Jamie Gonzales Esguerra (9) SACF PROJECT REPORT PA12

# What is Wordpress

Wordpress is a content management system (CMS) and uses either MySQL or MariaDB database. I was able to create a website using Wordpress. Relevant AWS instances were used to create my Wordpress.

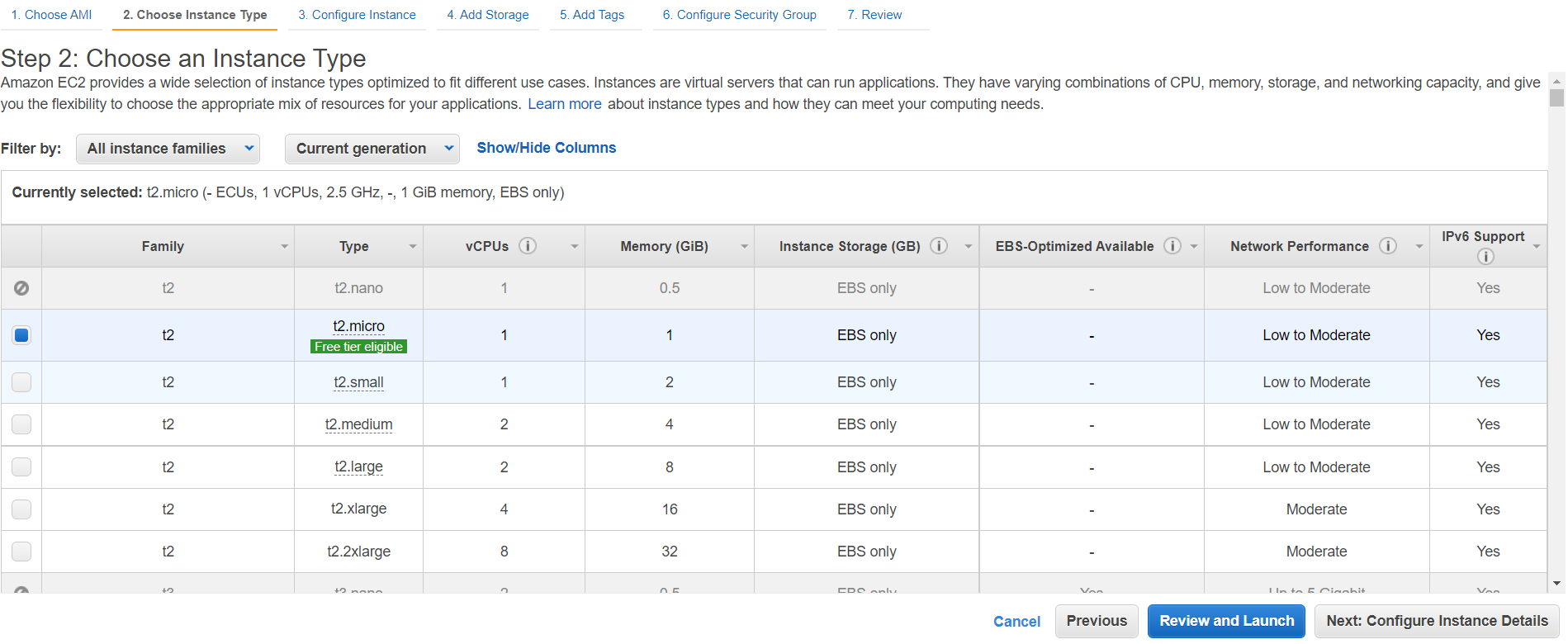
# Screenshots of my installation

**EC2**

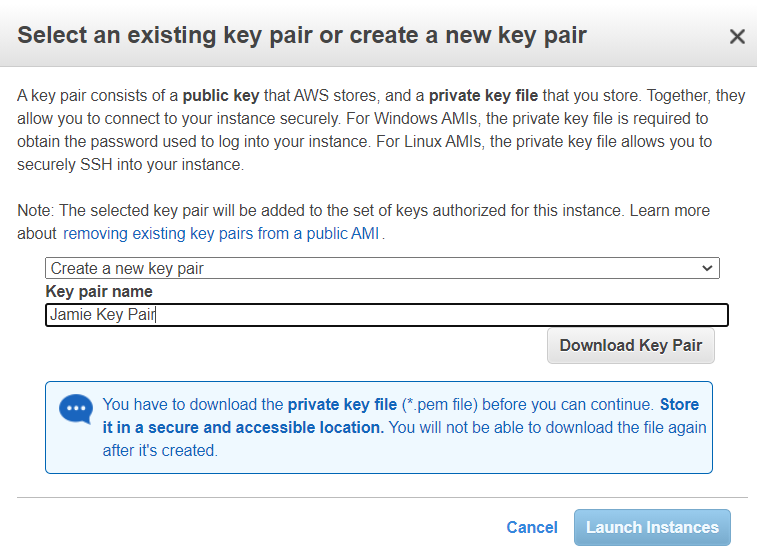
1. For AMI, I selected “**Red Hat Enterprise Linux 8**”



