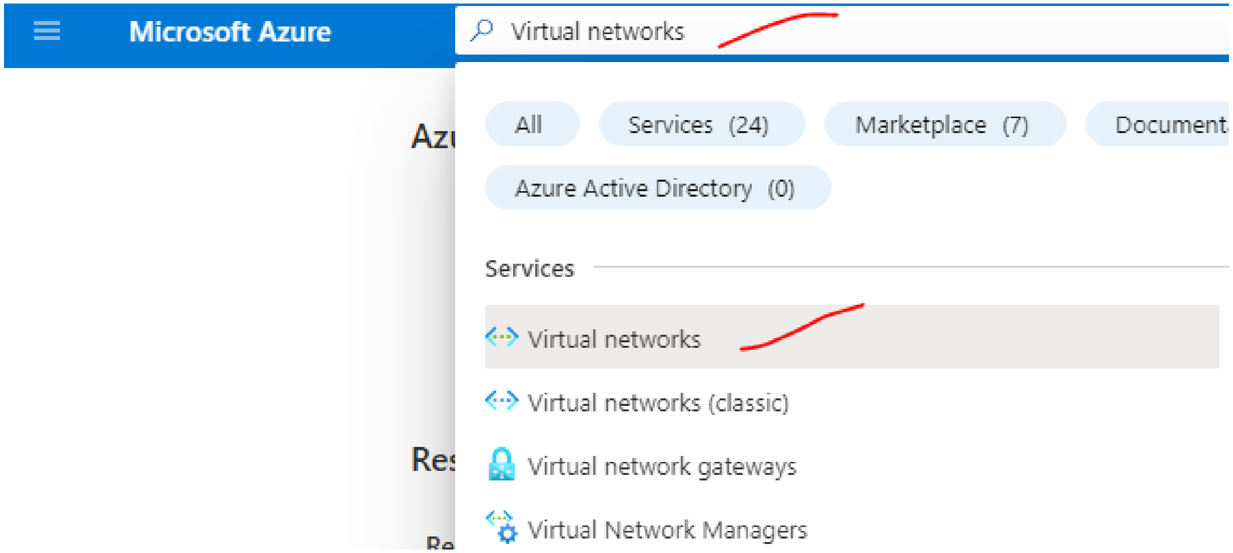
Please use a Screen or Video Capture software to save your works!

OBJECTIVE & PREPARATION

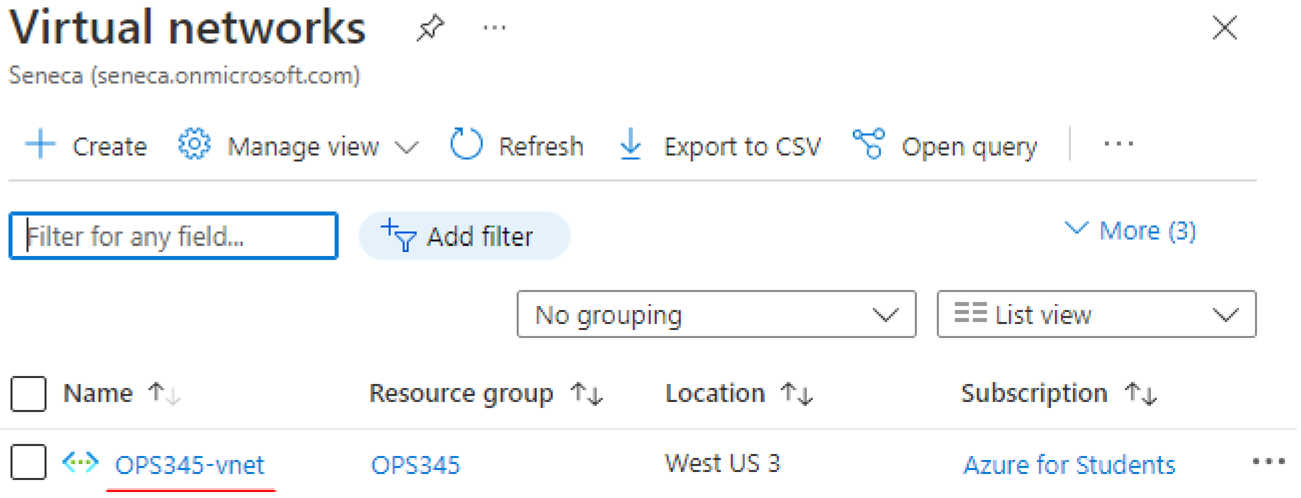
In OPS345, you will set up a static network connection for all of your VMs. All of the services that we install and configure for this course **require a working network connection**; therefore, it is very important that you know how to configure a network connection for your VMs, whether via command line for trouble-shooting purposes, or to create a persistent (permanent) network connection that uses static IP address (as opposed to DHCP).

Case 1: Prepare VMs for Lab 01 on Azure

1. Please make sure your **vm1** that created in PrepLab still functional.
2. Please created another VM named as "**vm2**". (refer to PrepLab for details)
3. Each VM you have created will have a Public IP address NAT to a Private 10.0.0.0/24 IP. Now, let's add another Private 192.168.0.0/24 IP address. You must not modify the 10.0.0.0/24 network, otherwise, you will lose the connection to the VMs. All of our network configuration tasks will be done on 192.168.0.0/24.
4. Search "Virtual networks"



