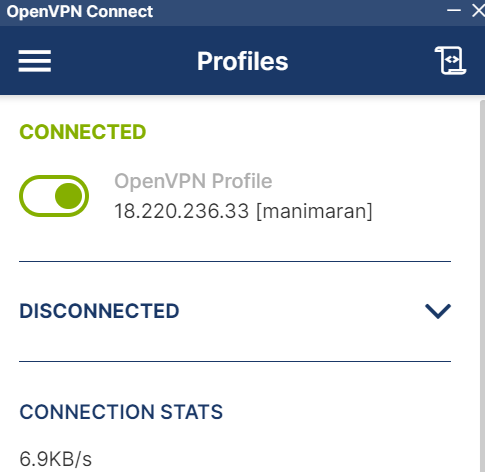
**FCD Pipeline Deployment Process**

# **Step1: Access the AWS console using OpenVPN.**

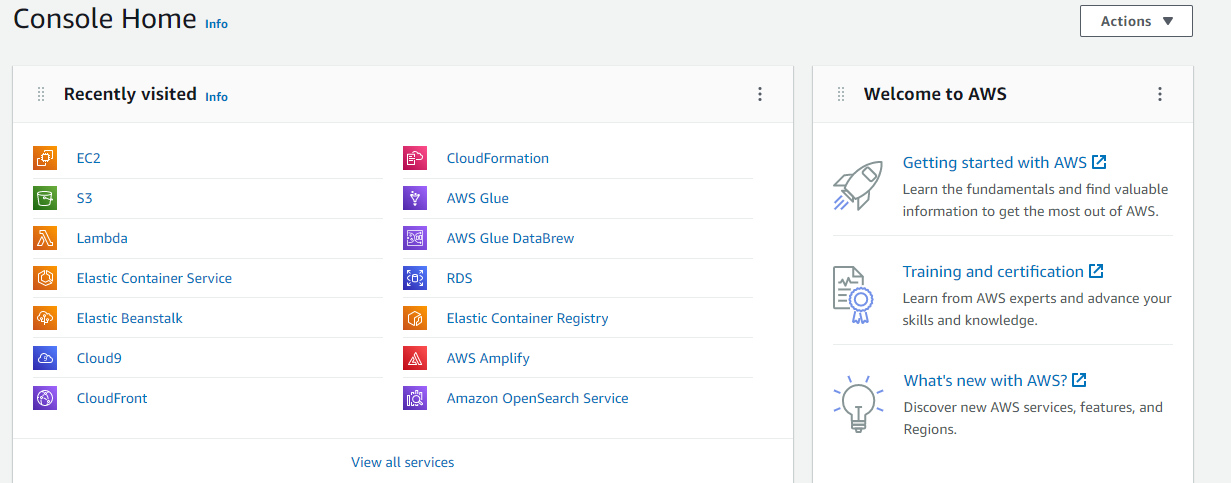
* Download the OpenVPN from the below URL

URL: <https://openvpn.net/downloads/openvpn-connect-v3-windows.msi>

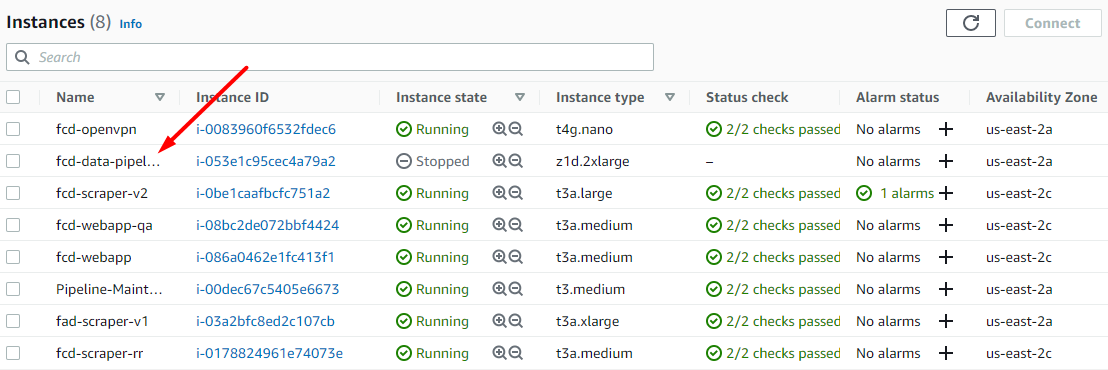
* Once downloaded, open the OpenVPN then access the AWS console using the right OpenVPN pem file



