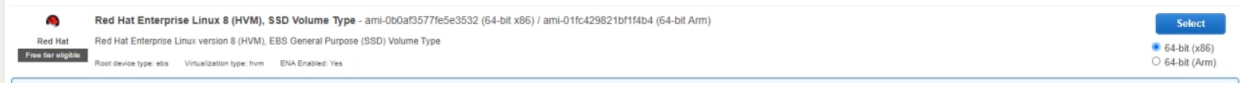
What is Wordpress

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**"



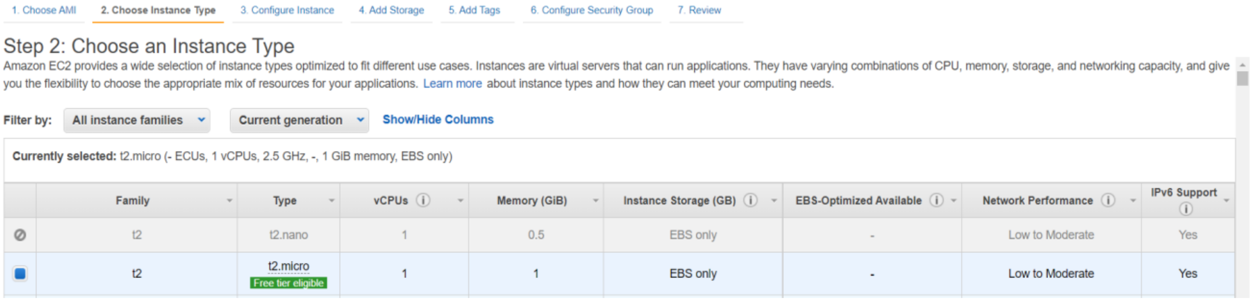
Red Hat
Free tier eligible

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0b0af3577fe5e3532 (64-bit x86) / ami-01fc429821bf1f4b4 (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

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



Step 2: Choose an Instance Type

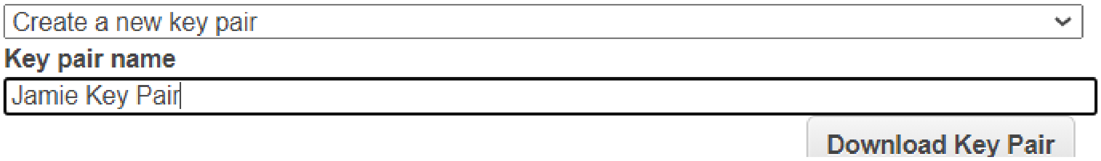
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
⊘	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

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



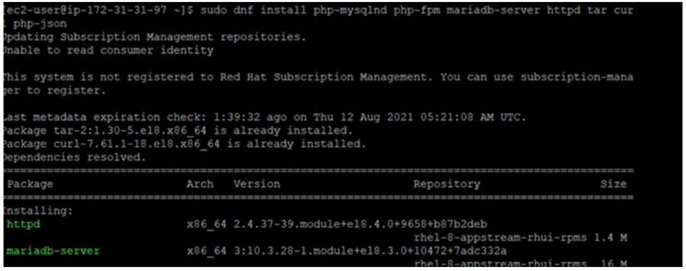
Create a new key pair

Key pair name

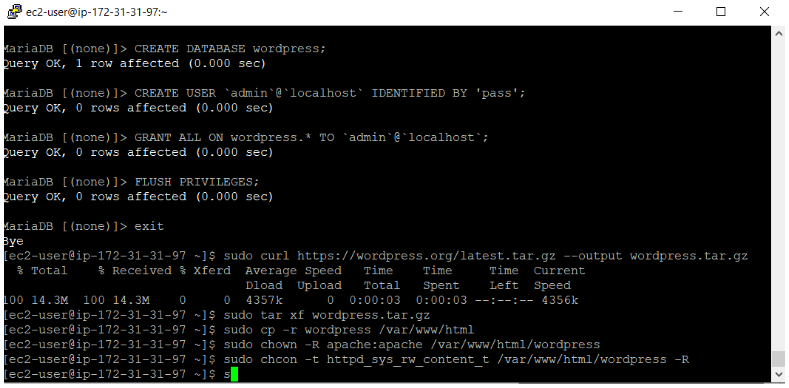
Jamie Key Pair

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 :
    a. Sudo yum install php-mysqlnd php-fpm mariadb-server httpd tar curl php-json
    b. Sudo systemctl start mariadb
    c. Sudo systemctl start httpd
    d. Sudo systemctl enable mariadb
    e. Sudo systemctl enable httpd
    f. mysql_secure_installation
    g. mysql -u root -p
    h. CREATE DATABASE wordpress;
    i. CREATE USER \`admin\`@\`localhost\` IDENTIFIED BY 'pass';
    j. GRANT ALL ON wordpress.* TO \`admin\`@\`localhost\`;
    k. FLUSH PRIVILEGES;
    l. Exit
    m. Sudo curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
    n. tar xf wordpress.tar.gz
    o. sudo cp -r wordpress /var/www/html
    p. sudo chown -R apache:apache /var/www/html/wordpress
    q. sudo chcon -t httpd_sys_rw_content_t /var/www/html/wordpress -R



```
[ec2-user@ip-172-31-31-97 ~]$ sudo dnf install php-mysqlnd php-fpm mariadb-server httpd tar cur
l php-json
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered to Red Hat Subscription Management. You can use subscription-mana
ger to register.

Last metadata expiration check: 1:39:32 ago on Thu 12 Aug 2021 05:21:08 AM UTC.
Package tar-2:1.30-5.el8.x86_64 is already installed.
Package curl-7.61.1-18.el8.x86_64 is already installed.
Dependencies resolved.
================================================================================================
 Package                   Arch    Version                         Repository              Size
================================================================================================
Installing:
 httpd                     x86_64  2.4.37-39.module+el8.4.0+9658+b87b2deb
                                                                   rhel-8-appstream-rhui-rpms 1.4 M
 mariadb-server            x86_64  3:10.3.28-1.module+el8.3.0+10472+7adc332a
                                                                   rhel-8-appstream-rhui-rpms  16 M
```

