Ubuntu Template Image Preparation and Cloning

About

Just some documentation I use to keep track of how to provision a new, fresh Ubuntu VM template on my homelab. Some values are going to differ in your environment, such as:

- Name
- Username
- Interface names (could vary as `enX0`, `ens160`, etc depending on your hypervisor or lack thereof)
- IP Addresses, Subnets, DNS servers, and search domains

Fresh Install of VM

Fresh Ubuntu Installation

- Select and hit enter on the **Try or Install Ubuntu Server** option (loads by default)
- Hit **Enter**, leaving the the default **English** option
- Hit **Enter**, leaving the defaults of **English (US)** and **English (US)**
- Hit **Enter**, leaving the default of **Ubuntu Server**
- **Arrow up** on the keyboard to **enX0 eth** and hit **Enter**
- **Arrow down** on the keyboard to **Edit IPv4** and hit **Enter**
- Hit **Enter** on **Automatic (DHCP)** and then **arrow down** on the keyboard to select **Manual**, then hit **Enter**
- Tab between fields:
  - Subnet: 192.168.160.0/23
  - Address: 192.168.160.xxx (Choose a free temporary IP from IP Address Allocations)
  - Gateway: 192.168.160.1
  - Name Servers: 192.168.160.105, 8.8.8.8, 8.8.4.4
  - Search domains: (blank)

    - Save (hit **Enter**)
- Tab to **Done** and hit **Enter**
- Hit **Enter**, leaving the proxy set to blank
- Hit **Enter** to accept the default mirror
- **Tab** to **Done** and hit **Enter** to accept the defaults for the storage layout
- **Arrow up** to **ubuntu-lv** (aka "mounted at /") and hit **Enter**
- Select **Edit** and hit **Enter**
- **Tab** to **Size** and change the value to the same as is listed next to **max** (probably ~?.???G)
- **Tab** down to **Save** and hit **Enter**
- **Tab** down to **Done** and hit **Enter**
- **Tab** to **Continue** and hit **Enter** on the Confirm destructive action dialog
- Profile Setup
    - Your Name: John Doe
    - Your server's name: ubuntu2404 (or ubuntu24041 etc for newer revisions)
    - Pick a username: jdoe
    - Choose a password: (my standard)
    - Confirm your password: (same)
    - **Tab** to **Done** and hit **Enter**
- Hit **Enter**, leaving the **Skip for now** option selected on the Upgrade to Ubuntu Pro
- Hit **Spacebar** on the option to **Install OpenSSH Server**
- If you have a GitHub account, you can optionally add your public SSH key
- **Tab** to **Done** and hit **Enter**
- **Tab** to **Done** and hit **Enter**, leaving defaults of nothing selected on Featured Server Snaps
- Wait for the "Cancel update and reboot" option to go away, which will be replaced with **Reboot now**

Post-install Ubuntu Configuration

- Return to the Remote Console window and **Power On** the VM
- Once it comes up, SSH to your VM
- Run any available upgrades and reboot

```
sudo apt update && sudo apt upgrade -y && sudo reboot
```

- Reconnect via SSH
- Set up Chrony for time sync

```
sudo apt install -y chrony && \
sudo systemctl restart chrony && \
chronyc tracking | grep --color=auto -e ^ -e "Last offset.*"
```

VIM Tweaks

While cool, some of the newer VIM plugins that come with Ubuntu can cause unexpected weirdness. Lets tweak the room VIM config so that it's less of a pain when doing a `sudo vi`

> Apparently `sudoedit` is a thing. That loads your personal .vimrc while executing edits as root. In which case, if you use sudoedit, you may want to add these tweaks to your personal ~/.vimrc

- Fix the auto-indentation of YAML when commenting out an existing line (for both your username and root)

```
cat << EOF >> ~/.vimrc
" Fix auto-indentation for YAML files
augroup yaml_fix
    autocmd!
    autocmd FileType yaml setlocal ts=2 sts=2 sw=2 expandtab indentkeys-=0#
indentkeys-=<:>
augroup END

EOF

sudo bash -c 'cat << EOF >> ~/.vimrc
" Fix auto-indentation for YAML files
augroup yaml_fix
    autocmd!
    autocmd FileType yaml setlocal ts=2 sts=2 sw=2 expandtab indentkeys-=0#
indentkeys-=<:>
augroup END

EOF'
```

- Set VIM as the default editor

```
sudo update-alternatives --config editor
```

Template Image Prep

Set up OpenSSH Key Reconfiguration

>

> If you simply clone an Ubuntu image without resetting the OpenSSH server host keys, an attacker can take those host keys and perform a MITM SSH attack on any system that was cloned from the same image. So we have to make sure those are reset before we make the image, and then automatically regenerated on the next boot.

- Copy/Paste/Run this entire chunk of script into your terminal (creates process that checks for missing keys at boot, and regenerates them if missing):

```
if [ `systemctl is-enabled openssh-reconfigure.service 2> /dev/null > /dev/null ||
true && false` ] ; then \
  echo "OpenSSH Key Reconfiguration Service already installed." ; \
else
  sudo bash -c 'cat << EOF > /usr/local/sbin/openssh-reconfigure
#!/bin/bash
test -f /etc/ssh/ssh_host_dsa_key || dpkg-reconfigure openssh-server
EOF'
sudo chmod 700 /usr/local/sbin/openssh-reconfigure
sudo bash -c 'cat << EOF > /etc/systemd/system/openssh-reconfigure.service
[Unit]
Description=OpenSSH Key Reconfiguration Service
Before=ssh.service

[Service]
Type=simple
ExecStart=/usr/local/sbin/openssh-reconfigure

[Install]
WantedBy=multi-user.target
EOF' ; \
  sudo chmod 644 /etc/systemd/system/openssh-reconfigure.service ; \
  sudo systemctl enable openssh-reconfigure.service ; \
fi
```

