                    "VLAN 1161 - 10.10.100.0 - 22"
                ],
                "resource_pool": ""
            },
            "master_group": {
                "gpus": [],
                "labels": null,
                "name": "master-group",
                "size": 1,
                "taints": null,
                "template": "ccp-tenant-image-1.13.5-ubuntu18-
4.0.1.ova",
                "vcpus": 2,
                "memory_mb": 16384,
                "ssh_key": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINhzxv/Zy/uHF567CqR1o71Z7Wo4Wk/3+H5APXv1c
RM6",
                "ssh_user": "ccpuser",
                "nodes": [
                    {
                        "name": "demo-cluster-0-master-0",
                        "status": "ERROR",
                        "phase": "Running",
                        "private_ip": "10.10.100.109",
                        "public_ip": "10.10.100.109"
                    }
                ]
            },
            "node_groups": [
                {
                    "gpus": [],
                    "labels": null,
                    "name": "node-group",
                    "size": 1,
                    "taints": null,
```

```
                    "template": "ccp-tenant-image-1.13.5-
ubuntu18-4.0.1.ova",
                    "vcpus": 2,
                    "memory_mb": 16384,
                    "ssh_key": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINhzxv/Zy/uHF567CqR1o71Z7Wo4Wk/3+",
                    "ssh_user": "ccpuser",
                    "nodes": [
                        {
                            "name": "demo-cluster-1-node-gr-0",
                            "status": "READY",
                            "phase": "Running",
                            "private_ip": "10.10.100.108",
                            "public_ip": "10.10.100.108"
                        }
                    ]
                }
            ],
            "network_plugin_profile": {
                "details": {
                    "typhaReplicas": "1",
                    "pod_cidr": "192.168.0.0/16",
                    "ssh_user": "ccpuser"
                },
                "name": "calico"
            },
            "kubernetes_config_secret": "demo-cluster-
kubeconfig",
            "ingress_as_lb": true,
            "nginx_ingress_class": "",
            "etcd_encrypted": false,
            "skip_management": null,
            "docker_no_proxy": []
        },
        "kubeconfig": "…",
        "kubernetes_version": "1.13.5",
        "kubernetes_config_secret": null,
        "ip_allocation_method": "ccpnet",
        "master_vip": "",
        "load_balancer_num": 1,
        "subnet_id": "ea042d99-9c69-43f8-ac44-ab0b9c843dcf",
        "ntp_pools": [],
        "ntp_servers": [],
        "root_ca_registries": [],
        "self_signed_registries": {},
        "insecure_registries": [],
        "docker_http_proxy": "",
        "docker_https_proxy": "",
        "vsphere_infra": {
            "datacenter": "HX3",
            "datastore": "hx3-data",
            "networks": [
                "VLAN 1161 - 10.10.100.0 - 22"
            ],
            "cluster": "HX3",
            "resource_pool": "",
            "folder": ""
        },
        "master_group": {
            "name": "master-group",
            "size": 1,
```

```
            "template": "ccp-tenant-image-1.13.5-ubuntu18-
4.0.1.ova",
            "vcpus": 2,
            "memory_mb": 16384,
            "gpus": [],
            "ssh_user": "ccpuser",
            "ssh_key": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINhzxv/Zy/uHF567CqR1o71Z7Wo4Wk/3+H5APXvlc
RM6
            "nodes": [
                {
                    "name": "demo-cluster-0-master-0",
                    "status": "ERROR",
                    "phase": "Running",
                    "private_ip": "10.10.100.109",
                    "public_ip": "10.10.100.109"
                }
            ]
        },
        "node_groups": [
            {
                "name": "node-group",
                "size": 1,
                "template": "ccp-tenant-image-1.13.5-ubuntu18-
4.0.1.ova",
                "vcpus": 2,
                "memory_mb": 16384,
                "gpus": [],
                "ssh_user": "ccpuser",
                "ssh_key": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINhzxv/Zy/uHF567CqR1o71Z7Wo4Wk/3+H5APXvlc
RM6",
                "nodes": [
                    {
                        "name": "demo-cluster-1-node-gr-0",
                        "status": "READY",
                        "phase": "Running",
                        "private_ip": "10.10.100.108",
                        "public_ip": "10.10.100.108"
                    }
                ]
            }
        ],
        "network_plugin_profile": {
            "details": {
                "typhaReplicas": "1",
                "pod_cidr": "192.168.0.0/16",
                "ssh_user": "ccpuser"
            },
            "name": "calico"
        },
        "ingress_as_lb": true,
        "nginx_ingress_class": "",
        "etcd_encrypted": false,
        "skip_management": false,
        "docker_no_proxy": [],
        "routable_cidr": null,
        "image_prefix": null,
        "aci_profile": null
    }
]
```

6.9 Retrieving Specific Clusters

Command
curl -k -X GET -H "X-Auth-Token":"$TOKEN" $CCP/v3/clusters/<your_cluster_uuid>/

Example

```
curl -k -X GET -H "X-Auth-Token":"$TOKEN"
$CCP/v3/clusters/35de61b9-5175-40d5-bea3-1b058fb22c45/
```

Response


```
 {
    "id": "35de61b9-5175-40d5-bea3-1b058fb22c45",
    "type": "vsphere",
    "name": "demo-cluster",
    "provider": "b54efda6-78c7-4418-9b89-955da6585984",
    "status": "READY",
    "spec": {
        "name": "demo-cluster",
        "type": "vsphere",
        "kubernetes_version": "1.13.5",
        "ip_allocation_method": "ccpnet",
        "master_vip": "",
        "load_balancer_num": 1,
        "subnet_id": "ea042d99-9c69-43f8-ac44-ab0b9c843dcf",
        "ntp_pools": [],
        "ntp_servers": [],
        "root_ca_registries": [],
        "self_signed_registries": {},
        "vsphere_infra": {
            "cluster": "HX3",
            "datacenter": "HX3",
            "datastore": "hx3-data",
            "folder": "",
            "guestOS": "",
            "hostSystem": "",
            "networks": [
                "VLAN 1161 - 10.10.100.0 - 22"
            ],
            "resource_pool": ""
        },
        "master_group": {
            "gpus": [],
            "labels": null,
            "name": "master-group",
            "size": 1,
            "taints": null,
            "template": "ccp-tenant-image-1.13.5-ubuntu18-
4.0.1.ova",
        …
    "kubernetes_version": "1.13.5",
    "kubernetes_config_secret": null,
    "ip_allocation_method": "ccpnet",
    "master_vip": "",
    "load_balancer_num": 1,
    "subnet_id": "ea042d99-9c69-43f8-ac44-ab0b9c843dcf",
    "ntp_pools": [],
    "ntp_servers": [],
    "root_ca_registries": [],
```