5. There should be an OPS345-vnet created already.

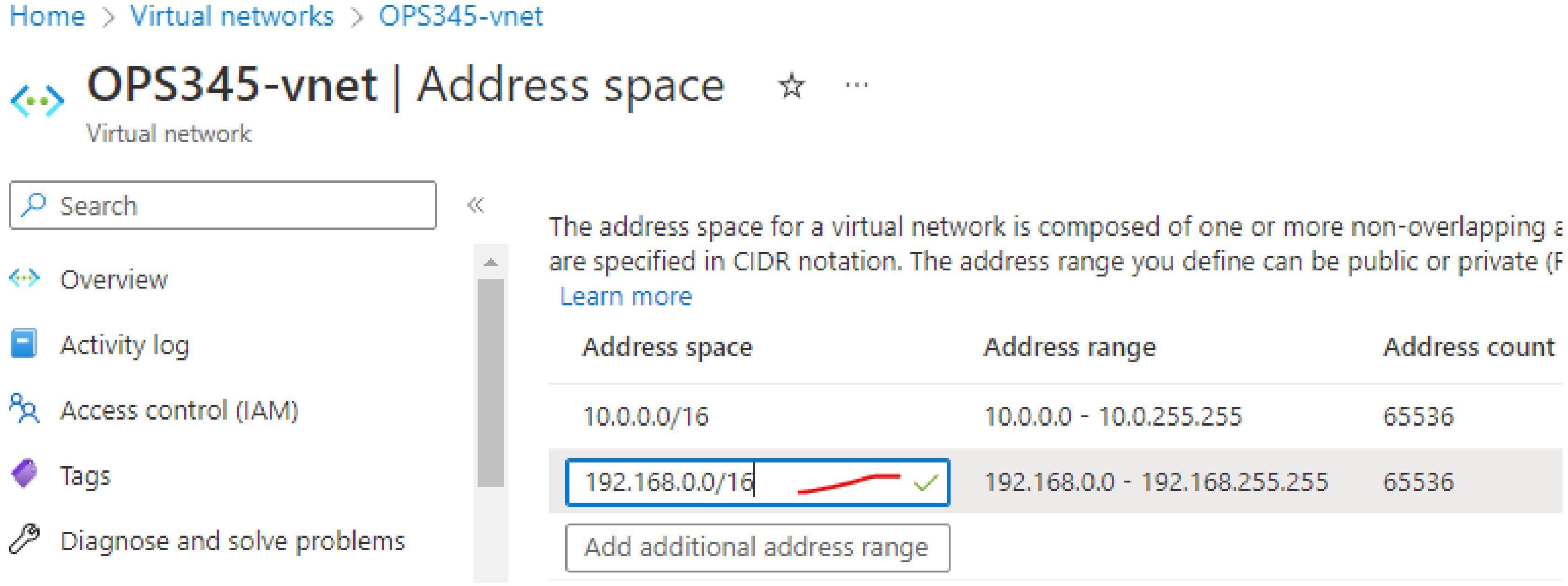


Virtual networks

Seneca (seneca.onmicrosoft.com)

Name	Resource group	Location	Subscription
OPS345-vnet	OPS345	West US 3	Azure for Students

6. Address space: add 192.168.0.0/16


Home > Virtual networks > OPS345-vnet
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Learn more
Add additional address range

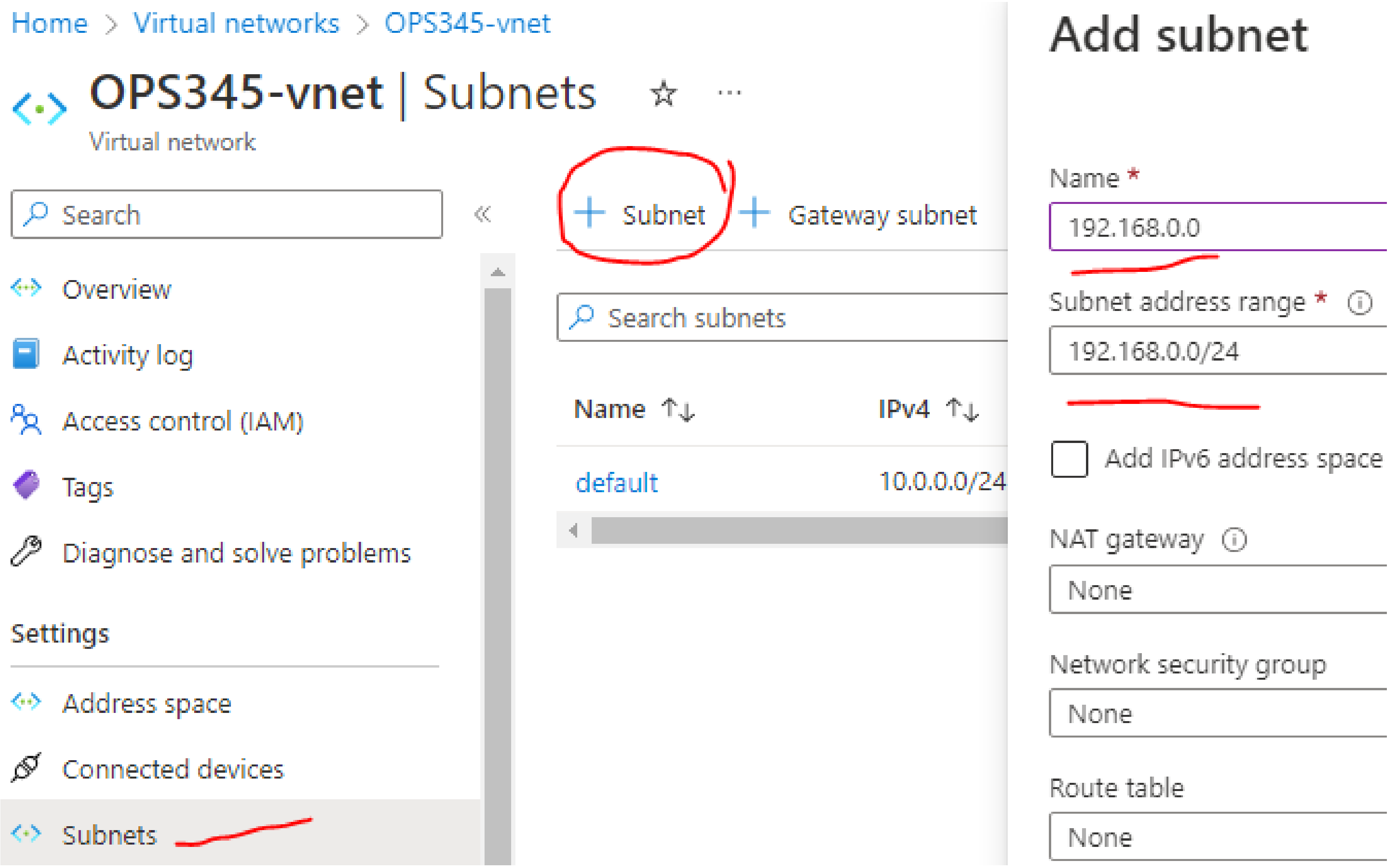
OPS345-vnet | Address space

Virtual network

The address space for a virtual network is composed of one or more non-overlapping a
are specified in CIDR notation. The address range you define can be public or private (F

Address space	Address range	Address count
10.0.0.0/16	10.0.0.0 - 10.0.255.255	65536
192.168.0.0/16	192.168.0.0 - 192.168.255.255	65536

