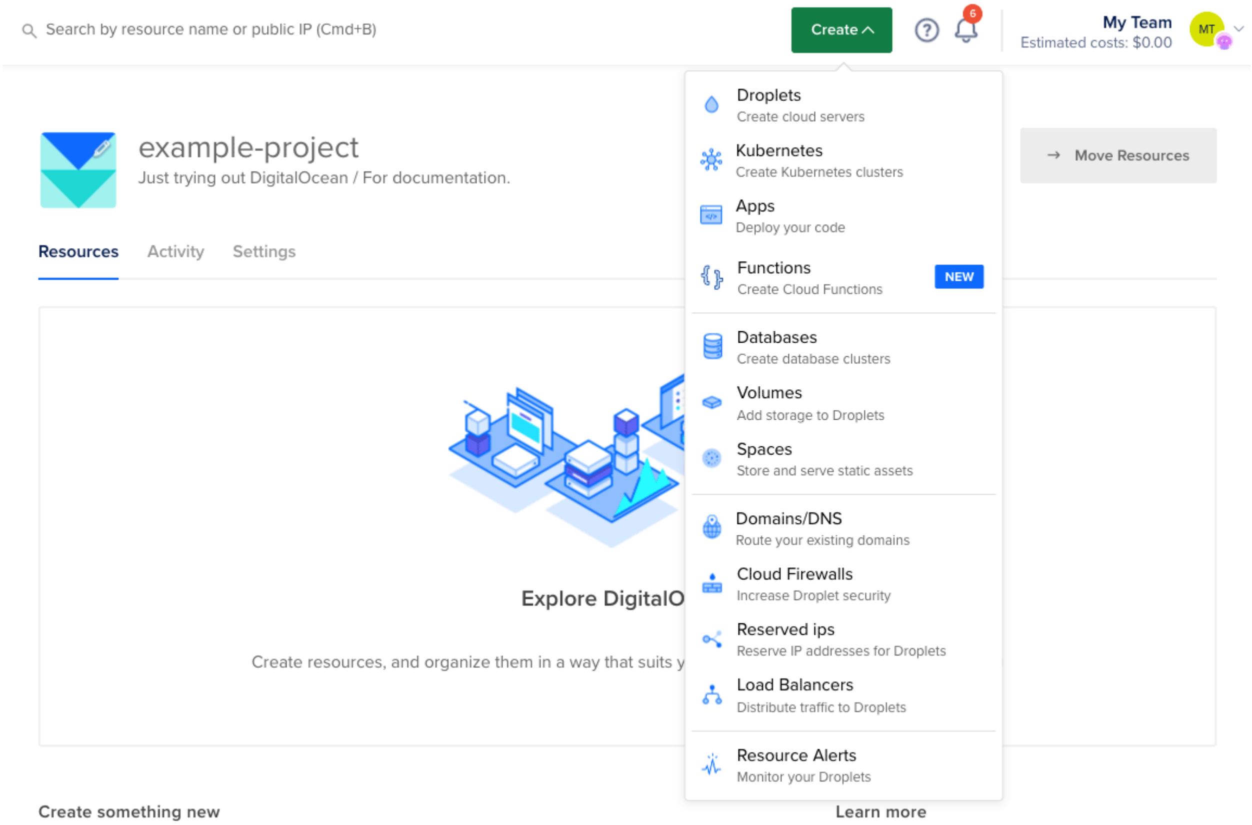
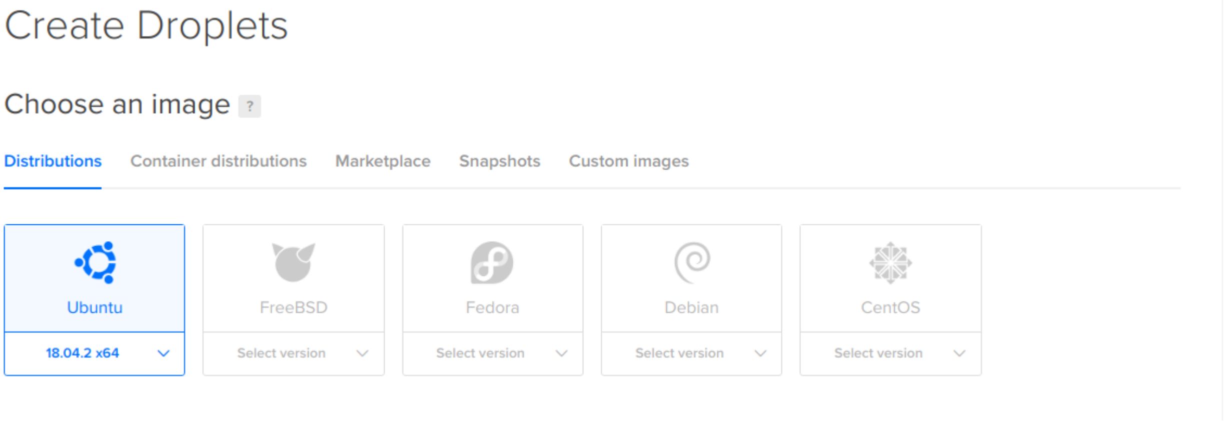
1) Digital Ocean server setup from scratch

1. **you log in to the control panel, and click the green Create button in the top right to open the create menu.**

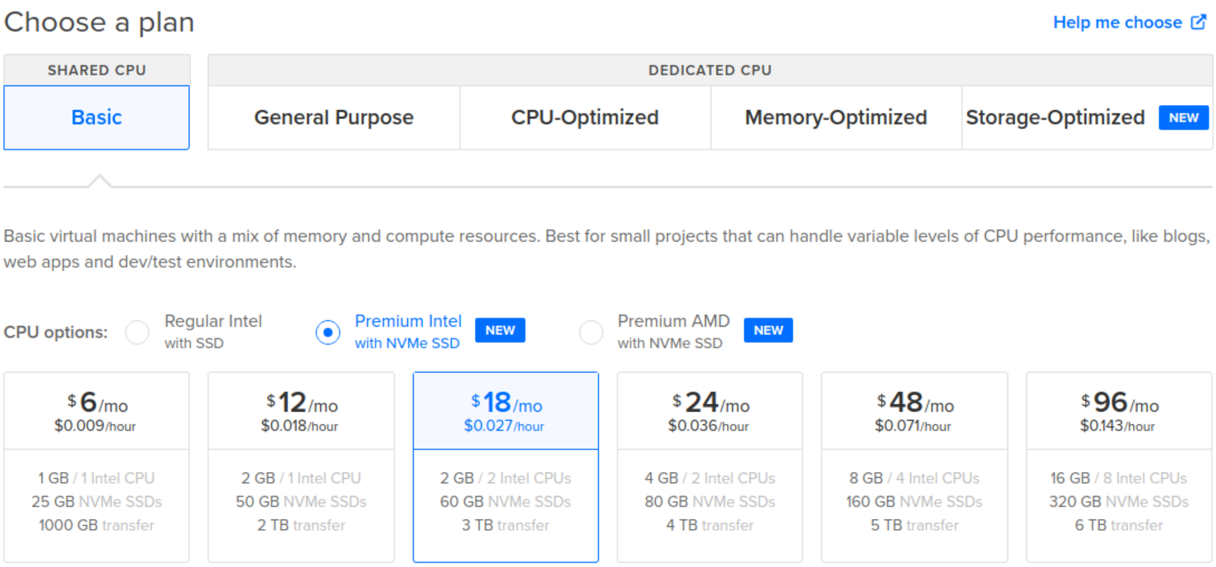


2. **Choose an image**
   ⇒ In the Choose an image section, choose the image your Droplet will be created from. Select ubuntu's latest version.



3. **Choose a plan**

⇒ In the Choose a plan section, you choose the amount of RAM, storage space, and CPU cores your Droplet will have.



Choose a plan

Help me choose

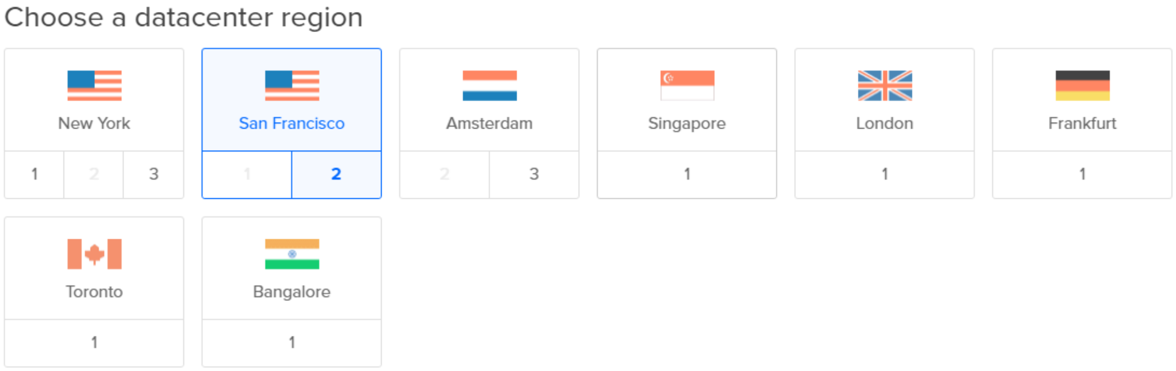
SHARED CPU	DEDICATED CPU			
Basic	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized NEW

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options: ○ Regular Intel with SSD ● Premium Intel with NVMe SSD NEW ○ Premium AMD with NVMe SSD NEW

$6/mo $0.009/hour	$12/mo $0.018/hour	$18/mo $0.027/hour	$24/mo $0.036/hour	$48/mo $0.071/hour	$96/mo $0.143/hour
1 GB / 1 Intel CPU 25 GB NVMe SSDs 1000 GB transfer	2 GB / 1 Intel CPU 50 GB NVMe SSDs 2 TB transfer	2 GB / 2 Intel CPUs 60 GB NVMe SSDs 3 TB transfer	4 GB / 2 Intel CPUs 80 GB NVMe SSDs 4 TB transfer	8 GB / 4 Intel CPUs 160 GB NVMe SSDs 5 TB transfer	16 GB / 8 Intel CPUs 320 GB NVMe SSDs 6 TB transfer

4. **Choose a data centre region**
   ⇒A good default is selected for you, but for the best performance and minimal latency, choose the datacenter nearest to you and your users. You can also make a decision based on which products and features are available in which regions.



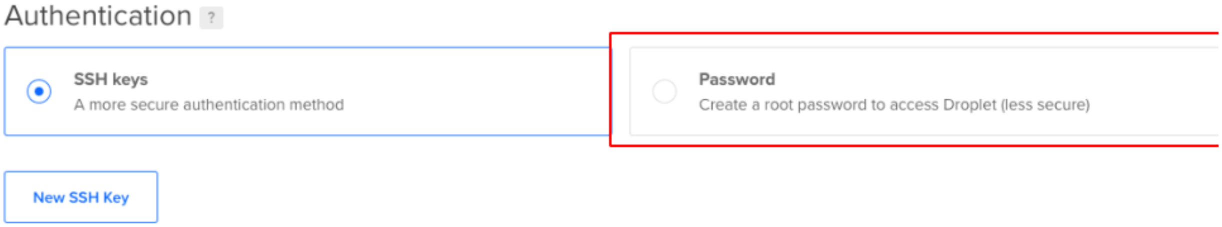
Choose a datacenter region

New York	San Francisco	Amsterdam	Singapore	London	Frankfurt
1 2 3	1 2	2 3	1	1	1

Toronto	Bangalore
1	1

5. **Authentication**
   ⇒ In the Authentication section, you choose the method you want to use to log in to your Droplet. There are two options:

   1. **SSH keys, which provide more security than a password.**
   2. **Password, which allows you to create your own password for the new Droplet.**



Authentication ?

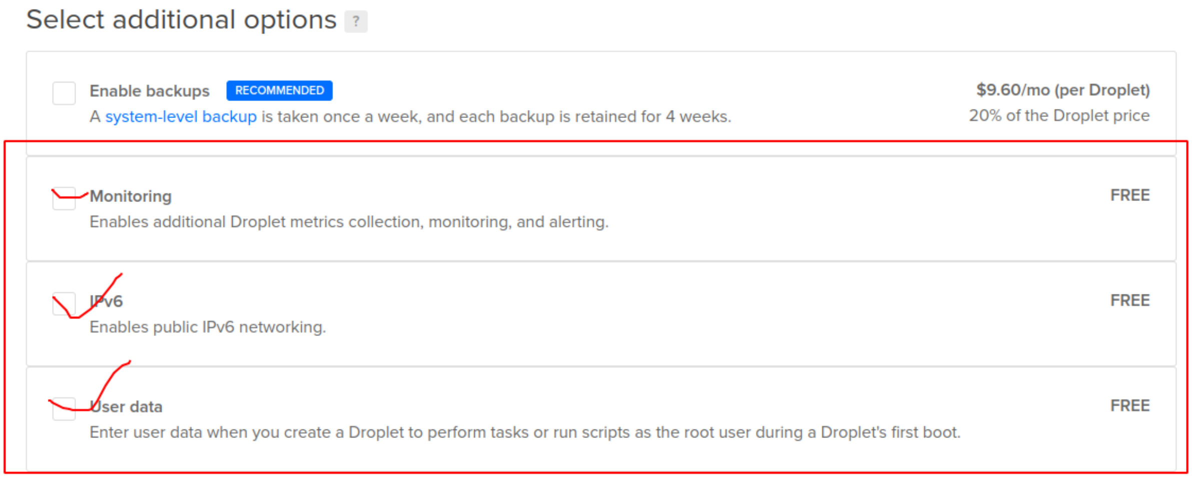
● SSH keys
A more secure authentication method

○ Password
Create a root password to access Droplet (less secure)

6. Select additional options

⇒ In the Select additional options section, you can enable several optional services that add functionality to your Droplet.

- Monitoring (free) adds the DigitalOcean agent to collect extended metrics and create alert policies.
- IPv6 (free) enables IPv6 access for your Droplet.
- User data (free) is arbitrary data that you specify which is written to the user-data field of the DigitalOcean metadata service. Droplets running distributions with cloud-init can consume and execute the data from this field, which are generally cloud-config files used for initially configuring a server on first boot.



Select additional options ?

□ Enable backups RECOMMENDED — $9.60/mo (per Droplet)
A system-level backup is taken once a week, and each backup is retained for 4 weeks. — 20% of the Droplet price

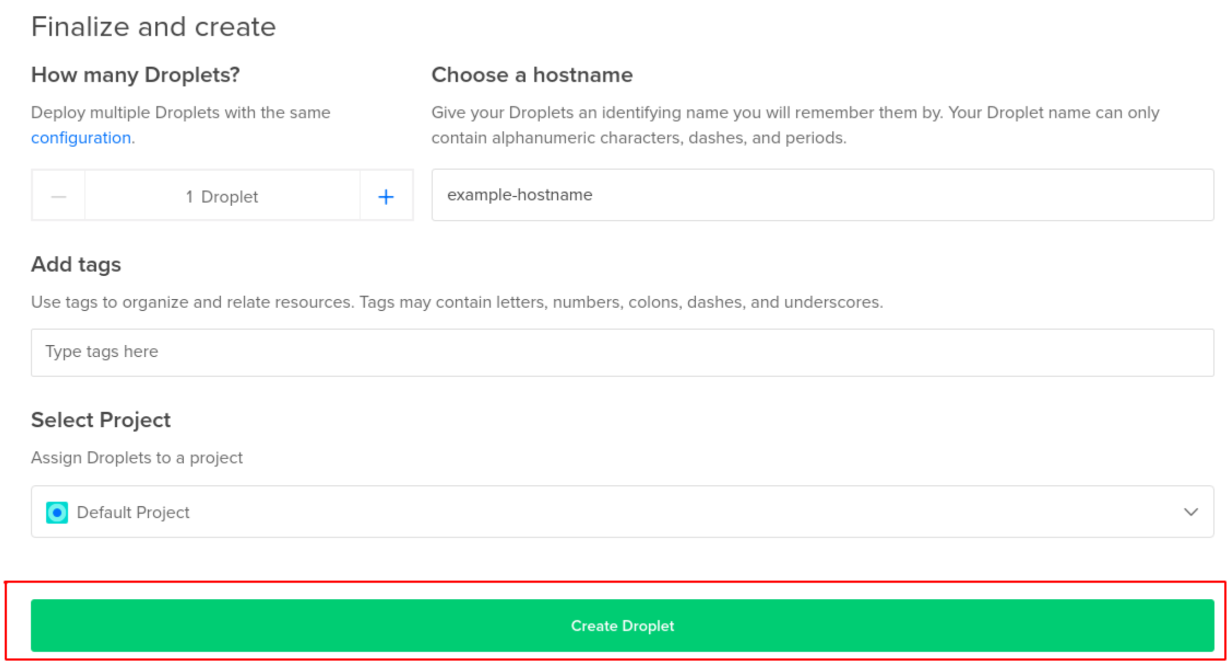
☑ Monitoring — FREE
Enables additional Droplet metrics collection, monitoring, and alerting.

☑ IPv6 — FREE
Enables public IPv6 networking.

☑ User data — FREE
Enter user data when you create a Droplet to perform tasks or run scripts as the root user during a Droplet's first boot.

7. Finalize and create

⇒ In the Finalize and create section, you specify the quantity, name, tags, and project for the Droplet you're creating.



Finalize and create

How many Droplets?

Deploy multiple Droplets with the same configuration.

— 1 Droplet +

Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

example-hostname

Add tags

Use tags to organize and relate resources. Tags may contain letters, numbers, colons, dashes, and underscores.

Type tags here

Select Project

Assign Droplets to a project

Default Project

Create Droplet

Once you have selected your options, click Create. A progress bar displays how close your Droplet is to being ready.

8. Connect to Droplets with SSH
⇒ How to Connect to your Droplet with PuTTY on Windows
⇒ Step of connecting to droplets using ssh

2) How To Install Linux, Apache, MySQL, PHP (LAMP)

1. Installing Apache

The Apache web server is among the most popular web servers in the world. It's well documented, has an active community of users, and has been in wide use for much of the history of the web, which makes it a great choice for hosting a website.

