☁️

Cloud Computing

DATA LAKE -

a system of data stored in natural/raw format | a single store of data including raw copies of source system data/sensor data/social data/transformed data used for reporting/visualization/advanced analytics/machine learning | includes structured data from relational databases/semi-structured data (CSV, logs, XML, JSON)/unstructured data (emails, documents, PDFs)/binary data (images, audio, video) | can be established "on premises" (within an organization's data centers) or "in the cloud" (using cloud services from Amazon/Microsoft/Google).

DATA SWAMP -

poorly managed data lakes have been facetiously called data swamps

SECURITY GROUP -

controls traffic that is allowed to reach/and leave resources | controls the inbound and outbound traffic for the instance

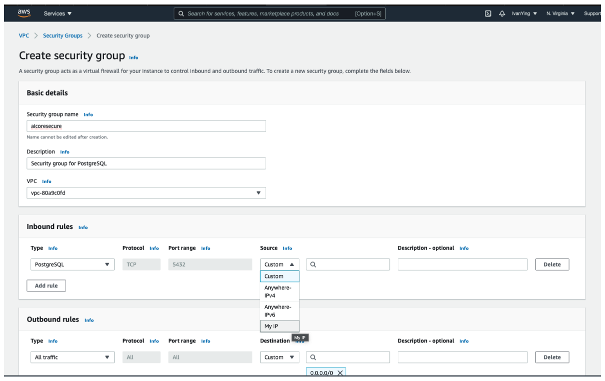
Amazon Web Services (AWS)

https://aws.amazon.com/rds/

Go to the console https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1#

1. Click on the ‘services’ tab
2. Establish a virtual network (this will limit the range of IP addresses that can be used to access the service)
3. Click ‘VCP’
4. Click ‘security groups’
5. Click ‘create security group’
6. Give it a name and description

1. Add an inbound rule, connect to the RDS instance from out local IP (allow the local machine to connect to the RDS ports)
2. In source, choose My IP (so you are the only one with access)
3. If you have a dynamic IP, this will work the first time, but next time your connection will be refused. In that case, change the source to Anywhere IPv4
4. Click ‘ create security group’


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
IvanYing
N. Virginia
Support
VPC > Security Groups > Create security group

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name: aicoresecure

Name cannot be edited after creation.

Description: Security group for PostgreSQL

VPC: vpc-80a9c0fd

Inbound rules

Type	Protocol	Port range	Source		Description - optional
PostgreSQL	TCP	5432	Custom		

Outbound rules

Type	Protocol	Port range	Destination		Description - optional
All traffic	All	All	Custom	0.0.0.0/0	

Relational Database Service (RDS)

AWS RDS is a service that allows you to create a database in the cloud. It is a highly scalable database that can be used for a variety of purposes. It allows creation of databases of different types, such as PostgreSQL, MySQL, Oracle, and Amazon Aurora

1. Click on the ‘services’ tab
2. Click ‘RDS
3. Click ‘create database’
4. Click ‘PostgreSQL’
5. Click ‘Standard Create’
6. Click ‘Free Tier’
7. Give it a name
8. Dont change ‘Master Username’
9. Make a password
10. Check ‘public access’
11. Add ‘existing security group’
12. Click ‘create’
13. Wait for it to be available

Take note of the Endpoint, which is the IP address of the database

```
from sqlalchemy import create_engine

DATABASE_TYPE = 'postgresql'
DBAPI = 'psycopg2'
```

```
ENDPOINT = 'datacollectionpipeline.c9nuz10uy6hn.us-east-1.rds.amazonaws.com' # Change it for your AWS endpoint
USER = 'postgres'
PASSWORD = 'Kilapila7'
PORT = 5432
DATABASE = 'postgres'

engine = create_engine(f"{DATABASE_TYPE}+{DBAPI}://{USER}:{PASSWORD}@{ENDPOINT}:{PORT}/{DATABASE}")
engine.connect()
```

Simple Storage Service (S3)

Go to the console https://us-east-1.console.aws.amazon.com/console/home?nc2=h_ct&region=us-east-1&src=header-signin#

1. Type in ‘S3’
2. Click ‘create bucket’
3. Give it a name
4. Change the region
5. Type in IAM
6. Click ‘users’
7. Click ‘add user’
8. Make a username
9. Tick ‘programmatic access’
10. Click ‘next’
11. Click ‘attach existing policies directly’
12. Tick ‘administrator access’
13. Click ‘next’ until you can create a user
14. Click ‘create user’
15. Download the csv (because you can only see it once)

