# **Setting Up MAAS (Metal-as-a-Service) on Ubuntu 22.04 with UDM Pro**

## **Introduction**

MAAS (Metal-as-a-Service) allows you to manage bare-metal servers just like cloud instances. In this guide, we combine MAAS with a **UDM Pro** to handle DHCP, VLANs, and PXE booting, enabling automated Ubuntu deployments.

Prerequisites:

1. While using UDM pro as DHCP server you need to make sure on your UDM pro for option 66 & option 67 under network boot you need to give absolute path for bootfile like in my case it is “bootloader/uefi/amd64/grubx64.efi”
2. Also you need to configure firewall rule for port open from 69-65535

Rule details:

Rule Name: Maas server

Type: LAN IN

Action: Accept

Protocol:UDP

Source: Select Source as your maas network

Destination: Maas server ip and port as 69-65535

## **1. MAAS Installation**

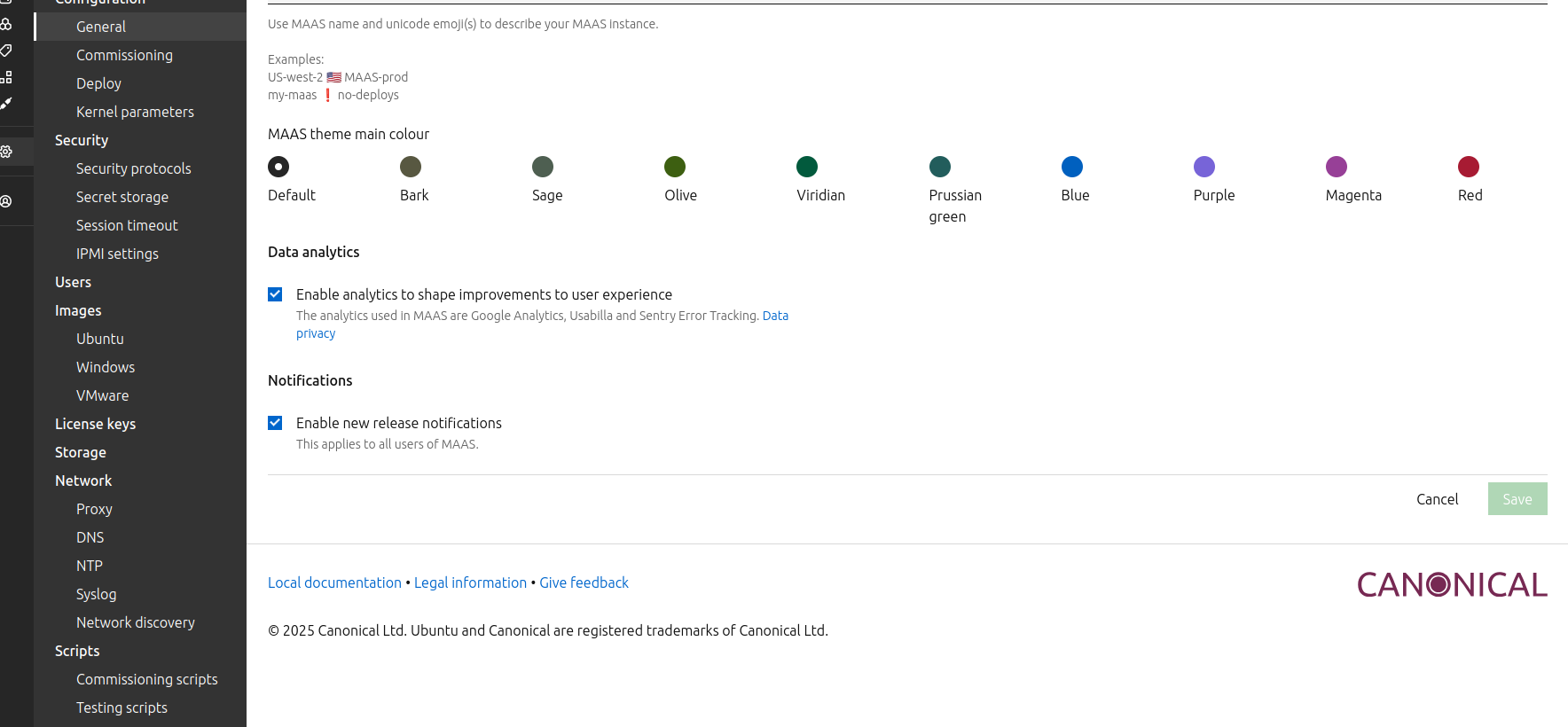
Install MAAS and its CLI tools:

sudo add-apt-repository ppa:maas/3.4 -y

sudo apt update

sudo apt install maas maas-cli maas-dhcp maas-dns -y

sudo maas init



## **2. Logging into MAAS CLI**

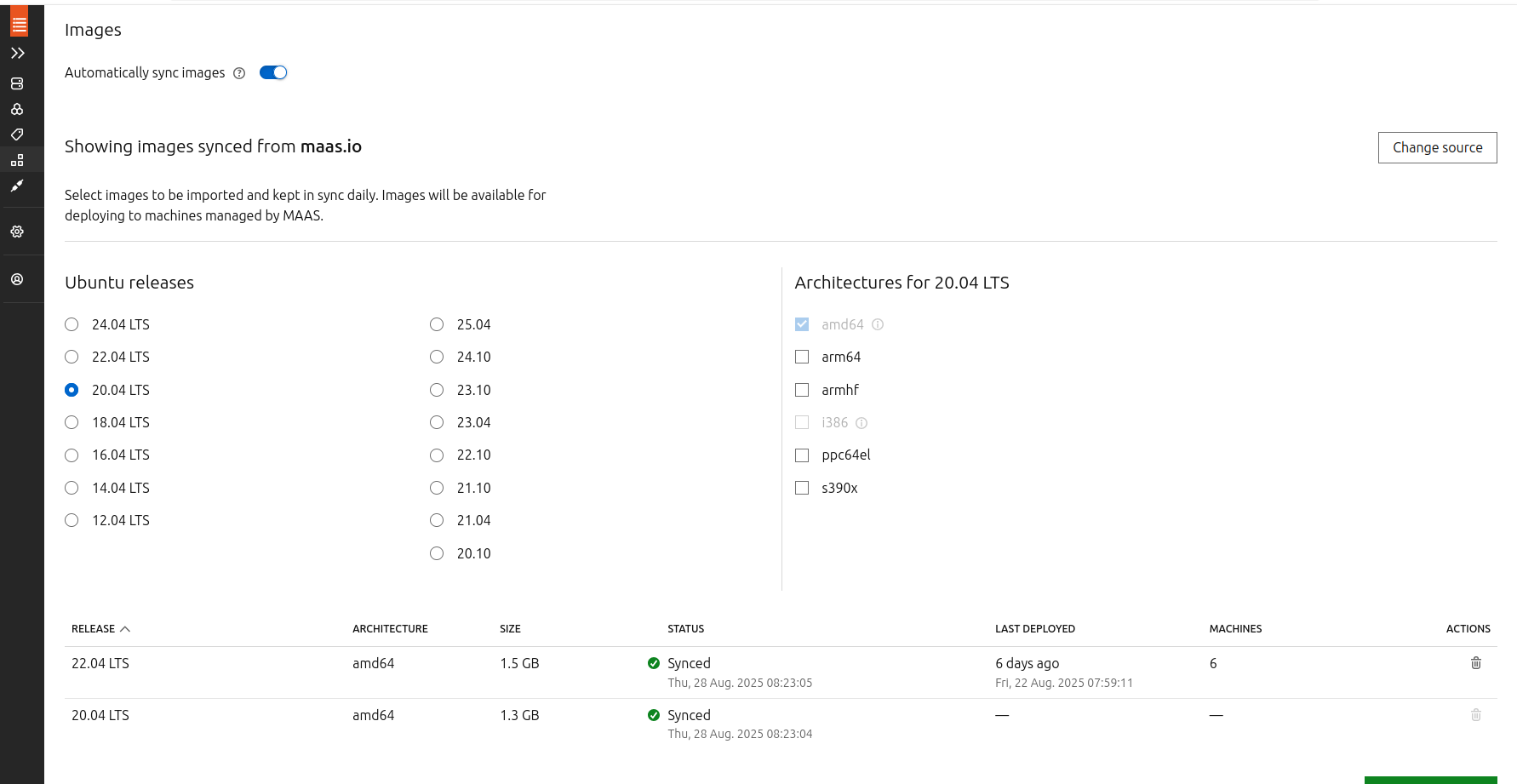
maas login admin http://yourdns:5240/MAAS <API\_KEY>

Replace <API\_KEY> with your MAAS admin API key.

