Building a virtual Beowulf cluster

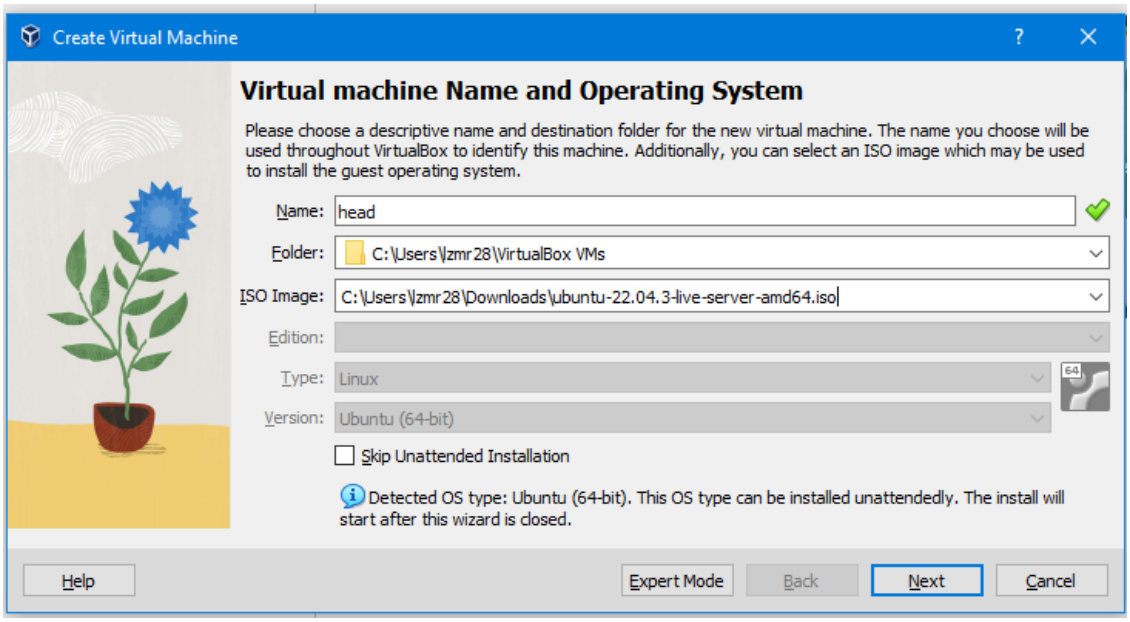
The goal of this practical is to configure a virtual mini cluster that can run distributed applications via MPI. We will start out by installing and configuring an operating system image for the head node, which we will then clone and use for both our compute nodes. Next, we will configure the shared file system NFS to enable access to shared data from any of the nodes. Afterwards, we will enable password-less ssh logins between all nodes such that our MPI processes are able to login to all nodes non-interactively. And finally, you will run your favourite MPI application on your cluster.

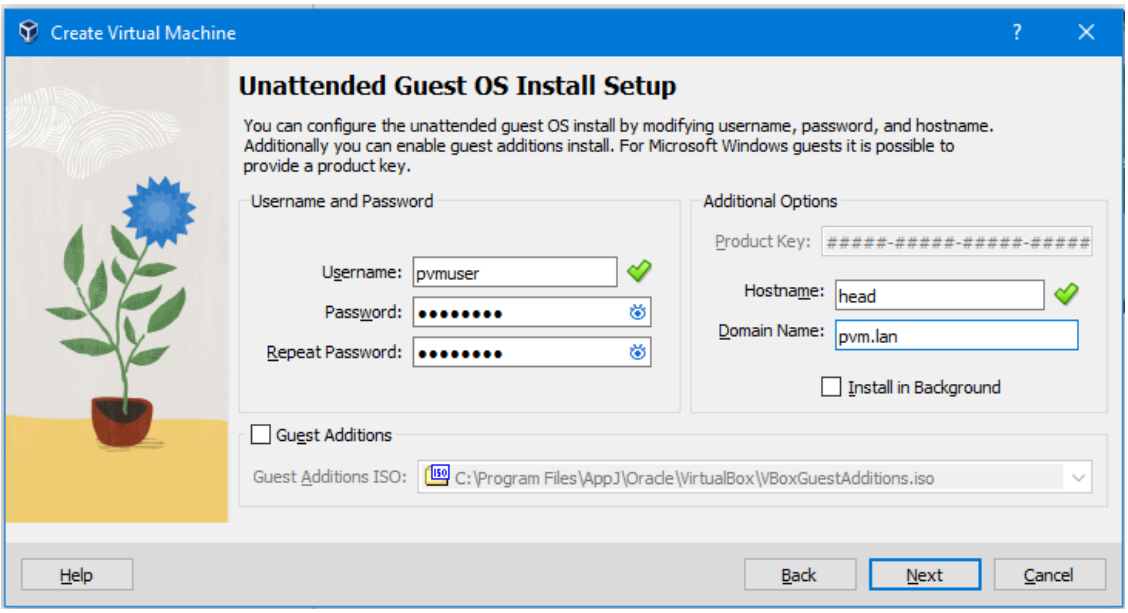
PS: While my instructions are (hopefully) rather detailed and complete for this practical, your goal should not be to blindly follow them but to understand the actual commands and interplay of the different concepts – so please refer to the man pages or external documentation to look up commands and parameters that we will use throughout the practical.

