

Databases and Big Data

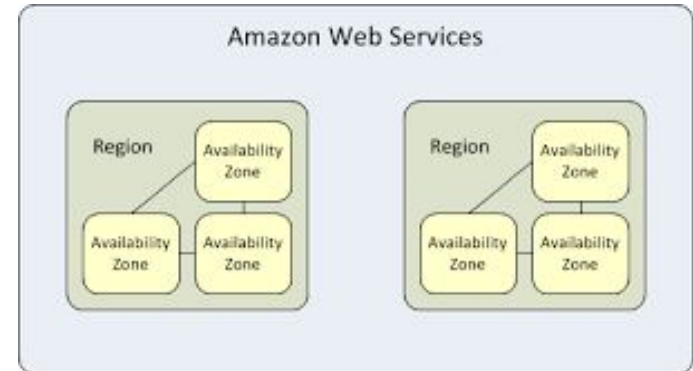
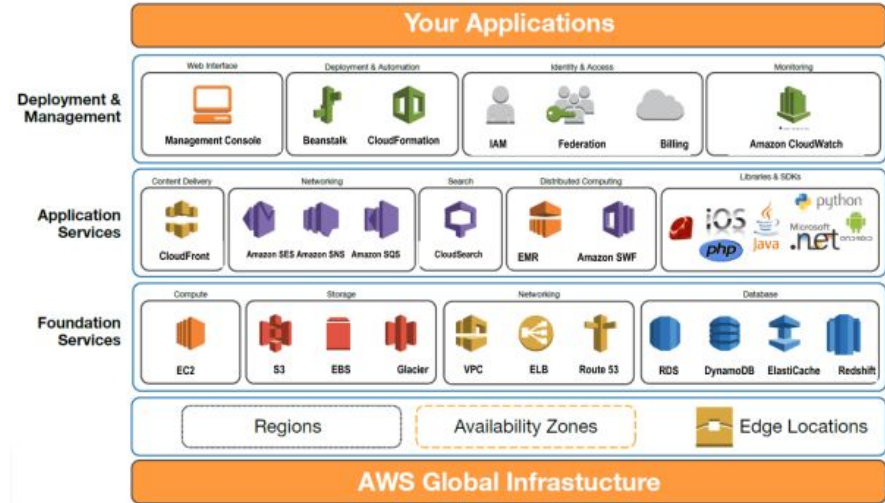
Lab 1

Today

- AWS Basics
- MySQL setup
- Boto3 (AWS Command Line Interface Wrapper for python)

AWS

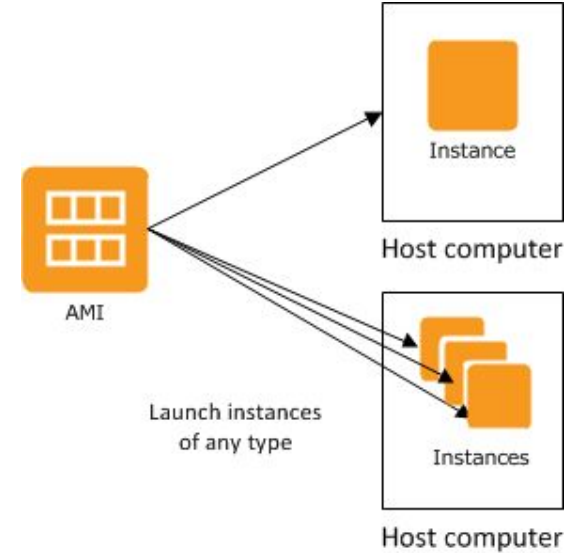
- Most popular
 - The earliest
- Only cover key concepts here
 - You have \$100 to play with
- **Region**
 - Where servers are
 - Geographical region, e.g. ap-southeast-1
 - Multiple **availability zones** per regions, e.g. ap-southeast-1a



AWS Services

- EC2

- Amazon Machine Image (**AMI**): like a virtual machine image
- **Instance**: running image on specific hardware
 - Like a virtual machine
- Instance **type**: what hardware

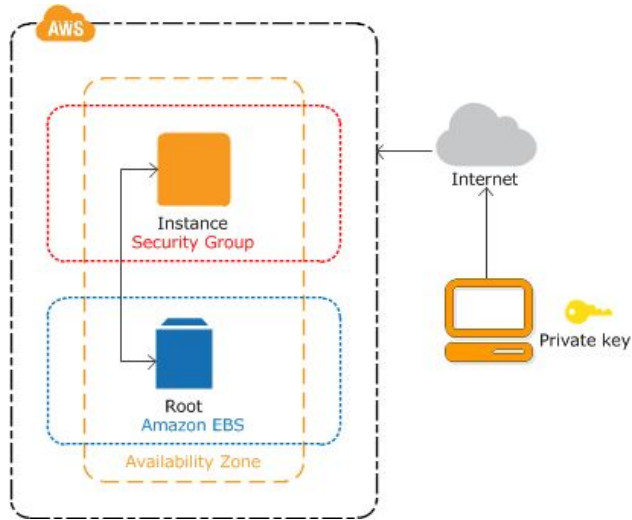


Instance	vCPU*	CPU Credits / hour	Mem (GiB)	Storage	Network Performance
t2.nano	1	3	0.5	EBS-Only	Low
t2.micro	1	6	1	EBS-Only	Low to Moderate
t2.small	1	12	2	EBS-Only	Low to Moderate
t2.medium	2	24	4	EBS-Only	Low to Moderate

Free*

AWS Services

- EC2



- Elastic Block Storage acts as hard disk.
- EC2 Instance and EBS storage should be on same region and same availability zone.





