Installing Medulla

Installation documentation for a single-site Medulla server using Ansible.

Debian OS installation for Medulla server

Technical Specifications

Prerequisites – Server Sizing		
Main Server	OS	Debian 12.x
	Architecture	x86-64
	CPU	8 cores
	RAM	8 GB
	Partition /	20 GB in EXT4
	/var partition	400 GB minimum in XFS or mount point on a storage array
Multi-site relay servers (if applicable)	OS	Debian 12.x
	Architecture	x86-64
	CPU	4 cores
	RAM	8 GB
	Partition /	20 GB in EXT4
	/var partition	At least 400 GB in XFS or mount point on a storage array

Debian Server Installation

Summary:

parate only /var from / and place them in LVM

stall the SSH server and standard system utilities

not install antivirus software or a firewall

t up an account that

- o can switch to sudo without a password
- o can log in from the IP address **94.130.207.190**

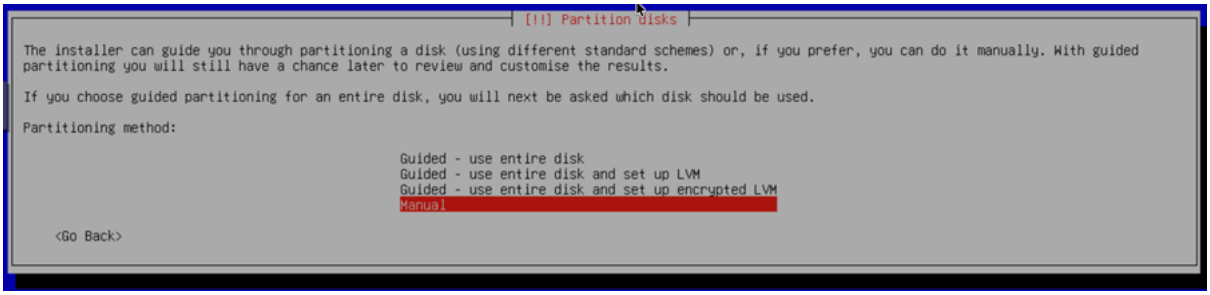
- o can log in with the following key:

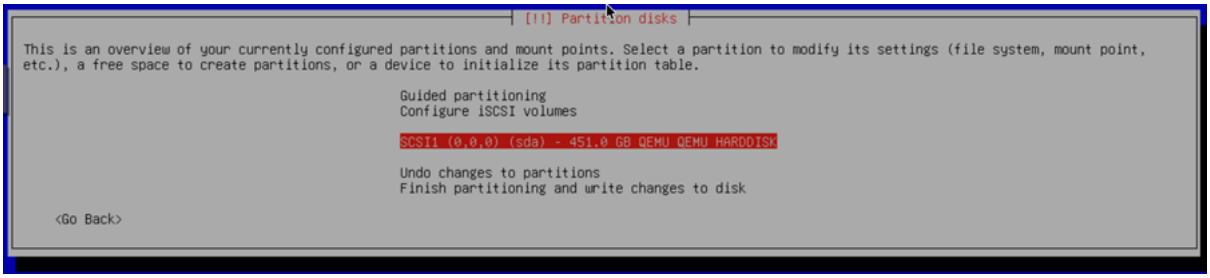
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCScgwfwJKM5BtgzAYu6FEeJ5jW3onkzFp8D8piLR22kWbRcT/AJ1z0jhS5ZDtn6mumfidVPFbLkDf382u54pOU6JGwy9GhvEIXOSlzgxZMH5kcfeBE/8Ovr9zLtbRKsWQN9YUSt5y6lmcSxuQNVhkRy49/593oamVJACSitSVJ68716hj0gp4N8gUMVkvNgEBDZVSPe0DXz2h7JEzOKx2ejjRaw22ve+qARTw+60gMP0aCLGt/m0cyv+90AZigQwWIPcUk+bBRJn3Ku+Bkw+JuLYURlVc4xoTvT1JTWKXAzMln4nrlisIc9Ex5eEHSkvs/fgJCgU28Fza5n5mBj/pbQRY+/AWLjvBVuLiVReO7hq60fhrX9+j7MWMCYCZQiHbk/r7OprLyl2yGFX1DbgRGF1Sk2R9DtqRhwPzPxtQ7ZtKSjIhLjrZxJ/YJLHSoUsw+4CHprjzU0gXBt1RCQoyhYqEGcnuFyfd9dIBXCINkmp4jfz7CQjrC8uPqAtS1zQU= support@support

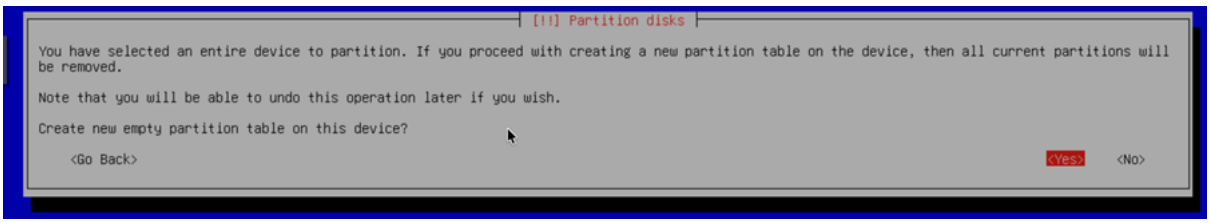
Disk Partitioning Configuration

Partition the disks according to the instructions below:

