

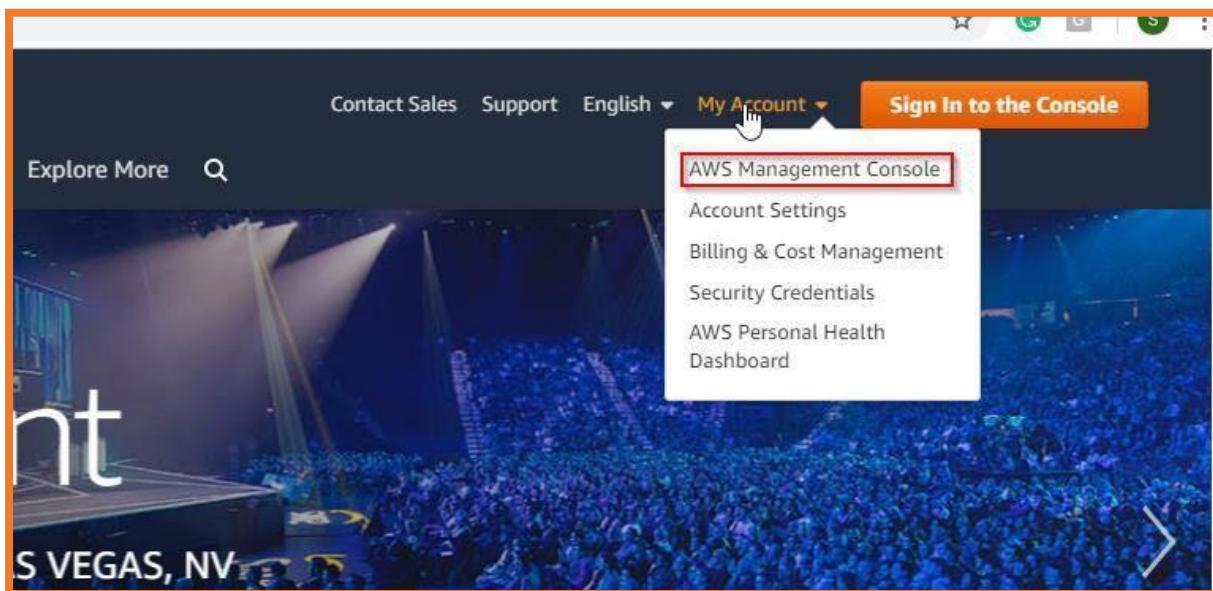
SOLUTION

Connect your system with your EC2 Instance

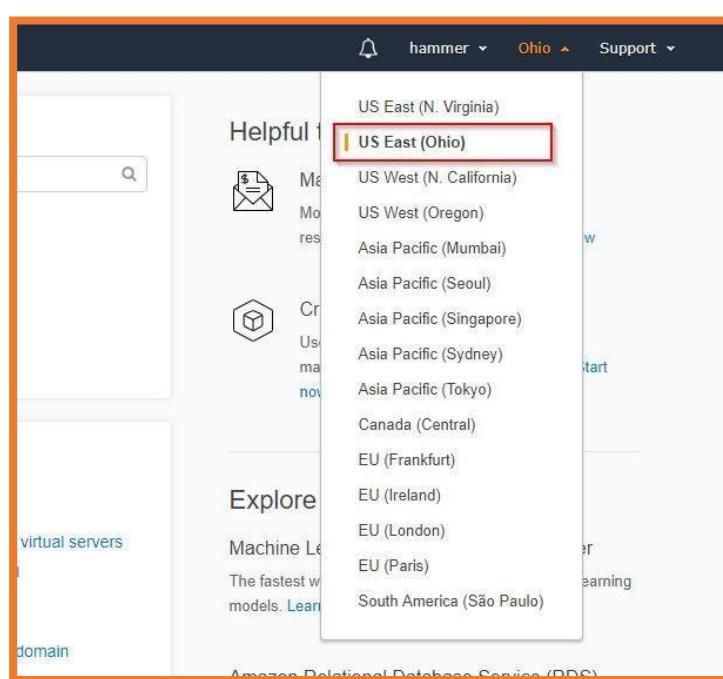
First you need to install **Putty** on your system and then connect it with your EC2 instance.

Below are the steps for it:

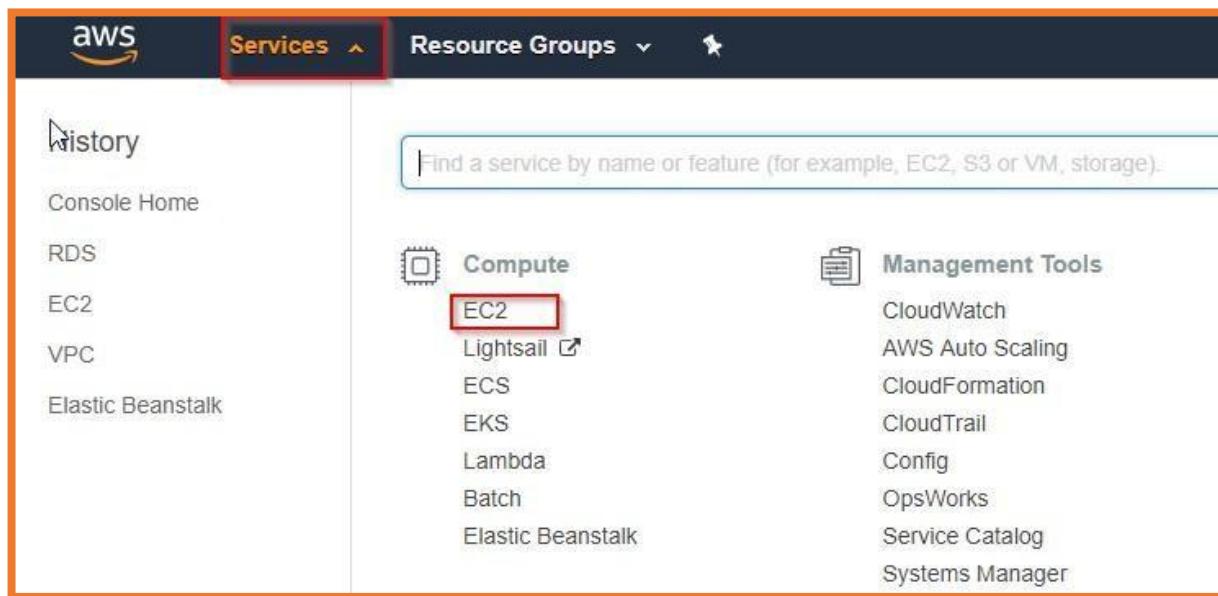
- First sign into the AWS Management Console.



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

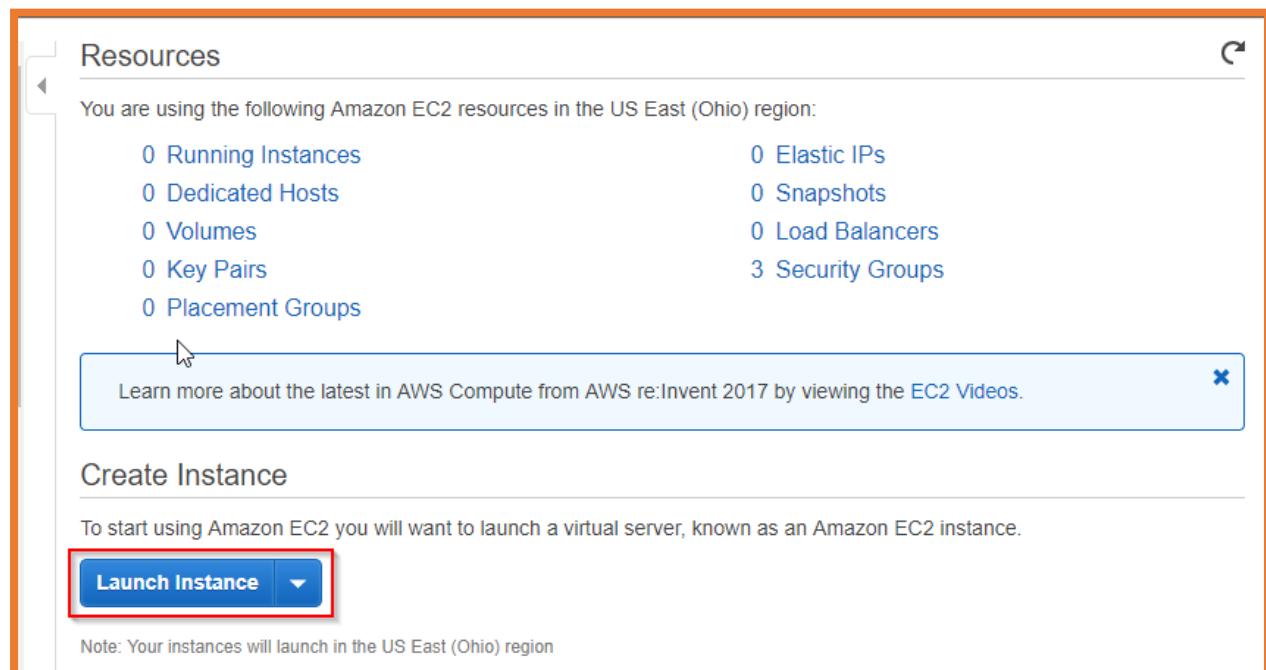


- In the **Services** section, you must see **Compute** where you need to choose **EC2**



The screenshot shows the AWS Services menu. The 'Compute' section is expanded, and 'EC2' is selected. Other options in the Compute section include Lightsail, ECS, EKS, Lambda, Batch, and Elastic Beanstalk. The 'Management Tools' section includes CloudWatch, AWS Auto Scaling, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, and Systems Manager.