Start by updating the package manager cache. If this is the first time you're using `sudo` within this session, you'll be prompted to provide your user's password to confirm you have the right privileges to manage system packages with `apt`.

```
sudo apt update
```

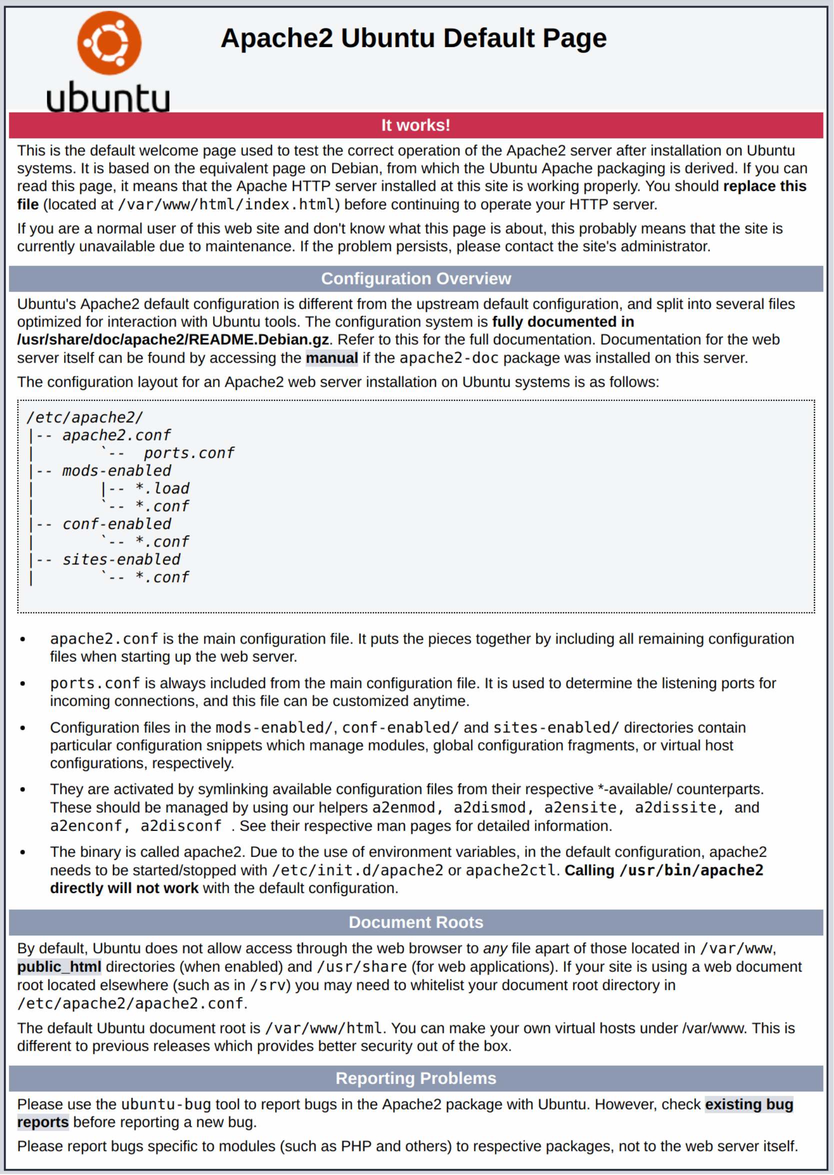
Then, install Apache with:

```
sudo apt install apache2
```

You can do a spot check right away to verify that everything went as planned by visiting your server's public IP address in your web browser (see the note under the next heading to find out what your public IP address is if you do not have this information already):

```
http://your_server_ip
```

You'll see the default Ubuntu 20.04 Apache web page, which is there for informational and testing purposes. It should look something like this:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

2. Installing MySQL

Now that you have a web server up and running, you need to install the database system to be able to store and manage data for your site. MySQL is a popular database management system used within PHP environments.

Again, use `apt` to acquire and install this software:

```
sudo apt install mysql-server
```

When prompted, confirm installation by typing `Y`, and then `ENTER.`

When the installation is finished, it's recommended that you run a security script that comes pre-installed with MySQL. This script will remove some insecure default settings and lock down access to your database system. Start the interactive script by running:

```
sudo mysql_secure_installation
```

This will ask if you want to configure the `VALIDATE PASSWORD PLUGIN`.

Answer `Y` for yes, or anything else to continue without enabling.

```
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?
Press y|Y for Yes, any other key for No:
```

If you answer "yes", you'll be asked to select a level of password validation. Keep in mind that if you enter `2` for the strongest level, you will receive errors when attempting to set any password which does not contain numbers, upper and lowercase letters, and special characters, or which is based on common dictionary words.

```
There are three levels of password validation policy:
LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
file
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
```

If you enabled password validation, you'll be shown the password strength for the root password you just entered and your server will ask if you want to continue with that password. If you are happy with your current password, enter `Y` for "yes" at the prompt:

```
Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any
other key for No)
```

For the rest of the questions, press `Y` and hit the `ENTER` key at each prompt. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes you have made.

When you're finished, test if you're able to log in to the MySQL console by typing:

```
sudo mysql
```

To exit the MySQL console, type:

```
Output

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.19-0ubuntu5 (Ubuntu)
Copyright (c) 2000, 2020, 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> exit
```

3. Installing PHP

You have Apache installed to serve your content and MySQL installed to store and manage your data. PHP is the component of our setup that will process code to display dynamic content to the final user. In addition to the `php` package, you'll need `php-mysql`, a PHP module that allows PHP to communicate with MySQL-based databases. You'll also need `libapache2-mod-php` to enable Apache to handle PHP files. Core PHP packages will automatically be installed as dependencies.

To install these packages, run:

```
sudo apt install php libapache2-mod-php php-mysql
```

Once the installation is finished, you can run the following command to confirm your PHP version:

```
php -v
```

```
Output
PHP 7.4.3 (cli) (built: Jul  5 2021 15:13:35) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
```

4. Installing PHPMYADMIN

You can use APT to install phpMyAdmin from the default Ubuntu repositories.

As your non-root sudo user, update your server's package index:

```
sudo apt update
```

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

Here are the options you should choose when prompted in order to configure your installation correctly:

- For the server selection, choose `apache2`

Warning: When the prompt appears, "apache2" is highlighted, but not selected. If you do not hit `SPACE` to select Apache, the installer will *not* move the necessary files during installation. Hit `SPACE`, `TAB`, and then `ENTER` to select Apache.

- Select `Yes` when asked whether to use `dbconfig-common` to set up the database
- You will then be asked to choose and confirm a MySQL application password for phpMyAdmin

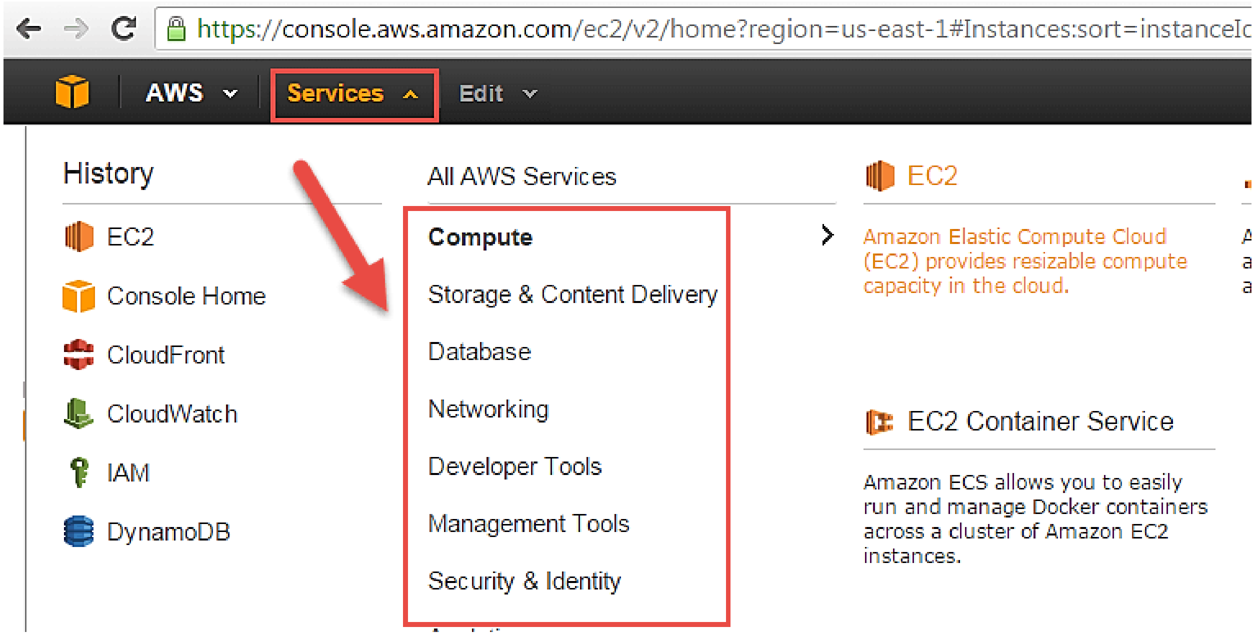
  ```
  sudo phpenmod mbstring
  sudo systemctl restart apache2
  ```

3) How to setup aws ec2 instance

Login and access to AWS services

Step 1) In this step,

- **Login to your AWS account and go to the AWS Services tab at the top left corner.**
- **Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Computeà EC2 as in the next step.**



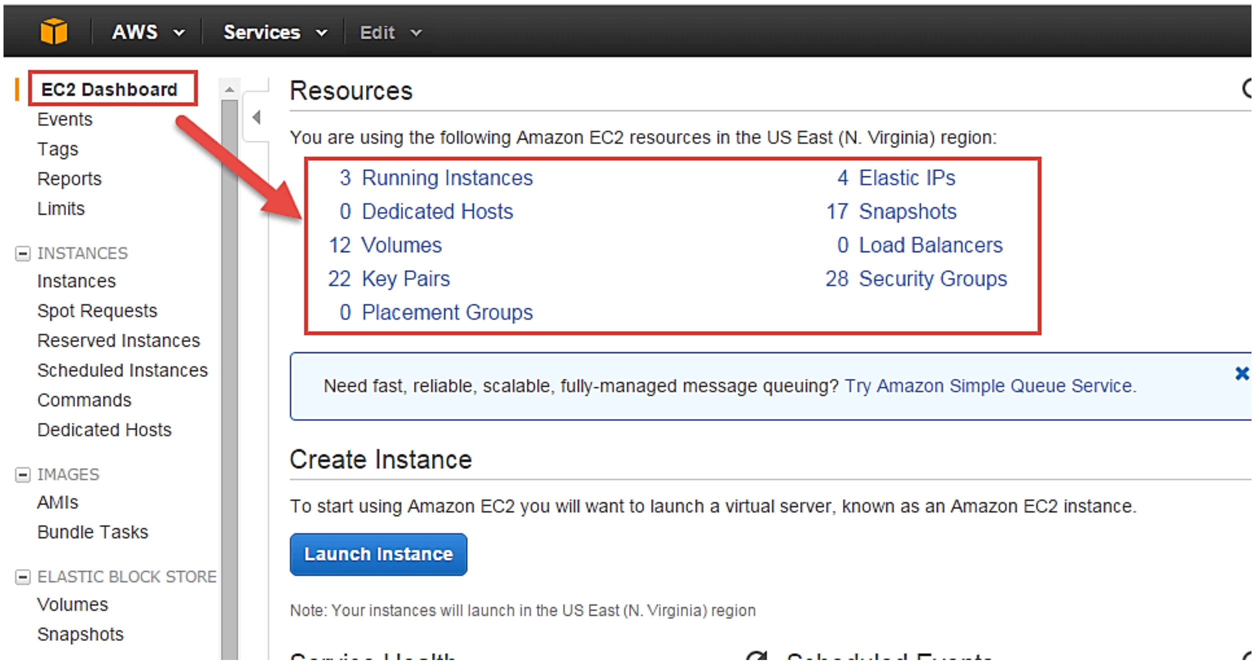
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceIc

Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.

Amazon ECS allows you to easily run and manage Docker containers across a cluster of Amazon EC2 instances.

- **Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.**

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.



Resources

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

3 Running Instances	4 Elastic IPs
0 Dedicated Hosts	17 Snapshots
12 Volumes	0 Load Balancers
22 Key Pairs	28 Security Groups
0 Placement Groups	

Need fast, reliable, scalable, fully-managed message queuing? Try Amazon Simple Queue Service.

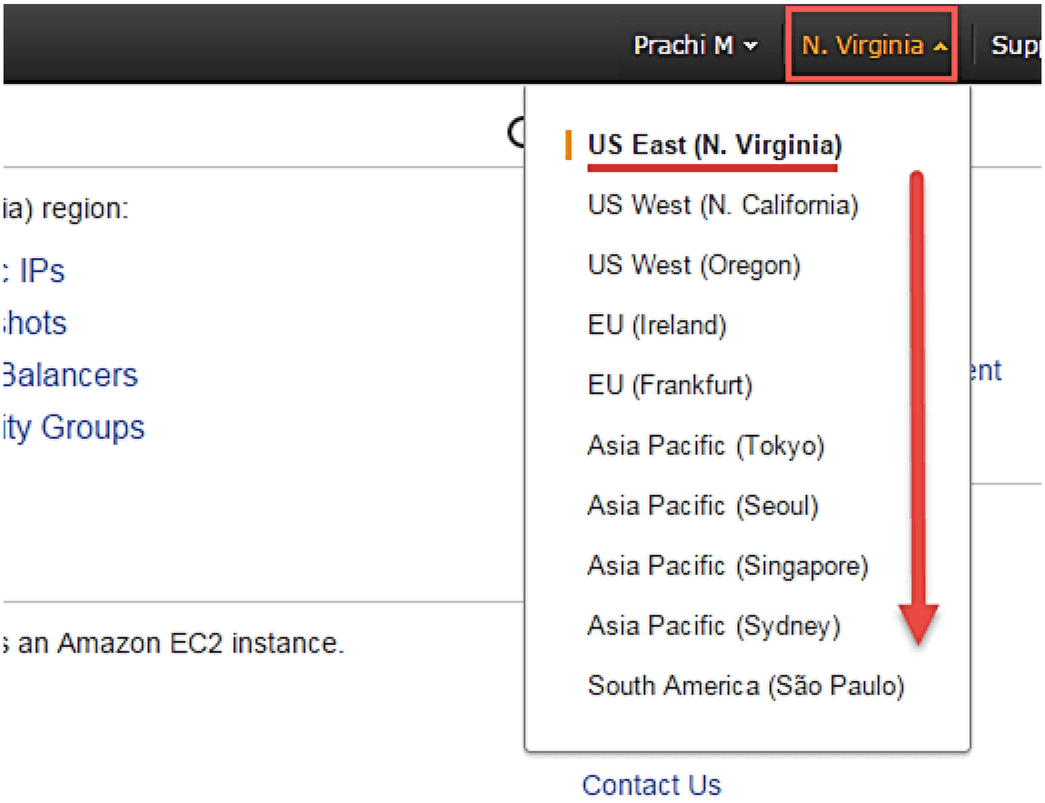
Create Instance

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

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

Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe.



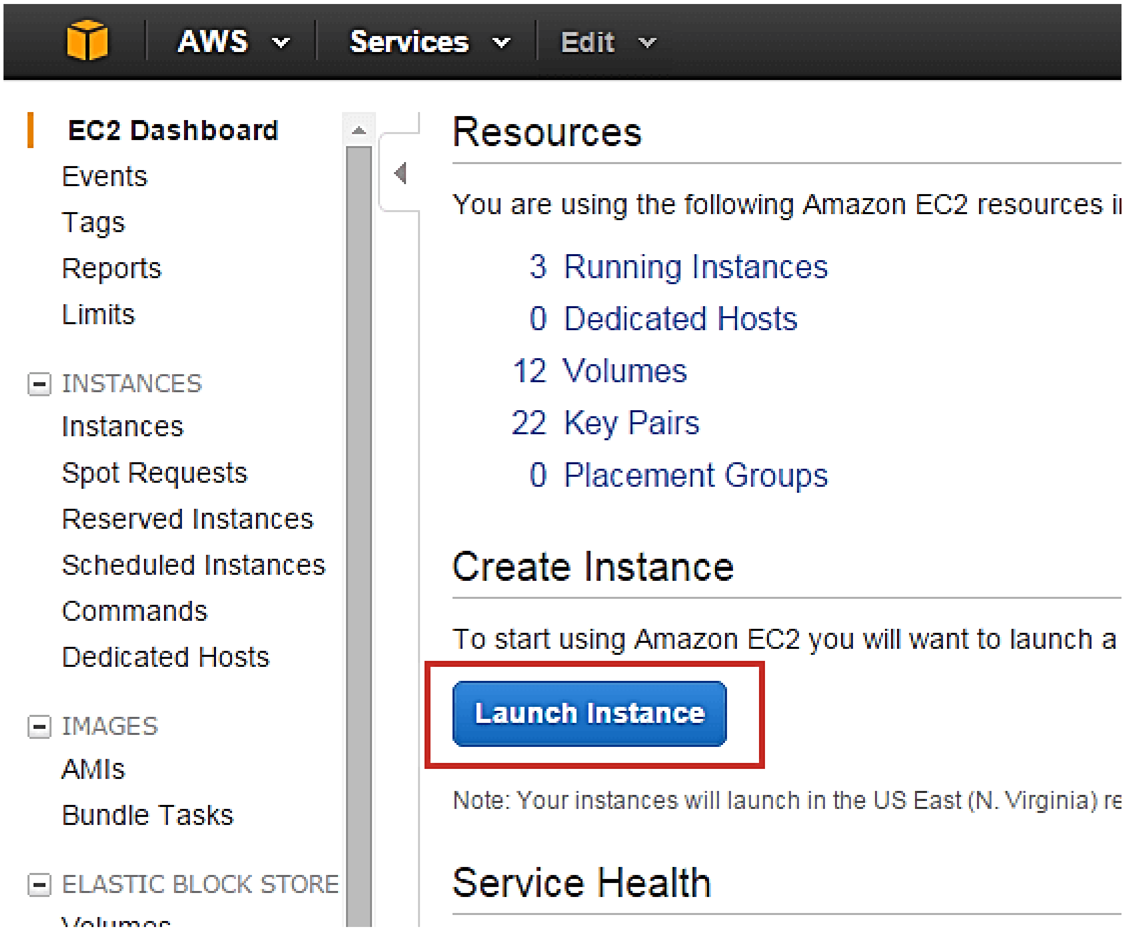
ia) region:

: IPs
hots
3alancers
ity Groups

s an Amazon EC2 instance.