AWS Services

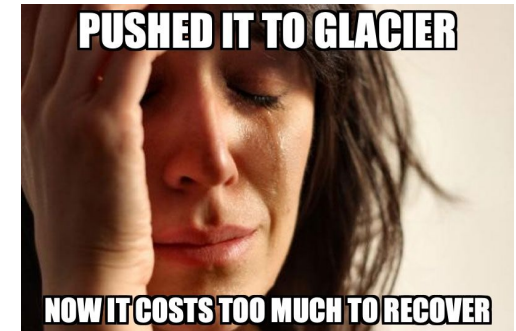
- EC2 - Security group
 - Firewall: rules for network access
 - Each instance belongs to one security group
 - You can create many groups
 - Default option: **publicly available**



AWS Services

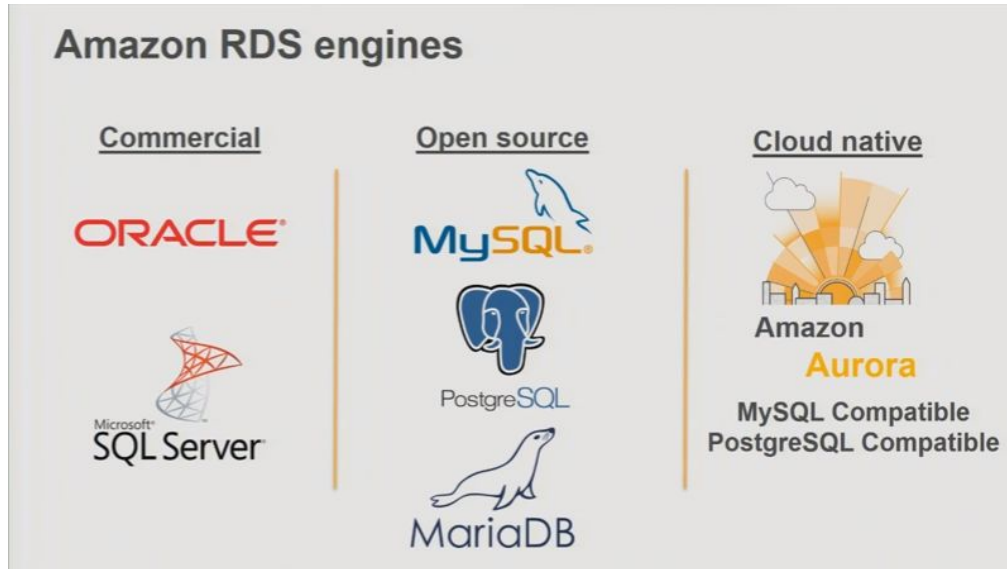
- Storage services

	 Amazon EBS	 Amazon S3	 Amazon Glacier	 Amazon EFS
Interface	Block	HTTP(S)/API	API	NFSv4
Concurrency	Single interface	Many connections	Low	Thousands
Consistency	Strongly consistent	Eventually consistent	Consistent	Strongly consistent
Latency	Low latency	Higher latency	3 options available	Low latency
Size	16 TiB/volume	5 TiB/item, unlimited	40 TiB/archive, unlimited	47.9 TiB/file
Durability	0.1-0.2% AFR AZ-specific volumes	99.999999999%	99.999999999%	Multi-AZ storage
Use Case	DBs, file servers	Static storage, high concurrency	Archival	Shared files, Big data analysis



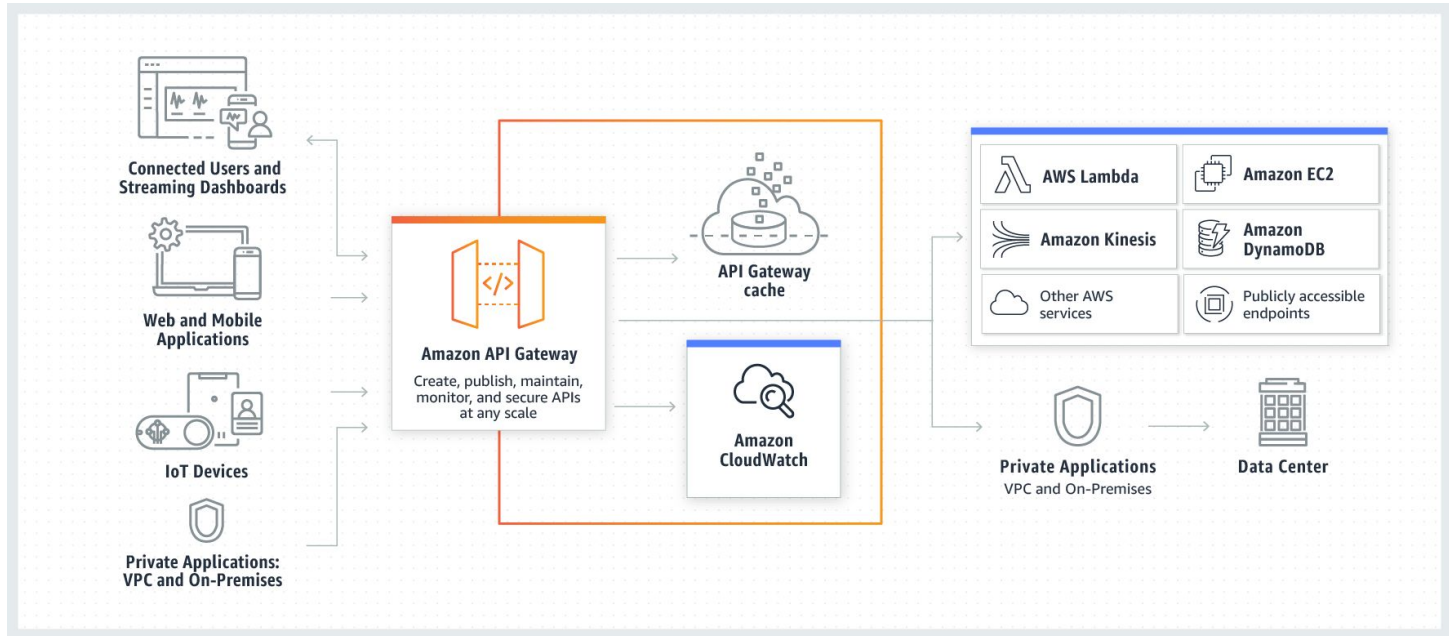
AWS Services

- RDS



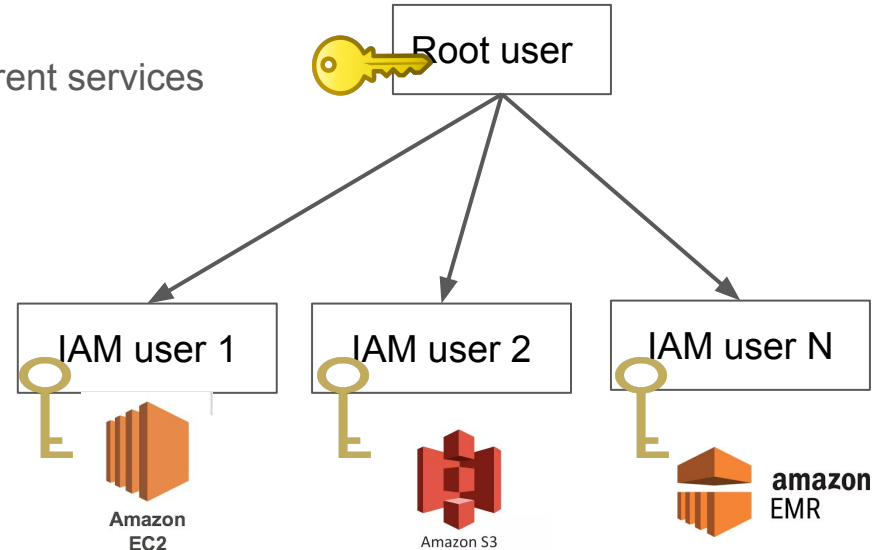
AWS Services

- API Gateway



AWS Services

- Identity and Access Management (IAM):
 - Your account has 1 root user (when sign up)
 - Root user can:
 - Create **IAM users**
 - Grant different IAM user access to different services



Why all these troubles?