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



The screenshot shows the AWS EC2 Resources page. It displays resource counts: 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 0 Key Pairs, 0 Placement Groups, 0 Elastic IPs, 0 Snapshots, 0 Load Balancers, and 3 Security Groups. A callout box provides information about AWS Compute from AWS re:Invent 2017. Below this, the 'Create Instance' section is shown, featuring a prominent 'Launch Instance' button, which is highlighted with a red box. A note at the bottom states: '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-0f65671a86f061fcf

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

Select
64-bit (x86)

- 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
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	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	4	8	EBS only	-	Low to Moderate	Yes

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

Cancel Previous Review and Launch Next: Configure Instance Details

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.

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

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 Zone) 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

Cancel Previous Review and Launch Next: Add Storage

- Then click on Add Tags

This screenshot shows the AWS EBS volume configuration interface. A specific step, '5. Add Tags', is highlighted with a red border. The page displays a table with one row for a 'Root' volume. The 'Volume Type' is 'General Purpose SSD (gp2)'. The 'Delete on Termination' checkbox is checked. A note at the bottom states: '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.' Navigation buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (which is blue), and 'Next: Add Tags' (which is also blue and has a red border).

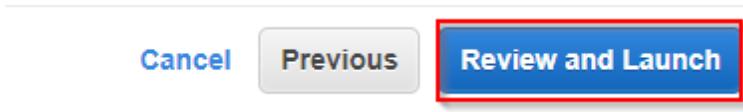
- Add tags then name the key and a value, click Configure Security Group

This screenshot shows the AWS EC2 instance creation wizard, specifically Step 5: Add Tags. The step is highlighted with a red border. The wizard has seven steps in total: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (highlighted), 6. Configure Security Group, and 7. Review. The 'Add Tags' section allows users to add key-value pairs. One tag is shown: 'Key' is 'intellipaat' and 'Value' is 'ec2-demo'. There is a link to 'Configure Security Group' at the bottom right.

- Keep the configuration of security group as it

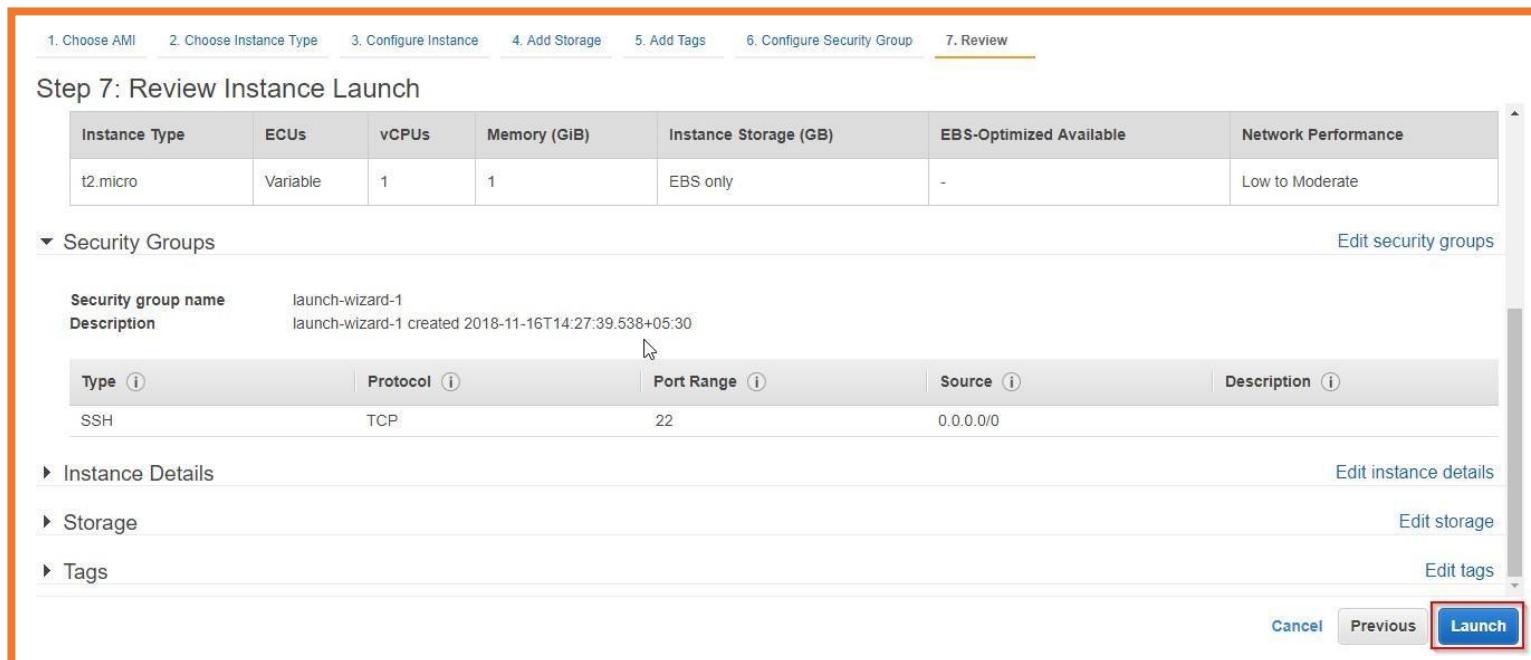
This screenshot shows the AWS Security Groups configuration page. A specific step, '6. Configure Security Group', is highlighted with a red border. The page includes instructions: 'Assign a security group: Create a new security group or Select an existing security group'. A new security group named 'launch-wizard-1' is being created. The 'Description' field shows 'launch-wizard-1 created 2018-11-16T14:20:59.237+05:30'. Below this, a table lists a single rule: 'Type: SSH, Protocol: TCP, Port Range: 22, Source: Custom 0.0.0.0/0, Description: e.g. SSH for Admin Desktop'. There is a 'Add Rule' button at the bottom left.

- Then click **Review & Launch**



Cancel Previous **Review and Launch**

- Then directly **Launch it**



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

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

Type <i>i</i>	Protocol <i>i</i>	Port Range <i>i</i>	Source <i>i</i>	Description <i>i</i>
SSH	TCP	22	0.0.0.0/0	launch-wizard-1 created 2018-11-16T14:27:39.538+05:30

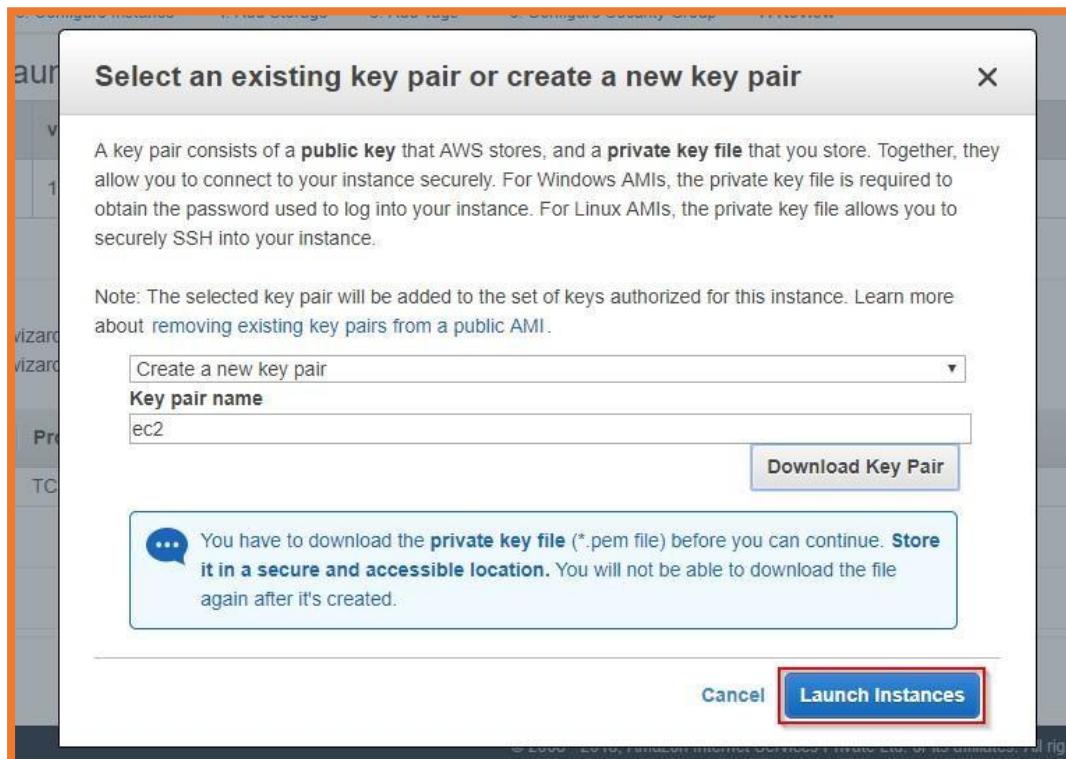
► Instance Details Edit instance details

