***Chapter 11*: Building a Hybrid Cloud on Oracle Cloud Infrastructure using Oracle Cloud VMware Solution**

Oracle and VMware jointly developed a fully certified and supported **software-defined data center solution** (**SDDC**) known as **Oracle Cloud VMware Solution** (**OCVS**). This solution leverages the underlying **Oracle Cloud Infrastructure** (**OCI**) to host a highly available VMware SDDC, allowing seamless migration of all on-premises VMware workloads to OCVS.

OCVS is a fully secure VMware SDDC that's comprised of OCI resources and VMware software and licenses. The base configuration of the OCVS includes three OCI bare-metal compute hosts (3x BM.DenseIO2.52 instances) to achieve high availability, along with other OCI products such as OCI **virtual cloud network** (**VCN**). The OCVS is highly scalable, starts with three hosts, and can scale up to 64 hosts in a single OCVS cluster. The base OCVS configuration comes with 156 OPCU, 2,304 GB of physical memory, and 153 TB of **Non-Volatile Memory Express** (**NVMe**)-based raw storage. The OCVS includes VMware software such as vSphere, vSAN, NSX-T, and vCenter Server. The vSAN converged storage technology ensures the availability of data and replicates data across all the bare-metal hosts that are used for the OCVS cluster.

OCVS is the only VMware SDDC on the public cloud that provides L2 network capabilities. It allows the applications that depend on L2 networking to run on the public cloud.

Unlike other cloud provider's VMware SDDC solutions, the OCVS is fully secured and controlled by the customer, and it can be customized based on the customer's needs. OCVS is also well integrated with other OCI services through the OCI VCN, configured with a different gateway to communicate with the other network and OCI services.

In this chapter, we're going to cover the following main topics:

* Understanding the solution overview of the OCVS solution
* Deploying an OCVS cluster
* Accessing an OCVS cluster
* Connecting an OCVS cluster to the internet

**Understanding the solution overview of the OCVS solution**

In this section, we will illustrate the basics of how OCVS is designed and deployed in OCI and its integration with other OCI services running natively, as well as other Oracle PaaS solutions.

OCVS is designed as per the VMware validated design guidelines, and the architecture uses from 3 to 64 BM.DenseIO2.52 nodes to provide full VMware SDDC capabilities.

The OCVS architecture shown in the following diagram illustrates how each component of the VMware SDDC stack is configured and deployed on an OCI compute BM instance.

The OCVS architecture can be divided into three key components: network (VMware NSX-T), compute (VMware vSphere), and storage (VMware vSAN). This section will describe the different components that are used for the VMware SDDC stack in OCI:

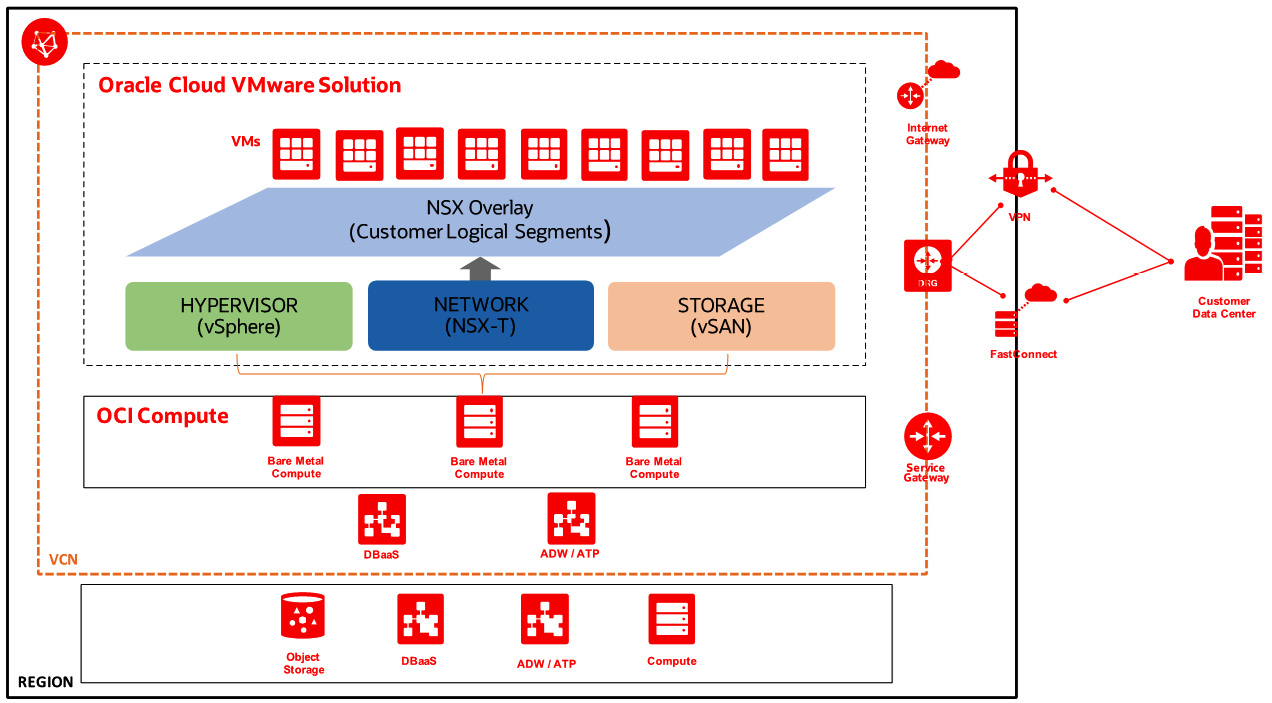


Figure 11.1 – OCVS solution integration within OCI

There are some restrictions as to what you can do on top of OCVS, and they are as follows:

* Customers cannot bring their own VMware licenses.
* The vSAN layer in OCVS will use local NVMe drives.
* The OCVS cluster will boot its ESXi image off a remote **Internet Small Computer Systems Interface** (**iSCSI**) boot volume.
* The smallest cluster a customer can provision contains three hosts.
* The largest cluster a customer can provision contains 64 hosts.
* **Border Gateway Protocol** (**BGP**) is not supported in the environment and all routing with the external networks (outside overlay) will be done using static routes.
* Following the initial bring-up, hosts can be added to an OCVS using the provisioning service. The customers are responsible for adding/removing these new nodes to/from the VMware cluster.

Now, let's dive into each of the solution components of OCVS.

**Virtual cloud network (VCN)**

One of the key components or resources of OCI is VCN, and this is the foundational component for OCVS as well. VCN carries out the connection point for all the topologies that OCVS provides, such as connecting on-premises data centers, connecting OCI resources, connecting to the internet, and connecting to the Oracle services network. You can think of it as an on-premises traditional network that provides switching, routing, and firewall capabilities. This VCN is spanned across an OCI region that provides a single IPv4 space.

Different properties and components comprise a VCN, such as subnets, VLANs, route tables, security lists, gateways, and **network security groups** (**NSGs**). The next one on the component list is ESXi hosts. Let's take a look at them.

**Compute – VMware vSphere (ESXi)**

This is the fundamental layer of the solution and provides an OCI BM DenseIO instance to run VMware vSphere Hypervisor. VMware ESXi is an enterprise-class Type-1 hypervisor for running virtual computers known as **virtual machines** (**VMs**). VMware ESXi, when deployed on a BM instance, provides a strong foundation for the entire SDDC stack. At the time of writing, only the BM.DenseIO.2.52 OCI shape is supported to run the VMware SDDC on OCI.

The VMware vSphere cluster in OCVS offers a three-node ESXi cluster that delivers 156 OCPUs and 2.25 TB of memory. This can be scaled up to 64 nodes in a cluster. The high availability of the ESXi host's BM instances is taken care of by OCI. In the following diagram, you can see the placement of the ESXi host (OCI instances) and how the solution components of OCVS sit on top of it:

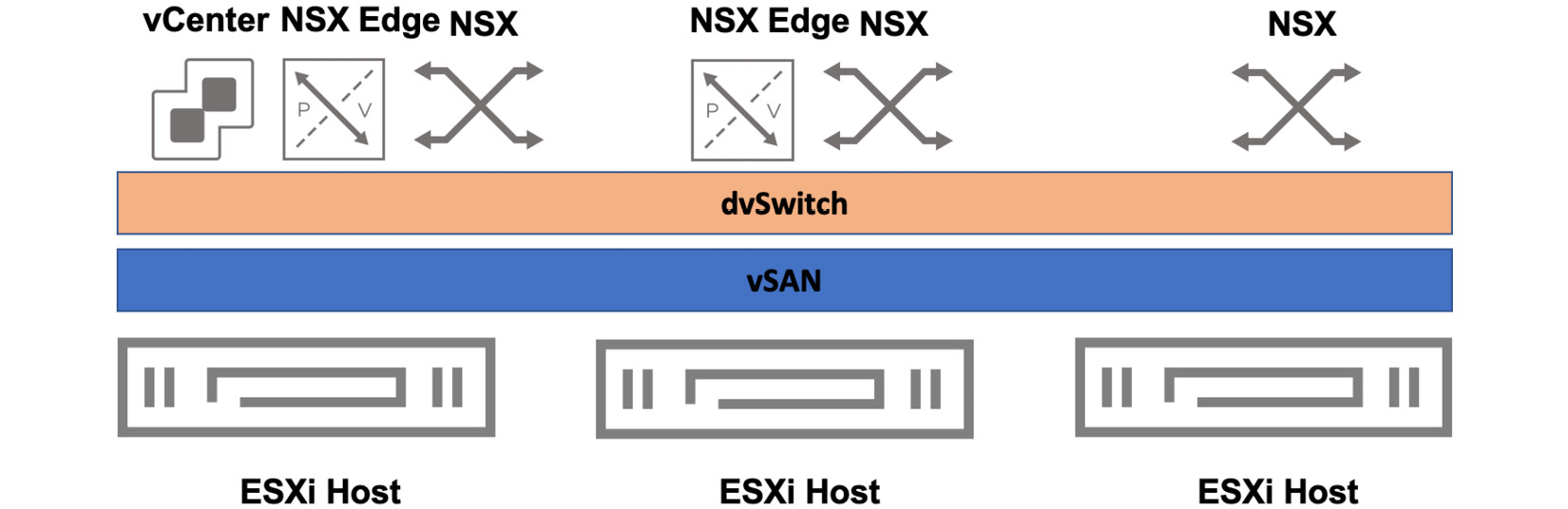


Figure 11.2 – OCVS solution logical diagram

At this stage, we are ready to look at various VMware components. Within these, the main component for all layer 2 networking is NSX. Let's look at what this is.

