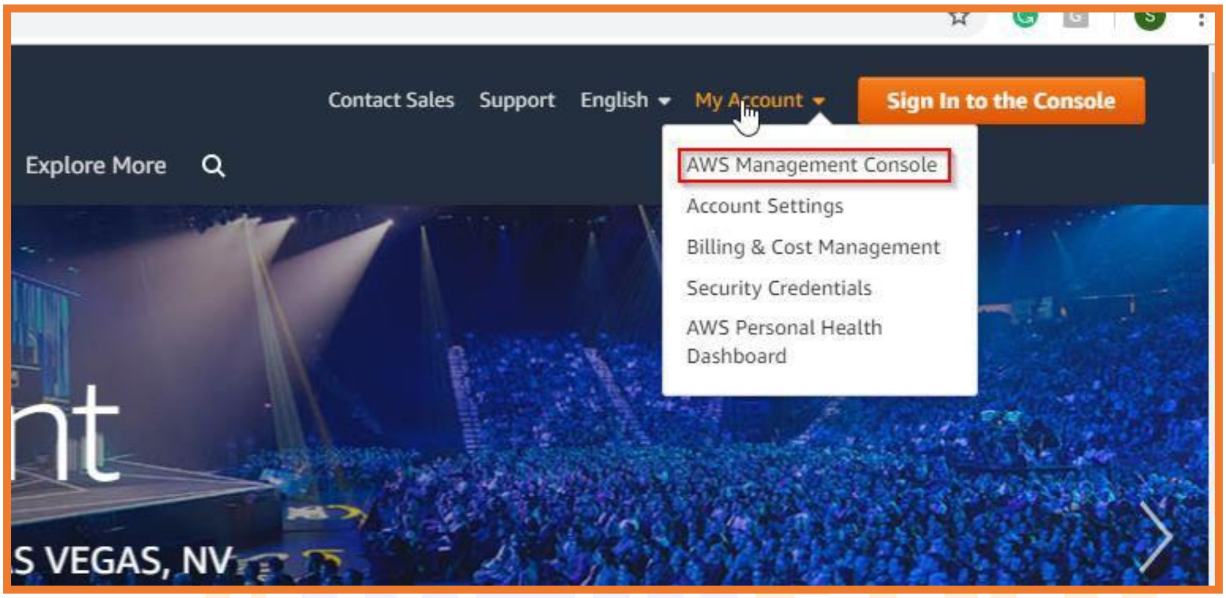
PROJECT 1 - SOLUTION

Connect your system with your EC2 Instance

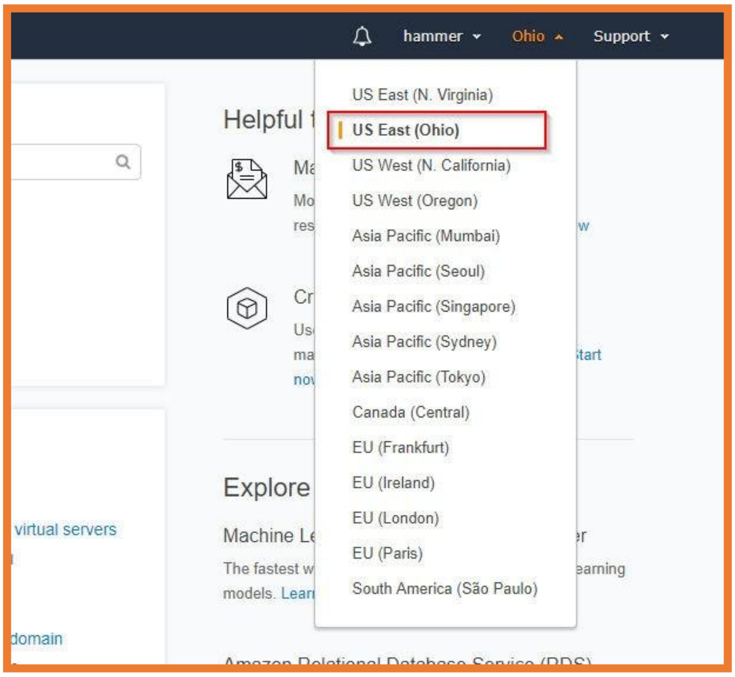
First you need to install **Putty** on your system and then connect it with your EC2 instance.

Below are the steps for it:

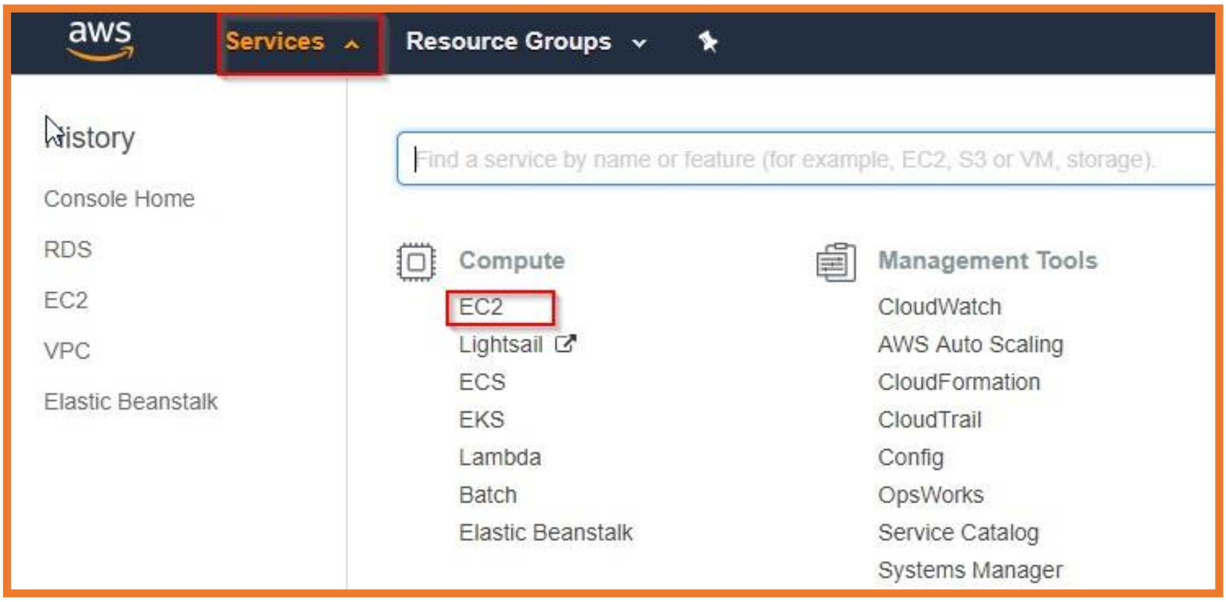
- First sign into the AWS Management Console.



- Select any region you want, like we've selected Ohio here



- In the **Services** section, you must see **Compute** where you need to choose **EC2**



aws Services Resource Groups

History

Console Home

RDS

EC2

VPC

Elastic Beanstalk

Find a service by name or feature (for example, EC2, S3 or VM, storage).

Compute

EC2

Lightsail

ECS

EKS

Lambda

Batch

Elastic Beanstalk

Management Tools

CloudWatch

AWS Auto Scaling

CloudFormation

CloudTrail

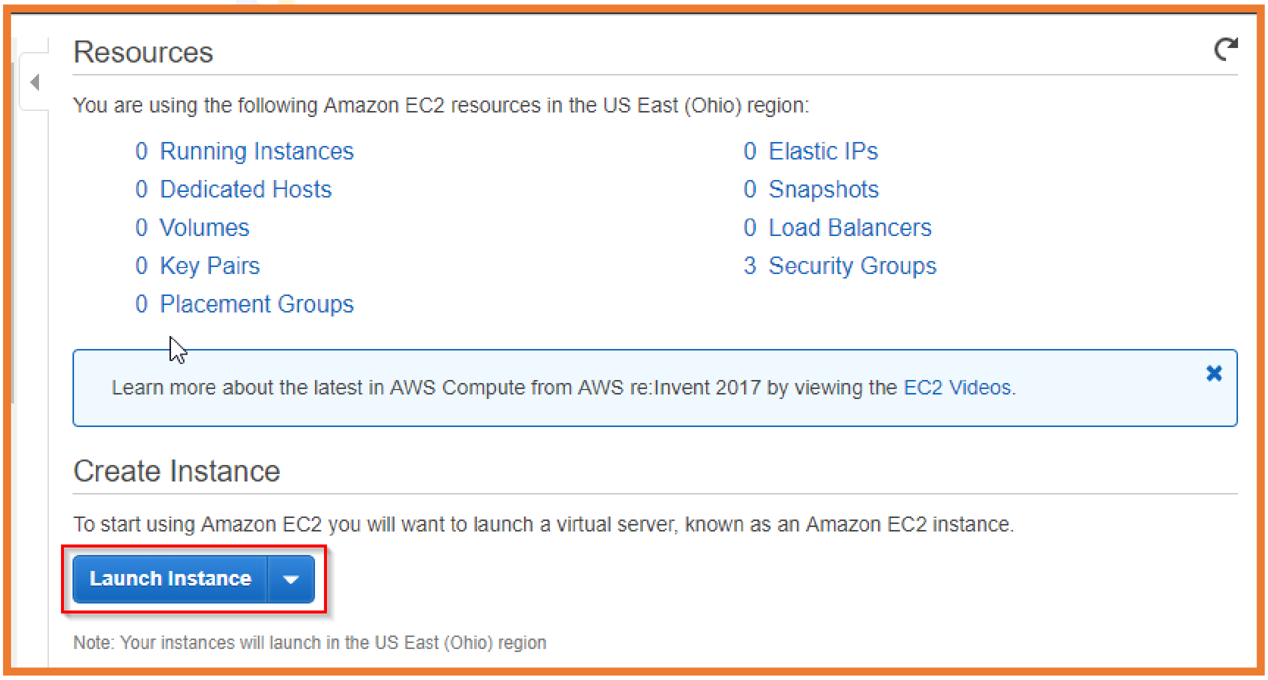
Config

OpsWorks

Service Catalog

Systems Manager

- Then you will see in the Create section, there is **Launch Instance** option, select it



Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

0 Running Instances

0 Dedicated Hosts

0 Volumes

0 Key Pairs

0 Placement Groups

0 Elastic IPs

0 Snapshots

0 Load Balancers

3 Security Groups

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.

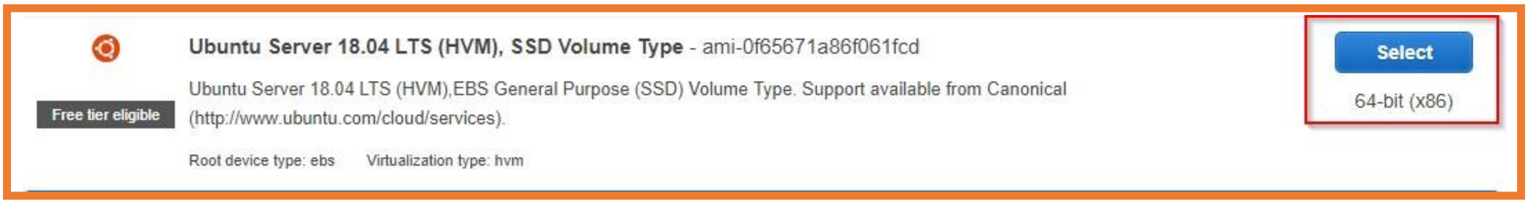
Create Instance

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

Launch Instance

Note: Your instances will launch in the US East (Ohio) region

- Then Select an **AMI** or **Amazon Machine Image**



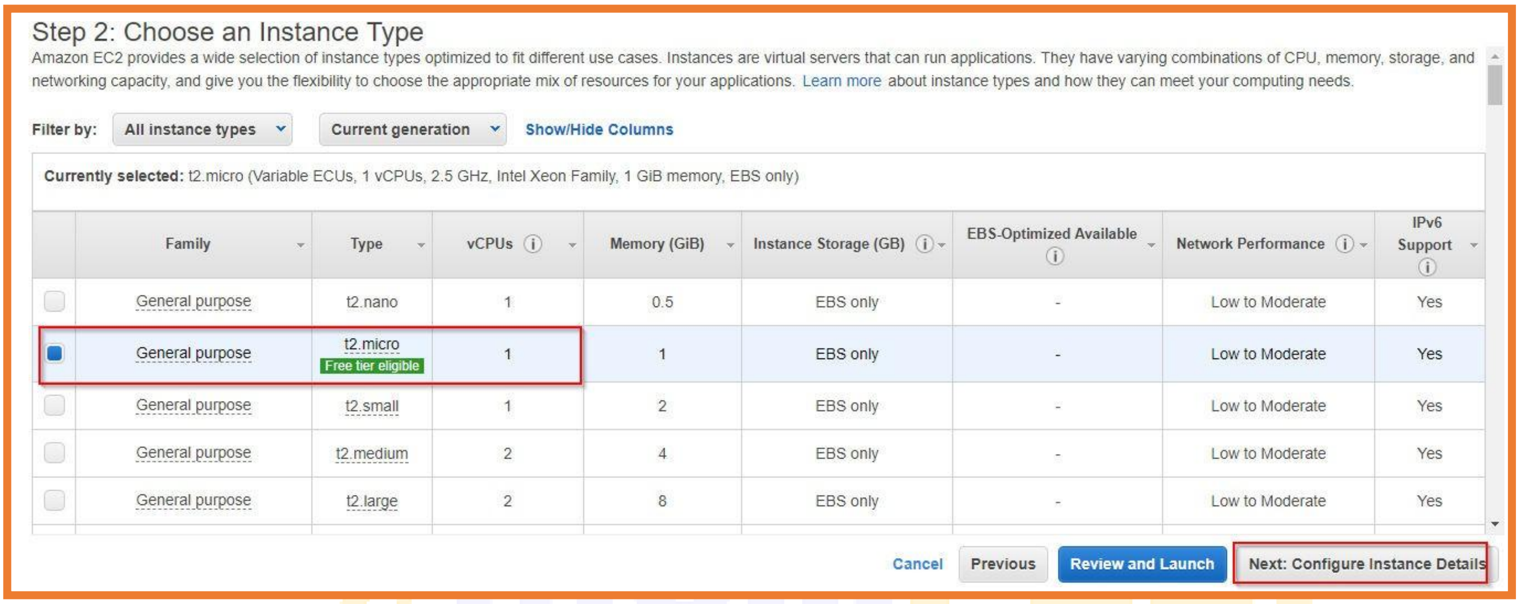
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0f65671a86f061fcd

