ACM install platform none clusters

Installing a hybrid cluster **1**
- Managed cluster environment configuration 1
- Create a Host inventory 3
- Create new VMs 5
- Advance network configuration 14
- Download the discovery ISO 17
- Upload the discovery ISO to vsphere 18
- Boot the VMs with the discovery ISO 19
- Create the cluster 23

TODO **26**

Is it supported to have mixed environment (VMware+Baremetal) setup for RHOCP 4.x cluster?

The next link could be outdated because the support is supposed to be available in 4.14, according to the Epic. But the docs still state that platform:none is only supported on SNO.
https://access.redhat.com/solutions/7008469

There is a documentation bug opened for that
https://issues.redhat.com/browse/OCPBUGS-29306

The final response is that YES, it is supported

Installing a hybrid cluster

Create 2 **VMware Cloud Public Cloud Open Environment**. One will be used to deploy an IPI OCP 4.14 cluster hub cluster, the other will be used to create the managed cluster.

Deploy an IPI cluster on vsphere, on demo.redhat.com using the **VMware Cloud Public Cloud Open Environment**
Deploy the ACM operator

Managed cluster environment configuration

The goal here is to assign the IP addresses for api and *.apps DNS records to the bastion host, and install haproxy in the bastion to redirect requests to the OCP 4 managed cluster.

On the Environment for the managed cluster add a new network interface to the bastion host.

Configure the network interface to have the IP addresses assigned to the api and *.apps DNS records

API DNS api.rscp4.dynamic.opentlc.com points to NAT IP to 192.168.188.201
Wildcard DNS *.apps.rscp4.dynamic.opentlc.com points to NAT IP to 192.168.188.202

```
Unset
$ sudo nmcli con down "Wired connection 1"
$ sudo nmcli con mod "Wired connection 1" ipv4.method manual ipv4.addr
"192.168.188.201, 192.168.188.202"
$ sudo nmcli con up "Wired connection 1"
$ ip -4 a
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
       inet 192.168.188.201/32 scope global noprefixroute eth1
       valid_lft forever preferred_lft forever
       inet 192.168.188.202/32 scope global noprefixroute eth1
```

Clone the following git repository:

```
Unset
$ git clone https://github.com/naps-product-sa/vmc-openshift-install-lab.git
$ cd vmc-openshift-install-lab
```

Install ansible:

```
Unset
$ sudo python3 -m pip install ansible
```

Update the vars file. Add some properties to ansible/vars.yml. We're interested in this section:

```
Unset
###### YOUR PROPERTIES HERE ######
api_ip: 192.168.188.201
apps_ip: 192.168.188.202
guid: 'rscp4'
ocp_version: 4.14.13
ocp4_pull_secret: '{"auths":{"cloud.opensh...
##################################
```

The ansible playbook is simplified so that only the tasks related to installing and enabling haproxy, and the task that downloads the oc client are run.

Run the playbook

```
Unset
$ ansible-playbook -vvv ansible/main.yml
```

Verify that haproxy is listening on the expected ports

```
Unset
$ ss -tlnp
State         Recv-Q        Send-Q                Local Address:Port
       Peer Address:Port           Process
...
LISTEN        0             128                   192.168.188.202:443
       0.0.0.0:*
LISTEN        0             128                   192.168.188.201:22623
       0.0.0.0:*
LISTEN        0             128                   192.168.188.201:6443
       0.0.0.0:*
LISTEN        0             128                   192.168.188.202:80
       0.0.0.0:*
```

Create a Host inventory

https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.9/html/clusters/cluster_mce_overview#create-host-inventory-console-steps

Before creating the first host inventory, the host inventory settings must be set.
From the All cluster web console -> Host inventory -> Configure host inventory settings
This creates the following pods in the project **multicluster-engine**

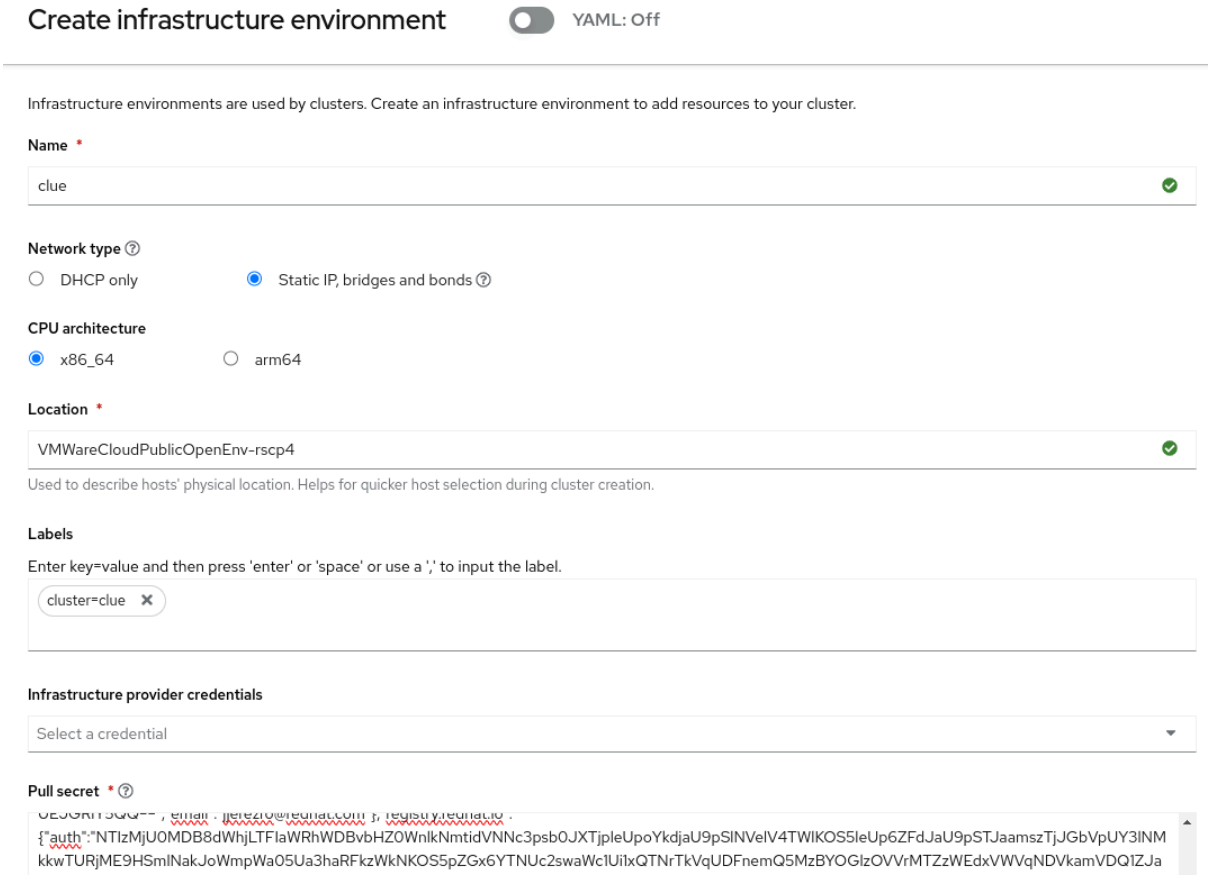
```
Unset
$ oc get pods -n multicluster-engine
agentinstalladmission-6bdf8c65d4-8pszw         1/1     Running   0     8m15s
agentinstalladmission-6bdf8c65d4-ltjnp         1/1     Running   0     8m15s
assisted-image-service-0                       1/1     Running   0     8m15s
```

```
assisted-service-5c84fc8bf9-rj5dj              2/2    Running  0    8m15s
```

And the following PVCs

```
Unset
$ oc get pvc -n multicluster-engine
NAME                         STATUS  VOLUME                    CAPACITY  ACCESS
MODES  STORAGECLASS  AGE
assisted-service                   Bound
pvc-c3802bb2-bb29-4d34-b90d-6e318cf6a660  50Gi        RWO           thin-csi
       9m53s
image-service-data-assisted-image-service-0  Bound
pvc-dd039e55-d750-4d62-936a-b7447e26750e  30Gi        RWO           thin-csi
       9m52s
postgres                   Bound  pvc-5de1ba28-41e1-4c7d-bad1-9d55cd2a5f9a
10Gi   RWO           thin-csi      9m53s
```

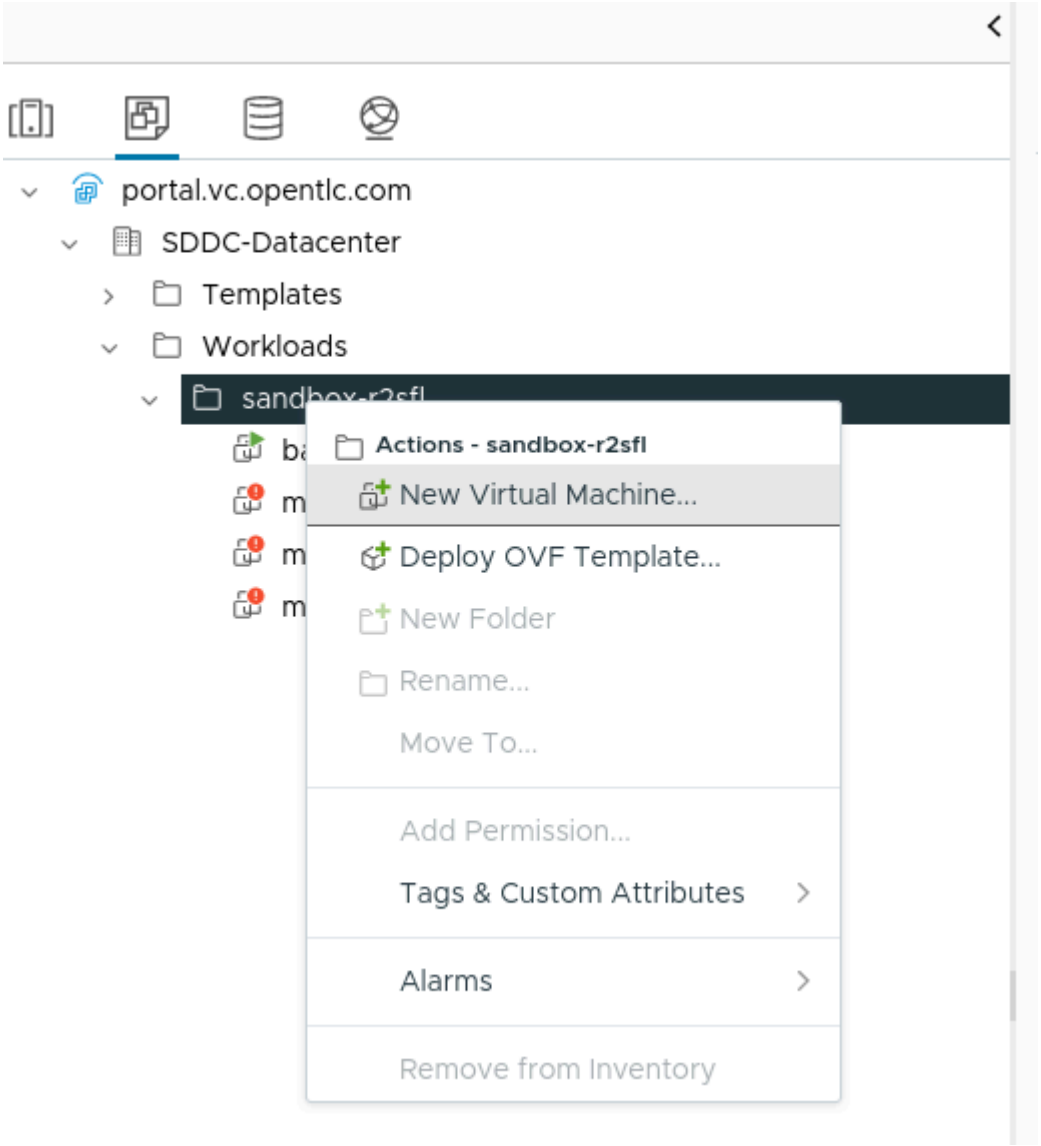
Create the infrastructure environment by clicking on the blue button on the middle of the Host inventory page.

