

### **1. Differences between ELB, ALB, and NLB. Where will you use which one?**

Elastic Load Balancer	Application Load Balancer	Network Load Balancer
No target groups needed	Target Groups are involved	Target groups are involved
Works on layer 4 of OSI	Works on layer 7 of OSI	Works on layer 4 of OSI

- If we need to redirect loads on EC2 classic instances or on individual instances, use ELB
- If we need to redirect traffic on the basis of instances, IP's and Lambdas or we need to load balance HTTP and HTTPS traffic, or we need to redirect load based on hosts and paths use ALB.
- If we need to redirect TCP traffic , use NLB.

### **2.Differences between step scaling and target scaling.**

Step Scaling	Target Scaling
We can specify multiple thresholds for different responses.	We specify threshold based on a particular target, example if we have CPU utilization above 30, it will launch a new instance.
Used for fine grained control.	Used generally

### **3. Differences between Launch configuration and launch template.**

Launch Configuration	Launch Template
It cannot be versioned	It can be versioned
It provides less options	We can specify advanced options
No T2 unlimited access	T2 unlimited access is there

#### 4. Differences between EC2 health check and load balancer health check.

EC2 Health check	Load Balancer Health check
Watches for instance availability from hypervisor and networking point of view.	Verifies network level availability.
If instance is wrongly configured and don't respond to network requests, it is marked as unhealthy.	Verifies whether the specified tcp and http port is accepting requests or not.

#### 5. Create 2 auto-scaling groups with launch configuration and launch template.

#### USING LAUNCH CONFIGURATION

Go to launch configurations in the left side pane, Create a launch configuration

#### Select AMI

## Select instance

The screenshot shows the 'Create Launch Configuration' wizard at step 2. The 'Choose Instance Type' section is active. A table lists three instance types: t2.nano, t2.micro (selected), and t2.small. The t2.micro row is highlighted with a green background and has a 'Free tier eligible' badge.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate

## Create launch configuration

The screenshot shows the 'Create Launch Configuration' wizard at step 3. The 'Configure details' section is active. It includes fields for Name (empty), Purchasing option (Request Spot Instances checked), IAM role (None selected), and Monitoring (CloudWatch detailed monitoring checked). A note states: '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.'

## Add storage

The screenshot shows the 'Create Launch Configuration' wizard at step 4. The 'Add Storage' section is active. It shows a table for adding storage volumes. One volume is listed: 'Root' device '/dev/sda1' using snapshot 'snap-0e078112eedeecc9db', size '8 GiB', volume type 'General Purpose (SSD)', IOPS '100 / 3000', throughput 'N/A', delete on termination checked, and encrypted 'No'. A note says: 'Free tier eligible customers can get up to 30 GB of EBS storage. Learn more about free usage tier eligibility and usage restrictions.'

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0e078112eedeecc9db	8	General Purpose (SSD)	100 / 3000	N/A	<input checked="" type="checkbox"/>	No

## Add security group

The screenshot shows the 'Create Launch Configuration' wizard at step 5, 'Configure Security Group'. It displays a table for defining security rules. A warning message at the bottom states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere (0.0.0.0/0)

**Assign a security group:**  
 Create a new security group  
 Select an existing security group  
**Security group name:** AutoScaling-Security-Group-2  
**Description:** AutoScaling-Security-Group-2 (2020-02-25 23:38:38.536+05:30)

## Review and launch

The screenshot shows the 'Create Launch Configuration' wizard at step 6, 'Review'. A modal window titled 'Select an existing key pair or create a new key pair' is open. It shows a dropdown for 'Choose an existing key pair' set to 'srima-TTN-bootcamp'. A checkbox below it is checked, stating: 'I acknowledge that I have access to the selected private key file (Srima-TTN-bootcamp.pem), and that without this file, I won't be able to log into my instance.' The main review screen shows the launch configuration details, including the security group 'AutoScaling-Security-Group-2' and the instance type 't2.micro'.

## Create an auto scaling group

The screenshot shows the EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Scheduled Instances. At the top, there's a promotional banner for saving up to 90% on Compute by creating an Auto Scaling group. Below the banner, there are buttons for 'Create launch configuration', 'Create Auto Scaling group', 'Copy to launch template', and 'Actions'. A table lists one launch configuration: 'SrimaLC1' with AMI ID 'ami-07ebfd5b3...', Instance Type 't2.micro', and Creation Time 'February 25, 2020 at 11:41:57 ...'.

Srimali Pal | AWS Certified Solutions Architect - Professional | LinkedIn

Services ▾ Resource Groups ▾

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

### Create Auto Scaling Group

Group name: SrimaliASG1

Launch Configuration: SrimaliLC1

Group size: Start with 1 instances

Network: vpc-01d9bc1ea53fdce9 (10.0.0.0/16) | t34ak | Create new VPC

Subnet: subnet-01f98a351ed87a94b(10.0.100.0/24) | valbhav1a | us-east-1a | subnet-097975492bf68a9df(10.0.101.0/24) | valbhav1b | us-east-1b | Create new subnet

**No public IP addresses will be assigned**  
None of the instances in this Auto Scaling group will be assigned a public IP address because you have not chosen to launch in your default VPC and subnet.

## Configure scaling policy

Srimali Pal | AWS Certified Solutions Architect - Professional | LinkedIn

Services ▾ Resource Groups ▾

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

### Create Auto Scaling Group

Scale between 1 and 3 instances. These will be the minimum and maximum size of your group.

**Scale Group Size**

Name: Scale Group Size

Metric type: Average CPU Utilization

Target value: 30

Instances need: 300 seconds to warm up after scaling

Disable scale-in:

Scale the Auto Scaling group using step or simple scaling policies (i)

Cancel Previous Review Next: Configure Notifications

## Review settings

Srimali Pal | AWS Certified Solutions Architect - Professional | LinkedIn

Services ▾ Resource Groups ▾

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

### Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

**Auto Scaling Group Details**

Group name	SrimaliASG1
Group size	1
Minimum Group Size	1
Maximum Group Size	3
Subnet(s)	subnet-01f98a351ed87a94b, subnet-097975492bf68a9df
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None
Service-Linked Role	AWSServiceRoleForAutoScaling

**Scaling Policies**

Scale Group Size: Maintain metric type Average CPU Utilization at target value 30, with 300 seconds for instances to warm up.