**Networking – VMware NSX-T**

The network is another important function of the architecture. The BM instance that's hosting VMware ESXi is deployed in an OCI VCN and its subnets to provide core networking underlay capabilities, in addition to security lists and route tables. The VMware NSX-T is a separate product suite from VMware that's deployed as part of the OCVS, which effectively provides agile, software-defined networking capabilities.

The OCVS BM instance that's used for the SDDC stack is backed by 2x25 Gbps network bandwidth and supports 52 **virtual Network Interface Cards** (**vNICs**) in total (26 per physical NIC), ensuring high throughput, low latency, and a fully redundant network.

Three different management components are deployed as part of NSX-T called NSX Manager, NSX Controller, and NSX Edge. All these components reside in a vSphere cluster, running as VMs on top of ESXi hosts backed by an OCI BM instance. The high availability of the software NSX components depends on the availability of the ESXi host and VMware vSphere cluster's native HA/DRS functionality. The high availability of NSX's underlay networking depends on OCI VCN. The OCI VCN regional subnet provides a greater extension to SDDC cluster scale-out operations. In the following diagram, you can see what the physical architecture of an ESXi instance looks like when it is deployed as part of an OCVS solution:

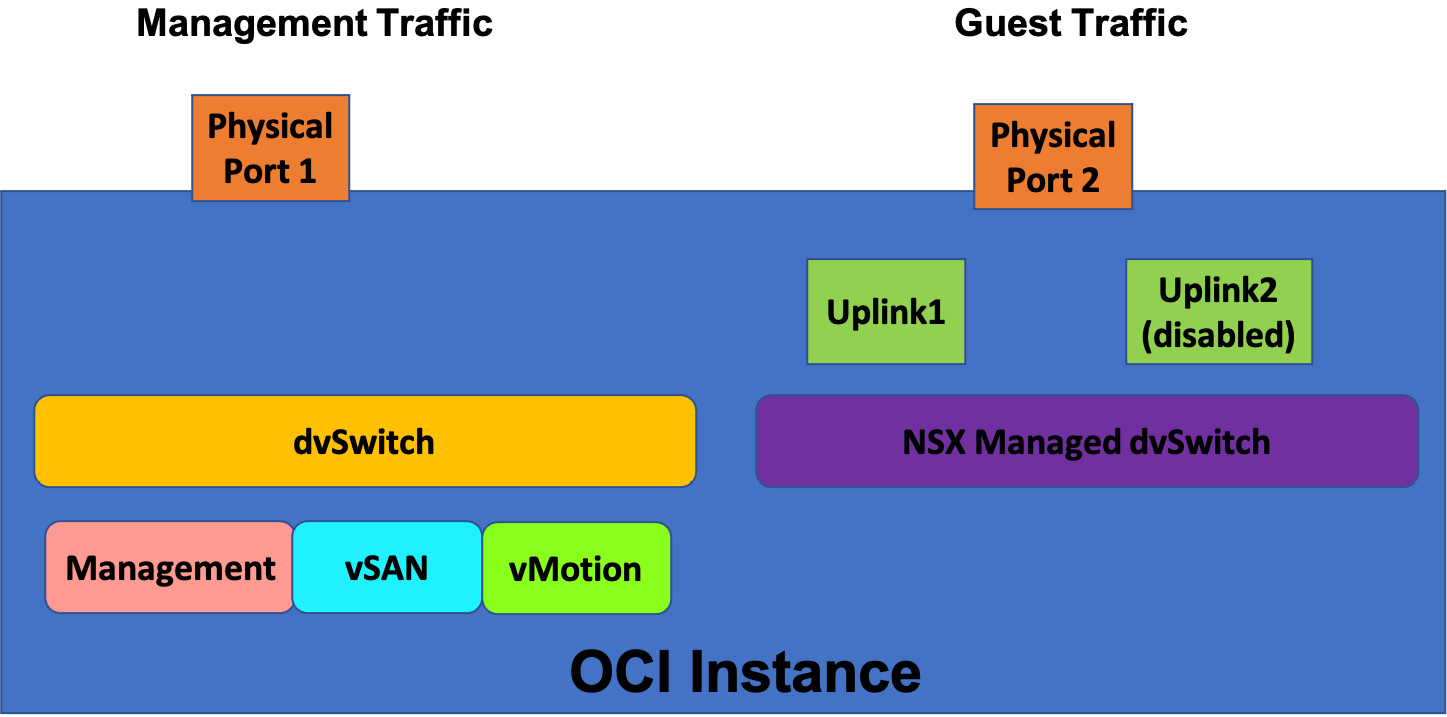


Figure 11.3 – OCVS solution logical diagram

After networking, you need to know about the storage solution that comes with OCVS. Let's take a look at the VMware vSAN solution.

**Storage – VMware vSAN**

vSAN has been chosen as the storage platform for OCVS to provide a hyper-converged, enterprise-class, and software-defined storage. vSAN was built to take advantage of cheap servers since they have locally attached flash disks.

The OCVS includes VMware vSAN storage technology, which provides a single shared datastore (vSAN Datastore) for compute and management workloads (VMs). The OCVS solution provides high-performance storage that has low latency and is built using the NVMe disks from the DenseIO instances. In the following diagram, you can see a logical representation of the vSAN all-flash construct:

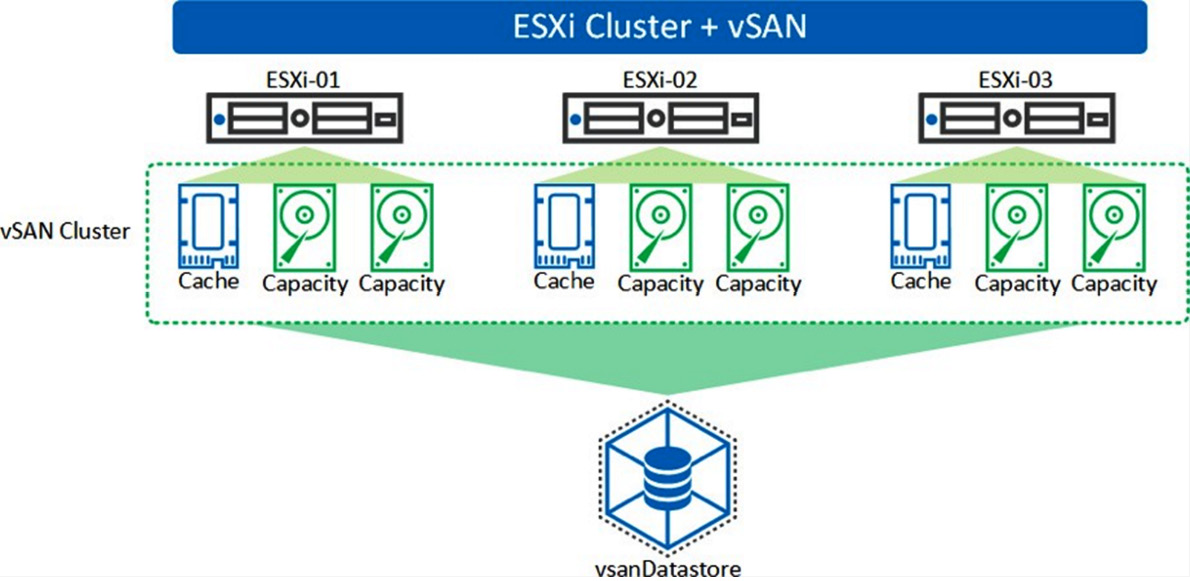


Figure 11.4 – OCVS solution logical diagram

vSAN implements a concept known as fault domains. This term is different from OCI Fault Domain and is referred to in the context of vSAN only. The idea behind the vSAN fault domain is to group the hosts that participate in creating a vSAN datastore. This is a logical grouping, not physical. Using the fault domain construct, vSAN maintains multiple replicas of the same storage objects so that in case of a failure, just one replica is affected.

vSAN storage policies are used to decide on the high availability of individual VMs. Different failures to tolerate policies can be configured with OCVS to decide on the number of failures to tolerate. These storage policies will be configured by OCVS tenants.

OCVS does not support stretched clusters for DR scenarios.