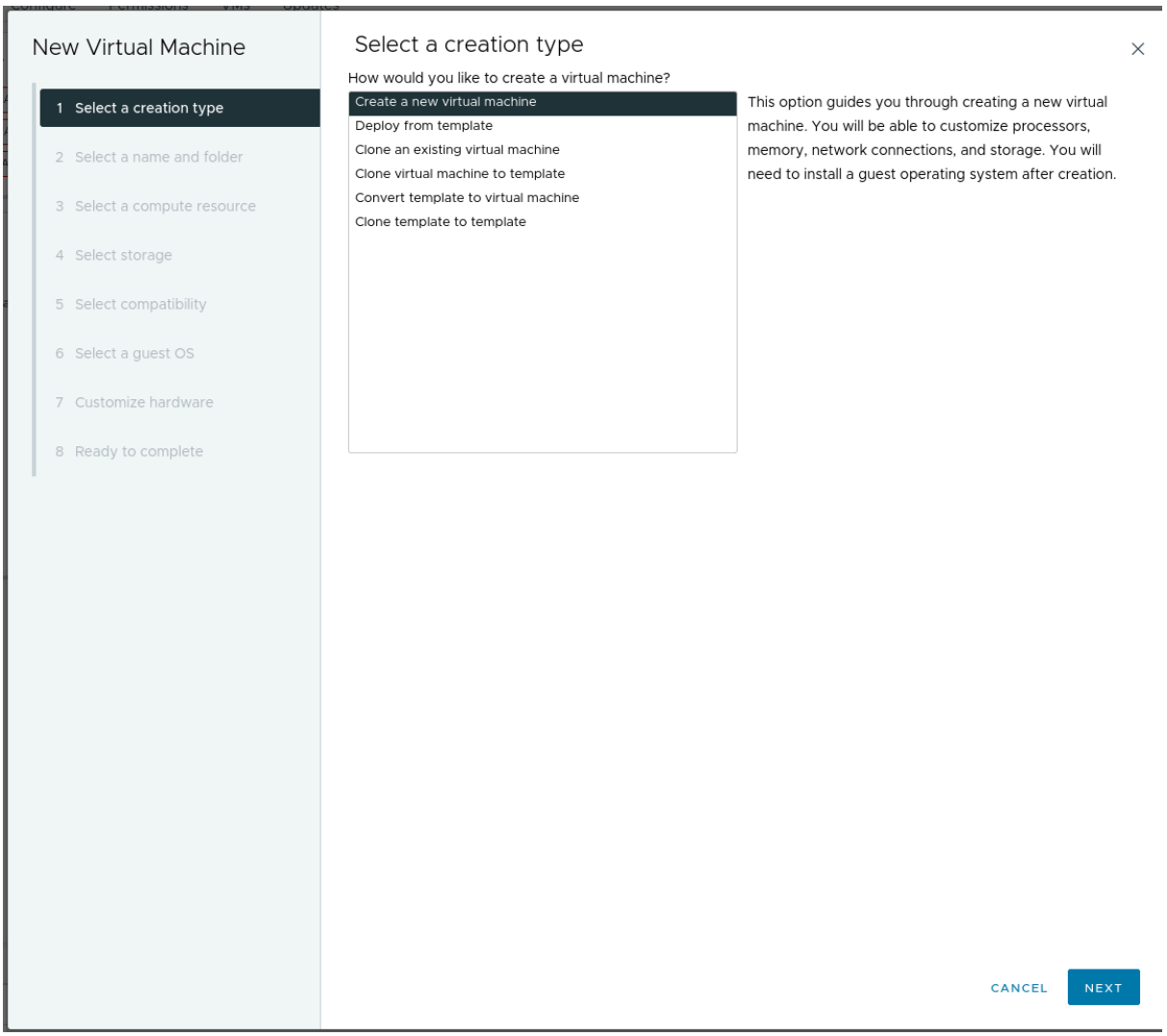


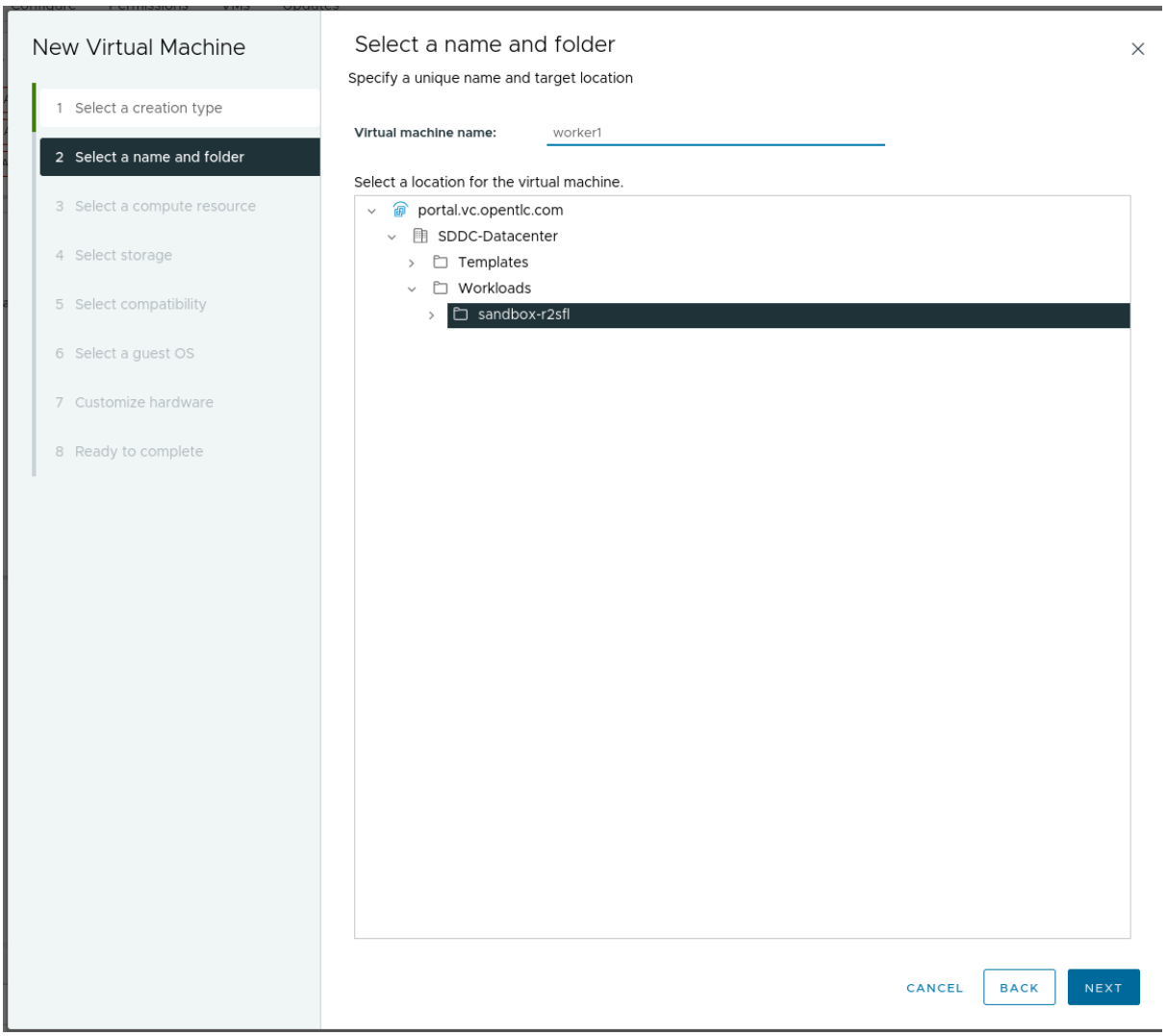
Create new VMs

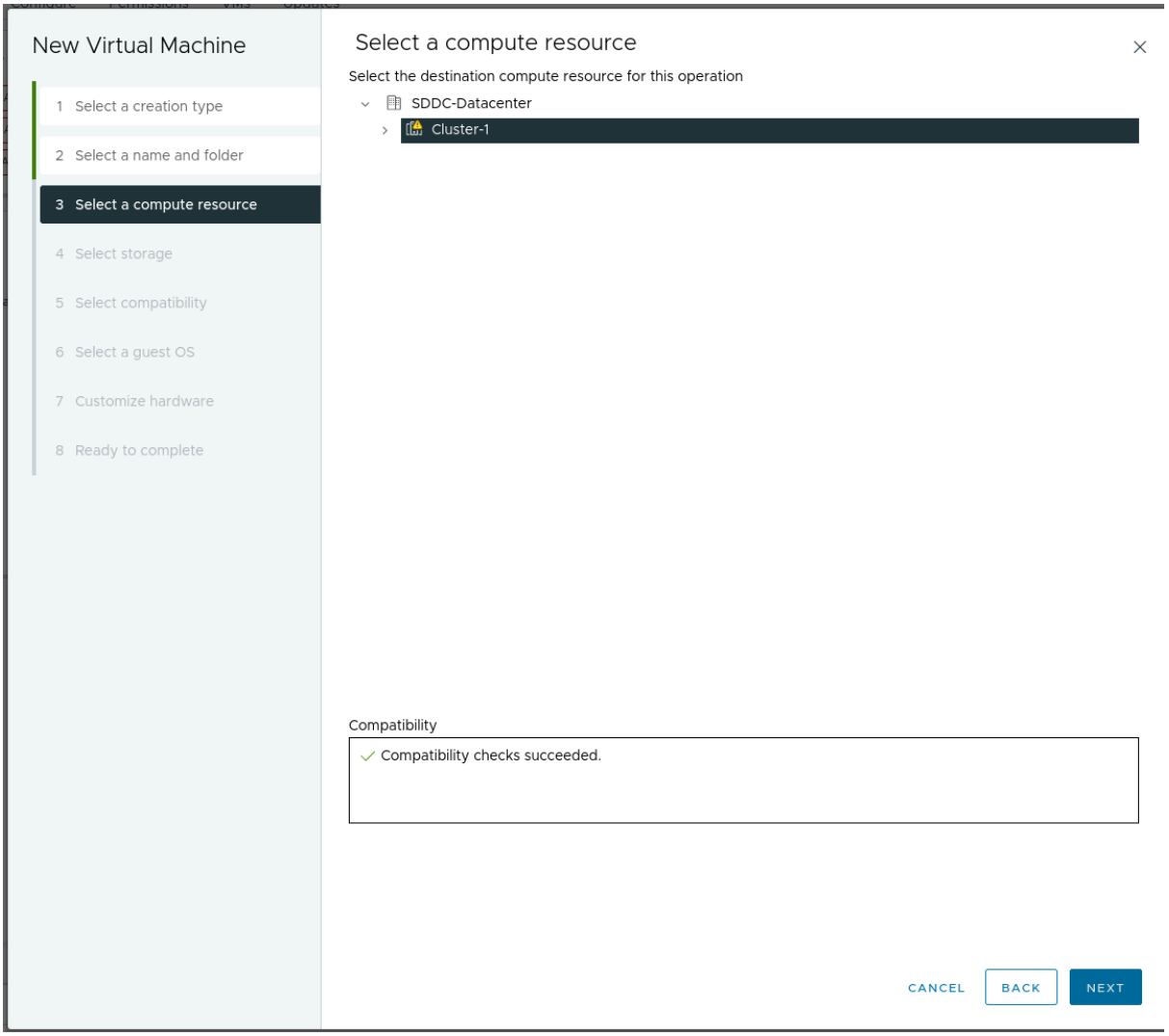
Create new virtual machines in vsphere.
Most of the information is left as default, exceptions are:

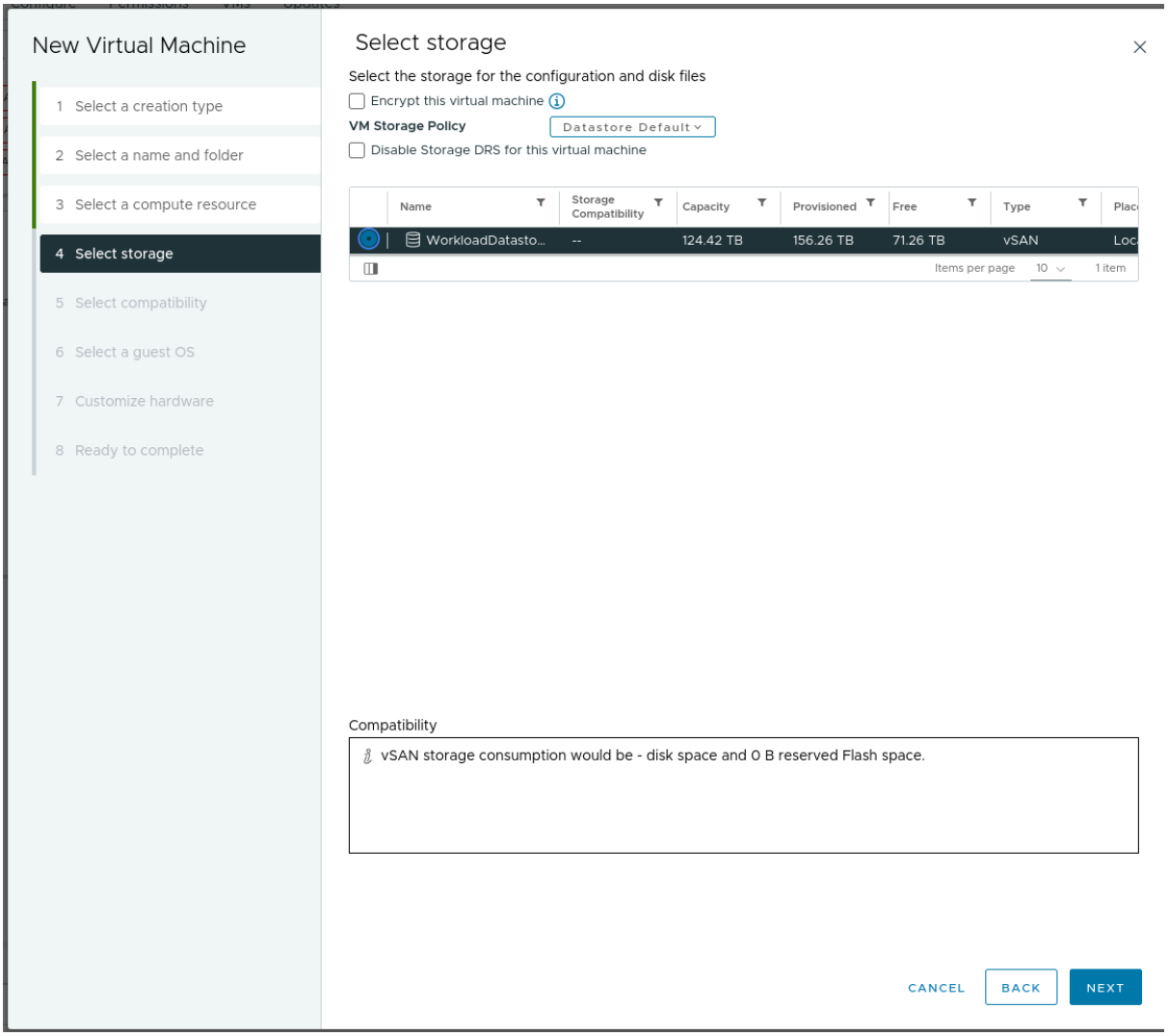
- The guest OS is changed to Linux - RHEL 8
- Hardware resources are updated to 4 CPU; 16GB RAM; 120GB disk
- In advance parameters tab, the diskEnableUUID=TRUE is added

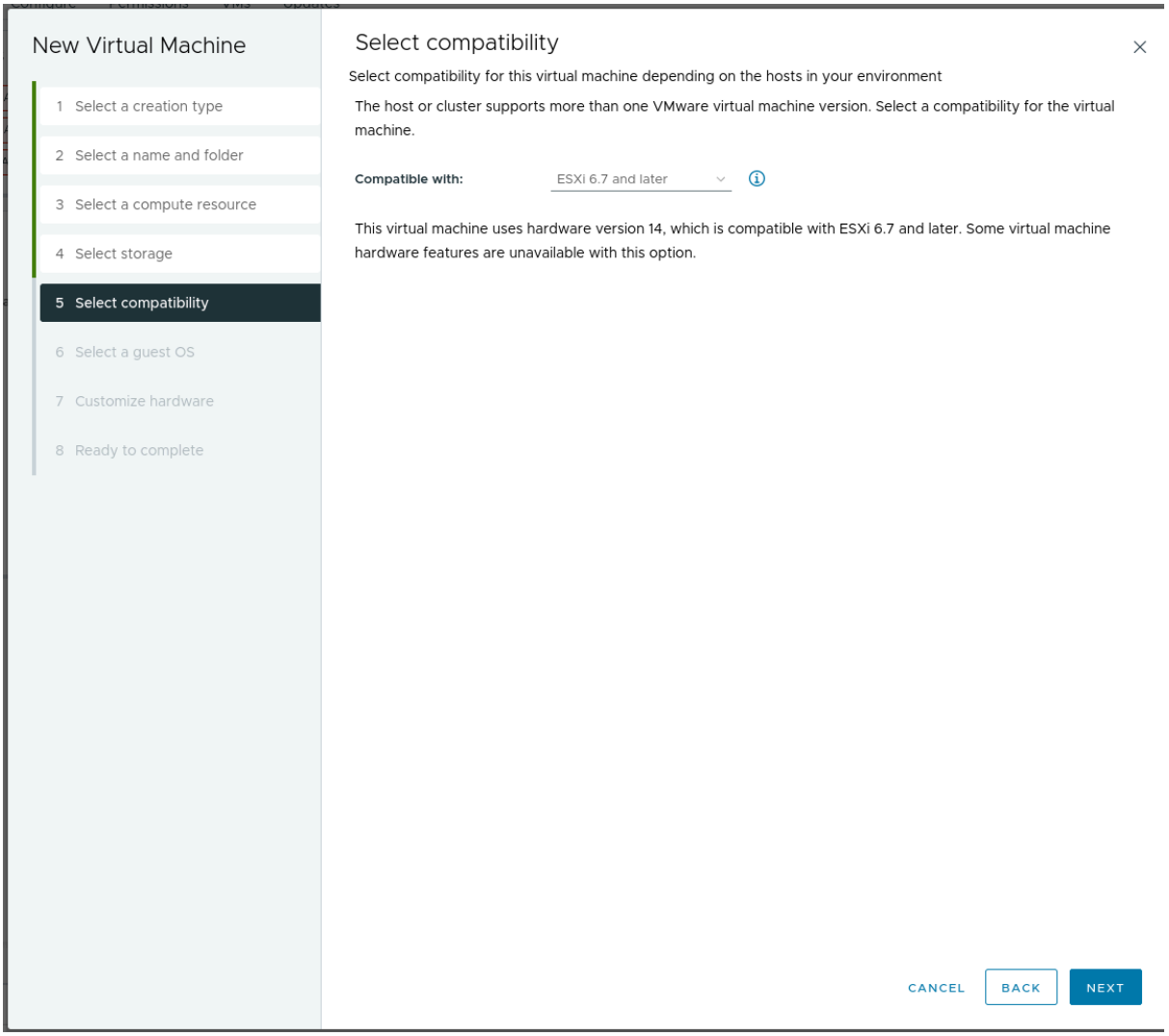


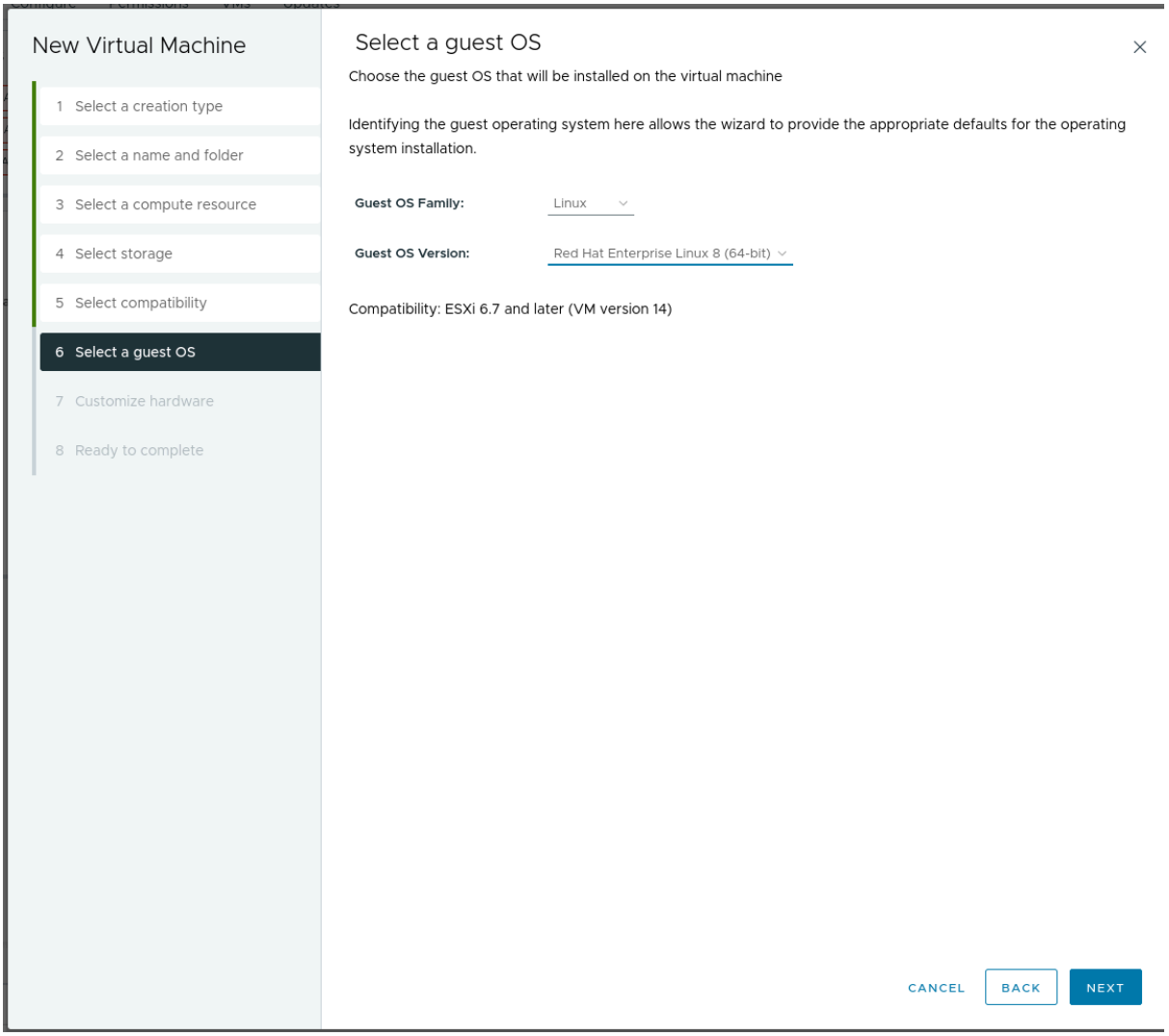

New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Select a creation type
How would you like to create a virtual machine?
Create a new virtual machine
Deploy from template
Clone an existing virtual machine
Clone virtual machine to template
Convert template to virtual machine
Clone template to template
This option guides you through creating a new virtual machine. You will be able to customize processors, memory, network connections, and storage. You will need to install a guest operating system after creation.
CANCEL
NEXT