```
    "self_signed_registries": {},
    "insecure_registries": [],
    "docker_http_proxy": "",
    "docker_https_proxy": "",
    "vsphere_infra": {
        "datacenter": "HX3",
        "datastore": "hx3-data",
        "networks": [
            "VLAN 1161 - 10.10.100.0 - 22"
        ],
        "cluster": "HX3",
        "resource_pool": "",
        "folder": ""
    },
    "master_group": {
        "name": "master-group",
        "size": 1,
        "template": "ccp-tenant-image-1.13.5-ubuntu18-4.0.1.ova",
        "vcpus": 2,
        "memory_mb": 16384,
        "gpus": [],
        "ssh_user": "ccpuser",
        "ssh_key": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINhzxv/Zy/uHF567CqR1o71Z7Wo4Wk/3+H5APXvlc
RM6",
        "nodes": [
            {
                "name": "demo-cluster-0-master-0",
                "status": "ERROR",
                "phase": "Running",
                "private_ip": "10.10.100.109",
                "public_ip": "10.10.100.109"
            }
        ]
    },
    "node_groups": [
        {
            "name": "node-group",
            "size": 1,
            "template": "ccp-tenant-image-1.13.5-ubuntu18-
4.0.1.ova",
            "vcpus": 2,
            "memory_mb": 16384,
            "gpus": [],
            "ssh_user": "ccpuser",
            "ssh_key": "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAINhzxv/Zy/uHF567CqR1o71Z7Wo4Wk/3+H5APXvlc
RM6",
            "nodes": [
                {
                    "name": "demo-cluster-1-node-gr-0",
                    "status": "READY",
                    "phase": "Running",
                    "private_ip": "10.10.100.108",
                    "public_ip": "10.10.100.108"
                }
            ]
        }
    ],
    "network_plugin_profile": {
        "details": {
```

```
                "typhaReplicas": "1",
                "pod_cidr": "192.168.0.0/16",
                "ssh_user": "ccpuser"
            },
            "name": "calico"
        },
        "ingress_as_lb": true,
        "nginx_ingress_class": "",
        "etcd_encrypted": false,
        "skip_management": false,
        "docker_no_proxy": [],
        "routable_cidr": null,
        "image_prefix": null,
        "aci_profile": null
    }
```

6.10 Deleting clusters

Command
curl -k -X DELETE -H "x-auth-token: $TOKEN" $CCP/v3/clusters/cluster_uuid/

Example

```
curl -k -X DELETE -H "x-auth-token: $TOKEN" $CCP/v3/clusters/
35de61b9-5175-40d5-bea3-1b058fb22c45/
```

6.11 Creating ACI Profile

```
curl -XPOST -d '{
  "name": "example-aci-profile5",
  "apic_username": "username",
  "apic_password": "password",
  "aci_tenant": "aci_tenant",
  "apic_hosts": "apic_hosts",
  "aci_vmm_domain_name": "aci_vmm_domain_name",
  "vrf_name": "vrf_name",
  "l3_outside_policy_name": "l3_outside_policy_name",
  "l3_outside_network_name": "l3_outside_network_name",
  "aaep_name": "aaep_name",
  "nameservers": "nameservers",
  "aci_infra_vlan_id": 1234,
  "node_vlan_start": 1,
  "node_vlan_end": 100,
  "multicast_range": "10.0.0.0\/16",
  "service_subnet_start": "20.15.1.1\/16",
  "pod_subnet_start": "10.2.0.0\/16",
  "aci_profile_name": "asdf"
}' -H 'content-type: application/json' localhost:8000/v3/aci-
profiles/
```

6.12 Creating ACI-enabled vSphere Cluster

```
curl -d '{"type":"vsphere", "provider": "276ed502-1b95-4329-859e-
12289d37953b", "name":"example-vsphere-cluster",
"kubernetes_version":"1.12.7", "vsphere_infra":{"folder":"yeet",
"datacenter":"foo",
"datastore":"foo", "networks":["foo"], "cluster":"foo",
"resource_pool":"ayyy"}, "master_group":{"name":"foo","size":1234},
```

```
"network_plugin_profile":{"details":{"pod_cidr":"10.0.0.0/24"}},
"node_groups":[], "ip_allocation_method":"ccpnet"
, "master_vip":"1.2.3.4","skip_management":true,
"docker_no_proxy":["foo","bar"],
"load_balancer_num":3,"subnet_id":"5c2f63d5-5821-439f-acd5-
fb8ddd559cac","aci_profile":"aadb0435-775d-445d-9bac-37dfcad1eb89",
"routable_cidr":"10.10.123.1/
24", "image_prefix":"this is not validated yet"}'
localhost:8000/v3/clusters/
```

6.13 Updating ACI Profile

Command

curl -XPATCH -d '{"aaep_name":"new_aaep_name"}' localhost:8000/v3/aci-profiles/aadb0435-775d-445d-9bac-37dfcad1eb89/

Note: The cluster has to be PATCHed to pick up the new ACI details (this is by design).

Example

```
curl -s -XPATCH -d '{}' localhost:8000/v3/clusters/d7dc05c7-78a6-
4ff7-9657-1ac48ee09dcb/
```

6.14 Deleting ACI Profile

Example

```
curl -XDELETE localhost:8000/v3/aci-profiles/aadb0435-775d-445d-
9bac-37dfcad1eb89/
```

6.15 Listing Addons

Helm charts can be managed using the addons API.

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN" $CCP/v3/clusters/<your_cluster_uuid>/addons/

Example

