

Servers
/v3

Authorize

ACI Profiles

GET	/aci-profiles/	List all ACI Profiles	
POST	/aci-profiles/	Create an ACI Profile	
GET	/aci-profiles/{aciProfileId}/	Info for a specific ACI Profile	
PATCH	/aci-profiles/{aciProfileId}/	Update a specific ACI Profile	
DELETE	/aci-profiles/{aciProfileId}/	Delete a specific ACI Profile	

Addons

GET	/clusters/{clusterId}/addons/	List all addons for the cluster	
POST	/clusters/{clusterId}/addons/	Create a cluster addon	
GET	/clusters/{clusterId}/addons/{addonId}/	Info for a specific cluster addon	
PATCH	/clusters/{clusterId}/addons/{addonId}/	Update a specific cluster addon	
DELETE	/clusters/{clusterId}/addons/{addonId}/	Delete a specific cluster addon	

Clusters

GET	/clusters/	List all clusters	
POST	/clusters/	Create a cluster	
GET	/clusters/{clusterId}/	Info for a specific cluster	
PATCH	/clusters/{clusterId}/	Update a specific cluster	
DELETE	/clusters/{clusterId}/	Delete a specific cluster	
GET	/clusters/{clusterId}/addons/	List all addons for the cluster	
POST	/clusters/{clusterId}/addons/	Create a cluster addon	
GET	/clusters/{clusterId}/addons/{addonId}/	Info for a specific cluster addon	
PATCH	/clusters/{clusterId}/addons/{addonId}/	Update a specific cluster addon	
DELETE	/clusters/{clusterId}/addons/{addonId}/	Delete a specific cluster addon	
GET	/clusters/{clusterId}/catalog/	Get the catalog for a specific cluster	
GET	/clusters/{clusterId}/upgrade-versions/	List all Kubernetes versions that the master/nodes can be upgraded to	
GET	/clusters/{clusterId}/node-groups/	List all cluster node groups	
POST	/clusters/{clusterId}/node-groups/	Create a cluster node group	
GET	/clusters/{clusterId}/node-groups/{nodeGroupId}/	Info for a specific cluster node groups	
PATCH	/clusters/{clusterId}/node-groups/{nodeGroupId}/	Update a specific cluster node group	
DELETE	/clusters/{clusterId}/node-groups/{nodeGroupId}/	Delete a specific cluster node group	

LDAP

GET	/ldap/groups/	List all LDAP Groups	
POST	/ldap/groups/	Create an LDAP Group	

GET	/ldap/groups/{ldapGroupId}/	Info for a specific LDAP Group	🔒
PATCH	/ldap/groups/{ldapGroupId}/	Update a specific LDAP Group	🔒
DELETE	/ldap/groups/{ldapGroupId}/	Delete a specific LDAP Group	🔒
GET	/ldap/setup/	Current LDAP setup	🔒
PUT	/ldap/setup/	Update the LDAP setup	🔒

Local Users

GET	/local-users/	List all Local Users	🔒
POST	/local-users/	Create a Local User	🔒
GET	/local-users/{localUserId}/	Info for a specific Local User	🔒
PATCH	/local-users/{localUserId}/	Update a specific Local User	🔒
DELETE	/local-users/{localUserId}/	Delete a specific Local User	🔒
PATCH	/local-users/{localUserId}/password/	Update a specific Local User password	🔒

Node Groups

GET	/clusters/{clusterId}/node-groups/	List all cluster node groups	🔒
POST	/clusters/{clusterId}/node-groups/	Create a cluster node group	🔒
GET	/clusters/{clusterId}/node-groups/{nodeGroupId}/	Info for a specific cluster node groups	🔒
PATCH	/clusters/{clusterId}/node-groups/{nodeGroupId}/	Update a specific cluster node group	🔒
DELETE	/clusters/{clusterId}/node-groups/{nodeGroupId}/	Delete a specific cluster node group	🔒