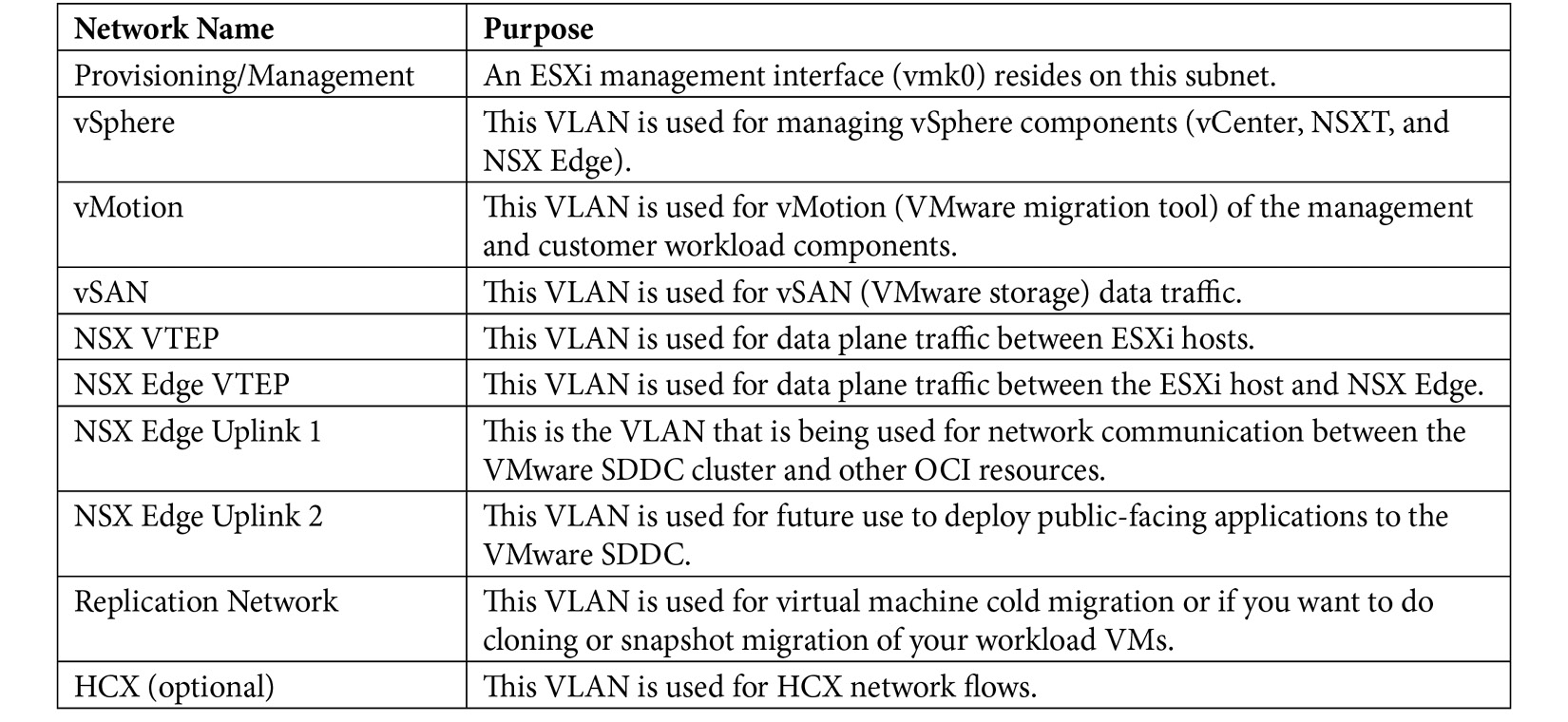
**Deploying OCVS**

OCVS is deployed as per VMware's recommended best practices; all VMware components are highly distributed across different fault domains within a given OCI region's AD.

To get started with OCVS, a user will need to create a VCN or use an existing VCN to deploy an OCVS cluster.

If you plan to use an existing VCN, Oracle recommends using a VCN that has a **/23** or larger IP address CIDR available for running a VMware cluster.

As a part of VMware provisioning, the provisioning cluster will create the following network segments for various VMware functionalities. These disparate network segments ensure that the appropriate traffic segregation is provided:



So far, you have been provided with an overview of the OCVS solution and the components that comprise it. In the next section, we'll deploy an OCVS cluster.

**Deploying an OCVS cluster**

Deploying an OCVS cluster is done in two parts. First, we must deploy the cluster, and then, we must deploy a Windows host on the same VCN to access the cluster. First, let's learn how to deploy a cluster:

1. Sign into the OCI console.
2. Open the navigation menu, select **Hybrid**, and then **VMware Solution**.
3. Click on **Create SDDC**.
4. Provide an **SDDC Name**.
5. Select which compartment you want to place this cluster on.
6. You can **Enable HCX (default option)**. **Hybrid Cloud Extension** (**HCX**) hides the complexity of moving applications between on-premise and OCI. It is a plugin that will be installed once you select it. However, you cannot add HCX once you've created the SDDC.
7. If you've selected HCX, then you must select the appropriate license type as well. The **Advanced HCX** license is suitable for a smaller number of workload migrations, whereas **Enterprise License** is suitable for a higher number of application migrations.
8. Select **vSphere version**. At the time of writing, OCI allows you to deploy vSphere 7.0 update 2, 6.7 update 3, and 6.5 update 3.
9. Choose **Pricing Interval Commitment**. In OCI, there are four different types of commits. You can see some sample output in the following screenshot:

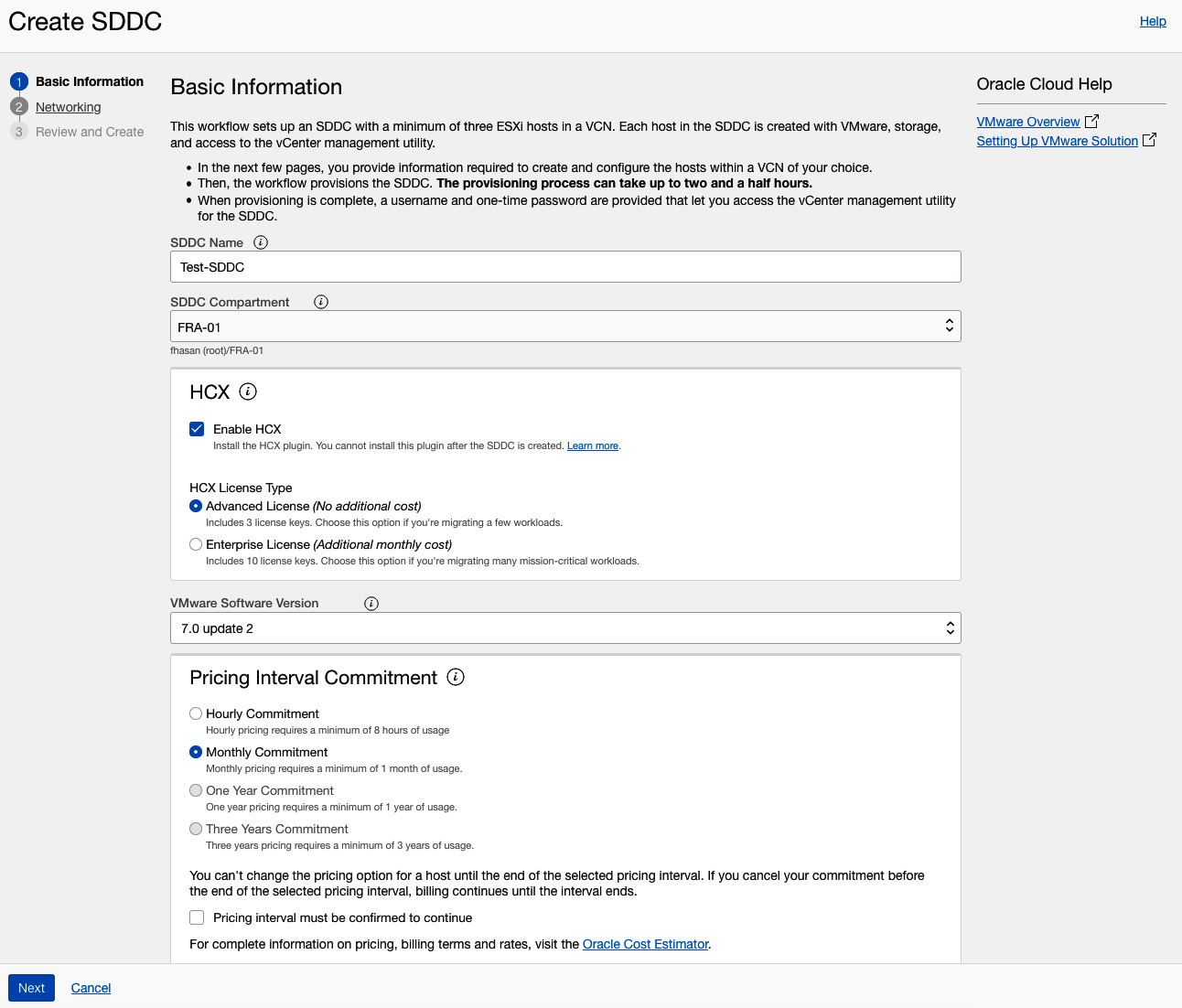


Figure 11.5 – OCVS cluster creation – Basic Information

1. Select the number of ESXi nodes that you want to deploy. The minimum number of nodes is 3.
2. Provide a **name prefix**.
3. Provide an **SSH key**. You will need this to SSH to the ESXi node, in case you want to do some troubleshooting.
4. Choose **Availability Domain** from the drop-down menu and click on **Next**. The following screenshot shows some sample output:

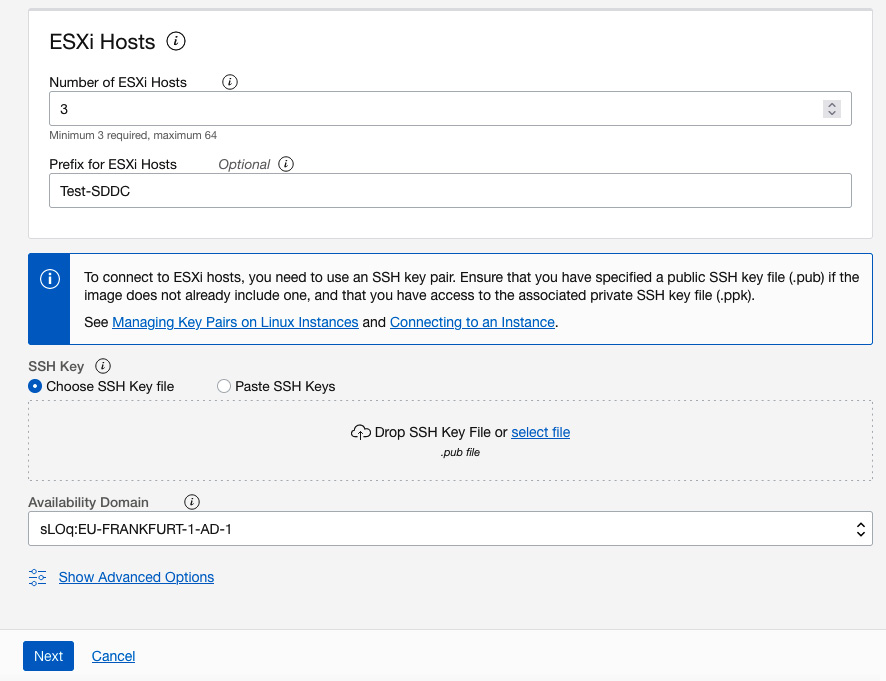


Figure 11.6 – OCVS cluster creation – Basic Information continued

1. The **Networking selection** section is crucial for OCVS cluster deployment. First of all, choose a VCN that will host this SDDC cluster. As a rule of thumb, you need to have at least a **/21** subnet for this VCN if you are deploying vSphere 7.x, or a **/22** subnet for this VCN if you are going to deploy vSphere 6.x.

OCI will chunk the VCN subnet into multiple **/25** segments for the logical segment, which you learned about in the *Deploying OCVS* section. For vSphere 7.x, OCI will create nine segments, while for vSphere 6.x, OCI will create seven segments.

OCI recommends that you let the workflow create the new subnet and VLANs. However, you can use your existing subnet and VLANs if you have pre-created those.

1. In the **SDDC Networks** section, provide a **/22** since we have chosen vSphere 6.7. Once you've typed in the subnet, it will validate on the fly. If the provided subnet is available, then you will see **This CIDR block is available**.
2. You will notice that the workflow is also looking for a **NAT gateway** since we selected HCX on the first screen. If there is an existing gateway connected to the same VCN, then the workflow will use it; otherwise, it will create one. This NAT gateway is required for HCX license activation, updates, and internet connectivity. You can see an example output in the following screenshot:

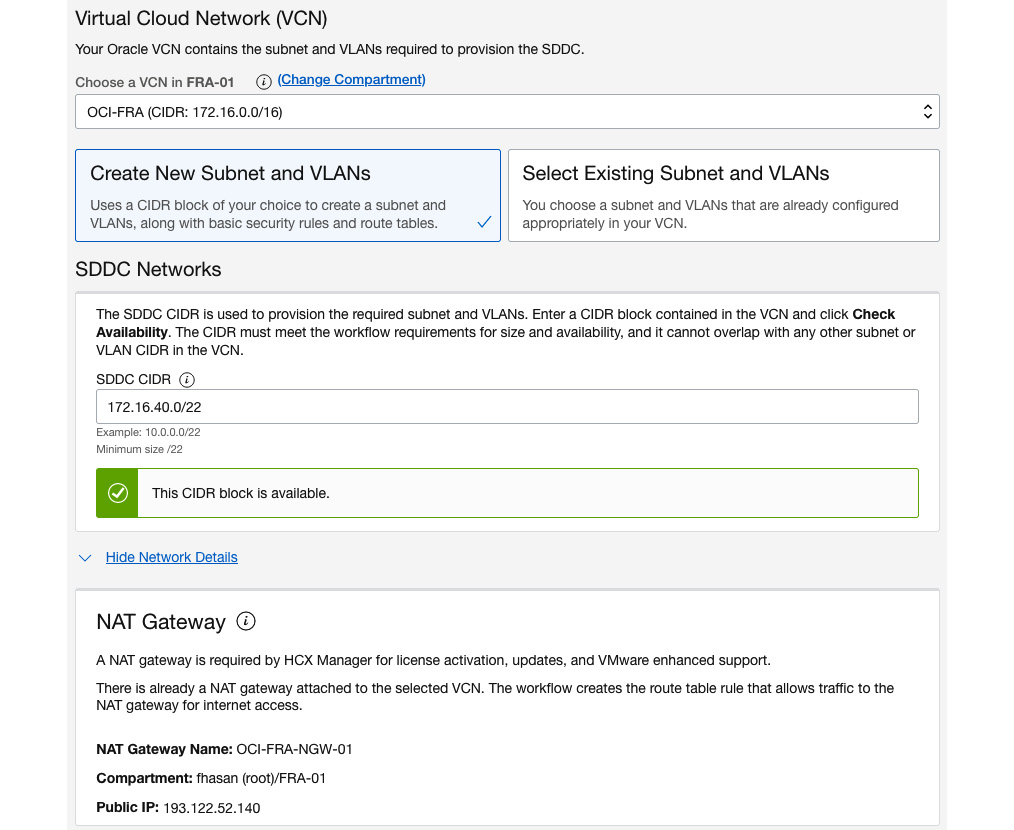


Figure 11.7 – OCVS cluster creation – VCN details

1. At this point, you will see that OCI has populated the **Subnet** and **VLANs**information.

The last thing that you need to provide (although it is optional) is the logical network segment that the workload will be connected to using an NSX logical switch. A minimum of **/30** is required. You can see an example output in the following screenshot:

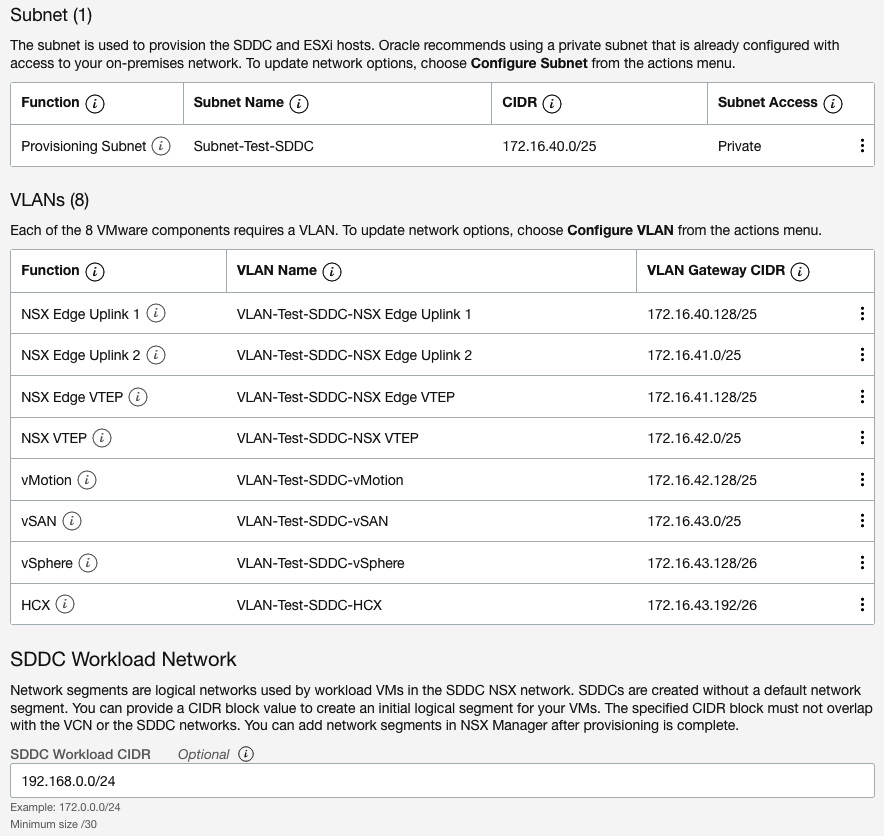


Figure 11.8 – OCVS cluster creation – Subnet and VLAN details

1. Click on **Next**.
2. Review and edit this information from this screen if you need to.
3. Click on **Create SDDC**. This should look as follows:

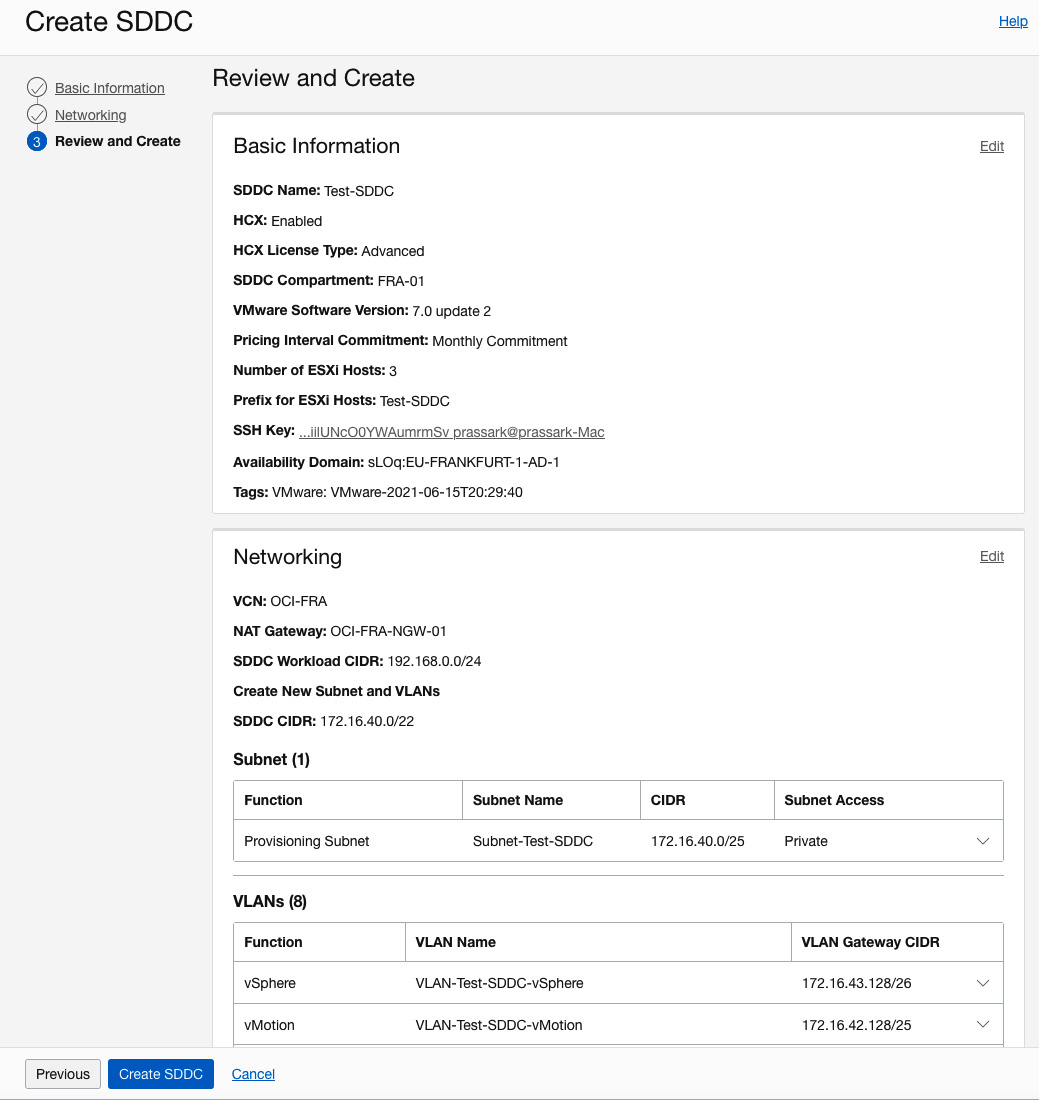


Figure 11.9 – OCVS cluster creation – Review and Create

1. It takes a considerable amount of time to finish deploying the SDDC cluster. Once done, you can go to the **Software-Defined Data Center Details** screen and check the status of this. You can also use this page to get the vCenter and NSX Manager login information, as well as the URL.

So far, you have learned how to create a cluster. You may have noticed that OCI has provided a neat workflow for hiding a lot of the complexity surrounding it. In the next section, we will show you how to access the SDDC cluster. You will also learn how to access vCenter Server and NSX Manager.

**Accessing an OCVS cluster**

Accessing an OCVS cluster requires a Windows host on the same subnet where you deployed the SDDC cluster. Let's create one and then use it to connect to the OCVS cluster.

Let's create a standard OCI compute instance:

1. Sign into the OCI console.
2. Open the navigation menu, select **Compute**, and then select **Instances**.
3. Click **Create Instance**.
4. Provide a **Name** and select a **Compartment** where you want to deploy it.
5. In the **Availability Domain** section of the **Configure placement and hardware** section, choose where you want to place the AD. Additionally, you can click on the **Choose a Fault Domain for this Instance** checkbox and select a **Fault Domain** from the drop-down menu.
6. Select **Windows** from the **Image** page. Click on **Select Image**.
7. In the **Shape** section, by default, the **VM.Standard.E4.Flex** shape type will be selected, which has 1 core OCPU, 16 GB memory, and 1 Gbps network bandwidth shape. You can see an example of this in the following screenshot:

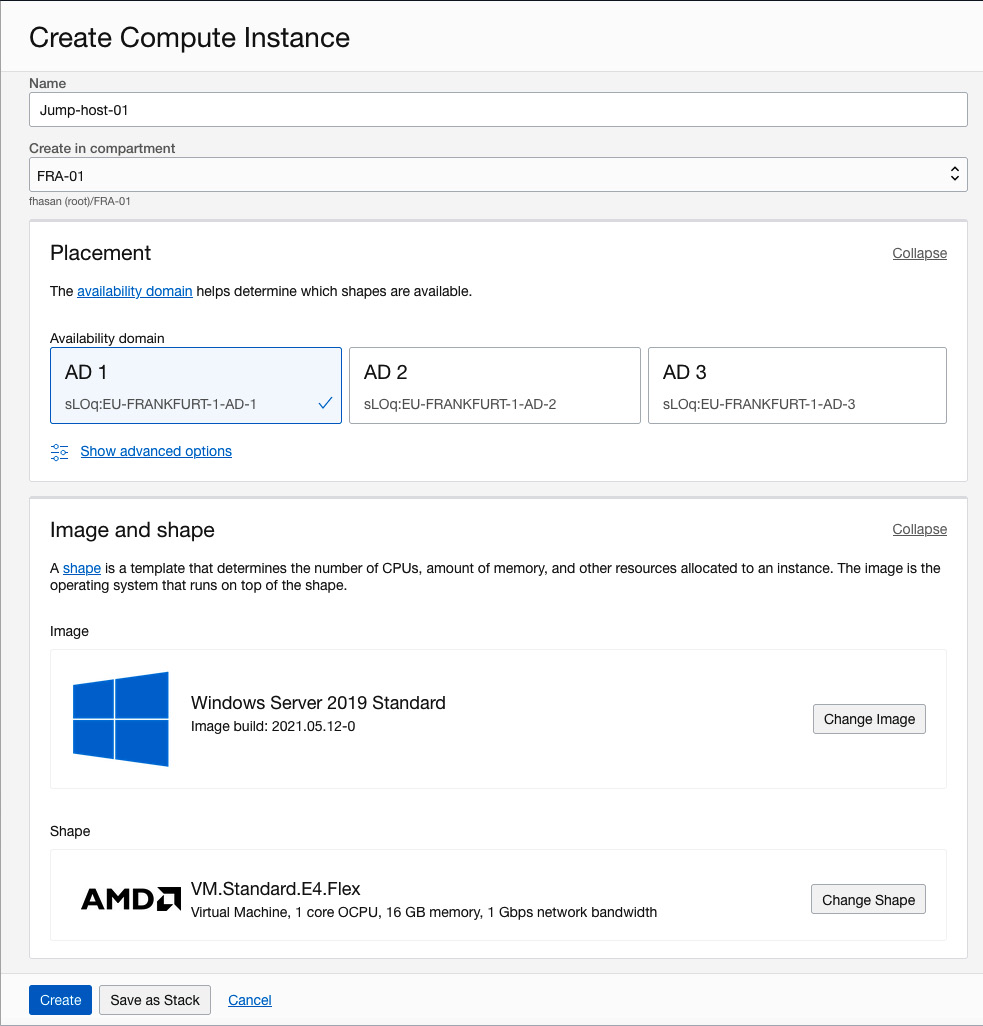


Figure 11.10 – Instance image selection

1. In the **Configure Networking** section, choose the VCN and the subnet that you want to connect this instance to.
2. Select the **Assign a Public IP Address** radio button so that you have access to this instance over the public internet. You can see an example of this in the following screenshot:

