Cloud Computing (INGI2145) - Lab Session 1

Muhammad Bilal and Marco Canini

1. Background

In this lab session, you will learn how to create Amazon Elastic Cloud Computing (EC2) instance, how to connect to them, and how to manage them using the Amazon Web Services (AWS) Command Line Interface (CLI).

2. Creating and Managing Amazon EC2 Instances

In order to manage EC2 instances from command line, you will need to have the AWS CLI installed (http://aws.amazon.com/cli/). You can install AWS CLI tools using `pip install awscli`, which in turns requires Python and pip (a python package manager) to be installed.

- Log in on the AWS console: https://mcanini.signin.aws.amazon.com/console/
  (use this exact URL or it won't work)
  Account: `mcanini`
  User Name: `ingi2145-<AWS username>`
  `AWS username` is the username you provided during the course sign up
  Password: supplied to you during this session

- As shown in the demo, go to Identity and Access Management (IAM) and create/download a new access key for yourself.

- As shown in the demo, setup the Multi-factor Authentication (MFA) for your account.

- Run `aws configure` and enter your key information, the default region (`us-west-2`) and the output format (`table`). Verify everything is working by running `aws ec2 describe-instances`
  The output should be empty or contain your friends' instances, if you're slow.

- Launch an EC2 instance via the console as shown in the demo.
  Things to keep in mind:
  - Use Amazon Linux AMI.
  - Create a micro instance.
  - Use a magnetic disk.
  - Tag your instance (`key="user" value = <AWS username>`).

    - Use the SSH Only INGI2145 security group.
    - After clicking on “Launch”, create and download a new key pair named after your user name (`ingi2145-<AWS username>`).
    - You will need to change the permission of the key pair:
      `chmod 400 ingi2145-<AWS username>`

- Look up your instance’s public IP, then connect to it via:
  `ssh -i <your_key>.pem ec2-user@<ip>`
  e.g., `ssh -i ingi2145-xyz.pem ec2-user@54.69.32.120`
  Then exit (`exit`).

- Create a new EBS volume, either from the AWS console (as shown in the demo), or via the CLI (if you’re a real macho (wo)man).
    - Hint: `aws ec2 create-volume help`
    - Hint: use the `--dry-run` option to verify your command.
    - Set the size to 1GB, and the storage type to magnetic.
    - Your volume must be created in the same availability zone as your instance.

- Attach the volume to the instance:
  `aws ec2 attach-volume --device /dev/sdf`
  `--instance-id <iid> --volume-id <vid>`
  e.g., `aws ec2 attach-volume --device /dev/sdf`
  `--instance-id i-705b4c7b --volume-id vol-efa7eaee`
  The device name matters!
  (http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html)

- Re-connect to VM via SSH, then format the volume and mount it:
  ```
  sudo mkfs -t ext4 /dev/xvdf
  sudo mkdir /mnt/sdf
  sudo mount /dev/xvdf /mnt/sdf
  sudo chmod ubuntu:ubuntu /mnt/sdf
  ```

- Make sure you can read and write the volume via `/mnt/sdf`. Then unmount it and exit the VM.
  ```
  sudo umount /mnt/sdf
  exit
  ```

- Detach the volume from the instance:
  `aws ec2 detach-volume --volume-id <vid>`

  At this point, the volume still exists (as you can tell from the console or from `aws ec2 describe-volumes`). We won’t do it, but we could reattach it to the same

instance or to another and read what was written on it.

- Instead, delete the volume and terminate your instance (it shouldn't be too hard to figure for a brilliant bunch like you). Take care not to delete your friends' volumes/instances (or they won't stay your friends for long).

Appendix A: Installation commands

Install Python:
`apt-get install python (ubuntu)` or `brew install python (mac)`
Install pip:
`apt-get install python-pip (ubuntu)` or `sudo easy_install pip (mac)`
though if you have install python using homebrew on Mac, you will have pip installed with it by default.

Appendix B: Note on EC2 SSH Login

Depending on the type of AMI that you chose, the default username might be different
For example, Ubuntu images would have `ubuntu`, while Amazon linux images would have `ec2-user` as the default user account.