**Notifications**

Edit scaling policies Edit notifications

Cancel Previous Create Auto Scaling group

## Autoscaling group created

### Auto Scaling group creation status

✓ Successfully created Auto Scaling group  
[View creation log](#)

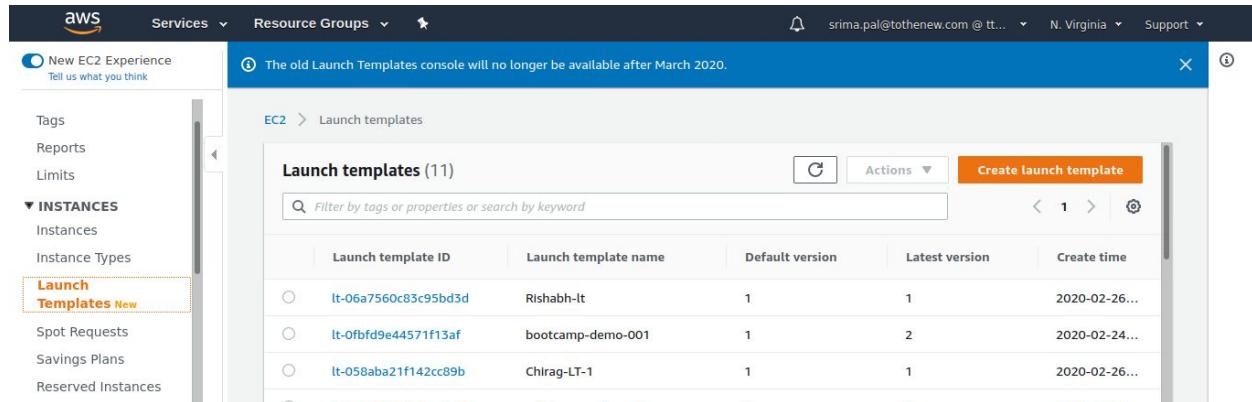
▼ View  
[View your Auto Scaling groups](#)  
[View your launch configurations](#)

► Here are some helpful resources to get you started

[Close](#)

## USING LAUNCH TEMPLATES

Go to Launch templates and click on Create Launch Template



The screenshot shows the AWS EC2 Launch Templates console. The left sidebar has a 'Launch Templates New' section highlighted with a dashed orange border. The main area displays a table titled 'Launch templates (11)'. The table columns are: Launch template ID, Launch template name, Default version, Latest version, and Create time. The data rows are:

Launch template ID	Launch template name	Default version	Latest version	Create time
lt-06a7560c83c95bd3d	Rishabh-LT	1	1	2020-02-26...
lt-0fbfd9e44571f13af	bootcamp-demo-001	1	2	2020-02-24...
lt-058aba21f142cc89b	Chirag-LT-1	1	1	2020-02-26...
...	...	...	...	...



Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

<b>Amazon machine image (AMI)</b> <small>Info</small>
AMI
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type ami-07ebfd5b3428b6f4d Catalog: Quick Start   architecture: 64-bit (x86)   virtualization: hvm
<b>Instance type</b> <small>Info</small>
Instance type
t2.micro Family: General purpose   1 vCPU   1 GiB Memory On-Demand Linux pricing: 0.0116 USD per Hour On-Demand Windows pricing: 0.0162 USD per Hour

## Created launch template

The screenshot shows the 'Launch templates' list. It has a header 'Launch templates (13)' with a search bar and a 'Create launch template' button. Two rows are visible: one for 'lt-0032557639a03a697' (name 't34ak-wordpress-Image') and another for 'lt-0f95bf9a4e760bb22' (name 'Srima-LT01').

Launch template ID	Launch template name	Default version	Latest version
lt-0032557639a03a697	t34ak-wordpress-Image	1	1
lt-0f95bf9a4e760bb22	Srima-LT01	1	1

Select the launch template and create auto scaling group

The screenshot shows the AWS EC2 Launch Templates console. On the left, there's a sidebar with navigation links like 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'IMAGES' (AMIs), 'ELASTIC BLOCK STORE' (Volumes, Snapshots), 'NETWORK & SECURITY' (Security Groups), and 'New EC2 Experience'. The main area is titled 'Launch templates (14)' and shows a table with columns 'Launch template ID' and 'Launch template name'. The 'Srima-LT02' entry is highlighted with a blue border. At the top right of the table, there's a 'Create launch template' button. Below the table, there's a section for the selected 'Srima-LT02' launch template.

Create another auto scaling group in the same way

The screenshot shows the 'Create Auto Scaling Group' wizard, step 5: Review. The page title is 'Create Auto Scaling Group'. It says 'Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.' Below this, there's a 'Auto Scaling Group Details' section with various configuration options. At the bottom right, there are 'Cancel', 'Previous', and 'Create Auto Scaling group' buttons.

## 6. Setup autoscaling Wordpress application with the Application load balancer. Auto-scaling should be triggered based on CPU usage of EC2 instances.

