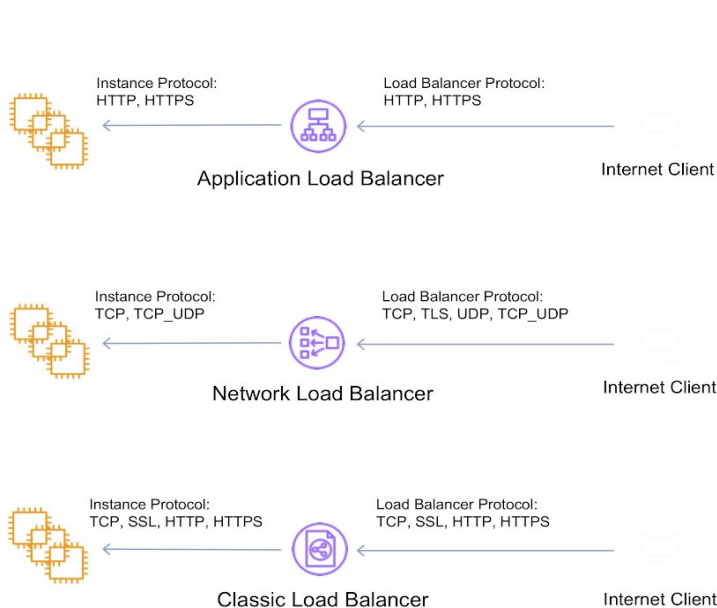


1.Differences between ELB, ALB, and NLB. Where will you use which one?



Application Load Balancer

- Operates at the request level
- Routes based on the content of the request (layer 7)
- Supports path-based routing, host-based routing, query string parameter-based routing, and source IP address-based routing
- Supports IP addresses, Lambda Functions and containers as targets

Network Load Balancer

- Operates at the connection level
- Routes connections based on IP protocol data (layer 4)
- Offers ultra high performance, low latency and TLS offloading at scale
- Can have static IP / Elastic IP
- Supports UDP and static IP addresses as targets

Classic Load Balancer

- Old generation; not recommended for new applications
- Performs routing at Layer 4 and Layer 7
- Use for existing applications running in EC2-Classic

2.Differences between step scaling and target scaling.

(SIMPLE SCALING) You pick ANY Cloud Watch metric For this and other examples in THIS POST I am choosing CPU Utilization You specify, a SINGLE THRESHOLD beyond which you want to scale and specify your response **EXAMPLE:** how many EC2 instances do you want to add or take away when the CPU UTILIZATION breaches the threshold. The scaling policy then acts. **THRESHOLD - add 1 instance when CPU Utilization is between 40% and 50% NOTE: This is the ONLY Threshold**

(STEP SCALING) You specify MULTIPLE thresholds Along with different responses. **Threshold A - add 1 instance when CPU Utilization is between 40% and 50% Threshold B - add 2 instances when CPU Utilization is between 50% and 70% Threshold C - add 3 instances when CPU Utilization is between 70% and 90% And so on and so on Note: There are multiple thresholds**

(TARGET TRACKING SCALING) You don't want to have to make so many decisions Makes the experience simple as compared to the previous 2 scaling options It's automatic All you do is pick CPU Utilization(Your metric and example for this post) Set the value and that's it Auto scaling does the rest adding and removing the capacity in order to keep your metric(CPU utilization) as close as possible to the target value

3.Differences between Launch configuration and launch template.

Launch templates allow control of the T2 unlimited (credits) feature, at least that is the one feature I needed, so had to use launch templates.

However, it seems that a launch template specifies exactly one subnet, rather than a list (like launch configurations).

They're quite different and CloudFormation and it has sucked up several hours of trial and error on my part to get them working.

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

EC2 health check watches for instance availability from hypervisor and networking point of view. For example, in case of a hardware problem, the check will fail. Also, if an instance was misconfigured and doesn't respond to network requests, it will be marked as faulty.

ELB health check verifies that a specified TCP port on an instance is accepting connections OR a specified web page returns 2xx code. Thus ELB health checks are a little bit smarter and verify that actual app works instead of verifying that just an instance works.