Providers

GET	/providers/	List all providers	🔒
POST	/providers/	Create a provider	🔒
GET	/providers/regions/	Regions for a given provider type	🔒
GET	/providers/{providerId}/	Info for a specific provider	🔒
PATCH	/providers/{providerId}/	Update a specific provider	🔒
DELETE	/providers/{providerId}/	Delete a specific provider	🔒
GET	/providers/{providerId}/availability-zones/	Availability Zones for a specific provider	🔒
GET	/providers/{providerId}/clusters/	Clusters (vSphere) for a specific provider (not CCP tenant clusters)	🔒
GET	/providers/{providerId}/datacenters/	Datacenters for a specific provider	🔒
GET	/providers/{providerId}/datastores/	Datastores for a specific provider	🔒
GET	/providers/{providerId}/dns-servers/	DNS Servers for a specific provider	🔒
GET	/providers/{providerId}/flavors/	Flavors (openstack) for a specific provider	🔒
GET	/providers/{providerId}/gpus/	GPUs for a specific provider	🔒
GET	/providers/{providerId}/gpu-vms/	GPU VMs for a specific provider	🔒
GET	/providers/{providerId}/hx-overrides/	HX Overrides for a specific provider	🔒
GET	/providers/{providerId}/images/	Images for a specific provider	🔒
GET	/providers/{providerId}/instance-types/	Instance types for a specific provider	🔒
GET	/providers/{providerId}/locations/	Locations for a specific provider	🔒
GET	/providers/{providerId}/networks/	Networks for a specific provider	🔒
GET	/providers/{providerId}/resource-groups/	Resource Groups for a specific provider	🔒

GET	/providers/{providerId}/resource-pools/	Resource Pools for a specific provider	🔒
GET	/providers/{providerId}/roles/	Roles for a specific provider	🔒
GET	/providers/{providerId}/routers/	Routers for a specific provider	🔒
GET	/providers/{providerId}/ssh-keys/	SSH Keys for a specific provider	🔒
GET	/providers/{providerId}/vms/	VMs for a specific provider	🔒

System

POST	/clusters/{clusterId}/health-check/	Cluster health check	🔒
GET	/system/healthz/	Health check	
POST	/system/login/	Login to a local user or LDAP account	
POST	/system/logout/	Logout and release token	🔒
GET	/system/profile/	Returns profile data for current user	🔒
GET	/system/version/	Returns CCP semver and API version	

Schemas

```

Aciprofile {
  control_plane_contract_name* string
  nameservers*
  node_vlan_start integer
  node_vlan_end integer
  multicast_range string
  service_subnet_start string
  pod_subnet_start string
  name* string
  apic_hosts* string
  apic_username* string
  apic_password* string
  aci_vmm_domain_name* string
  aci_infra_vlan_id* integer
  vrf_name* string
  l3_outside_policy_name* string
  l3_outside_network_name* string
  aaep_name* string
  aci_tenant* string
}

```

```

Addon {
  name* string
  namespace string
  overrides default: default
  overrideFiles string
  status {
  }
  url string
}

```

```

BaseCluster {
  name* string
  provider* string
  type* string
}

```

```

AKSCluster {
  name* string
  provider* string
  type* string
  agent_pool_name* string
  kubernetes_version* string
  location* string
  pod_cidr string
  resource_group_name* string
  service_cidr string
  virtual_kubelet_enabled boolean
  worker_instance_type* string
  worker_count* integer
  network_plugin string
  vnet_subnet_id string
  docker_bridge_cidr string
  dns_service_ip string
}

```

```

EKSCluster {
  name* string
  provider* string
  type* string
  region* string
  status string
  status_detail {
  }
  access_role_arn* string
  kubeconfig string
  vpc_sizing* {
    subnet string
    public_subnets {
    }
    private_subnets {
    }
  }
  ami* string
  instance_type* string
  k8s_version* string
  ssh keys
}

```

```

    }
    worker_count* integer
    vpc_id* string
  }
  }

```

```

GKECluster {
  name* string
  provider* string
  type* string
  status string
  current_master_version string
  kubeconfig string
  kubernetes_version* string
  node_pools* {
    GKENodeGroup {
      autoscaling boolean
      autoscaling_min_nodes integer
      autoscaling_max_nodes integer
      image_type* string
      initial_node_count* integer
      locations* {
        machine_type* string
        name* string
        preemptible boolean
        current_node_version string
      }
    }
  }
  master_upgrade boolean
  worker_upgrade boolean
}