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

- Choose your instance type, we're choosing Free tier for the demo purpose



Step 2: Choose an Instance Type

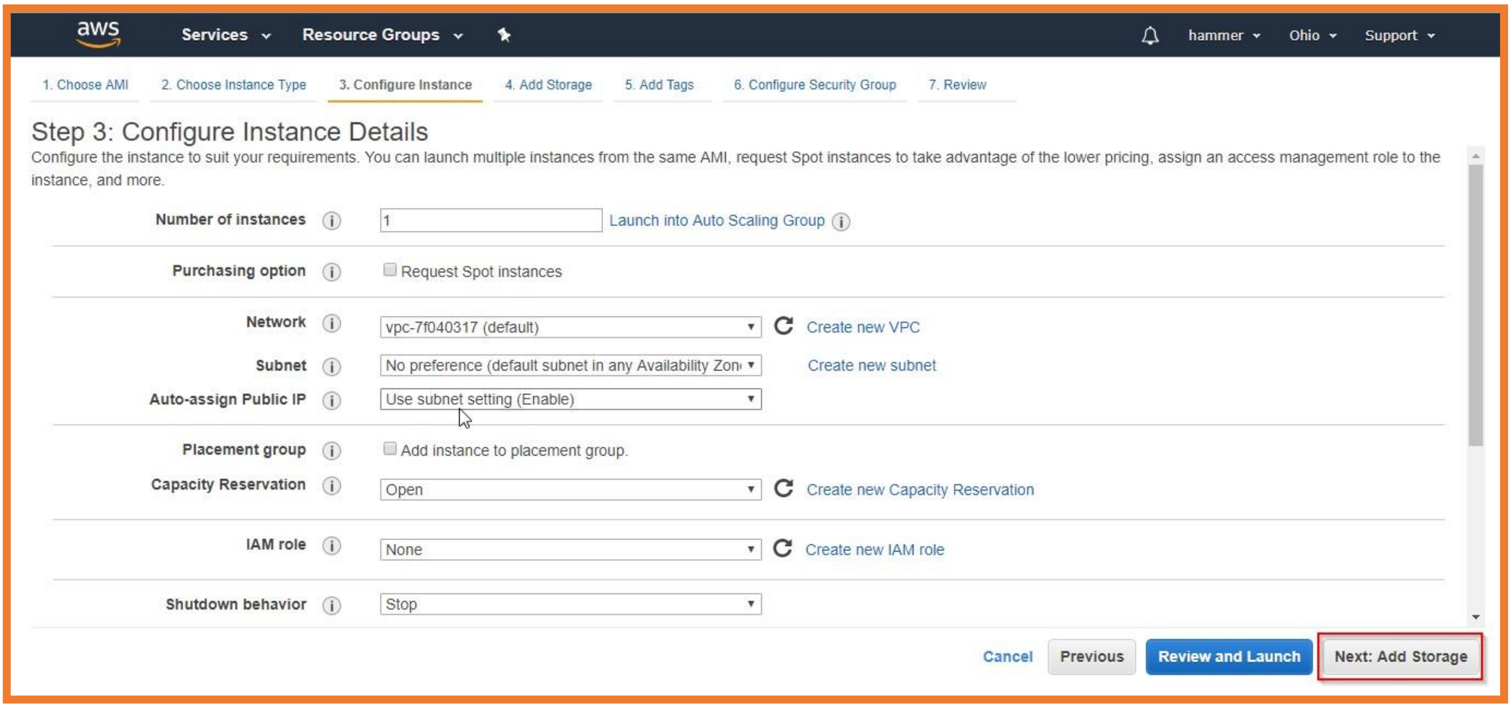
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources 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 (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

- Next step is to configure your instance details and then there will and **Add storage** option, select it



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-7f040317 (default) — Create new VPC

Subnet: No preference (default subnet in any Availability Zon — Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

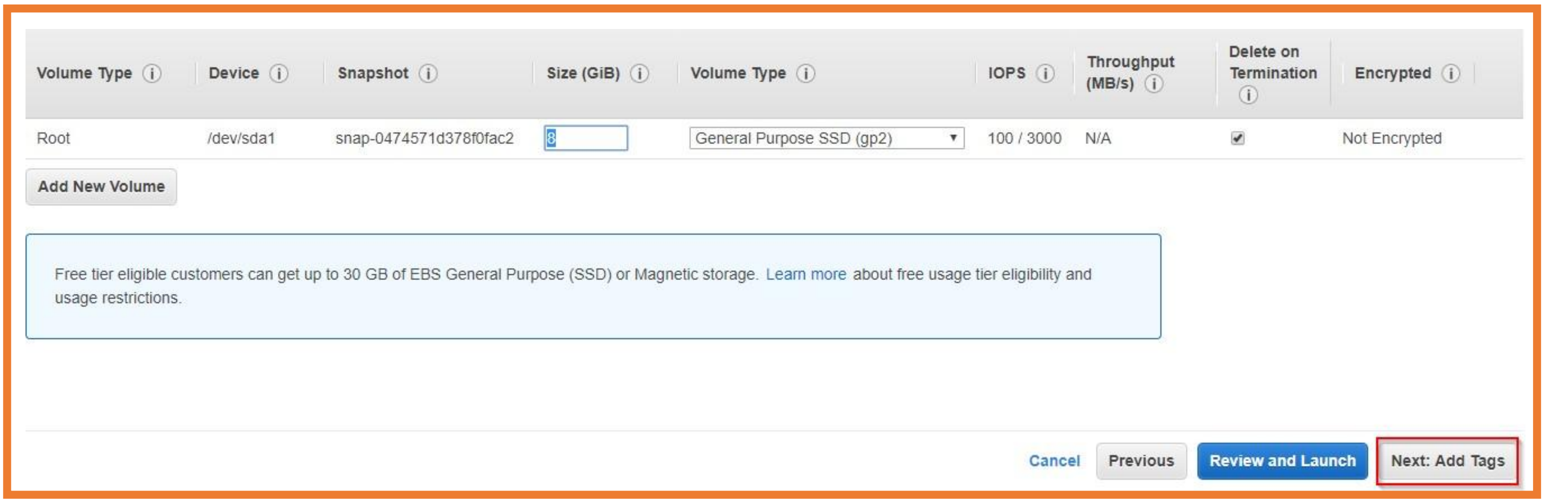
Placement group: ☐ Add instance to placement group.

Capacity Reservation: Open — Create new Capacity Reservation

IAM role: None — Create new IAM role

Shutdown behavior: Stop

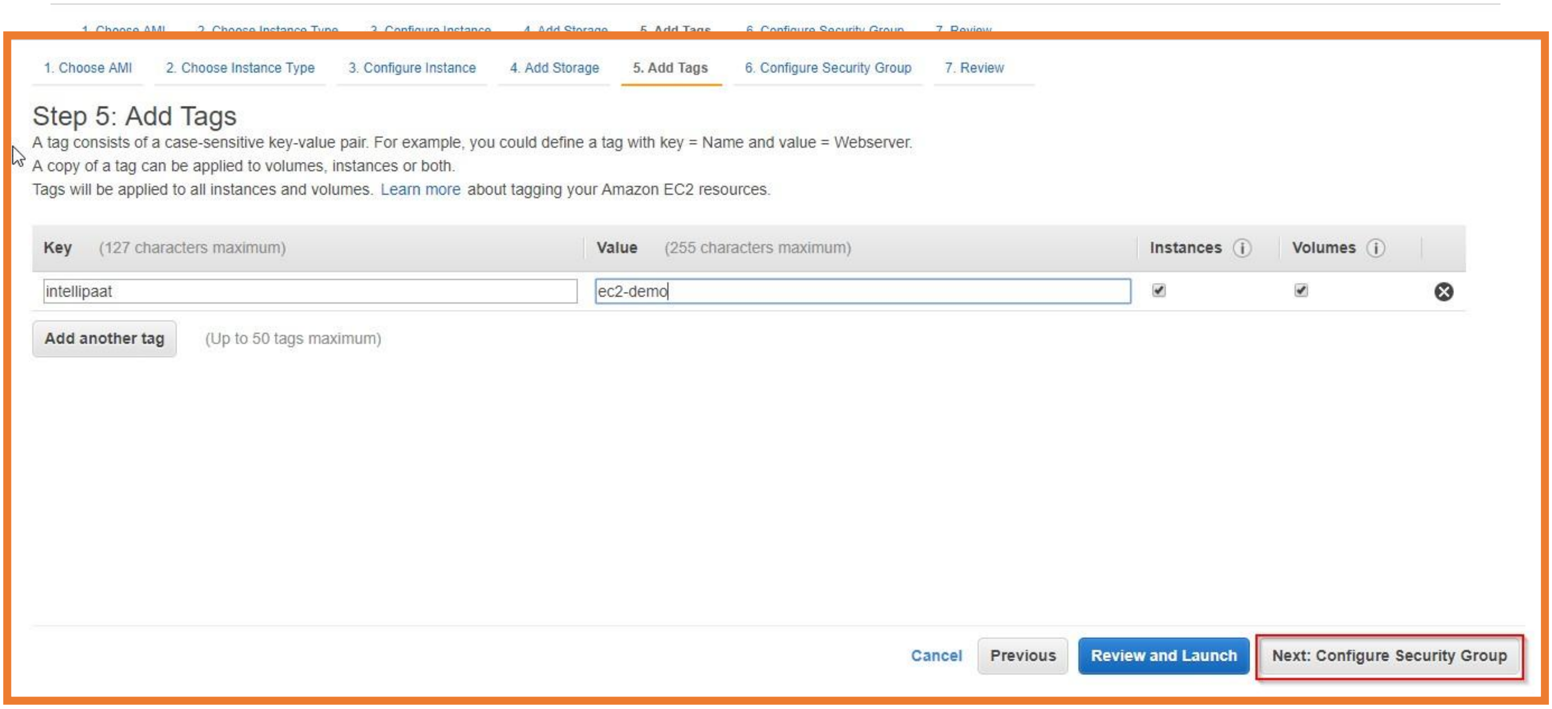
- Then click on Add Tags



Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0474571d378f0fac2	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

- Add tags then name the key and a value, click Configure Security Group

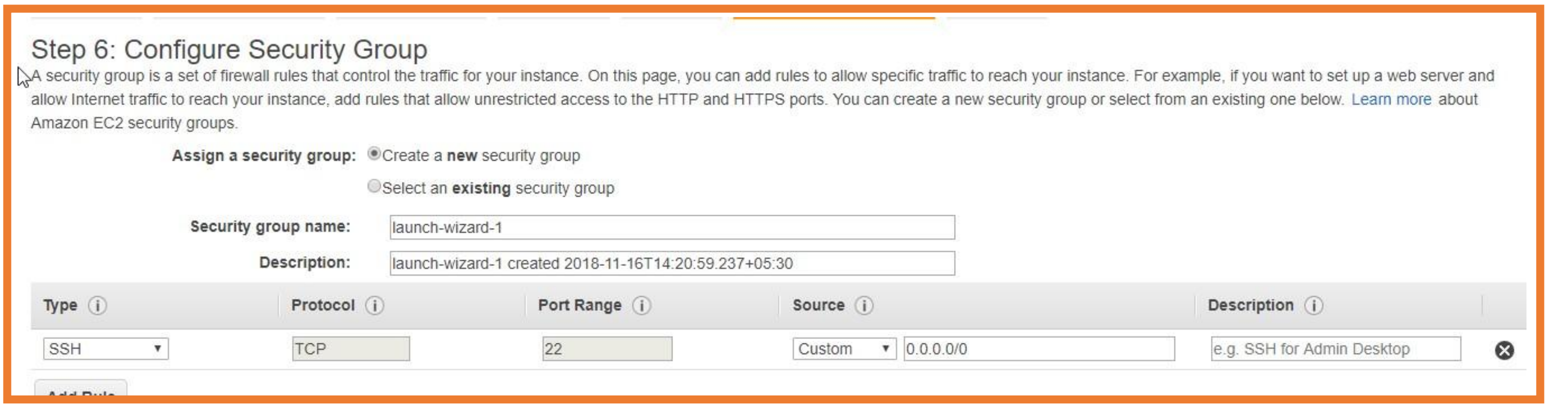


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
intellipaat	ec2-demo	☑	☑

- Keep the configuration of security group as it



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

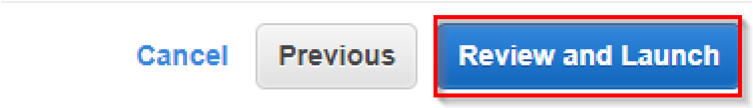
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-1

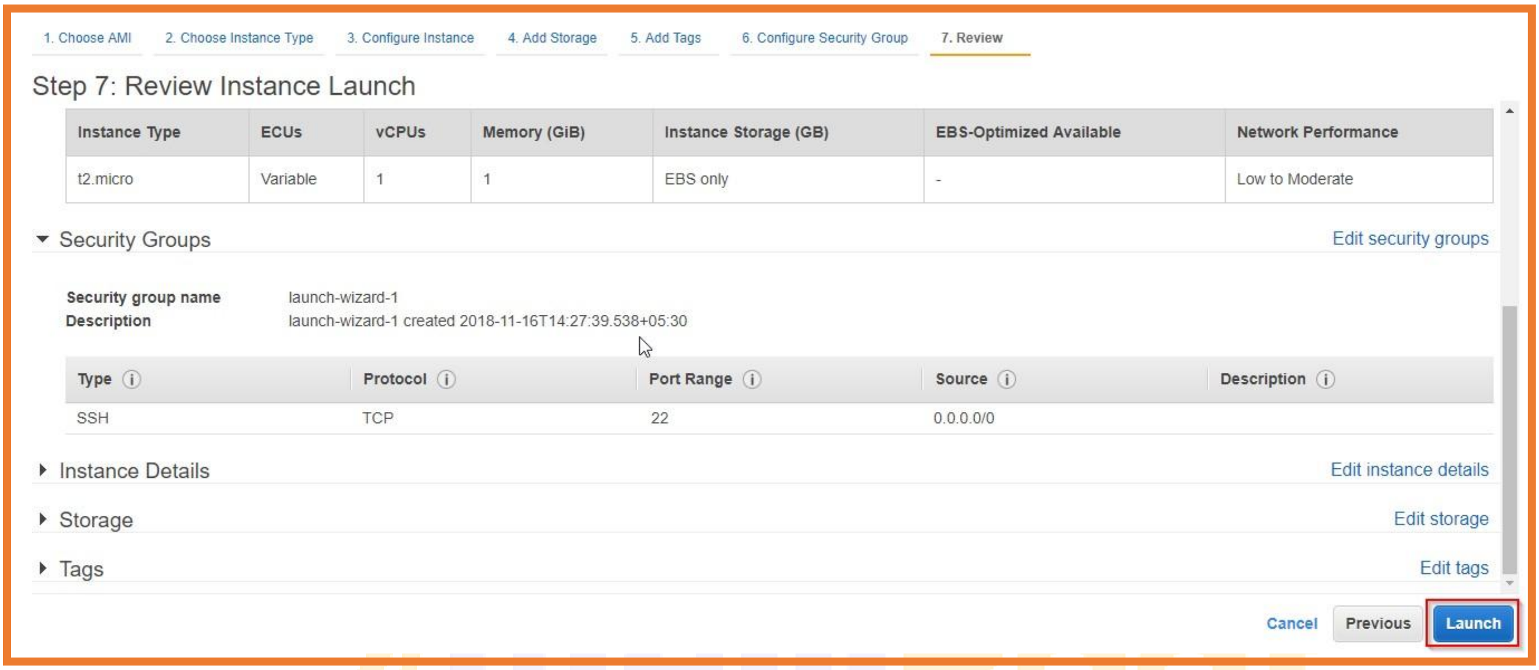
Description: launch-wizard-1 created 2018-11-16T14:20:59.237+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

