

Project

Set Up a WordPress Instance for Your Organization

Project agenda: You are hired as a cloud architect in a global media company. You have been asked to set up a WordPress instance to publish blogs for your company per the defined specifications.

Description: Your organization publishes blogs and provides documentation services for other businesses and technologies. You have been asked to:

- Set up a live WordPress instance to publish blogs
- Set up a WordPress instance that can be used for development and testing purposes so that any work done on this instance will not impact the live blog
- Configure the WordPress instance for development and testing purposes, which will be available only for the business hours (9 AM–6 PM)
- Monitor the health of the WordPress instance

Tools required: AWS account

Prerequisites: Create a key pair

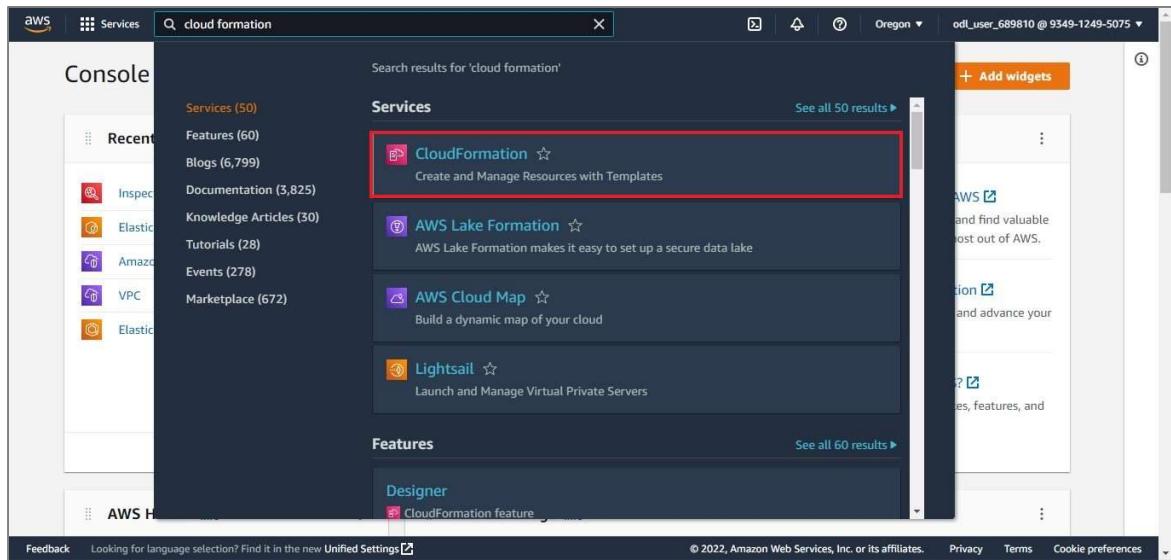
Expected Deliverables: WordPress instance that is operational between 9 AM-6 PM, and the instance health is continuously monitored.

Steps to be followed:

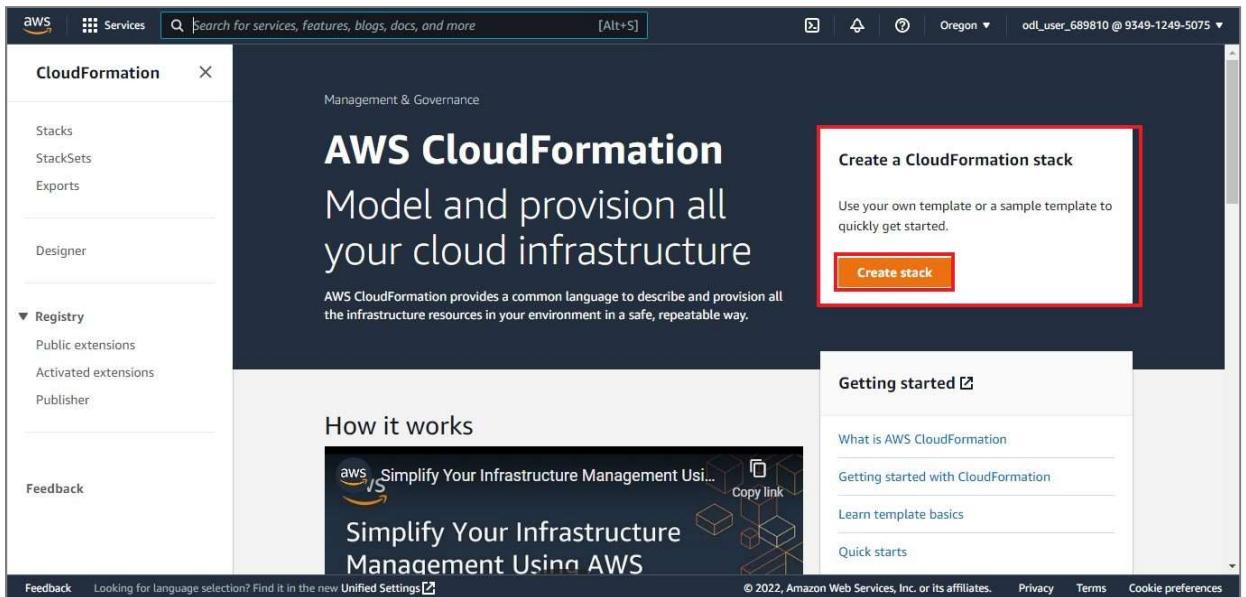
1. Create a CloudFormation stack
2. Create an AMI of the WordPress instance
3. Configure Auto Scaling to launch a new WordPress instance
4. Configure the new WordPress instance to shut down automatically
5. Monitor the instance using Availability Monitoring feature of the R53

Step 1: Create a CloudFormation new stack

- 1.1 From the AWS management console, search for **CloudFormation**, then click on **CloudFormation** from the search result



- 1.2 Select **Create New Stack**



1.3 In the **Create stack** section do the following:

- Choose **Select a sample template**
- In **Sample templates** choose **WordPress blog** from the drop down

The screenshot shows the AWS CloudFormation 'Create stack' wizard. On the left, a sidebar lists steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main area is titled 'Create stack'. Under 'Prerequisite - Prepare template', there are three options: 'Template is ready' (radio button), 'Use a sample template' (radio button, highlighted with a red box), and 'Create template in Designer'. Under 'Select a sample template', it says 'This collection of sample templates will help you get started with AWS CloudFormation and quickly build your own templates'. A dropdown menu shows 'WordPress blog' (highlighted with a red box). At the bottom right are 'View in Designer' and 'Next' buttons.

- Click on the **Next**

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The sidebar and main area are identical to the previous screenshot, but the 'Next' button at the bottom right is now highlighted with a red box.

1.4 In the **Create stack** section do the following:

- Enter an arbitrary name in the **Stack Name** field

Specify stack details

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Stack name
Stack name
Project1

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

DBName
The WordPress database name
wordpressdb

DBPassword
The WordPress database admin account password

- Enter an arbitrary value for **DBPassword**, **DBRootPassword**, **DBUser**
- Change the **Instance Type** to **t2.micro**
- Select **KeyName** from the drop-down

CloudFormation X

Stacks

StackSets

Exports

Designer

Registry

Public extensions

Activated extensions

Publisher

Feedback

DBPassword
The WordPress database admin account password
.....

DBRootPassword
MySQL root password
.....