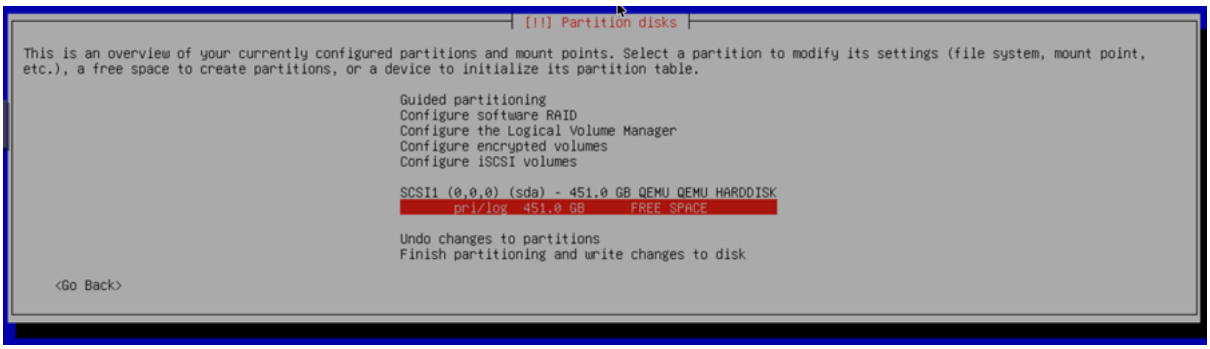
Perform manual partitioning

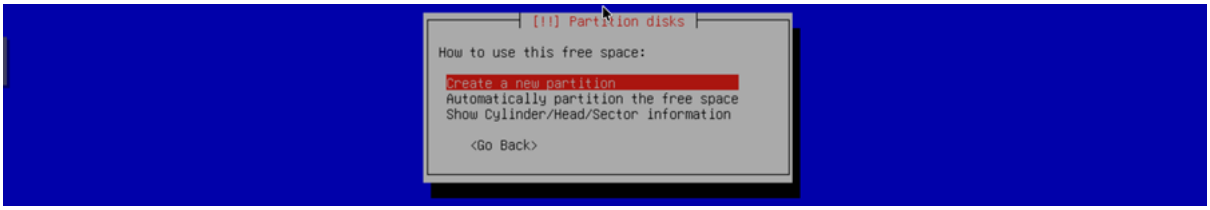


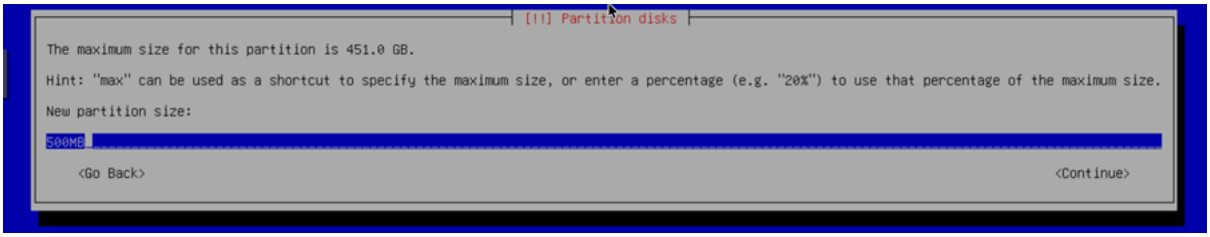


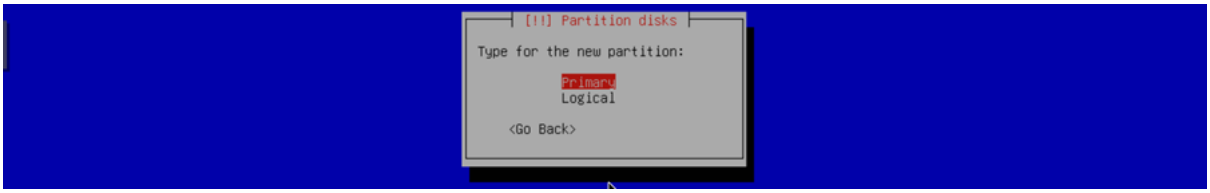


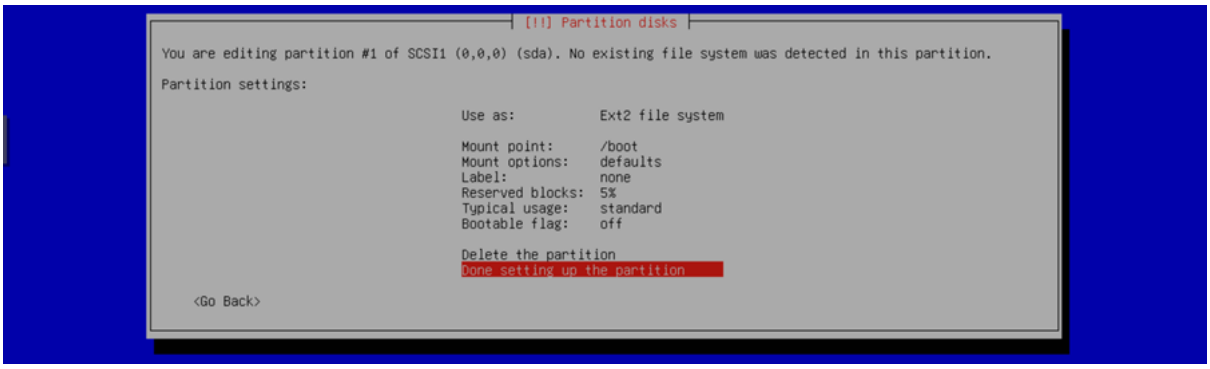
Creating the /boot partition



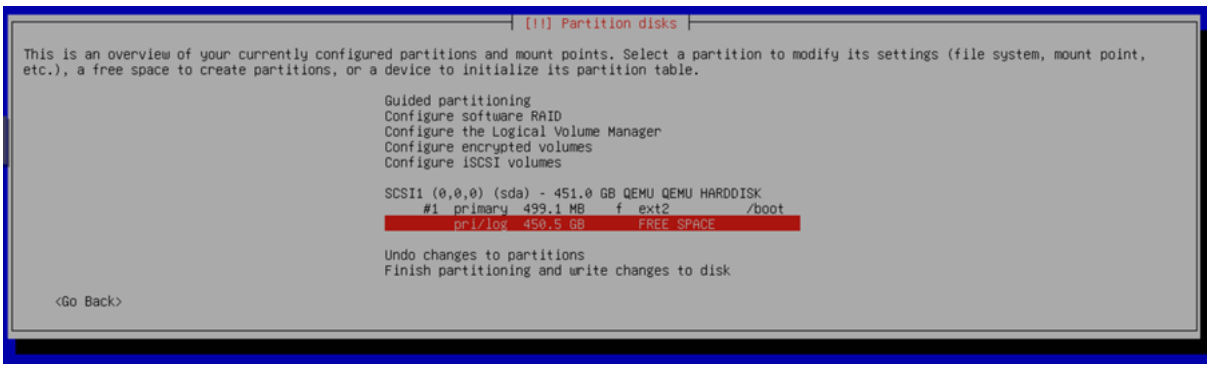


