Microsoft Azure Red Hat OpenShift with hosted control planes

The Microsoft Azure Red Hat OpenShift with hosted control planes service is a fully-managed turnkey application platform that allows you to focus on what matters most, delivering value to your customers by building and deploying applications. Red Hat and Microsoft SRE experts manage the underlying platform so you do not have to worry about infrastructure management. Azure Red Hat OpenShift with hosted control planes provides seamless integration with a wide range of Azure compute, database, analytics, machine learning, networking, mobile, and other services to further accelerate the building and delivering of differentiating experiences to your customers.

Azure Red Hat OpenShift with hosted control planes offers a reduced-cost solution to create an Azure Red Hat OpenShift with hosted control planes cluster with a focus on efficiency. You can quickly create a new cluster and deploy applications in minutes.

You subscribe to the service directly from your Azure account. After you create clusters, you can operate your clusters with the OpenShift web console, the Azure CLI, or the Azure Red Hat OpenShift REST API.

You receive OpenShift updates with new feature releases and a shared, common source for alignment with OpenShift Container Platform.

Azure Red Hat OpenShift with hosted control planes uses user-assigned managed identities to obtain credentials to manage infrastructure in your Azure account.

Cluster node scaling

Microsoft Azure Red Hat OpenShift with hosted control planes requires a minimum of only two nodes, making it ideal for smaller projects while still being able to scale to support larger projects and enterprises. Easily add or remove compute nodes to match resource demand. Autoscaling allows you to automatically adjust the size of the cluster based on the current workload.

Fully managed underlying control plane infrastructure

Control plane components, such as the API server and etcd database, are hosted in a Red Hat-owned Azure account.

Flexible, consumption-based pricing

Scale to your business needs and pay as you go with flexible pricing and an on-demand hourly or annual billing model.

Single bill for Red Hat OpenShift and Azure usage

Customers will receive a single bill from Microsoft for both Red Hat OpenShift and Azure consumption.

Fully integrated support experience

Installation, management, maintenance, and upgrades are performed by Red Hat site reliability engineers (SREs) with joint Red Hat and Microsoft support and a 99.95% service-level agreement (SLA).

Azure service integration

Azure has a robust portfolio of cloud services, such as compute, storage, networking, database, analytics, and machine learning. All of these services are directly accessible through Azure Red Hat OpenShift with hosted control planes. This makes it easier to build, operate, and scale globally and on-demand through a familiar management interface.

Maximum availability

Deploy node pools across multiple availability zones in supported regions to maximize availability and maintain high availability for your most demanding mission-critical applications and data.

Optimized clusters

Choose from memory-optimized, compute-optimized, or general purpose virtual machine sizes with clusters sized to meet your needs.

Microsoft Azure Red Hat OpenShift with hosted control planes service definition

This documentation outlines the service definition for the Microsoft Azure Red Hat OpenShift with hosted control planes managed service.

Account management

This section provides information about the service definition for Azure Red Hat OpenShift with hosted control planes account management.

Billing and pricing

Azure Red Hat OpenShift with hosted control planes is billed directly to your Azure account. Azure Red Hat OpenShift with hosted control planes pricing is consumption based, with annual commitments or three-year commitments for greater discounting. The total cost of Azure Red Hat OpenShift with hosted control planes consists of two components:

- Azure Red Hat OpenShift with hosted control planes service fees
- Azure infrastructure fees

Visit the Microsoft Azure Red Hat Openshift pricing page for more details.

Cluster self-service

Customers can self-service their clusters, including, but not limited to:

- Create a cluster
- Delete a cluster
- Add or remove an identity provider
- Add or remove a user from an elevated group
- Configure cluster privacy
- Add or remove node pools and configure autoscaling

You can perform these self-service tasks using the Microsoft Azure Red Hat Openshift APIs, Azure CLI, or Azure Portal.

Additional resources

- Red Hat Operator support

Azure compute types

All Azure Red Hat OpenShift with hosted control planes clusters require a minimum of 2 worker nodes. Shutting down the underlying infrastructure through the cloud provider console is unsupported and can lead to data loss.

> **NOTE**
>
> Approximately one vCPU core and 1 GiB of memory are reserved on each worker node and removed from allocatable resources. This reservation of resources is necessary to run processes required by the underlying platform. These processes include system daemons such as udev, kubelet, and container runtime among others. The reserved resources also account for kernel reservations. OpenShift Container Platform core systems such as audit log aggregation, metrics collection, DNS, image registry, SDN, and others might consume additional allocatable resources to maintain the stability and maintainability of the cluster. The additional resources consumed might vary based on usage. For additional information, see the Kubernetes documentation.

Additional resources

- For a detailed listing of supported compute types, see Sizes for virtual machines in Azure.

Regions and availability zones

The following Azure regions are currently available for Azure Red Hat OpenShift with hosted control planes:

- UK South
- Canada Central
- Australia East
- Switzerland North
- Brazil South
- Central India

> **WARNING**
>
> The region cannot be changed after a cluster has been deployed.

Service Level Agreement (SLA)

Any SLAs for the service itself are defined in Appendix 4 of the Red Hat Enterprise Agreement Appendix 4 (Online Subscription Services).

Limited support status

When a cluster transitions to a *Limited Support* status, Red Hat no longer proactively monitors the cluster, the SLA is no longer applicable, and credits requested against the SLA are denied. It does not mean that you no longer have product support. In some cases, the cluster can return to a fully-supported status if you remediate the violating factors. However, in other cases, you might have to delete and recreate the cluster.