► Storage Edit storage

► Tags Edit tags

Cancel Previous **Launch**

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



- Status

Launch Status

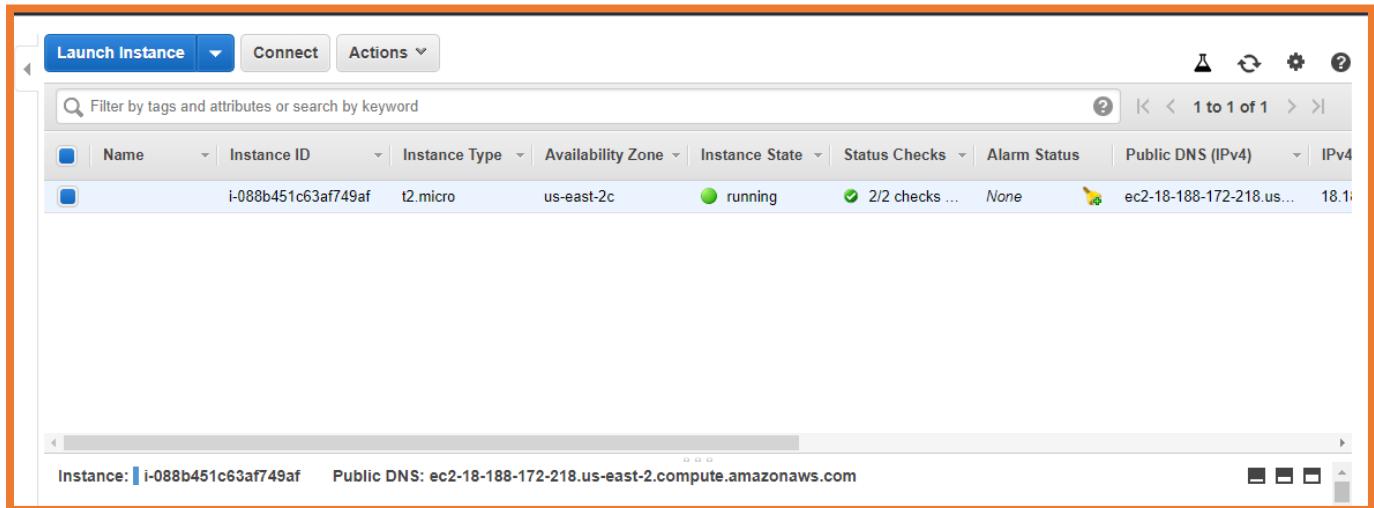
Your instances are now launching
The following instance launches have been initiated: i-088b451c63af749af [View launch log](#)

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).

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

Launch Instance	Connect	Actions	Actions	Actions	Actions	Actions	Actions	Actions	Actions	Actions	
<input type="button" value="Launch Instance"/>	<input type="button" value="Connect"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	<input type="button" value="Actions"/>	
<input type="text"/> Filter by tags and attributes or search by keyword											
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			

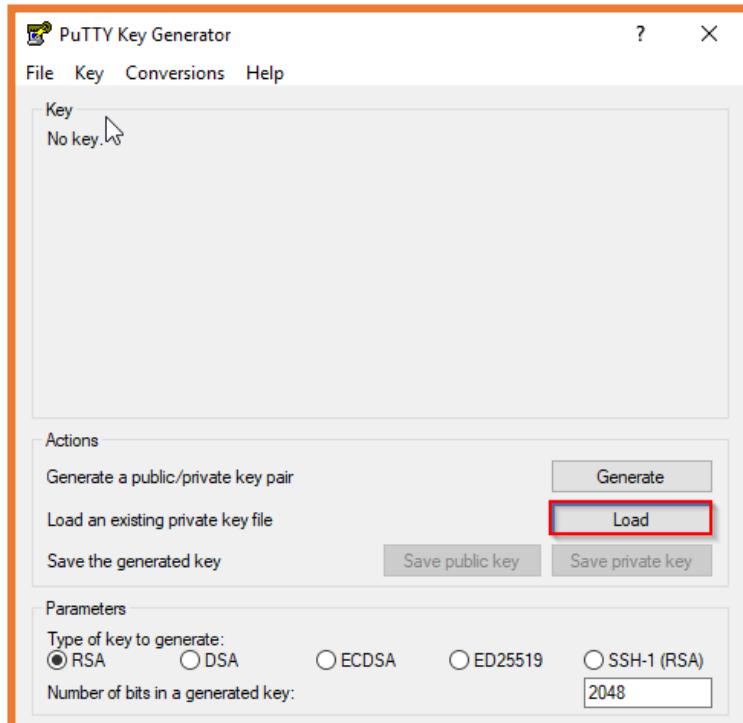
- Then after few minutes, you will see that now your instance is in running stage



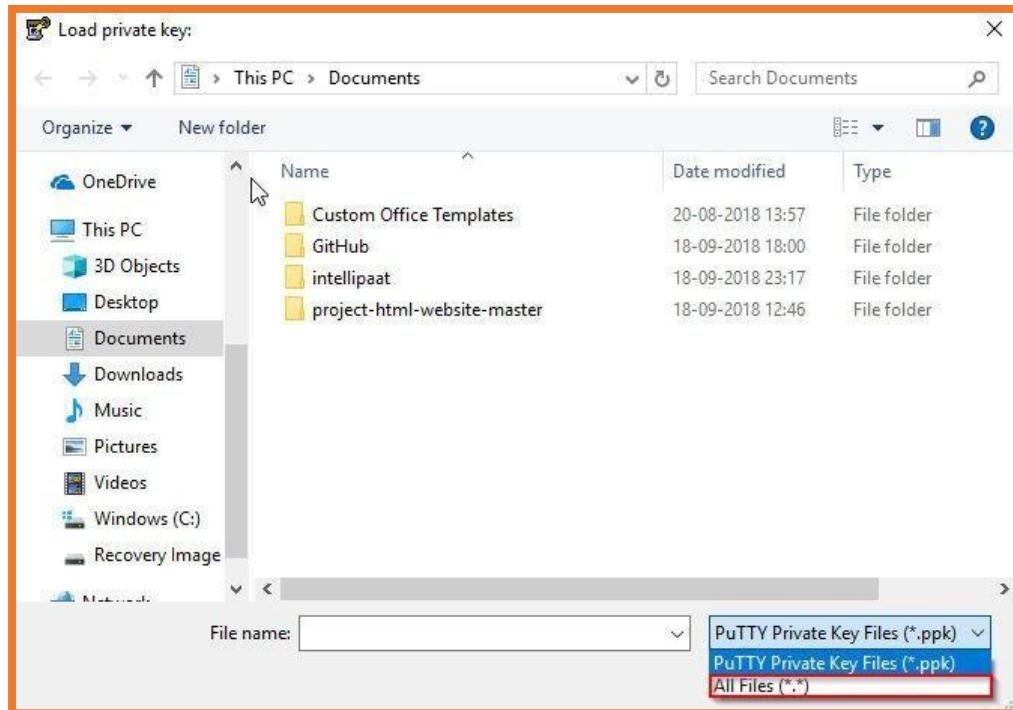
- Now it's time to convert your private key using PuTTYgen

