Linux emulator - QEMU

Ngoc Nguyen Tran

Contents

- Simulator(s) and virtualizator(s)
- QEMU introduction
- QEMU applications
- QEMU basic usages
- Exercises

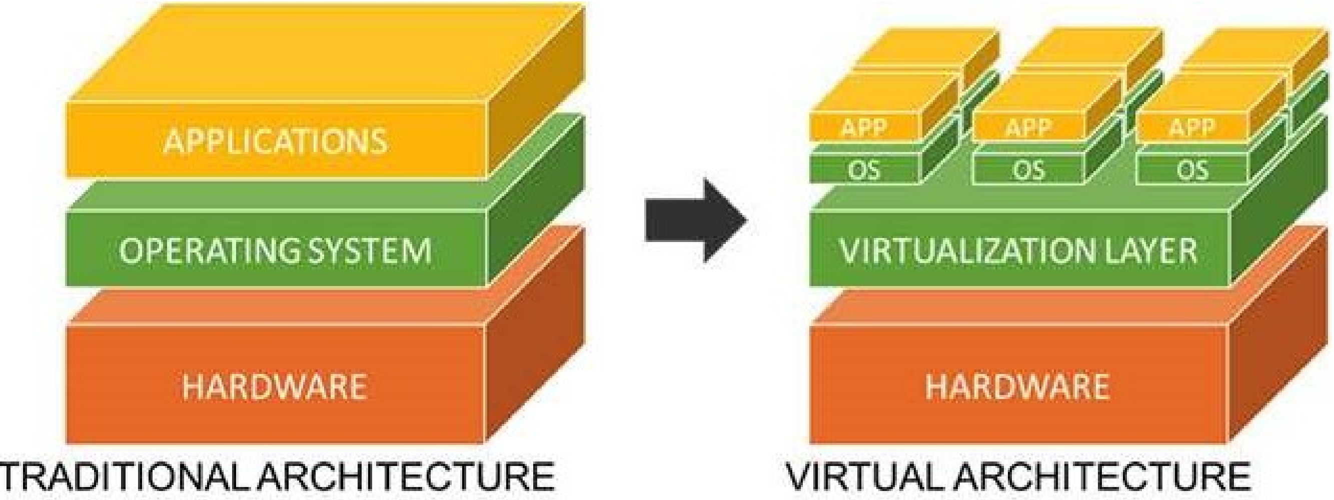
Why simulators/ virtualizators?

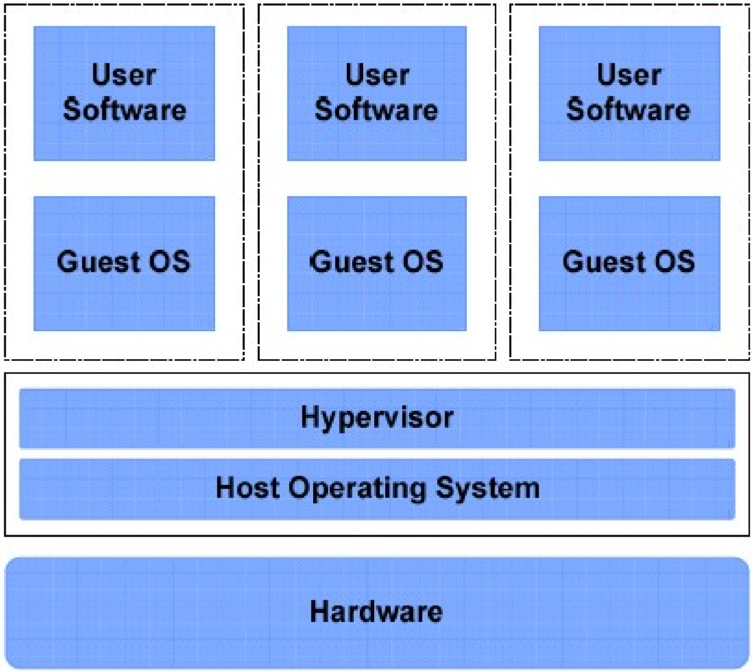
- Need a simulated/emulated environment for embedded Linux systems if needed
- Optimise our PC systems
- Easily test multiple systems before working with the real ones
- Easier and faster for testing

Simulation vs emulation vs virtualisation?

