AWS - Deploy WordPress with Amazon RDS

Overview:

- WordPress is a highly popular content management system (CMS) that is used for over 30% of all sites on the internet. It is most commonly used for blogs but can also be used for running e-commerce sites, message boards, and many other popular use cases.

- In this Project, we will learn how to set up a WordPress site to run a blog. WordPress requires a MySQL database to store its data. For this Project, we will use Amazon RDS for MySQL to run our MySQL database.

- In the modules that follow, we'll see how to configure a WordPress installation using Amazon RDS for MySQL. To configure this WordPress site, we will create the following resources in AWS:
  a. An Amazon EC2 instance to install and host the WordPress application;
  b. An Amazon RDS for MySQL database to store our WordPress data.

Pre-req:

- AWS Account with admin level access.
- Browsers – Chrome, Firefox, Brave (No Ads), etc.,

Creating AWS Account:

- Use a personal account or Create a New AWS account for this project rather than using an organization's account to ensure that you have full access to the necessary services and do not leave behind any resources from the project.

- Once you have an AWS account set up, move on to the next module where you launch a MySQL database instance with Amazon RDS.

Modules:

This Labs are divided into five short modules. We must complete each module in order before moving on to the next one.

- Creating a MySQL Database with RDS: Configure the RDS database to allow access to specific entities.

- Creating an EC2 Instance: Create an EC2 instance to run your WordPress site.

- Configuring Your RDS Database: Configure the RDS database to allow access to specific entities.

- Configuring WordPress on EC2: Install the WordPress application and dependencies on the EC2 instance.

- Explore your New Web Site and Clean Up: Clean up the resources created in this lab so that you will not be charged.

Lab – 1: Creating a MySQL Database with RDS.

- Here, in this session we will create a MySQL database for your WordPress website.
- At this point, we have created an RDS database and an EC2 instance. In this session, we will configure the RDS database to allow access to specific entities.
- How to create MySQL Database for a WordPress website.
- Why WordPress needs a MySQL database and why Amazon RDS is a good choice for your database needs.

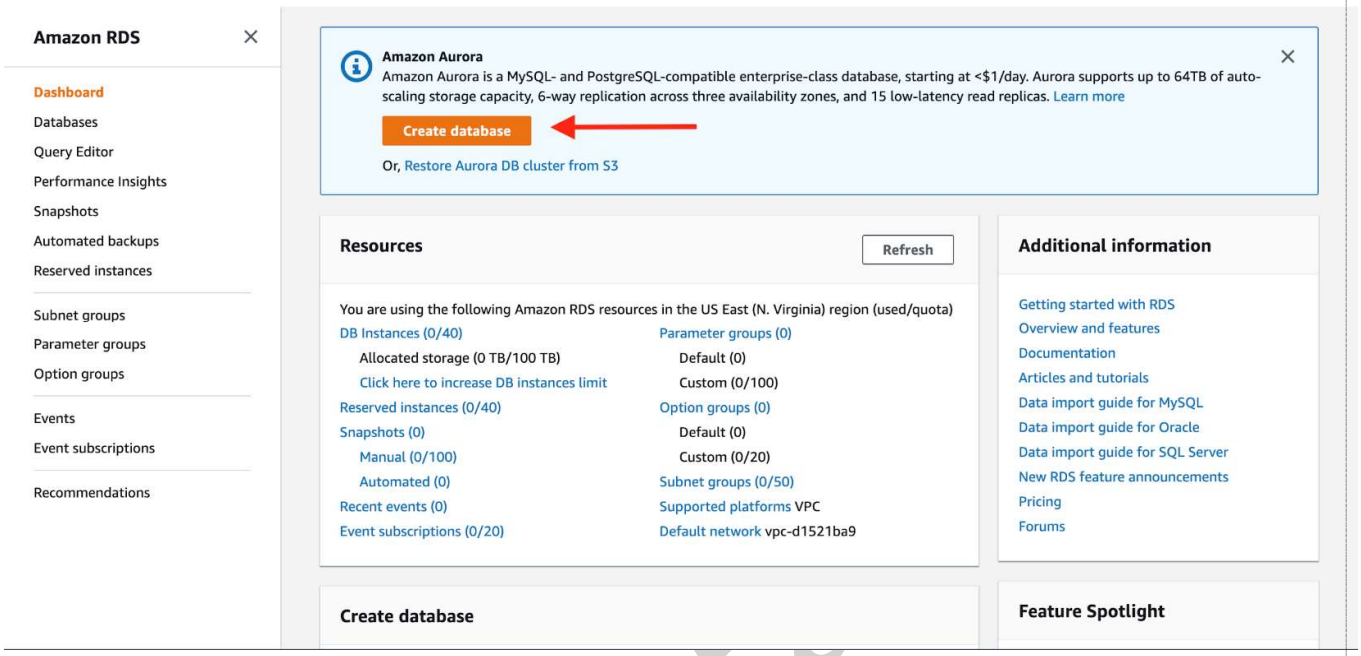
Why does WordPress need MySQL?

- WordPress is a flexible content management system (CMS) for building blogs, e-commerce sites, discussion boards, and more. For whatever kind of website, you're making, you will have content to store. In a blog, this will be your blog posts and comments. In an e-commerce site, it will be your products and user accounts.
- This content needs to be permanently stored somewhere. WordPress uses MySQL to store this content. A lot of the data in a WordPress application is hierarchical, structured data. For example, your application may have blog posts which have user-submitted comments. A relational database is a good choice for storing hierarchical data like this. Further, MySQL is the most popular open-source database, and it is a reliable, performant choice for this application.

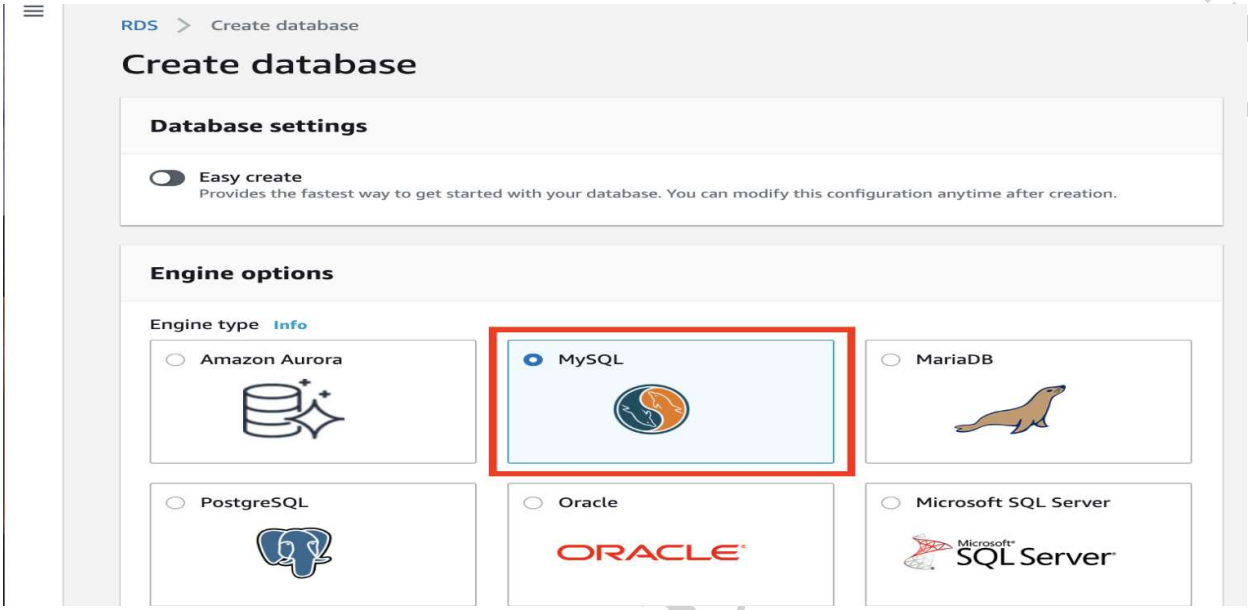
Why use Amazon RDS for your WordPress database?

- Many installation guides for WordPress use a MySQL database that is on the same server as the WordPress installation. While this may be sufficient to start, there are a number of reasons you may not want your MySQL database on the same server as your WordPress installation:
- MySQL and WordPress will be competing for compute resources on the same server, potentially hurting your site's performance.
- You are unable to horizontally scale WordPress by adding additional WordPress servers as your site becomes more popular.
- You are responsible for all database maintenance tasks, including database backups and security upgrades.
- By using Amazon RDS for MySQL, these concerns go away. Your database will be on a separate instance than your WordPress installation, so they won't be competing for resources.
- Further, you can create multiple WordPress installations that connect to a single MySQL instance on RDS, allowing you to scale your site horizontally.
- Finally, Amazon RDS for MySQL has automated backups and security patches to help you with your database administration.
- In the steps below, you will launch a MySQL database using the AWS management console.

- To begin, go to Amazon RDS in the AWS console. Click the orange Create database button to get started.



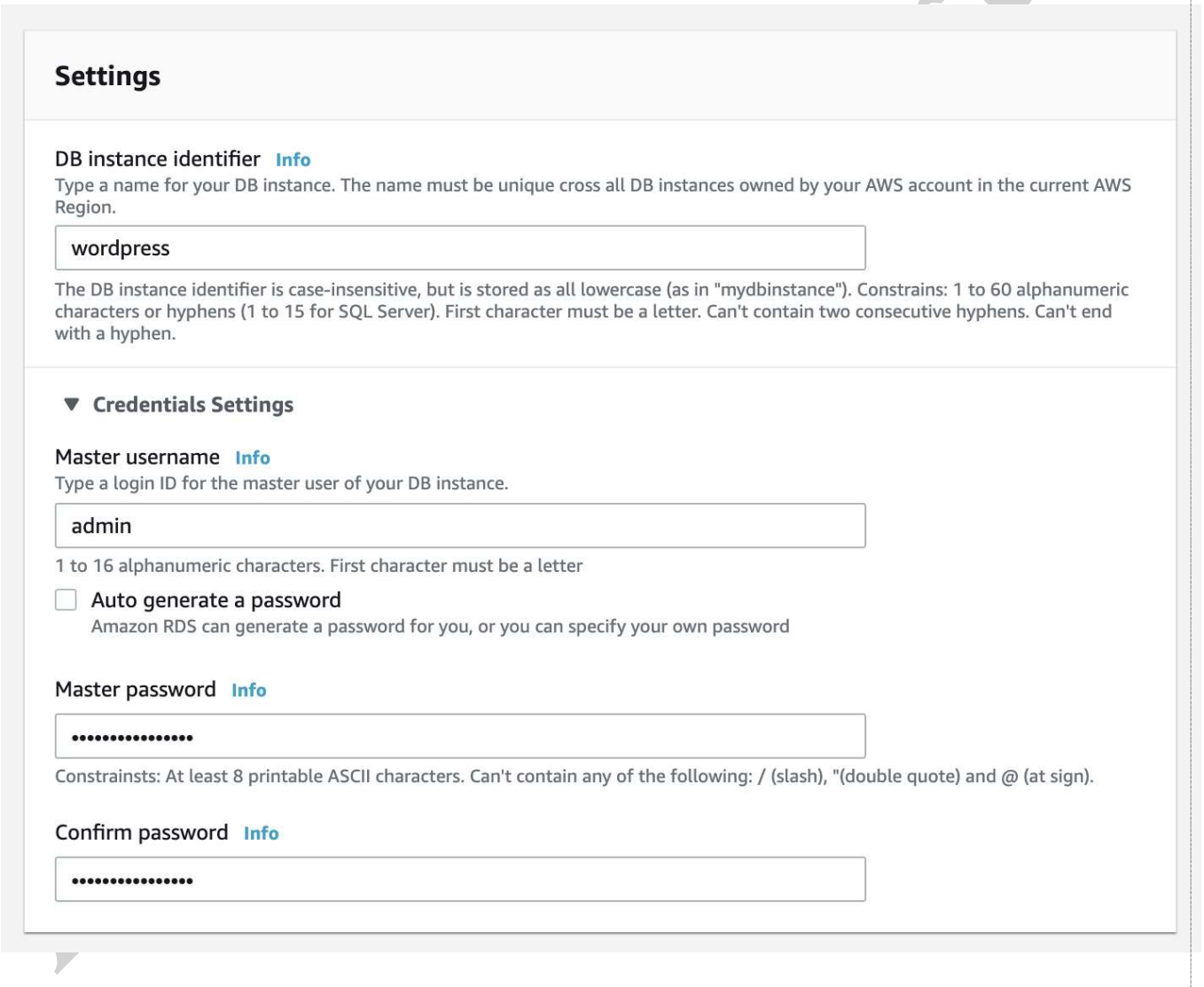
- The first step is to choose the database engine you want to use.
- Amazon RDS supports six different engines, from popular open-source options like MySQL and PostgreSQL, to commercial options like Oracle and Microsoft SQL Server, to a cloud-native option called Amazon Aurora that was custom-built to take advantage of the cloud.
- WordPress uses MySQL, so select that engine now.