That being said there is a third check type: custom health check. If your application can't be checked by simple HTTP request and requires advanced test logic, you can implement a custom check in your code and set instance health through API: Health Checks for Auto Scaling Instances

5. Create 2 auto-scaling groups with

- launch configuration and

Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

▼ Ownership

Owned by me

ayush

ayush-wordpress - ami-023c44c73c2a6763f

Root device type: ebs Virtualization type: hvm Owner: 187632318301

Select

64-bit

Create Launch Configuration

Name ayush-LC-wp

Purchasing option ☐ Request Spot Instances

IAM role None

Monitoring ☐ Enable CloudWatch detailed monitoring
[Learn more](#)

▼ Security Groups

| Security Group ID | Name | Description |
|----------------------|---------------|--------------------------------------------------------|
| sg-0d2f62d8a7b2177ec | ayush-sg-test | launch-wizard-96 created 2020-02-21T10:30:23.189+05:30 |

All selected security groups inbound rules

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

Edit

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

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.](#)

Choose an existing key pair

Select a key pair

ayush-ec2

☒ I acknowledge that I have access to the selected private key file (ayush-ec2.pem), and that without this file, I won't be able to log into my instance.

Cancel

Create launch configuration

- launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto scaling guidance [Info](#)

Select this if you intend to use this template with auto scaling

☐ Provide guidance to help me set up a template that I can use with auto scaling

► Template tags

► Source template

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.

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

Amazon machine image (AMI) [Info](#)

AMI

ayush-wordpress
ami-023c44c73c2a6763f
Catalog: My AMIs architecture: 64-bit (x86) virtualization: hvm

Instance type [Info](#)

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

Key pair (login) [Info](#)

Key pair name

ayush-ec2



[EC2](#) > [Launch templates](#) > Create launch template



Success

Successfully created **ayush-LT** (lt-0f0de362b4c2b47c2)

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

Create Auto Scaling Group

[Cancel and Exit](#)

Complete this wizard to create your Auto Scaling group. First, choose either a launch configuration or a launch template to specify the parameters that your Auto Scaling group uses to launch instances.

Launch Configuration

You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)

[Create a new launch configuration](#)

Launch Template

[New](#)

Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned. [Learn more](#)

[Create new launch template](#)



| Name | Launch Template Id | Default Version | Latest Version | Create Time | Created by |
|----------|----------------------|-----------------|----------------|----------------------------------|------------------------------------------------------------|
| ayush-LT | lt-0f0de362b4c2b47c2 | 1 | 1 | Sun Mar 01 10:15:08 GMT+530 2020 | arn:aws:iam::187632318301:user/ayush.tripathi@tothenew.com |

[Cancel](#) [Next Step](#)

Create Auto Scaling Group

Group name ⓘ

ayush-asg-LT-wp

Launch Template ⓘ

lt-0f0de362b4c2b47c2

Launch Template Version ⓘ

1 (Default)



[Create new launch template](#)

Launch Template Description ⓘ

for assignment

Fleet Composition

☒ Adhere to the launch template

The launch template determines the instance type and purchase option (On-Demand or Spot).

☐ Combine purchase options and instances

Choose a mix of On-Demand Instances and Spot Instances and multiple instance types. Spot Instances are automatically launched at the lowest price available.

Group size ⓘ

Start with 1 instances

Network ⓘ

vpc-d38d68b7 (172.31.0.0/16) | default (default)



[Create new VPC](#)

Subnet ⓘ

[Create new subnet](#)

Create Alarm



You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

☒ Send a notification to: [cancel](#)

With these recipients:

Whenever: Average ▼ of CPU Utilization ▼

Is: >= ▼ Percent

For at least: consecutive period(s) of 5 Minutes ▼

Name of alarm:

CPU Utilization Percent



☒ ayush-asg-LT-wp

[Cancel](#)

[Create Alarm](#)

Create Alarm



You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

☒ Send a notification to: [create topic](#)

Whenever: Average ▼ of CPU Utilization ▼

Is: < ▼ Percent

For at least: consecutive period(s) of 5 Minutes ▼

Name of alarm:

CPU Utilization Percent



☒ ayush-asg-LT-wp

[Cancel](#)

[Create Alarm](#)

Create Auto Scaling Group

Increase Group Size

Name:

Execute policy when: [awsec2-ayush-asg-LT-wp-CPU-Utilization](#) [Edit](#) [Remove](#)
breaches the alarm threshold: CPUUtilization >= 60 for 300 seconds
for the metric dimensions AutoScalingGroupName = ayush-asg-LT-wp

Take the action:

And then wait: seconds before allowing another scaling activity

[Create a scaling policy with steps](#) ⓘ

Decrease Group Size

Name:

Execute policy when: [awsec2-ayush-asg-LT-wp-High-CPU-Utilization](#) [Edit](#) [Remove](#)
breaches the alarm threshold: CPUUtilization < 40 for 300 seconds
for the metric dimensions AutoScalingGroupName = ayush-asg-LT-wp

Take the action: when >= CPUUtilization > -infinity

[Add step](#) ⓘ

[Create a simple scaling policy](#) ⓘ

Create Auto Scaling Group

Configure your Auto Scaling group to send notifications to a specified endpoint, such as an email address, whenever a specified event takes

If you created a new topic, check your email for a confirmation message and click the included link to confirm your subscription. Notifications

Send a notification to: [create topic](#)

Whenever instances:

- ☒ launch
- ☒ terminate
- ☒ fail to launch
- ☒ fail to terminate

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 | ayush-asg-LT-wp |
| Launch Template | lt-0f0de362b4c2b47c2 |
| Launch Template Version | 1 |
| Launch Template Description | for assignment |
| Group size | 1 |
| Minimum Group Size | 1 |
| Maximum Group Size | 1 |
| Subnet(s) | subnet-06680a5b651f104dc |
| Health Check Grace Period | 300 |
| Detailed Monitoring | No |
| Instance Protection | None |
| Service-Linked Role | AWSServiceRoleForAutoScaling |

▼ Scaling Policies

| | |
|---------------------|--------------------------------------------------------------------------------------------------------------|
| Increase Group Size | With alarm = awsec2-ayush-asg-LT-wp-CPU-Utilization; Add 1 capacity units and 300 seconds between activities |
| Decrease Group Size | With alarm = awsec2-ayush-asg-LT-wp-High-CPU-Utilization; Remove 1 capacity units |

▼ Notifications

| | |
|-----------------------------------------------|------------------------------------------------------|
| ayush-asg-wp (ayush.tripathi@tothenew.com) | launch, terminate, fail to launch, fail to terminate |
|-----------------------------------------------|------------------------------------------------------|

▼ Tags

| | | |
|-------|--------------|-------------------|
| Name | ASG-WP-ayush | tag new instances |
| Owner | ayush | tag new instances |

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

Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers.

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 >](#)

Step 1: Configure Load Balancer

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 |
|-------------------------|--------------------|
| <div>HTTP</div> | <div>80</div> |
| <div>Add listener</div> | |

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.

| | |
|--------------------|----------------------------------------------------------------------------------------------|
| VPC | <div>vpc-d38d68b7 (172.31.0.0/16) default (default)</div> |
| Availability Zones | <div><input checked="" type="checkbox"/> us-east-1c subnet-06680a5b651f104dc (default)</div> |
| IPv4 address | Assigned by AWS |

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets

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

Remove

| <input type="checkbox"/> | Instance | Name | Port | State | Security groups | Zone |
|--------------------------|---------------------|----------|------|----------------------|-----------------|------------|
| <input type="checkbox"/> | i-05b1a8c786f575a57 | ayush-SG | 80 | running | ayush-http | us-east-1b |

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

Search Instances

X

| <input type="checkbox"/> | Instance | Name | State | Security groups | Zone | Subnet ID | Subnet CIDR |
|-------------------------------------|---------------------|----------|--------------------|-----------------|------------|--------------------------|-------------|
| <input checked="" type="checkbox"/> | i-05b1a8c786f575a57 | ayush-SG | <div>running</div> | ayush-http | us-east-1b | subnet-078f8e416344e3d9d | 10.0.2.0/24 |

Load Balancer Creation Status



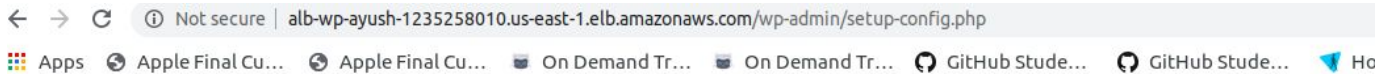
Successfully created load balancer

Load balancer **ALB-WP-Ayush** 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.