Create an AMI for wordpress to use in launch template

Launch an instance and install nginx

```
ubuntu@ip-172-31-116-175:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor prese
  Active: active (running) since Wed 2020-02-26 12:04:09 UTC; 11min ago
    Docs: man:nginx(8)
   Main PID: 10620 (nginx)
      Tasks: 1 (limit: 1902)
     CGroup: /system.slice/nginx.service
             └─ 10620 /usr/sbin/nginx -g daemon on; master process runn
# systemctl enable mariadb.service
```

Install maria db

```
ubuntu@ip-172-31-116-175:~$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.1.44-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

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

The installation of MariaDB is complete in your Ubuntu 18.04 LTS system.

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

MariaDB [(none)]>
```

### Step 3: Install PHP

Install php

```
ubuntu@ip-172-31-116-175:~$ sudo !!
sudo apt-get install php7.2 php7.2-cli php7.2-fpm php7.2-mysql php7.2-json
php7.2-opcache php7.2-mbstring php7.2-xml php7.2-gd php7.2-curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libsodium23 php-common php7.2-common php7.2-readline
Suggested packages:
```

Create wordpress database

```
ubuntu@ip-172-31-116-175:~$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.1.44-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

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

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

MariaDB [(none)]> CREATE DATABASE wordpress_db;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL ON wordpress_db.* TO 'srima'@'localhost' IDENTI
FIED BY 'pass' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

Install php

Create root folder for wordpress installation

```
ubuntu@ip-172-31-116-175:~$ sudo !!
sudo mkdir -p /var/www/html/wordpress/public_html
```

Create wordpress.conf in sites-available

The screenshot shows a terminal window with the following content:

```
ubuntu@ip-172-31-116-175:~$ sudo nano /etc/nginx/sites-available/wordpress.conf
Server {
    listen 80;
    root /var/www/html/wordpress/public_html;
    index index.php index.html;
    server_name abc.com;

    access_log /var/log/nginx/abc.access.log;
    error_log /var/log/nginx/abc.error.log;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        #mkdir -p /var/www/html/wordpress/public_html/.php snippets/fastcgi-php.conf;
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.2-fpm.sock;
    }
}
```

Annotations in the screenshot:

- A tooltip for the `try_files` directive says: "Let us now proceed with configuring NGINX server block to start with create the root folder for your WordPress instance".
- A tooltip for the `fastcgi_pass` directive says: "To create NGINX server block for your WordPress domain".

Test and reload nginx

```
ubuntu@ip-172-31-116-175:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-172-31-116-175:/etc/nginx/sites-available$ sudo systemctl reload
nginx
In this step, download the archived WordPress file using
ubuntu@ip-172-31-116-175:/etc/nginx/sites-available$ cd ..
ubuntu@ip-172-31-116-175:/etc/nginx$ cd sites-enabled/
ubuntu@ip-172-31-116-175:/etc/nginx/sites-enabled$ sudo ln -s ..sites-available/wordpress.conf .
```

Download wordpress using wget

```
ubuntu@ip-172-31-116-175:/var/www/html/wordpress/public_html$ ls
latest.tar.gz
```

Untar wordpress

tar -zxf latest.tar.gz

```
ubuntu@ip-172-31-116-175:/var/www/html/wordpress/public_html$ ls
latest.tar.gz  wordpress
ubuntu@ip-172-31-116-175:/var/www/html/wordpress/public_html$
```

```
mv wordpress/* .
rm -rf wordpress
```

Change the ownership and apply correct permissions to the extracted WordPress files and folders.

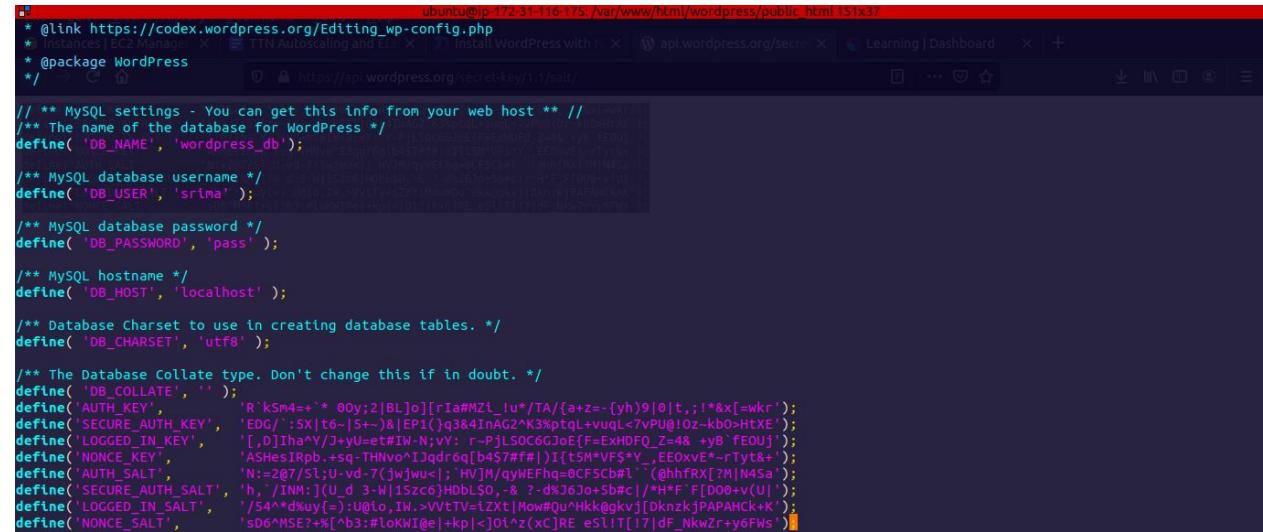
```
cd /var/www/html/wordpress/public_html
chown -R www-data:www-data *
chmod -R 755 *
```