- Then click **Review & Launch**



- Then directly **Launch** it



Step 7: Review Instance Launch

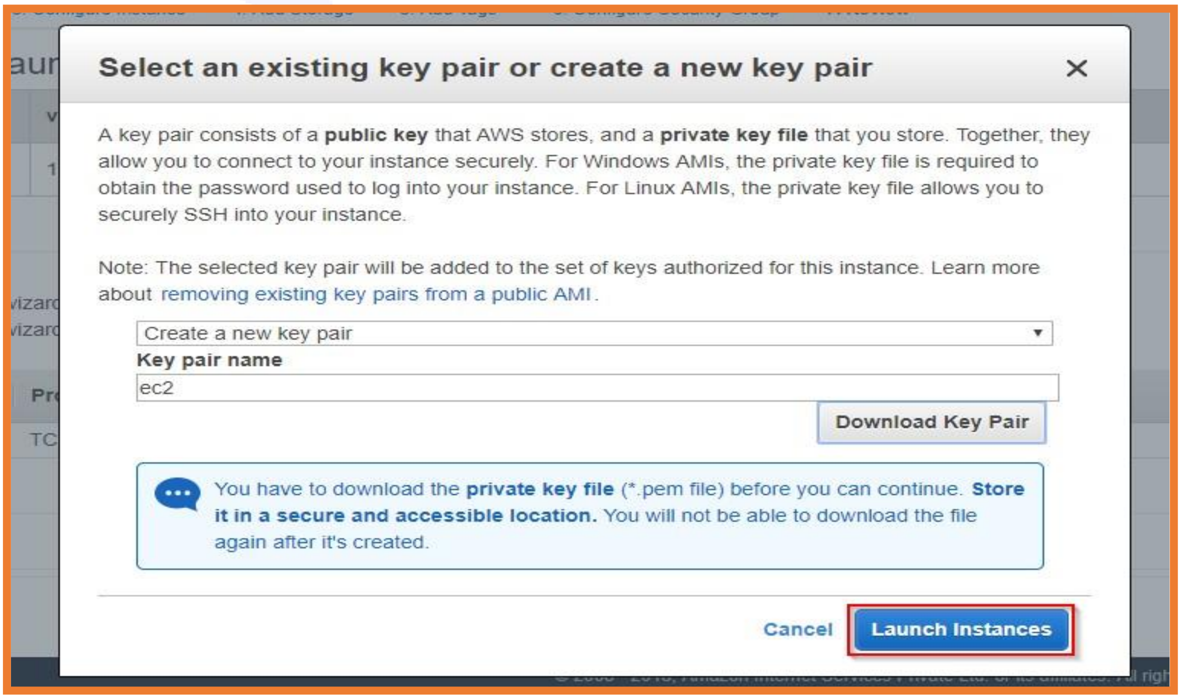
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name launch-wizard-1
Description launch-wizard-1 created 2018-11-16T14:27:39.538+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

- Then Create a key pair, **download** it and **then Launch your instance**


Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

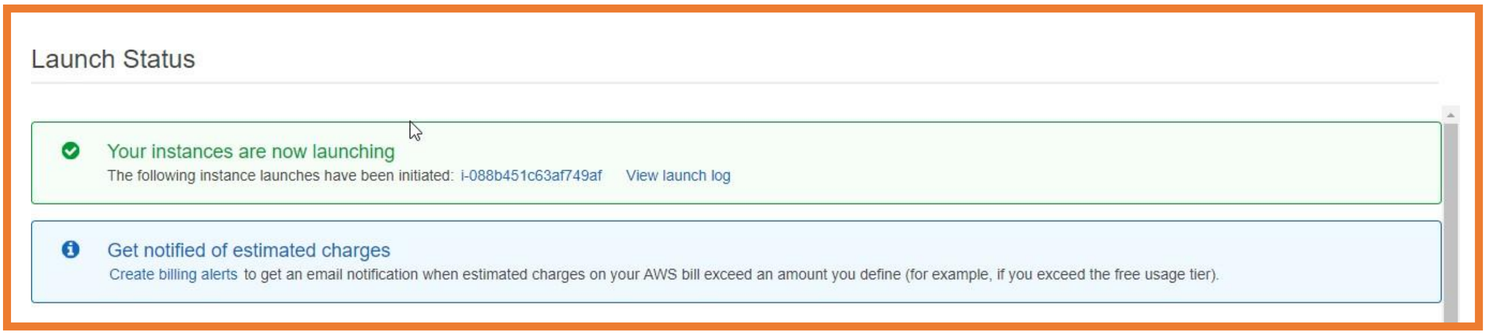
Create a new key pair

Key pair name

ec2

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.

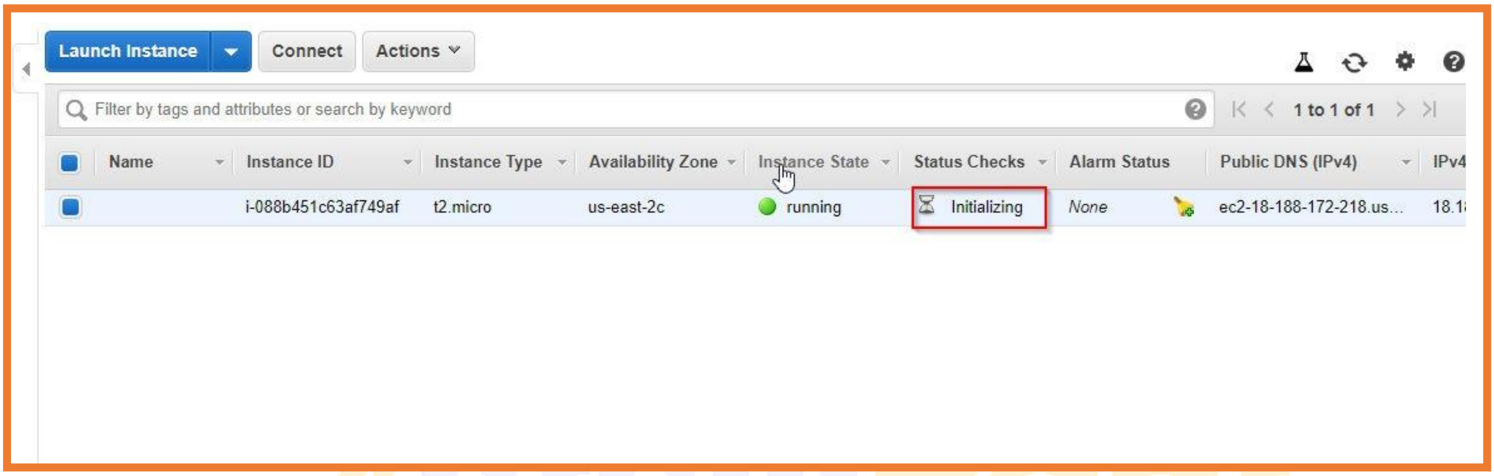
- Status
  You will be able to see in your status that your Instance is on Initializing stage



Launch Status

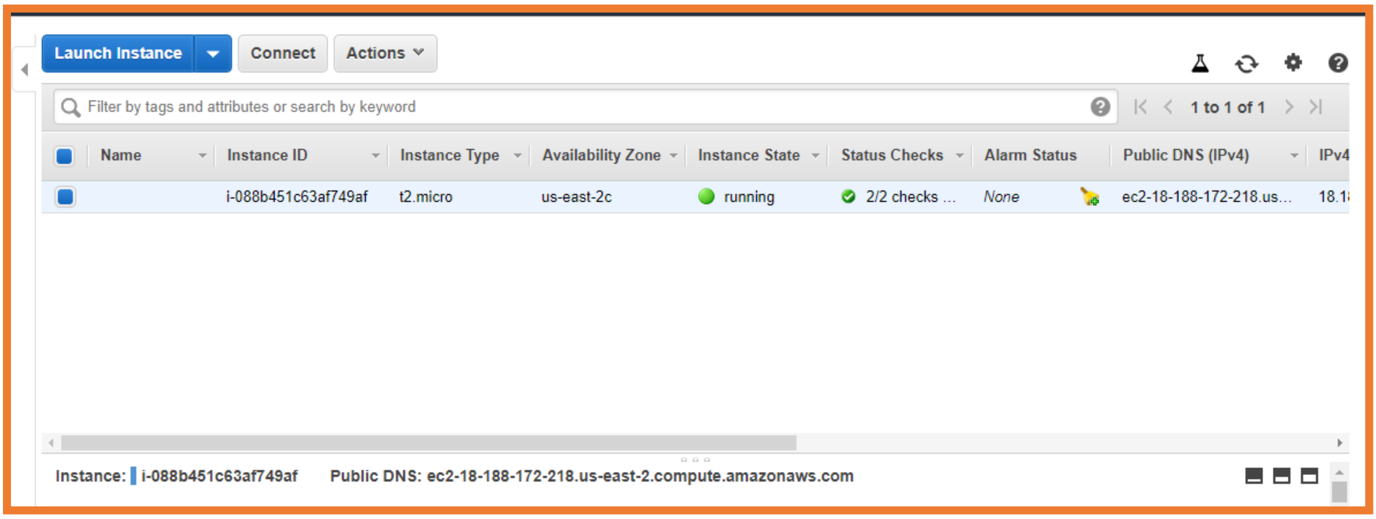
✓ Your instances are now launching
The following instance launches have been initiated: i-088b451c63af749af View launch log

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
■		i-088b451c63af749af	t2.micro	us-east-2c	● running	Initializing	None	ec2-18-188-172-218.us...	18.1

- Then after few minutes, you will see that now your instance is in running stage



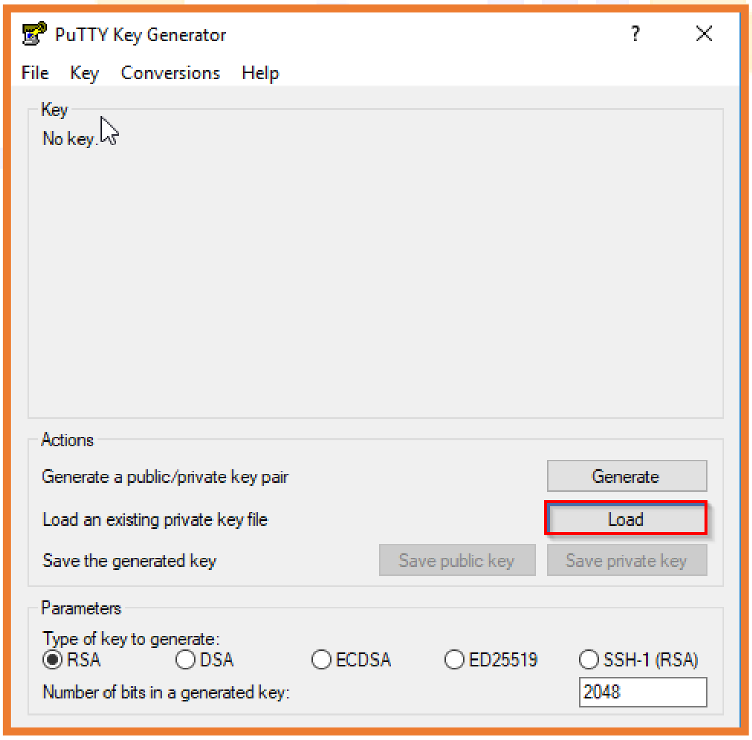
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
	i-088b451c63af749af	t2.micro	us-east-2c	● running	✓ 2/2 checks ...	None	ec2-18-188-172-218.us...	18.1

Instance: i-088b451c63af749af   Public DNS: ec2-18-188-172-218.us-east-2.compute.amazonaws.com

- Now it's time to convert your private key using PuTTYgen

  PuTTY won't be able to support this .pem file, so you'd require a PuTTY gen tool which can convert your .pem file into .ppk format, because you need a .ppk file in order to connect it with your instance

- Click Load in your PuTTY gen

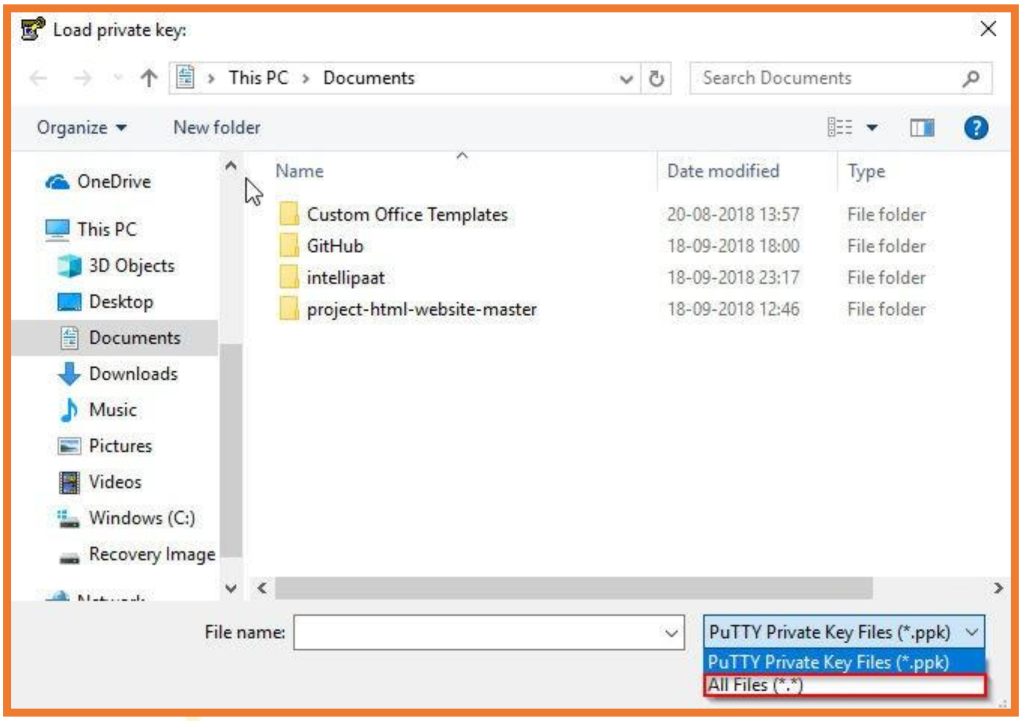


Key
No key.