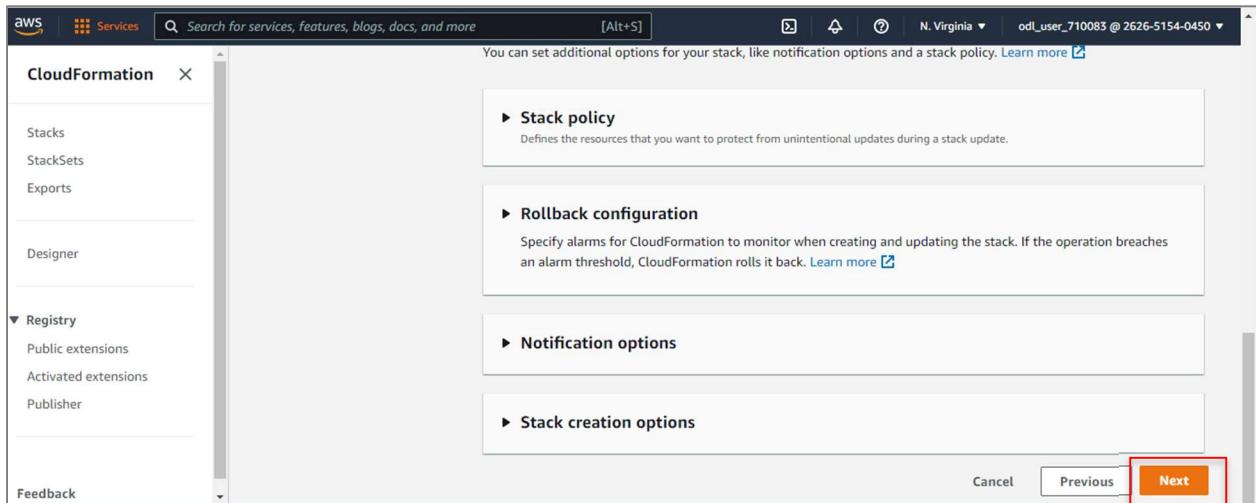
DBUser
The WordPress database admin account username
.....

InstanceType
WebServer EC2 instance type
t2.micro

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instances
Kp12345

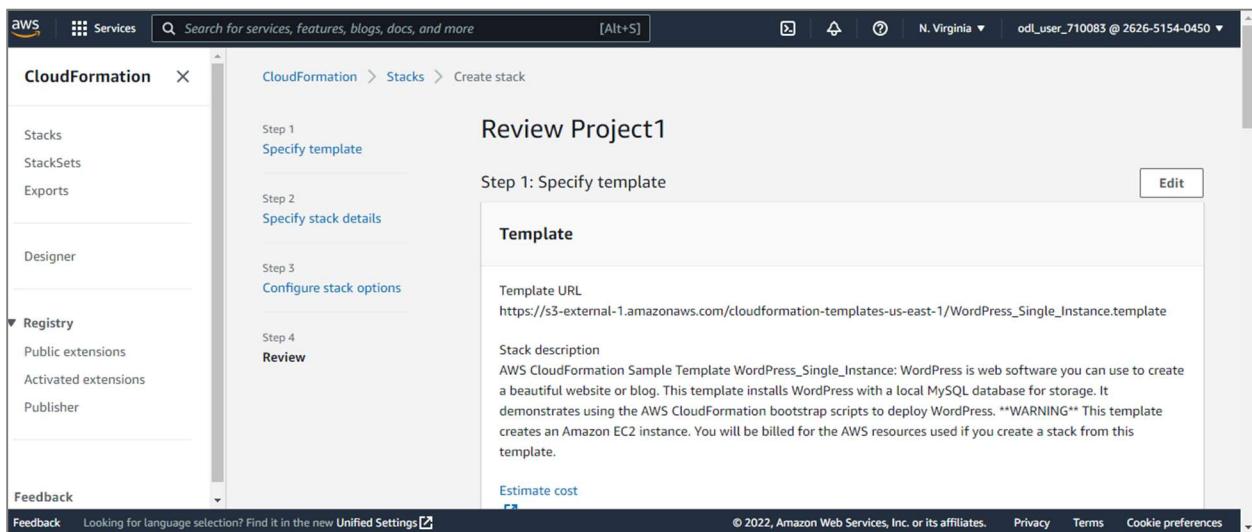
SSH location

- Click on the **Next**

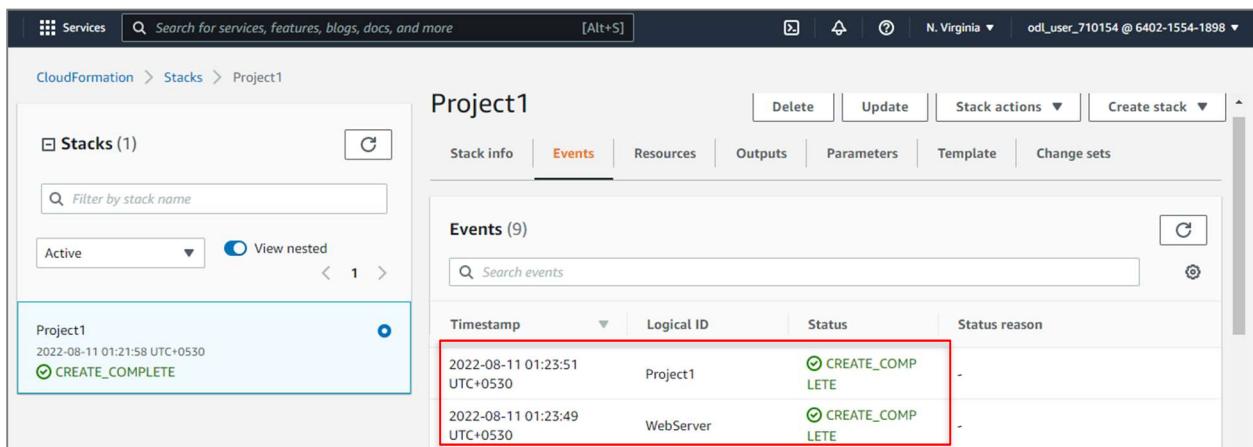


Similarly, click on Next in Configure stack options

1.5 Review the settings, and then click on the Create stack



1.6 Wait until the status of the stack changes from Pending to CREATE_COMPLETED



Step 2: Create an AMI of the WordPress instance

2.1 Switch to the EC2 dashboard and verify that your new instance is available for use

The screenshot shows the AWS EC2 Instances page. A single instance is listed with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
-	i-09f868b389ebb8021	Running	t2.micro	2/2 checks passed	No alarms

Note: Here we have renamed the instance created in step 1 to WordPress_Instance

2.2 Select the new instance then click on the Actions button and do the following:

- Go to **Image and templates** option
- Click on the **Create Image** button

The screenshot shows the AWS EC2 Instances page with the 'Actions' dropdown menu open. The 'Image and templates' and 'Create image' options are highlighted with red boxes.

2.3 Enter an arbitrary **Image Name** and **Image description**, and then click on the Create Image

The screenshot shows the 'Create image' configuration page. The 'Image name' field contains 'WordPress_AMI' and the 'Image description - optional' field contains 'this is wordpress image'.