```
39 mv wordpress/* .
40 sudo mv wordpress/* .
41 rm -rf wordpress
42 sudo rm -rf wordpress
43 chown -R www-data:www-data *
44 sudo chown -R www-data:www-data *
45 sudo chmod -R 755 *
46 history
```

folders.

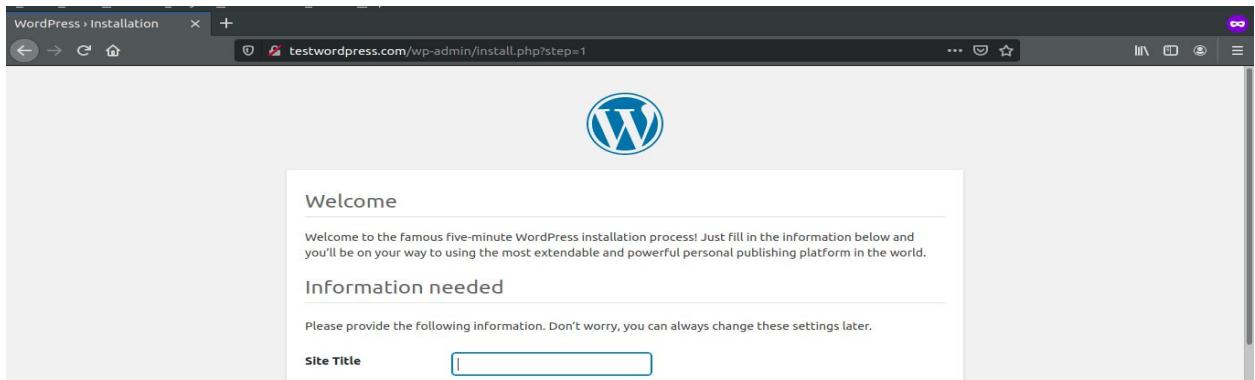
```
cd /var/www/html/wordpress/public_html
chown -R www-data:www-data *
chmod -R 755 *
```

Change the wrd-config.php file



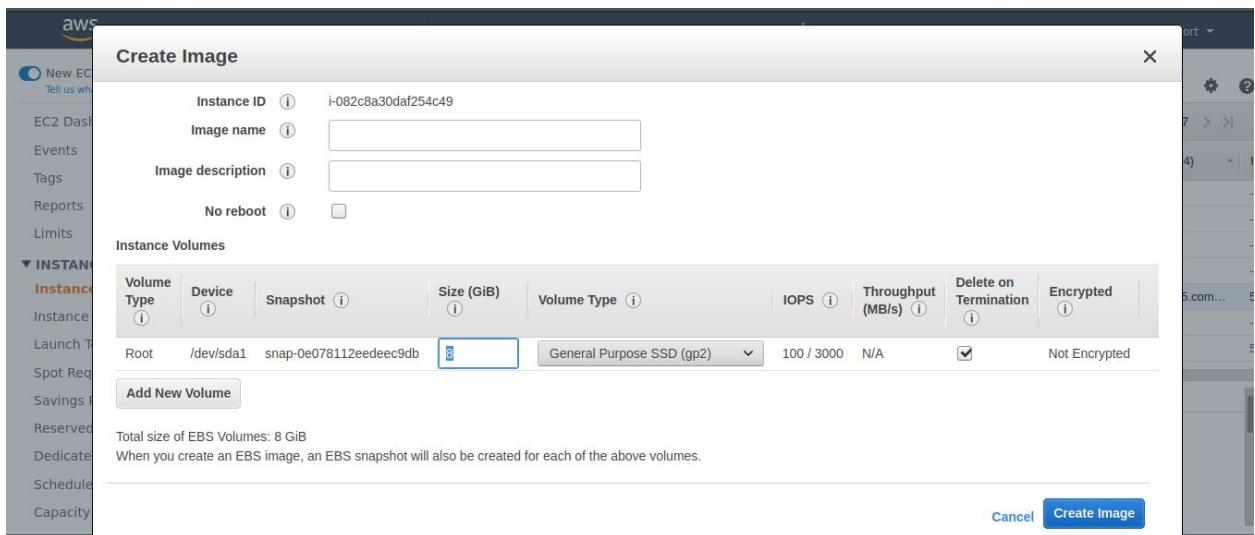
```
ubuntu@ip-172-31-116-175:~$ cd /var/www/html/wordpress/public_html/151x37
* @link https://codex.wordpress.org/Editing_wp-config.php
* Instances | EC2 Manager | T1A AutoScaling and EC2 | Install WordPress with ...
* @package WordPress
*/
** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress_db' );
/** MySQL database username */
define( 'DB_USER', 'srima' );
/** MySQL database password */
define( 'DB_PASSWORD', 'pass' );
/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
define('AUTH_KEY', 'R`kSm4=+^* 0Oy:2|BL]o|[rIa#MzL_u*/TA/[a+z=-[yh)9|0|t,;!*8x[wkr'];
define('SECURE_AUTH_KEY', 'EDG/ :5X|t6-|5+->|EP1(|q384InAG2^K3kptql+vul<7vPU@10z-kb0+HtxE');
define('LOGGED_IN_KEY', '[,D]ha#Y/J+yU+et#IN-N;VY: r-PjL5OC6GJoE(F=ExHDFQ_Z=48 +yB'FOUj');
define('NONCE_KEY', 'ASHeSIRpb,+sq-THnvo^I3qdr6q[b457#/#])I{tSM*VFS*Y_,EOoxVe*-rTyt&+');
define('AUTH_SALT', 'N:=207/S1:U-vd-7{jwJu<];HV]M/qyHEFhq=0CF5Cb1``(hhfRX[?M|N45a');
define('SECURE_AUTH_SALT', 'h_;/IM:M](U,d 3-W15zc6)HbL5O_, & ?-d%26Jo+5b#c|/*H*F'F'D08+v(U)');
define('LOGGED_IN_SALT', '/54^*dxuy(y=:U@lo,IM->VVtTV=LZxt|lowl#Qu^Hkk@gkvj[DknzkjPAPAHck+k');
define('NONCE_SALT', 's6^*MSE+%^[b3:#loKHI@e+kp|<0|^z(x]RE eS!|1[!7|dF_NkwZr+y6Fws')
```

## Wordpress up



## Create AMI

Go to actions and create image



## CREATE AUTO SCALING GROUP

Create Launch Configuration

## Select instance type

### Create Launch Configuration

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
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.micro <span style="background-color: #00AEEF; color: white; padding: 2px;">Free tier eligible</span>	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate

## Configure Security Group

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput	Delete on Termination	Encrypted
Root	/dev/sda1	snap-07314738644eea331	8	General Purpose (SSD)	100 / 3000	N/A	<input checked="" type="checkbox"/>	No
<a href="#">Add New Volume</a>								

Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

## Configure security group

The screenshot shows the 'Create Launch Configuration' step in the AWS wizard. The 'Configure Security Group' tab is selected. A warning message states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The security group name is 'SrimaASG' and the description is 'AutoScaling-Security-Group-14 (2020-02-26 22:13:17.341+05:30)'. Two inbound rules are defined:

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
HTTP	TCP	80	Anywhere 0.0.0.0/0

## Select key-pair and create launch configuration

The screenshot shows the 'Create Launch Configuration' step. A modal dialog titled 'Select an existing key pair or create a new key pair' is open. It contains instructions about key pairs and a note: '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.' Below are dropdown menus for 'Choose an existing key pair' (set to 'Srima-TTN-bootcamp') and 'Select a key pair' (also set to 'Srima-TTN-bootcamp'). A checkbox is checked with the text: 'I acknowledge that I have access to the selected private key file (Srima-TTN-bootcamp.pem), and that without this file, I won't be able to log into my instance.' At the bottom are 'Cancel' and 'Create launch configuration' buttons.