```
export CLUSTER=35de61b9-5175-40d5-bea3-1b058fb22c45
curl -k -X GET -H "X-Auth-Token":"$TOKEN"
$CCP/v3/clusters/$CLUSTER/addons/
```

Response


```
{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
        {
            "name": "ccp-monitor",
            "namespace": "default",
            "overrides": "",
            "overrideFiles": [],
            "status": {},
```

```
            "url": "/opt/ccp/charts/ccp-monitor.tgz"
        },
        {
            "name": "metrics",
            "namespace": "default",
            "overrides": "",
            "overrideFiles": [],
            "status": {},
            "url": "metrics-server"
        }
    ]
}
```


6.16 Listing Catalog

Built-in addons can be listed using the catalog.

Command

curl -k -X GET -H "X-Auth-Token":"$TOKEN" $CCP/v3/clusters/<your_cluster_uuid>/catalog/

Example

```
export CLUSTER=35de61b9-5175-40d5-bea3-1b058fb22c45
curl -k -X GET -H "X-Auth-Token":"$TOKEN"
$CCP/v3/clusters/$CLUSTER/addons/
```

Response


```
{
    "_ccp-monitor": {
        "name": "ccp-monitor",
        "description": "Monitoring",
        "chart": "/opt/ccp/charts/ccp-monitor.tgz",
        "url": "/opt/ccp/charts/ccp-monitor.tgz"
    },
    "_ccp-efk": {
        "name": "ccp-efk",
        "description": "Logging",
        "chart": "/opt/ccp/charts/ccp-efk.tgz",
        "url": "/opt/ccp/charts/ccp-efk.tgz"
    },
    "_ccp-kubernetes-dashboard": {
        "name": "kubernetes-dashboard",
        "description": "Dashboard",
        "chart": "/opt/ccp/charts/kubernetes-dashboard.tgz",
        "overrideFiles": [
            "/opt/ccp/charts/kubernetes-dashboard.yaml"
        ],
        "url": "/opt/ccp/charts/kubernetes-dashboard.tgz"
    }
}
```


6.17 Adding an Addon

Command

curl -k -v -H "Content-Type:application/json" -H "X-Auth-Token:$TOKEN" https://$HOST/v3/clusters/$CLUSTER/addons/ -d '{"name":"addon_name", "url": "addn_url"}'

For built-in add-ons, the response for an addon for the /catalog listing can be used as payload for the addon creation.

Example

```
curl -k -H "Content-Type:application/json" -X POST -H "X-Auth-
Token":"$TOKEN" $CCP/v3/clusters/$CLUSTER/addons/ -d '{"name":
"ccp-monitor",
        "description": "Monitoring",
        "chart": "/opt/ccp/charts/ccp-monitor.tgz",
        "url": "/opt/ccp/charts/ccp-monitor.tgz"}'
```

Response

```
{
    "name": "ccp-monitor",
    "namespace": "default",
    "url": "/opt/ccp/charts/ccp-monitor.tgz"
}
```

6.18 Adding a Cisco Container Platform Addon

Command

curl -k -v -H "Content-Type:application/json" -H "X-Auth-Token:$TOKEN" https://$HOST/v3/clusters/$CLUSTER/addons/ -d '{"name":"addon_name", "url": "addn_url"}'

For built-in add-ons, the response for an addon for the /catalog listing can be used as payload for the addon creation.

Example

```
curl -k -H "Content-Type:application/json" -X POST -H "X-Auth-
Token":"$TOKEN" $CCP/v3/clusters/$CLUSTER/addons/ -d '{"name":
"ccp-monitor",
        "description": "Monitoring",
        "chart": "/opt/ccp/charts/ccp-monitor.tgz",
        "url": "/opt/ccp/charts/ccp-monitor.tgz"}'
```

Response

```
{
    "name": "ccp-monitor",
    "namespace": "default",
    "url": "/opt/ccp/charts/ccp-monitor.tgz"
}
```

6.19 Adding an Addon with Overrides

For example, consider the following override:

```
prometheus:
  nodeExporter:
    enabled: false
```

This override translates to: {"overrides": "prometheus:\n nodeExporter:\n enabled: false"}

```
curl -k -v -H "Content-Type:application/json" -H "X-Auth-
Token:$TOKEN" $CCP/v3/clusters/$CLUSTER/addons/ -d
```

```
'{
  "name": "ccp-monitor",
  "url": "_ccp-monitor",
  "namespace": "ccp",
  "overrides": "prometheus:\n  nodeExporter:\n    enabled: false"
}'

curl -k -v -H "Content-Type:application/json" -H "X-Auth-
Token:$TOKEN" http://127.0.0.1:8000/v3/clusters/$CLUSTER/addons/ -d
'{
  "name": "ccp-monitor",
  "url": "_ccp-monitor",
  "namespace": "ccp",
  "overrides": "hx:\n  url: 10.10.51.9\n  token:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2Vycy9hZG1pbmlzdHJhdG9yQHZzcGhlcmUu
bG9jYWwiLCJ1c2VyIjoiYWRtaW5pc3RyYXRvckB2c3BoZXJlLmxvY2FsIiwibGFiZWxzI
jp7Im5hbWUiOiJhYmkiLCJjb21wYW55IjoiY2lzY28ifSwic2NvcGUiOiJSRUFELE1PRE
lGWSIsImlzc3VlZEF0IjoxNTY1MjQ5OTY4NjM0LCJ0b2tlbkxpZmVUaW1lIjotMX0.DkQ
jyBqSO8py3625ki9X3na8vLNS2QDQUC5SOlVHL9M"
}'

curl -k -v \
  -H "Content-Type:application/json" \
  -H "X-Auth-Token:$TOKEN" \
  https://$HOST/v3/clusters/$CLUSTER/addons/ \
  -d
'{
  "name": "ccp-monitor",
  "url": "\/opt\/ccp\/charts\/ccp-monitor.tgz",
  "namespace": "ccp",
  "overrides": "prometheus:\n  server:\n    persistentVolume:\n
size: 16Gi\n    extraArgs:\n      storage.tsdb.size: 8Gi\n
storage.tsdb.retention.size: 2Gi"
}'
```

6.20 Deleting an Addon

Command

curl -k -v -X DELETE -H "X-Auth-Token:$TOKEN" $CCP/v3/clusters/$CLUSTER/addons/<addon-name>/

Example