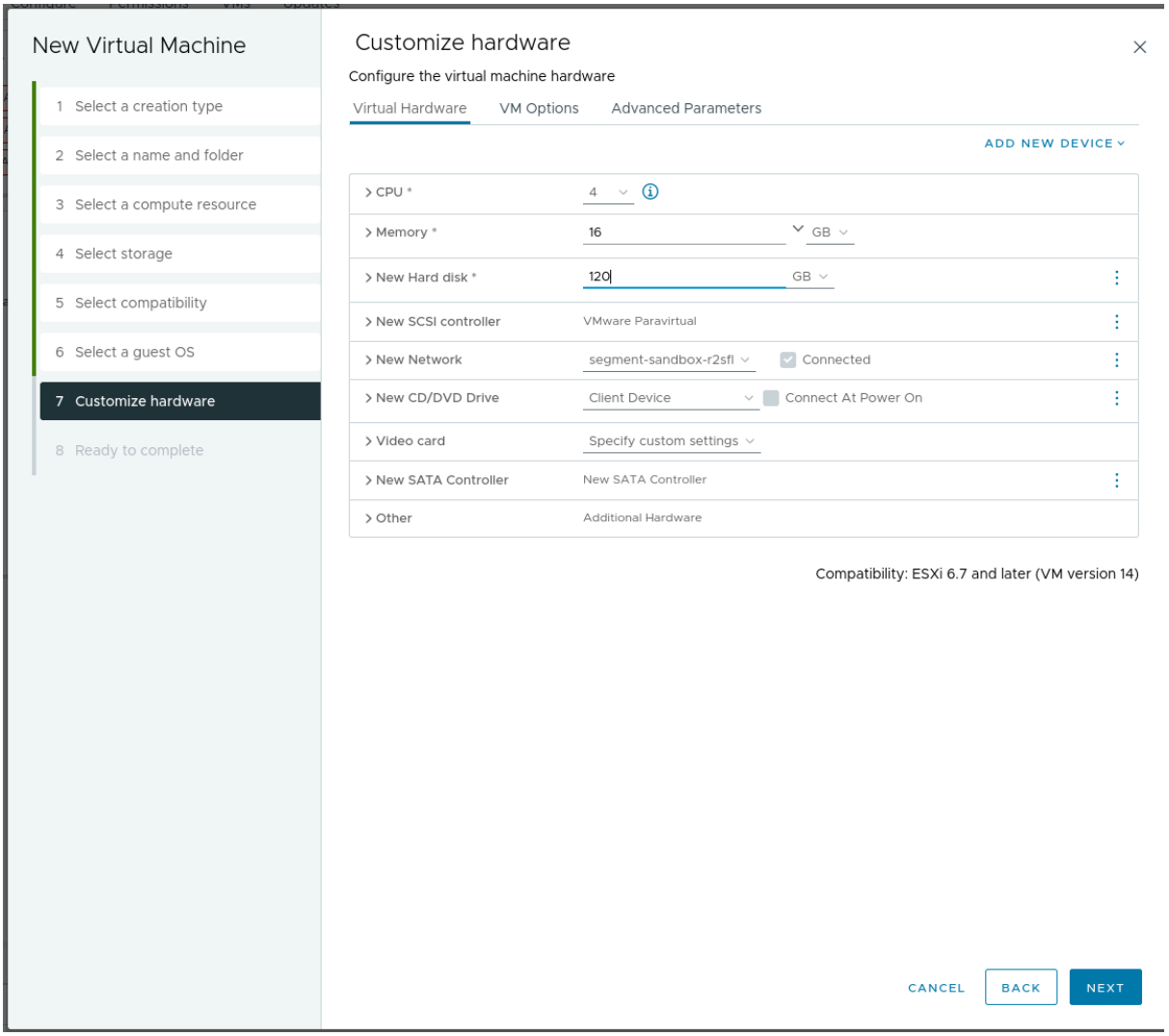

New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Select a name and folder
Specify a unique name and target location
Virtual machine name:
worker1
Select a location for the virtual machine.
portal.vc.opentlc.com
SDDC-Datacenter
Templates
Workloads
sandbox-r2sfl
CANCEL
BACK
NEXT

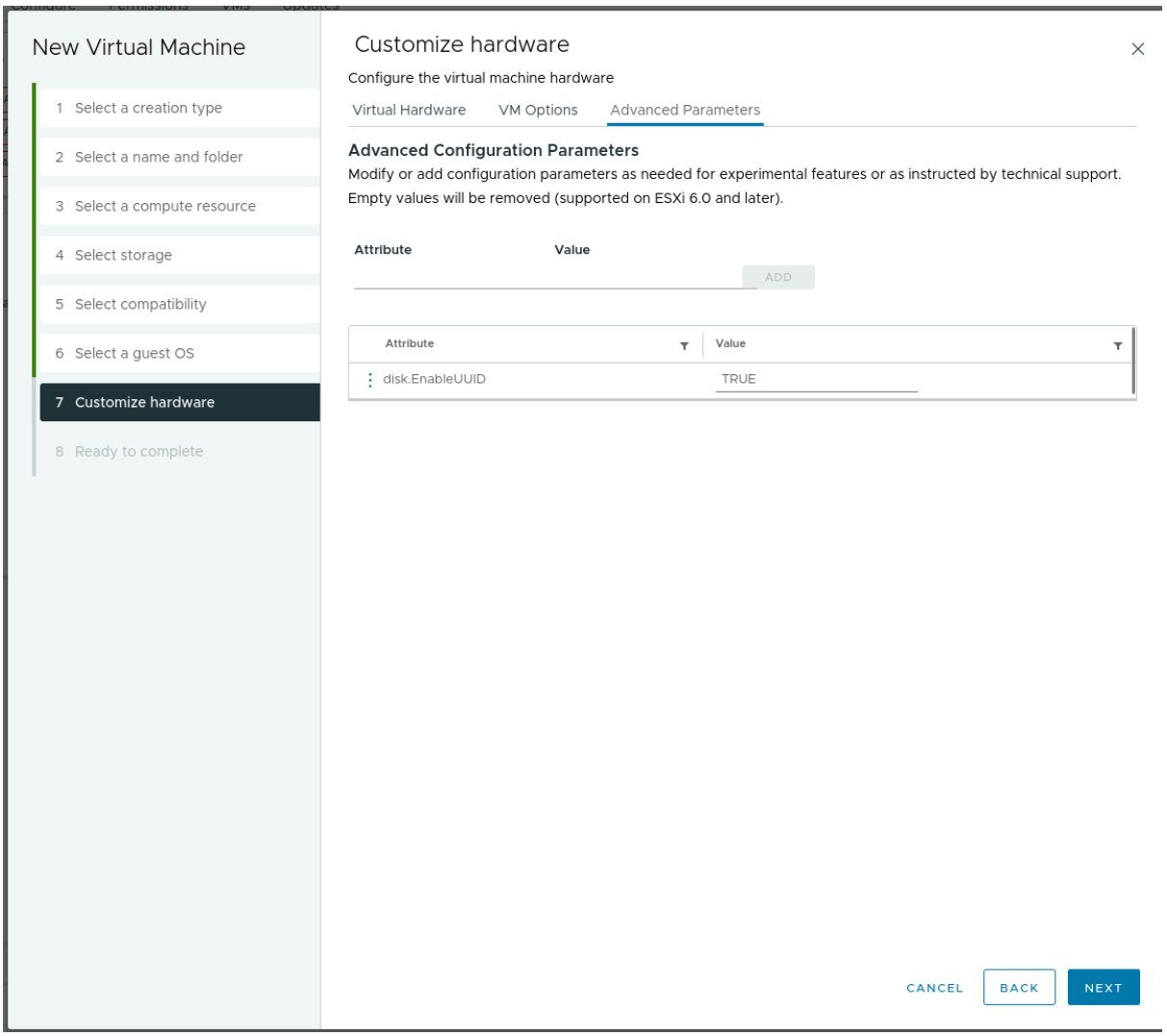

New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Select a compute resource
Select the destination compute resource for this operation
SDDC-Datacenter
Cluster-1
Compatibility
Compatibility checks succeeded.
CANCEL
BACK
NEXT

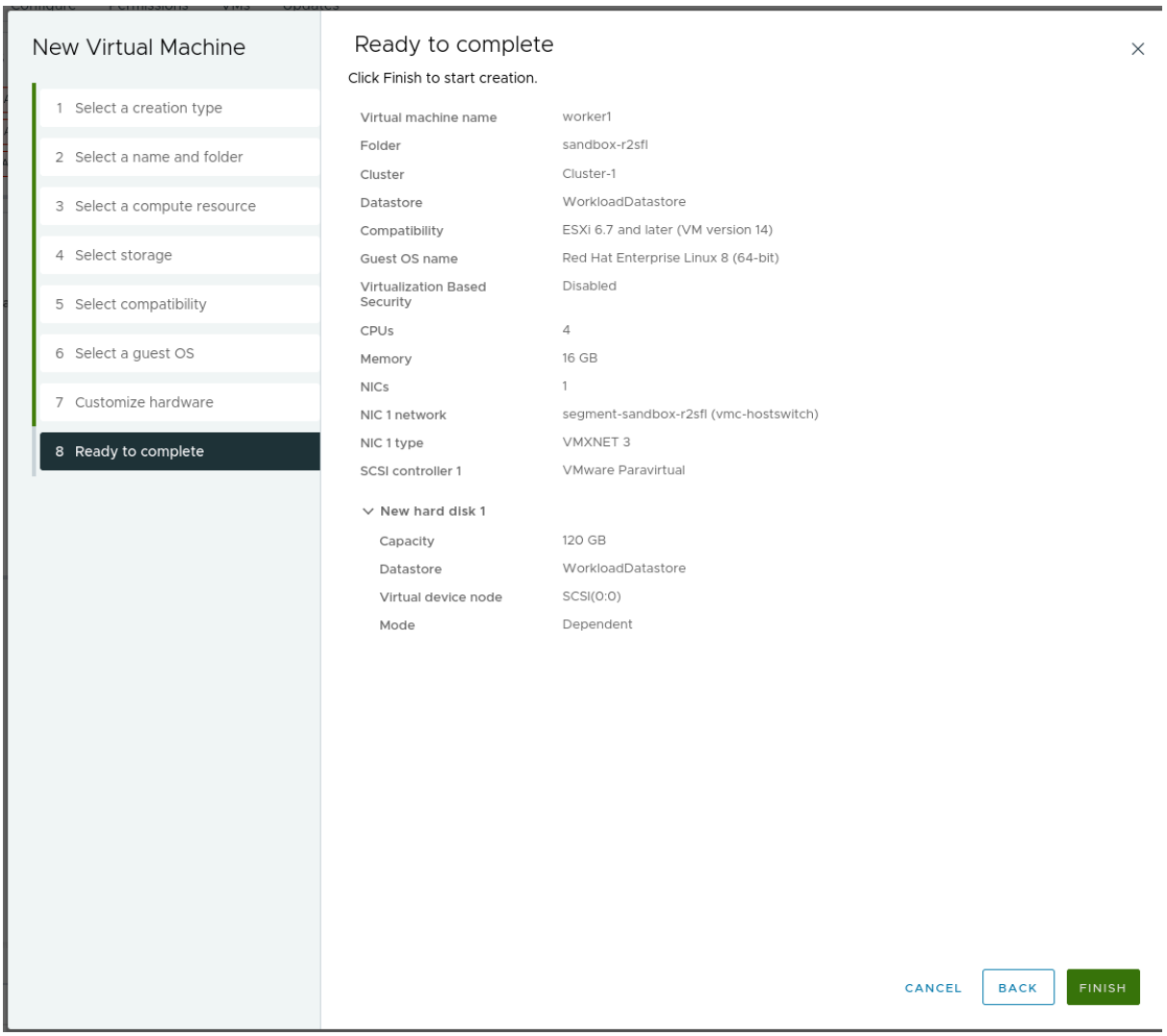

New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Select storage
Select the storage for the configuration and disk files
Encrypt this virtual machine
VM Storage Policy
Datastore Default
Disable Storage DRS for this virtual machine
Name
Storage Compatibility
Capacity
Provisioned
Free
Type
WorkloadDatasto...
--
124.42 TB
156.26 TB
71.26 TB
vSAN
Items per page
10
1 item
Compatibility
vSAN storage consumption would be - disk space and 0 B reserved Flash space.
CANCEL
BACK
NEXT


New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Select compatibility
Select compatibility for this virtual machine depending on the hosts in your environment
The host or cluster supports more than one VMware virtual machine version. Select a compatibility for the virtual machine.
Compatible with:
ESXi 6.7 and later
This virtual machine uses hardware version 14, which is compatible with ESXi 6.7 and later. Some virtual machine hardware features are unavailable with this option.
CANCEL
BACK
NEXT