To connect the local system to AWS

```
pip install awscli
aws configure
aws s3 ls
```

1. Install `awscli`
2. Then `aws configure`
3. Type in the info exactly as in the csv file
4. Make sure the region name matches the one on aws (eu-west-2)
5. Skip the last one by pressing enter

To use your AWS resources

```
pip install boto3
```

```
import boto3

# Uploads files to bucket
s3_client = boto3.client('s3')
# (file_name, bucket, object_name)
response = s3_client.upload_file('cat_0.jpg', 'cat-scraper', 'cat.jpg')

# Displays bucket contents
s3 = boto3.resource('s3')
my_bucket = s3.Bucket('data-collection-pipeline-bucket')
for file in my_bucket.objects.all():
    print(file.key)

# Download files from bucket
s3 = boto3.client('s3')
s3.download_file('pokemon-sprites', 'zubat/front.png', 'zubat.png')

# Download via bucket url
import requests
url = 'https://pokemon-sprites.s3.amazonaws.com/blastoise/front.png'

response = requests.get(url)
with open('blastoise.png', 'wb') as f:
    f.write(response.content)
```

Examples

```
# Link to S3 Bucket (millie-pokemon-bucket, pokemon_user)
# Upload to S3 Bucket
# !pip install boto3

import boto3

s3 = boto3.client('s3')
with open("Sprites/vulpix_front.png", "rb") as f:
    s3.upload_fileobj(f, "millie-pokemon-bucket", "vulpix_front.png")
```

6. Use `aws s3 ls` to check its working

1. Install `boto3`
2. Tell boto that you want to use a bucket
3. Upload something
4. Display contents
5. Download something

1. Click on an object in your S2 and it will give you a url
2. use this url to download the object

Make bucket public

1. In the console, in your bucket
2. Click ‘permissions’
3. Disable ‘Block public access’
4. Go here http://awspolicygen.s3.amazonaws.com/policy
5. Click ‘type of policy’
6. Select ‘S3 bucket policy’
7. Type ‘*’ into ‘principle’
8. In ‘actions’ select ‘get object’
9. In ‘Amazon Resource Name (ARN)’ type ‘arn:aws:s3:::your_bucket_name/*’
10. Click ‘add statement’
11. Click ‘Generate Policy’
12. Copy the text
13. In the console click ‘permissions’
14. Edit ‘bucket policy’
15. Paste the text
16. Click ‘save changes’

Security Groups

SECURITY GROUP - controls traffic that is allowed to reach/and leave resources | controls the inbound and outbound traffic for the instance

When you create a VPC, it comes with a default security group.

You can create additional security groups for each VPC.

You can associate a security group only with resources in the VPC for which it is created.

For each security group, you add *rules* that control the traffic based on protocols and port numbers.

There are separate sets of rules for inbound traffic and outbound traffic.

You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

Elastic Compute Cloud (EC2)

“remote computers that can be employed to run code on a cluster or a single computer”

