Infoblox

DEPLOYMENT GUIDE

Deploying Infoblox-Flex based Grid on VMware vCloud Director

Table of Contents

Introduction

VMware Cloud Director(VCD) is a leading cloud service-delivery platform used by some of the world's most popular cloud providers to operate and manage successful cloud-service businesses. Using VMware Cloud Director, cloud providers deliver secure, efficient, and elastic cloud resources to thousands of enterprises and IT teams across the world.

VCD users can deploy and build Infoblox-Grid using vNIOS ova image, which can be downloaded from here.

This deployment guide covers deploying Infoblox-Grid(Grid master *1 , Flex Members*2, and a windows client) onto VCD

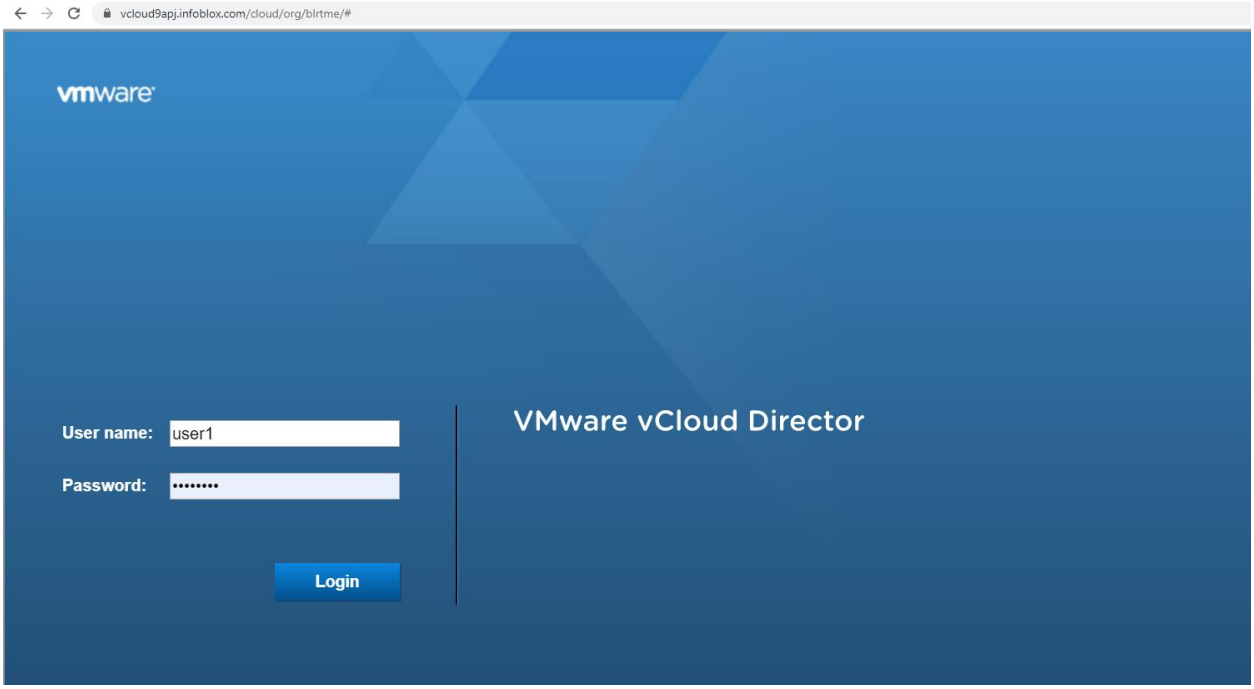
Pre-requisites to deploy Infoblox Infoblox-Flex based grid on VCD.

- Target VDC(Virtual Datacenter) must have enough resources to accommodate Infoblox-Grid.
- User must have a minimum vapp author level permissions to deploy and build Infoblox-Grid.
- If a user is trying to deploy a vApp from scratch, a minimum of 2 vapp networks must be provisioned.
- Access to atleast one of the catalogs where vNIOS ova image can be uploaded.
- Access and availability of vNIOS OVA image.
- Access and availability of Windows client(Any Windows OS, preferable windows7 or 10)

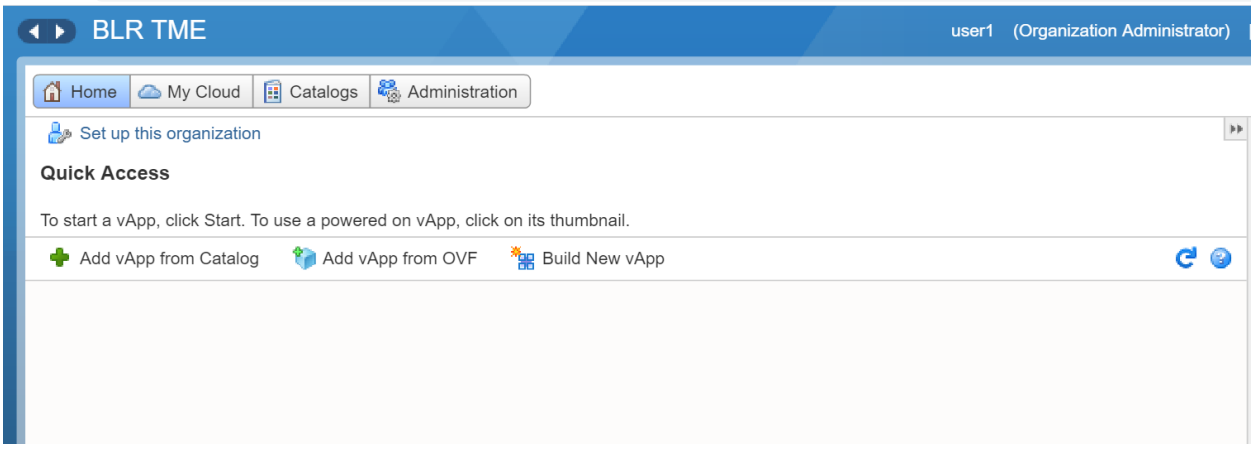
Deploying Infoblox Flex based grid on VCD via GUI

Upload vNIOS OVF to a catalog.

1. Login to your VCD environment using flash-based GUI.

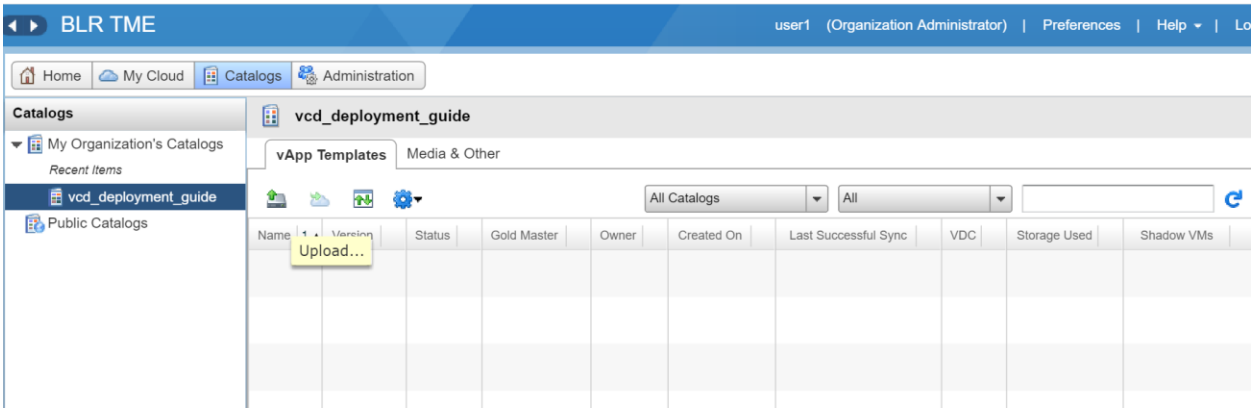


2. Once logged in click on **Catalogs** , present at the top.



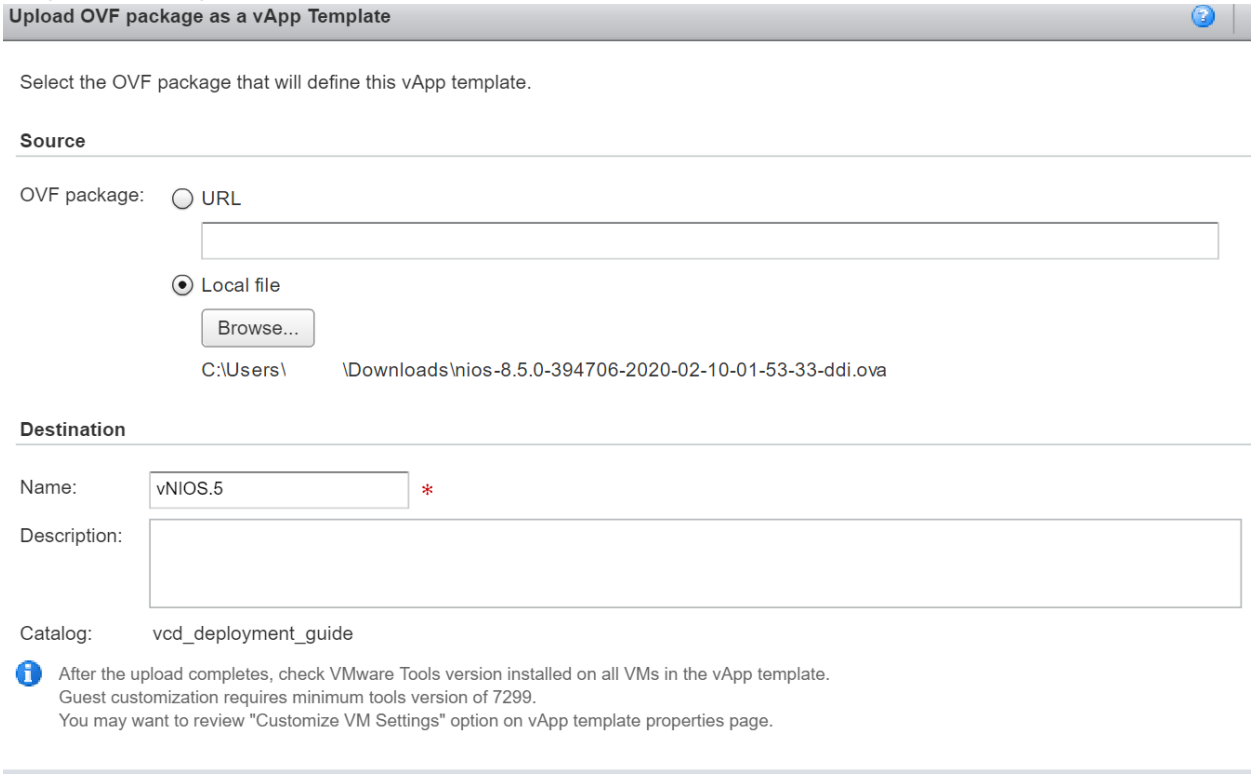
3. You will be presented with a list of catalogs, click on the catalog which you have access to.
Note: We will be using "***vcd_deployment_guide"*** catalog in this deployment guide.

4. Click on upload icon to upload vNIOS OVF file as template to this catalog.

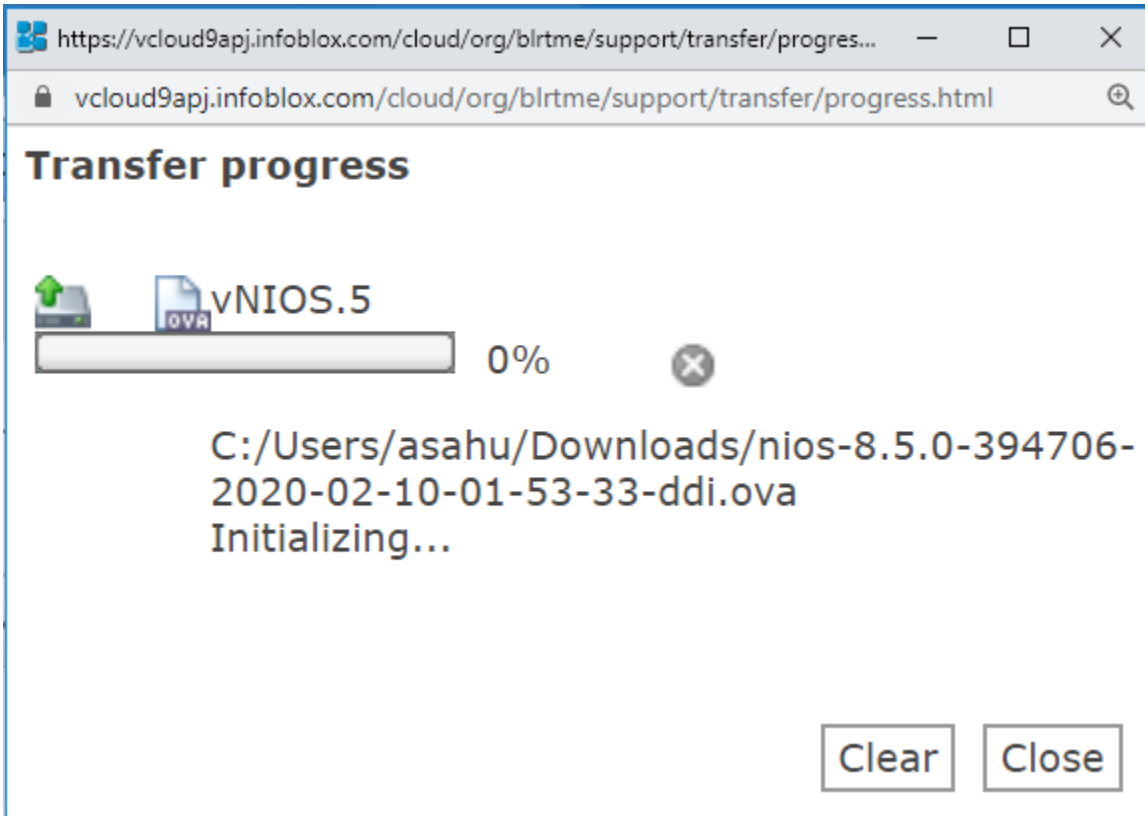


5. vNIOS OVF file can be either uploaded from a URL or from the local machine. We will use **Local file** option to upload an vNIOS OVA image is which is present locally in the system. Enter a name for this

image and click upload.

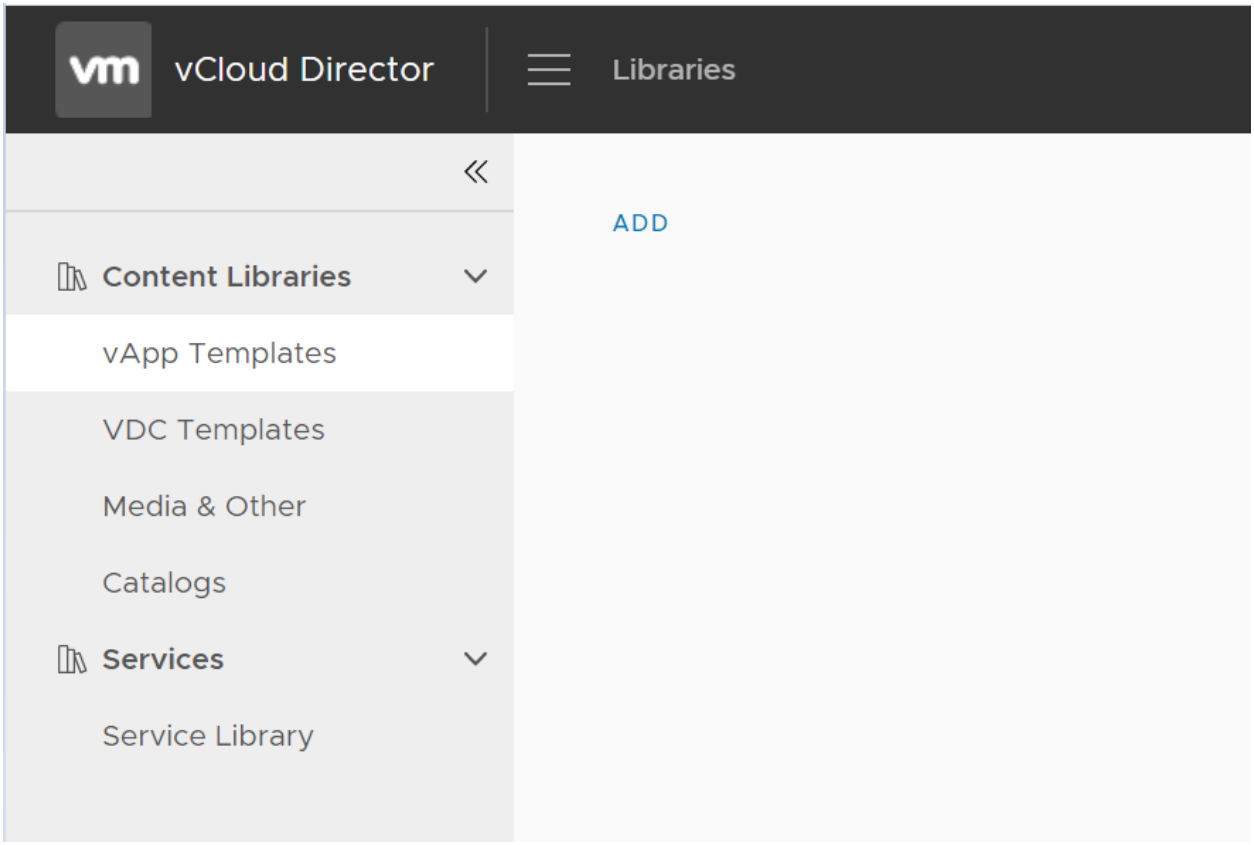


6. You should see a **Transfer progress** bar showing disc upload progress.

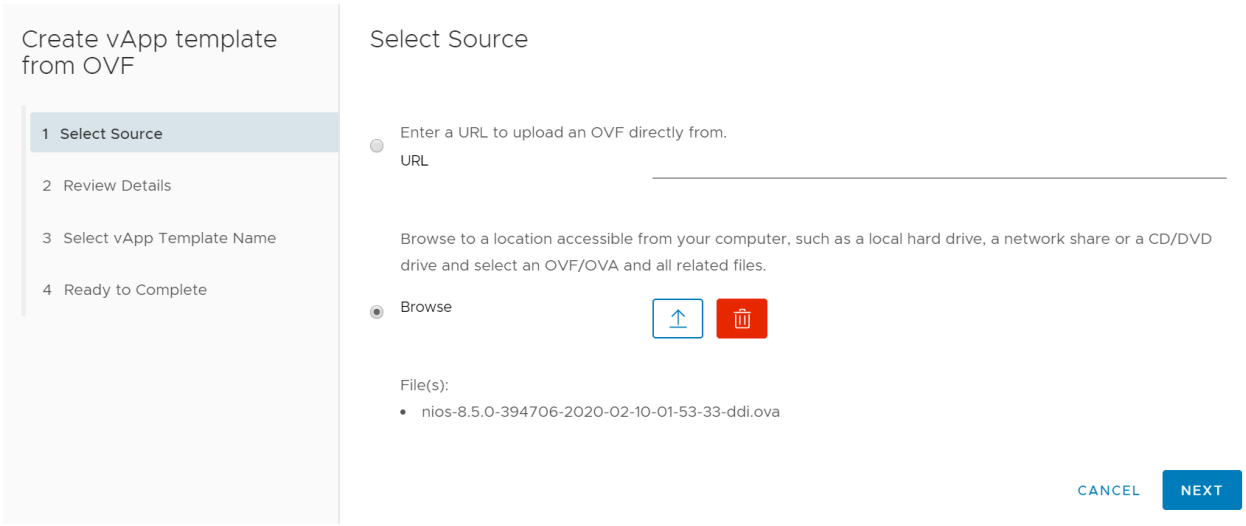


Note: If Transfer Progress bar gets stuck at 0% and does not proceed, try following steps.