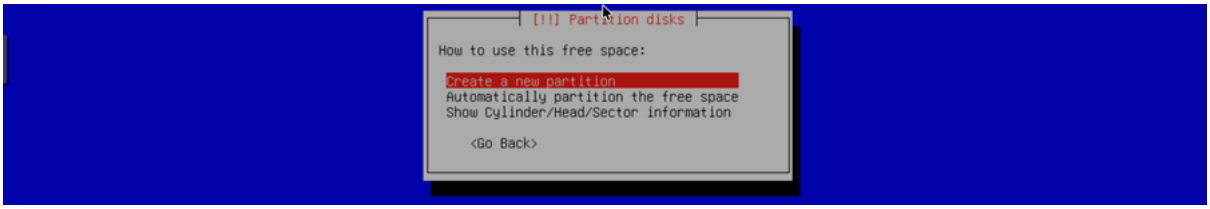


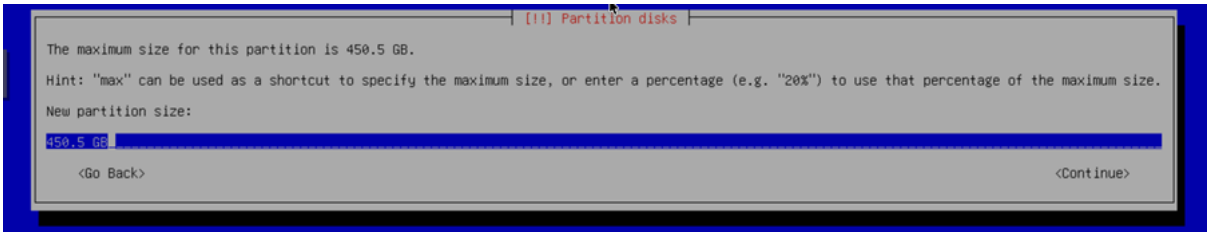


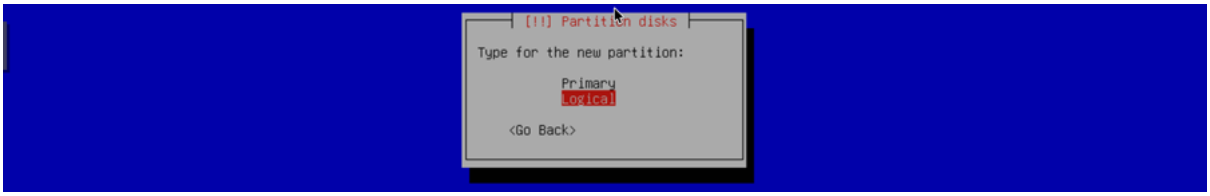


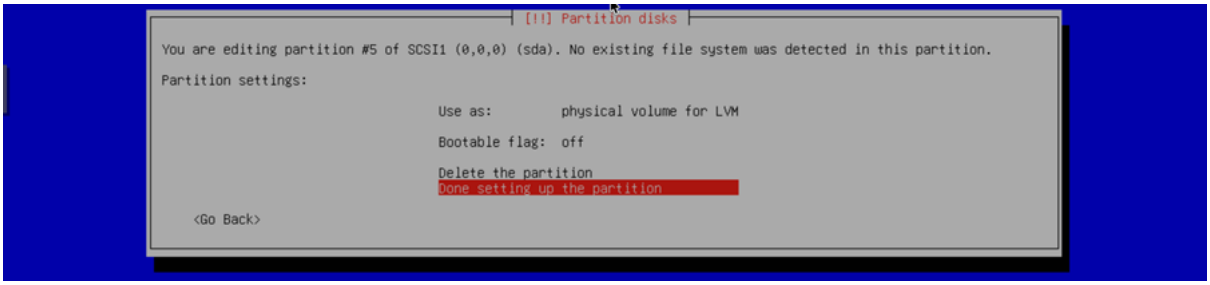
Creating the LVM

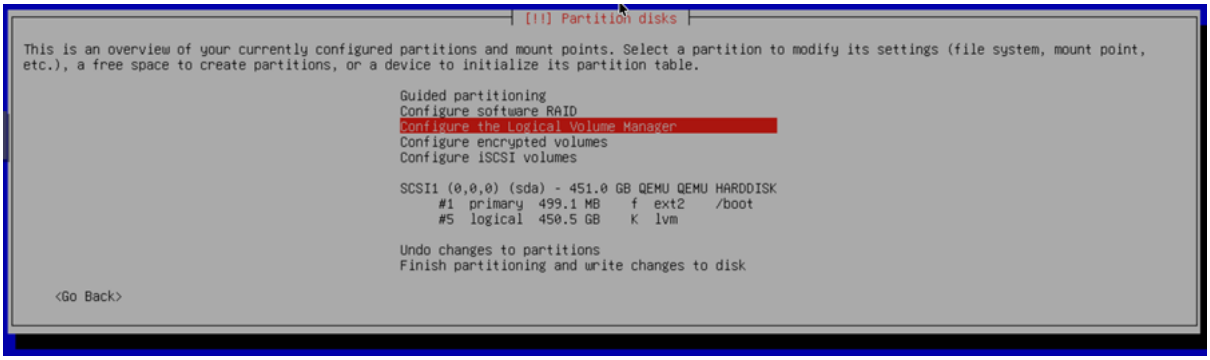


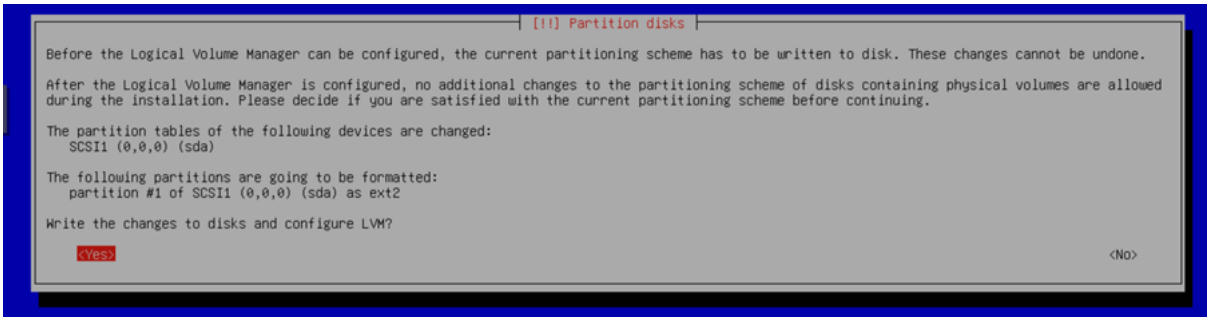




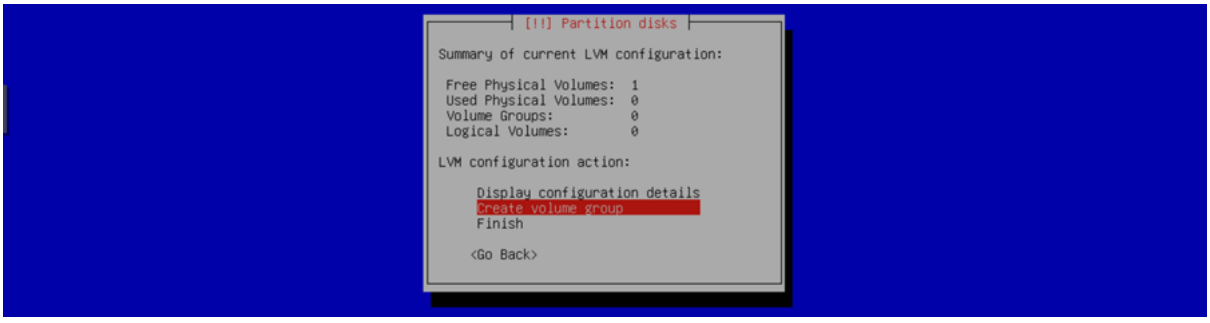


