Contents

[Create AWS account and EC2 instance 1](#_Toc197220533)

[Take note/Copy your Public IPv4 address and Connect to EC2 instance 7](#_Toc197220534)

[Get a permanent IPv4 address: 8](#_Toc197220535)

[Launch instances 9](#_Toc197220536)

[First time logging into EC2 instance (Windows) 9](#_Toc197220537)

[Subsequent time logging into EC2 instance 10](#_Toc197220538)

[To run analysis on WSL 10](#_Toc197220539)

[To run analysis on Jupyter Notebook\* 10](#_Toc197220540)

[\*To install Jupyter Notebook (only if you want to record your analysis) 11](#_Toc197220541)

[Taking a break from the analysis: 13](#_Toc197220542)

[If running analysis on WSL 13](#_Toc197220543)

[If running analysis on Jupyter Notebook 13](#_Toc197220544)

[Stop instance 14](#_Toc197220545)

# Create AWS account and EC2 instance

Create AWS account at <https://signin.aws.amazon.com/signup?request_type=register>

1. Search for and select EC2 to access the EC2 console.
2. Make sure that you are in the N. Virginia region
3. Select Launch Instances.

A screenshot of a computer

AI-generated content may be incorrect.

1. Name and tags: Name your instance (e.g., **Firstname\_Lastname**).
2. Amazon Machine Image: click on **Browse more AIMs**, chose **Community AMIs** tab, search for the course's AMI ID: **ami-00029a06cacbe647c,** select **cshl-seqtec-2024.**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Instance type: Select “**m6a.xlarge**”.

A screenshot of a computer

AI-generated content may be incorrect.

1. Key pair (login): because the existing key pair (e.g., cshl\_2024\_student) doesn’t pop up 🡪 **create a new one** (e.g., “RNAseq-AWS”; mine is “RNAseq2025”) and store it somewhere safe.

A screenshot of a computer error

AI-generated content may be incorrect.

Screens screenshot of a computer

AI-generated content may be incorrect.

After creating the key pair, the RNAseq-AWS.pem file will be downloaded to your laptop. Note the download location. This .pem file is important to log into EC2 instance.

1. Network settings: Because existing security group called “SSH/HTTP/Jupyter” doesn’t pop up **🡪 create one with** **SSH and HTTP access**

A screenshot of a computer

AI-generated content may be incorrect.

1. Configure storage: Make sure that you see two volumes.

A screenshot of a computer

AI-generated content may be incorrect.

1. Click on **Advanced details** dropdown menu: Select “**Termination Protection: Enable**”.
2. Summary: Review and then Launch instance.

A screenshot of a computer

AI-generated content may be incorrect.

1. Select **View all instances**

## Take note/Copy your Public IPv4 address and Connect to EC2 instance

A screenshot of a computer

AI-generated content may be incorrect.

This is what my instance looks like:

A screenshot of a computer

AI-generated content may be incorrect.

**When you stop and start your EC2 instance (not reboot), or in some cases even if the network refreshes (like connecting from a new location), AWS can assign a new public IPv4 IP address.**

## Get a permanent IPv4 address:

If you **don’t want the public IP to change**, you can assign an **Elastic IP (EIP)**:

1. Click on **Dashboard** (left bar), then click **Elastic IPs**
2. Click **Allocate Elastic IP Adress (orange button, top right corner) 🡪** scroll down, click **Allocate (orange button, bottom right corner)**
3. After it's created, click the **check box** in line with the **Allocated public IPv4 address**, then click **Actions → Associate Elastic IP address**
4. Choose your instance (eg, FirstnameLastName), then click **Associate**
5. Click on **Instances** (left bar), click the **check box** in line with your instance, the public IPv4 address you see now is permanent!

Now you’ll have a **permanent IP address**, even if the instance is restarted.

A screenshot of a computer

AI-generated content may be incorrect.

## Launch instances

A screenshot of a computer

AI-generated content may be incorrect.