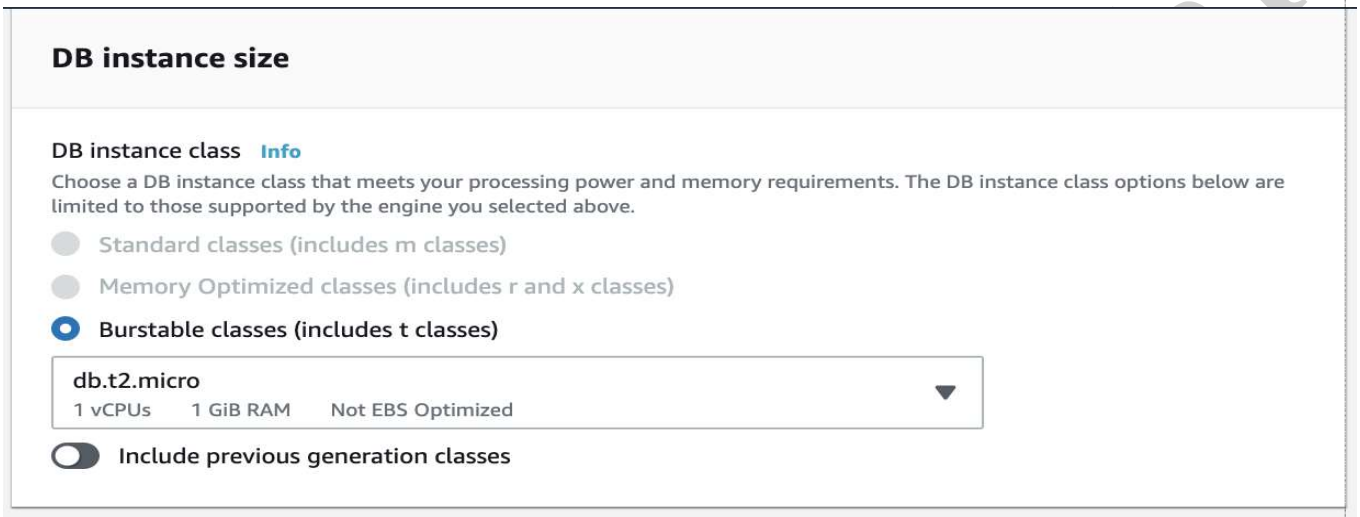
- In the Templates section of the creation wizard, there is an option to only show options that are available in the AWS Free Tier. Select this option now if you would like to use this lab for learning without spending any money.
- In a production setup, you may want to use features of Amazon RDS that are outside the free tier. These include:
- A larger database instance class, for improved performance;
- Multi-AZ deployments, for automatic failover and recovery in the event of an infrastructure issue;
- Provisioned IOPS for disk storage, for faster I/O performance.

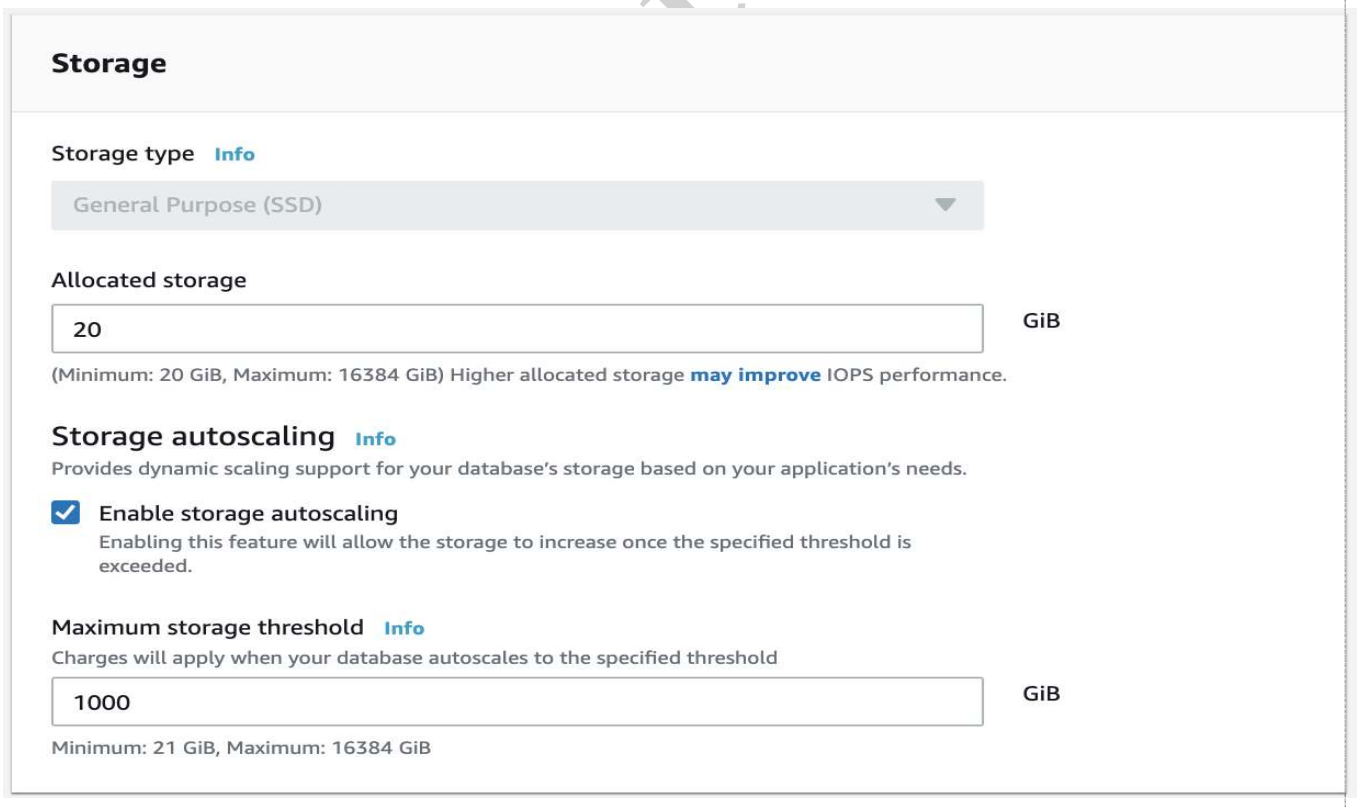
Next, you will specify the authentication settings for your MySQL deployment. This includes the database name and the master username and password.

- In the Settings section, enter WordPress as your DB instance identifier.
- Then specify the master username and password for your database.
- Choose a strong, secure password to help protect your database.
- Store the username and password for safekeeping as you will need it in a later module.

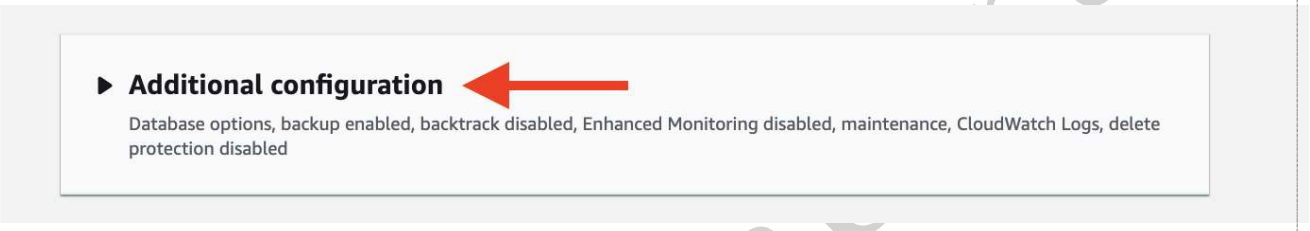


- After setting your username and password, you can select key details about your MySQL deployment. This includes the instance class and storage details.
- The default settings will work for this lab. We will use a small instance class that is suitable for testing or small-scale applications, and it fits within the AWS Free Tier.
- If you don't want to use the AWS Free Tier, you could set a larger instance class or alter the storage configuration options.

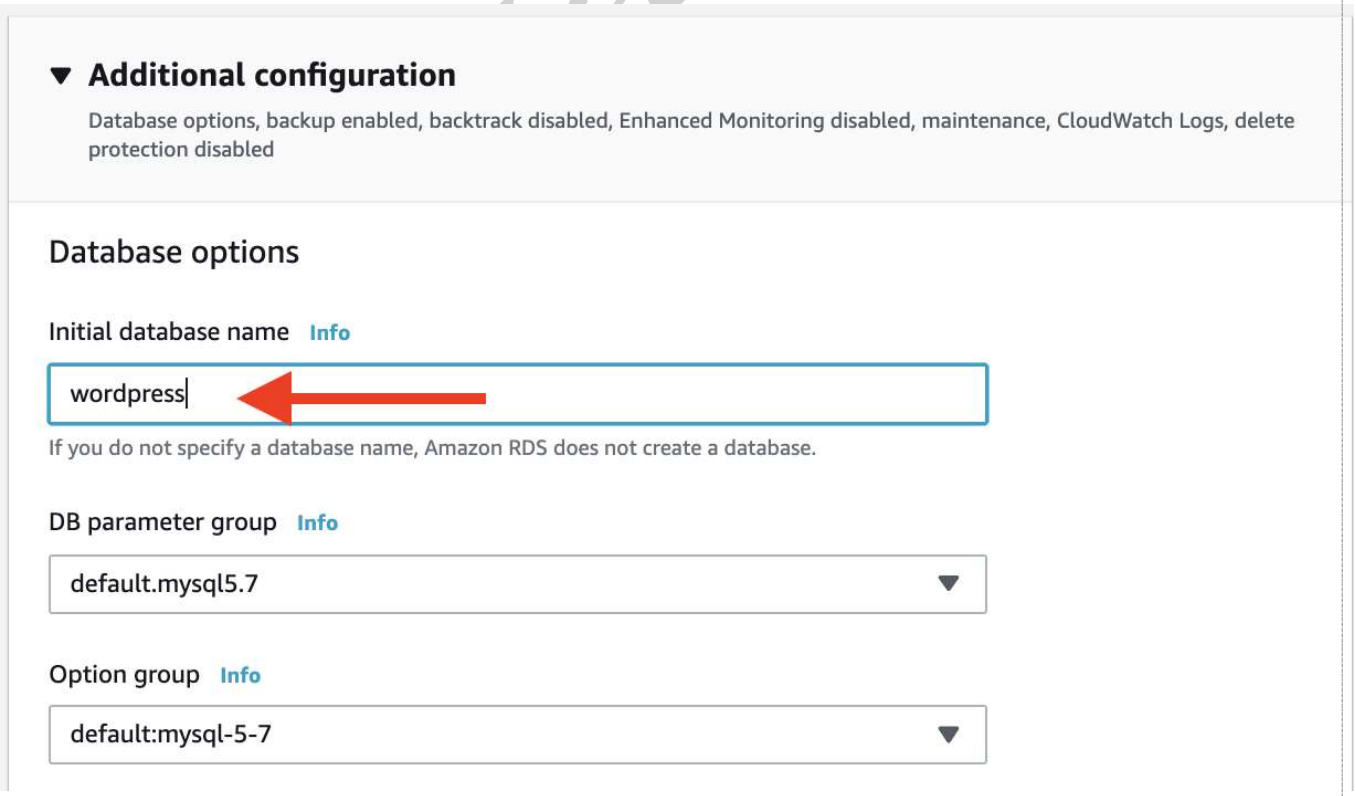




- Next, you can configure connectivity and network configuration. Amazon RDS instances must be created in an Amazon VPC, which is a logically-separate network where your provisioned resources will live.
- VPCs are an advanced topic outside the scope of this lab. Fortunately, AWS has created a default VPC in each region in your account.
- The default VPC is already selected for you, and you can launch your RDS instance in this VPC.
- Finally, RDS provides a number of additional configuration options to customize your deployment. You need to make one change in this area. Click on the Additional configuration line to expand the options.

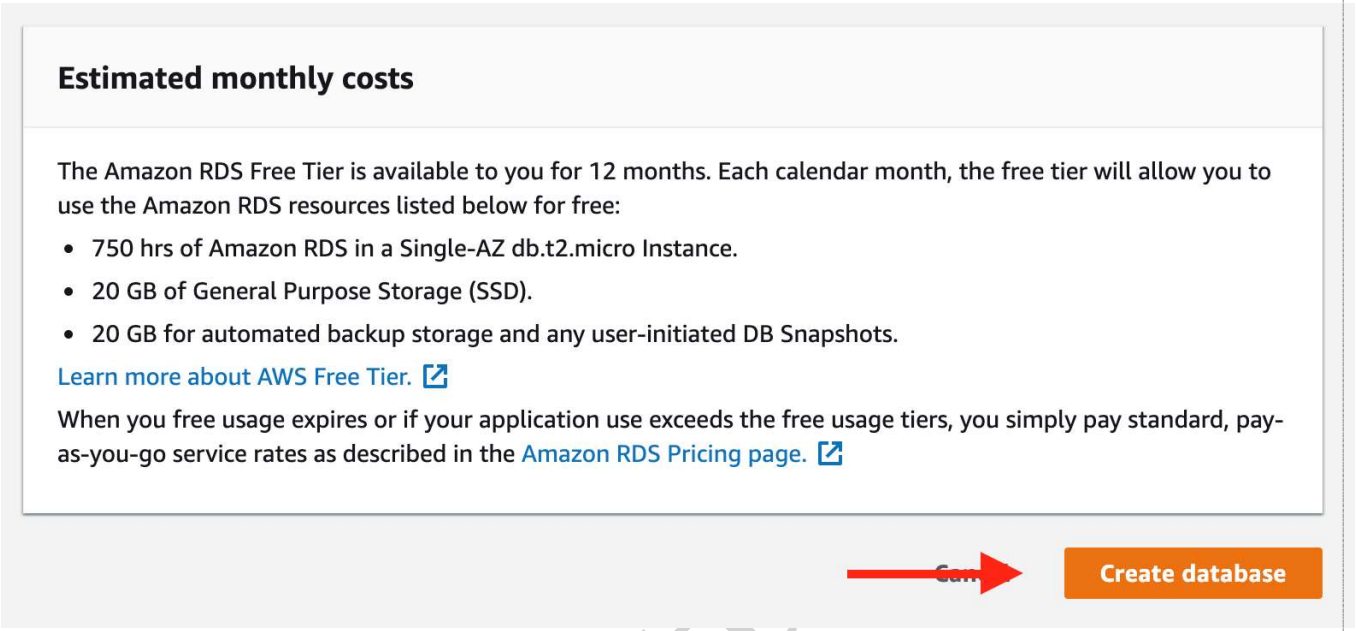


- Set the initial database name to WordPress. This will ensure RDS creates the database in your MySQL instance upon initialization.
- We will use this database name when connecting to your database.