- *Simulators* and *emulators* both let you run software in one environment that's meant for another
  - Emulator: duplicating every aspect of the original device's behavior
  - Simulator: sets up a similar environment to the original device's OS but doesn't attempt to simulate the real device's hardware. Its performance is usually slower than emulator
- Virtualisation:
  - Create a virtual computer architecture
  - Modern virtualization tools create a completely software-based facsimile of a real computer hardware profile and run a genuine operating system
  - Can be considered as *upgrading emulators*

Virtualization diagram





Virtualisastion/Emulation for embedded Linux

- Some notable VM application
  - VMWare
  - Oracle VirtualBox
- However, they are not designed for embedded systems
- Kernel-based Virtual Machine (KVM): virtualization module in the Linux kernel that allows the kernel to function as a hypervisor
- QEMU
  - Free and open-source emulator/virtualizator
  - Native Linux
  - Emulates the machine's processor
    - Dynamic binary translation
    - Provides a set of different hardware and device models for the machine
  - *Allow applications compiled for one architecture to run on another*
  - *High performance*
  - VirtualBox used some of QEMU's virtual hardware devices, and had a built-in dynamic recompiler based on QEMU
  - Easy to integrate with KVM

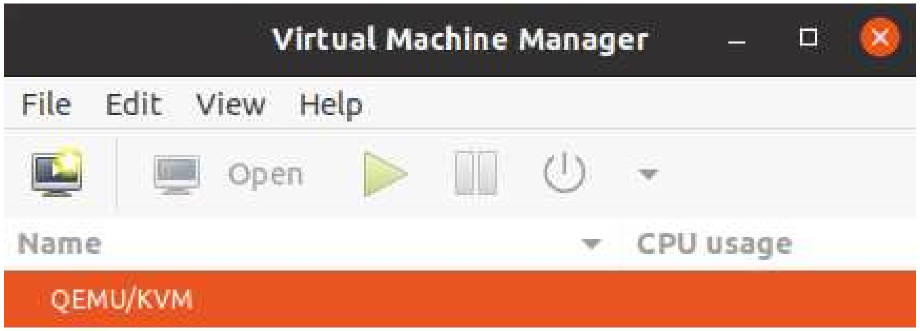
QEMU emulated hardware platforms

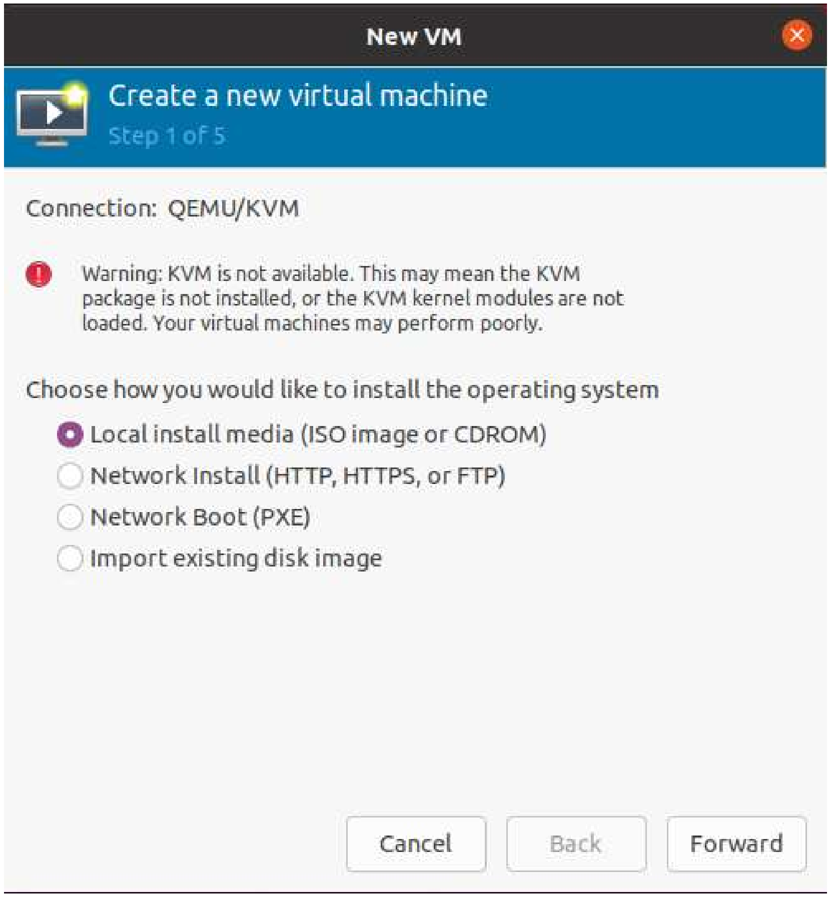
- x86
- PowerPC
- ARM
- SPARC
- MicroBlaze
- LatticeMico32
- CRIS
- OpenRISC