AWS Services

- IAM

- 2 types of credentials

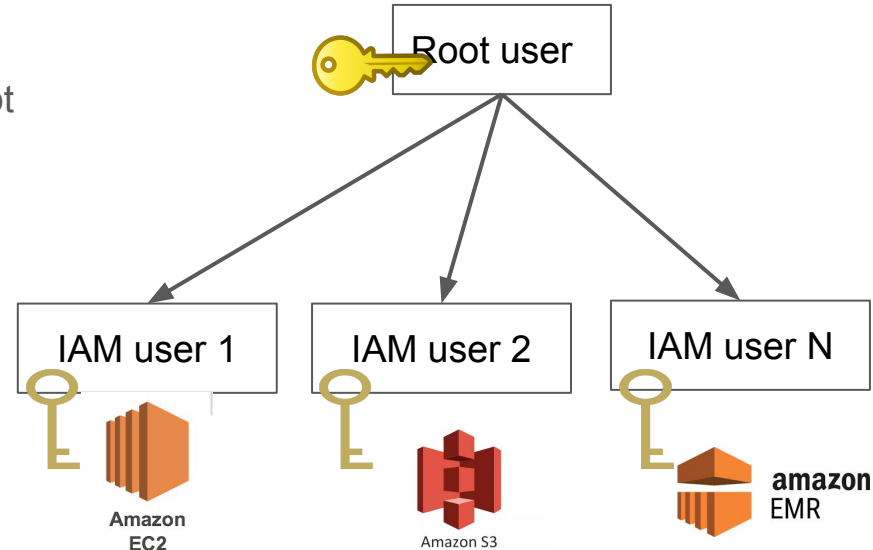


- **Access key:**

- Short, 40-character string
 - To invoke AWS APIs, must be kept secret.

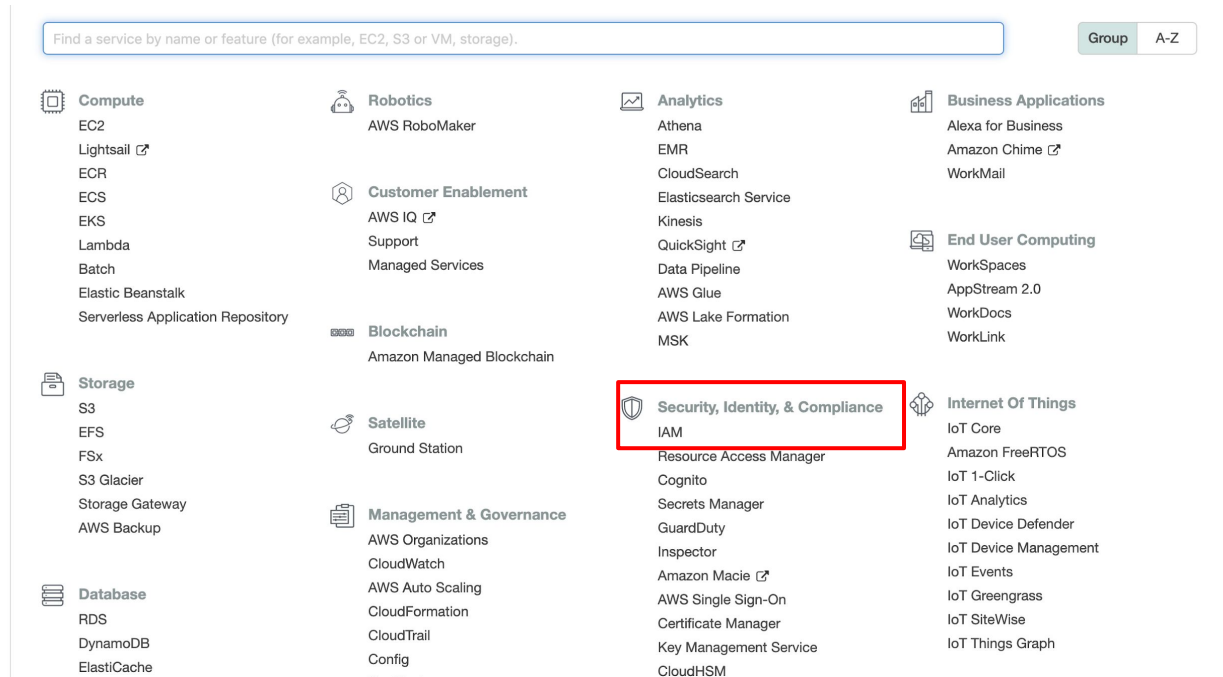
- **SSH key:**

- Standard SSH key
 - To log-in to the EC2 instances



AWS Services

- Access key created via IAM



Access Key

Add user



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://273066595606.signin.aws.amazon.com/console>



Download .csv

	User	Access key ID	Secret access key
▶	✓ dinhtta-test	A[REDACTED]ED	K[REDACTED] 2[REDACTED]

Key Pair

- SSH keys
 - To login to your instances

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

Lifecycle Manager

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Create Key Pair Import Key Pair Delete

Filter by attributes or search by keyword

<input type="checkbox"/>	Key pair name	Fingerprint
<input type="checkbox"/>	dinhhta	8:0f:56
<input type="checkbox"/>	dinhhta-mac	8:6f:19
<input type="checkbox"/>	keypair	

Select a key pair

AWS

- How to interact with AWS:
 - AWS GUI console
 - Doesn't scale beyond a handful of instances
 - This course:
 - Use command lines
 - Or boto3 (Python)