# First time logging into EC2 instance (Windows)

Open WSL terminal. Follow this page to install WSL: <https://learn.microsoft.com/en-us/windows/wsl/install> (This YouTube video will walk you through it: <https://www.youtube.com/watch?v=pdoNbNKVs3k&t=4s>)

Now I’m in WSL terminal, I want to see if “.ssh” directory already exit, yes it does:

(base) lam@LamPhan:~$ ls -**a**  
. .bashrc .config .jupyter .ssh KnotFold  
.. .cache .gitconfig .local .sudo\_as\_admin\_successful Miniconda3-latest-Linux-x86\_64.sh  
.bash\_history .conda .ipython .motd\_shown DIY\_scRNAseq miniconda3  
.bash\_logout .condarc .java .profile Desktop

If “.ssh” doesn’t exit yet, create it by:

(base) lam@LamPhan:~$ mkdir -p ~/.ssh

Because my RNAseq2025.pem file was downloaded to Download folder in my laptop, I need to move (**mv** command) it to WSL:

(base) lam@LamPhan:~$ mv /mnt/c/Users/lammu/Downloads/[Your key pair name].pem ~/.ssh/

To see if pem file is successfully moved to “.ssh”:

(base) lam@LamPhan:~$ ls ~/.ssh  
[Your key pair name].pem known\_host

Change permission of pem file and log into EC2 instance. Type “yes” when asked if you want to connect:

(base) lam@LamPhan:~$ chmod 400 ~/.ssh/[Your key pair name].pem  
(base) lam@LamPhan:~$ ssh -i ~/.ssh/[Your key pair name].pem ubuntu@[Your **public** IPv4 address]  
The authenticity **of** host '3.236.179.99 (3.236.179.99)' can't be established.  
ED25519 **key** fingerprint **is** SHA256:aVi6eYlvKyylfgyKnkVZssCfH9Sj3/R3zxKl23Mlfds.  
This **key** **is** **not** known **by** any other names  
Are you sure you want **to** **continue** connecting (yes/no/[fingerprint])? yes

You’re inside the EC2 instance when you see:

ubuntu@ip-172-31-4-43:~$

To see what is listed in this parent directory (**ls** command):

ubuntu@ip-172-31-4-43:~$ ls  
R bin workspace

While you’re still logged into EC2 through WSL, set up your environment following the course’s instruction: <https://rnabio.org/module-00-setup/0000/09/01/Environment/>

# Subsequent time logging into EC2 instance

|  |  |
| --- | --- |
| To run analysis on WSL | To run analysis on Jupyter Notebook\* |
| Open WSL, type: ssh -i ~/.ssh/[Your key pair name].pem ubuntu@[Your public IPv4 address] | 1. Open PowerShell, type: ssh -i "C:\Users\lammu\[Your key pair name].pem" -L 8888:localhost:8888 ubuntu@[ Your public IPv4 address] 2. Open WSL, type:   ssh -i ~/.ssh/[Your key pair name].pem ubuntu@[Your public IPv4 address]  conda activate jupyter\_env  jupyter notebook   1. Copy/Paste one of the links provided to web browser 2. Create new bash file |

## \*To install Jupyter Notebook (only if you want to record your analysis)

**Step 1: SSH into EC2 instance through WSL (skip this step if you’re already logged in)**

(base) lam@LamPhan:~$ ssh -i ~/.ssh/[Your key pair name].pem ubuntu@[Your public IPv4 address]

**Step 2: Install miniconda**

ubuntu@ip-172-31-4-43:~$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh  
ubuntu@ip-172-31-4-43:~$ bash Miniconda3-latest-Linux-x86\_64.sh

When you install Miniconda, it usually adds a few lines (shown below) to .bashrc so the conda command is available every time you open a new shell.