QEMU installation

- Available from the official repository of Ubuntu
- Update Ubuntu repository: sudo apt update
- Install QEMU as a normal application
  - sudo apt install qemu-kvm virtinst qemu virt-manager libvirt-daemon libvirt-daemon-system bridge-utils virt-viewer libvirt-clients
  - qemu is the application
  - qemu-kvm is needed for process virtualisation
  - virt-manager and virt-view are installed for GUI purposes
  - libvirt contains binaries for both QEMU and KVM
- Check the status of libvirt
  - sudo systemctl status libvirtd
  - systemctl: a command to configure services/applications (will be mentioned next lecture)

Install an emutated OS using Virt Manager





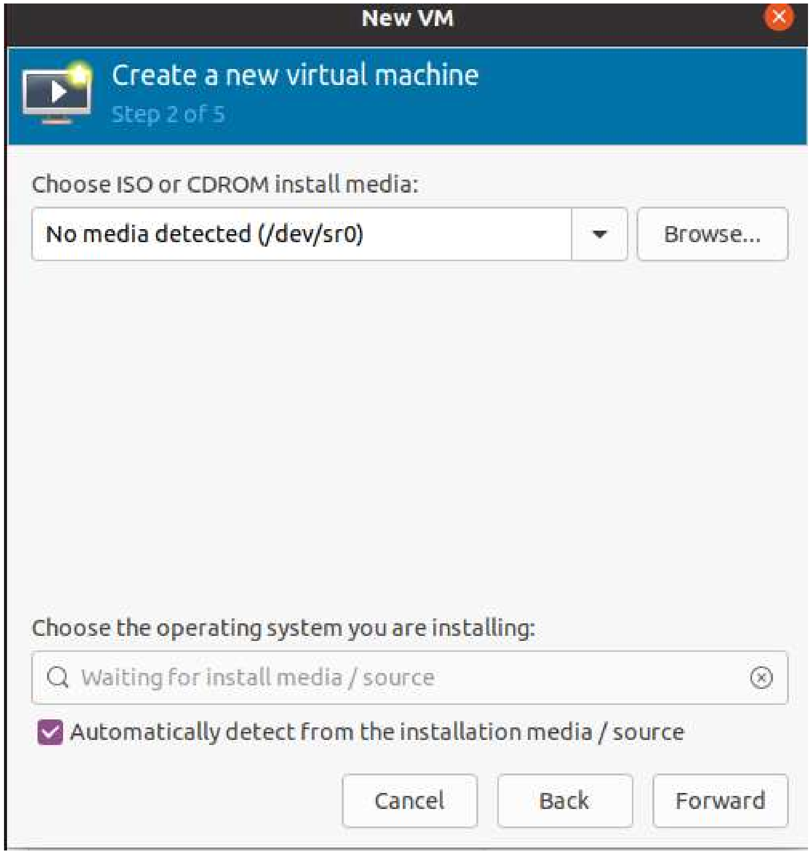
Connection: QEMU/KVM

Warning: KVM is not available. This may mean the KVM package is not installed, or the KVM kernel modules are not loaded. Your virtual machines may perform poorly.