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

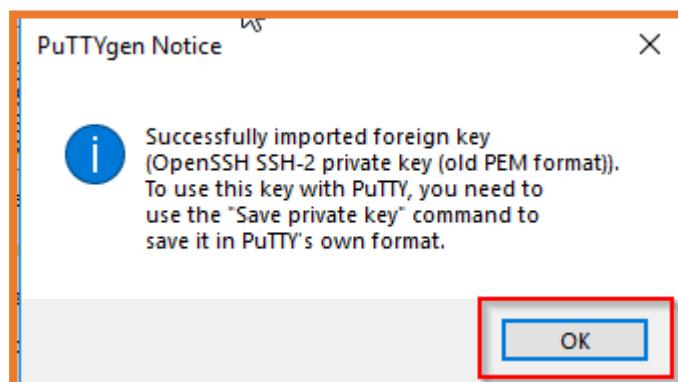
- Click Load in your PuTTY gen



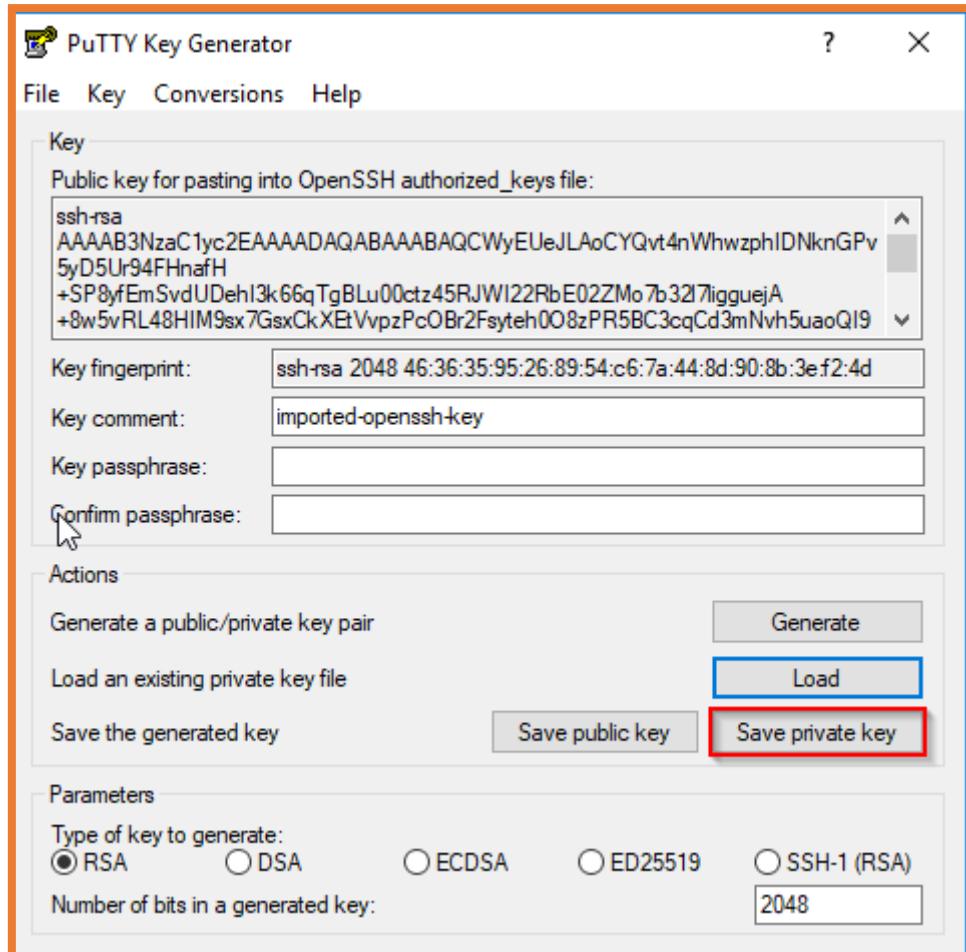
- PuTTY key gen always shows the .ppk format file, so go to the right bottom bar and select the All files option as shown below



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



- 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



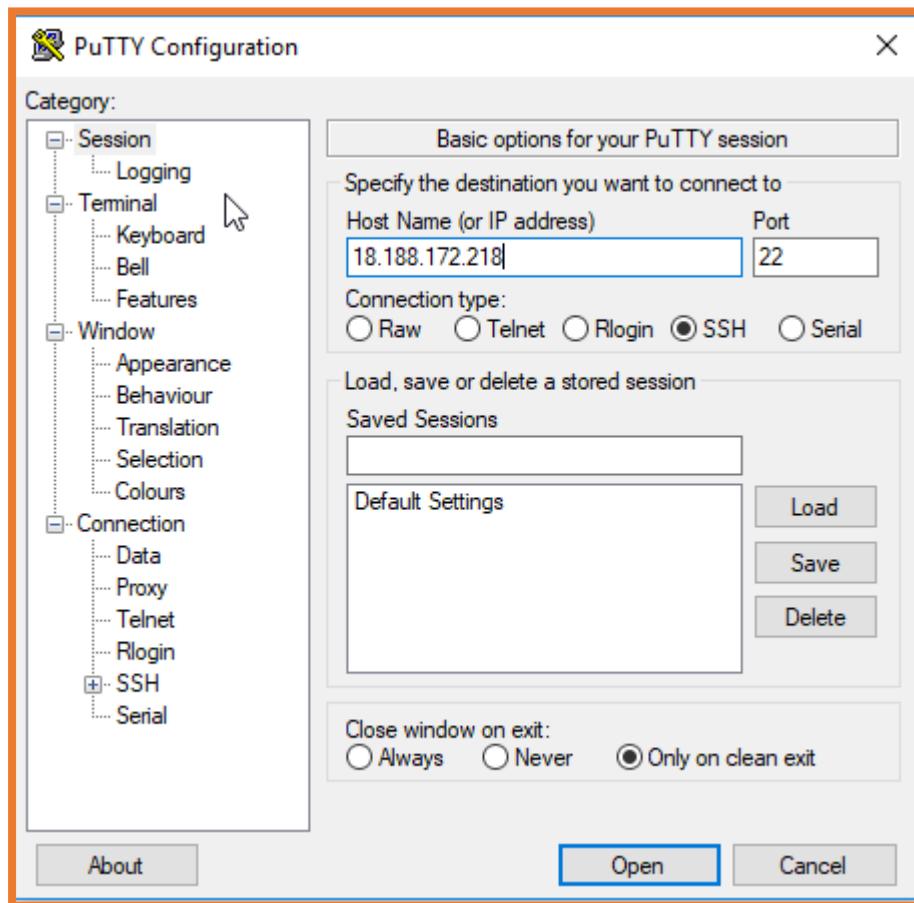
- Now you will see that in your folder, the .ppk file is already added with that name you had given (in our case, it's ec2)

Connecting to your EC2 Instance using SSH & PuTTY

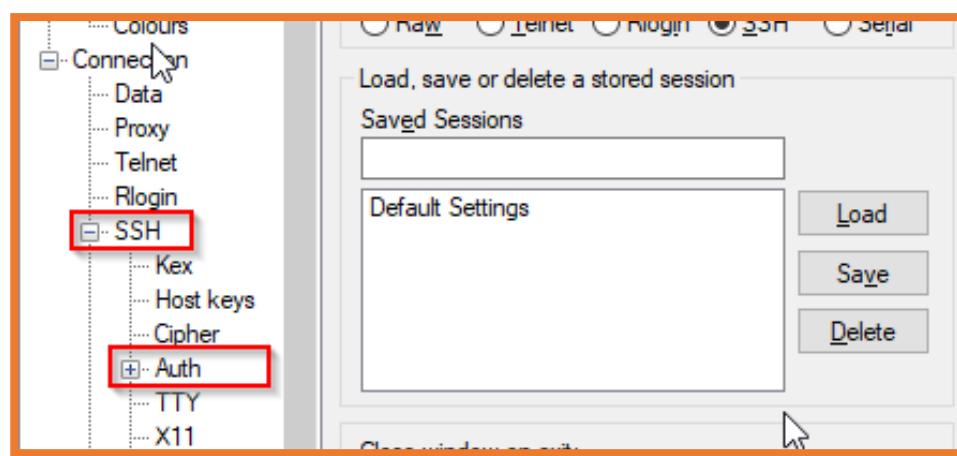
- First open PuTTY.exe then in the Host Name box, add the Public IP of your Instance



