Ubuntu 20.04 Template for XenServer

Introduction

This document details the steps required to configure a template XVA for XenServer.

XVA Template Creation

Follow these instructions to create an Ubuntu-20.04 XVA that can be used to quickly deploy Kubernetes to XenServer. Note that these steps will only be **rarely necessary** to regenerate the XVA from scratch. Most of the time, you’ll be able to quickly **clone the pre-built XVA** downloaded from Amazon S3 from (where **#** is the Ubuntu revision):

https://s3.amazonaws.com/neonforge/kube/xenserver-ubuntu-20.04.#.xva

or the latest version from:  
  
 <https://s3.amazonaws.com/neonforge/kube/xenserver-ubuntu-20.04.latest.xva>  
  
NOTE: These files are compressed via **GZIP**.

**Setup Instructions**

1. Download the **Ubuntu-20.04 Server ISO** from (where **#** is the desired revision) from:   
     
   http://releases.ubuntu.com/20.04/

Edit the downloaded file name to remove the last version field (probably “.0”) and the “live-server-amd64” and then archive the ISO file to (making it public):  
  
<https://s3.amazonaws.com/neonforge/kube/ubuntu-20.04.#.iso>

1. Burn the ISO to a DVD (or add it to a XenServer ISO library).
2. Power up XenServer and wait for the UX to display on the monitor.
3. Run XenCenter/XCPCenter on your workstation and connect to the XenServer.
4. Create the setup VM (You’ll need an **XCP-ng v7.5.0** server, newer versions (like v8.1.0) generate VM templates that are **not compatible with older servers**:  
   1. **VM/New VM…**
   2. Scroll to the end and select **Ubuntu Xenial Xerus 16.04** and click **Next**
   3. Name the VM: **xenserver-ubuntu-20.04-neon**
   4. **Insert the DVD** (or select the ISO from the library) into the XenServer machine (the new VM will mount this by default).
   5. Click **Next** until you get processors and memory page. Set **4 procs** and **1024MB RAM**.
   6. Click **Next** until the last page and click **Create Now** to create the VM.
   7. **Wait for the VM to be created** in the left panel, select it, and then click the **Console tab** and wait for the VM to boot.
   8. **Wait a minute or two** for the VM to obtain an IP address via DHCP.
5. Wait for the installation UX to start.
6. Press enter to select **English**.
7. **DO NOT UPDATE** to the **latest installer** if asked.
8. Press enter to **Install**.
9. Press enter to **skip proxy** configuration.
10. Press enter to choose the **default mirror**.
11. **Use the entire disk**.
12. **Confirm the file system settings**.
13. **Continue with setup** (there’s no going back 😊)
14. Enter **sysadmin** as your **name** and **username** and enter **sysadmin0000** as the password. Set the server name to **ubuntu** and select **done**.
15. Install **OpenSSH**.
16. Install **NO Server Snaps**.
17. Let the installation get to the **security update** step and then **cancel the update and reboot** (the idea is that we want our templates to match the Ubuntu releases and that we’ll handle updates separately).
18. Select **Reboot Now** and then press **ENTER** to reboot (you may see a failed CDROM eject message: ignore that and just press **ENTER**).
19. **Login** with the credentials you specified earlier to verify that the VM works.
20. Use the command below to discover the VM’s **IP address** for the **eth0** interface. You’ll need this later to finish preparing the node.

ip address

1. **IMPORTANT:** Be sure to **logout** of any PuTTY or XenServer consoles terminals so the next step will work.
2. Run this command **on your workstation** to complete the preparation, passing the IP address of the VM:  
     
   neon prepare node-template --xenserver IPADDRESS
3. Install the **XenServer/XCP-ng tools**:  
   1. **right-click** on the VM in XenCenter and click **Install XenServer Tools**. You may need to wait a minute or two for the menu item to show up.
   2. **SSH into the VM** and login.
   3. **mount** the tools virtual DVD drive, install the tools:  
        
      sudo bash  
      mount /dev/dvd /mnt  
      /mnt/Linux/install.sh -n  
      eject /dev/dvd
   4. **Eject** the **DVD drive** in XenCenter (at the top-right of the VMs storage tab).
   5. **Cleanup and shut down:**apt-get cleanrm -rf /var/lib/dhcp/\*  
      sfill -fllz /  
      shutdown -h now
4. Create the **XVA template**:  
   1. Right-click the VM and **Convert to Template**.
   2. Right-click on the template and select **Export to file…** Save the file to your workstation as **xenserver-ubuntu-20.04.#.xva** where **#** is the revision, set the **format to XVA** and then export to a local file.
5. Copy the file somewhere and then GZIP it:  
     
   gzip --best xenserver-ubuntu-20.04.#.xva
6. Use the AWS Console to **Upload** the image to the location below (where **#** is the revision:

Then upload the ZIP file to S3 (**removing the .gz** extension and setting **Content-Encoding=gzip**):  
  
