Cloud Computing Applications and Services

(Aplicações e Serviços de Computação em Nuvem)

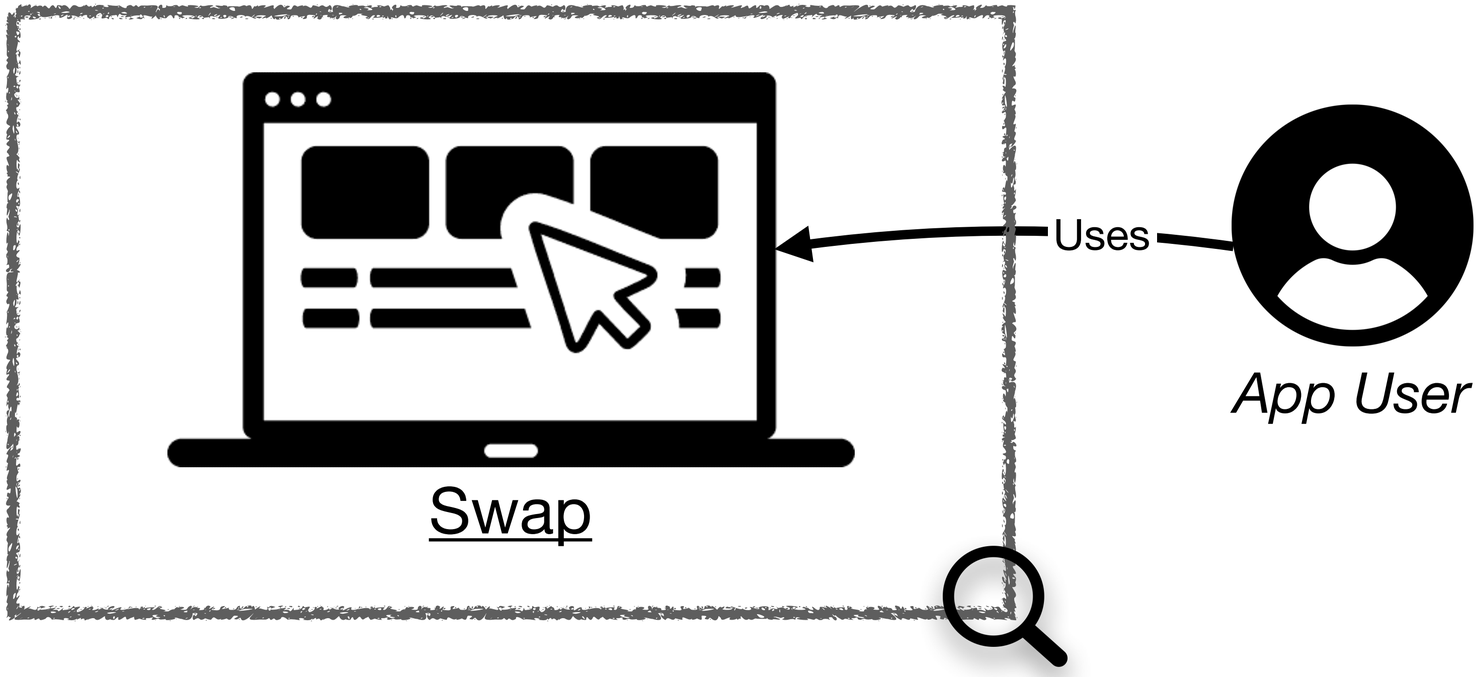
Guide 0: Warm-Up

University of Minho

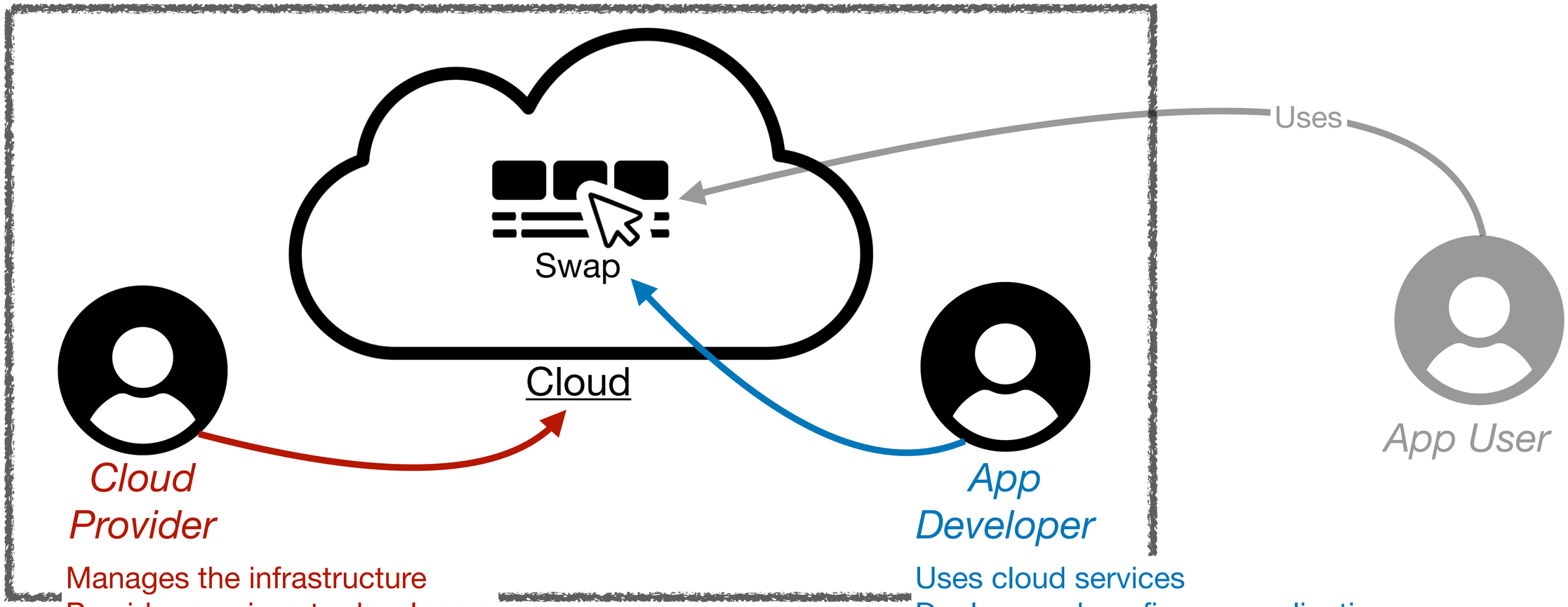