7. Create a subnet.



Name	IPv4
default	10.0.0.0/24

Name * 192.168.0.0

Subnet address range * 192.168.0.0/24

□ Add IPv6 address space

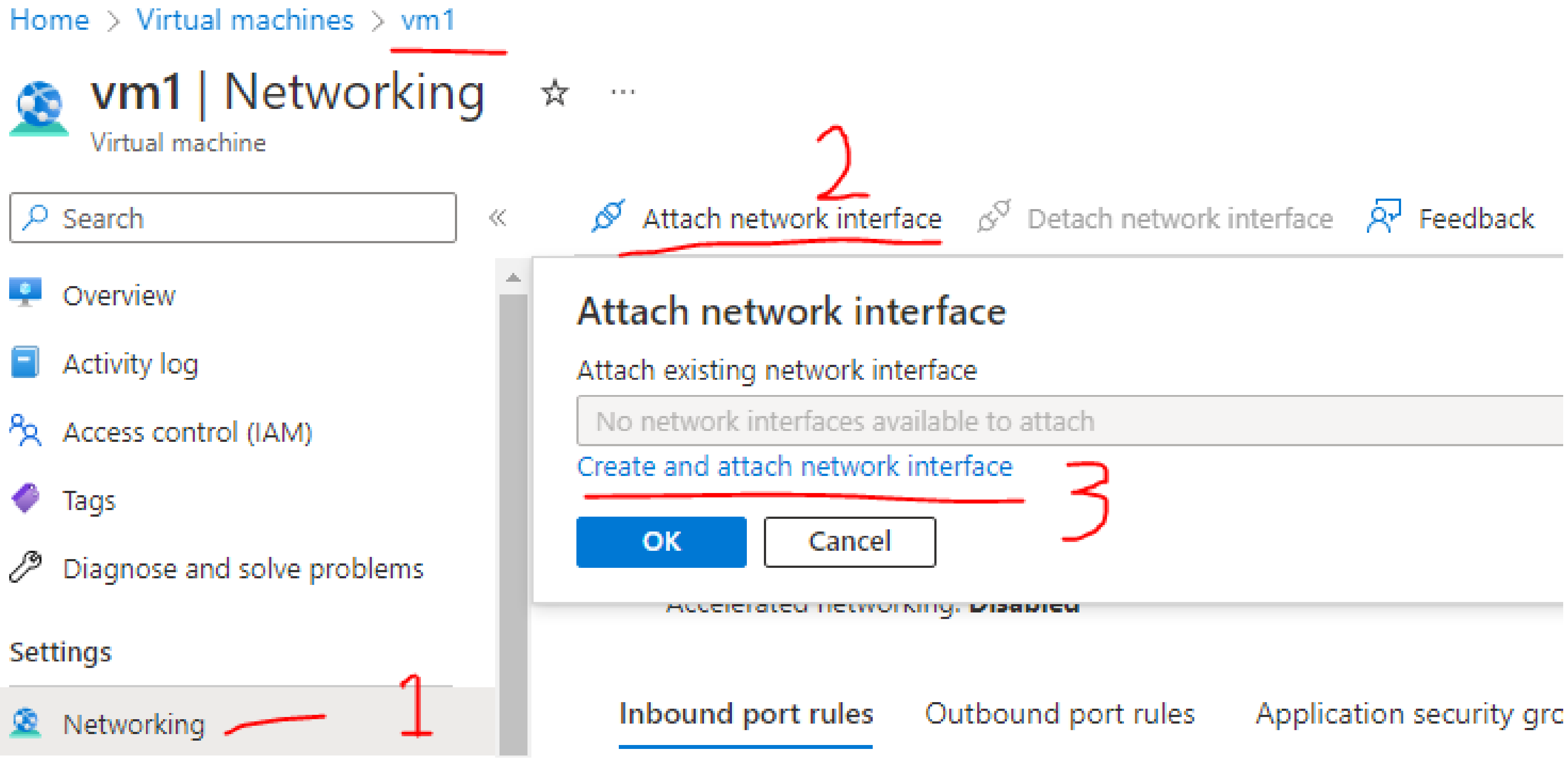
NAT gateway: None

Network security group: None

Route table: None

8. We now have the “Software” part ready. IP addresses/subnet can be used. However, we will need to have a “Physical” network adapter on each VM to use it. Make sure both vm1 and vm2 are Stopped (deallocated).

9. On vm1, "Create and attach network interface"



Home > Virtual machines > vm1

vm1 | Networking

Virtual machine

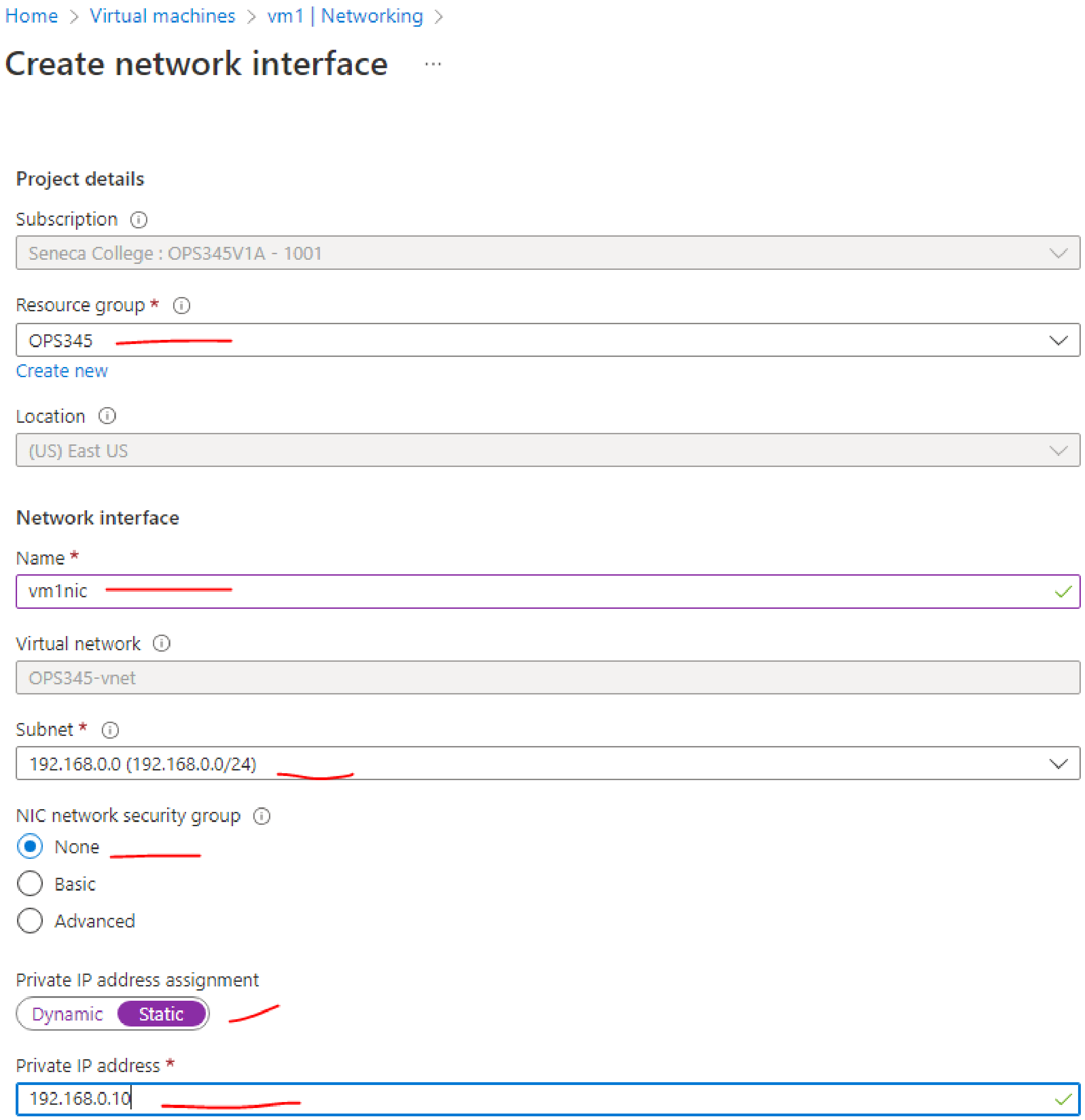
Attach network interface

Attach existing network interface

No network interfaces available to attach

Create and attach network interface

10.Create a network adaptor as “vm1nic”. Make sure “NIC network security group” selected as None. Otherwise, it may conflict with the firewall rules we will build in the future labs. Assign a Static IP. Please assign 192.168.0.10 for **vm1**



Home > Virtual machines > vm1 | Networking >

Create network interface ...

