vSphere with Tanzu Quick Start Guide

Getting vSphere 7.0 Update 1 with Tanzu set up in a lab network for Proof of Concept testing

vmware®

Table of contents

MORE INFO

- *vSphere Blogs*
- *Go download vSphere 7 Update 1!*
- *Learn how to activate the in-product evaluation*.
- *HAProxy Load Balancer Download*

ADDITIONAL RESOURCES

- *vSphere website*
- *vSphere Academy*
- Video: *A Quick Look at What's New in vSphere 7 Update 1*
- Video: *vSphere with Tanzu Overview in 3 Minutes*

Introduction

vSphere with Tanzu is the latest update to Kubernetes running natively on vSphere. The biggest change with vSphere with Tanzu is that introduces the ability to enable Kubernetes on vSphere clusters using a vSphere Distributed Switch.

vSphere with Tanzu utilizes vSphere Distributed Switch Portgroups and a "bring your own" network strategy for load balancing Kubernetes workloads. The initial release will support HAProxy for load balancing via our new Load Balancer API. Look for additional load balancers coming soon.

Scope of this Document

The guiding principle of this document is to get to a working evaluation of vSphere with Tanzu. You can create this environment on physical hardware or via nested virtual machines. You can also do everything in this document within your standard VMware evaluation licensing window.

Setting up and installing vSphere with Tanzu, regardless of using NSX or vSphere Distributed Switch requires custom networking configuration depending on your environment. Because so many customer's configurations are unique it's very difficult to test every configuration.

With that in mind and to ensure you can get vSphere with Tanzu up and running as quickly as possible on an evaluation basis, we have limited the networking scope of this guide to using one subnet for workloads and virtual IP's (VIP) and one subnet for vSphere components (vCenter, ESXi).

Note: This is NOT a replacement for the documentation, nor should this configuration be used in a production environment. This configuration is purely for "kicking the tires" or creating a Proof of Concept of vSphere with Tanzu. We hope you enjoy it!

Prerequisites

Installation/Configuration

This document assumes you know how to install and configure ESXi and VCSA, enable DRS and HA and configure networking and shared storage. If you are not comfortable with that, then we highly encourage you to take a lab at VMware's Hands-On Labs. It's free! Go to *https://hol.vmware.com*

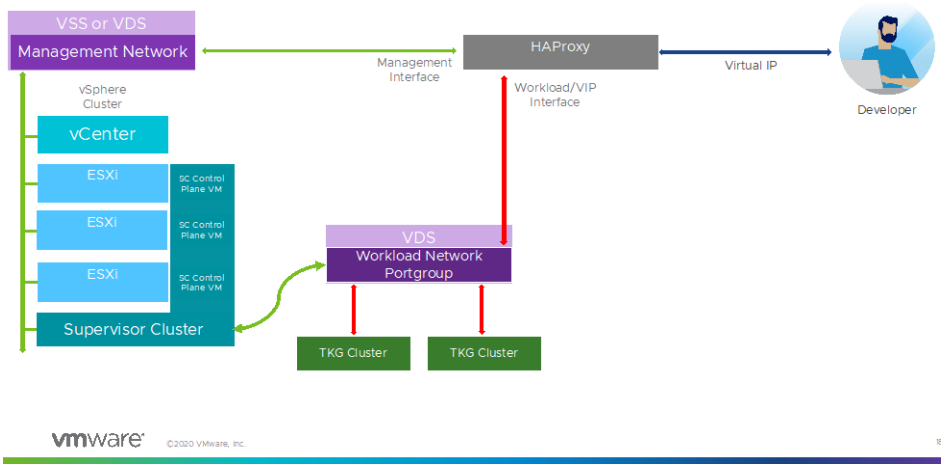
Networking

To make networking as easy as possible we recommend the following setup for your PoC/lab environment.

You will need two separate, **routable** subnets configured. One subnet will be for Management Networking. This is where vCenter, ESXi, the Supervisor Cluster and the Load Balancer will live. The other subnet will be used for Workload Networking. This is where your TKG clusters will live.

As you will see, these two subnets are going to be configured on separate portgroups. If these subnets are on separate VLANs then you will need to configure the portgroups accordingly.

"Simplified" vSphere with Tanzu Network Topology



Note: In the documentation you will see mention of a "frontend network" configuration. This guide purposely did not include that configuration in the goal of making this as simple as possible. The Frontend configuration would be used in a production environment to isolate the nodes of your clusters from the network used by developers to access the cluster.

Subnets

The size of each subnet is dependent on your configuration needs. If you are just installing this to "try it out" and have limited subnet resources, the Management network could be very small (see below), and your Workload Network could be as small as a /28. That would give you 14 addresses.

Let's look at the bare minimum requirements. First, we will start with subnet masks. This will give you an idea of how many IP addresses to request for your evaluation.

Subnet Mask	/28	/27		/26	/25	/24
IP Addresses	14	30	62	162	254	

Management Network

The Supervisor Cluster and Load Balancer are "dual homed". They have a virtual NIC attached to both the Management Network and the Workload Network.

- This is to allow the Supervisor Cluster to program the load balancer
- The VM's IP address on this network should be static since the Supervisor Cluster will not be able to program the load balancer if the load balancer's control plane IP address changes.
- This is also the network to which the VM's default gateway should belong.
- Finally, other system activity, such as DNS queries, will occur via this network

For Lab purposes the "Management Network" can be on a VSS or a portgroup on a VDS.

Components	Management Network IP Address Minimal Requirements
Supervisor Cluster	3 IPs
ESXi Hosts	1 per host
VCSA	1 IP
Load Balancer	1 IP

Workload Network

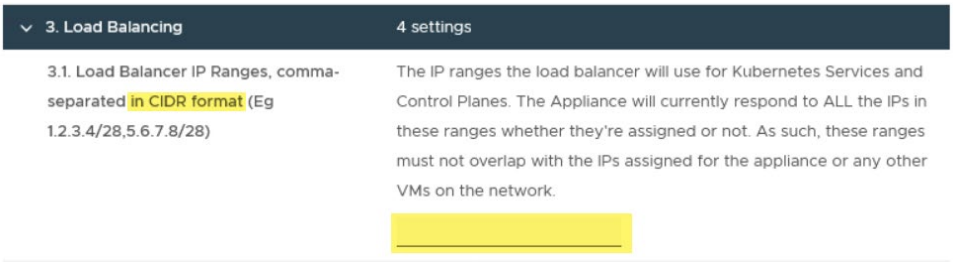
The Workload Network has the following characteristics:

- This network is used by the load balancer to access the services on the Supervisor and Guest clusters.
- When the HAProxy VM is deployed with only two NICs, the *Workload* network must also provide the logical networks used to access the load balanced services.

Components	Workload Network IP Address Minimal Requirements
Supervisor Cluster	3 IPs
TKG Cluster Controller	1 IP per Controller
TKG Cluster Worker	1 IP per Worker
Load Balancer	1 IP per Kubernetes LB Service
Virtual IPs	TBD

The main takeaway here is that there are two ranges of IP addresses in use in the **Workload Network** subnet.

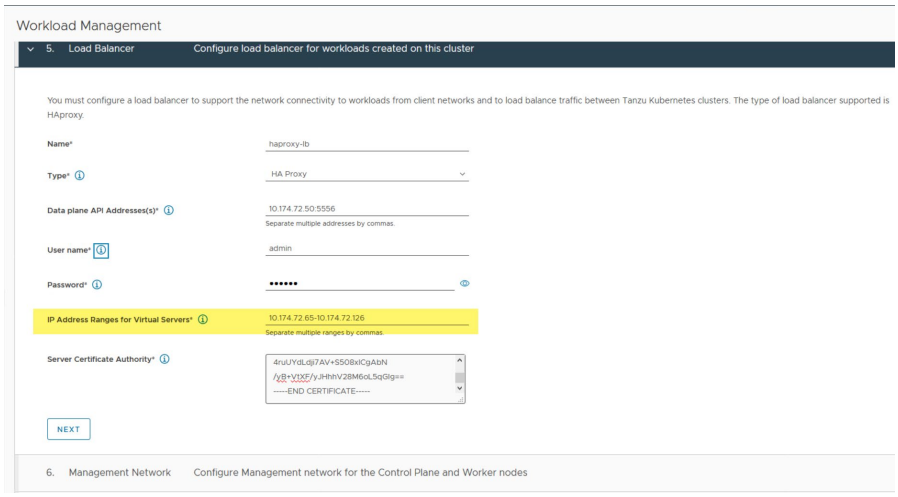
- The Cluster Node Range
  - The range for Supervisor and Guest Cluster nodes on the workload network.
  - In the UI during the deployment of the Load Balancer OVA (in this case HAProxy) this is referred to as the Load Balancer IP Ranges



- The Virtual IP Range
  - These are the IP addresses offered up by the Load Balancer that will route to a TKG cluster or application you set up. Developers running Kubectl will connect to

their vSphere Namespace and TKG clusters using one of these IP Addresses. These IP Addresses will be provisioned from this range when a developer creates a Kubernetes Service of "Type: Load Balancer"'

- In the UI during the deployment of the Workload Network this is referred to as "IP Address Ranges for Virtual Servers"



*Note: If you have a single TKG cluster with one Control Plane VM and three Worker VMs then you are now at ten IP's leaving you four VIPs. That's the **absolute** bare minimum. We would recommend at least a /27 (30 IP addresses) or a /26 (62 IP Addresses).*

If you are looking to set this up for developers to try, then you probably want a /25 (126 IP Addresses) or a /24 (255 IP Addresses). This will allow them to create several TKG clusters for their testing and validation.

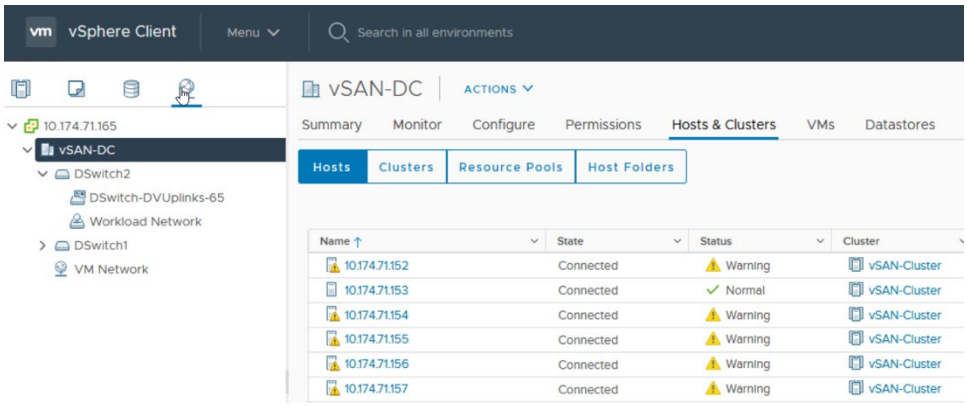
vSphere Distributed Switch Setup

First, you will need a vSphere Distributed Switch (VDS) configured on all of the hosts in your cluster. Please see the documentation for how to set that up. This should be a version 7 VDS. (The default)

Next, you will need at least one VDS Portgroup set up. If you have vSphere Standard Switching (VSS) set up already you can use that as your "**Management Network**". This is where you'll be deploying your load balancer VM to so it should have direct connectivity to ESXi and vCenter.

If you prefer, you can create a separate, new VDS Portgroup and call it “Management”. If you do that and you are using VLANs then ensure that both the VSS and VDS Management portgroup are on the same VLAN.

Next, you will create a “Workload Network” portgroup. If you are using VLANs then configure this portgroup accordingly.