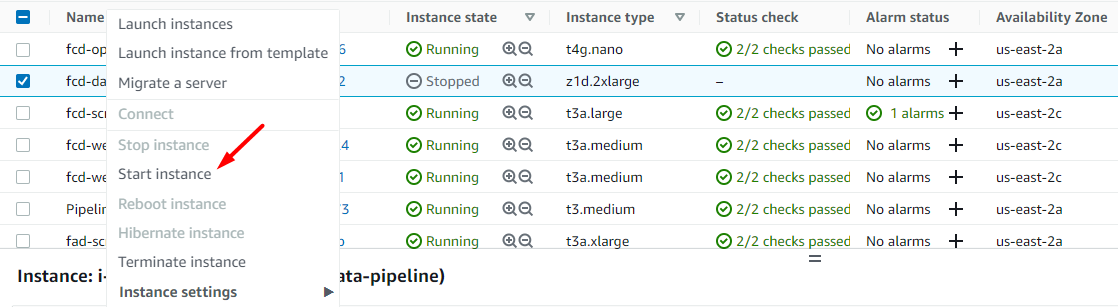
* Now Login into the AWS console using the respective IAM user credential.



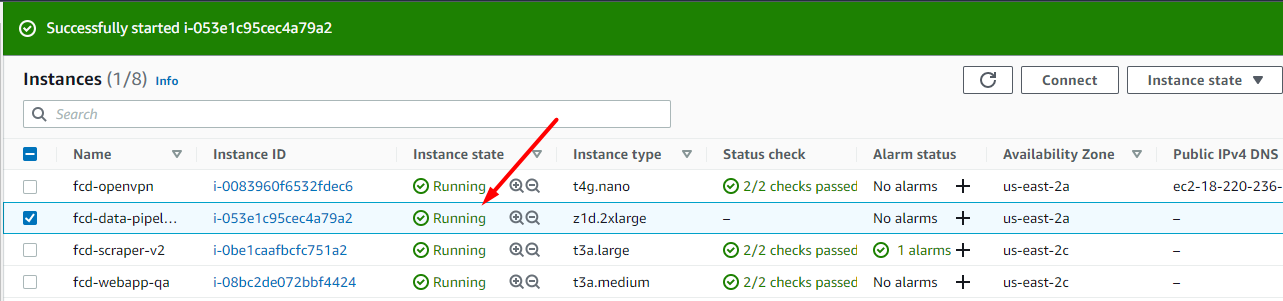
* Now Click on Ec2 service. In the Ec2 dashboard click on the instance
* Here we can able to see the Running instances and stopped instances in our AWS console.



* Here, we need to start the “FCD-DATA-PIPELINE” instance to execute the fcd pipeline script.
* Right, click the Instance then choose the “start instance option”



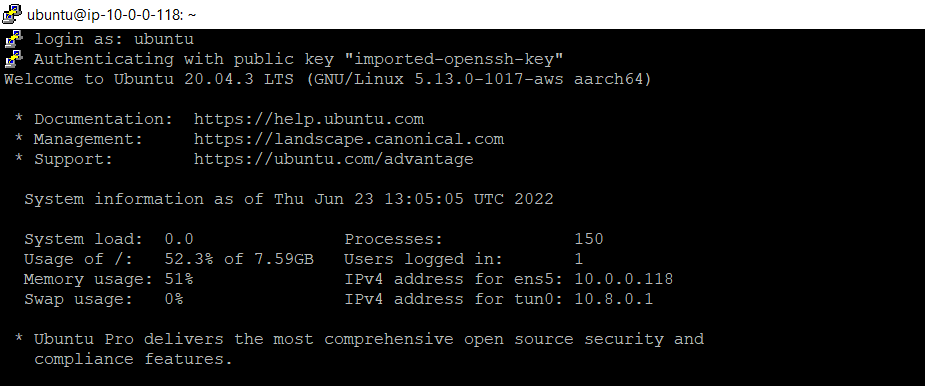
* Once the start the fcd-data-pipeline instance