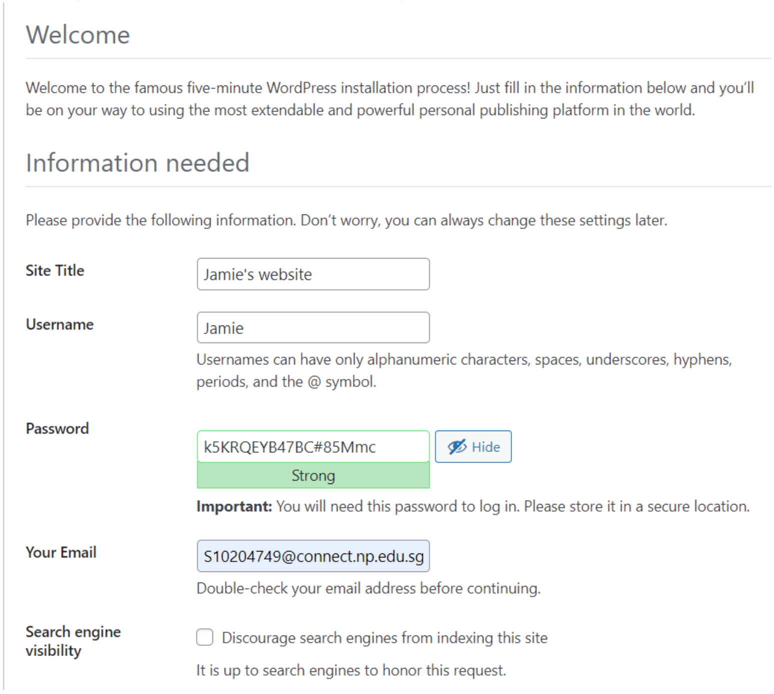


```
MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER `admin`@`localhost` IDENTIFIED BY 'pass';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL ON wordpress.* TO `admin`@`localhost`;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit
Bye
[ec2-user@ip-172-31-31-97 ~]$ sudo curl https://wordpress.org/latest.tar.gz --output wordpress.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14.3M  100 14.3M    0     0  4357k      0  0:00:03  0:00:03 --:--:-- 4356k
[ec2-user@ip-172-31-31-97 ~]$ sudo tar xf wordpress.tar.gz
[ec2-user@ip-172-31-31-97 ~]$ sudo cp -r wordpress /var/www/html
[ec2-user@ip-172-31-31-97 ~]$ sudo chown -R apache:apache /var/www/html/wordpress
[ec2-user@ip-172-31-31-97 ~]$ sudo chcon -t httpd_sys_rw_content_t /var/www/html/wordpress -R
[ec2-user@ip-172-31-31-97 ~]$ s
```

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:**



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title: Jamie's website

Username: Jamie

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: k5KRQEYB47BC#85Mmc Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

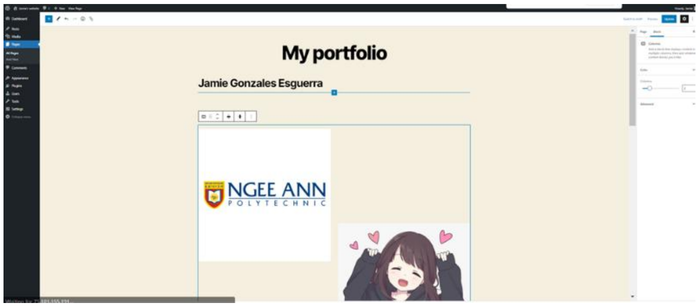
Your Email: S10204749@connect.np.edu.sg

Double-check your email address before continuing.

Search engine visibility: ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

2) Hence, I can now then modify my website. As an example, I made my website look as if it was a portfolio

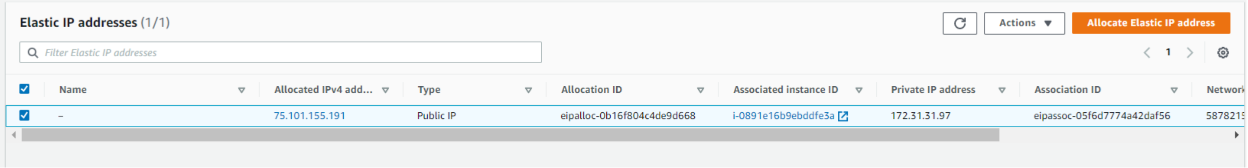


Screenshots of enhancements and explanation why I added these enhancements

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

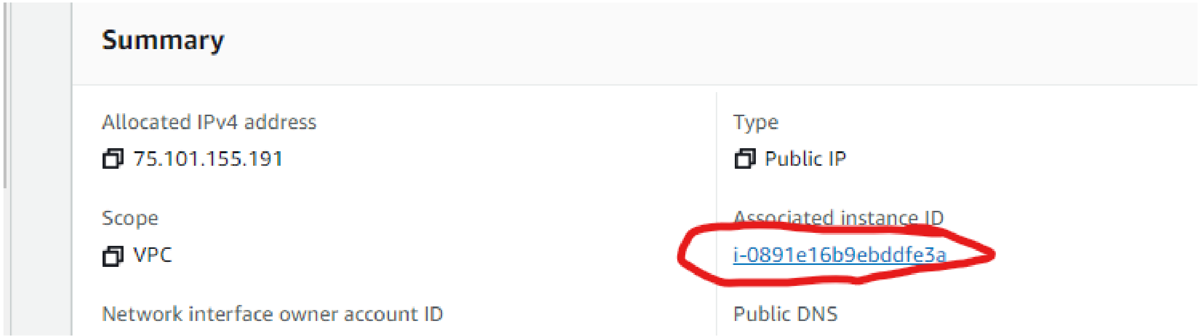
1) Click “Allocate Elastic IP address”
2) Publick IPv4 address pool : Amazon’s pool of IPv4 address



Elastic IP addresses (1/1)

☑	Name	Allocated IPv4 add...	Type	Allocation ID	Associated instance ID	Private IP address	Association ID	Network
☑	–	75.101.155.191	Public IP	eipalloc-0b16f804c4de9d668	i-0891e16b9ebddfe3a	172.31.31.97	eipassoc-05f6d7774a42daf56	5878213

3)
4) Under the elastic ip address that I created, I can see the Associated Instance ID of my ec2