A cluster might move to a Limited Support status for many reasons, including the following scenarios:

If you remove or replace any native Azure Red Hat OpenShift with hosted control planes components or any other component that is installed and managed by Red Hat
If cluster administrator permissions were used, Red Hat is not responsible for any of your or your authorized users' actions, including those that affect infrastructure services, service availability, or data loss. If Red Hat detects any such actions, the cluster might transition to a Limited Support status. Red Hat notifies you of the status change and you should either revert the action or create a support case to explore remediation steps that might require you to delete and recreate the cluster.

If you have questions about a specific action that might cause a cluster to move to a Limited Support status or need further assistance, open a support ticket.

Support

Azure Red Hat OpenShift with hosted control planes includes Red Hat Premium Support, which can be accessed by using the Red Hat Customer Portal.

See the Red Hat Production Support Terms of Service for support response times.

Azure support is subject to a customer's existing support contract with Azure.

Monitoring

This section provides information about the service definition for Azure Red Hat OpenShift with hosted control planes monitoring.

Cluster metrics

Azure Red Hat OpenShift with hosted control planes clusters come with an integrated Prometheus stack for cluster monitoring including CPU, memory, and network-based metrics. This is accessible through the web console. These metrics also allow for horizontal pod autoscaling based on CPU or memory metrics provided by an Azure Red Hat OpenShift with hosted control planes user.

Cluster notifications

Cluster notifications are messages about the status, health, or performance of your cluster.

Cluster notifications are the primary way that Red Hat Site Reliability Engineering (SRE) communicates with you about the health of your managed cluster. SRE may also use cluster notifications to prompt you to perform an action in order to resolve or prevent an issue with your cluster.

Cluster owners and administrators must regularly review and action cluster notifications to ensure clusters remain healthy and supported.

You can view cluster notifications in the Red Hat Hybrid Cloud Console, in the **Cluster history** tab for your cluster. By default, only the cluster owner receives cluster notifications as emails. If other users need to receive cluster notification emails, add each user as a notification contact for your cluster.

Networking

This section provides information about the service definition for Azure Red Hat OpenShift with hosted control planes networking.

Custom domains for applications

> **WARNING**
>
> Starting with Azure Red Hat OpenShift with hosted control planes 4.14, the Custom Domain Operator is deprecated. To manage Ingress in Azure Red Hat OpenShift with hosted control planes 4.14 or later, use the Ingress Operator. The functionality is unchanged for Azure Red Hat OpenShift with hosted control planes 4.13 and earlier versions.

To use a custom hostname for a route, you must update your DNS provider by creating a canonical name (CNAME) record. Your CNAME record should map the OpenShift canonical router hostname to your custom domain. The OpenShift canonical router hostname is shown on the *Route Details* page after a route is created. Alternatively, a wildcard CNAME record can be created once to route all subdomains for a given hostname to the cluster's router.

Domain validated certificates

Azure Red Hat OpenShift with hosted control planes includes TLS security certificates needed for both internal and external services on the cluster. For external routes, there are two separate TLS wildcard certificates that are provided and installed on each cluster: one is for the web console and route default hostnames, and the other is for the API endpoint. OneCert is the certificate authority used for certificates. Routes within the cluster, such as the internal API endpoint, use TLS certificates signed by the cluster's built-in certificate authority and require the CA bundle available in every pod for trusting the TLS certificate.

Custom certificate authorities for builds

Azure Red Hat OpenShift with hosted control planes supports the use of custom certificate authorities to be trusted by builds when pulling images from an image registry.

Load balancers

Azure Red Hat OpenShift with hosted control planes only deploys load balancers from the default ingress controller. All other load balancers can be optionally deployed by a customer for secondary ingress controllers or Service load balancers.

Cluster ingress

Project administrators can add route annotations for many different purposes, including ingress control through IP allow-listing.

Ingress policies can also be changed by using `NetworkPolicy` objects, which leverage the `ovs-networkpolicy` plugin. This allows for full control over the ingress network policy down to the pod level, including between pods on the same cluster and even in the same namespace.

All cluster ingress traffic will go through the defined load balancers. Direct access to all nodes is blocked by cloud configuration.

Cluster egress

Pod egress traffic control through `EgressNetworkPolicy` objects can be used to prevent or limit outbound traffic in Azure Red Hat OpenShift with hosted control planes.

Cloud network configuration

Azure Red Hat OpenShift with hosted control planes allows for the configuration of a private network connection through Azure services, such as:

- Vnet
- Vnet peering
- Transit Gateway
- Direct Connect

IMPORTANT

Red Hat site reliability engineers (SREs) do not monitor private network connections. Monitoring these connections is the responsibility of the customer.

DNS forwarding

Azure Red Hat OpenShift with hosted control planes clusters that have a private cloud network configuration, a customer can specify internal DNS servers available on that private connection, that should be queried for explicitly provided domains.

Network verification

Network verification checks run automatically when you deploy a Azure Red Hat OpenShift with hosted control planes cluster into an existing Virtual Private Cloud (VPC) or create an additional machine pool with a subnet that is new to your cluster. The checks validate your network configuration and highlight errors, enabling you to resolve configuration issues prior to deployment.

You can also run the network verification checks manually to validate the configuration for an existing cluster.

Storage

This section provides information about the service definition for Azure Red Hat OpenShift with hosted control planes storage.

Encrypted-at-rest OS and node storage

Worker nodes use encrypted-at-rest Amazon Elastic Block Store (Amazon EBS) storage.