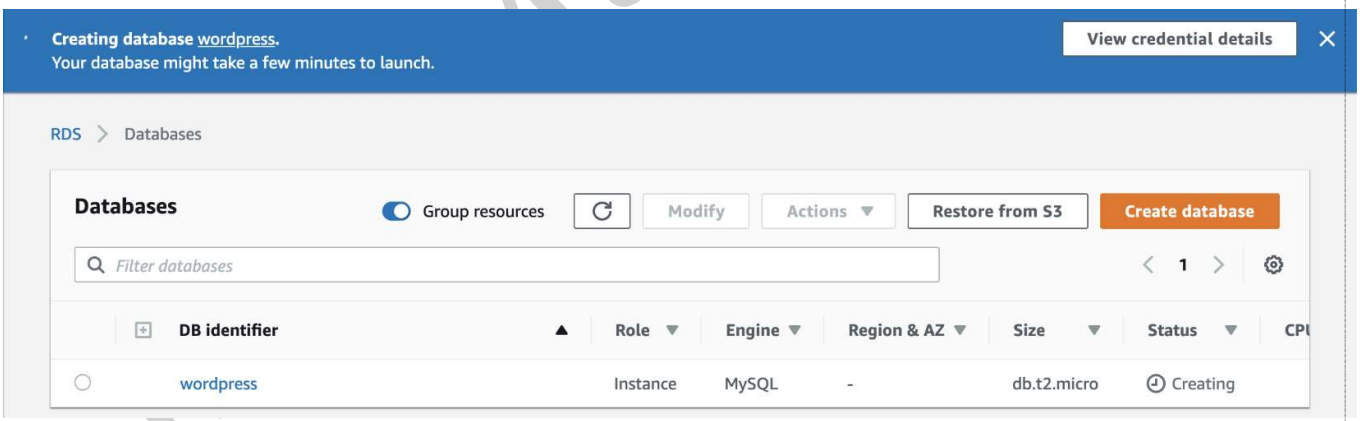


- At the bottom of the creation wizard, AWS will show you estimated monthly costs for your RDS database. If you are still eligible for the Amazon RDS Free Tier, you will see a note that the database will be free to you for up to 12 months.

- Click the orange Create database button to create your database.



- You should see a success notice indicating that your database is being created.



- In this module, you created a fully-managed MySQL database using Amazon RDS.

- **In the next module, we will create an Amazon EC2 instance for running your WordPress site.**

Lab – 2: Creating an EC2 Instance to run WordPress site.

- In this session, we will create an Amazon EC2 instance to run our WordPress site.
- Amazon EC2 provides highly-configurable server instances on-demand. On an EC2 instance, we can run a WordPress site that will be accessible by users anywhere.
- Create an Amazon EC2 instance to run your WordPress site.

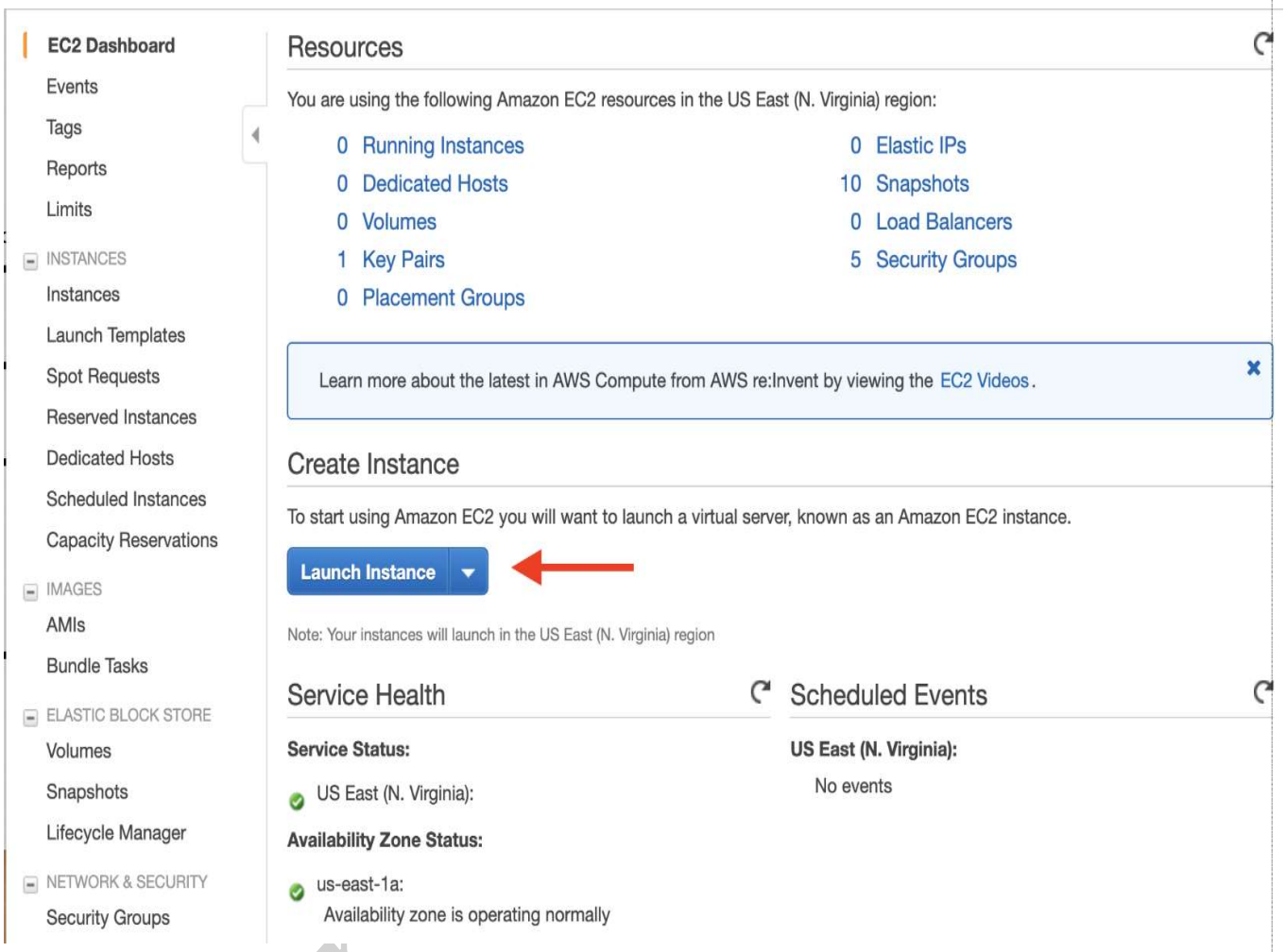
Why use Amazon EC2 for your WordPress site?

- When getting started with WordPress, you may test it out by installing and running it on your laptop or desktop. This is fine for a test but you will quickly hit its limitations.
- Your WordPress site will only be running as long as your laptop or desktop is running. Further, the site will only be accessible by you -- it won't be available over the public internet.
- A better approach is to use a *server*.
- Amazon EC2 provides on-demand server provisioning. With Amazon EC2, you rent out server instances with varying sizes, each with different CPU, RAM, and network configuration.
- You pay by the hour for these servers, and you can use them to host websites, like your WordPress site. With an EC2 instance, your WordPress site will remain up and running and will be accessible by anyone over the internet.

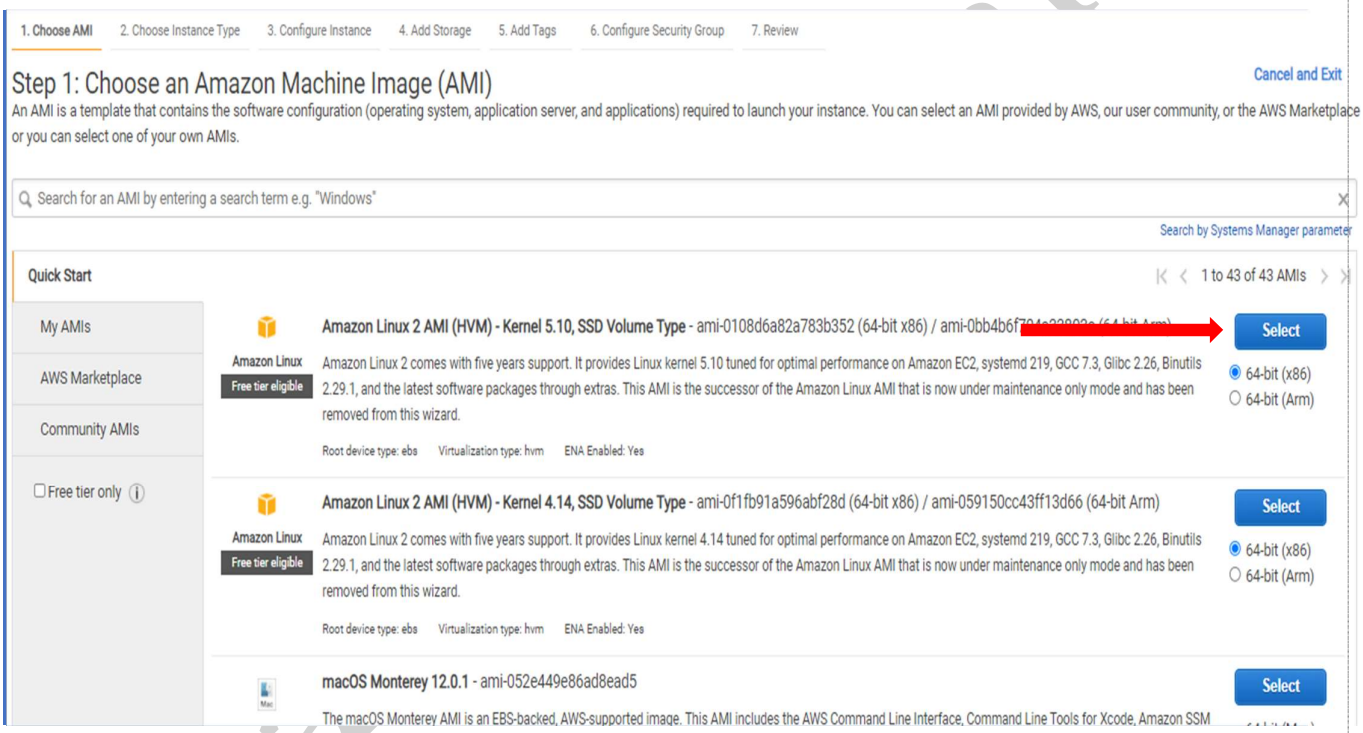
In the steps below, we will launch an EC2 instance to host your WordPress site.

Choosing Amazon Machine Image:

- To create your EC2 instance, go to Amazon EC2 in the AWS console. Click the blue button that says Launch instance to open the instance creation wizard.

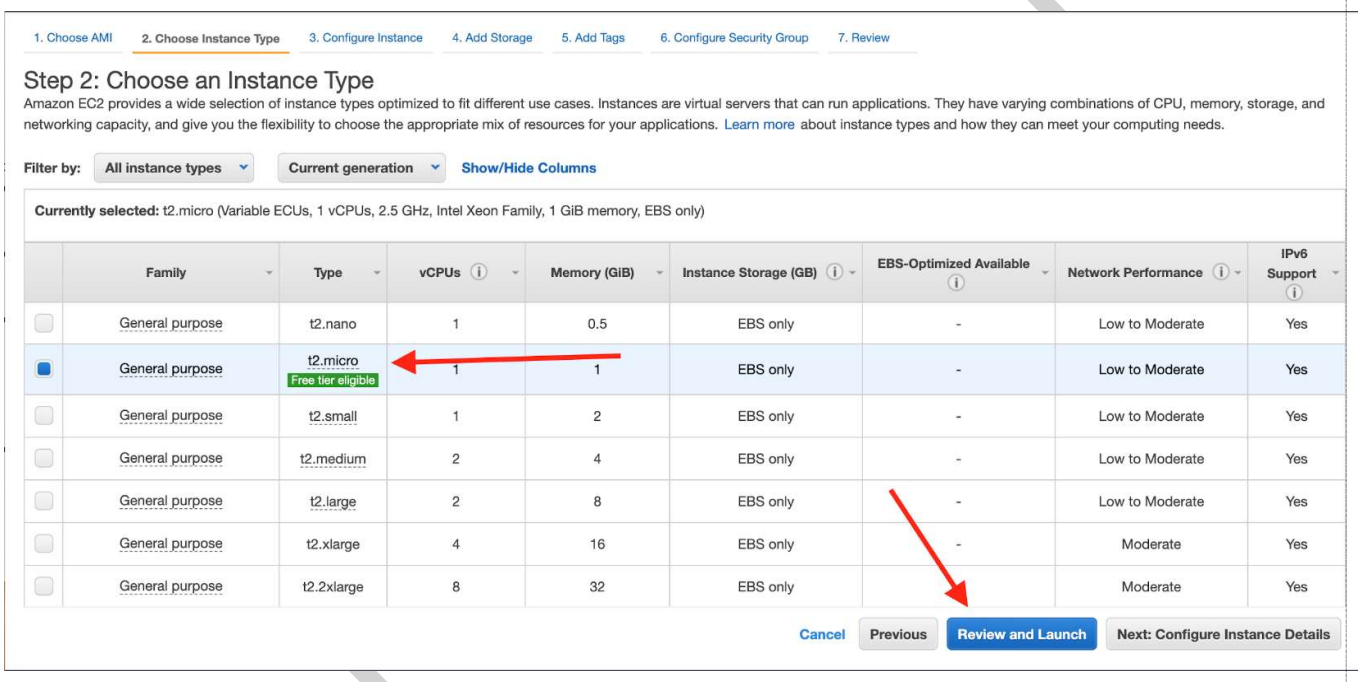


- In the first page, we will choose an Amazon Machine Image (“AMI”).
- The AMI you choose will determine the base software that is installed on your new EC2 instance.
- This includes the operating system (Amazon Linux, Red Hat Enterprise Linux, Ubuntu, Microsoft Server, etc.), as well as the applications that are installed on the machine.
- Many AMIs are general-purpose AMIs for running many different applications, but some are purpose-built for specific use cases, such as the Deep Learning AMI or various AWS Marketplace AMIs.
- The Amazon Linux distro is a popular choice, so choose the Amazon Linux 2 AMI (HVM) in the AMI selection view.