Project details

Subscription: Seneca College : OPS345V1A - 1001

Resource group *: OPS345

Create new

Location: (US) East US

Network interface

Name *: vm1nic

Virtual network: OPS345-vnet

Subnet *: 192.168.0.0 (192.168.0.0/24)

NIC network security group

● None
○ Basic
○ Advanced

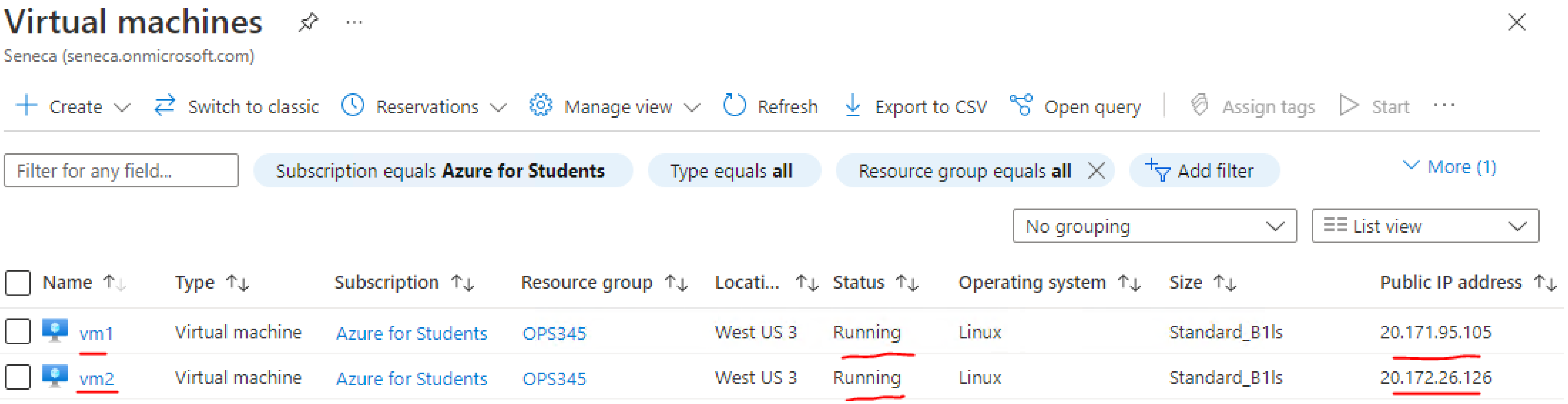
Private IP address assignment: Dynamic [✓ Static]

Private IP address *: 192.168.0.10

11.Repeat the above step to create another as “vm2nic”and attach it to vm2. Assign 192.168.0.20 for **vm2**

Case 2: Basic Network Configuration

1. From Virtual Machines, start both vm1 and vm2 and record the “Public IP address”



Virtual machines

Seneca (seneca.onmicrosoft.com)

Filter for any field... | Subscription equals **Azure for Students** | Type equals **all** | Resource group equals **all** | No grouping | List view

Name	Type	Subscription	Resource group	Locati...	Status	Operating system	Size	Public IP address
vm1	Virtual machine	Azure for Students	OPS345	West US 3	Running	Linux	Standard_B1ls	20.171.95.105
vm2	Virtual machine	Azure for Students	OPS345	West US 3	Running	Linux	Standard_B1ls	20.172.26.126

Confirm network connectivity between two VMs

1. With Bitvise or your preferred SSH client. Login to vm1 and vm2 with the public IP. (Two instances of the SSH client)
   a. Find the 192.168 IPs from vm1 and vm2

      ip address

   b. Ping the 192.168.*.* IP of vm2 from vm1. Confirm it replies.

      ping 192.168.0.5 -c 1

   c. How about ping the 192.168.*.* IP of vm1 from vm2?

Checking Your Current Network Settings

In OPS245, you have used the ifconfig and route commands. In this course we'll use the **ip** command instead, so that you'll be familiar with both sets of commands.

Perform the following steps on the VMs:

1. View the table below comparing older vs newer methods of obtaining network setting information for a Linux machine.

Comparison of Older and Newer Methods of Obtaining Network Settings

Purpose	**Older Method** *(command)*	**Newer Method** *(command)*
Obtain IP ADDRESS and Subnet Mask	**ifconfig**	**ip address**
Obtain Default Gateway	**route -n**	**ip route**
Obtain DNS Server	**nslookup**	**more /etc/resolv.conf**
Obtain Hostname	**uname -n**	**uname -n**
See MAC cache	**arp -n**	**ip neighbour**

2. Note: nslookup is part of DNS utility that not installed yet. You may get command not found error. Install with:
   sudo yum install bind-utils
3. Refer to the man pages or refer to following article to see how to issue the above commands to create a temporary connection to your existing network.

Making Persistent (Permanent) Network Setting Changes

In your OPS245 course, you used a series of commands (ifconfig, route, and nameserver) to setup a temporary network connection. You can use the **ip** command in a similar way to create a temporary network connection. The problem with this network connection method is that those changes will be lost if you restart your Linux machine, although you may want to do that to create a temporary network connection for troubleshooting purposes.

In order to have your network settings become permanent, you need to edit and save the settings changes in a file. For the IP address, subnet mask, default gateway, and DNS server you edit a configuration file in a directory called network-scripts.

Perform the following steps:

1. Change to the /etc/sysconfig/network-scripts/ directory.
2. The name of the file that contains your persistent network settings has the following name format:
   ifcfg-interfacename

3. Which file-name in your network-scripts directory do you think contains your current network settings?

   ```
   more /etc/sysconfig/network-scripts/ifcfg-eth0
   more /etc/sysconfig/network-scripts/ifcfg-eth1          # Not exist as of this step
   ```

4. View the contents of the file to see if it contains the IP address, subnet mask, and default gateway.
5. What is the MAC address of your current VM/NIC?
6. Does this file contain the hostname of your machine? If not, what command can allow you to change your machine's hostname? (hostnamectl set-hostname your-new-hostname) or, edit /etc/hostname

All the Virtual Machines in this course must have **static** 192.168 network configuration (as opposed to Automatic or DHCP). Sometimes, you will be required to debug networking problems quickly by changing the network configuration of your VMs.