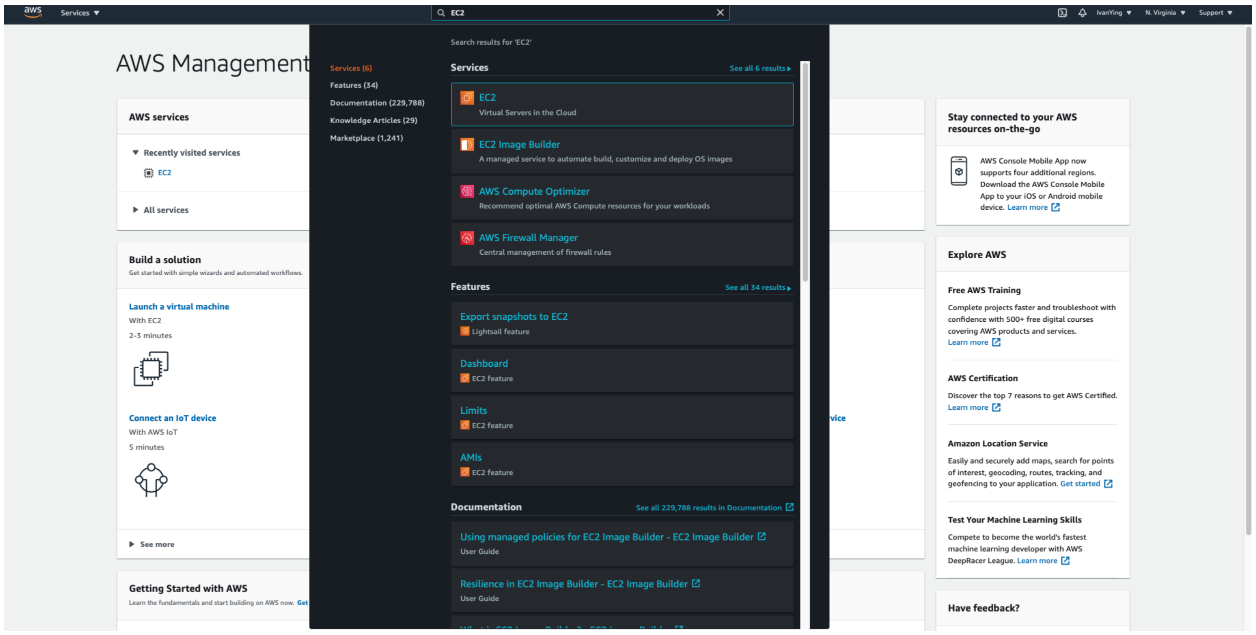
An EC2 instance can be accessed through a terminal. The instance can be accessed in the CLI by establishing an SSH connection to the instance. Once an SSH connection has been established, you can run commands as you would in your CLI.

SSH Key

The first step is to create a key-pair file, which would enable you to connect to the instance.

To create a key pair

1. Go to the AWS console
2. Click on the EC2 section
3. Click on the `Create Key Pair` button.



Get started with simple wizards and automated workflows.

A managed service to automate build, customize and deploy OS images

Recommend optimal AWS Compute resources for your workloads

Central management of firewall rules

Using managed policies for EC2 Image Builder - EC2 Image Builder

Resilience in EC2 Image Builder - EC2 Image Builder

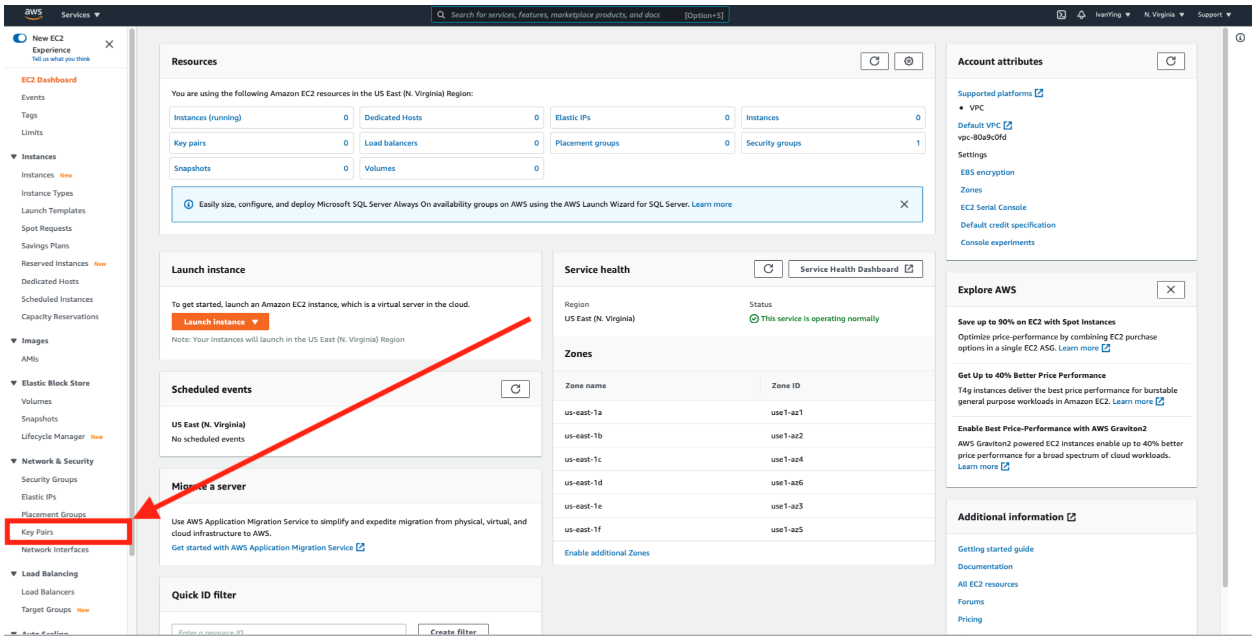
AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Complete projects faster and troubleshoot with confidence with 500+ free digital courses covering AWS products and services. Learn more