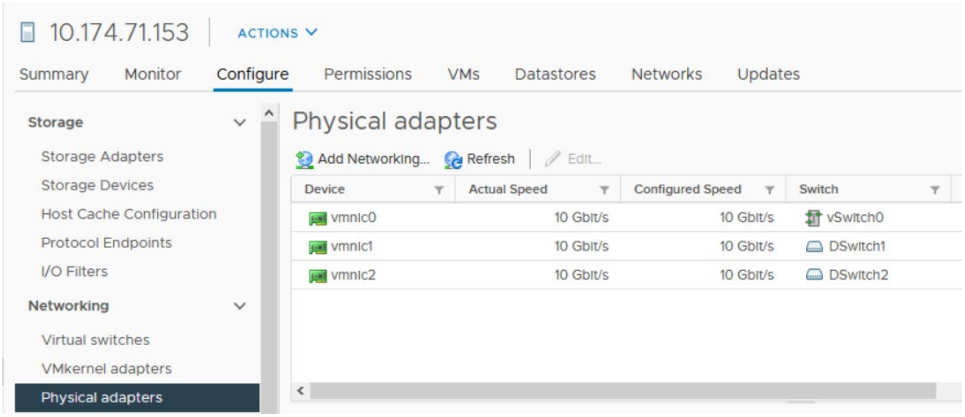


You will need IP addresses in two separate, routable subnets. The first subnet will be the one that we’ll call “**Management**”. This is where your VCSA, ESXi hosts, Control Plane VMs and Load Balancer (e.g. HAProxy). On the ESXi hosts this subnet will connect via vmnic0.

This network can live either on a vSphere Standard Switch or a vSphere Distributed Switch portgroup. In the image above you can see a **VM Network** VSS portgroup and a VDS with a **Workload Network** portgroup. **VM Network** will be used at the **Management Network**.

The **Workload Network** subnet will be “carved” into two IP ranges, one each for the Supervisor Cluster and TKG Cluster systems and one for Load Balancer virtual IPs (VIPs). On the ESXi hosts this subnet will live on vmnic1. This network is **required** to be on a vSphere Distributed Switch. (Version 7, the default value)

Here’s an example of how the vmnic physical adapters are configured.



Installation

It is recommended that a minimum of three ESXi hosts be used for this configuration. They can be physical hosts or virtualized/nested hosts. As part of the installation we will assume that the hosts have two NIC cards. (vmnic0, vmnic1) My example has three nic cards and vmnic1 is not used.

If you are comfortable setting up vSphere in a nested environment, then you can use that for your proof of concept. It is highly recommended that you reference Nested Virtualization content on William Lam's website. From there you can subscribe to his Content Library where he has pre-built ESXi virtual machine OVA's available for installation. The link for his page on nested virtualization is: http://vmwa.re/nestedesxi

Note: Use of nested virtual hosts is not supported in production

ESXi Installation

Physical hosts

Install ESXi on 3 hosts according to the documentation. You will need to use vSphere supported **shared storage** solution. Typically, this is vSAN, NFS, iSCSI or Fibre Channel. Shared storage is required. Presenting storage volumes directly is not.

Note: vSAN is NOT required for vSphere with Tanzu! Any supported shared storage will work.

ESXi Network Configuration

As guided above, ensure that the ESXi VMs have two NICs configured.

The vmnic0 card will be the uplink for the "**Management Network**". On this network will be the VCSA, ESXi hosts and the Supervisor Control Plane. This network needs access to NTP, DNS and DHCP services.

The vmnic1 card will be used as the uplink for the "**Workload Network"** portgroup on the vDS.

VCSA Installation

Install the VCSA according to the documentation. It should be on the same network as your ESXi hosts. The configuration option to choose for this installation is:

VCSA Size: Small

Configuring vCenter

When the VCSA is up and running, log in to *administrator@vSphere.local* and do the following tasks.

- Create a cluster
- Enable vSphere HA and DRS on the cluster
    - *DRS should be set to fully automated*
- Add hosts to the cluster
- On all hosts, enable vmk0 for all traffic types required. E.g. vMotion, vSAN, etc.
- If you are using NFS or iSCSI shared storage, configure this now
- If you are using vSAN then configure this now and create a vSAN datastore (ensure vmk0 is enabled for vSAN traffic)

Below is some sample PowerCLI code to help you set vmk0 to the correct settings.

Configure vSwitch and Host Network Adapter PowerCLI Example

```
Get-VMHost|Get-VirtualSwitch -Name vSwitch0|Set-VirtualSwitch -Mtu 9000 -
Confirm:$false
Get-VMHost|Get-VMHostNetworkAdapter -name vmk0|Set-VMHostNetworkAdapter -
VsanTrafficEnabled $true -VMotionEnabled $true -Confirm:$false
```

VDS Configuration

- Create a vDS called "Dswitch" (default name) and distributed portgroup called "**Workload Network**"
- Uplink the vDS to vmnic1 on each ESXi host.

Creating Workload Network VDS Switch and Portgroup PowerCLI Example

Below is an example of some PowerCLI code that can help you automate the proper configuration of the Workload Network for this evaluation. This is given as an example only.

```
$workloadhosts = get-cluster $Cluster | get-vmhost
New-VDSwitch -Name "Dswitch" -MTU 9000 -NumUplinkPorts 1 -location vSAN-DC
Get-VDSwitch "Dswitch" | Add-VDSwitchVMHost -VMHost $workloadhosts
Get-VDSwitch "Dswitch" | Add-VDSwitchPhysicalNetworkAdapter -VMHostNetworkAdapter
($workloadhosts | Get-VMHostNetworkAdapter -Name vmnic1) -Confirm:$false
```

Storage Configuration

In this section we are going to create a tagging-based storage profile. The datastore you use needs to be seen by all ESXi hosts in the cluster. When adding the tag, you will also need to create a new tag category. These storage policies will be used in the Supervisor Cluster and namespaces.

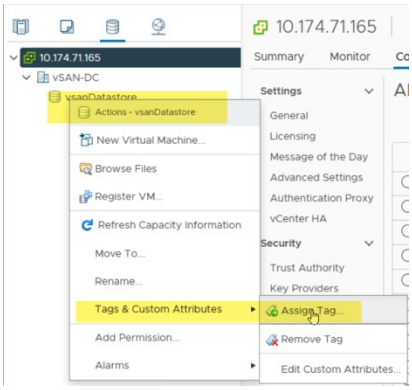
The policies represent datastores available in the vSphere environment. They control the storage placement of such objects as control plane VMs, pod ephemeral disks, container images, and persistent storage volumes. If you use VMware Tanzu™ Kubernetes Grid™ Service, the storage policies also dictate how the Tanzu Kubernetes cluster nodes are deployed. Let's get started. The following includes PowerCLI to automate these steps and then a UI version.

Storage Policies PowerCLI Example

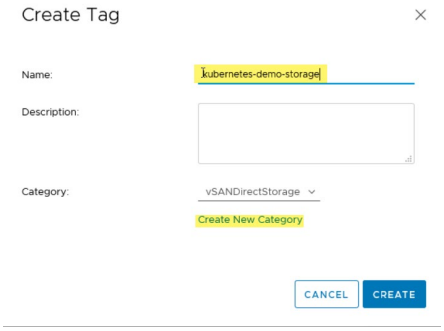
```
#
#  Set up tags for vSphere with  Tanzu
#
$StoragePolicyName = "kubernetes-demo-storage"
$StoragePolicyTagCategory = "kubernetes-demo-tag-category"
$StoragePolicyTagName = "kubernetes-gold-storage-tag"
New-TagCategory -Name $StoragePolicyTagCategory -Cardinality single -EntityType
Datastore
New-Tag -Name $StoragePolicyTagName -Category $StoragePolicyTagCategory
Get-Datastore -Name $datastore | New-TagAssignment -Tag $StoragePolicyTagName
New-SpbmStoragePolicy -Name $StoragePolicyName -AnyOfRuleSets (New-SpbmRuleSet -
Name "wcp-ruleset" -AllOfRules (New-SpbmRule -AnyOfTags (Get-Tag
$StoragePolicyTagName)))
```

Storage Policies vCenter UI Example

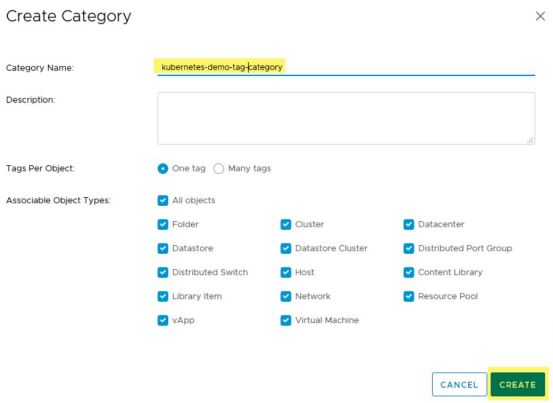
i. Right-click on the datastore you want to use and select Tags and Custom Attributes and Assign Tags. If you want to use more than one datastore then at the end you can just assign the tag we are about to create to that datastore.



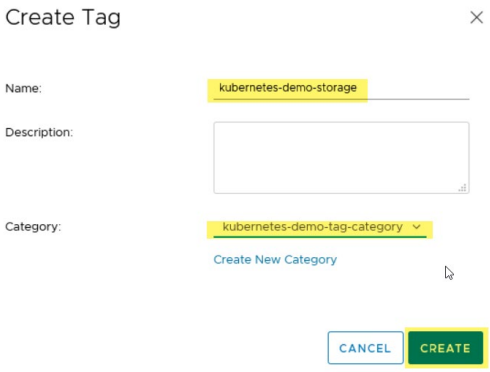
ii. Click on Add Tag and fill the Tag Name as 'kubernetes-demo-storage'



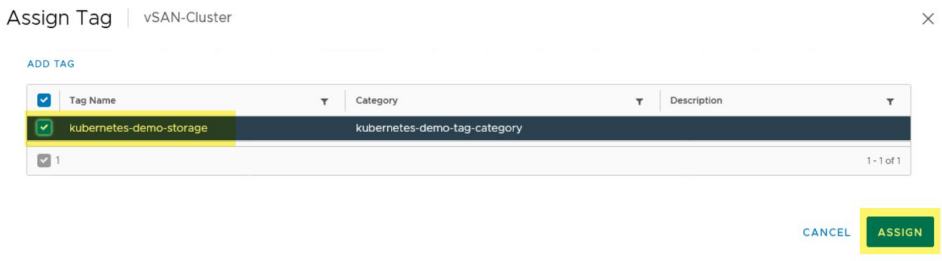
i. Click create category, enter the Category Name: 'kubernetes-demo-tag-category' then click Create.



ii. In the Create Tag box you will see the new tag. Select the category you just created. Click Create

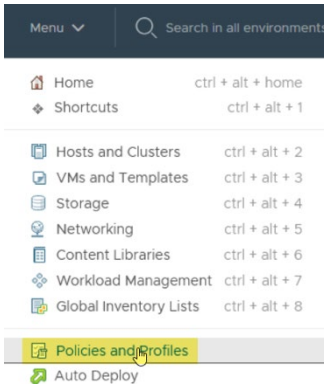


iii. Select this newly created Tag and click Assign.

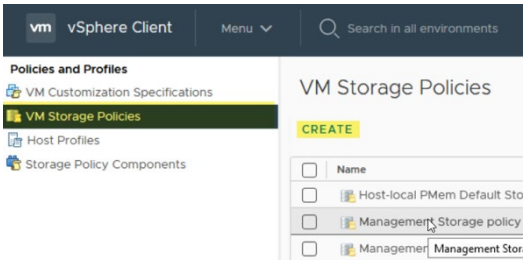


Now we need to create a tag-based storage policy.