Create image Info
An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-09f868b389ebb8021

Image name
WordPress_AMI

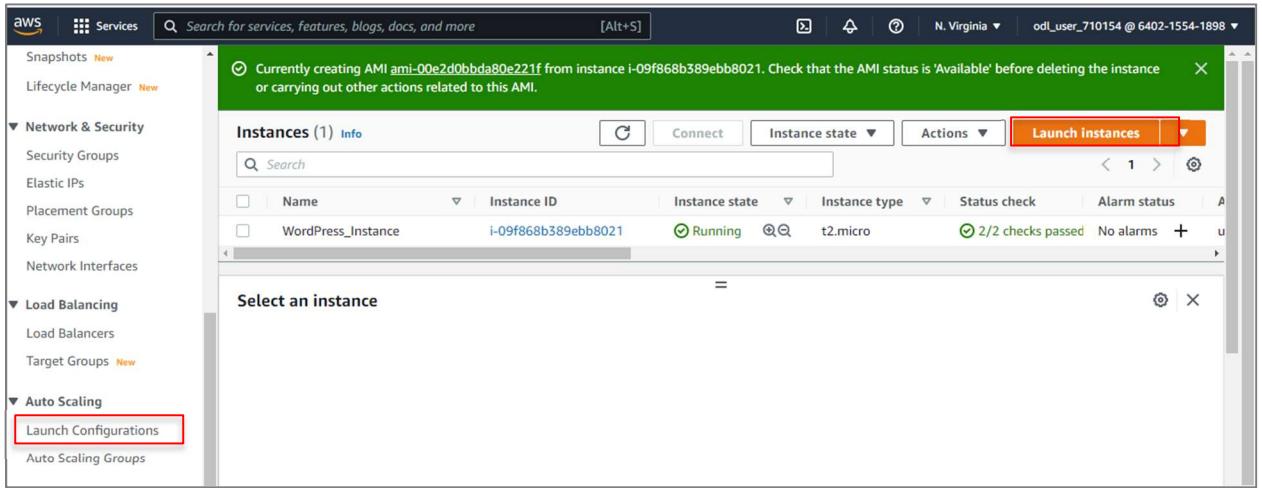
Image description - optional
this is wordpress image

No reboot
 Enable

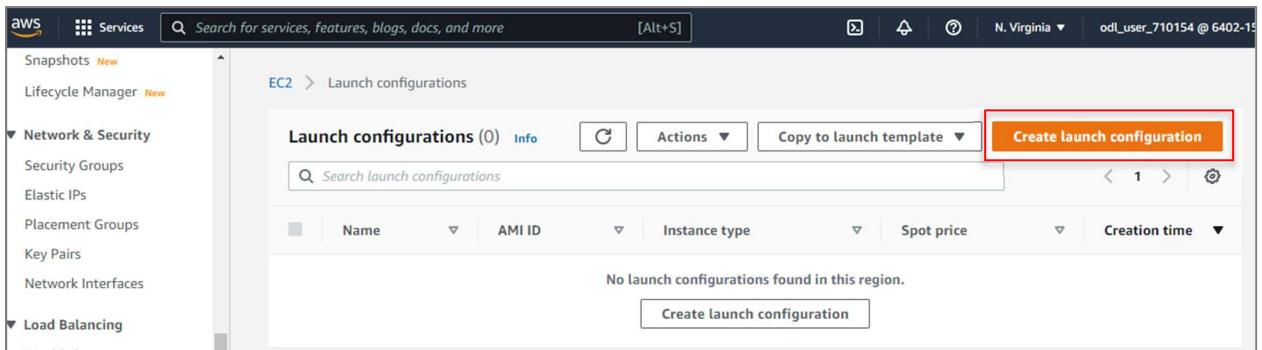
- Switch to the AMI dashboard and wait until the status of your new AMI changes from pending to available:

Step 3: Create launch configuration

3.1 Switch to Launch Configurations



Click on the **Create launch configuration**



3.2 In the **Create launch configuration**, do the following:

- Enter an arbitrary name for the launch configuration
- In AMI field select the AMI created in step 2

Screenshot of the AWS EC2 'Create launch configuration' page.

The 'Launch configuration name' field contains 'launch_config_1' and is highlighted with a red box.

The 'AMI' dropdown menu shows 'WordPress_AMI' and is highlighted with a red box.

3.3 In the security group, select Create security group

Screenshot of the AWS Security Groups 'Create security group' page.

The 'Security group name' field contains 'AutoScaling-Security-Group-1'.

The 'Description' field contains 'AutoScaling-Security-Group-1 (2022-08-10T20:06:01.075Z)'.

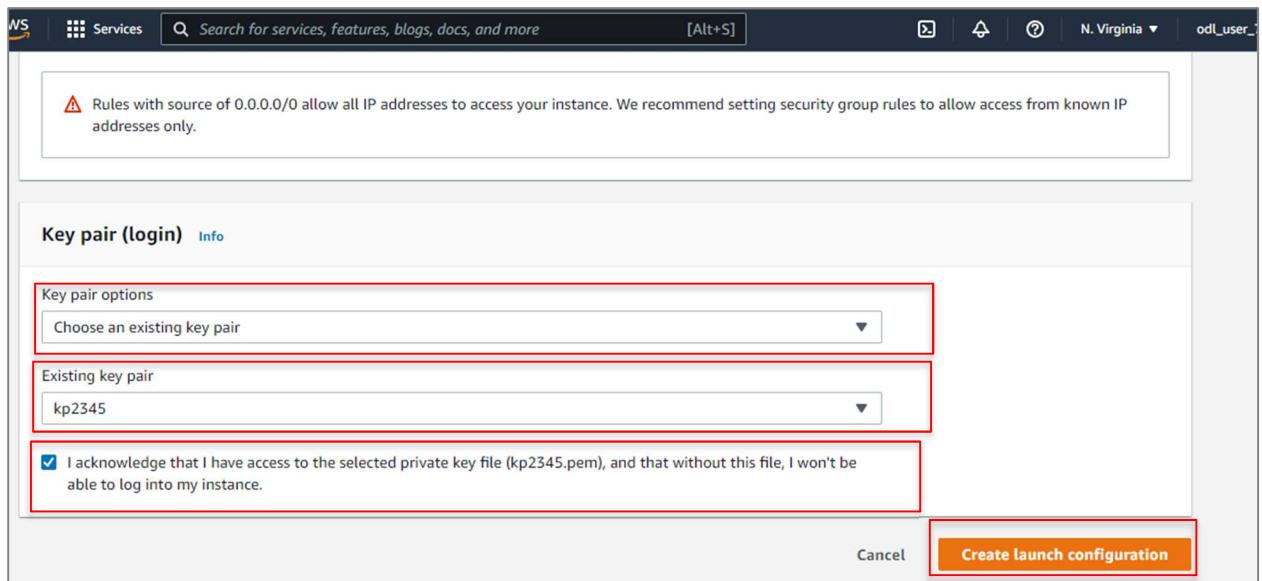
The 'Rules' section shows one rule:

Type	Protocol	Port range	Source type	Source
SSH	TCP	22	Anywhere	0.0.0.0/0

A note 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."