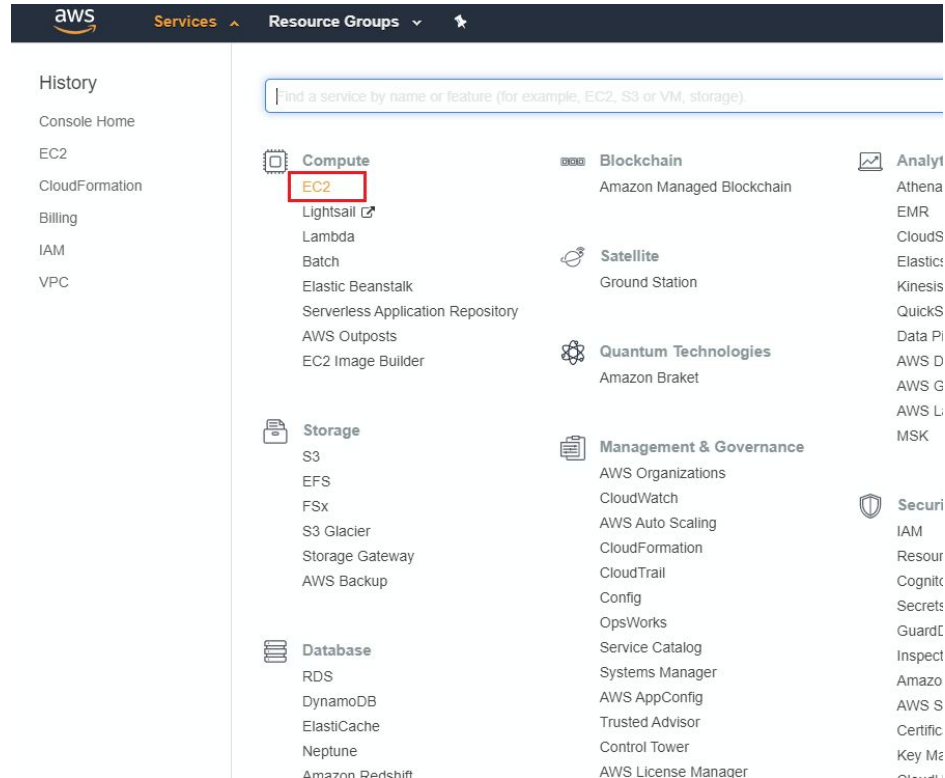


AWS EC2 - Creating a EC2 Instance

- Log into <https://console.aws.amazon.com> or <https://www.awseducate.com/signin>
- Make note of the region for logged in aws user

AWS EC2 - Creating a EC2 Instance

- Go to EC2 Service



AWS EC2 - Creating a EC2 Instance

- Go to Launch Instance Wizard

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', and 'Resource Groups'. The left sidebar contains the 'EC2 Dashboard' and a list of navigation links under 'Instances' and 'Images'. The main content area is divided into several sections: 'Resources', 'Launch instance', 'Scheduled events', and 'Service health'. The 'Launch instance' section has a red box around the 'Launch Instance' button. The 'Resources' section shows a summary of EC2 resources in the US East (N. Virginia) Region. The 'Service health' section shows the region and zone status.

Resources

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

Running instances	0	Elastic IPs	0	Dedicated Hosts
Volumes	0	Load balancers	0	Key pairs
Placement groups	0			

Launch instance

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

Launch Instance

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

Scheduled events

US East (N. Virginia)
No scheduled events

Service health

Region
US East (N. Virginia)

Zone status

Zone

us-east-1a (use1-az6)
us-east-1b (use1-az1)

AWS EC2 - Creating a EC2 Instance

- Select an AMI

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

Search by Systems Manager parameter






Quick Start < 1 to 40 of 40 AMIs >

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only ⓘ

 Amazon Linux Free tier eligible	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0c94855ba95c71c99 (64-bit x86) / ami-0d29b48622869dfd9 (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. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
 Amazon Linux Free tier eligible	Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-00514a528eadbc95b The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	Select 64-bit (x86)
 Red Hat Free tier eligible	Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-098f16afa9edf40be (64-bit x86) / ami-029ba835ddd43c34f (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	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
 SUSE Linux Free tier eligible	SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0a782e324655d1cc0 (64-bit x86) / ami-06ec4eaf39ca724d4 (64-bit Arm) SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
 Ubuntu Free tier eligible	Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-06b263d6ceff0b3dd (64-bit x86) / ami-0d324124b7b7eec66 (64-bit Arm) Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).	Select <input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)

AWS EC2 - Creating a EC2 Instance

- Select an Instance Type

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GiB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

AWS EC2 - Creating a EC2 Instance

- Select an Instance Type and Launch

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance typesCurrent generationShow/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel

Previous

Review and Launch

Next: Configure Instance Details

AWS EC2 - Creating a EC2 Instance

● Launch

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group


7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ AMI Details

Edit AMI

 **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-06b263d6ceff0b3dd**

Free tier eligible

Ubuntu Server 18.04 LTS (HVM).EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

▼ Instance Type

Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups

Edit security groups

Security group name

launch-wizard-1

Description

launch-wizard-1 created 2020-09-08T07:55:31.702+08:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
This security group has no rules				