Choose how you would like to install the operating system

- ● Local install media (ISO image or CDROM)
- ○ Network Install (HTTP, HTTPS, or FTP)
- ○ Network Boot (PXE)
- ○ Import existing disk image

Install an emutated OS using Virt Manager (2)



Create a new virtual machine

Step 2 of 5

Choose ISO or CDROM install media:

No media detected (/dev/sr0)

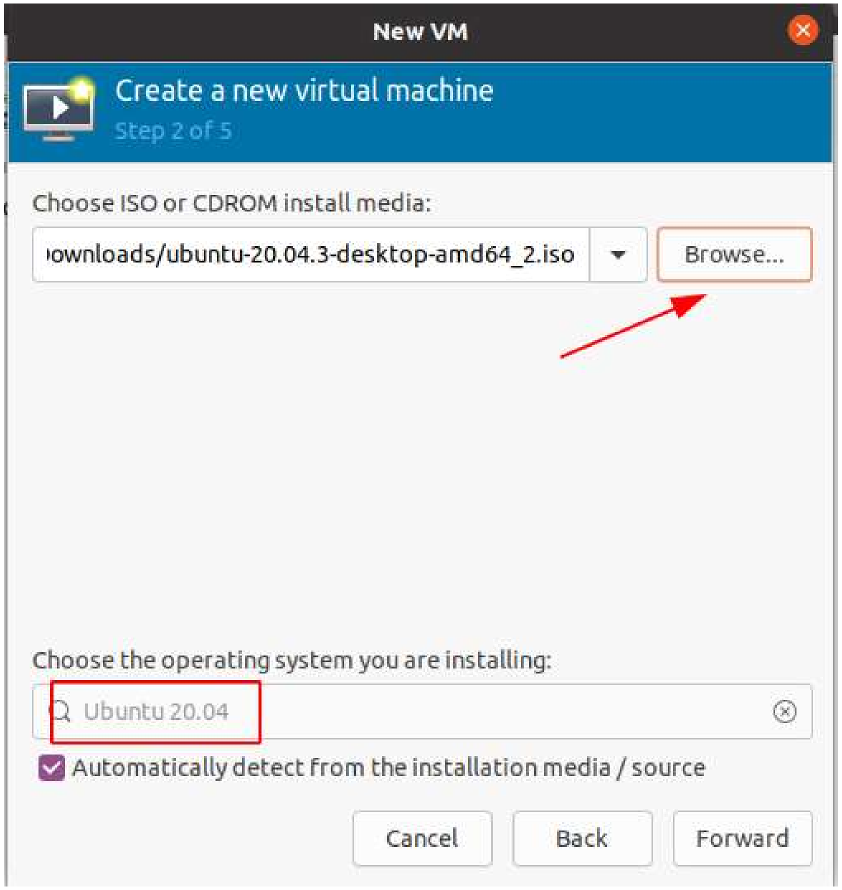
Choose the operating system you are installing:

Waiting for install media / source

☑ Automatically detect from the installation media / source

- We need the installation file, please download it

Install an emutated OS using Virt Manager (3)



Create a new virtual machine

Step 2 of 5

Choose ISO or CDROM install media:

›ownloads/ubuntu-20.04.3-desktop-amd64_2.iso

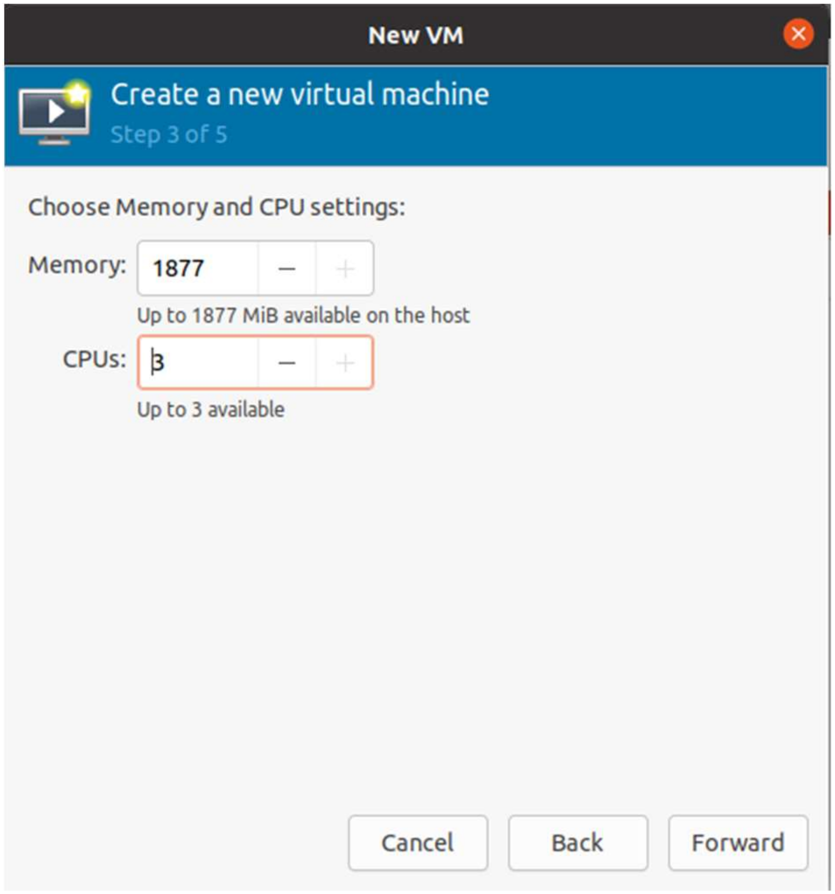
Choose the operating system you are installing:

Ubuntu 20.04

☑ Automatically detect from the installation media / source

- Example of installing Ubuntu 20.04.3
- Automatically dectect the OS from the installation media

Install an emutated OS using Virt Manager (4)



New VM

Create a new virtual machine

Step 3 of 5

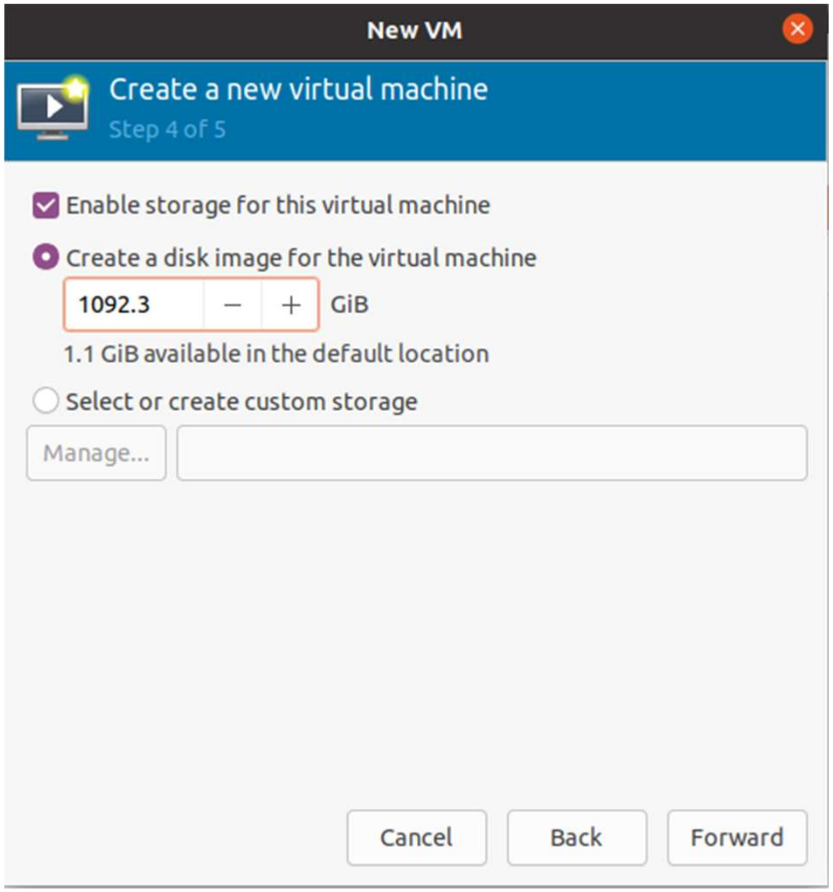
Choose Memory and CPU settings:

Memory: 1877

Up to 1877 MiB available on the host

CPUs: 3

Up to 3 available



New VM

Create a new virtual machine

Step 4 of 5

☑ Enable storage for this virtual machine

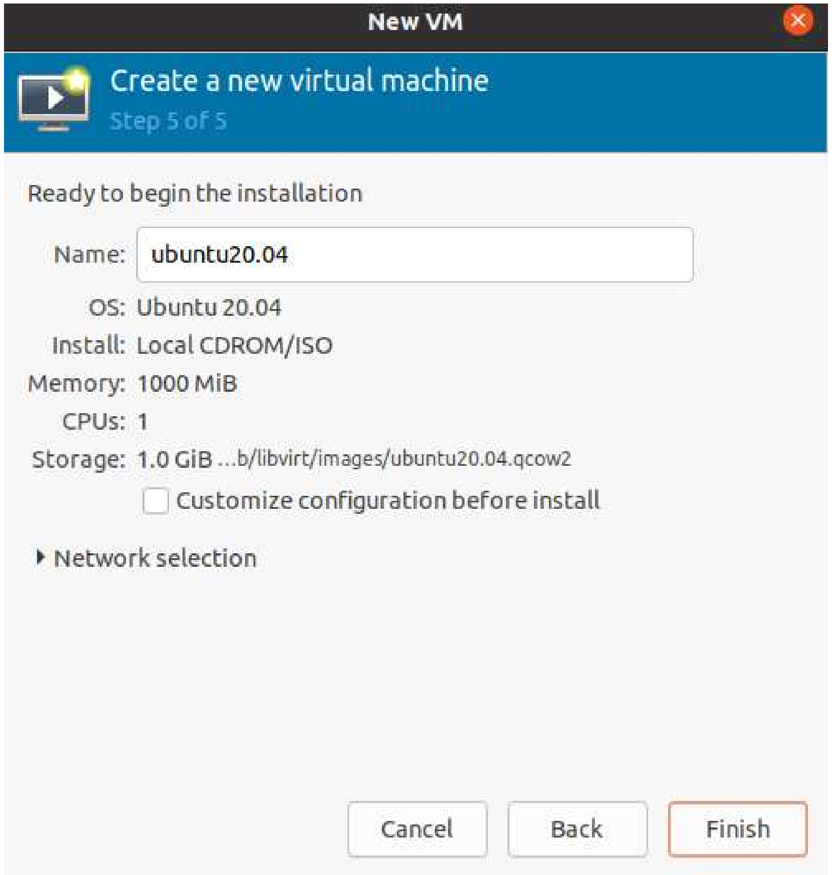
● Create a disk image for the virtual machine

1092.3 GiB

1.1 GiB available in the default location

○ Select or create custom storage

Install an emutated OS using Virt Manager (5)