- Copy paste this Public IP in your PuTTY Hostname

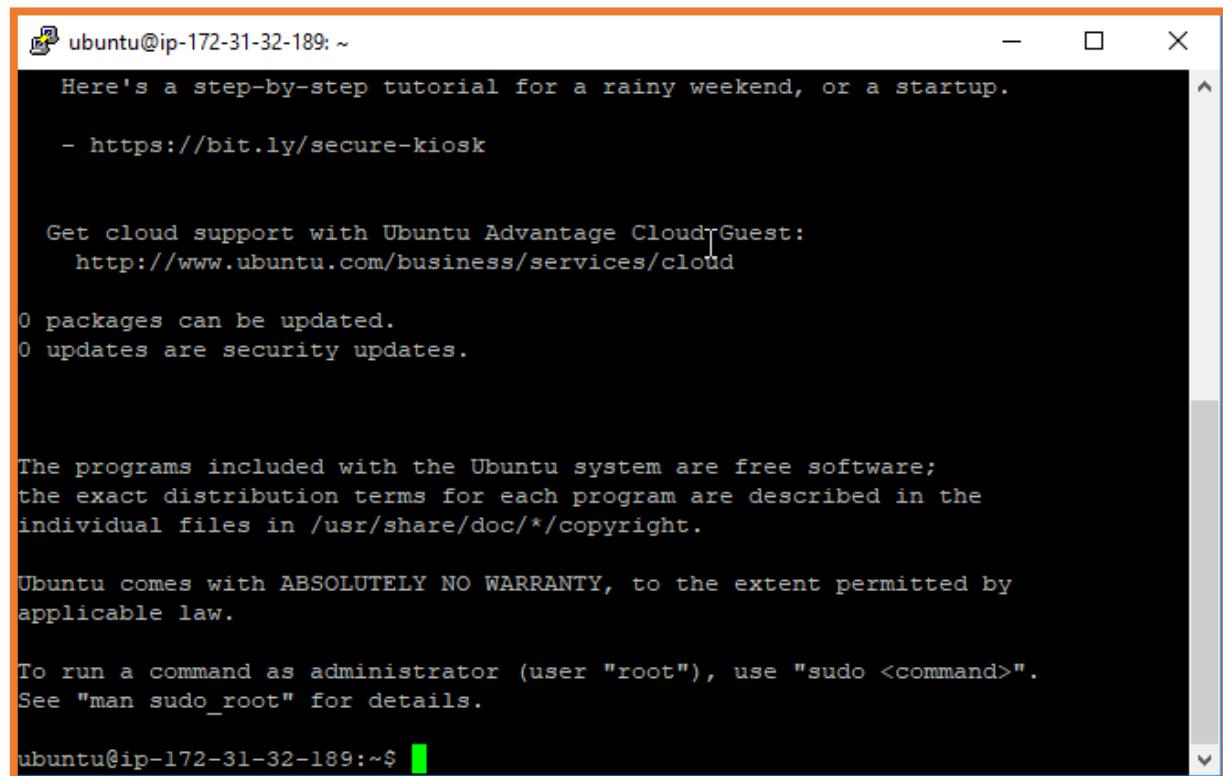


- Then in the category list, expand the SSH and Click on AUTH (but don't expand it)



- Then Click Open

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



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

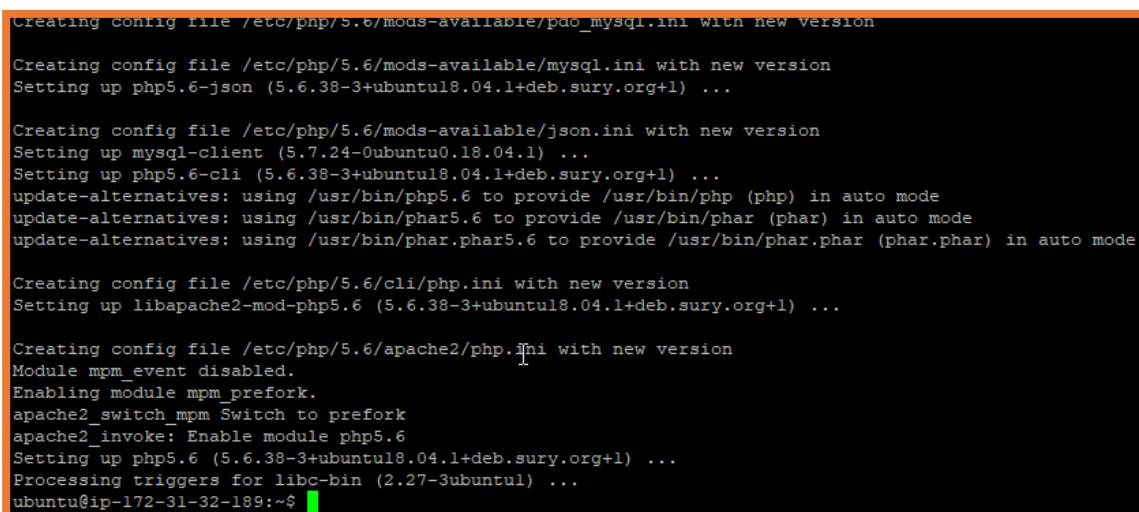
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

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/pnp/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+ubuntul8.04.1+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+ubuntul8.04.1+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+ubuntul8.04.1+deb.sury.org+1) ...

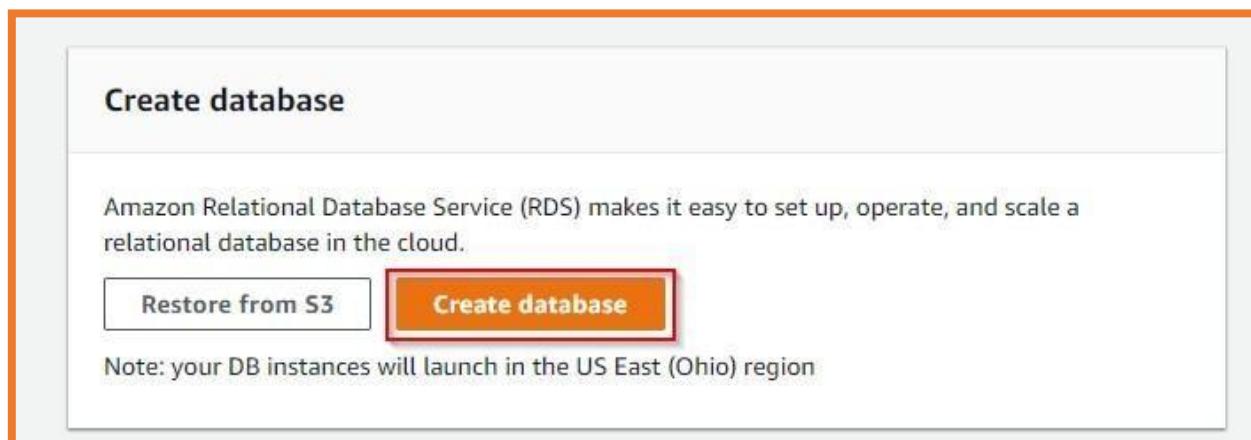
Creating config file /etc/php/5.6/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php5.6
Setting up php5.6 (5.6.38-3+ubuntul8.04.1+deb.sury.org+1) ...
Processing triggers for libc-bin (2.27-3ubuntul) ...
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**



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

Engine options

Amazon Aurora

Amazon Aurora

MySQL

MySQL

MariaDB



PostgreSQL



Oracle

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.

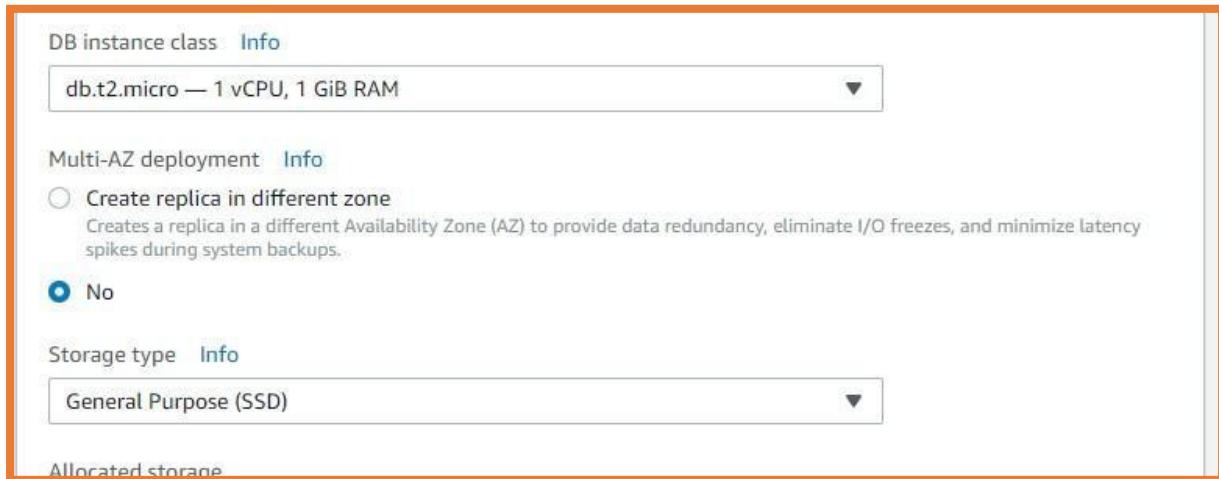
Production - MySQL
Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance.

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](#).