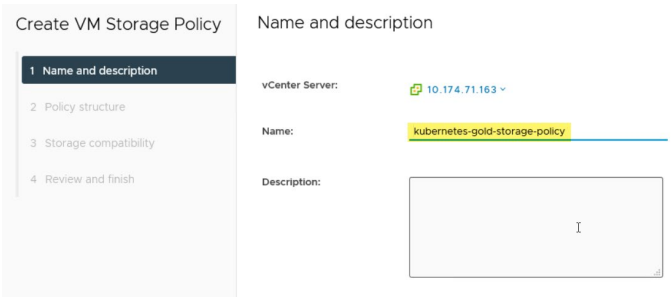
Menu -> Profiles and Policies



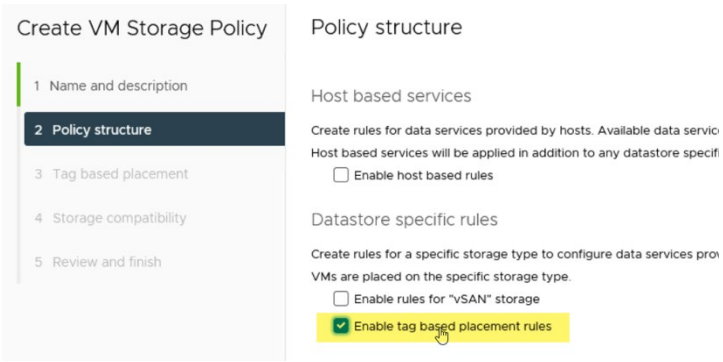
VM Storage Policies -> Create VM Storage Policy



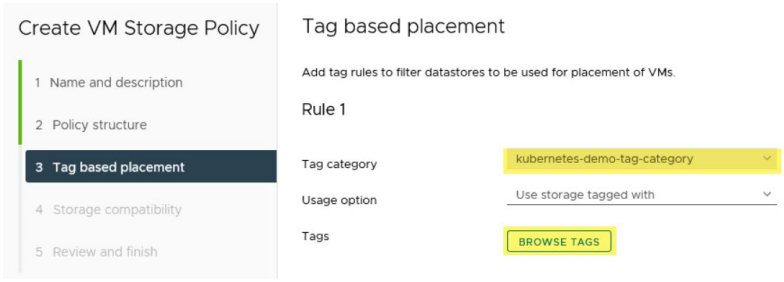
Name: '**kubernetes-gold-storage-policy**' then click Next



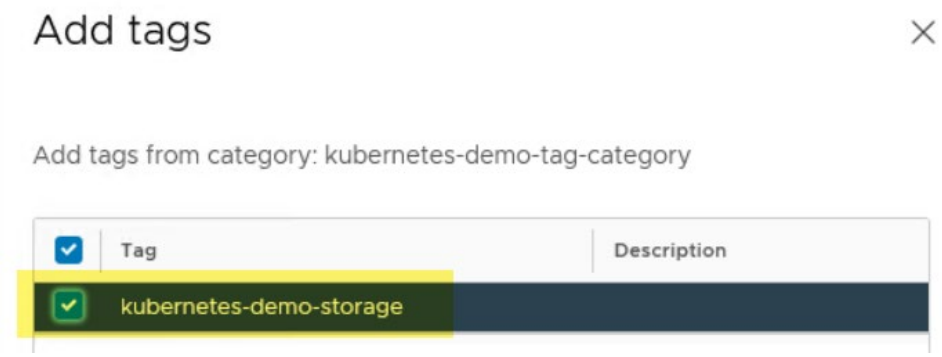
Select "Enable tag based placement rules " then click Next



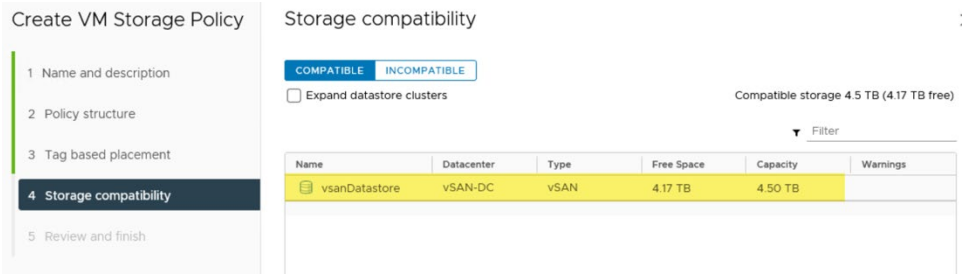
Tag-based placement -> For Tag Category select **kubernetes-demo-tag-category**



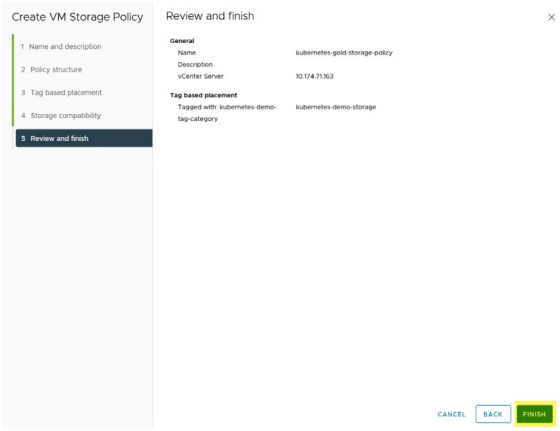
Click browse and select **'kubernetes-demo-storage'** then click OK then Next



Under Storage Compatibility you should see the datastore selected in above steps. Click Next



Click Finish



Storage policies visible to a vSphere Namespace determine which datastores the namespace can access and use for persistent volumes. The storage policies appear as matching Kubernetes storage classes in the namespace. They are also propagated to the Tanzu Kubernetes cluster on this namespace.

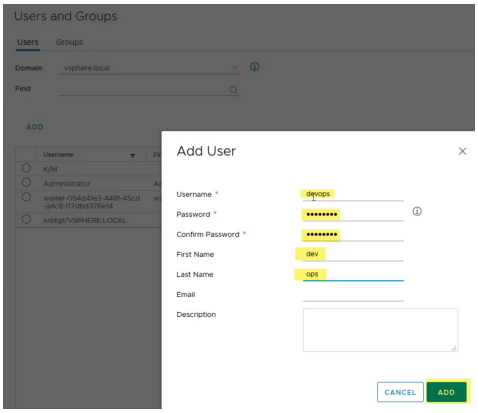
Add a DevOps user

If your vCenter is joined to an LDAP or Active Directory you can substitute the "**devops**" user with a user from that identity store. For the purposes of the demo we are assuming you do not have that available. Instead, we will create a user called "**devops**" in the **vSphere.local** identity store. This user will be the one running the **kubectl** commands later in the document.

To create the **devops** user do the following:

1. Menu-> Administration -> Users and Groups

2. Select Domain -> **vSphere.local** -> Add User
3. Add the user "**devops**" password **VMware1!** - confirm password -> Add



Create Content Library

In this step we will add a subscribed Content Library. This Content Library contains the latest TKG cluster images that will be deployed to create TKG clusters.

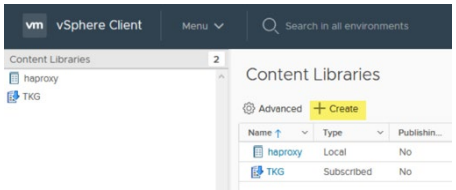
Create Content Library PowerCLI Example

```
#Set up the content library needed by vSphere with Tanzu
New-ContentLibrary -Datastore $datastore -name "tkg-cl" -AutomaticSync -
SubscriptionUrl "http://wp-content.vmware.com/v2/latest/lib.json" -Confirm:$false
```

Create Content Library vCenter UI Example

Do the following:

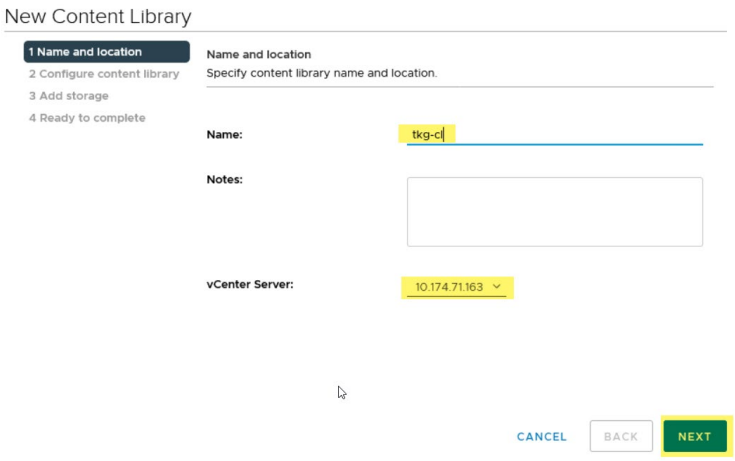
1. Menu -> Content Libraries
2. Click “Create”



3. Enter a name and location
    a. Enter the name: **tkg-cl**

b. Choose the vCenter Server

c. Click Next



4. Configure the Content Library

   a. Select “Subscribed Content Library”

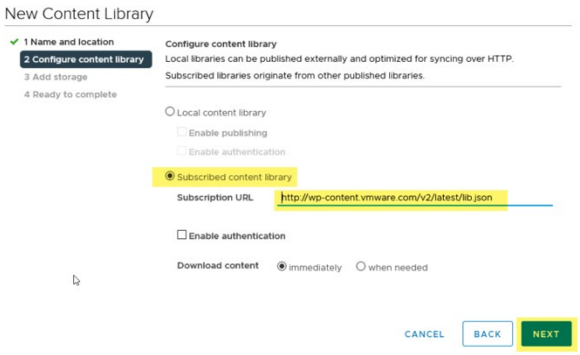
   b. Enter the following URL:

      i. *http://wp-content.vmware.com/v2/latest/lib.json*

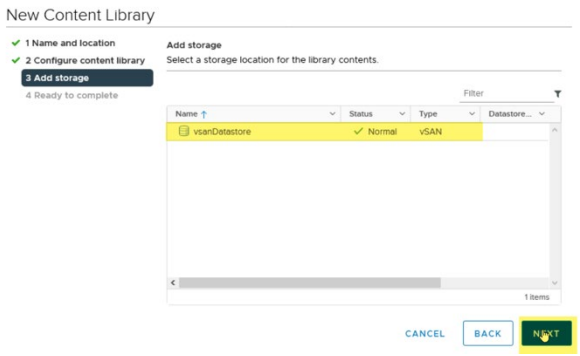
   Note: This will pull down the latest TKG content directly from VMware. All content is digitally signed and regularly updated.

   If you are running systems that are not connected to the Internet there are steps documented in the vSphere documentation on how to get the TKG content.

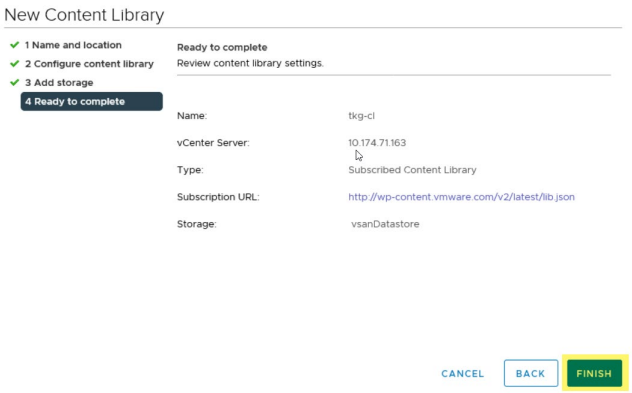
   c. Click Next



5. Add Storage
    a. Choose the shared storage option containing your storage policy you configured earlier
    b. Click Next



6. Click Finish



HAProxy Installation

You are now ready to deploy the HAProxy Load Balancer. Let's decide on our network configuration and then collect the information we are going to need to accomplish this task.

Note: You can get a copy of HAProxy from github.com. The location for the HAProxy OVA that has been updated to work with vSphere with Tanzu is here:

https://github.com/haproxytech/vmware-haproxy

First, we will need an IP address and DNS address on the Management Network. This must be a static IP.

Management Network Load Balancer IP	10.174.71.50
Management Network Gateway IP	10.174.71.253

Note: Deploying this appliance is the equivalent of deploying a piece of L2, networking infrastructure.

For example: The IP range selected for the virtual servers will be reserved by the load balancer appliance. This means if the VIP range is 10.174.72.0/24, and there happens to be a gateway on 10.174.72.253, anyone or anything trying to access a host on 10.174.72.0/24 is going to have a bad time. The appliance will argue it owns 10.174.72.253, any routes that require the gateway

10.174.72.253 failing in the process. Please plan carefully when you decide how you want to configure the Workload Network.

/24 Example

For the example below we are going to use a full /24 subnet for the Workload Network.

Workload Network Load Balancer IP	10.174.72.2
Workload Network Gateway IP	10.174.72.253
Cluster Node Range = 10.174.72.0/25	10.174.72.1-10.174.72.126
Virtual IP Range = 10.174.72.208/28	10.174.72.209-10.174.72.222

We will fine tune down to the IP so that the Load Balancer doesn't attempt to own the Gateway IP. Based on the values above you will see that we have approximately 124 usable IP addresses set aside for Supervisor Clusters, TKG Clusters, etc.