## Configure auto scaling group

The screenshot shows the 'Create Auto Scaling Group' step. A red error message says: 'No default subnet found. Please choose another subnet in your default VPC, or choose another VPC.' Below are configuration fields:

- Group name: ASG-Srima
- Launch Configuration: SrimaL1
- Group size: Start with 1 instances
- Network: vpc-00470a42fc196d84e (10.0.0.0/16) | sarthak
- Subnet: subnet-01d770a77bb69a1f8(10.0.1.0/24) | sarthak-load-balancer-1 | us-east-1b  
subnet-0b14d740243a0165c(10.0.2.0/24) | sarthak-loadbalancer-2 | us-east-1c

Buttons at the top right include 'Cancel and Exit' and 'Don't show me this again'.

## Set Scaling Policy

Create Auto Scaling Group

Use scaling policies to adjust the capacity of this group

Scale between  and  instances. These will be the minimum and maximum size of your group.

Scale Group Size

Name: Scale Group Size

Metric type: Average CPU Utilization

Target value: 30

Instances need: 300 seconds to warm up after scaling

Cancel Previous Review Next: Configure Notifications

## Configure tags

1. Configure Auto Scaling group details   2. Configure scaling policies   3. Configure Notifications   4. Configure Tags   5. Review

### Create Auto Scaling Group

A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group when they launch. [Learn more](#).

Key	Value	Tag New Instances <small>i</small>
Name	Srima	<input checked="" type="checkbox"/>

Add tag 49 remaining

## Review the settings

1. Configure Auto Scaling group details   2. Configure scaling policies   3. Configure Notifications   4. Configure Tags   5. Review

### Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

#### Auto Scaling Group Details

[Edit details](#)

Group name	ASG-Srima
Group size	3
Minimum Group Size	3
Maximum Group Size	5
Subnet(s)	subnet-01d770a77bb69a1f8,subnet-0b14d740243a0165c
Health Check Grace Period	300
Detailed Monitoring	No
Instance Protection	None
Service-Linked Role	AWSServiceRoleForAutoScaling

#### Scaling Policies

[Edit scaling policies](#)

Scale Group Size Maintain metric type Average CPU Utilization at target value 30, with 300 seconds for instances to warm up.

[Cancel](#) [Previous](#) [Create Auto Scaling group](#)

## Create the ASG

### Auto Scaling group creation status

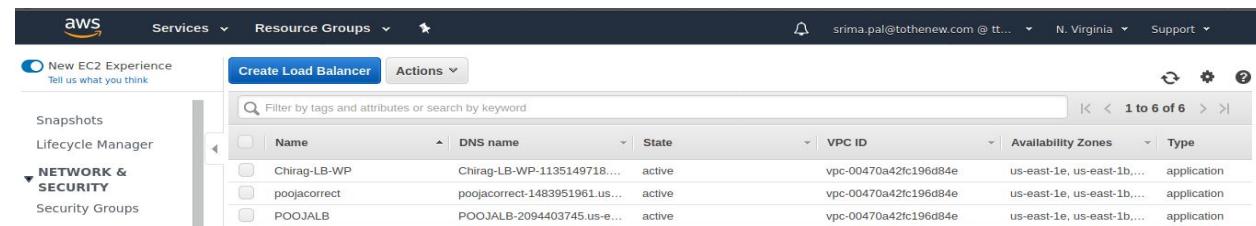
✓ Successfully created Auto Scaling group  
[View creation log](#)

▼ View  
[View your Auto Scaling groups](#)  
[View your launch configurations](#)

## Launching instances

Filter: Any Status ▾				Filter scaling history...	1 to 10 of 10 History Items
Status	Description	Start Time	End Time		
▶ Not yet in service	Launching a new EC2 instance: i-02501aea4e9a86835	2020 February 26 22:39:45 UTC+5:30			
▶ Not yet in service	Launching a new EC2 instance: i-0320027240a631a08	2020 February 26 22:39:45 UTC+5:30			

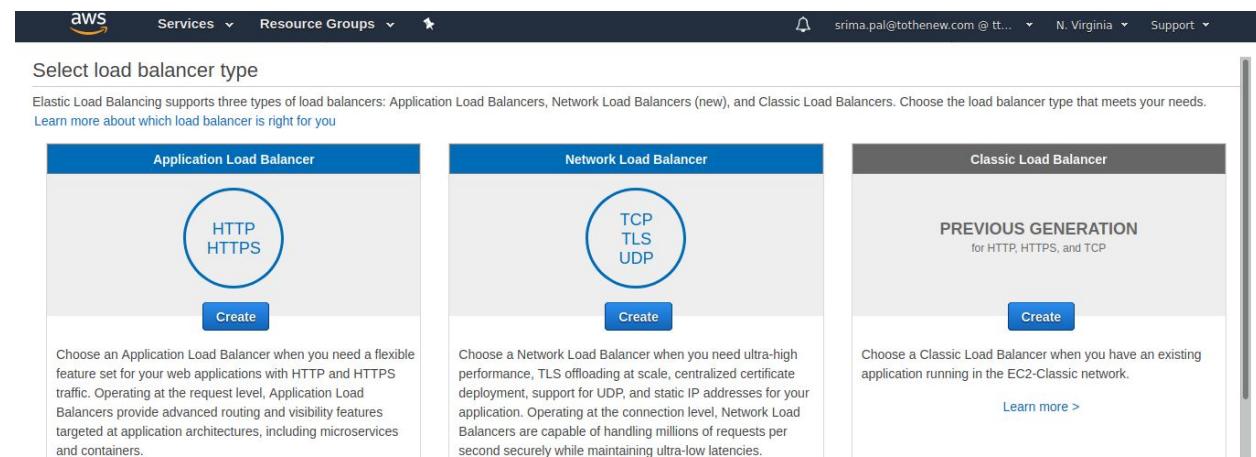
## Go to load balancer and create load balancer



The screenshot shows the AWS CloudFront console. On the left, there's a sidebar with 'New CloudFront Experience' and links for 'Schemas', 'Lifecycle Manager', and 'NETWORK & SECURITY'. The main area displays a table of existing CloudFront distributions:

Name	DNS name	State	VPC ID	Availability Zones	Type
Chirag-LB-WP	Chirag-LB-WP-1135149718....	active	vpc-00470a42fc196d84e	us-east-1e, us-east-1b...	application
poojacorrect	poojacorrect-1483951961.us...	active	vpc-00470a42fc196d84e	us-east-1e, us-east-1b...	application
POOJALB	POOJALB-2094403745.us-e...	active	vpc-00470a42fc196d84e	us-east-1e, us-east-1b...	application