7. Login to the VCD HTML5 interface by entering following URL https://***vcd_fqdn_or_ip_address***/tenant/***orgnanization_name***
8. Click on the 3 horizontal lines next to the **Datacenters** and select **Libraries**
9. Navigate to **vApp Templates** option, present in the left side and click on **Add**.

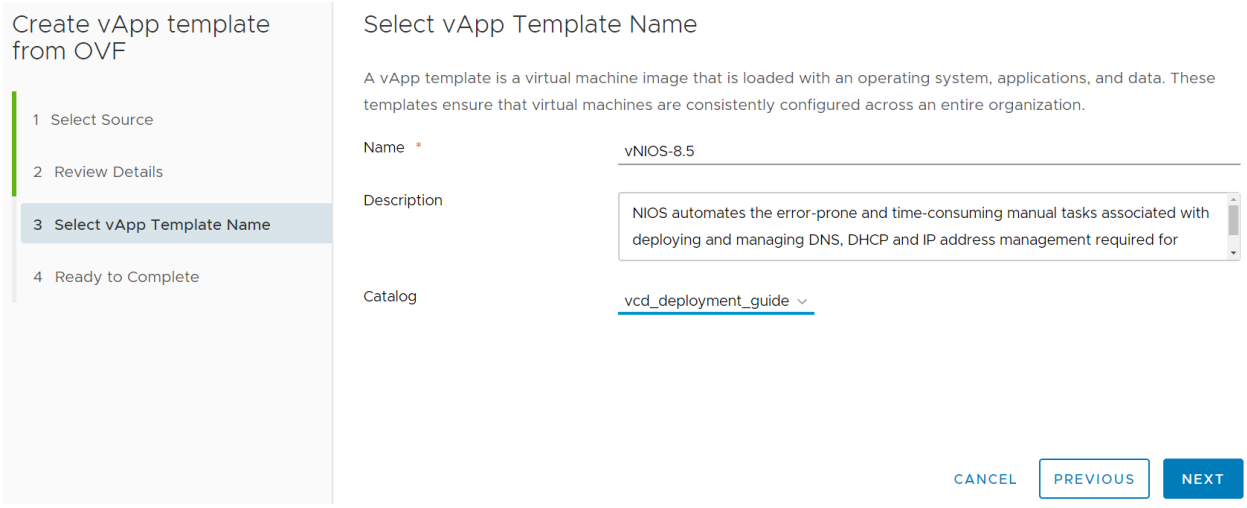


10. vNIOS OVF file can be either uploaded from a URL or from the local machine. We will use **Local file** option to upload an vNIOS OVF file is which is present locally in the system. Once OVF file is selected click on NEXT.

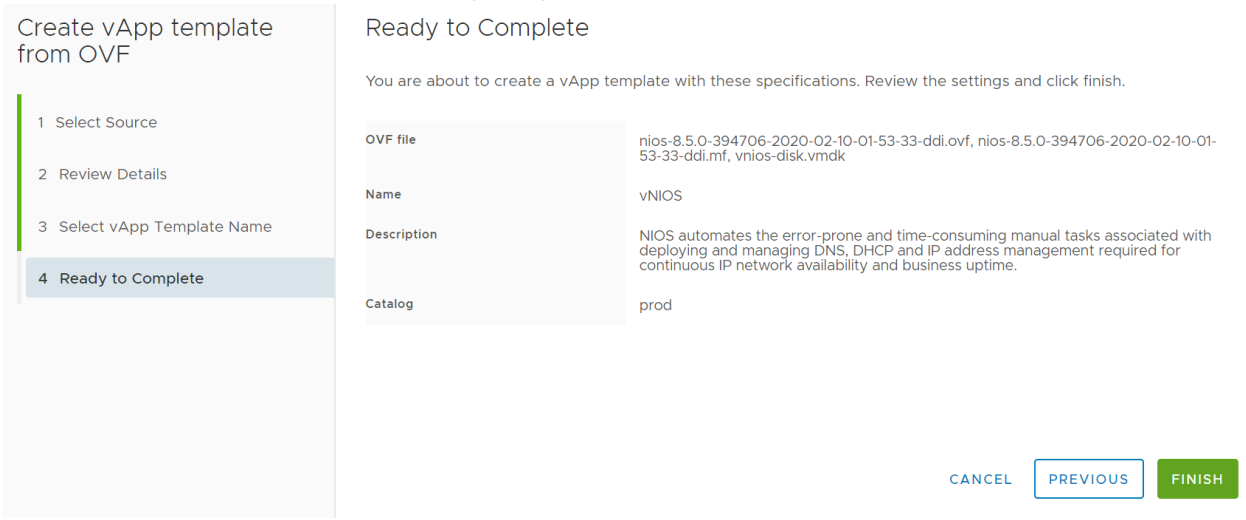


11. You will presented with **Review Details** section. Review the details and click on **NEXT**.

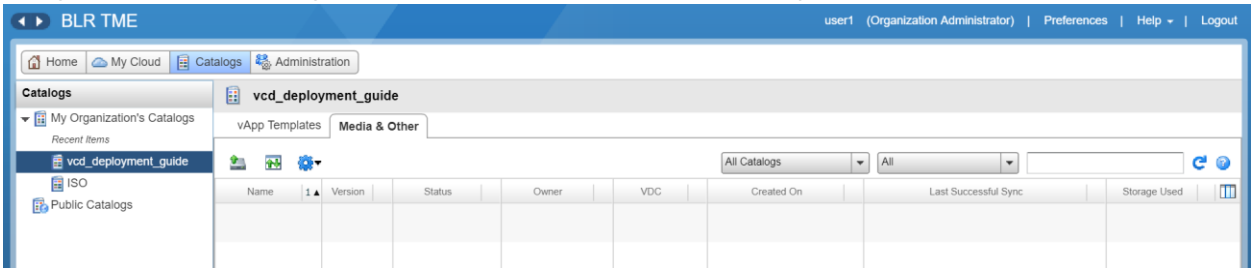
12. Enter a name for this vApp template and select the catalog from the dropdown menu.Click on **NEXT**.



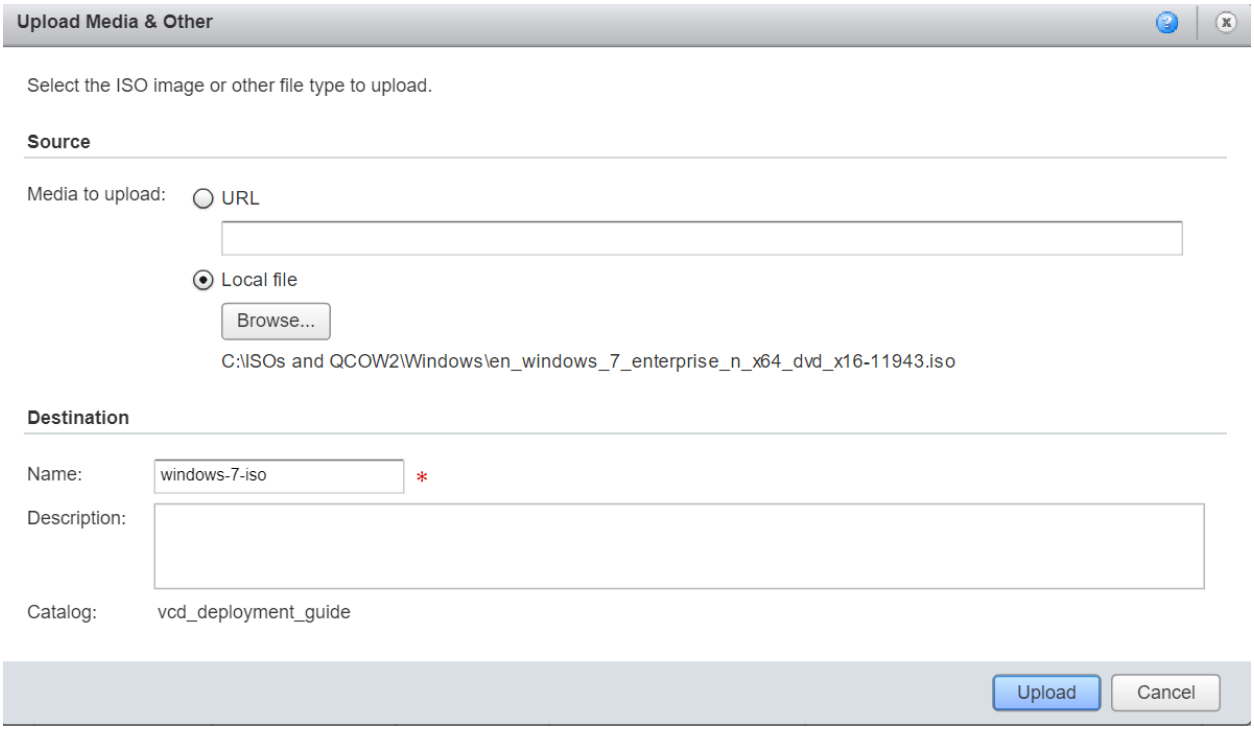
13. Click on **FINISH** to start the vNIOS OVF upload process.



14. You can view the task bar to monitor the file upload progress.
15. Navigate back to the catalog view and select **Media & other**, click on upload icon.