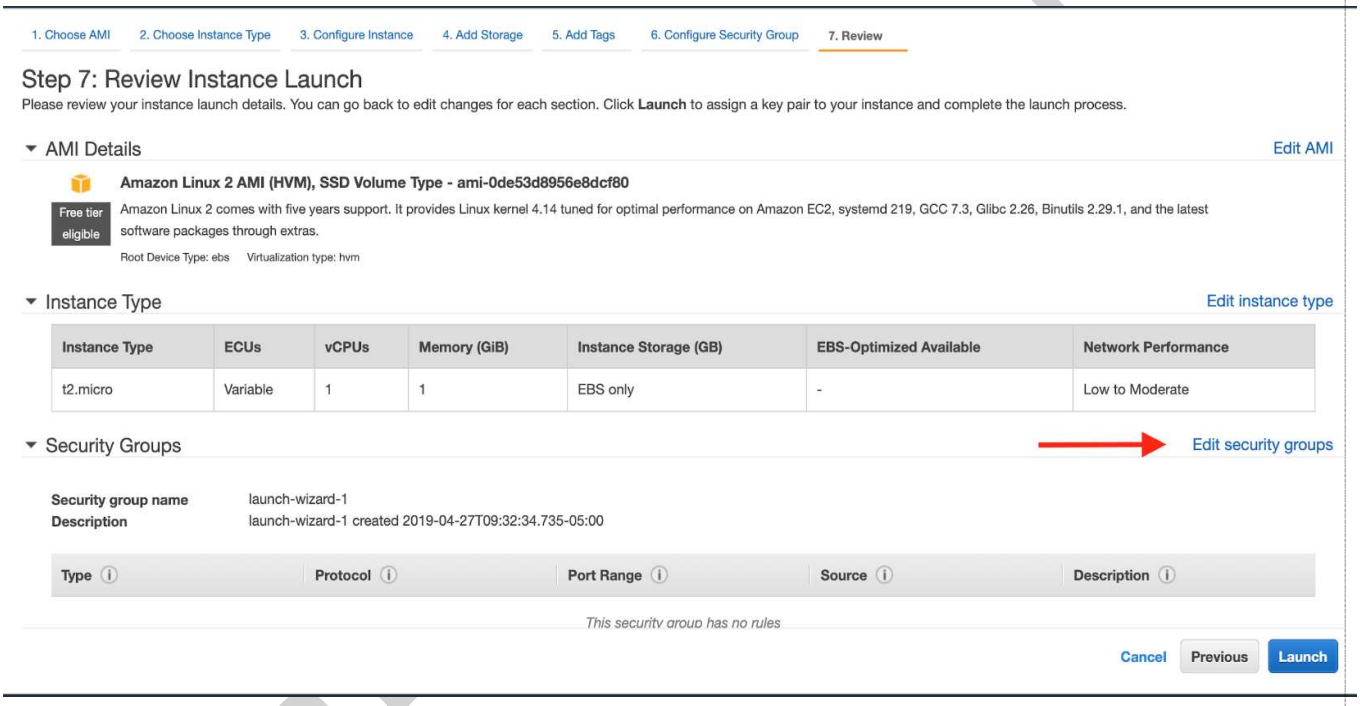
Choosing an Instance Type:

- On the second screen of the EC2 wizard, you will select an EC2 instance type. An instance type is a particular configuration of CPU, memory (RAM), storage, and network capacity.
- AWS has a huge selection of instance types that cover many different workloads. Some are geared toward memory-intensive workloads, like databases and caches, while others are aimed at compute-heavy workloads like image processing or video encoding.
- Amazon EC2 allows you to run 750 hours per month of a t2.micro instance under the AWS Free Tier. Select this option for this lab so that you won't incur any costs on your bill.
- After selecting the t2.micro instance, click the blue Review and Launch button to skip some of the advanced configuration steps.

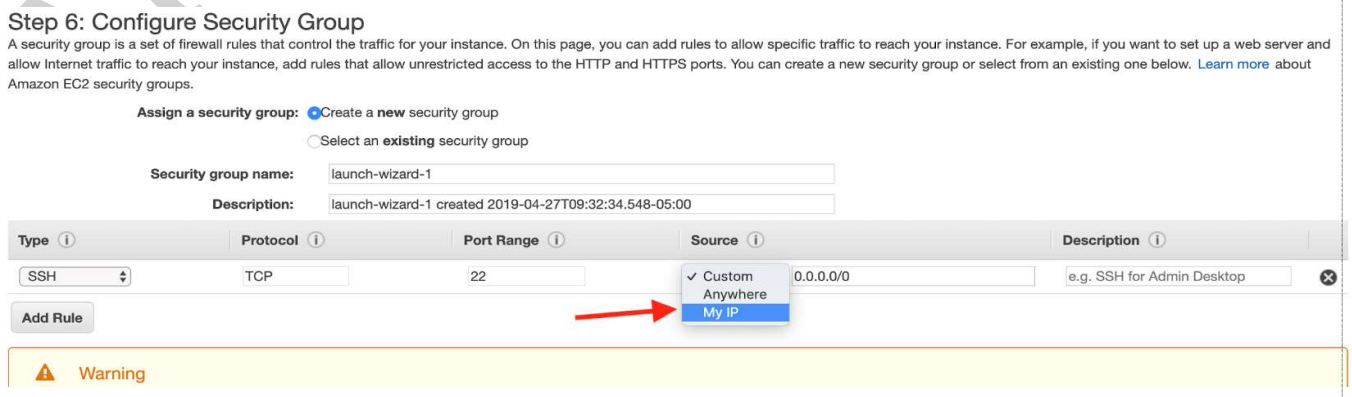


Configuring a Security Group:

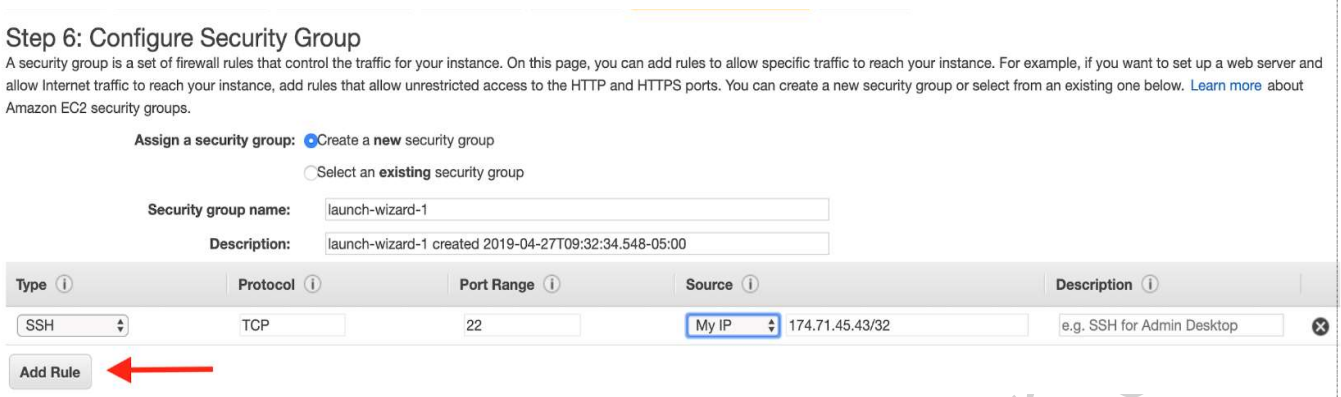
- After clicking the Review and Launch button, you will be at a Review Instance Launch screen. You need to configure one more thing before launching your instance.
- Security groups are networking rules that describe the kind of network traffic that is allowed to your EC2 instance. You want to allow two kinds of traffic to your instance:
- SSH traffic from your current IP address so you can use the SSH protocol to log into your EC2 instance and configure WordPress;
- HTTP traffic from all IP addresses so that users can view your WordPress site.
- To configure this, click the Edit security groups link on the review page.



- It will show the current rules in your security group.
- There is an SSH rule configured, but it allows SSH access from any IP address. Click under Source to restrict it to your current IP address.



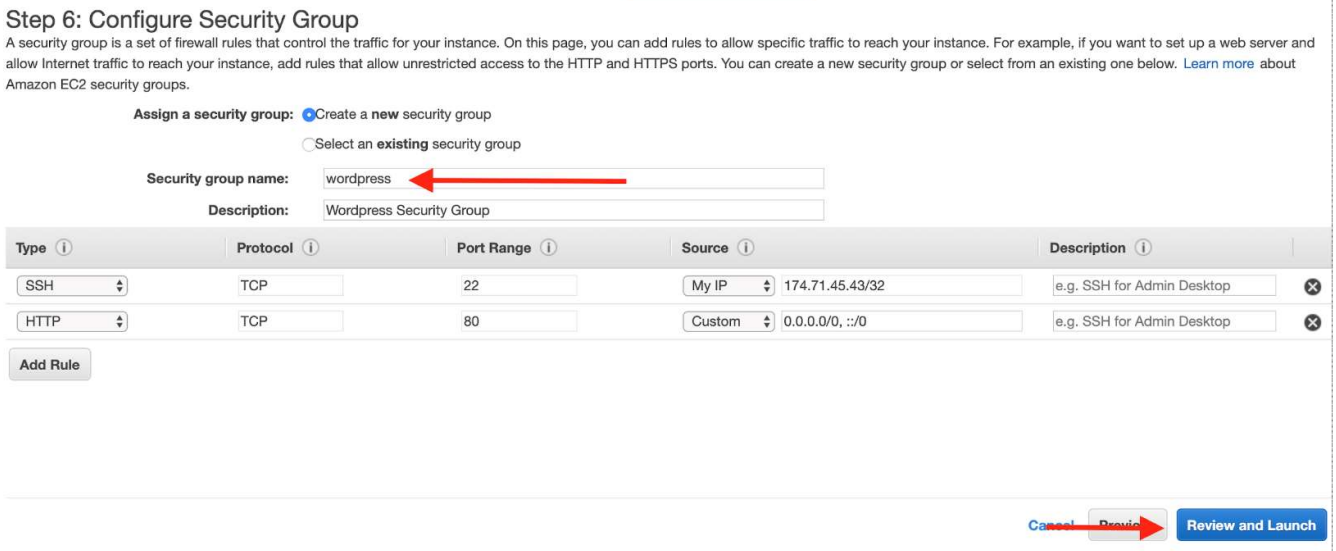
- Then, you need to add a new rule to allow HTTP traffic. Click Add Rule.



- In the new rule that shows up, click the dropdown under the Type column. Select HTTP, and it will autofill default values for an HTTP rule.

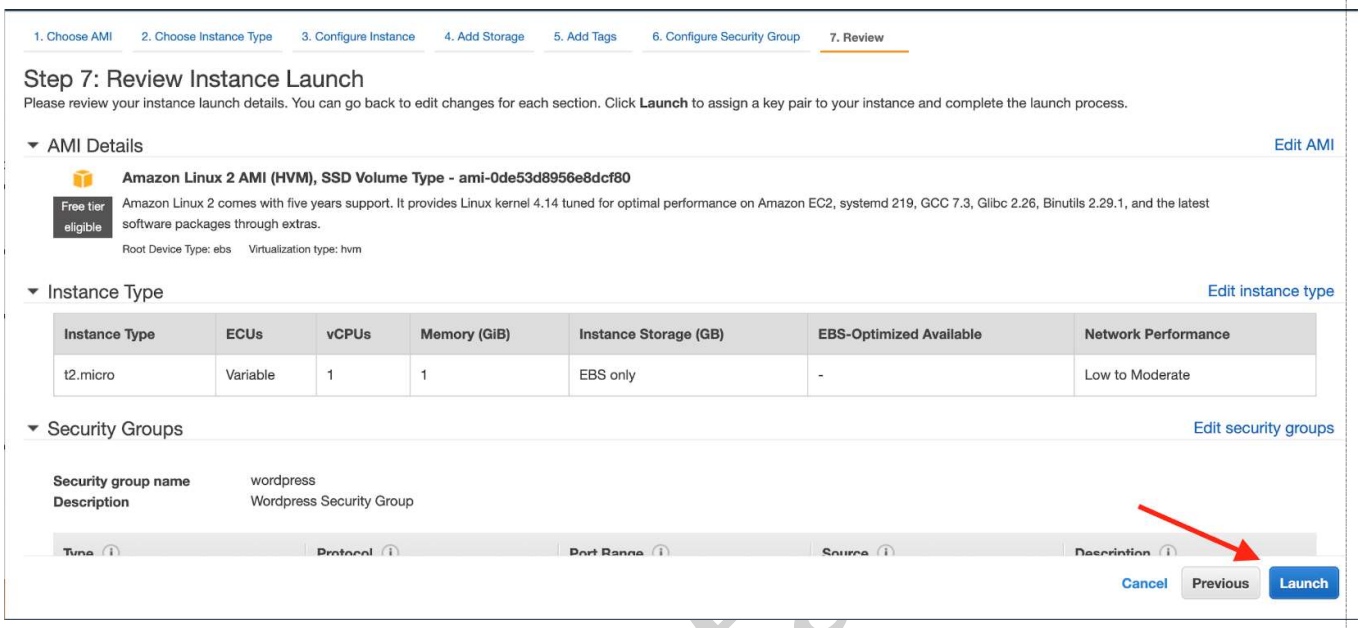
- Once you have the security group rules in place, give your security group a name in the Security group name input box. Name the group “WordPress” so that it will be easy to find.

- Once you’ve named it, click the blue Review and Launch button.



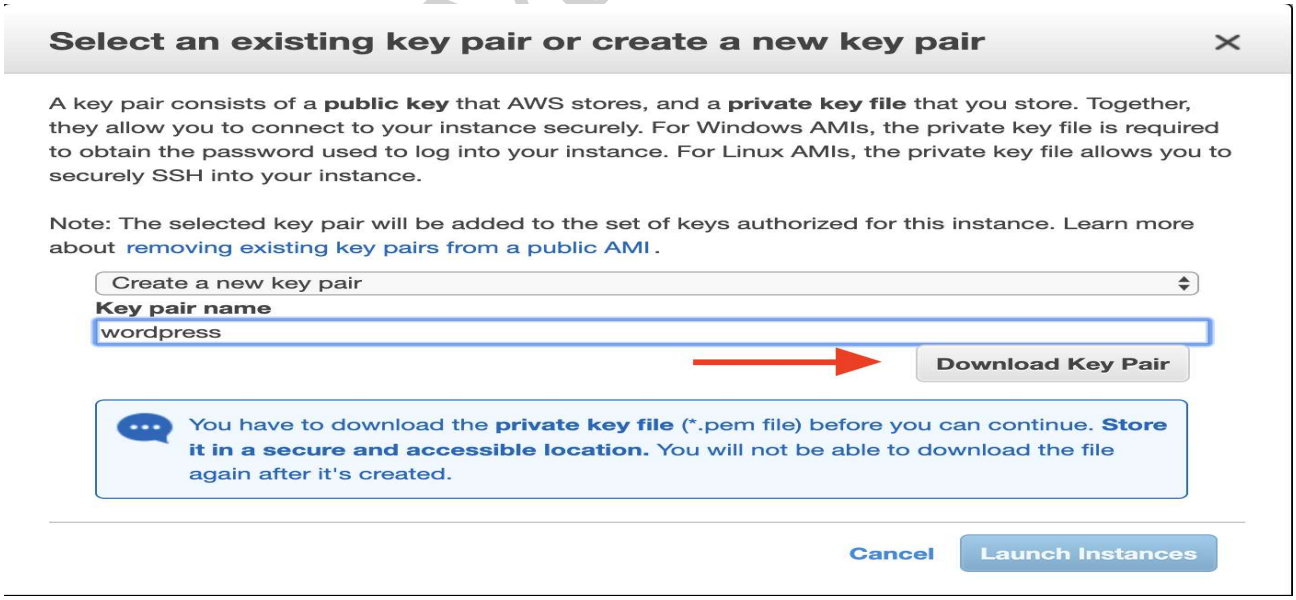
Launch and get an SSH Key:

- It is now time to launch your EC2 instance. Click the blue Launch button to create your EC2 instance.