Step 3) In this step

- **Once your desired Region is selected, come back to the EC2 Dashboard.**
- **Click on 'Launch Instance' button in the section of Create Instance (as shown below).**



Resources

You are using the following Amazon EC2 resources i

3 Running Instances
0 Dedicated Hosts
12 Volumes
22 Key Pairs
0 Placement Groups

Create Instance

To start using Amazon EC2 you will want to launch a

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

Service Health

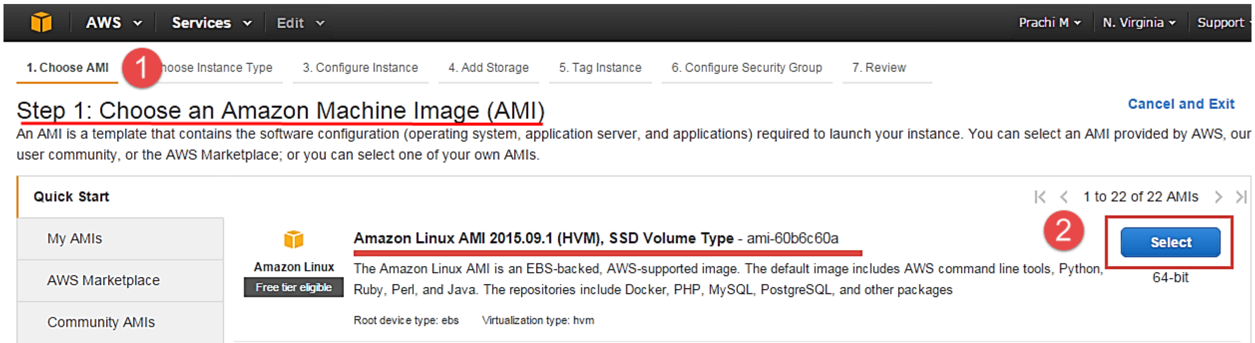
- **Instance creation wizard page will open as soon as you click 'Launch Instance'.**

Choose AMI

Step 1) In this step we will do,

1. **You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance). Once you launch an EC2 instance from your preferred AMI, the instance will automatically be booted with the desired OS. (We will see more about AMIs in the coming part of the tutorial).**

2. Here we are choosing the default Amazon Linux (64 bit) AMI.



Step 1: Choose an Amazon Machine Image (AMI)

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

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

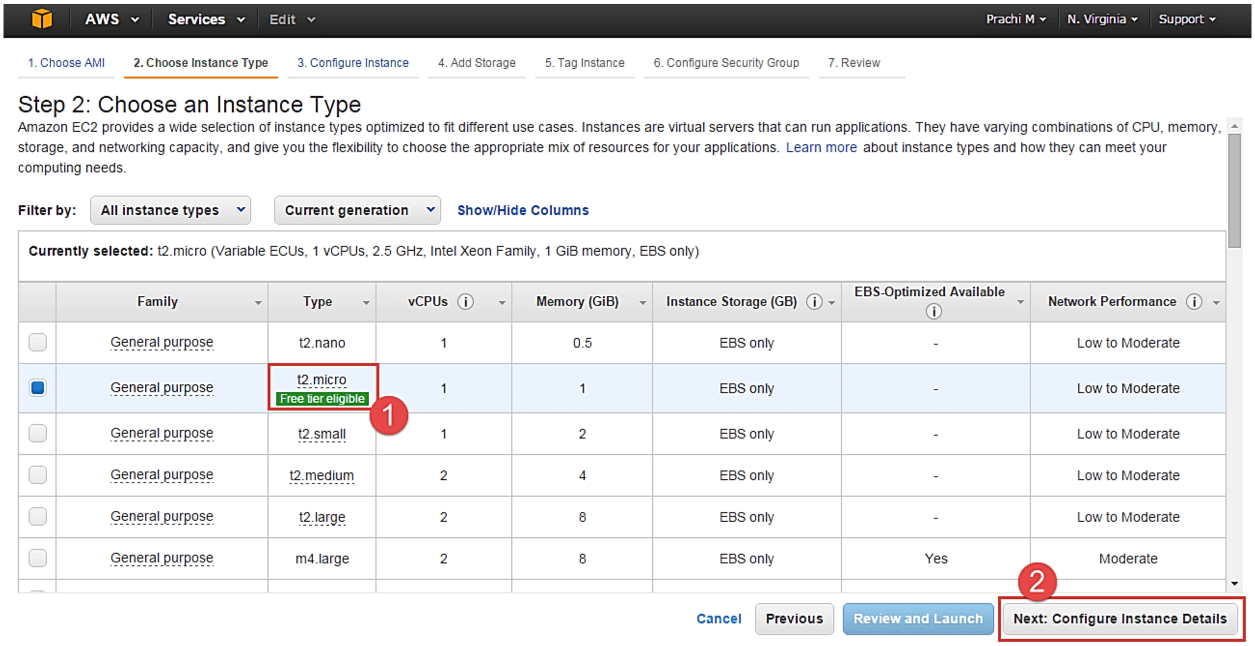
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

Root device type: ebs   Virtualization type: hvm

Choose EC2 Instance Types

Step 1) In the next step, you have to choose the type of instance you require based on your business needs.

1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on “Configure Instance Details” for further configurations



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.

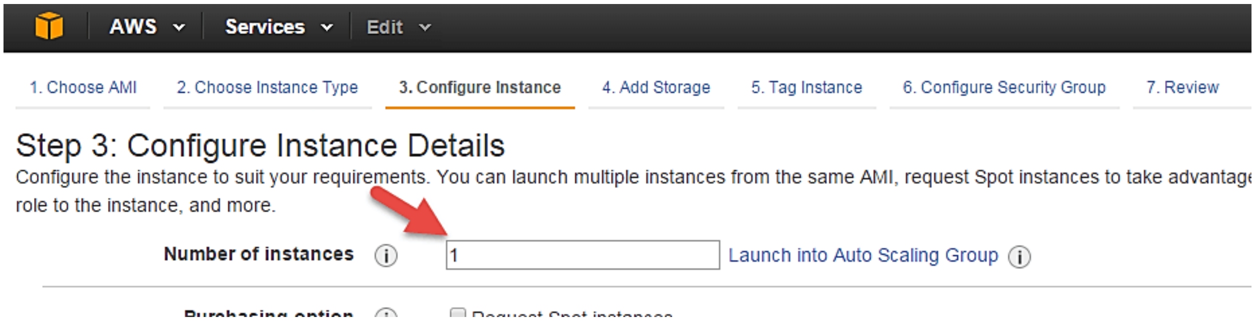
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate

- In the next step of the wizard, enter details like no. of instances you want to launch at a time.
- Here we are launching one instance.

Configure Instance

Step 1) No. of instances- you can provision up to 20 instances at a time. Here we are launching one instance.



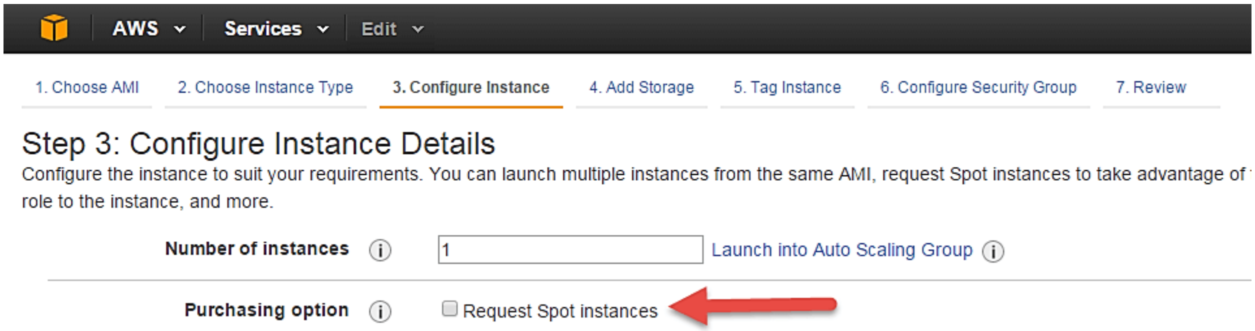
Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantag
role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Step 2) Under Purchasing Options, keep the option of 'Request Spot Instances' unchecked as of now. (This is done when we wish to launch Spot instances instead of on-demand ones. We will come back to Spot instances in the later part of the tutorial).



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
role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

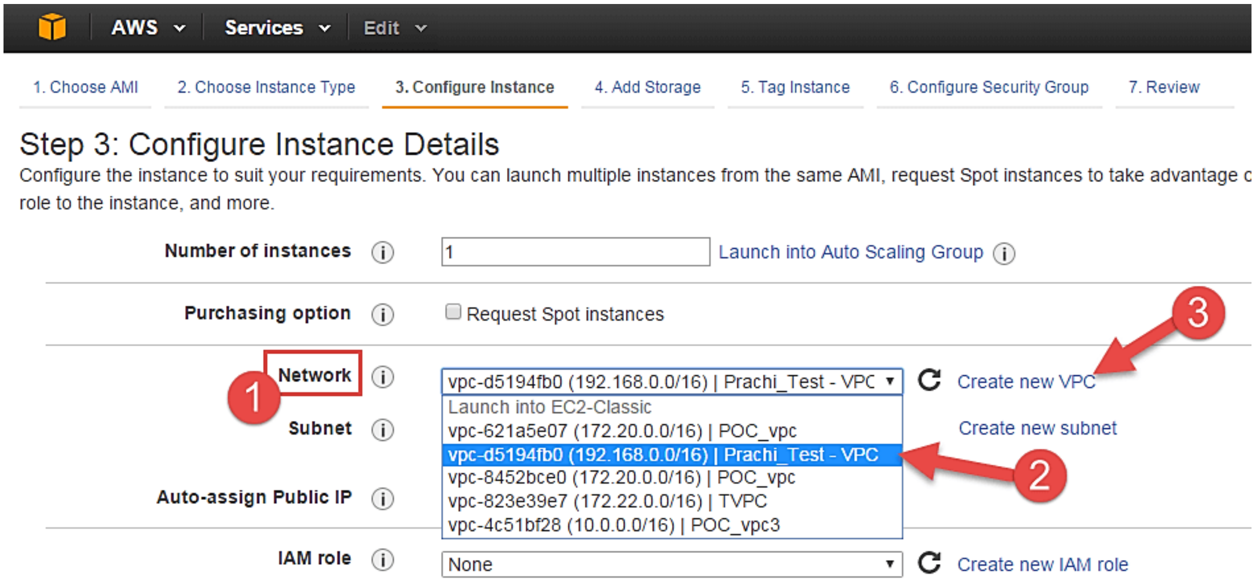
Step 3) Next, we have to configure some basic networking details for our EC2 server.

- **You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for better management. (We will see how to create a new VPC in Networking section of the tutorial.**
- **Subnetting should also be pre-planned. E.g.: If it's a web server you should place it in the public subnet and if it's a DB server, you should place it in a private subnet all inside your VPC.**

Below,

1. **Network section will give a list of VPCs available in our platform.**
2. **Select an already existing VPC**
3. **You can also create a new VPC**

Here I have selected an already existing VPC where I want to launch my instance.



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 c role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

- Launch into EC2-Classic
- vpc-621a5e07 (172.20.0.0/16) | POC_vpc
- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC
- vpc-8452bce0 (172.20.0.0/16) | POC_vpc
- vpc-823e39e7 (172.22.0.0/16) | TVPC
- vpc-4c51bf28 (10.0.0.0/16) | POC_vpc3

Subnet

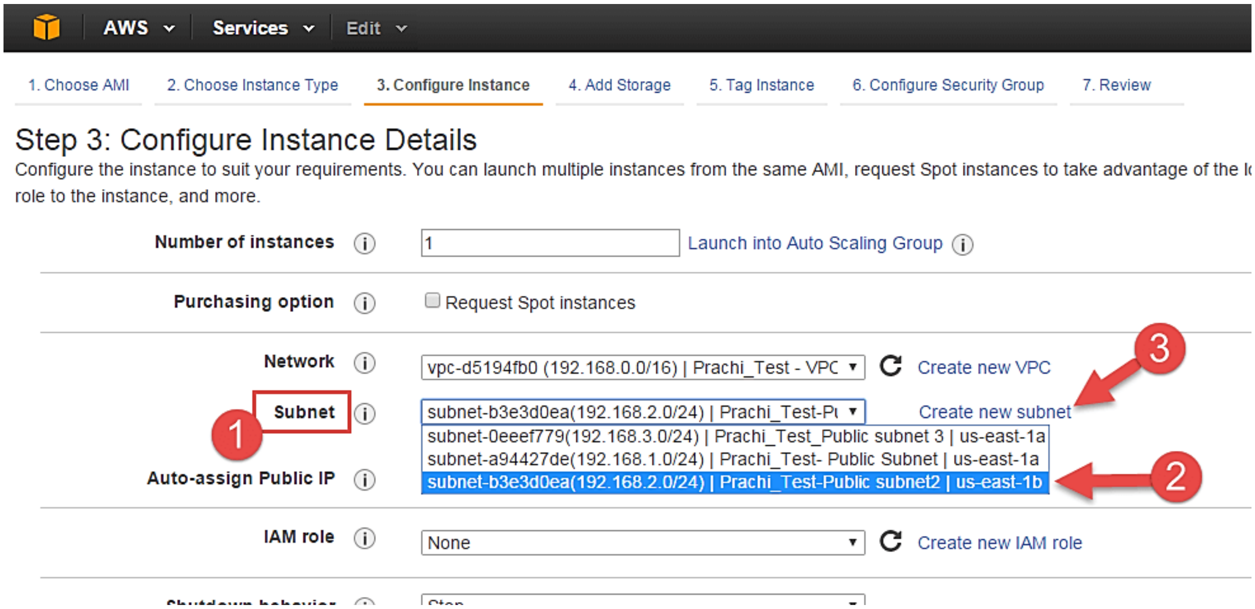
Auto-assign Public IP

IAM role: None

Step 4) In this step,

- **A VPC consists of subnets, which are IP ranges that are separated for restricting access.**
- **Below,**

1. **Under Subnets, you can choose the subnet where you want to place your instance.**
2. **I have chosen an already existing public subnet.**
3. **You can also create a new subnet in this step.**



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 l role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu

- subnet-0eeef779(192.168.3.0/24) | Prachi_Test_Public subnet 3 | us-east-1a
- subnet-a94427de(192.168.1.0/24) | Prachi_Test- Public Subnet | us-east-1a
- subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public subnet2 | us-east-1b

Auto-assign Public IP

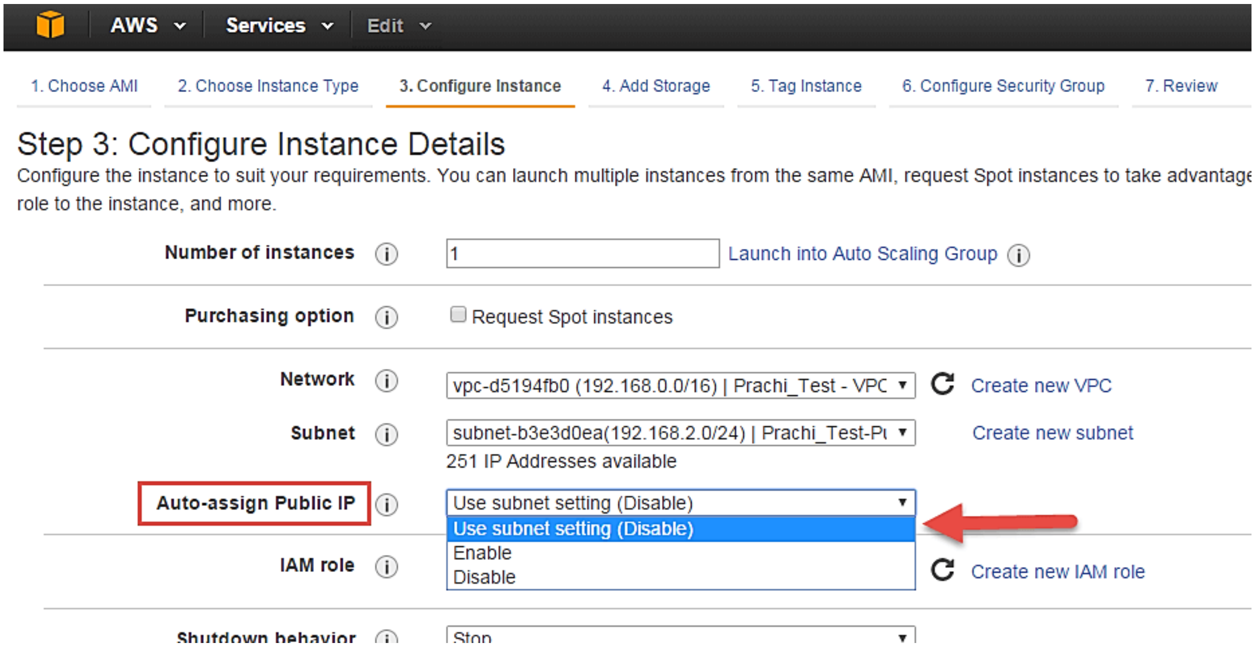
IAM role: None

Shutdown behavior: Stop

- **Once your instance is launched in a public subnet, AWS will assign a dynamic public IP to it from their pool of IPs.**

Step 5) In this step,

- **You can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.**
- **Here we are going to assign this instance a static IP called as EIP (Elastic IP) later. So we keep this feature disabled as of now.**



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
role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet
251 IP Addresses available

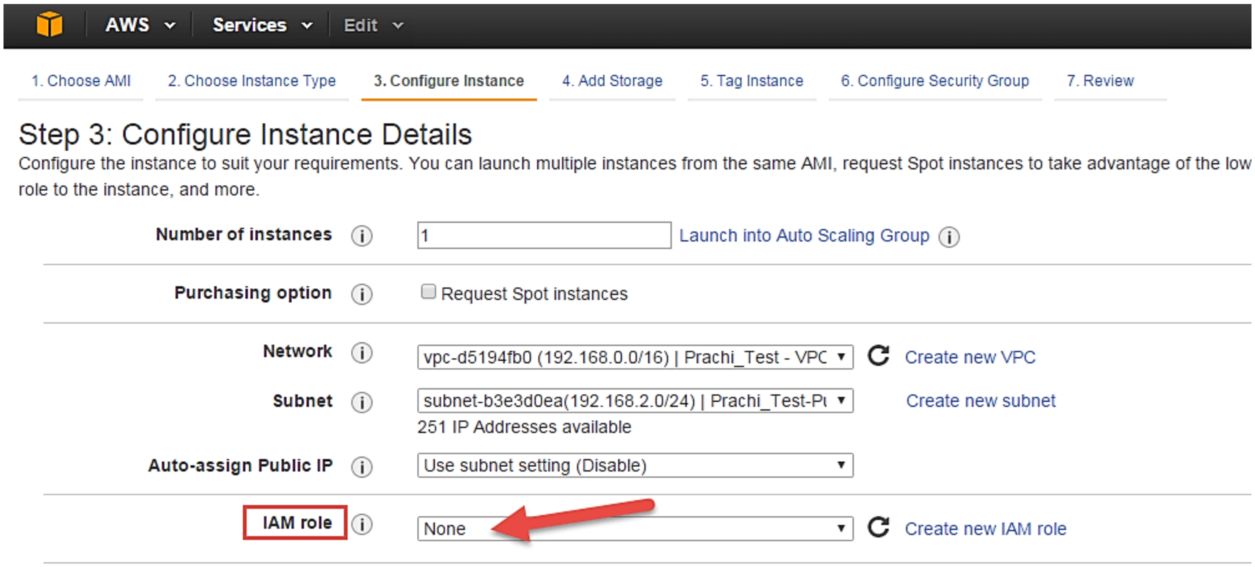
Auto-assign Public IP: Use subnet setting (Disable)
- Use subnet setting (Disable)
- Enable
- Disable

IAM role — Create new IAM role

Shutdown behavior: Stop

Step 6) In this step,

- **In the following step, keep the option of IAM role 'None' as of now. We will visit the topic of IAM role in detail in IAM services.**



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 low role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet

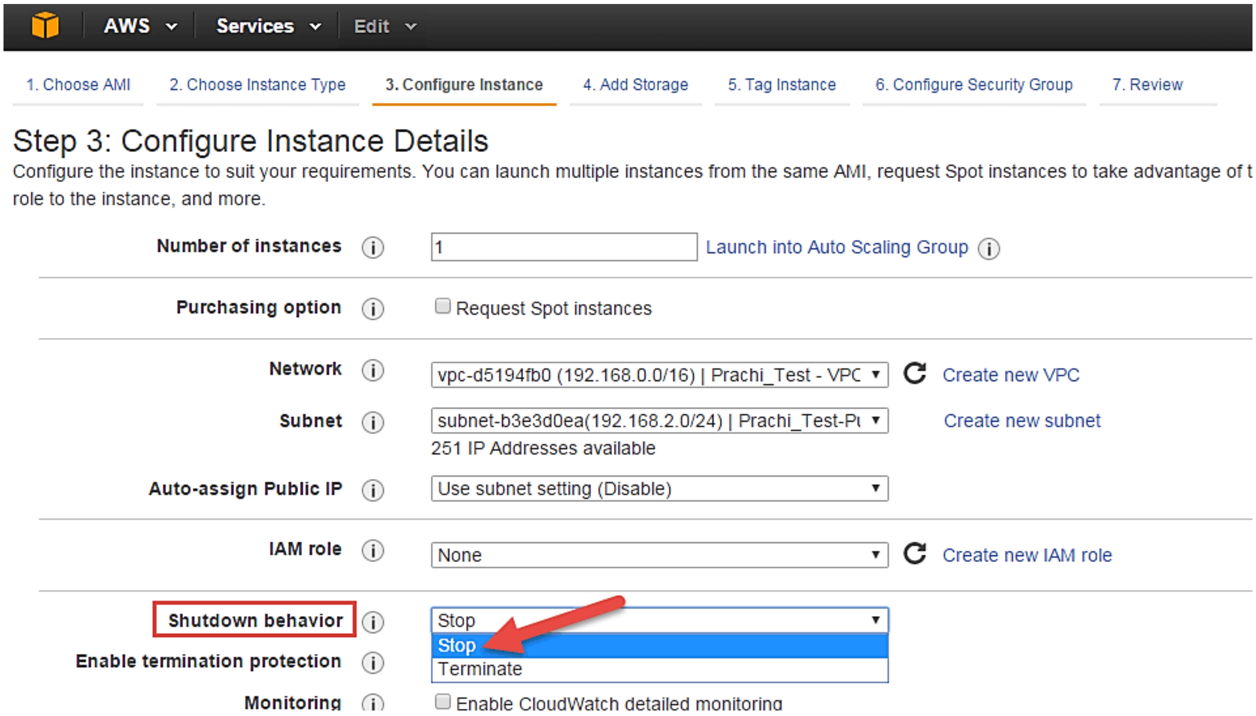
251 IP Addresses available

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

IAM role: None — Create new IAM role

Step 7) In this step, you have to do following things

- **Shutdown Behavior – when you accidently shut down your instance, you surely don't want it to be deleted but stopped.**
- **Here we are defining my shutdown behavior as Stop.**



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 t role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet

251 IP Addresses available

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

IAM role: None — Create new IAM role

Shutdown behavior: Stop

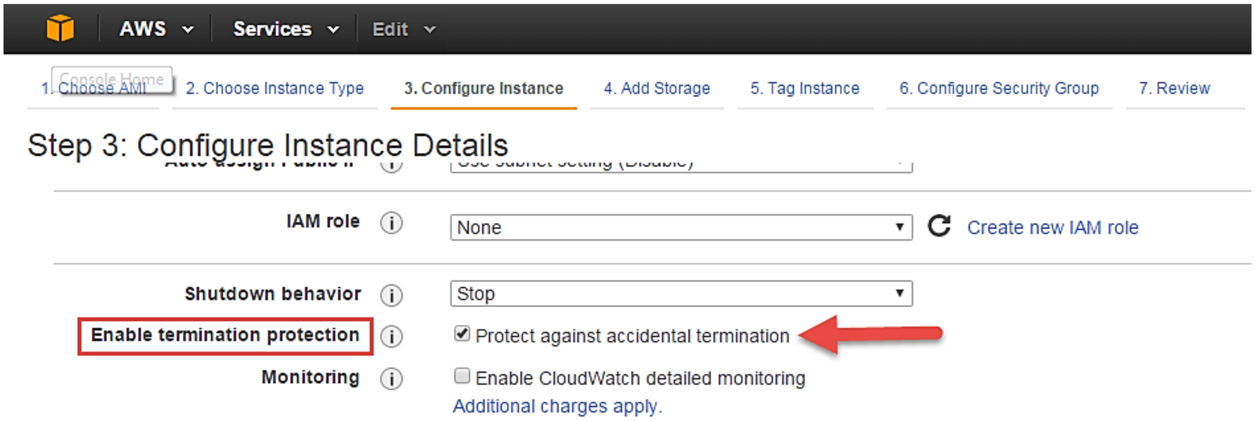
Enable termination protection

Monitoring: □ Enable CloudWatch detailed monitoring

Step 8) In this step,

- **In case, you have accidently terminated your instance, AWS has a layer of security mechanism. It will not delete your instance if you have enabled accidental termination protection.**

- **Here we are checking the option for further protecting our instance from accidental termination.**



Step 3: Configure Instance Details

IAM role: None

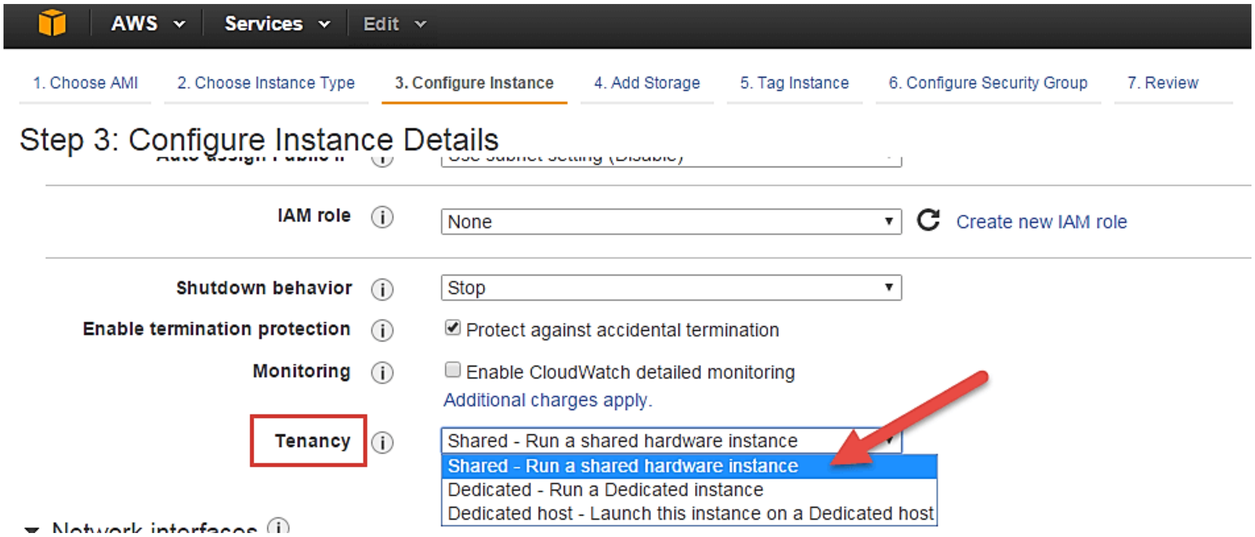
Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring

Step 9) In this step,

- **Under Monitoring- you can enable Detailed Monitoring if your instance is a business critical instance. Here we have kept the option unchecked. AWS will always provide Basic monitoring on your instance free of cost. We will visit the topic of monitoring in AWS Cloud Watch part of the tutorial.**
- **Under Tenancy- select the option if shared tenancy. If your application is a highly secure application, then you should go for dedicated capacity. AWS provides both options.**



Step 3: Configure Instance Details

IAM role: None

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

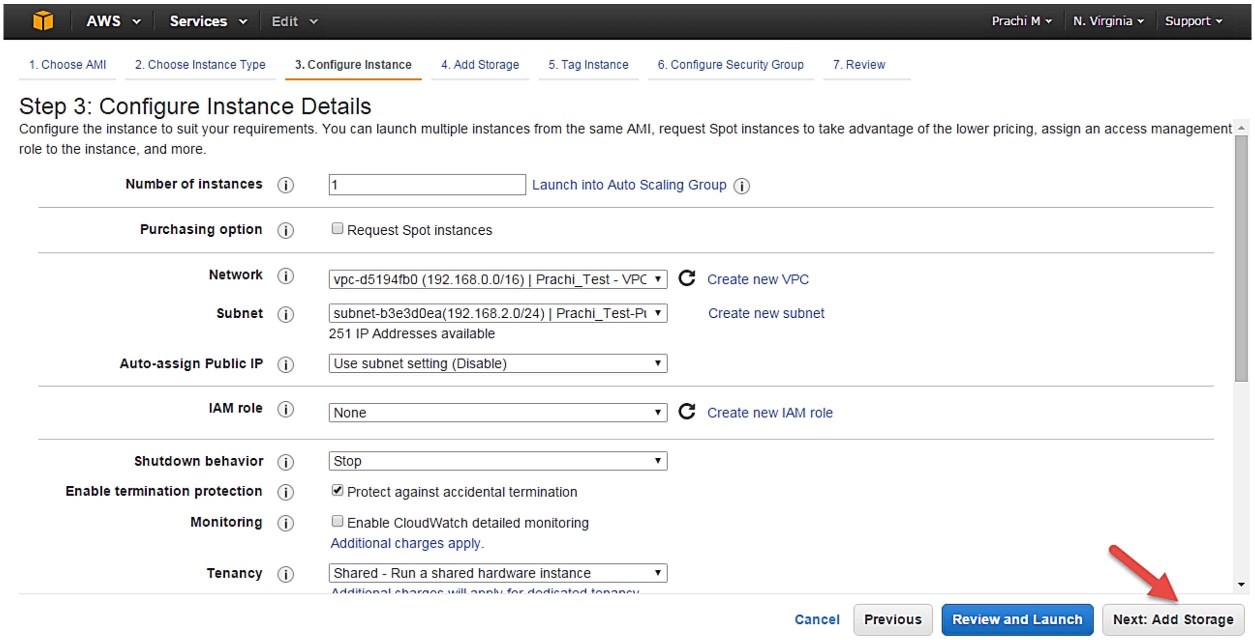
Monitoring: □ Enable CloudWatch detailed monitoring

Tenancy: Shared - Run a shared hardware instance

Shared - Run a shared hardware instance
Dedicated - Run a Dedicated instance
Dedicated host - Launch this instance on a Dedicated host

Step 10) In this step,

- **Click on ‘Add Storage’ to add data volumes to your instance in next step.**



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-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet

251 IP Addresses available

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

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring

Additional charges apply.

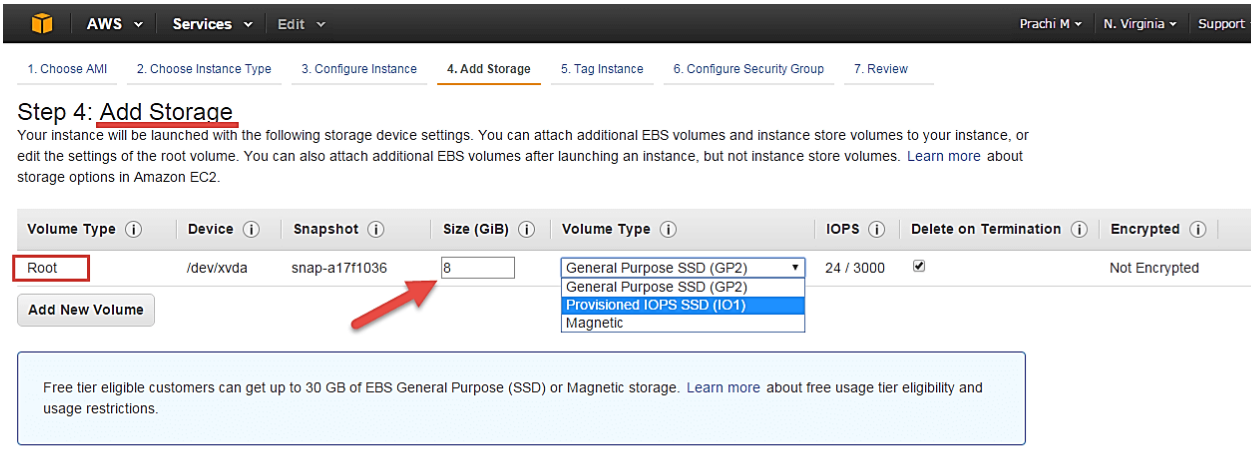
Tenancy: Shared - Run a shared hardware instance

Additional charges will apply for dedicated tenancy

Add Storage

Step 1) In this step we do following things,

- **In the Add Storage step, you'll see that the instance has been automatically provisioned a General Purpose SSD root volume of 8GB. (Maximum volume size we can give to a General Purpose volume is 16GB)**
- **You can change your volume size, add new volumes, change the volume type, etc.**
- **AWS provides 3 types of EBS volumes- Magnetic, General Purpose SSD, Provisioned IOPs. You can choose a volume type based on your application's IOPs needs.**



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-a17f1036	8	General Purpose SSD (GP2)	24 / 3000	☑	Not Encrypted

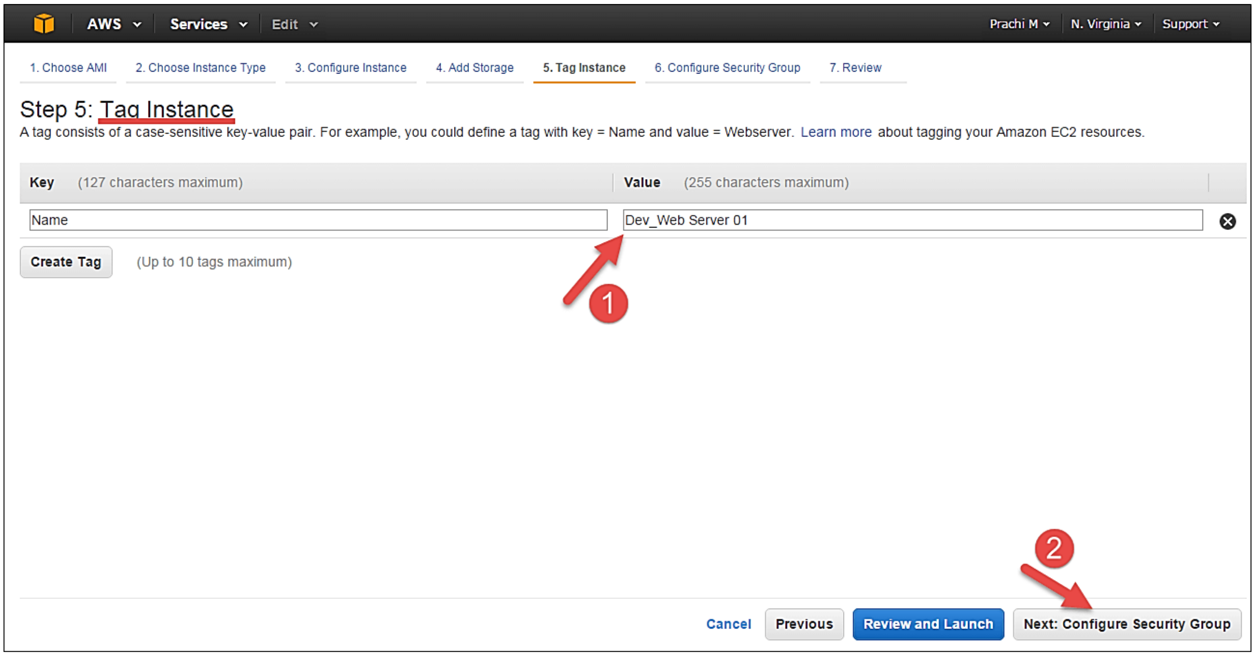
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Tag Instance

Step 1) In this step

- you can tag your instance with a key-value pair. This gives visibility to the AWS account administrator when there are lot number of instances.
- The instances should be tagged based on their department, environment like Dev/SIT/Prod. Etc. this gives a clear view of the costing on the instances under one common tag.

1. Here we have tagged the instance as a Dev_Web server 01
2. Go to configure Security Groups later



Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	Dev_Web Server 01

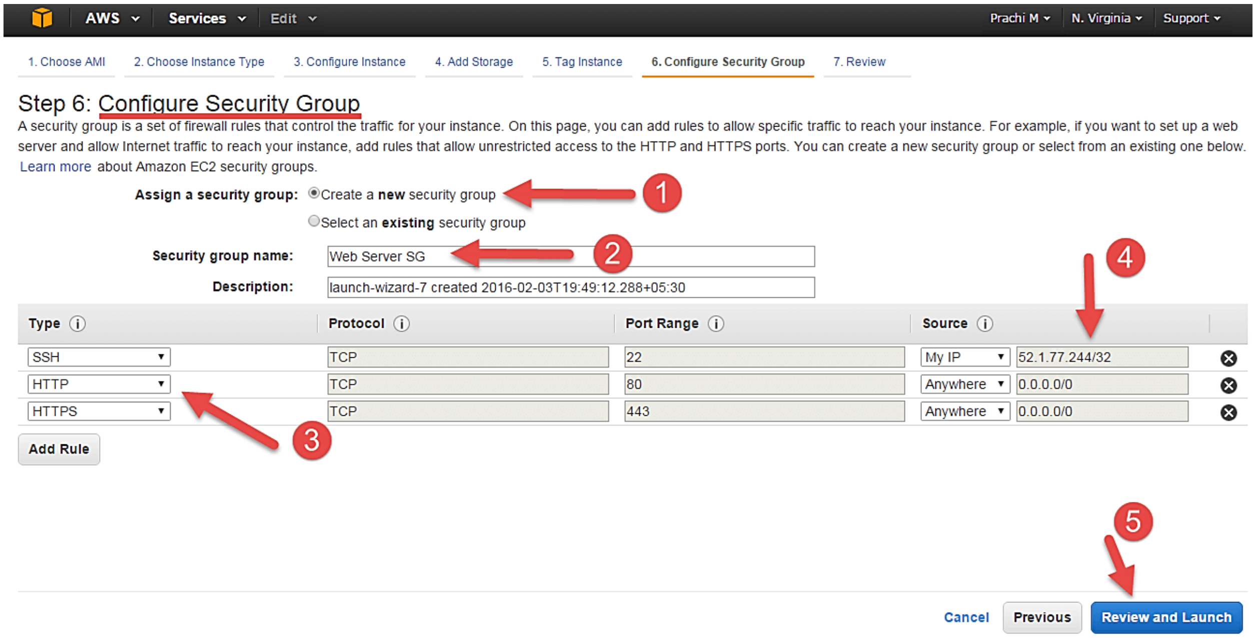
(Up to 10 tags maximum)

Configure Security Groups

Step 1) In this next step of configuring Security Groups, you can restrict traffic on your instance ports. This is an added firewall mechanism provided by AWS apart from your instance's OS firewall.
You can define open ports and IPs.