Summary

Allocated IPv4 address
75.101.155.191

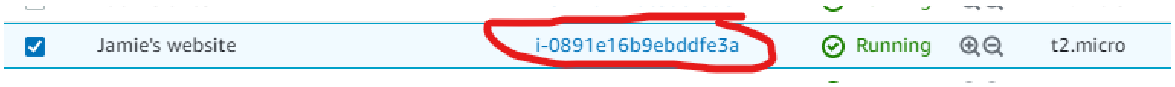
Type
Public IP

Scope
VPC

Associated instance ID
i-0891e16b9ebddfe3a

Network interface owner account ID

Public DNS

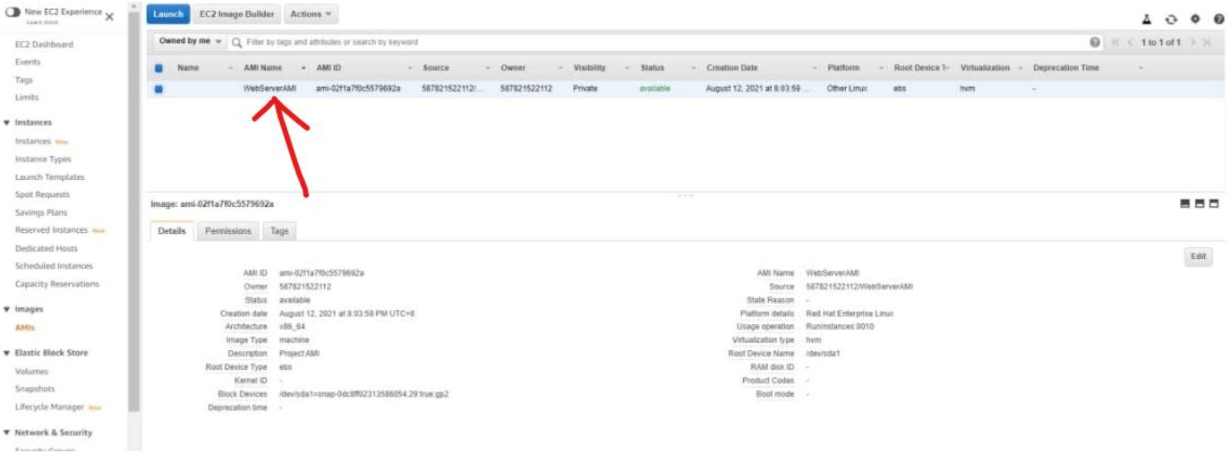


☑	Jamie's website	i-0891e16b9ebddfe3a	✓ Running	t2.micro

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”



Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform	Root Device T	Virtualization	Deprecation Time
	WebServerAMI	ami-02f1a7f0c5579692a	587821522112/...	587821522112	Private	available	August 12, 2021 at 8:03:59 ...	Other Linux	ebs	hvm	-

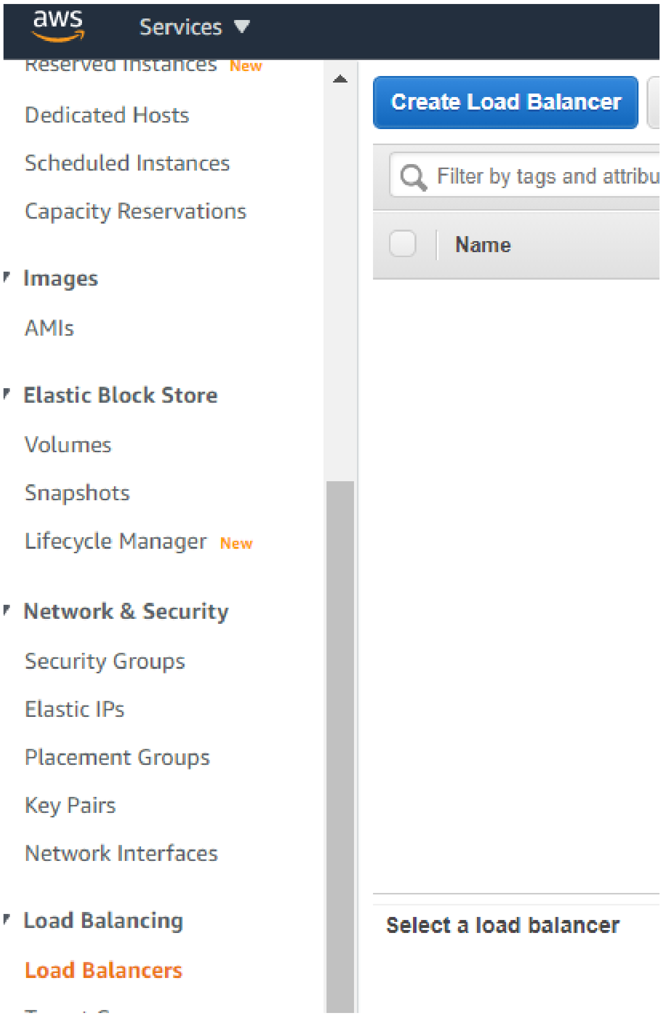
Image: ami-02f1a7f0c5579692a

AMI ID	ami-02f1a7f0c5579692a	AMI Name	WebServerAMI
Owner	587821522112	Source	587821522112/WebServerAMI
Status	available	State Reason	-
Creation date	August 12, 2021 at 8:03:59 PM UTC+8	Platform details	Red Hat Enterprise Linux
Architecture	x86_64	Usage operation	RunInstances:0010
Image Type	machine	Virtualization type	hvm
Description	Project AMI	Root Device Name	/dev/sda1
Root Device Type	ebs	RAM disk ID	-
Kernel ID	-	Product Codes	-
Block Devices	/dev/sda1=snap-0dc8ff02313586054:29:true:gp2	Boot mode	-
Deprecation time	-		

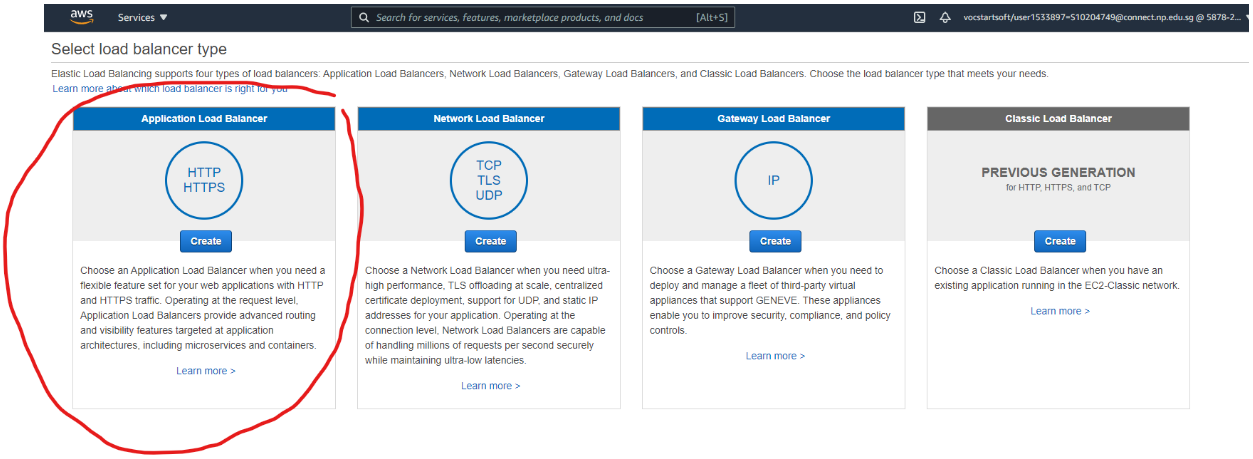
Load balancer

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**



3) As the Load Balancer I chose **"HTTP HTTPS"**



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1533897=S10204749@connect.np.edu.sg @ 5878-2...