Encrypted-at-rest PV

EBS volumes that are used for PVs are encrypted-at-rest by default.

Block storage (RWO)

Persistent volumes (PVs) are backed by Azure Disk which is Read-Write-Once.

PVs can be attached only to a single node at a time and are specific to the availability zone in which they were provisioned. However, PVs can be attached to any node in the availability zone.

Each cloud provider has its own limits for how many PVs can be attached to a single node. See the specific instance types for documented maximums

Shared Storage (RWX)

The Azure Files CSI Driver can be used to provide RWX support for Azure Red Hat OpenShift with hosted control planes. A Red Hat Operator is provided and installed to simplify usage.

Platform

This section provides information about the service definition for the Azure Red Hat OpenShift with hosted control planes platform.

Autoscaling

Node autoscaling is available on Azure Red Hat OpenShift with hosted control planes. You can configure the autoscaler option to automatically scale the number of machines in a cluster.

Daemonsets

Customers can create and run daemonsets on Azure Red Hat OpenShift with hosted control planes. To restrict daemonsets to only running on worker nodes, use the following `nodeSelector`:

```yaml
spec:
  nodeSelector:
    role: worker
```

Node labels

Custom node labels are created by Red Hat during node creation and cannot be changed on Azure Red Hat OpenShift with hosted control planes clusters at this time. However, custom labels are supported when creating new machine pools.

Node lifecycle

Worker nodes are not guaranteed longevity, and may be replaced at any time as part of the normal operation and management of OpenShift.

A worker node might be replaced in the following circumstances:

- Machine health checks are deployed and configured to ensure that a worker node with a `NotReady` status is replaced to ensure smooth operation of the cluster.

- An Azure instance may be terminated when Azure detects irreparable failure of the underlying hardware that hosts the instance.
- During upgrades, a new, upgraded node is first created and joined to the cluster. Once this new node has been successfully integrated into the cluster via the previously described automated health checks, an older node is then removed from the cluster.

For all containerized workloads running on a Kubernetes based system, it is best practice to configure applications to be resilient of node replacements.

Cluster backup policy

Red Hat recommends object-level backup solutions for Azure Red Hat OpenShift with hosted control planes clusters. OpenShift API for Data Protection (OADP) is included in OpenShift but not enabled by default. Customers can configure OADP on their clusters to achieve object-level backup and restore capabilities.

Red Hat does not back up customer applications or application data. Customers are solely responsible for applications and their data, and must put their own backup and restore capabilities in place.

> ⓘ **WARNING**
>
> Customers are solely responsible for backing up and restoring their applications and application data. For more information about customer responsibilities, see "Shared responsibility matrix".

OpenShift version

Azure Red Hat OpenShift with hosted control planes is run as a service and is kept up to date with the latest OpenShift Container Platform version. Upgrade scheduling to the latest version is available.

Windows Containers

Red Hat OpenShift support for Windows Containers is not available on Azure Red Hat OpenShift with hosted control planes at this time.

Container engine

Azure Red Hat OpenShift with hosted control planes runs on OpenShift 4 and uses CRI-O as the only available container engine.

Operating system

Azure Red Hat OpenShift with hosted control planes runs on OpenShift 4 and uses Red Hat CoreOS as the operating system for all control plane and worker nodes.

Red Hat Operator support

Red Hat workloads typically refer to Red Hat-provided Operators made available through Operator Hub. Red Hat workloads are not managed by the Red Hat SRE team, and must be deployed on worker nodes. These Operators may require additional Red Hat subscriptions, and may incur additional cloud infrastructure costs. Examples of these Red Hat-provided Operators are:

- Red Hat Quay
- Red Hat Advanced Cluster Management
- Red Hat Advanced Cluster Security
- Red Hat OpenShift Service Mesh
- OpenShift Serverless
- Red Hat OpenShift Logging
- Red Hat OpenShift Pipelines

Kubernetes Operator support

All Operators listed in the OperatorHub marketplace should be available for installation. These Operators are considered customer workloads, and are not monitored by Red Hat SRE.

Security

This section provides information about the service definition for Azure Red Hat OpenShift with hosted control planes security.

Authentication provider

Authentication for the cluster can be configured using either ARM API or Azure CLI. Azure Red Hat OpenShift with hosted control planes is not an identity provider, and all access to the cluster must be managed by the customer as part of their integrated solution. The use of multiple identity providers provisioned at the same time is supported.

Privileged containers

Privileged containers are available for users with the `cluster-admin` role. Usage of privileged containers as `cluster-admin` is subject to the responsibilities and exclusion notes in the Red Hat Enterprise Agreement Appendix 4 (Online Subscription Services).

Cluster administration role

The administrator of Azure Red Hat OpenShift with hosted control planes has default access to the cluster-admin role for your organization’s cluster. While logged into an account with the cluster-admin role, users have increased permissions to run privileged security contexts.

Project self-service

By default, all users have the ability to create, update, and delete their projects. This can be restricted if a member of the dedicated-admin group removes the self-provisioner role from authenticated users:

```
$ oc adm policy remove-cluster-role-from-group self-
provisioner system:authenticated:oauth
```

Restrictions can be reverted by applying:

```
$ oc adm policy add-cluster-role-to-group self-provisioner
system:authenticated:oauth
```

Network security

With Azure Red Hat OpenShift with hosted control planes, Azure provides a standard DDoS protection on all load balancers, called Azure DDoS. This provides 95% protection against most commonly used level 3 and 4 attacks on all the public facing load balancers used for Azure Red Hat OpenShift with hosted control planes. A 10-second timeout is added for HTTP requests coming to the haproxy router to receive a response or the connection is closed to provide additional protection.