For the virtual IP range, we have 126 IP addresses set aside.

/26 Example

If a /24 is too much you could go with a smaller subnet and change the values accordingly. For example, let's say you were given 10.174.72.0/26 which is 62 addresses.

Workload Network Load Balancer IP	10.174.72.2
Workload Network Gateway IP	10.174.72.253
Cluster Node Range = 10.174.72.0/27	10.174.72.3-10.174.72.30
Virtual IP Range = 10.174.72.32/27	10.174.72.33-10.174.72.62

Fine Tuning IPs

To “fine tune” to exclude the gateway or other IP addresses you can put in multiple CIDRs to create “blocks” of IP addresses. For example, if your gateway was 10.174.72.1 and you were given 10.174.72.0/25 as your CIDR range to work with then you would have 126 IP addresses starting at 10.174.72.1 to .126.

To exclude the .1 address and set aside approximately 50 addresses for VIPs you could create some or all of the following CIDR ranges.

CIDR	Host Min	Host Max	Usable Ips
10.174.72.8/29	10.174.72.9	10.174.72.14	6
10.174.72.16/28	10.174.72.17	10.174.72.30	14
10.174.72.32/27	10.174.72.33	10.174.72.62	30

Deploy the Load Balancer

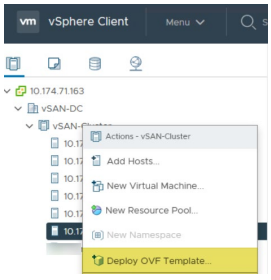
For vSphere 7 Update 1 With VDS networking, you need to supply your own load balancer. The first load balancer that is supported is HAProxy. In this section we will deploy it and use some of the values we have talked about above.

If you wish to view the full script using in this document and automate the deployment of the OVA and setup of the Content Library, tags and VDS, please check out the PowerCLI script available at the vSphere Tech Marketing Github page here:

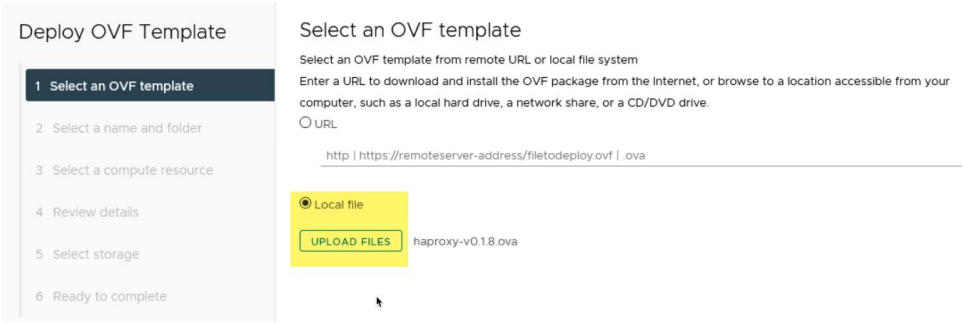
https://github.com/vsphere-tmm/Deploy-HAProxy-LB

Download the OVA from: *https://github.com/haproxytech/vmware-haproxy*

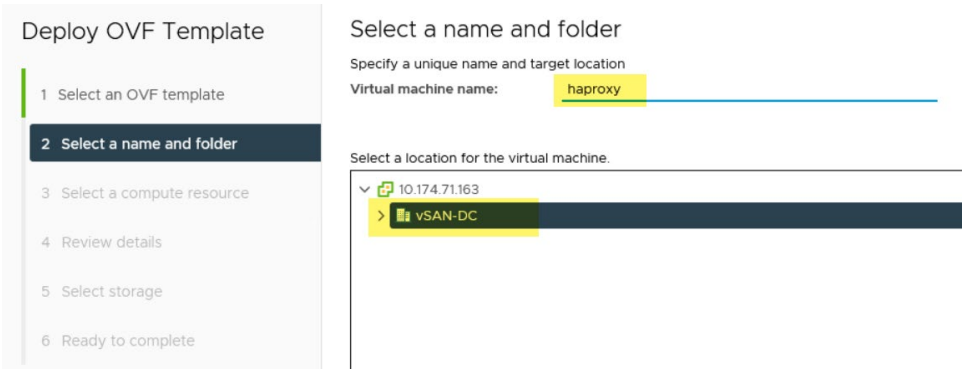
Right-Click on your cluster and select Deploy OVF Template.



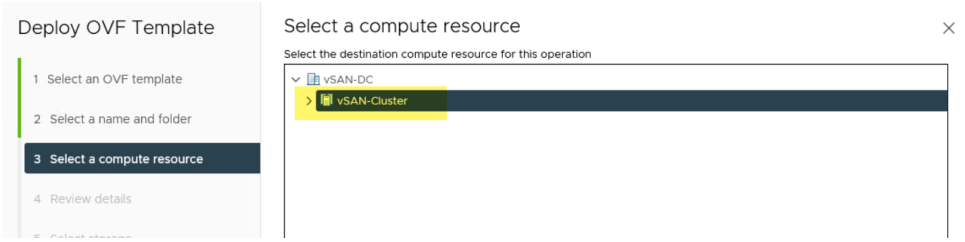
Next, select Local File and click on Upload Files. Find the HAProxy OVA and click Next



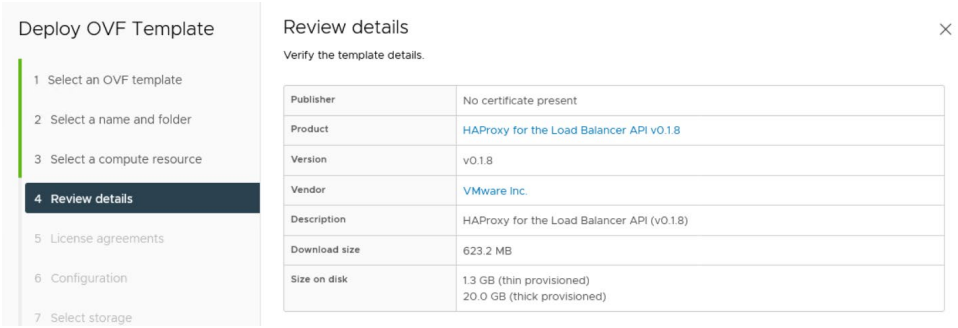
Next, give the VM a name and select where you are going to deploy it to in the folder hierarchy. Click Next.



Select the compute resource you are going to deploy to and click Next.

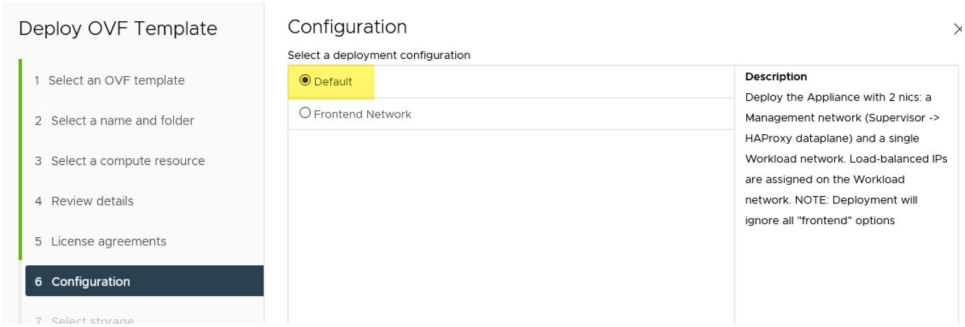


Review the details and click Next

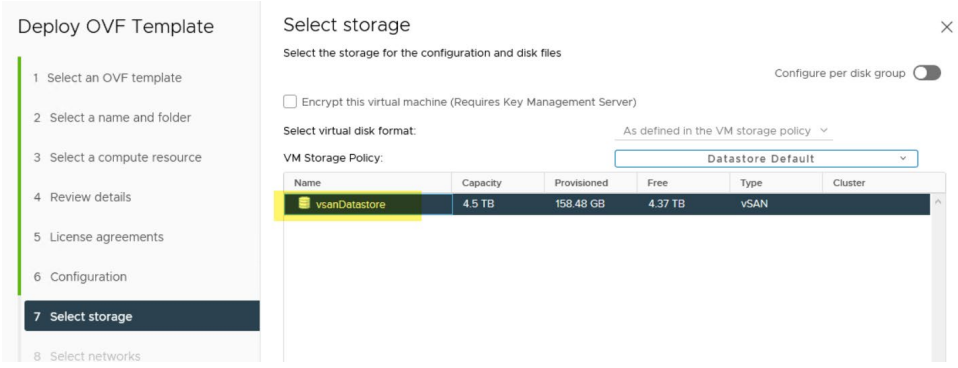


Accept the License Agreements and click Next

In the next screen you are asked to select Default or Frontend Network. For the purposes of the evaluation we will select Default. You can read about the different options on this screen.

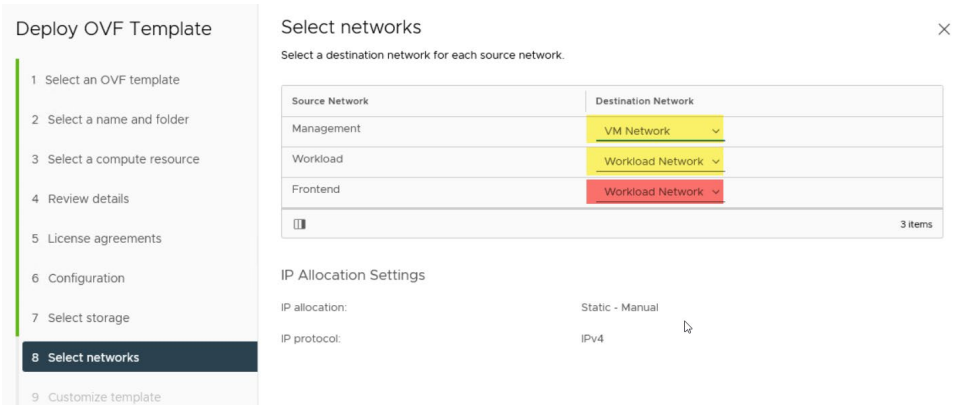


Select the storage you will be using for the VM



The next screen is where we select the networks used by the Load Balancer. When using HAProxy in the Default configuration the 3rd option of "Frontend" is displayed but not used when configured.

Leave it selected to whatever default network is in the dropdown. For Management, if you are still using “VM Network” for that network then select that. For Workload Management, select the Workload Management VDS Portgroup we created earlier. Click Next.