Discover the top 7 reasons to get AWS Certified. Learn more

Easily and securely add maps, search for points of interest, geocoding, routes, tracking, and geofencing to your application. Get started

Compete to become the world's fastest machine learning developer with AWS DeepRacer League. Learn more



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

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	0	Load balancers	0	Placement groups	0	Security groups	1
Snapshots	0	Volumes	0				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

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

US East (N. Virginia)
No scheduled events

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.
Get started with AWS Application Migration Service

Region: US East (N. Virginia)
Status: This service is operating normally

Zones

Zone name	Zone ID
us-east-1a	use1-az1
us-east-1b	use1-az2
us-east-1c	use1-az4
us-east-1d	use1-az6
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

Supported platforms
- VPC

Default VPC
vpc-80a9c0fd

Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

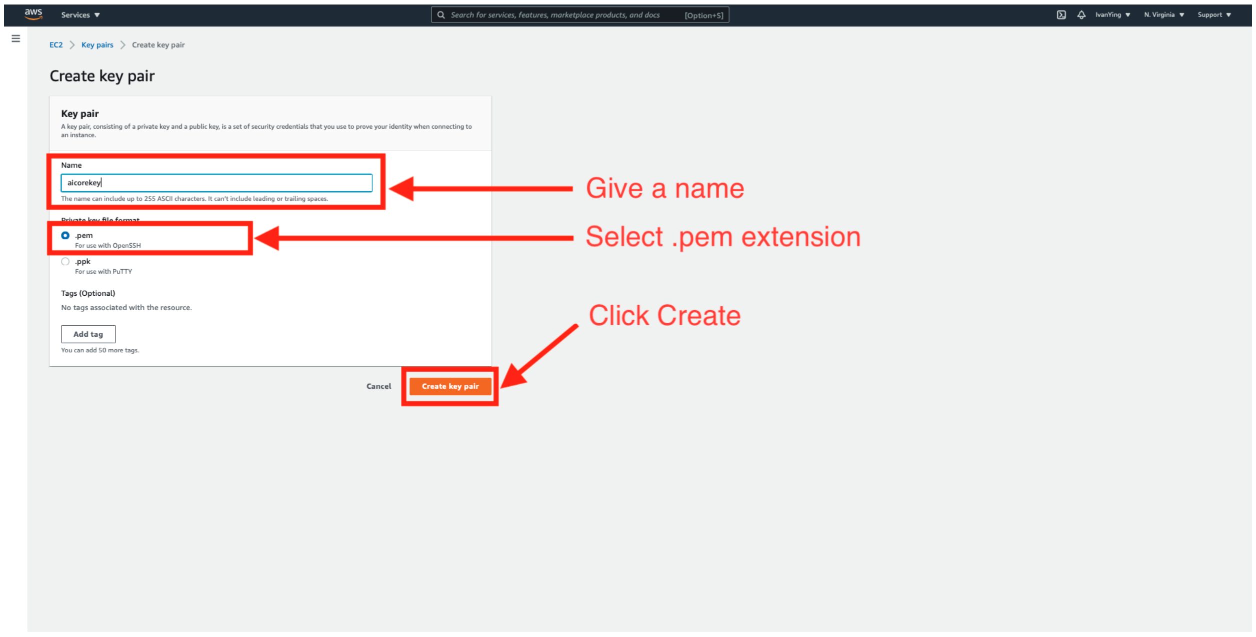
Save up to 90% on EC2 with Spot Instances
Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. Learn more

Get Up to 40% Better Price Performance
T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

Enable Best Price-Performance with AWS Graviton2
AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. Learn more

4. You can provide a name, as well as an extension

5. A `.pem` file with the name of the key pair will be created. This file contains the key pair that you will use to connect to the instance, and it will be automatically downloaded immediately after you click `'Create key pair'`.



Create key pair

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

aicorekey

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Private key file format

[✓ .pem] For use with OpenSSH

.ppk For use with PuTTY

Tags (Optional)

No tags associated with the resource.

You can add 50 more tags.