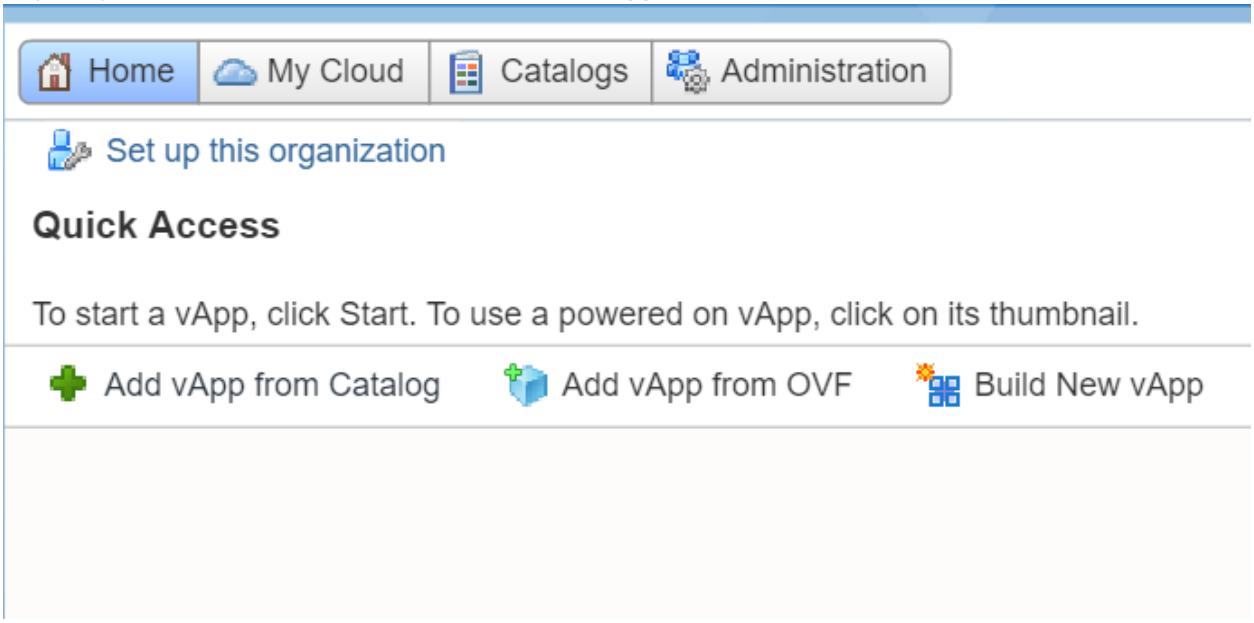


16. Browse for the locally downloaded windows iso file, by selecting Local file. Enter a name and click on **Upload**.

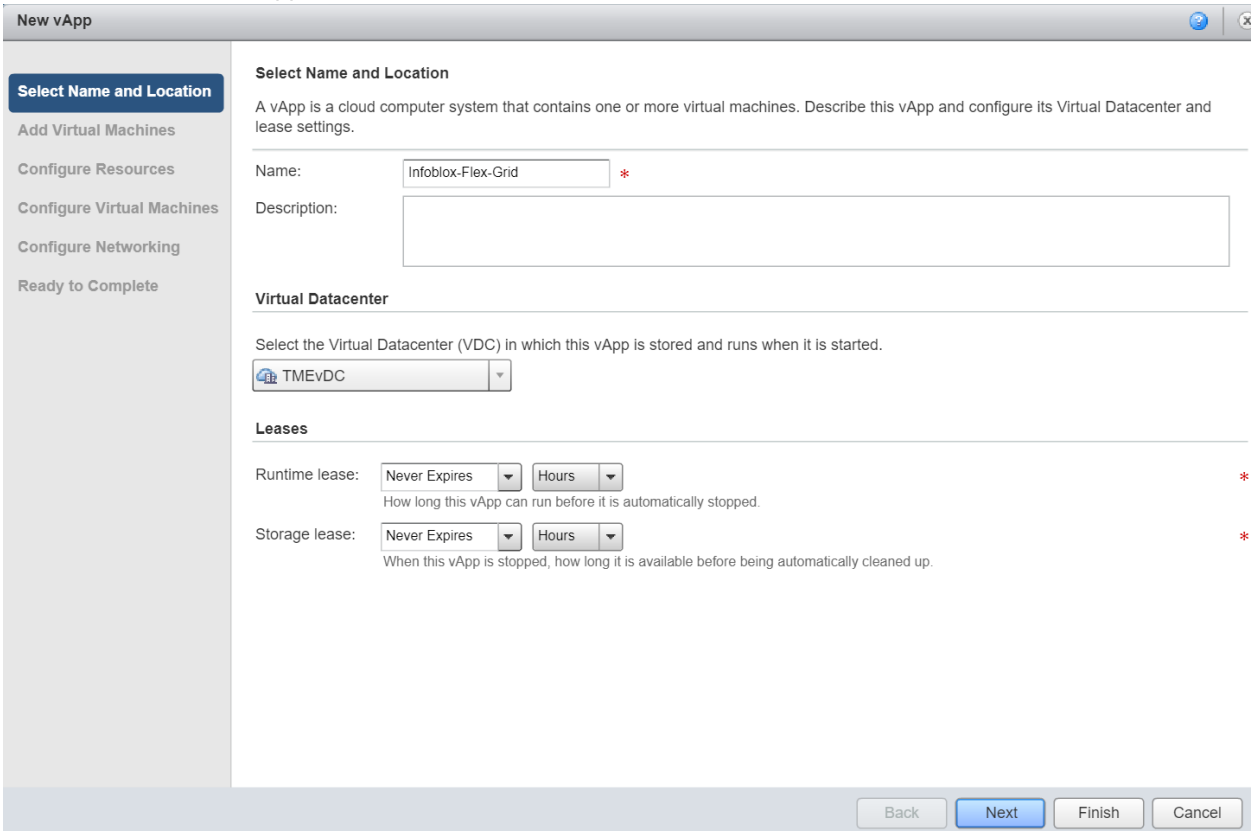


Provisioning a vApp

1. Login to your vcd enviroment, and select **Build New vApp**

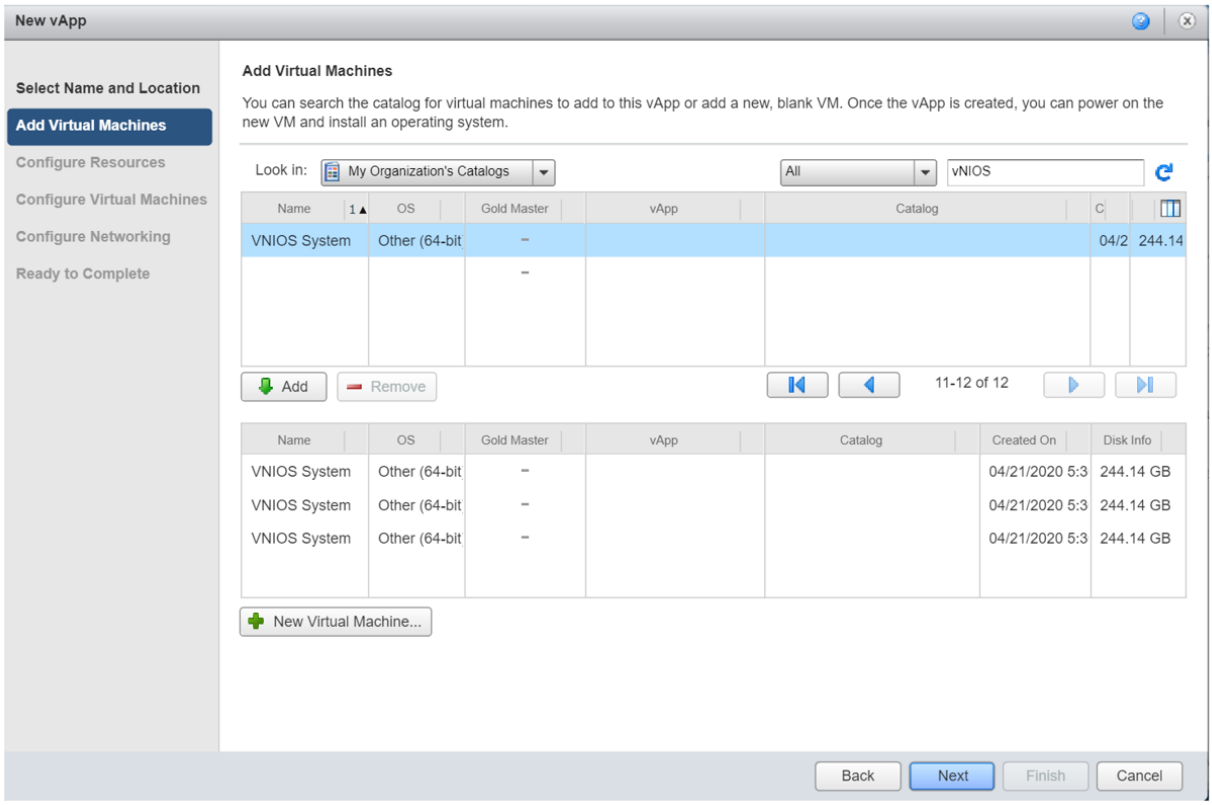


2. Enter a name for this vApp and click on next .



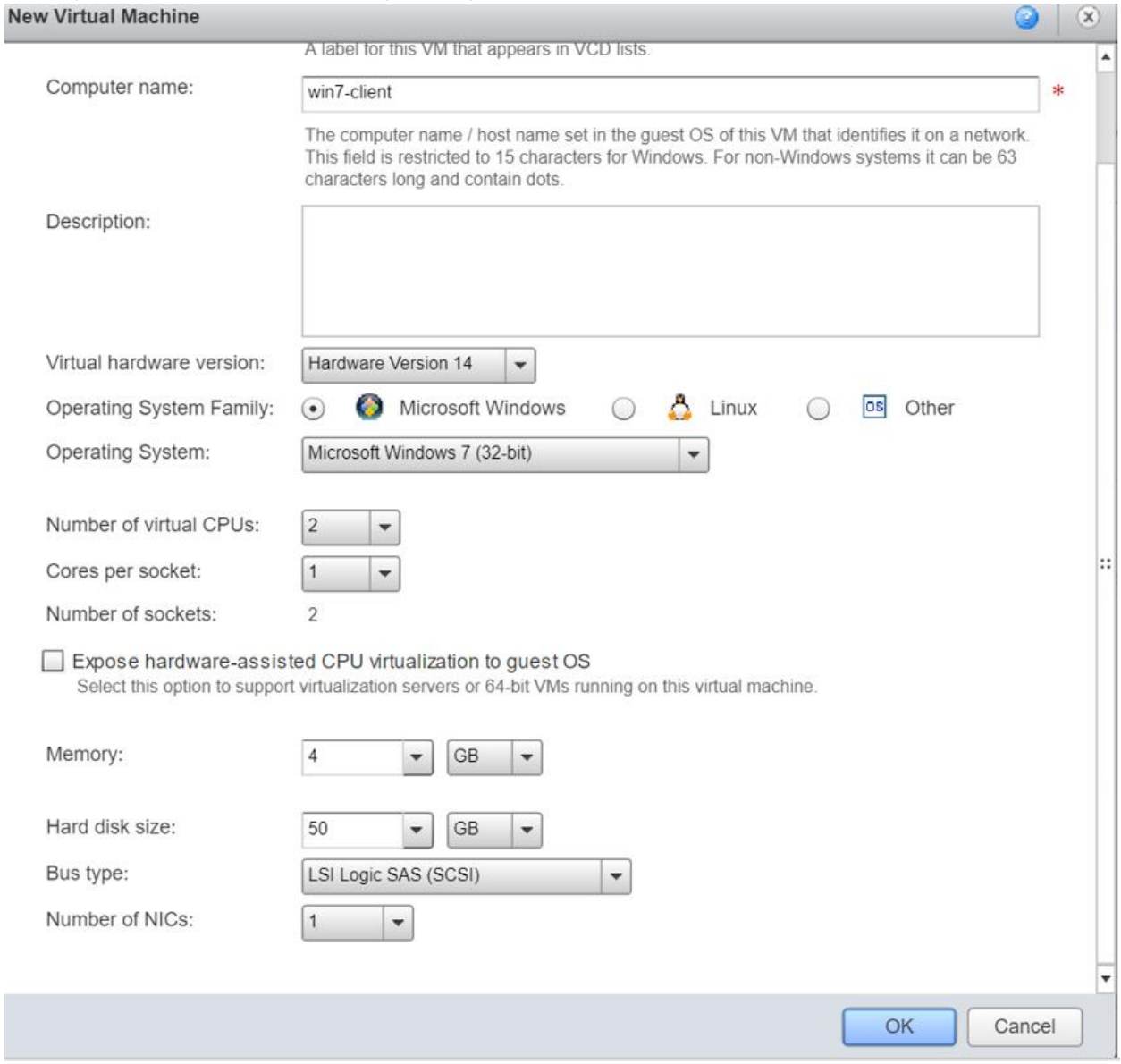
3. Make sure you select the appropriate catalog from the **Look in** drop down menu and search for the vNIOS ova file which you uploaded in the previous step. Click on **Add** option 3 times to add 3

instances of vNIOS ova image into the vApp. **Do not click on Next.**

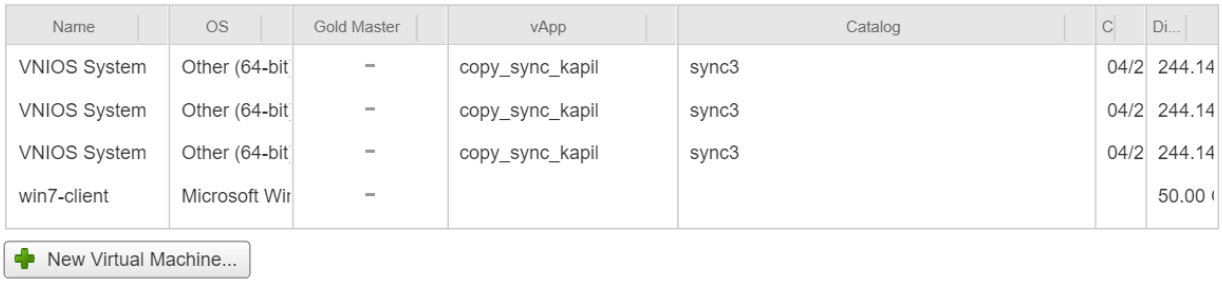


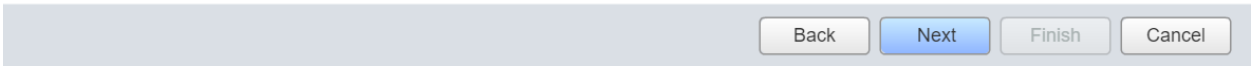
4. From the same wizard, click on **New Virtual Machine** option(to install a windows based client). Enter the name for this VM, and select correct **Operating System**. Give atleast 2 virtual CPUs,4 GB

memory, and 50 GB hard disk for a lag free experience. Click **OK**.



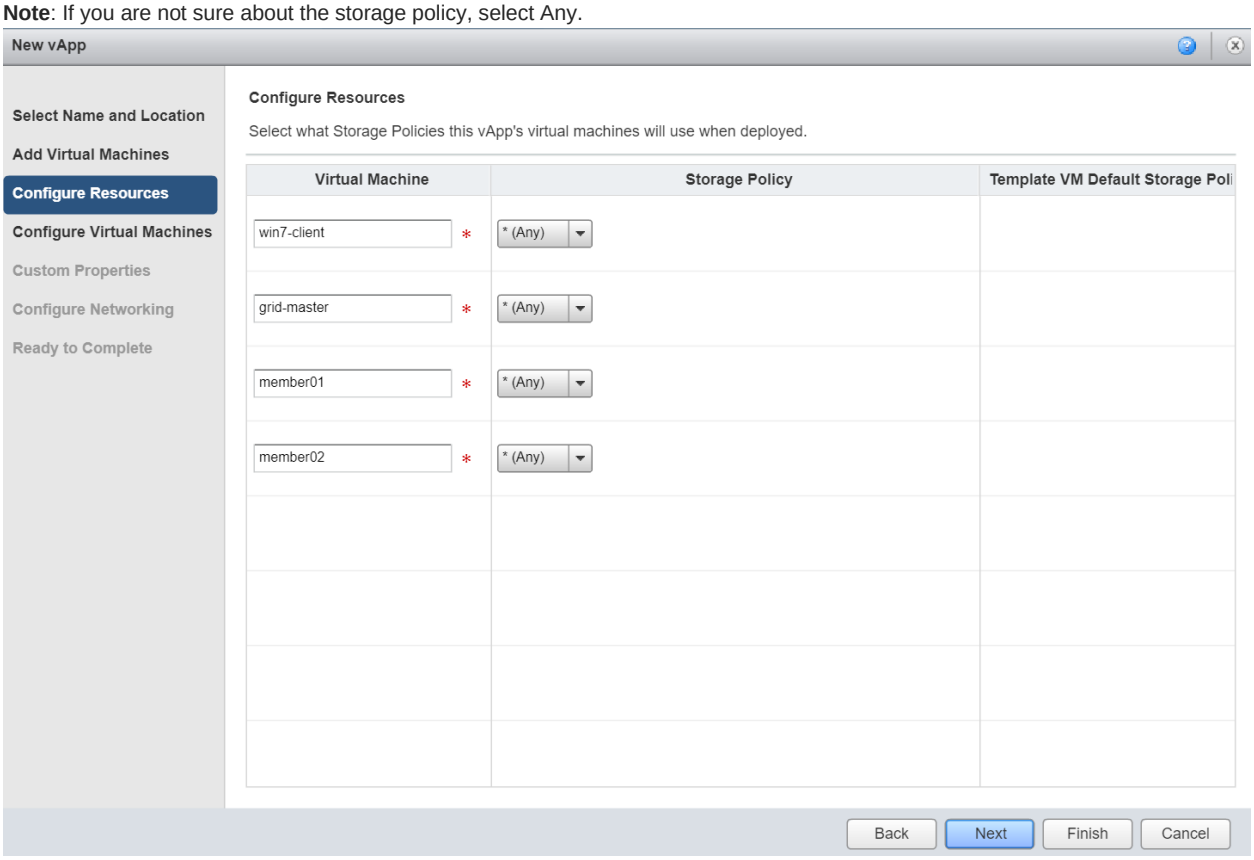
5. You should now see 4 VMs(3* vNIOS VMs and 1* Windows VM) which will be deployed in the vApp. Click on **Next**.





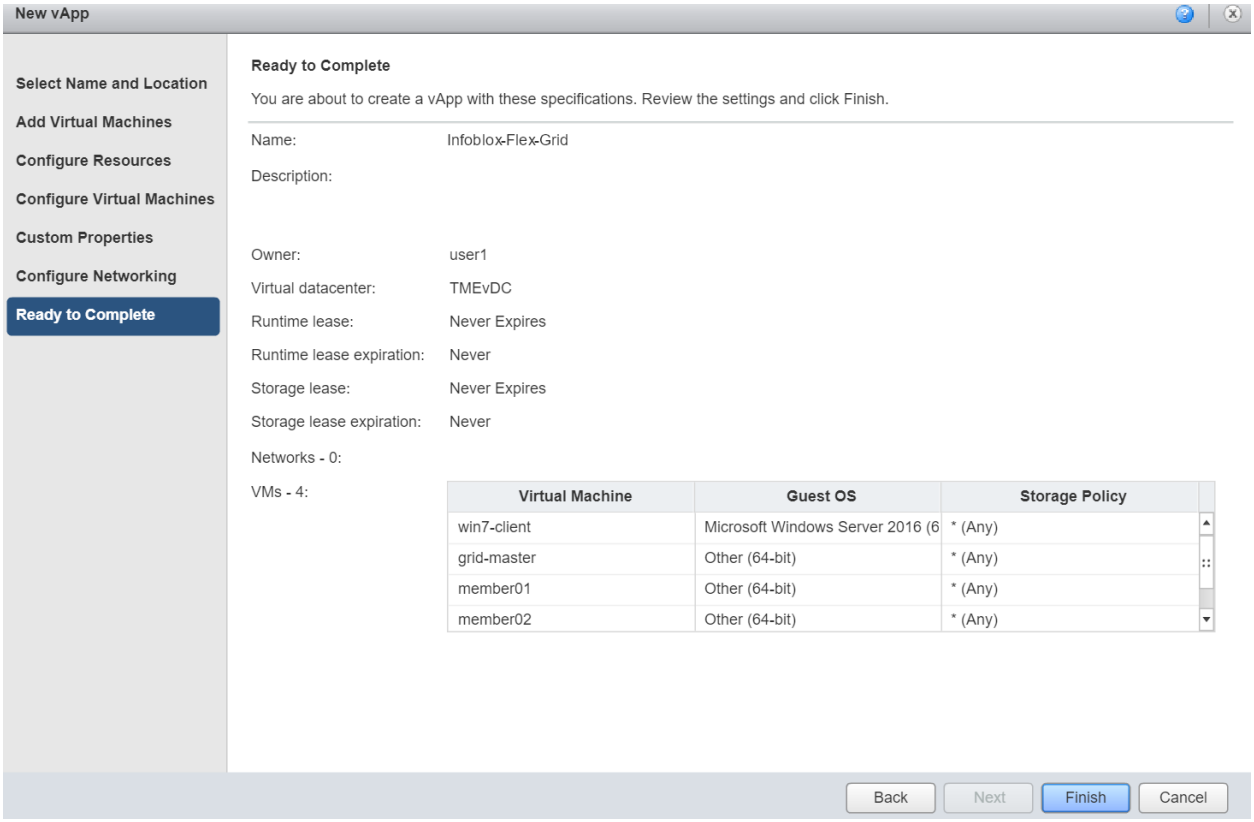
6. Enter virtual machine names and select the appropriate **Storage Policy** from the **Configure Resources** wizard and click on **Next**.

Note: If you are not sure about the storage policy, select Any.

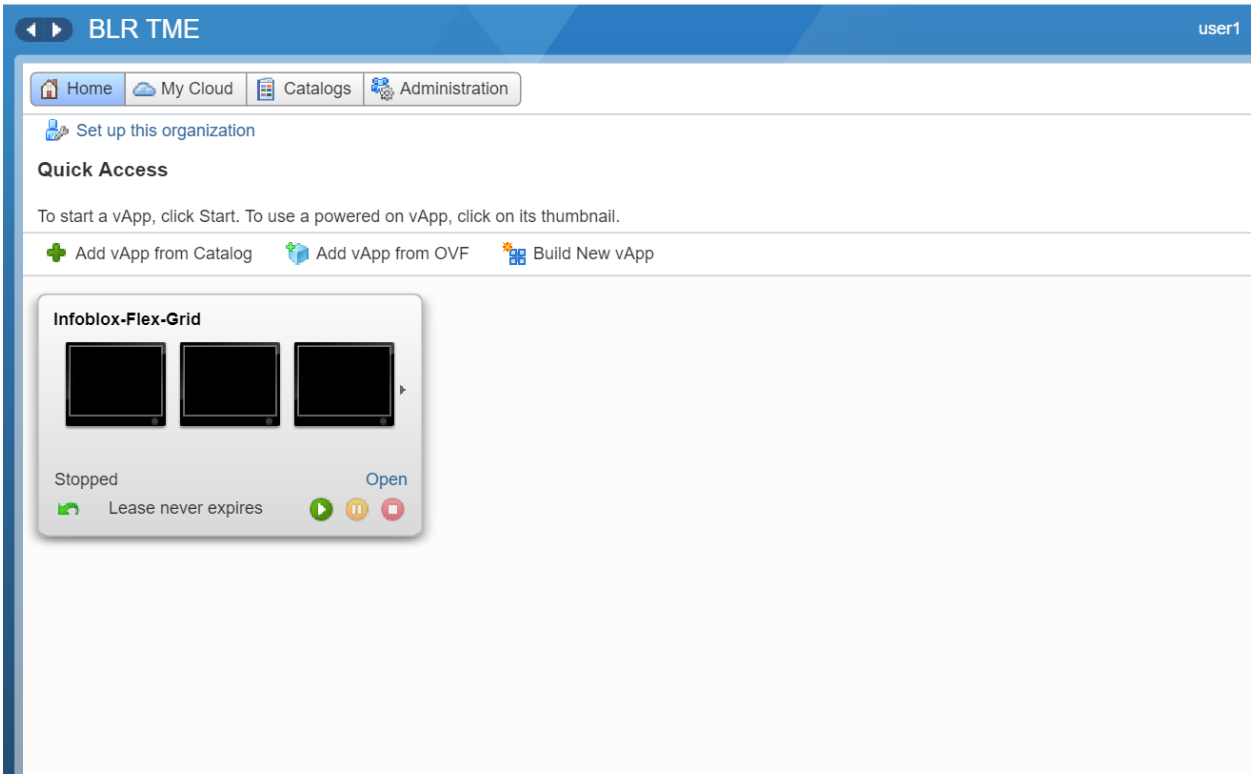


7. You will be presented with **Configure Virtual Machines** wizard, leave all the settings to default and click on **Next**.
8. You will be presented with **Custom Properties wizard**, leave all the settings to default and click on **Next**.
9. You will be presented with **Configure Networking**, leave all the settings to default and click on **Next**

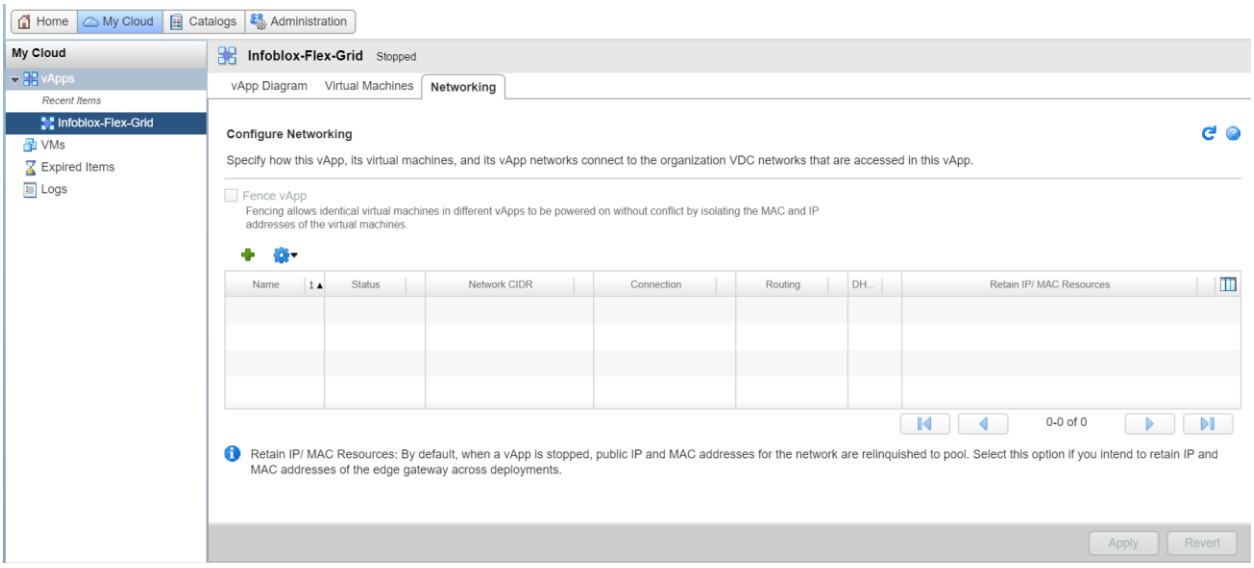
10. You can review the settings from the Ready to Complete wizard. Click on **Finish** to start vApp provisioning.