* We need to SSH into the “fcd-data-pipeline” instance using OpenVPN
* We can use putty to connect the OpenVPN server, Now we need to SSH in to OpenVPN server.
* Connect with OpenVPN server using the below credential

IP Address: 10.0.0.118

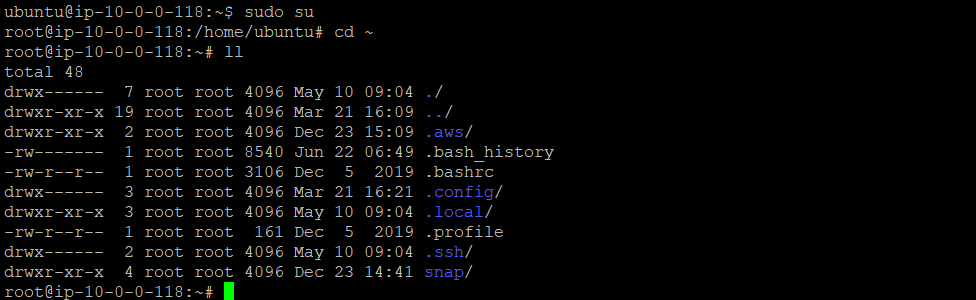
UserName: ubuntu



* Switch to root user using below commands

$ sudo su

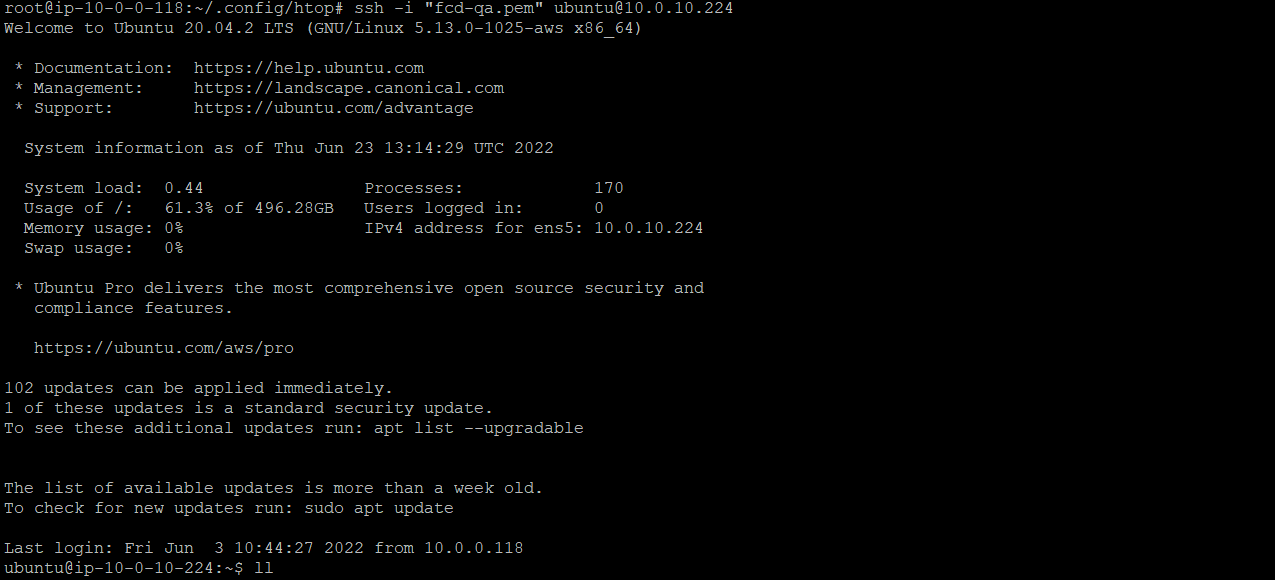
$ cd ~



$ cd ~/.config/htop/

* Execute the below commands in htop folder to connect the fcd pipeline server.