# >>> conda initialize >>>  
\_\_conda\_setup="$('/home/ubuntu/miniconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"  
eval "$\_\_conda\_setup"  
# <<< conda initialize <<<

To view these extra lines for conda inside bashrc

ubuntu@ip-172-31-4-43:~$ nano ~/.bashrc

You can see that miniconda3 directory was created:

ubuntu@ip-172-31-4-43:~$ ls  
Miniconda3-latest-Linux-x86\_64.sh R bin miniconda3 workspace

Reload the .bashrc file with “source ~/.bashrc” so your current shell session picks up the new settings and make Conda’s base environment available:

ubuntu@ip-172-31-4-43:~$ source ~/.bashrc  
(base) ubuntu@ip-172-31-4-43:~$

Before source ~/.bashrc: the (base) prefix was missing, because the Conda environment wasn’t initialized. After “source ~/.bashrc”, the (base) shows up, which is confirmation that Conda is now active and working.

**Step 3: Create a conda environment (optional but clean)**

(base) ubuntu@ip-172-31-4-43:~$ conda create -n jupyter\_env python=3.10 -y  
(base) ubuntu@ip-172-31-4-43:~$ conda activate jupyter\_env

**Step 4: Install Jupyter Notebook**

(jupyter\_env) ubuntu@ip-172-31-4-43:~$ conda install -c conda-forge notebook -y

**Step 5: Configure Jupyter for remote access**

(jupyter\_env) ubuntu@ip-172-31-4-43:~$ jupyter notebook --generate-config

answer yes when ask if want to overwrite. Then edit the config:

(jupyter\_env) ubuntu@ip-172-31-4-43:~$ nano ~/.jupyter/jupyter\_notebook\_config.py

Make the following changes in the file:

c.ServerApp.ip = '0.0.0.0' # Accept connections from any IP (needed for remote access)   
c.ServerApp.port = 8888 # Use a fixed port  
c.ServerApp.open\_browser = False # Prevent browser from opening on the server

**Step 6: Set a password (optional but recommended-I didn't do)**

(jupyter\_env) ubuntu@ip-172-31-4-43:~$ jupyter notebook password

**Step 7: Install jupyter bash kernel**

(jupyter\_env) ubuntu@ip-172-31-4-43:~$ conda install -c conda-forge bash\_kernel -y  
(jupyter\_env) ubuntu@ip-172-31-4-43:~$ python -m bash\_kernel.install

**Step 8: Start Jupyter on the EC2**

(jupyter\_env) ubuntu@ip-172-31-4-43:~$ jupyter notebook

**Step 9: Open a new WSL to make a copy of pem file to /mnt/c/Users/lammu/ so PowerShell can access (step 10)**

(base) lam@LamPhan:~$ cp ~/.ssh/[Your key pair name].pem /mnt/c/Users/lammu/

**Step 10: Open PowerShell and run**

PS C:\Users\lammu> ssh -i "C:\Users\lammu\[Your key pair name].pem" -L 8888:localhost:8888 ubuntu@[Your public IPv4 address]

Note: the -L flag create an SSH tunnel from your local machine (through PowerShell) to the EC2 instance. The PowerShell window now shows as ubuntu@ip-xxxx because you're logged into the EC2 instance. Leave the used-to-be PowerShell open.

**Step 11: Go back to WSL terminal that has Jupyter Notebook initiated (step 8) open a browser and paste one of the URLs.**

Note: You can leave the PowerShell window open: The SSH connection and the tunnel will stay active until you close the PowerShell window or terminate the session. As long as the SSH session is running, your Jupyter Notebook will be accessible through localhost:8888.