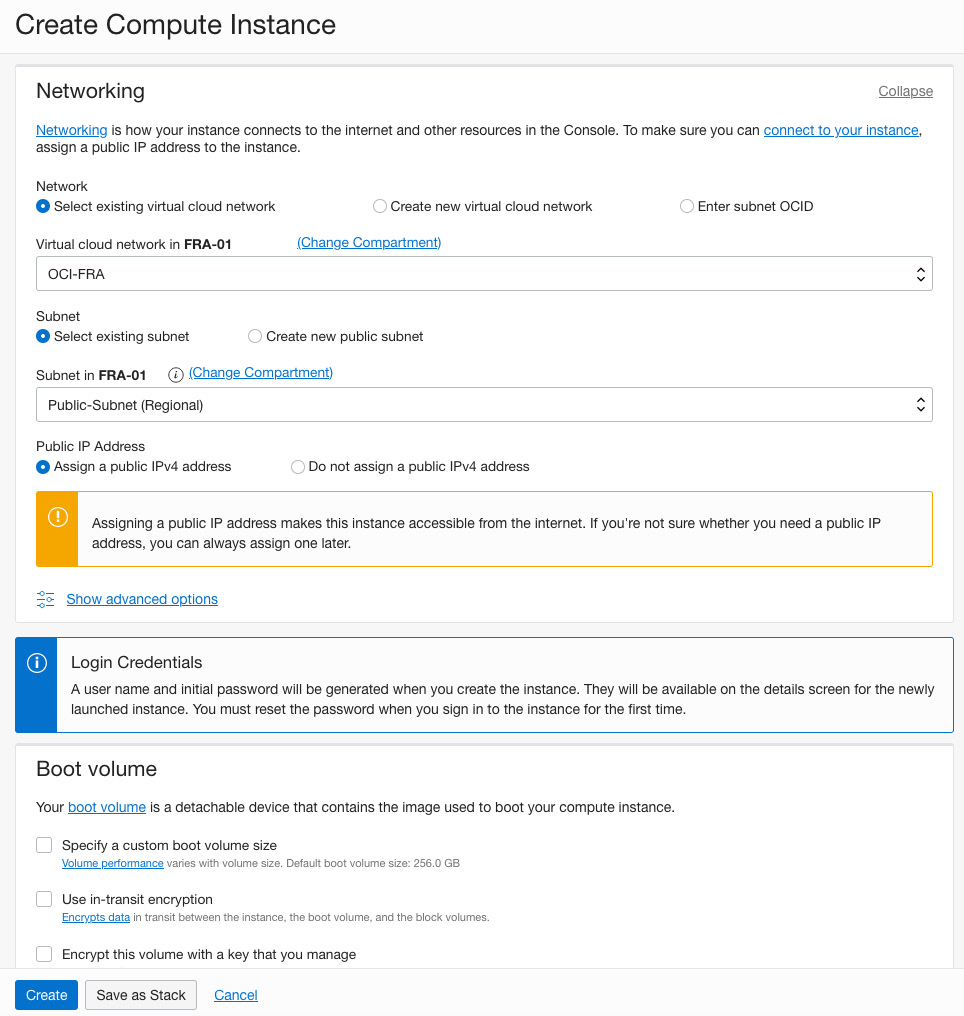


Figure 11.11 – Create Compute Instance wizard – Networking section

1. Optionally, you can **Specify a custom boot volume size**. While creating this instance, you can choose the default boot volume size of this instance or a custom boot volume size up to 32 TB.

If you are provisioning a Linux image-based instance, then you must set your custom boot volume size so that it's more than its default volume size, which is 50 GB. If you want to create an instance based on Windows operating system images, then the same rule applies; that is, you must set the custom boot volume size so that it's more than the default boot volume size, which is 256 GB.

The reason behind setting up this default boot volume size is so that you have enough space for Windows patches and a page file.

1. Optionally, you can specify **Use in-transit encryption**. In-transit encryption allows you to encrypt the volume when it's being created.
2. Optionally, you can specify **Encrypt this volume with a key that you manage**.
3. Click on **Create**.
4. Once it is in a **Running** state, you must copy the instance's public IP address and connect to it. You will get the initial password from this page as well. An example of this can be seen in the following screenshot:

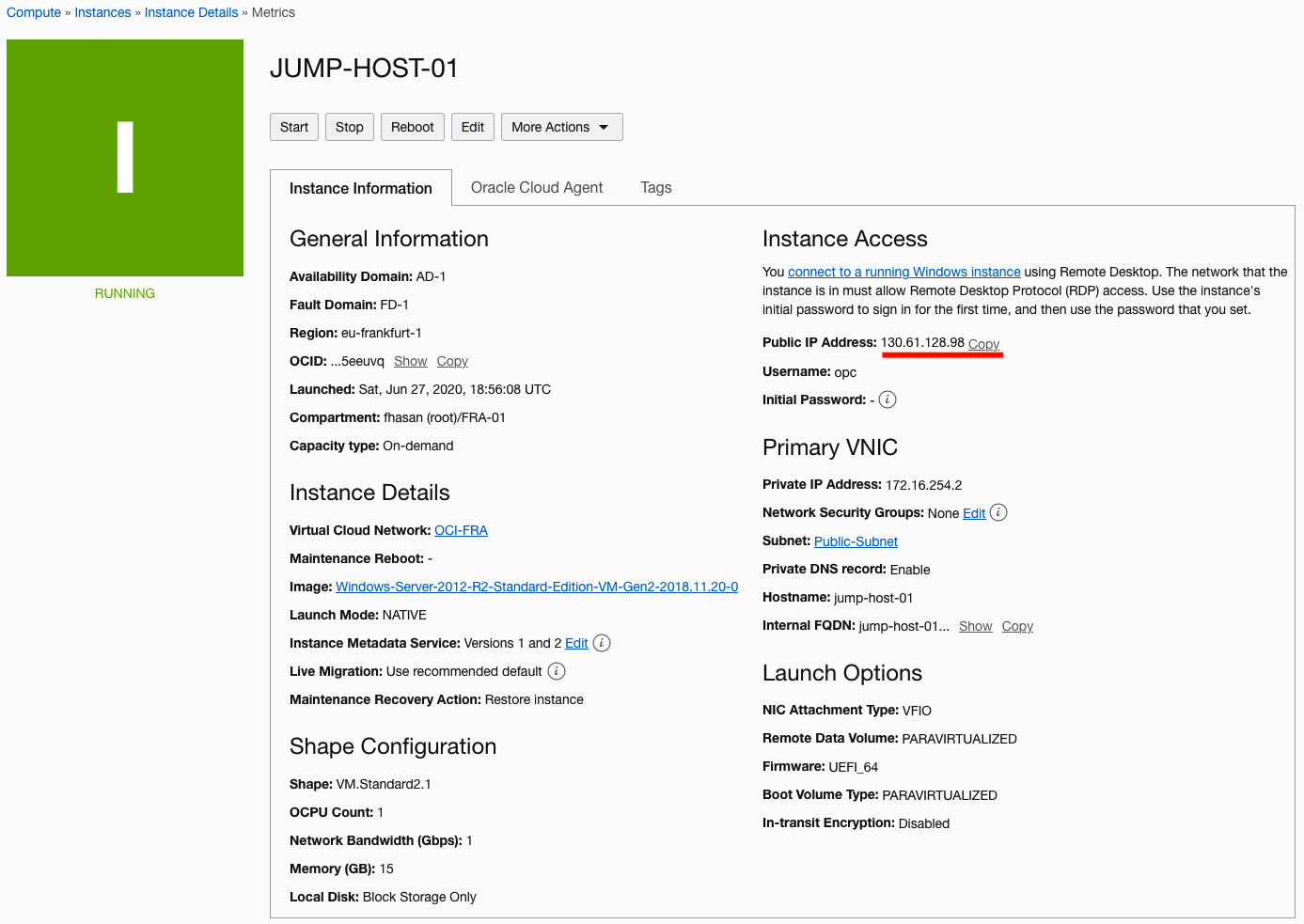


Figure 11.12 – Instance details

1. When you get the instance up and running, use your favorite RDP client to log into it.
2. Once you've logged into the Windows host, you need to open a browser and paste the URL of the vCenter Server that you saved from the SDDC details page. You will have also saved the initial credentials for both vCenter and NSX Manager.
3. Paste in your **vCenter Server URL** and press *Enter*.
4. Click on **Launch vSphere Client (HTML5)**, as shown in the following screenshot:

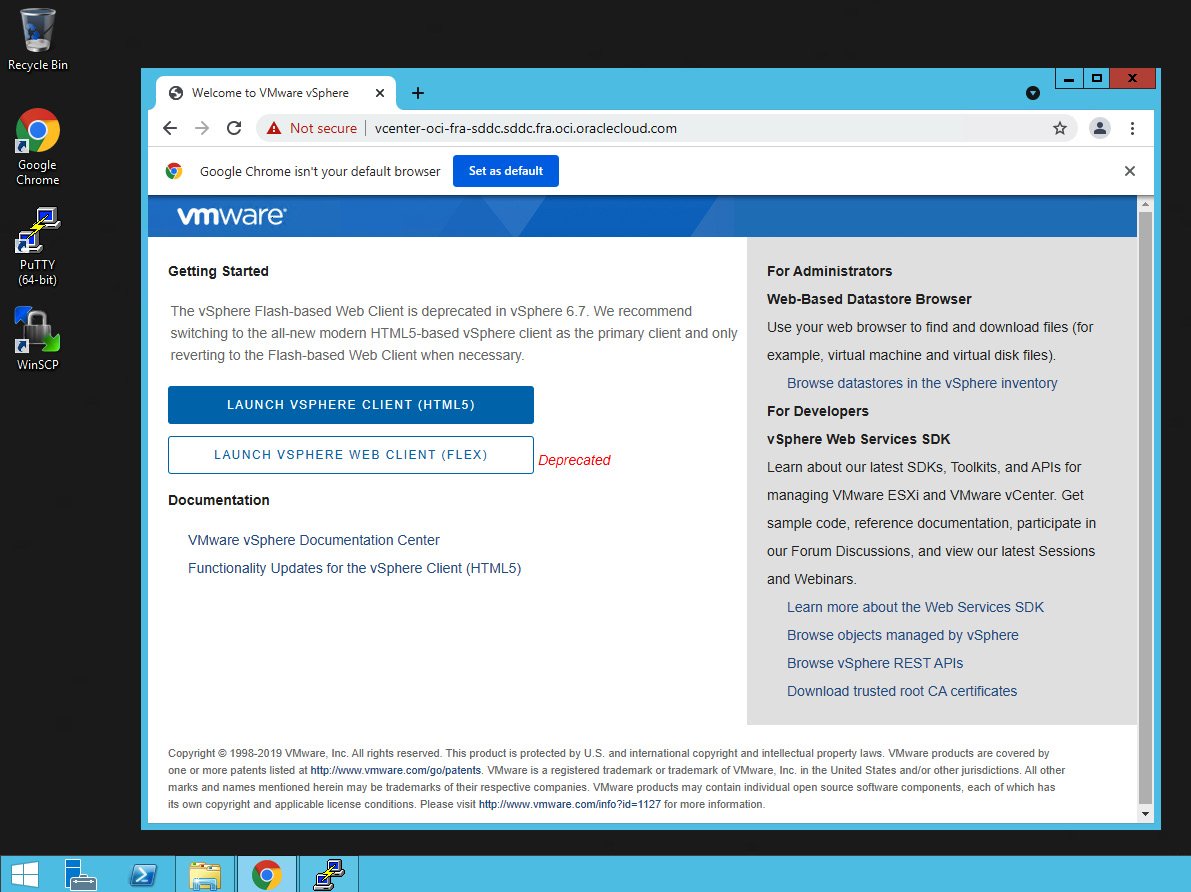


Figure 11.13 – vSphere client

1. Provide the necessary credentials and click **Login**.
2. You will be logged into vCenter Server, as shown in the following screenshot:

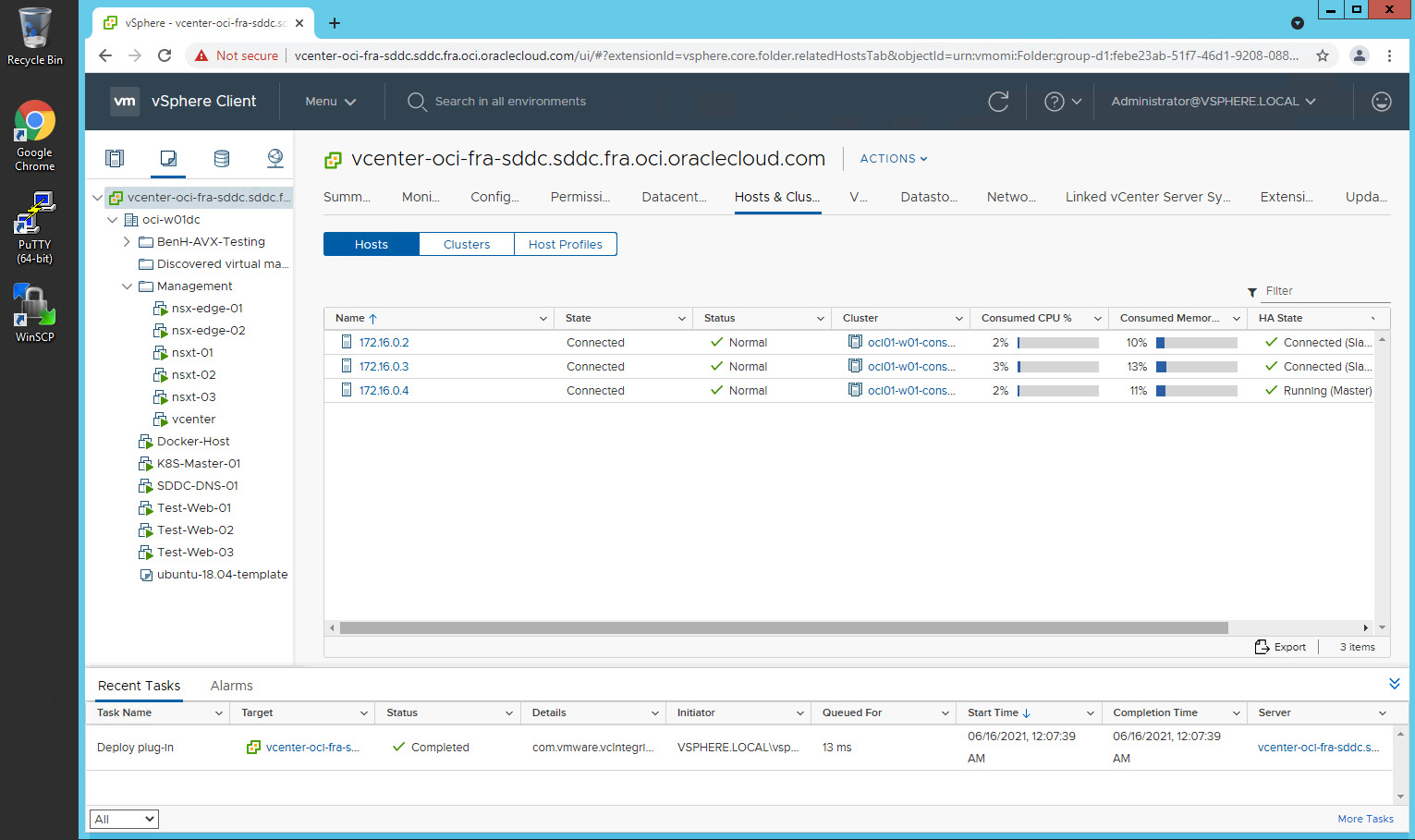


Figure 11.14 – vCenter Server

1. Follow the same procedure for NSX Manager as well. When you log into NSX Manager, you will see the following screen:

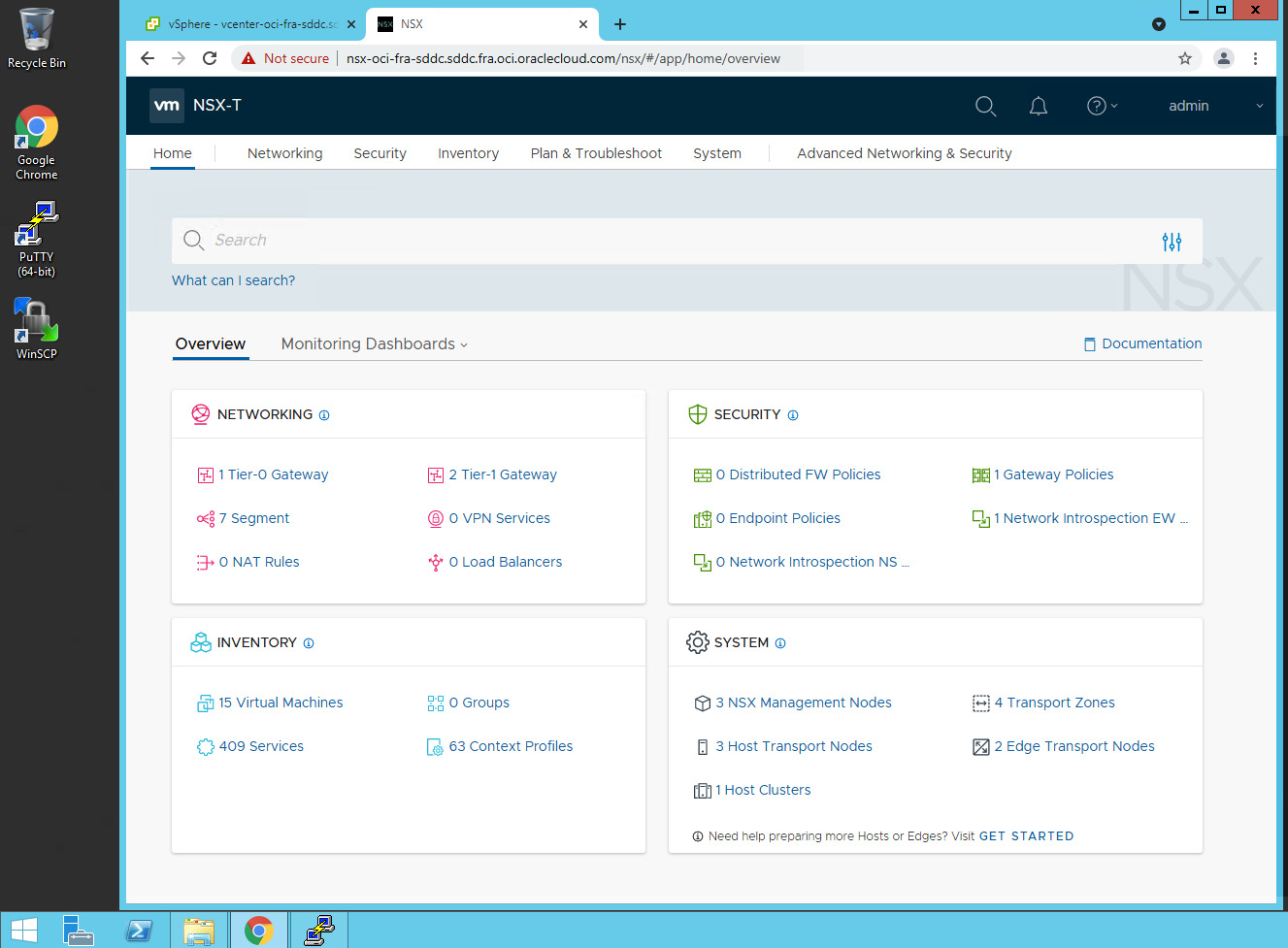


Figure 11.15 – NSX manager

In this section, you learned how to access an SDDC cluster by deploying a Windows host onto the same VCN. In the next section, we will show you how to connect your SDDC to the internet using the workflow that OCI provides.

