cPouta exercises

8.2. 2018

Table of Contents

1. Launching a virtual machine

1.1 Login to cPouta interface in

https://pouta.csc.fi

Check what is the current project number you are using and how much resources your project has.
Project > Compute >Overview

1.2 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.

Depending on the operating system of your local computer, continue using either A (mac and linux) or B (instructions).

A. Using SSH key in Linux (local linux or Taito!) and Mac OS X:

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

```
mkdir -p .ssh
chmod 700 .ssh
```

... and move the key into it

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

2. Make the key file read-write only and add a password to it (recommended)

```
chmod 600 lastname_firstname.pem
```

… and add a password to it (recommended)

```
ssh-keygen -p -f lastname_firstname.pem
```

B. Usingh SSH key in Windows:

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: 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

1.3 Launch a Virtual Machinine in cPouta

To create a new virtual image, you will use an existing Ubuntu 16 image and the access settings 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 the Default security group 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.

1.4 Associate Floating IP to your cloud machine

To be able to connect to 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 drop-down 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 to the instance, you will use the public-IP address (floating IP) you just assigned.

1.5 Create your own Security Group

Security Groups are nothing but OpenStack level firewall rules. In this exercise, you will create and take in use a security group that will allow ssh connections from your machine (IP address) to the virtual machine.

- To find your computer's IP address you can visit to http://v4.ident.me/.
- In the cPouta interface, navigate to **Access and Security** -> **Security Groups** and click on **Create Security Group.** Name the new security group as your ***lastname_firstname_ssh.***
- Click on **Manage Rules** of your Security Group.
- Define the **port** to be **22**.
- Leave **Remote** as **CIDR**.

- In the **CIDR** field, you should change the default value (0.0.0.0/0) to the IP address of your local server (from https://v4.ident.me). That way connections to your VM are allowed only from your local computer.
- Remember to add /32 IP mask after your IP address.
- Move back to the *Instances* view. In the "*Actions*" column of your Virtual Machine, that by default has value "*Create snapshot*" , select action "**Edit Security Groups**". In this tool, use the + button to add the security group you just created to your virtual machine.

1.6 Building your own Rstudio Server

Now that you have successfully created your own Virtual Machine in cPouta machine you can now log in ssh or Putty. Note that the default user account name is always: **cloud-user**

In Linux and Mac / ssh

In terminal, give command:

ssh -i $HOME/.ssh/*lastname_firstname*.pem cloud-user@*IP-of-your-machine*

Windows /Putty

Open PuTTY and use the IP address of you Virtual Machine as the name of the computer to be connected.

When connecting, the usage of the private key is defined in Putty's Configuration menu under **Connection | SSH | Auth**
Private key file for authentication.
Use the **Browse...** button to select the *lastname_firstname*_key.ppk file you previousky created.

As a sample case for adding software to a your virtual machine we install a web server hosting Rstudio server to your VM.

Download Rstudio rpm installation package:

```
wget https://download2.rstudio.org/rstudio-server-rhel-1.1.383-x86_64.rpm
```

Then install rstudio

```
sudo yum install --nogpgcheck rstudio-server-rhel-1.1.383-x86_64.rpm
sudo yum install R
sudo rstudio-server verify-installation
```

To use Rsudio server you need to add a new user account, that has password, to your system:

```
sudo useradd Ruser
sudo passwd Ruser
```

The Rstudio server WWW interface uses by dafeult port: 8787

To be able to access the server with browser running in your local machine, you must add a security group rule that allows connections to port 8787 from you local machine.
Afrer that open browser connection to your Rstudio server:

```
http://ip-address-of-your-vm:8787/
```

and log in with the user account you just created

2 Creating and using volumes

If you are working with datasets that are larger than just some tens of gigabytes you should attach a volume to your Virtual Machine. In this exercise, we will learn how to create add additional volume to instance. Volumes are like your virtual external hard disks, you can save data in them and reattach to any other instance of your choice.