- Delete the existing keys

```
sudo /bin/rm -v /etc/ssh/ssh_host_*
```

Clear the Machine ID

- Run this:

```
sudo bash -c "truncate -s0 /etc/machine-id ; \
rm /var/lib/dbus/machine-id ; \
ln -s /etc/machine-id /var/lib/dbus/machine-id"
```

Genericize the netplan config

- Make these alterations to `/etc/netplan/50-cloud-init.yaml` in the `enX0` section. Leave the comments for the image/template user to understand what needs to happen to re-activate networking
  Ubuntu 24.04:

```
link-local: [ ipv4 ]   # Post-cloning, comment this out or remove
#link-local: [ ]       # Post-cloning, un-comment this line
#addresses:            # Post-cloning, un-comment this line
#- 192.168.160.xxx/23  # Post-cloning, un-comment this line and set appropriately
```

> Setting link-local allows the network interface to come up on boot, but without DHCP or a Static IP assigned. Additionally, setting link-local to a blank array ([ ]) after initial config disabled link-local addressing which can cause problems for default route handling in some cases.

Clear the Bash, VIM, and other history

- Run this:

```
rm -rf ~/.viminfo ~/.Xauthority ~/.cache
sudo bash -c 'rm -rf ~/.viminfo ~/.Xauthority ~/.cache'
sudo bash -c 'echo -n "" > /var/log/wtmp'
sudo bash -c 'echo -n "" > /var/log/btmp'
sudo bash -c 'echo -n "" > /var/log/lastlog'

### These should always run last
sudo bash -c 'truncate -s0 ~/.bash_history ; history -c'
truncate -s0 ~/.bash_history ; history -c
```

Shutdown

- Run this:

```
sudo shutdown -h now
```

Create the Template or Image

XCP-NG

Using XOA Xen Orchestra

- Go to **Home → VMs**
- Change the **Filters** to blank
- Click on your new VM
- Click on the name of the VM
- Change the name to: `TPL_U24.04_20240709172110`
  - `TPL` means this is a user-generated template
  - `U24.04` indicates this is Ubuntu Server 24.04
  - `20240709172110` Indicates the revision date of THE DOCUMENT YOU ARE READING in UTC, ie 2024 July 9th, 17:21:10. This is used as a means of revision control.
- Click on **Advanced**
- Click **Convert to template**
- Click **OK** on the confirmation dialog

Cloning to a New VM

PVE

- Right-click on the prepped VM and select **Convert to template**
- Click **Yes** on the confirmation dialog

XCP-NG

Using XOA Xen Orchestra

- Go to **Home → VMs**
- Click **New VM**
- Select your pool
- Set...
  - Template: (Type `TPL_U` to start auto-filling the options, and select your template)
  - Name: (Name of your choosing)
  - Description: (Something descriptive like: `Docker server on Ubuntu 24.04`)
  - VCPU: 16 (or whatever the number of cores is on the system)
  - RAM: 16 GiB (or whatever you need)
  - Topology: (Default behavior)
  - Click **Show advanced settings**
  - Uncheck **Boot VM after creation**
  - If you wish to have this system auto-power-on when the host system/hypervisor reboots...
    - Check **Auto power on**
- Click **Create**
- Click **Disks**
- Click the disk size in the **Size** column
- Increase the disk size to whatever your needs are
- You may optionally migrate the VDI to a faster SR or bulk data SR if desired by using the arrow button in the last column
- When you are finished, click the start button () at the top of the page
- Click the Console to watch the startup and proceed with initial configuration

Post-Cloning Configuration

- Log into the VM via the console
- Edit `/etc/netplan/50-cloud-init.yaml` and configure as appropriate to your use (reserve an IP from your firewall's DHCP pool).
  - Follow the instructions in the comments we made in the template
- Apply the new netplan config

```
sudo netplan apply
```

- SSH to the VM via the assigned IP using the account you created during initial installation and templating
- Configure the hostname

```
sudo hostnamectl set-hostname myhostname
```

- If you resized the disk, resize the filesystem

```
# Check the current partition configuration
lsblk /dev/xvda | grep -e ^ -e ^xvda.* -e ".*—xvda3.*"
```

```
# Resize xvda partition 3 to consume the available space
sudo growpart /dev/xvda 3

# Re-check the current partition configuration
lsblk /dev/xvda | grep -e ^ -e ^xvda.* -e ".*─xvda3.*"

# Check the current Free PE on the partition
sudo pvdisplay | grep -e ^ -e "Free PE.*"

# If you do not see any free PE's, run
sudo pvresize /dev/xvda3

# Then re-check the current Free PE on the partition
sudo pvdisplay | grep -e ^ -e "Free PE.*"

# Check the Free PE on the volume group
sudo vgdisplay | grep -e ^ -e "Free  PE.*"

# Check the LV Size on the logical volume
sudo lvdisplay | grep -e ^ -e "LV Size.*"

# Resize the LV, increasing by the number of PE's available above
sudo lvextend -l +7680 /dev/ubuntu-vg/ubuntu-lv

# Re-check the LV Size on the logical volume
sudo lvdisplay | grep -e ^ -e "LV Size.*"

# Check the Free PE on the volume group, confirming we used all available PE's
sudo vgdisplay | grep -e ^ -e "Free  PE.*"

# Check the filesystem Size
df -h /

# Resize the filesystem
sudo resize2fs /dev/ubuntu-vg/ubuntu-lv

# Re-check the filesystem to confirm larger Size
df -h /
```

- Optionally Install Docker

Created 2024-07-09 16:07:17 UTC by jholmstadt
Updated 2025-12-16 18:18:12 UTC by jholmstadt