▶ Instance Details

Edit instance details

▶ Storage

Edit storage

▶ Tags

Edit tags

Cancel

Previous

Launch

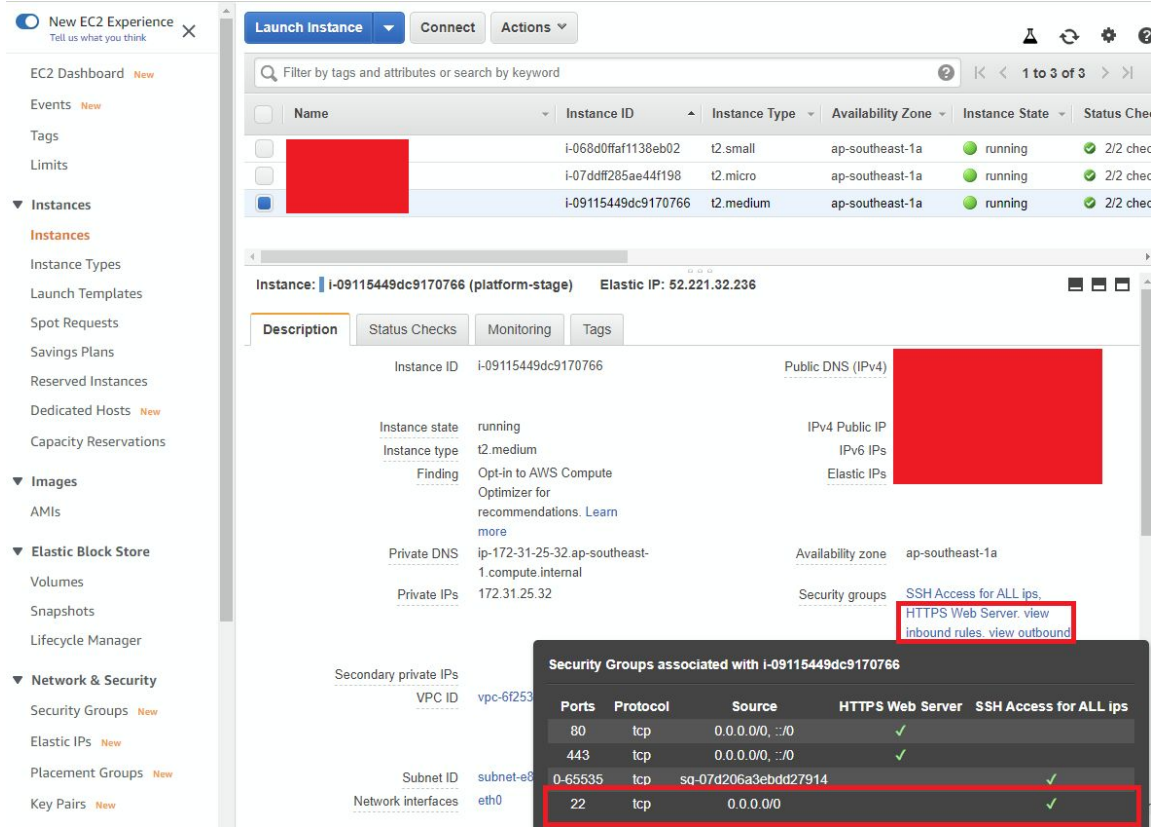
AWS EC2 - Creating a EC2 Instance

- Create Key and Launch

The screenshot shows the AWS Management Console interface during the 'Step 7: Review Instance Launch' process. The main window displays the instance configuration details, including the AMI (Ubuntu Server 18.04 LTS), Instance Type (t2.micro), and Security Groups. A modal dialog is open, titled 'Select an existing key pair or create a new key pair'. The dialog explains that a key pair consists of a public key and a private key file. It provides a text input field for the 'Key pair name' with the value 'newKey'. A red box highlights the 'Download Key Pair' button. Below the input field, a blue information box states: 'You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.' At the bottom of the dialog, there are 'Cancel' and 'Launch Instances' buttons, with the latter highlighted by a red box. The background shows the 'Review' step of the instance launch process, with various configuration options visible on the left and right sides.

AWS EC2 - Creating a EC2 Instance

- Security Group - Make sure that you allow SSH access



The screenshot displays the AWS Management Console interface for an EC2 instance. The left sidebar shows the navigation menu with options like EC2 Dashboard, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main content area shows the details of the instance **i-09115449dc9170766** (platform-stage) in the **ap-southeast-1** region. The instance is running and has a public IP address of **52.221.32.236**. The security groups associated with the instance are listed as **SSH Access for ALL ips** and **HTTPS Web Server**. A red box highlights the **SSH Access for ALL ips** security group, and a red box highlights the **SSH Access for ALL ips** rule in the **Security Groups associated with i-09115449dc9170766** table.

Ports	Protocol	Source	HTTPS Web Server	SSH Access for ALL ips
80	tcp	0.0.0.0/0 ::/0	✓	
443	tcp	0.0.0.0/0 ::/0	✓	
0-65535	tcp	sq-07d206a3ebdd27914		✓
22	tcp	0.0.0.0/0		✓

AWS EC2 - Accessing instance

- Locate the public IP of your instance

platform-stage i-09115449dc9170766 t2.medium ap-southeast-1a running

Instance: **i-09115449dc9170766 (platform-stage)** Elastic IP: 52.221.32.236

Description Status Checks Monitoring Tags

Instance ID	i-09115449dc9170766	Public DNS (IPv4)
Instance state	running	IPv4 Public IP
Instance type	t2.medium	IPv6 IPs
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs
Private DNS	ip-172-31-25-32.ap-southeast-1.compute.internal	Availability zone
Private IPs	172.31.25.32	ap-southeast-1a
Secondary private IPs		Security groups
VPC ID	vpc-6f253308	SSH Access for ALL ips, HTTPS Web Server. view inbound rules . view outbound rules
Subnet ID	subnet-e849b28e	Scheduled events
		No scheduled events
		AMI ID
		ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200112 (ami-09a4a9ce71f3f20b)
		Platform
		-

AWS EC2 – Accessing Instance

- Connect using SSH

- `ssh ubuntu@3.81.11.139 -i path/to/key.pem`

- Copy files to server

- `scp -i path/to/key.pem local/path/file.txt user@host:server/path/file.txt`

MySQL Installation

- Install mysql-server-5.7 with apt-get
 - Leave root password blank
- mysql -u root (not using sudo!)
 - If successful, great!
 - If not:
 - Need to explicitly grant access to “root” user
 - Then restart the server

```
sudo mysql -e 'update mysql.user set plugin = "mysql_native_password" where user="root"'
sudo mysql -e 'create user "root"@"%" identified by ""'
sudo mysql -e 'grant all privileges on *.* to "root"@"%" with grant option'
sudo mysql -e 'flush privileges'
sudo service mysql restart
```

- Now you should see the console

MySQL Hands-on

- Show databases
- Create a database
- Switch to a database
- Configure mysql so it can be accessed remotely.

MySQL connecting remotely

- Change the binding address of the mysql server
 - Edit /etc/mysql/mysql.conf.d/mysqld.cnf
 - Change bind-address = 127.0.0.1
 - To bind-address = 0.0.0.0
 - Save the file and restart mysql service
- Add a new Inbound rule for TCP port 3306 to the security group
- Install mysql client (on your pc)
 - `sudo apt-get install mysql-client`
- Connect to server
 - `mysql -u root -p --host=<PublicIP> --port=3306`

Why not to use default config for mysql

- <https://www.zdnet.com/article/hackers-are-scanning-for-mysql-servers-to-deploy-gandcrab-ransomware/>
- Change default port, secure root user, run mysql_secure_installation
- Implement granular access for users



Exercise

- Create a index.html on ec2 instance
- Create and launch a simple http server on ec2 instance to serve files using python.
- Test whether you can access it on your pc.

AWS: Creating a Snapshot of MySQL server

- Go to EC2 Dashboard and locate VolumeID

The screenshot displays the AWS Management Console interface. On the left, the navigation menu includes sections for EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area shows a list of EC2 instances. One instance is highlighted with a red box, showing its details in a modal window.