```
curl -k -X DELETE -H "Content-Type:application/json" -H "X-Auth-
Token":"$TOKEN" $CCP/v3/clusters/$CLUSTER/addons/metrics/
```

Response

```
None
```

7 Cisco Container Platform API Reference

swagger-api.json Explore

Note: This section applies to v2 Clusters.

Cisco Container Platform Control Plane API Documentation

[Base URL: https://Cisco Container Platform Control Plane IP/2/]
swagger-api.json

Schemes
HTTP

/v3 CCP v3 API

DELETE `/v3/{resource}` forwards v3 API requests to the v3 API service

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

HEAD `/v3/{resource}` forwards v3 API requests to the v3 API service

PATCH `/v3/{resource}` forwards v3 API requests to the v3 API service

POST `/v3/{resource}` forwards v3 API requests to the v3 API service

PUT `/v3/{resource}` forwards v3 API requests to the v3 API service

2/aci_api accessing ACI api

POST `/2/aci_api/login` ACI login

2/aci_profiles List of ACI profile endpoints

GET `/2/aci_profiles` Get all ACI profiles

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

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

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

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

2/clusters List of cluster endpoints

GET `/2/clusters` Get all clusters

POST `/2/clusters` Create a cluster with the given specification

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

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

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

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

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

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

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

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

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

GET `/2/clusters/{clusterUUID}/helmcharts` Get HelmCharts object for a given cluster

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

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

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

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

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

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

2/keyvalues List of endpoints for key values

GET `/2/keyvalues/{key}`

POST `/2/keyvalues/{key}`

2/ldap List of ldap endpoints

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

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

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

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

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

GET `/2/ldap/setup` Get LDAP parameters

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

2/license List of licensing endpoints

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

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

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

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

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

2/localusers

GET `/2/localusers` Get CX local users

POST `/2/localusers` Create CX local user

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

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

PATCH `/2/localusers/{username}/password` Update

2/providerclientconfigs List of provider client config endpoints

GET	`/2/providerclientconfigs`	Get provider client configuration list
POST	`/2/providerclientconfigs`	Add provider client configuration
DELETE	`/2/providerclientconfigs/{clientconfigUUID}`	Delete provider client configuration
GET	`/2/providerclientconfigs/{clientconfigUUID}`	Get provider client configuration
PATCH	`/2/providerclientconfigs/{clientconfigUUID}`	Update provider client configuration
GET	`/2/providerclientconfigs/{clientconfigUUID}/clusters`	Get list of clusters who are using providerclientconfig
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter`	Gets the list of vSphere Data Centers.
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/cluster`	Gets the list of vSphere Clusters in a datacenter.
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/cluster/{clusterName}/gpu`	Gets the list of VSphere GPUs.
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/cluster/{clusterName}/pool`	Gets the list of vSphere Pools.
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/datastore`	Gets the list of vSphere Datastores.
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/network`	Gets the list of vSphere Networks.
GET	`/2/providerclientconfigs/{clientconfigUUID}/vsphere/datacenter/{datacenterName}/vm`	Gets the list of vSphere Virtual Machines.

2/rbac

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

2/system List of system endpoints

GET	`/2/system/CorcHealth`	Get corc health

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

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

POST `/2/system/login` Management server login

Models

api.ACILoginReply {

token*	string

}

api.ACILoginRequest {

apic_ips*	string
apic_password*	string
apic_username*	string

}

api.AddAuthorization {

Local*	boolean
Name*	string

}

api.AddAuthorizationReply {

AuthID*	string
Local*	boolean
Name*	string

}

api.CorcHealthReply {

}

api.CorcHealthRequest {

}

api.CreateLocalUserRequest {

```
    Disable*              boolean
    FirstName*            string
    LastName*             string
    Password*             string
    Role*                 string
    Token*                string
    UserName*             string
}
```

api.CreateLocalUserResponse {

```
}
```

api.CreateNodePoolReply {

```
    NodePool*
```

api.CreateNodePoolReply.NodePool {...}

```
}
```

api.CreateNodePoolReply.NodePool {

```
}
```

api.DeleteNodePoolReply {

```
}
```

api.GetVSphereClustersReply {

```
    Clusters*
                          [...]
}
```

api.GetVSphereDatacentersReply {

```
    Datacenters*
                          [...]
}
```

api.GetVSphereDatastoresReply {

```
    Datastores*
                          [...]
}
```

api.GetVSphereGpusReply {

```
    gpus*                       [...]
}
```

api.GetVSphereNetworksReply {

```
    Networks*                   [...]
}
```

api.GetVSpherePoolsReply {

```
    Pools*                      [...]
}
```

api.GetVSphereVMsReply {

```
    VMs*                        [...]
}
```

api.GpuHostIndex {

```
    gpu_type*           string
    hosts*                      [...]
}
```

api.HostGpuCount {

```
    count*              integer($int32)
    hostname*           string
}
```

api.LdapGroup {

