2019

Designing Computational Biology Workflows with Perl - Part 1

Esma Yildirim
CUNY Queensborough Community College

Title: Designing Computational Biology Workflows with Perl – Part1

Author/Affiliation: Esma Yildirim / Queensborough Community College

Date: 05/15/2019

Material Type: Lab

CS + Computational Biology

Software/Equipment Dependencies:
An Amazon Web Services (AWS) account, a web browser and a command line interpreter program (e.g. Putty on Windows, Terminal on Linux/MacOSX)

Prior Knowledge Needed (if any): None

Keywords: Linux file system, Perl, gene-sequencing file formats

Approximate time needed: 1 hour

Description: This material introduces the AWS console interface, describes how to create an instance on AWS with the VMI provided, connect to that machine instance using the SSH protocol. Once connected, it requires the students to write a script to enter the data folder, which includes gene-sequencing input files and print the first five line of each file remotely. The same exercise can be applied if the VMI is installed on a local machine using virtualization software (e.g. Oracle VirtualBox). In this case, the Terminal program of the VMI can be used to do the exercise.

Designing Computational Biology Workflows with Perl – Part I

In this lab, you will learn how to create a virtual machine instance on AWS cloud and write a Perl script to remotely display the contents of gene-sequencing input files.

1. What is a Virtual Machine Image(VMI)?

A virtual machine image is the software representation of a machine with its operating system, hardware settings, installed programs and all the data in its file system. It can be configured on any matching hardware settings that might be running on any type of operating system through the use of *virtualization software* (e.g. Oracle VirtualBox).

Example: An Ubuntu Linux Virtual Machine with 10GB hard disk, 4 GB memory settings can be launched on top of a computer that runs Windows operating system and has 250GB of disk, 8GB of memory and a 4-core CPU.

The Virtual Machine Image provided as part of this course is a Ubuntu Linux machine prepared via Oracle VirtualBox Version 5.2.20 r125813 (Qt5.6.2) virtualization software that ran on a Windows operating system.

2. Why do we need a VMI?

The open source software packages used in gene sequencing analysis usually run on top of a UNIX/Linux based system which is not a common operating system type used in a Lab setting. However, they can easily be installed on a UNIX/Linux machine and exported as a VMI to be launched in a Cloud Computing environment through the use of a web browser and a command line interpreter program.

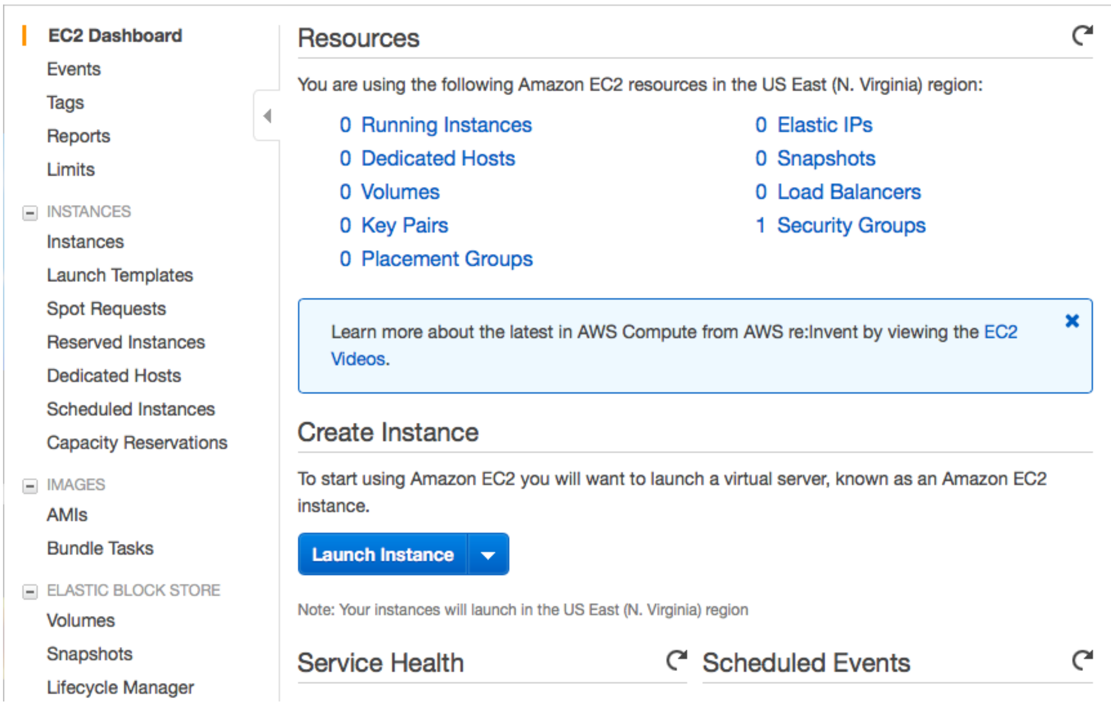
Also, the installation process of gene sequencing software packages is a cumbersome process and they have a lot of dependencies. The VMI comes with all the necessary software pre-installed along with the input data needed to do the lab exercises. After the instructor converts the VMI to an AWS compatible AMI (Amazon Machine Image), shares that AMI through his/her AWS account. You can then launch an instance on AWS cloud using the converted AMI in a few seconds and connect to the machine to do the lab exercises.