Select load balancer type

Elastic Load Balancing supports four types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers. Choose the load balancer type that meets your needs.
Learn more about which load balancer is right for you

Application Load Balancer

HTTP
HTTPS

Create

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Learn more >

Network Load Balancer

TCP
TLS
UDP

Create

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Learn more >

Gateway Load Balancer

IP

Create

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Learn more >

Classic Load Balancer

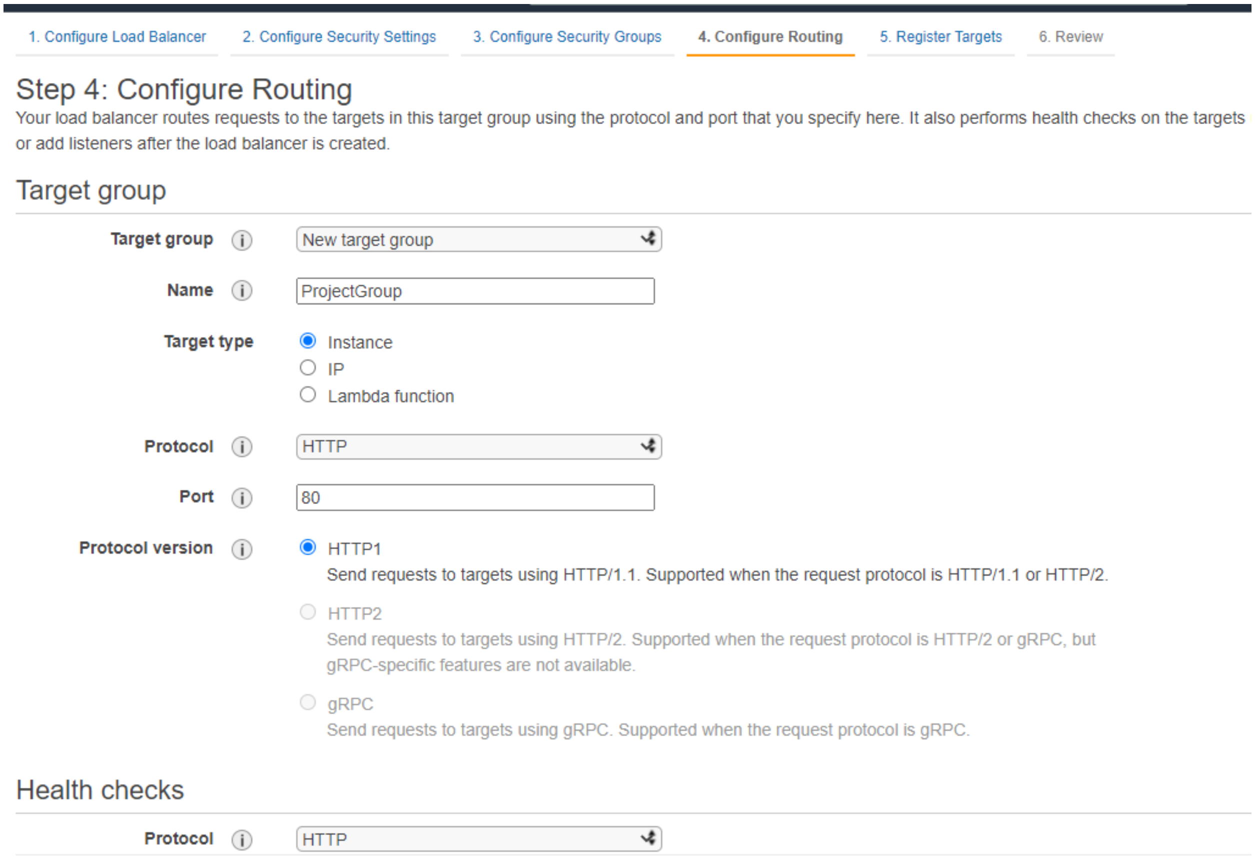
PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

Create

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Learn more >

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



1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 4: Configure Routing

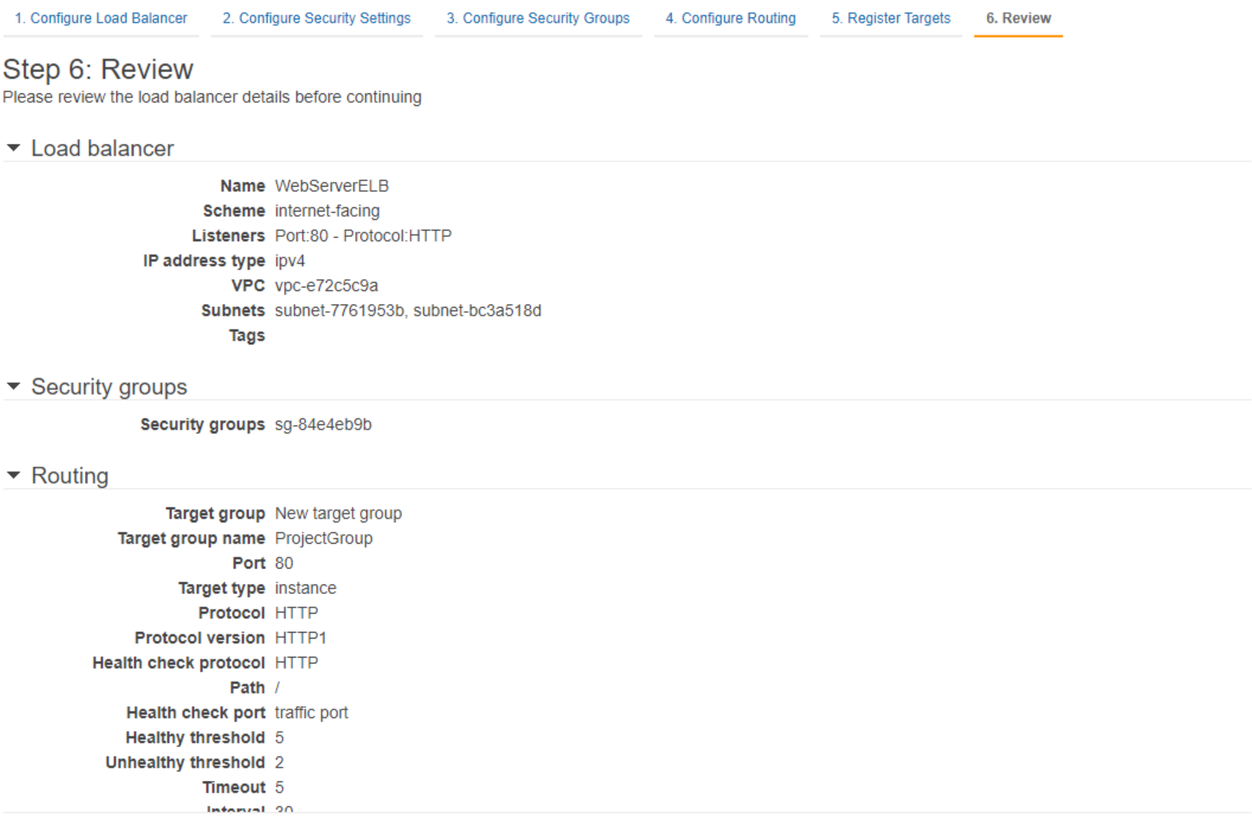
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on the targets or add listeners after the load balancer is created.