## Select application load balancer



The screenshot shows the AWS Load Balancers console. It starts with a header: "Select load balancer type" and a note: "Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs." Below this, there are three cards:

- Application Load Balancer**: Features a large circle with "HTTP HTTPS" inside. A "Create" button is at the bottom. Description: "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."
- Network Load Balancer**: Features a large circle with "TCP TLS UDP" inside. A "Create" button is at the bottom. Description: "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."
- Classic Load Balancer**: Features a large circle with "PREVIOUS GENERATION for HTTP, HTTPS, and TCP" inside. A "Create" button is at the bottom. Description: "Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network." A "Learn more >" link is also present.

## Configure the load balancer

**Step 1: Configure Load Balancer**

**Basic Configuration**

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name: Srima-ALB

Scheme: internet-facing

IP address type: ipv4

**Listeners**

A listener is a process that checks for connection requests, using the protocol and port that you configured.

**Listeners**

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener

**Availability Zones**

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	vpc-d38d68b7 (172.31.0.0/16)   default (default)
Availability Zones	<input checked="" type="checkbox"/> us-east-1c subnet-06680a5b651f104dc

IPv4 address: Assigned by AWS

## Select an existing security group for load balancing

**Step 3: Configure Security Groups**

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-027522c779c12164c	Chirag-ALB-SG	load-balancer-wizard-1 created on 2020-02-26T19:40:34.731+05:30	<a href="#">Copy to new</a>
sg-0f87044e4ccf3814	default	default VPC security group	<a href="#">Copy to new</a>
sg-062b1633ad7e65c21	launch-wizard-177	launch-wizard-177 created 2020-02-26T18:25:24.592+05:30	<a href="#">Copy to new</a>
sg-051d671c160aea760	sarthak	ssh https http	<a href="#">Copy to new</a>
sg-0112cabec48317b21	Srima-ASG1	ASG	<a href="#">Copy to new</a>
sg-010a51e206309b0c9	VedantSG	VPC-sarthak	<a href="#">Copy to new</a>

Cancel Previous Next: Configure Routing

## Configure routing

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, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

#### Target group

Target group

Name

Target type  Instance  
 IP  
 Lambda function

Protocol

Port

[Cancel](#) [Previous](#) [Next: Register Targets](#)

### Step 4: Configure Routing

Path

Advanced health check settings

Port  traffic port  
 override

Healthy threshold

Unhealthy threshold

Timeout  seconds

Interval  seconds

Success codes

[Cancel](#) [Previous](#) [Next: Register Targets](#)

## Add instances to the target group

### Step 5: Register Targets

#### Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

Q	Srim	X					
	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input type="checkbox"/>	i-02604fc1fc2fe79	Srima-ASG	<span>running</span>	Srima-ASG1	us-east-1b	subnet-01d770a77bb69a1f8	10.0.1.0/24
<input type="checkbox"/>	i-02501aea4e9a86...	Srima-ASG	<span>running</span>	Srima-ASG1	us-east-1b	subnet-01d770a77bb69a1f8	10.0.1.0/24
<input type="checkbox"/>	i-0320027240a631...	Srima-ASG	<span>running</span>	Srima-ASG1	us-east-1e	subnet-008dc90bf26a9055	10.0.3.0/24
<input type="checkbox"/>	i-0fa11371c7c6dcce7	Srima-ASG	<span>running</span>	Srima-ASG1	us-east-1e	subnet-008dc90bf26a9055	10.0.3.0/24