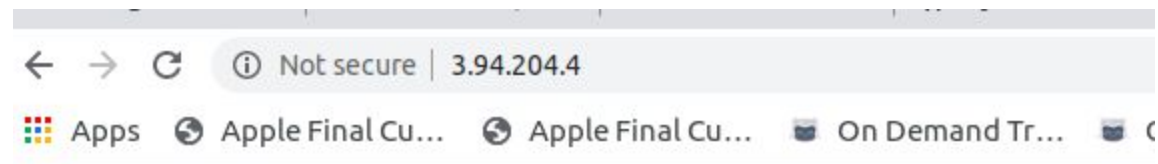
Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within **ALB-WP-Ayush**
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

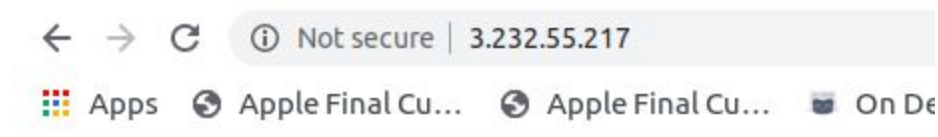


7.Create another Wordpress website and use the ALB created above to send traffic to this website based on the hostname(PATH based routing)

Create two instance and attach those instance to two target groups



this is PAGE~ 1



this is PAGE~ 1 this is the second page appended

Create target group

Actions

search : ayush

Add filter

| | Name | Port | Protocol | Target type | Load Balan | VPC ID | Monitoring |
|-------------------------------------|--------------|------|----------|-------------|-------------|-----------------------|------------|
| <input type="checkbox"/> | ayush-ALB-TG | 80 | HTTP | instance | ALB-WP-A... | vpc-0bce5df601296bb8a | |
| <input type="checkbox"/> | ayush-ALB-1 | 80 | HTTP | instance | | vpc-0bce5df601296bb8a | |
| <input checked="" type="checkbox"/> | ayush-ALB-2 | 80 | HTTP | instance | | vpc-0bce5df601296bb8a | |

Target group: ayush-ALB-2

Description

Targets

Health checks

Monitoring

Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

| Instance ID | Name | Port | Availability Zone | Status | Description |
|---------------------|---------------|------|-------------------|--------|---------------------------------------------|
| i-0ba25c512bf099848 | ayush-nginx-2 | 80 | us-east-1b | unused | Target group is not configured to receive t |

Create target group

Actions

search : ayush

Add filter

| | Name | Port | Protocol | Target type | Load Balanc | VPC ID |
|-------------------------------------|--------------|------|----------|-------------|-------------|-----------------------|
| <input type="checkbox"/> | ayush-ALB-TG | 80 | HTTP | instance | ALB-WP-A... | vpc-0bce5df601296bb8a |
| <input checked="" type="checkbox"/> | ayush-ALB-1 | 80 | HTTP | instance | | vpc-0bce5df601296bb8a |
| <input type="checkbox"/> | ayush-ALB-2 | 80 | HTTP | instance | | vpc-0bce5df601296bb8a |

Target group: ayush-ALB-1

Description

Targets

Health checks

Monitoring

Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health check. You can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

| Instance ID | Name | Port | Availability Zone | Status | Description |
|-------------------------------------|-------------|------|-------------------|--------|--------------------------------|
| i-032624a89dbf54d13 | nginx-ALB-1 | 80 | us-east-1a | unused | Target group is not configured |

Availability Zones

< Rules

+

||

−

ALB-WP-Ayush | HTTP:80

↺ ⓘ

Click a location for your new rule. Each rule must include one action of type forward, redirect, fixed response.

✓

New rule was created successfully.

✕

ALB-WP-Ayush | HTTP:80 (3 rules)

▶ Rule limits for condition values, wildcards, and total rules.

+

Insert Rule

1

arn...552d4

IF

✓ Path is /app2/*

THEN

Forward to

ayush-ALB-2: 1 (100%)

Group-level stickiness: Off

+

Insert Rule

2

arn...f6177

IF

✓ Path is /app1/*

THEN

Forward to

ayush-ALB-1: 1 (100%)

Group-level stickiness: Off

← → ↻ ⓘ Not secure | alb-wp-ayush-1235258010.us-east-1.elb.amazonaws.com/app1/

📱 Apps 🌐 Apple Final Cu... 🌐 Apple Final Cu... 📱 On Demand Tr... 📱 On Demand Ti

this is PAGE~ 1

← → ↻ ⓘ Not secure | alb-wp-ayush-1235258010.us-east-1.elb.amazonaws.com/app2/

📱 Apps 🌐 Apple Final Cu... 🌐 Apple Final Cu... 📱 On Demand Tr... 📱 On Demand

this is PAGE~ 1 this is the second page appended

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

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 best fits your needs.

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 >](#)

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.

Name ⓘ NLB-WP-ayush

Scheme ⓘ ☒ internet-facing ☐ internal

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 |
|------------------------|--------------------|
| TCP | 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 or specific addresses for your load balancer.