Virtual machine sizes

For a list of available worker node virtual machine types and sizes, see Worker nodes.

NOTE

Microsoft Azure Red Hat OpenShift with hosted control planes supports a maximum of 500 worker nodes. |

Azure Red Hat OpenShift with hosted control planes architecture

With Microsoft Azure Red Hat OpenShift with hosted control planes, the control plane is hosted in a Red Hat account and the worker nodes are deployed in the customer’s Azure account. The Azure Red Hat OpenShift with hosted control planes service hosts a highly-available, single-tenant OpenShift control plane.

The worker nodes are deployed in your Azure account and run on your virtual network. You can add additional private subnets from one or more availability zones to ensure high availability. Worker nodes are shared by OpenShift components and applications. OpenShift components such as the ingress controller, image registry, and monitoring are deployed on the worker nodes hosted on your virtual network.

Create a cluster

Microsoft Azure Red Hat OpenShift with hosted control planes is a managed OpenShift service that lets you quickly deploy and manage clusters. With Azure Red Hat OpenShift with hosted control planes, each cluster has a dedicated control plane that is isolated in an Azure service account.

This article shows you how to deploy an Azure Red Hat OpenShift with hosted control planes cluster using a Bicep file.

Prerequisites

Before deploying an Microsoft Azure Red Hat OpenShift with hosted control planes cluster, verify each of the following prerequisites.

Azure CLI requirement

Ensure you're using Azure CLI version 2.67.0 or higher. Use `az --version` to find the version of Azure CLI you installed. If you need to install or upgrade, see Install Azure CLI.

Verify resource quota

Ensure you have enough resource quota. Azure Red Hat OpenShift with hosted control planes requires a minimum of 20 cores to create and run an OpenShift cluster. The default Azure resource quota for a new Azure subscription doesn't meet the minimum cores requirement. To request an increase in your resource limit, see Standard quota: Increase limits by VM series.

For example, to check the current subscription quota of the smallest supported virtual machine family SKU "Standard DSv5":

```
$ LOCATION=eastus
$ az vm list-usage -l $LOCATION \
  --query "[?contains(name.value, 'standardDSv5Family')]" -o table
```

Verify permissions

In this article, you create a resource group which contains the virtual network and managed identities for the cluster. To create a resource group, you need Contributor and User Access Administrator permissions or Owner permissions on the resource group or subscription containing it. For more information, see Verify your permissions.

Register resource providers

The following resource providers must be registered in your Azure subscription:

- `Microsoft.RedHatOpenShift`
- `Microsoft.Compute`
- `Microsoft.Storage`
- `Microsoft.Authorization`

For more information about how to register any of these resource providers, see [Register the resource providers](#).

Create a Bicep file

Create a Bicep file that defines an Azure Red Hat OpenShift with hosted control planes cluster.

Default resources to define

To create an Azure Red Hat OpenShift with hosted control planes cluster, you must define the following resources:

- Network Security Group (NSG)
- Virtual network with an empty subnet
- A user-assigned managed identity and role assignment for each OpenShift cluster Operator
- A service managed identity and role assignments
- Hosted control plane (`Microsoft.RedHatOpenShift/hcpOpenShiftClusters` resource type)
- Node pool (`Microsoft.RedHatOpenShift/hcpOpenShiftClusters/nodePools` resource type)

Resources to define for customer-managed etcd encryption

If you want to use your own key to encrypt etcd, you must also define these additional resources:

- Azure Key Vault
- Custom KMS key to encrypt the etcd database
- A user-assigned managed identity to access the KMS key, and a role assignment that grants Key Vault permissions
- A role assignment that grants `Reader` permissions to the service managed identity

Example Bicep file

This example Bicep file defines a cluster that uses a custom KMS key to encrypt the etcd database. You can customize the cluster and node pool configuration properties as needed.