- **Since our server is a webserver=, we will do following things**

1. **Creating a new Security Group**
2. **Naming our SG for easier reference**
3. **Defining protocols which we want enabled on my instance**
4. **Assigning IPs which are allowed to access our instance on the said protocols**
5. **Once, the firewall rules are set- Review and launch**



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

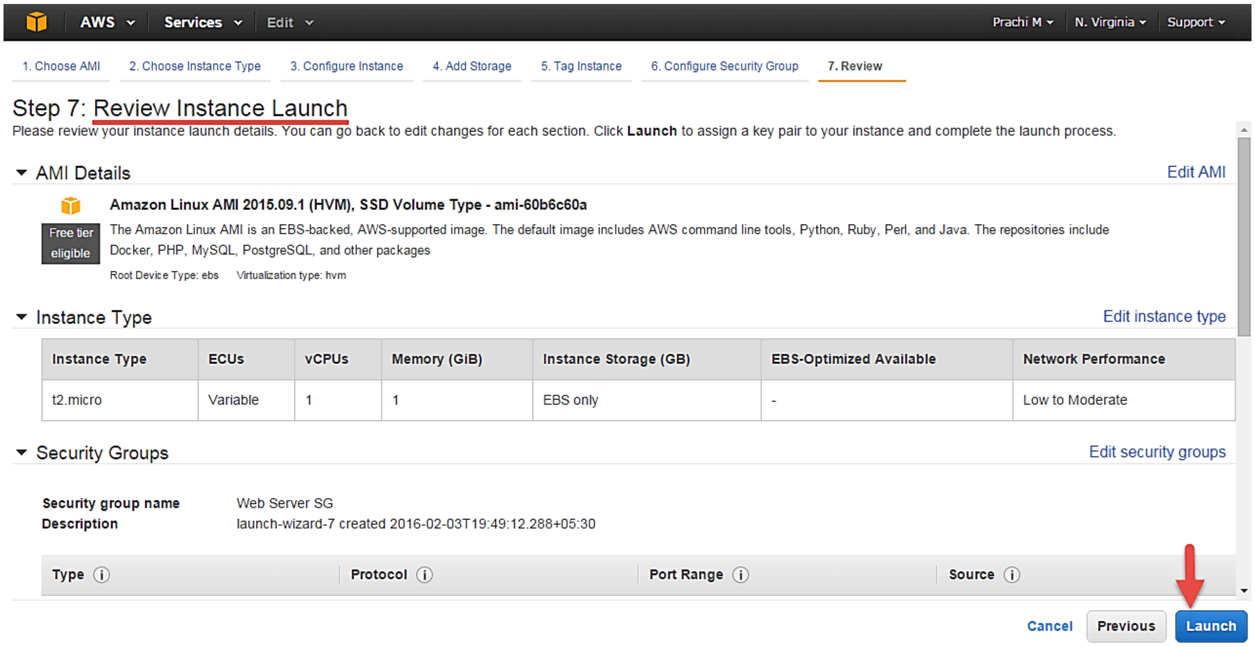
Security group name: Web Server SG

Description: launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source	
SSH	TCP	22	My IP	52.1.77.244/32
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0

Review Instances

Step 1) In this step, we will review all our choices and parameters and go ahead to launch our instance.



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

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

Free tier eligible

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

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

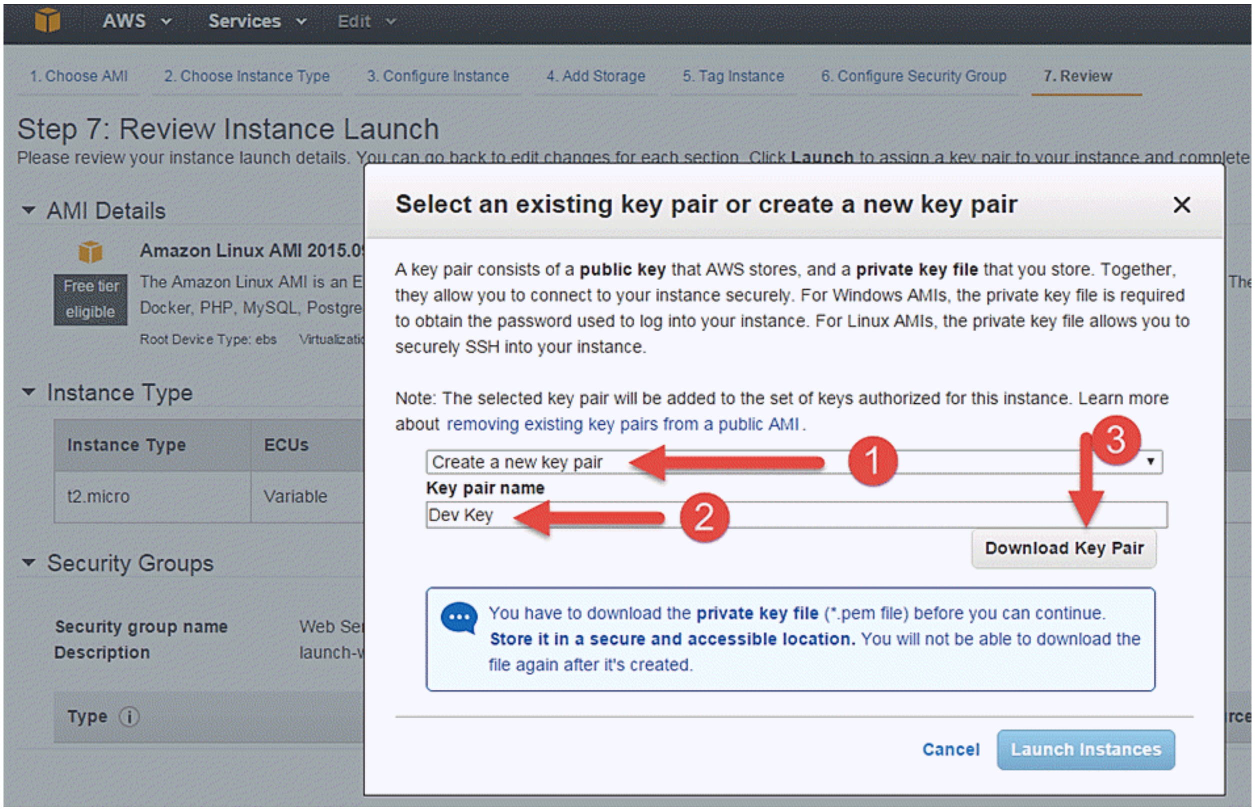
Security group name Web Server SG
Description launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source

Step 2) In the next step you will be asked to create a key pair to login to you an instance. A key pair is a set of public-private keys.

AWS stores the private key in the instance, and you are asked to download the private key. Make sure you download the key and keep it safe and secured; if it is lost you cannot download it again.

1. **Create a new key pair**
2. **Give a name to your key**
3. **Download and save it in your secured folder**



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

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 .

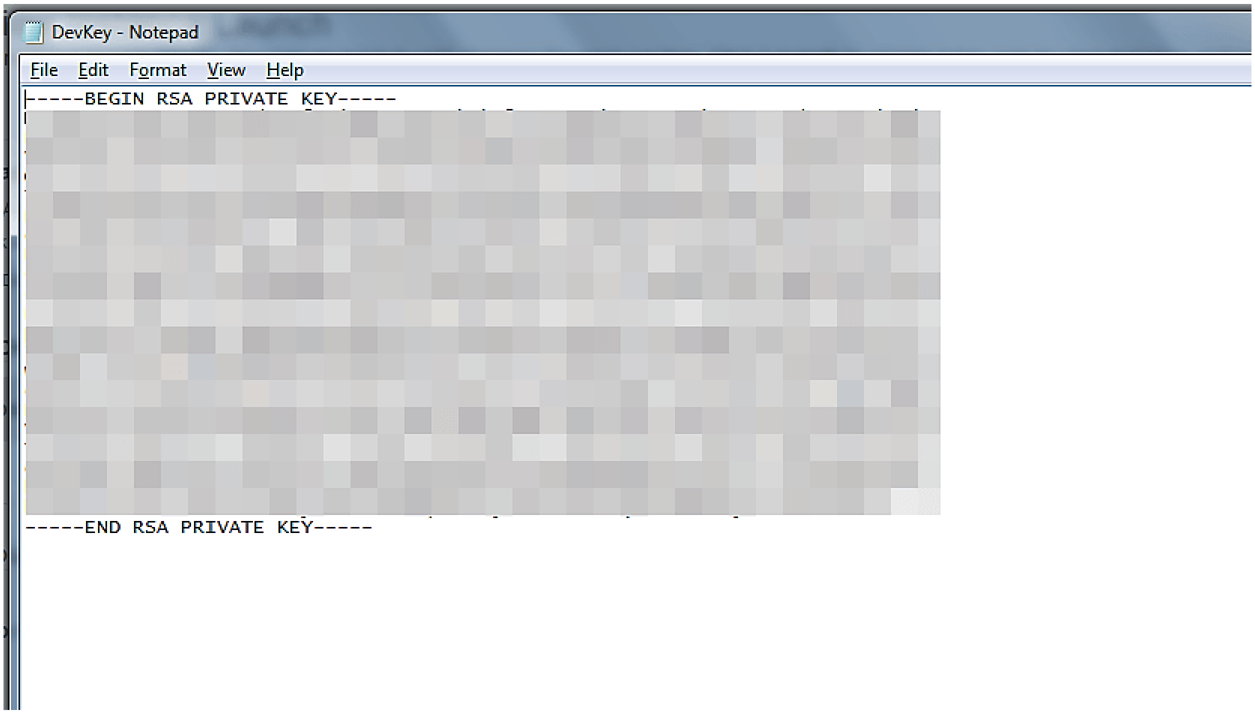
Create a new key pair

Key pair name

Dev Key

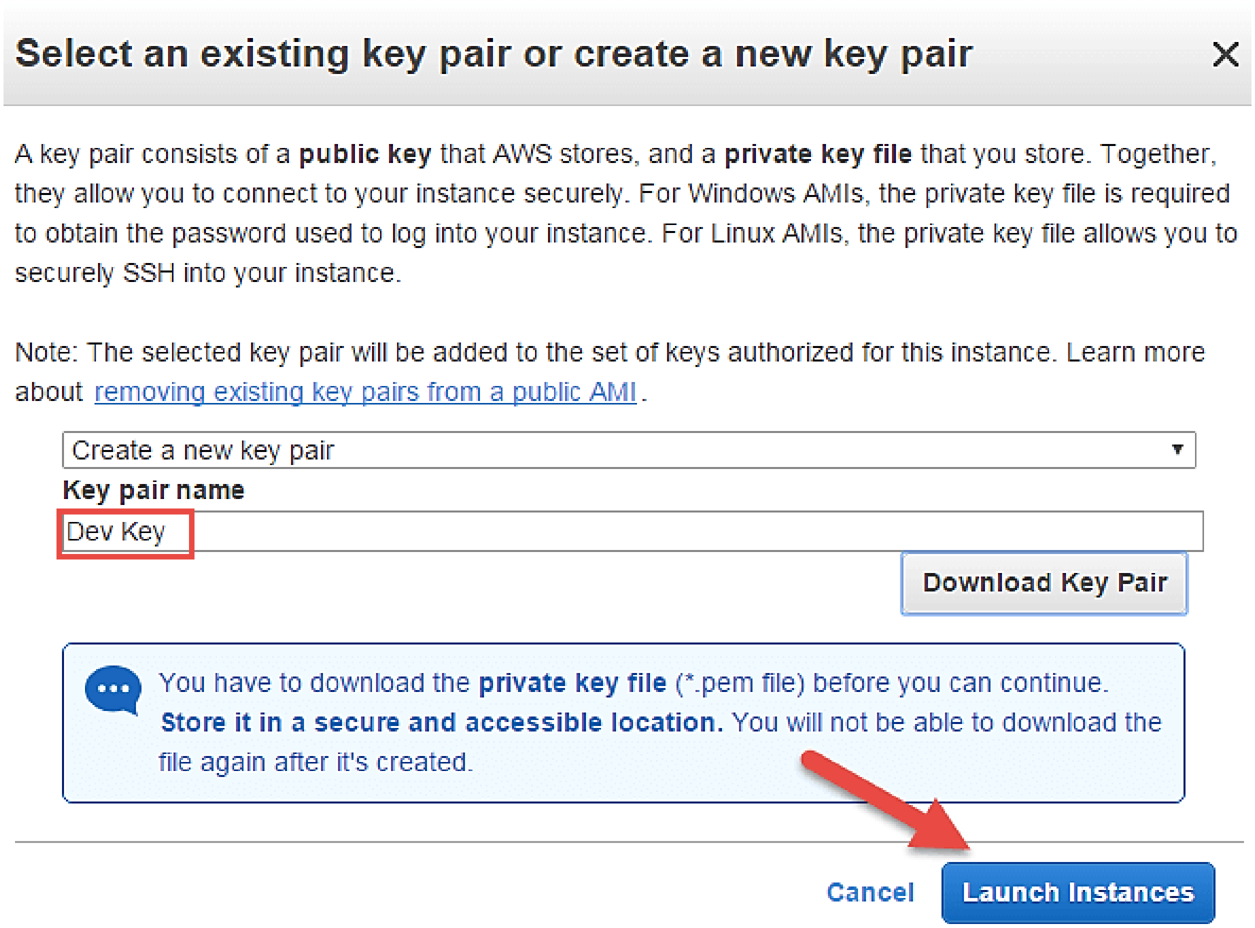
You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

- **When you download your key, you can open and have a look at your RSA private key.**



```
-----BEGIN RSA PRIVATE KEY-----
[redacted]
-----END RSA PRIVATE KEY-----
```

Step 3) Once you are done downloading and saving your key, 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 .

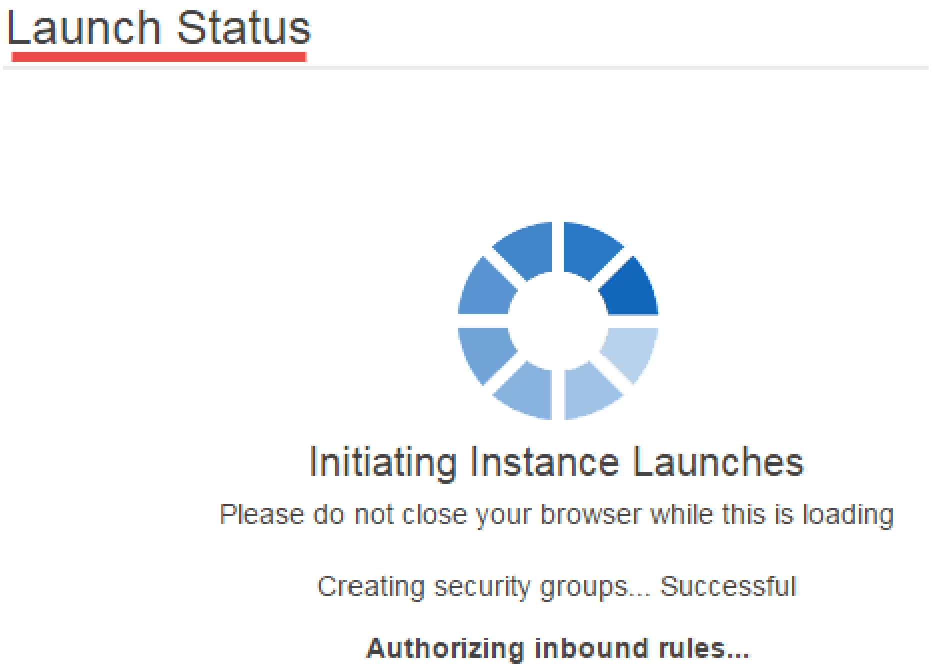
Create a new key pair

Key pair name

Dev Key

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.

- **You can see the launch status meanwhile.**



Launch Status

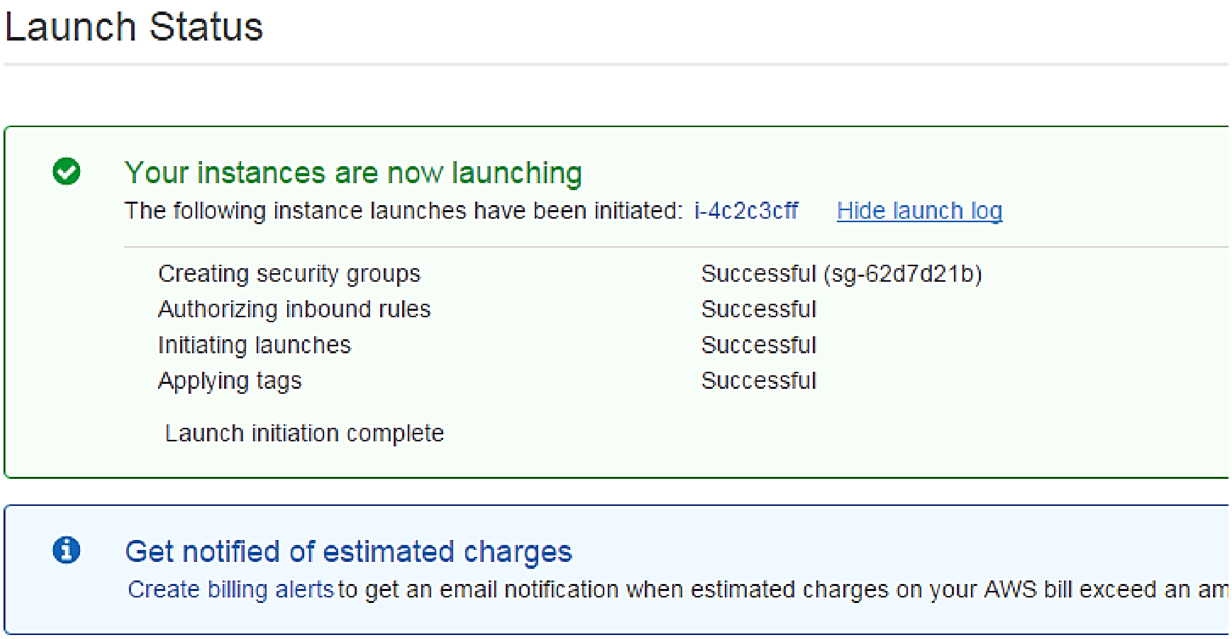
Initiating Instance Launches

Please do not close your browser while this is loading

Creating security groups... Successful

Authorizing inbound rules...