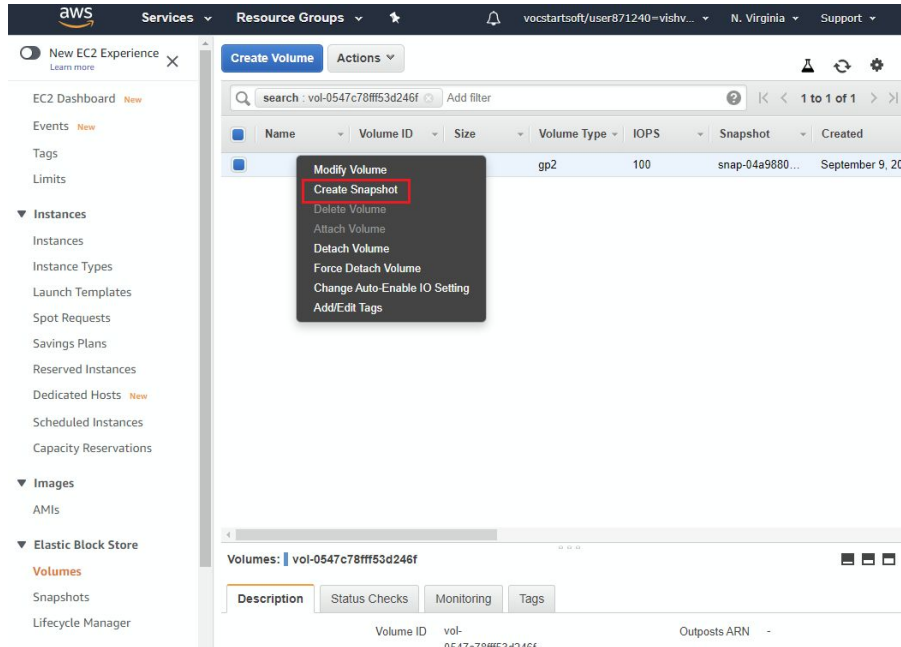
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-0456784854163c990	t2.micro	us-east-1c	running	2/2 checks

Block Device /dev/sda1

EBS ID	vol-0547c78ff53d246f
Root device type	EBS
Attachment time	2020-09-09T02:45:38.000Z
Block device status	attached
Delete on termination	True

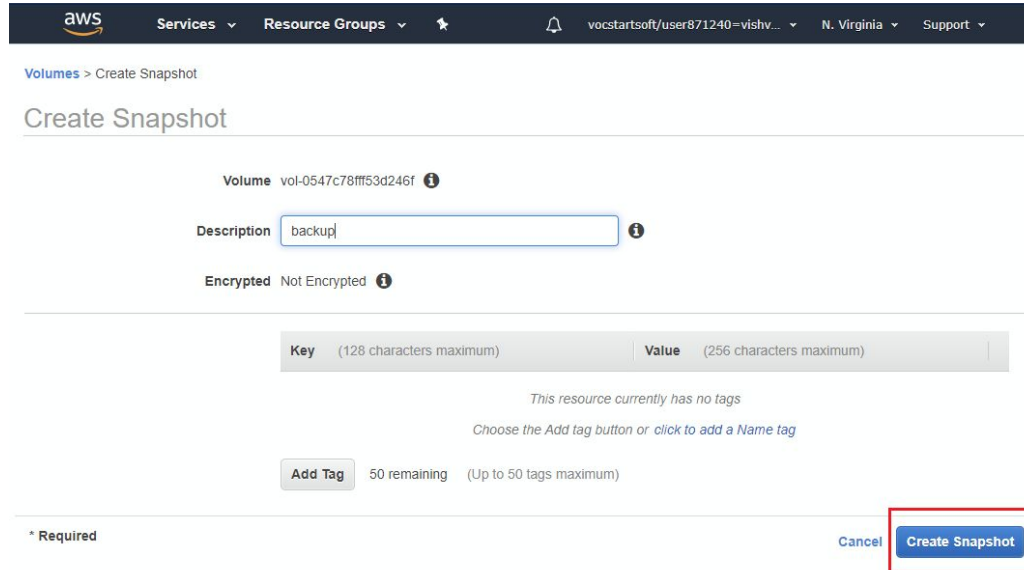
AWS: Creating a Snapshot of MySQL server

- Navigate to Volume dashboard and select create Snapshot from context menu.



AWS: Creating a Snapshot of MySQL server

- Create snapshot



The screenshot shows the AWS Management Console interface for creating a snapshot of an EBS volume. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a user profile dropdown, a region dropdown set to 'N. Virginia', and a 'Support' link. The breadcrumb trail indicates the path 'Volumes > Create Snapshot'. The main heading is 'Create Snapshot'. Below this, the 'Volume' field is populated with 'vol-0547c78ff53d246f'. The 'Description' field contains the text 'backup'. The 'Encrypted' field is set to 'Not Encrypted'. A section for tags is present, showing a table with 'Key' and 'Value' columns, and a message stating 'This resource currently has no tags' with a link to 'Choose the Add tag button or click to add a Name tag'. At the bottom left, there is a note '* Required'. At the bottom right, there are two buttons: 'Cancel' and 'Create Snapshot', with the latter being highlighted by a red rectangular box.

aws Services Resource Groups

Volumes > Create Snapshot

Create Snapshot

Volume vol-0547c78ff53d246f

Description backup

Encrypted Not Encrypted

Key	Value
(128 characters maximum)	(256 characters maximum)