Example `azuredeploy.bicep` file

```
@description('Network Security Group Name')
param customerNsgName string

@description('Virtual Network Name')
param customerVnetName string

@description('Subnet Name')
param customerVnetSubnetName string

@description('Name of the hypershift cluster')
param clusterName string

@description('The Hypershift cluster managed resource group name')
param managedResourceGroupName string

@description('The name of the node pool')
param nodePoolName string

var randomSuffix = toLower(uniqueString(clusterName))
var addressPrefix = '10.0.0.0/16'
var subnetPrefix = '10.0.0.0/24'

resource customerNsg 'Microsoft.Network/networkSecurityGroups@2023-05-01' = {
  name: customerNsgName
  location: resourceGroup().location
  tags: {
    persist: 'true'
  }
}

resource customerVnet 'Microsoft.Network/virtualNetworks@2023-05-01' = {
  name: customerVnetName
  location: resourceGroup().location
  tags: {
    persist: 'true'
  }
  properties: {
    addressSpace: {
      addressPrefixes: [
        addressPrefix
      ]
    }
    subnets: [
      {
        name: customerVnetSubnetName
        properties: {
          addressPrefix: subnetPrefix
          networkSecurityGroup: {
            id: customerNsg.id
          }
        }
      }
    ]
  }
}

resource subnet 'Microsoft.Network/virtualNetworks/subnets@2022-07-01' existing = {
  name: customerVnetSubnetName
  parent: customerVnet
}

//
// C O N T R O L   P L A N E   I D E N T I T I E S
//

// Reader
var readerRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  'acdd72a7-3385-48ef-bd42-f606fba81ae7'
)

//
// C L U S T E R   A P I   A Z U R E   M I
//

resource clusterApiAzureMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-31' = {
  name: '${clusterName}-cp-cluster-api-azure-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift Hosted Control Planes Cluster API Provider
var hcpClusterApiProviderRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  '88366f10-ed47-4cc0-9fab-c8a06148393e'
)

resource hcpClusterApiProviderRoleSubnetAssignment 'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, clusterApiAzureMi.id, hcpClusterApiProviderRoleId,
subnet.id)
  scope: subnet
  properties: {
    principalId: clusterApiAzureMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: hcpClusterApiProviderRoleId
  }
}

resource serviceManagedIdentityReaderOnClusterApiAzureMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
clusterApiAzureMi.id)
  scope: clusterApiAzureMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// CONTROL PLANE OPERATOR MANAGED IDENTITY
//

resource controlPlaneMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-
31' = {
  name: '${clusterName}-cp-control-plane-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift Hosted Control Planes Control Plane Operator
var hcpControlPlaneOperatorRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  'fc0c873f-45e9-4d0d-a7d1-585aab30c6ed'
)

resource hcpControlPlaneOperatorVnetRoleAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, controlPlaneMi.id, hcpControlPlaneOperatorRoleId,
customerVnet.id)
  scope: customerVnet
  properties: {
    principalId: controlPlaneMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: hcpControlPlaneOperatorRoleId
  }
}

resource hcpControlPlaneOperatorNsgRoleAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, controlPlaneMi.id, hcpControlPlaneOperatorRoleId,
customerNsg.id)
  scope: customerNsg
  properties: {
    principalId: controlPlaneMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: hcpControlPlaneOperatorRoleId
  }
}

resource serviceManagedIdentityReaderOnControlPlaneMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
controlPlaneMi.id)
  scope: controlPlaneMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// CLOUD CONTROLLER MANAGER MANAGED IDENTITY
//

resource cloudControllerManagerMi
'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-31' = {
  name: '${clusterName}-cp-cloud-controller-manager-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift Cloud Controller Manager
var cloudControllerManagerRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  'a1f96423-95ce-4224-ab27-4e3dc72facd4'
)

resource cloudControllerManagerRoleSubnetAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, cloudControllerManagerMi.id,
cloudControllerManagerRoleId, subnet.id)
  scope: subnet
  properties: {
    principalId: cloudControllerManagerMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: cloudControllerManagerRoleId
  }
}

resource cloudControllerManagerRoleNsgAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, cloudControllerManagerMi.id,
cloudControllerManagerRoleId, customerNsg.id)
  scope: customerNsg
  properties: {
    principalId: cloudControllerManagerMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: cloudControllerManagerRoleId
  }
}

resource serviceManagedIdentityReaderOnCloudControllerManagerMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
cloudControllerManagerMi.id)
  scope: cloudControllerManagerMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// I N G R E S S   M A N A G E D   I D E N T I T Y
//

resource ingressMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-31' = {
  name: '${clusterName}-cp-ingress-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift Cluster Ingress Operator
var ingressOperatorRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  '0336e1d3-7a87-462b-b6db-342b63f7802c'
```

```
)

resource ingressOperatorRoleSubnetAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, ingressMi.id, ingressOperatorRoleId, subnet.id)
  scope: subnet
  properties: {
    principalId: ingressMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: ingressOperatorRoleId
  }
}

resource serviceManagedIdentityReaderOnIngressMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
ingressMi.id)
  scope: ingressMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// D I S K   C S I   D R I V E R   M A N A G E D   I D E N T I T Y
//

resource diskCsiDriverMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-
31' = {
  name: '${clusterName}-cp-disk-csi-driver-${randomSuffix}'
  location: resourceGroup().location
}

resource serviceManagedIdentityReaderOnDiskCsiDriverMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
diskCsiDriverMi.id)
  scope: diskCsiDriverMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}
```

```
//
// F I L E   C S I   D R I V E R   M A N A G E D   I D E N T I T Y
//

resource fileCsiDriverMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-31' = {
  name: '${clusterName}-cp-file-csi-driver-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift File Storage Operator
var fileStorageOperatorRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  '0d7aedc0-15fd-4a67-a412-efad370c947e'
)

resource fileStorageOperatorRoleSubnetAssignment 'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, fileCsiDriverMi.id, fileStorageOperatorRoleId, subnet.id)
  scope: subnet
  properties: {
    principalId: fileCsiDriverMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: fileStorageOperatorRoleId
  }
}

resource fileStorageOperatorRoleNsgAssignment 'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, fileCsiDriverMi.id, fileStorageOperatorRoleId, customerNsg.id)
  scope: customerNsg
  properties: {
    principalId: fileCsiDriverMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: fileStorageOperatorRoleId
  }
}

resource serviceManagedIdentityReaderOnFileCsiDriverMi 'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId, fileCsiDriverMi.id)
  scope: fileCsiDriverMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// IMAGE REGISTRY MANAGED IDENTITY
//

resource imageRegistryMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-
31' = {
  name: '${clusterName}-cp-image-registry-${randomSuffix}'
  location: resourceGroup().location
}

resource serviceManagedIdentityReaderOnImageRegistryMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
imageRegistryMi.id)
  scope: imageRegistryMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// CLOUD NETWORK CONFIG MANAGED IDENTITY
//

resource cloudNetworkConfigMi
'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-31' = {
  name: '${clusterName}-cp-cloud-network-config-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift Network Operator
var networkOperatorRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  'be7a6435-15ae-4171-8f30-4a343eff9e8f'
)

resource networkOperatorRoleSubnetAssignment
```

```