7. Edit the ifcfg-interfacename (most likely ifcfg-eth1) file for each of your VMs to use a static IP address.
   Note: Never make any change to your Public Facing network (10.0.0.0/24). You will lose connection to the VMs.
   If the ifcfg-eth1 file doesn't exist, create a new file or make a copy of ifcfg-eth0 file.
   You should at least configure the BOOTPROTO (**static** instead of **dhcp**), IPADDR, PREFIX (or NETMASK). Optional parameters: HWADDR. (Leave GATEWAY, DNS1 out or remove it from this file because you will not connect to Internet with this NIC).
   Note the following information for this setup:
   a. Change the IP address of your vm1nic to static as 192.168.0.10 by modify the configuration file.
   b. Change the IP address of your vm2nic to static as 192.168.0.20 by modify the configuration file.
   c. You can use the sample template. Replace the highlight part with appropriate value.

   ```
   BOOTPROTO=static
   IPADDR=192.168.0.10
   NETMASK=255.255.255.0
   DEVICE=eth1
   ONBOOT=yes
   IPV6INIT=no
   NM_CONTROLLED=no
   ```

   Save your editing session, restart each VM and run the following command to ensure they still have the network configuration you set. Note: Cloud Hosting Platform like Azure may override your settings with default value.

- Try to ping each other VM with the new / hardcoded 192.168.0.0/24 IP addresses.
- ssh (Objective: Test SSH connection from vm1 to vm2 with the 192.168.0.0/24 IP addresses)
    - SSH to vm1 with the Public IP.
        - Create a folder **ops345.vm1** in the home directory
        - SSH to the vm2.

          ```
          ssh ops345@192.168.0.20
          ```
        - Once connected, Create a folder **ops345.made4vm2** in the home directory
    - SSH to vm2 with the Public IP.
        - Verify if the above **ops345.made4vm2** created in the home directory.
        - Think about the logic for the above steps?

8. After setting the network configuration for EACH VM, then either use **ip** commands to bring **down** and bring **up** of the NIC or **reboot** each VM, to verify a working network connection for each VM from boot-up. Be careful, do not bring down the NIC for the 10.0.0.0/24 network, you will lose connection to the VM if you do so. If you did already, restart VM from Azure console to bring the connection back.

   ```
   sudo ip link set eth1 down
   sudo ip link set eth1 up
   ```

If you are uncertain how to perform the above-listed operations, take time to practice them. If everything works and you are comfortable with these operations then you may proceed to the next section.

Linux Network Connection Configuration Troubleshooting

If the network not work in your Virtual Machine, you should perform the following routine steps to troubleshoot the network connection:

1. **Is the network on vm plugged in?** On a physical network you would check whether the cable is plugged in and the link light is “ON” on your network card. In a virtual network environment, you don't have a physical network adapter. Instead, you will need to check the NIC settings in the virtual machine details to view and confirm the appropriate network connection.

2. **Is the network enabled/attached?** This is a problem more common with virtual networks than physical networks. Compare to the physical network card removed from computer, is the NIC attached in Azure?
3. **Do you have an ip address?** Run ip address to check.
4. **Can you ping the host by ip?** (by its public/internal IP address). If not - check all of the above, check if you have an IP address confliction, and check that your subnet mask is correct.
5. **Can you ping 8.8.8.8?** If all of the above work - check that your default gateway is set correctly with ip route and that you can ping the default gateway.
6. **Can you resolve google.ca?** Run nslookup google.ca. If the output doesn't provide an IP address, check that your DNS server is configured correctly and that you can ping that address.

There are several other problems that could prevent your network connection from functioning but the above are the most common problems. Contact your instructor if you still have network issue.

Case 3: Configuring SSH

The default (and often the only way) to administer a Linux server is via SSH. Even if you work in a graphical Linux environment, it is very useful to open a terminal and use SSH to monitor and manage your VMs. Using SSH to connect to remote servers on a network helps to protect your Linux machine from being penetrated. You can also generate a private and public encryption key for the root user, and copy that public key from your host to your VMs in order to allow certain backup programs to run via a scheduling daemon (called cron) without having to be required to enter the password for the remote machine. You will be doing those operations later in this lab.

Managing Services

The SSH server should already be installed and running in your VMs. If not, you can install openssh-server using yum.

```
sudo yum install openssh-server
```

It is essential for CNS/CSN/CTY students to become comfortable managing services since you will need to constantly stop services, change their configuration, and start them for the configuration changes to take effect in nearly every topic this semester, and for other courses involving Linux network management.

Perform the following steps, and become comfortable using them.

```
systemctl list-units --all               # List all loaded services on your system
systemctl start/stop servicename         # Temporarily start or stop a service
systemctl enable/disableservicename      # Permanently change bootup behaviour of a service
systemctl status servicename             # Get current status of a service
```

1. SSH and login to your vm1 Virtual Machine.
2. Use one of the commands above to check the status of your SSH server (i.e. service: sshd).
   ```
   systemctl status sshd
   ```

3. Issue a command to stop the ssh server and run a command to verify that the ssh server is no longer running.
   If you get disconnected from the VMs and unable to reconnect, restart the VMs from Azure Portal.
   Do NOT use Bastion service from Azure, it is a very expensive service that will consume your credit in few days.
   ```
   sudo systemctl stop sshd
   systemctl status sshd
   ```
4. Issue another command to start the SSH server and verify that it is running.
   ```
   sudo systemctl start sshd
   systemctl status sshd
   ```
5. Issue a command to confirm that the ssh service will run upon vm1 restarts (i.e. "enabled").
   ```
   systemctl status sshd
   ```

Configuring the SSH Service

A common way to try to hack into a machine is to try to ssh as root and brute-force root's password. The root user always exists, meaning the attacker doesn't need to try guessing what user names are on your system. If they can get access to root, they can do anything. To prevent this, we will edit the configuration file for the ssh service to prevent root from ssh'ing into your host machine.

(Note: VMs on Cloud Hosting Platform like Azure have the root account disabled already, the following steps only provide protection with locally/on-prem hosted VMs with root account enabled.)

Perform the following steps:

1. Login to vm1 and change the “root” user’s password to a lengthy password.
   ```
   sudo passwd root
   ```
2. Use the more command to display /etc/ssh/sshd_config. This file contains the configuration parameters for the ssh service.
   ```
   sudo more /etc/ssh/sshd_config
   ```
3. Take a few moments to view this file. Lines that begin with # are comments. Either simple explanations of parameters, or parameters that have not been set.

4. Use a text editor to edit the file sudo vi /etc/ssh/sshd_config. This lists all the possible parameters in alphabetical order along with a brief explanation of what each one does. The parameter we are looking for is PermitRootLogin, read its description. By default, it is set to yes, allowing the root user to ssh in to the machine.
5. Uncomment PermitRootLogin, and change the value to no. (Azure has Root Login disabled and this parameter doesn't exist. You can double assure it by add "PermitRootLogin no")
6. Try to use ssh from one of your VMs to log into your host as root. What happened?
7. This is because (for most services) the changes you make to the configuration file will not take effect until the service restarts.
8. Restart the sshd service on your host and try to ssh in again. Now it should prevent you.