```

```

OpenstackCluster {
  name* string
  provider* string
  type* string
  status string
  kubeconfig string
  network_plugin string
  network_type* string
  public_network_uuid* string
  vm_network_dns_servers {
    default: List []
  }
  kubernetes_version* string
  pod_cidr* string
  ssh_key_name* string
  master_count* integer
  force_id string
  flavor* string
  image* string
  worker_count* integer
  master_vip string
  vm_network_subnet string
  vm_network_uuid string
  http_proxy string
  https_proxy string
  ntp_pools {
    default: List []
  }
  ntp_servers {
    default: List []
  }
  root_ca_registries {
    default: List []
  }
  router_uuid string
  self_signed_registries {
  }
  etcd_encrypted boolean
  skip_management boolean
  nginx_ingress_class string
  cinder_as nova_as string
  docker_no_proxy {
    default: List []
  }
  control_plane boolean
}

```

```

VsphereCluster {
  name* string
  provider* string
  type* string
  description string
  aci_profile_name string
  control_plane boolean
  kubernetes_version string
  ip_allocation_method* string
  master_vip string
  load_balancer_num integer
  subnet_id string
  ntp_pools {
    default: List []
  }
  ntp_servers {
    default: List []
  }
  root_ca_registries {
    default: List []
  }
  self_signed_registries {
  }
  insecure_registries {
    default: List []
  }
  docker_http_proxy string
  docker_https_proxy string
  docker_bip string
  ingress_as_lb boolean
  nginx_ingress_class string
  etcd_encrypted boolean
  skip_management boolean
  docker_no_proxy {
    default: List []
  }
  routable_cidr string
  image_prefix string
  vsphere_infra* {
    datacenter* string
    datastore* string
    networks* {
      cluster* {
        resource_pool string
        folder string
      }
    }
  }
  master_group* VsphereNodeGroup > {...}
}

```

```

node_groups
  v {VsphereNodeGroup v {
    name* string
    size* integer
      minimum: 1
    template string
    vcpus integer
      default: 2
    memory_mb integer
      default: 16384
    gpus
      v {
        default: List []
      }
      v {
        type string
        count integer
          minimum: 1
      }
    ssh_user string
      default:
    ssh_key string
      default:
    nodes
      v {
        readOnly: true
        default: List []
      }
      v {
        name string
        status string
        phase string
        private_ip string
        public_ip string
      }
      kubernetes_version* string
    }
  }
network_plugin_profile
  v {
  }
}

```

```

Clusters v {Clusters v {
  anyof ->
    AKSCluster v {
      name* string
      provider* string
      type* string
      agent_pool_name* string
      kubernetes_version* string
      location* string
      pod_cidr string
      resource_group_name* string
      service_cidr string
      virtual_kubelet_enabled boolean
      worker_instance_type* string
      worker_count* integer
        minimum: 1
      network_plugin string
      vnet_subnet_id string
      docker_bridge_cidr string
      dns_service_ip string
    }
    EKSCluster v {
      name* string
      provider* string
      type* string
      region* string
      status string
      status_detail
        v {
        }
      access_role_arn* string
      kubeconfig string
      vpc_sizing*
        v {
          subnet string
          public_subnets v {string}
          private_subnets v {string}
        }
      ami* string
      instance_type* string
      k8s_version* string
      ssh_keys v {string}
      worker_count* integer
        minimum: 1
      vpc_id* string
    }
    GKECluster v {
      name* string
      provider* string
      type* string
      status string
      current_master_version string
      kubeconfig string
      kubernetes_version* string
      node_pools*
        v {GKENodeGroup v {
          autoscaling boolean
            default: false
          autoscaling_min_nodes integer
          autoscaling_max_nodes integer
          image_type* string
          initial_node_count* integer
          locations* v {...}
          machine_type* string
          name* string
          preemptible boolean
          current_node_version string
            readOnly: true
        }
      master_upgrade boolean
        default: false
      worker_upgrade boolean
        default: false
    }
    OpenstackCluster v {
      name* string
      provider* string
      type* string
      status string
      kubeconfig string
      network_plugin string
      network_type* string
      public_network_uuid* string
      vm_network_dns_servers
        v {
          default: List []
          string
        }
      kubernetes_version* string
      pod_cidr* string
      ssh_key_name* string
      master_count* integer
        minimum: 1
      force_id string
      flavor* string
      image* string
      worker_count* integer
        minimum: 1
      master_vip string
      vm_network_subnet string
      vm_network_uuid string
      http_proxy string
      https_proxy string
      ntp_pools
        v {
          default: List []
          string
        }
      ntp_servers
        v {
          default: List []
          string
        }
      root_ca_registries
        v {
          default: List []
          string
        }
      router_uuid string
      self_signed_registries
        v {