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, cloudNetworkConfigMi.id, networkOperatorRoleId,
subnet.id)
  scope: subnet
  properties: {
    principalId: cloudNetworkConfigMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: networkOperatorRoleId
  }
}

resource networkOperatorRoleVnetAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, cloudNetworkConfigMi.id, networkOperatorRoleId,
customerVnet.id)
  scope: customerVnet
  properties: {
    principalId: cloudNetworkConfigMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: networkOperatorRoleId
  }
}

resource serviceManagedIdentityReaderOnCloudNetworkMi
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id, readerRoleId,
cloudNetworkConfigMi.id)
  scope: cloudNetworkConfigMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: readerRoleId
  }
}

//
// D A T A P L A N E   I D E N T I T I E S
//

// Azure Red Hat OpenShift Federated Credential
// give the ability to perform OIDC federation to the service managed identity
// over the corresponding dataplane identities
var federatedCredentialsRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  'ef318e2a-8334-4a05-9e4a-295a196c6a6e'
)
```

```
resource dpDiskCsiDriverMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-
01-31' = {
  name: '${clusterName}-dp-disk-csi-driver-${randomSuffix}'
  location: resourceGroup().location
}

resource dpDiskCsiDriverMiFederatedCredentialsRoleAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, dpDiskCsiDriverMi.id, federatedCredentialsRoleId)
  scope: dpDiskCsiDriverMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: federatedCredentialsRoleId
  }
}

resource dpFileCsiDriverMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-
01-31' = {
  name: '${clusterName}-dp-file-csi-driver-${randomSuffix}'
  location: resourceGroup().location
}

resource dpFileCsiDriverMiFederatedCredentialsRoleAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, dpFileCsiDriverMi.id, federatedCredentialsRoleId)
  scope: dpFileCsiDriverMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: federatedCredentialsRoleId
  }
}

resource dpFileCsiDriverFileStorageOperatorRoleSubnetAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, dpFileCsiDriverMi.id, fileStorageOperatorRoleId,
subnet.id)
  scope: subnet
  properties: {
    principalId: dpFileCsiDriverMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: fileStorageOperatorRoleId
  }
}
```

```
resource dpFileCsiDriverFileStorageOperatorRoleNsgAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, dpFileCsiDriverMi.id, fileStorageOperatorRoleId,
customerNsg.id)
  scope: customerNsg
  properties: {
    principalId: dpFileCsiDriverMi.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: fileStorageOperatorRoleId
  }
}

resource dpImageRegistryMi 'Microsoft.ManagedIdentity/userAssignedIdentities@2023-
01-31' = {
  name: '${clusterName}-dp-image-registry-${randomSuffix}'
  location: resourceGroup().location
}

resource dpImageRegistryMiFederatedCredentialsRoleAssignment
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, dpImageRegistryMi.id, federatedCredentialsRoleId)
  scope: dpImageRegistryMi
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: federatedCredentialsRoleId
  }
}

//
// S E R V I C E   M A N A G E D   I D E N T I T Y
//

resource serviceManagedIdentity
'Microsoft.ManagedIdentity/userAssignedIdentities@2023-01-31' = {
  name: '${clusterName}-service-managed-identity-${randomSuffix}'
  location: resourceGroup().location
}

// Azure Red Hat OpenShift Hosted Control Planes Service Managed Identity
var hcpServiceManagedIdentityRoleId = subscriptionResourceId(
  'Microsoft.Authorization/roleDefinitions',
  'c0ff367d-66d8-445e-917c-583feb0ef0d4'
)
```

```
// grant service managed identity role to the service managed identity over the user
provided subnet
resource serviceManagedIdentityRoleAssignmentVnet
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id,
hcpServiceManagedIdentityRoleId, customerVnet.id)
  scope: customerVnet
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: hcpServiceManagedIdentityRoleId
  }
}

// grant service managed identity role to the service managed identity over the user
provided subnet
resource serviceManagedIdentityRoleAssignmentSubnet
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id,
hcpServiceManagedIdentityRoleId, subnet.id)
  scope: subnet
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: hcpServiceManagedIdentityRoleId
  }
}

// grant service managed identity role to the service managed identity over the user
provided NSG
resource serviceManagedIdentityRoleAssignmentNSG
'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  name: guid(resourceGroup().id, serviceManagedIdentity.id,
hcpServiceManagedIdentityRoleId, customerNsg.id)
  scope: customerNsg
  properties: {
    principalId: serviceManagedIdentity.properties.principalId
    principalType: 'ServicePrincipal'
    roleDefinitionId: hcpServiceManagedIdentityRoleId
  }
}

resource hcp 'Microsoft.RedHatOpenShift/hcpOpenShiftClusters@2024-06-10-preview' = {
  name: clusterName
  location: resourceGroup().location
  properties: {
```

```
version: {
  id: '4.19'
  channelGroup: 'stable'
}
dns: {}
network: {
  networkType: 'OVNKubernetes'
  podCidr: '10.128.0.0/14'
  serviceCidr: '172.30.0.0/16'
  machineCidr: '10.0.0.0/16'
  hostPrefix: 23
}
console: {}
etcd: {
  dataEncryption: {
    keyManagementMode: 'PlatformManaged'
  }
}
api: {
  visibility: 'Public'
}
clusterImageRegistry: {
  state: 'Enabled'
}
platform: {
  managedResourceGroup: managedResourceGroupName
  subnetId: subnet.id
  outboundType: 'LoadBalancer'
  networkSecurityGroupId: customerNsg.id
  operatorsAuthentication: {
    userAssignedIdentities: {
      controlPlaneOperators: {
        'cluster-api-azure': clusterApiAzureMi.id
        'control-plane': controlPlaneMi.id
        'cloud-controller-manager': cloudControllerManagerMi.id
        #disable-next-line prefer-unquoted-property-names
        'ingress': ingressMi.id
        'disk-csi-driver': diskCsiDriverMi.id
        'file-csi-driver': fileCsiDriverMi.id
        'image-registry': imageRegistryMi.id
        'cloud-network-config': cloudNetworkConfigMi.id
      }
      dataPlaneOperators: {
        'disk-csi-driver': dpDiskCsiDriverMi.id
        'file-csi-driver': dpFileCsiDriverMi.id
        'image-registry': dpImageRegistryMi.id
        }
        serviceManagedIdentity: serviceManagedIdentity.id
      }
    }
  }
}
identity: {
  type: 'UserAssigned'
  userAssignedIdentities: {
    '${serviceManagedIdentity.id}': {}
    '${clusterApiAzureMi.id}': {}
    '${controlPlaneMi.id}': {}
    '${cloudControllerManagerMi.id}': {}
    '${ingressMi.id}': {}
    '${diskCsiDriverMi.id}': {}
    '${fileCsiDriverMi.id}': {}
    '${imageRegistryMi.id}': {}
    '${cloudNetworkConfigMi.id}': {}
  }
}
dependsOn: [
  hcpClusterApiProviderRoleSubnetAssignment
  hcpControlPlaneOperatorVnetRoleAssignment
  hcpControlPlaneOperatorNsgRoleAssignment
  cloudControllerManagerRoleSubnetAssignment
  cloudControllerManagerRoleNsgAssignment
  ingressOperatorRoleSubnetAssignment
  fileStorageOperatorRoleSubnetAssignment
  fileStorageOperatorRoleNsgAssignment
  networkOperatorRoleSubnetAssignment
  networkOperatorRoleVnetAssignment
  dpDiskCsiDriverMiFederatedCredentialsRoleAssignment
  dpFileCsiDriverMiFederatedCredentialsRoleAssignment
  dpImageRegistryMiFederatedCredentialsRoleAssignment
  serviceManagedIdentityRoleAssignmentVnet
  serviceManagedIdentityRoleAssignmentSubnet
  serviceManagedIdentityRoleAssignmentNSG
  dpFileCsiDriverFileStorageOperatorRoleSubnetAssignment
  dpFileCsiDriverFileStorageOperatorRoleNsgAssignment
  serviceManagedIdentityReaderOnControlPlaneMi
  serviceManagedIdentityReaderOnCloudControllerManagerMi
  serviceManagedIdentityReaderOnIngressMi
  serviceManagedIdentityReaderOnDiskCsiDriverMi
  serviceManagedIdentityReaderOnFileCsiDriverMi
  serviceManagedIdentityReaderOnImageRegistryMi
  serviceManagedIdentityReaderOnCloudNetworkMi
```

```
    serviceManagedIdentityReaderOnClusterApiAzureMi
  ]
}

resource nodepool 'Microsoft.RedHatOpenShift/hcpOpenShiftClusters/nodePools@2024-06-10-preview' = {
  parent: hcp
  name: nodePoolName
  location: resourceGroup().location
  properties: {
    version: {
      id: '4.19.0'
      channelGroup: 'stable'
    }
    platform: {
      subnetId: hcp.properties.platform.subnetId
      vmSize: 'Standard_D8s_v3'
      osDisk: {
        sizeGiB: 64
        diskStorageAccountType: 'StandardSSD_LRS'
      }
    }
    replicas: 2
  }
}
```