   ```
   sudo systemctl stop sshd
   sudo systemctl start sshd
   ```
9. Above step to make sure a root account cannot login with SSH. If it is disabled already, you can permit root login and try (extremely dangers), make sure you disable the root login with SSH when you done.

Note: Configuration files for most services follow a very similar format. Some use an = between the parameter and its value, some require you to group certain parameters together, and most use # to be a comment. You will get lots of experience working with the configuration files for services in this course.

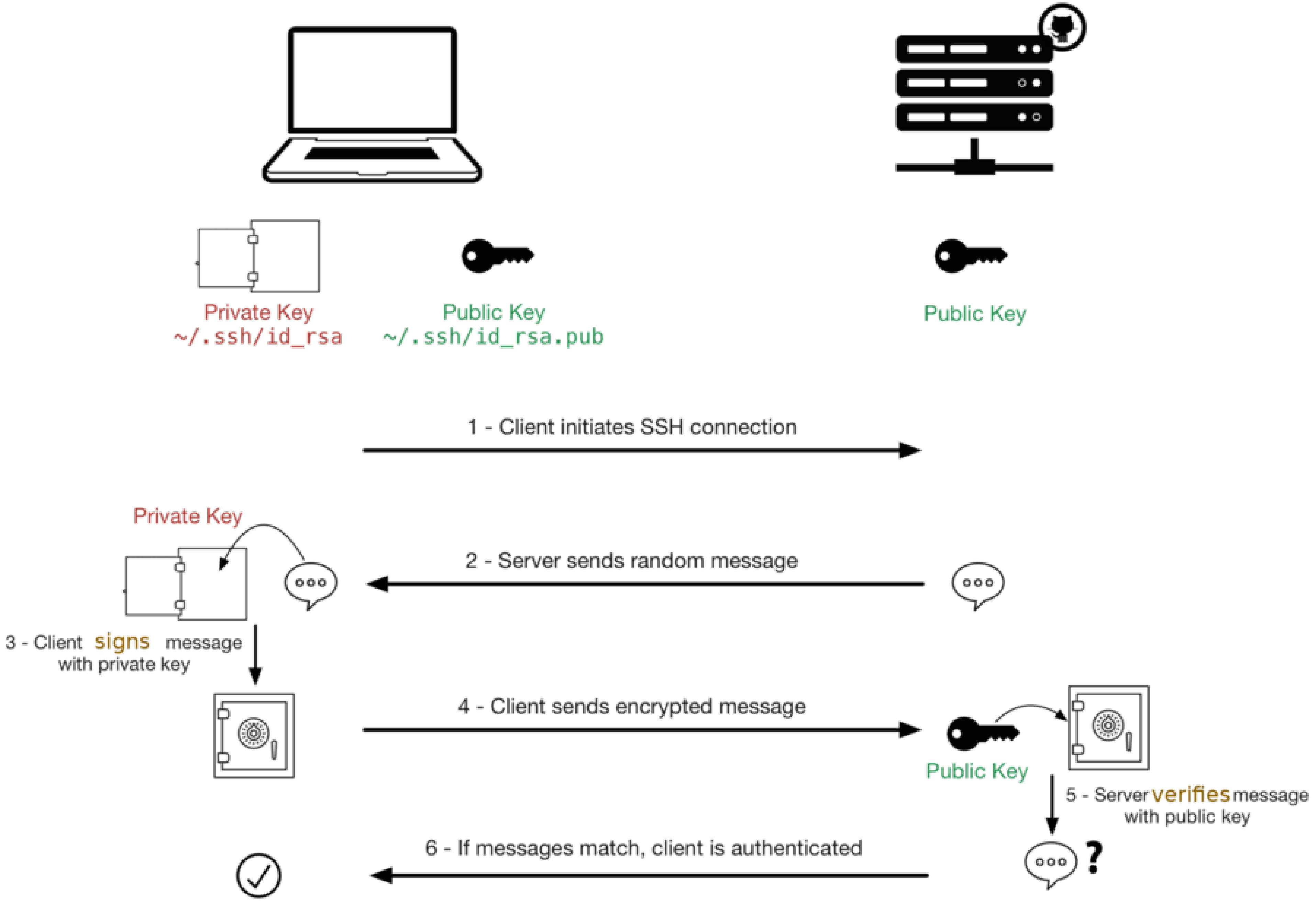
SSH Key Concepts

You should have a basic understanding of ssh and public/private key cryptography to create secure connections between servers after you took the OPS245.

The public key can be "shared" with other server accounts, and can be used in conjunction with your private key in order to help encrypt/de-crypt data.

The diagram below is shared from Sébastien Saunier's blog. It demonstrates how SSH key authentication works. It's not a complete diagram, but it helps see all the parts of ssh key authentication in one place.

A diagram explaining how public / Private keys work. Another term to represent this process is called PKI (Public/Private Key Infrastructure)


Private Key
~/.ssh/id_rsa
Public Key
~/.ssh/id_rsa.pub
Public Key
1 - Client initiates SSH connection
Private Key
2 - Server sends random message
3 - Client signs message with private key
4 - Client sends encrypted message
Public Key
5 - Server verifies message with public key
6 - If messages match, client is authenticated
?

Generating a Public/Private Key Pair & Sharing the Public Key

The public/private key pair needs to be generated on and used on your VMs. The private key is the equivalent of a password (that is why it is considered to be private - only to be used by the owner). That is why the private key is stored in the owner's ~/.ssh/ directory.

One very common mistake that students make is to either generate the key pair for the wrong account, or copy the public key to the wrong account on the intended remote machine.

Perform the following Steps:

1. Make sure login to your **vm1**. (Which will act as a backup destination/server in this lab.)
2. Make sure login to your **vm2**. (Which will act as a backup source/client in this lab.)
3. You will create a key-pair on your **vm2** with no password (i.e. when generating keypair press enter for all prompts including the password). Generate the key-pair by issuing the command:
   ```
   ssh-keygen -t rsa
   ```
4. Note that we only generate one pair of Public/Private key on vm2. The purpose of this lab is to connect from vm2 to vm1 without a password. If vm1 “Trust” the Public key generated by vm2, then whoever has the Private key will also be “Trusted”. The Private key MUST NOT be shared, and to be **Revoked** if suspect leaked.
5. The ssh key authentication provide a way of automated script running without user intervention.

NOTE: When issuing this command, you will end up with the files: ~/.ssh/id_rsa and ~/.ssh/id_rsa.pub (private and public keys). So far, this topic is generally a repeat of OPS245 lab. What you may not know is that by using a "trick" (the magic of public key cryptography), you can SSH to a Linux machine without using a password! Learning to perform this trick is essential in this course and in the industry in general. SSH keys are used everywhere that Linux servers are used.