Cancel Previous **Next**

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



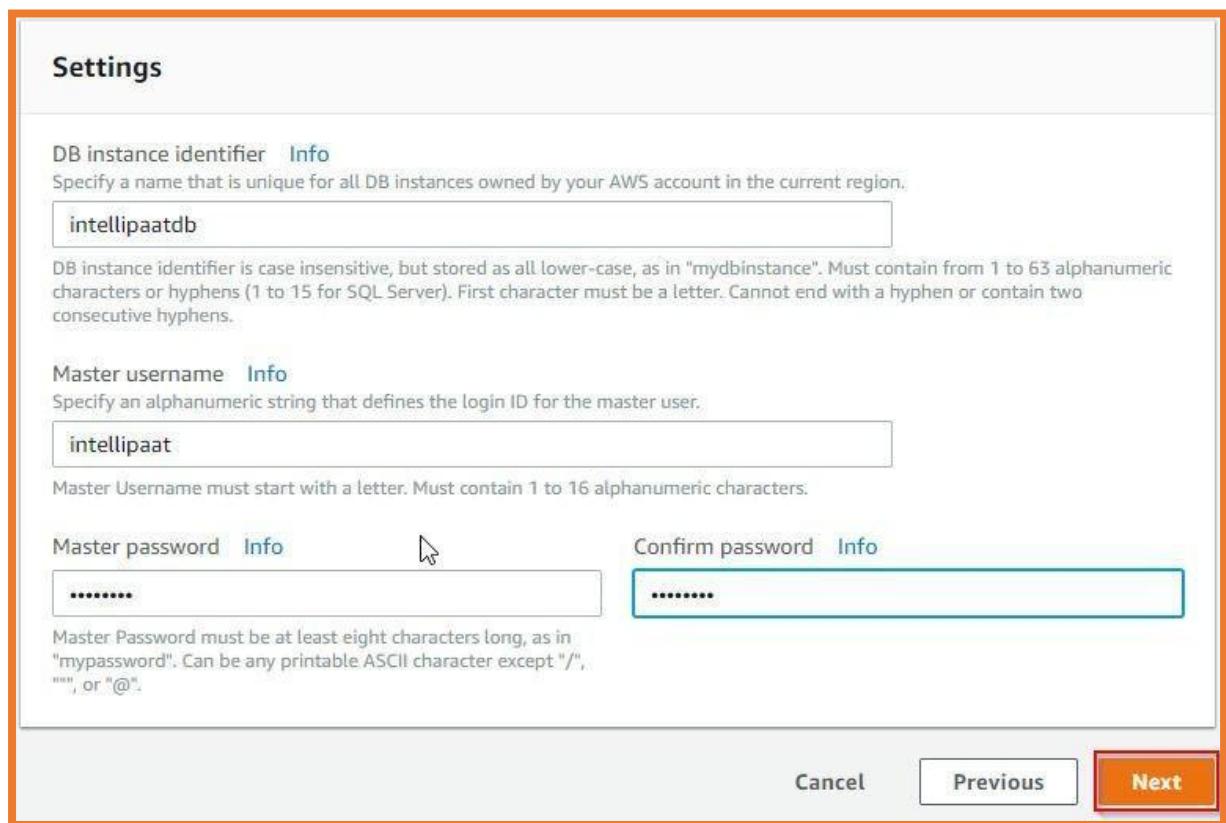
DB instance class [Info](#)
 db.t2.micro — 1 vCPU, 1 GiB RAM

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.

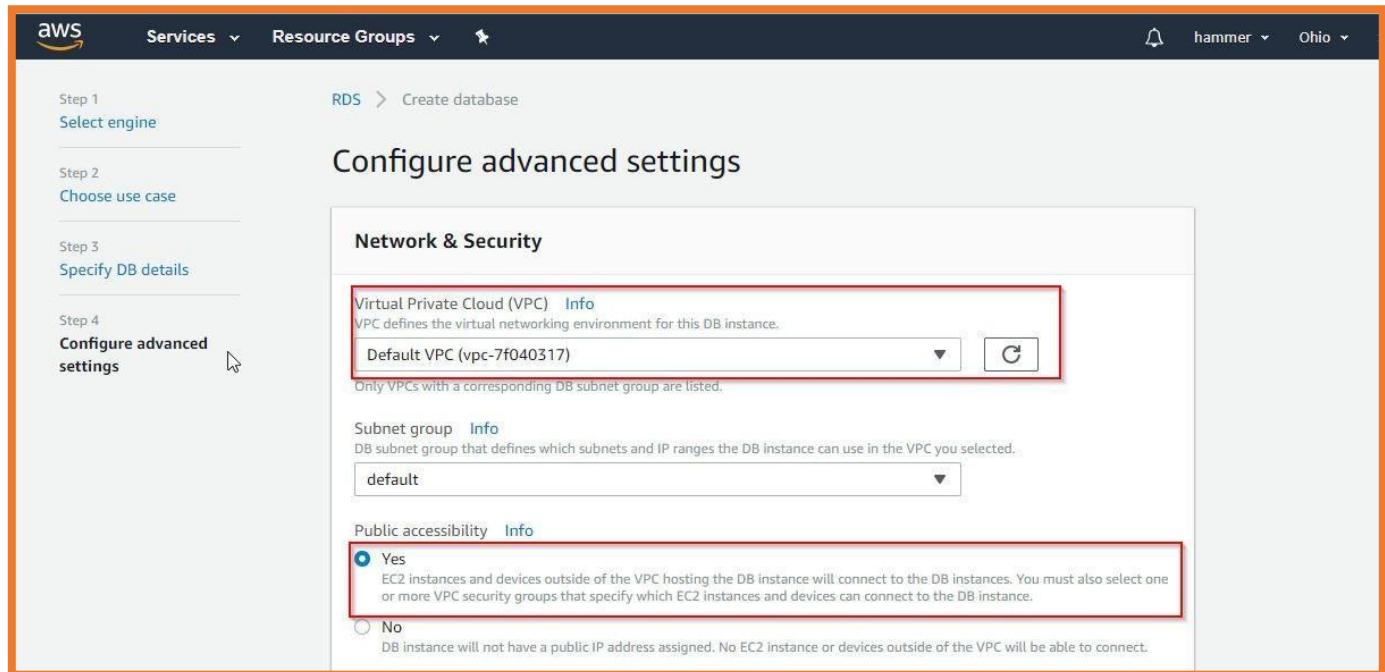
Master password [Info](#) 

Confirm password [Info](#)

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

Cancel **Previous** **Next**

- 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

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.

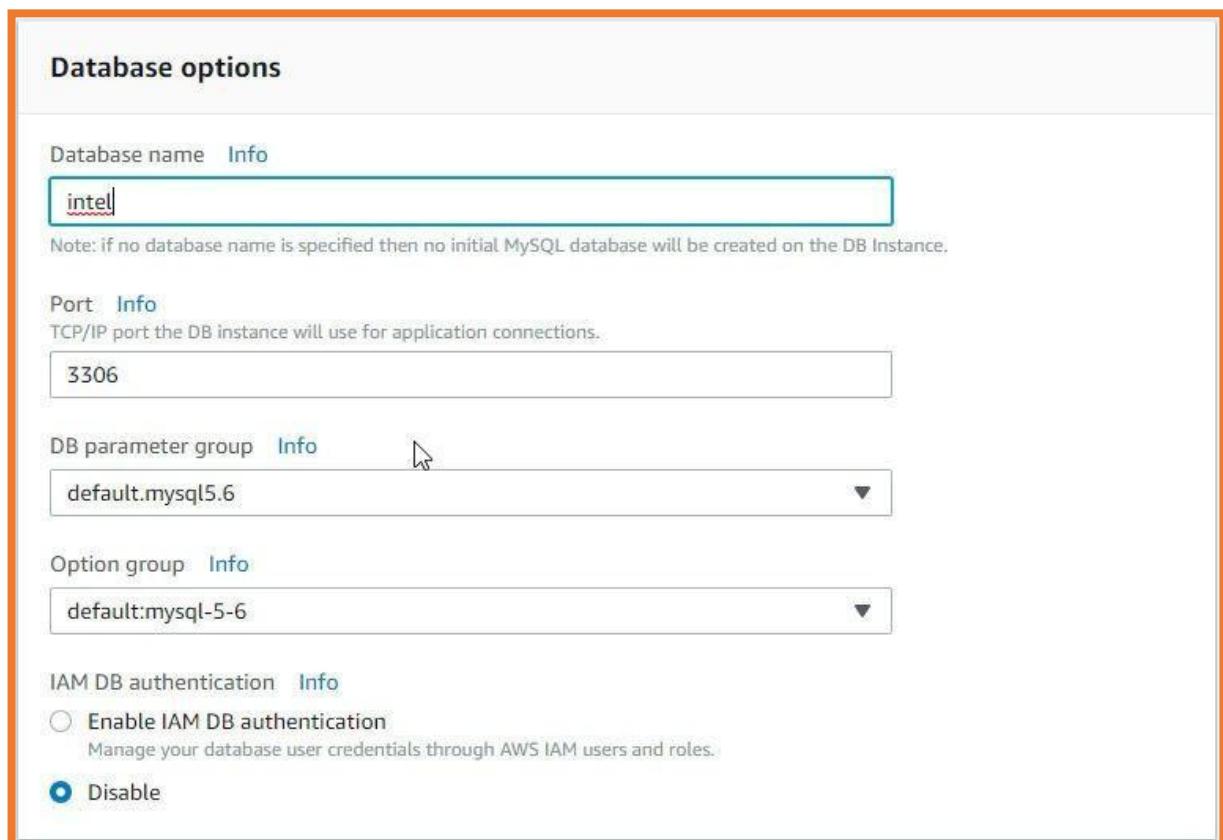
default

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](#)
intell

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

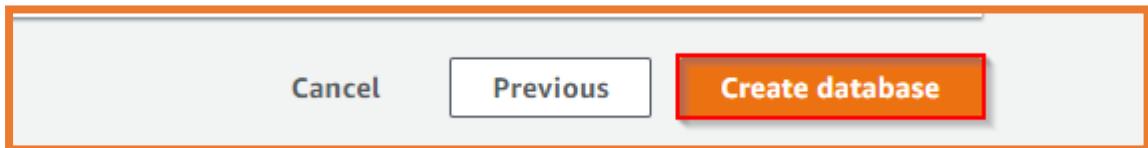
Option group [Info](#)
default:mysql-5-6

