

6. Benchmark procedure

Now that the benchmarking suite, systems, and targets are ready, the benchmarks can be run. It is very important to have a fixed procedure for applying the benchmark on the different targets to ensure that the results are in fact comparable. Every chapter of the procedure is applicable in parallel on multiple identic hosts to speedup the entire benchmarking of a type of system.

For the three targets, the following chapters were run in this exact order.

6.1. Bare metal

For this, one need to use the iso [ubuntu-22.04.1-live-server-amd64.iso](#). This installation image has all the necessary packages self-contained, so it sure to have exactly the same versions each time. One can then start the server on the iso and apply the following configuration:

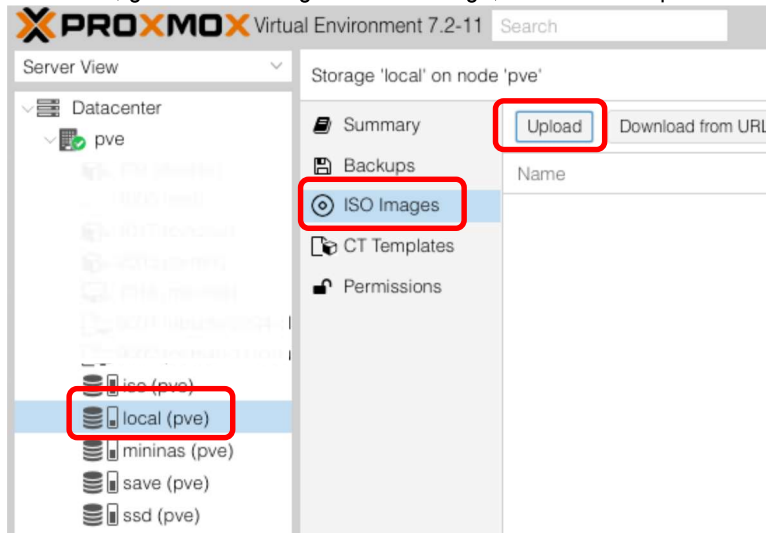
1. Language: English
2. Don't update the installer
3. Choose Switzerland keyboard
4. Select Ubuntu Server as a base installation
5. Configure the network with the IP specified in chapter 5.2
6. Don't specify proxy
7. Choose default mirror address
8. Use default storage configuration
9. Enter `user` as username and `TMhes2022` as password
10. Import SSH public key from GitHub
11. Don't add additional features
12. Reboot when installation is ready
13. Connect with the user
14. Copy the SSH key with `sudo cp .ssh/authorized_keys /root/.ssh/authorized_keys`

Once this is done, one can use the benchmark scheduling tool to schedule the normal, Docker and k8s benchmarks.

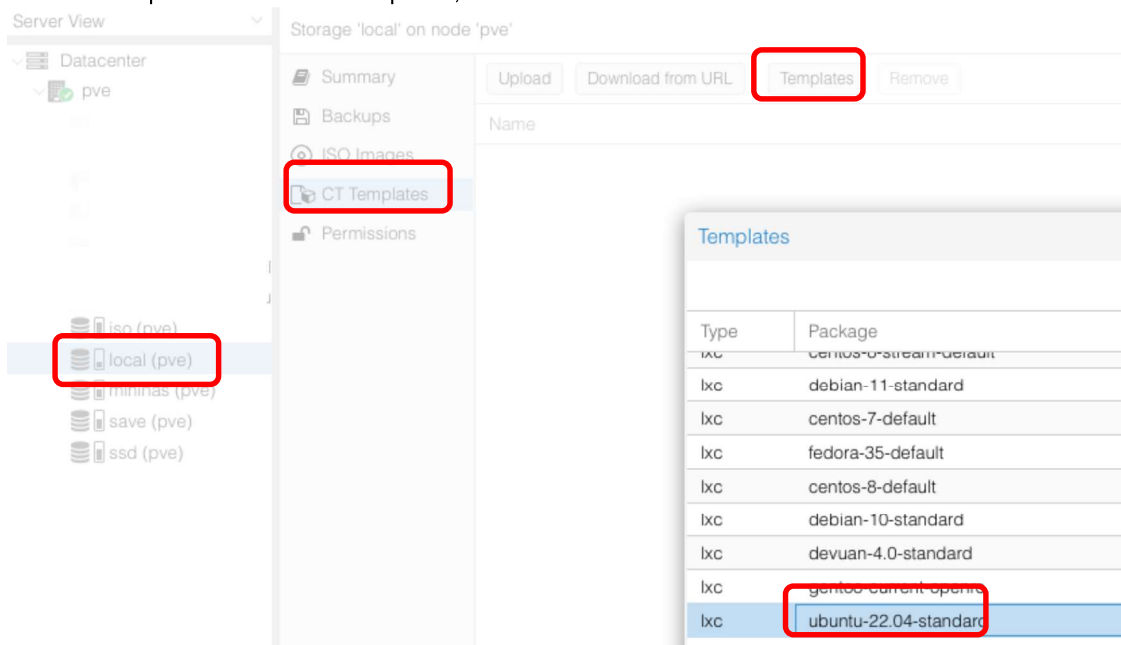
6.2. Proxmox VE

Boot on ISO [Proxmox VE 7.2 ISO Installer](#) and apply the following config:

1. Accept EULA
2. Select the target installation disk
3. Enter Switzerland for the country, Zurich as the time zone and swiss-french keyboard layout
4. Choose `TMhes2022` as password and enter an email address
5. Configure network interface as specified in chapter 3.2
6. Click install and wait for the server to reboot
7. Go on <https://<host>:8006> and login with user `root` and password `TMhes2022`
8. On the left, go to local storage and ISO image, then click on upload



9. Select the ISO and click upload.
10. Go to CT template and click on Templates, choose Ubuntu-22.04-standard



11. In the top right corner, click on Create a VM, choose a name
12. Select the previously uploaded ISO
13. Keep default settings for the system
14. Change the size of the disk to 50 GB and activate discard
15. Give all the available CPU cores, check chapter 5.2 for the system specifications
16. Give all the available memory minus 2GB

17. Disable firewall on the network interface
18. Check start after created, connect to the VM console and proceed with the installation and benchmark as instructed in chapter 6.1
19. Once benchmarks on the VM are finished, power it off and click on create CT in the top right corner
20. Choose a name, a password and upload the SSH key
21. Select the previously downloaded template
22. Create a 50GB disk
23. Give all available core and all the memory minus 2GB
24. Configure the network and disable the firewall
25. Keep default DNS configs and confirm
26. Schedule only the normal benchmark

6.3.XCP-ng

Use the ISO [xcp-ng-8.2.1.iso](#) Apply the following settings:

1. Select the fr_CH keyboard
2. Press Ok to start the setup
3. Accept EULA
4. Select the disk and press Ok
5. Choose `Local media` as the installation source
6. Skip the media verification
7. Enter the password `TMhes2022`
8. Enter the network information
9. Choose Europe/Zurich as the location
10. Use NTP with server `pool.ntp.org`
11. Don't install supplemental packages
12. Reboot once installation is completed
13. Install [XCP-ng Center](#) on a Windows machine
14. Click on Add New Server on the top left
15. Enter the IP of the server and use the `root` username to login.
16. Click on save and restore server connection state on startup
17. Click on new storage, choose ISO storage NFS
18. Enter the information to connect to the on-site storage server which contains the ubuntu ISO.
19. Click on new VM, choose Ubuntu 20.04 (template for 22.04 isn't available yet but this one works)
20. Select the right ISO
21. Give all the CPU and all the memory available
22. Attach a GPU if available
23. Create a 50GB disk
24. Start the VM and follow instructions of chapter 6.1

Due to a bug, it is currently impossible to install XCP-ng on an AMD-B550 platform due to the installer crashing at boot. According to github.com/xcp-ng/xcp/issues/555, the problem is still present, and none of the proposed solutions worked. Consequences are that no XCP-ng benchmark were run on one of the systems.

6.4. Hyper-V Server

Boot ISO [hyperv.iso](#) and follow these instructions:

1. Set the right regional options
2. Accept EULA
3. Delete all partitions on disk and choose empty disk to install
4. Once the installation is finished, enter the password `TMhes2022` two times
5. Choose `8` to configure the network
6. Join an AD domain
7. Use the Hyper-V manager tool from another computer or server on the same domain to create a VM
8. Follow the instructions in chapter Bare metal

At first, I wanted to manage the Hyper-V server without joining a domain. But after multiple days of following guide or official documentation with no result, I came to the conclusion that it was quicker to install an AD domain on a side server. This server was also used to manage the Hyper-V host.

6.5. VMWare ESXi

The biggest problem with VMWare is the hardware compatibility. The different versions are not compatible with the same hardware, but it was important to use the same version to have meaningful benchmarking results. The main problem was the compatibility with the network cards. The drivers cannot be found easily and they are not compatible with every version of ESXi. After a lot of tries, I finally found that network and storage drivers that were compatible with ESXi 7.0 which worked on most of the targets. The only remaining problem is the storage driver not recognized on the most recent AMD system. CPU, memory and network benchmark have still been executed but none of the ones who uses storage were.

To inject the drivers and create the ISO, I used ESXi-Customizer-PS. As it is a PowerShell script it needs a windows PC to be run.

Second problem I had was that VMWare needs a second hard drive for the VM storage. To resolve it, I installed ESXi on a secondary disk and used the SSD as a VM datastore.

Once the ISO is ready, the installation is:

1. Accept the license
2. Choose disk for the installation
3. Reboot and configure network
4. Connect to the web interface
5. Create the VM datastore on the SSD
6. Connect to a SMB or NFS server for the ISO
9. Create the VM
10. Follow the instructions in chapter Bare metal