2.1 Create your volume

- Navigate to Volume section.
- Click on **create volume**, name volume as your ***lastname_firstname_*vol** and size as 10GiB

2.2 Attach your volume to instance

- After creating volume, open the pop up of your volume (in the *Actions* column)
- Use **Manage attachments** to attach your volume to your Virtual Machine

  (choose your virtual machine from Attach to instance list)

Next steps are done inside your virtual machine using the terminal connection

2.3 Mount your volume

Connect to your cloud virtual machine. Check disk partition using *fdisk* command and find your volume

```
sudo fdisk -l
```

Create file system of your choice on your volume, *xfs* or *ext4* give good performance in cPouta. Remember to enter correct path for volume based on output from *fdisk* command you executed above. In the example command below the end of the path may be different than **vdc1 (e.g. vdb or vdb1). Note that creating files system removes all old data from the volume. Thus you should skip this command if you are re-attaching an old volume.**

```
sudo mkfs.xfs /dev/vdc1
```

Mount your volume to instance's file system:

```
sudo mkdir /media/volume
sudo mount /dev/vdc1 /media/volume
```

Once you have mounted your volume you can start using in it. If you now move to your volume and check that available disk space, you should see that you have now 10GB or free space in your volume.

```
cd /media/volume
```

```
df -h ./
```

However if you try to add some data to the volume, you will get "permission denied" as the volume directory is owned by the root user (sudo). With command *chown*, we change the directory so that the owner is "cloud-user"

```
sudo chown cloud-user /media/volume
```

Now, locate some data set from internet and download it to your volume with *wget* command:

```
wget url-of-a-dataset
```

In case you want to detach volume, you should un-mount it first, you can do so with command

```
sudo umount /dev/vdc1
```

2.4 Mounting Taito $WRKDIR to your Virtual Machine

The $HOME and $WRKDIR directories of Taito can be remotely mounted to your virtual machine with **sshfs** tool. As sshfs is not installed in the Linux images of cPouta you have to first install it.

In Ubuntu this is done with command:

```
sudo apt install sshfs
```

In Centos installation is done with command:

```
sudo yum install sshfs
```

Once sshfs installation is done, create a new directory (e.g. taito_wrk) to be used as a mount point.

```
mkdir taito_wrk
ls -l taito_wrk
```

Then do the remote mounting with command (in case of training accounts, replace XX with your account number. If you have a personal account in Taito, you can use that instead of the training account).

```
sshfs trngXX@taito.csc.fi:/wrk/trngXX taito_wrk
```

Now check the content of the mounted directory

```
ls -l taito_wrk
```

Once you stop using the remotely mounted directory, un-mount it with command:

```
fusermount -u taito_wrk
```

3 Create Snapshot and re-use it

In order to back up a VM state (or to save billing units when a VM is not used) you can create a snapshot from it, so that you can continue later with the same machine. A snapshot can be thought of a version of your virtual machine that can be launch later on as an instance (this instance will be the same as that you have in the moment of taking the snapshot). The snapshots are stored in Pouta as **Images**

3.1 Create a snapshot of your machine

In the Pouta web interface:

- Shut down your instance: **Compute > Instances >** instance_name **> Shut Off Instance**
- Then, **Compute > Instances >** instance_name **> Actions: Create Snapshot**
- Name it: *lastname_firstname_vm_date*
- This creates a new Image in **Compute > Image**

Review that the snapshot was created properly

- Go to **Compute > Images**
- Click on the **name of your image (snapshot)** to see its details

Delete your machine

- Go to **Instances**
- Click on your instance and delete it.

3.2 Relaunch new instance with same state

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 Snapshot"*
- Find your Snapshot and select it.
- Navigate to **Access and Security** in the same pop-up
- Select **Key Pair** you created.
- Select Security Group you created in same pop-up.
- You can leave the rest as defaults and click on **Launch Instance**

This will launch new cloud machine which is in same state as of instance you deleted.

Attach the Volume previously used to your VM. This is applied with the same commands that were used in exercise 2 with the exceptions that :

1. You don't need to create a new volume
2. You don't need to create directory /media/volume as it already exists
2. You don't need to create the file system (sudo mkfs.xfs /dev/vdc1) before mounting the

volume. Running the filesystem creation command would erase all the data from the volume

After re-attaching the volume, check that you can you see the data that was stored to the volume?