New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Select a guest OS
Choose the guest OS that will be installed on the virtual machine
Identifying the guest operating system here allows the wizard to provide the appropriate defaults for the operating system installation.
Guest OS Family:
Linux
Guest OS Version:
Red Hat Enterprise Linux 8 (64-bit)
Compatibility: ESXi 6.7 and later (VM version 14)
CANCEL
BACK
NEXT


New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Customize hardware
Configure the virtual machine hardware
Virtual Hardware
VM Options
Advanced Parameters
ADD NEW DEVICE
CPU *
4
Memory *
16
GB
New Hard disk *
120
GB
New SCSI controller
VMware Paravirtual
New Network
segment-sandbox-r2sfl
Connected
New CD/DVD Drive
Client Device
Connect At Power On
Video card
Specify custom settings
New SATA Controller
New SATA Controller
Other
Additional Hardware
Compatibility: ESXi 6.7 and later (VM version 14)
CANCEL
BACK
NEXT


New Virtual Machine
1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete
Customize hardware
Configure the virtual machine hardware
Virtual Hardware
VM Options
Advanced Parameters
Advanced Configuration Parameters
Modify or add configuration parameters as needed for experimental features or as instructed by technical support.
Empty values will be removed (supported on ESXi 6.0 and later).
Attribute
Value
ADD
Attribute
Value
disk.EnableUUID
TRUE
CANCEL
BACK
NEXT



Advance network configuration

If the hosts need to get advanced network configuration, like is the case in the demo.redhat.com environment, where DHCP is available but the IP is unpredictable, so static IPs must be assigned to the nodes so they can be used in the DNS and LB configurations, then add a nmstateconfig object for each of the nodes that are going to be added to the cluster.
https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.9/html/clusters/cluster_mce_overview#cim-network-steps

https://access.redhat.com/documentation/es-es/openshift_container_platform/4.12/html/installing/installing-an-on-premise-cluster-with-the-agent-based-installer#sample-ztp-custom-resources_installing-with-agent-based-installer

The nmstateconfig definitions are similar to the following, one for each node in the cluster. The MACs are obtained from the vsphere VMs, the IPs are obtained from the haproxy configuration, the label from the infraenv object in ACM, the DNS IP is obtained from the /etc/resolv.conf file, the default route IP is obtained from the command “ip route”

Unset

```
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
 name: master-1
 namespace: hybridcluster
 labels:
       infraenvs.agent-install.openshift.io: hybridcluster
spec:
 config:
  interfaces:
    - name: eth0
      type: ethernet
      state: up
      mac-address: 00:50:56:a2:5d:4d
      ipv4:
       enabled: true
       address:
        - ip: 192.168.64.101
          prefix-length: 24
       dhcp: false
  dns-resolver:
   config:
    server:
     - 192.168.64.10
  routes:
   config:
    - destination: 0.0.0.0/0
      next-hop-address: 192.168.64.1
      next-hop-interface: eth0
      table-id: 254
 interfaces:
   - name: "eth0"
     macAddress: 00:50:56:a2:5d:4d
---
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
  name: master-2
  namespace: hybridcluster
  labels:
       infraenvs.agent-install.openshift.io: hybridcluster
spec:
  config:
       interfaces:
       - name: eth0
       type: ethernet
       state: up
       mac-address: 00:50:56:a2:e9:43
```

```
      ipv4:
      enabled: true
      address:
            - ip: 192.168.64.102
            prefix-length: 24
      dhcp: false
      dns-resolver:
      config:
      server:
      - 192.168.64.10
      routes:
      config:
      - destination: 0.0.0.0/0
      next-hop-address: 192.168.64.1
      next-hop-interface: eth0
      table-id: 254
  interfaces:
      - name: "eth0"
      macAddress: 00:50:56:a2:e9:43
---
apiVersion: agent-install.openshift.io/v1beta1
kind: NMStateConfig
metadata:
  name: master-3
  namespace: hybridcluster
  labels:
      infraenvs.agent-install.openshift.io: hybridcluster
spec:
  config:
      interfaces:
      - name: eth0
      type: ethernet
      state: up
      mac-address: 00:50:56:a2:4a:1d
      ipv4:
      enabled: true
      address:
            - ip: 192.168.64.103
            prefix-length: 24
      dhcp: false
      dns-resolver:
      config:
      server:
      - 192.168.64.10
      routes:
      config:
      - destination: 0.0.0.0/0
      next-hop-address: 192.168.64.1
```

```
      next-hop-interface: eth0
      table-id: 254
  interfaces:
      - name: "eth0"
      macAddress: 00:50:56:a2:4a:1d
```

Apply the objects to the cluster:

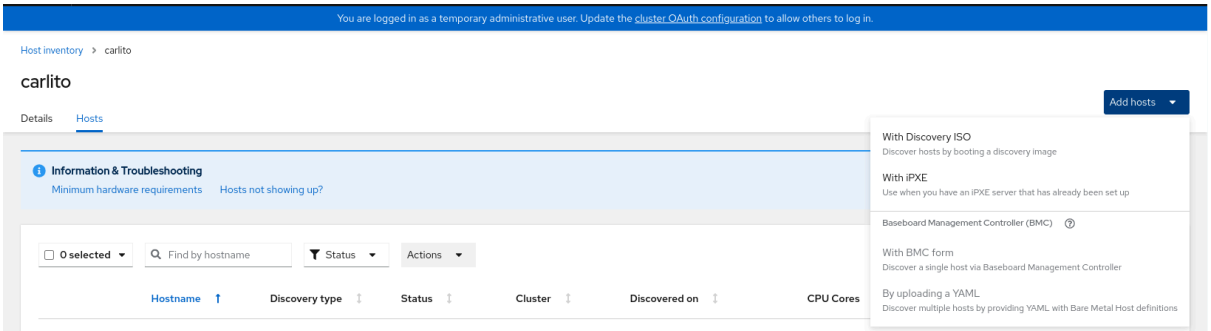
```
Unset
$ oc apply -f nmstate-master-1.yaml
$ oc apply -f nmstate-master-2.yaml
$ oc apply -f nmstate-master-3.yaml

$ oc get nmstateconfig -n hybridcluster
NAME          AGE
master-1   6m25s
master-2   5m40s
master-3   5m35s
```

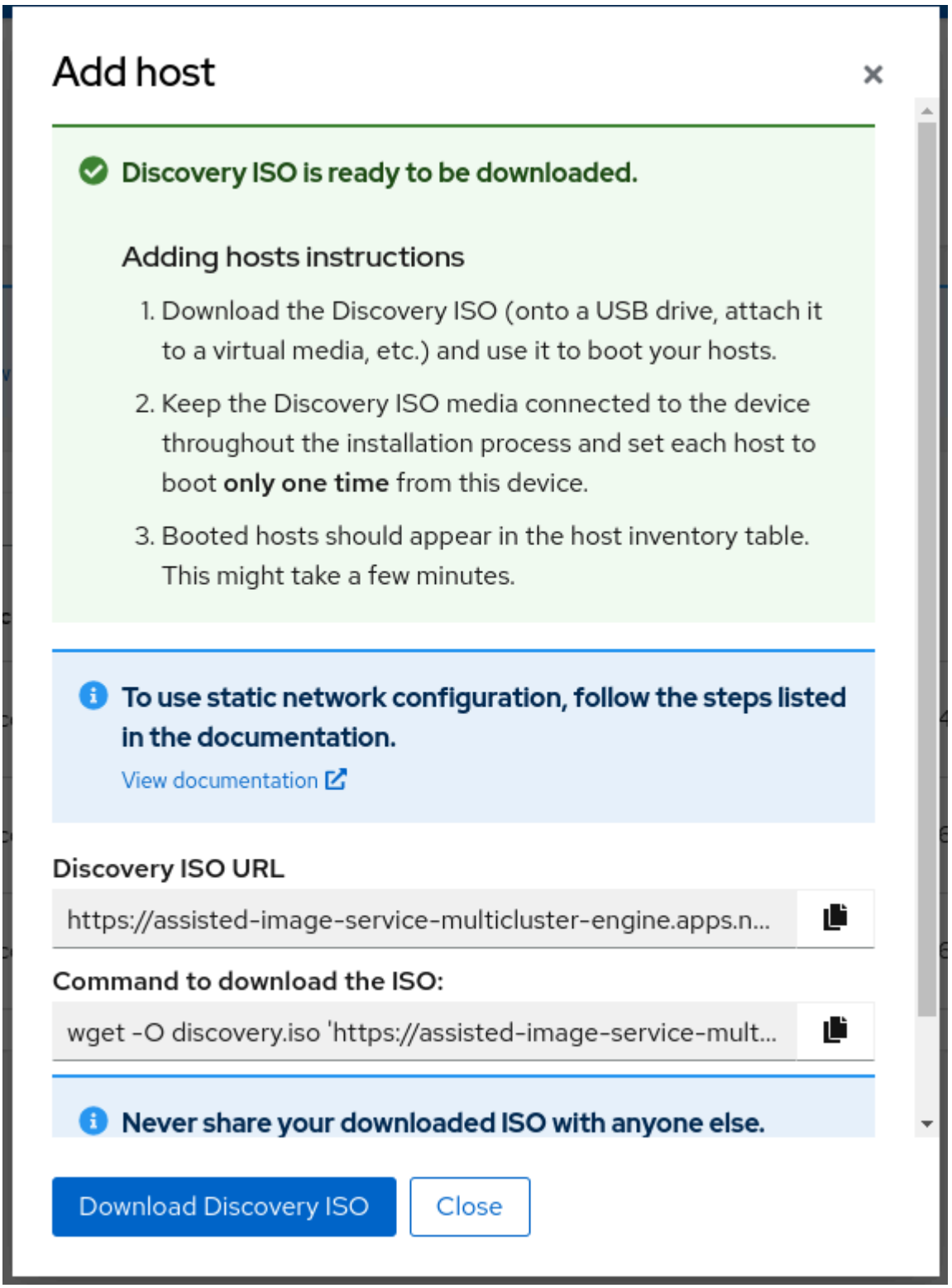
After applying the nmstateconfig objects, a new discovery ISO Image is created