1. I created an EC2 with an Instance type of **t2.micro**.

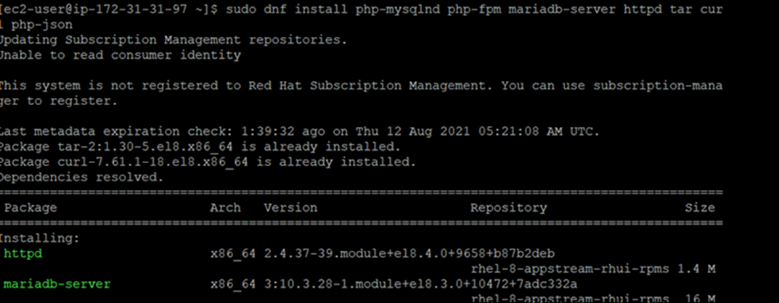


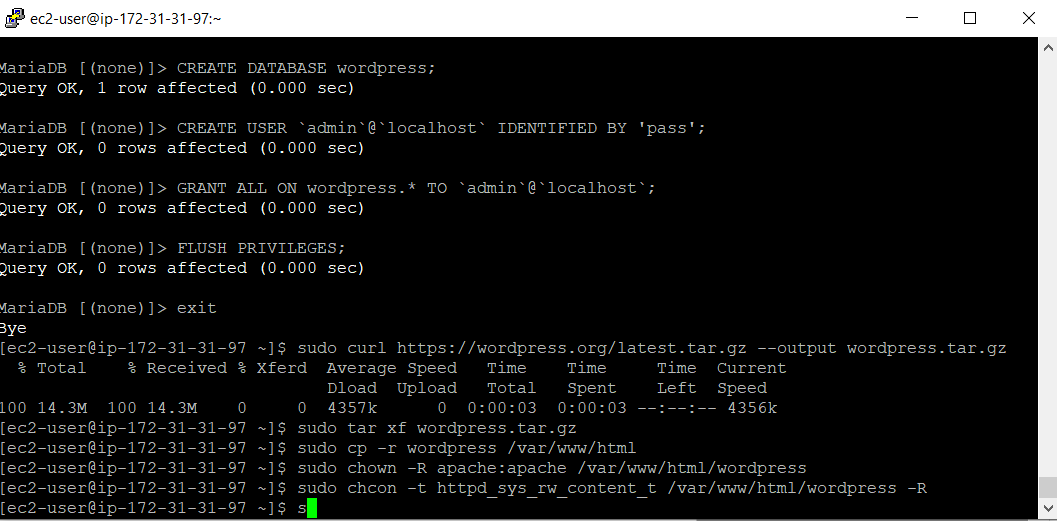
1. For my Configure Instance Details, I created a subnet in us-east-1a.
2. In the tag session, **“key: Name”** and **“Value: Jamie’s Website:”**
3. For Security Group “**Type: HTTP”** and **“Source: Anywhere”**
4. While creating my EC2, I created a new key pair to access my ec2 on putty. **“Key pair name:Jamie Key Pair”**



**Putty and Puttygen**

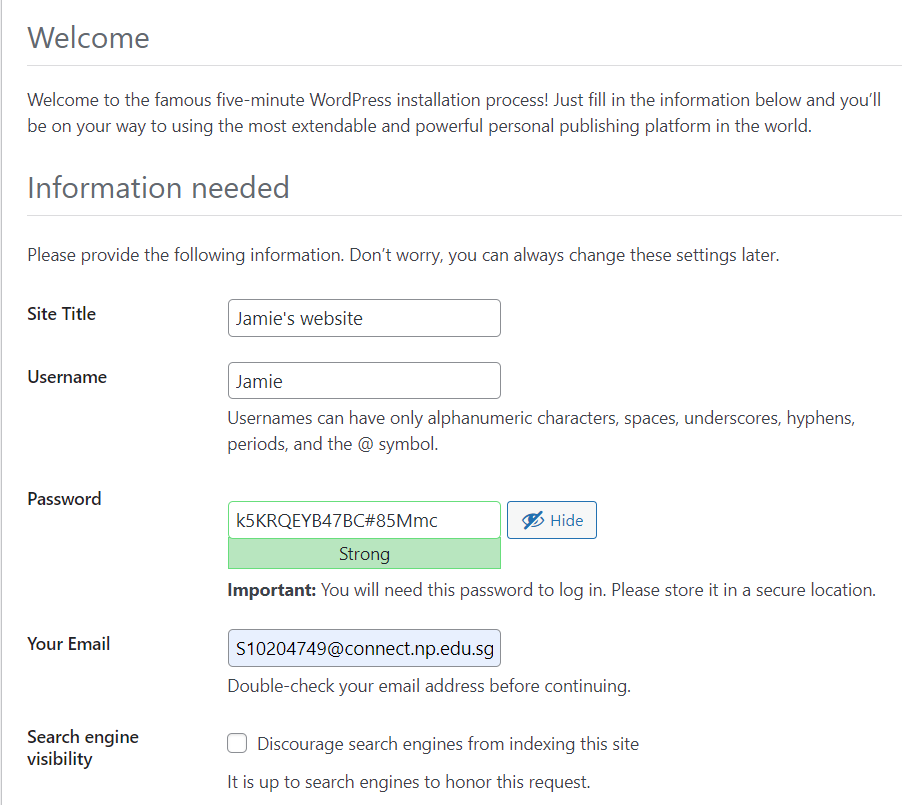
1. Download putty and puttygen
2. Use puttygen to convert “JamieKeyPair.pem” to “JamieKeyPair.ppk”
3. After converting “JamieKeyPair.pem” to “JamieKeyPair.ppk”, I can then now use this for my putty’s private key.
4. For the **“HostName”** , I paste the Public IPv4 address of my EC2 which is **75.101.155.191**
5. Hence I can then now use putty to download apache web server, httpd and create my wordpress database. The commands I used are :
   1. Sudo yum install php-mysqlnd php-fpm mariadb-server httpd tar curl php-json
   2. Sudo systemctl start mariadb
   3. Sudo systemctl start httpd
   4. Sudo systemctl enable mariadb
   5. Sudo systemctl enable httpd
   6. mysql\_secure\_installation
   7. mysql -u root -p
   8. CREATE DATABASE wordpress;
   9. CREATE USER `admin`@`localhost` IDENTIFIED BY 'pass';
   10. GRANT ALL ON wordpress.\* TO `admin`@`localhost`;
   11. FLUSH PRIVILEGES;
   12. Exit
   13. Sudo curl [https://wordpress.org/latest.tar.gz --output wordpress.tar.gz](https://wordpress.org/latest.tar.gz%20--output%20wordpress.tar.gz)
   14. tar xf wordpress.tar.gz
   15. sudo cp -r wordpress /var/www/html
   16. sudo chown -R apache:apache /var/www/html/wordpress
   17. sudo chcon -t httpd\_sys\_rw\_content\_t /var/www/html/wordpress -R

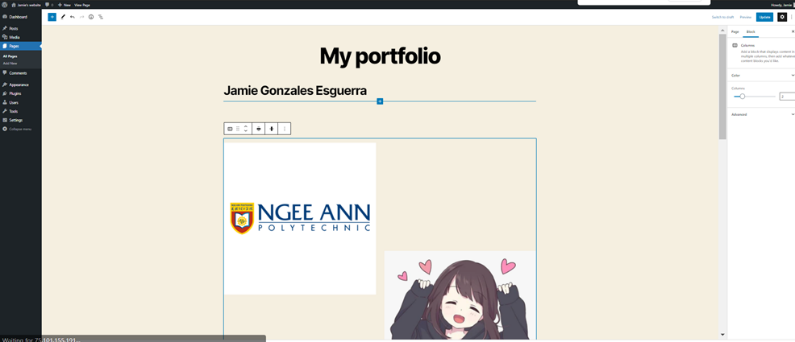




I can now access my wordpress website by using the public IPv4 address : http:// **75.101.155.191**/wordpress/