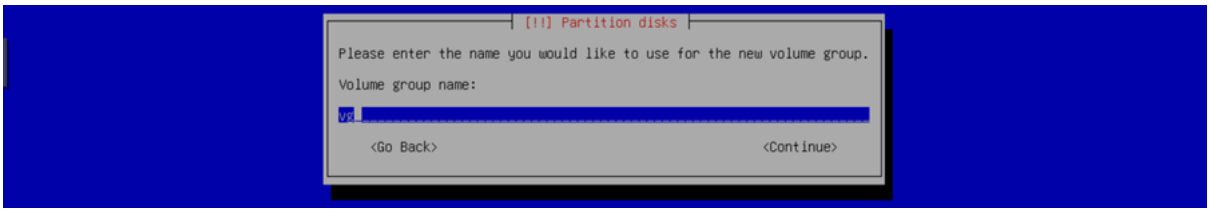


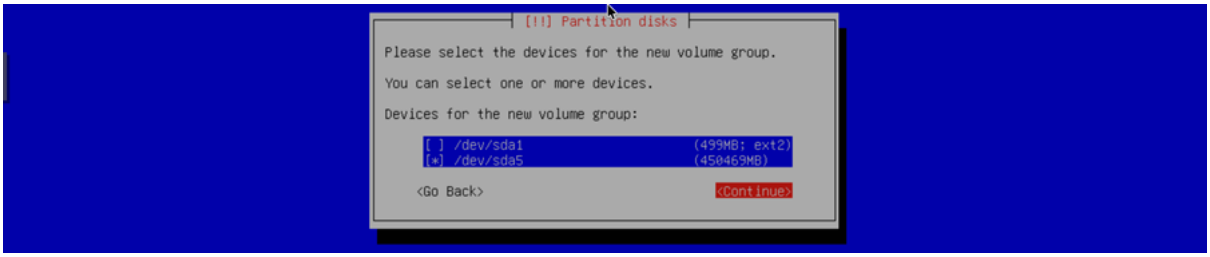




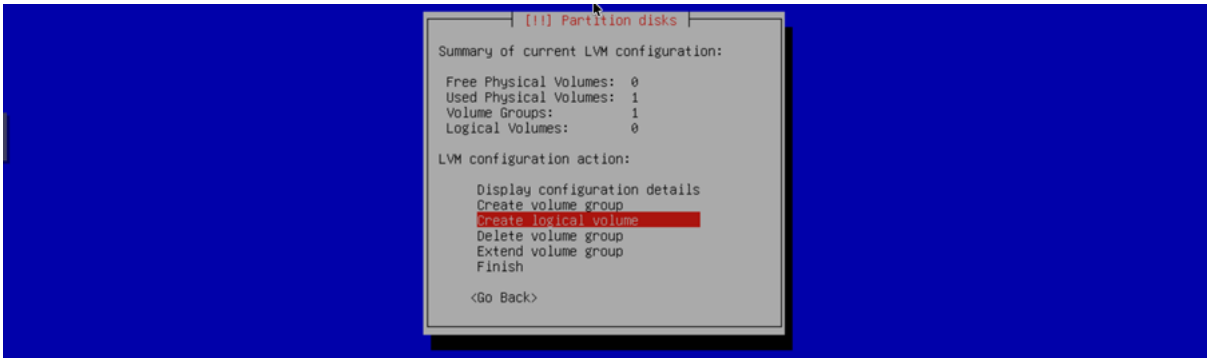
Creating the vg volume group

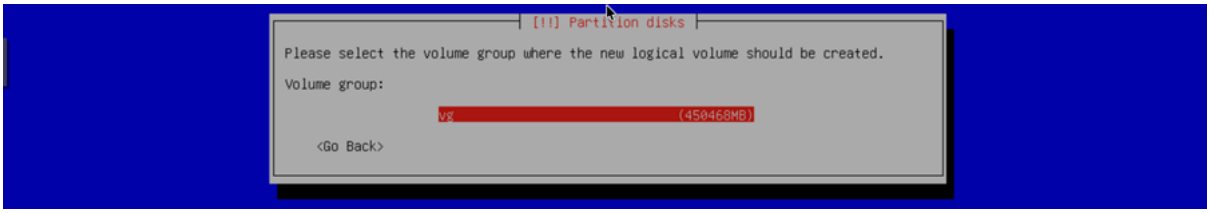


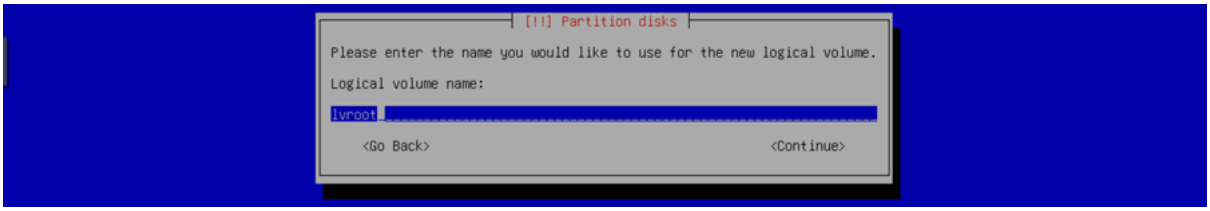


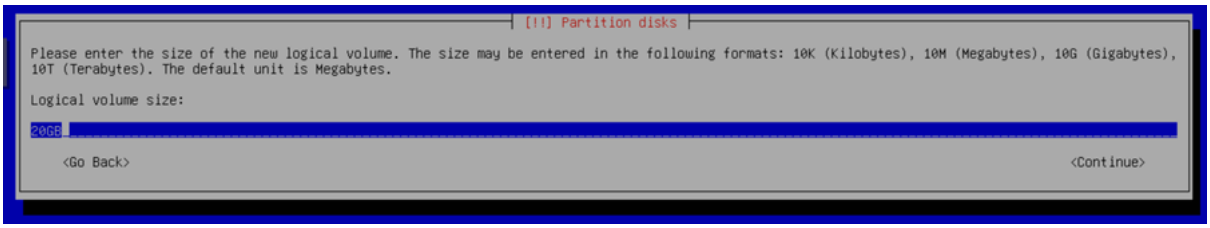


