A1 - Introduction to SNIC Science Cloud

Introduction:

The aim of this assignment is to

- Give you hands-on experience with the cloud computing infrastructure used in this course
- Give you some experience with working in the Linux command prompt on the Virtual Machines (VMs)
- Introduce you to Git

We will use SNIC Science Cloud (SSC). SSC is a national scale community cloud that provides Infrastructure-as-a-Service (IaaS). It is built using OpenStack cloud software and Ceph storage and offers the following key cloud services we will use in the course:

1. Instance management (virtual machines)
2. Storage management
3. Identity management
4. Image management
5. Network management

Important links:

1. Information page: https://cloud.snic.se
2. OpenStack user guide: https://docs.openstack.org/train/user/
3. OpenStack dashboard: https://east-1.cloud.snic.se/

Your goal in this assignment is to complete the tasks below. There are instructions for the tasks, but they are **not comprehensive**. You will **also need to** use the OpenStack user-guide, as well as external resources, *depending on your previous experience*. **Using the Internet to find instructions and troubleshoot issues is also part of the objectives of this assignment.** Remember that you can also ask (and answer) questions on the course Slack.

Rules for cloud resources in the course:

1. Always name your instances something containing your first and last name. This way, we teachers can manage resources more easily.
2. Always change the name of volumes backing your instances from the uuid to a name containing your name (same reason as above).
3. You are not allowed to create new networks or modify the existing networks.
4. Do not change any settings in the “default” security group.
5. Select "yes" for "Delete Volume on Instance Delete" when launching instances.
6. Please remember to clean up resources that you no longer use, since we are paying for all deployed resources, even if you are not actively using them. **Terminate all your running instances, delete the volume, and delete the snapshot.**

Task 0 - Getting started with Git

Git is an open-source version control system used to keep track of changes to code. Together with web-based hosting services such as GitHub or BitBucket, it can also be used for remote access control and various collaboration features.

For all coding tasks of this course, as well as the project, you are required to use Git.
This allows you to:

a) Have your code protected in case cloud instances fail.

b) Work on code on your private computer (and therefore use any code editor you like) and subsequently sync the code on the cloud.

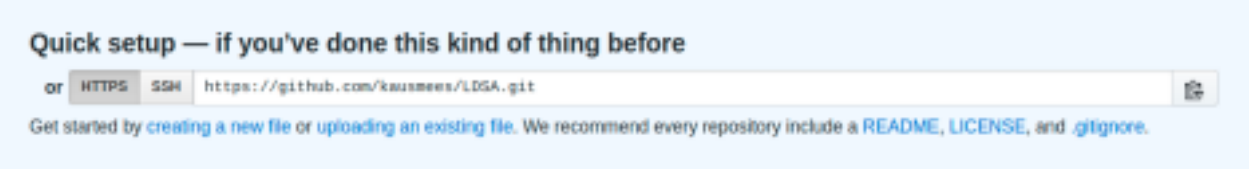
For further learning we recommend the book: https://git-scm.com/book/en/v2 and the documentation of Git: https://git-scm.com/docs

The Git commands we provide you with below work if you are using a Linux system. If you use another system, you might have to modify them. If you have a Windows computer, we recommend using a linux-like terminal such as Windows Subsystem for Linux (WSL), Git, PuTTY or whatever you prefer.

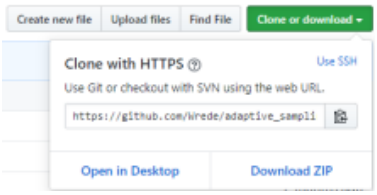
The learning curve for Git is steep, but it's extensively used both in industry and academia.