6. Ensure that the `.pem` file has only read permission. To do this, run the following command: `chmod 400 <key-pair-name>.pem` . `chmod` stands for `change mode`, and `400` is the permission. When you run the command, ensure that you are in the same directory as the `.pem` file.

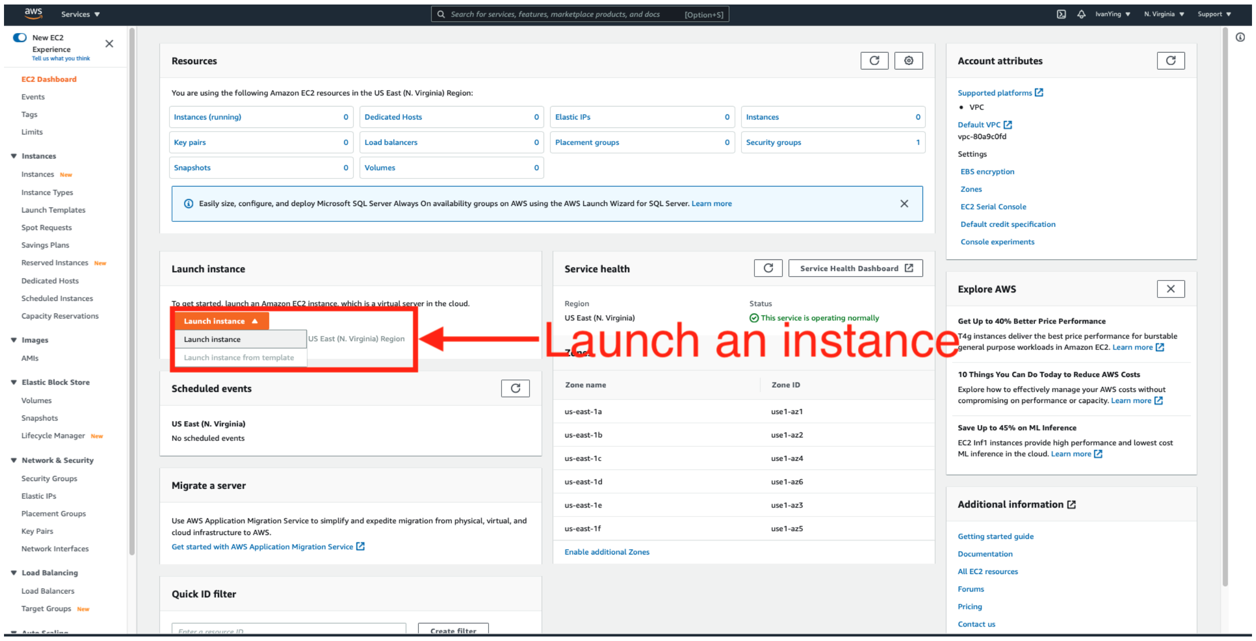
Create EC2 Instance

Remember that to enable access to the EC2 instance, a security group is required.

7. To create a security group, go to the AWS console and click on the EC2 section. You will see a list of security groups. Click on the `Create Security Group` button, and ensure that you set the correct inbound rules:

- HTTP: Anywhere IPv4
- HTTPS: Anywhere IPv4
- SSH: My IP

8. Once completed, remember to provide a memorable name. After creating the security group, click on the `Launch Instance` button on the EC2 main page.



Resources

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

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	0	Load balancers	0	Placement groups	0	Security groups	1
Snapshots	0	Volumes	0				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▲
Launch instance
Launch instance from template
US East (N. Virginia) Region

Service health

Service Health Dashboard

Region
US East (N. Virginia)

Status
This service is operating normally

Zones

Zone name	Zone ID
us-east-1a	use1-az1
us-east-1b	use1-az2
us-east-1c	use1-az4
us-east-1d	use1-az6
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

Scheduled events

US East (N. Virginia)
No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.
Get started with AWS Application Migration Service

Quick ID filter

Enter a resource ID
Create filter

Account attributes

Supported platforms
- VPC

Default VPC
vpc-80a9c0fd

Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Explore AWS

Get Up to 40% Better Price Performance
T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