Download the discovery ISO

From Host inventory -> Add hosts -> With Discovery ISO
D

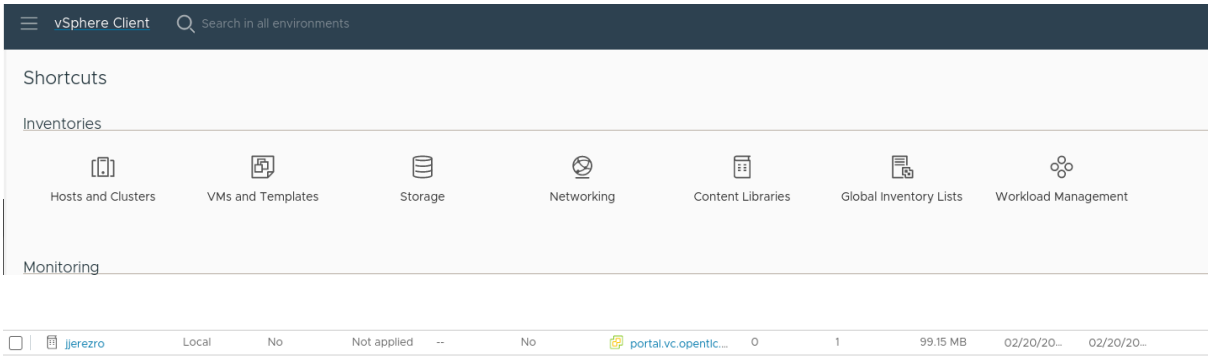


Download Discovery ISO

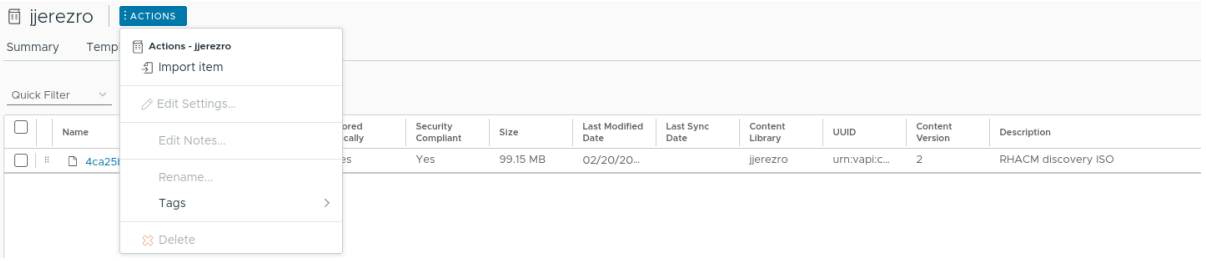


Upload the discovery ISO to vsphere

Create a content library by going to Content Libraries -> Create

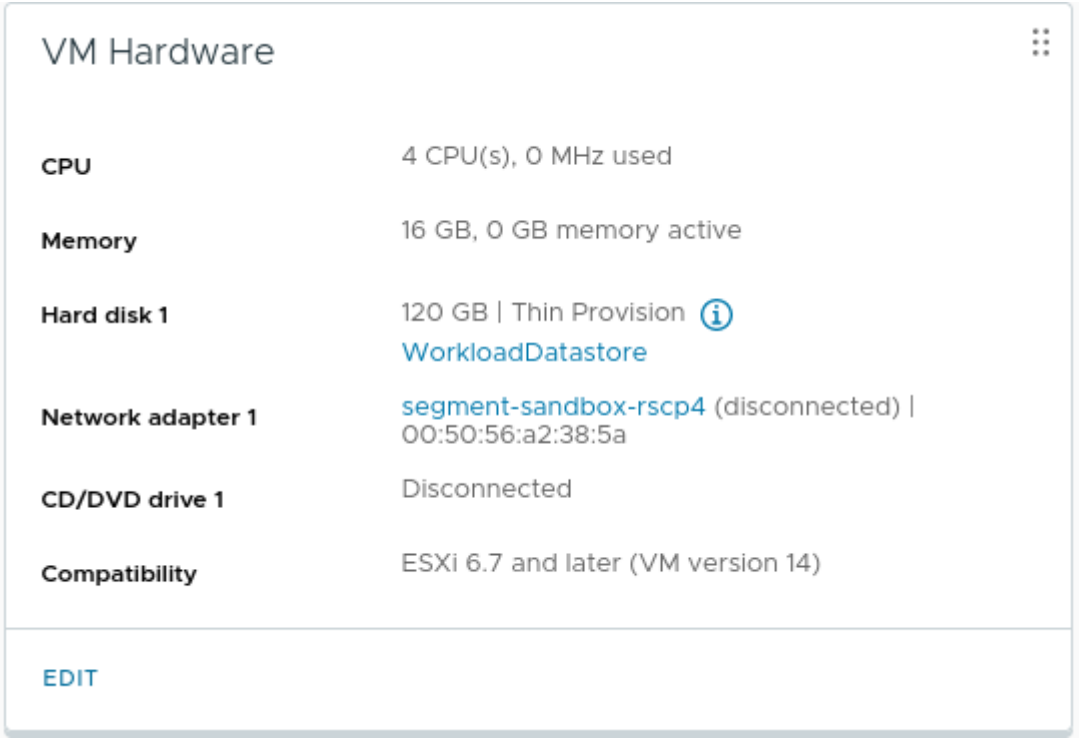


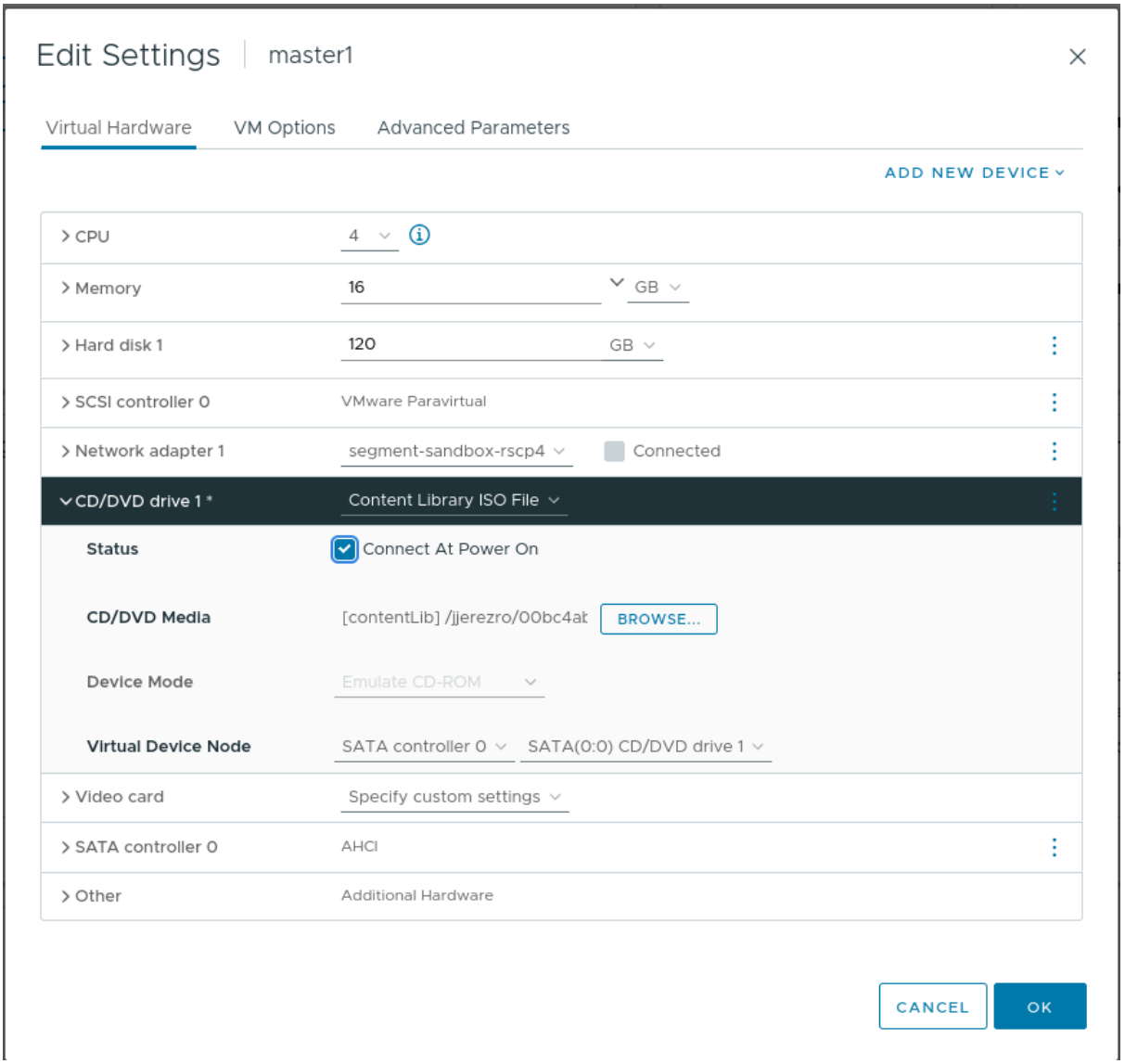
Once created, go to the content library -> Actions -> Import Item



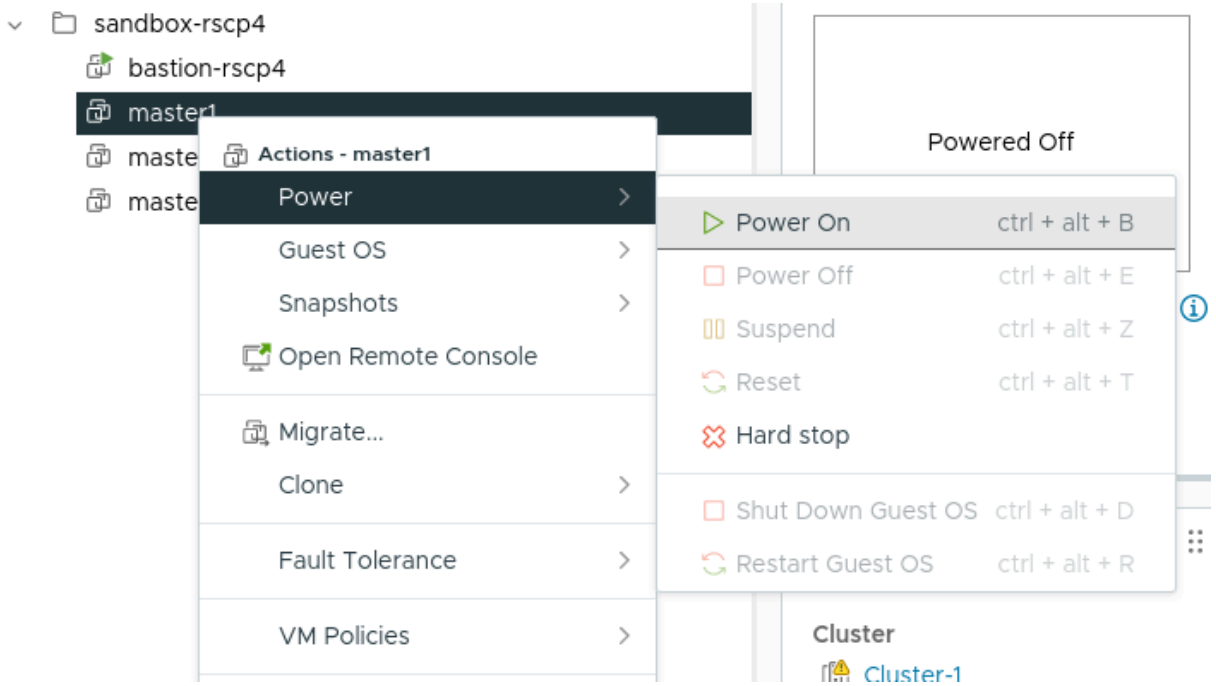
Boot the VMs with the discovery ISO

Edit the VM hardware and attach the discovery ISO from the content library to the CD.
Enable the option connect at Power On so that the VM boots from the ISO.
Repeat for all VMs