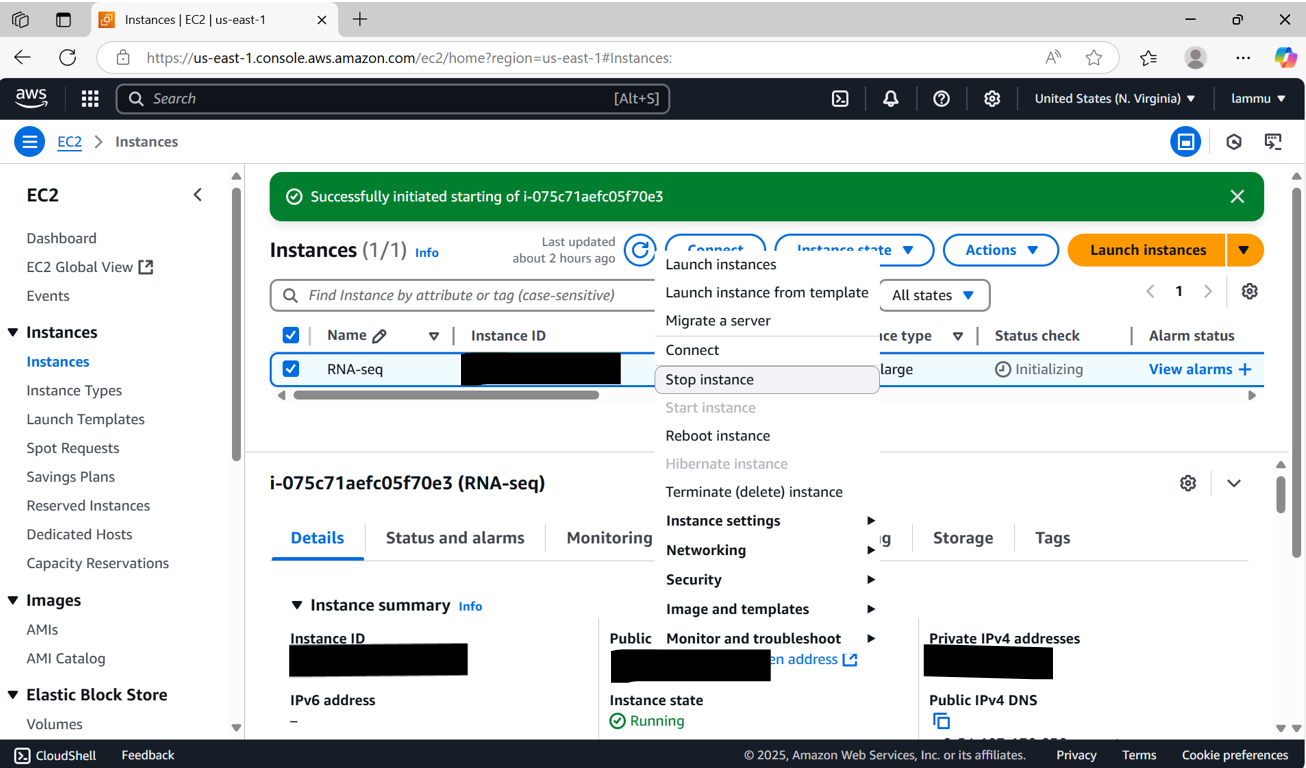
# Taking a break from the analysis:

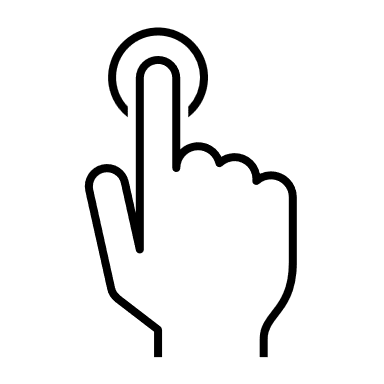
|  |  |
| --- | --- |
| If running analysis on WSL | If running analysis on Jupyter Notebook |
| On WSL, type: exit | 1. Close Jupyter Notebook bash file 2. On Jupyter Notebook’s Home page: File > Shut down 3. On WSL, type:   conda deactivate  exit   1. On PowerShell, type:   exit |

A screenshot of a computer

AI-generated content may be incorrect.

## Stop instance



A screenshot of a computer

AI-generated content may be incorrect.