```
    LdapDN*             string
    Role*               string
}
```

api.NodePoolRequest {

gpus*	[...]
labels*	string
memory*	integer($int64)
name*	string
node_ip_pool_uuid*	string
size*	integer($int32)
taints*	string
template*	string
vcpus*	integer($int32)

}

api.ResizeNodePoolRequest {

size*	integer($int32)

}

api.UpdateLocalUserPasswordRequest {

logged_in_user_password*	string
new_password*	string

}

api.UpdateLocalUserRequest {

Disable*	boolean
FirstName*	string
LastName*	string
Role*	string

}

ipam.IPInfo {

gateway*	string
id*	integer
ip*	string
mtu*	integer($int32)
nameservers*	[...]
netmask*	string
subnet	string
uuid*	string

}

ipam.LoadBalancerIPInfo {

IPInfo*	ipam.IPInfo {...}
never_release*	boolean

}

ipam.NodeIPInfo {

IPInfo*	ipam.IPInfo {...}
if_name*	string
type*	{...}

}

main.GetRoleResonse {

role*	string

}

types.ACIProfile {

aaep_name*	string
aci_allocator	types.ACIProfileAllocatorConfig {...}
aci_infra_vlan_id*	integer
aci_tenant*	string
aci_vmm_domain_name*	string
apic_hosts*	string
apic_password*	string
apic_username*	string
control_plane_contract_name*	string
l3_outside_network_name*	string
l3_outside_policy_name*	string
name*	string
nameservers*	[...]
uuid*	string
vrf_name*	string

}

types.ACIProfileAllocatorConfig {

multicast_range*	string
node_vlan_end*	integer
node_vlan_start*	integer
pod_subnet_start*	string
service_subnet_start*	string

}

types.Cluster {

Infra*	types.Cluster.Infra {...}
aci_profile_uuid*	string
aws_iam_enabled*	boolean
aws_iam_role_arn*	string
ccp_private_ssh_key*	string
ccp_public_ssh_key*	string
cluster_dashboard_url*	string
cluster_env_url*	string
deployer*	types.Kubeadm {...}
description*	string
etcd_encrypted*	boolean
harbor_admin_server_password*	string
harbor_registry_size*	string
helm_charts*	[...]
ingress_vip_pool_id*	string
ingress_vips*	[...]
is_adopt*	boolean
is_control_cluster*	boolean
is_harbor_enabled*	boolean
is_istio_enabled*	boolean
kubernetes_version*	string
labels*	[...]
load_balancer_ip_info_list*	[...]
load_balancer_ip_num*	integer($int32)
master_mac_addresses*	[...]
master_node_pool	types.Cluster.master_node_pool {...}
master_vip*	string
master_vip_addr_id*	string
masters*	integer($int32)
name*	string
network_plugin*	types.NetworkPluginProfile {...}
node_ip_pool_uuid	string
node_pools*	[...]
nodes*	[...]
ntp_pools*	[...]
ntp_servers*	[...]
provider_client_config_uuid*	string
registries_insecure*	[...]
registries_root_ca*	[...]
registries_self_signed*	[...]
secure_multitenancy_enabled*	boolean
ssh_key*	string
ssh_user*	string
state*	string
storage_class*	string
template*	string
tsig_key*	string
type*	{...}
uuid*	string
worker_node_pool	types.Cluster.worker_node_pool {...}
workers*	integer($int32)

}

types.Cluster.Infra {

}

types.Cluster.master_node_pool {

}

types.Cluster.node_pools {

}

types.Cluster.worker_node_pool {

}

types.GpuTypeCount {

```
count*                  integer($int32)
gpu_type*               string
```

}

types.HelmChart {

```
chart_url*              string
cluster_UUID*           string
helmchart_uuid*         string
name*                   string
options*                string
```

}

types.K8SNodeStatus {

```
LastTransitionTime*     string
NodeCondition*          string
NodeName*               string
NodeStatus*             string
```

}

types.K8SPodStatus {

```
LastTransitionTime*     string
PodCondition*           string
PodName*                string
PodStatus*              string
```

}

types.Kubeadm {

provider*	types.VsphereCloudProvider {...}
provider_type*	string

}

types.Label {

key*	string
value*	string

}

types.LdapSetup {

BaseDN*	string
InsecureSkipVerify*	boolean
Port*	integer
Server*	string
ServiceAccountDN*	string
ServiceAccountPassword*	string
StartTLS*	boolean

}

types.LoginStatus {

from_host*	string
last_fail*	string($date-time)
last_success*	string($date-time)
login_id*	string
proto*	string
status*	string
to_host*	string
total_fail*	integer($int32)

}

types.NetworkPluginProfile {

details*	string
name*	string
status*	string

}

types.Node {

cloud_init_data*	string
error_log*	string
ip_info*	[...]
is_master*	boolean
kubernetes_version*	string
mac_addresses*	[...]
name*	string
node_pool_id*	integer
node_pool_type*	string
private_ip*	string
public_ip*	string
state*	string
template*	string
uuid*	string

}

types.ProviderClientConfig {

config*	types.ProviderClientConfig.config {...}
name*	string
type*	{...}
uuid*	string

}

types.ProviderClientConfig.config {

}

types.SystemHealth {

CurrentNodes*	integer($int32)
ExpectedNodes*	integer($int32)
NodesStatus*	[...]
PodStatusList*	[...]
TotalSystemHealth*	string

}

types.VsphereClientConfig {

ip*	string
password	string
port*	integer
username*	string

}

types.VsphereCloudProvider {

client_config;omitempty*	**types.VsphereClientConfig** {...}
vsphere_client_config_uuid*	string
vsphere_datacenter*	string
vsphere_datastore*	string
vsphere_scsi_controller_type*	string
vsphere_working_dir*	string

}

ERROR {···}

Note: This section applies to v3 Clusters.

CCP v3 API (v3)

CCP v3 API documentation

Authentication

api_key

Security scheme type:	API Key
Header parameter name:	X-Auth-Token

aci-profiles

aci-profiles_list

AUTHORIZATIONS: api_key

Responses

^ 200

RESPONSE SCHEMA: application/json

Array [

id	string (Id)
cluster_count	integer (Cluster count)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable

multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

]

GET /aci-profiles/

Response samples

200

application/json

Copy Expand all Collapse all


```
[
  - {
      "id": "string",
      "cluster_count": 0,
      "name": "string",
      "apic_hosts": "string",
      "apic_username": "string",
```

