Pouta Exercise set

Exercise 1: Create an SSH key pair for secure login to instance

When you set up a new virtual machine, you are creating a new "cloud instance" with specific:

- Compute resources & hardware (Based on Pouta flavors you choose),
- Operating system (Based on OS image you select)
- Access configurations (Based on Security Groups & SSH Key pairs you create)

The end result is a new vanilla server with desired resources, hardware, OS & access configurations running remotely in CSC's datacenters.

In the first phase of this exercise, you will create your own cloud instance! To start with you need to follow these steps

A. Log in to Cloud Dashboard

- Open a web browser and navigate **https://pouta.csc.fi**
- Log in with training account provided to you.

Since cloud virtual machines are accessible via the internet, it very important and necessary to configure different access and security rules. You will start this exercise by setting a basic set of access rules (that can be reused) to access Pouta virtual machines:

- a Key pair, that you can add to VMs for secure access,
- a Security Group (Set of firewall rules at OpenStack level), to allow access to specific IP addresses.

B. Create your own SSH key pair

- Navigate to **Access & Security > Key Pairs**, click on **Create Key Pair**
  - Name it: *lastname_firstname_key*

An automatic download will start for SSH key pair.

To use your SSH key in **Windows** based machine:

1) You need to first convert the key file to a Windows format. Use **Puttygen** tools to converts your key to *pkk* format (if not installed in your computer, download Putty

tools from: http://putty.tx.se/latest.html or https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)

2) In **Puttygen**, go to **File > Load private key**, and load your *lastname_firstname_key.pem* key (note that you have to select **All Files (*.*)** in the file browse window to see it)

3) Add a **Key passphrase** and click on **Save private key**, save as *lastname_firstname_key.ppk*

To use your SSH key in **Linux** and **Mac OS** based machines:

1) Create .ssh directory in ~ (the users home directory) if it is not there already

```
$ cd ~
$ mkdir -p .ssh
$ chmod 700 .ssh
$ mv keyname.pem .ssh
```

2) ... and move the key into it

```
$ cd .ssh
$ mv ../Downloads/lastname_firstname.pem .
```

3) Make the key file read only and add a password to it (recommended)

```
$ chmod 400 lastname_firstname.pem
```

4) Password protect the key (recommended but not necessary)

```
$ ssh-keygen -p -f keyname.pem
```

C. Create your cloud machine

To create a new virtual machine, you will use an existing CentOS-7 OS image and the access configurations you just created.

Navigate to the **Instances** section and click **Launch Instance**

- **Name** it: *lastname_firstname_vm*
- Select **Flavor** as *standard.tiny*
- Select Instance Boot Source as "*Boot from image*"
- Find the image named **CentOS-7** and select it.
- Navigate to **Access and Security** in the same pop-up
- Select **Key Pair** you just created.
- Select predefined security group **SSH-World** in same pop-up.
- You can leave the rest as defaults and click on **Launch Instance**

Your instance should be visible in the **Instances** tab, wait until it has started. You can check its details by clicking on the name of your instance.

D. Associate Floating IP to your cloud machine

To connect your new instance you need to **assign it a public IP address**:

Go to the **Instances** page

- Find your VM's name and from the dropdown on the right, select **Associate Floating IP**.
- **Select an IP** from the drop down (if there are no available IPs, click on the **"+"** sign)
  - You can see the IP you assigned to your VM in the **Instances** page, next to the name of your virtual machine

To connect this instance, you will use the **public-IP** address (floating IP) you just assigned

E. SSH into your cloud machine

The CSC OS images have a sudo user by default: **cloud-user**. This user has no password by default, so the only way to connect virtual machines running with CSC OS images is via SSH using cloud-user.

To connect to your VM from **Windows** based machine, use **Putty**:

- Open **Putty** and add the **public-ip** you assigned to your VM as the **Host Name (or IP address)**.
- Go to **Connection > SSH > Auth** then add it in **Private key file for authentication** and add the key pair file in ppk format (*lastname_firstname_key.ppk*) under
- You can also connect to your instance with **WinSCP** for transferring files

To connect to your VM from **Linux** based machines, use these commands to add your key to your keys archive:

- $ **ssh-agent /bin/bash**
  $ **ssh-add lastname_firstname_key.pem**
  $ **ssh -A cloud-user@public-ip**

F. Celebrate the success!

Now that you have successfully created your own cloud machine, you can start playing with it: install your favorite software package, create some files, run some linux commands etc..

Finally, install telnet and enjoy Star Wars show!

- $ **sudo yum install telnet**
  $ **telnet towel.blinkenlights.nl**

Exercise 2: Install Docker CE & run RStudio server in Docker Container

In this exercise, you will install Docker CE to your Pouta VM & then run RStudio server as a docker container in your VM.

A. Install Docker CE on your VM

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
sudo yum install docker-ce
```

B. Start Docker CE on your VM

```
sudo systemctl start docker
```

C. Test your Docker CE installation

Test if Docker CE is successfully started on your VM, run this Hello world test container, you should see Hello World from docker if Docker CE is running.

```
sudo docker run hello-world
```

D. Run your RStudio Container

Run Rstudio docker container, replace <password> with a password of your choice

```
sudo docker run -d -p 8787:8787 -e PASSWORD=<password> --name rstudio rocker/rstudio
```

REMEMBER to open port 8787 for VM using Security Groups. You can use pre created security group rule ***Allow-8787*** for opening port 8787 or create a new security group of your own and allow TCP ingress traffic for port 8787. You can access RStudio server using http://Your_VM-IP:8787 Login for RStudio server rstudio/your password