Target group

Target group	New target group
Name	ProjectGroup
Target type	● Instance ○ IP ○ Lambda function
Protocol	HTTP
Port	80
Protocol version	● HTTP1 Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2. ○ HTTP2 Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available. ○ gRPC Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol	HTTP


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 6: Review

Please review the load balancer details before continuing

Load balancer

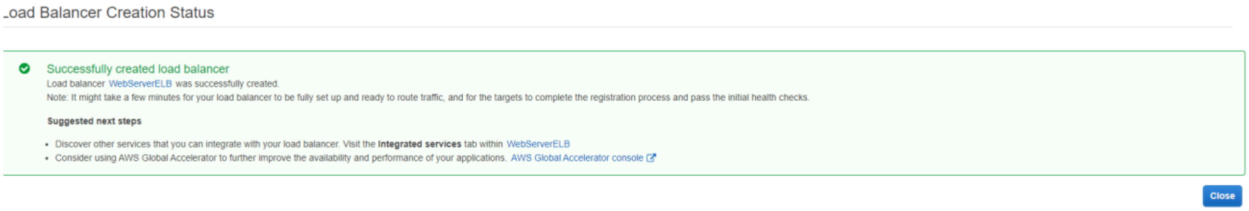
Name	WebServerELB
Scheme	internet-facing
Listeners	Port:80 - Protocol:HTTP
IP address type	ipv4
VPC	vpc-e72c5c9a
Subnets	subnet-7761953b, subnet-bc3a518d
Tags	

Security groups

Security groups	sg-84e4eb9b

Routing

Target group	New target group
Target group name	ProjectGroup
Port	80
Target type	instance
Protocol	HTTP
Protocol version	HTTP1
Health check protocol	HTTP
Path	/
Health check port	traffic port
Healthy threshold	5
Unhealthy threshold	2
Timeout	5
Interval	30



.oad Balancer Creation Status

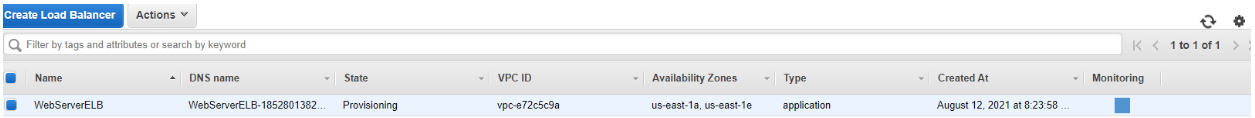
Successfully created load balancer

Load balancer WebServerELB was successfully created.

Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within WebServerELB
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. AWS Global Accelerator console

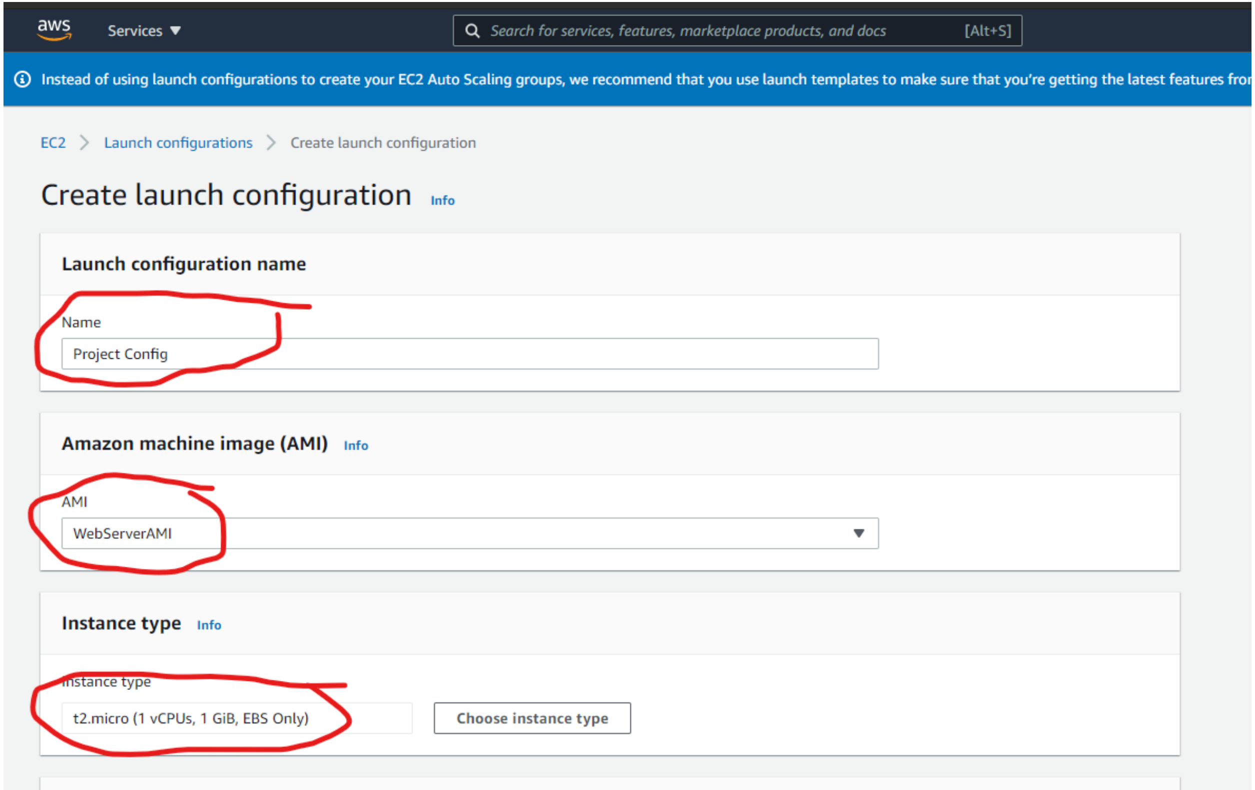


■	Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
■	WebServerELB	WebServerELB-1852801382...	Provisioning	vpc-e72c5c9a	us-east-1a, us-east-1e	application	August 12, 2021 at 8:23:58 ...	■

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:


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
EC2 > Launch configurations > Create launch configuration
Info
Choose instance type

Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates to make sure that you're getting the latest features fro