Create the cluster using Bicep

This section describes how to use Bicep to create the cluster.

1. Set the following variables in the shell environment that you plan to execute the az commands.

   ```
   # location of your cluster
   LOCATION="<location>"

   # name of your subscription
   SUBSCRIPTION="<subscription-name>"

   # name of the resource group where you want to create your cluster
   CUSTOMER_RG_NAME="$USER-net-rg-03"

   # name of the nsg
   CUSTOMER_NSG="<nsg-name>"
   ```

```
# name of the vnet
CUSTOMER_VNET_NAME="<vnet-name>"

# name of the subnet within the vnet
CUSTOMER_VNET_SUBNET1="<subnet-name>"

# name of your cluster
CLUSTER_NAME="<cluster-name>"

# name of the managed resource group
MANAGED_RESOURCE_GROUP="$CLUSTER_NAME-rg-03"

# name of the node pool
NP_NAME="<node-pool-name>"

# optional variables for creating tags
SUBSCRIPTION_ID=$(az account show --query id --output tsv)
POLICY_DEFINITION="<policy-name>"
POLICY_ASSIGNMENT="${POLICY_DEFINITION}-${CLUSTER_NAME}"
```

2. Create a resource group to hold the cluster resource, node pool resource, and the cluster virtual network and identities.

```
$ az group create \
  --name "${CUSTOMER_RG_NAME}" \
  --subscription "${SUBSCRIPTION}" \
  --location "${LOCATION}"
```

3. (Optional) If desired, define tags that will be applied to each of the resources in your managed resource group.