- **You can also see the launch log.**



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-4c2c3cff

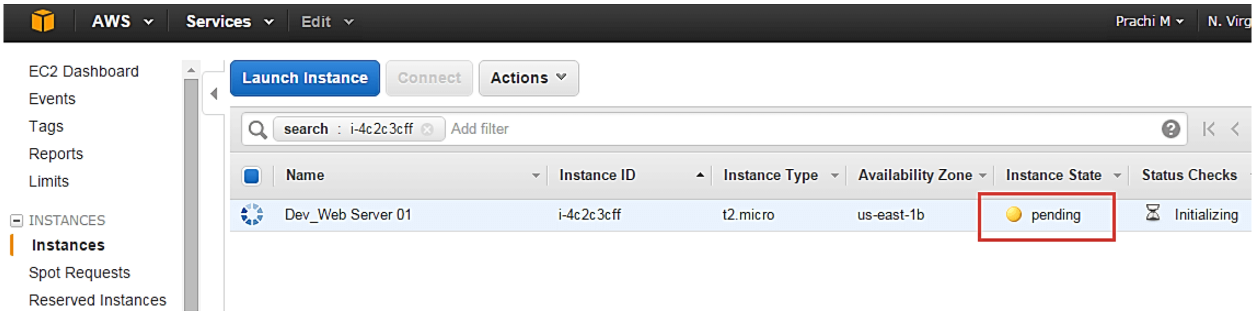
Creating security groups	Successful (sg-62d7d21b)
Authorizing inbound rules	Successful
Initiating launches	Successful
Applying tags	Successful

Launch initiation complete

Get notified of estimated charges

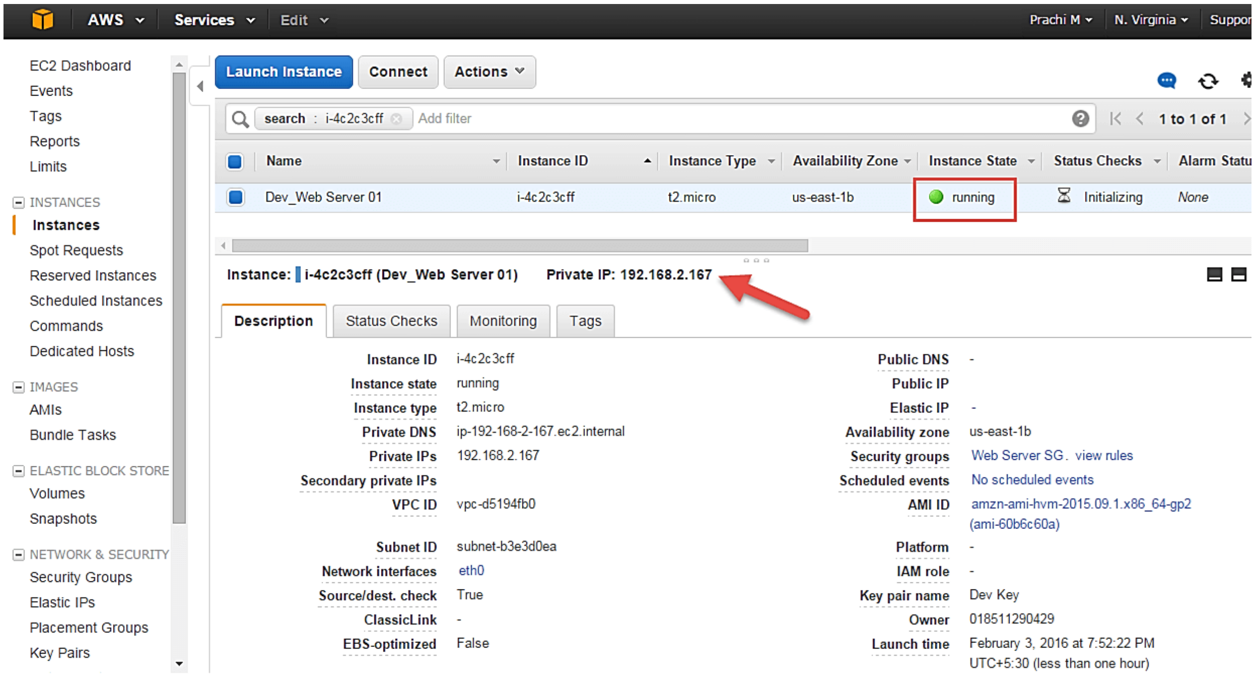
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an am

- **Click on the ‘Instances’ option on the left pane where you can see the status of the instance as ‘Pending’ for a brief while.**



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	pending	Initializing

- **Once your instance is up and running, you can see its status as 'Running' now.**
- **Note that the instance has received a Private IP from the pool of AWS.**



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	running	Initializing	None

Instance: i-4c2c3cff (Dev_Web Server 01) Private IP: 192.168.2.167

Instance ID	i-4c2c3cff	Public DNS	-
Instance state	running	Public IP	
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-192-168-2-167.ec2.internal	Availability zone	us-east-1b
Private IPs	192.168.2.167	Security groups	Web Server SG. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-d5194fb0	AMI ID	amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c60a)
Subnet ID	subnet-b3e3d0ea	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	Dev Key
ClassicLink	-	Owner	018511290429
EBS-optimized	False	Launch time	February 3, 2016 at 7:52:22 PM UTC+5:30 (less than one hour)

4) Other helpful commends with explanation

1. pwd

Use the pwd command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all

the directories that starts with a forward slash (/). An example of an absolute path is /home/username.

2. cd

To navigate through the Linux files and directories, use the cd command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.
the directory's absolute path: `cd /home/username/Movies`
here are some shortcuts to help you navigate quickly:

`cd ..` **(with two dots) to move one directory up**
`cd` **to go straight to the home folder**
`cd-` **(with a hyphen) to move to your previous directory**

3. ls

The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.
For example, enter `ls /home/username/Documents` **to view the content of Documents.**
There are variations you can use with the ls command:

`ls -R` **will list all the files in the sub-directories as well**
`ls -a` **will show the hidden files**
`ls -al` **will list the files and directories with detailed information like the permissions, size, owner, etc.**

4. cat

cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (sdout). To run this command, type cat followed by the file's name and its extension. For instance: `cat file.txt`**.**

5. cp

Use the cp command to copy files from the current directory to a different directory. For instance, the command `cp scenery.jpg path/xyz.jpg`

6. mv

The primary use of the mv command is to move files, although it can also be used to rename files or move file.

Move file

```
mv scenery.jpg path/xyz.jpg
```

rename file

```
mv scenery.jpg xyz.jpg
```

7. rm

The rm command is used to delete directories and the contents within them.

If you only want to delete the directory — as an alternative to rmdir — use `rm -r`.
Note: Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo.

Remove file

```
rm scenery.jpg
```

8. sudo

Short for "SuperUser Do", this command enables you to perform tasks that require administrative or root permissions. However, it is not advisable to use this command for daily use because it might be easy for an error to occur if you did something wrong.

9. df

Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type `df -m`.

10. du

If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers

instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the `-h` argument to the command line.

11. chmod or How we can change file permission

chmod is another Linux command, used to change the read, write, and execute permissions of files and directories. As this command is rather complicated, you can read the full tutorial in order to execute it properly.

12. chmod

As a terminal equivalent to Task Manager in Windows, the top command will display a list of running processes and how much CPU each process uses. It’s very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

13. kill a process on a port on ubuntu

```
sudo kill -9 `sudo lsof -t -i:port_number`
```

14. restart server

```
sudo reboot
```

15. shutdown server

```
sudo poweroff
```

16. restart apache

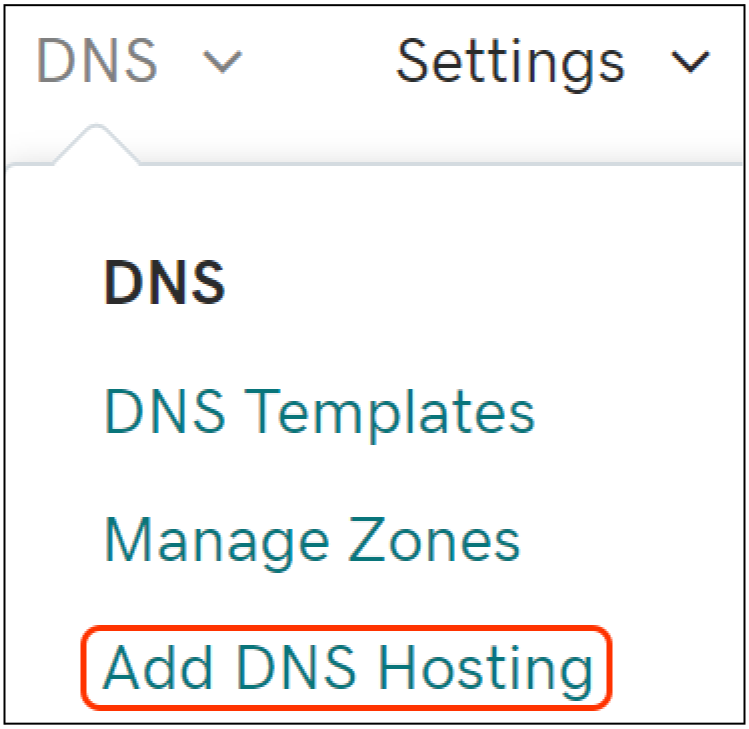
```
sudo systemctl restart apache2
```

17. restart other server

```
sudo systemctl restart service_name
```

18. Add DNS hosting

1. **Log in to your GoDaddy Domain Control Center. (Need help logging in? Find your username or password.)**
2. **Select DNS > Add DNS Hosting.**



3. **Type your domain in the Domain Name field, then select Next.**
4. **Assign the provided nameservers to your domain name through your domain registrar. The zone file for your domain will not be active on DNS hosting until after you've updated your nameservers.**

19. Manage DNS records and How to route with domain which parameter needs to change and IPv4 and IPv6

How and where you add, edit or delete your DNS records depends on where your DNS is hosted. This is determined by where your nameservers are pointing. There are three possible options for where you'll manage your DNS:

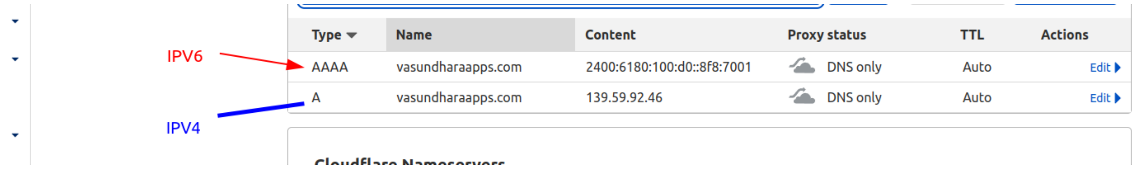
1. Your domain is registered with GoDaddy and is using our nameservers: you'll manage DNS settings in your GoDaddy account.
2. Your domain is *not* registered with GoDaddy, but *is* using our nameservers: you'll manage DNS settings in your GoDaddy account. This is usually the case if you're hosting a website with us, or using DNS Hosting.
3. Your domain is registered with any company, but is *not* using our nameservers: you won't manage DNS with us at all. You'll need to work with your DNS and/or website hosting company instead.

If your DNS is with us, you can add, edit or delete DNS records in your account.

- A record: The primary DNS record used to connect your domain to an IPV4 address that directs visitors to your website. Add / Edit / Delete
- AAAA record: The primary DNS record used to connect your domain to an IPV6 address that directs visitors to your website. Add / Edit / Delete
- Subdomain: Any DNS record that's on a prefix of your domain name such as blog.coolexample.com. A subdomain can be created using an *A record that points to the IP address* (the most common), *a CNAME that points to a URL*, or even an MX record. Add / Edit / Delete
- CNAME: A type of record that also adds a prefix to your domain name and is sometimes referred to as a type of subdomain. A CNAME can't point to an IP address. It can only point to another domain name or URL address. For example, you can create a CNAME for store.coolexample.com that points to a different URL, such as a store built with Shopify. Add / Edit / Delete
- MX record: Manages your email address and makes sure your email messages get to your inbox. Different email services use different MX records, and email with GoDaddy is automatically set up for you. Add / Edit / Delete
- TXT record: Allows you to verify domain ownership and setup email sender policies. Add / Edit / Delete
- SPF record: A type of TXT record that lets you set up email sender policies. This is an *advanced* type of DNS record. Add / Edit / Delete

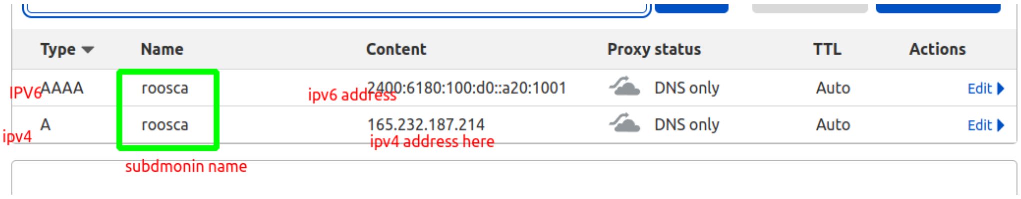
- **NS record: Contains information about your nameservers. Use these records to identify which nameservers you should use if your domain is *not* registered with GoDaddy, but you want to manage your DNS with us. This is an *advanced* custom DNS record. Add / Edit / Delete**

For Example



Type	Name	Content	Proxy status	TTL	Actions
AAAA	vasundharaapps.com	2400:6180:100:d0::8f8:7001	DNS only	Auto	Edit
A	vasundharaapps.com	139.59.92.46	DNS only	Auto	Edit

Subdomain



Type	Name	Content	Proxy status	TTL	Actions
AAAA	roosca	2400:6180:100:d0::a20:1001	DNS only	Auto	Edit
A	roosca	165.232.187.214	DNS only	Auto	Edit

20. How To Set Up Apache Virtual Hosts

1. Create the Directory Structure

We’ll first make a directory structure that will hold the site data that we will be serving to visitors in our top-level Apache directory. We’ll be using example domain names, highlighted below. You should replace these with your actual domain names.

```
sudo mkdir -p /var/www/example.com/public_html
```

2. Grant Permissions

Additionally, we’ll ensure that read access is permitted to the general web directory and all of the files and folders it contains so that pages can be served correctly.

```
sudo chmod -R 755 /var/www
```

3. Create New Virtual Host Files

Apache comes with a default virtual host file called `000-default.conf` that we'll use as a template. We'll copy it over to create a virtual host file for each of our domains.

Create the First Virtual Host File

Start by copying the file for the first domain:

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/example.com.conf
```

Open the new file in your editor (we're using nano below) with root privileges:

```
sudo nano /etc/apache2/sites-available/example.com.conf
```

We will customize this file for our own domain. Modify the highlighted text below for your own circumstances.

/etc/apache2/sites-available/example.com.conf

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/rootpath
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

At this point, save and close the file.

4. Enable the New Virtual Host Files

With our virtual host files created, we must enable them. We'll be using the `a2ensite` tool to achieve this goal.

```
sudo a2ensite example.com.conf
```

When you are finished, you need to restart Apache to make these changes take effect and use `systemctl status` to verify the success of the restart.

```
sudo systemctl restart apache2
```

5. Test your Results

Now that you have your virtual hosts configured, you can test your setup by going to the domains that you configured in your web browser:

```
http://example.com
```

21. How to set up SSL? And How we can apply self SSL.

Certbot provides a variety of ways to obtain SSL certificates through plugins. The Apache plugin will take care of reconfiguring Apache and reloading the configuration whenever necessary. To use this plugin, type the following:

```
sudo certbot --apache
```

This script will prompt you to answer a series of questions in order to configure your SSL certificate. First, it will ask you for a valid e-mail address. This email will be used for renewal notifications and security notices:

```
Output

Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter
'c' to
cancel): you@your_domain
```

After providing a valid e-mail address, hit `ENTER` to proceed to the next step. You will then be prompted to confirm if you agree to Let's Encrypt terms of service. You can confirm by pressing `A` and then `ENTER`:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You
must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
(A)gree/(C)ancel: A
```

Next, you'll be asked if you would like to share your email with the Electronic Frontier Foundation to receive news and other information. If you do not want to subscribe to their content, type `N`. Otherwise, type `Y`. Then, hit `ENTER` to proceed to the next step.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
Would you be willing to share your email address with the Electronic
Frontier
Foundation, a founding partner of the Let's Encrypt project and the
non-profit
organization that develops Certbot? We'd like to send you email about our
work
encrypting the web, EFF news, campaigns, and ways to support digital
freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
(Y)es/(N)o: N
```

The next step will prompt you to inform Certbot of which domains you'd like to activate HTTPS for. The listed domain names are automatically obtained from your Apache virtual host configuration, that's why it's important to make sure you have the correct `ServerName` and `ServerAlias` settings configured in your virtual host. If you'd like to enable HTTPS for all listed domain names (recommended), you can leave the prompt blank and hit `ENTER` to proceed. Otherwise, select the domains you want to enable HTTPS for by listing each appropriate number, separated by commas and/ or spaces, then hit `ENTER`.

```
Which names would you like to activate HTTPS for?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
1: your_domain
2: www.your_domain
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
Select the appropriate numbers separated by commas and/or spaces, or leave
input
blank to select all options shown (Enter 'c' to cancel):1
```

You'll see output like this:

```
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for your_domain
http-01 challenge for www.your_domain
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at
/etc/apache2/sites-available/your_domain-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
Deploying Certificate to VirtualHost
/etc/apache2/sites-available/your_domain-le-ssl.conf
Enabling available site:
/etc/apache2/sites-available/your_domain-le-ssl.conf
Deploying Certificate to VirtualHost
/etc/apache2/sites-available/your_domain-le-ssl.conf
```

Next, you’ll be prompted to select whether or not you want HTTP traffic redirected to HTTPS. In practice, that means when someone visits your website through unencrypted channels (HTTP), they will be automatically redirected to the HTTPS address of your website. Choose `2` to enable the redirection, or `1` if you want to keep both HTTP and HTTPS as separate methods of accessing your website.

```
Please choose whether or not to redirect HTTP traffic to HTTPS, removing
HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose
this for
new sites, or if you're confident your site works on HTTPS. You can undo
this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

After this step, Certbot’s configuration is finished, and you will be presented with the final remarks about your new certificate, where to locate the generated files, and how to test your configuration using an external tool that analyzes your certificate’s authenticity:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -
Congratulations! You have successfully enabled https://your_domain and
https://www.your_domain

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=your_domain
https://www.ssllabs.com/ssltest/analyze.html?d=www.your_domain
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - -

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/your_domain/fullchain.pem
   Your key file has been saved at:
```

```
   /etc/letsencrypt/live/your_domain/privkey.pem
   Your cert will expire on 2020-07-27. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

Your certificate is now installed and loaded into Apache's configuration. Try reloading your website using `https://` and notice your browser's security indicator. It should point out that your site is properly secured, typically by including a lock icon in the address bar.

22. How to create a subdomain. And point with server

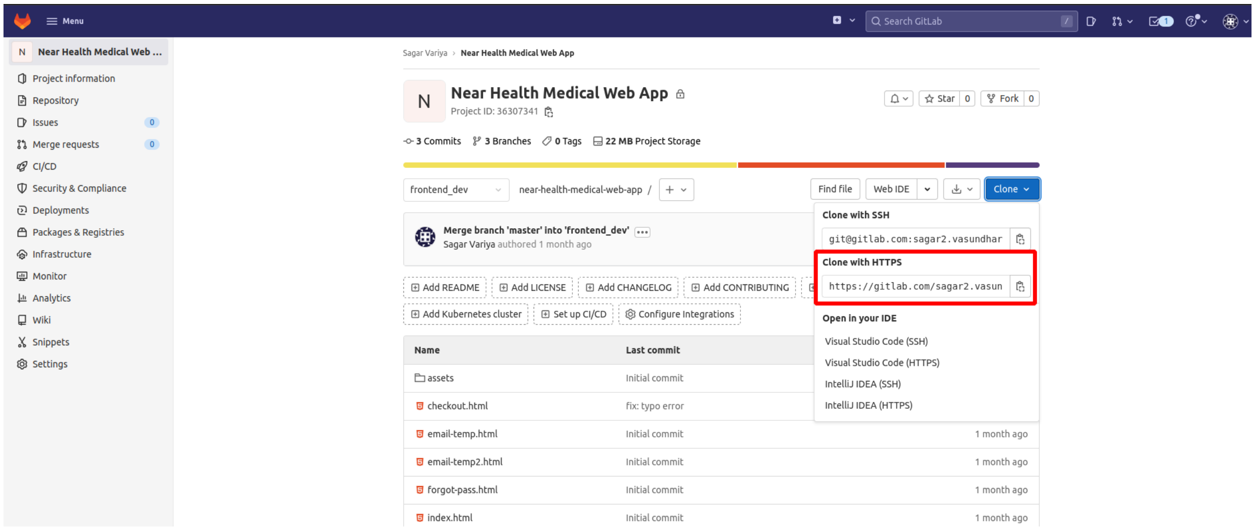
Follow step [19 Manage DNS records]and [20 How To Set Up Apache Virtual Hosts]

23. How to deploy project on server using GitLab.

1. Clone with HTTPS

Clone with HTTPS when you want to authenticate each time you perform an operation between your computer and GitLab.