Create launch configuration

Launch configuration name

Name: Project Config

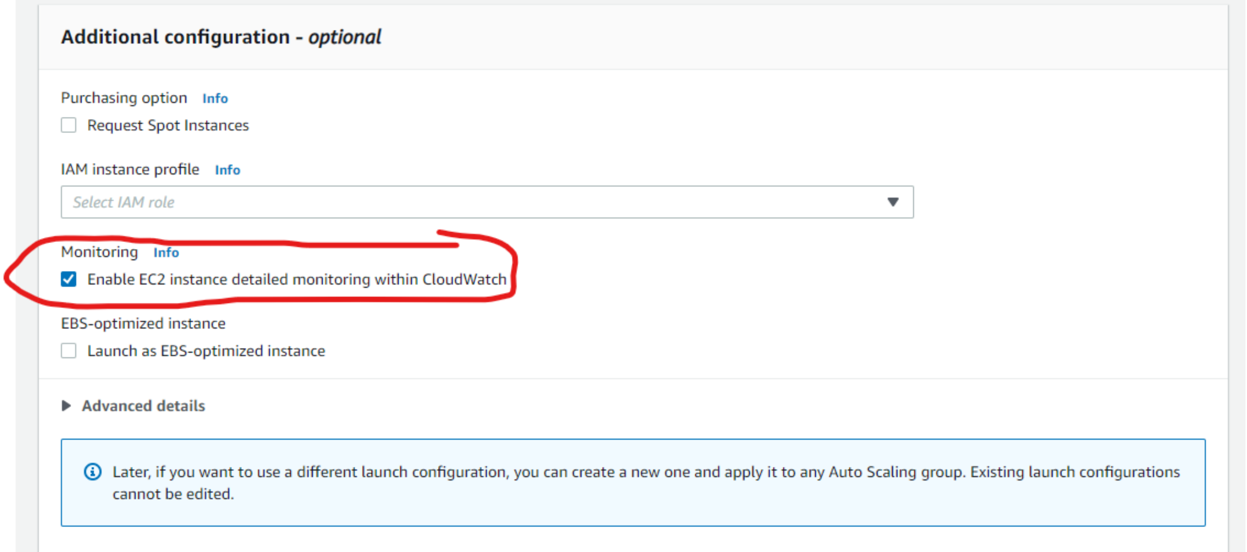
Amazon machine image (AMI)

AMI: WebServerAMI

Instance type

Instance type: t2.micro (1 vCPUs, 1 GiB, EBS Only)

4) I selected "Enable EC2 instance detailed monitoring within CloudWatch" for auto scaling to react quickly to changing utilization



Additional configuration - *optional*

Purchasing option
□ Request Spot Instances

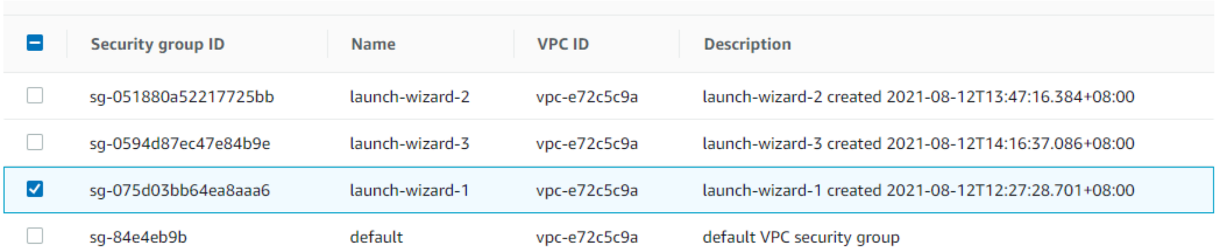
IAM instance profile

Monitoring
☑ Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance
□ Launch as EBS-optimized instance

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

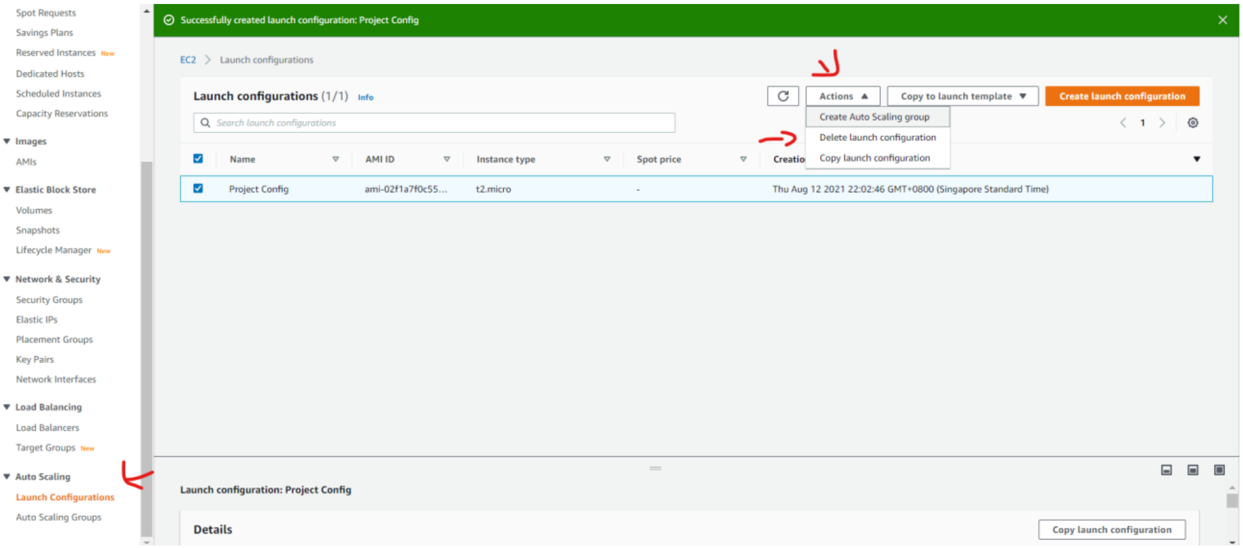
5) Security group I used “launch-wizard-1”



☐	Security group ID	Name	VPC ID	Description
☐	sg-051880a52217725bb	launch-wizard-2	vpc-e72c5c9a	launch-wizard-2 created 2021-08-12T13:47:16.384+08:00
☐	sg-0594d87ec47e84b9e	launch-wizard-3	vpc-e72c5c9a	launch-wizard-3 created 2021-08-12T14:16:37.086+08:00
☑	sg-075d03bb64ea8aaa6	launch-wizard-1	vpc-e72c5c9a	launch-wizard-1 created 2021-08-12T12:27:28.701+08:00
☐	sg-84e4eb9b	default	vpc-e72c5c9a	default VPC security group

6) I used an existing key pair which is “JamieKeyPair.pem”.

7) Under Launch configuration, I created an auto scaling group by clicking the following:



Successfully created launch configuration: Project Config

Launch configurations (1/1) Info

☑	Name	AMI ID	Instance type	Spot price	Creation
☑	Project Config	ami-02f1a7f0c55...	t2.micro	-	Thu Aug 12 2021 22:02:46 GMT+0800 (Singapore Standard Time)

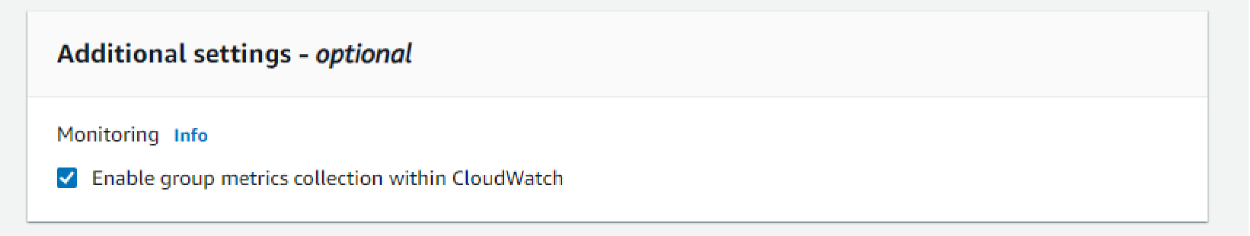
Launch configuration: Project Config

Details

8) I set the name as “Project Auto Scaling Group”