IAM DB authentication [Info](#)

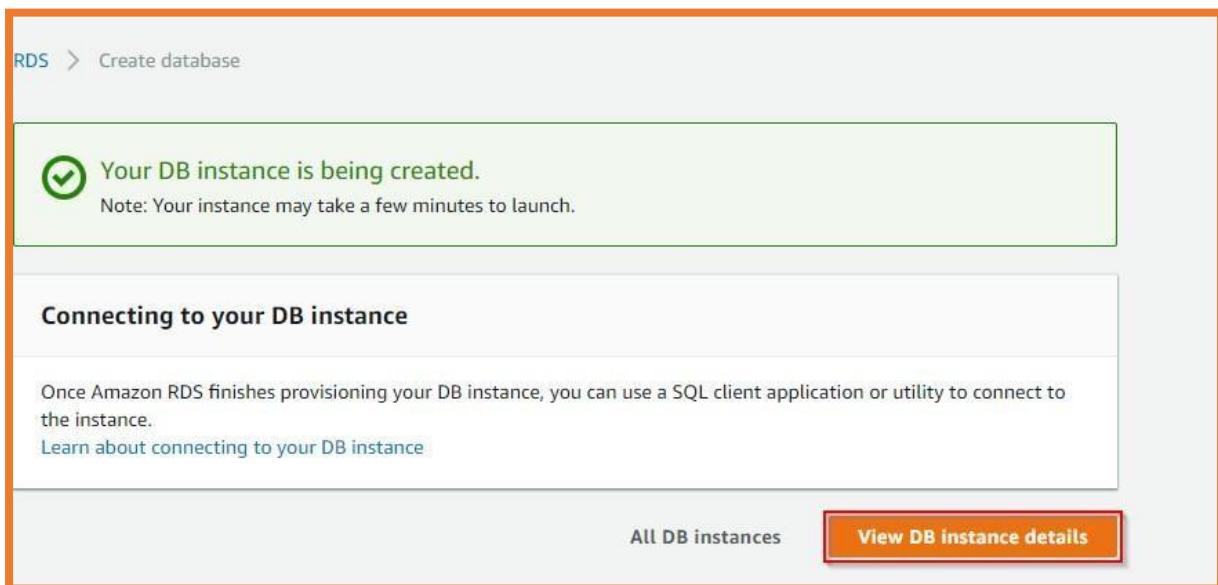
Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.

Disable

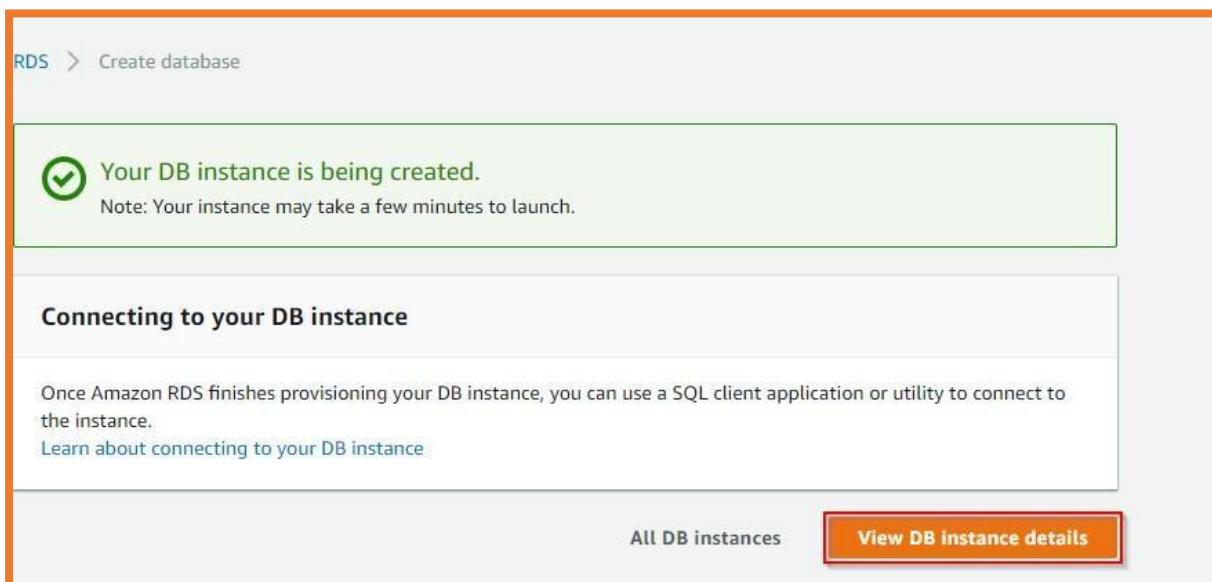
- Then click on Create database



- Then you can check your instance status

A screenshot of the Amazon RDS 'Create database' status page. It shows a success message: 'Your DB instance is being created.' with a green checkmark icon. Below it says 'Note: Your instance may take a few minutes to launch.' A section titled 'Connecting to your DB instance' contains the text: 'Once Amazon RDS finishes provisioning your DB instance, you can use a SQL client application or utility to connect to the instance.' and a link 'Learn about connecting to your DB instance'. At the bottom are two buttons: 'All DB instances' (grey) and 'View DB instance details' (orange). An orange border surrounds the entire screenshot.

- 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.

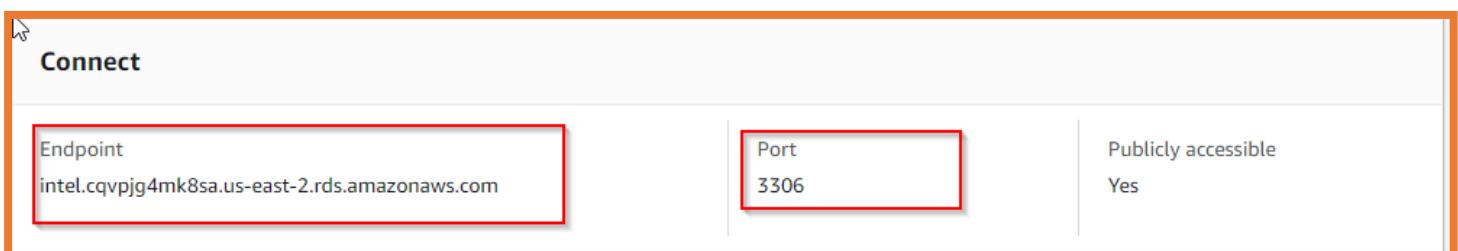
Connecting to your DB instance

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](#)

All DB instances View DB instance details

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

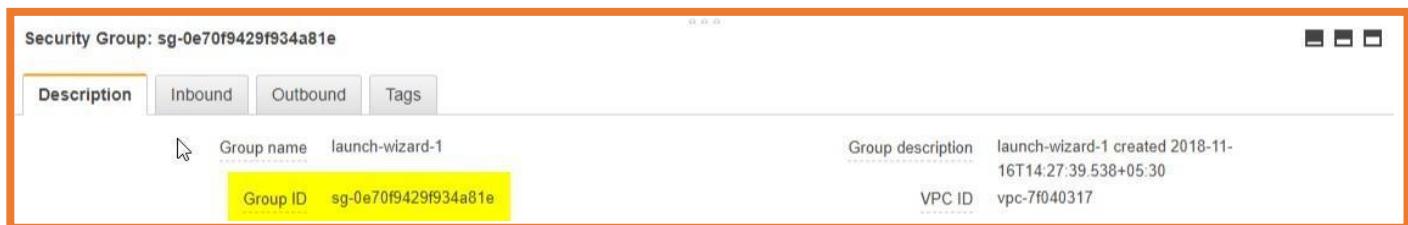


Endpoint: intel.cqvpjg4mk8sa.us-east-2.rds.amazonaws.com

Port: 3306

Publicly accessible: 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

Description	Inbound	Outbound	Tags
Group name: launch-wizard-1	Group description: launch-wizard-1 created 2018-11-16T14:27:39.538+05:30	VPC ID: vpc-7f040317	
Group ID: sg-0e70f9429f934a81e			

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



Type: MySQL/Aurora

Protocol: TCP

Port Range: 3306

Source: Custom 182.75.139.2/32

Description: 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

- Then paste the EC2 Security ID in Source> Custom > **Security Group** by keeping



the Type as MySQL/Aurora

- 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.

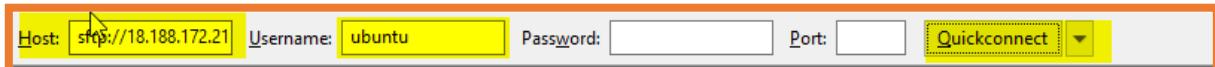
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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.