If you have the private key, you can prove to someone who has your public key that you are indeed the actual owner of that public key. That is how ssh key/PKI authentication works. You are then only required to transfer your public key to a remote server.

You are going to share the public key from the ops345 user in your vm2 to the ops345 user of your vm1.

1. Copy the contents of your ~/.ssh/id_rsa.pub from your host machine and append to ~/.ssh/authorized_keys

   ```
   ssh-copy-id -i ~/.ssh/id_rsa.pub ops345@192.168.0.10
   ```

NOTE: Press ENTER for all prompted information including the password (although this may seem counter-intuitive!).

2. Use the ssh command to test ssh connection from your vm2 to vm1 without having to use a password. This is essential to create backups from VMs to your backup server without being prompted for password.
3. Test if you can connect from vm2 to vm1 without prompt for password.
4. If it is failed to connect or prompt for password, read the error message carefully and fix the issue.
5. DO NOT Proceed to next section when there are still issues!

NOTE: Always remember that these keys are per-user, not per machine. This means that sharing a user's public key will only work for that specific user.

Case 4: Performing & Automating Backups

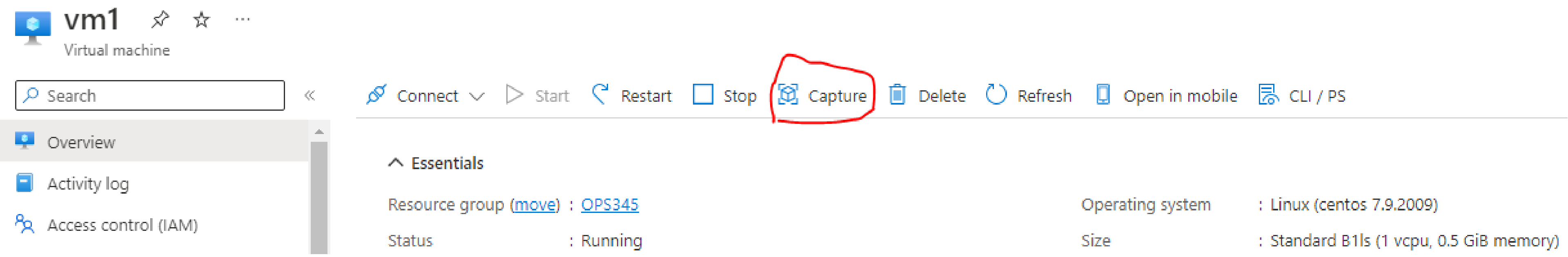
Data backups are considered to be an insurance policy. Running backup can be tedious, but they MUST be performed in an accurate and consistent basis, since loss of data can be expensive (For example: cost of hiring staff to re-enter data).

When performing labs or assignments in this class, if you fail to make backups and something bad occurs and there is loss of data, it only affects you. On the other hand, if you are supporting a client, or working for a company and fail to adequately perform backups and there is loss of data, then other users are affected by failure to backup essential data.

Performing Full Backups

A full backup represents backing up of all of the files of a computer machine (in our case, a VM). A full backup should be performed at the end of each lab or assignment working session. However, we are using the Cloud Platform, that will incur a charge for data storage. Please make sure you remove the backup once you have done the lab.

- Backup VM on Azure is easy, just open the VM page and click the Capture button. No need to perform a real backup.



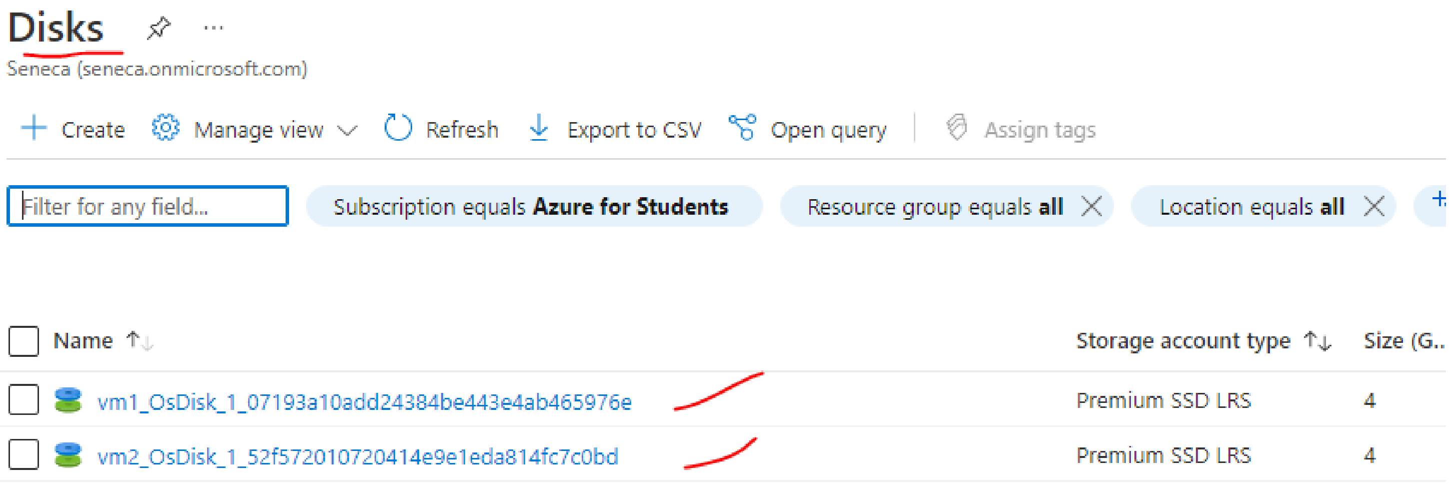
Resource group (move) : OPS345

Status : Running

Operating system : Linux (centos 7.9.2009)

Size : Standard B1ls (1 vcpu, 0.5 GiB memory)

- You can also backup the disk(s) of an image, just search “Disks”. No need to perform a real backup.



Subscription equals **Azure for Students**

Resource group equals **all**

Location equals **all**

□	Name	Storage account type	Size (G..
□	vm1_OsDisk_1_07193a10add24384be443e4ab465976e	Premium SSD LRS	4
□	vm2_OsDisk_1_52f572010720414e9e1eda814fc7c0bd	Premium SSD LRS	4

Performing Incremental Backups with rsync

An incremental backup is a backup of only files that have changed since the last backup. In your case, it may be a good idea to perform incremental backups of your /etc/ directory for your VMs upon startup. We will be using the rsync command to perform incremental backups for **vm2**, again the **vm1** will be the backup destination/server.

rsync is a very versatile backup tool. As the name suggests, rsync is used for synchronizing files typically across a network. It works over the SSH protocol, which is useful in our situation since we are running ssh on our server and VMs. You are going to use your vm2 to backup files to the **vm1**.

Perform the following steps:

1. rsync Needs to be Installed on ALL VMs (not just the source or destination). `sudo yum install rsync`

   The VM template I shared is a minimum install of CentOS 7.9, the rsync command was not installed by default. You need the rsync command to be available on all of your VMs.

