# Solution M1: Core Infrastructure Services

Main goal is to build further on what was demonstrated during the practice

## Tasks (Azure Portal)

One possible solution, using the Azure Portal, may include the following steps:

* Open and login to <https://portal.azure.com>
* Navigate to **Resource Groups** and create a new one, for example **Homework-M1**
* Create a new **Virtual Machine** with the following parameters and steps:
  + Virtual machine name set to **VM1**
  + You may adjust the region to match the one of the resource group
  + For image select the **Ubuntu Server 20.04 LTS – Gen2** option. If not listed, then click on the **Browse all public and private images** link and filter the list to find it easier
  + Set the size to **Standard B1s**
  + This time leave the authentication details as they are (SSH public key)
  + On the **Disks** blade click the **Create and attach a new disk** to add a second hard disk:
    1. Click on the **Change size** link
    2. In the **Custom disk size (GiB)** field in the bottom enter **20**
    3. Click the **OK** button
  + Confirm disk creation with **OK**
  + Finalize the machine creation with **Review + create** button
  + Initiate the creation with the **Create** button
  + When asked to download the key, click on the **Download private key and create resource** button
  + You will be asked to open or save the file. Save it to a known location, you will need it later
* Once the machine deployment is done, open an SSH session to it using your favorite tool
* For example, is we are using a terminal SSH client the connection command will look like:

**ssh -i <path-to-the-key>\VM1\_key.pem azureuser@<public-ip-address>**

* The actual connection command can be seen from the **Connect** option in **Settings** section (in the left menu) of the virtual machine. There you must switch to the **SSH** tab
* You should note that you may receive an error as initially the key file has a wider permission set, which must be reduced (the file should be readable only by your user). Once you are done with the permissions, you may try to connect again
* Once logged on the machine, let us do the hard disk related tasks first:
* We can check what block devices are seen by the OS with

**lsblk**

* The empty 20GB disk will appear like **sdX** where **X** may be any letter in the range **a** to **z**. Let’s assume that in our case it is **sdc**
* Then we can start the partitioning utility with:

**sudo fdisk /dev/sdc**

* Execute the following sequence of actions to create a partition that covers the whole disk:
  + Enter **n** and hit **Enter** to start creating a partition
  + Enter **p** and hit **Enter** to create a primary partition
  + Enter **1** and hit **Enter** to create the first (and only) primary partition
  + Hit **Enter** to confirm the beginning of the partition
  + Hit **Enter** again to confirm the ending of the partition
  + Finally, enter **w** and hit **Enter** to write the changes
* If you execute once more the **lsblk** command, you will see that there is a partition on our disk. There will be an index after the disk name, for example **sdc1**
* Now that we have the partition created, let us create the file system:

**sudo mkfs.ext4 /dev/sdc1**

* In order to use the new file system, we must mount it in an existing empty folder. Let us create one:

**sudo mkdir /data**

* The actual mounting is accomplished with this command:

**sudo mount /dev/sdc1 /mnt**

* If we want the file system to be auto-mounted on boot, we must edit the **/etc/fstab** file

**sudo nano /etc/fstab**

* And add the following line at the end

**/dev/sdc1 /mnt ext4 defaults 0 0**

* To save the changes, we must hit the **Ctrl+O** and then confirm by hitting the **Enter** key
* Press **Ctrl+X** to close the editor
* Before we install any new packages, we must update the list of available packages with:

**sudo apt-get update**

* Then we can install Apache web server with:

**sudo apt-get install apache2**

* When asked if the installation should continue, we must enter **y** and press the **Enter** key
* After the installation is done, under Ubuntu the service is automatically started and configured to start on boot. If we were using another distribution, we must execute the following two commands to accomplish the same:

**sudo systemctl enable apache2**

**sudo systemctl start apache2**

* Anyway, we can check the status of the service with:

**systemctl status apache2**

* Now, let us change the port Apache is listening on
* Open the main configuration file for editing:

**sudo nano /etc/apache2/ports.conf**

* Change the value of the **Listen** directive to **8080**
* Hit **Ctrl+O** and confirm with **Enter** to save the changes
* Close the file with **Ctrl+X**
* Normally, we should adjust other files like the ones that are used to configure the virtual hosts. In our case we can skip this step
* Now, restart the Apache web server with:

**sudo systemctl restart apache2**

* We can easily create new **index.html** file with the following command:

**echo '<h1>SoftUni User</h1>' | sudo tee /var/www/html/index.html**

* Now, we can test if the web page is accessible locally with:

**curl http://localhost:8080**

* Finally, we must return to the Azure Portal, navigate to the virtual machine, and click on **Networking** option under **Settings**
* Then create a new inbound rule by clicking on the **Add inbound port rule** button. Set the **Destination port** to **8080** and the **Protocol** to **TCP**. Click the **Add** button
* Find the public IP address of the virtual machine and test in a new browser window by navigating to the following URL: **http://<public-ip-address>:8080**

## A Reminder

Do not forget to remove the resources created during the homework solution practice. The easiest option to accomplish this would be to delete the resource group