```



```
base_on* string
service_account_dn* string
service_account_password* string
start_tls* boolean
insecure_skip_verify* boolean
validate_before_save* boolean
}
```

```
LocalUser {
  username* string
  first_name string
  last_name string
  role* string
  disable* boolean
  password* string
  password_info PasswordInfo {
    < * >:
    {
      expired boolean
      warning boolean
      expires_on string
      change_required boolean
    }
  }
}
```

```
GKENodeGroup {
  autoscaling boolean
  autoscaling_min_nodes default: false integer
  autoscaling_max_nodes integer
  image_type* string
  initial_node_count* integer
  locations*
  machine_type*
  name* string
  preemptible boolean
  current_node_version string
  readOnly: true
}
```

```
VsphereNodeGroup {
  name* string
  size* integer
  minimum: 1
  template string
  vcpus integer
  default: 2
  memory_mb integer
  default: 16384
  gpus
  {
    type string
    count integer
    minimum: 1
  }
  ssh_user string
  default:
  ssh_key string
  default:
  nodes
  {
    name string
    status string
    phase string
    private_ip string
    public_ip string
  }
  readOnly: true
  default: List []
  kubernetes_version* string
}
```

```
NodeGroups { NodeGroups {
  anyOf ->
  GKENodeGroup {
    autoscaling boolean
    default: false
    autoscaling_min_nodes integer
    autoscaling_max_nodes integer
    image_type* string
    initial_node_count* integer
    locations*
    machine_type*
    name* string
    preemptible boolean
    current_node_version string
    readOnly: true
  }
  VsphereNodeGroup {
    name* string
    size* integer
    minimum: 1
    template string
    vcpus integer
    default: 2
    memory_mb integer
    default: 16384
    gpus
    {
      type string
      count integer
      minimum: 1
    }
    ssh_user string
    default:
    ssh_key string
    default:
    nodes
    {
      name string
      status string
      phase string
      private_ip string
      public_ip string
    }
    readOnly: true
    default: List []
    kubernetes_version* string
  }
}
```

```
BaseProvider {
  name* string
  type* string
}
```

```
AKSProvider {
  name* string
  type* string
  app_name* string
  client_id* string
  client_secret* string
  tenant_id* string
}
```

```
subscription_id* string
}
```

```
EKSProvider {
  name* string
  type* string
  role_arn* string
  access_key_id* string
  secret_access_key* string
}
```

```
GKEProvider {
  name* string
  type* string
  credentials* {
    }
}
```

```
OpenstackProvider {
  name* string
  type* string
  auth_url* string
  ca_cert* string
  domain_name* string
  insecure_skip_verify boolean
  password* string
  region* string
  tenant_name* string
  username* string
}
```

```
VsphereProvider {
  name* string
  type* string
  address* string
  port* integer
  username* string
  password* string
  insecure_skip_verify* boolean
}
```

```
Providers { Providers {
  anyOf ->
    AKSProvider {
      name* string
      type* string
      app_name* string
      client_id* string
      client_secret* string
      tenant_id* string
      subscription_id* string
    }
    EKSProvider {
      name* string
      type* string
      role_arn* string
      access_key_id* string
      secret_access_key* string
    }
    GKEProvider {
      name* string
      type* string
      credentials* {
        }
    }
    OpenstackProvider {
      name* string
      type* string
      auth_url* string
      ca_cert* string
      domain_name* string
      insecure_skip_verify boolean
      password* string
      region* string
      tenant_name* string
      username* string
    }
    VsphereProvider {
      name* string
      type* string
      address* string
      port* integer
      username* string
      password* string
      insecure_skip_verify* boolean
    }
  }
}
```

```
Error {
  code* integer($int32)
  message* string
}
```

```
PasswordInfo {
  < * >:
  {
    expired boolean
    warning boolean
    expires_on string
    change_required boolean
  }
}
```