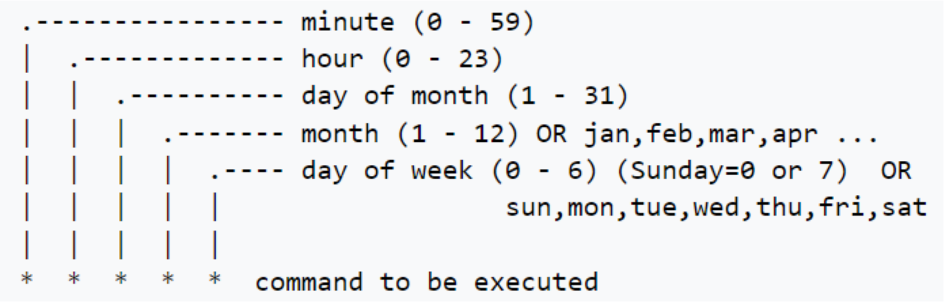
2. Make sure that both vm1 and vm2 are running.
3. On vm1, run the following command: `sudo mkdir -p /backup/incremental/vm2`
4. On vm2, run the following command: `rsync -avz /etc/ ops345@192.168.0.10:/backup/incremental/vm2/`
   You can see the process started, but why nothing copied? (user ops345@vm2 doesn't have access to the folder on vm1)
5. On vm1, run the following command: `sudo chmod -R 777 /backup/incremental/vm2/`
6. Now back to step 4, is backup successful?
7. Run the step 4 rsync command again. Notice that this time nothing is copied over to vm1 since none of the files have changed on your vm2 machine.
8. Create a new file in vm2's /etc/ directory, and rerun rsync. Confirm on your vm1 that only that file that was created on your vm2 actually got backed up to your OPS335.Client.
9. In general, it is a good idea for server to "grab" files from client instead of client "push" to server, because the command will be run as an account from the server, it is easier to manage access. Especially when there is not a Directory Service in this lab (yet). (though both VMs has ops345 account, they are two different accounts). However, to illustrate the ssh key authentication, this section is designed as client **push** to server.

Automating Backups (cron)

Since your vm1 and vm2 are not continuously running, and your VMs are sitting on the Cloud. Scheduled Cloud backup involves extra knowledge that not related to the Linux Administration. You are not required to schedule to perform your FULL BACKUPS periodically (eg. every week at 2:00 AM). Instead, you will use cron to perform scheduled incremental backups.

Cron is a daemon (i.e. a program that runs in the background). The term "Cron" is short for Chronograph which was an old-fashioned term for a stop watch or timer. The role of Cron is to run tasks periodically. It can run tasks for the system (as root) or for a user (including regular users). Every user has a crontab (Cron Table) which is a list of tasks they want to run periodically. You do not edit this file manually: instead, you edit this table using the command crontab -e. Once you run the command, you will get an empty file where you have to insert a line like this:

Perform the following steps:



```
.---------------- minute (0 - 59)
|  .------------- hour (0 - 23)
|  |  .---------- day of month (1 - 31)
|  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
|  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7)  OR
|  |  |  |  |                 sun,mon,tue,wed,thu,fri,sat
|  |  |  |  |
*  *  *  *  *  command to be executed
```

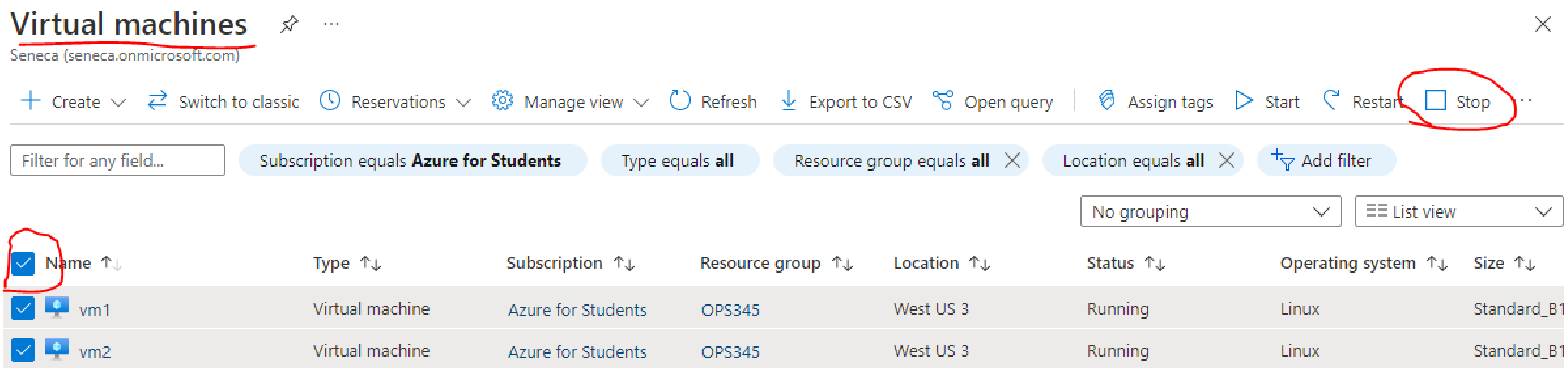
1. Login to vm2, modify the setting so it will run that echo command every minute by creating a crontab (via crontab -e) entry with the following line:

   * * * * * echo "Cron ran this job at: "`date` >> /tmp/cron.log

2. Save and exit your crontab edit session.
3. Wait for one minute to pass, and check the /tmp/cron.log file to see if it was created with the expected contents. (You can also check /var/log/cron file to see what jobs were run).
4. Perform an internet search to see how to configure that crontab entry to run every two minutes instead of every minute.

5. Edit your crontab entry to run same command every two minutes, save and exit, and then confirm by viewing /tmp/cron.log and /var/log/cron files.
6. Edit your crontab to make automatic backups using the rsync command of the /etc directory from vm2 into /backup/incremental/vm2 every hour and overwrite the previous backup.
   0 * * * * rsync -avz /etc/ ops345@192.168.0.10:/backup/incremental/vm2/

Stop and Deallocate your VMs now



Subscription equals **Azure for Students** · Type equals **all** · Resource group equals **all** · Location equals **all**

☑ Name ↑↓	Type ↑↓	Subscription ↑↓	Resource group ↑↓	Location ↑↓	Status ↑↓	Operating system ↑↓	Size ↑↓
☑ vm1	Virtual machine	Azure for Students	OPS345	West US 3	Running	Linux	Standard_B1
☑ vm2	Virtual machine	Azure for Students	OPS345	West US 3	Running	Linux	Standard_B1

Deallocating stops the VM and releases all the compute resources so you are no longer charged for the VM compute resources. However, all persistent disks remain, such as the operating system disk and the attached data disks.

Save the captured file(s) as OPS345_Lab01_yourusername and upload to Blackboard.

If it is video recordings, upload to OneDrive and share with jason.pang@senecacollege.ca