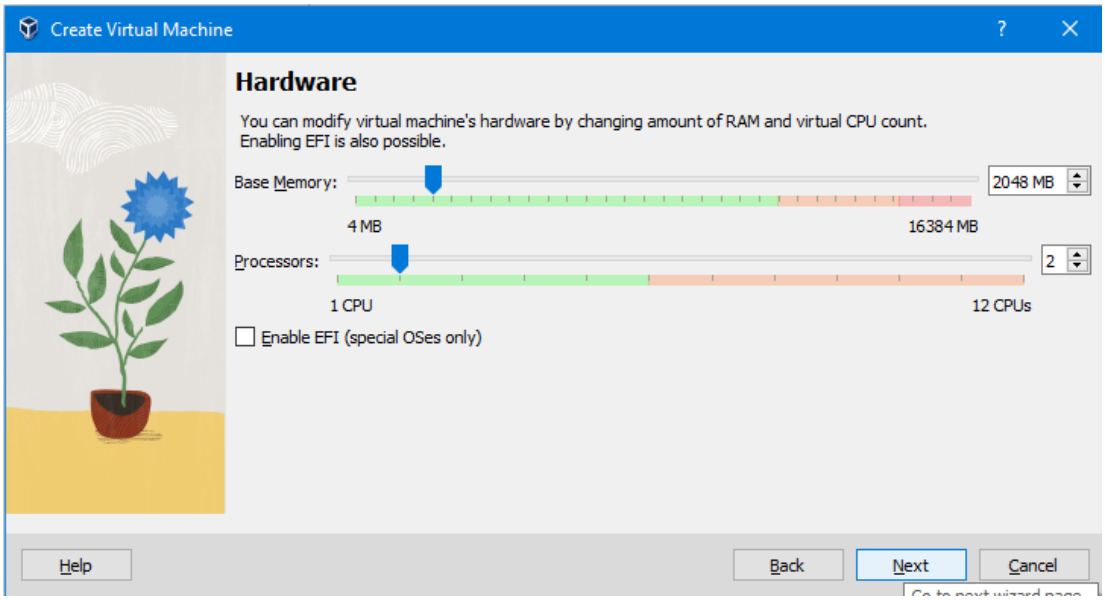
PPS: You are absolutely welcome to configure the cluster to your likings - as long as you can comfortably run MPI applications on it afterwards. For instance, you are free to choose your favourite Linux distro or MPI implementation as well as use a different network configuration or file system.

On the host system:

1. Download the Ubuntu Server 22.04.3 LTS image from https://ubuntu.com/download/server.
2. Open VirtualBox via AppsAnywhere (https://appsanywhere.durham.ac.uk/login).
3. In VirtualBox, create a new VM based on the image by clicking "New" and following the guided VM creation to achieve settings similar to the screenshots.