9) I chose the subnets: “us-east-1a” and “us-east-1e”

10) I attached an existing load balancer which is the “Project Group|HTTP”



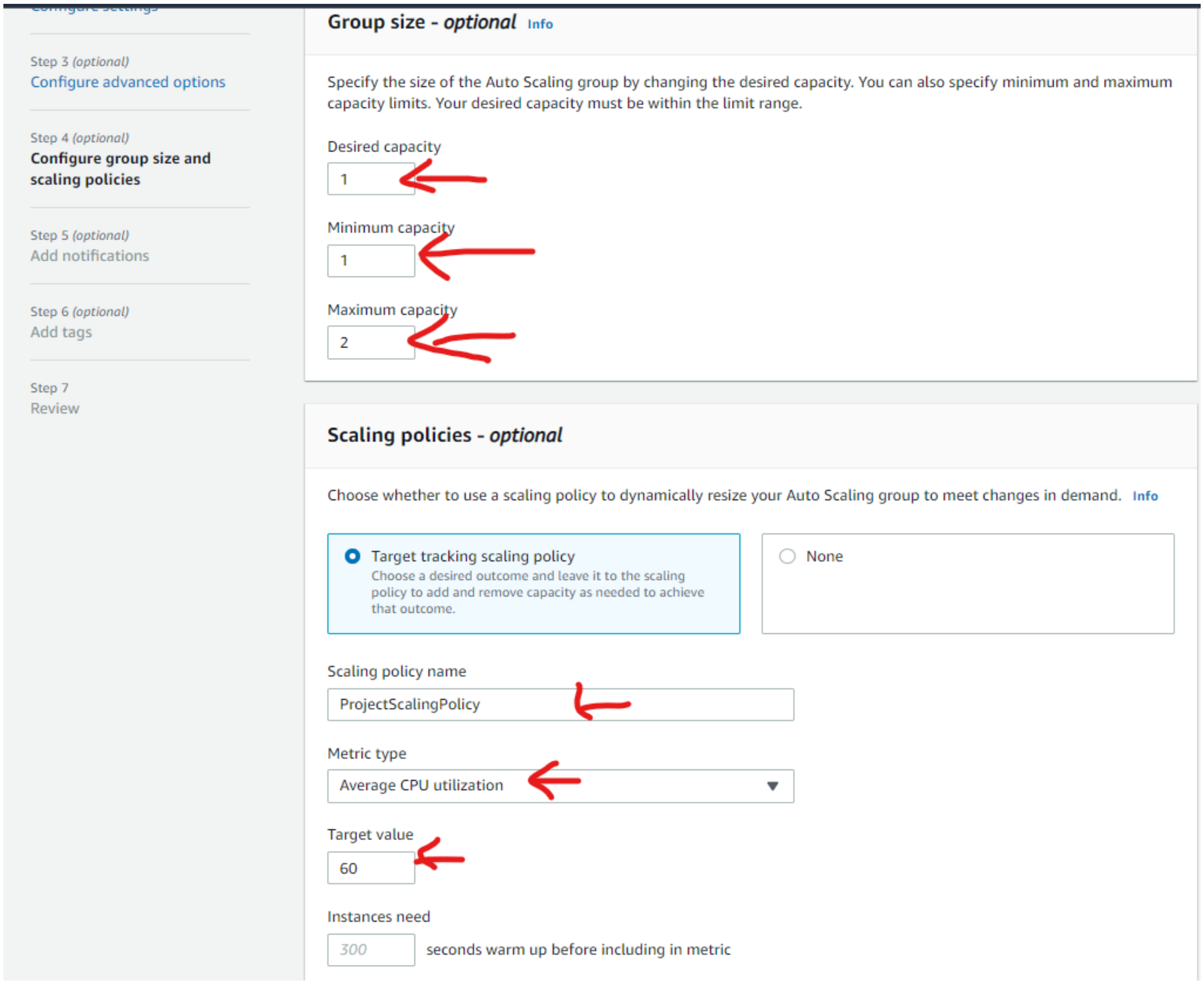
Additional settings - *optional*

Monitoring Info

☑ Enable group metrics collection within CloudWatch

This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns

11) I set the desired capacity, minimum capacity dn maximum capacity as follows:



Step 3 *(optional)*
Configure advanced options

Step 4 *(optional)*
Configure group size and scaling policies

Step 5 *(optional)*
Add notifications

Step 6 *(optional)*
Add tags

Step 7
Review

Group size - *optional* Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity: 1

Minimum capacity: 1

Maximum capacity: 2

Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

● Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

○ None

Scaling policy name: ProjectScalingPolicy

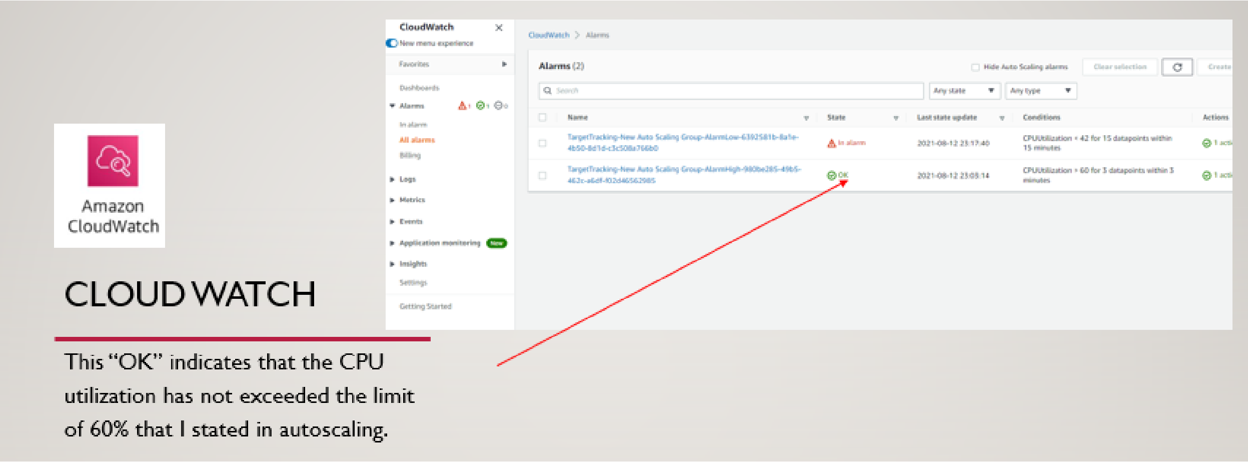
Metric type: Average CPU utilization

Target value: 60

Instances need: 300 seconds warm up before including in metric

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

12)