Set up GitHub or Bitbucket:

1. Create a GitHub or a BitBucket account (https://github.com/ or https://bitbucket.org/product/). Both offer private repositories, but GitHub also offers student accounts**. You do not need to create a new account for the course if you have one already.**
2. Login to your account and create a new **private** repository (name it something suitable, e.g. DataEngineering). **For security reasons:** make sure that the repository you create for the course is private.
3. For authentication you need to setup SSH key-based access (follow these instructions: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/about-ssh).
4. Copy the SSH URL of your new repository. In GitHub this can be done as the figures below demonstrate



or



5. Working on your own machine, clone the repository onto a local computer (your private one or a lab computer). Unix-like systems have Git installed by default (including WSL). For Windows you have to install it.

```
git clone <repository-url>
```

A folder (the repository) with the name DataEngineering now exists. If you look inside it, there is a hidden folder named .git

Note: if you are using WSL do NOT alter or edit files that lie inside the WSL filesystem using a Windows editor (might cause corrupted files). However you can modify files in the Windows file system using WSL. Therefore, we recommend you to Git clone inside the Windows file system (/mnt/c/) and then use Windows editors for your coding.

6. Add a file and commit it to the repository: Create a new folder A1 inside your DataEngineering repository and create a file inside it called test.txt.
(DataEngineering/A1/test.txt). In your repository folder DataEngineering, execute the command:

```
git status
```

Notice the new folder and the file under "untracked files". To add it to the "staging area", run the command:

```
git add A1/test.txt
```

Changes to files that have been staged can be saved to the repository by committing:

```
git commit -m "My commit message"
```

(The message should describe the changes made, e.g. "Added a file called test.txt".)

To see the commit history run the command:

```
git log
```

press “q” to quit.

7. To sync your changes back to the GitHub or BitBucket repository run the command:

```
git push
```

Type in your account username and credentials. Go to your GitHub/BitBucket account and notice that the new folder and file is in your repository.

In task 2.1, you will launch a virtual machine on the cloud and sync this repository to it.

Part 1 - Instance and volume management

First of all, read the security guidelines: https://cloud.snic.se/index.php/user-security-guidelines/
Log into the cloud here: https://east-1.cloud.snic.se/

Task 1.1: Create a new SSH-keypair

The only method allowed to access the cloud instances is via ssh-keypairs. Username/Password is disabled by default on all cloud instances (according to best practice) and should never be enabled for security reasons. If you are not familiar with the use of ssh-keys, here is a simple explanation of how it works:
http://blakesmith.me/2010/02/08/understanding-public-key-private-key-concepts.html

The OpenStack software helps you create/import keys, and will make sure that your public keys are injected in the instances you create. The private key should be private and is for you to safekeep on your clients. In the OpenStack dashboard:

1. Create a new SSH-keypair (**Compute -> KeyPairs**)

2. Save the downloaded .pem file in a secure location on your computer (and remember where you store it). You will use this key throughout the course, and you **don’t** need to create a new ssh-keypair each time you create a new virtual machine.

Task 1.2: Provisioning a Virtual Machine

In this task you will "launch" an instance of a VM by booting from an existing image(which has an installed operating system). In the OpenStack dashboard, go to **Instances -> Launch Instance**.

In the Launch Instance menu, select the following settings (for settings not specified below, leave the default values):

1. Follow the “Rules for provisioning cloud resources in the DataEngineering course” at the beginning of this document when selecting a name.
2. Select the most recent “Ubuntu 22.04” image as the boot source.
3. Choose the “Yes” for "Create New Volume"
4. Choose "no" for "Delete Volume on Instance Delete" (this is **for this lab only**, always choose “yes” for this option when creating VMs in this course in the future, as specified in the rules at the start of the document).
5. Select the flavor (size) that has 1 VCPUs and 1GiB RAM.
6. Make sure the “default” security group is selected for the instance.
7. Choose to inject the keypair you created in Task 1.1.

You have now created a VM. To be able to access (“login”) to this VM you need to:

1. In the OpenStack dashboard, in the Instances tab, locate your VM. 2. In the Actions column choose “Associate Floating IP” to your instance.

2. Select an IP address from the list. Note: if there are no IP addresses available in the list press the “+” sign to allocate a new IP from the pool.

3. Access the instance from your laptop / lab computer using SSH. a. To use SSH on a Linux system:

```
ssh -i /path/to/yourkey.pem ubuntu@yourfloatingIP
```

4. To use SSH on a Windows system, you can either use an SSH client such as PuTTy, or use WSL, in which case you can use the same command as above.