Actions
Generate a public/private key pair
Load an existing private key file
Save the generated key

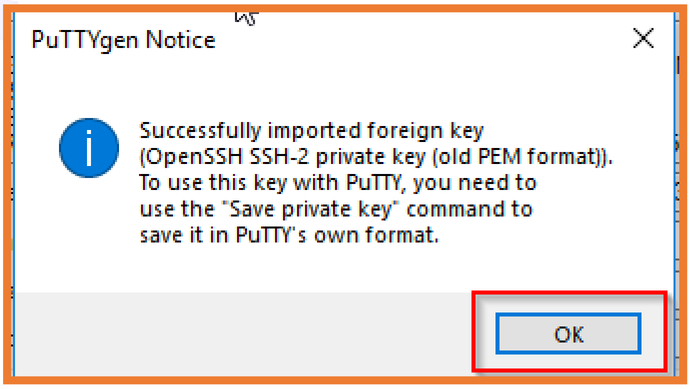
Parameters
Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)
Number of bits in a generated key: 2048

- PuTTY key gen always shows the .ppk format file, so go to the right bottom bar and select the All files option as shown below



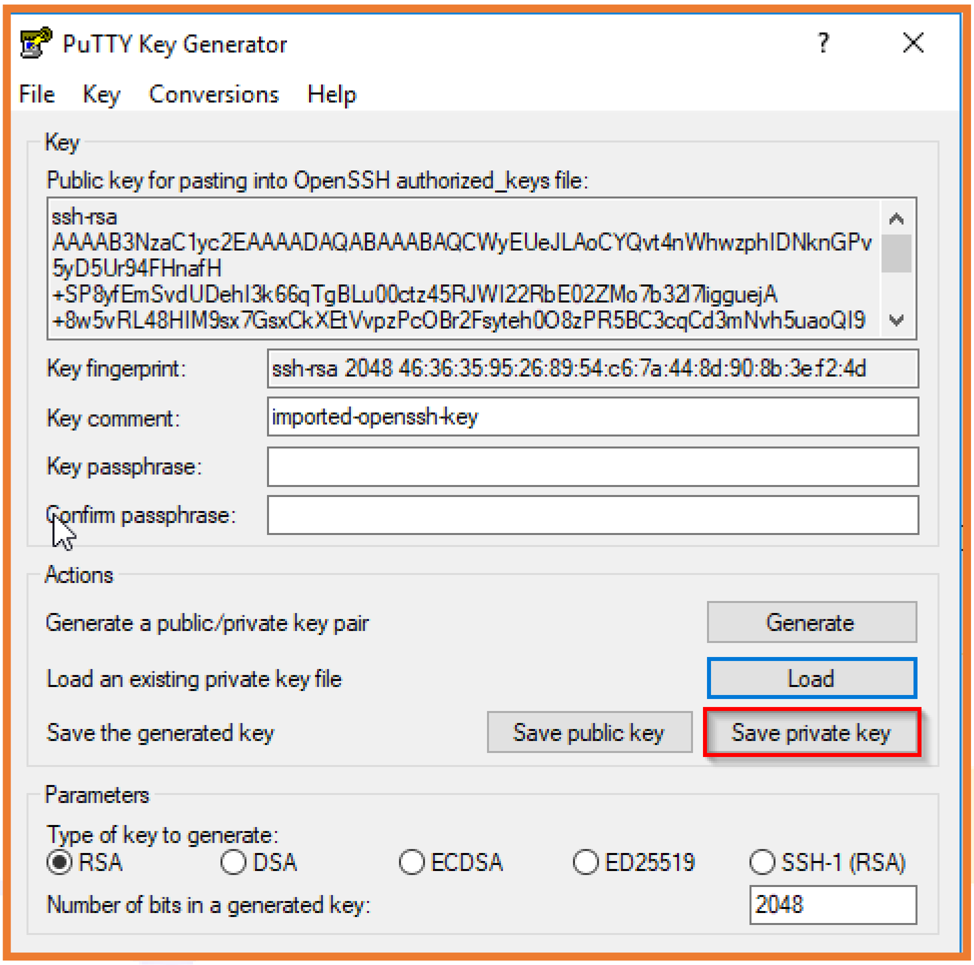
Name	Date modified	Type
Custom Office Templates	20-08-2018 13:57	File folder
GitHub	18-09-2018 18:00	File folder
intellipaat	18-09-2018 23:17	File folder
project-html-website-master	18-09-2018 12:46	File folder

- Then select the folder where you downloaded this keypair and load it there
- You will see this option then click OK


PuTTYgen Notice

Successfully imported foreign key (OpenSSH SSH-2 private key (old PEM format)). To use this key with PuTTY, you need to use the "Save private key" command to save it in PuTTY's own format.

- Then click on Save the Private key, PuTTY gen will give a warning about saving the key without Key passphrase, click Yes and specify the same name for your file that you gave it in the key pair



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCWyEUeJLAoCYQvt4nWhwzphIDNknGPv
5yD5Ur94FHnafH
+SP8yfEmSvdUDehI3k66qTgBLu00ctz45RJWI22RbE02ZMo7b32l7ligguejA
+8w5vRL48HIM9sx7GsxCkXEtVvpzPcOBr2Fsyteh0O8zPR5BC3cqCd3mNvh5uaoQI9
```

Key fingerprint: ssh-rsa 2048 46:36:35:95:26:89:54:c6:7a:44:8d:90:8b:3e:f2:4d

Key comment: imported-openssh-key

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

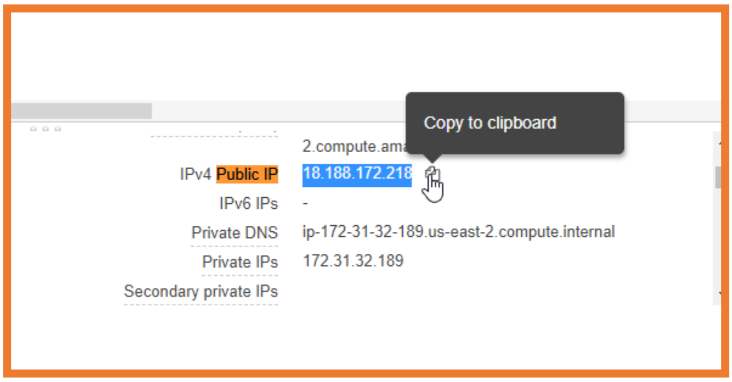
Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

- Now you will see that in your folder, the .ppk file is already added with that name you had given (in our case, it’s ec2)

Connecting to your EC2 Instance using SSH & PuTTY

- First open PuTTY.exe then in the Host Name box, add the Public IP of your Instance



2.compute.ama

IPv4 Public IP 18.188.172.218

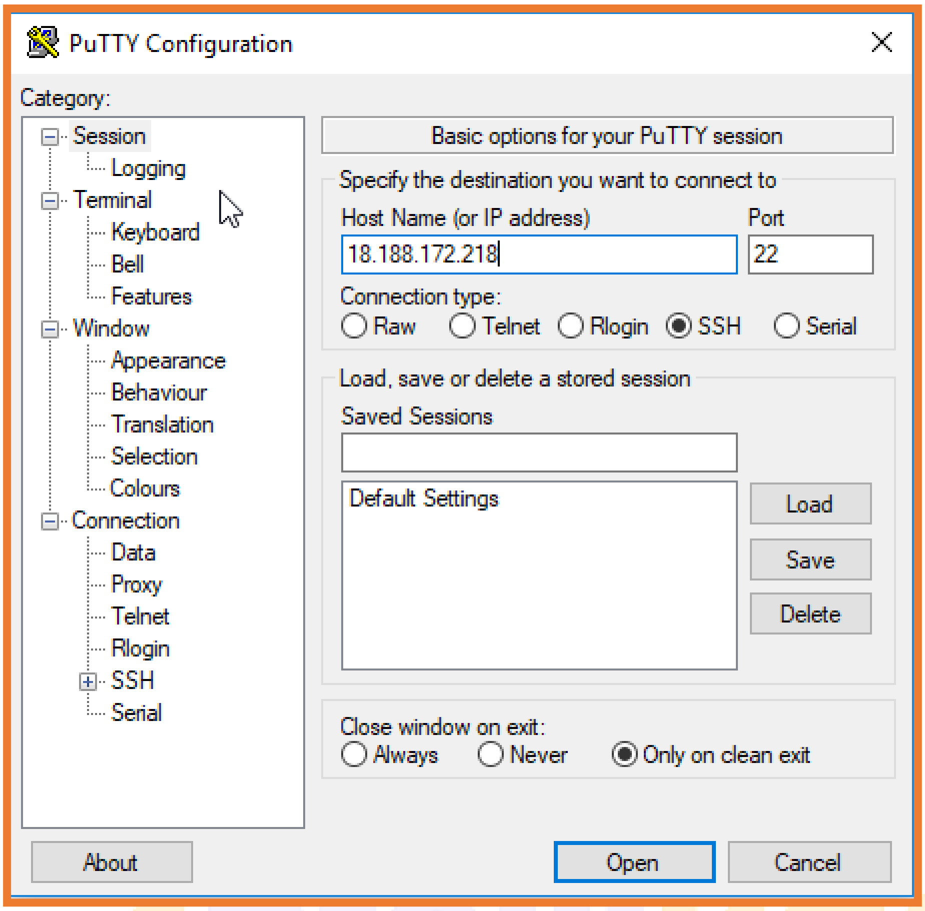
IPv6 IPs -

Private DNS ip-172-31-32-189.us-east-2.compute.internal

Private IPs 172.31.32.189

Secondary private IPs

- Copy paste this Public IP in your PuTTY Hostname



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 18.188.172.218

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

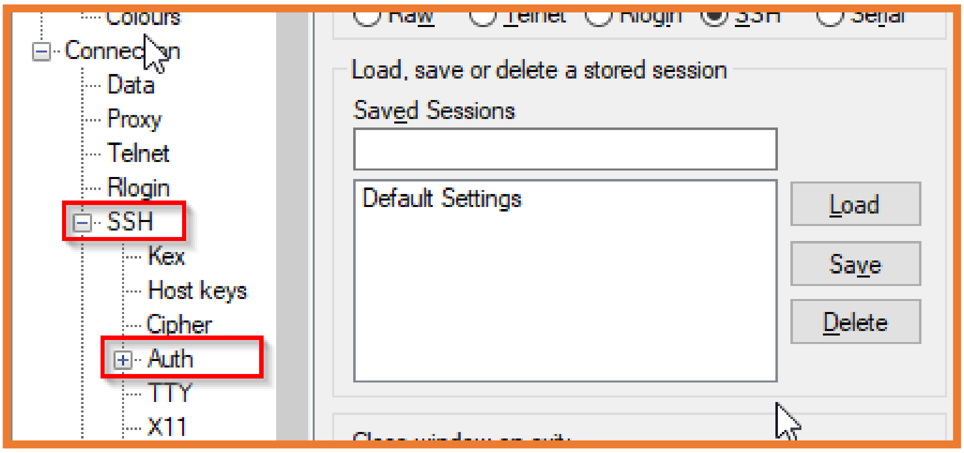
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

- Then in the category list, expand the SSH and Click on AUTH (but don't expand it)



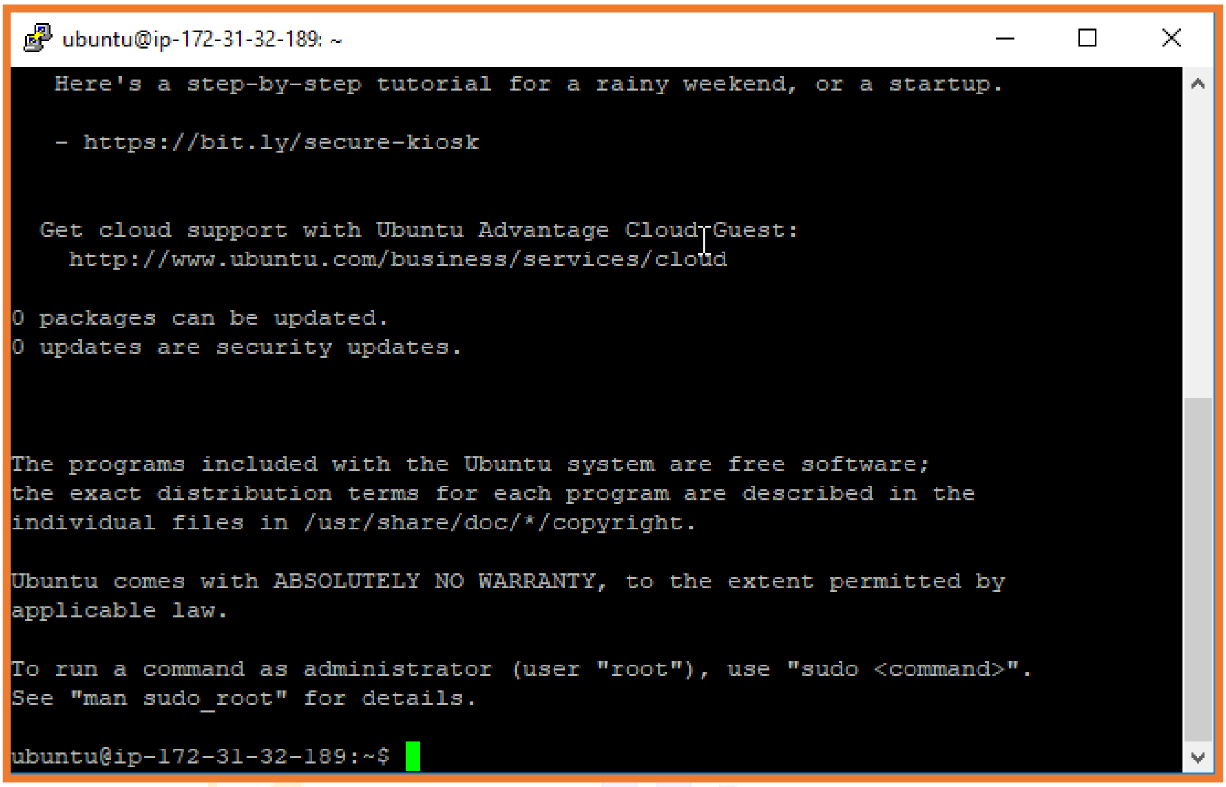
Load, save or delete a stored session

Saved Sessions

Default Settings

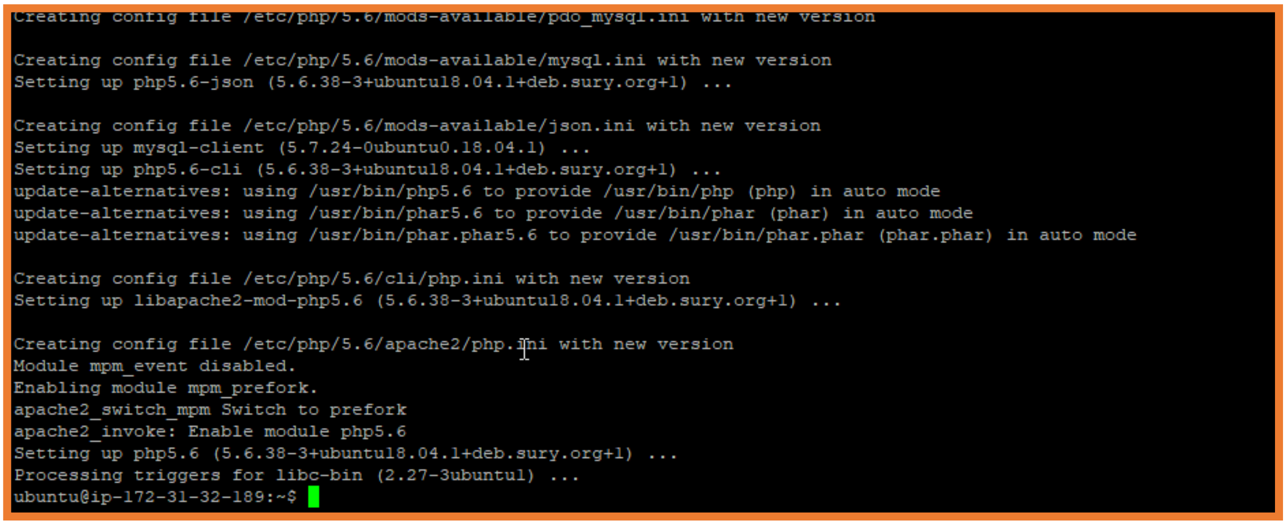
- Then Click Open

- Login as per your OS, in our case it is ubuntu, so we will **Login as: Ubuntu**



```
   Here's a step-by-step tutorial for a rainy weekend, or a startup.