This resource currently has no tags

Choose the Add tag button or click to add a Name tag

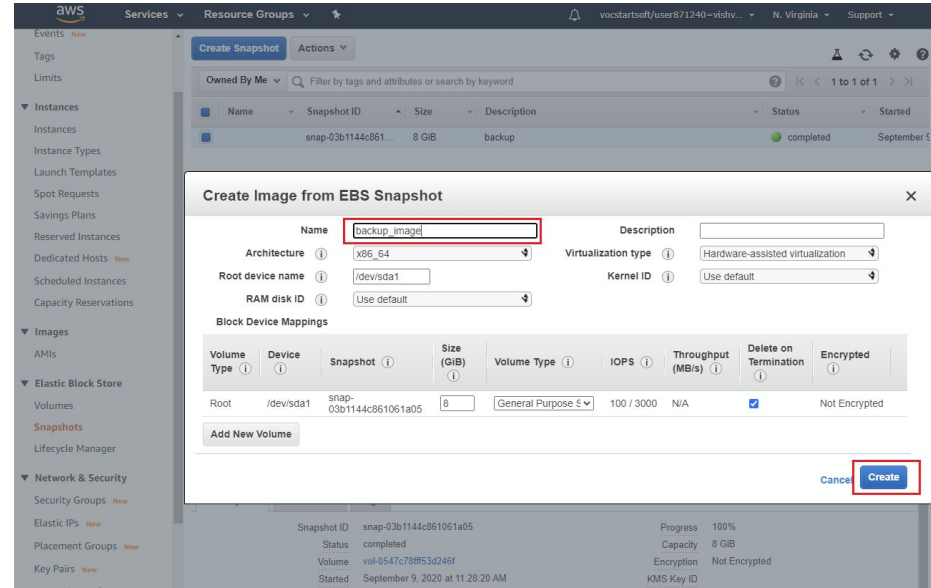
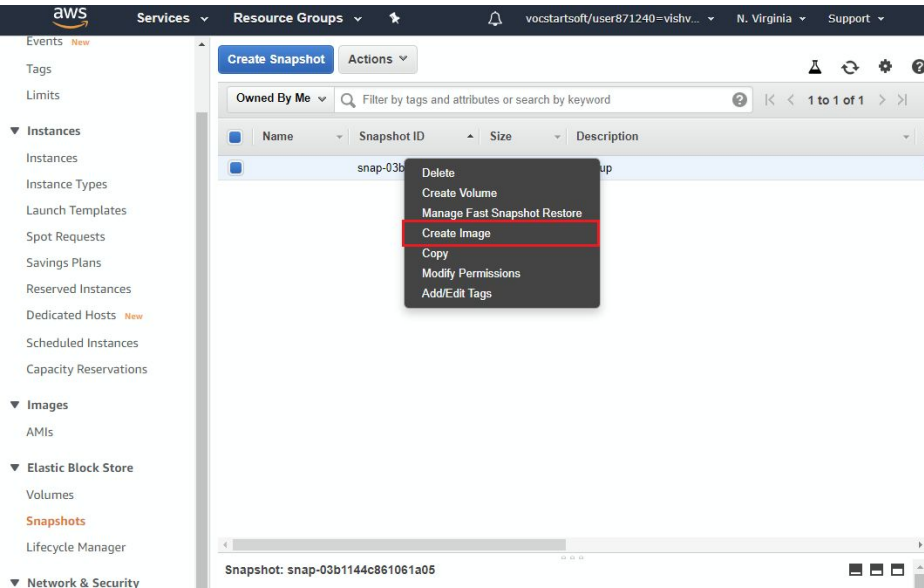
Add Tag 50 remaining (Up to 50 tags maximum)

* Required

Cancel Create Snapshot

AWS: Launching EC2 using snapshot

- Create Image



AWS: Launching EC2 using snapshot

- Launch Image

The screenshot displays the AWS Management Console interface. On the left sidebar, the 'Images' section is expanded, and the 'AMIs' link is highlighted with a red rectangle. The main content area shows a table of AMIs owned by the user. A context menu is open over the first AMI, with the 'Launch' option highlighted. Below the table, the 'Details' tab for the selected AMI is visible, showing its ID, name, owner, and source.

Owned by me	search	ami-0c747ce725628e973	Add filter	1 to 1 of 1			
Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
<input checked="" type="checkbox"/>		ami-0c747ce725628e973	829583823500/...	829583823500	Private	available	September 9, 2021

Context menu options:

- Launch
- Spot Request
- Deregister
- Register New AMI
- Copy AMI
- Modify Image Permissions
- Add/Edit Tags
- Modify Boot Volume Setting
- EC2 Image Builder

Image: ami-0c747ce725628e973

Details | Permissions | Tags

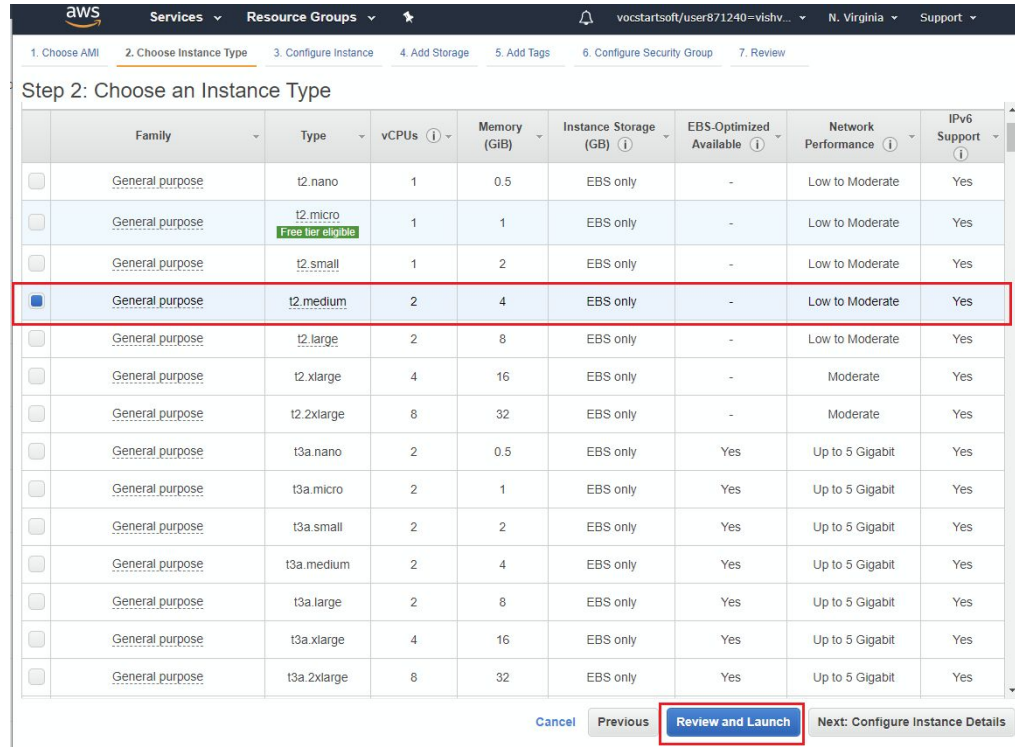
AMI ID: ami-0c747ce725628e973
Owner: 829583823500