3.4 In the **Key pair(login)**, do the followings:

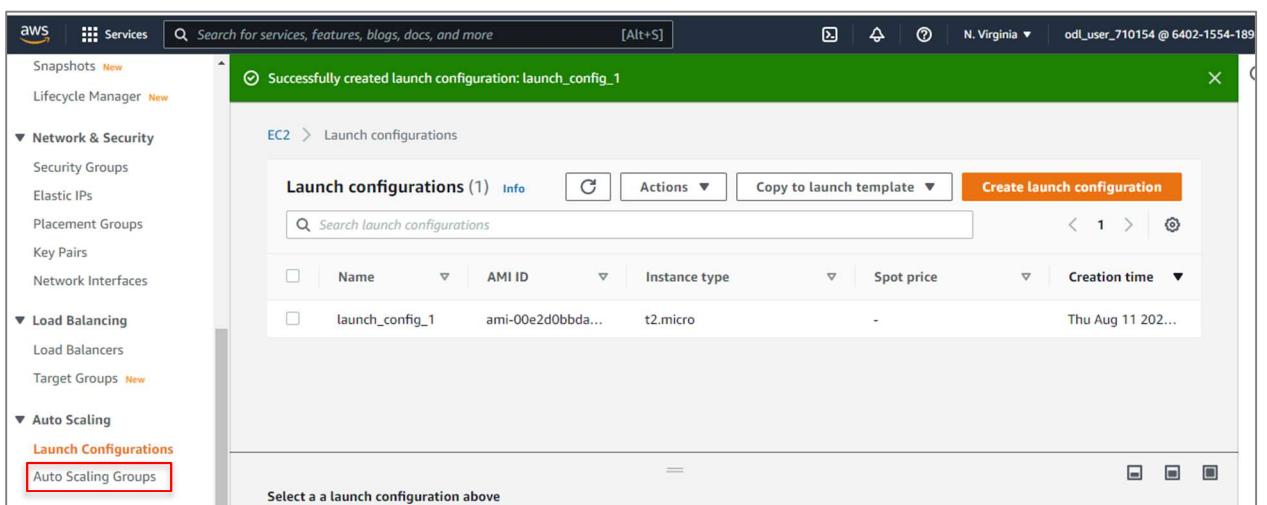
- Select **Choose an existing key pair** and select the **key pair**
- Click on the **Acknowledgement**
- Click in **Create launch configuration**



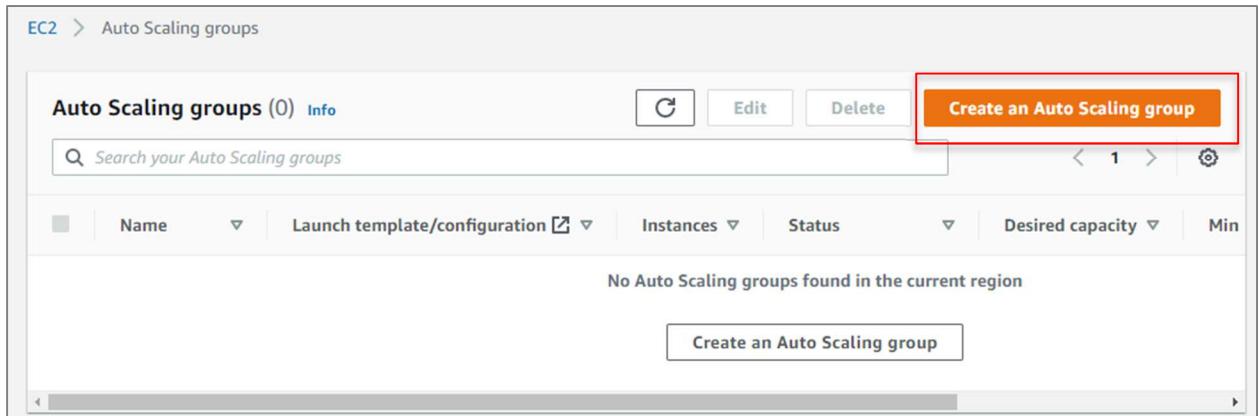
You can now use the new launch configuration to create a new WordPress instance
during the working hours (9 AM - 6 PM)

Step 4: Configure the new WordPress instance to shut down automatically

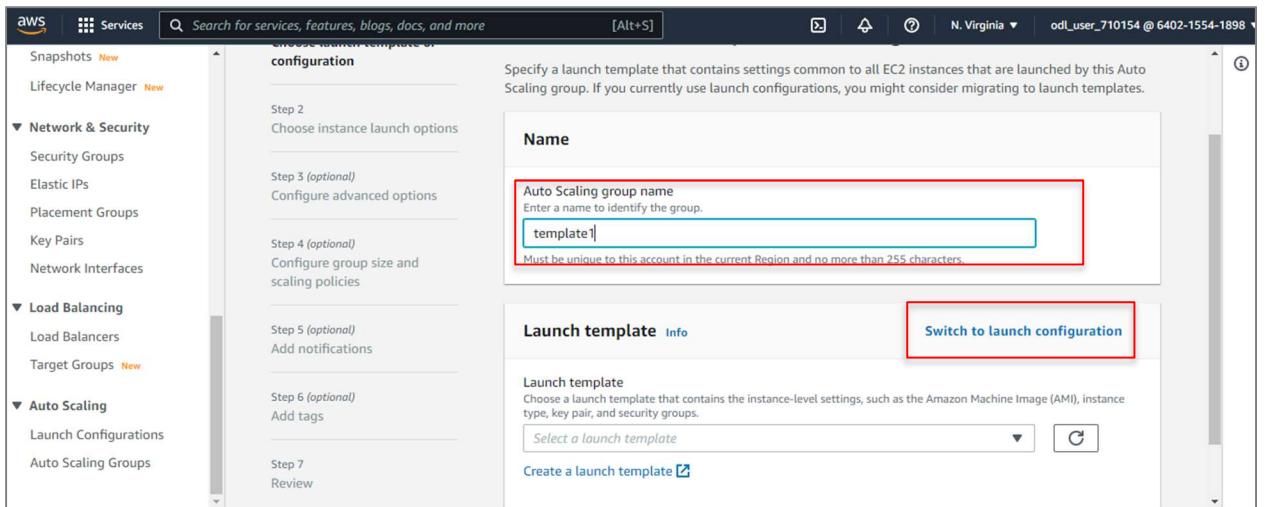
4.1 Click on **Create Auto Scaling group** from the left pane of the Launch configurations



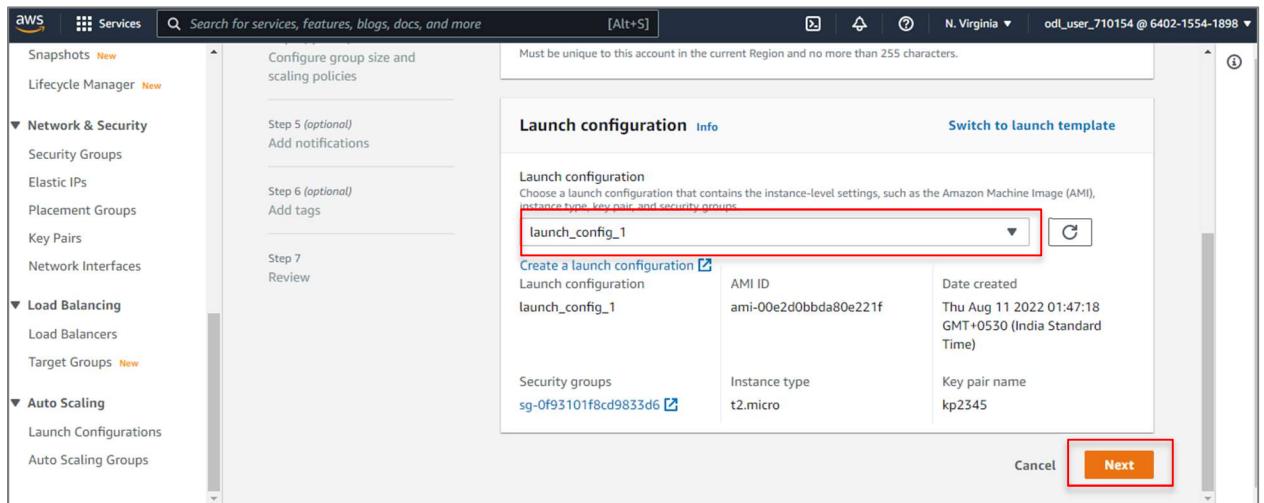
4.2 In the **Auto Scaling Dashboard** click on **Create an Auto Scaling group:**