2024-2025

Context

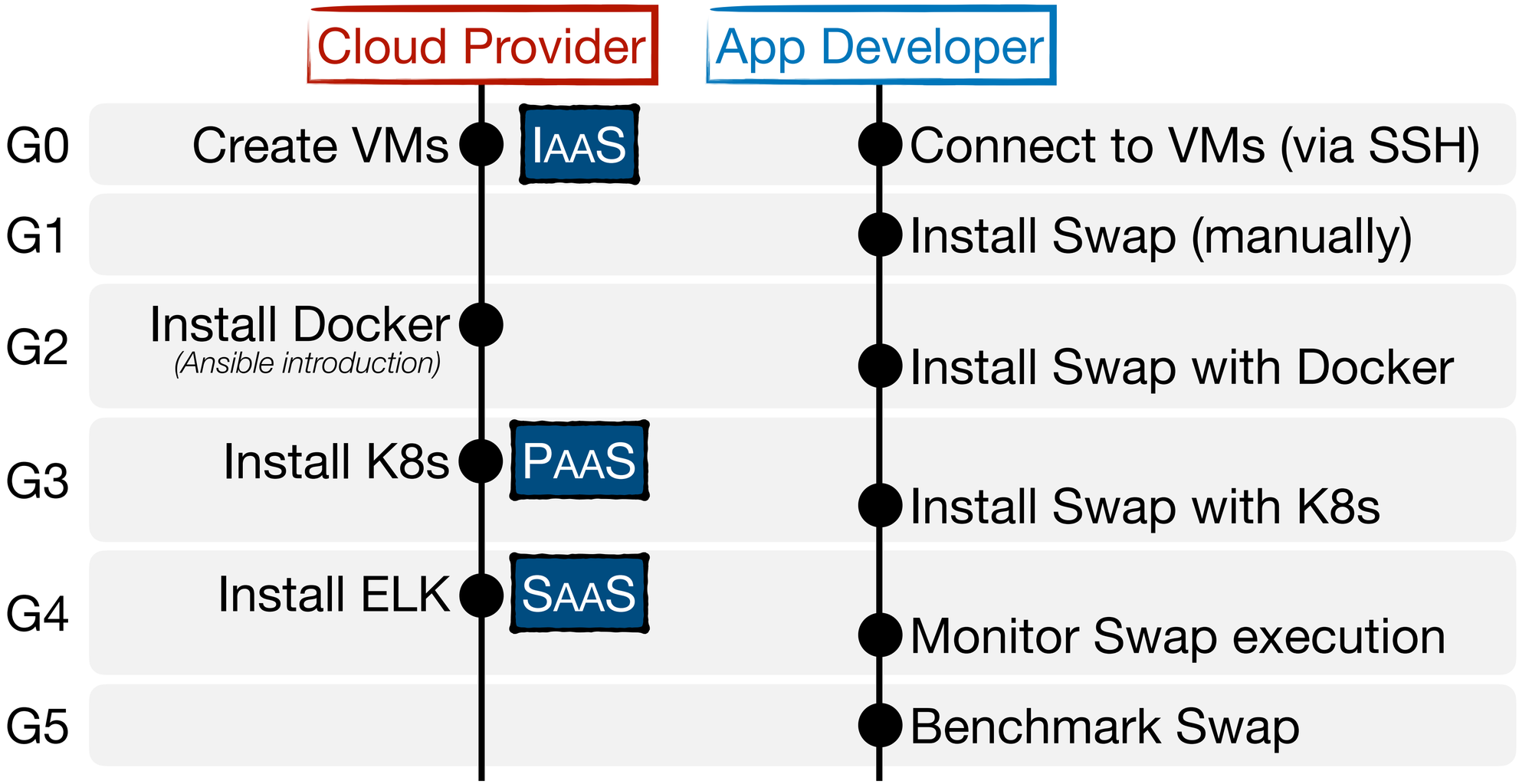
- The University of Minho hired you to deploy and manage Swap, a service for students swapping practical shifts.