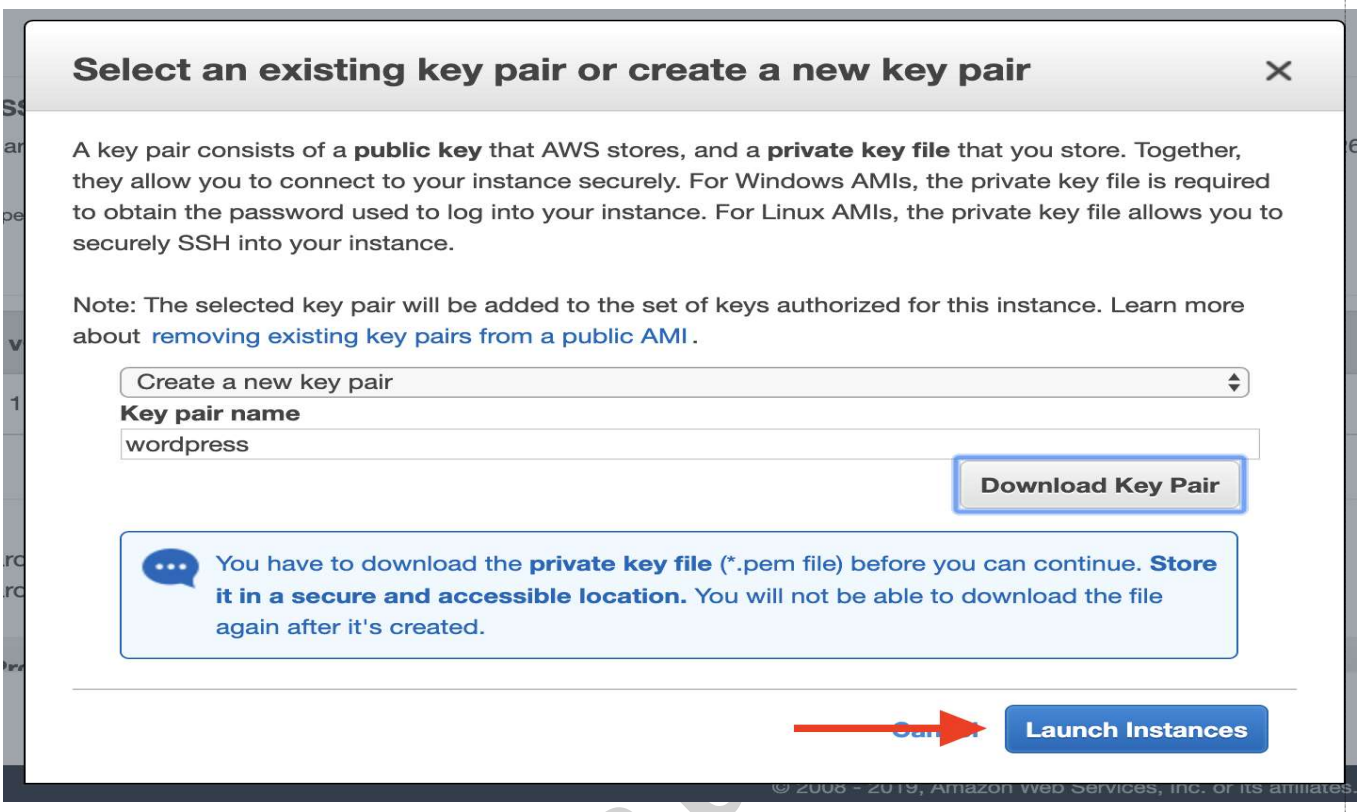


- You will see a details on how to configure a key pair for your instance. You will use the key pair to SSH into your instance, which will give you the ability to run commands on your server.

- Create a new key pair for your instance and give it a name. Then click the Download Key Pair button to download the .pem file to your machine, which you will use in the next module.



- Once your key pair is downloaded, click the blue Launch Instances button to launch your EC2 instance.



- You have successfully launched your EC2 instance.

In the next module, we will configure your RDS database to work with your EC2 instance.

Lab – 3: Configuring your RDS database.

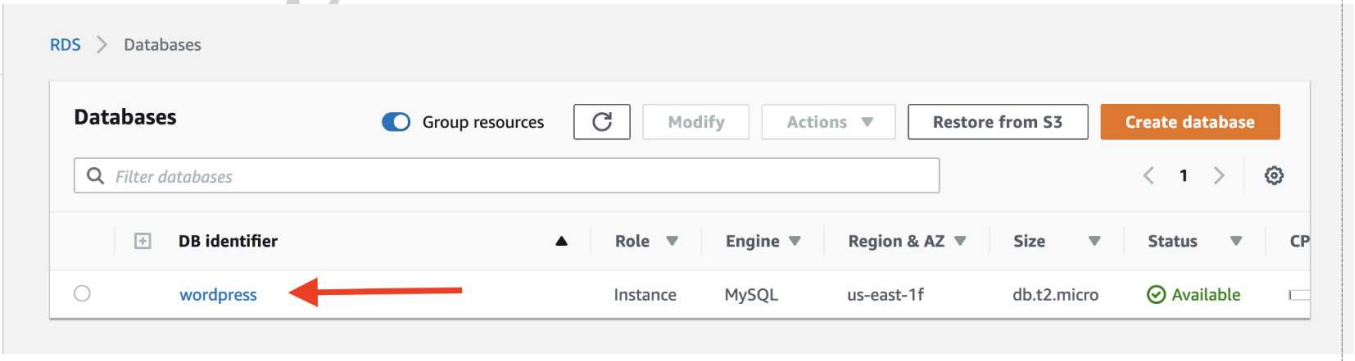
- At this point, you have created an RDS database and an EC2 instance. In this module, we will configure the RDS database to allow access to specific entities.
- Configure the RDS database to allow access to specific entities

Database Security Methods

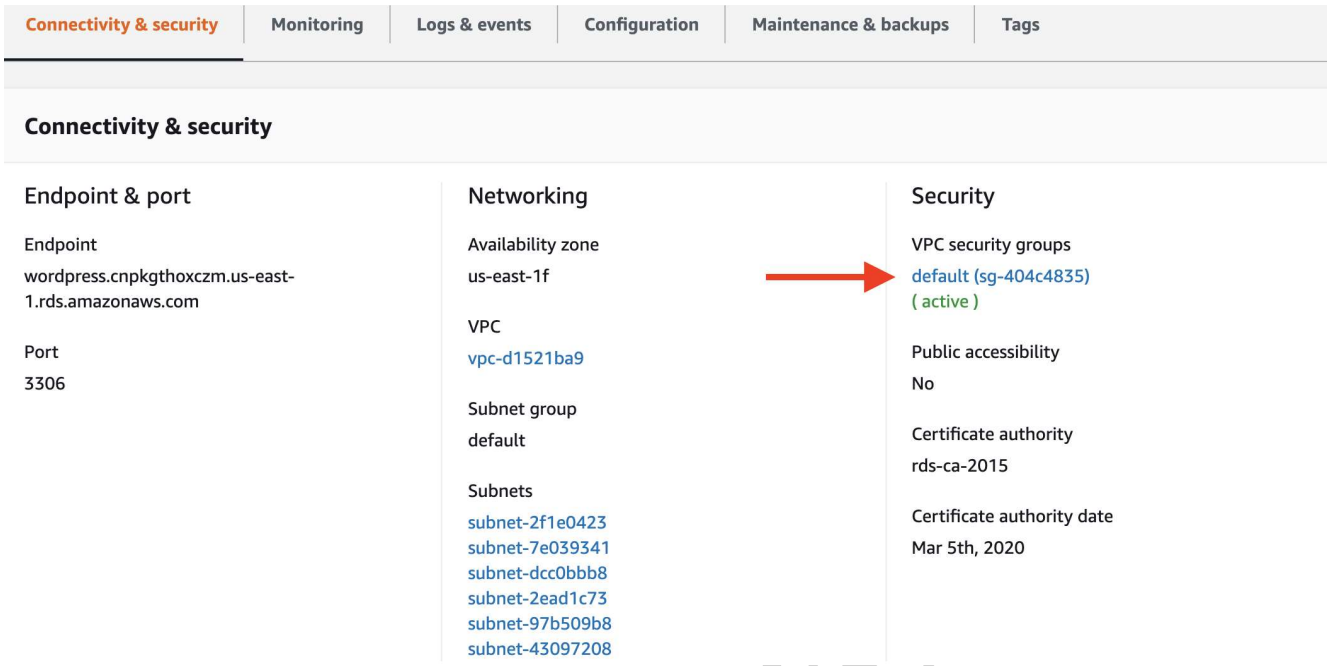
- It is critical to secure your database from unauthorized access, and there are a number of strategies you can use to add security to your database. You will learn two of them in this module. They are:
    a. Network security: Limiting access to your database instance by rejecting traffic that's not from authorized IP addresses
    b. Password authentication and authorization: Limiting access to your database by requiring a username and password to access.
- You will configure each of these in the steps below.

Implementation:

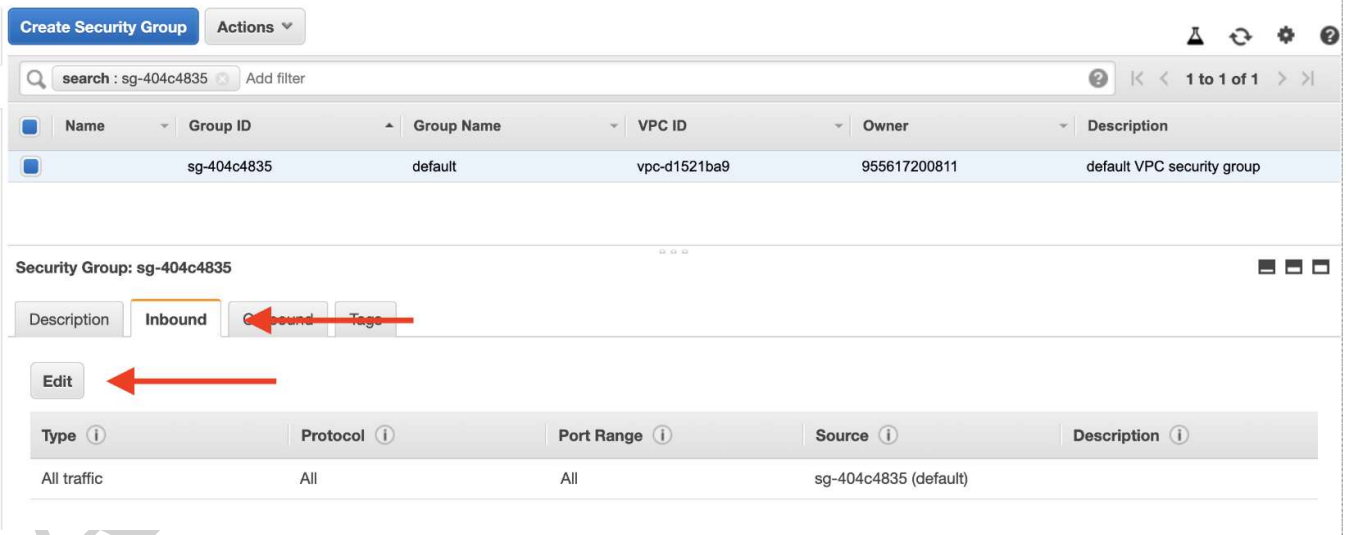
- Allow your EC2 Instance to access your RDS Database
- First, you will modify your RDS database to allow network access from your EC2 instance.
- In the previous module, you created security group rules to allow SSH and HTTP traffic to your WordPress EC2 instance. The same principle applies here.
- This time, you want to allow certain traffic from your EC2 instance into your RDS database.
- To configure this, go to the RDS databases in the AWS console. Click on the MySQL database you created in an earlier module in this lab.



- Scroll to the Connectivity & security tab in the display, and click on the security group listed in VPC security groups.

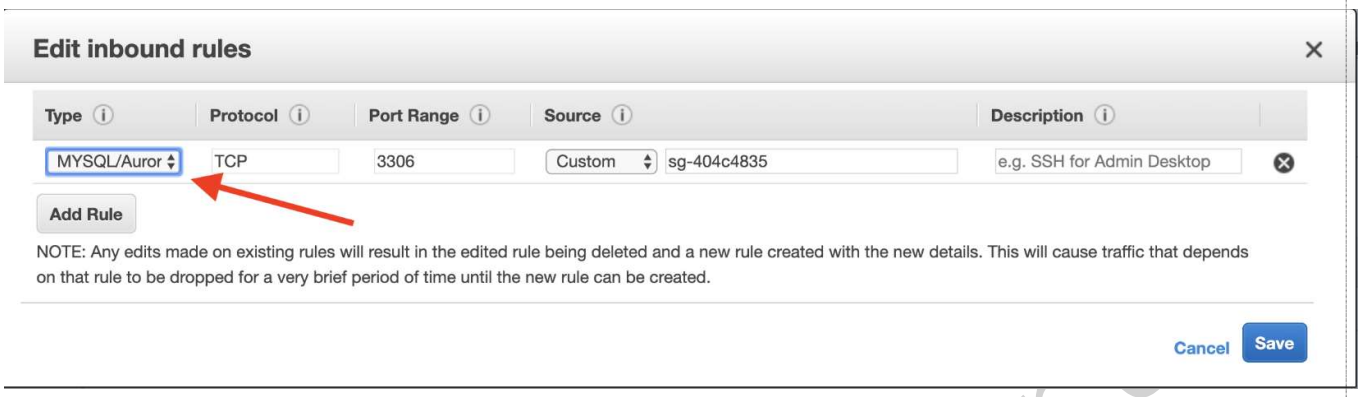


- The console will take you to the security group configured for your database.
- Click the Inbound tab, then click the Edit button to change the rules for your security group.