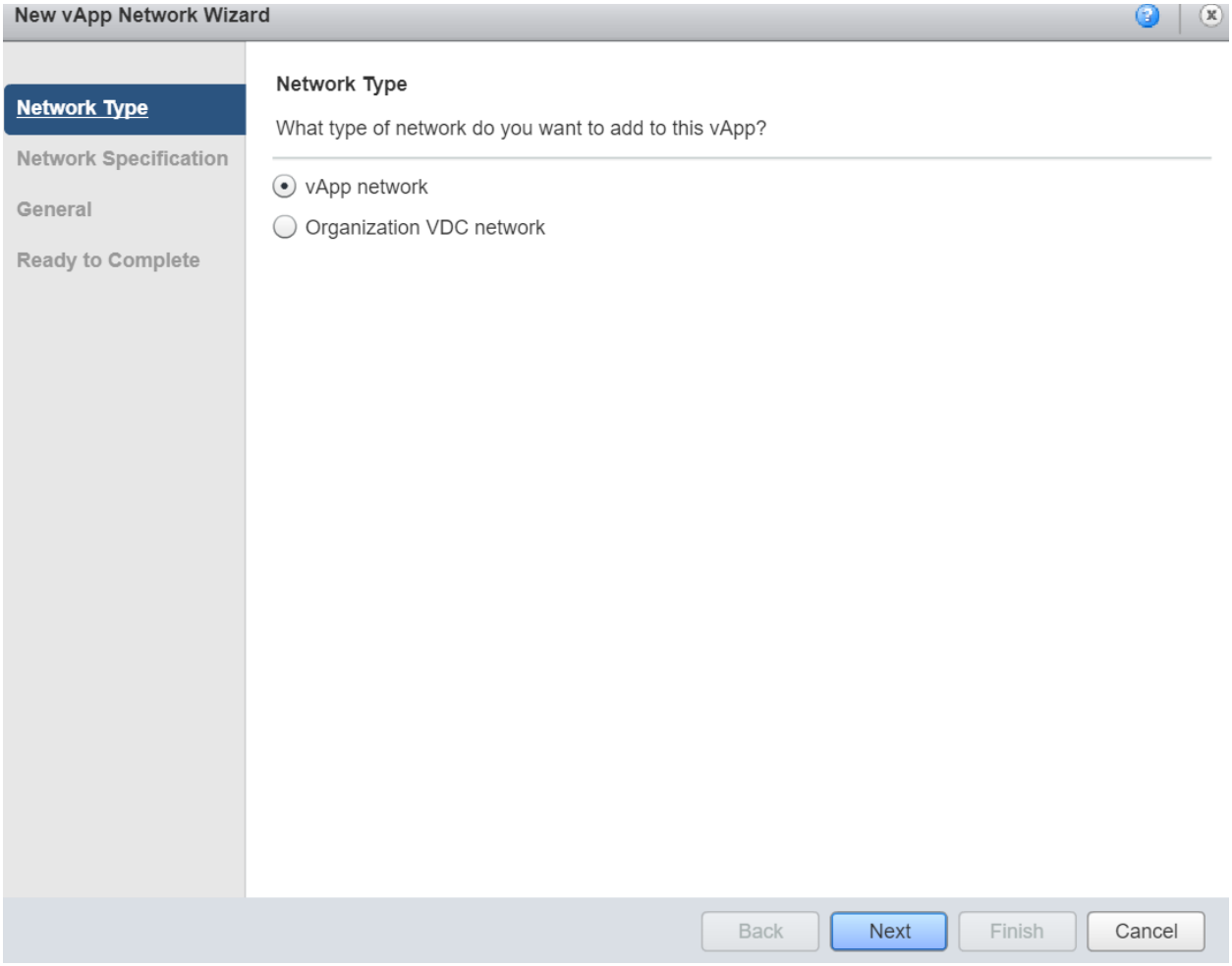
11. vApp provisioning will take 2-3 minutes. Double click on the vApp to access its resources.



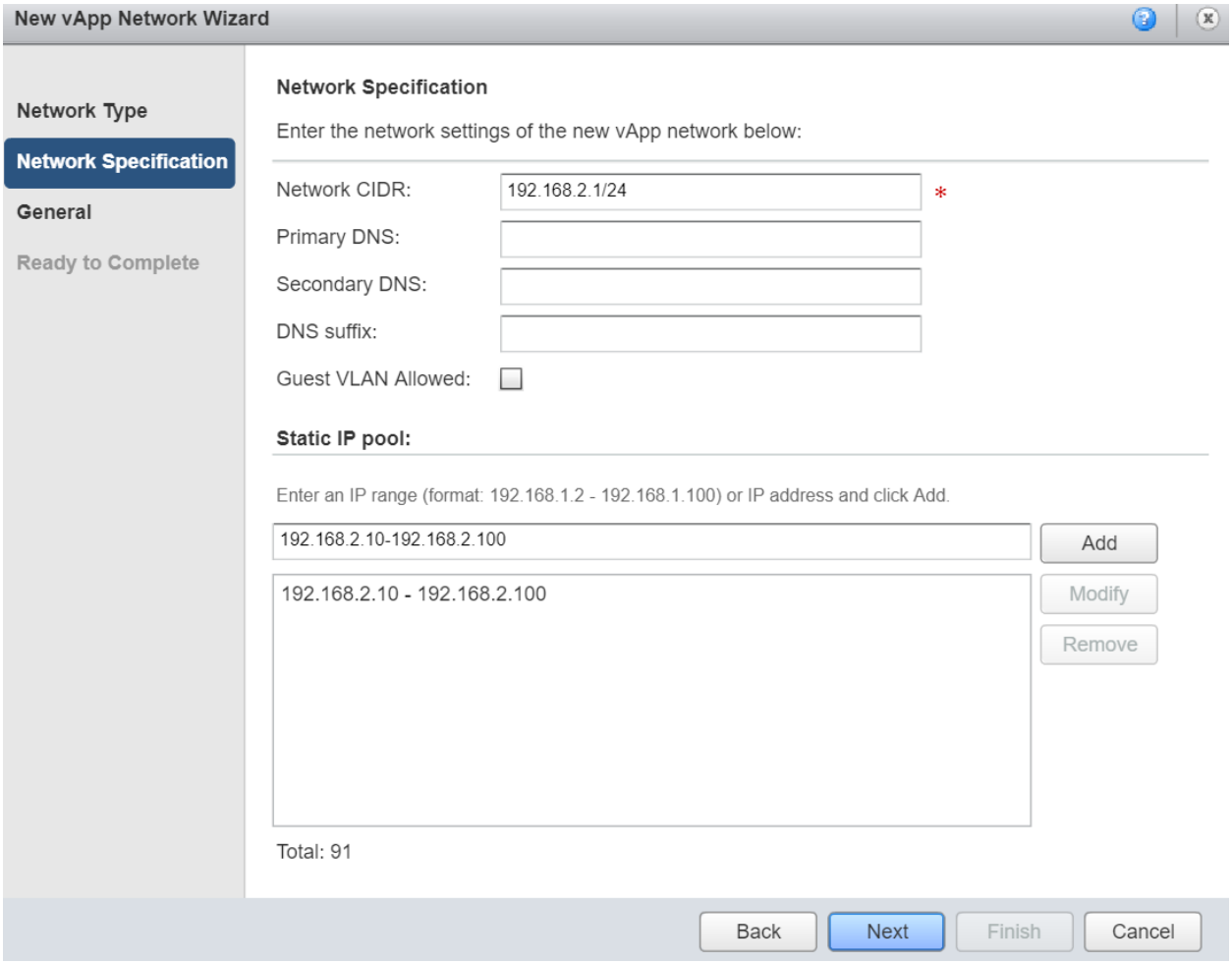
12. First thing which you will notice that the vApp does not have any networks. Click Networking tab to start creating networks for Infoblox-Flex-Grid. Click green plus icon to start creating networks.



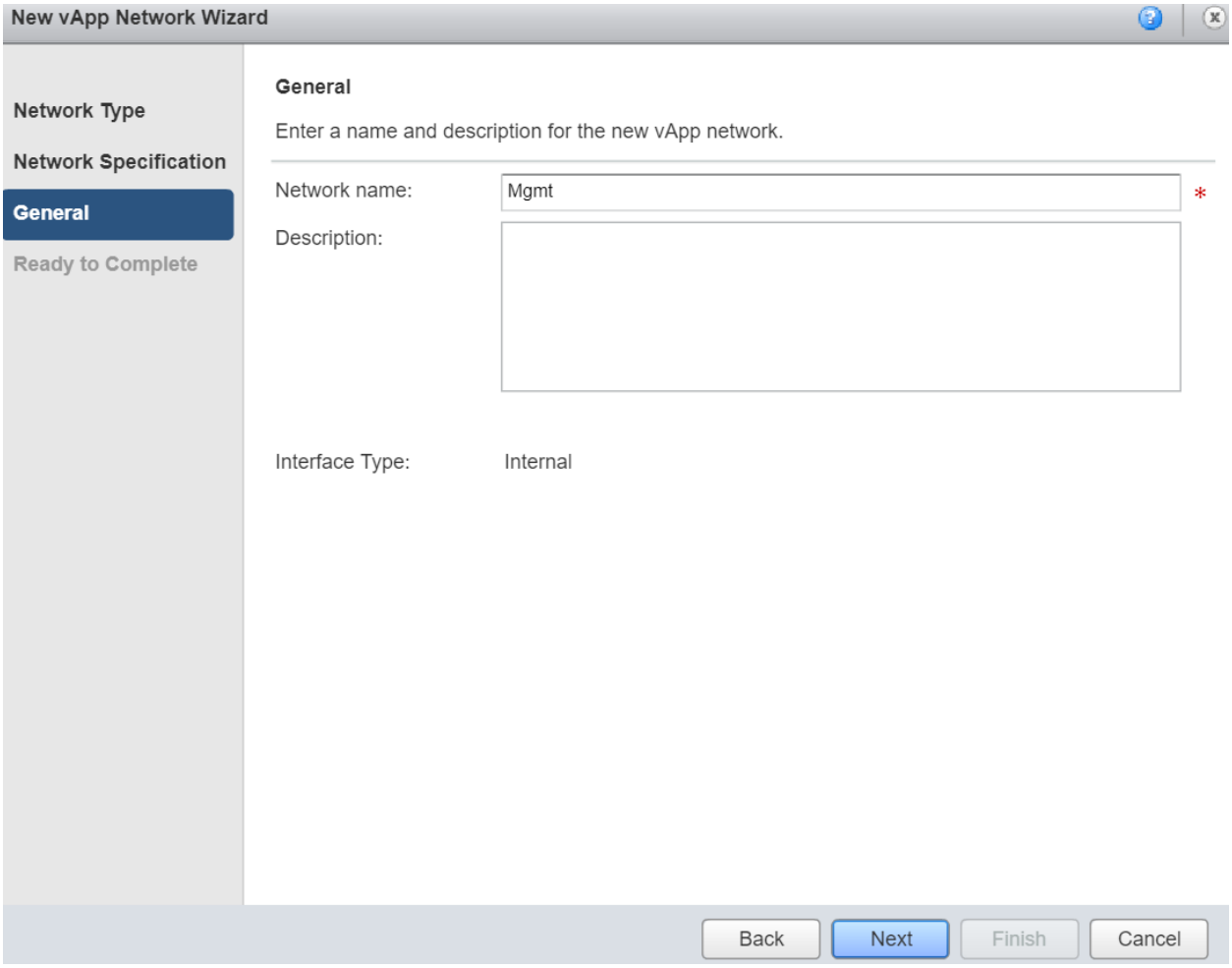
13. Since we are creating vApp network, select **vApp network** and click on Next.



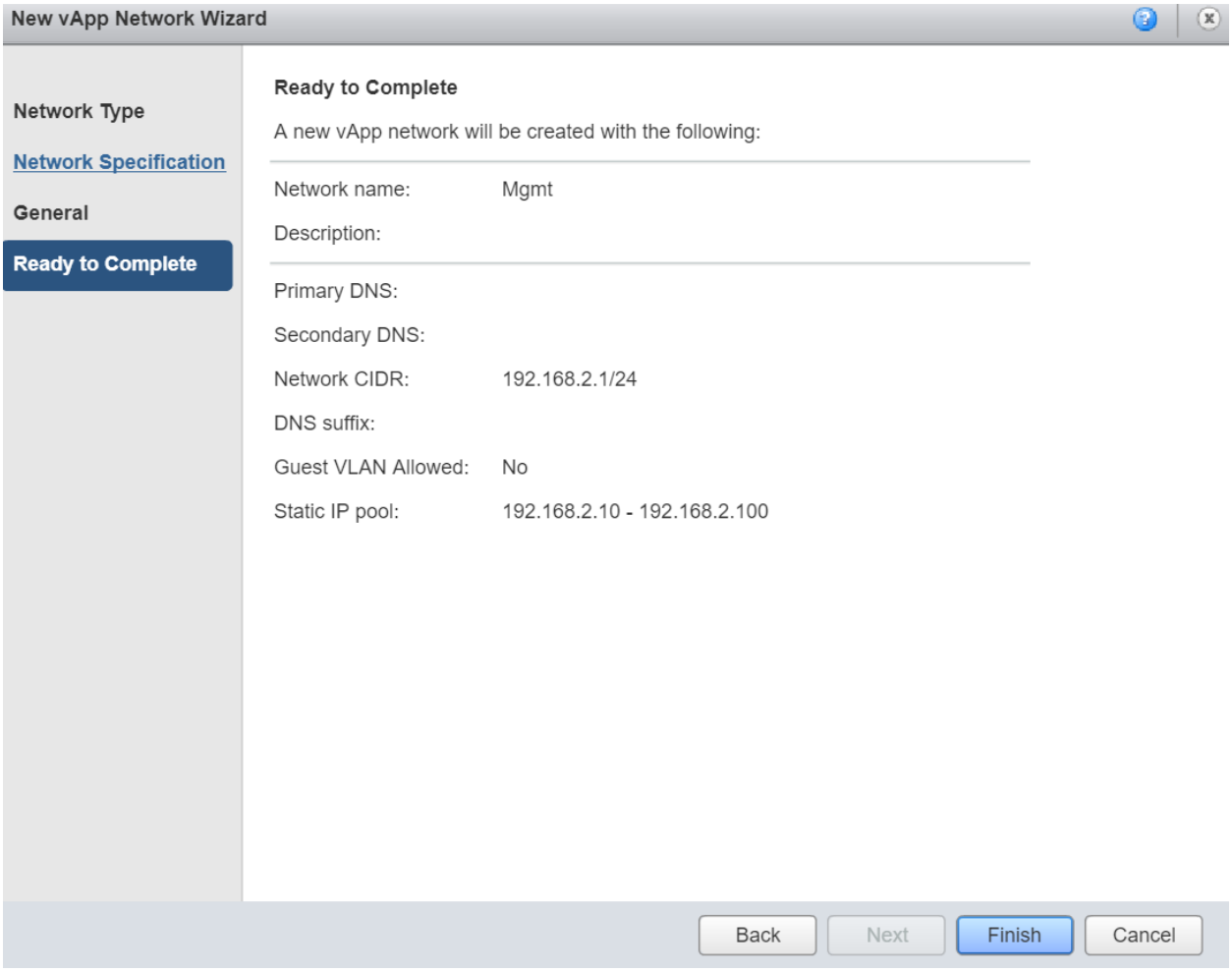
14. Enter network CIDR and network related inputs as depicted in the screen shot. Enter a **static IP pool** range and click on **Add**. Click on **Next** to go to the next screen.



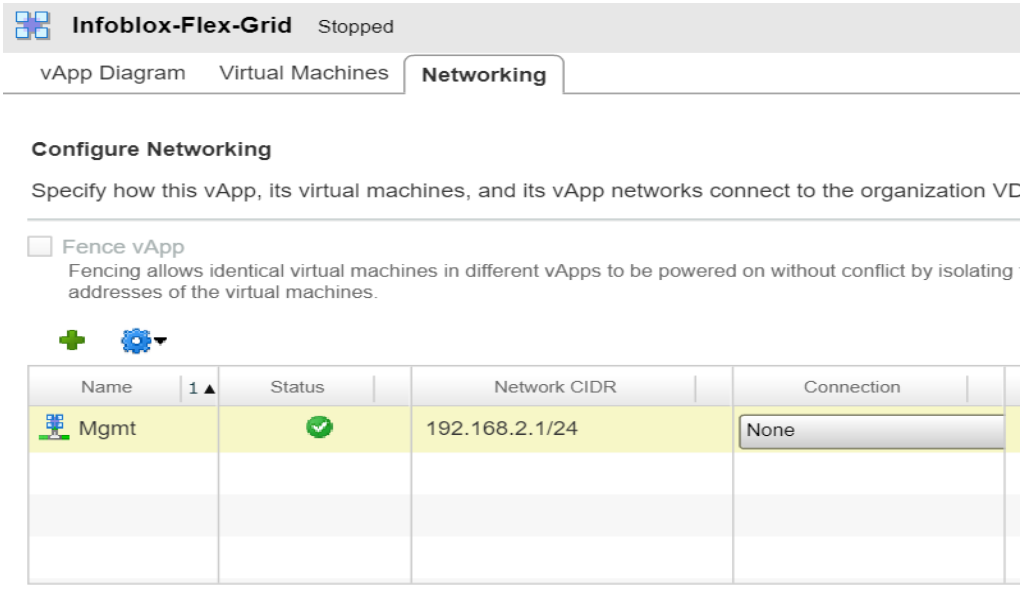
15. Enter a **Network name** for this network. Click on **Next**.