Create the logical volume lvroot

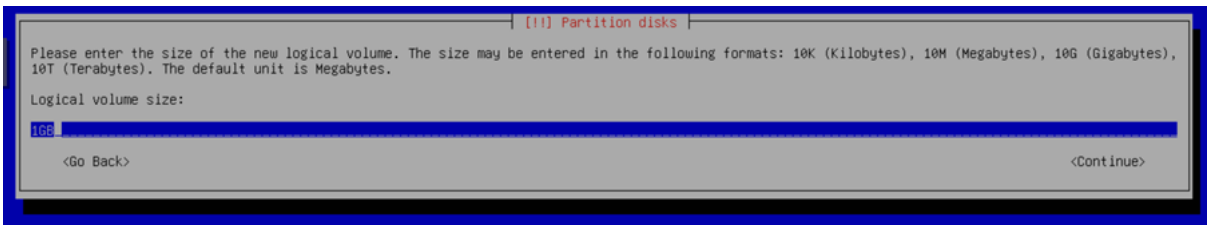


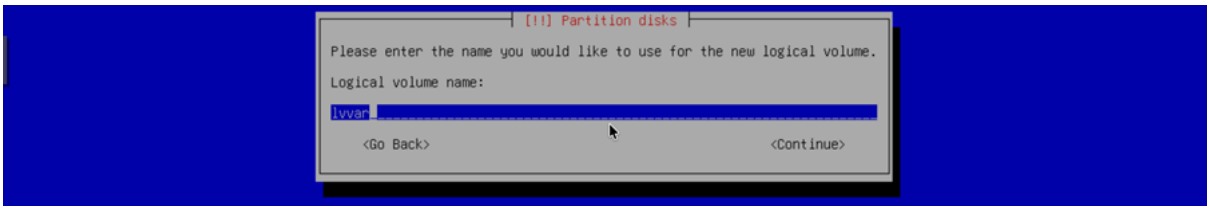




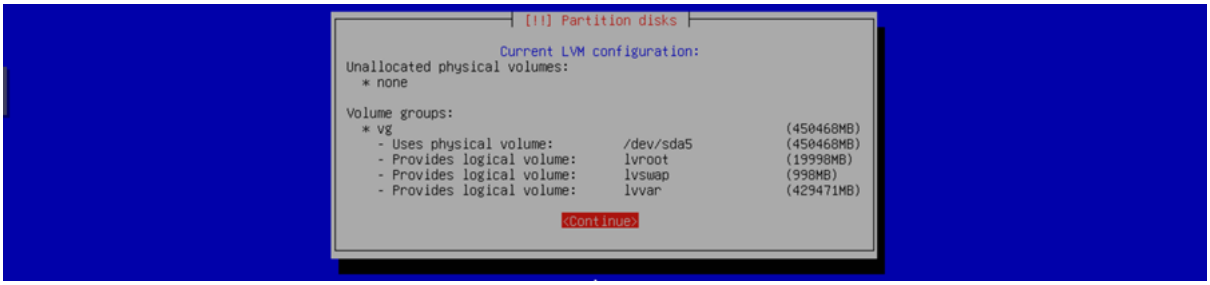


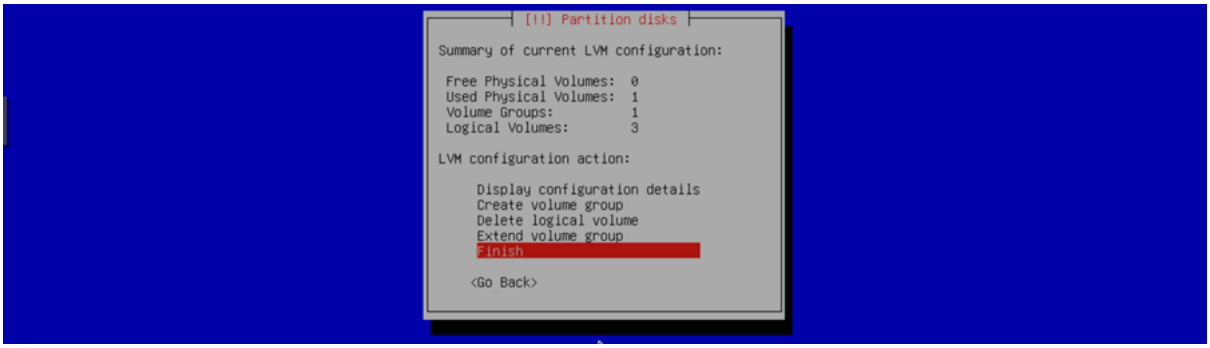
Repeat for the lvswap and lvvar volumes



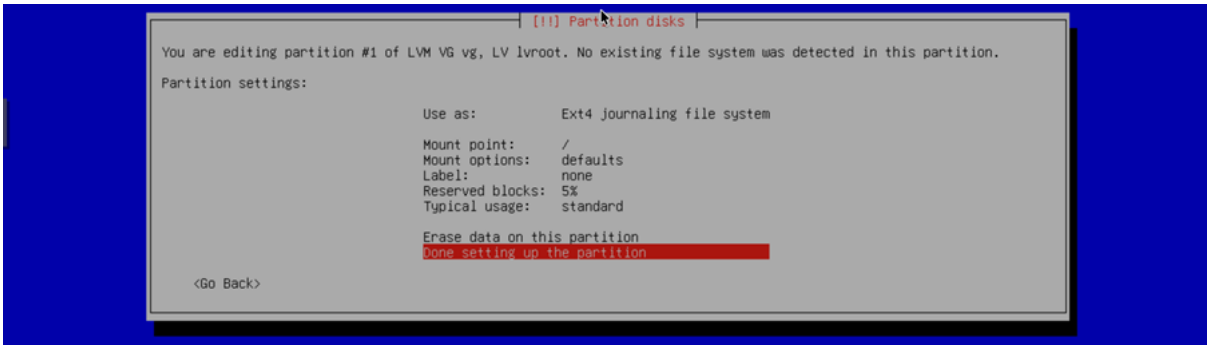


To obtain the following LVM layout:

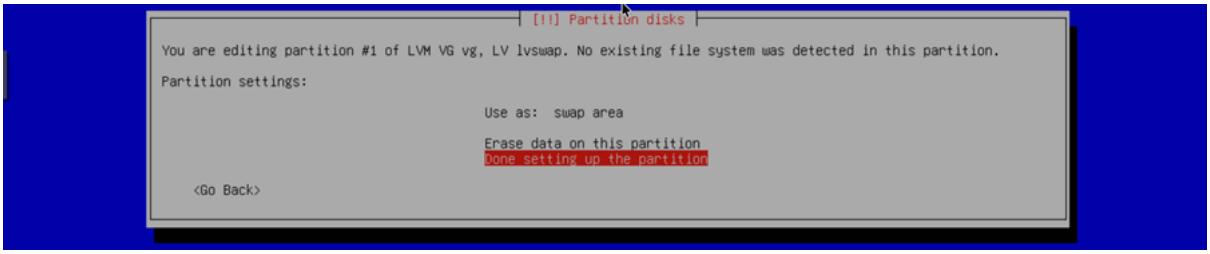


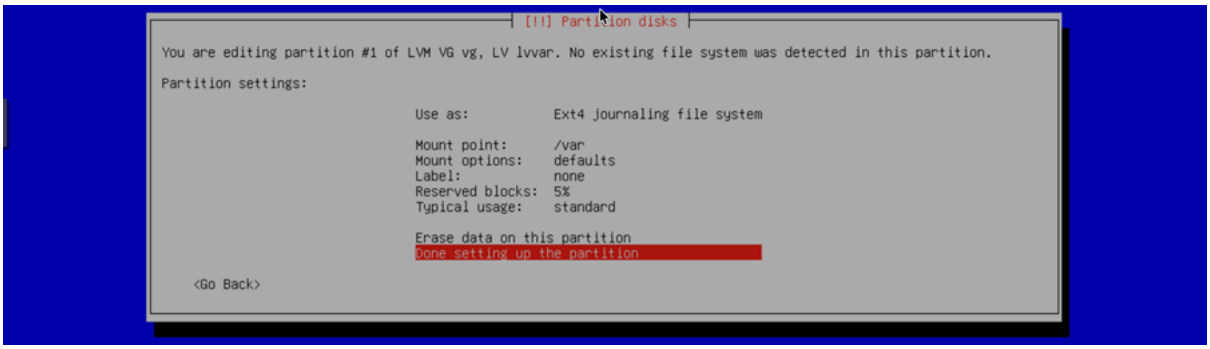


Configure the / partition

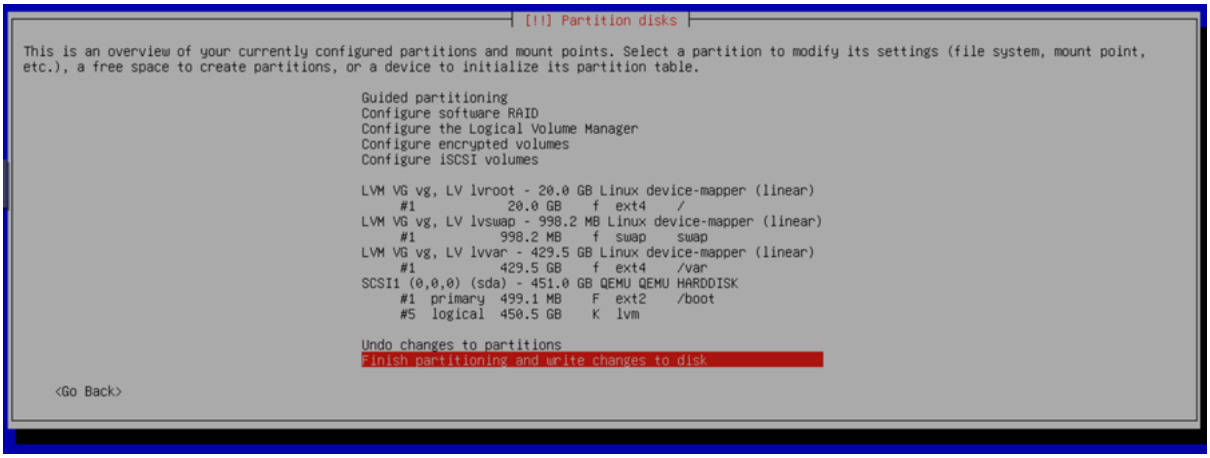


Repeat for each of the swap and /var partitions:

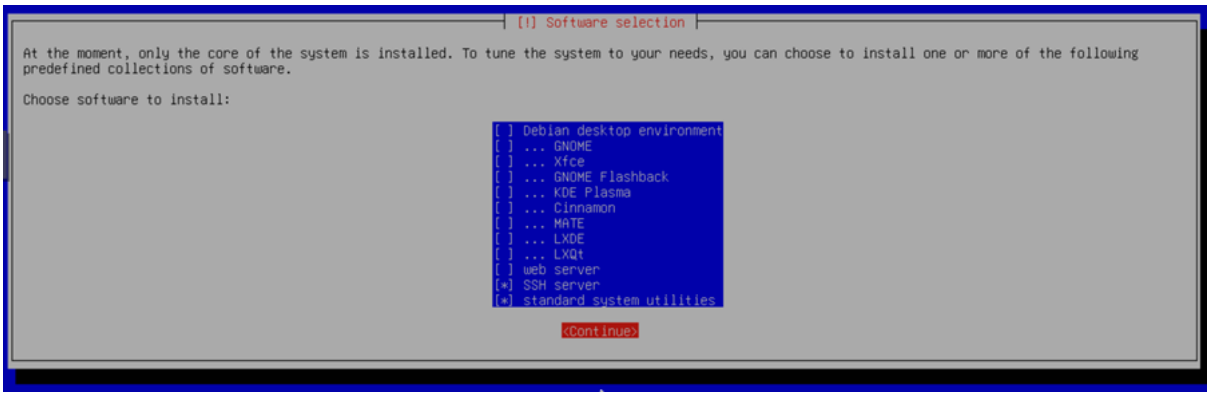




To obtain the following partitioning scheme:



Installing packages



Server verification

Download the verification script from https://dl.medulla-tech.io/nc/check_server_before_install.sh

```
wget https://dl.medulla-tech.io/nc/check_server_before_install.sh
```

Run the following commands:

```
chmod +x check_server_before_install.sh

./check_server_before_install.sh
```

All script tests must pass. Once completed, you can request the installation script via the contact form:

https://github.com/medulla-tech/medulla/blob/master/README.fr.md

If you have a support contract, please send the results to delivery@medulla-tech.io. If not, please contact the "Sales" department via our websitemedulla-tech.io.

e are the most common errors:

1. Core Dump Limits

Context: The file `/etc/security/limits.d/10-coredump-debian.conf` defines the maximum size of "core dump" files. Our script expects specific values that do not match the current configuration.