$ ssh -i "fcd-qa.pem" ubuntu@10.0.10.224



# **Step2: Update the Production server environment credential in .env file**

* We can export the below production credential in bashrc file and activate the bashrc file in current user.

export SECRET\_KEY='-$$@c!dm9#3h-g2$$f(=4c@6@u2epuwcu9#+#2$$c0&+-fsg8ox$$q'

export SQL\_ENGINE=django.db.backends.postgresql

export SQL\_DATABASE=postgres

export SQL\_USER=fcd\_app

export SQL\_PASSWORD=cgew2oqYU5lQ5w4m

export SQL\_HOST=fcd-prod-v1.cndbcmevjl4s.us-east-2.rds.amazonaws.com

export SQL\_PORT=5432

export ES\_HOST=https://vpc-fcd-prod-fsy66vojsrahpoo52hq25jfp34.us-east-2.es.amazonaws.com:443

export DATABASE=postgres

export DJANGO\_DEBUG=True

export DJANGO\_IMPORT\_DIR=/home/ubuntu/first-class-data-backend/import

export SENTRY\_DSN=https://52a6cee2ccd74de59217d236de21a05f@o1135282.ingest.sentry.io/6184032

export DATA\_PIPELINE\_CONCURRENCY=8

* We have already done the process for staging and production environment.
* We need to copy the bashrc-prod to .bashrc we use below command to copy the file

$ sudo cp bashrc-prod .bashrc

For staging :

$ sudo cp bashrc-staging .bashrc

* Now activate the .bashrc file using bellow command.