**Wordpress**

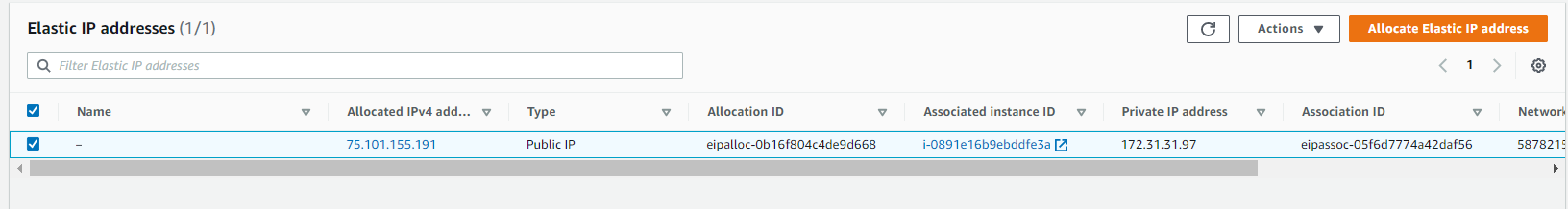
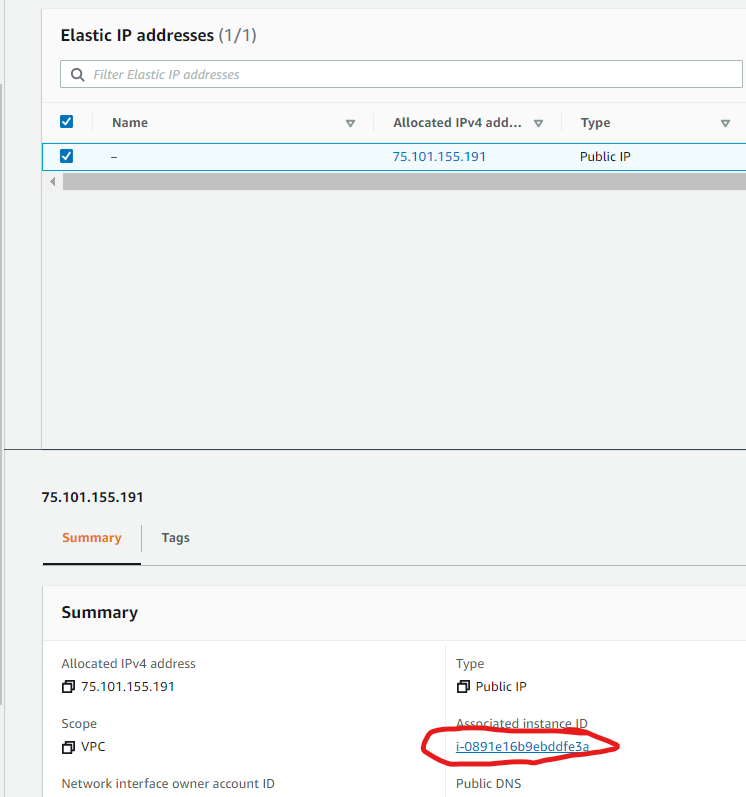
1. I set up the **Site Title, username, password and email as follows:**
2. Hence, I can now then modify my website. As an example, I made my website look as if it was a portfolio

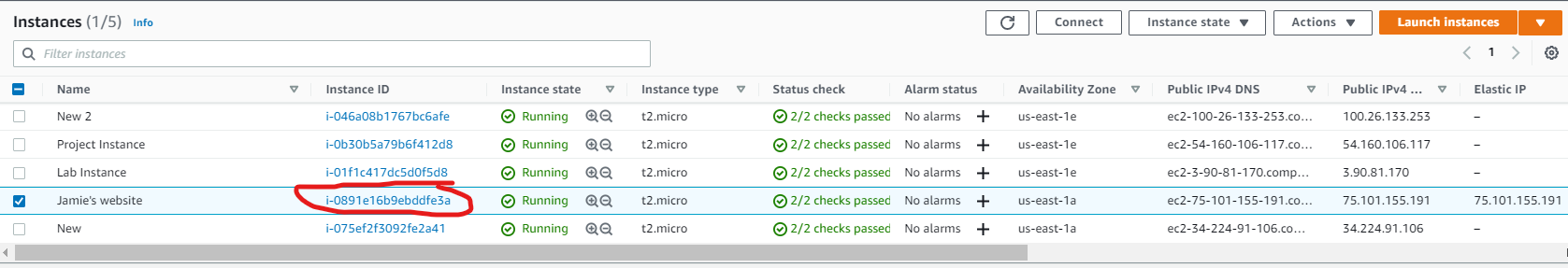
****

# Screenshots of enhancements and explanation why I added these enhancements

**Elastic IP address**

I created this is so that when you restart the instance, it will restart with the same ip address. Hence, it is easier for me to go to my website as it would have the same public ipv4 address

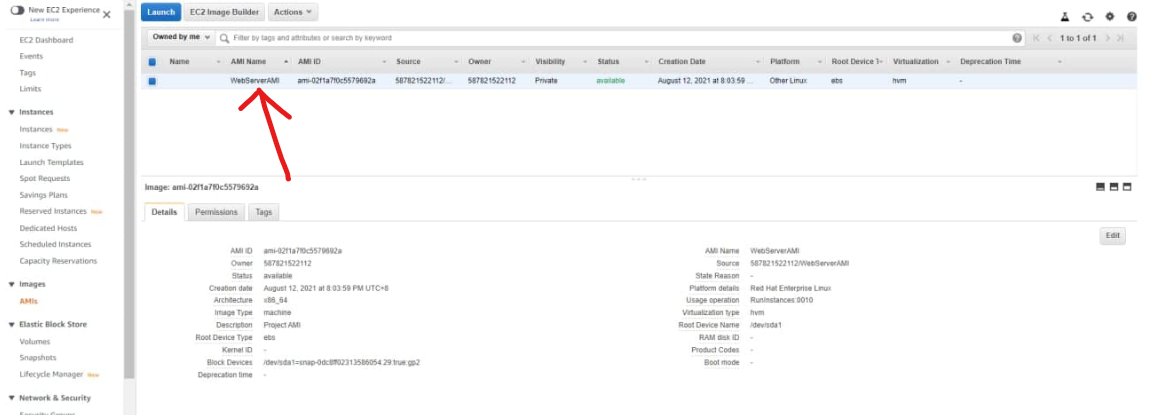
1. Click “Allocate Elastic IP address”
2. Publick IPv4 address pool : Amazon’s pool of IPv4 address
3. 
4. Under the elastic ip address that I created, I can see the Associated Instance ID of my ec2 



**AMI**

This AMI is used when launching the Auto Scaling group. This is for me to have more instances with the same configuration.