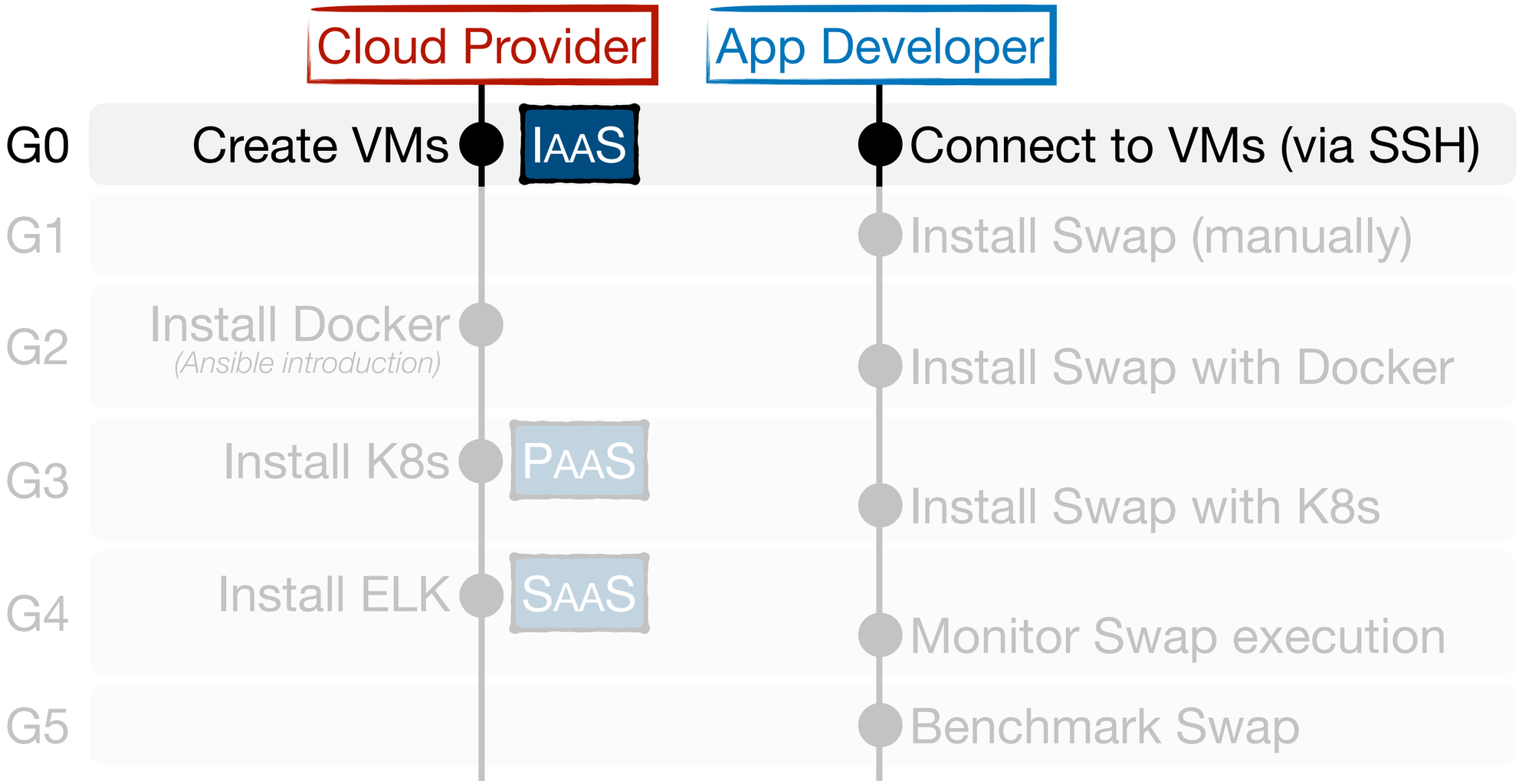
Context


Swap
Cloud
Uses
App User
Cloud Provider
App Developer
Manages the infrastructure
Provides services to developers
Uses cloud services
Deploys and configures applications

Road Map


Cloud Provider
App Developer
G0
Create VMs
IAAS
Connect to VMs (via SSH)
G1
Install Swap (manually)
G2
Install Docker
(Ansible introduction)
Install Swap with Docker
G3
Install K8s