<https://s3.amazonaws.com/neonforge/kube/ubuntu-20.04.#.vhdx>   
  
Be sure to: **Remove the .gz extension.**Be sure to: **Add AWS metadata: Content-Encoding = gzip**Be sure to: **Make these files public on AWS!**

1. If this is the latest Ubuntu image, we’ll also build a fully upgraded template:  
   1. In XenCenter, rename the most recent Ubuntu point release template from **xenserver-xenserver-20.04-neon** to **x-xenserver-ubuntu-20.04-neon**.
   2. Create a VM named **xenserver-ubuntu-20.04-neon** from the **x-xenserver-ubuntu-20.04-neon** template.
   3. Start the VM, login via the Console tab and **get the IP address**.
   4. SSH into the VM using PuTTY and run these commands to **upgrade Ubuntu**:  
        
      sudo bash  
      apt-get update  
      apt-get dist-upgrade -yq
   5. And then these commands to **clean the disk** and **shutdown**:  
        
      apt-get clean   
      rm -rf /var/lib/dhcp/\*  
      sfill -fllz /  
      shutdown -h now
   6. You can **delete** the **x-xenserver-ubuntu-20.04-neon** template.
   7. Convert the VM to a template and export the XVA file like we did above and then compress and upload it to S3:  
        
      <https://s3.amazonaws.com/neonforge/kube/ubuntu-20.04.latest.vhdx>   
        
      Be sure to: **Remove the .gz extension.**Be sure to: **Add AWS metadata: Content-Encoding = gzip**Be sure to: **Make these files public on AWS!**

**XenServer Template Upload:** XenServer templates are GZIP compressed rather than using ZIP so the XenServer tooling can download them. The steps for creating a XenServer image is much like those for Hyper-V. Essentially, you’ll use XenCenter to create the VM by mounting the Ubuntu setup ISO, setting up the VM, installing the XenServer tools, and then exporting it as an **\*.xva** file.  
  
I generally go through the full template build from ISO only for Ubuntu point releases. For intermediate package upgrades, I do the following:  
  
In XenCenter, rename the most recent Ubuntu point release template from **xenserver-ubuntu-20.04-neon** to something like **temp-neon**.  
  
Create a VM named **xenserver-ubuntu-20.04-neon** from the **temp-neon** template and run these commands to upgrade it:   
  
sudo bash  
apt-get update  
apt-get dist-upgrade -yq  
rm -rf /var/lib/dhcp/\*  
shutdown -h now  
  
Convert the VM into a Xen Template and download/upload it to S3 as described further below.  
  
Delete the **kube-neon** VM and then **rename x-kube-neon** back to **xenserver-ubuntu-20.04-neon**.  
  
Follow the steps below compress and upload it to AWS S3:

* 1. Create and initialize an Ubuntu virtual machine on XenServer much like we did above for Hyper-V. Name the VM **xenserver-ubuntu-20.04-neon (this is important)** You’ll be using the console in XenCenter.
  2. After you **finished installing Ubuntu**: **right-click** on the VM in XenCenter and click **Install XenServer Tools**. You may need to wait a minute or two for the menu item to show up.
  3. Wait a minute or two for the tools driver to be mounted.
  4. SSH into the VM and **mount** the tools DVD drive:  
       
     sudo bash  
     mount -o ro,exec /dev/sr0 /mnt  
       
     You may need to modify the /dev/sr0 path. Execute blkid and use the device with the XenServer Tools label. You may need to wait a minute or two for the DVD to mount.
  5. **Install** the tools via and then reboot:  
       
     /mnt/Linux/install.sh  
     reboot
  6. **Login** back into the VM.
  7. **Eject** the **DVD drive** in XenCenter (at the top-right of the VMs storage tab) and shutdown the VM:  
       
     rm -rf /var/lib/dhcp/\*  
     shutdown -h now
  8. Right-click the VM and **Convert to Template**.
  9. Right-click on the template and select **Export to file…** Save the file as **xenserver-ubuntu-20.04.#.xva** where **#** is the revision, set the **format to XVA** and then export to a local file.
  10. GZIP compress the file and create a latest version if this is the latest. Note that we’re going to rename the file to remove the **.gz** file type after uploading to AWS:  
        
      gzip --best xenserver-ubuntu-20.04.#.xva
  11. If this is the latest Ubuntu image, restart and log back into the VM and then run:  
        
      sudo bash  
      apt-get update  
      apt-get dist-upgrade -yq  
      rm -rf /var/lib/dhcp/\*  
      shutdown -h now  
        
      After the VM has down, export it to **xenserver-ubuntu-20.04.latest.xva** and then GZIP it like:  
        
      gzip --best xenserver-ubuntu-20.04.latest.xva

1. Manually upload the files to S3 setting metadata to:  
     
   **S3** at [https://s3.amazonaws.com/neonforge/kube/\*](https://s3.amazonaws.com/neonforge/kube/*)   
     
   Be sure to: **Remove the .gz extension.**Be sure to: **Add AWS metadata: Content-Encoding = gzip**Be sure to: **Make these files public on AWS!**