Hint: you may have to configure the file permissions of the .pem file to be able to use it for SSH. (chmod 400)

You are now logged in to the VM. Perform the following actions on the VM (remember that you are using a machine with the Linux operating system, running the Ubuntu distribution, in case you have to search the Internet for how to do these steps).

1. Install the package “cowsay”, use the `apt` tool
2. Run cowsay:
   `cowsay hej`
3. Create a text file in the home directory of the instance.

Go back to the OpenStack dashboard in your browser and complete the following steps.

1. In the “Volumes” menu, locate the volume that has been created to backup your instance and change its name to an identifier containing your name.
2. In the “Instances” menu, delete your instance.
3. Create a new instance. For the boot source, select the volume that was created for your previous instance, that you renamed in step 1. Select “no” for “Delete volume on instance delete”.
4. Access the new instance. Is the file you created in the home directory still there?
5. Delete the instance.
6. Create a snapshot of the volume.
7. Boot a new instance from the volume snapshot. Access the instance. Is the "cowsay" program still installed?
8. With a basic understanding of instance provisioning, please review the SSC user security guidelines: https://cloud.snic.se/index.php/user-security-guidelines/

Task 1.3 : Block Storage

In this task we will not give you as much instructions as the previous tasks. The purpose is for you to find external sources of information and to navigate the OpenStack dashboard yourself.

1. Create a volume of size 1GB.
2. Attach your newly created volume to your instance.
3. Access the volume from your instance and copy a file to the attached volume. *Hint: you will need to format and mount the volume, here is a good tutorial:*
https://github.com/naturalis/openstack-docs/wiki/Howto:-Creating-and-using-Volumes-on-a-Linux-instance

Part 2 - Using the virtual machine

Task 2.1 : Syncing Git
On your virtual machine, clone the repository you created in Task 0. You should be able to see your file DataEngineering/A1/test.txt

To update the local repository with the latest commits that have been pushed to the remote repository, run the command:

```
git pull
```

Important: The VMs are not always reliable, and if they unexpectedly fail, you may lose files that have been stored on them. If you create files that you do not want to lose on your VM, make sure to add them to your git repository and push changes frequently to back up the files.

Task 2.2 : Running a jupyter notebook

On your virtual machine, install pip for python3 (“python3-pip”), and the Python3 packages “pandas”, “notebook”, “matplotlib” and version 3.0 of “jinja2”. Your VM should already have Python3 installed.

Launch a notebook server, and access it via the web-browser:
You will need to create a new security group and associate it with your VM. Do not modify the default security group. Also check the security group to make sure the port for jupyter is accessible.

2. When you have the Jupyter notebook server running on your VM, create a new Python3 notebook by locating the folder DataEngineering/A1 via the file tree in the browser and selecting new->Python 3. Rename it.

3. Run the program cowsay via the notebook by writing

```
! cowsay yello
```

in the first cell and press the “Run” button. (What is the effect of the “!” ?)

4. Run the following Python code in the jupyter notebook:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.DataFrame(np.random.rand(10, 4), columns=['a', 'b', 'c', 'd'])

df.plot.area()
```

A plot should appear in the notebook.

5. Take a screenshot showing the entire browser window (including the address fields so that the IP is shown).

6. On your VM, there should now be a file DataEngineering/A1/yourNotebookName.ipynb.
If you created it in another directory, move it to DataEngineering/A1. Push it to the repository.

Submission Instructions

7. On the website of GitHub or BitBucket, open the folder A1 in the DataEngineering repository. It should contain two files: test.txt and yourNotebookName.ipynb. Take a screenshot of the browser window, showing your github username.

8. Upload the two screenshots you have taken to Studium for Assignment A1.

9. Delete all instances, volumes and volume snapshots you created during this lab.

During the rest of the DataEngineering course, you can refer back to this document for instructions on how to provision resources in the cloud.

Optional Tasks

- Instead of opening a port for accessing the Jupyter server, use SSH port forwarding.