How to fix: Edit the file mentioned to match the requirements.

1. Open the file: `sudo nano /etc/security/limits.d/10-coredump-debian.conf`
2. Adjust the lines so they look like this:
   - `* hard core infinity`
   - `root hard core infinity`
   - `* soft core 0`
   - `root soft core 0`

Suggested sources: Debian documentation on `limits.conf` and `core dump`.

---

2. Number of open files (lsof)

Context: The `lsof` lines for users xxx and `messagebus` indicate that the number of currently open files deviates from the value expected by the verification script (often because services are already running or are misconfigured).

How to fix: This is often informative, but if you need to reduce these numbers:

- Identify what these users are doing: `ps -u user_id_in_error`.
- Restart the associated services (e.g., `sudo systemctl restart dbus` for messagebus).
- If the expected values are too strict for your use case, you may need to adjust the validation script itself or the global `ulimit` settings.

Suggested sources: `lsof` manual and file descriptor management in Linux.

---

3. systemd settings (NPROC and SIGPENDING)

Context: The `DefaultLimitNPROC` (maximum number of processes) and `DefaultLimitSIGPENDING` (pending signals) values must be**31541**.

How to fix: You must force these values in the global systemd configuration so that they match the expected values exactly.

1. Edit the configuration file: `sudo nano /etc/systemd/system.conf`
2. Uncomment or add the following lines:
   - `DefaultLimitNPROC=31541`
   - `DefaultLimitSIGPENDING=31541`
3. Reload the configuration and reboot (or use `systemctl daemon-reexec`).

Suggested sources: `systemd-system.conf` documentation on freedesktop.org.

Installing Medulla on Linux

Prerequisites:

The Medulla installation script supports the Debian (Bookworm) distribution.

You must have a machine with:

- **at least 8 GB of available RAM**,
- **at least 40 GB of free disk space for / and 200 GB for /var**.

Verify that your **/etc/hosts** file is properly configured with the full FQDN.

For example, if the machine is named medulla and the domain is named domain.lan, the /etc/hosts file must contain a line like this:

```
127.0.1.1    medulla.domain.lan    medulla
```

Verify that the **/etc/resolv.conf** file **is** properly configured and that the machine is connected to the internet.

Installing the Medulla server:

To install the Medulla server: Use a user with sudo privileges to run the installation script and playbook.

1- Transfer the script to the server.

2- Set execute permissions on the file:

```
type: chmod +x install_from_ansible.py
```

3- Start the server installation:

```
./install_from_ansible.py
```

Wait for the installation process to finish; a summary will display all the necessary passwords (copy them to a safe place).

Installing the Medulla Agent

The Medulla agent can be downloaded from

http://medulla.domaine.lan/downloads/win/Medulla-Agent-windows-FULL-latest.exe

The Medulla agent can be installed manually or in silent mode:

> "Medulla-Agent-windows-FULL-latest.exe /S"

The installation process will continue after the installation is complete; it will install all dependencies.

It is complete when the computer appears in Medulla in blue (online).

Nom de la Machine	Description	Système d'exploitation	Type
demo-win-2		Microsoft Version d'évaluation de Windows 11 Entreprise	Standard PC (Q35 + ICH9, 2009)

Medulla Agent

By default, we provide a global agent for Medulla (as explained in this post). **Entity management per workstation is handled via GLPI.**

However, if you wish **to benefit from automatic assignment of machines to entities directly via the agent**, it is also possible to **generate entity-specific agents**. This option is not enabled by default, but we can assist you in setting it up if it meets your needs. To view agents by Entity:

Entity Management

Agent Deployment via WinRM SSH

Preparing the Machines

Debian Machines (Medulla Main and Medulla Relay)

- Ensure that netcat is installed on the Debian machines:
- `sudo apt update && sudo apt install netcat-openbsd`

Windows Machine (Client Workstation)

- Make sure PowerShell is configured to run scripts:
- `Set-ExecutionPolicy RemoteSigned -Scope CurrentUser`
- Answer [A] "Yes to all"

Listening Scripts

On Debian Machines (Medulla Main and Medulla Relay)
Use the listen_ports_debian.sh script to listen on the necessary ports.

Steps:

1. **Download the**listen_ports_debian.sh**script**to the machine.
2. **Make it executable (**and convert to Unix if necessary):

```
chmod +x listen_ports_debian.sh
dos2unix listen_ports_debian.sh
```

1. **Run the script**:
   - For**Medulla Main**:
   - `./listen_ports_debian.sh --medulla`
   - For**Medulla Relay**:
   - `./listen_ports_debian.sh --relay`

On the Windows Machine (Client)

Use the listen_ports_windows.ps1 script to listen on the necessary ports.

Steps:

1. **Download the**listen_ports_windows.ps1**script**to the machine.
2. **Run the script**:

```
.\listen_ports_windows.ps1
```

Connection Test Scripts

On Debian Machines (Medulla Main, Medulla Relay, and Client Station)

Use the medulla_connection_check.sh, medulla_relay_connection_check.sh, and windows_connection_check.ps1 scripts to test connections to the machines.

Steps:

1. **Download the scripts**to the three machines (Medulla Main, Relay, and Client Workstation).
2. **Make them executable**:
3. ```
   chmod +x medulla_connection_check.sh
   chmod +x medulla_relay_connection_check.sh
   chmod +x listen_ports_debian.sh
   ```
4. If necessary (error when running the Debian scripts), convert them to Unix:
5. ```
   dos2unix medulla_connection_check.sh
   dos2unix medulla_relay_connection_check.sh
   dos2unix listen_ports_debian.sh
   ```
6. **Run the script**:
   - To test connections from**Medulla Main**to the Medulla Relay:
   - `./medulla_connection_check.sh --relay <Medulla_Relay_IP>`
   - To test connections from**Medulla Main**to a Client Workstation:
   - `./medulla_connection_check.sh --client <Client_Workstation_IP>`
   - To test connections from**Medulla Relay**to Medulla Main:
   - `./medulla_relay_connection_check.sh --medulla <Medulla_Main_IP>`
   - To test connections from**Medulla Relay**to a Client Workstation:
   - `./medulla_relay_connection_check.sh --client <Client_Workstation_IP>`
   - To test connections from**a Client Station to**Medulla Main and Medulla Relay:

- ```
  .\windows_connection_check.ps1 -Target <Medulla_Main_IP> -Mode pulse
  .\windows_connection_check.ps1 -Target <Medulla_Relay_IP> -Mode relay
  ```