```
        "apic_password": "string",
        "aci_vmm_domain_name": "string",
        "aci_infra_vlan_id": 1,
        "vrf_name": "string",
        "l3_outside_policy_name": "string",
        "l3_outside_network_name": "string",
        "aaep_name": "string",
      + "nameservers": [ … ],
        "control_plane_contract_name": "string",
        "aci_tenant": "string",
        "node_vlan_start": 1,
        "node_vlan_end": 1,
        "multicast_range": "string",
        "service_subnet_start": "string",
        "pod_subnet_start": "string"
    }
]
```


aci-profiles_create

AUTHORIZATIONS: api_key

REQUEST BODY SCHEMA: application/json

id	string (Id)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]

vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable
multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

Responses

^ 201

RESPONSE SCHEMA: application/json

id	string (Id)
cluster_count	integer (Cluster count)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters

apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable
multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

POST /aci-profiles/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
    "id": "string",
    "name": "string",
    "apic_hosts": "string",
    "apic_username": "string",
    "apic_password": "string",
    "aci_vmm_domain_name": "string",
    "aci_infra_vlan_id": 1,
    "vrf_name": "string",
    "l3_outside_policy_name": "string",
    "l3_outside_network_name": "string",
    "aaep_name": "string",
  - "nameservers": [
        "string"
    ],
    "control_plane_contract_name": "string",
    "aci_tenant": "string",
    "node_vlan_start": 1,
    "node_vlan_end": 1,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```


Response samples

201

application/json

Copy Expand all Collapse all


```
{
    "id": "string",
    "cluster_count": 0,
    "name": "string",
    "apic_hosts": "string",
    "apic_username": "string",
    "apic_password": "string",
    "aci_vmm_domain_name": "string",
```

```
    "aci_infra_vlan_id": 1,
    "vrf_name": "string",
    "l3_outside_policy_name": "string",
    "l3_outside_network_name": "string",
    "aaep_name": "string",
  - "nameservers": [
        "string"
    ],
    "control_plane_contract_name": "string",
    "aci_tenant": "string",
    "node_vlan_start": 1,
    "node_vlan_end": 1,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```


aci-profiles_read

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string A unique value identifying this aci profile.

Responses

200

RESPONSE SCHEMA: application/json

id	string (Id)
cluster_count	integer (Cluster count)

name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable
multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

GET /aci-profiles/{id}/

Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "id": "string",
    "cluster_count": 0,
    "name": "string",
    "apic_hosts": "string",
    "apic_username": "string",
    "apic_password": "string",
    "aci_vmm_domain_name": "string",
    "aci_infra_vlan_id": 1,
    "vrf_name": "string",
    "l3_outside_policy_name": "string",
    "l3_outside_network_name": "string",
    "aaep_name": "string",
  - "nameservers": [
        "string"
    ],
    "control_plane_contract_name": "string",
    "aci_tenant": "string",
    "node_vlan_start": 1,
    "node_vlan_end": 1,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```


---

aci-profiles_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string A unique value identifying this aci profile.

REQUEST BODY SCHEMA: application/json

id	string (Id)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable
multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)

pod_subnet_start	string (Pod subnet start)

Responses

^ 200

RESPONSE SCHEMA: application/json

id	string (Id)
cluster_count	integer (Cluster count)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters

node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable
multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

PUT /aci-profiles/{id}/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
   "id": "string",
   "name": "string",
   "apic_hosts": "string",
   "apic_username": "string",
   "apic_password": "string",
   "aci_vmm_domain_name": "string",
   "aci_infra_vlan_id": 1,
   "vrf_name": "string",
   "l3_outside_policy_name": "string",
   "l3_outside_network_name": "string",
   "aaep_name": "string",
 - "nameservers": [
      "string"
   ],
   "control_plane_contract_name": "string",
   "aci_tenant": "string",
   "node_vlan_start": 1,
   "node_vlan_end": 1,
   "multicast_range": "string",
```

```
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "id": "string",
    "cluster_count": 0,
    "name": "string",
    "apic_hosts": "string",
    "apic_username": "string",
    "apic_password": "string",
    "aci_vmm_domain_name": "string",
    "aci_infra_vlan_id": 1,
    "vrf_name": "string",
    "l3_outside_policy_name": "string",
    "l3_outside_network_name": "string",
    "aaep_name": "string",
  - "nameservers": [
        "string"
    ],
    "control_plane_contract_name": "string",
    "aci_tenant": "string",
    "node_vlan_start": 1,
    "node_vlan_end": 1,
    "multicast_range": "string",

    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```


aci-profiles_partial_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string A unique value identifying this aci profile.

REQUEST BODY SCHEMA: application/json

id	string (Id)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings
control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable

multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

Responses

200

RESPONSE SCHEMA: application/json

id	string (Id)
cluster_count	integer (Cluster count)
name required	string (Name) [1 .. 255] characters
apic_hosts required	string (Apic hosts) [1 .. 4096] characters
apic_username required	string (Apic username) non-empty
apic_password required	string (Apic password) non-empty
aci_vmm_domain_name required	string (Aci vmm domain name) [1 .. 255] characters
aci_infra_vlan_id required	integer (Aci infra vlan id) [1 .. 4094]
vrf_name required	string (Vrf name) [1 .. 255] characters
l3_outside_policy_name required	string (L3 outside policy name) [1 .. 255] characters
l3_outside_network_name required	string (L3 outside network name) [1 .. 255] characters
aaep_name required	string (Aaep name) [1 .. 255] characters
nameservers required	Array of strings

control_plane_contract_name required	string (Control plane contract name) non-empty
aci_tenant required	string (Aci tenant) [1 .. 255] characters
node_vlan_start	integer (Node vlan start) [1 .. 4094] Nullable
node_vlan_end	integer (Node vlan end) [1 .. 4094] Nullable
multicast_range	string (Multicast range)
service_subnet_start	string (Service subnet start)
pod_subnet_start	string (Pod subnet start)

PATCH /aci-profiles/{id}/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
    "id": "string",
    "name": "string",
    "apic_hosts": "string",
    "apic_username": "string",
    "apic_password": "string",
    "aci_vmm_domain_name": "string",
    "aci_infra_vlan_id": 1,
    "vrf_name": "string",
    "l3_outside_policy_name": "string",
    "l3_outside_network_name": "string",
    "aaep_name": "string",
  - "nameservers": [
        "string"
    ],
    "control_plane_contract_name": "string",
```

```
    "aci_tenant": "string",
    "node_vlan_start": 1,
    "node_vlan_end": 1,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "id": "string",
    "cluster_count": 0,
    "name": "string",
    "apic_hosts": "string",
    "apic_username": "string",
    "apic_password": "string",
    "aci_vmm_domain_name": "string",
    "aci_infra_vlan_id": 1,
    "vrf_name": "string",
    "l3_outside_policy_name": "string",
    "l3_outside_network_name": "string",
    "aaep_name": "string",
  - "nameservers": [
        "string"
    ],
    "control_plane_contract_name": "string",