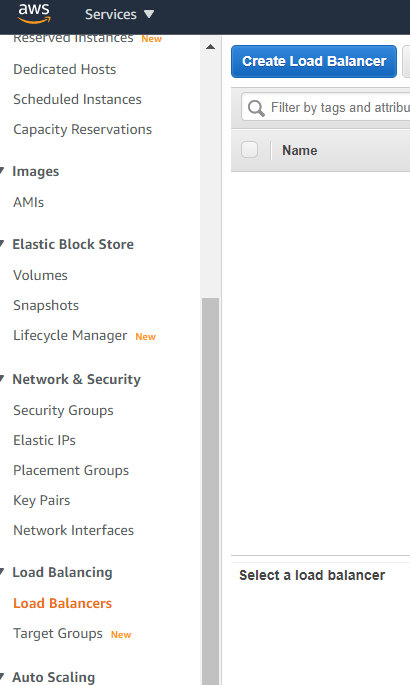
1. Image name : WebServer AMI
2. Image description : Project AMI
3. Click “Create image”



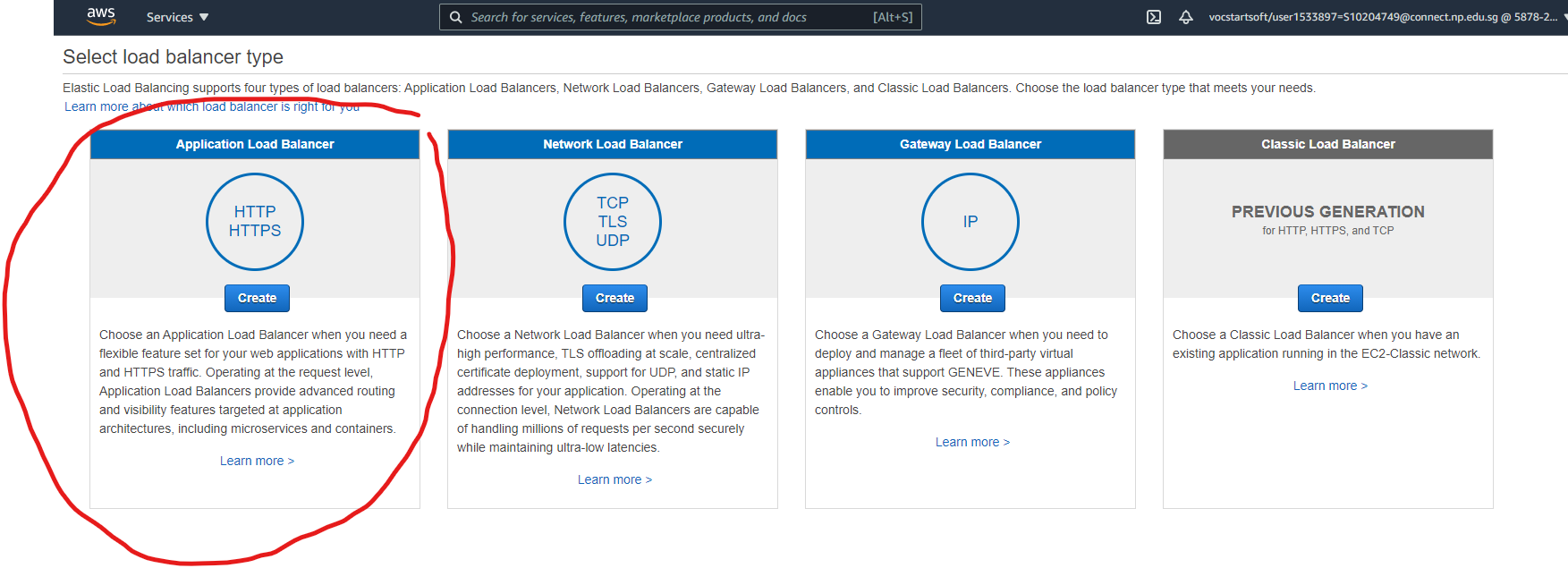
**Load balancer**

When the load balancer identifies an unhealthy target, it redirects traffic to a healthy target. When it determines that the target is healthy again, it resumes directing traffic to it. Hence it enables be to have fault tolerance.

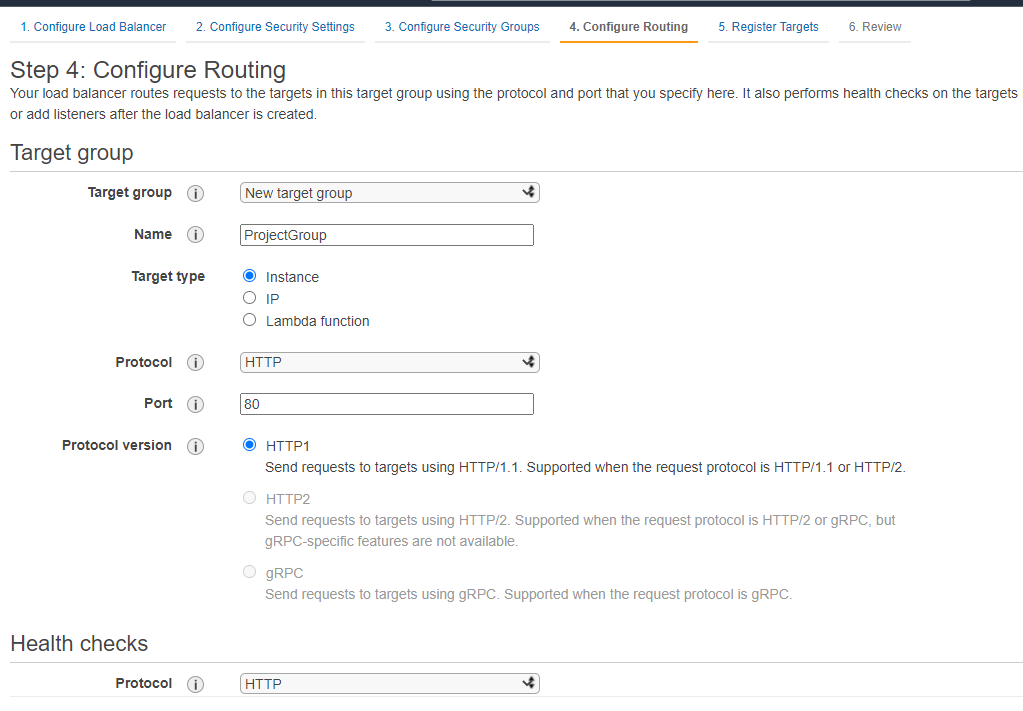
1. In the left navigation pane, click **Load balancer**.
2. Click **Create Load Balancer**