Alternatively, use virtualization software like Oracle VirtualBox to import the VMI provided by your instructor and start it on your local desktop machine.

3. Create a machine instance on AWS [Optional]

This section can be used only after the students register and activate their AWS accounts or the instructor creates IAM user accounts for the students through his/her account. The services provided by AWS cloud are many. But the one service that allows us to create machine instances is called the “EC2” service.

Step 1. Go to AWS console web page using this link and sign in. From the services menu at the top, select EC2.



You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Figure 1. EC2 Dashboard

Step2. From the left menu, select “AMIs” under “IMAGES”. The list of AMIs will appear on the right hand side frame. From the drop down menu, select “Private images” and you will see the AMI added by your instructor.

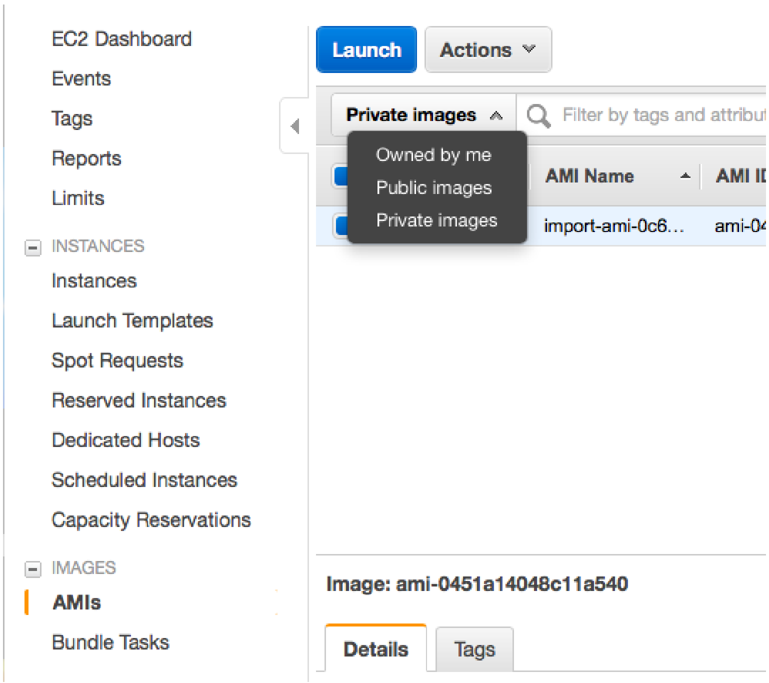


Image: ami-0451a14048c11a540

Figure 2. AMI list.

Step 3. Select the AMI from the list and select “Launch” from the “Actions” menu.

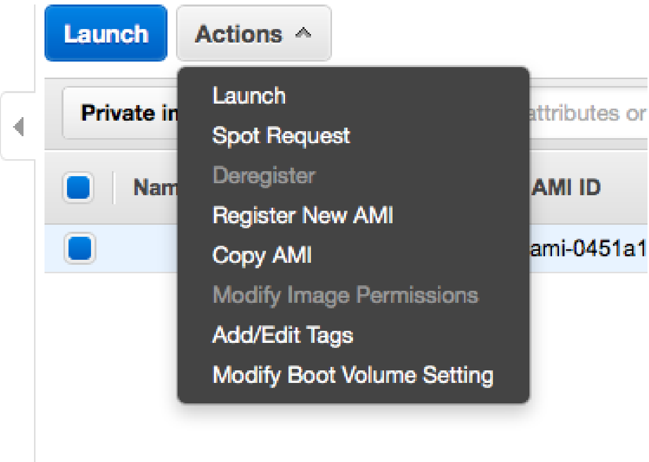
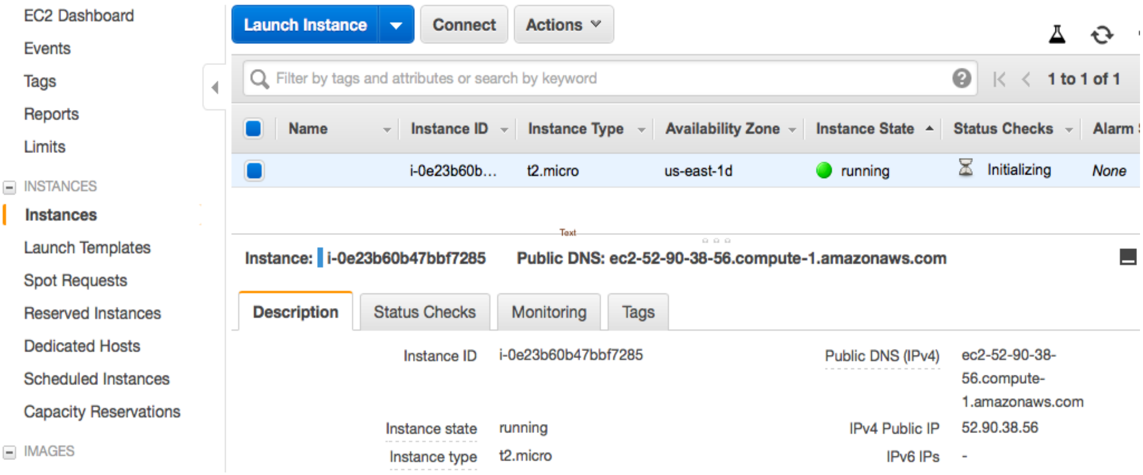