## Review

### Load Balancer Creation Status

**✓ Successfully created load balancer**  
Load balancer **Srima-LB1** 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 **Srima-LB1**
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

[Close](#)

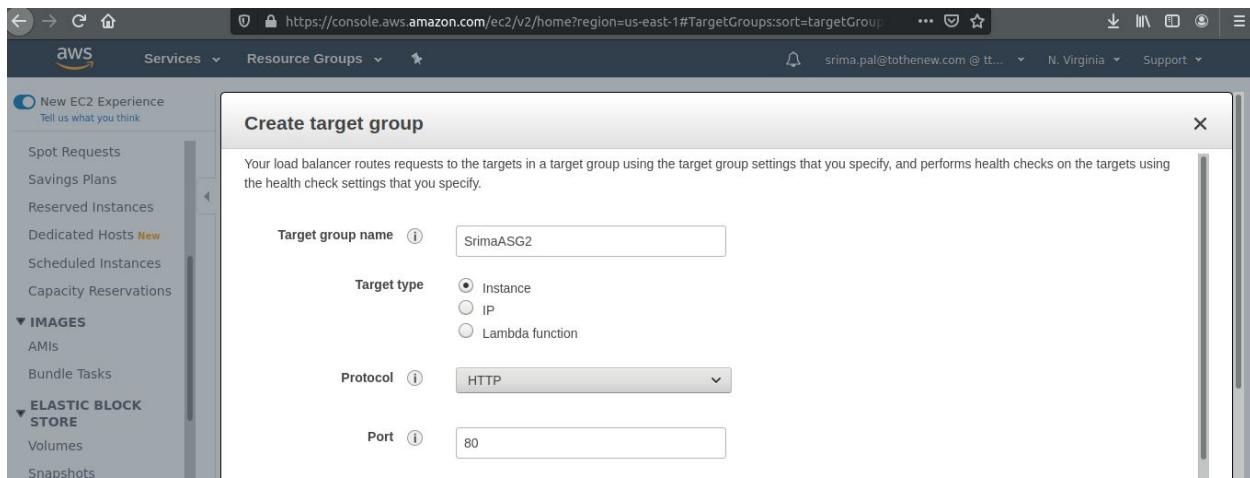
The screenshot shows the final review step of creating a new target group. The target group is named "Srima-TG" and uses port 80. The protocol is HTTP, and the health check protocol is also HTTP. The path for the health check is "/". The success codes are listed as 200. Below this, the "Targets" section lists four instances: i-02604fc1fcbb2ffe79 (Srima-ASG):80, i-02501aea4e9a86835 (Srima-ASG):80, i-0320027240a631a08 (Srima-ASG):80, and i-0fa11371c7c6dcce7 (Srima-ASG):80. At the bottom right, there are "Cancel", "Previous", and "Create" buttons.

Load balancer was successfully created

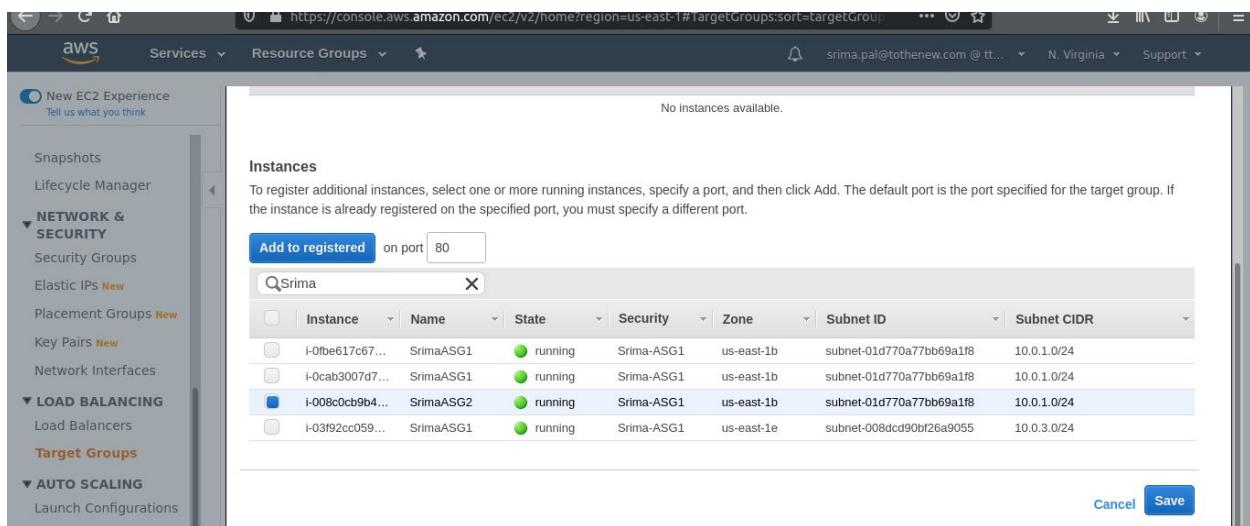
The screenshot shows a WordPress website titled "srima" with the URL "srimaalb-36473445.us-east-1.elb.amazonaws.com". The page displays a single post titled "Hello world!" under the "UNCATEGORIZED" category. The post was published on February 27, 2020, by user "srima". A comment section indicates there is 1 comment. The footer of the page says "Welcome to WordPress. This is your first post. Edit or delete it, then".

## 7. Create another Wordpress website and use the ALB created above to send traffic to this website based on the hostname.

Create another target group



Add instances to target group (ASG2)



Select load balancer and go to view and edit rules



In the rules, edit the rules and give the if and forward configuration

RULE ID	IF (all match)	THEN
1 arn:...:57206	Path is /wp-admin	1. Forward to SrimaASG2: 1 (100%) Group-level stickiness: Off

last action: HTTP 80: default  
IF Requests otherwise not routed

Now output with only ALB dns

Output with ALB/wp-admin

## 8. Use NLB that replaces the ALB in the above setup.

### Select Network Load balancer

#### Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs. [Learn more about which load balancer is right for you](#)

The screenshot shows a comparison of three load balancer types:

- Application Load Balancer:** Handles HTTP and HTTPS traffic. Description: 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.
- Network Load Balancer:** Handles TCP, TLS, and UDP traffic. Description: 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.
- Classic Load Balancer:** Previous generation for HTTP, HTTPS, and TCP. Description: Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network. [Learn more >](#)

### Select routing and healthcheck configurations

Step 3: Configure Routing

Target group

Target group <a href="#">i</a>	New target group <a href="#">▼</a>
Name <a href="#">i</a>	SrimaNLB-TG
Target type <a href="#">i</a>	<input checked="" type="radio"/> Instance <input type="radio"/> IP
Protocol <a href="#">i</a>	TCP <a href="#">▼</a>
Port <a href="#">i</a>	80

Health checks

Protocol <a href="#">i</a>	HTTP <a href="#">▼</a>
Path <a href="#">i</a>	/

Advanced health check settings

Port <a href="#">i</a>	<input checked="" type="radio"/> traffic port <input type="radio"/> override
Healthy threshold <a href="#">i</a>	3
Unhealthy threshold <a href="#">i</a>	3
Timeout <a href="#">i</a>	6 seconds

[Cancel](#) [Previous](#) [Next: Register Targets](#)

## Register targets in target group

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

## Step 4: Register Targets

To deregister instances, select one or more registered instances and then click Remove.