AMI Name: backup_image
Source: 829583823500/backup_image

Edit

AWS: Launching EC2 using snapshot

- Launch a different instance type



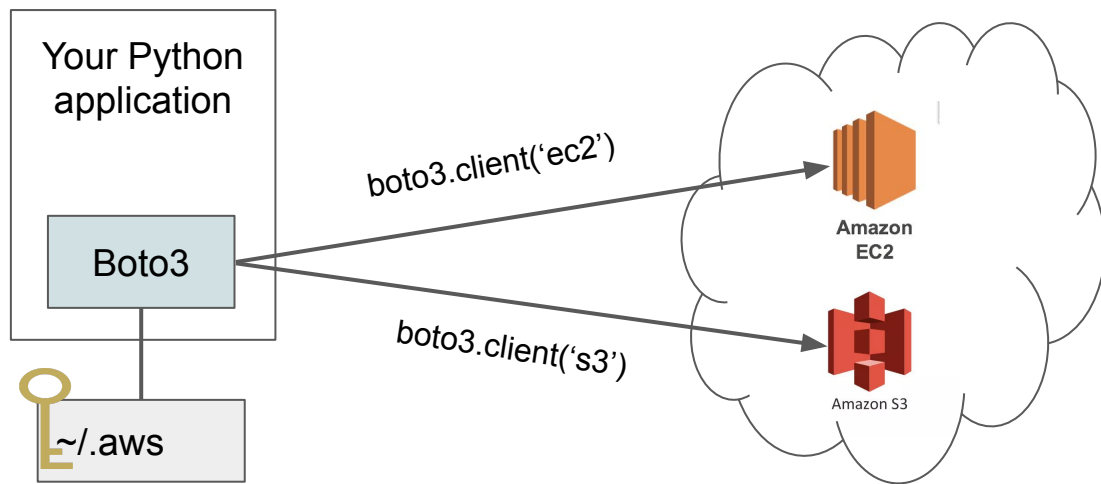
The screenshot shows the AWS Management Console interface for launching an EC2 instance. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The main content area is titled 'Step 2: Choose an Instance Type' and displays a table of available instance types. The 't2.medium' instance type is selected and highlighted with a red box. The 'Review and Launch' button at the bottom right is also highlighted with a red box.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

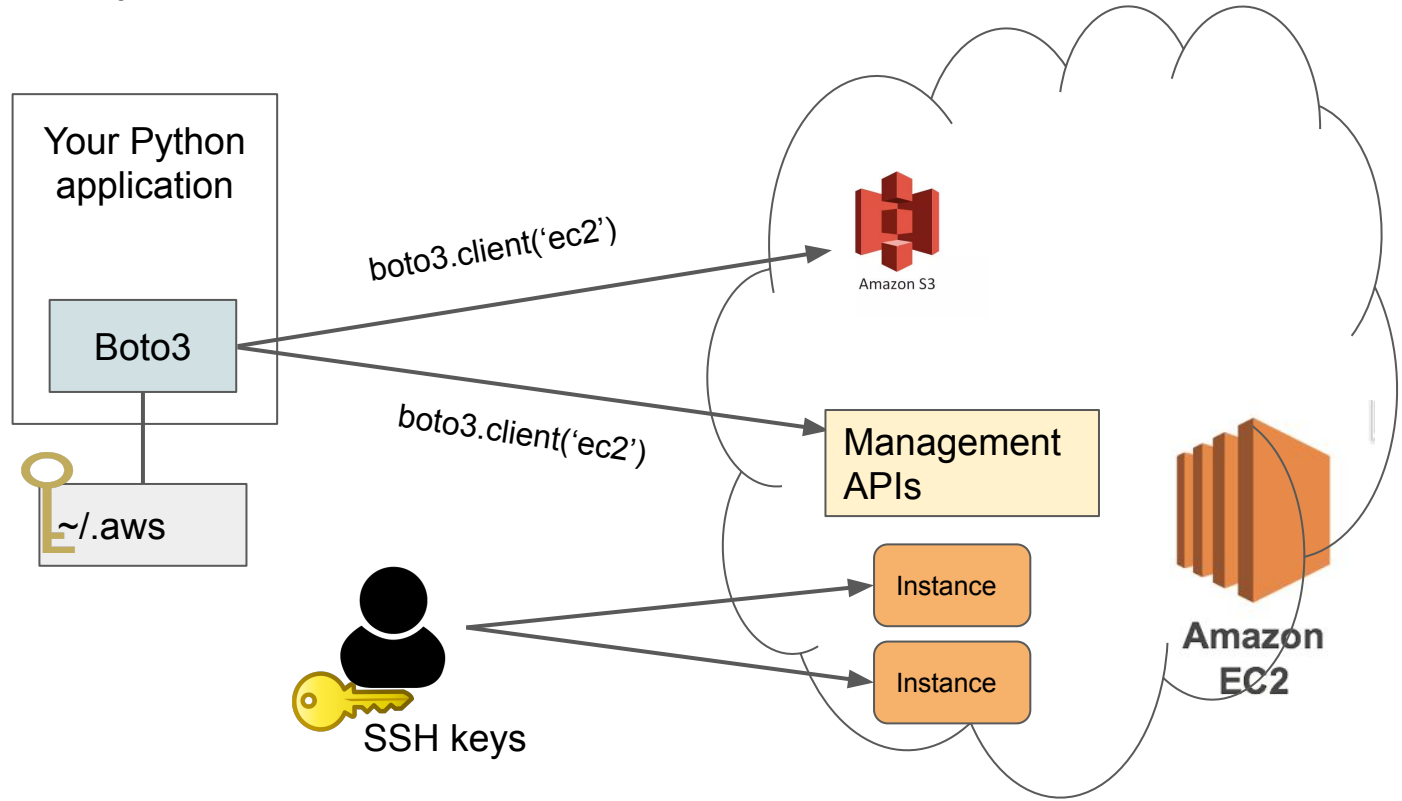
Boto3

- Python library for accessing AWS services



Boto3

Access key vs. Key pair



boto3

- Prerequisites

- Install aws-cli
 - <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-linux.html>
- Configure aws-cli with access keys
 - `aws configure`
- Optional – install python pip module if not already installed
 - `sudo apt-get install python3-pip`

- Setting up boto

- `sudo apt-get update`
- `sudo apt-get install -y python3-pip`
- `python3 -m pip install boto3`

boto3

Create a file using nano and add following.

- nano createEC2.py

- Example: start an instance

```
import boto3

ec2 = boto3.resource('ec2')

# create a new EC2 instance

instances = ec2.create_instances(

    ImageId='ami-06b263d6ceff0b3dd',

    MinCount=1,

    MaxCount=1,

    InstanceType='t2.micro',

    KeyName='test-vm'

)
```

- Example: start an instance (aws educate)

```
import boto3

session = boto3.session.Session(aws_access_key_id='',
aws_secret_access_key='', aws_session_token='', region_name='')

ec2 = session.resource('ec2')

# create a new EC2 instance

instances = ec2.create_instances(

    ImageId='ami-06b263d6ceff0b3dd',

    MinCount=1,

    MaxCount=1,

    InstanceType='t2.micro',

    KeyName='test-vm'

)
```