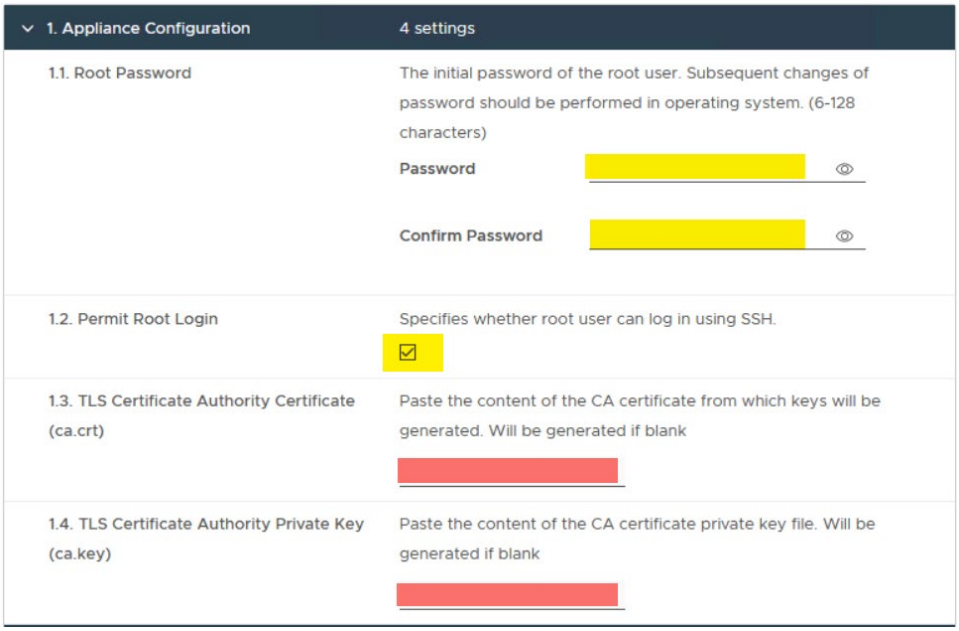
Any selection in Frontend will be ignored when using the Default Configuration

Customize HAProxy OVA Template

Appliance Configuration

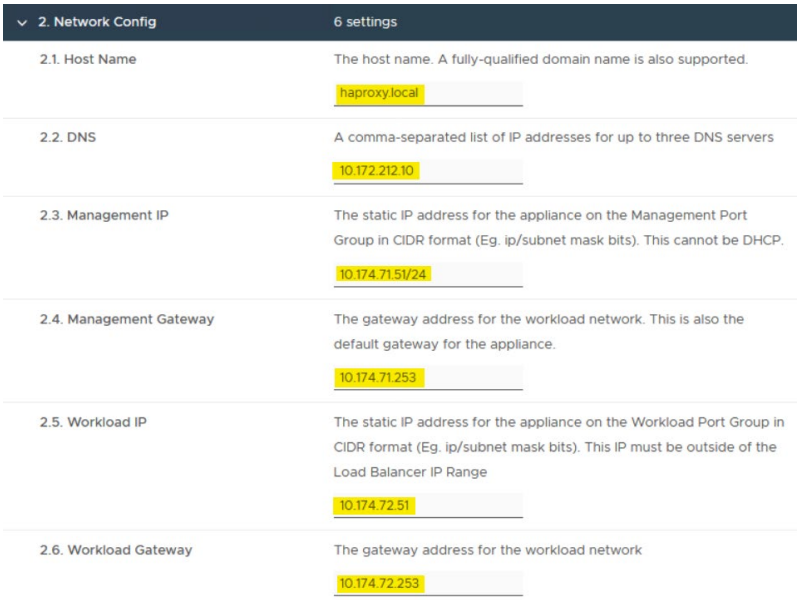
This is the section that requires you to have done your networking homework!

- Enter a password for the root account
- Select whether you wish to permit root login
- If you are using your own TLS certificate, then 1.3 and 1.4 should include the certificate (ca.crt) from which keys will be generated and the CA certificate private key. If you don’t wish to enter these values, then a self-signed certificate will be generated. For the purpose of the evaluation leave these values blank.



Network Configuration

- Enter the fully qualified host name for your load balancer
- Enter the DNS Address. If more than one, separate them using commas
- Enter the Management IP. This is the static IP address of the appliance on the Management Network. You can't use a DHCP address here. The value must be in CIDR format. E.g. 10.174.71.50/24.
- Enter the Management Gateway IP Address
- Enter the Workload IP. E.g. 10.174.72.2
- Enter the Workload Gateway IP Address

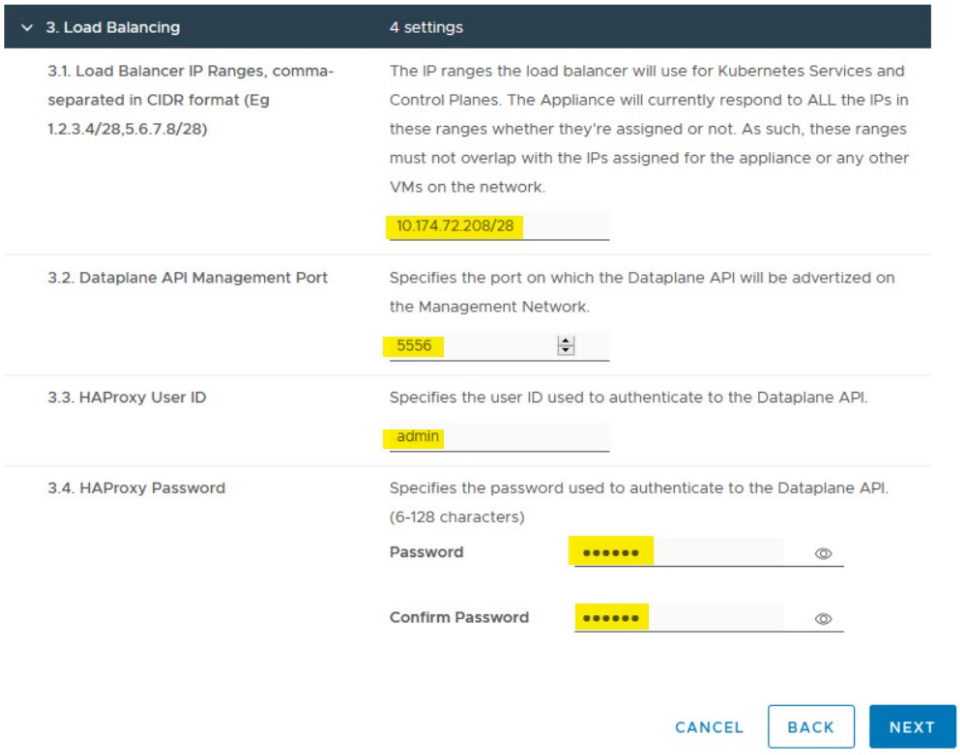


Load Balancing

- Enter the Load Balancer IP Ranges. These are the addresses for the virtual IP Addresses or VIPs used by the load balancer. The load balancer will respond to each of these IP addresses so once you select this range you can't "give them up" to something else. In the example below I'm using 10.174.72.208/28. This gives me 14 addresses for VIPS. In the examples above this was set to 10.174.72.128/25, giving us 126 VIPs.

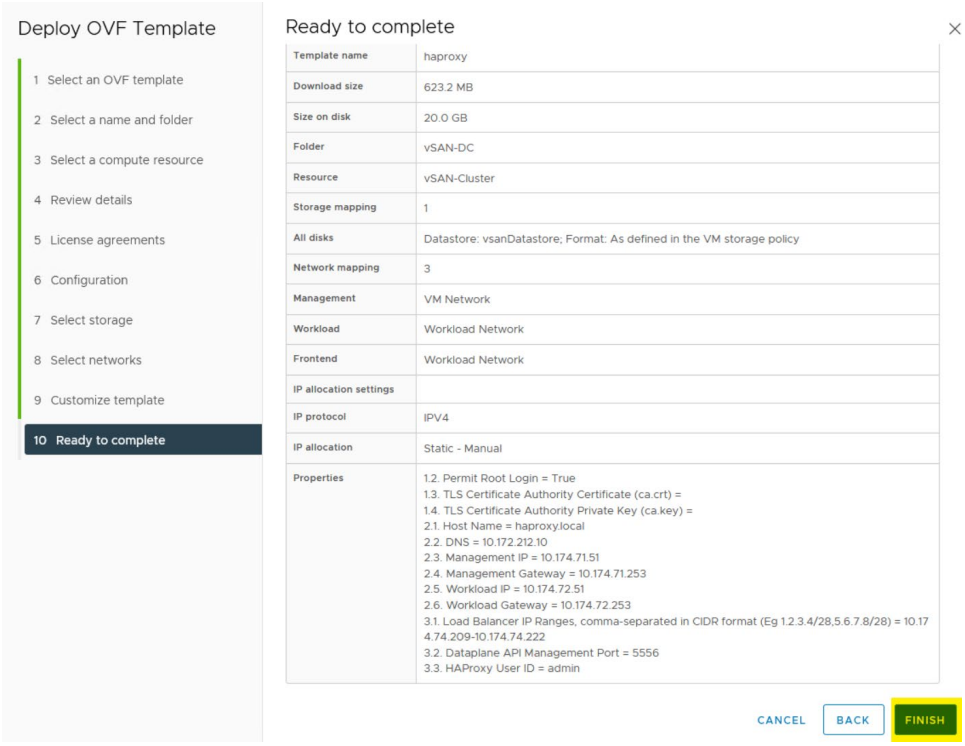
- Enter the Dataplane API Management Port. This is typically 5556. This will be combined with the Management IP address when we set up vCenter.

- Enter a username and password for the Load Balancer Dataplane API and click Next



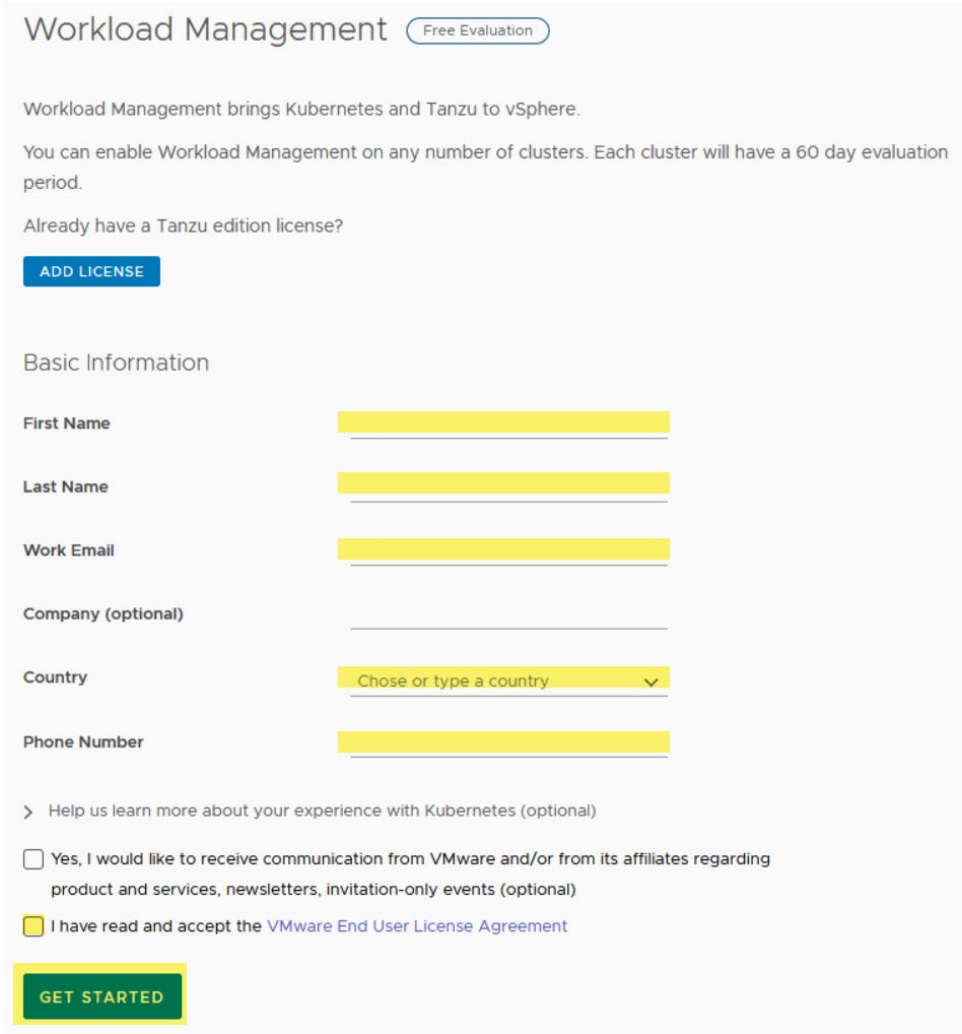
Ready to Complete

We are now ready to deploy the Load Balancer. Review the values you set and click Finish.