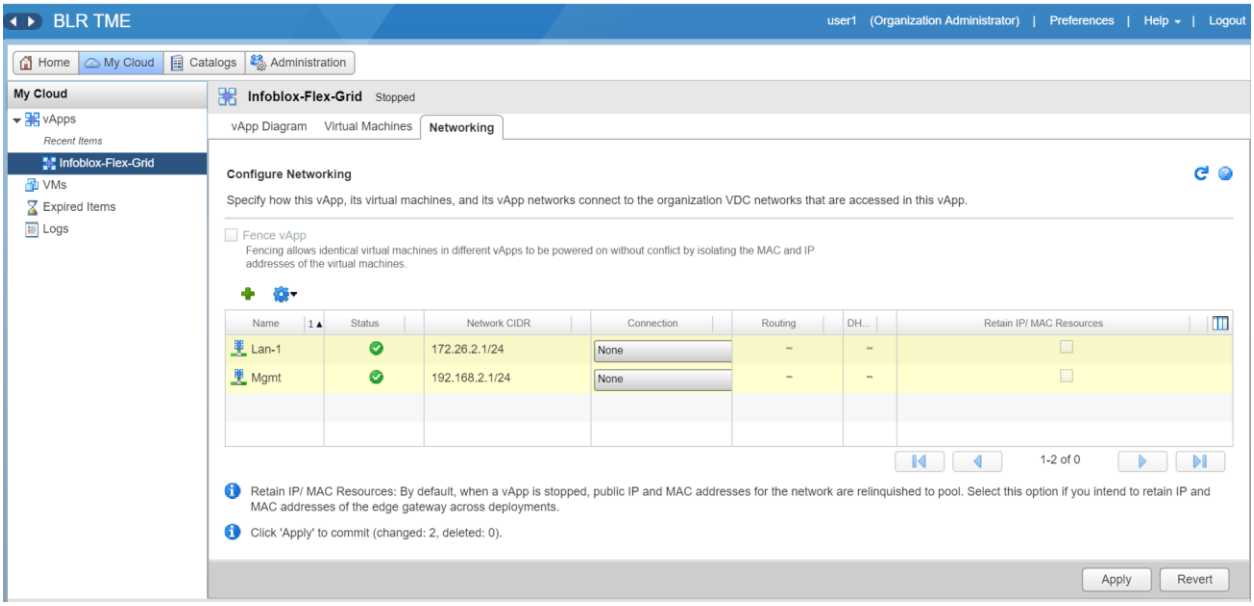
16. You can review the network details. If everything looks good, click on **Finish**.



17. Click on **Add** network icon again to create second network.

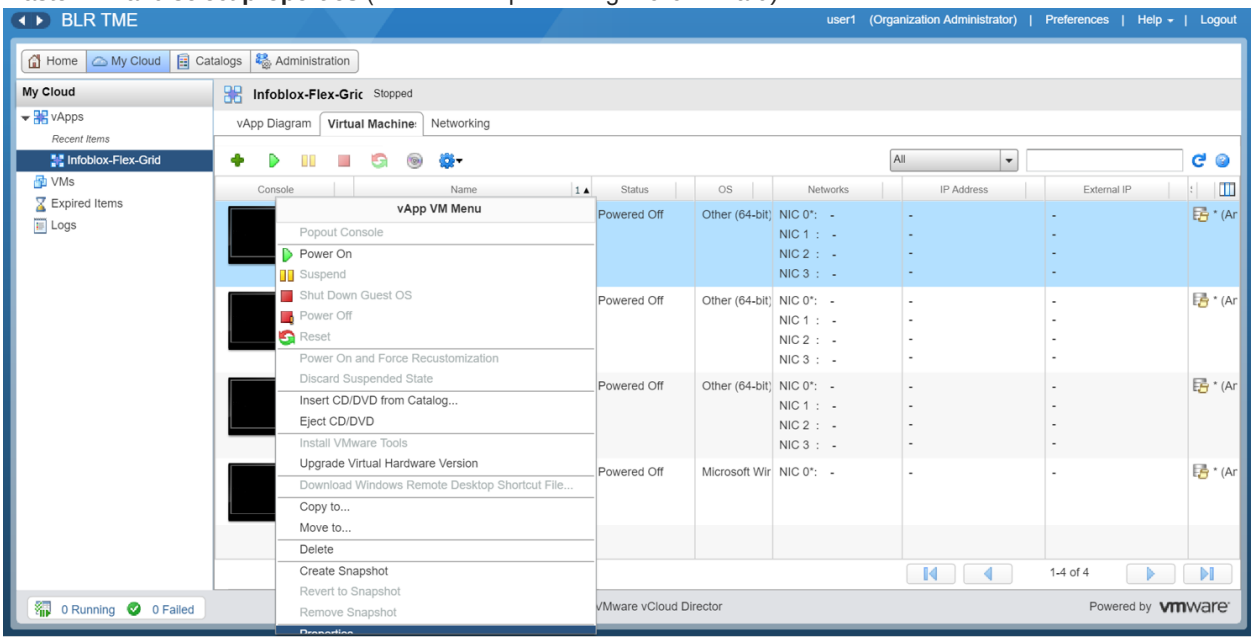


18. Follow the same steps and enter a new network CIDR, static ip range and a network name**(Lan-1).**
19. Please ensure to click on **Apply** to implement the changes.



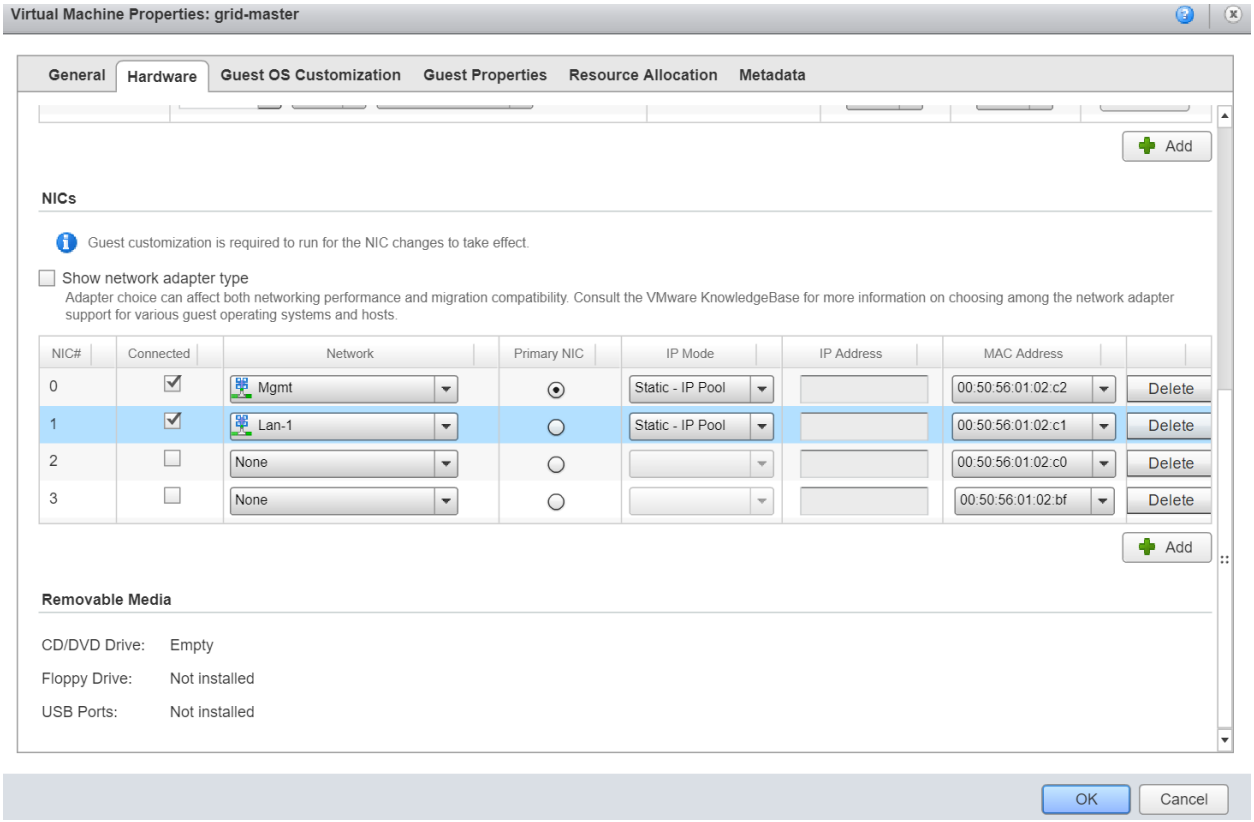
Assigning networks to VMs.

1. Click **Virtual Machines** tab. You will be presented with the list of VMs present in the vApp. Right click grid-master VM and select **properties**.(It's the last option in right-click wizard)

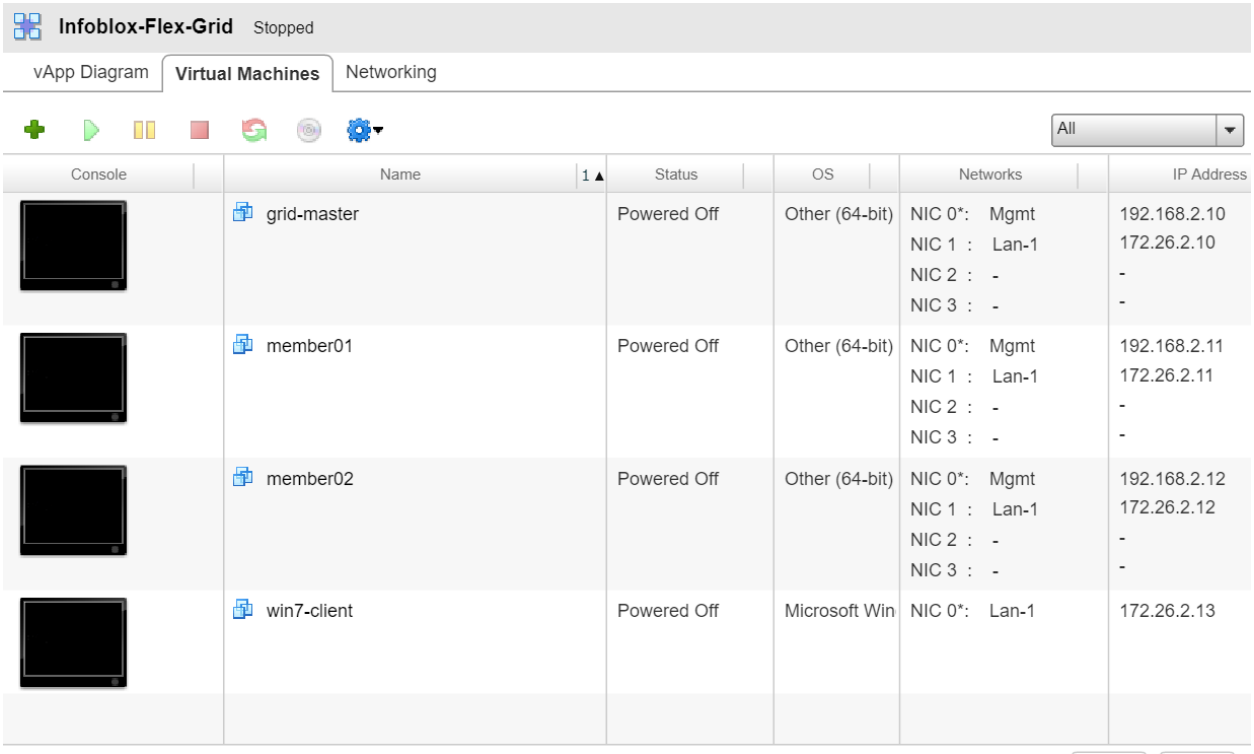


2. You will be presented with Virtual Machine Properties wizard. Navigate to **Hardware** section and scroll down to assign network interfaces. Assign first interface as Mgmt and second interface as Lan-1. Click **OK** to save

changes.

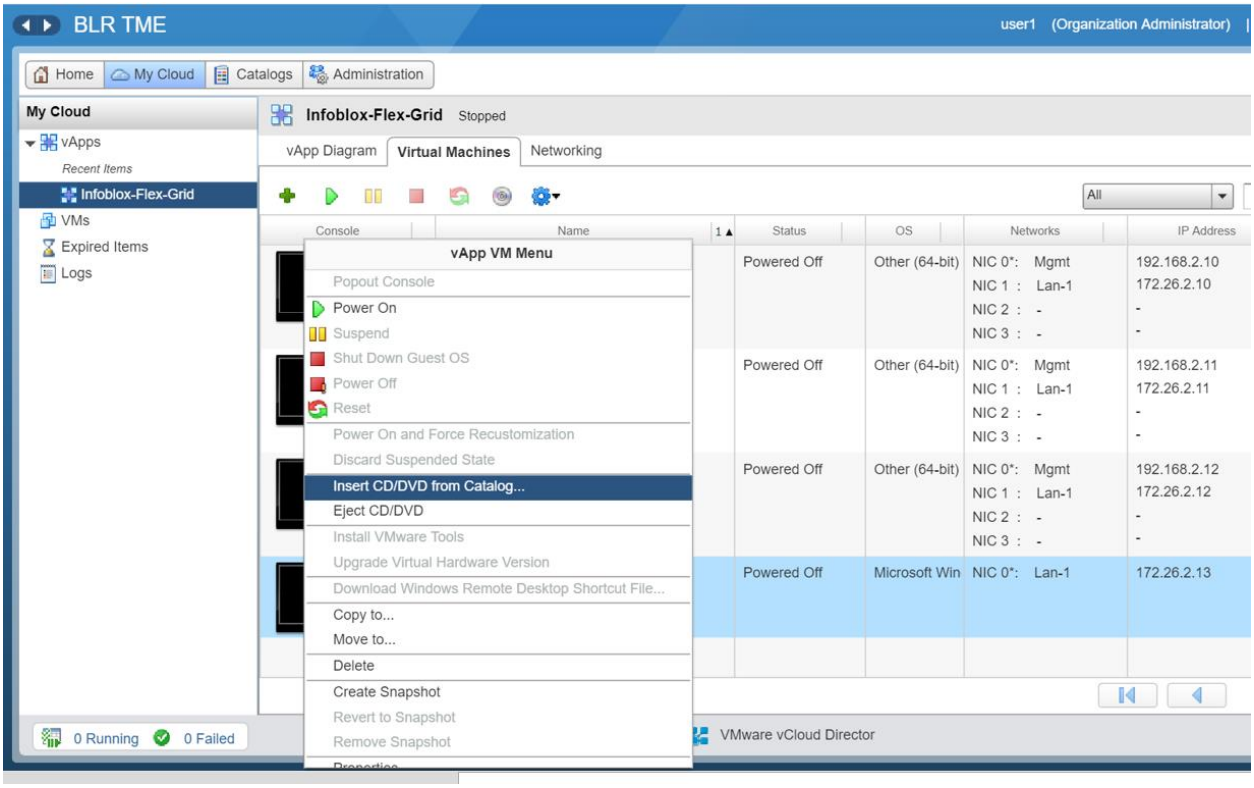


3. Follow the same steps to assign interfaces to member01 and member02 VMs.
4. For win-7 client machine we will be assigning only one interface i.e is Lan-1. After network interface assignment to all VMs, **Virtual Machines** list view will look like this.

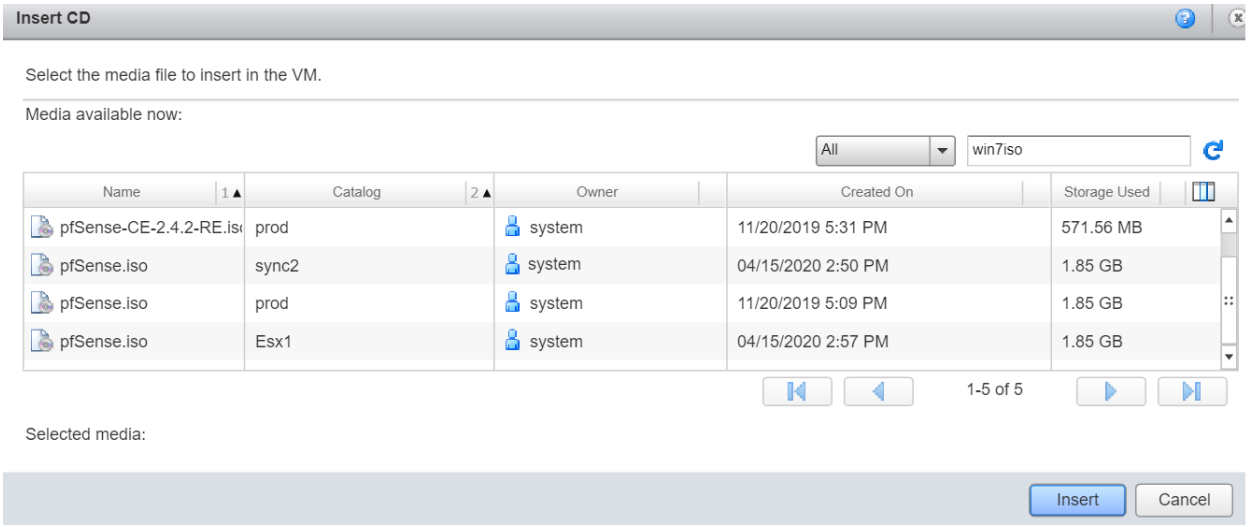


Installing Operating system onto win7-client VM

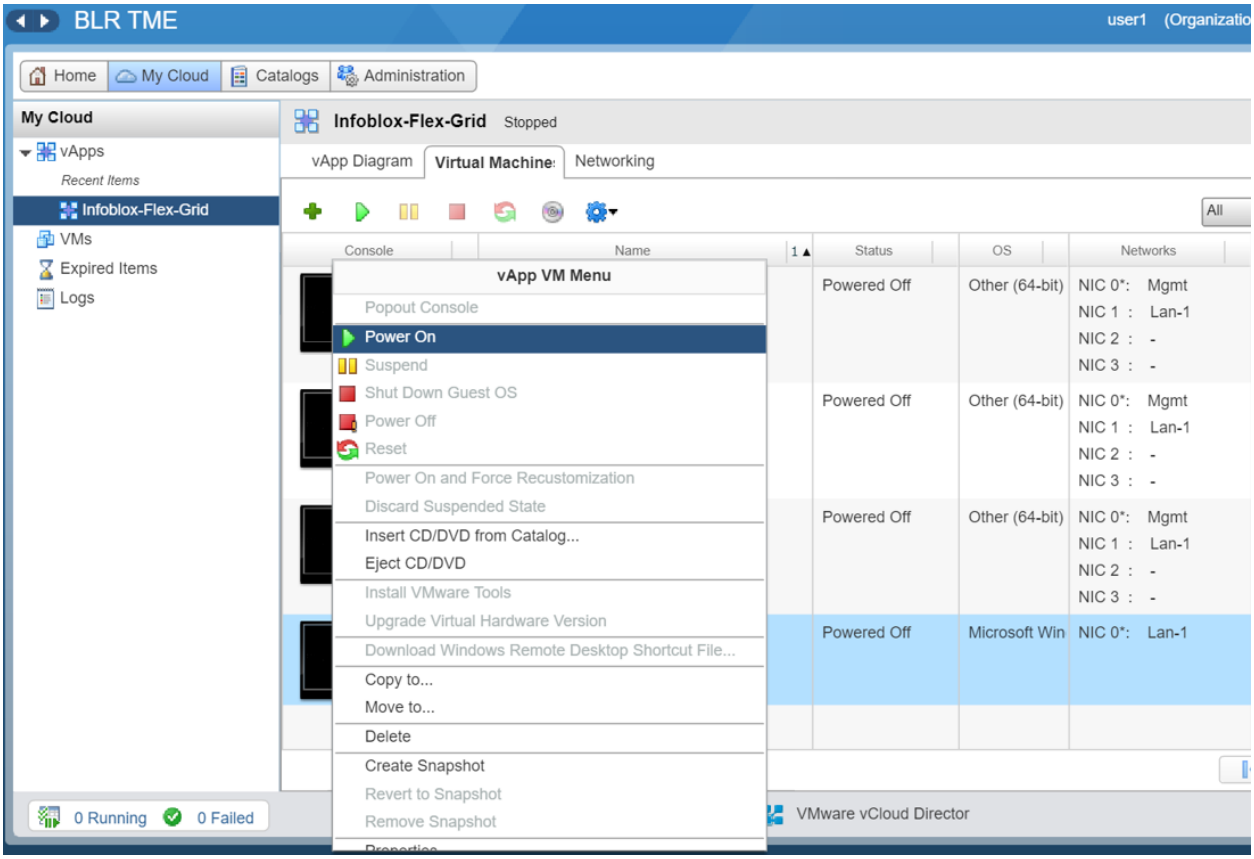
1. From virtual machine tab, right click win7-client VM and select **Insert CD/DVD from Catalog.**