PAAS
Install Swap with K8s
G4
Install ELK
SAAS
Monitor Swap execution
G5
Benchmark Swap

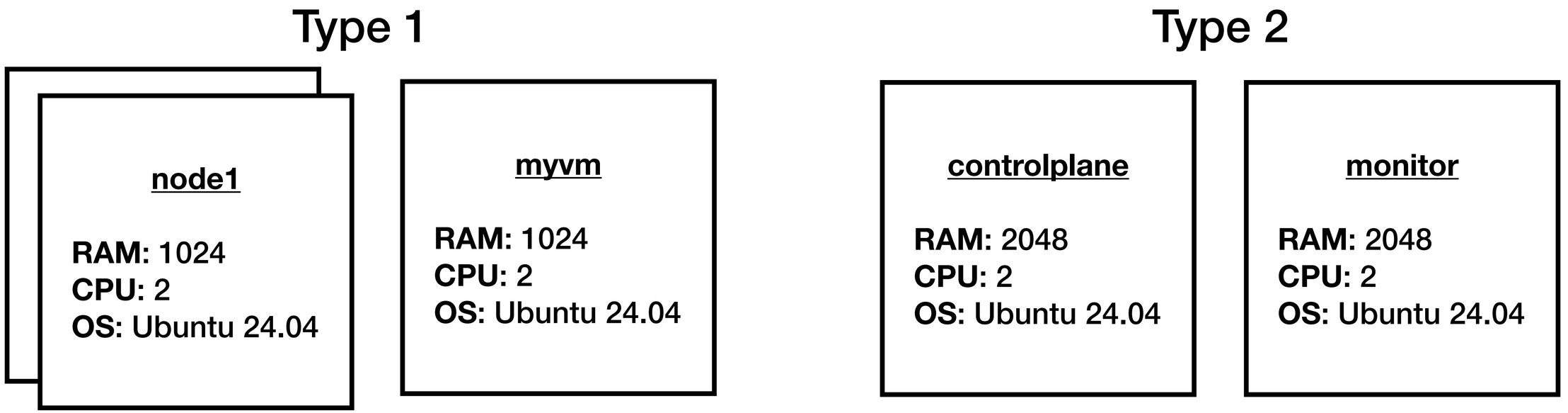
Road Map


Cloud Provider
App Developer
G0
Create VMs
IaaS
Connect to VMs (via SSH)
G1
Install Swap (manually)
G2
Install Docker
(Ansible introduction)
Install Swap with Docker
G3
Install K8s
PaaS
Install Swap with K8s
G4
Install ELK
SaaS
Monitor Swap execution
G5
Benchmark Swap