4.3 In the **Create Auto Scaling group section** add an arbitrary name for the auto scaling Group, and click on **Switch to launch configuration**

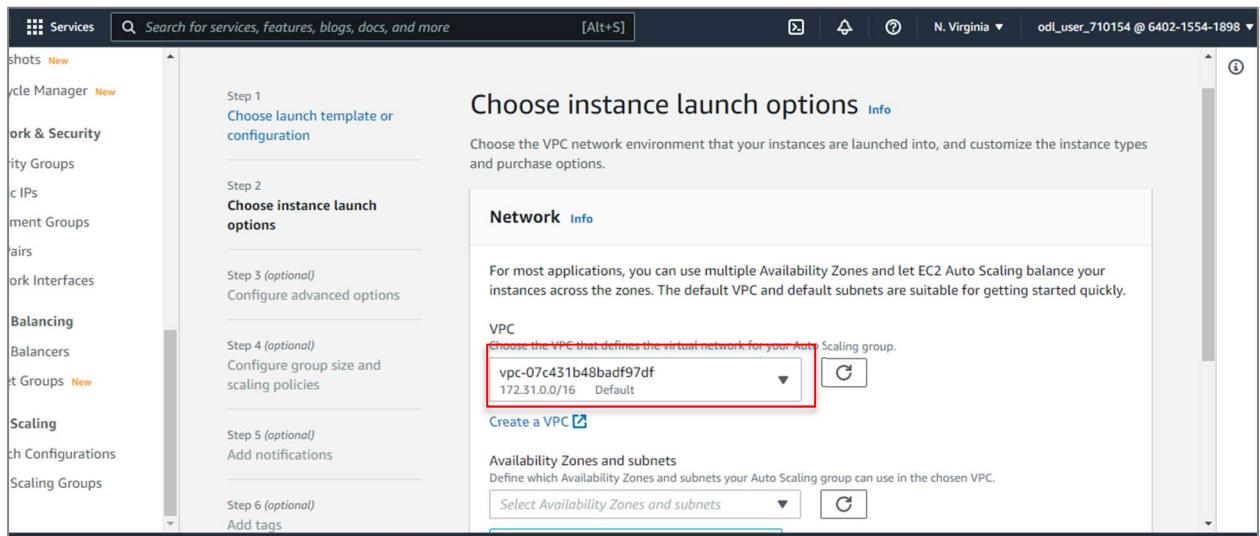


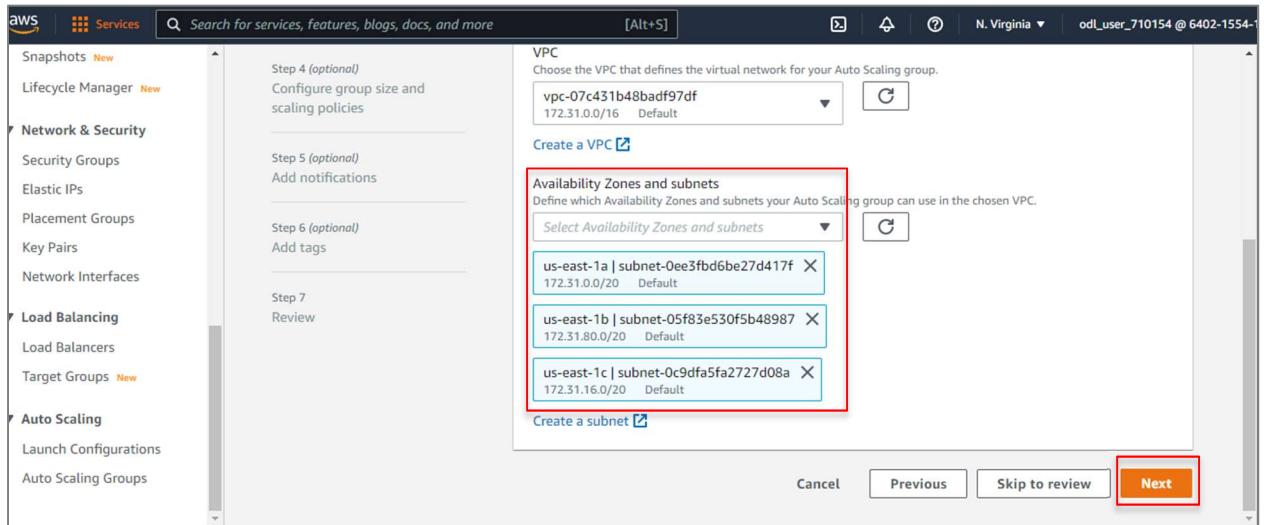
- Select the launch configuration created in step 3
- Click on **Next**



4.4 In the **Choose instance launch options** do the following:

- Select the **default VPC**
- Select all the **availability zones and subnets** then click on **Next**
- Again, click on **Next**:





4.5 In the Configure advanced options select No load balancer:

Configure advanced options

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

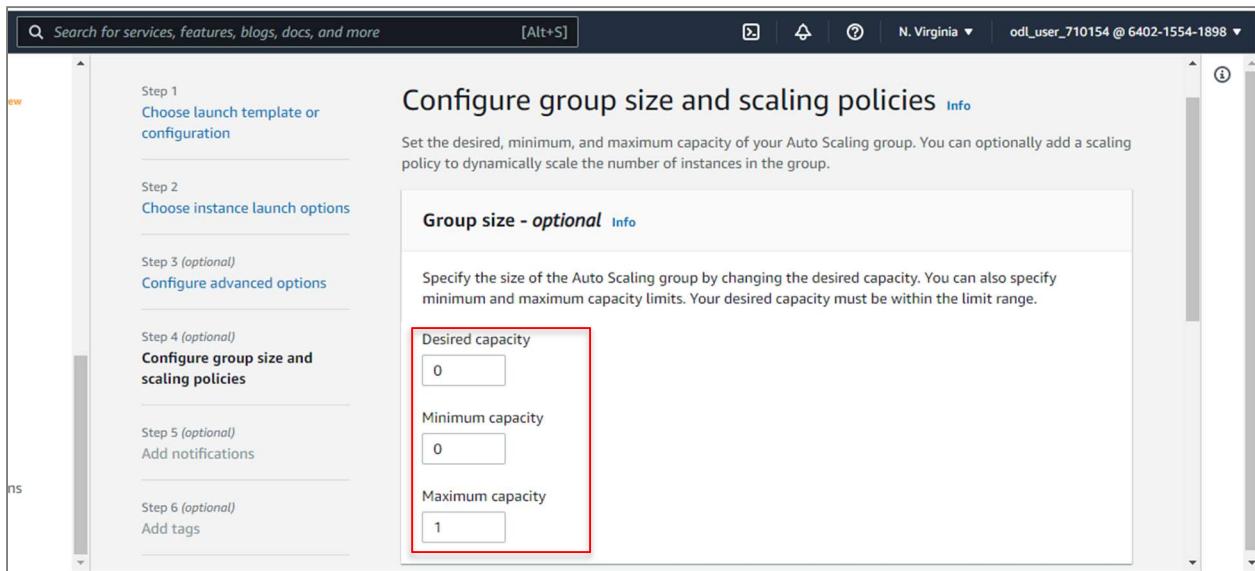
Load balancing - optional

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

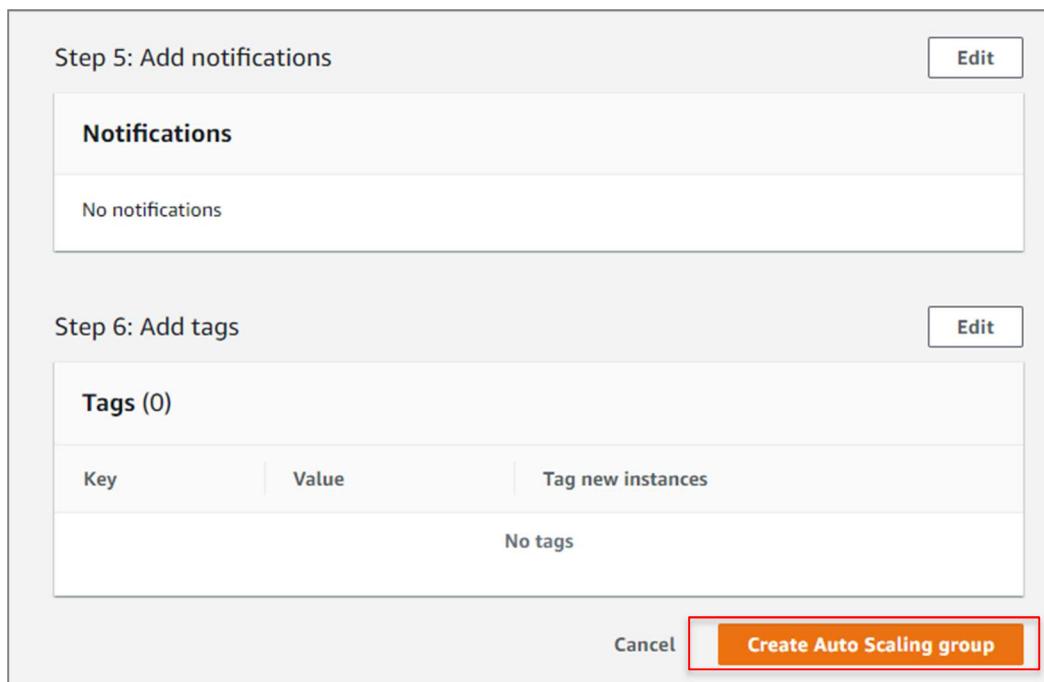
- No load balancer**
Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer
Choose from your existing load balancers.
- Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

4.6 In the Configure group size and scaling policies do the following:

Desired capacity = 0, Minimum capacity = 0 and, Maximum capacity = 1 then click on Next:



4.7 Now skip all the section by clicking on **Next** and then finally click on **Create Auto Scaling group**:



Finally, you will see the **AutoScalingGroup1** in **Auto Scaling group Dashboard** which indicates that the Auto Scaling group has been launched successfully:

The screenshot shows the AWS Auto Scaling Groups dashboard. At the top, a green success message box displays "template1 created successfully". Below it, the main table header is "Auto Scaling groups (3) Info". The table has columns: Name, Launch template/configuration, Instances, Status, Desired capacity, and Min. A single row is visible: "template1" with "launch_config_1", "0" instances, and "0" for both desired capacity and min. The "Load Balancing" section in the left sidebar is highlighted with a red box.

4.8 In the auto scaling group dashboard, click on the **Scheduled Actions** tab, and then click on

Create Scheduled Action to create the scheduled actions

The screenshot shows the same AWS Auto Scaling Groups dashboard as above, but now with the "Scheduled actions" tab selected. The main table header is "Scheduled actions (0) Info". The "Actions" button in the top right of the table area is highlighted with a red box. The "Create scheduled action" button, located just below the table, is also highlighted with a red box.

4.9 In the Create schedule action do the following

- Enter a **Name**, for example, **SCALEUP_9AM**
- set the **Desired Capacity** as **one**, and then set the time when you want the job to run