1. **Go to your project's landing page and select Clone. Copy the URL for Clone with HTTPS.**



Merge branch 'master' into 'frontend_dev'
Sagar Variya authored 1 month ago

```
git@gitlab.com:sagar2.vasundhar
https://gitlab.com/sagar2.vasun
```

Name	Last commit	
assets	Initial commit	
checkout.html	fix: typo error	
email-temp.html	Initial commit	1 month ago
email-temp2.html	Initial commit	1 month ago
forgot-pass.html	Initial commit	1 month ago
index.html	Initial commit	1 month ago

2. Open a terminal and go to the directory where you want to clone the files.
3. Run the following command. Git automatically creates a folder with the repository name and downloads the files there.

```
git clone https://gitlab.com/gitlab-tests/sample-project.git
```

2. Change dir name `mv sample-project foldername`
3. Go to the folder application using cd command on your cmd or terminal

```
cd foldername
```

4. Run `composer install` on your cmd or terminal
5. Copy .env.example file to .env on the root folder. You can type `cp .env.example .env` if using terminal, Ubuntu
6. Open your .env file and change the database name (DB_DATABASE) to whatever you have, username (DB_USERNAME) and password (DB_PASSWORD) fields correspond to your configuration.
7. Run

```
php artisan key:generate
php artisan migrate
```

24. How to deploy project on server using GitLab.

Follow step 19 Manage DNS records and 20 How To Set Up Apache Virtual Hosts

25. Steps to create an RDS instance

Following are the steps to create an RDS Instance:

1. Sign into **AWS Management Console**.

2. Open the **RDS console**.

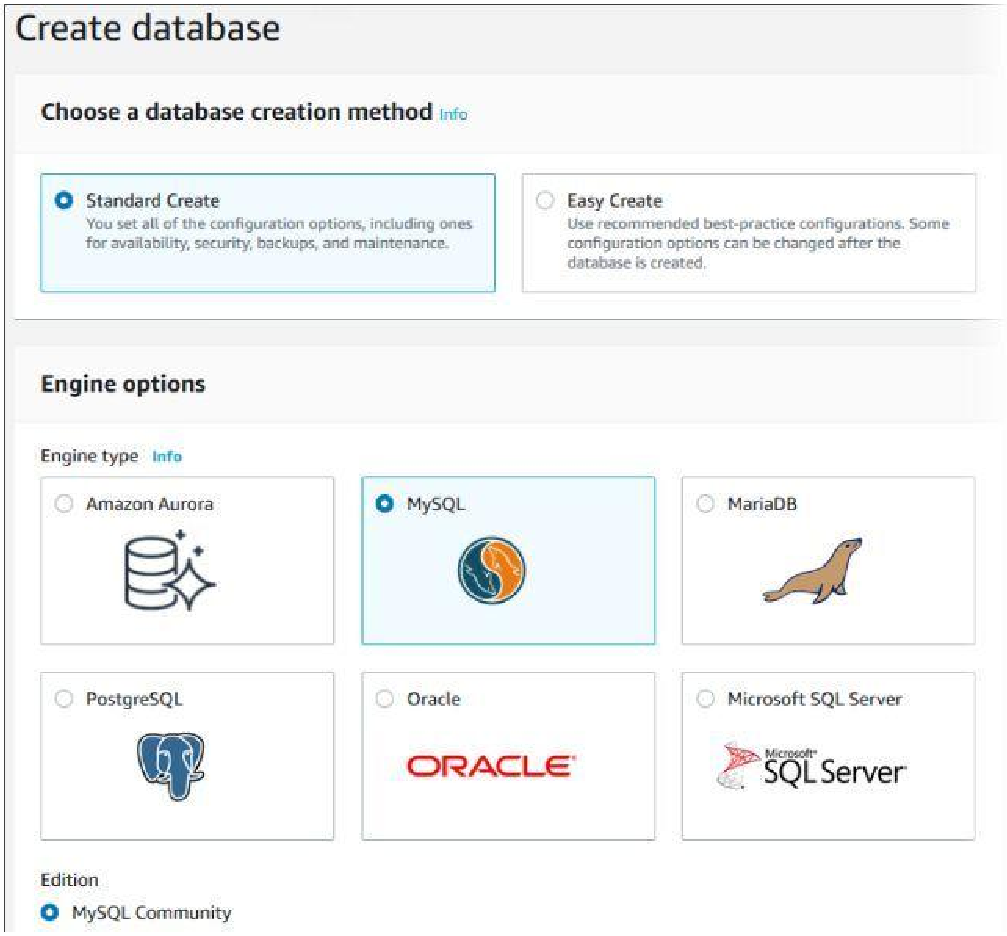
3. In the upper-right corner, choose the region where you wish to create your instance.

4. In the navigation pane, click on ‘**Databases’**.

5. Click on ‘**Create database’**.

6. Make sure **‘Standard create**’ is chosen, then click on MySQL (or the database which you wish to create an RDS database instance).

Below is the snip that shows this operation.



Create database

Choose a database creation method Info

● Standard Create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy Create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Amazon Aurora

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

Edition

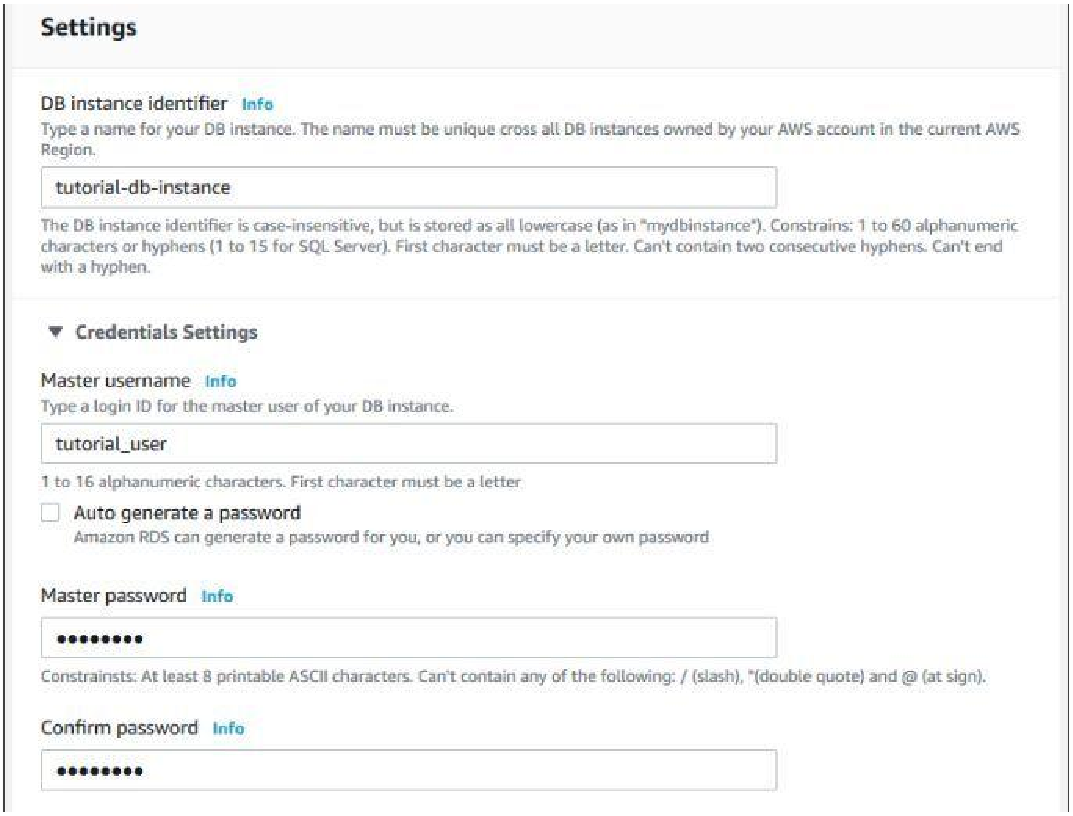
● MySQL Community

Image credit: aws.amazon.com

7. In the ‘**Templates’** tab, click on the ‘**Dev/Test’** option.

8. In the ‘**Setting’** tab, set the following values:

- DB instance identifier
- Master username
- Auto Generate a password
- Master password
- Confirm password



Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

tutorial-db-instance

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constrains: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

tutorial_user

1 to 16 alphanumeric characters. First character must be a letter

□ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

Master password Info

••••••••

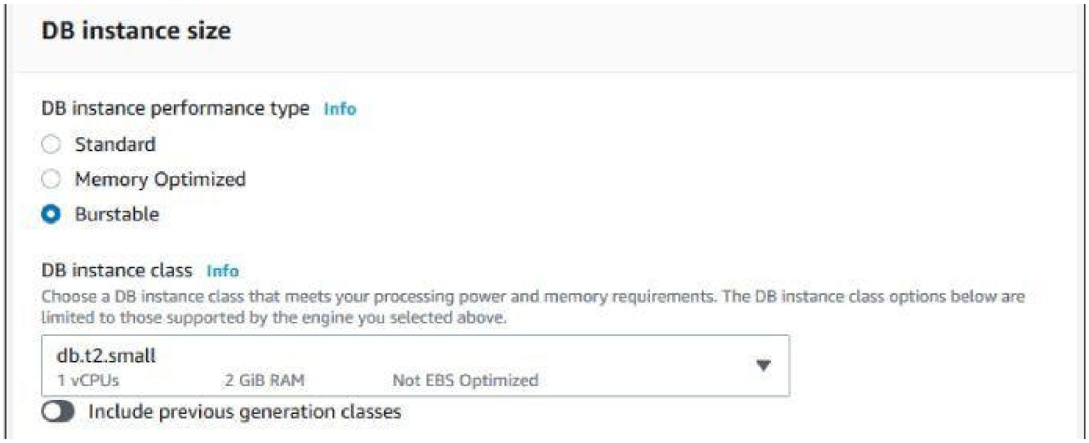
Constrainsts: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password Info

••••••••

9. In the ‘DB instance size’ option, give a value for the following variables:

- DB instance performance types
- DB instance class



DB instance size

DB instance performance type Info

○ Standard

○ Memory Optimized

● Burstable

DB instance class Info

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

db.t2.small
1 vCPUs 2 GiB RAM Not EBS Optimized

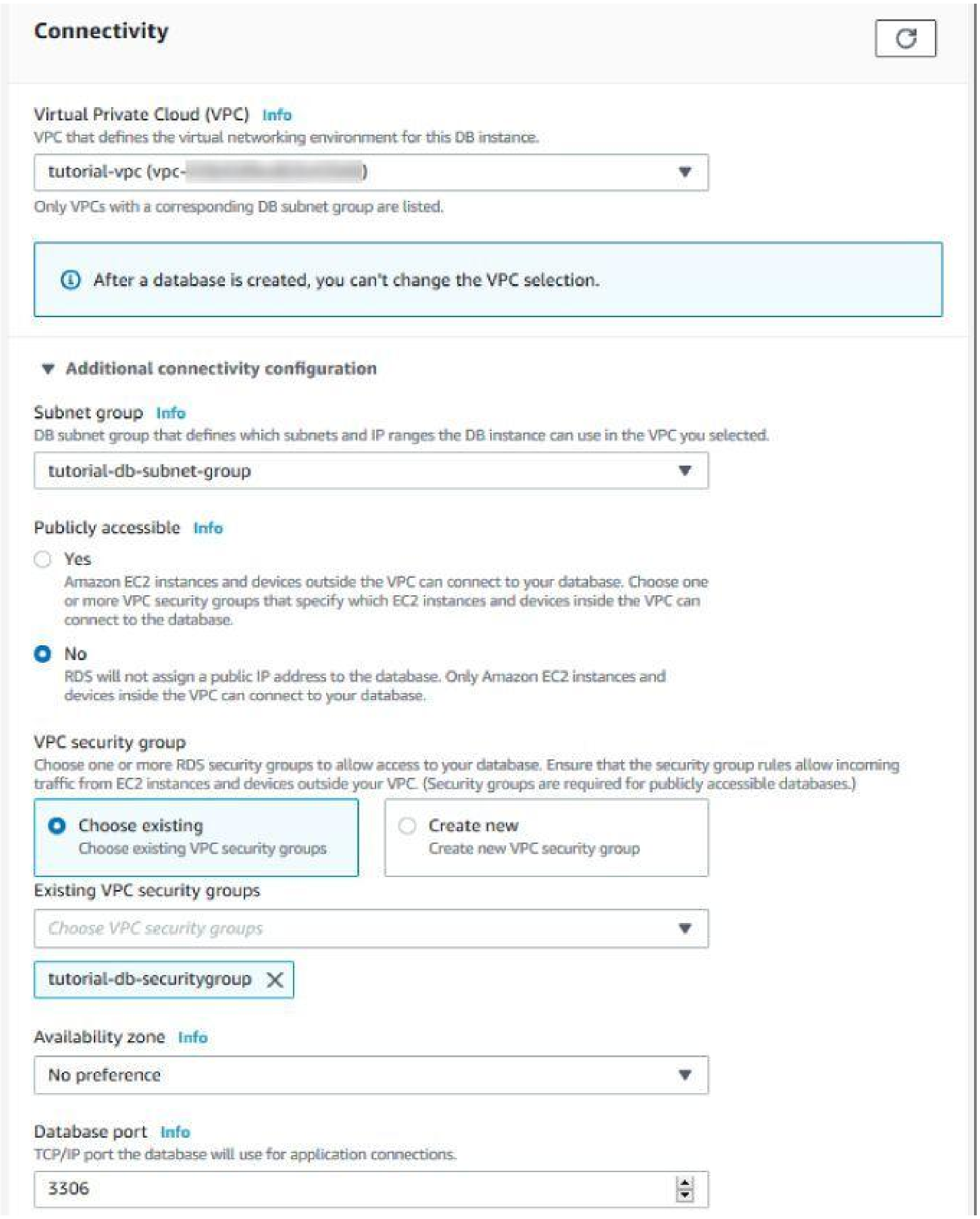
Include previous generation classes

10. In the ‘**Storage’** and ‘**Availability & durability’** section, leave the default values as is.

11. In the ‘**Connectivity’** section, click on the ‘Additional connectivity configuration’ and set the below values in it:

- Virtual Private Cloud (VPC)
- Subnet group
- Publicly accessible- No
- VPC security groups
- Availability zone- No preference
- Database port- 3306

The same is displayed in the below screenshot:



Connectivity

Virtual Private Cloud (VPC) Info
VPC that defines the virtual networking environment for this DB instance.
tutorial-vpc (vpc-[redacted]) ▼
Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

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

Publicly accessible Info

○ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

● No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

Existing VPC security groups
Choose VPC security groups ▼
tutorial-db-securitygroup ✕

Availability zone Info
No preference ▼

Database port Info
TCP/IP port the database will use for application connections.
3306

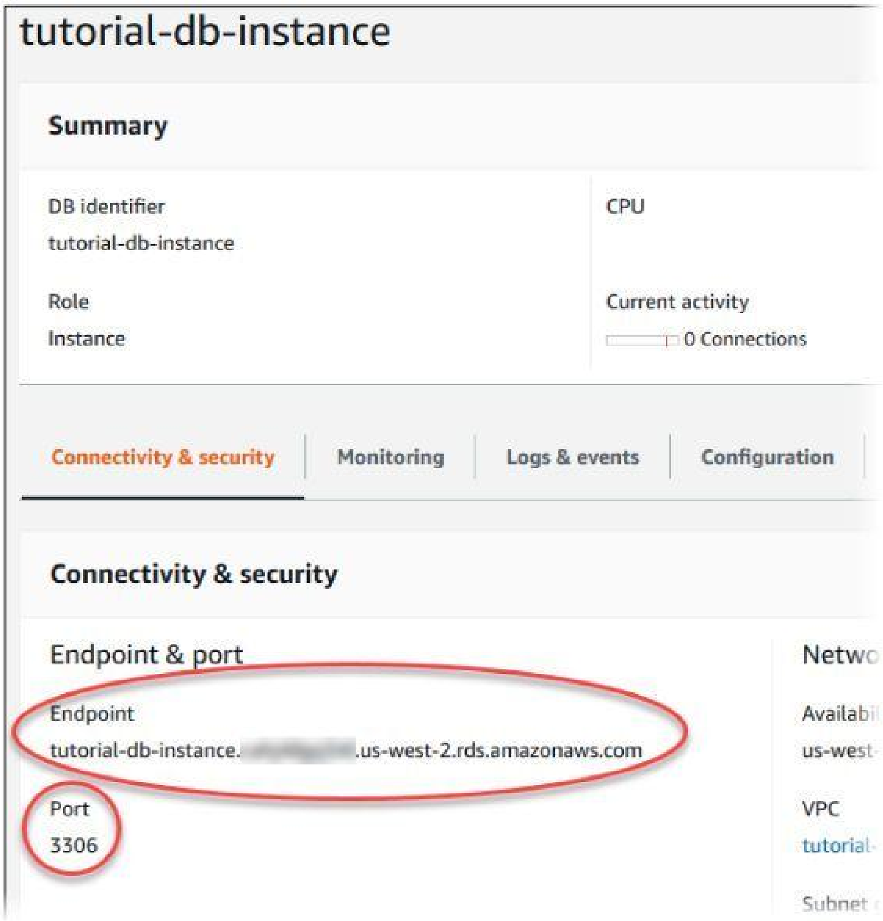
12. Click on the ‘**Additional configuration’**tab, and provide a name for the **‘Initial database name’** variable. The default settings for other options need to be kept the same.