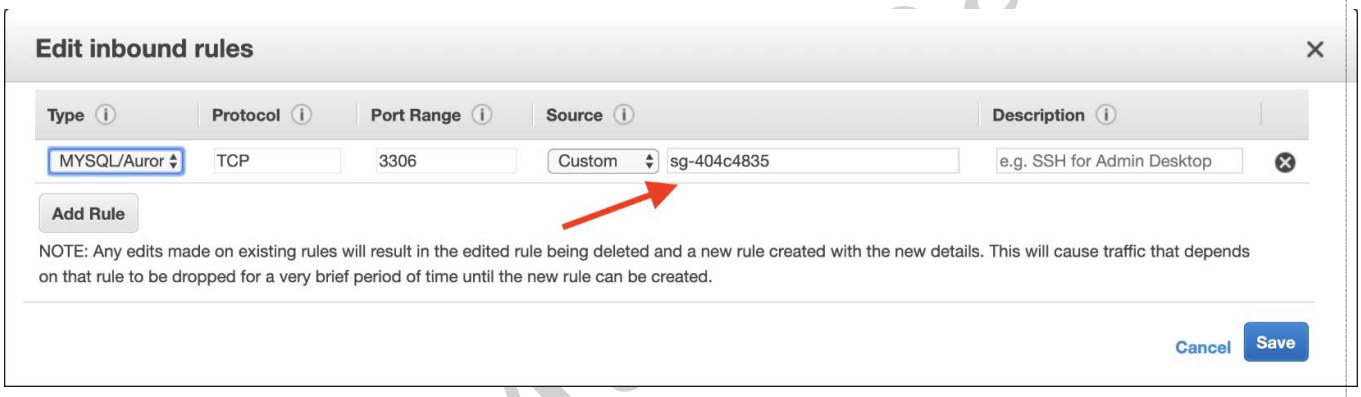


- The default security group has a rule that allows all inbound traffic from other instances in the default security group.
- However, since your WordPress EC2 instance is not in that security group, it will not have access to the RDS database.

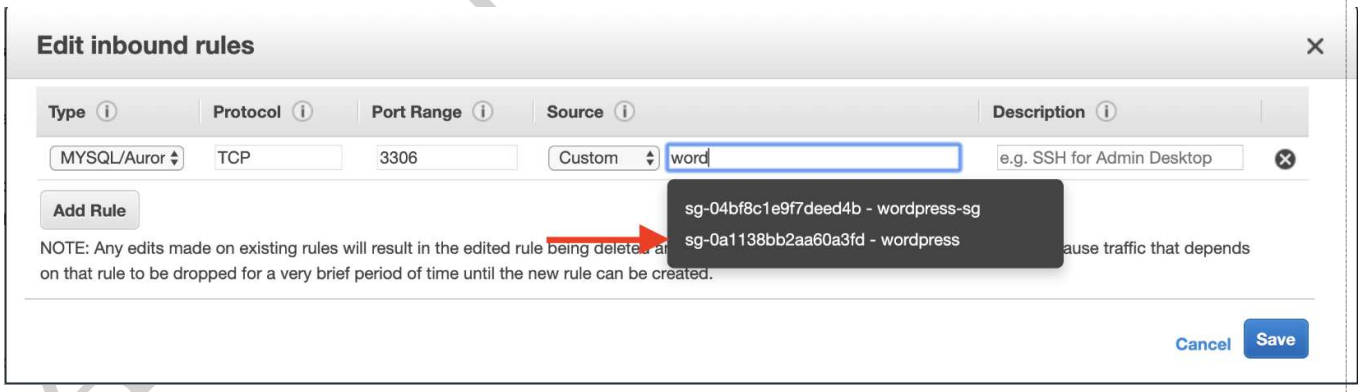
- Change the Type property to MYSQL/Aurora, which will update the Protocol and Port Range to the proper values.



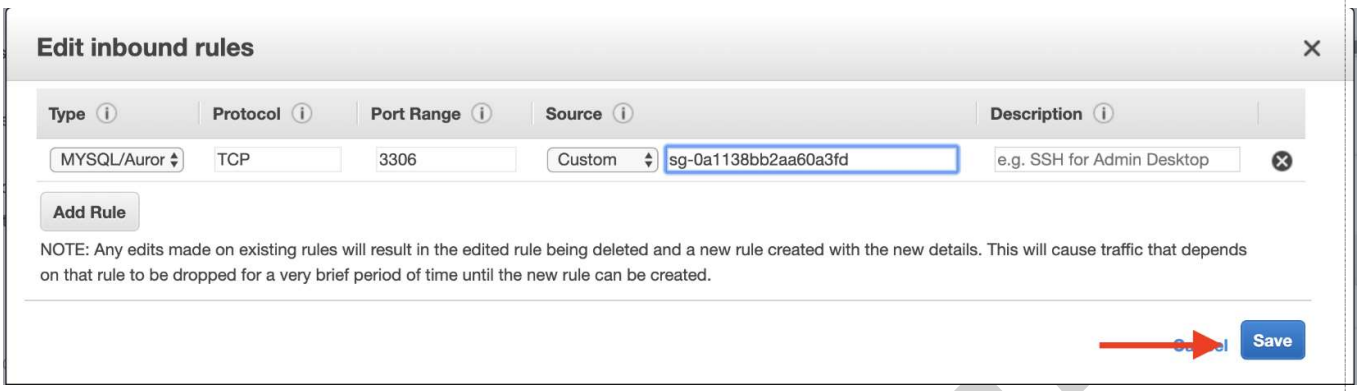
- Then, remove the current security group value configured for the rule, and type "WordPress" instead. The console will show the available security groups that are configured.



- Click on the "WordPress" security group that you used for your EC2 instance.

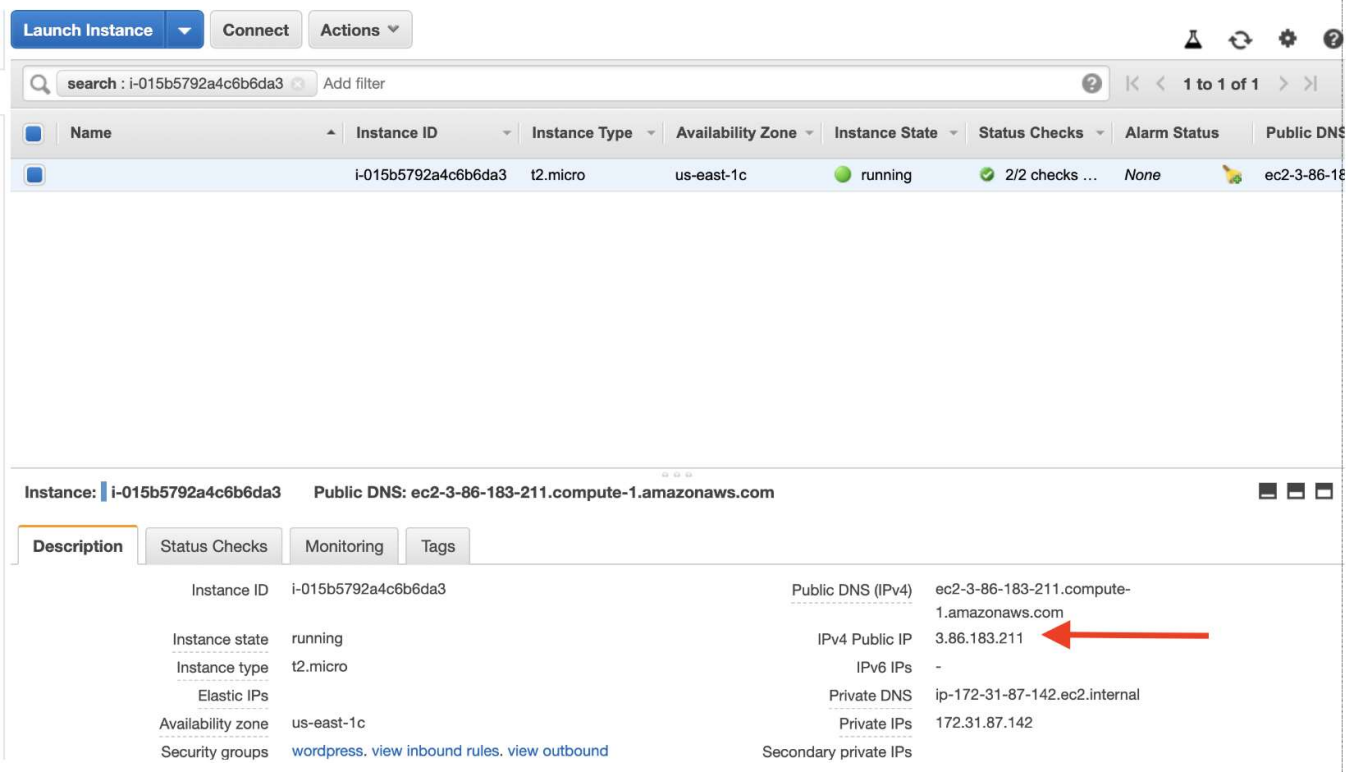


- After you click, it will fill in the security group ID. This rule will allow MySQL access to any EC2 instance with that security group configured.
- When you're finished, hit the blue Save button to save your changes.



SSH into your EC2 Instance

- Now that your EC2 instance has access to your RDS database, you will SSH into your EC2 instance and run some configuration commands.
- Go to the EC2 instances page in the AWS console. You should see the EC2 instance you created for the WordPress installation.
- Click on it, and you will see a public IP address labeled IPv4 Public IP in the instance description.



- Save this IP address, as you will need it when you SSH into your instance.

- Previously, you downloaded the .pem file for the key pair of your instance. Locate that file now. It will likely be in a Downloads folder on your desktop.

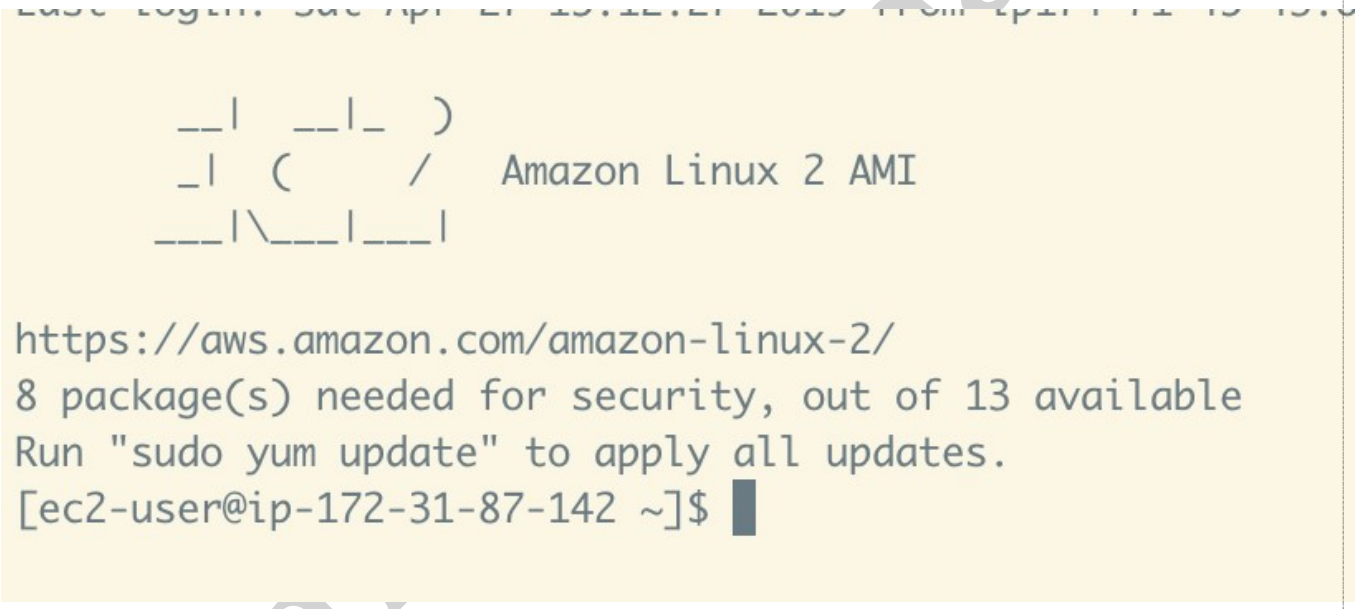
For Mac or Linux users:

- Open a terminal window. If you are on a Mac, you can use the default Terminal program that is installed, or you can use your own terminal.
- In your terminal, run the following commands to SSH into your instance. Replace the **"<path/to/pem/file>"** with the path to your file, e.g. **"~/Downloads/wordpress.pem"**, and the **"<publicIpAddress>"** with the public IP address for your EC2 instance.

  chmod 600 <path/to/pem/file>

  ssh -i <path/to/pem/file> ec2-user@<publicIpAddress>

- You should see the following in your terminal to indicate that you connected successfully:



For Windows users:

- You will need to use PuTTY, an SSH client for Windows, to connect to your EC2 instance.
- For instructions on doing this, see this guide for Connecting to your Linux instance from Windows using PuTTY.
- You will need the .pem file you downloaded and the public IP address of your EC2 instance.
- In this step, you connected to your EC2 instance via SSH. In the next step, you will connect to your RDS database from your EC2 instance and create a database user for the WordPress application.