2. Search for win7 iso from the **Insert CD** wizard. Click **Insert** to mount the iso.

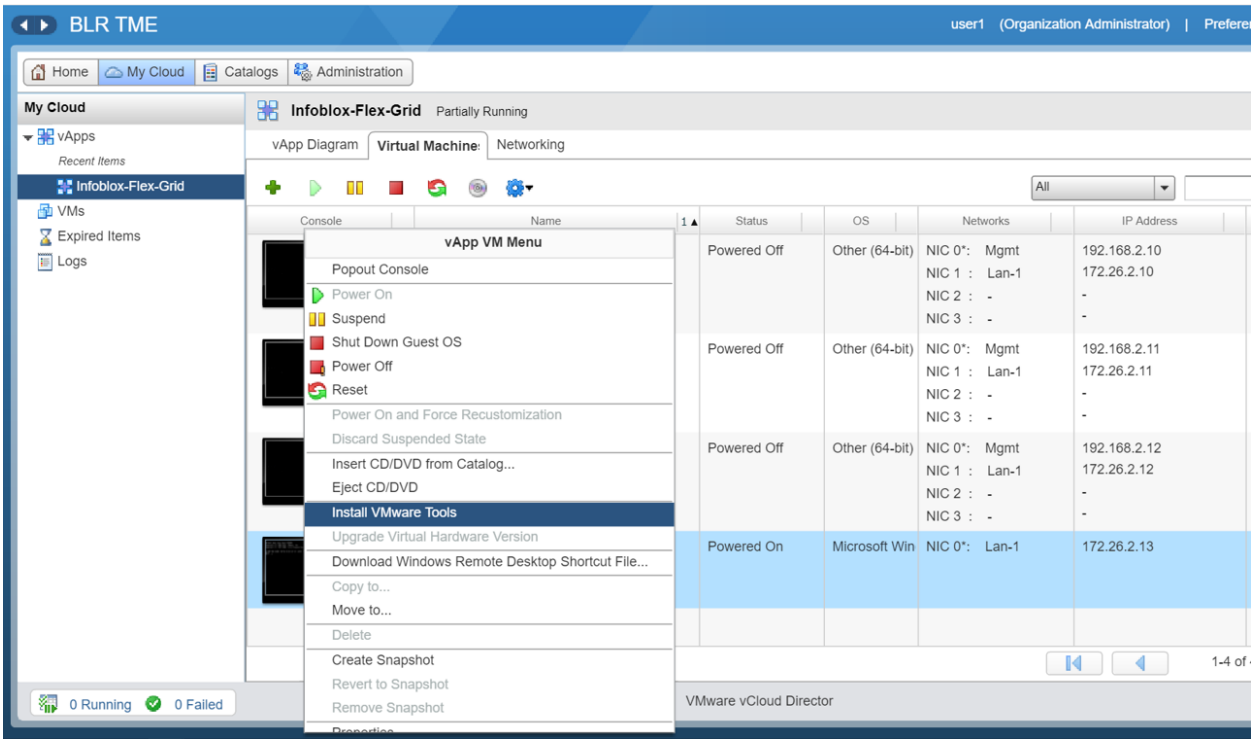


3. Power on win7-client VM by right clicking and selecting **Power on** .



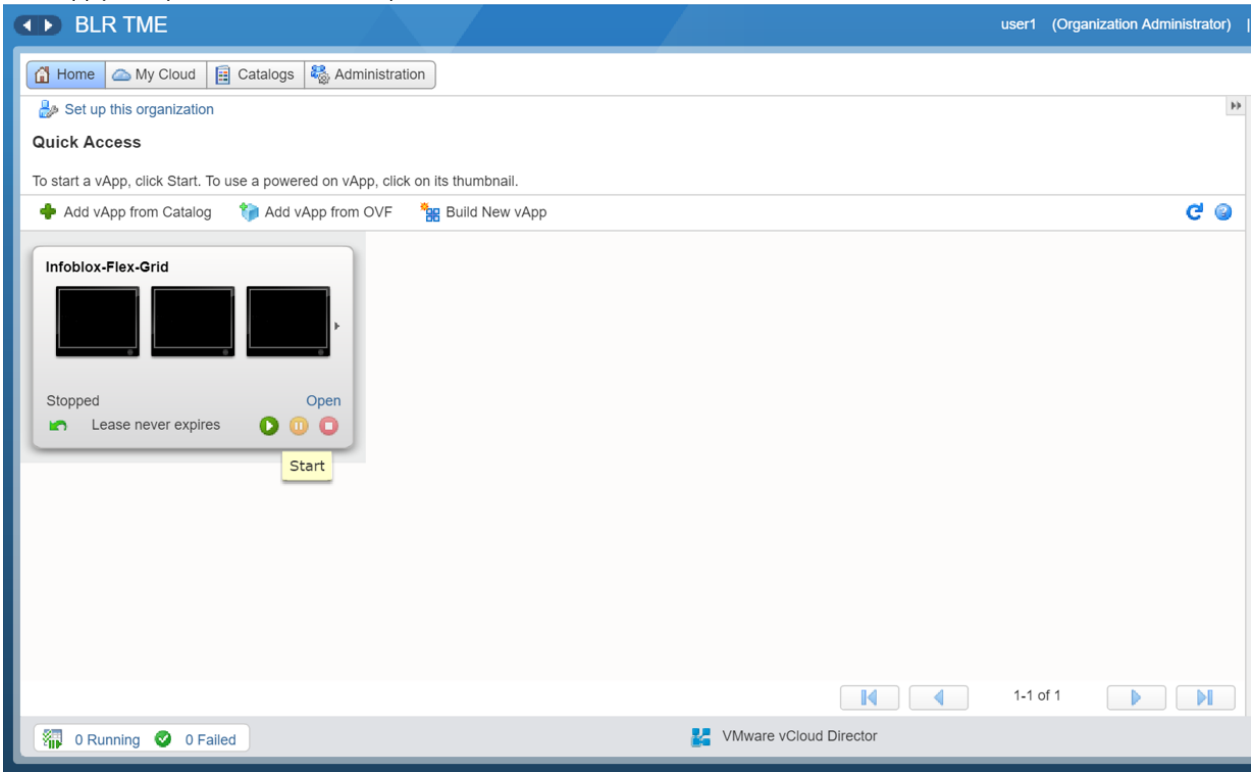
4. To commence win7 OS installation double click the win7-client icon to open up VM console. From here you can install the OS by following the screen instructions.
5. Post OS installation come back to the **Virtual Machine** list view, right click win7-client VM and select **Eject CD/DVD** to unmount iso image.
6. From the **Virtual Machine** list view and right click win7-client VM and select **Install VMware Tools**. This step will mount VMware tools iso to win7-client VM. Login to the win7-client VM and continue with the

installation of VMware tools.

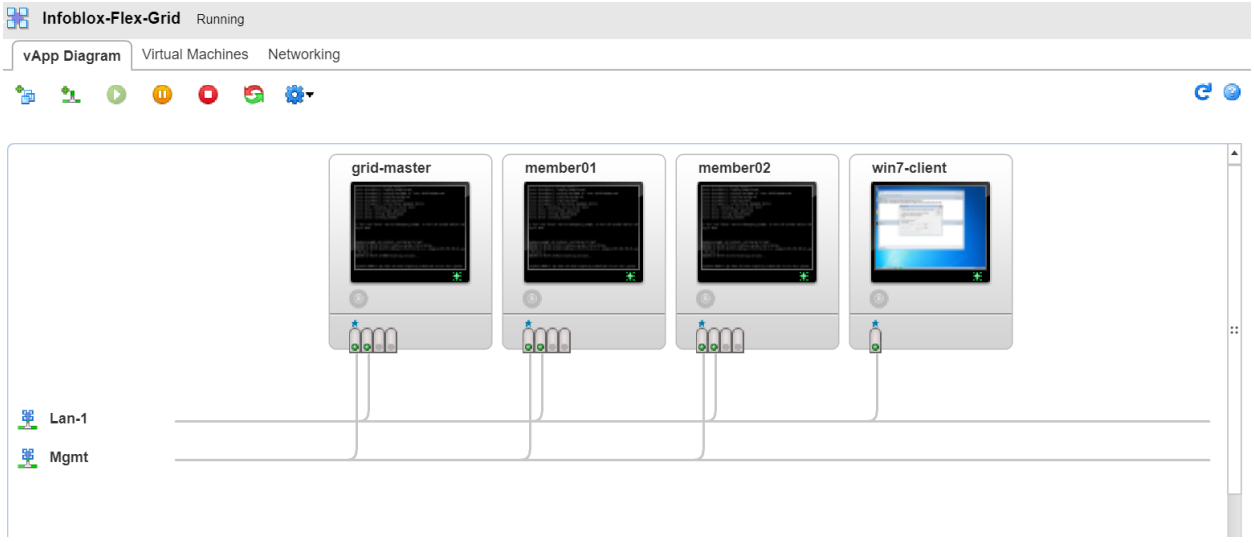


Powering on the vApp, assigning licenses, networking, and setting up the Grid.

1. Navigate to the **Home** option and you will be presented with the vApp view.Click on power on, to power on the vApp(It will power on all the VMs)



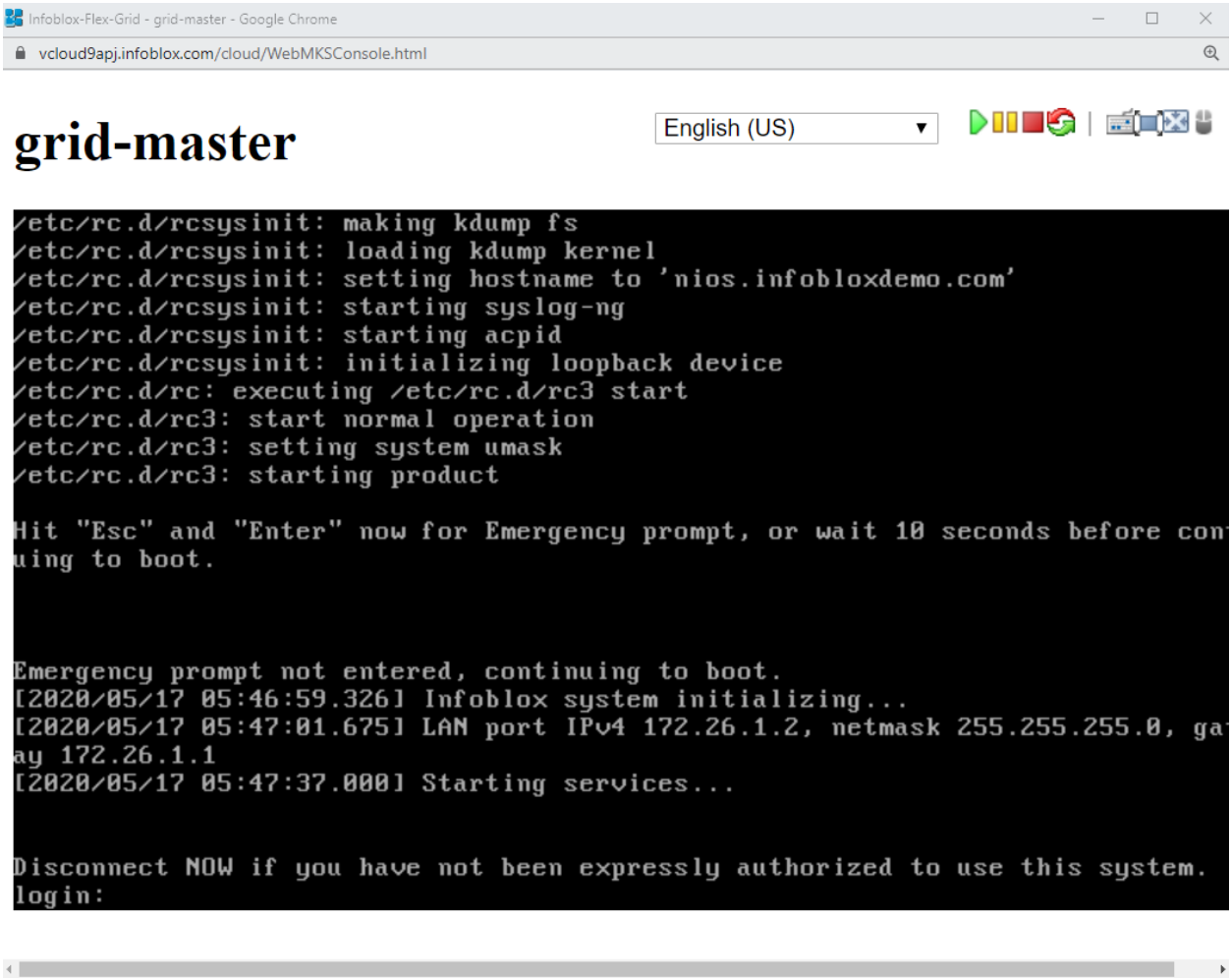
2. Double click the vApp to access its resources. You will now see that vApp has network connectivity.



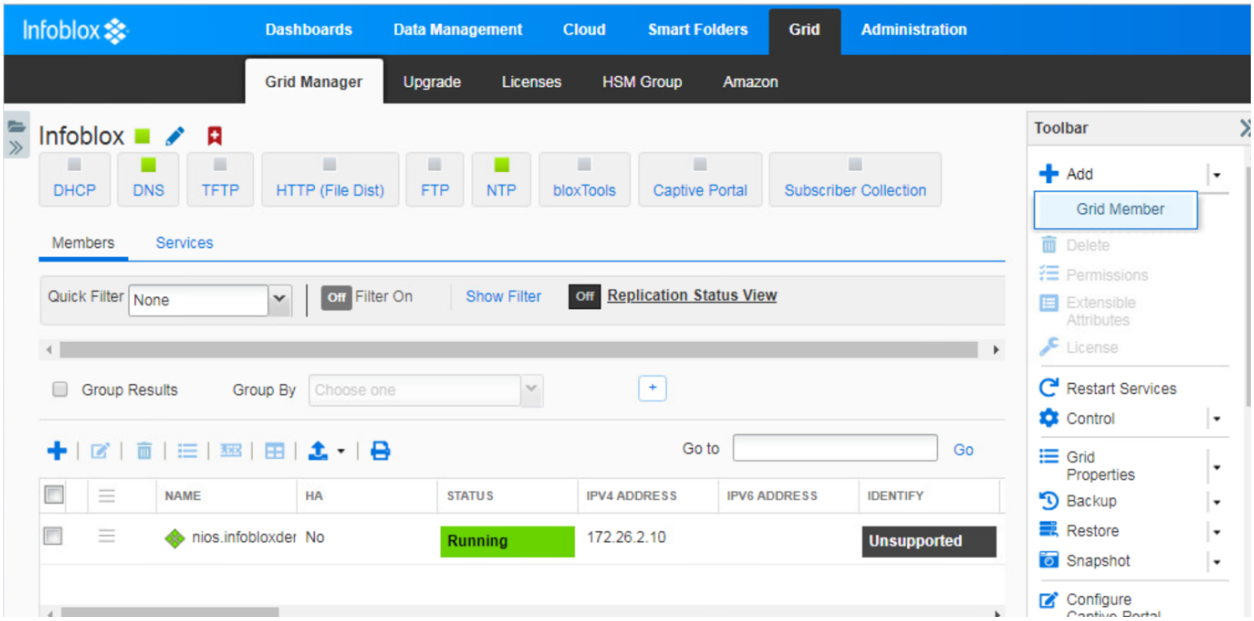
3. Navigate to the **Virtual Machines** tab and make a note of Lan-1 ipaddreses of the VMs.These ip addresses will be assigned in the respective VMs.

Console	Name	Status	OS	Networks	IP Address
	grid-master	Powered On	Other (64-bit)	NIC 0*: Mgmt NIC 1 : Lan-1 NIC 2 : - NIC 3 : -	192.168.2.10 172.26.2.10 - -
	member01	Powered On	Other (64-bit)	NIC 0*: Mgmt NIC 1 : Lan-1 NIC 2 : - NIC 3 : -	192.168.2.11 172.26.2.11 - -
	member02	Powered On	Other (64-bit)	NIC 0*: Mgmt NIC 1 : Lan-1 NIC 2 : - NIC 3 : -	192.168.2.12 172.26.2.12 - -
	win7-client	Powered On	Microsoft Win	NIC 1*: Lan-1	172.26.2.13

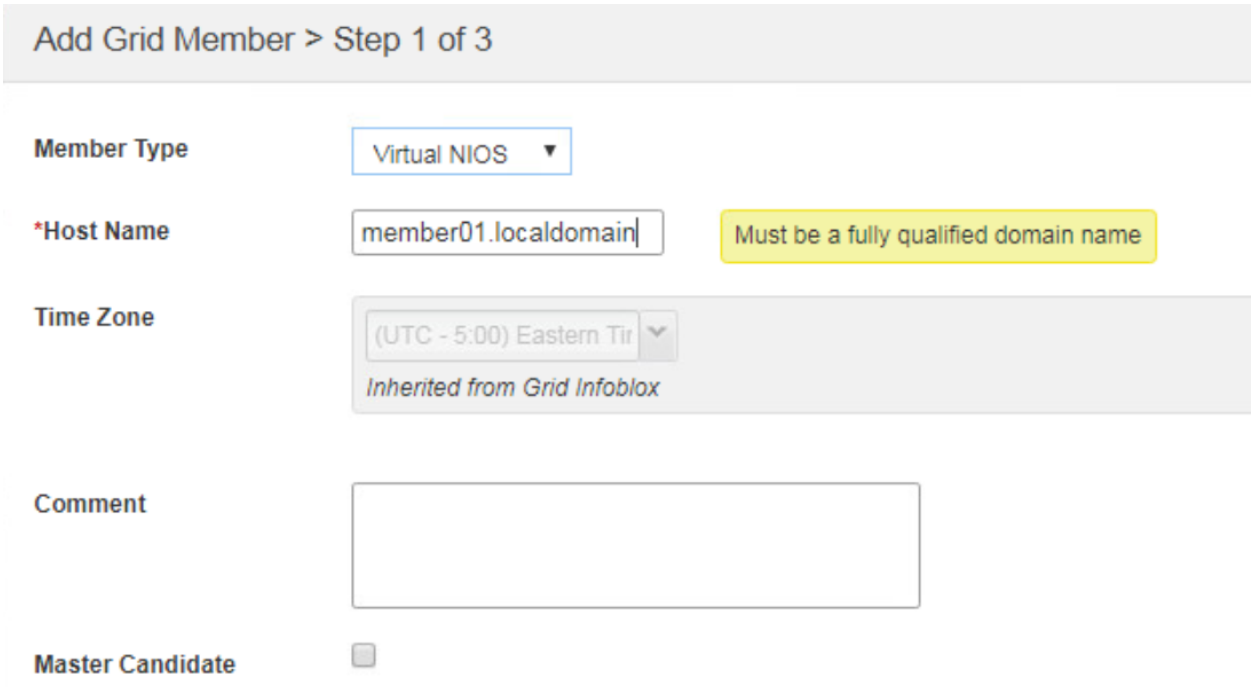
4. Double click **grid-master** VM console icon to access its console.