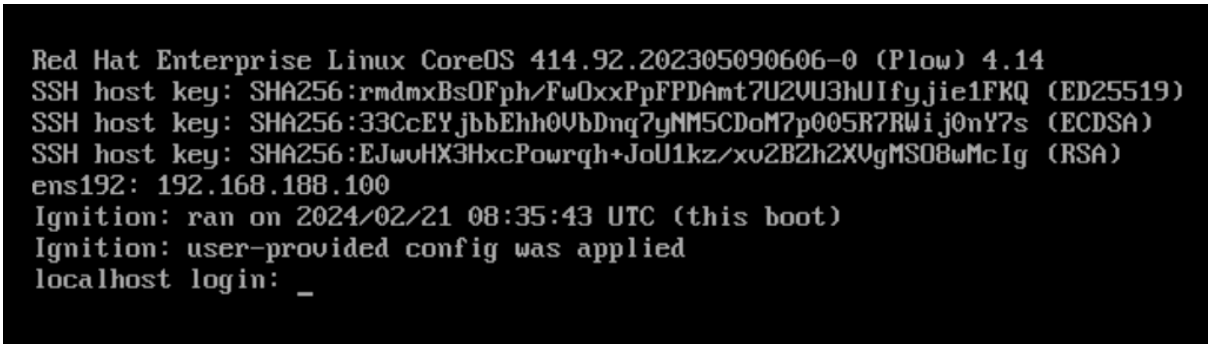




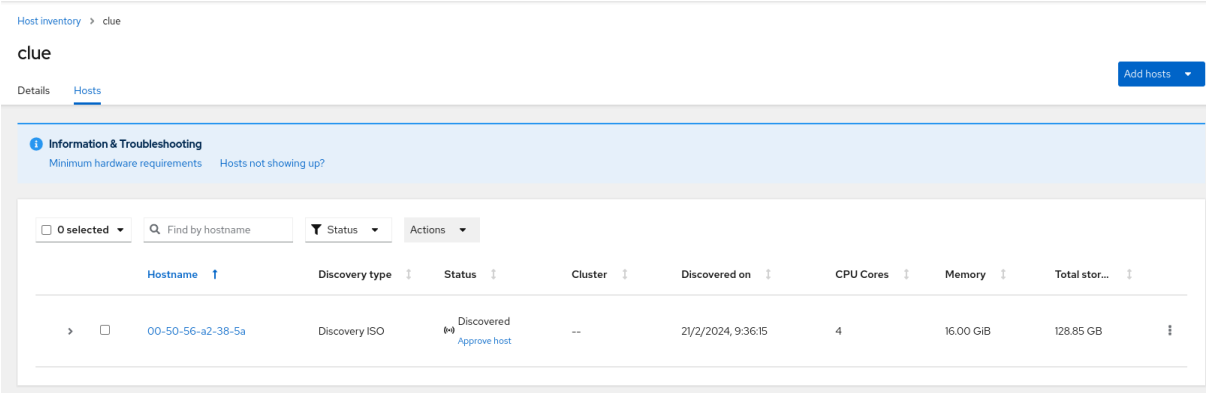
Boot the VMs, one by one



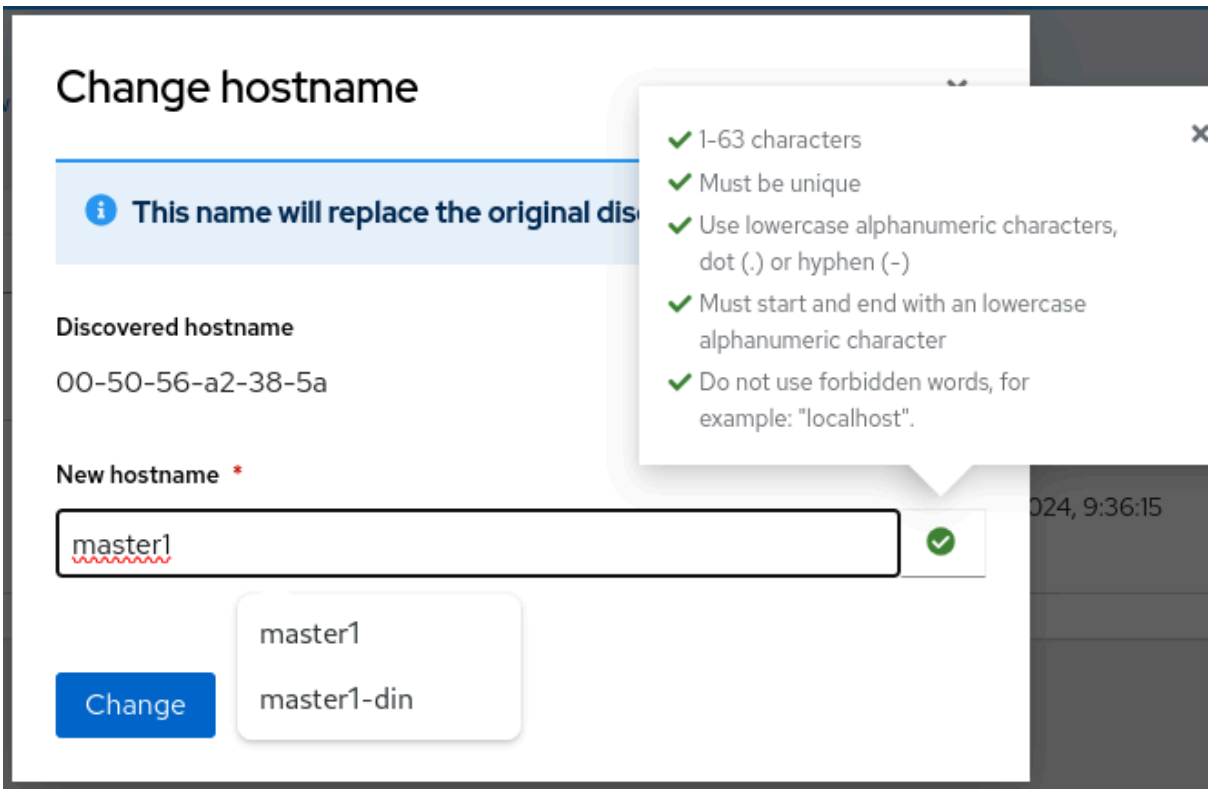
After a couple minutes the VM should be fully booted



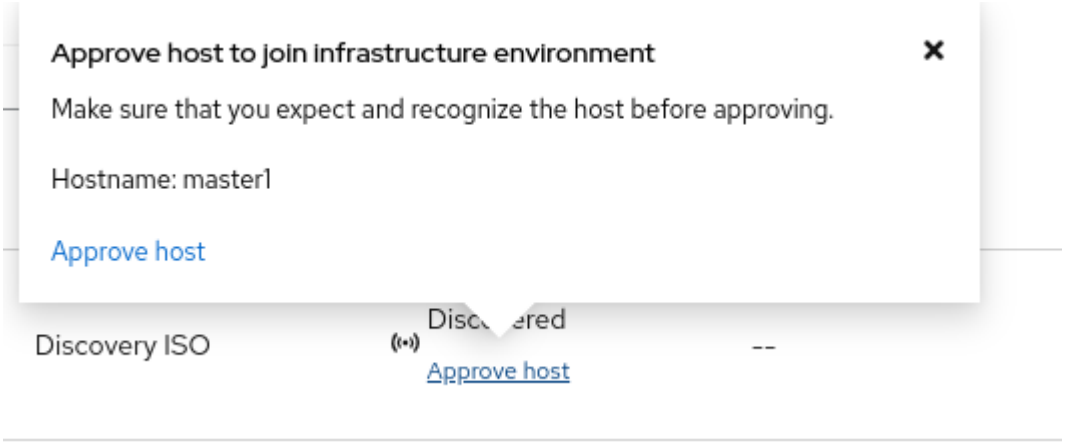
And the new host should appear in the Hosts list of the ACM inventory:



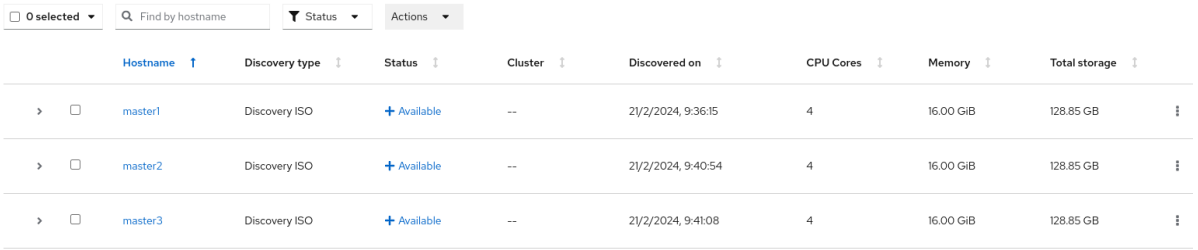
If the name cannot be resolved by DNS in the ACM host, it appears as the mac address, this can be changed here.



Approve the discovered host



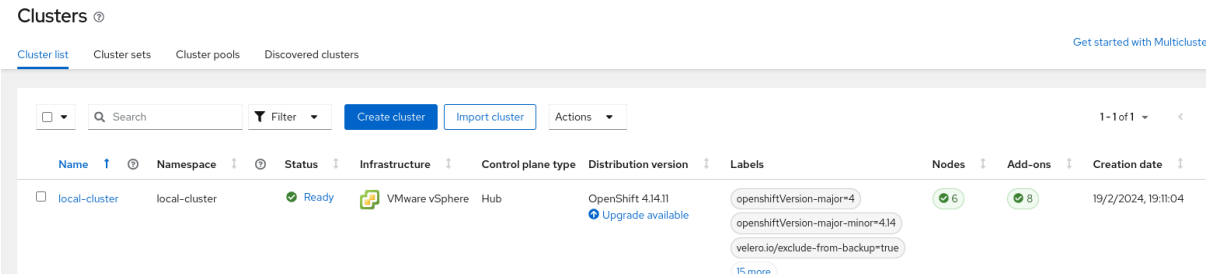
The final result looks like this:



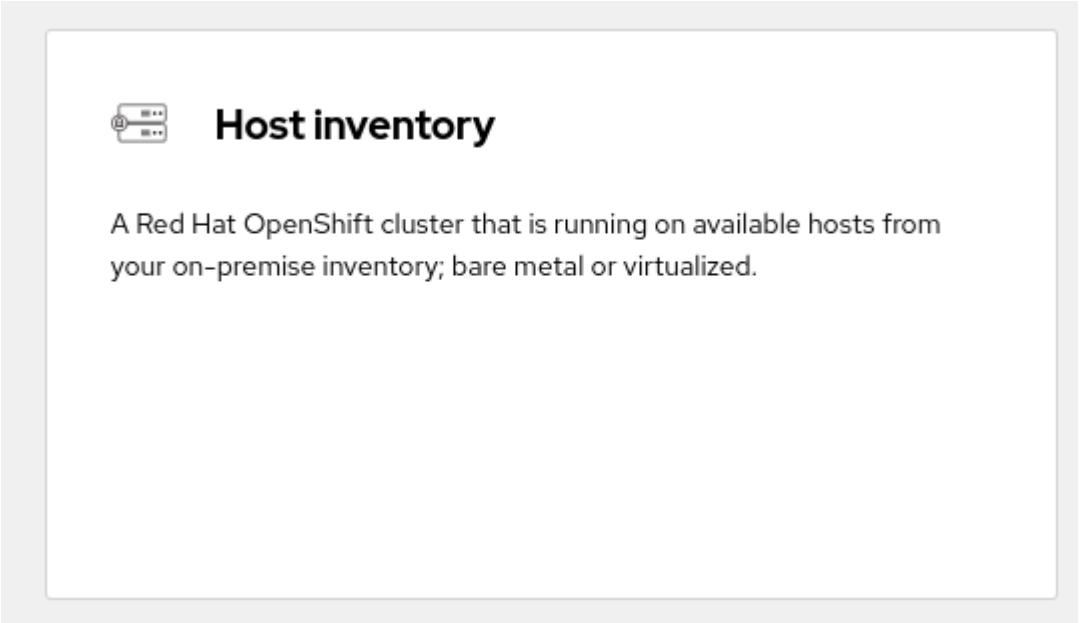
Create the cluster

https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.9/html/clusters/cluster_mce_overview#creating-a-cluster-on-premises

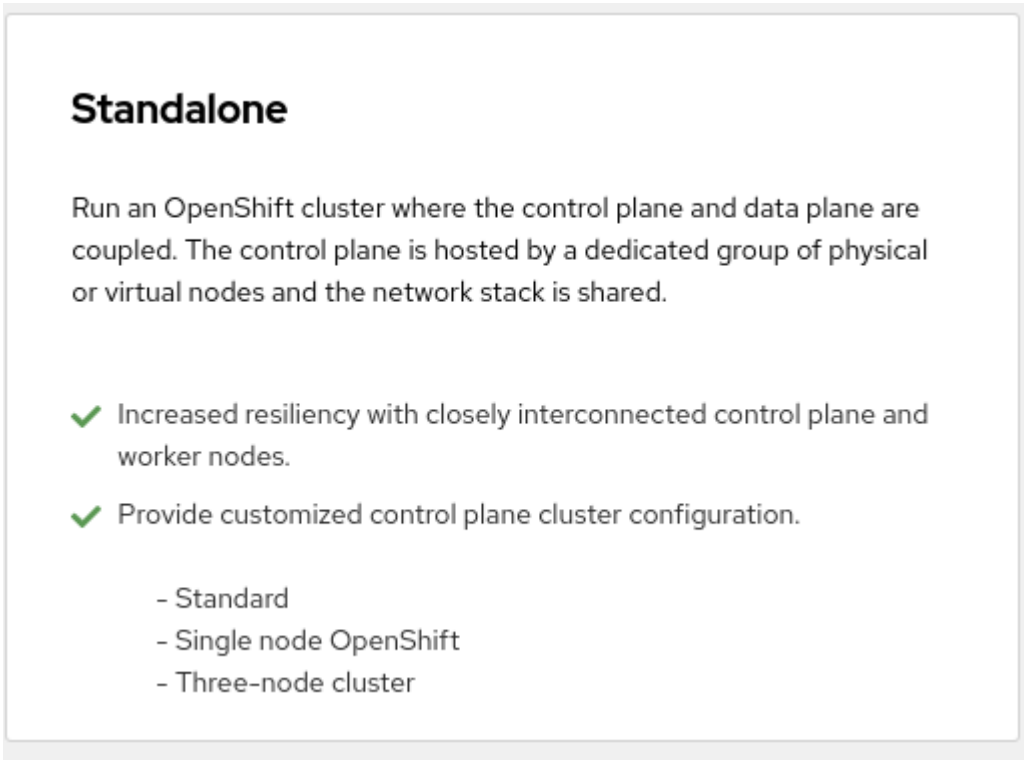
Go to Infrastructure -> Clusters -> Create cluster