**Connecting an OCVS cluster to the internet**

Connecting an OCVS cluster to the internet is very straightforward and can be done using a simple workflow. If you selected the HCX option while deploying the cluster, then half of the work has already been done, as that workflow would have created a NAT Gateway, if one hadn't already been provided. Let's run the workflow to connect the cluster to the internet:

1. From the **SDDC Cluster details** page, select **Configure connectivity to the internet through NAT gateway**, as shown in the following screenshot:

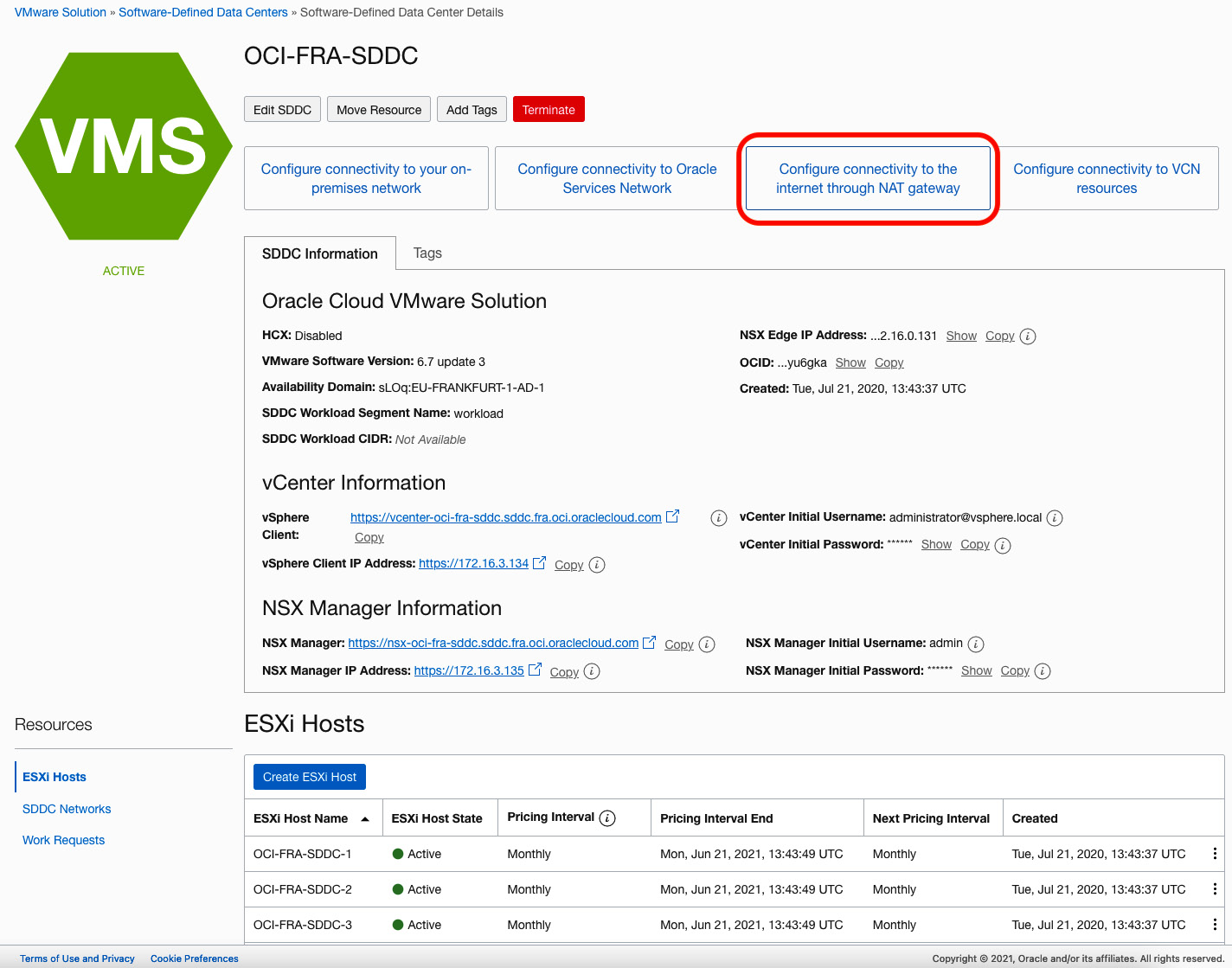


Figure 11.16 – Internet connectivity workflow

1. Since a NAT gateway has already been provided, this workflow will use it.
2. It will add a route rule to the Route Table entry for the NSD Edge Uplink 1 VLAN subnet. Basically, it will allow all the internal traffic to go out but via the NAT gateway.
3. It will also add an allow any-any egress rule to the NSX Edge uplink VLAN's network security group, as shown in the following screenshot:

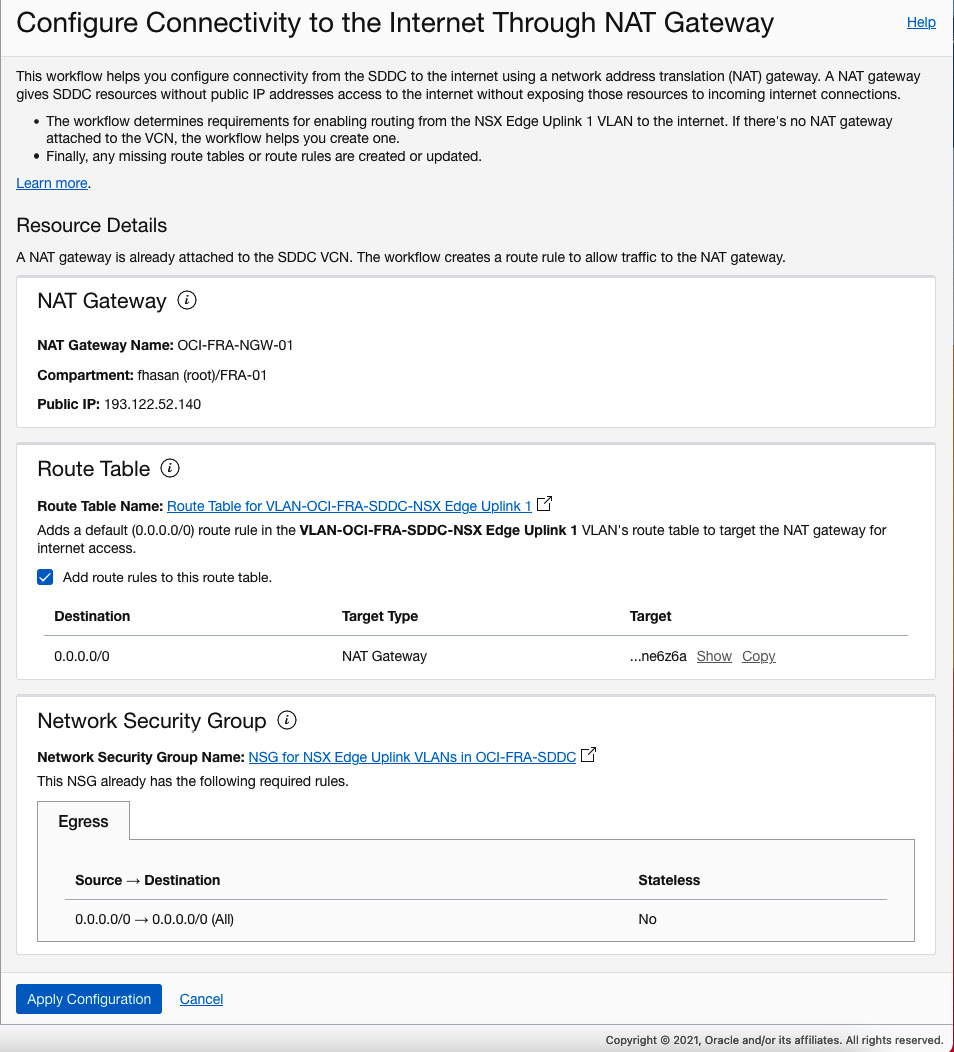


Figure 11.17 – The Configure Connectivity to the Internet Through NAT Gateway page

1. Click on **Apply Configuration**.
2. Click on **Close**.

Your SDDC cluster now has access to the internet, which means you can use the HCX plugin to activate licenses and more!

**Summary**

In this chapter, you learned how Oracle and VMware have jointly worked together and created a solution that customers use to not only extend their on-premises data centers, but also to migrate workloads from on-premises to OCI. You have learned about the basic architecture of the OCVS solution, as well as how to create and access it. You have also seen how OCI-provided workflows can help you connect the SDDC cluster to the internet.

Throughout the last 11 chapters, you have learned about different topics surrounding OCI, and we thank you for joining us. This book will benefit you if you are looking to learn Oracle Cloud from an Architect's perspective and can easily uplift your career as an Oracle Cloud Architect. This book will also certainly help you prepare for your Oracle Cloud certifications, such as Oracle Cloud Infrastructure Architect Associate or Architect Professional.