   - https://bit.ly/secure-kiosk


  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-32-189:~$
```

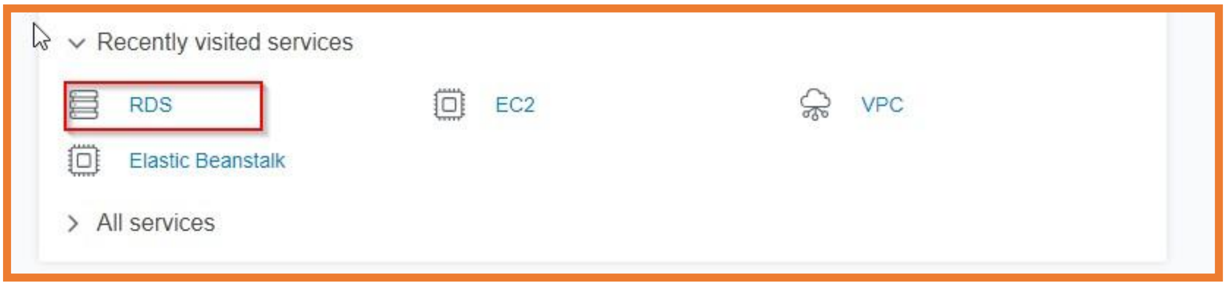
- First Update your system using the command
  sudo apt-get update
- Then use this command in PuTTY to install Apache2
  sudo apt-get install apache2
- Then install php-mysql using the following command
  sudo add-apt-repository -y ppa:ondrej/php
  sudo apt install php5.6 mysql-client php5.6-mysqli
  Now everything is updated in your system



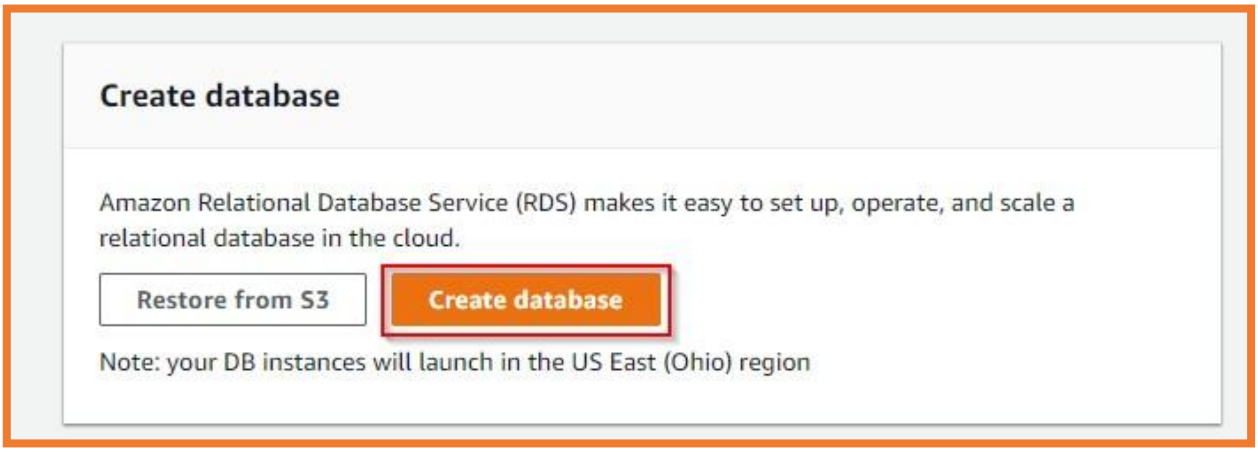
```
Creating config file /etc/php/5.6/mods-available/pdo_mysql.ini with new version

Creating config file /etc/php/5.6/mods-available/mysql.ini with new version
Setting up php5.6-json (5.6.38-3+ubuntu18.04.1+deb.sury.org+1) ...

Creating config file /etc/php/5.6/mods-available/json.ini with new version
Setting up mysql-client (5.7.24-0ubuntu0.18.04.1) ...
Setting up php5.6-cli (5.6.38-3+ubuntu18.04.1+deb.sury.org+1) ...
update-alternatives: using /usr/bin/php5.6 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar5.6 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar5.6 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/5.6/cli/php.ini with new version
Setting up libapache2-mod-php5.6 (5.6.38-3+ubuntu18.04.1+deb.sury.org+1) ...