Create scheduled action

Name
SCALEUP_9AM

i Provide at least one value for Desired, Min, or Max Capacity

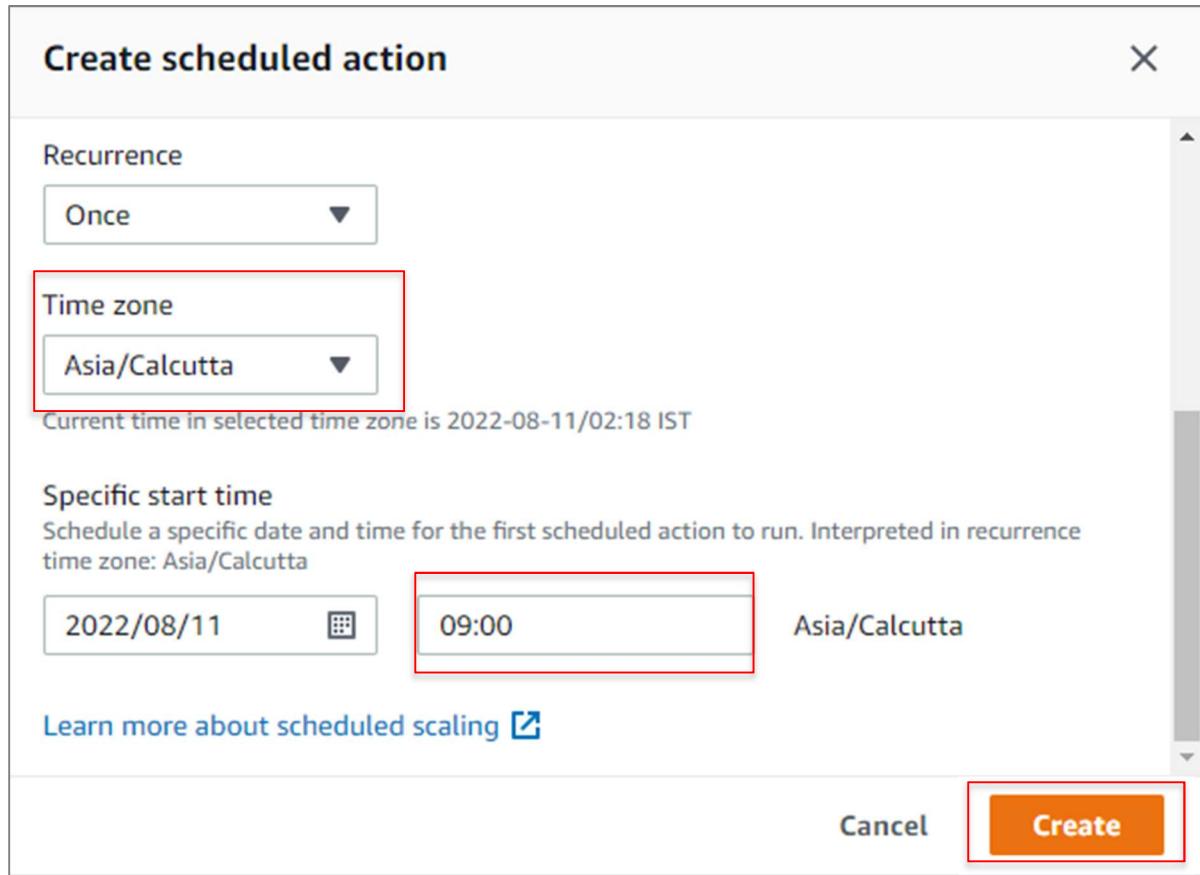
Desired capacity
1 Min Max

Recurrence
Once ▾

Cancel Create

The screenshot shows a modal dialog titled "Create scheduled action". Inside, there's a "Name" input field containing "SCALEUP_9AM", which is also highlighted with a red border. Below it is a note: "(i) Provide at least one value for Desired, Min, or Max Capacity". Under "Capacity", there are three input fields: "Desired capacity" (containing "1", also highlighted with a red border), "Min", and "Max". Under "Recurrence", there's a dropdown set to "Once". At the bottom right are two buttons: "Cancel" and a prominent orange "Create" button, which is also highlighted with a red border.

- Change the time zone to **Asia/Calcutta**
- Once complete, click on the **Create**



Similarly, create another schedule action.

4.10 In the Create schedule action do the following

- Enter a **Name**, for example, **SCALEDOWN_6PM**
- set the **Desired Capacity** as **one**, and then set the time when you want the job to run

Create scheduled action

Name
SCALEDOWN_6PM

Desired capacity 1 **Min** **Max**

Recurrence
Once ▾

Cancel **Create**

ⓘ Provide at least one value for Desired, Min, or Max Capacity

- Change the time zone to **Asia/Calcutta**
- Once complete, click on the **Create**

Create scheduled action

Recurrence

Once

Time zone

Asia/Calcutta

Current time in selected time zone is 2022-08-11/02:21 IST

Specific start time

Schedule a specific date and time for the first scheduled action to run. Interpreted in recurrence time zone: Asia/Calcutta

2022/08/11 06:00 Asia/Calcutta

Learn more about scheduled scaling [↗](#)

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

4.11 Verify that the actions have been created successfully

template1 created successfully

EC2 > Auto Scaling groups

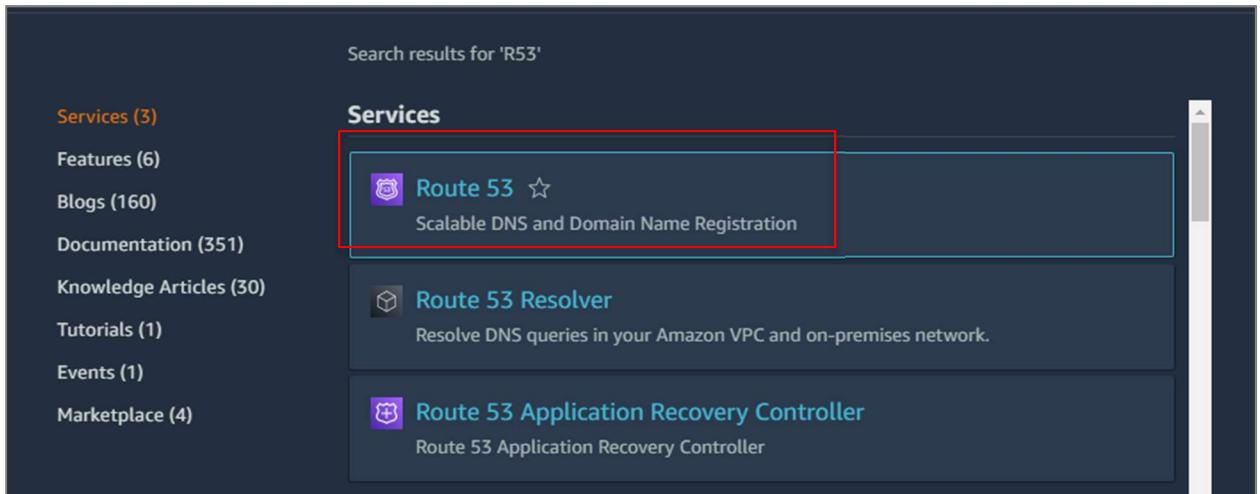
Name	Launch template/configuration	Instances	Status	Desired capacity
template1	launch_config_1	0	-	0

Filter scheduled actions

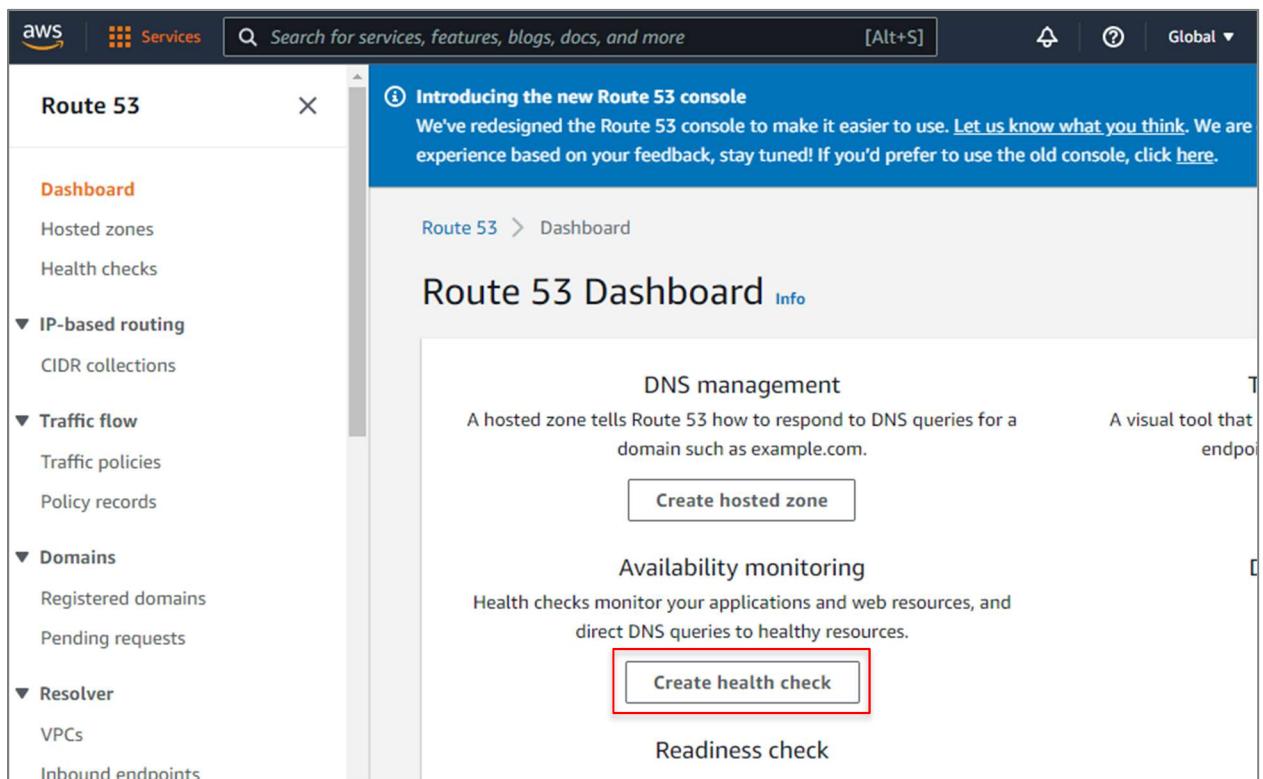
Name	Start time	End time	Recurrence	Time zone	Desired capac...	Min
SCALEDOWN...	2022 Augu...			Asia/Calcutta	1	
SCALEUP...	2022 Augu...			Asia/Calcutta	1	