Create and manage Elastic IPs in the VPC console [↗](#)

VPC ⓘ vpc-0bce5df601296bb8a (10.0.0.0/16) | ayush-VPC

Step 3: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and perform

Target group

Target group

New target group

Name

ayush-NLB-TG

Target type

☒ Instance

☐ IP

Protocol

TCP

Port

80

Health checks

Protocol

HTTP

Path

/

Advanced health check settings

Port

☒ traffic port

☐ override

Healthy threshold

3

Unhealthy threshold

3

Timeout

6

seconds

Step 4: Register Targets

Configure Security Groups

The security groups for your instances must allow traffic from the VPC CIDR on the health check port.

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the t

Registered targets

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

Remove

| <input type="checkbox"/> | Instance | Name | Port | State |
|--------------------------|---------------------|----------|------|---------------------|
| <input type="checkbox"/> | i-05b1a8c786f575a57 | ayush-SG | 80 | <div></div> running |

Load Balancer Creation Status



Successfully created load balancer

Load balancer [NLB-WP-ayush](#) 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 c

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within [NLB-WP](#)
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS G](#)

→ ↻ ⓘ Not secure | nlb-wp-ayush-d0d5b8f48079f1a4.elb.us-east-1.amazonaws.com/wp-admin/setup-config.php

Apps 🍏 Apple Final Cu... 🍏 Apple Final Cu... 📺 On Demand Tr... 📺 On Demand Tr... 🔄 GitHub Stude... 🔄 GitHub Stude... 🚀 H



Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. **If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? [We got it.](#)

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready...

Let's go!

9.Take an instance out of the ASG.

We can detach instance from AGS

Create Auto Scaling group

Actions

Filter:

| <input type="checkbox"/> | Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Check Grace |
|--------------------------|--------|------------------------|-----------|---------|-----|-----|------------------------|------------------|--------------------|
| <input type="checkbox"/> | aws-SG | ayush-LT | 1 | 1 | 1 | 1 | us-east-1a, us-east-1b | 300 | 300 |

Auto Scaling Group: aws-SG

Details

Activity History

Scaling Policies

Instances

Monitoring

Notifications

Tags

Scheduled Actions

Lifecycle Hooks

Actions

Detach

Set to Standby

Set to InService

Instance Protection

status

Any Lifecycle State

| <input type="checkbox"/> | | Lifecycle | Launch Configuration / Template | Availability Zone | Health Status |
|--------------------------|---------------------|-----------|---------------------------------|-------------------|---------------|
| <input type="checkbox"/> | i-05b1a8c786f575a57 | InService | ayush-LT | us-east-1b | Healthy |

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

Edit details - aws-SG

Min

1

Max

1

Availability Zone(s)

us-east-1a x us-east-1b x

Subnet(s)

subnet-078f8e416344e3d9d(10.0.2.0/24) | ayush-SN-public-2 | us-east-1b x
subnet-016a6e052a2fac4ac(10.0.1.0/24) | ayush-SN-public-1 | us-east-1a x

Classic Load Balancers

Target Groups

Health Check Type

EC2

Health Check Grace Period

300

Instance Protection

Protect From Scale In x |

Termination Policies

Default x

Suspended Processes

Max Instance Lifetime

Placement Groups

Default Cooldown

300

Cancel

Save

11.Put Schedules in ASG to:

- Remove all instances of the ASG at 8 PM

14:30 UTC is 20:00 in GMT

Edit Scheduled Action ✕

Name

remove-at-8

Auto Scaling Group

aws-SG

Provide at least one of Min, Max and Desired Capacity

Min

0

Max

0

Desired Capacity

0

Recurrence

Every day ▼

(Cron) 30 14 ***

Start Time

2020-03-01

14 : 30

UTC

Specify the start time in UTC

The first time this scheduled action will run

End Time

[Set End Time](#)

[Cancel](#)

[Save](#)

- Launch a minimum of 2 instances at 10 AM

16:30 UTC is 22:00 in GMT

Create Scheduled Action ✕

Name

add-at-10

Auto Scaling Group

aws-SG

Provide at least one of Min, Max and Desired Capacity

Min

2

Max

2

Desired Capacity

2

Recurrence

Every day ▼
(Cron) 30 16 ***

Start Time

2020-03-01 16 : 30 UTC Specify the start time in UTC
The first time this scheduled action will run

End Time

[Set End Time](#)

Cancel

Create