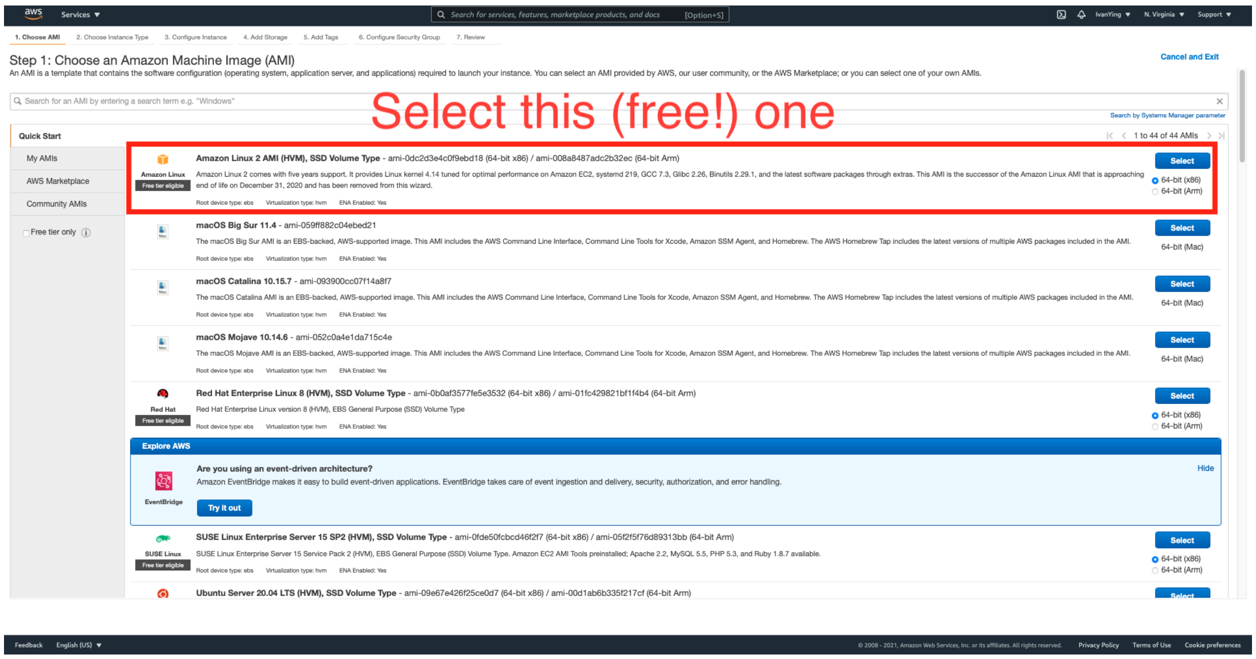
10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity. Learn more

Save Up to 45% on ML Inference
EC2 Inf1 instances provide high performance and lowest cost ML inference in the cloud. Learn more

Additional information

Getting started guide
Documentation
All EC2 resources
Forums
Pricing
Contact us

9. You will be asked to choose an Amazon machine image (AMI). For this lesson, we will use the `Amazon Linux 2 AMI` image.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Amazon Linux (Free tier eligible) — **Amazon Linux 2 AMI (HVM), SSD Volume Type** - ami-0dc2d3e4c0f9ebd18 (64-bit x86) / ami-008a8487adc2b32ec (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
[✓ 64-bit (x86)] 64-bit (Arm)

macOS Big Sur 11.4 - ami-059ff882c04ebed21
The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (Mac)

macOS Catalina 10.15.7 - ami-093900cc07f14a8f7
The macOS Catalina AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (Mac)

macOS Mojave 10.14.6 - ami-052c0a4e1da715c4e
The macOS Mojave AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (Mac)

Red Hat (Free tier eligible) — **Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** - ami-0b0af3577fe5e3532 (64-bit x86) / ami-01fc429821bf1f4b4 (64-bit Arm)
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
[✓ 64-bit (x86)] 64-bit (Arm)

