

Project Documentation: Hosting a Website with Auto Scaling & Load Balancer on AWS EC2

Objective

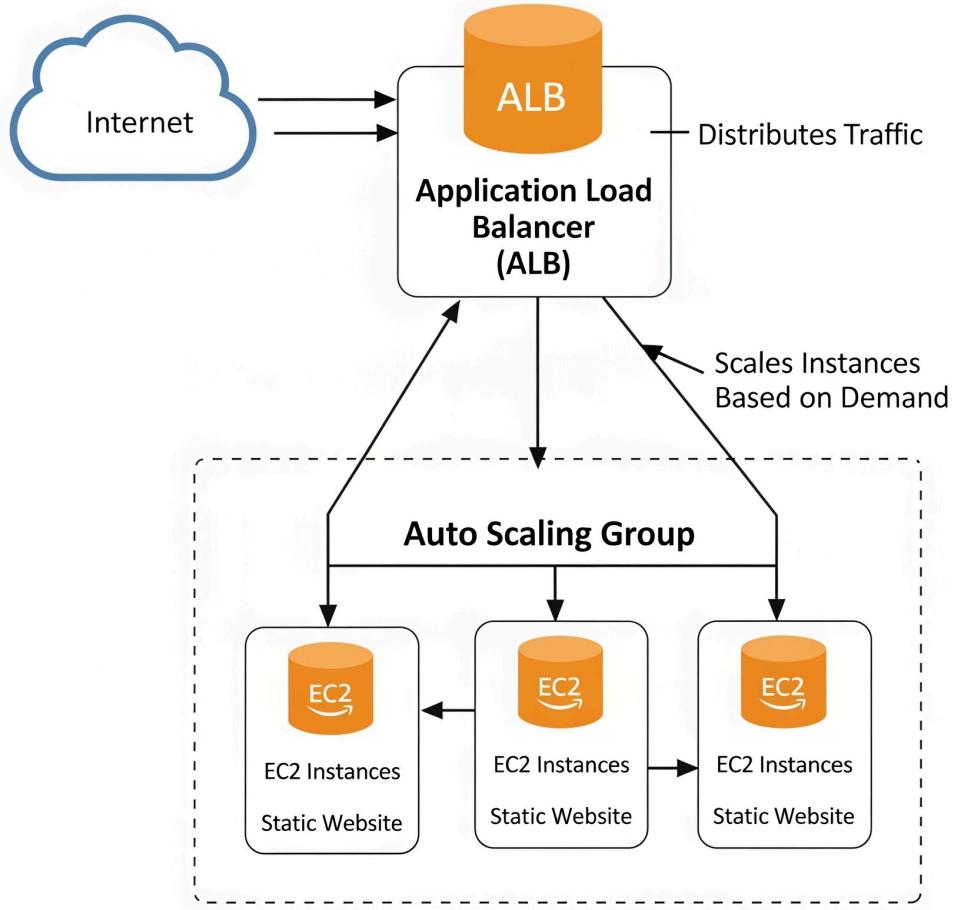
Set up a **highly available and scalable static website** on AWS using **EC2, Auto Scaling Group (ASG), and Application Load Balancer (ALB)** to simulate real-world cloud architecture.

Prerequisites

- AWS account (Free Tier eligible)
- Basic knowledge of EC2, Security Groups, and website hosting

Architecture Overview

- **ALB** → Distributes traffic across multiple instances for high availability.
- **ASG** → Maintains minimum/desired number of EC2 instances and scales based on load.
- **EC2 Instances** → Hosts the static website on EC2



Step 1: Prepare EC2 Instance

1. Launch an EC2 instance with **Amazon Linux 2**.
2. Install and configure Apache HTTP server:
 - `sudo yum update -y`
 - `sudo yum install -y httpd`
 - `sudo systemctl start httpd`
 - `sudo systemctl enable httpd`

3. Deploy your static website to `/var/www/html`.
4. Test website accessibility via public IP.

The screenshot shows the AWS EC2 Instances page. The left sidebar has 'EC2' selected under 'Instances'. The main area shows a table with one instance listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Auto-Scaling&...	i-0b1b5127e2e616d15	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a

Below the table, the instance details for `i-0b1b5127e2e616d15` are shown. The 'Details' tab is selected. Key information includes:

- Instance ID:** i-0b1b5127e2e616d15
- Public IPv4 address:** 3.91.11.255 (with a link to open address)
- Private IPv4 addresses:** 172.31.39.125
- Instance state:** Running
- Public DNS:** (link)

Step 2: Create an AMI

1. Go to **EC2 → Instances**, select your configured instance.
2. Click **Actions → Image → Create Image**.
3. Provide a name (e.g., **Static-Website-AMI**) and create the image.
- This AMI will be used by Auto Scaling to launch identical instances.