Complete Procedure

Port Listening

On Medulla Main (Debian):

```
./listen_port_debian.sh --medulla
```

On Medulla Relay (Debian):

```
./listen_port_debian.sh --relay
```

On Client Workstation (Windows):

```
.\listen_ports_windows.ps1
```

Connection Test From Medulla Main (Debian):

```
./medulla_connection_check.sh --relay <Medulla_Relay_IP>
./medulla_connection_check.sh --client <Client_Workstation_IP>
```

From Medulla Relay (Debian):

```
./medulla_relay_connection_check.sh --medulla <Medulla_Main_IP>
./medulla_relay_connection_check.sh --client <Client_Workstation_IP>
```

From the Client Workstation (Windows):

```
.\windows_connection_check.ps1 -Target <Medulla_Main_IP> -Mode pulse
.\windows_connection_check.ps1 -Target <Medulla_Relay_IP> -Mode relay
```

Troubleshooting

Connection Issues

- **Check IP addresses**: Make sure the IP addresses used are correct.
- **Check firewalls**: Make sure that the firewalls on the machines allow connections on the necessary ports.
- **Check services**: Make sure the necessary services are running.
```

Listening Issues

- **Check listening ports**: Use `netstat -ano` on Windows or `ss -tulnp` on Debian to verify that the ports are listening.
- **Check for errors**: Review the error messages in the scripts to identify any issues.

Medulla interface

To access the Medulla console, open a web browser and enter the following address:

http://medulla.domaine.lan/mmc

or

http://ip-serveur/mmc/

To learn about Medulla and how to get started with the solution:

Medulla Documentation

DHCP / PXE Configuration

Enable UEFI boot with PXE

DHCP server:

The DHCP server requires a special option to enable PXE boot,

The DHCP options are:

- Option 66
- Option 67

Windows DHCP server configuration

First of all, the DHCP server needs to determine which type of computer is requesting the PXE server and the correct boot file.

To sort computers, vendor classes must be defined.

Vendor Classes

Vendor Classes as a Detection Method are used to determine how devices request a boot image from the DHCP server.

- Open the DHCP Console and expand the IPv4 Node
- Right-click on ‘IPv4 Node’ and select ‘Define Vendor Classes’
- Click ‘Add’
- First, create the UEFI 64-Bit Vendor class by entering the following information
- Enter the following information in the respective fields:
  - **DisplayName:**PXEClient (UEFI x64)
  - **Description:**PXEClient:Arch:00007
  - **ASCII:**PXEClient:Arch:00007
- Click ‘OK’
- Click ‘Add’
  - **DisplayName:**PXEClient (BIOS x86 & x64)
  - **Description:**PXEClient:Arch:00000
  - **ASCII:**PXEClient:Arch:00000
- Click ‘OK’

Creating Custom DHCP Policies

32-Bit & 64-Bit BIOS DHCP Policy

- Right-click ‘Policies’ and click ‘New Policy’
- Give the policy a descriptive name that matches your vendor’s class naming scheme:
  - **PolicyName**: PXEClient (BIOS x86 & x64)
  - **Description**: Delivers the correct boot file for BIOS machines
- Click ‘Next’
- On the ‘Configure Conditions for the policy’ page, click ‘Add’
- Select the ‘Value’ drop-down box and select the**PXEClient (BIOS x86 & x64)**vendor class that you created in the previous steps
- Make sure to check the**‘Append wildcard(*)’** box
- Select ‘Add’
- Select ‘OK’
- Click “Next”
- If you want the policy to apply only to a specific range within your scope, configure it; otherwise, select “No” and click “Next”
- On the Configure settings for the policy page, ensure that ‘DHCP Standard Options’ is selected from the drop-down box
- Configure the following scope options:
  - **066**:*IP Address of Medulla*
  - ***067****: bootloader/undionly*.kpxe
- Click ‘Next’
- **On the Summary page, click “Finish”**

UEFI DHCP Policy

- Right-click ‘Policies’ and click ‘New Policy’
- Give the policy a descriptive name that matches your vendor’s class naming scheme:
  - **PolicyName**: PXEClient (UEFI)
  - **Description**: Delivers the correct boot file for (UEFI)
- Click ‘Next’
- On the ‘Configure Conditions for the policy’ page, click ‘Add’
- Select the ‘Value’ drop-down box and select the**PXEClient (UEFI)**vendor class that you created in the previous steps
- Make sure to check the**‘Append wildcard(*)’** box
- Select ‘Add’
- Select ‘OK’
- Click “Next”
- If you want the policy to apply only to a specific range within your scope, configure it; otherwise, select “No” and click “Next”
- On the Configure settings for the policy page, ensure that ‘DHCP Standard Options’ is selected from the drop-down box
- Configure the following scope options:
  - **066**:*IP Address of Medulla*
  - ***067****: bootloader-uefi64/ipxe*.efi
  - Click ‘Next’

- On the **Summary** page, click “Finish”

Remove Default PXE Options

Ensure that you have removed the 067, 066, and 060 options from the default scope options to ensure that the policies take precedence; otherwise, you will end up with a conflict. As long as you have configured everything correctly, you should now be able to boot machines from BIOS or UEFI.

Linux DHCP Server

```
# PXE definitions
option space PXE;
option PXE.mtftp-ip code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
option PXE.discovery-control code 6 = unsigned integer 8;
option PXE.discovery-mcast-addr code 7 = ip-address;
option arch code 93 = unsigned integer 16;
# In the initial DHCP DISCOVER packet, the PXE client sets option 93 to its architecture.
# 0000 == IA x86 PC (BIOS boot)
# 0006 == x86 EFI boot
# 0007 == x64 EFI boot

# PXE boot following the PXE specifications
class "PXE" {
match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;
if option arch = 00:07 {
filename "bootloader-uefi64/ipxe.efi";
} else {
filename "/bootloader/undionly.kpxe";
}
}

# Etherboot boot
class "Etherboot" {
match if substring (option vendor-class-identifier, 0, 11) = "Etherboot-5";
option vendor-encapsulated-options 3c:09:45:74:68:65:72:62:6f:6f:74:ff;
option vendor-class-identifier "Etherboot-5.0";
```

```
vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;
}

subnet ##MEDULLA_NETWORK## netmask ##MEDULLA_NETMASK## {
option broadcast-address ##MEDULLA_BCAST##; # broadcast address
option domain-name ##MEDULLA_DOMAIN##; # domain name
option domain-name-servers ##MEDULLA_DNS##; # DNS servers
option routers ##MEDULLA_GW##; # default gateway

pool { # Only defined pool

# uncomment the following two lines for PXE-only boot
#allow members of "PXE"; # PXE-only
#allow members of "Etherboot"; # PXE-only
range ##MEDULLA_START## ##MEDULLA_END##;
next-server ##MEDULLA_IP##;
}
}
```