Ubuntu Host Setup

Here are the minimal steps to setup a clean Ubuntu VM.

> Note: These instructions are tested on on Ubuntu v22 through v24.
> They may require updates for other versions.

Spawn VM Clone

The first step is to spawn a clone of the template VM.

Be sure to do the following:

1. Give it an inventory name that fits its hostname.
2. Update the CPU count.
3. Set the memory size.
4. Set its disk space for the intended service.
5. Assign the VM's NIC to the Provisioning portgroup (VLAN 170).
   This will ensure that we can access it via SSH, for faster setup.

Once the VM is started, log into its console session.

Update Packages

From the console, update packages of the VM.
Do this before anything else, to ensure the latest package versions are used.

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

Ubuntu: SSH Server

See this page for setting up the SSH Server: Ubuntu: SSH Server

Initial Remote Access (VLAN 170)

Once the VM is started up, and in the Provisioning VLAN (VLAN 170), it should have a DHCP address that we can reach.

From the VM's local console, run this to get its IP on the provisioning VLAN:

```
ifconfig
```

Open an SSH session to the VM, for remote setup.

Setting the Host Name

See this page for how to set the hostname: Ubuntu: Set Hostname

Other Packages

We will install net-tools on each host, for diagnostic purposes.

```
sudo apt install net-tools
```

Managed Host Setup

If the VM will be managed by Ansible and deployment tools, see this page for setup steps: Managed Host User Setup

User Setup

Configure any users and groups that the VM will need.

This may include a deployment user.

SSH Keys

We need to add SSH public keys for users added, above.

See this page for the various method of adding SSH keys to a remote linux host: Linux SSH Key Management

Switch SSH to Key Auth

Once you have installed SSH public keys in the VM, you need update the SSH config to SSH authentication.

To do so, open the sshd config file, with this:

```
sudo nano /etc/ssh/sshd_config
```

Look for the line with this directive, 'PasswordAuthentication'.

Uncomment the line, and set it to no, like this:

```
PasswordAuthentication no
```

In the config file, set Key Auth to yes:

```
PubkeyAuthentication yes
```

Save and close the SSH config file.

Restart SSH with this:

```
sudo systemctl restart ssh
```

Verify SSH Keys

Now that SSH key authentication is enabled, you need to verify that each added SSH public key works.

Attempt to connect with the VM, using each configured SSH key.
Verify each one works.

See this page for steps on how to connect to a Linux host from Windows: Connecting to SSH Server from Windows

Routing and Firewall

So far, we are accessing the VM on a temporary provisioning network.
We need to set things up for its final location.

Create a proper firewall rule for accessing the VM at its final IP address, in its target VLAN.

Static IP Address

With the above firewall rule in place, we will be able to access the VM, once it's moved to its target VLAN.

But first, we need to set its static IP address: Ubuntu: Setup Static IP Address

Change VLAN PortGroup

Setting the static IP address, above, means that we lost temporary SSH access.
We need to fix that.

In the hypervisor, change the VM's portgroup to the target VLAN.

Remote Access

With the VM at its assigned static IP, in the target VLAN, and with access firewall rule exists, we can remote SSH to it.

Attempt to open a remote SSH session to the VM.

Root CA Certs

If you are looking for the local intranet Root CA certificate, see this: Current Intranet Root CA Certificate

If your institution or company uses has its own certificate authority (CA), you should install its root CA into the following folder:

```
/usr/local/share/ca-certificates/
```

For example, to add the root CA for the local network, create the file with this:

```
sudo nano /usr/local/share/ca-certificates/ogsofttech.lan_ca.crt
```

Save and close the file.

Once all root CA certs are added, you need to update the runtime's CA list, with this:

```
sudo update-ca-certificates
```

See this page for how to add Root CA certificates: How to Add Root CAs to Ubuntu

NTP Client Needs

If the VM will be located in an isolated VLAN with restricted internet access, it may need access to the local NTP server.

See this page for how to setup each host/VM to use the local private NTP server: Ubuntu: Use Private NTP Server

Further Setup

With the above things done, we can continue on with other setup.

Revision #30
Created 21 May 2025 02:44:12 by glwhite
Updated 7 May 2026 05:27:01 by glwhite