Enable Workload Management

With vSphere 7 with Tanzu you get a 60-day evaluation period. In order to enable this, you go to Menu…Workload Management and fill in the contact details so that you can receive communication from VMware. The image below shows all the required fields. Clicking on "I have read and accept the VMware End User License Agreement" will validate the entries. Now click Get Started



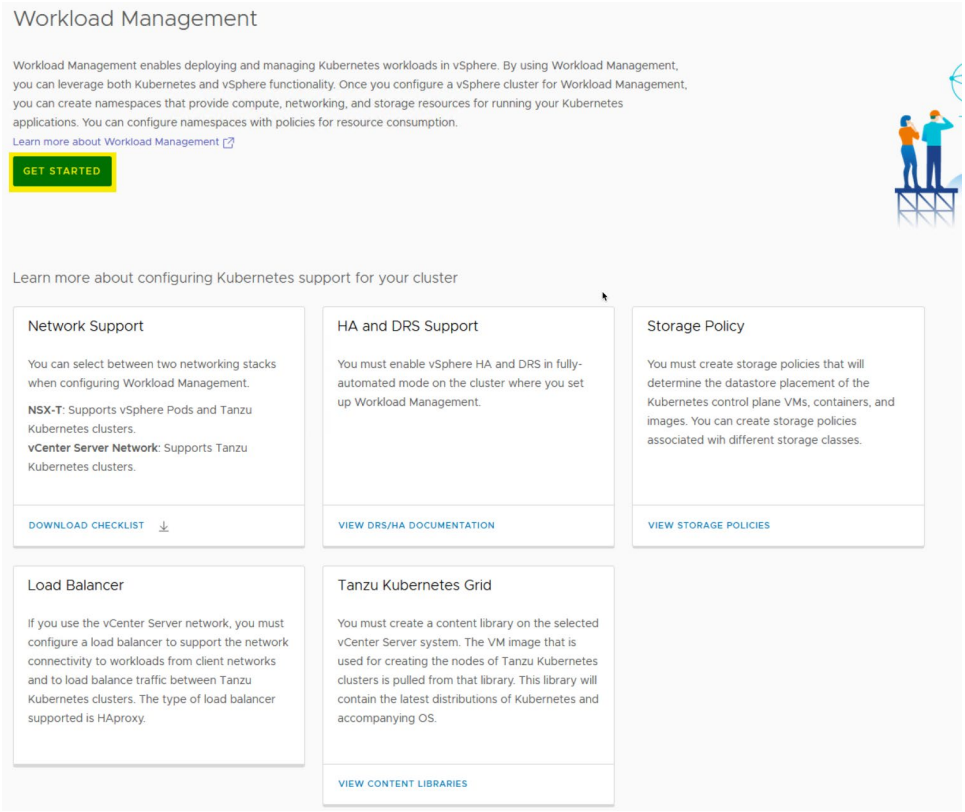
You can use the Workload Management functionality during a 60-day evaluation period. However, you must assign the Tanzu Edition license to the Supervisor Cluster before the evaluation period expires.

When the evaluation period of a Supervisor Cluster expires, or the Tanzu Edition license expires, as a vSphere administrator you cannot create new namespaces or update the Kubernetes version of the cluster. As a DevOps engineer, you cannot deploy new Tanzu Kubernetes clusters or change the configuration of the existing ones, such as add a new node and similar. You can still deploy workloads on Tanzu Kubernetes clusters, and all

existing workloads continue to run as expected. All Kubernetes workloads that are already deployed continue their normal operation.

Workload Management Setup

After you've filled out the license or evaluation screen you are presented with the Workload Management setup screen. From here we will set up the networking support. At this stage we have enabled HA/DRS, set up storage policies, deploy the load balancer and set up the content libraries necessary to continue. That leaves us with the network support setup.

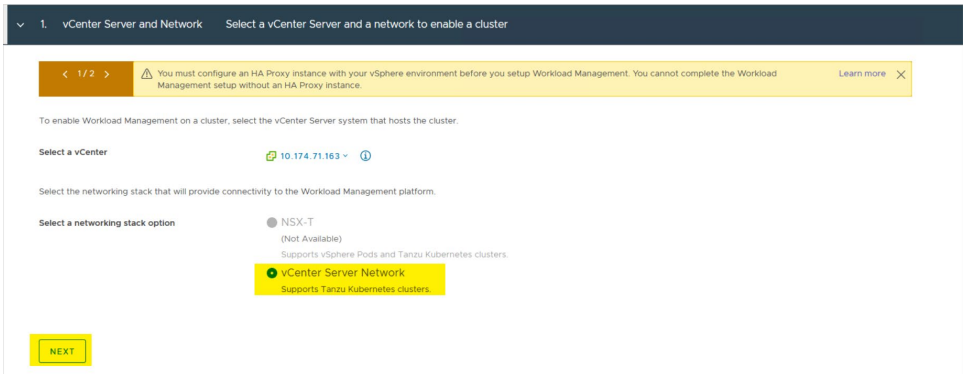


Review the content on the screen. Consider downloading the checklist. It is an Excel spreadsheet and is an excellent item to ensure you've covered all the bases. Click on Get Started.

vCenter Server and Network

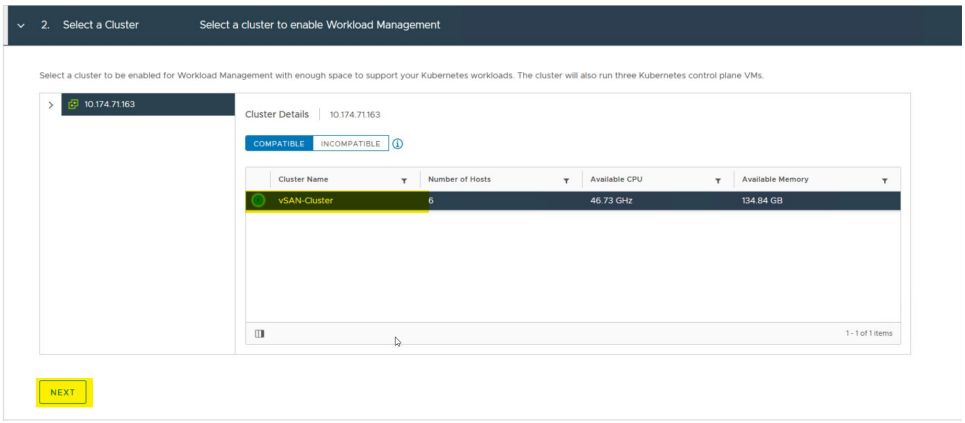
- Your vCenter should already be selected. Ensure it is correct.

- You will see that you have a choice of networking stacks. Because we haven't loaded NSX-T it will be greyed out and unavailable.
- Click Next.



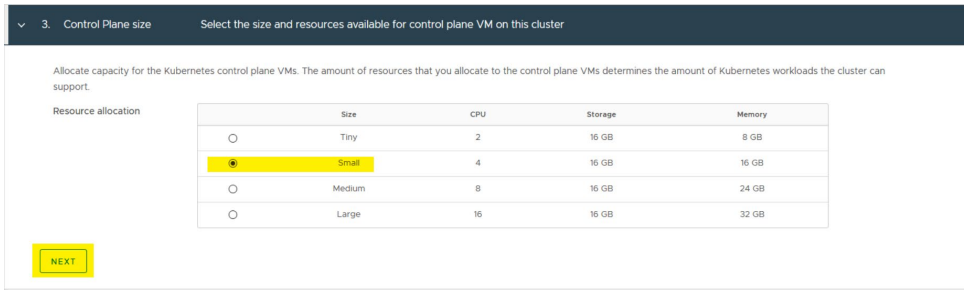
Select a Cluster

- Select the cluster you're using
- Click Next



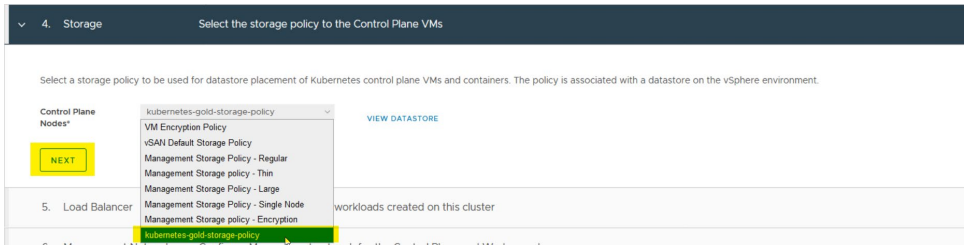
Control Plane Size

- Select the size of the resource allocation you need for the Control Plane. For the evaluation, Tiny or Small should be enough.
- Click Next.



Storage

- Here we will select the storage policy we configured previously
- Click Next

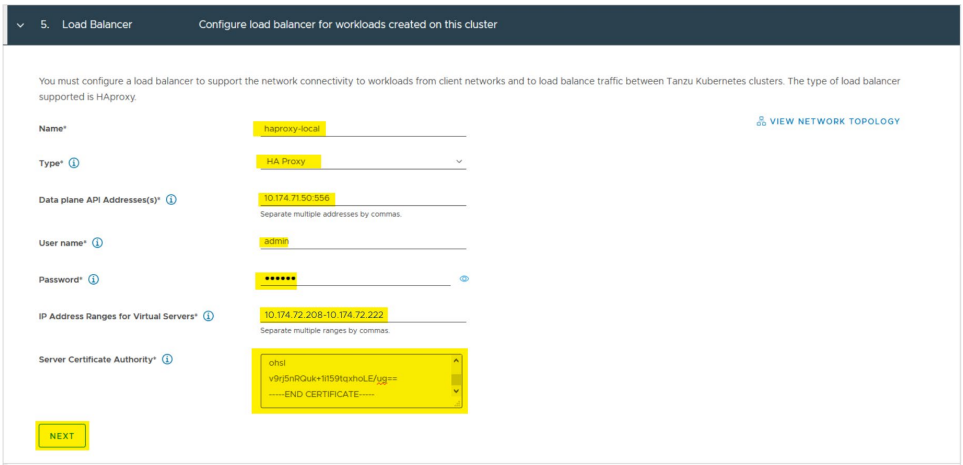


Load Balancer

In this section we will use some of the data collected during the deployment of the load balancer.

- Enter a DNS-compliance, immutable name. No underscores. Use lower case letters a-z, numbers 0-9, and hyphens E.g. “haproxy-local”
- Select the type of Load Balancer: HA Proxy
- Enter the data plane IP Address. This is the Management IP address AND the port number. In the example here it’s 10.174.72.50:5556. In the worksheet above it would be 10.174.72.2:5556
- Enter the username and password used during deployment for the Data plane API user.

- Enter the IP Address Ranges for Virtual Server. This is NOT the example used for the VIPs. This is the range of IP addresses that will be used in the Workload Network by TKG clusters.
- Finally, enter in the Server CA cert. If you have added a cert during deployment, you'd use that. If you have used a self-signed cert then you can retrieve that data from the VM. The easiest method and does not require you to log into the VM is to get the information from the HAProxy VM's advanced settings. This code sample of PowerCLI will retrieve it. Alternatively, you can get it from the vCenter UI, but you will have to convert the string from BASE64.



HAProxy Certificate Retrieval Code Example

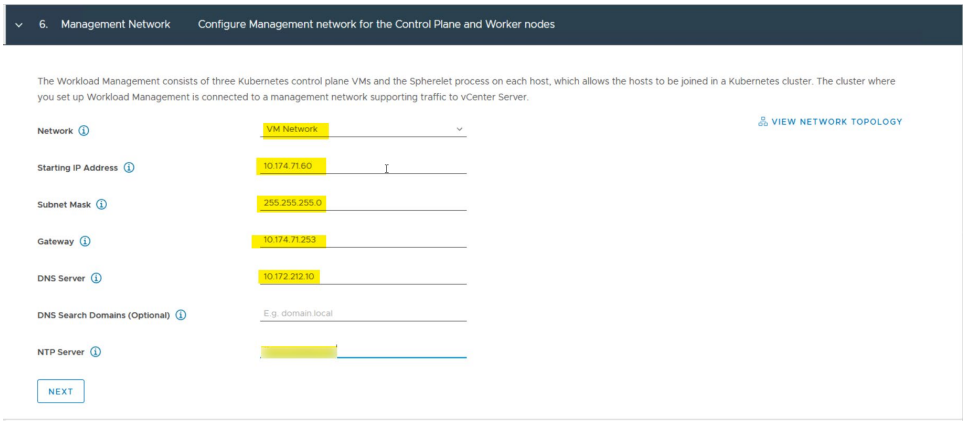
Change the value of $vc, $vc_user, $vc_password and $VMname to match yours.

```
$vc = "10.174.71.163"

$vc_user = "administrator@vsphere.local"

$vc_password = "Admin!23"

Connect-VIServer -User $vc_user -Password $vc_password -Server $vc

$VMname = "haproxy-demo"

$AdvancedSettingName = "guestinfo.dataplaneapi.cacert"

$Base64cert = get-vm $VMname |Get-AdvancedSetting -Name $AdvancedSettingName
```

```
while ([string]::IsNullOrEmpty($Base64cert.Value)) {

        Write-Host "Waiting for CA Cert Generation... This may take a under 5-10 minutes as the VM needs to boot and generate the CA Cert (if you haven't provided one already)."

        $Base64cert = get-vm $VMname |Get-AdvancedSetting -Name $AdvancedSettingName

        Start-sleep -seconds 2

    }

    Write-Host "CA Cert Found... Converting from BASE64"

    $cert = [Text.Encoding]::Utf8.GetString([Convert]::FromBase64String($Base64cert.Value))

Write-Host $cert
```