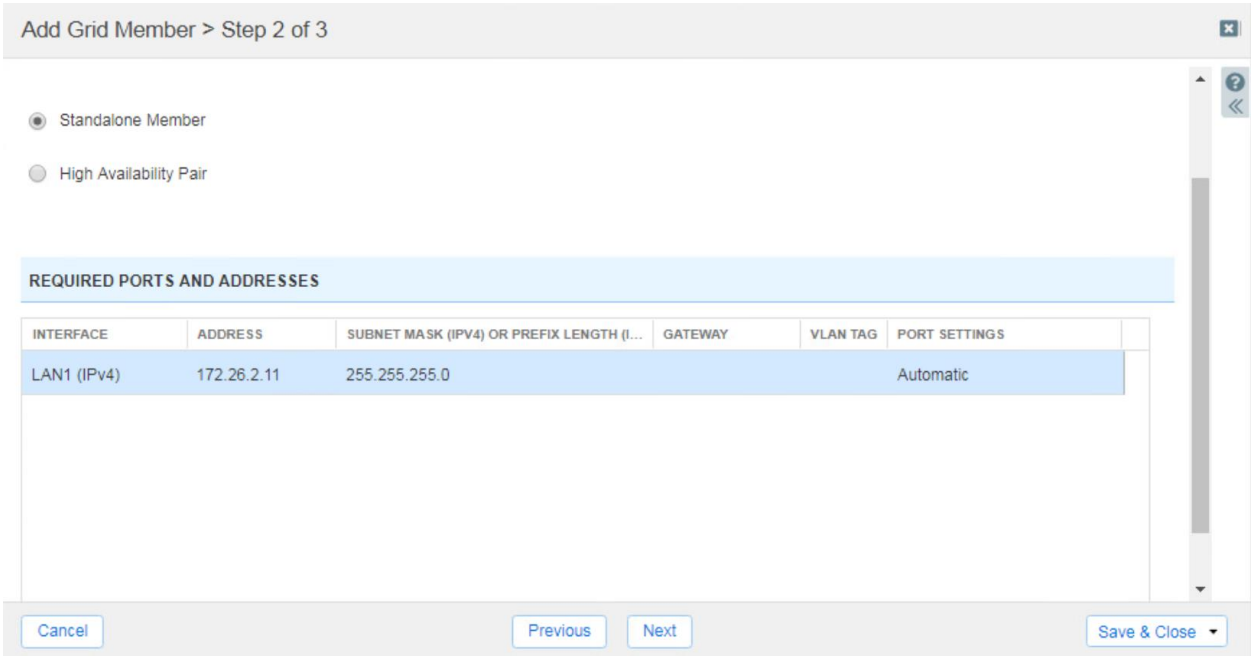
5. Use default user id and password(admin / infoblox) to login into the console of grid-master.
6. Use `set_temp` command license to activate desire licenses. After license assignment use `set_network` command to set lan-1 ip address. Use the same lan-1 IP address which is reflected under IP Address section of the vcd GUI. (For more details on setting and activating flex based licenses please refer this deployment guide.
7. For members(member01 and member02) first set the hardware type to IB-FLEX by executing `set hardware -type IB-FLEX` command. Post this step, set up the networking. Please refer this guide for more details.
8. Open the console of win7-client VM and make sure that it has the correct lan-1 IP address. If not, then assign the correct lan-1 IP address.
9. Open a browser and login to the infoblox grid, by typing https://*grid_master_lan-1_ip* After login page shows up use default user-id and password(admin/infoblox) to access Infoblox Grid.
10. Navigate to Grid → Grid Manager → Members and click on **add** to pre-provision members.



11. Select **Virtual NIOS** from the **Member Type drop** down box and give a hostname.

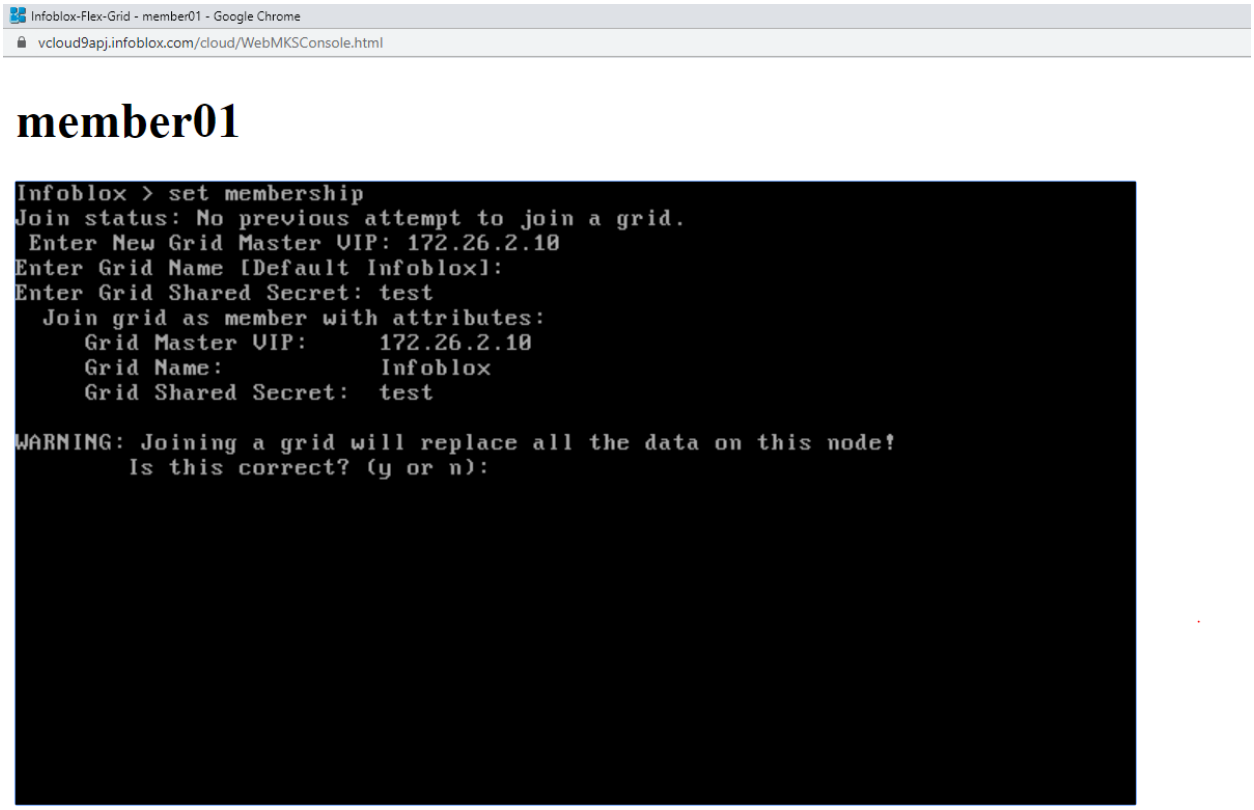


12. Assign correct lan-1 ip address as per the vcd gui. Click on **Save & Close**.



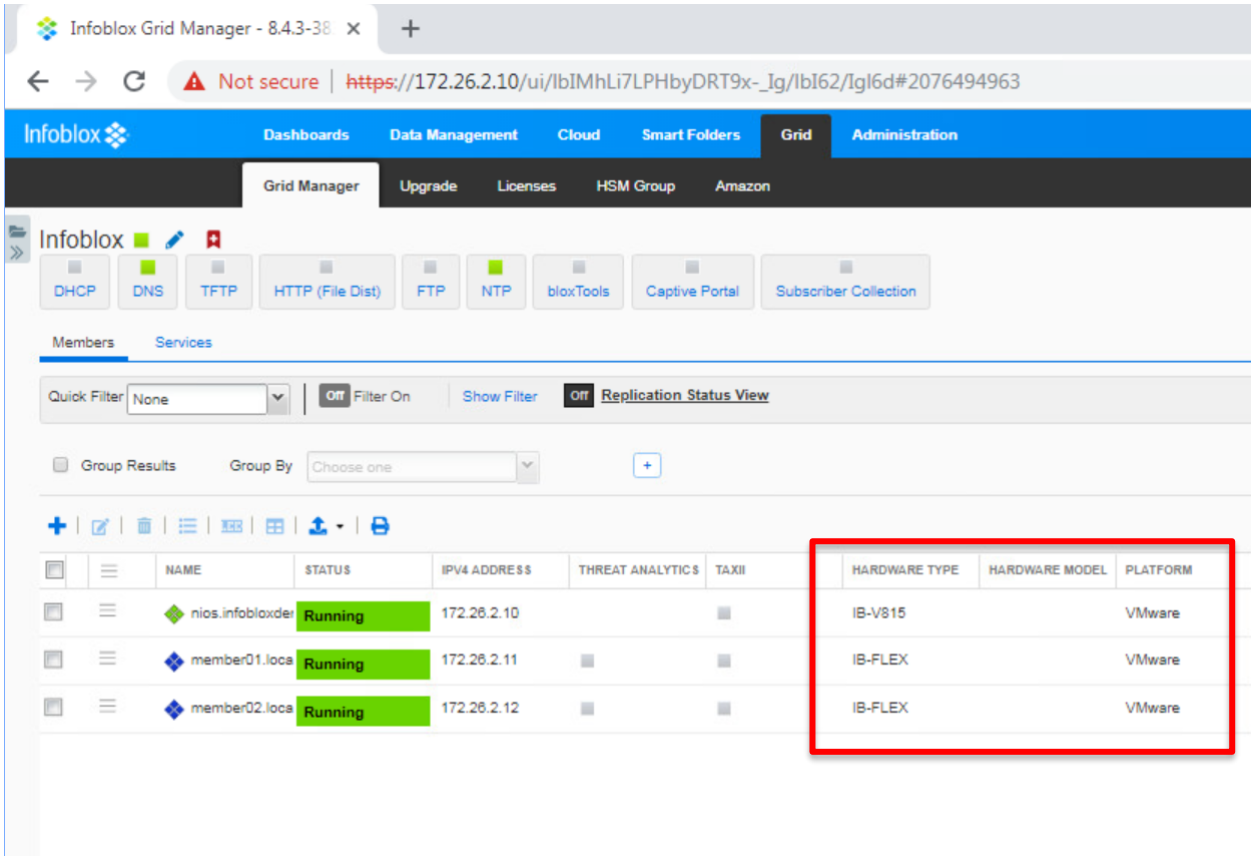
13. Follow the same set of steps and pre-provison member02 as well.

14. Afterprovisnioning members open the member01 console and execute `set membership` command. Key in grid-master lan-1 Ip address and other grid details.



15. Repeat same steps in member02 console window.

16. Post adding member01 and member02 to the grid, login to the grid and verify that members are connected and are showing up as IB-FLEX.

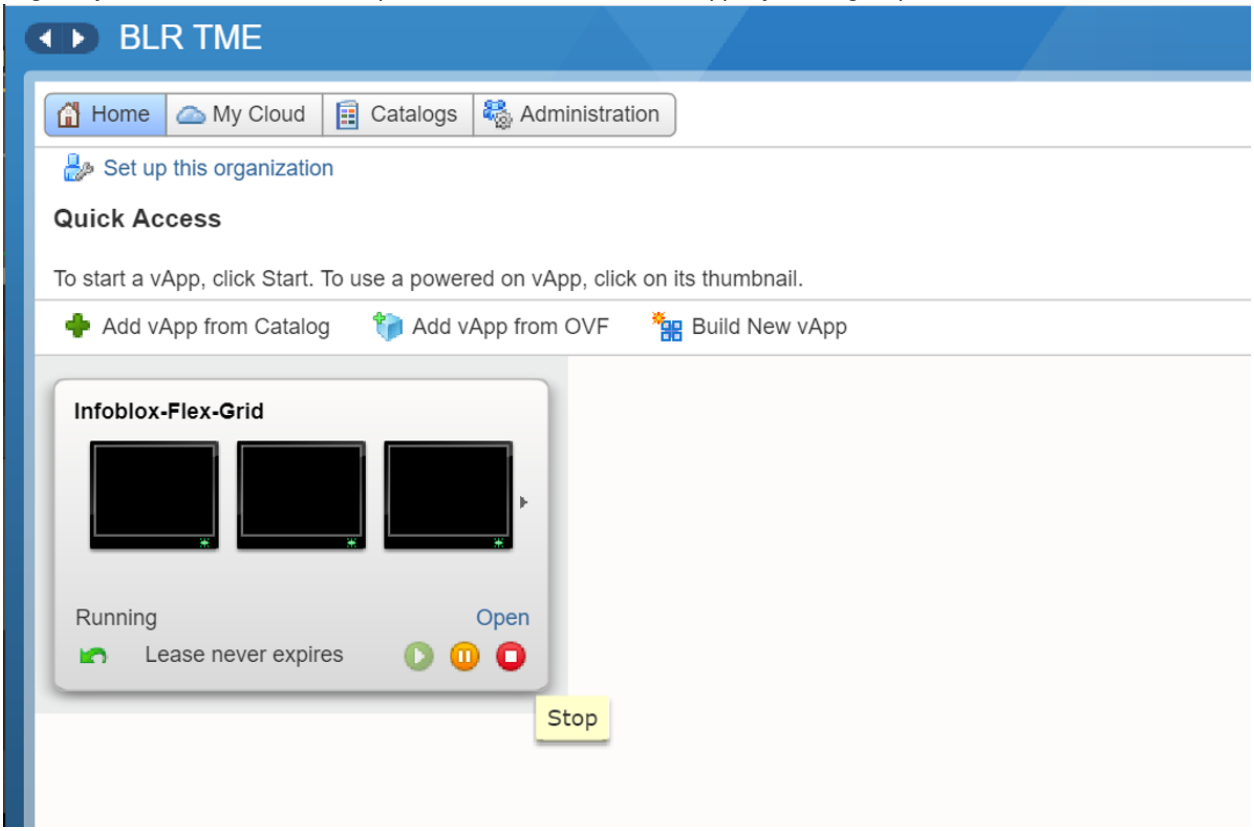


Converting a vApp to a template.

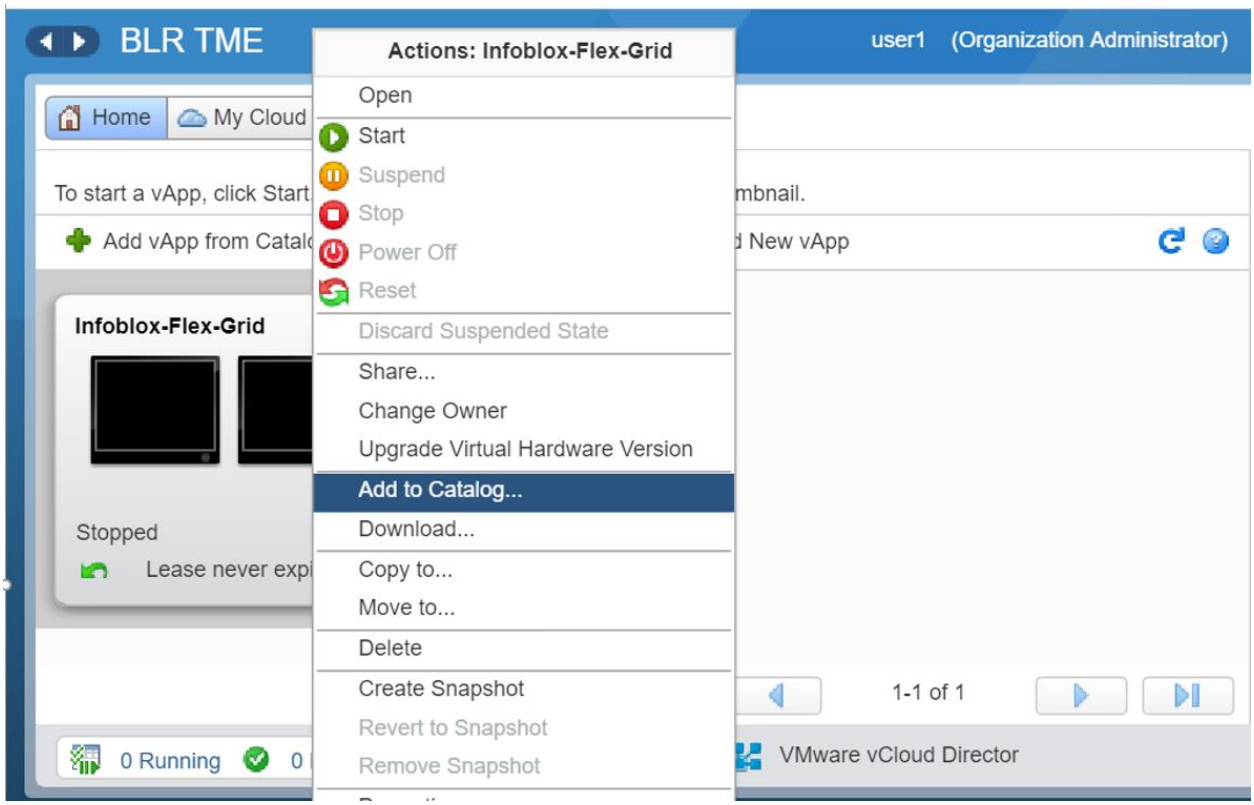
A vApp template is an image of a vApp which contains one or more VM images and one or more vApp networks.

vApp templates are like master copies of a vApp. If need be, a user can deploy multiple vApps(with same or modified configuration) from a vApp template.