    "aci_tenant": "string",
    "node_vlan_start": 1,
    "node_vlan_end": 1,
    "multicast_range": "string",
    "service_subnet_start": "string",
    "pod_subnet_start": "string"
}
```

aci-profiles_delete

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string A unique value identifying this aci profile.

Responses

— 204

```
DELETE  /aci-profiles/{id}/
```

clusters

clusters_list

AUTHORIZATIONS: api_key

Responses

— 200

GET /clusters/

---

clusters_create

AUTHORIZATIONS: api_key

Responses

— 201

POST /clusters/

---

clusters_addons_list

AUTHORIZATIONS: api_key

PATH PARAMETERS

cluster_pk required	string

QUERY PARAMETERS

page	integer A page number within the paginated result set.

page_size	integer Number of results to return per page.

Responses

^ 200

RESPONSE SCHEMA: application/json

count required	integer
next	string <uri> Nullable
previous	string <uri> Nullable
results > required	Array of objects

GET /clusters/{cluster_pk}/addons/

Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "count": 0,
    "next": "http://example.com",
    "previous": "http://example.com",
  - "results": [
      + { … }
    ]
}
```

clusters_addons_create

AUTHORIZATIONS: api_key

PATH PARAMETERS

cluster_pk required	string

REQUEST BODY SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
url required	string (Url) non-empty

Responses

^ 201

RESPONSE SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
status >	object (Status)

url required	string (Url) non-empty

POST /clusters/{cluster_pk}/addons/

Request samples

Payload

application/json

Copy Expand all Collapse all

```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
    "url": "string"
}
```

Response samples

201

application/json

Copy Expand all Collapse all

```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
```

```
  - "status": {
        "property1": "string",
        "property2": "string"
    },
    "url": "string"
}
```


clusters_addons_read

AUTHORIZATIONS: api_key

PATH PARAMETERS

cluster_pk required	string
id required	string

Responses

200

RESPONSE SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
status ›	object (Status)
url required	string (Url) non-empty

GET /clusters/{cluster_pk}/addons/{id}/

Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
  - "status": {
        "property1": "string",
        "property2": "string"
    },
    "url": "string"
}
```


---

clusters_addons_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

cluster_pk required	string
id required	string

REQUEST BODY SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
url required	string (Url) non-empty

Responses

200

RESPONSE SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
status >	object (Status)
url required	string (Url) non-empty

PUT /clusters/{cluster_pk}/addons/{id}/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
    "url": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
  - "status": {
        "property1": "string",
        "property2": "string"
    },
    "url": "string"
}
```


---

clusters_addons_partial_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

cluster_pk required	string
id required	string

REQUEST BODY SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
url required	string (Url) non-empty

Responses

200

RESPONSE SCHEMA: application/json

name required	string (Name) non-empty
namespace	string (Namespace) non-empty Default: `"default"`
overrides	string (Overrides) non-empty
overrideFiles	Array of strings
status >	object (Status)
url required	string (Url) non-empty

PATCH /clusters/{cluster_pk}/addons/{id}/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
    "url": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "name": "string",
    "namespace": "default",
    "overrides": "string",
  - "overrideFiles": [
        "string"
    ],
  - "status": {
        "property1": "string",
        "property2": "string"
    },
    "url": "string"
}
```

clusters_addons_delete

AUTHORIZATIONS: api_key

PATH PARAMETERS

cluster_pk required	string
id required	string

Responses

— 204

```
DELETE  /clusters/{cluster_pk}/addons/{id}/
```

clusters_read

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /clusters/{id}/

---

clusters_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

PUT /clusters/{id}/

---

clusters_partial_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

PATCH /clusters/{id}/

---

clusters_delete

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 204

DELETE /clusters/{id}/

---

clusters_catalog

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

```
GET  /clusters/{id}/catalog/
```

ldap

ldap_groups_list

AUTHORIZATIONS: api_key

Responses

^ 200

RESPONSE SCHEMA: application/json

```
Array [
```

all_clusters_auth	string (All clusters auth)

clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

]

GET /ldap/groups/

Response samples

200

application/json

Copy Expand all Collapse all

```
[
  - {
        "all_clusters_auth": "string",
      + "clusters": [ … ],
        "ldap_dn": "string",
        "role": "string"
    }
]
```

ldap_groups_create

AUTHORIZATIONS: api_key

REQUEST BODY SCHEMA: application/json

clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters

role required	string (Role)

Responses

^ 201

RESPONSE SCHEMA: application/json

all_clusters_auth	string (All clusters auth)
clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

POST /ldap/groups/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```


Response samples

201

application/json

Copy Expand all Collapse all


```
{
    "all_clusters_auth": "string",
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```


ldap_groups_read

AUTHORIZATIONS: api_key

PATH PARAMETERS

ldap_dn required	string A unique value identifying this ldap group.

Responses

200

RESPONSE SCHEMA: application/json

all_clusters_auth	string (All clusters auth)
clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

GET /ldap/groups/{ldap_dn}/

Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "all_clusters_auth": "string",
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```


ldap_groups_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

ldap_dn required	string A unique value identifying this ldap group.

REQUEST BODY SCHEMA: application/json

clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

Responses

200

RESPONSE SCHEMA: application/json

all_clusters_auth	string (All clusters auth)
clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

PUT /ldap/groups/{ldap_dn}/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "all_clusters_auth": "string",
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```


---

ldap_groups_partial_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

ldap_dn required	string A unique value identifying this ldap group.

REQUEST BODY SCHEMA: application/json

clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

Responses

^ 200

RESPONSE SCHEMA: application/json

all_clusters_auth	string (All clusters auth)

clusters >	Array of objects
ldap_dn required	string (Ldap dn) [1 .. 255] characters
role required	string (Role)

PATCH /ldap/groups/{ldap_dn}/

Request samples

Payload

application/json

Copy Expand all Collapse all


```
{
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "all_clusters_auth": "string",
  - "clusters": [
      + { … }
    ],
    "ldap_dn": "string",
    "role": "string"
}
```

ldap_groups_delete

AUTHORIZATIONS: api_key

PATH PARAMETERS

ldap_dn required	string A unique value identifying this ldap group.