Step 5: Monitoring the Solution using Availability Monitoring feature of the R53

5.1 In the AWS management console search for Route and then click on **Route 53** from the search results:



5.2 In the **Route 53 Dashboard** click on **Create health check** under **Availability monitoring**:



5.3 In the **Configure health check** console do the following:

- Enter an **arbitrary name** in the **Name** column
- Select **Domain name** in **Specify endpoint by**
- Enter the **Domain name** of your website
- Enter **80** in **Port**:

Note: User's can add their own website domain name if they have and we have used **www.google.com** for this course end project.

Configure health check

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name i

What to monitor Endpoint i
 Status of other health checks (calculated health check)
 State of CloudWatch alarm

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy.
[Learn more](#)

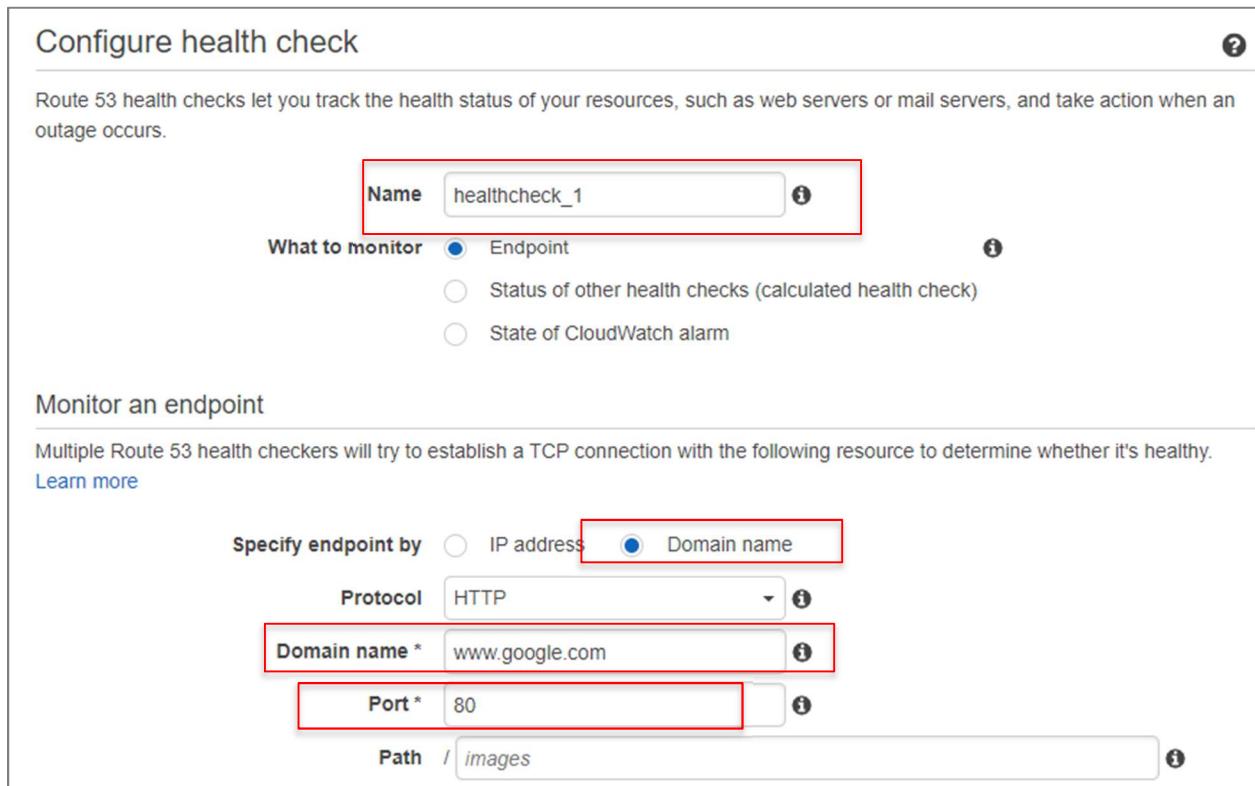
Specify endpoint by IP address Domain name i

Protocol i

Domain name * i

Port * i

Path / i



- Now click on **Next**:

Port * 80

Path /images

Advanced configuration

URL http://www.google.com:80/

Health check type Basic - no additional options selected ([View Pricing](#))

* Required

Cancel **Next**

5.4 In **Get notified when health check fails** section do the following:

- Select **Yes** for **Create alarm**
- Select **New SNS topic** in **Send notification to**
- Enter an arbitrary name in **Topic name**
- Enter your email address where you want to receive the notification in **Recipient email address**
- Click on **Create health check**:

Create health check

Step 1: Configure health check

Step 2: Get notified when health check fails

Get notified when health check fails

If you want CloudWatch to send you an Amazon SNS notification, such as an email, when the status of the health check changes to unhealthy, create an alarm and specify where to send notifications.

Create alarm Yes No

CloudWatch sends you an Amazon SNS notification whenever the status of this health check is unhealthy for at least one minute. The alarm will be located in the **us-east-1** region.

Send notification to Existing SNS topic New SNS topic

Topic name * topic1

Recipient email addresses * prakhar.gupta@simplilearn.net

Separate multiple addresses with a comma, a semicolon, or a space

* Required

Cancel Previous **Create health check**

After Successful creation of the health check, you will see it in your health check

Dashboard:

The screenshot shows the AWS Health Checks service dashboard. A success message at the top states: "Health check with id 4637e5c3-79f8-48e5-90de-798b0e5a4f5c has been created successfully". Below this, a table lists one health check entry:

Name	Status	Description	Alarms	ID
healthcheck_1	Unknown	http://www.google.com:80/	⚠ 1 of 1 in INSUFFICIE...	46:

The "Status" column for the health check is currently "Unknown".

Initially the **Status** of the health check will be unknown, then refresh it after 2 mins the

Status will change:

The screenshot shows the AWS Health Checks service dashboard after some time has passed. The same success message is present. The table now shows the health check status has changed:

Name	Status	Description	Alarms	ID
healthcheck_1	Healthy	16 minutes ago now	✓ 1 of 1 in OK	46:

The "Status" column for the health check is now "Healthy".

The above image shows that the domain www.google.com is healthy, similarly you can check for your domain also.