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-0fbe617c6782fdf9	SrimaASG1	80	<span style="color: green;">running</span>	Srima-ASG1	us-east-1b
<input type="checkbox"/>	i-0cab3007d7f09e466	SrimaASG1	80	<span style="color: green;">running</span>	Srima-ASG1	us-east-1b
<input type="checkbox"/>	i-03f92cc059874020e	SrimaASG1	80	<span style="color: green;">running</span>	Srima-ASG1	us-east-1e

**Instances**

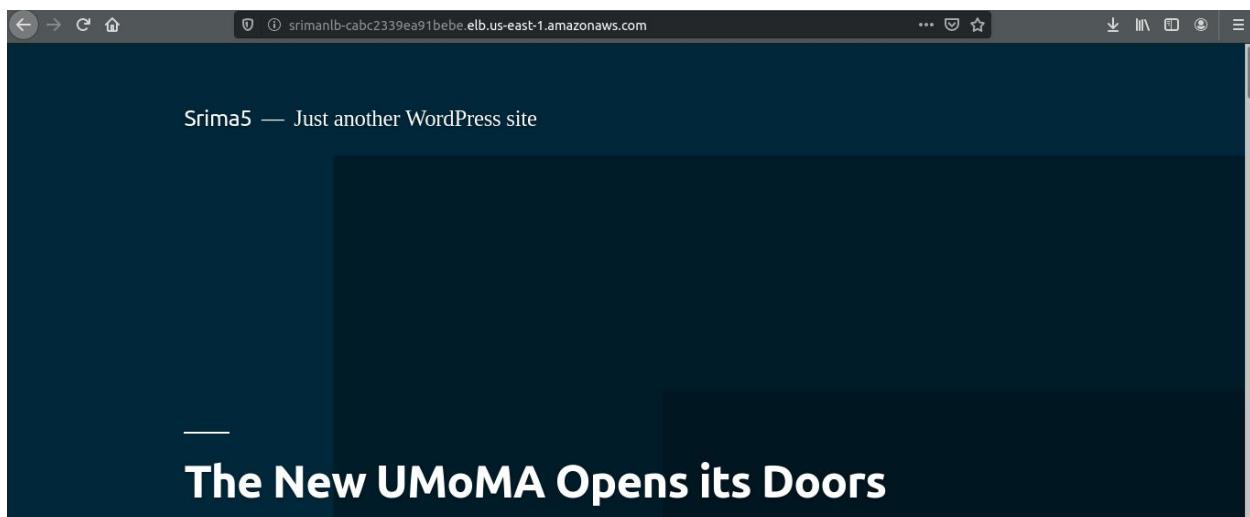
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

**Add to registered** on port **80**

**Cancel** **Previous** **Next: Review**

NLB Created

The screenshot shows the AWS Management Console with the AWS logo at the top left. The top navigation bar includes 'Services' (with a dropdown arrow), 'Resource Groups' (with a dropdown arrow), a search bar containing 'srima.pal@tothenew.com @ tt...', and account information for 'N. Virginia' and 'Support'. Below the navigation is a title 'Load Balancer Creation Status'. A green success message box contains a checkmark icon, the text 'Successfully created load balancer', and details about the newly created load balancer named 'SrimaNLB'. It also includes a note about setup time and a 'Close' button. A 'Suggested next steps' section lists two items: 'Discover other services...' and 'Consider using AWS Global Accelerator...'. The bottom right corner has a 'Close' button.



## 9. Take an instance out of the ASG.

Reduce minimum size of the auto scaling group

Go to instances pane, select the instances to be detached

## 10. Put scale-in protection on an instance in the ASG.

Select the ASG , go to instances, select the instance to enable scale in protection and then choose instance scale in

### Set scale in protection

## 11. Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM
- Launch a minimum of 2 instances at 10 AM

Select auto scaling group and go to scheduled policies

The screenshot shows the AWS EC2 Dashboard with the 'Resource Groups' dropdown open. Under 'INSTANCES', 'Instances' is selected. In the main pane, the 'Auto Scaling Groups' section is shown, with 'SrimaASG1' listed. The 'Scheduled Actions' tab is active. A modal window titled 'Create Scheduled Action' is overlaid on the main page, containing fields for 'Name' (Srima Scheduled Policy), 'Auto Scaling Group' (SrimaASG1), and various scheduling parameters.

### Schedule policy

The 'Create Scheduled Action' modal window is displayed. The 'Name' field is filled with 'Srima Scheduled Policy'. The 'Auto Scaling Group' dropdown is set to 'SrimaASG1'. The 'Min' field is '0', 'Max' is '0', and 'Desired Capacity' is '0'. The 'Recurrence' dropdown is set to 'Every day'. The 'Start Time' field shows '20:00 UTC' with a note: 'Specify the start time in UTC'. The 'End Time' field has a link 'Set End Time'. At the bottom right are 'Cancel' and 'Create' buttons.

EC2 Dashboard

- Events
- Tags
- Reports
- Limits

INSTANCES

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Scheduled Instances
- Capacity Reservations

IMAGES

- AMIs
- Bundle Tasks

ELASTIC BLOCK STORE

### Create Scheduled Action

Name: Srima Launch 2 instances

Auto Scaling Group: SrimaASG1

Provide at least one of Min, Max and Desired Capacity

Min: 2

Max: 5

Desired Capacity: 2

Recurrence: Every day

(Cron) 0 10 \* \* \*

Start Time: 2020-02-28 10:00 UTC Specify the start time in UTC  
The first time this scheduled action will run

End Time: Set End Time

[Cancel](#) [Create](#)

Scaling Groups > SrimaASG1

Health Check Grace Period: 300

Actions

Scheduled Actions >

Min	Max
0	0