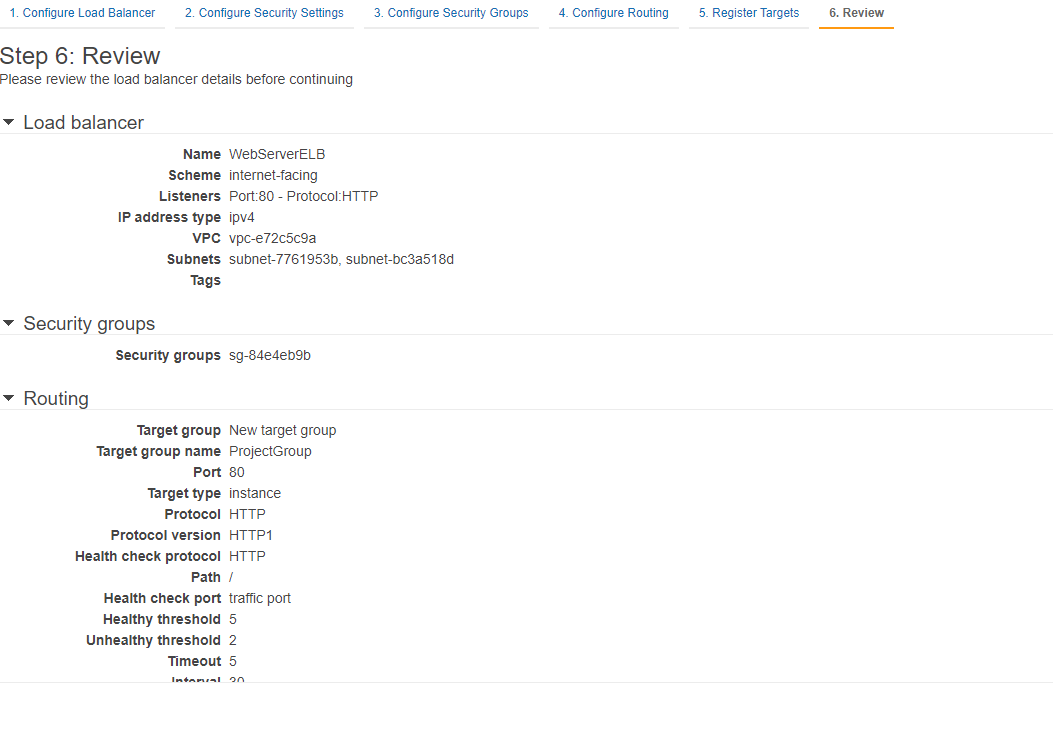


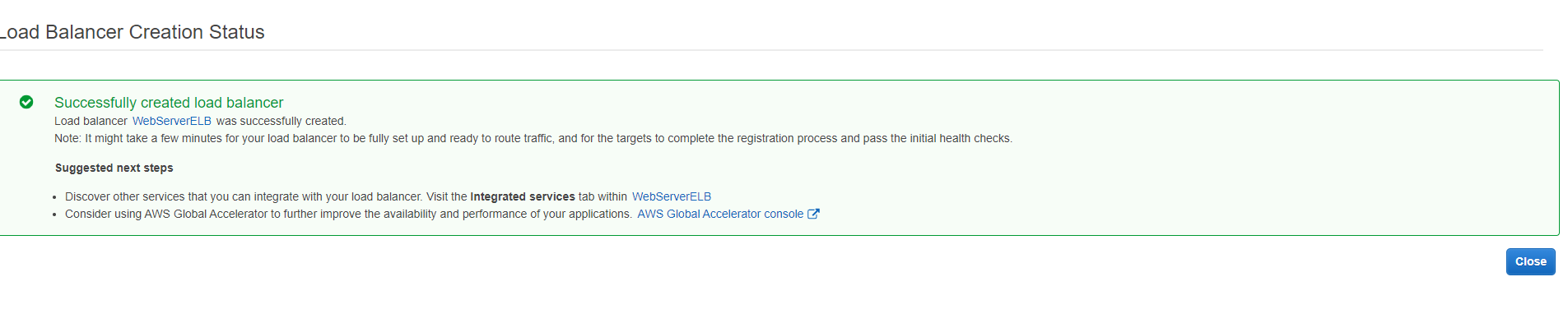
1. As the Load Balancer I chose **“HTTP HTTPS”**

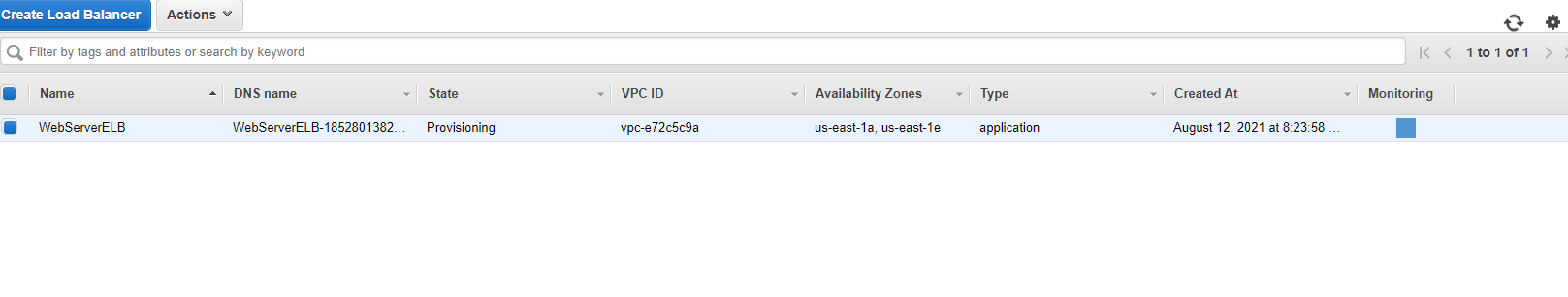


1. I put the name as “WebServerELB”.
2. The load balancer protocol as HTTP.
3. For Availability Zone, I selected us-east-1a and us-east-1e.
4. I used the default VPC security group
5. I configured my routing as follows:





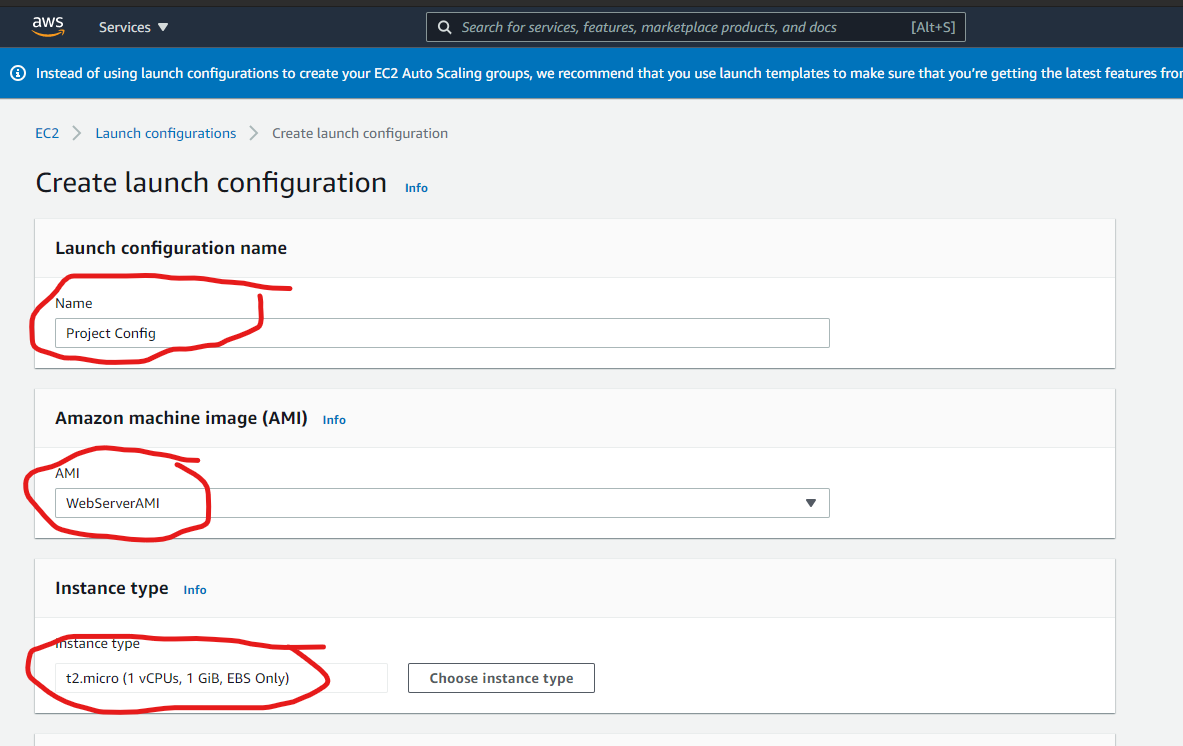


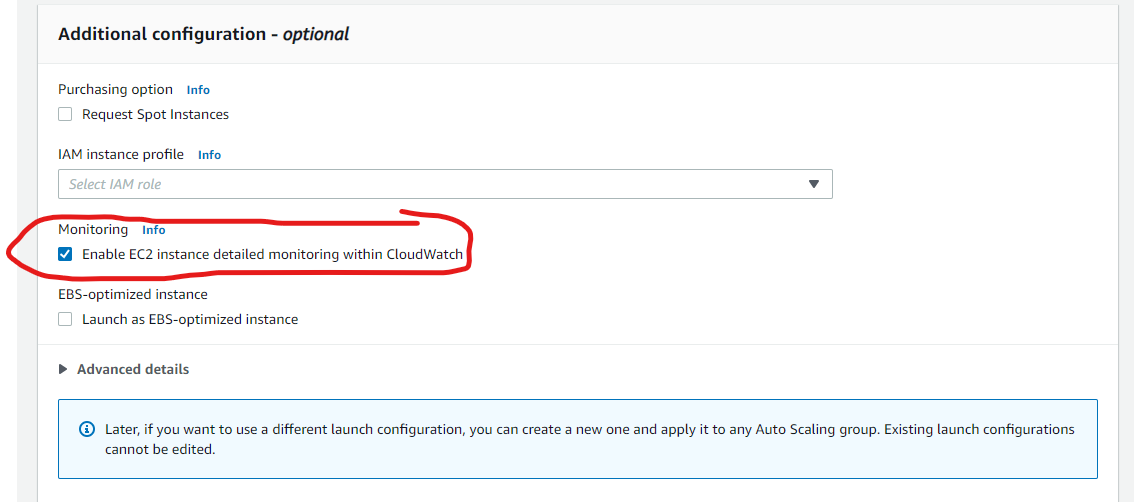
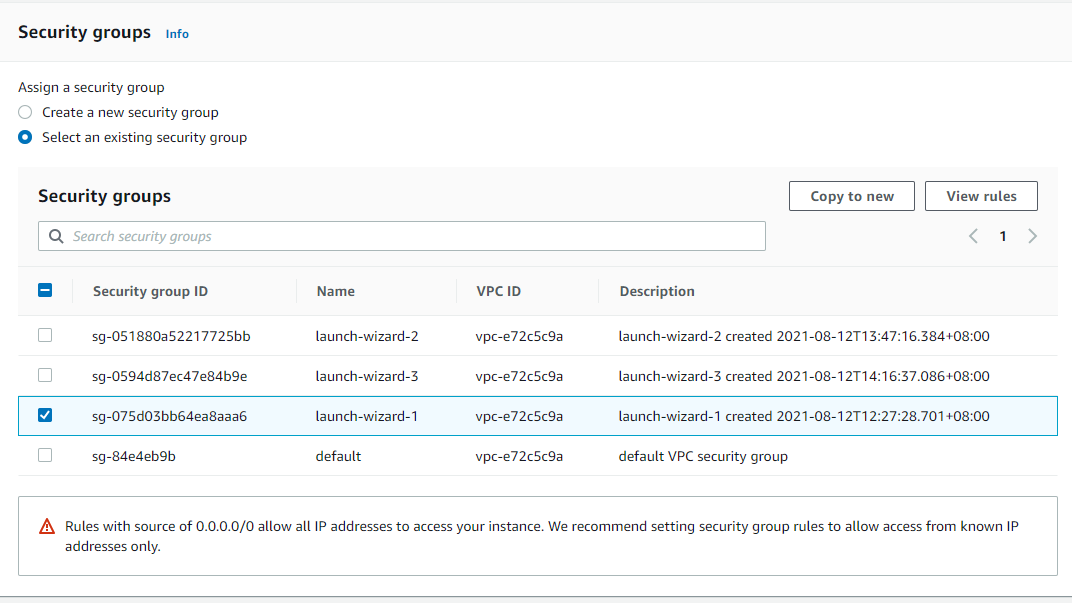