Responses

— 204

```
DELETE  /ldap/groups/{ldap_dn}/
```

ldap_setup_list

AUTHORIZATIONS: api_key

Responses

— 200

GET /ldap/setup/

---

ldap_setup_update

AUTHORIZATIONS: api_key

Responses

— 200

PUT /ldap/setup/

---

local-users

local-users_list

AUTHORIZATIONS: api_key

Responses

^ 200

RESPONSE SCHEMA: application/json

```
Array [
```

username required	string (Username)	[1 .. 255] characters
first_name	string (First name)	non-empty Default: " "
last_name	string (Last name)	non-empty Default: " "
disable	boolean (Disable)	
role required	string (Role)	
password required	string (Password)	[1 .. 127] characters

```
]
```

GET /local-users/

Response samples

200

application/json

Copy Expand all Collapse all


```
[
  - {
      "username": "string",
      "first_name": "",
      "last_name": "",
      "disable": true,
      "role": "string",
      "password": "string"
    }
]
```

local-users_create

AUTHORIZATIONS: api_key

REQUEST BODY SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

Responses

^ 201

RESPONSE SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)

role required	string (Role)
password required	string (Password) [1 .. 127] characters

POST /local-users/

Request samples

Payload

application/json

Copy Expand all Collapse all

```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```

Response samples

201

application/json

Copy Expand all Collapse all

```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```

local-users_read

AUTHORIZATIONS: api_key

PATH PARAMETERS

username required	string A unique value identifying this local user.

Responses

200

RESPONSE SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

GET /local-users/{username}/

Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```


local-users_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

username required	string A unique value identifying this local user.

REQUEST BODY SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

Responses

^ 200

RESPONSE SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

PUT /local-users/{username}/

Request samples

Payload

application/json

Copy Expand all Collapse all

```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```

Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```


local-users_partial_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

username required	string A unique value identifying this local user.

REQUEST BODY SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: ""
last_name	string (Last name) non-empty Default: ""
disable	boolean (Disable)
role required	string (Role)

password required	string (Password) [1 .. 127] characters

Responses

200

RESPONSE SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

PATCH /local-users/{username}/

Request samples

Payload

application/json

Copy Expand all Collapse all

```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
```

```
    "disable": true,
    "role": "string",
    "password": "string"
}
```


Response samples

200

application/json

Copy Expand all Collapse all


```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```


---

local-users_delete

AUTHORIZATIONS: api_key

PATH PARAMETERS

username required	string A unique value identifying this local user.

Responses

— 204

DELETE /local-users/{username}/

local-users_set_password

AUTHORIZATIONS: api_key

PATH PARAMETERS

username required	string A unique value identifying this local user.

REQUEST BODY SCHEMA: application/json

username required	string (Username) [1 .. 255] characters
first_name	string (First name) non-empty Default: " "
last_name	string (Last name) non-empty Default: " "
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

Responses

^ 200

RESPONSE SCHEMA: application/json

username required	string (Username) [1 .. 255] characters

first_name	string (First name) non-empty Default: ""
last_name	string (Last name) non-empty Default: ""
disable	boolean (Disable)
role required	string (Role)
password required	string (Password) [1 .. 127] characters

PATCH /local-users/{username}/password/

Request samples

Payload

application/json

Copy Expand all Collapse all

```
{
    "username": "string",
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```

Response samples

200

application/json

Copy Expand all Collapse all

```
{
    "username": "string",
```

```
    "first_name": "",
    "last_name": "",
    "disable": true,
    "role": "string",
    "password": "string"
}
```


providers

providers_list

AUTHORIZATIONS: api_key

Responses

— 200


```
GET /providers/
```


providers_create

AUTHORIZATIONS: api_key

Responses

— 201

POST /providers/

---

providers_regions

AUTHORIZATIONS: api_key

Responses

— 200

GET /providers/regions/

---

providers_read

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/

---

providers_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

PUT /providers/{id}/

---

providers_partial_update

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

PATCH /providers/{id}/

---

providers_delete

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 204

DELETE /providers/{id}/

providers_availability_zones

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

```
GET  /providers/{id}/availability-zones/
```

providers_clusters

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/clusters/

---

providers_datacenters

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/datacenters/

---

providers_datastores

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/datastores/

---

providers_dns_servers

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/dns-servers/

---

providers_flavors

AUTHORIZATIONS: api_key

PATH PARAMETERS

id

required string

Responses

— 200

GET /providers/{id}/flavors/

providers_gpu_vms

AUTHORIZATIONS: api_key

PATH PARAMETERS

id
required string

Responses

— 200

GET /providers/{id}/gpu-vms/

providers_gpus

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/gpus/

providers_hx_overrides

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/hx-overrides/

---

providers_images

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/images/

---

providers_instance_types

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/instance-types/

---

providers_locations

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/locations/

---

providers_networks

AUTHORIZATIONS: api_key

PATH PARAMETERS

id

required string

Responses

— 200

GET /providers/{id}/networks/

---

providers_resource_groups

AUTHORIZATIONS: api_key

PATH PARAMETERS

id
required string

Responses

— 200

GET /providers/{id}/resource-groups/

---

providers_resource_pools

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/resource-pools/

providers_roles

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/roles/

---

providers_routers

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/routers/

---

providers_ssh_keys

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/ssh-keys/

---

providers_vms

AUTHORIZATIONS: api_key

PATH PARAMETERS

id required	string

Responses

— 200

GET /providers/{id}/vms/

---

system

system_entitlements_list

AUTHORIZATIONS: api_key

Responses

— 200

GET /system/entitlements/

---

system_healthz_list

AUTHORIZATIONS: api_key

Responses

— 200

GET /system/healthz/

---

system_login_create

AUTHORIZATIONS: api_key

Responses

— 201

```
POST  /system/login/
```

---

system_profile_read

AUTHORIZATIONS: api_key

Responses

— 200

```
GET  /system/profile/
```