Goal

Cloud Provider

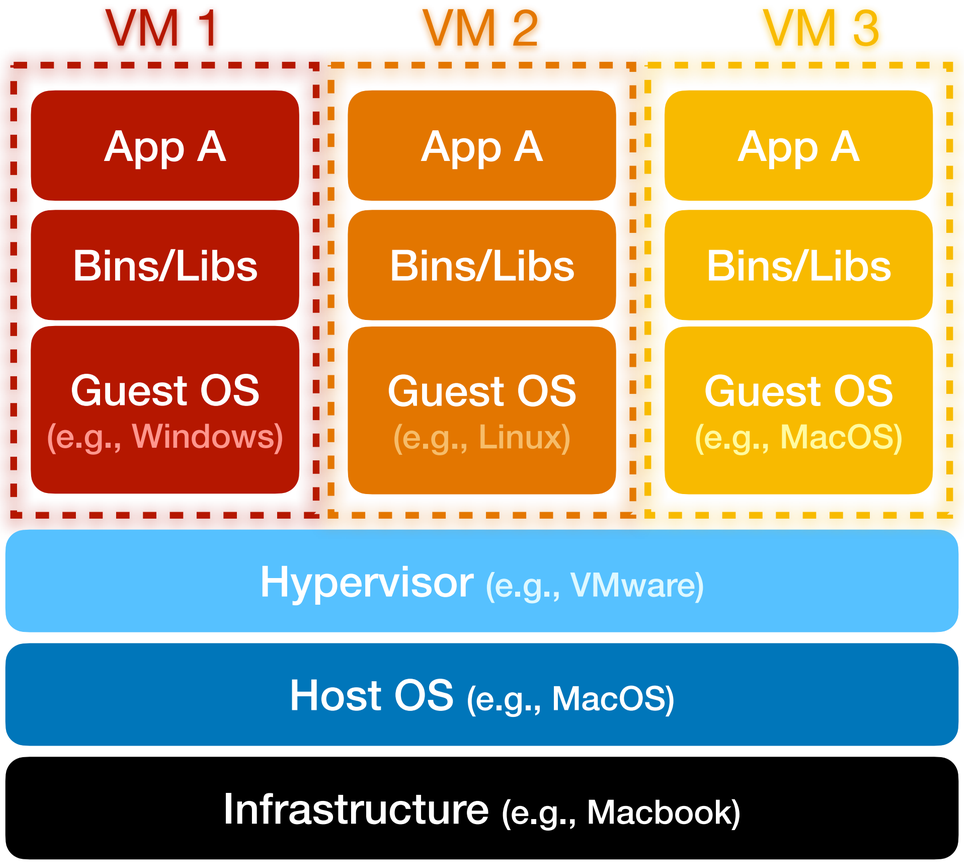
- Use Vagrant to create and configure 5 VMware virtual machines



Useful Tools and Concepts

VMware

- **VMware** is a virtualization software for running multiple operating systems in a single host.
- Want to know more about virtualization and virtual machines? Stay tuned for theoretical classes!