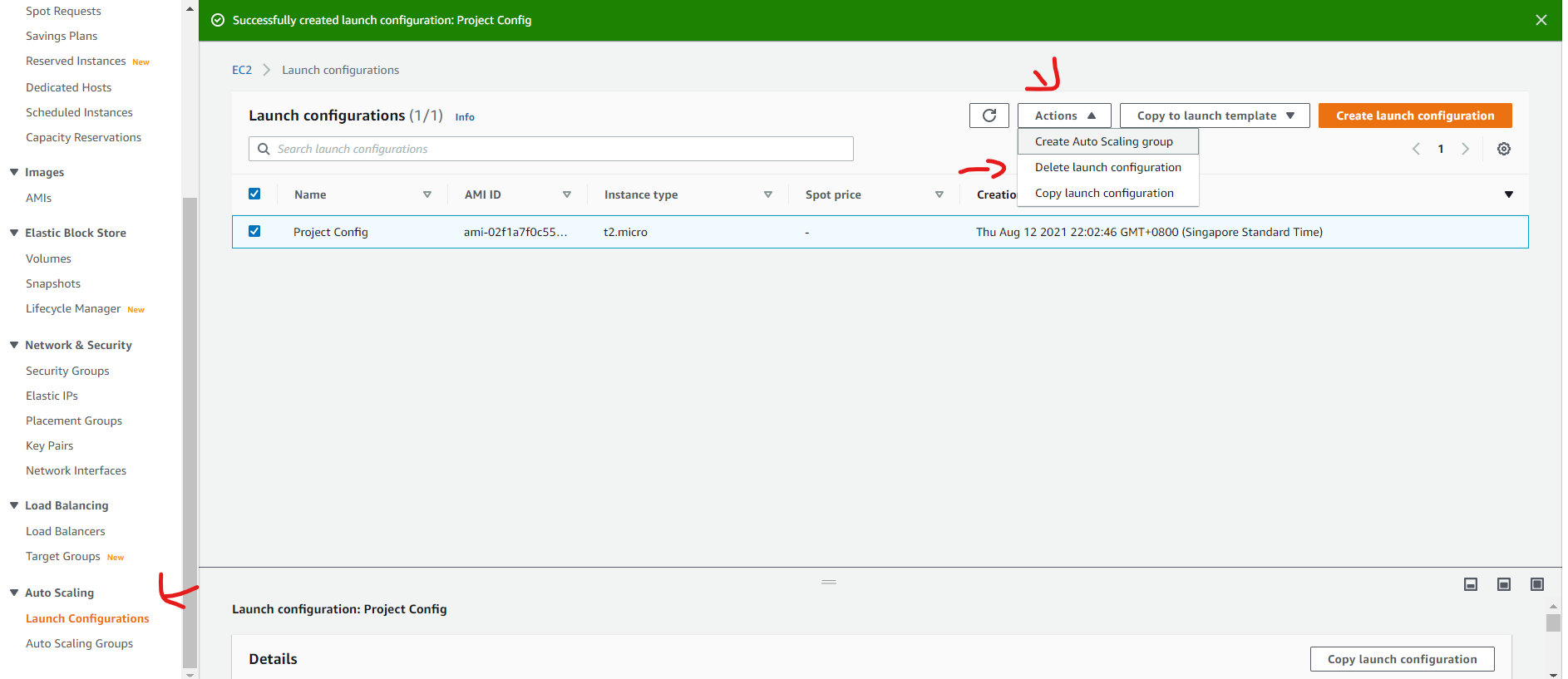
**AUTO-SCALING GROUP**

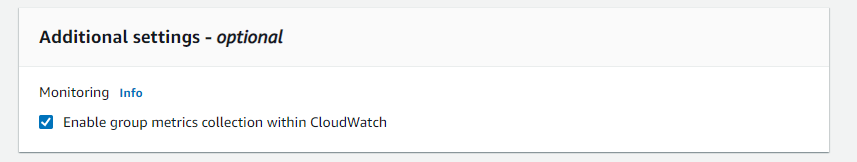
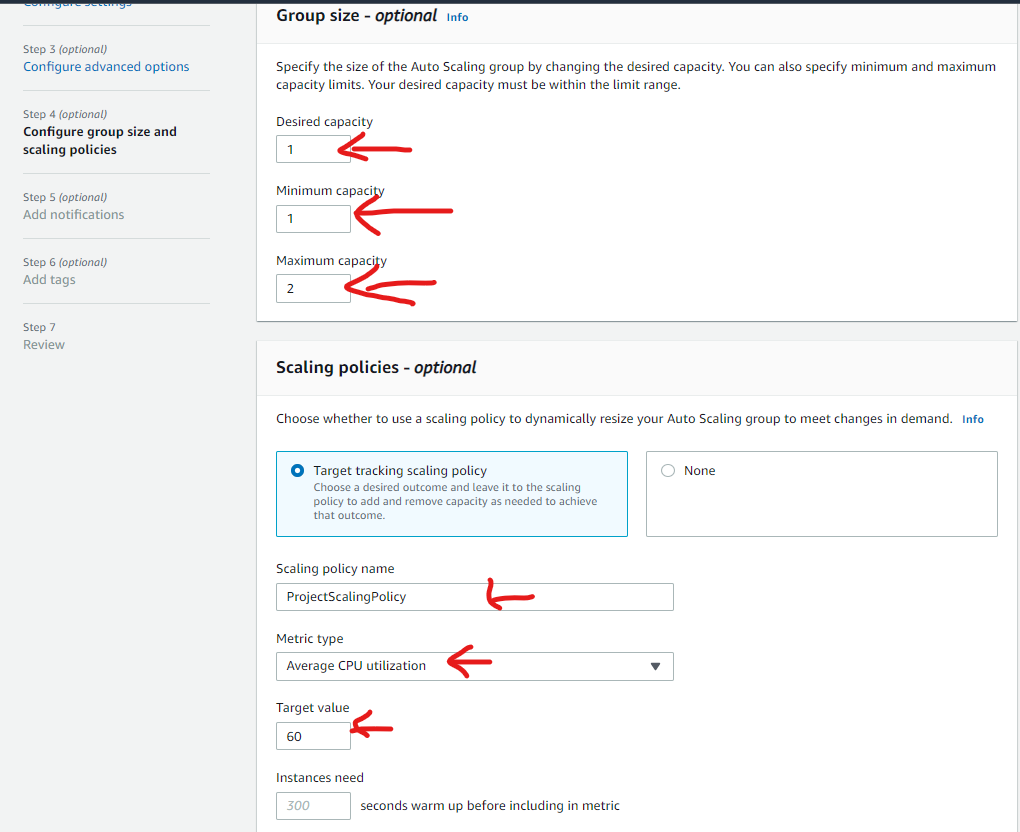
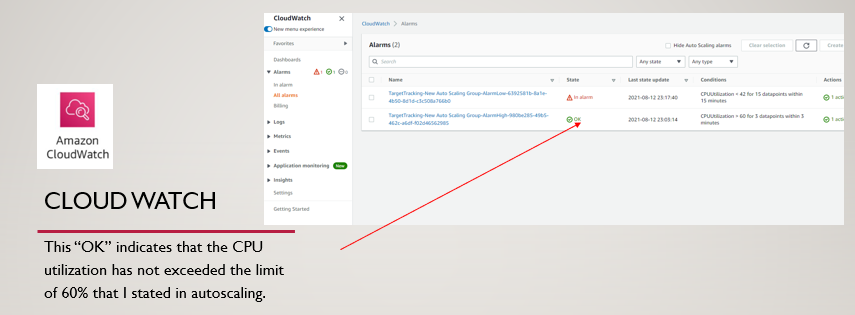
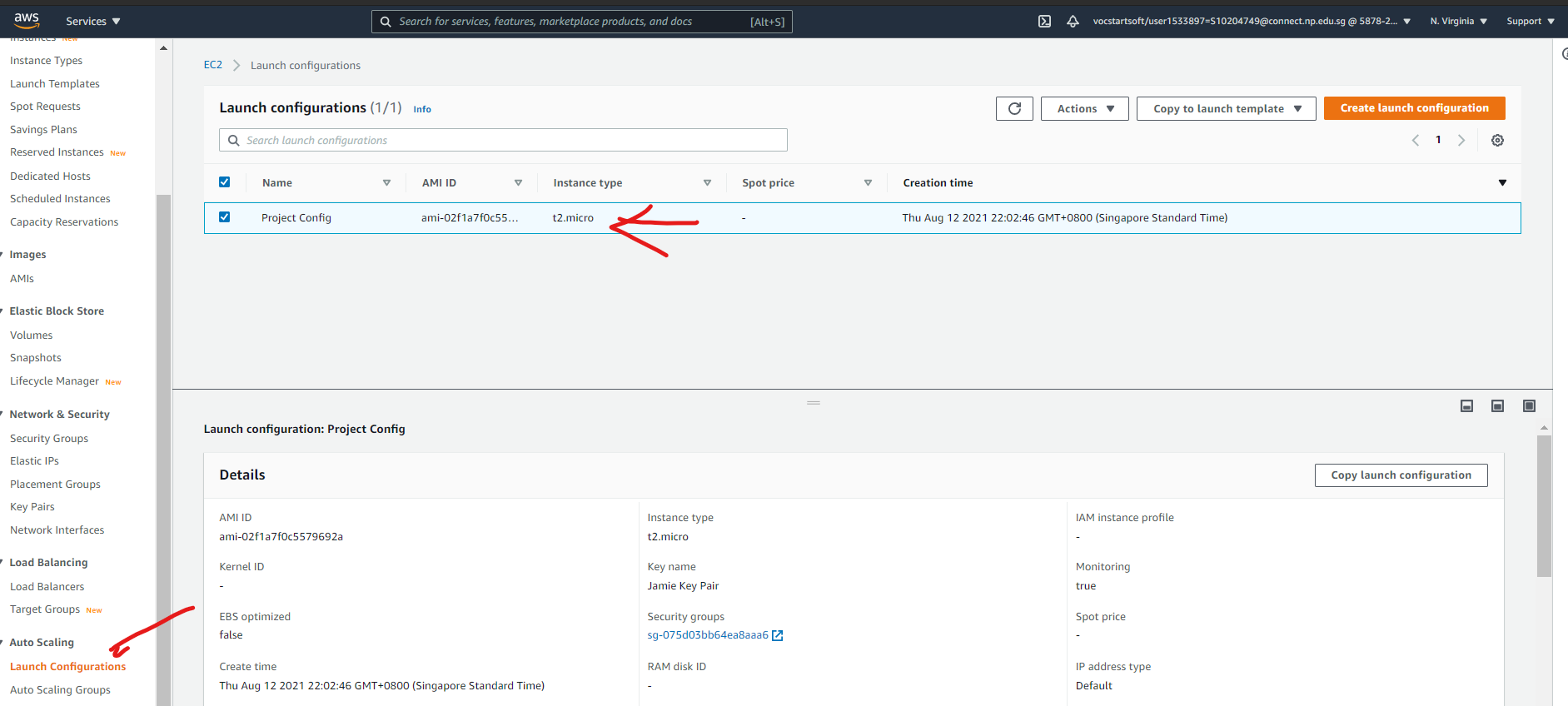
Amazon EC2 Auto Scaling adapts to changing conditions by adding or terminating instances, launching instances from an AMI, and ensuring that a minimum number of Amazon EC2 instances are operating.

1. In the left navigation pane, click **Launch Configurations.**
2. Click **“Create** **Launch configuration”**
3. I set up the **name, AMI and Instance type** as follows:



1. I selected “Enable EC2 instance detailed monitoring within CloudWatch” for auto scaling to react quickly to changing utilization
2. Security group I used “launch-wizard-1”
3. I used an existing key pair which is “JamieKeyPair.pem”.
4. Under Launch configuration, I created an auto scaling group by clicking the following:



1. I set the name as “Project Auto Scaling Group”
2. I chose the subnets: “us-east-1a” and “us-east-1e”
3. I attached an existing load balancer which is the “Project Group|HTTP”This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns
4. I set the desired capacity, minimum capacity dn maximum capacity as follows:This tells Auto Scaling to maintain an average CPU untilization across all instances 60%. Auto scaling will automatically add or remove capacity as required to keep the metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern
5. 
6. For the tags : “Key: Name” and “Value: Project Instance”