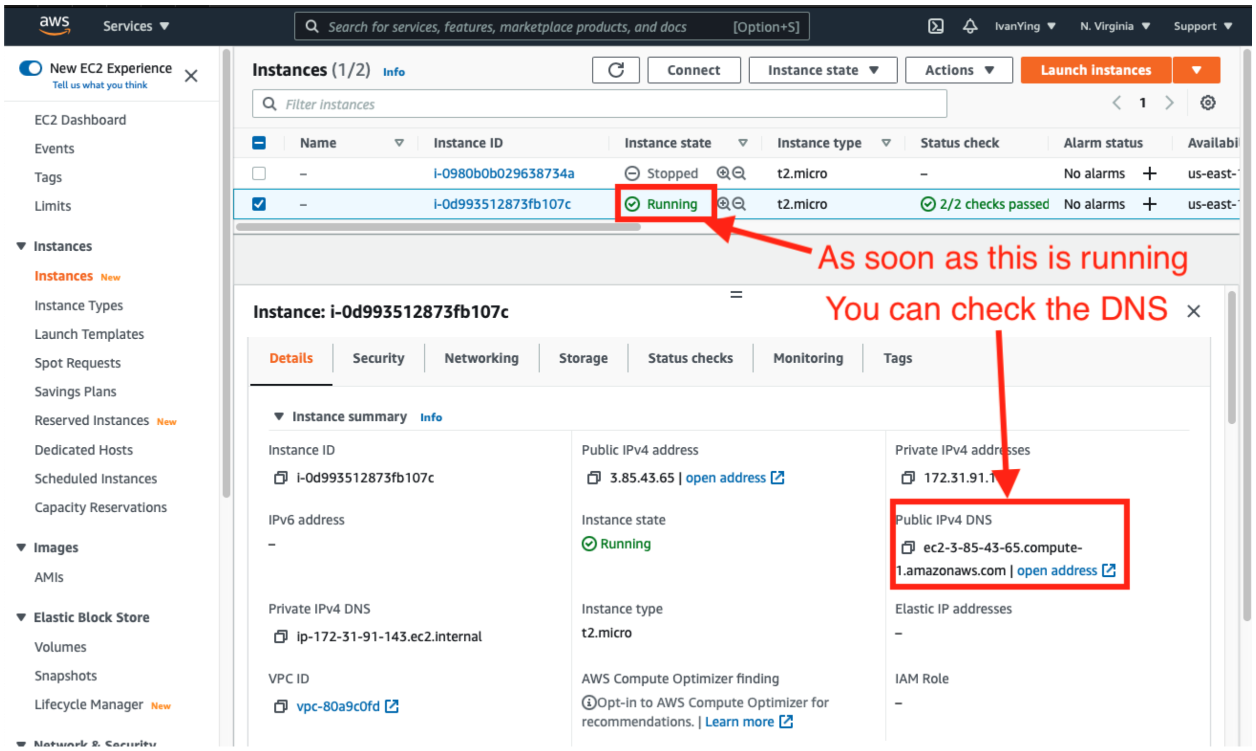
Explore AWS
EventBridge — **Are you using an event-driven architecture?**
Amazon EventBridge makes it easy to build event-driven applications. EventBridge takes care of event ingestion and delivery, security, authorization, and error handling.

SUSE Linux (Free tier eligible) — **SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type** - ami-0fde50fcbcd46f2f7 (64-bit x86) / ami-05f2f5f76d89313bb (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
[✓ 64-bit (x86)] 64-bit (Arm)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7 (64-bit x86) / ami-00d1ab6b335f217cf (64-bit Arm)

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10. Click on Launch, and you will be prompted to select a key pair.
11. Select the key-pair file you created, and click Launch Instance. The launch process should take a few minutes.
12. Once launched, view its public DNS:



Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabi
–	i-0980b0b029638734a	Stopped	t2.micro	–	No alarms	us-east-
–	i-0d993512873fb107c	Running	t2.micro	2/2 checks passed	No alarms	us-east-

Instance: i-0d993512873fb107c

Instance summary Info

Instance ID i-0d993512873fb107c	Public IPv4 address 3.85.43.65 | open address	Private IPv4 addresses 172.31.91.1
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-3-85-43-65.compute-1.amazonaws.com | open address
Private IPv4 DNS ip-172-31-91-143.ec2.internal	Instance type t2.micro	Elastic IP addresses –
VPC ID vpc-80a9c0fd	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more	IAM Role –

13. If everything works well, you should be able to connect to the instance through SSH. Ensure that your `.pem` file has read permission and that you specified the right path to the `.pem` file.
14. Connect to the EC2 instance, run the following command in the CLI: `ssh -i <key-pair-name>.pem ec2-user@<public-dns>` Remember to specify the path to the `.pem` file in the key name (unless, of course, you are in the same directory). You might be prompted to provide a fingerprint. This is the fingerprint of the key pair. You can ignore this fingerprint, and type `yes` to continue.
15. Populating the instance using secure copying, since it is empty. In your terminal, you can copy files to the EC2 instance by typing `scp -i </path/my-key-pair.pem> </path/my-file> ec2-user@<public-dns>:<path/>`