13. Now click on ‘**Create database**’.

14. It takes a few minutes for the instance to get created. It can be seen in the ‘Databases’ list as ‘Creating’.

15. Once it is created, it shows as ‘Available’.

16. The ‘**Endpoint’** and ‘**Port’** of the database instance can be viewed in the ‘Connectivity & security’ section.



tutorial-db-instance

Summary

DB identifier
tutorial-db-instance

Role
Instance

CPU

Current activity
0 Connections

Connectivity & security

Endpoint & port

Endpoint
tutorial-db-instance.[redacted].us-west-2.rds.amazonaws.com

Port
3306

Netwo

Availabi
us-west

VPC
tutorial

Subnet

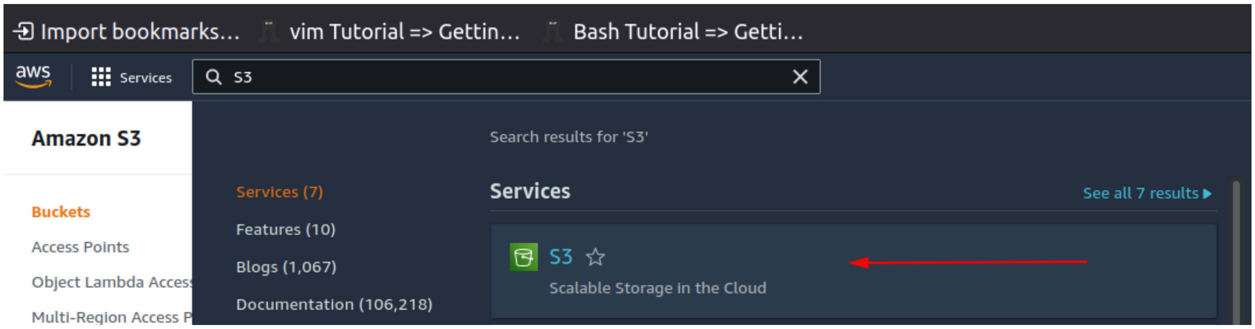
Note: Make sure that your database instance is secure, by verifying that sources outside of the VPC can’t connect to the RDS database instance.

https://stackoverflow.com/questions/4402482/using-phpmyadmin-to-administer-amazon-rds

26. Steps S3

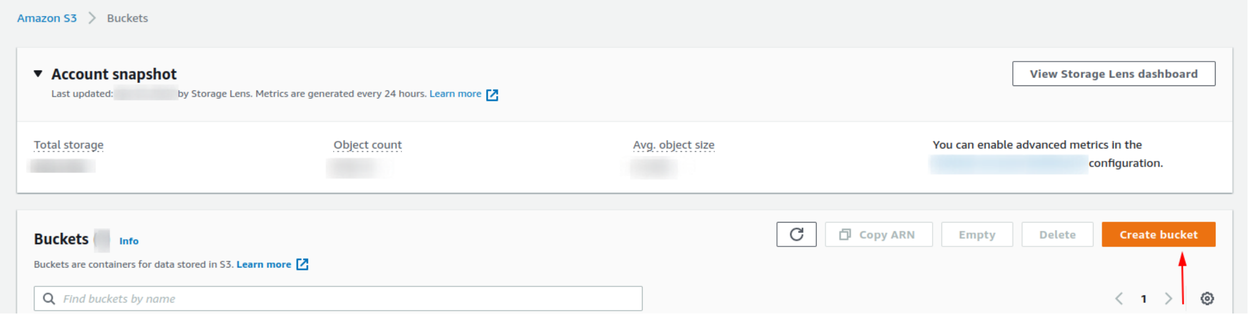
1. Log in to your AWS account via console (https://aws.amazon.com/console/)

2. Once you have logged in, you can search for S3 in the Search bar



3. Click on S3 from search results under Services. This will take you to your S3 page where you can create and manage your S3 buckets

4. Click on “Create bucket” button to set up a new S3 bucket



Account snapshot

Last updated: [redacted] by Storage Lens. Metrics are generated every 24 hours. Learn more

Total storage: [redacted]

Object count: [redacted]

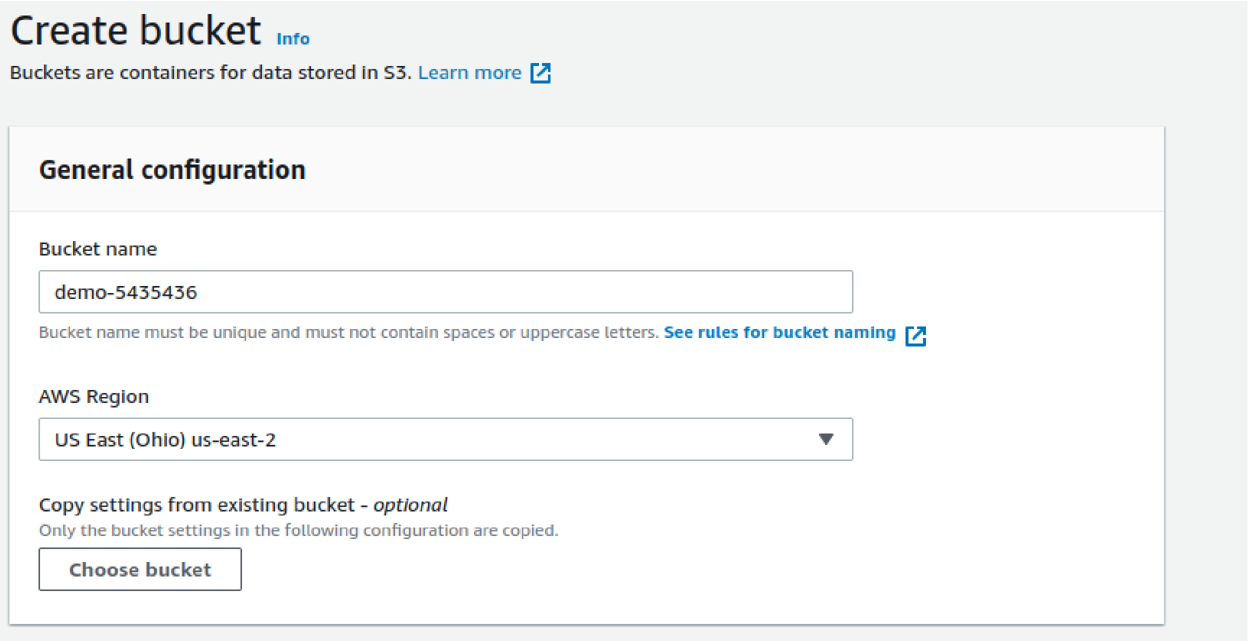
Avg. object size: [redacted]

You can enable advanced metrics in the [redacted] configuration.

Buckets [redacted] Info

Buckets are containers for data stored in S3. Learn more

5. You will then be presented with the bucket configuration page
6. Enter a unique name for your S3 bucket without using any spaces or upper case letters. The bucket names must be unique since the S3 bucket namespace is global
7. Select the region in which you would like to create your S3 bucket. It is generally suitable to create the bucket in a region that is geographically closer to the users of the bucket



Create bucket

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: demo-5435436

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: US East (Ohio) us-east-2

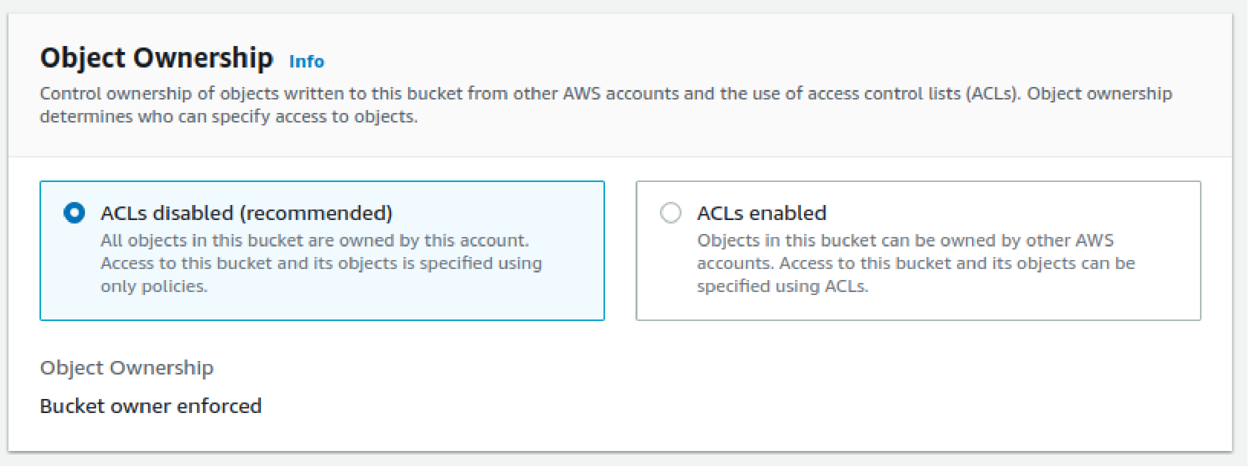
Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

8. The next configuration section allows you to manage the ownership of objects. S3 provides two options for object ownership: ACLs disabled and ACLs enabled

ACLs disabled means all the objects in the bucket are owned by the account that created the bucket

ACLs enabled means that the objects can be owned by the other AWS accounts as well



Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

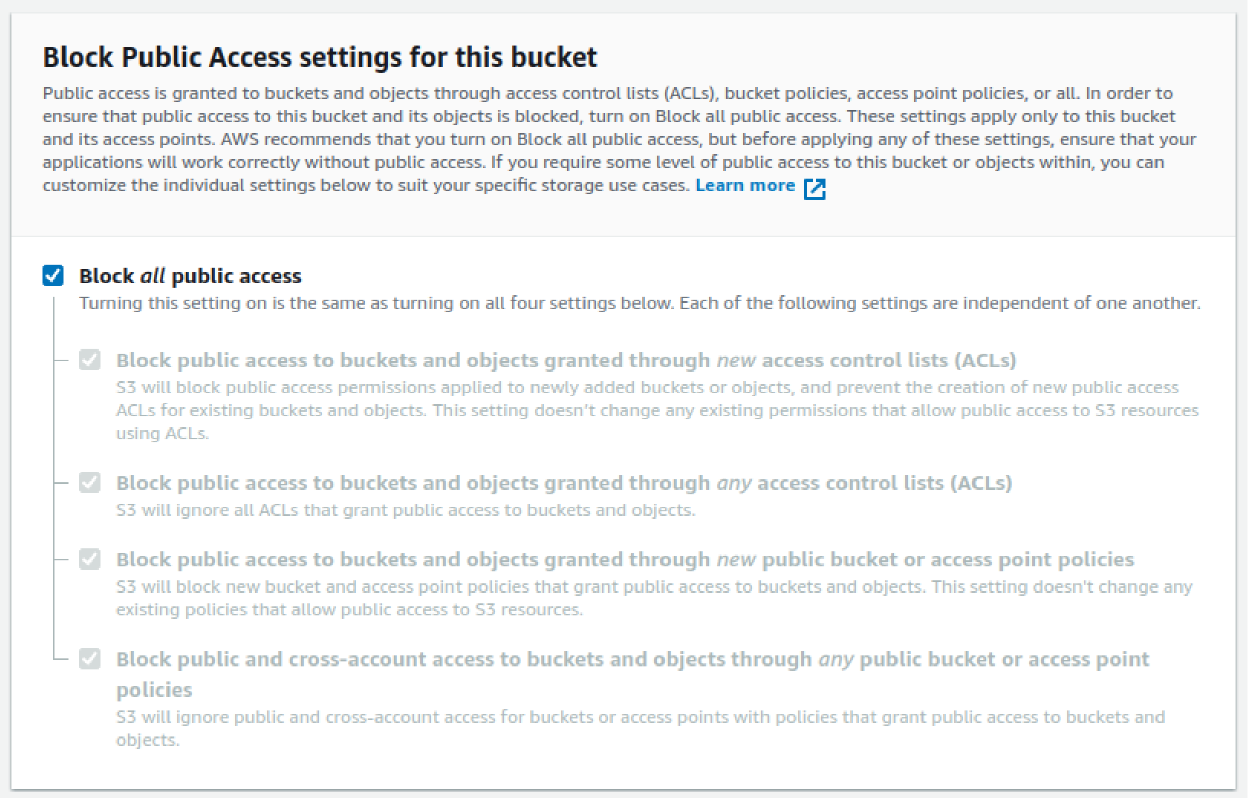
● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

9. Select the accessibility option for your bucket. If your bucket contains any sensitive data or any data which you do not want to make public then select “Block all Public access” setting for this bucket



Block Public Access settings for this bucket

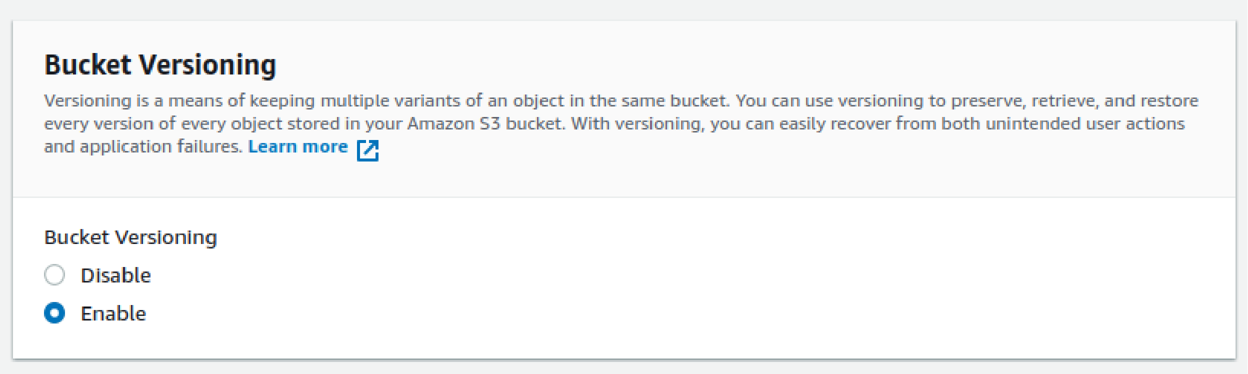
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☑ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- ☑ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

10. Then you can enable or disable Bucket Versioning for your S3 bucket.

Bucket Versioning is a feature provided by S3 which allows you to have multiple versions of an object. When it is important for you to ensure that your objects do not get overwritten or deleted, you can enable this feature.

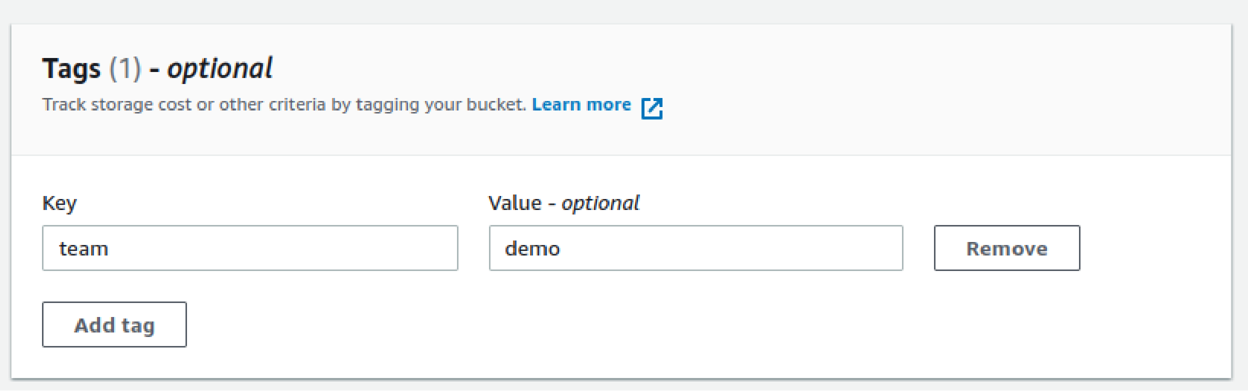


Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
○ Disable
● Enable

11. Assign Tags to your bucket. Though this is an optional feature, Tags can be attached to categorise your bucket in various ways like environment, teams, createdby, etc.



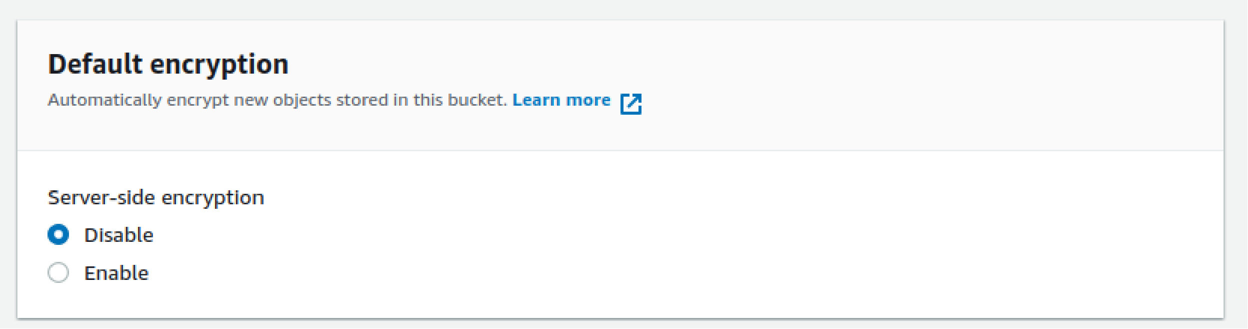
Tags (1) - *optional*

Track storage cost or other criteria by tagging your bucket. Learn more

Key: team

Value - *optional*: demo

12. Select whether you want to enable or disable default encryption for your objects. Default encryption is a great security feature when you want to protect your objects stored in S3 buckets



Default encryption

Automatically encrypt new objects stored in this bucket. Learn more

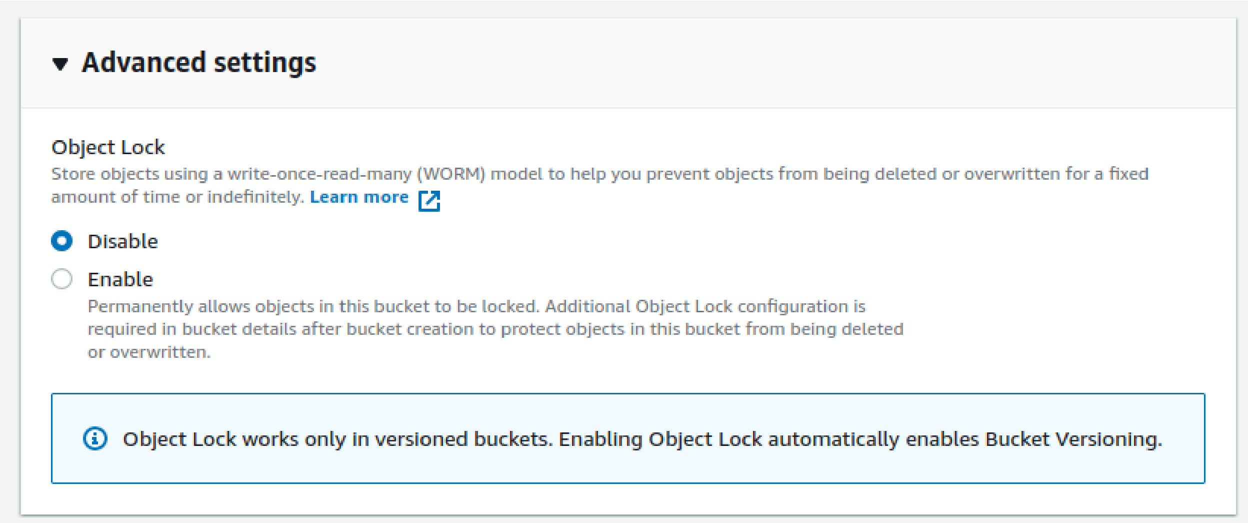
Server-side encryption

● Disable

○ Enable

13. There’s also a section of Advanced settings in S3 configurations which allows you to enable or disable object locking

Object Lock is helpful when you want to prevent the stored object in S3 bucket from getting deleted or overwritten. Enabling this setting automatically enables the bucket versioning if not enabled



▼ Advanced settings

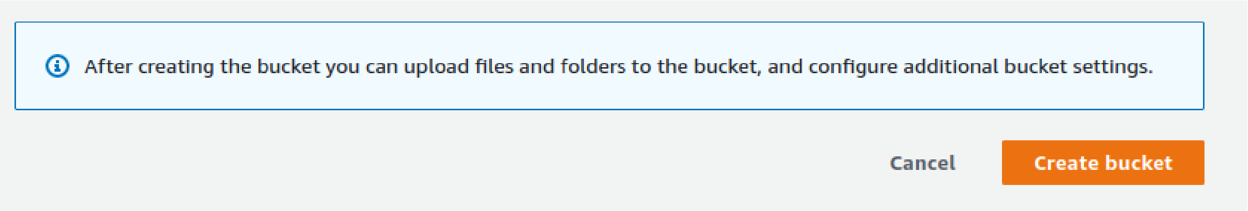
Object Lock
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Learn more

● Disable

○ Enable
Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

ⓘ Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Bucket Versioning.

14. Finally click on “Create bucket”. The bucket is created and you can start storing data to your newly created S3 bucket



ⓘ After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.