New VM

Create a new virtual machine

Step 5 of 5

Ready to begin the installation

Name: ubuntu20.04

OS: Ubuntu 20.04

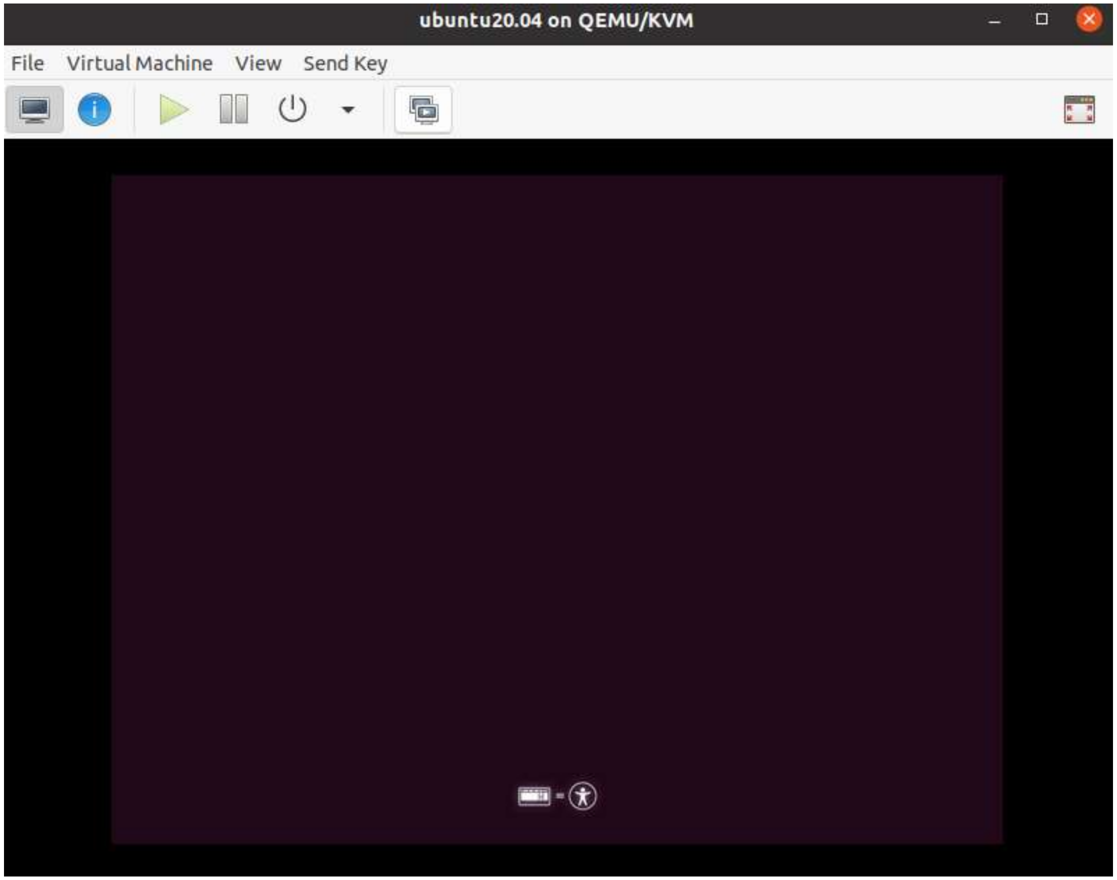
Install: Local CDROM/ISO

Memory: 1000 MiB

CPUs: 1

Storage: 1.0 GiB ...b/libvirt/images/ubuntu20.04.qcow2

□ Customize configuration before install



ubuntu20.04 on QEMU/KVM

Exercise(s)

- Try to run a 64-bit OS image (Fedora, CentOS, Ubuntu, etc.) using command lines of qemu-system-x86_64
- Type *qemu-system-x86_64 –h* to see options
- Syntax: *qemu-system-x86_64 OPTIONS IMAGE*
  - *IMAGE is the file name of the disk for the VM*
  - *OPTIONS are provided for VM options like Ram size, features, CPU, Graphics, etc.*
- You will need a virtual machine images (in various formats like raw, qcow2, vmdk, etc) of one OS
  - Example: http://www.nic.funet.fi/pub/mirrors/fedora.redhat.com/pub/fedora/linux/releases/30/Cloud/x86_64/images/Fedora-Cloud-Base-30-1.2.x86_64.qcow2
  - Use wget or curl to download
- Some important options:
  - -smp: multiple cores
  - -boot: select boot options
  - -m: memory size
  - -name: setting name
  - -drive: specify disk file or image
  - -nographic: disable GUI
  - -net: set networking for your Linux machine