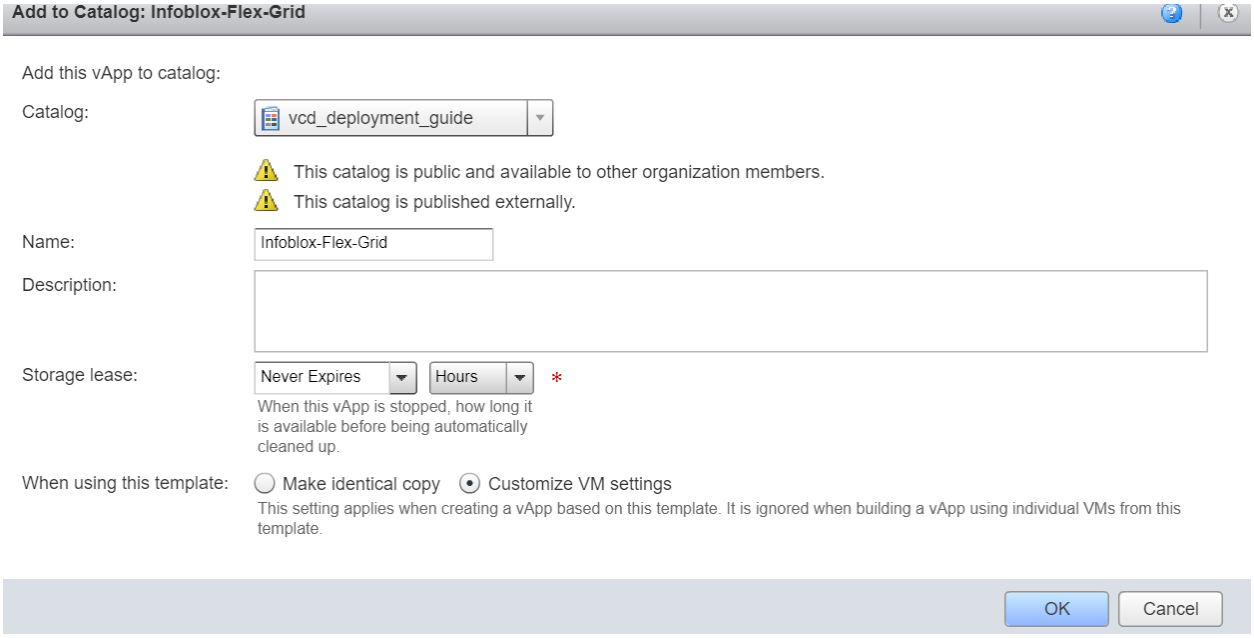
1. Login to your vcd environment and power off Infoblox-Flex-Grid vApp, by clicking stop icon.



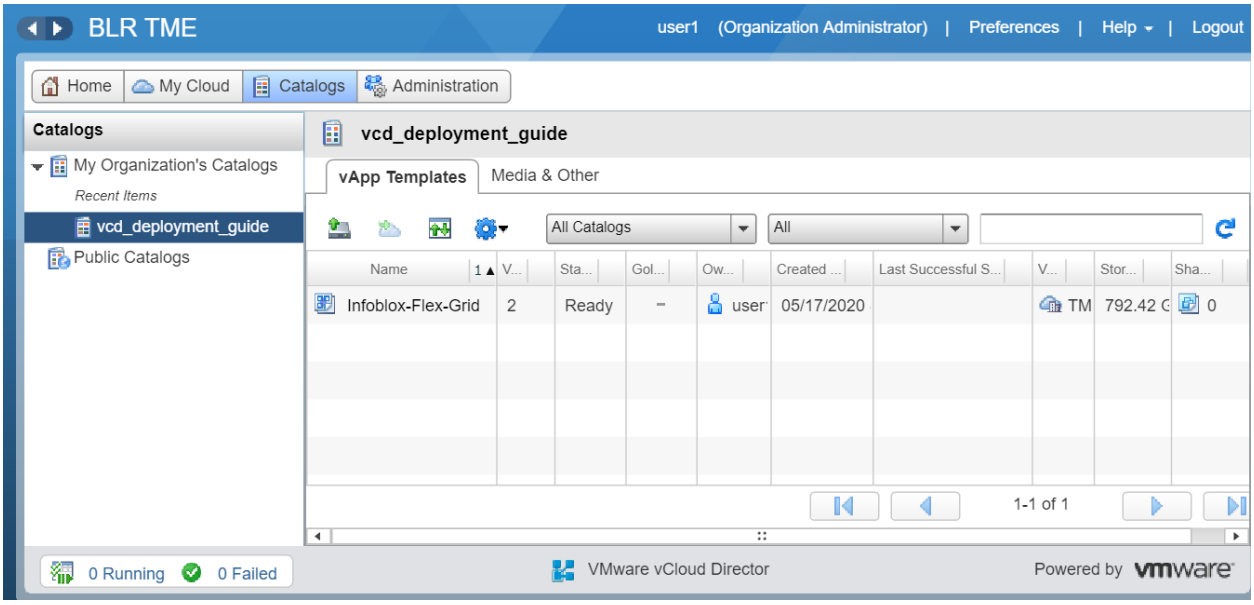
2. Right click the vApp and select **Add to catalog**.



3. Select the name of your **Catalog**(for which you have access) and enter a name for the template. Click **OK .**



4. From the home screen navigate to **Catalogs** → **Catalogs** and select your catalog. You should see a template with status as **Ready**.



Deploying vApps through vcd APIs.

The vCloud Director(vcd) API is a powerful and easy to use solution for automating and orchestrating vApps creation. VMware vCloud Director supports several versions of the vCloud API. vCloud API clients communicate with servers over HTTP, exchanging representations of vCloud objects. These representations take the form of XML elements. You use HTTP GET requests to retrieve the current representation of an object, HTTP POST and PUT requests to create or modify an object, and HTTP DELETE requests to delete an object.

This deployment guide covers cURL, a Linux based utility to fire vcd APIs. cURL by default is not part of Linux distros and needs to be installed separately.

- For Debian based systems use `apt-get install curl -y`
- For Linux based systems use `yum install curl -y`

This deployment guide covers following vcd objects

S.no.	Action	Description
1	**`cloneVApp`**	Creates a copy of an existing vApp.
2	**`instantiateVAppTemplate`**	Edits a vApp to add, remove, or reconfigure virtual machines.

vCloud Director API Constructs

Construct	Description
Organization (org)	An organization is the unit of multi-tenancy that represents a single logical security boundary. An organization contains users, virtual data centers and networks.
Provider Virtual datacenter(vdc)	A provider virtual datacenter is a grouping of compute and storage resources from a single vCenter Server instance.
vApp	A vApp is a container for a distributed software solution and is the standard unit of deployment in vCloud director.
Org Networks	An Org network provides networking services to virtual machines or virtual appliances deployed inside of an Org vDC network.

Fetching vcd construct IDs.

Use following shell script to fetch various vcd construct IDs which we will need to fire subsequent vcd APIs. Replace the variable name (in red)with your environment values. Save this script as **vcd_ids.sh**, assign executable permission to it and then execute it.

```
#!/bin/bash

vcd_url="https://vcd_ip_or_fqdn"

org="org_name"
vdc="vdc_name"
vcd_pass="password"
vcd_user="user_id@$org"
vcd_catalog="catalog_name"
template="template_name"
vapp="name_of_the_vapp"
header="Accept:application/*+xml"
api="version=29.0"


auth_token=$(curl -I -k -s -H "$header;$api" -u "$vcd_user:$vcd_pass" -X POST
"$vcd_url/api/sessions"| awk '/x-vcloud-authorization/{print $2}'| awk '{ sub("\r$", "");print}' )

org_id=$(curl -B -k -s -H "$header;$api" -H "x-vcloud-authorization: $auth_token" -X GET
"$vcd_url/api/query?type=organization&format=references"| awk -F "org"  "/$org/{print
substr(\$2,2)}"|awk -F '"' "{print\$1}")

vdc_id=$(curl -B -k -s -H "Accept:application/*+xml;version=29.0" -H "x-vcloud-authorization:
$auth_token" -X GET "$vcd_url/api/admin/vdcs/query?type=vdc&format=references" |awk -F 'vdc'
"/$vdc/{ print  substr(\$2,2)}"|awk -F '"' '{print $1}')

catalog_id=$(curl -B -k -s -H "$header;$api" -H "x-vcloud-authorization: $auth_token" -X GET
"$vcd_url/api/org/$org_id"| awk -F "catalog" "/$vcd_catalog/{print substr(\$2,2)}"| awk -F '"'
'{print$1}')

catalog_item=$(curl -B -k -s -H "$header;$api" -H "x-vcloud-authorization: $auth_token" -X GET
"$vcd_url/api/catalog/$catalog_id"| awk "/$template/{print\$3}"| awk -F '"' "{print \$2}")

vapp_id=$(curl -B -k -s -H "$header;$api" -H "x-vcloud-authorization: $auth_token" -X GET
"$vcd_url/api/query?type=vApp&format=references" |awk -F "vApp" "/$vapp/{print substr(\$2,2)}"|awk
-F '"' "{print\$1}")

vapp_template=$(curl -B -k -s -H "$header;$api" -H "x-vcloud-authorization: $auth_token" -X GET
"$vcd_url/api/catalogItem/$catalog_item"| awk -F "vAppTemplate" "/$template/{print
substr(\$2,2)}"| awk -F '"' "{print\$1}")


echo "authtoken -->        $auth_token"
echo "org_id -->           $org_id"
echo "vdc_id -->           $vdc_id"
echo "catalog_id -->       $catalog_id"
echo "catalog_item_id -->  $catalog_item"
echo "vapp_template_id --> $vapp_template"
echo "vapp_id -->          $vapp_id"
```

```
root@ansible-control-server:~/vcd_automation# ./vcd_ids.sh
AuthToken -->  1158e17818864c899c0eb03223be7d0d
Org ID -->  a0e8bce1-2de1-4ffc-9dc4-a0a9cee5e18e
VDC ID -->  d1326dc5-45eb-4bec-ab37-003c3fc3cc99
Catalogid -->  1ca8e7af-500d-45e9-bd6b-50d844d0086a
Catalog_item_id -->  b508934a-7d1f-4365-ac8e-ac371c9f07f7
vapp_template_id -->
vappTemplate-9d069506-a2f1-4172-9f9b-76093841c08c
root@ansible-control-server:~/vcd_automation#
```

Cloning a vApp

1. Save the following file as **clone**. Edit the values marked in red. Use the vapp_id generated by the shell script and update it in the "<***Source href***" section.

```
<?xml version="1.0" encoding="UTF-8"?>

<CloneVAppParams xmlns="http://www.vmware.com/vcloud/v1.5"
xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1" name="name_of_the_vapp" deploy="true"
powerOn="false">

  <Description>Cloned vApp Example</Description>

  <Source href="https://vcloud9apj.infoblox.com/api/vApp/vapp_id_obtained_from_the_shell_script"/>

</CloneVAppParams>
```

2. Execute the following curl command to commence cloning process of an existing vapp.

```
curl -s -i -k -H 'Accept:application/*+xml;version=29.0' -H 'x-vcloud-
authorization:authtoken' -X POST
https://vcd_ip_address_or_fqdn/api/vdc/vdc_id/action/cloneVApp -H 'Content-
Type: application/vnd.vmware.vcloud.cloneVAppParams+xml' -d @clone
```

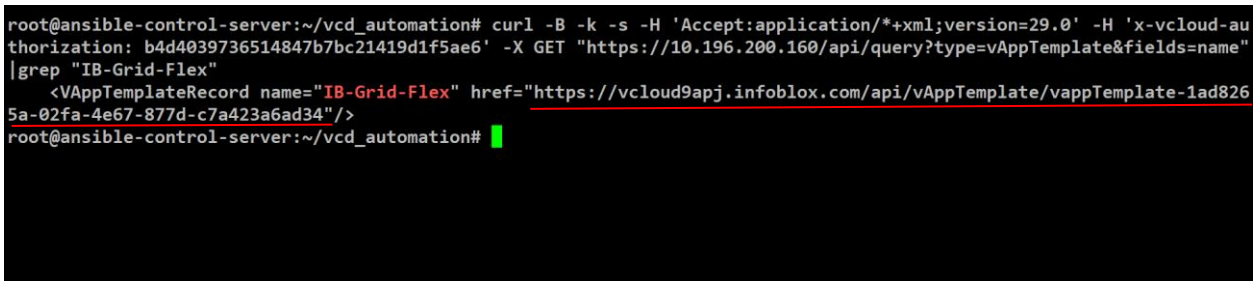
```
root@ansible-control-server:~/vcd_automation# curl -s -i -k -H 'Accept:application/*+xml;version=29.0' -H 'x-vcloud-au
947d1fc40f67b55d' -X POST https://10.196.200.160/api/vdc/d1326dc5-45eb-4bec-ab37-003c3fc3cc99/action/cloneVApp -H 'Con
mware.vcloud.cloneVAppParams+xml' -d @clone
```

3. Post executing the command, login to the vcd gui to verify that a new vapp creation has started or not.

Deploying a vApp from a template.

1. Retrieve the vApp template href link, by executing following curl command.

```
curl -B -k -s -H 'Accept:application/*+xml;version=29.0' -H 'x-vcloud-
authorization: authtoken' -X GET "https://vcd_ip_address_or_fqdn
/api/query?type=vAppTemplate&fields=name"|grep "template_name"
```



2. Command mentioned in step 1 will return matching template name and its href link(underlined in red). Make a note of this.
3. Run the following curl command against the href link obtained from the previous step. This command will redirect the vapp XML representation to `vApp_template.xml` file.

```
curl -B -k -s -H 'Accept:application/*+xml;version=29.0' -H 'x-vcloud-
authorization: authtoken' -X GET "href_link_obtained_from_step_1" >>
vApp_template.xml
```

4. Open the vApp_template.xml file and make a note of xml elements within **<NetworkConfigSection> </NetworkConfigSection>** tag. Skip external network section in case, your vapp template does not have/need external network connectivity.
5. Update the network config section copied from the previous step under **InstantiationParams** section file in the following xml schema and save it as **instantiate_vapp .**

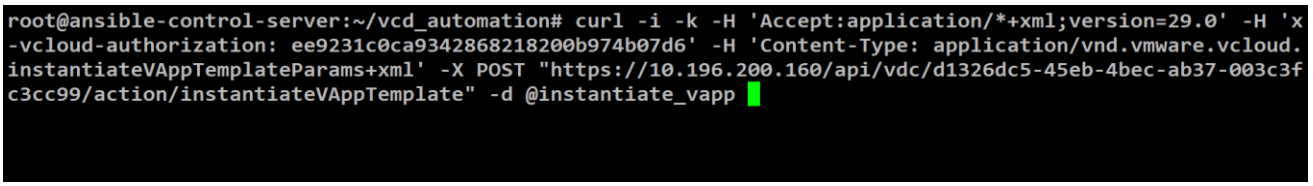
```
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="name"
   deploy="true"
   powerOn="true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>DDI automation</Description>
   <InstantiationParams>

      …network_section…

   </InstantiationParams>
   <Source
           href="href_link_obtained_from_step_1" />
      <AllEULAsAccepted>true</AllEULAsAccepted>
</InstantiateVAppTemplateParams>
```

6. After updating network section, xml file should look like this. Change the values in red as per your environment.

```
<?xml version="1.0" encoding="UTF-8"?>
<InstantiateVAppTemplateParams
   xmlns="http://www.vmware.com/vcloud/v1.5"
   name="name"
   deploy="true"
   powerOn="true"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1">
   <Description>DDI automation</Description>
   <InstantiationParams>
      <NetworkConfigSection>
         <ovf:Info>Configuration parameters for logical networks</ovf:Info>
         <NetworkConfig
           networkName="mgmt">
        <Description></Description>
        <Configuration>
            <IpScopes>
                <IpScope>
                    <IsInherited>false</IsInherited>
                    <Gateway>192.161.2.1</Gateway>
                    <Netmask>255.255.255.0</Netmask>
                    <SubnetPrefixLength>24</SubnetPrefixLength>
                    <Dns1>10.120.3.10</Dns1>
                    <IsEnabled>true</IsEnabled>
                    <IpRanges>
                        <IpRange>
                            <StartAddress>192.161.2.20</StartAddress>
                            <EndAddress>192.161.2.80</EndAddress>
                        </IpRange>
                    </IpRanges>
                </IpScope>
            </IpScopes>
            <FenceMode>isolated</FenceMode>
            <RetainNetInfoAcrossDeployments>false</RetainNetInfoAcrossDeployments>
            <GuestVlanAllowed>false</GuestVlanAllowed>
        </Configuration>
        <IsDeployed>false</IsDeployed>
    </NetworkConfig>
    <NetworkConfig networkName="lan1">
        <Description></Description>
        <Configuration>
            <IpScopes>
                <IpScope>
                    <IsInherited>false</IsInherited>
                    <Gateway>172.126.2.1</Gateway>
                    <Netmask>255.255.255.0</Netmask>
                    <SubnetPrefixLength>24</SubnetPrefixLength>
                    <Dns1>10.120.3.10</Dns1>
                    <IsEnabled>true</IsEnabled>
                    <IpRanges>
                        <IpRange>
                            <StartAddress>172.126.2.10</StartAddress>
                            <EndAddress>172.126.2.100</EndAddress>
                        </IpRange>
                    </IpRanges>
                </IpScope>
            </IpScopes>
            <FenceMode>isolated</FenceMode>
            <RetainNetInfoAcrossDeployments>false</RetainNetInfoAcrossDeployments>
            <GuestVlanAllowed>false</GuestVlanAllowed>
        </Configuration>
        <IsDeployed>false</IsDeployed>
    </NetworkConfig>
     </NetworkConfigSection>
      <LeaseSettingsSection
         type="application/vnd.vmware.vcloud.leaseSettingsSection+xml">
         <ovf:Info>Lease Settings</ovf:Info>
         <StorageLeaseInSeconds>172800</StorageLeaseInSeconds>
         <StorageLeaseExpiration>2010-04-11T08:08:16.438-07:00</StorageLeaseExpiration>
      </LeaseSettingsSection>
   </InstantiationParams>
   <Source
                  href="href_link_obtained_from_step_1" />
      <AllEULAsAccepted>true</AllEULAsAccepted>
</InstantiateVAppTemplateParams>
```

7. Use the following curl command to initiate vApp instantiation from a template.

```
curl -i -k -H 'Accept:application/*+xml;version=29.0' -H 'x-vcloud-authorization:
authtoken' -H 'Content-Type:
application/vnd.vmware.vcloud.instantiateVAppTemplateParams+xml' -X POST
"https://vcd_ip_or_fqdn/api/vdc/vdc_id/action/instantiateVAppTemplate" -d
@instantiate_vapp
```



Infoblox is the leader in modern, cloud-first networking and security services. Through extensive integrations, its solutions empower organizations to realize the full advantages of cloud networking today, while maximizing their existing infrastructure investments. Infoblox has over 12,000 customers, including 70 percent of the Fortune 500.

Corporate Headquarters | 2390 Mission College Boulevard, Ste. 501 | Santa Clara, CA | 95054
+1.408.986.4000 | info@infoblox.com | www.infoblox.com