Creating config file /etc/php/5.6/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php5.6
Setting up php5.6 (5.6.38-3+ubuntu18.04.1+deb.sury.org+1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@ip-172-31-32-189:~$
```

Now we connect mysql with the RDS

- Go to your AWS Management Console
- Select **RDS**



- Then click on **Create Database**

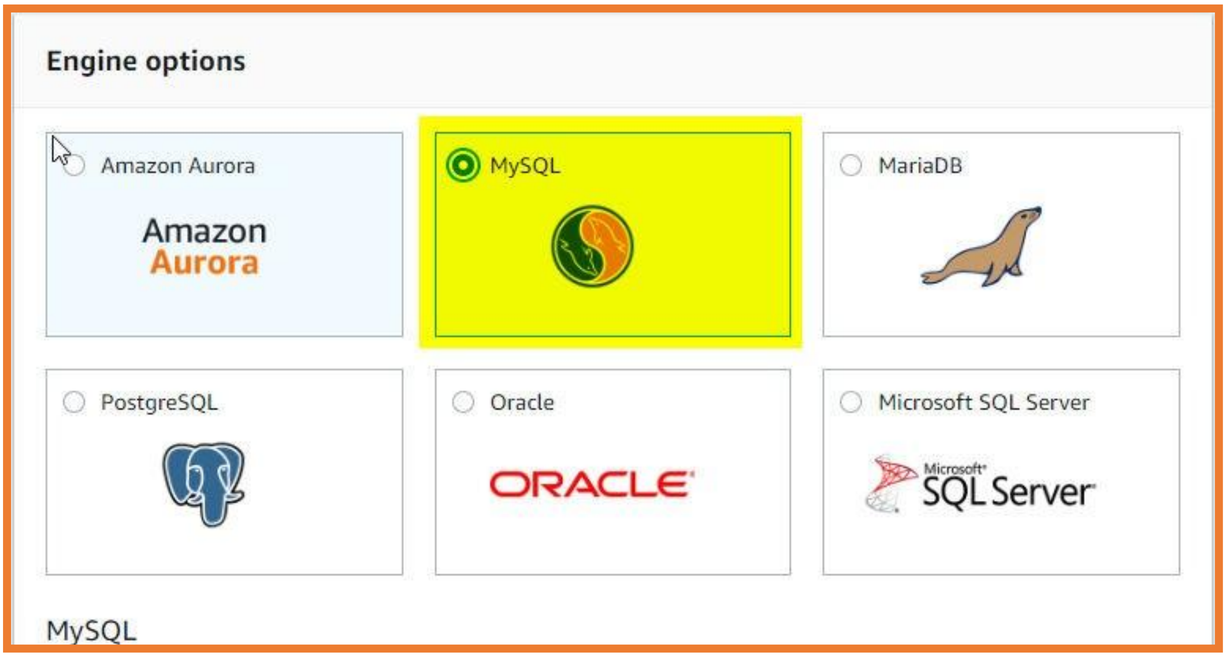


Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Note: your DB instances will launch in the US East (Ohio) region

- Select the MySQL Engine and click **Next**



Engine options

○ Amazon Aurora

● MySQL

○ MariaDB

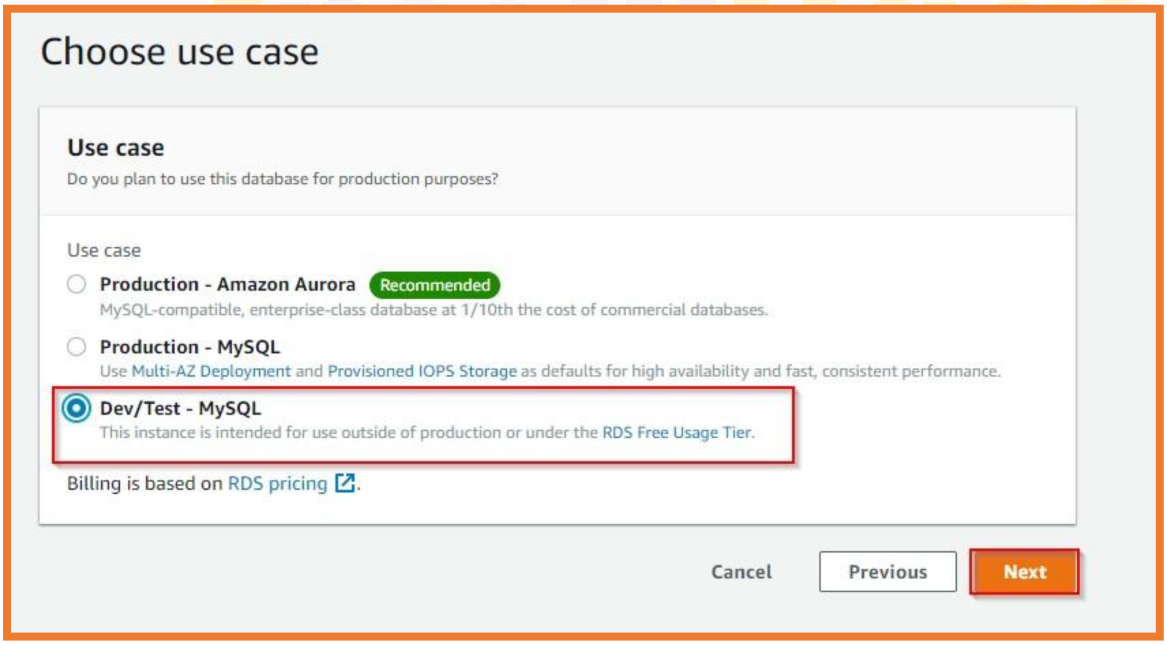
○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

MySQL

- Since we're using it for the demo purpose, so we'll choose the Dev/Test -MySQL option only and then click **Next**



Choose use case

Use case

Do you plan to use this database for production purposes?

Use case

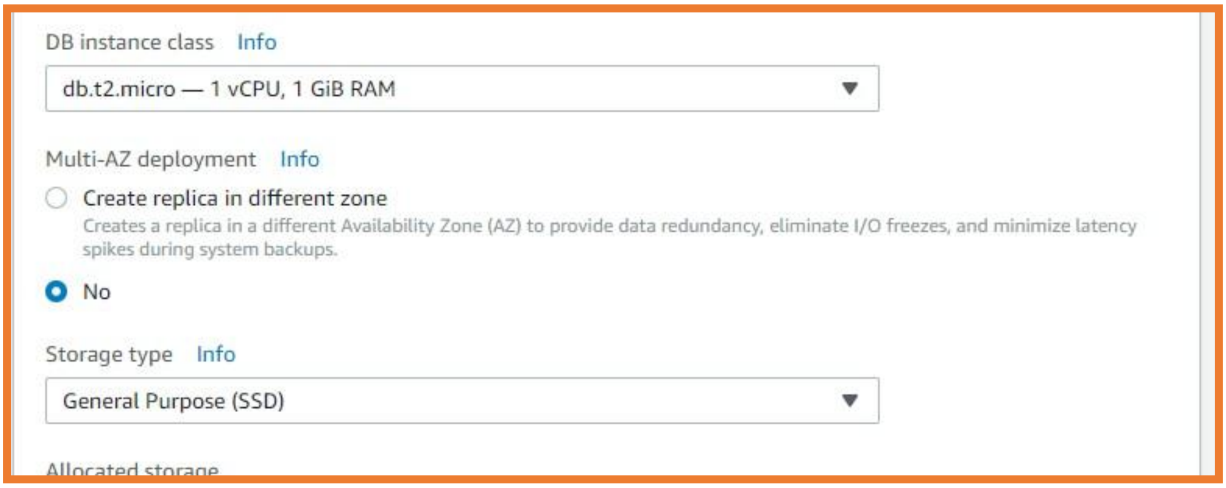
○ Production - Amazon Aurora Recommended
MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases.

○ Production - MySQL
Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance.

● Dev/Test - MySQL
This instance is intended for use outside of production or under the RDS Free Usage Tier.

Billing is based on RDS pricing.

- Specify DB Details, make sure to choose only **db.t2.micro** in DB Instance Class



DB instance class Info

db.t2.micro — 1 vCPU, 1 GiB RAM ▼

Multi-AZ deployment Info

○ Create replica in different zone

Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

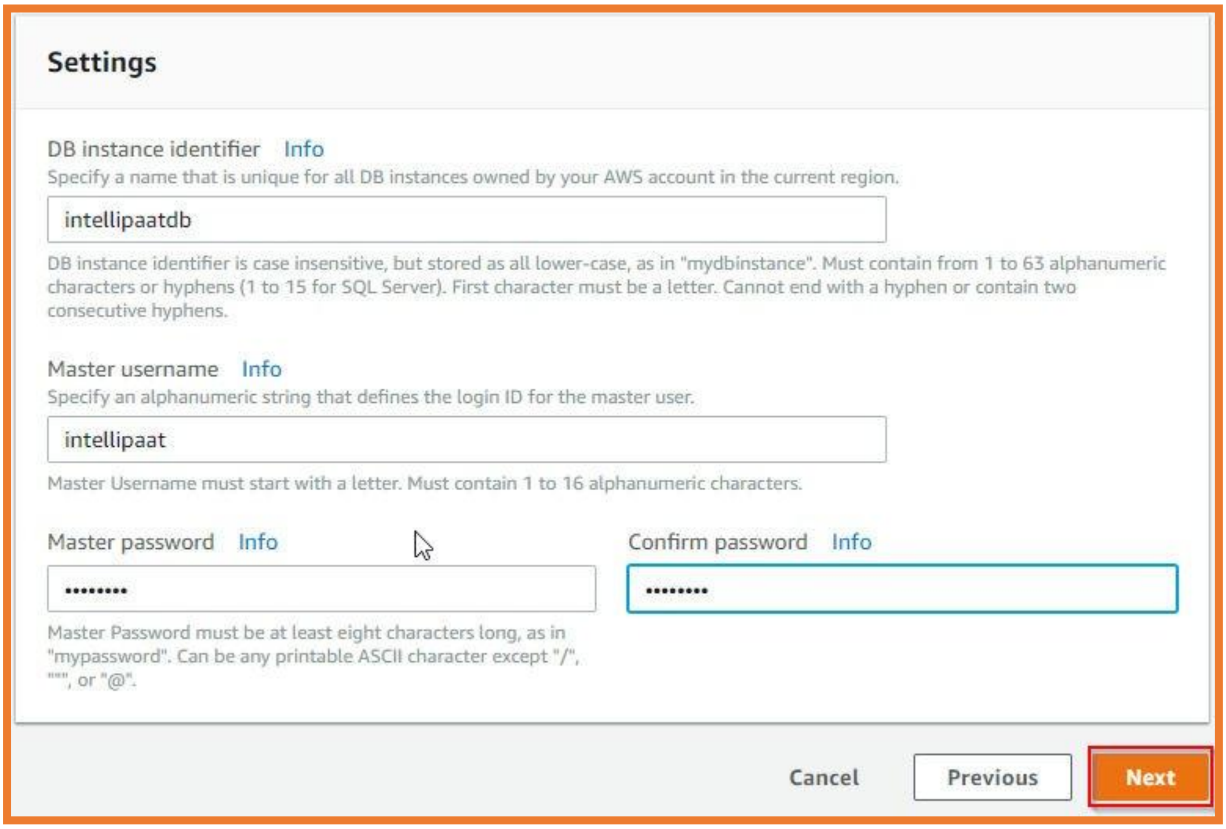
● No

Storage type Info

General Purpose (SSD) ▼

Allocated storage

- Enter these credentials (Note: Make sure you remember these credentials, as they will be required for connecting the RDS with your PuTTY



Settings

DB instance identifier Info

Specify a name that is unique for all DB instances owned by your AWS account in the current region.

intellipaatdb

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

Master username Info

Specify an alphanumeric string that defines the login ID for the master user.

intellipaat

Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

Master password Info

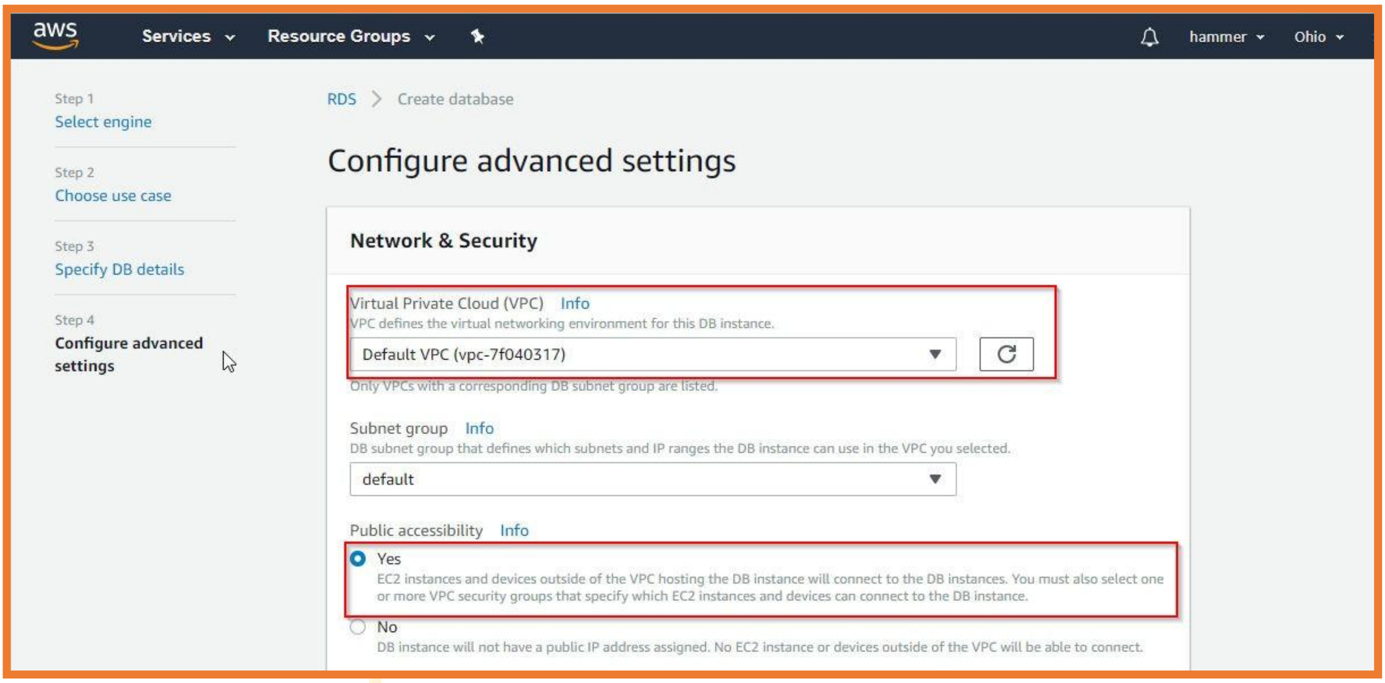
•••••••••

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", """, or "@".

Confirm password Info

•••••••••

- Then in the **Configure Advanced Option,** make sure to keep the VPC as default, along with the Public Accessibility as **Yes**



Step 1
Select engine

Step 2
Choose use case

Step 3
Specify DB details

Step 4
Configure advanced settings

RDS > Create database

Configure advanced settings

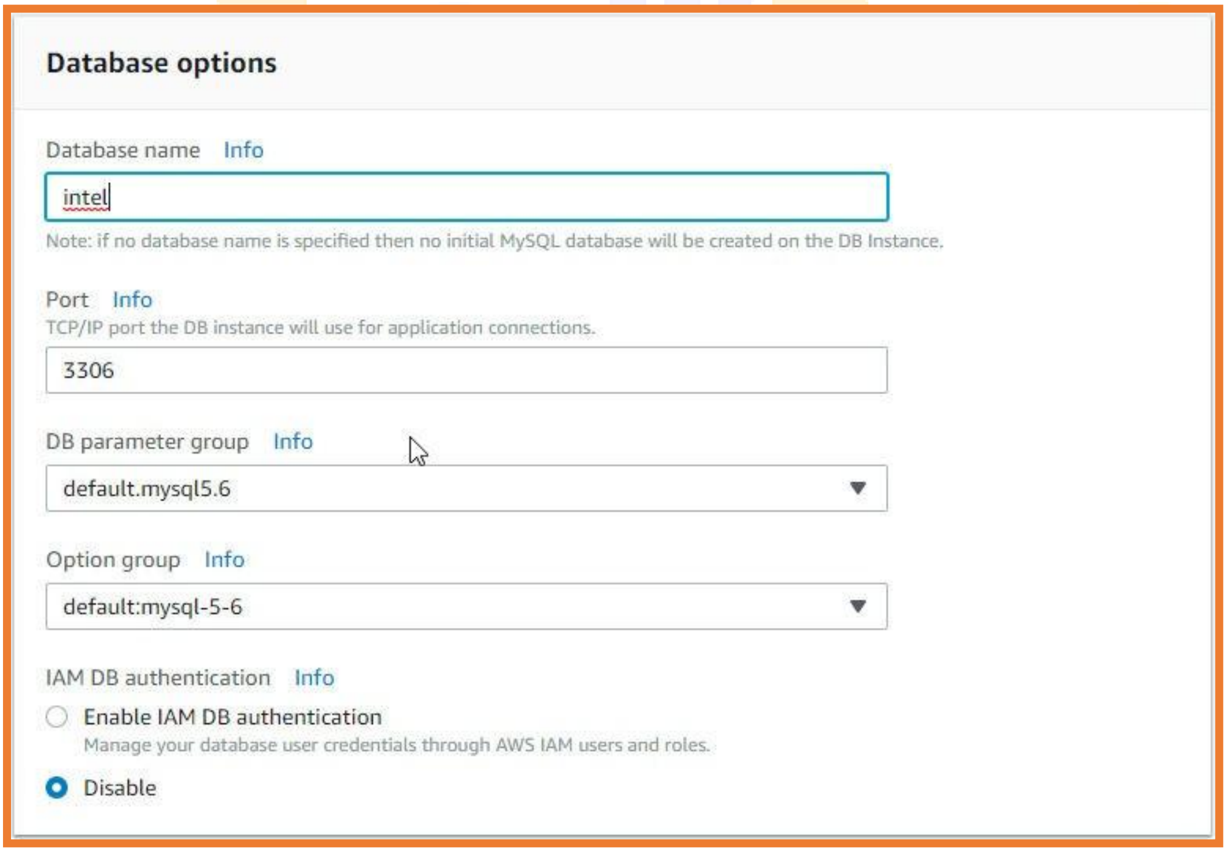
Network & Security

Virtual Private Cloud (VPC) Info
VPC defines the virtual networking environment for this DB instance.
Default VPC (vpc-7f040317)
Only VPCs with a corresponding DB subnet group are listed.

Subnet group Info
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.
default

Public accessibility Info
● Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.
○ No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

- In the Database Options, name the Database and keep the other artefacts as it is



Database options

Database name Info
intel
Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

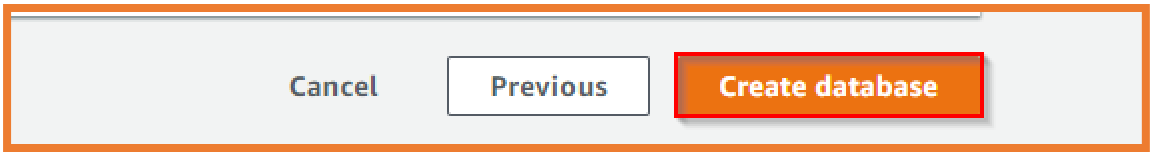
Port Info
TCP/IP port the DB instance will use for application connections.
3306

DB parameter group Info
default.mysql5.6

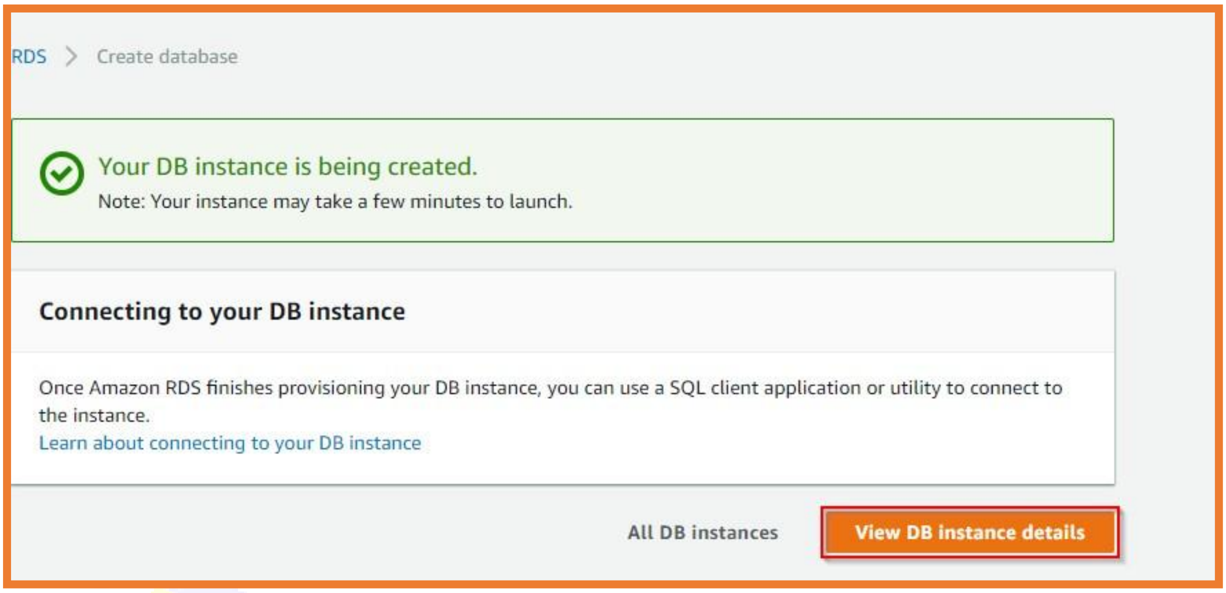
Option group Info
default:mysql-5-6

IAM DB authentication Info
○ Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.
● Disable

- Then click on Create database



- Then you can check your instance status



Your DB instance is being created.

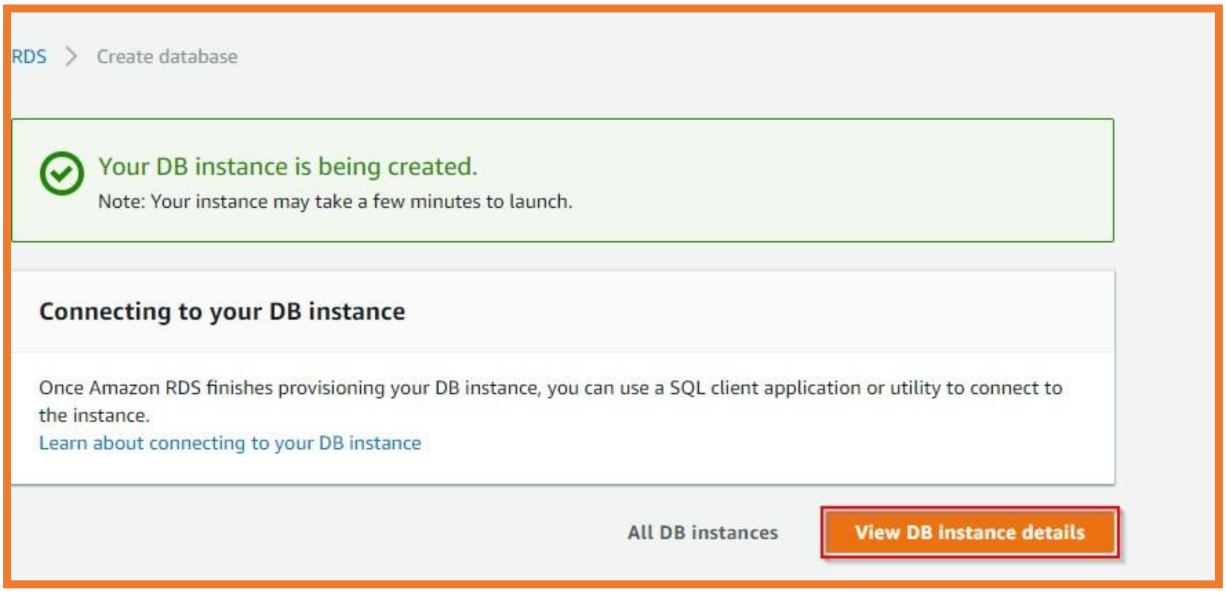
Note: Your instance may take a few minutes to launch.

Connecting to your DB instance

Once Amazon RDS finishes provisioning your DB instance, you can use a SQL client application or utility to connect to the instance.

Learn about connecting to your DB instance

- It may take few minutes for RDS to go from Initial to Running stage, you will observe that Endpoint and Port are not yet available (wait for few minutes)



Your DB instance is being created.

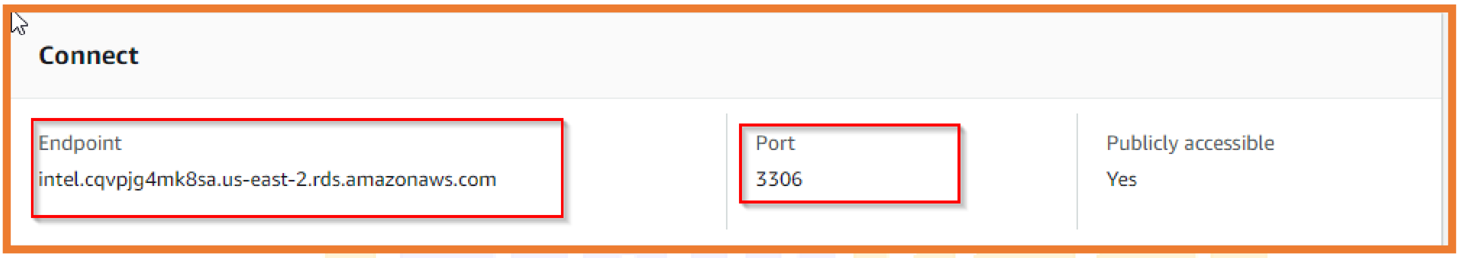
Note: Your instance may take a few minutes to launch.

Connecting to your DB instance

Once Amazon RDS finishes provisioning your DB instance, you can use a SQL client application or utility to connect to the instance.

Learn about connecting to your DB instance

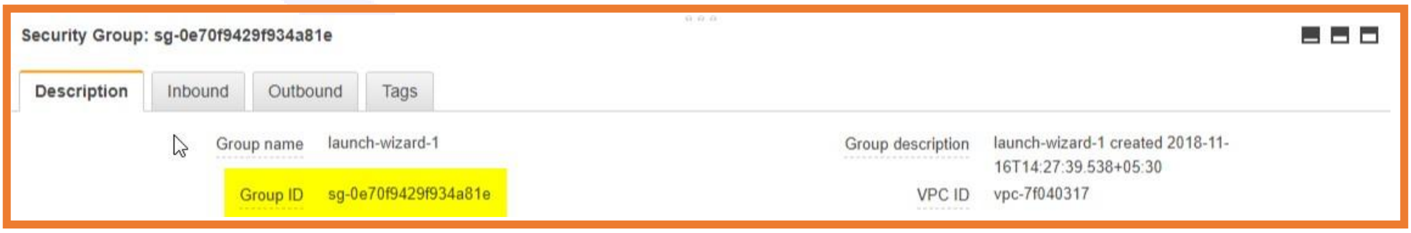
- In few minutes, you will be able to see the Endpoint and Port



Connect

Endpoint	Port	Publicly accessible
intel.cqvpjg4mk8sa.us-east-2.rds.amazonaws.com	3306	Yes

- Also, make sure to change some security configuration in the RDS
- Go to your EC2 Instance Security Groups and select your group ID



Security Group: sg-0e70f9429f934a81e

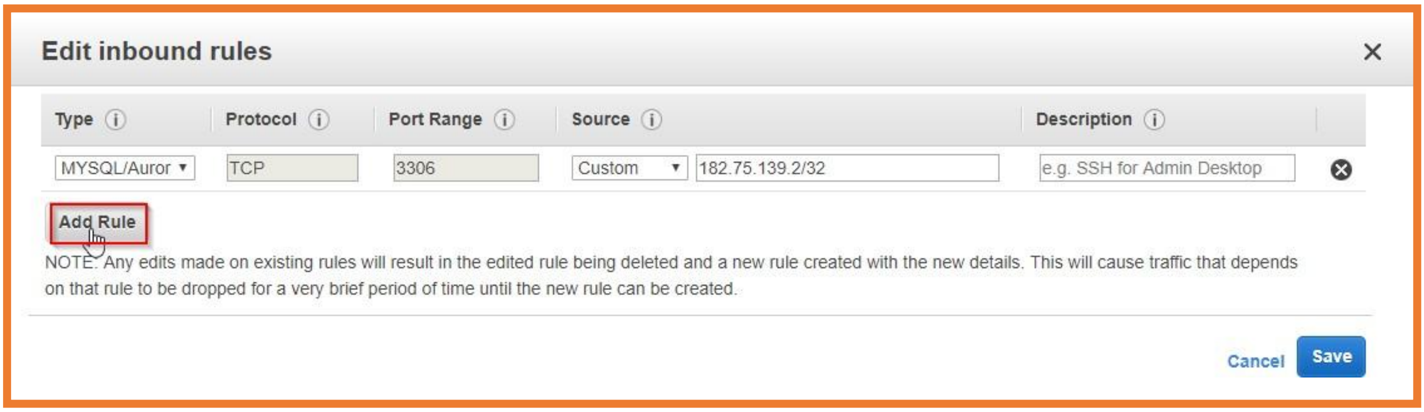
Group name: launch-wizard-1

Group ID: sg-0e70f9429f934a81e

Group description: launch-wizard-1 created 2018-11-16T14:27:39.538+05:30

VPC ID: vpc-7f040317

- Then go to RDS Security groups and select the Inbound rules panel there and click on Add Rule

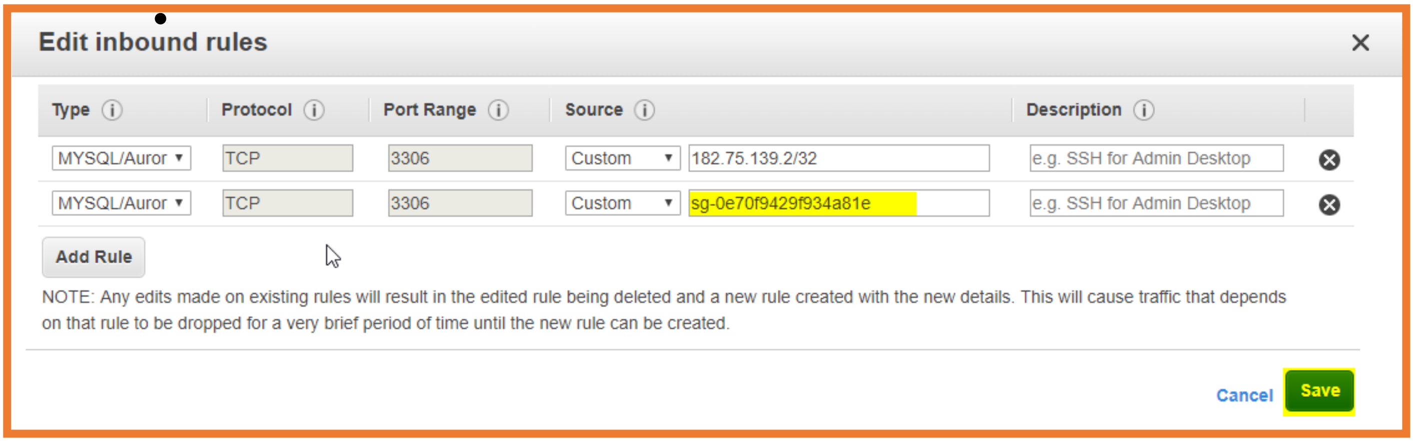


Edit inbound rules

Type	Protocol	Port Range	Source		Description
MYSQL/Auror	TCP	3306	Custom	182.75.139.2/32	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Then paste the EC2 Security ID in Source> Custom > **Security Group** by keeping the Type as MYSQL/Aurora



Edit inbound rules ✕

Type	Protocol	Port Range	Source		Description	
MYSQL/Auror ▾	TCP	3306	Custom ▾	182.75.139.2/32	e.g. SSH for Admin Desktop	⊗
MYSQL/Auror ▾	TCP	3306	Custom ▾	sg-0e70f9429f934a81e	e.g. SSH for Admin Desktop	⊗

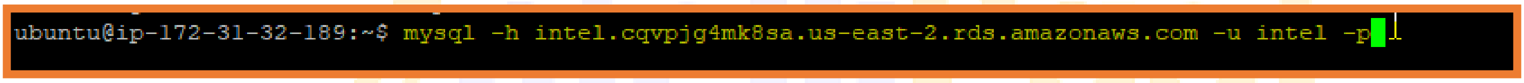
Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

- Now go back to your PuTTY and use this command as shown below
  mysql -h hostname -u username -p
  NOTE:
  - In place of hostname, make sure to use your Endpoint from RDS
  - Username which you created

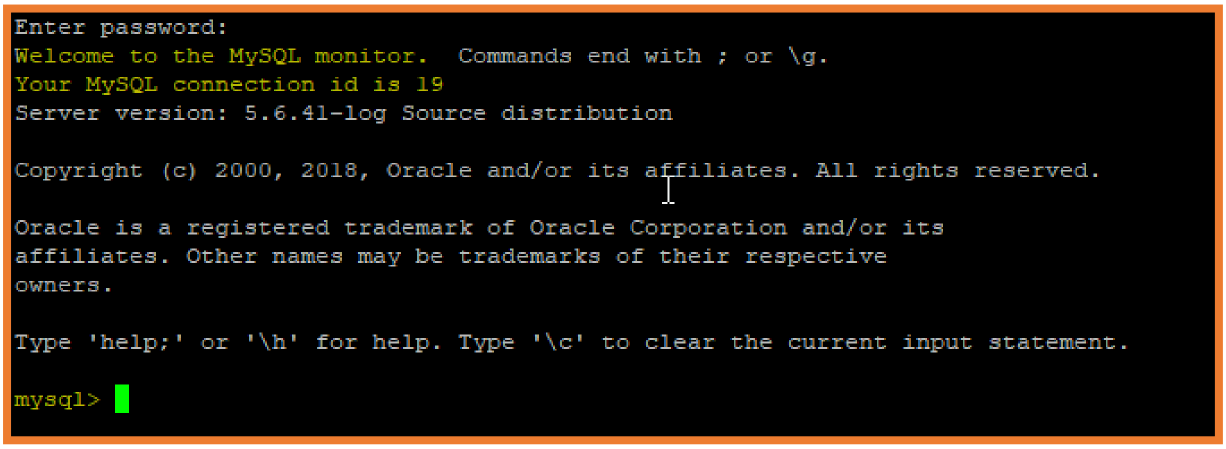
Here, we're using our own Endpoint and username and password used