Vagrant

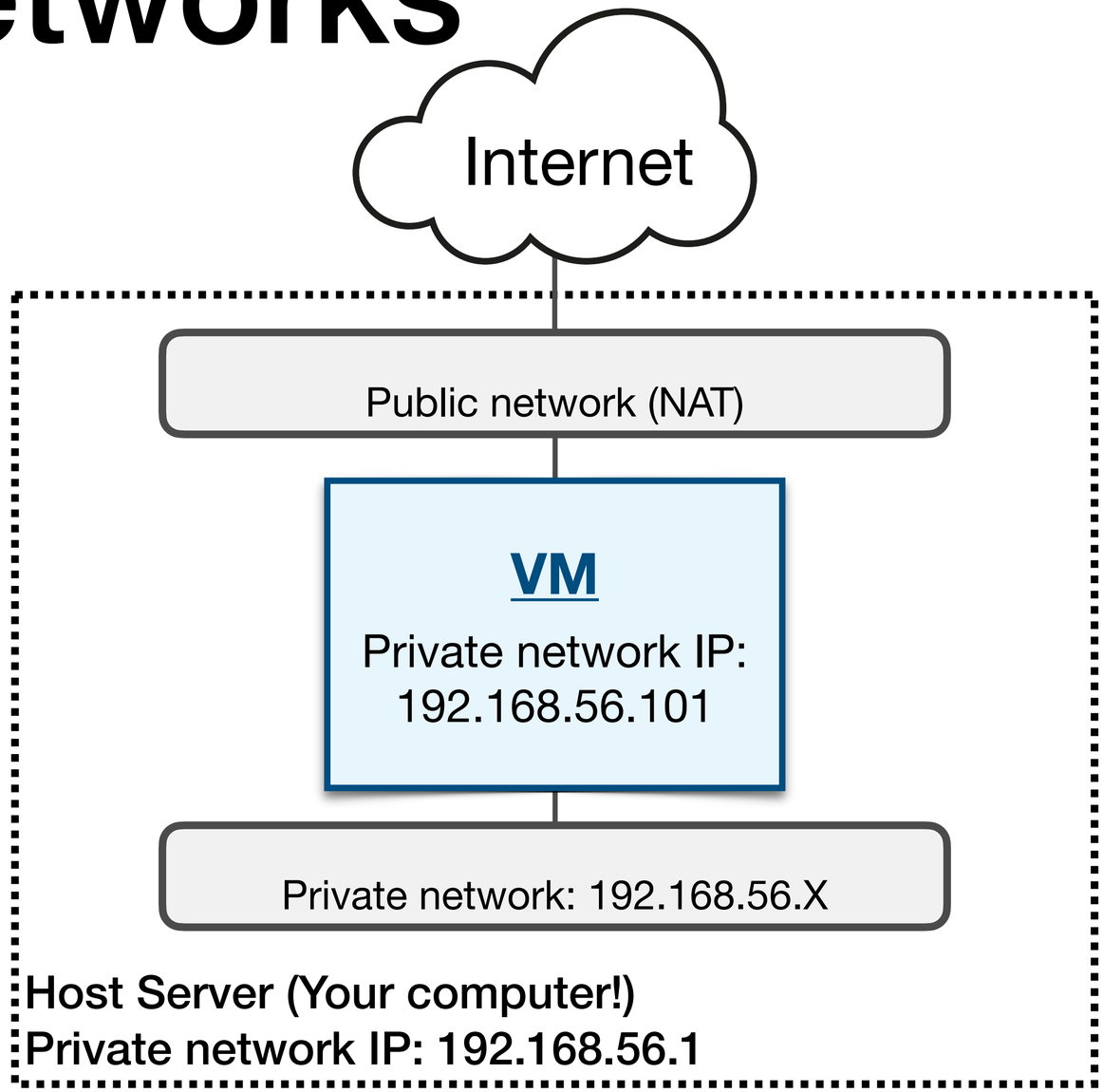
- **Vagrant** is a software for building and maintaining portable virtual software development environments (e.g., for VMware, VirtualBox, etc).
- Vagrant uses base images (known as **boxes**) to quickly clone a VM.
  - E.g., the box "bento/ubuntu-24.04" provides a base image of Ubuntu 24.04.
- The type of virtual machines and the tasks of how to configure and provision them are defined in a **Vagrantfile**.

Vagrant
Useful Commands

- **Start** the VMs:
  - vagrant up
  - vagrant up <vm_name>
- **Stop** the VMs:
  - vagrant halt
  - vagrant halt <vm_name>
- Output the **status** of the VMs:
  - vagrant status
  - vagrant status <vm_name>
- **Provision** the VMs (forces reprovisioning of the VMs):
  - vagrant provision
  - vagrant provision <vm_name>
- **Clean up** the VMs (stops and deletes the VMs):
  - vagrant destroy
  - vagrant destroy -f
  - vagrant destroy <vm_name>
  - vagrant destroy -f <vm_name>

Virtual Networks

- **NAT (Public network)**
  - VMs access the Internet through (masked as) the Host IP
- **Host-only (Private Network)**
  - Only the Host and VMs can communicate with each other



SSH

- **SSH (secure shell)** is a network protocol that allows secure remote access to systems over an unsecured network.
- SSH uses a client-server model, where the client connects to the server and establishes a secure channel for communication.
  - **SSH Server (sshd)**: program that runs on the remote system and listens for incoming SSH connections.
  - **SSH Client (ssh)**: program used to initiate an SSH connection to a remote server.