$ source .bashrc or source ~/.bashrc

* Then check the exported credential in source server

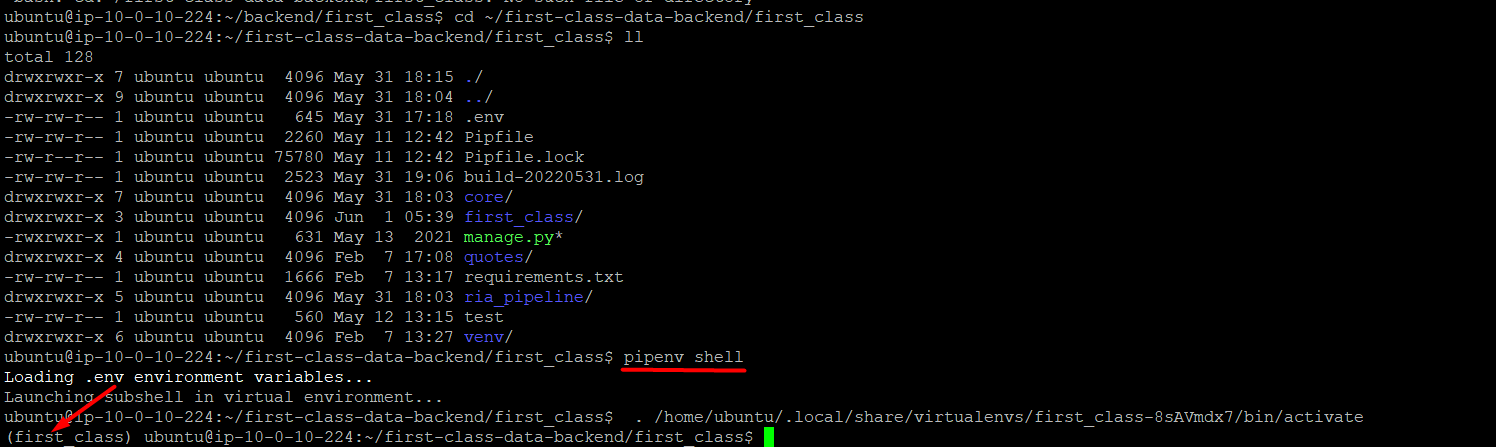
$ echo $SQL\_HOST

# **Step3: Activate the Virtual environment for production and staging server**

* We can use the below command to activate the virtual environment

$ cd ~/first-class-data-backend/first\_class

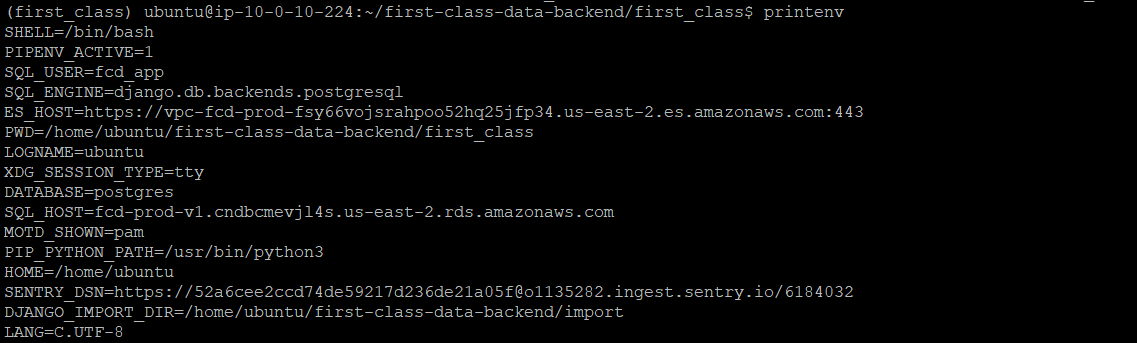
$ pipenv shell



# **Step4: Verify the Virtual environment credential in production server**

* Now check the Env credentials in current environment folder

$ printenv



# **Step5: We need to Synchronized the export folder to production S3 bucket**

* Need to Sync the import folder and S3 bucket in firstclass data environment

$ cd /first-class-data-backend

$ aws s3 sync --dryrun --exclude "\*.DS\_Store" s3://fcd-data-pipeline-sourcedata/stage0-sourcedata import/stage0-sourcedata

$ aws s3 sync --exclude "\*.DS\_Store" s3://fcd-data-pipeline-sourcedata/stage0-sourcedata import/stage0-sourcedata

# **Step6: start the build process for staging and production environment using below command.**

* We can use mini command to get the first 100K data from the each csv file and run the build process

$ pipenv run python manage.py build\_all --mini >> log.txt

* Mostly above command is used for local purpose
* Now , we use the below command in server to build the all process in fcd data

$ pipenv run python manage.py build\_all > build-20220628.log 2>&1 &

* When the build process is completed echo "All preparing steps completely, syncing these back to S3 bucket"

$ aws s3 sync --exclude "\*.DS\_Store" ../import/stage1-cleaned s3://fcd-data-pipeline-sourcedata/stage1-cleaned

$ aws s3 sync --exclude "\*.DS\_Store" ../import/stage2-prepared s3://fcd-data-pipeline-sourcedata/stage2-prepared

* Once done, we need to run the below command in environment to update the database and elastic search .

$ pipenv run python manage.py build\_tabledata\_from\_source\_csv

$ pipenv run python manage.py merge\_down\_agents

$ pipenv run python manage.py remove\_unwanted\_agents

$ pipenv run python manage.py remove\_common\_phones\_emails

$ pipenv run python manage.py reapply\_ncoa

$ pipenv run python manage.py infer\_data

$ pipenv run python manage.py chunk\_tabledata\_by\_agent\_id

$ pipenv run python manage.py tidy\_data

$ pipenv run python manage.py import\_tabledata\_to\_postgres

$ pipenv run python manage.py import\_es\_documents