## **3. Import Boot Resources**

sudo maas admin boot-resources import

maas admin boot-resources read | jq '.[] | {name, type, architecture, complete}'



* Ensures UEFI and legacy boot images are available.
* Verify files in /var/lib/maas/boot-resources/uefi/ for PXE boot.

## **4. Network Setup with UDM Pro**

* Configure a **VLAN for MAAS provisioning**.
* Assign **DHCP** to the VLAN or let MAAS manage it.
* Example MAAS CLI commands to set up VLANs and subnets:

maas admin fabrics read | jq '.[] | {id, name}'

maas admin vlans read 0 | jq '.[] | {id, name, vid, dhcp\_on, external\_dhcp, subnet: .space}'

maas admin vlan update 0 1 dhcp\_on=true primary\_rack=bareops

maas admin subnets create cidr=X.X.X.0/24 gateway\_ip=X.X.X.X fabric=0 vlan=1

## **5. Commissioning and Deploying Nodes**

* Commission a machine to verify hardware:

maas admin machine commission <SYSTEM\_ID>

* Accept the machine after commissioning:

maas admin machine accept <SYSTEM\_ID>

maas admin machine update <SYSTEM\_ID> status=Ready

* Deploy Ubuntu:

maas admin machine deploy <SYSTEM\_ID> distro\_series=jammy hwe\_kernel=hwe-22.04

* Monitor logs for TFTP and PXE boot issues:

sudo journalctl -u maas-rackd -f

sudo ss -ulnp | grep :69 # Check TFTP port

## **6. Verifying PXE Boot Resources**

sudo -u maas tftp localhost -c get bootloader/uefi/amd64/grubx64.efi

ls -lh /var/lib/maas/boot-resources/current/bootloader/uefi/amd64/

* Ensures all EFI boot files are present for UEFI-enabled servers.

# **2. Enabling Password Login on MAAS-Deployed Ubuntu 22.04**

## **Objective**

Allow SSH login via password while keeping SSH key authentication intact.

## **Steps**

### **2.1 Ensure SSH configuration allows password login**

sudo nano /etc/ssh/sshd\_config

Ensure the following lines are present and uncommented:

PasswordAuthentication yes  
PermitEmptyPasswords no  
UsePAM yes  
KbdInteractiveAuthentication yes

Save and exit (Ctrl+O, Enter, Ctrl+X).

### **2.2 Create a drop-in override file**

sudo mkdir -p /etc/ssh/sshd\_config.d  
echo -e "PasswordAuthentication yes\nKbdInteractiveAuthentication yes" **|** sudo tee /etc/ssh/sshd\_config.d/01-password.conf

### **2.3 Restart SSH service**

sudo systemctl restart sshd  
sudo systemctl status sshd

### **2.4 Verify SSH settings**

sudo sshd -T **|** grep -E "passwordauthentication|kbdinteractiveauthentication"

Expected output:

passwordauthentication yes  
kbdinteractiveauthentication yes

### **2.5 Set a password for the ubuntu user**

sudo passwd ubuntu

Enter and confirm your desired password.

### **2.6 Test SSH password login locally**

ssh ubuntu@localhost

### **2.7 Connect using Remmina**

* Protocol: SSH
* Username: ubuntu (default username is ubuntu you can change it later)
* Password: the one set above
* Port: 22

Now Remmina should connect successfully using password authentication.

**Note:** The drop-in file in /etc/ssh/sshd\_config.d/ is necessary because MAAS/cloud-init may override the main config. Always verify with sudo sshd

## **7. Tips & Troubleshooting**

* If DHCP or PXE isn’t working, verify VLAN and DHCP on UDM Pro.
* Clear MAAS caches if boot resources fail to load:

sudo systemctl stop maas-regiond maas-rackd

sudo rm -rf /var/lib/maas/regiond/cache/\*

sudo rm -rf /var/lib/maas/boot-resources/\*

sudo systemctl start maas-regiond maas-rackd

sudo maas admin boot-resources import

* Always check TFTP and rackd logs for errors:

sudo tail -f /var/log/maas/rackd.log

## **Conclusion**

Using MAAS with a UDM Pro allows **full automation of bare-metal server provisioning**. You can now commission, deploy, and manage Ubuntu servers via PXE boot in a controlled VLAN environment, all accessible remotely through SSH.