Management Network

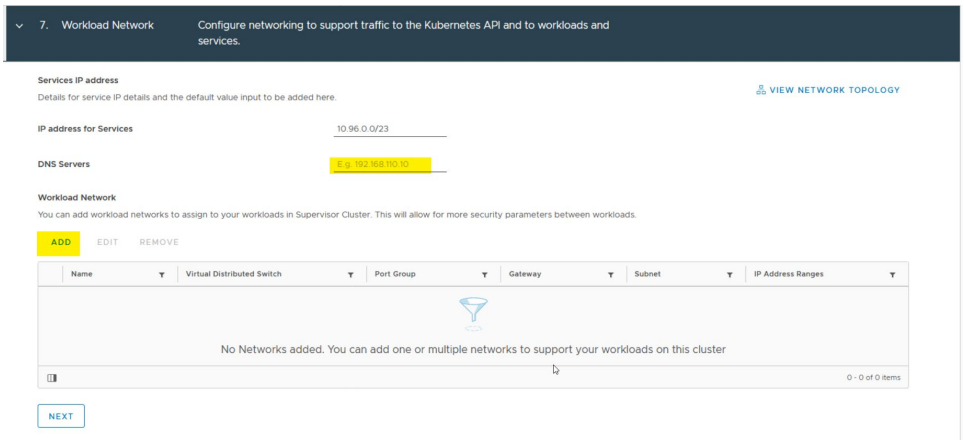
- Select the network used for the Management Network. In this case I'm selecting "VM Network"
- Enter the Starting IP Address. This is the first IP in a range of 5 IPs to assign to Supervisor control plane VMs' management network interfaces. 1 IP is assigned to each of the 3 Supervisor control plane VMs in the cluster, 1 IP is used for a Floating IP, and 1 is reserved for use during upgrade.
- Enter the subnet mask of the Management Network
- Enter the Gateway IP address
- Enter your DNS server(s)
- Optionally, enter your DNS Search Domains
- Enter your NTP Server
- Click Next



Workload Network

Here we are going to add your DNS Server and click on Add to start the process of adding the Workload Network. Typically, you can take the default network subnet for "IP Address for Services. Only change this if you are using that subnet elsewhere. This subnet is used for internal communication and it not routed.

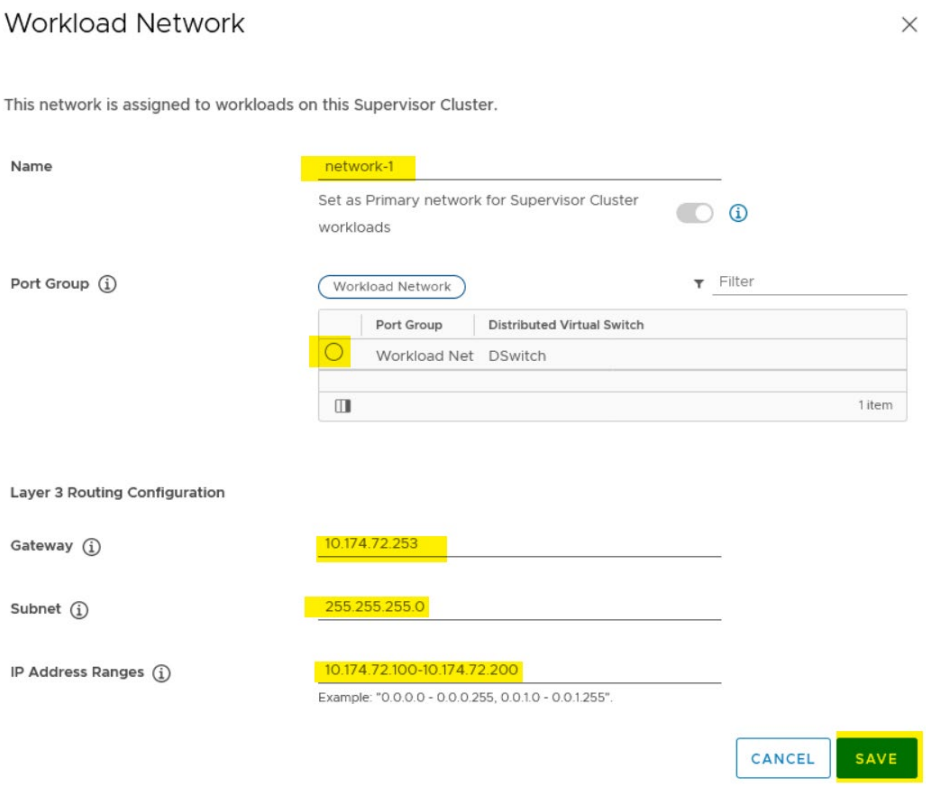
- Click Add



Adding the Workload Network

- Either create a new name or select the default
- Select the Workload Network Port Group on the vDS (Dswitch)

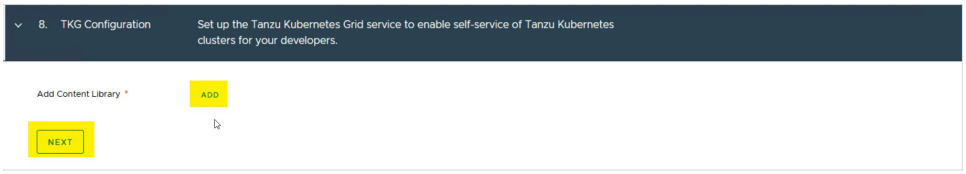
- Add the gateway for the Workload Network. To follow the worksheet above that would be 10.174.72.253
- Enter the subnet mask. For a /24 that is 255.255.255.0
- Enter the IP ranges used by resources like TKG clusters on this network. This is the "Cluster Node Range" referred to above. If you selected the whole /24 for your Workload Network when you configured your Load Balancer, then here is where you would be able to isolate out specific addresses by providing a range. To make things simple, let's put in 10.174.72.100-10.174.72.200.
- Click Save



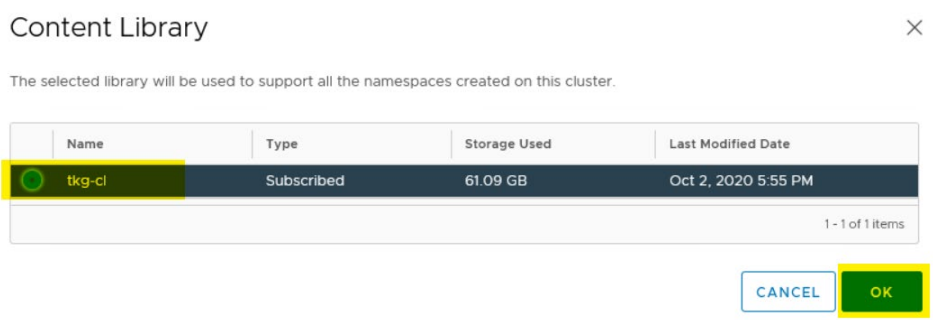
Now Click Next and we will move on to TKG Configuration

TKG Configuration

- Click on Add



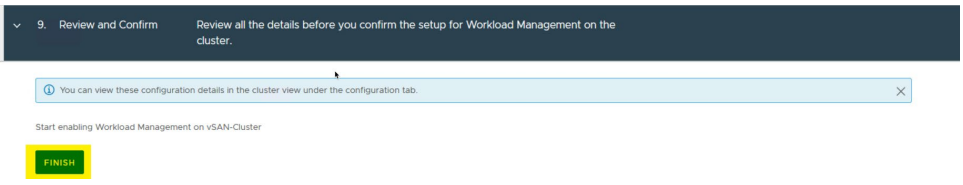
- Select the TKG Content Library we added previously
- Click OK
- Click Next



- Click Next

Review and Confirm

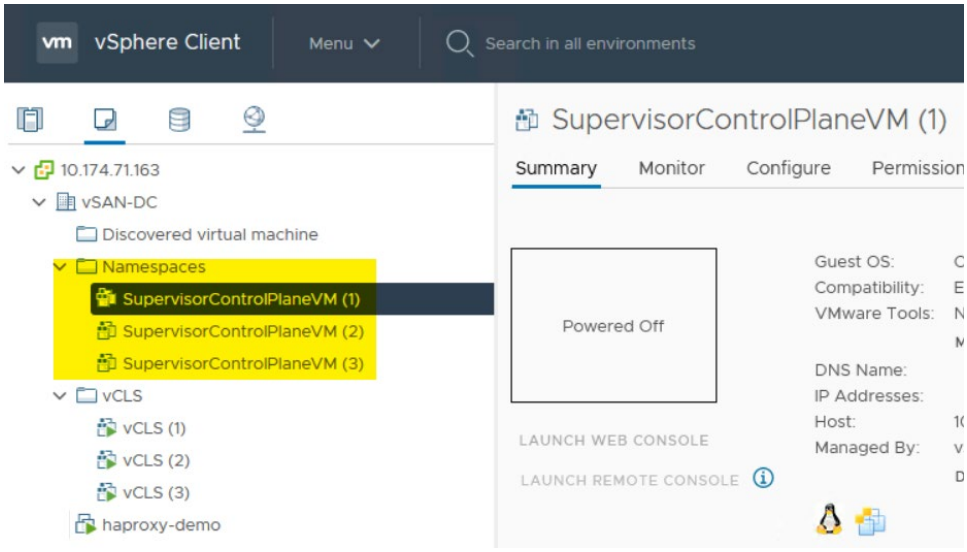
- Click Finish



During the process of configuring you will see the occasional message become available, updating you on the status of the configuration process. This will take a variable amount of time as several Supervisor Control Plane virtual machines are being provisioned.

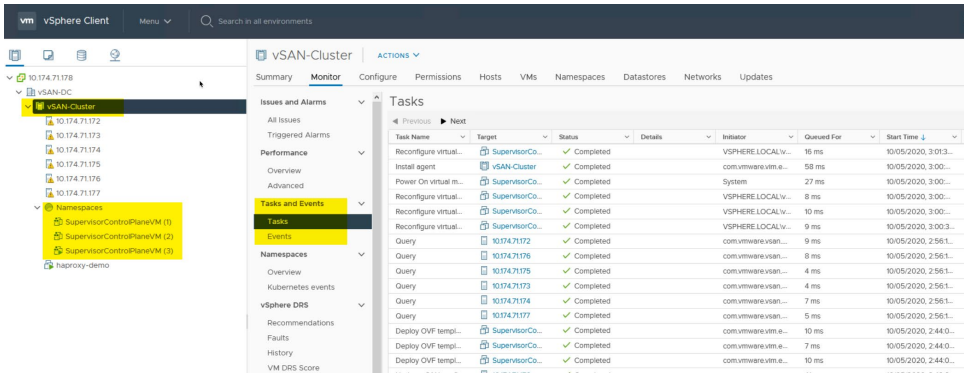
Monitoring Workload Network Configuration

During this process you will see a Namespaces folder be created and the Supervisor Control Plane virtual machines being provisioned into that folder.

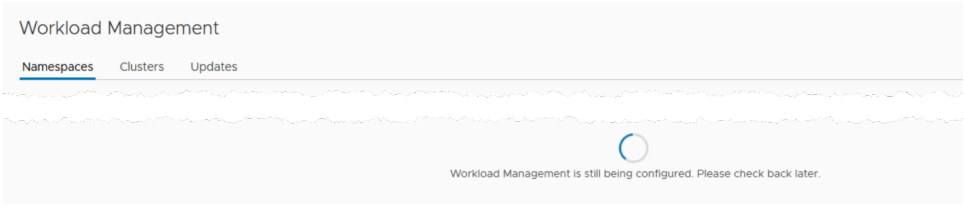


You can monitor the deployment of the VM's in the Tasks view for the vSphere Cluster. You may see some http errors from time to time. Not to worry, the Supervisor Cluster will keep retrying.