Figure 3. Launch your AMI

Step 4. For the following pages that come after “Launch”, do not make any changes and follow the instructions on the pages. The settings will automatically be selected by the specifications of the AMI. If not, choose a machine instance with at least 4GB of memory and 20GB of hard disk. Once you come to the page that asks for keypairs, select the option to “continue without a keypair” from the dropdown menu. Once you are done, go back to the EC2 dashboard and select instances on the left menu. Your launched instance along with its description will appear on the right frame. Make a note of the public IP address.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
	i-0e23b60b…	t2.micro	us-east-1d	● running	Initializing	None

Instance: i-0e23b60b47bbf7285 Public DNS: ec2-52-90-38-56.compute-1.amazonaws.com

Instance ID	i-0e23b60b47bbf7285	Public DNS (IPv4)	ec2-52-90-38-56.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	52.90.38.56
Instance type	t2.micro	IPv6 IPs	-

Figure 4. Instance specification

Once the status of your instance indicates “running”, you are ready to connect to the instance remotely.

Step5. Launch your Terminal program. Make sure that SSH is installed on it. In Windows machines, a Putty client can be used as well. Use the following command to connect to your remote machine:

```
$ ssh ubuntu@52.90.38.56
```

The numbers 52.90.38.56 indicate the IP address of the remote machine and it can be different for each instance. The username “ubuntu” is fixed, because that is the only username configured with this AMI.

Type ‘Yes’ to the question that appears and type ‘ubuntu2967’ for password. You will connect to the instance. After that, any command that you type will be executed on the remote machine instance.

Step6. Once you’re finished with the lab exercises, from the Actions menu click ‘Instance state’ and select ‘Terminate’. If the instance is not terminated, it will charge for the hours that the instance is alive. Therefore, it is very important that, once you are done, terminate the instance. Before that, save your work. Once the instance is terminated, all the changes you make in the file system will disappear.

4. Write a Perl script to access input files

The gene-sequencing input data resides under the directory /home/ubuntu/input/.

Step1. Create another directory called “scripts” under /home/ubuntu/ by using **mkdir** command.

Then, use **ls –l** command to see all files and directories under /home/ubuntu/ directory. If “scripts” directory is in this list, you are ready to move to the next step.

Step2. Go inside “scripts” directory by using the **cd** command. Then use **pwd** command to print the current working directory. If the output of the command is /home/ubuntu/scripts, then, move on to the next step.

Step3. Create a script called *lab1.pl* by using **vim** command. Go into *insert mode* and then do the following tasks one by one:

- Create a variable named $path and assign the value of $ARGV[0] to it. This path will come from the command line when you run the script. You will give the path of the input directory, which is /home/ubuntu/input. Print the path.

- Use backquotes ` ` to run **ls** command to list all the files that end with the extension *.fasta*. Use the wildcard character * to do that. Assign the returned output to an array named @ls_output. Use **print** function to print the entire array.

- Find the number of elements in the array by assigning the array @ls_output to a variable named $nums.

- For each file name in the array, print the entire path of the file. Then, use the **head** command and **system()** function to print the first five lines of each fasta file. Use a while loop to do that. The loop should iterate as many times as the value of $nums.

Step 4. Escape insert mode and exit vim text editor by saving your changes to the file.

Step 5. Execute the file as follows:

```
$ perl lab1.pl /home/ubuntu/input/
```

Your output should be similar to the one below:

```
$ perl lab1.pl /home/ubuntu/input/
/home/ubuntu/input/
/home/ubuntu/input/copy.fasta
/home/ubuntu/input/E.Coli_K12_MG1655.fasta
/home/ubuntu/input/copy.fasta

>NC_000913.3 Escherichia coli str. K-12 substr. MG1655, complete
genome
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAA
AAAGAGTGTCTGATAGCAGC
```

```
TTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAA
TATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACGCATTAGCACCACC
ATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAG
/home/ubuntu/input/E.Coli_K12_MG1655.fasta

>NC_000913.3 Escherichia coli str. K-12 substr. MG1655, complete genome
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
TTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAA
TATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACGCATTAGCACCACC
ATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAG
```

Step 6. Repeat all the steps for FASTQ files in the input directory as well.

Step 7. Submit your script and terminal outputs.

This OER material was produced as a result of the CS04ALL CUNY OER project.