# Difficulties I encountered

One if the difficulties I encountered is that I couldn’t get to my ec2 instance using the command promt. As an alternative, I used decided to use putty. While trying to connect with my putty I realized that my key is in the wrong format: “JamieKeyPair.pem”. Therefore, I downloaded puttygen to convert the format from “.pem” to “.ppk” for it to be suitable to be used in putty.

Furthermore, I didn’t know how to edit my Wordpress website. I was unsure of how to change the words and pictures that the wordpress automatically provides. After navigating Wordpress and learning, I was able to create a nice portfolio in the end.

# Reflection

Throughout this module I learnt how to use the necessary services to create a website and use database. I gained resilience by finding new ways to solve the issues that I encountered. I realized that one of the important concepts I needed when working with AWS is that there would always be a way for my instances to not work properly. Hence, I had to learn how to use other services to tackle these problems. Using new services such as the Auto-scaling group, Load balancer and cloud watch. After setting up the load balancer, I found a way to check if it was working by using its DNS name. By knowing if my services are working properly after creating, I become more confident in creating services in AWS. Furthermore leaning Linux commands have been a big help as I had to use the skills to download HTTP and Apache Web Server. While working through my way in AWS, I noticed that I also had to keep track of my spending as I was limited with $50. Therefore, I became more careful when creating new instances and made sure that I stop my instances when I am not using them. To sum up, this project have helped me the most in understanding what is cloud computing and AWS in general.