```
ubuntu@ip-172-31-32-189:~$ mysql -h intel.cqvpjg4mk8sa.us-east-2.rds.amazonaws.com -u intel -p
```

Use the command as shown below

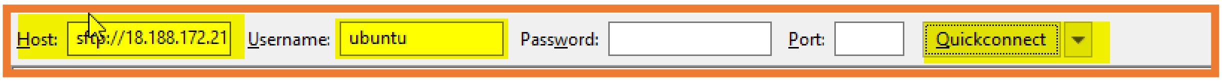
- After this, it will ask for your password, in our case, password is: intel123
- Then it will show that you're connected to the mysql



```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.6.41-log Source distribution

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

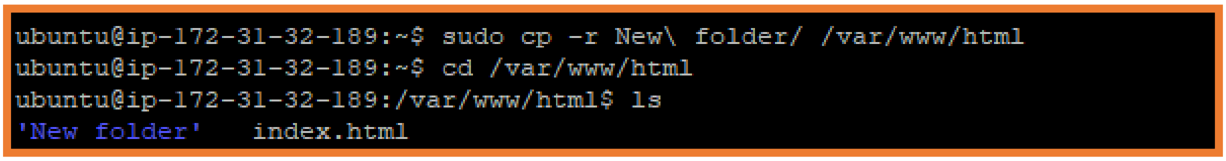
Filezilla

- Now install Filezilla
- In order to connect it, enter hostname as the Endpoint of EC2 and Username as Ubuntu and no need to keep the password, then quickconnect.



Host: sftp://18.188.172.21 Username: ubuntu Password: Port:

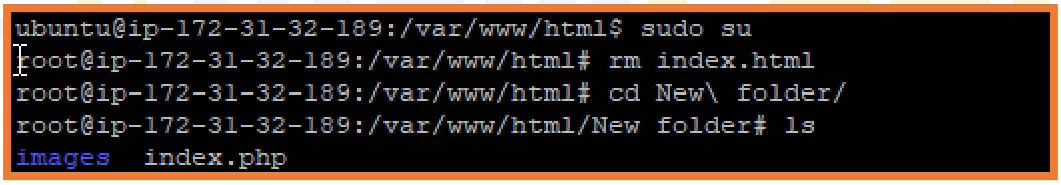
- Now your Filezilla is connected with your EC2 instance
- Create a **'New Folder'** of your website in your Desktop
- And copy paste it in your Filezilla Remote Site path: /home/ubuntu



```
ubuntu@ip-172-31-32-189:~$ sudo cp -r New\ folder/ /var/www/html
ubuntu@ip-172-31-32-189:~$ cd /var/www/html
ubuntu@ip-172-31-32-189:/var/www/html$ ls
'New folder'   index.html
```

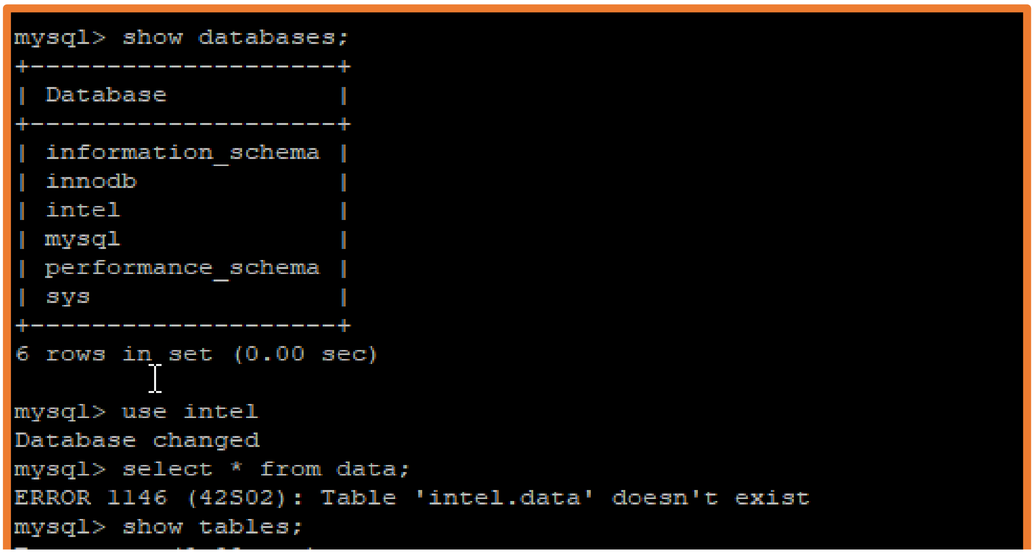
- Now go back to your PuTTY, where you will see that it contains the index.html file
- Now you need to remove this 'index.html' file and add 'index.php' in its place

  For that you need to use "**sudo su**" and remove this file using remove command



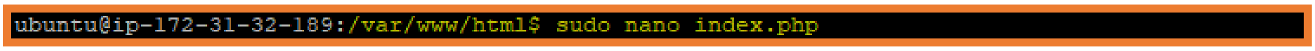
```
ubuntu@ip-172-31-32-189:/var/www/html$ sudo su
root@ip-172-31-32-189:/var/www/html# rm index.html
root@ip-172-31-32-189:/var/www/html# cd New\ folder/
root@ip-172-31-32-189:/var/www/html/New folder# ls
images  index.php
```

- Also, before running this website, you need to create a table in it (its database)