   Follow these steps to use Azure Policy to tag the resources in your managed resource group. When you create the Azure Red Hat OpenShift with hosted control planes cluster, these tags will be applied to the managed resource group, and to each of the resources within it:

   1. Create the following JSON files to define the tags:
      - Rules file
      - Parameter definitions
      - Parameter values
   2. Create the policy definition.

```
$ az policy definition create -n $POLICY_DEFINITION \
  --mode All \
  --rules rules.json \
  --params param-defs.json
```

3. Create the policy assignment.

```
$ az policy assignment create -n $POLICY_ASSIGNMENT \
  --policy $POLICY_DEFINITION \
  --scope "/subscriptions/${SUBSCRIPTION_ID}" \
  --location $LOCATION \
  --mi-system-assigned \
  --role "Tag Contributor" \
  --identity-scope "/subscriptions/${SUBSCRIPTION_ID}" \
  --params param-values.json
```

4. Apply the Bicep file.

```
$ az deployment group create \
--name 'aro-hcp' \
--subscription "${SUBSCRIPTION}" \
--resource-group "${CUSTOMER_RG_NAME}" \
--template-file azuredeploy.bicep \
--parameters \
  customerNsgName="${CUSTOMER_NSG}" \
  customerVnetName="${CUSTOMER_VNET_NAME}" \
  customerVnetSubnetName="${CUSTOMER_VNET_SUBNET1}" \
  clusterName="${CLUSTER_NAME}" \
  managedResourceGroupName="${MANAGED_RESOURCE_GROUP}" \
  nodePoolName="${NP_NAME}"
```

5. (Optional) If desired, create a tag for the managed resource group.

```
$ az tag create \
--resource-id
/subscriptions/${SUBSCRIPTION_ID}/resourcegroups/${MANAGED_RESOURCE_GROUP} \
--tags <tag-key>=<tag-value>
```

Access the cluster

After creating an Azure Red Hat OpenShift with hosted control planes cluster, you can request a temporary administrative credential to access your cluster.

Requesting an administrative credential generates a new cluster-admin `kubeconfig` file. The `kubeconfig` file contains information about the cluster that connects a client to the correct cluster and API server. You can use the newly generated `kubeconfig` file to allow access to the cluster.

Administrative credentials expire after 24 hours. You can request multiple administrative credentials for a cluster. You can also revoke all administrative credentials for the cluster.

1. Request an administrative credential.

   ```
   $ az rest \
     --method POST \
     --uri "/subscriptions/${SUBSCRIPTION_ID}/resourceGroups/${CUSTOMER_RG_NAME}/providers/Microsoft.RedHatOpenShift/hcpOpenShiftClusters/${CLUSTER_NAME}/requestAdminCredential?api-version=2024-06-10-preview" \
     --verbose
   ```

   The admin credential is generated as an asynchronous Azure operation.

2. Monitor the status of the operation.

   1. In the response headers, find the URL listed in the `Azure-AsyncOperation` field.

   2. Use the `Azure-AsyncOperation` URL to get the status.

      ```
      $ az rest \
        --method GET \
        --uri "<url-from-Azure-AsyncOperation-field>"
      ```

3. Get the newly-generated `kubeconfig` content and save it as a `kubeconfig` file.

   1. In the response headers from the original operation, find the URL listed in the `Location` field.

   2. Use the `Location` field URL to get the `kubeconfig` file.

      ```
      $ az rest \
        --method GET \
        --uri "<url-from-Location-field>" \
        |  jq -r '.kubeconfig' > kubeconfig
      ```

4. Verify that you have the correct credentials:

```
$ export KUBECONFIG=kubeconfig
$ kubectl auth whoami --insecure-skip-tls-verify=true
ATTRIBUTE    VALUE
Username     system:customer-break-glass:<user-id>
Groups       [system:masters system:authenticated]
Extra: authentication.kubernetes.io/credential-id   <credential-id>
```

5. If desired, revoke all administrative credentials for the cluster:

```
$ az rest \
  --method POST \
  --uri
"/subscriptions/${SUBSCRIPTION_ID}/resourceGroups/${CUSTOMER_RG_NAME}/providers/
Microsoft.RedHatOpenShift/hcpOpenShiftClusters/${CLUSTER_NAME}/revokeCredentials
?api-version=2024-06-10-preview"
```

Clean up

When you are done using your Azure Red Hat OpenShift with hosted control planes cluster, you can delete it.

1. Delete the cluster (the `hcpOpenShiftClusters` resource).

   Deleting the `hcpOpenShiftClusters` resource deletes the cluster, its node pools, and the managed resource group.

```
$ az rest \
  --method DELETE \
  --uri
"/subscriptions/${SUBSCRIPTION_ID}/resourceGroups/${CUSTOMER_RG_NAME}/providers/
Microsoft.RedHatOpenShift/hcpOpenShiftClusters/${CLUSTER_NAME}?api-version=2024-
06-10-preview"
```

2. Verify that the cluster is deleted.

```
$ az rest \
  --method GET \
  --uri
"/subscriptions/${SUBSCRIPTION_ID}/resourceGroups/${CUSTOMER_RG_NAME}/providers/
Microsoft.RedHatOpenShift/hcpOpenShiftClusters/${CLUSTER_NAME}?api-version=2024-
06-10-preview"
```

3. Delete the resource group.

```
$ az group delete --name ${CUSTOMER_RG_NAME}
```