The screenshot shows the AWS EC2 AMIs page. The left sidebar includes links for Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs), AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (CloudShell, Feedback). The main content area displays the 'Amazon Machine Images (AMIs) (1/1)' section. A table lists one AMI: 'Static-Website-AMI' (ami-0109c2c6454dd3675). The table columns are Name, AMI ID, Source, and Owner. Below the table, the 'AMI ID: ami-0109c2c6454dd3675' details page is shown, with tabs for Details, Permissions, Storage, My AMI usage - new, and Tags. The 'Details' tab shows the following information:

AMI ID	Image type	Platform details	Root device type
ami-0109c2c6454dd3675	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
Static-Website-AMI	771923946960	x86_64	RunInstances

At the bottom of the page are links for © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 3: Configure Target Group

1. Navigate to **EC2 → Target Groups → Create Target Group.**
2. Type: **Instances**
3. Protocol: **HTTP**, Port: **80**
4. Target group name: e.g., **Website-TG**
5. Health check path: **/**

The screenshot shows the AWS EC2 Target Groups console. The left sidebar has navigation links for Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), Auto Scaling (Auto Scaling Groups), and Settings. The main content area is titled "Website-TG". It shows the "Details" section with the ARN: am:aws:elasticloadbalancing:us-east-1:771923946960:targetgroup/Website-TG/3a1d9178de4b5634. It lists the Target type as Instance, Protocol as Port HTTP: 80, and Protocol version as HTTP1. The VPC is associated with vpc-09eaa5ecb34a57b94. Below this, a table shows 1 Total targets: 0 Healthy, 0 Unhealthy, 0 Anomalous, 1 Unused, 0 Initial, and 0 Draining. A section titled "Distribution of targets by Availability Zone (AZ)" indicates no targets are registered. At the bottom, tabs for Targets (selected), Monitoring, Health checks, Attributes, and Tags are visible.

Step 4: Create Application Load Balancer (ALB)

1. EC2 → Load Balancers → Create Load Balancer → Application Load Balancer
2. Internet-facing, IPv4, Protocol: HTTP, Port: 80
3. Choose at least 2 Availability Zones
4. Attach **Security Groups** allowing HTTP (80)
5. Register the **Target Group** from Step 3

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar has sections for Dashboard, EC2 Global View, Events, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (CloudShell, Feedback). The main area is titled "Website-LBR" and contains a "Details" section with fields: Load balancer type (Application), Status (Active), Scheme (Internet-facing), Hosted zone (Z35SXDOTRQ7X7K), VPC (vpc-09eaa5ecb34a57b94), Availability Zones (subnet-036788e70cd554b4b, subnet-0e641aacb5acbefc4, subnet-0ff4b64a6ddc9dc07, subnet-0a88936e8c57c3459, subnet-059ef28505cab3391, subnet-0ddf171ccbb92bd5b), Load balancer IP address type (IPv4), Date created (September 3, 2025, 19:45 (UTC+05:30)), and DNS name (Info). At the bottom, there's a footer with links for © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 5: Create Launch Template

1. EC2 → Launch Templates → Create New
2. Select **AMI** from Step 2
3. Instance type: **t2.micro**
4. Security Group: Allow HTTP (80) + SSH (22) from your IP
5. Name the template (e.g., **Website-LaunchTemplate**)

The screenshot shows the AWS EC2 Launch Templates page. The left sidebar is collapsed. The main content area displays the 'Website-LT (lt-002f647743edf32cd)' launch template. The 'Launch template details' section includes fields for Launch template ID (lt-002f647743edf32cd), Launch template name (Website-LT), Default version (1), and Owner (arn:aws:sts::771923946960:assumed-role/voclabs/user4088122=harshithagalla00@gmail.com). Below this, the 'Launch template version details' section shows one version (1 (Default)) created on 2025-09-03T14:24:23.000Z by the same owner. The 'Instance details' tab is selected, showing AMI ID (ami-0109c2c6454dd3675) and Instance type (t2.micro). Other tabs include Storage, Resource tags, Network interfaces, and Advanced details.

Step 6: Create Auto Scaling Group (ASG)

1. EC2 → Auto Scaling Groups → Create Auto Scaling Group
2. Select Launch Template
3. Set desired capacity: 2, min: 1, max: 2
4. Attach ALB Target Group
5. Configure scaling policies (optional)

The screenshot shows the AWS Auto Scaling Groups page for the 'Website-ASG' group. The top navigation bar includes the AWS logo, search bar, [Alt+S] key, account information (Account ID: 7719-2394-6960, vclabs/user4088122=harshithagalla00@gmail.com), and region selection (United States (N. Virginia)). Below the navigation is a breadcrumb trail: EC2 > Auto Scaling groups > Website-ASG. The main content area is titled 'Website-ASG Capacity overview' with an 'Edit' button. It displays the following details:

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
1	1 - 2	Units (number of instances)	-

Date created: Wed Sep 03 2025 19:59:49 GMT+0530 (India Standard Time)

Below this, there are tabs for Details, Integrations - new, Automatic scaling, