

Part 1: EC2 with ELB and ASG

Objective: Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

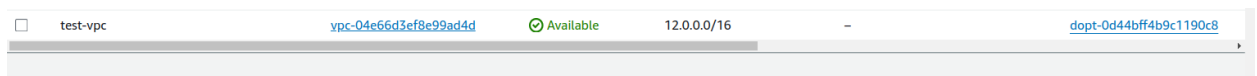
Approach:

1. **Launch EC2 Instances:** Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).
2. **Configure Load Balancer:** Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.
3. **Set Up Auto Scaling Group (ASG):** Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.
4. **Test Your Setup:** Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.
5. **Verify Website Functionality:** Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations.

Goal: By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

General steps

1. **Log in to your AWS Management Console and create a VPC with the necessary configuration.**



2. **Then create an internet gateway with the necessary configuration and attach to our newly create VPC.**

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="my-ig-test"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		

You can add 49 more tags.

3. Create two different public subnet

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block

 256 IPs
< > ^ v

▼ Tags - optional

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="test-public-subnet2"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		

You can add 49 more tags.

4. Now create a route table with our newly create VPC

VPC > Route tables > Create route table

Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

vpc-04e66d3ef8e99ad4d (test-vpc) ▼

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key
 X

Value - optional
 X

You can add 49 more tags.

5. Now associate the route table to newly created public subnet by clicking edit subnet association under subnet association and select all the subnet.

VPC > Route tables > rtb-014d6b52b24a78525 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/2)

< 1 > ⚙

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	test-public-subnet1	subnet-0f10112be1becb5d1	12.0.1.0/24	–	Main (rtb-016b95a5591807830)
<input checked="" type="checkbox"/>	test-public-subnet2	subnet-0afa6d141ae3f83aa	12.0.3.0/24	–	Main (rtb-016b95a5591807830)

Selected subnets

subnet-0f10112be1becb5d1 / test-public-subnet1 X subnet-0afa6d141ae3f83aa / test-public-subnet2 X

6. Add the routes on the route table by selecting our internet gateway

VPC > Route tables > rtb-014d6b52b24a78525 > Edit routes

Edit routes

Destination	Target	Status	Propagated
12.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

Buttons: Add route, Cancel, Preview, Save changes

7. Create a target group which will be responsible for pointing to ec2 instance so choose a target type to instances and our created VPC while creating target group.

EC2 > Target groups

Target groups (1) Info

Filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
test-target-ec2-apache2	arn:aws:elasticloadbalanci...	80	HTTP	Instance	None associated	vpc-04e66d3ef8e99ad4

8. Again create a load balancer choose application load balancer and select the our VPC the two subnet that we have created and also create new security group here, After creating load balancer wait until the balancer is in active state.

Successfully created load balancer: test-load-balancer
It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

EC2 > Load balancers > test-load-balancer

test-load-balancer

Details

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate
HTTP:80	Forward to target group <ul style="list-style-type: none">test-target-ec2-apache2 1 (100%)Group-level stickiness: Off	1 rule	ARN	Not applicable	Not applicable

9. Now create the auto scaling groups

Choose launch template [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name

Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#)

i For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.



[Create a launch template](#)

Cancel

Next

Here a create launch template(which is a primary configuration which specify what type of ec2 instance you want to setup so the auto scaling group will going to use the template to create ec2 instance) select ubuntu as ami, t2.micro as instance type, create a new keypair, http and ssh rule for the security group and attach the below script for running a apache server

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1> Hello World from $(hostname -f)</h1>" > /var/www/html/index.html
```

So our launch template is created

Launch Templates (1) [Info](#)

[Refresh](#) [Actions](#) [Create launch template](#)

	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
	lt-0b3f3587994a69a6a	lt-ec2-instances	1	1	2024-02-15T15:22:14.000Z	arn:aws:sts::992382801069:assu...

Now select the created launch template and click next and select our VPC

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

12.0.0.0/16

[Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

12.0.1.0/24

12.0.3.0/24

[Create a subnet](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

Then enable the attach an existing load balancer also enable choose from load balancer and security groups select the created test-target-group, enable ELB health check and click next.

Configure the group size and click next

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed Instances groups configured with a set of Instance attributes.

Desired capacity
Specify your group size.

Scaling [Info](#)
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

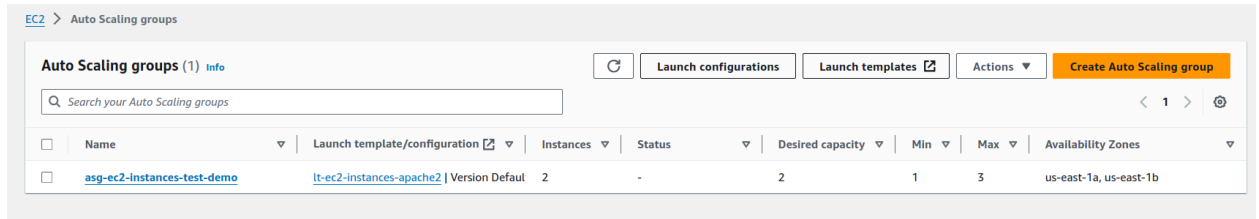
Equal or less than desired capacity

Max desired capacity

Equal or greater than desired capacity

Automatic scaling [optional](#)

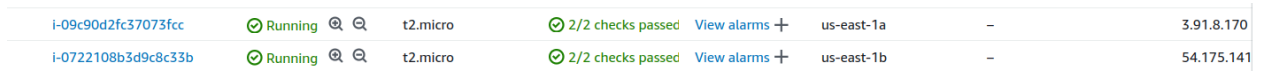
Now our auto scaling group is created, in the image below you can see the instance to 2 so it will automatically create 2 instance for us.



The screenshot shows the 'Auto Scaling groups (1)' page in the AWS console. A table lists the group 'asg-ec2-instances-test-demo' with details: Launch template 'lt-ec2-instances-apache2', Version 'Default', 2 instances, Desired capacity 2, Min 1, Max 3, and Availability Zones 'us-east-1a, us-east-1b'.

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
asg-ec2-instances-test-demo	lt-ec2-instances-apache2 Version Default	2	-	2	1	3	us-east-1a, us-east-1b

Here two instance are created



The screenshot shows two EC2 instances in a 'Running' state. Both are t2.micro instances with 2/2 checks passed. The first instance has IP 3.91.8.170 and the second has IP 54.175.141.

Instance ID	Status	Instance Type	Checks	Availability Zone	Private IP
i-09c90d2fc37073fcc	Running	t2.micro	2/2 checks passed	us-east-1a	3.91.8.170
i-0722108b3d9c8c33b	Running	t2.micro	2/2 checks passed	us-east-1b	54.175.141

10. Go to load balancer and copy the dns name and paste it in web browser



The screenshot shows the details of an Elastic Load Balancing (ALB) instance named 'test-alb-ec2-instance'. It is in an 'Active' state. The console displays various attributes including VPC, Availability Zones, and a DNS name: 'test-alb-ec2-instance-1009435130.us-east-1.elb.amazonaws.com'. A tooltip indicates 'DNS name copied'.

Attribute	Value
Load balancer type	Application
Status	Active
VPC	vpc-0049771f1f77c9c73
Availability Zones	subnet-03107fd6df56d63ef (us-east-1b), subnet-0870be9965d8969a6 (us-east-1a)
IP address type	IPv4
Date created	February 15, 2024, 21:28 (UTC+05:45)
Load balancer ARN	arn:aws:elasticloadbalancing:us-east-1:866388144037:loadbalancer/app/test-alb-ec2-instance/bba52225393776fc
DNS Name	test-alb-ec2-instance-1009435130.us-east-1.elb.amazonaws.com (A Record)

As we can see our web server is running simple application



Hello World from ip-12-0-1-139.ec2.internal

If we refresh we get new ip address



Hello World from ip-12-0-3-7.ec2.internal

So our load balancer is routing to our ec2 instances properly

11. Go to the load balancer target group int is running two instance in healthy condition

Targets									
Registered targets (2) Info									
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.									
Filter targets									
	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection...	
<input type="checkbox"/>	i-024fe37981bfe7ea8		80	us-east-1a	Healthy	-	February ...	Normal	
<input type="checkbox"/>	i-0ce7ed8bad9221b86		80	us-east-1b	Healthy	-	February ...	Normal	

12. If we terminate it will create a new instance,so i am termination a ec2 instance

Find Instance by attribute or tag (case-sensitive)									
Any state									
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
<input checked="" type="checkbox"/>		i-024fe37981bfe7ea8	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-	3.1
<input type="checkbox"/>		i-0ce7ed8bad9221b86	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-	3.1
<input type="checkbox"/>	Lab2	i-032f592c6f64ee2a7	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-
<input type="checkbox"/>	Lab1	i-006cae8a67eb2fd1b	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-

Now if we go to auto scaling group the new instance is being created

my-test-ASG									
Details Activity Automatic scaling Instance management Monitoring Instance refresh									
Instances (3)									
Filter instances									
	Instance ID	Lifecycle	Instance type	Weighted capac...	Launch templat...	Availability Zone	Health status	Protected from	
<input type="checkbox"/>	i-00d5398c1ab29e09e	Pending	t2.micro	-	asg-test-template \	us-east-1a	Healthy		
<input type="checkbox"/>	i-024fe37981bfe7ea8	Terminating	t2.micro	-	asg-test-template \	us-east-1a	Unhealthy		
<input type="checkbox"/>	i-0ce7ed8bad9221b86	InService	t2.micro	-	asg-test-template \	us-east-1b	Healthy		

Go to the ec2 dashboard to verify the new instance is being created if we terminate one

Instances (5) Info									
Find Instance by attribute or tag (case-sensitive)									
Any state									
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
<input type="checkbox"/>		i-00d5398c1ab29e09e	Running	t2.micro	Initializing	View alarms +	us-east-1a	-	3.1
<input type="checkbox"/>		i-024fe37981bfe7ea8	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-
<input type="checkbox"/>		i-0ce7ed8bad9221b86	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-	3.1

Part 2: Hosting a Static Portfolio Website on S3

Objective: Learn to host a static website (such as a personal portfolio) on Amazon S3.

Approach:

1. **Create an S3 Bucket:** Start by creating a new S3 bucket. Configure the bucket for website hosting, which includes setting permissions to make the content publicly accessible.
2. **Upload Website Files:** Upload the static files of your portfolio website (HTML, CSS, JavaScript, images) to the S3 bucket.
3. **Configure DNS:** Use Amazon Route 53 or another DNS service to point a domain name to the S3 bucket. This makes the website accessible via a user-friendly URL.
4. **Enable Additional Features** (Optional): Implement features like HTTPS for secure access and CloudFront for content delivery optimization.

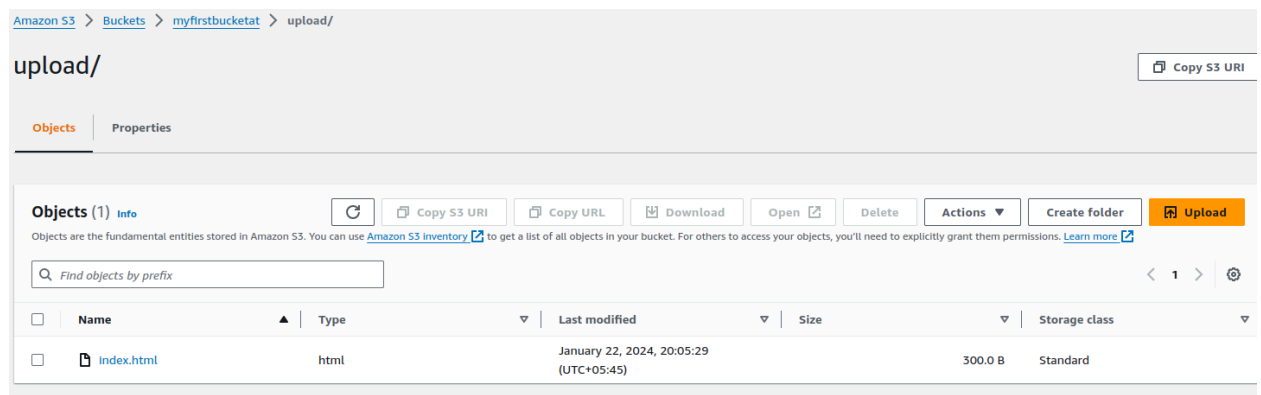
Goal: Students will understand how to use S3 for hosting static websites, manage bucket permissions, and integrate with other AWS services for a complete web hosting solution.

#####

Here i will try to map the s3 bucket static website with Domain name as **www.satishsubedi.com**

Basic Steps:

1. **Create a S3 bucket as a root bucket for a website with necessary details and add the getObject policy through policy generator and upload the file by clicking the upload file button also enable the static hosting.**



2. Create another sub-domain bucket with the same name as a root domain just add **www.** Before the bucket name

General purpose buckets (4) [info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	AWS Region	Access	Creation date
<input type="radio"/>	bucketfor-api-gateway	US East (N. Virginia) us-east-1	Public	January 23, 2024, 19:56:21 (UTC+05:45)
<input type="radio"/>	myfirstbucketat	US East (N. Virginia) us-east-1	Public	January 22, 2024, 16:55:10 (UTC+05:45)
<input type="radio"/>	satishsubedi.com	US East (N. Virginia) us-east-1	Public	February 14, 2024, 20:07:53 (UTC+05:45)
<input type="radio"/>	www.satishsubedi.com	US East (N. Virginia) us-east-1	Objects can be public	February 14, 2024, 20:35:48 (UTC+05:45)