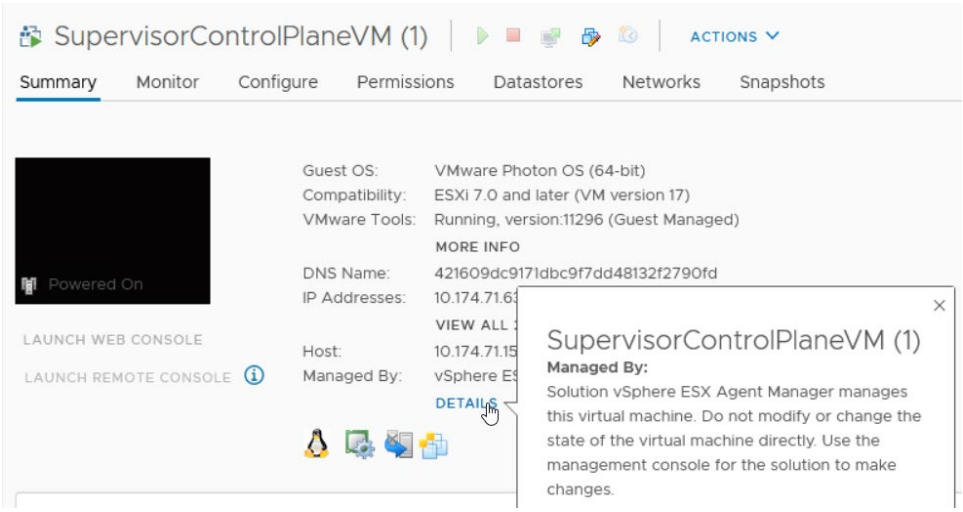
You can monitor the status of the configuration by watching the Tasks and Events pane in the vCenter UI for the vSphere Cluster you enabled Workload Management on.



If you go to Workload Management…Namespaces you will see this screen until configuration has completed. This can take a while (20+ minutes or more).

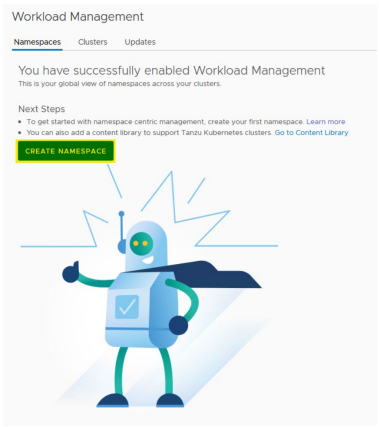


While you are waiting, notice that the SupervisorControlPlane VM's are somewhat unique. **You should not modify or change them in any way**. They are managed by vCenter.



Create a vSphere Namespace

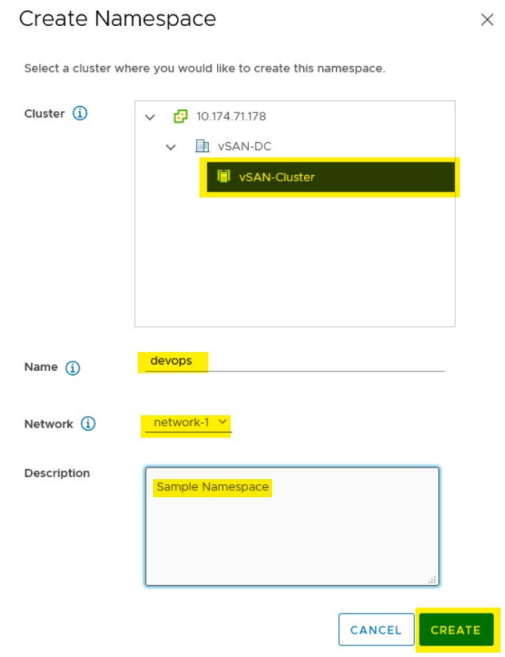
When the system is ready you will see under Workload Management-->Namespaces the following screen



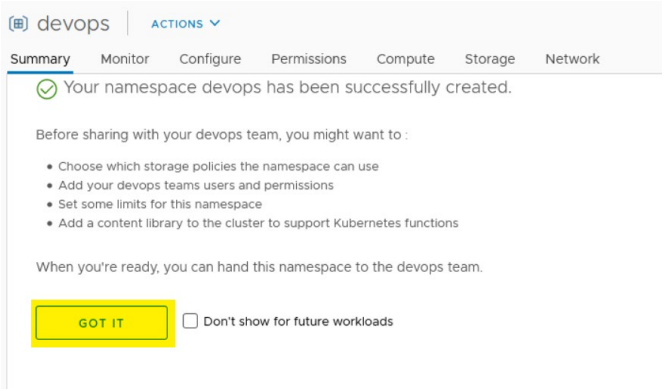
Click on Create Namespace.

Namespace Configuration

- Select the cluster
- Enter a name for your Namespace
- Select the network your Namespace will use
- Optionally add a description
- Click “Create”



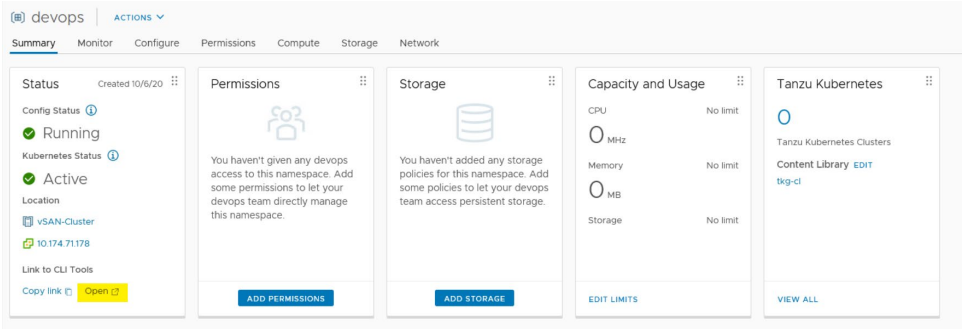
You will now be presented with the Namespace page for your new Namespace. Click on "Got It".



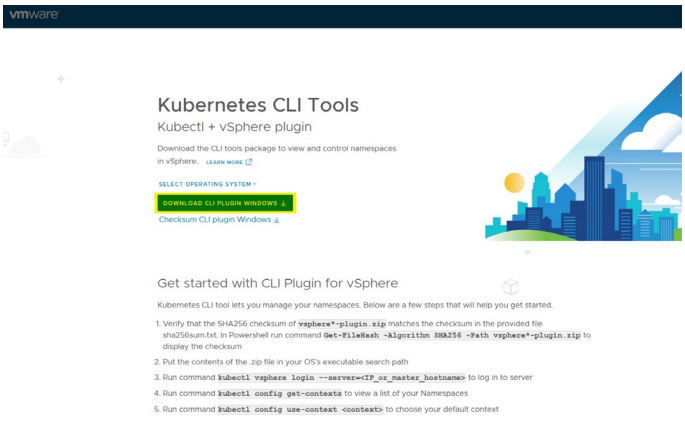
At this point you will configure several Namespace options

- Add Permissions

- Add Storage
- Configure Capacity and Usage



Let's break each one of these down. First, let's start with the link to the CLI tools. The CLI tools are the kubectl command that is the key method for developers to interact with Kubernetes. From here you can download a copy of Kubectl that will speak to vSphere. Click on Open and follow the instructions on downloading and installing Kubectl on your client OS.

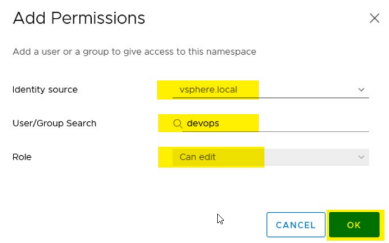


Namespace Permissions

Go back to the Namespace and click on Permissions. Here we will grant the previously created "devops" user the Edit permission on the namespace.

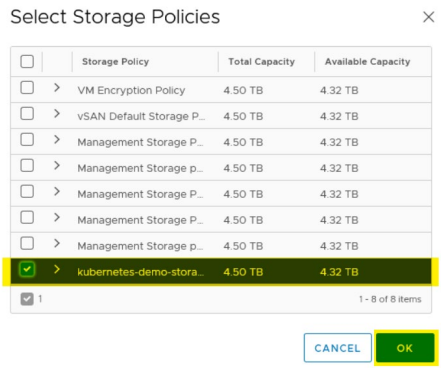
- Select the Identity Source (vsphere.local)
- Enter the username (devops)

- Select the Role (Edit)



Add Storage to the Namespace

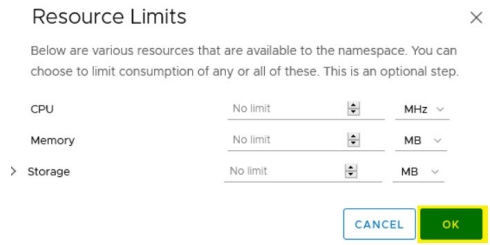
- Click on Add Storage
- Select the Kubernetes-demo-storage policy created earlier
- Click OK



Edit Namespace Resource Limits

Optionally you can edit the resource limits. After all, a Namespace IS a Resource Pool! For the purposes of this document and exercise, let's hold off on limits for now.

- Click on Edit Limits
- View the dialog box and optionally adjust the limits used by this namespace.
- Click OK



Use Case Examples

You are now ready for your first deployment of a TKG cluster!

Login as devops user

Let's confirm that you can login. From the system you have installed kubectl, enter the following:

```
kubectl vsphere login --server=https://10.174.72.209 --vsphere-username devops@vsphere.local --insecure-skip-tls-verify
```

Enter the password you chose for the devops user.

```
Password:
```

You should see the following response:

```
Logged in successfully.

You have access to the following contexts:

   10.174.72.209

   devops

If the context you wish to use is not in this list, you may need to try

logging in again later or contact your cluster administrator.

To change context, use `kubectl config use-context <workload name>`
```

Now change your kubectl context to the namespace you created

```
kubectl config use-context devops

Switched to context "devops".
```

Where to go for more on using Kubernetes

If all of that worked then you are now ready to move on to the VMware's Gitlab we have shared a set of instructions on how to deploy TKG cluster workloads in your new environment. Please see the link below.

Deploy a workload on the TKC Cluster

https://github.com/vsphere-tmm/vsphere-with-tanzu-quick-start

Share with users/developers

Ultimately you want to have your development team try out your new PoC. Create a namespace for them, give them permissions, set resources and share with them the IP address to download the kubectl binary and the IP address to connect kubectl to the PoC. You can then share with them the GitHub page that we have created, and they can try the example there or they can start uploading their own code to try out.

Next Steps

Now that you have a working Proof of Concept up and running you may want to now consider how you are going to enable vSphere with Tanzu in your existing vSphere installations.

As you've discovered in this exercise, your biggest challenge was probably the networking. The big takeaway is to plan, plan and then plan again. Ensuring your networking configuration is ready to be used by vSphere with Tanzu is key to a successful rollout. Ensuring you have your subnets, routers, gateways & VLANs all documented before deploying the load balancer and enabling Workload Management is key. Because so many networks are set up differently it is imperative that you work with your networking team to make this PoC a success.

In closing

We would like to take this opportunity to thank you for getting this far in this "quick" start guide. If you have feedback please send it via Twitter to @mikefoley and @mylesagray. We will be updating this document based on your feedback.

Glossary

TKG cluster	Tanzu Kubernetes Grid cluster. A fully upstream conformant Kubernetes cluster. a.k.a. TKC
Supervisor Cluster	a.k.a. SV or SC. This is the control plane running on vSphere that enables the deployment and management of TKG clusters.
Load Balancer or LB	A virtual machine used to load balance traffic between ingress networks and workloads. HAProxy is used in this configuration but more load balancers will be introduced.
vDS	vSphere Distributed Switch
WCP	Workload Control Plane
kubectl	The Kubernetes command-line tool, *kubectl*, allows you to run commands against Kubernetes clusters.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com.