Crontab

Windows = https://www.windowscentral.com/how-create-automated-task-using-task-scheduler-windows-10

Crontab.guru - The cron schedule expression editor

The quick and simple editor for cron schedule expressions by Cronitor We created Cronitor because cron itself can't alert you if your jobs fail or never start. Cronitor is easy to integrate and provides you with instant alerts when things go wrong. Learn more about cron job monitoring.

https://crontab.guru/

https://vim.rtorr.com/

Scheduled tasks can be managed using crontab. Here, we will focus on three use cases:

- `crontab -l`: List the tasks that have been scheduled.
- `crontab -r`: Remove the tasks that have been scheduled.
- `crontab -e`: Edit the tasks you want to schedule.

Before editing the crontab, we examine the structure of a crontab. First, set a schedule using five numbers:

`minutes hours day_of_month month day_of_week`

The number you give to each of these fields will set the frequency of the command. The `*` wildcard is always available for setting `every` field. Some examples:

- `* * * *`: Every minute, every hour, every day of the month, every month, and every day of the week.
- `30 4 * * *`: Every day at 4:30.
- `* 5 * 6`: Every minute on all Saturdays (6), on the fifth day of the month.
- `0 0 * 11 *`: Every day at midnight in November (11).
- `0 0 1 1 *`: Yearly on the 1st of January.

The second part of the crontab comprises the command to be run. Remember that you are in the terminal. To run a python script, you should run `python3 test.py` for example.

Thus, the following command:

- `0 0 * * 5 python3 test.py`,

will run `test.py` weekly at midnight on Fridays.

Now, we add some tasks to the crontab file using the last command. Upon running it, Vim, a relatively complicated text editor, will open.

There are only two relevant commands:

- `i`: Enter Insert mode. After this, you will be able to write.
- `:wq`: Write and Quit. This saves the changes and quits the editor afterwards.

The Vim commands are provided in its console. If you are in Insert mode, simply press Esc to enter the console.

EC2 + Selenium

The created scraper might be configured to run daily or weekly.

- Problem: Running it manually might be tedious.
- Solution: Use `crontab`.
- Problem: `crontab` does not work if the computer is off.
- Solution: Run it on an EC2 instance and leave it open.
- Problem: Chrome or chromedriver for running Selenium is not available.

- Solution: Download it.
- Eventual Problem: When logging out, the script halts.
- Eventual Solution: Create a terminal multiplexor that leaves the session running.

Note: This only works with the Amazon Linux 2 AMI instance.

1. First, download Google Chrome, and determine its version. In your EC2 instance, run the following commands:

```
sudo curl https://intoli.com/install-google-chrome.sh | bash
sudo mv /usr/bin/google-chrome-stable /usr/bin/google-chrome
```

1. If no problem occurs, run the following to check the version: `google-chrome --version`

▼ For Ubuntu 20.04

1. At the time of writing, the current version is 93.x. Accordingly, we must download the corresponding chromedriver version. Since you are in a Linux instance, search for the right version on the next webpage: https://sites.google.com/chromium.org/driver/. Copy the link; there is no need to download the file.

```
sudo wget https://chromedriver.storage.googleapis.com/93.0.4577.15/chromedriver_linux64.zip
```

1. Unzip the downloaded file using `sudo unzip chromedriver_linux64.zip`.
2. Move that file to the PATH using `sudo mv chromedriver /usr/bin/chromedriver`.
3. You should be set to use selenium. (Remember to download `pip install selenium`).

Multiplexing

Once you start using selenium, your process will halt after the SSH is terminated. A terminal multiplexer with `tmux` presents a solution to this issue. This, in essence, will run a new terminal (or set of terminals) and keep the process running even if you disconnect from it.

To install it, run: `sudo yum install tmux`.

To improve your knowledge, experiment with `tmux`, since it has many commands. In this notebook, however, we will focus on creating terminals:

- To create a new session and connect to it, run `tmux new-session -s session_name`.
- Once inside, you can run anything as if you were in the main terminal. To detach from the session, press `Ctrl + B` and subsequently `D`.
- Create as many sessions as you want. To list the active sessions, run `t`mux list-sessions`.
- To reconnect to an existing session, run `tmux attach-session -t session_name`.
- To end a session, run `tmux kill-session -t session_name`.

Once up and running, you can log out of the EC2 instance, without interrupting or halting the process.