3. Again go to the properties tab and enable the static hosting and add other information such as hosting type to redirect request host name to your root bucket name and protocol to http.

[Amazon S3](#) > [Buckets](#) > [www.satishsubedi.com](#) > Edit static website hosting

Edit static website hosting [info](#)

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable

☒ Enable

Hosting type

☐ Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

☒ Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

Host name

Target bucket website address or personal domain

Protocol - *Optional*

☐ none

☒ http

☐ https

Cancel [Save changes](#)

- Now go to aws console and click to the route 53 and create the hosted zone there note the name of the hosted zone should be you root domain name(which is the name of root s3 bucket name)

satishsubedi.com was successfully created.

Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.

Route 53 > Hosted zones > satishsubedi.com

Public

satishsubedi.com

Info

Delete zone

Test record

Configure query logging

Hosted zone details

Edit hosted zone

Records (2)

DNSSEC signing

Hosted zone tags (0)

Records (2) Info

Refresh

Delete record

Import zone file

Create record

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Filter records by property or value

Type

Routing policy

Alias

< 1 >

Settings

<input type="checkbox"/>	Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to	TT
<input type="checkbox"/>	satishsub...	NS	Simple	-	No	ns-1889.awsdns-44.co.uk. ns-283.awsdns-35.com. ns-1115.awsdns-11.org. ns-536.awsdns-03.net.	17
<input type="checkbox"/>	satishsub...	SOA	Simple	-	No	ns-1889.awsdns-44.co.uk. a...	90

- Now create a record for you root s3 bucket, under here enable the alias button and disable the evaluate target health and select region as your current region and route traffic to alias to s3 website endpoint etc. as shown in the figure

[Route 53](#) > [Hosted zones](#) > [satishsubedi.com](#) > Create record

Create record [Info](#)

Quick create record [Switch to wizard](#)

▼ Record 1 [Delete](#)

Record name [Info](#)

subdomain satishsubedi.com

Record type [Info](#)

A – Routes traffic to an IPv4 address and some AWS resources ▼

Keep blank to create a record for the root domain.

☒ Alias

Route traffic to [Info](#)

Alias to S3 website endpoint ▼

US East (N. Virginia) ▼

s3-website-us-east-1.amazonaws.com

Routing policy [Info](#)

Simple routing ▼

Evaluate target health

☐ No

Add another record

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

6. Again create a record for the sub domain as well all the steps are pretty similar except record name to www

▼ Record 1 [Delete](#)

Record name [Info](#)

www .satishsubedi.com

Record type [Info](#)

A – Routes traffic to an IPv4 address and some AWS resources ▼

Keep blank to create a record for the root domain.

☒ Alias

Route traffic to [Info](#)

Alias to S3 website endpoint ▼

US East (N. Virginia) ▼

s3-website-us-east-1.amazonaws.com

Routing policy [Info](#)

Simple routing ▼

Evaluate target health

☐ No

These are my records

► Hosted zone details

Edit hosted zone

Records (4)

DNSSEC signing

Hosted zone tags (0)

Records (4) Info



Delete record

Import zone file

Create record

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

🔍 Filter records by property or value

Type ▼

Routing policy ▼

Alias ▼



1



<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Differ... ▼	Alias ▼	Value/Route traffic to ▼	TTL (s...
<input type="checkbox"/>	satishsubedi.com	A	Simple	-	Yes	s3-website-us-east-1.amazo...	-
<input type="checkbox"/>	satishsubedi.com	NS	Simple	-	No	ns-1889.awsdns-44.co.uk. ns-283.awsdns-35.com. ns-1115.awsdns-11.org. ns-536.awsdns-03.net.	172800
<input type="checkbox"/>	satishsubedi.com	SOA	Simple	-	No	ns-1889.awsdns-44.co.uk. a...	900
<input type="checkbox"/>	www.satishsubedi.c...	A	Simple	-	Yes	s3-website-us-east-1.amazo...	-