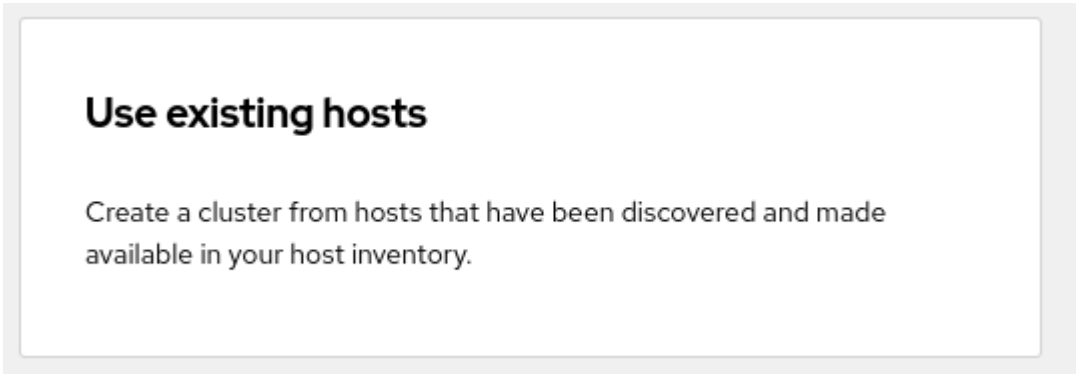
Select Host inventory



In this case, choose Standalone as opposed to Hosted control plane cluster.



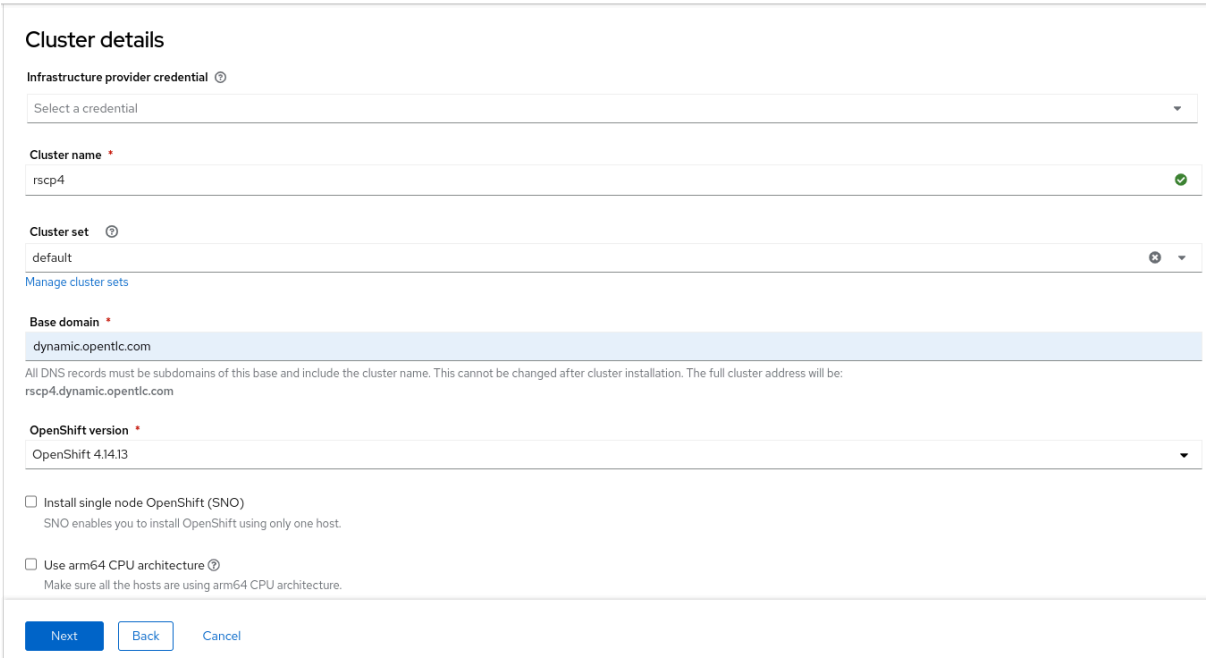
Use existing hosts



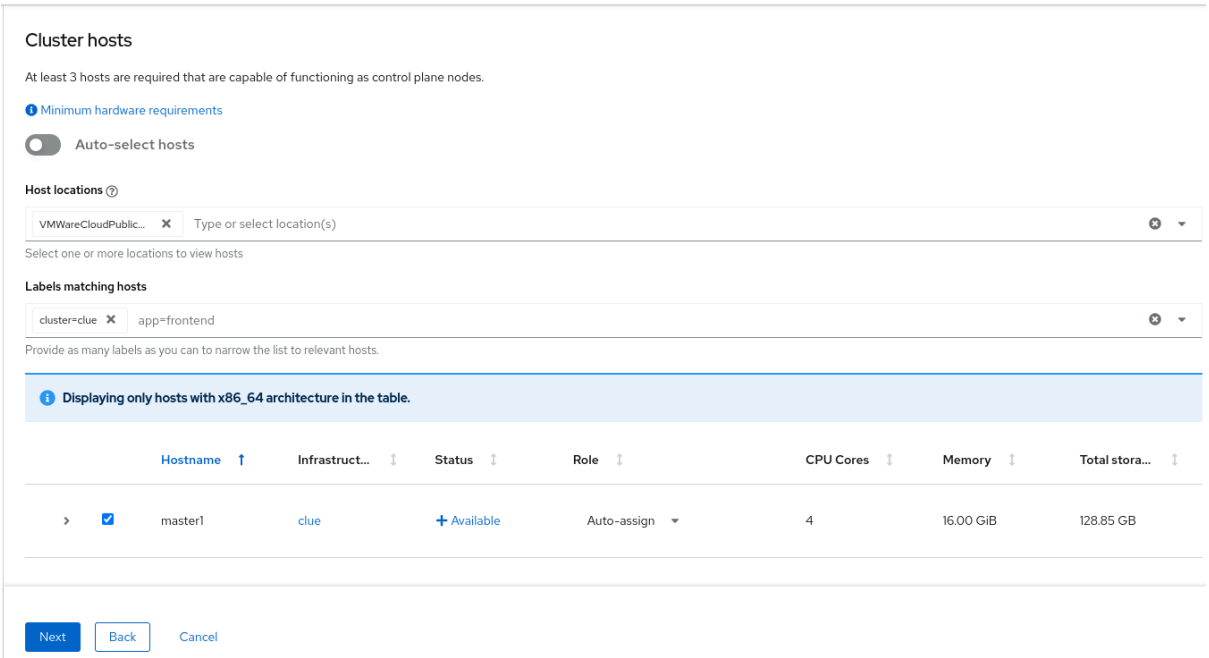
Enter the cluster details.
The cluster name must be the GUID assigned by demo.redhat.com

GUID `rscp4`

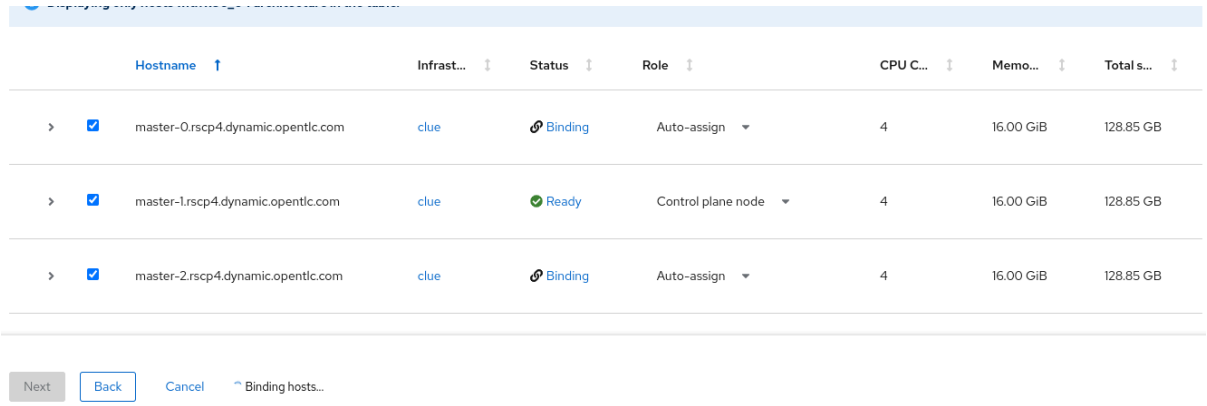
The base domain in the case of demo.redhat.com is dynamic.opentlc.com
Add a pull secret



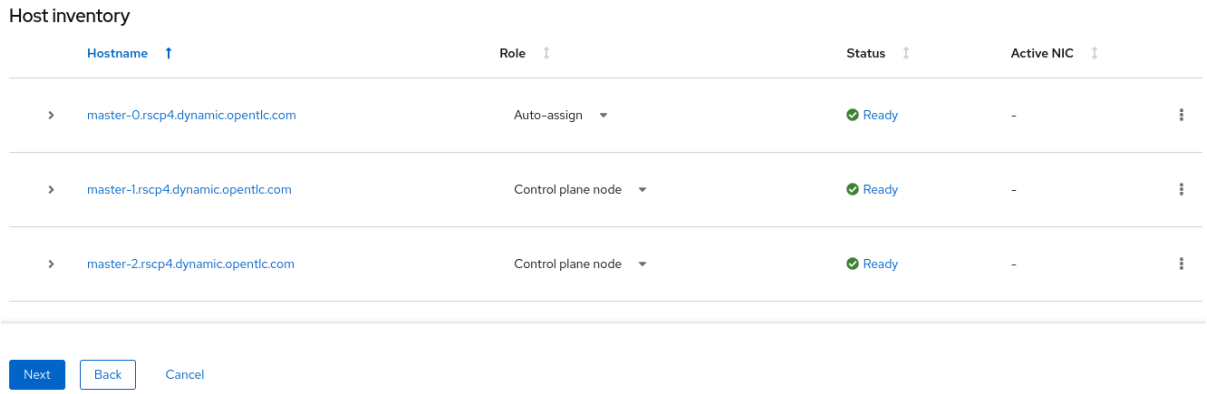
Assign the hosts for the cluster. In this case a 3 node compact cluster is being created, and the hosts are being assigned manually



After clicking Next on the previous page, the hosts are checked and bound



Add the networking configuration.
In this case User-Managed networking is used because the provide the LB and DNS
Add a public ssh key to propagate to the nodes.
It will take a short while until the hosts status goes from insufficient to Ready.



The next page shows the summary before proceeding to the actual cluster installation.

TODO

Cluster Image Sets

When the host inventory settings are defined, a collection of **clusterimagesets** are created:

- All belong to the fast channel.
- Only the ones with **visible: “true”** are shown as options when installing a new cluster

https://access.redhat.com/documentation/en-us/red_hat_advanced_cluster_management_for_kubernetes/2.9/html/clusters/cluster_mce_overview#release-images-intro

https://access.redhat.com/articles/6961617

https://github.com/stolostron/acm-hive-openshift-releases/blob/backplane-2.5/subscribe/subscription-stable.yaml

Install a cluster with customizations

How do I install a managed cluster with customizations, when I don’t have access to the install-config.yaml file?