In the VM:

1. Install the operating system. Ensure to install OpenSSH during this process.
2. Login with your chosen credentials.
3. Install the following:

```
sudo apt update
sudo apt install net-tools
sudo apt install nfs-kernel-server
sudo apt install nfs-common
sudo apt install build-essential
sudo apt install openmpi-bin openmpi-doc libopenmpi-dev
```

4. Shutdown the VM in order to allow cloning:

```
shutdown -h now
```

In VirtualBox on the host system:

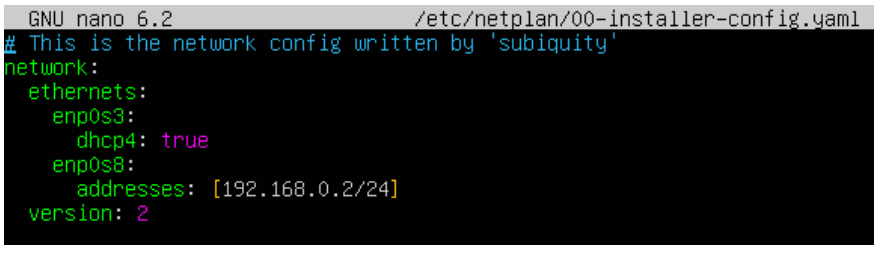
1. Create two clones of the VM: *compute1* and *compute2.*
2. Configure the *Network Adapter 1* for all nodes (head, compute1, compute2) as attached to *NAT*.
3. Configure the *Network Adapter 2* for all nodes (head, compute1, compute2) as attached to *Internal Network* (must be the same for all nodes).
4. Boot up *all* nodes.

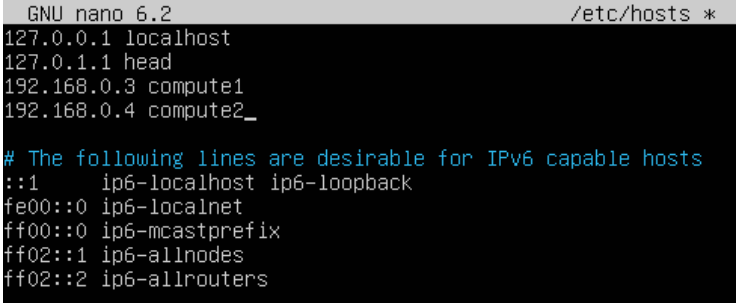
On compute1 and compute2:

1. Change hostname from head to compute1 or compute2, respectively, in the following files:
   ```
   sudo nano /etc/hostname
   sudo nano /etc/hosts
   ```
2. Reboot for the changes to become effective:
   ```
   reboot
   ```

On all VMs:

1. Determine the name of the second network interface via:
   ```
   ifconfig -a
   ```
2. Assign a unique static IP within the same subnet (e.g. 192.168.0.2/24; 192.168.0.3/24; 192.168.0.4/24) to each node by modifying the network configuration:
   ```
   sudo nano /etc/netplan/00-installer-config.yaml
   sudo netplan apply
   ```
3. Check whether all nodes can mutually reach each other:
   ```
   ping <ip_addr>
   ```
4. Adjust /etc/hosts such that the static IPs match hostnames:
   ```
   sudo nano /etc/hosts
   ```
5. Check whether all nodes can mutually reach each other:
   ```
   ping <hostname>
   ```





On head:

1. Start NFS server via command:
   ```
   sudo systemctl start nfs-kernel-server.service
   ```
2. Edit /etc/exports on login by adding line:
   ```
   /home *(rw,async,no_subtree_check,no_root_squash)
   ```
3. Safe and afterwards execute:
   ```
   sudo exportfs -a
   ```

On compute1 and compute2:

1. Mount home directory of head into home directory of node:
   ```
   sudo mount head:/home /home
   cd
   ```
2. Create a file in the home directory on compute1 and check whether it shows up in the home directory on head and compute2.
3. After reboot, the directory would not be mounted anymore. Therefore, we have to add the following line to /etc/fstab:
   ```
   head:/home /home nfs    defaults    0     0
   ```
4. Reboot and check whether the remote home directory is mounted:
   ```
   findmnt
   ```

On any node:

1. Generate ssh keys for password-less logins between all nodes (save to the default location and don't enter a password for the ssh keypair):
   ```
   ssh-keygen
   cp .ssh/id_rsa.pub .ssh/authorized_keys
   ```
2. Disable host key checking by generating a file ~/.ssh/config with the following content:
   ```
   Host *
         StricHostKeyChecking no
   ```

On any node:

1. Test your MPI installation by running:
   ```
   mpirun -np 3 head,compute1,compute2 hostname
   ```
2. Congratulations! You have now manually configured your very own mini cluster – and are now able to run your favourite MPI application.