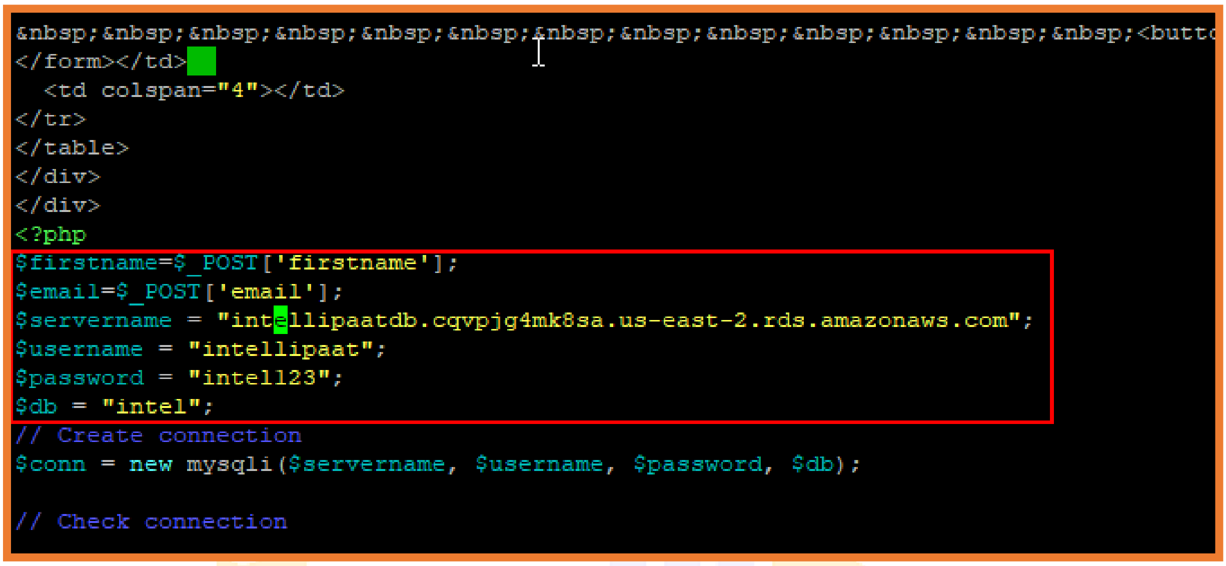
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| intel              |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> use intel
Database changed
mysql> select * from data;
ERROR 1146 (42S02): Table 'intel.data' doesn't exist
mysql> show tables;
```

- Now go to the path where website files are kept and run the **index.php** file by using **sudo nano index.php**



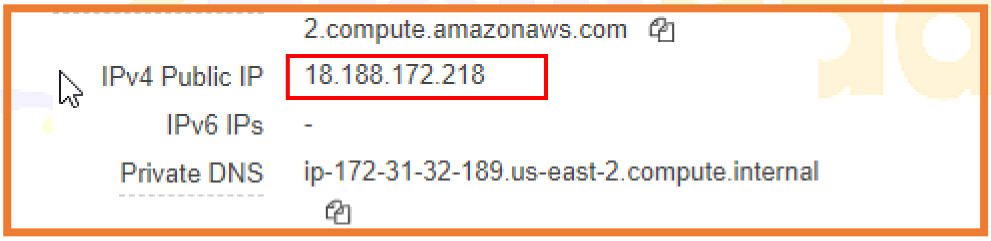
```
ubuntu@ip-172-31-32-189:/var/www/html$ sudo nano index.php
```

- Now after this, GNU nano will pop up where you have to make changes in your code, you have to check if in your server name, the endpoint of your RDS is there along with username, password and db name



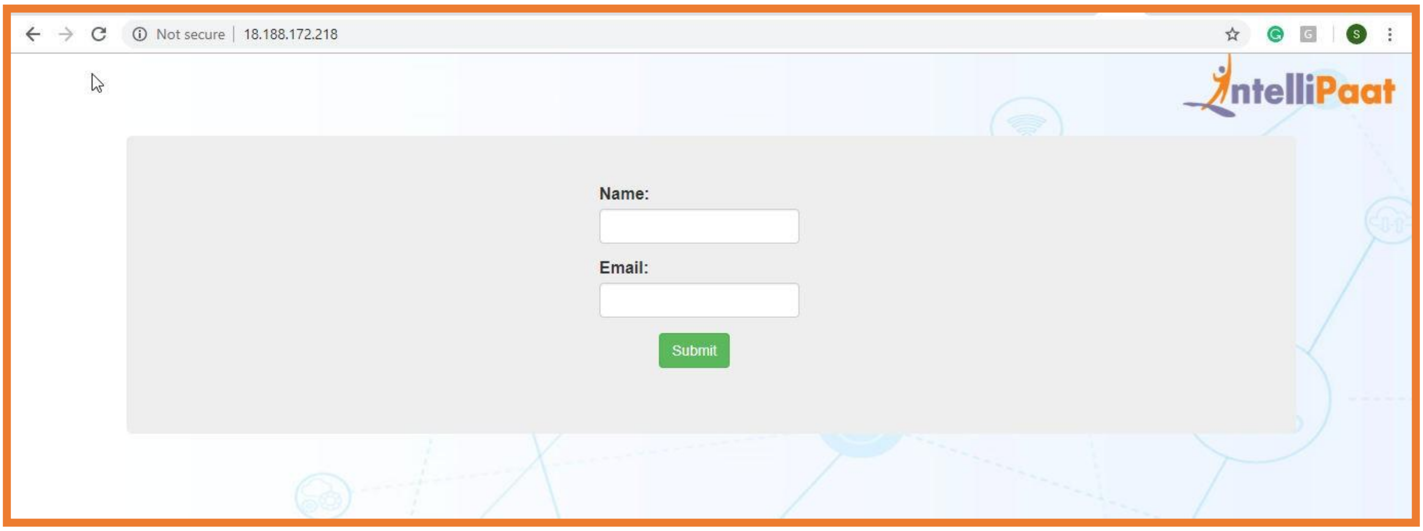
```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<butt
</form></td>
  <td colspan="4"></td>
</tr>
</table>
</div>
</div>
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "intellipaatdb.cqvpjg4mk8sa.us-east-2.rds.amazonaws.com";
$username = "intellipaat";
$password = "intel123";
$db = "intel";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

// Check connection
```

- Now when you will try, and copy paste the Public IP of your EC2 Instance



	2.compute.amazonaws.com
IPv4 Public IP	18.188.172.218
IPv6 IPs	-
Private DNS	ip-172-31-32-189.us-east-2.compute.internal

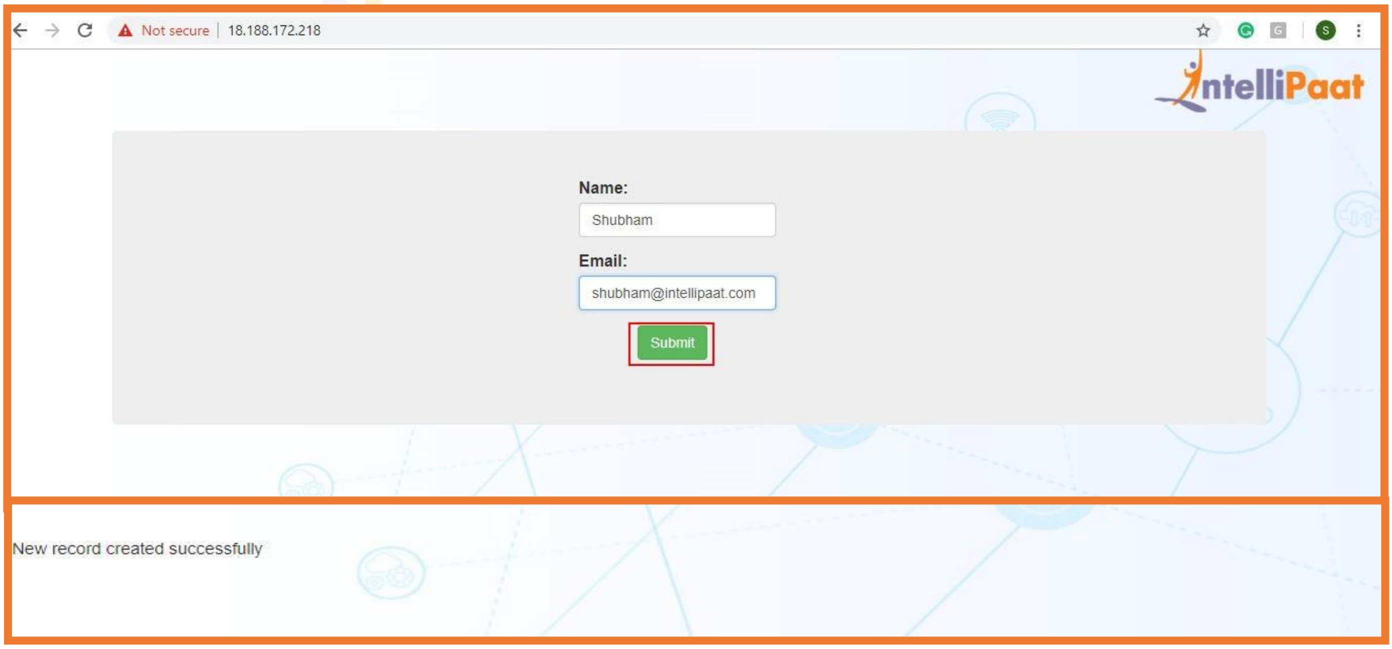
- After copying this IP to your browser, you will observe that your website is working on it



Name:

Email:

- Now when you enter these details in this website, you will see the following result



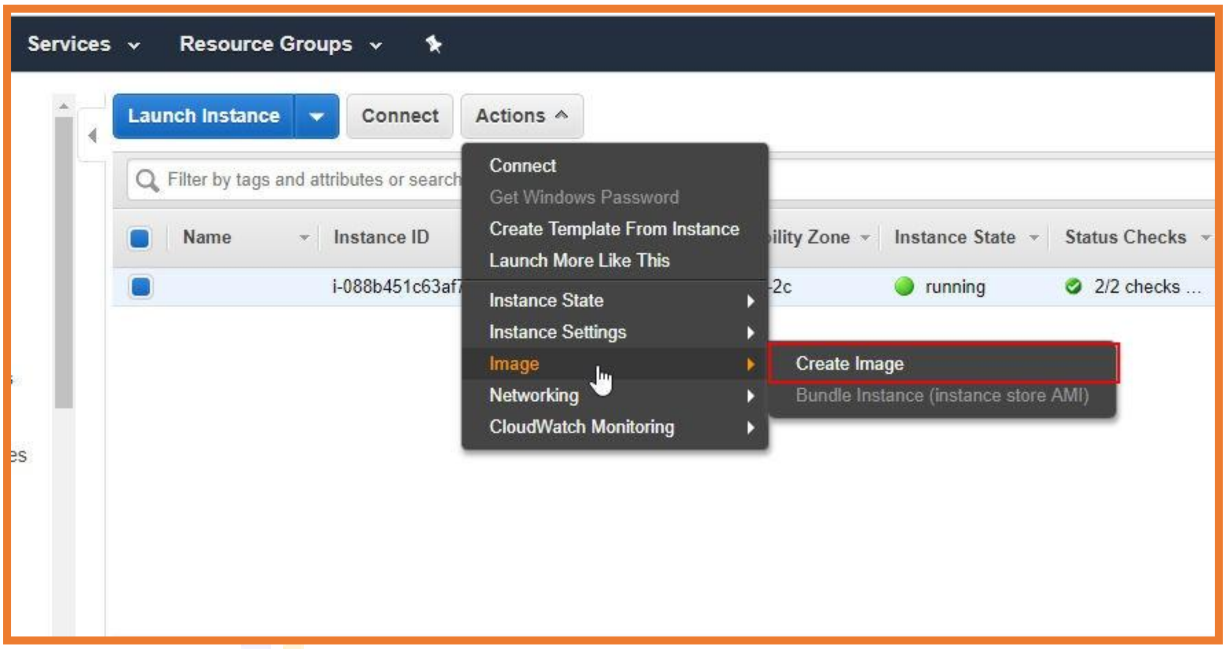
Name: Shubham

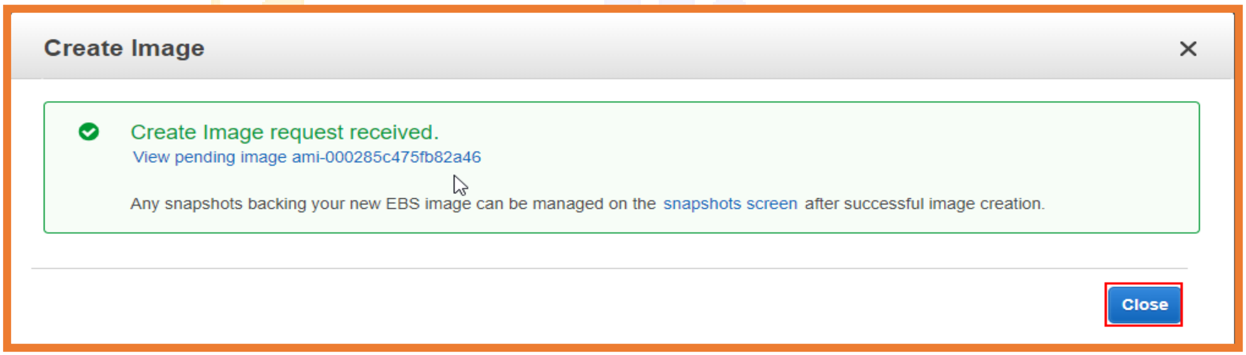
Email: shubham@intellipaat.com

New record created successfully

AUTO SCALING

Now, we'll do the autoscaling of our website by going to our EC2 Instance and then click on **Actions** and **Create Image**



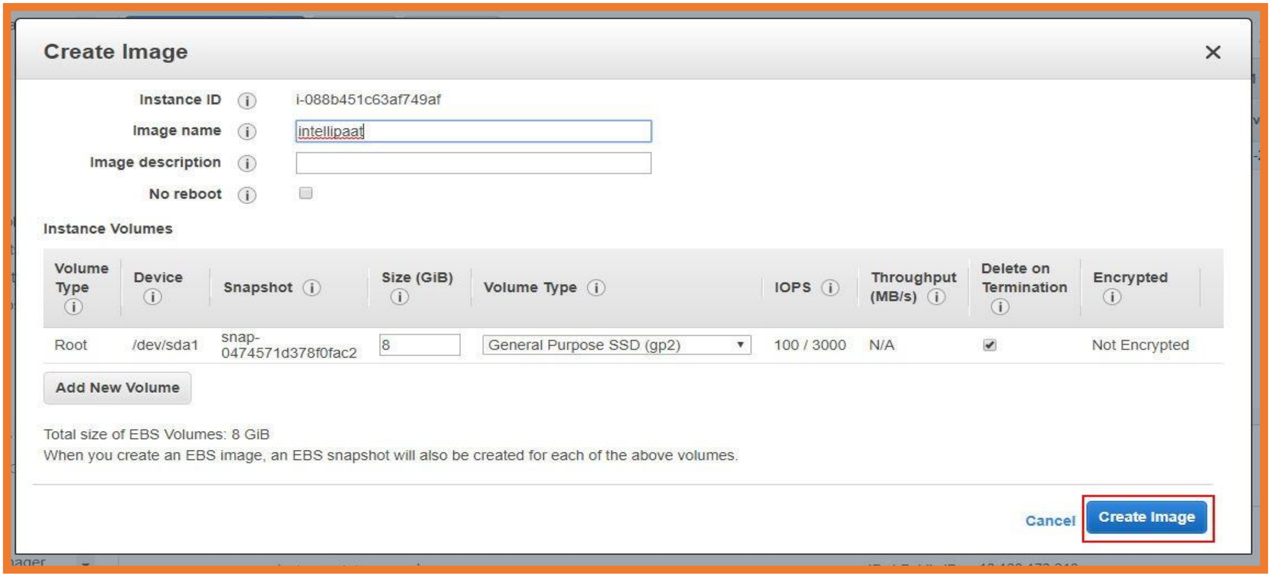


Create Image

Create Image request received.
View pending image ami-000285c475fb82a46

Any snapshots backing your new EBS image can be managed on the snapshots screen after successful image creation.

Close



Create Image

Instance ID: i-088b451c63af749af

Image name: intellipaat

Image description:

No reboot: □

Instance Volumes

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0474571d378f0fac2	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Add New Volume

Total size of EBS Volumes: 8 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Cancel Create Image

- Then further, activate its autoscaling and then its classic load balancer which directs the traffic to your website directly