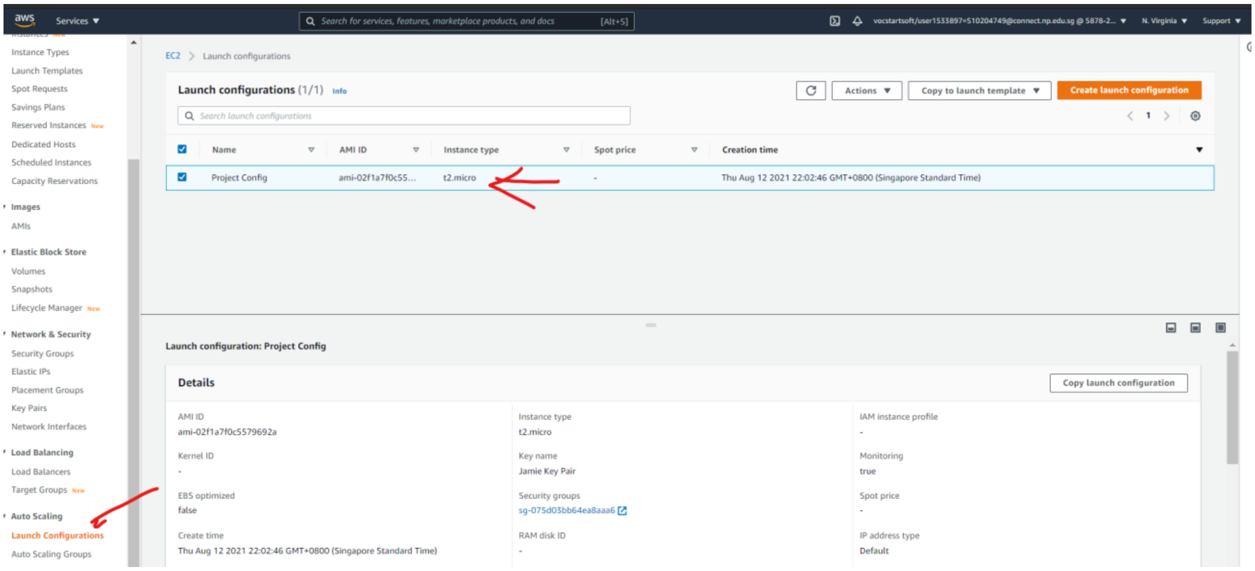
Alarms (2)

Name	State	Last state update	Conditions	Actions
TargetTracking-New Auto Scaling Group-AlarmLow-6392581b-8a1e-4b50-8d1d-c3c508a766b0	In alarm	2021-08-12 23:17:40	CPUUtilization < 42 for 15 datapoints within 15 minutes	1 acti
TargetTracking-New Auto Scaling Group-AlarmHigh-980be285-49b5-462c-a6df-f02d46562985	OK	2021-08-12 23:03:14	CPUUtilization > 60 for 3 datapoints within 3 minutes	1 acti

CLOUD WATCH

This "OK" indicates that the CPU utilization has not exceeded the limit of 60% that I stated in autoscaling.

13) For the tags : “Key: Name” and “Value: Project Instance”



Launch configurations (1/1)

☑	Name	AMI ID	Instance type	Spot price	Creation time
☑	Project Config	ami-02f1a7f0c55...	t2.micro	-	Thu Aug 12 2021 22:02:46 GMT+0800 (Singapore Standard Time)

Launch configuration: Project Config

Details

AMI ID: ami-02f1a7f0c5579692a
Instance type: t2.micro
IAM instance profile: -
Kernel ID: -
Key name: Jamie Key Pair
Monitoring: true
EBS optimized: false
Security groups: sg-075d03bb64ea8aaa6
Spot price: -
Create time: Thu Aug 12 2021 22:02:46 GMT+0800 (Singapore Standard Time)
RAM disk ID: -
IP address type: Default

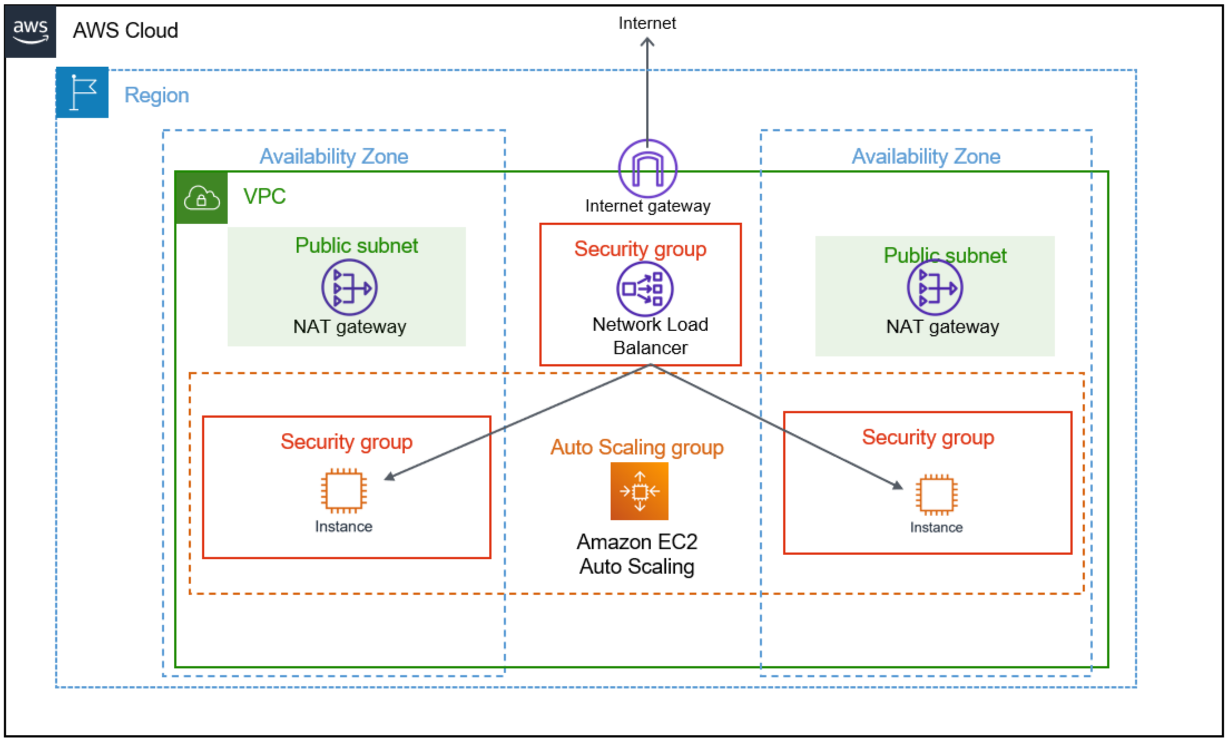
Difficulties I encountered

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.


AWS Cloud
Internet
Region
Availability Zone
Availability Zone
VPC
Internet gateway
Public subnet
NAT gateway
Security group
Network Load Balancer
Public subnet
NAT gateway
Security group
Instance
Auto Scaling group
Amazon EC2 Auto Scaling
Security group
Instance