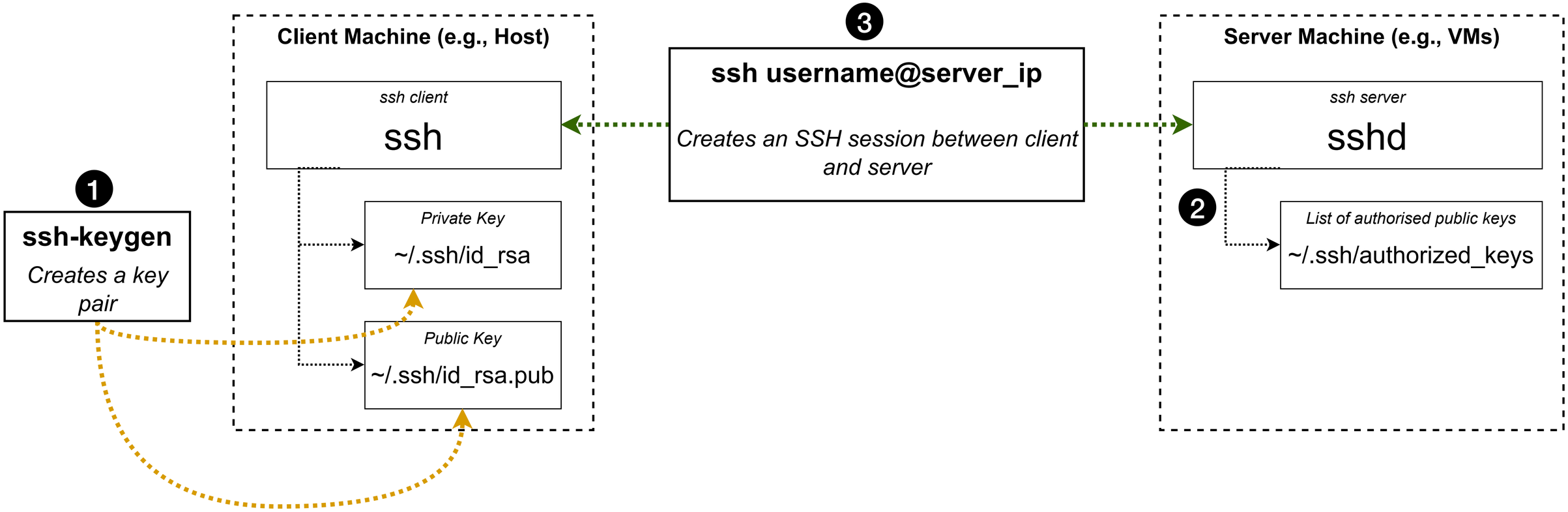
SSH

- **SSH keys**: SSH uses public-key cryptography to authenticate and establish secure connections.
  - The **private key (id_rsa)** is secret, known only to the user, and should remain stored safely (e.g., in the user's laptop accessing the remote server).
  - The **public key (id_rsa.pub)** can be shared freely with any SSH server to which the user wishes to connect (e.g., VM).
  - The **authorized Keys** (file **~/.ssh/authorized_keys** on the server) specifies the SSH public keys that can be used for logging into the remote server.

SSH
Useful Commands

- Generate **new SSH keys**:
  - ssh-keygen
- **Copy the Pub key** to a remote server:
  - ssh-copy-id <username>@<vm_ip>
- **Connect** to a remote server:
  - ssh <username>@<vm_ip>
- Copy a file from local to remote:
  - scp <local_file_path> <username>@<vm_ip>:<remote_dst_path>
- Copy a folder from local to remote:
  - scp -r <local_folder_path> <username>@<vm_ip>:<remote_dst_path>
- Copy a file from remote to local:
  - scp <username>@<vm_ip>:<remote_file_path> <local_dst_path>
- Copy a folder from remote to local:
  - scp -r <username>@<vm_ip>:<remote_folder_path> <local_dst_path>

SSH



Note: Step ❶ needs to be done manually. Step ❷ is specified at the Vagrantfile.

Example: For accessing *node1,* one should run ssh vagrant@192.168.56.101 at the host's terminal (step ❸).