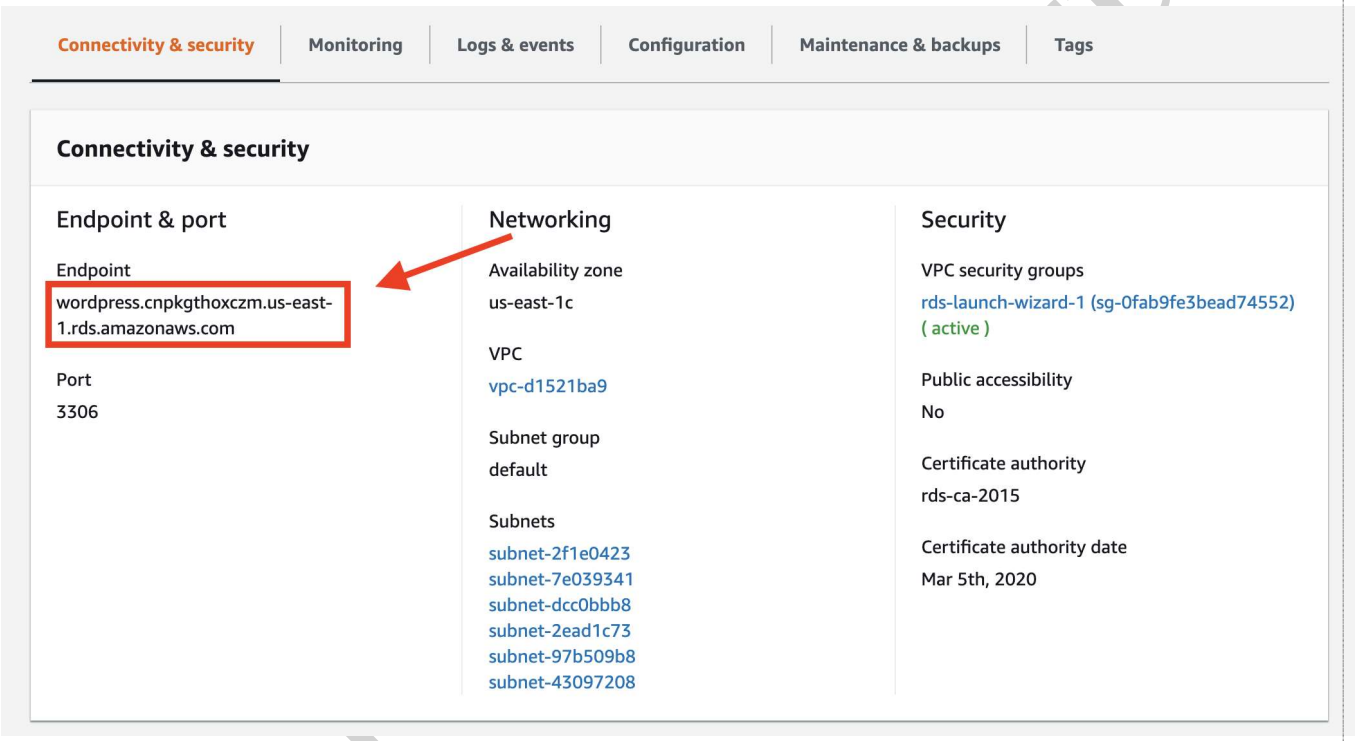
Create a Database User

- You should have an active SSH session to your EC2 instance in the terminal. Now, you will connect to your MySQL database.

- First, run the following command in your terminal to install a MySQL client to interact with the database.

  sudo yum install -y mysql

- Next, find the hostname for your RDS database in the AWS console. In the details of your RDS database, the hostname will be shown as the Endpoint in the Connectivity & security section.



- In your terminal, enter the following command to set an environment variable for your MySQL host. Be sure to replace "<your-endpoint>" with the hostname of your RDS instance.

  export MYSQL_HOST=<your – rds – endpoint>

- Next, run the following command in your terminal to connect to your MySQL database. Replace "<user>" and "<password>" with the master username and password you configured when creating your RDS database.

  mysql --user=<user> --password=<password> wordpress

- If you connected successfully, your terminal should indicate connection to the MySQL database as shown in the following image.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.6.40-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [wordpress]>
```

- Finally, create a database user for your WordPress application and give it permission to access the "WordPress" database.

Run the following commands in your terminal:

CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';

GRANT ALL PRIVILEGES ON wordpress. TO wordpress;*

FLUSH PRIVILEGES;

EXIT

- You should use a better password than "WordPress-pass" to secure your database.
- Write down both the username and password that you configure, as it will be needed in the next module when setting up your WordPress installation.
- In this module, you learned how to configure network and password security for your RDS database. Your EC2 instance now has network access to your RDS database. Further, you created a database user to be used by your WordPress application.

In the next module, you will configure your EC2 instance to run the WordPress application.

Lab – 4: Configuring WordPress on EC2.

- To this point, you have done a lot of configuration setup. You created an RDS instance and an EC2 instance.
- You allowed network access from your EC2 instance to your RDS instance. You learned how to SSH into your EC2 instance and configured a database user to be used by WordPress.
- In this module, you will finish up the work to make your WordPress site live. You will install the WordPress application and dependencies on the EC2 instance.
- At the end of this module, you will have a WordPress installation that is accessible in the browser from anywhere in the world.
- To complete the steps in this module, you will need to SSH into your EC2 instance. Please review the steps in the previous module if you need to reconnect to your EC2 instance via SSH.
- Install the WordPress application and dependencies on the EC2 instance.

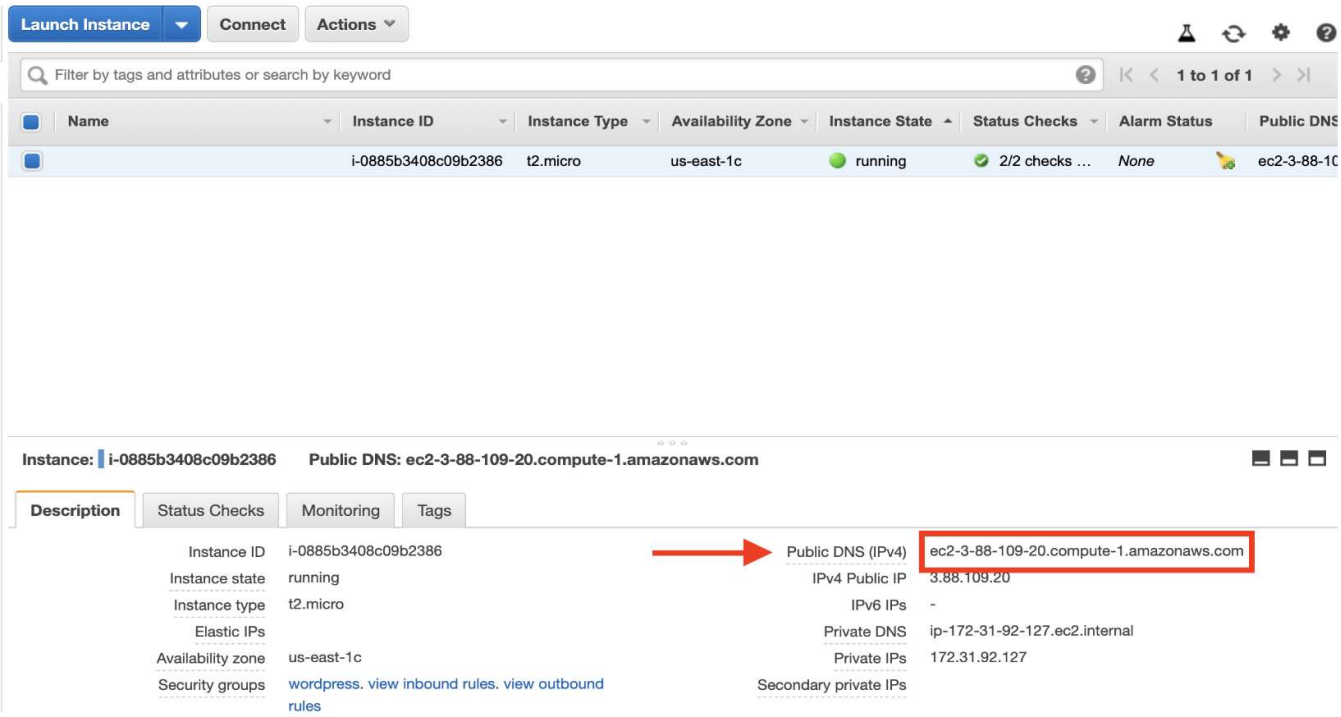
Implementation:

- Installing the Apache Web Server
- To run WordPress, you need to run a web server on your EC2 instance. The open source Apache web server is the most popular web server used with WordPress.
- To install Apache on your EC2 instance, run the following command in your terminal:

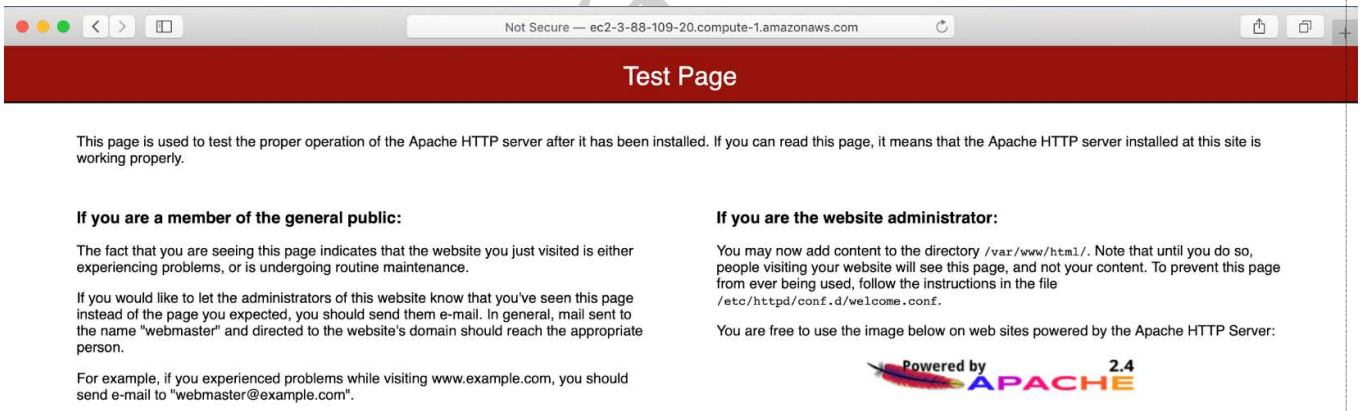
  sudo yum install -y httpd
- You should see some terminal output of the necessary packages being installed.
- To start the Apache web server, run the following command in your terminal:

  sudo service httpd start
- You can see that your Apache web server is working and that your security groups are configured correctly by visiting the public DNS of your EC2 instance in your browser.

- Go to the EC2 Instances page and find your instance. In the Description below, find the Public DNS (IPv4) of your instance.



- Enter this value into your web browser, and you should see an Apache test page.



- Now that your Apache web server is working, it's time to download and configure WordPress.

Download and Configure WordPress

- In this step, you will download the WordPress software and set up the configuration.
- First, download and uncompressing the software by running the following commands in your terminal:

  wget https://wordpress.org/latest.tar.gz
  tar -xzf latest.tar.gz

- If you run "ls" to view the contents of your directory, you will see a tar file and a directory called wordpress with the uncompressed contents.

  ls

  latest.tar.gz wordpress

- Change into the wordpress directory and create a copy of the default config file using the following commands:

  cd wordpress
  cp wp-config-sample.php wp-config.php

- Then, open the wp-config.php file using the vi editor by running the following command.

  vi wp-config.php

- You need to edit two areas of configuration.
- First, edit the database configuration by changing the following lines:

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );

/** MySQL database username */
define( 'DB_USER', 'username_here' );

/** MySQL database password */
define( 'DB_PASSWORD', 'password_here' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

- The values should be:

  DB_NAME: “wordpress”

  DB_USER: The name of the user you created in the database in the previous module

  DB_PASSWORD: The password for the user you created in the previous module.

  DB_HOST: The hostname of the database that you found in the previous module.

- The second configuration section you need to configure is the Authentication Unique Keys and Salts. It looks as follows in the configuration file:

```
/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',         'put your unique phrase here' );
define( 'SECURE_AUTH_KEY',  'put your unique phrase here' );
define( 'LOGGED_IN_KEY',    'put your unique phrase here' );
define( 'NONCE_KEY',        'put your unique phrase here' );
define( 'AUTH_SALT',        'put your unique phrase here' );
define( 'SECURE_AUTH_SALT', 'put your unique phrase here' );
define( 'LOGGED_IN_SALT',   'put your unique phrase here' );
define( 'NONCE_SALT',       'put your unique phrase here' );
```

- Go to this link to generate values for this configuration section. You can replace the entire content in that section with the content from the link.

  https://api.wordpress.org/secret-key/1.1/salt/

- You can save and exit from vi editor by entering Esc then type wq and hit Enter.

- With the configuration updated, you are almost ready to deploy your WordPress site. In the next step, you will make your WordPress site live.

Deploying WordPress Site

- In this step, you will make your Apache web server handle requests for WordPress.
- First, install the application dependencies you need for WordPress. In your terminal, run the following command.

  sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

- Second, change to the proper directory by running the following command:

  cd /home/ec2-user

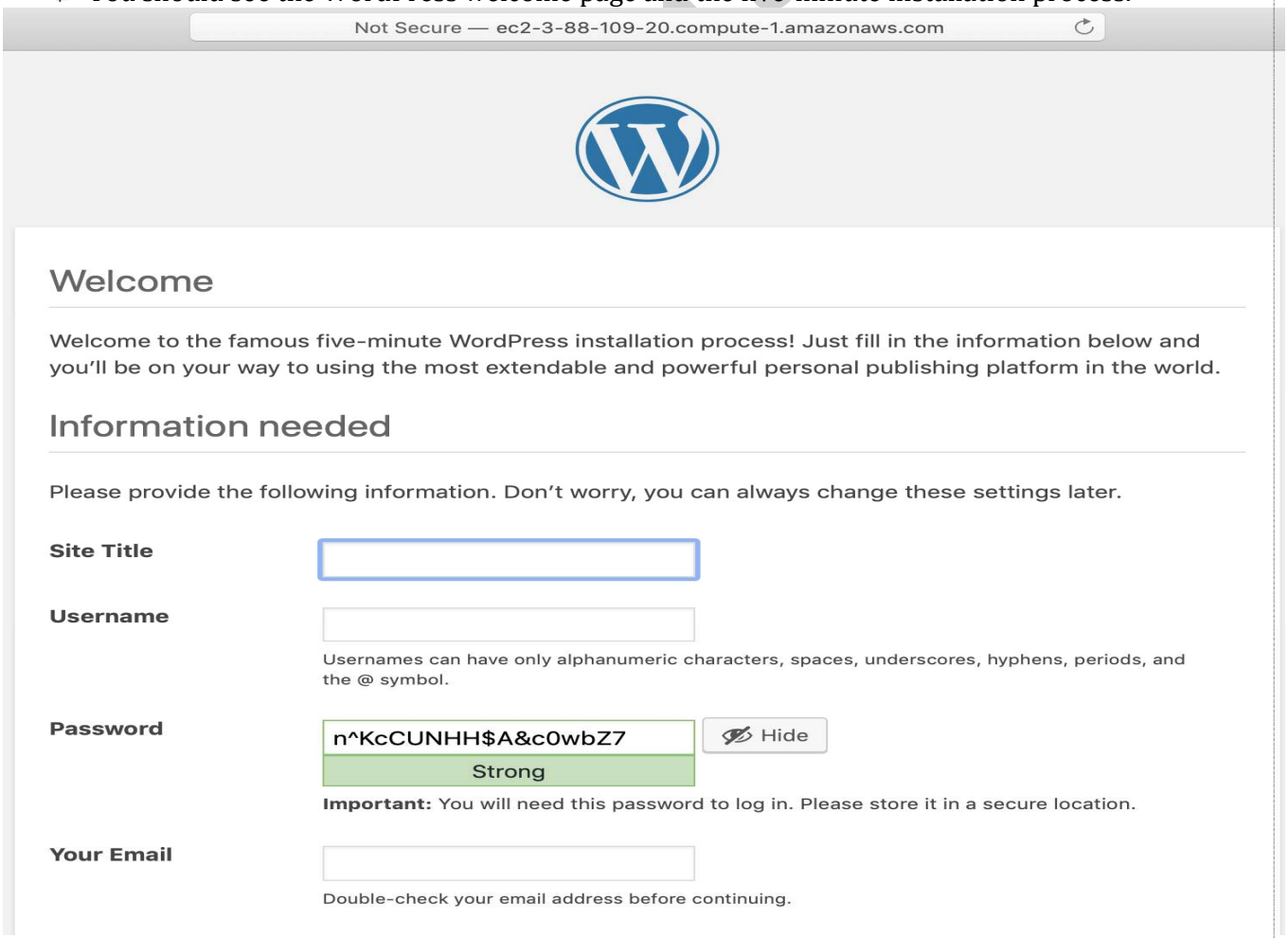
- Then, copy your WordPress application files into the /var/www/html directory used by Apache.

  sudo cp -r wordpress/* /var/www/html/

- Finally, restart the Apache web server to pick up the changes.

  sudo service httpd restart

- You should see the WordPress welcome page and the five-minute installation process.



- That's it. You have a live, publicly-accessible WordPress installation using a fully-managed MySQL database on Amazon RDS.

- In the next module, you will clean up your resources and see some next steps for your WordPress installation.

Lab – 5: Explore our New Website and do cleanup activity

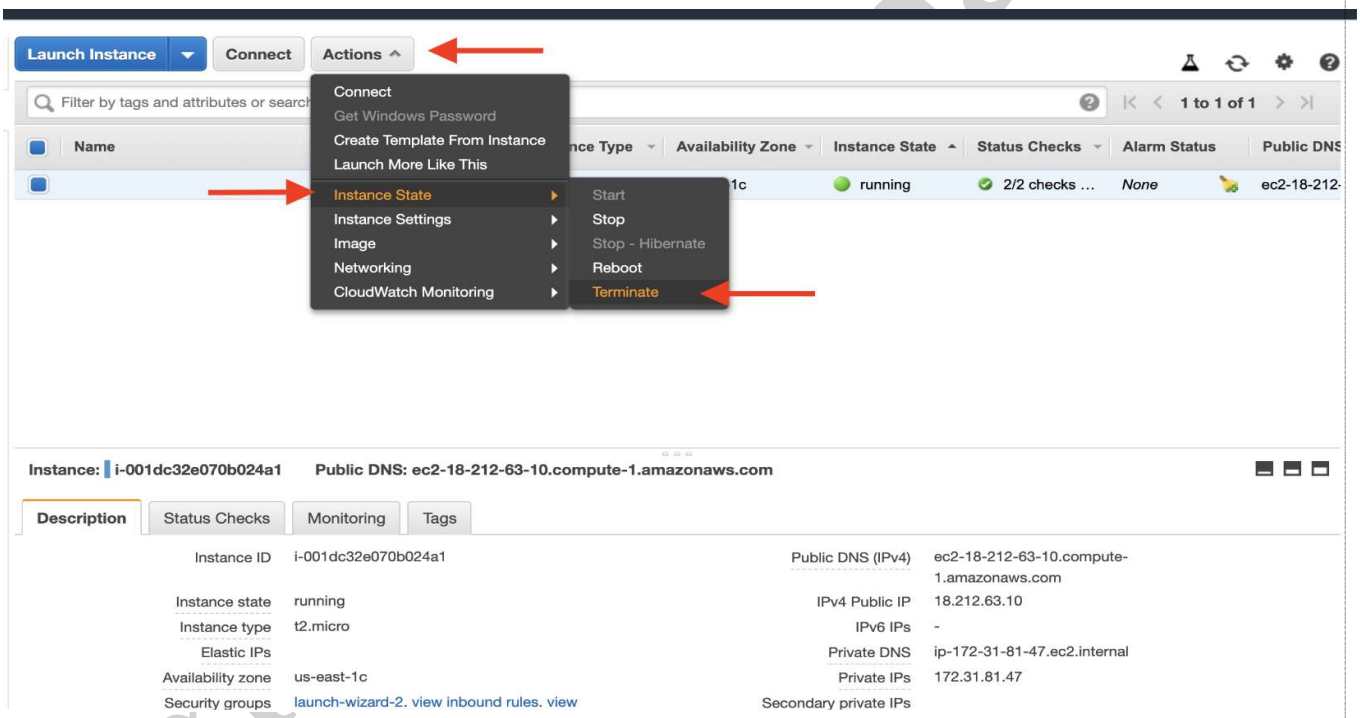
- In this lab, you learned how to set up a WordPress site on Amazon EC2.
- As part of this, you used a MySQL database from Amazon RDS. Amazon RDS provides fully-managed database options so you can focus on the things that matter for your business.
- Now it's time for the fun part -- go play with your new site. Configure the design, add pages and posts, and start getting users to your site.
- There are a number of ways to improve your WordPress installation, such as:
  a. Connecting a domain name to your WordPress site with Amazon Route53;
  b. Using Elastic Load Balancers and Amazon CloudFront to scale WordPress;
- For additional information, check out the whitepaper on WordPress Best Practices on AWS.

Note: If you ever stop and start your WordPress EC2 instance, use an Elastic Load Balancer in front of your EC2 instance, or associate a domain name with your site, you may need to update some WordPress configuration settings. Please see WordPress's reference documentation on changing the site URL for more information.

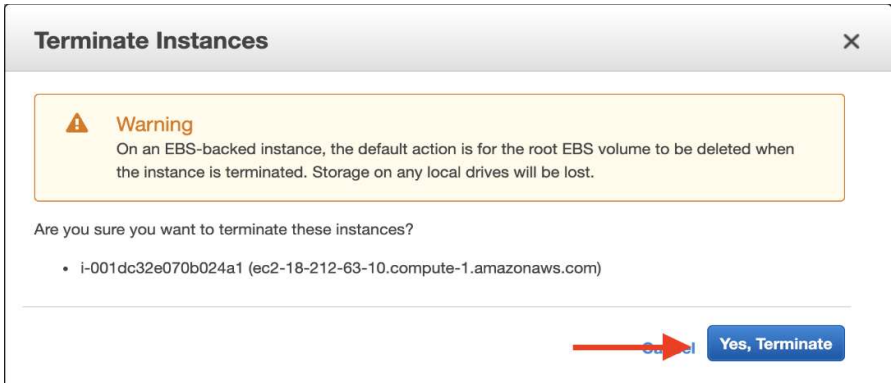
- In the steps below, you will clean up the resources created in this lab so that you will not be charged.
- After completing of all lab scenarios this is main task.
- Clean up the resources created in the lab:
  a. Remove your EC2 Instance
  b. Delete your RDS Instance

Remove your EC2 Instance:

- First, you can remove your EC2 instance that is hosting your WordPress site.
- Navigate to the EC2 Instances page and find your EC2 instance. Select it, and click the Actions button.
- Navigate to Instance State > Terminate and click it.

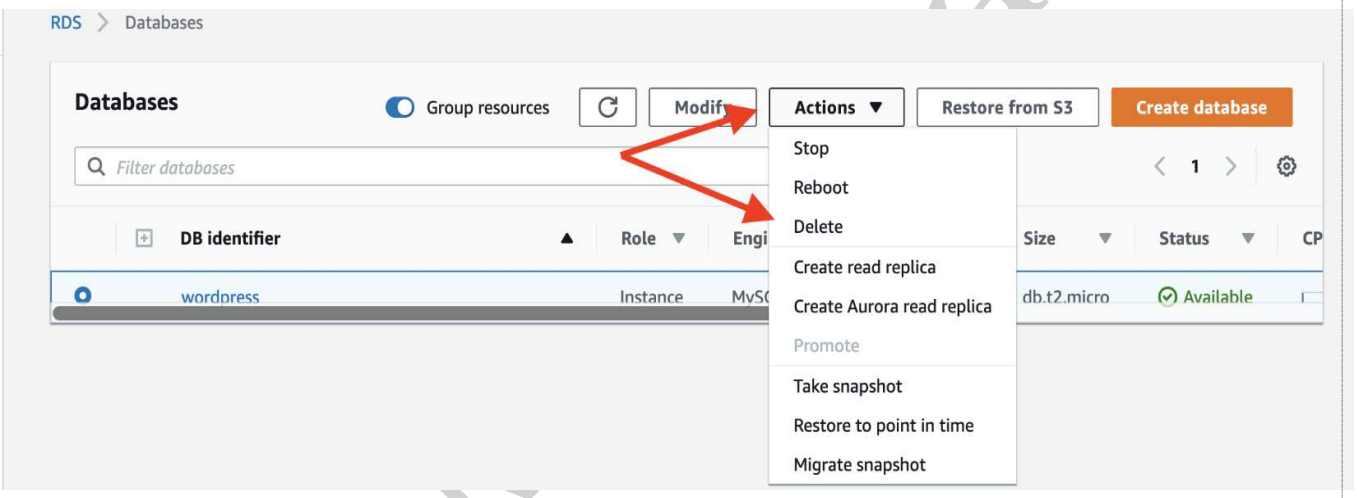


- Click Yes, terminate to terminate your instance.

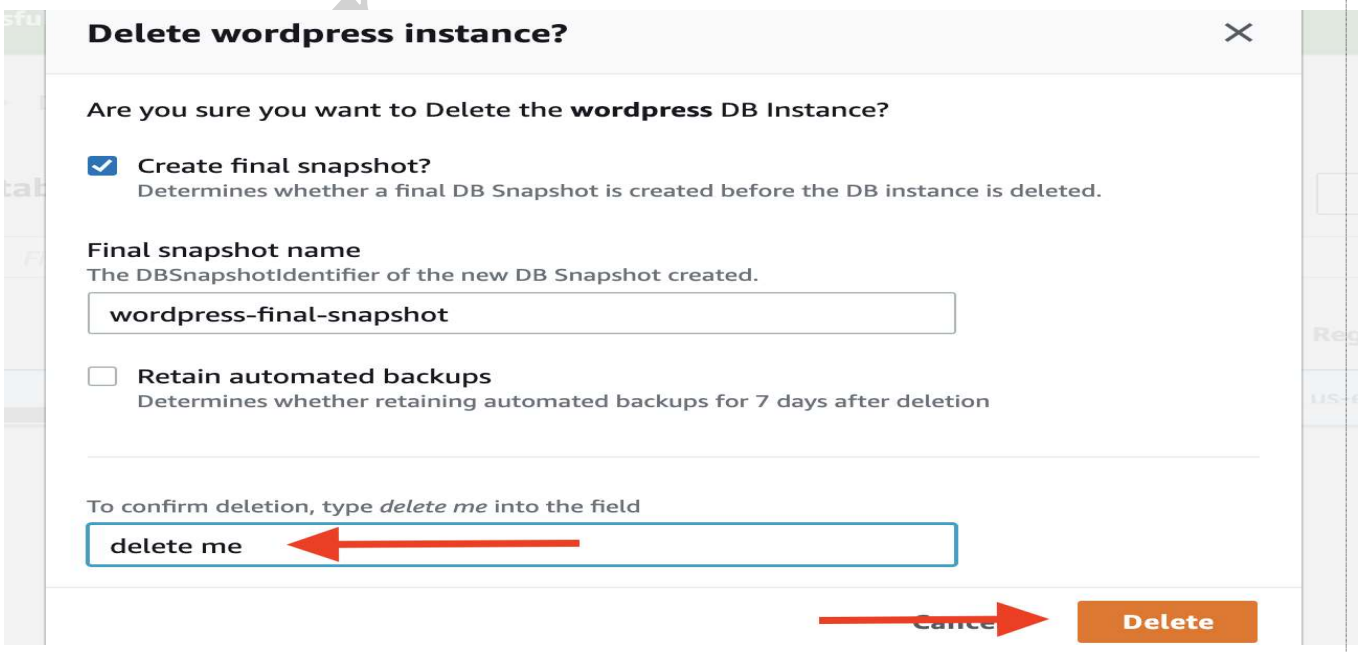


Delete your RDS Instance

- To delete your RDS instance, navigate to the [RDS Databases page] in the AWS console. Find your WordPress RDS instance and select it. Then, click Actions and Delete.



- Type “delete me” into the confirmation box and click the Delete button to finalize the process.



- You successfully set up a WordPress site to run a blog!

Now it's time for the fun part -- go play with your new site. Configure the design, add pages and posts, and start getting users to your site.