- 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

- Now go back to your PuTTY, where you will see that it contains the index.html file

```
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 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 removecommand

```
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

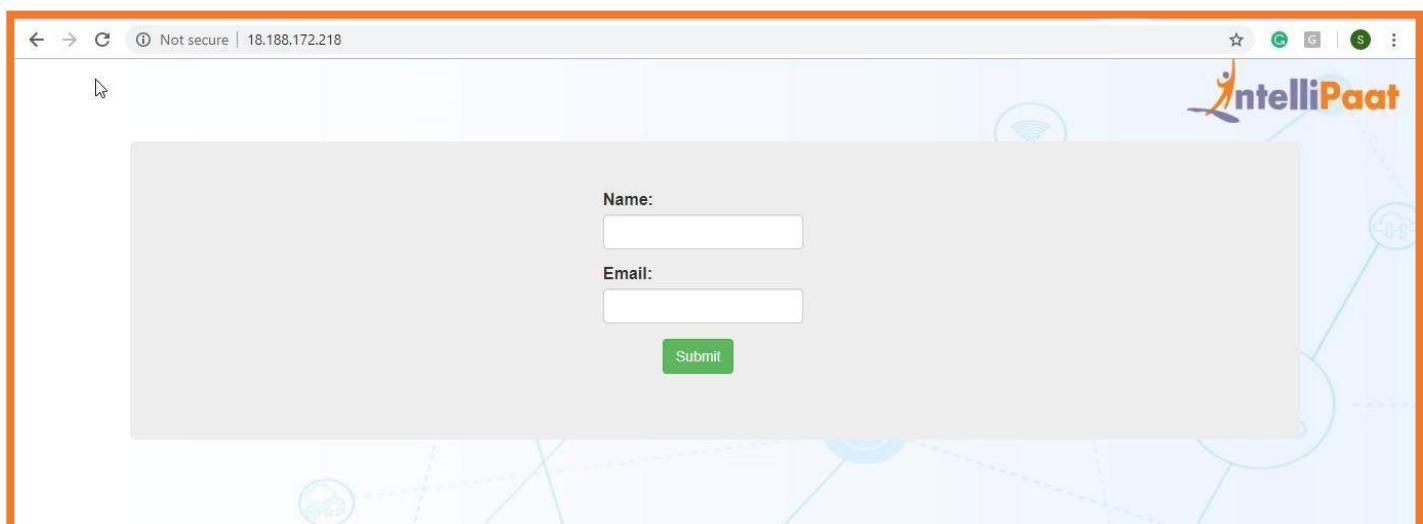
```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
</form></td>
<td colspan="4"></td>
</tr>
</table>
</div>
</div>
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "intellipaatdb.cqvpjg4mk8sa.us-east-2.rds.amazonaws.com";
$username = "intellipaat";
$password = "intell123";
$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

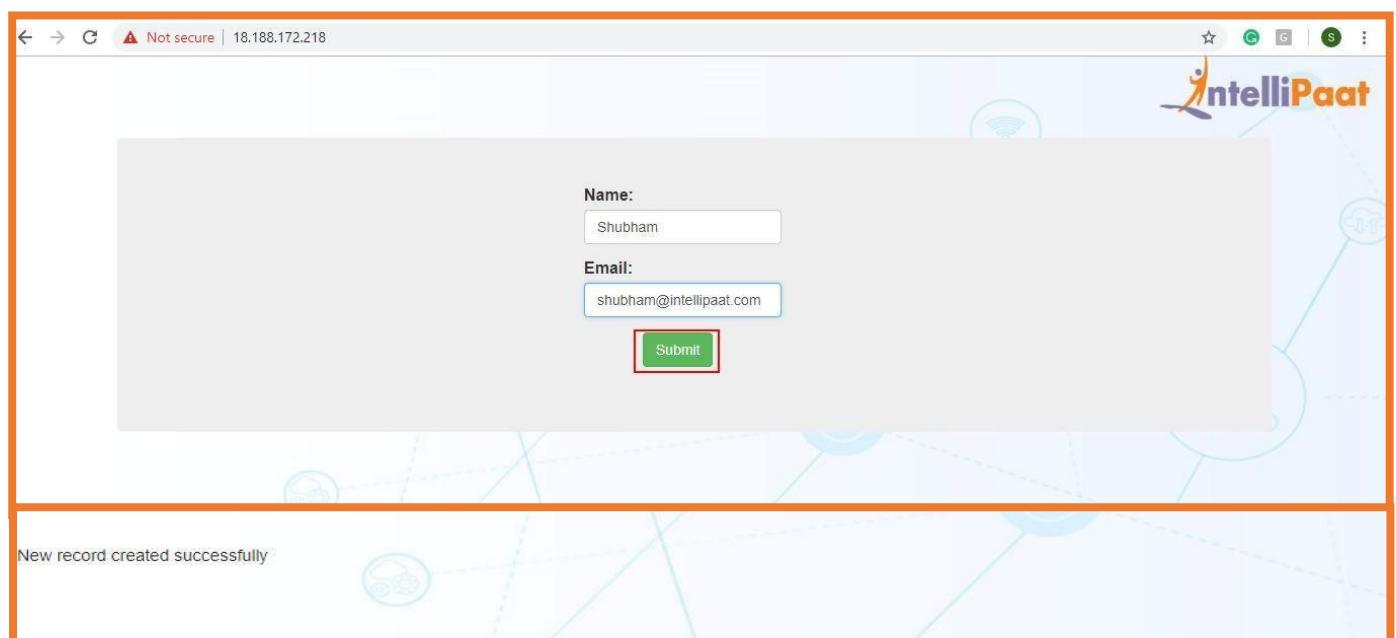


- After copying this IP to your browser, you will observe that your website is working on it



A screenshot of a web browser window. The address bar at the top shows "Not secure | 18.188.172.218". The main content area contains a form with two input fields: "Name:" and "Email:", both with empty white boxes. Below the fields is a green "Submit" button. The background of the page features a light blue and white abstract design with network-like nodes and lines.

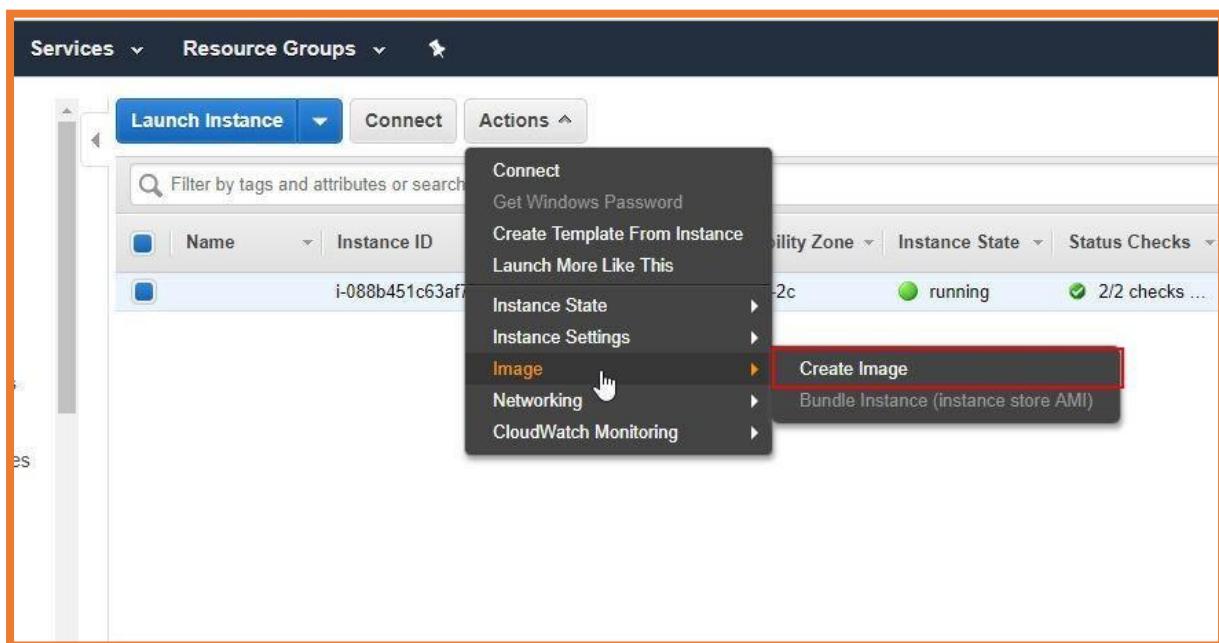
- Now when you enter these details in this website, you will see the following result



A screenshot of the same web browser window after the user has submitted the form. The "Name:" field now contains "Shubham" and the "Email:" field contains "shubham@intellipaat.com". The "Submit" button is highlighted with a red border. At the bottom of the page, a message "New record created successfully." is displayed in a green box.

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	<input checked="" type="checkbox"/>	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.

Create Image

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

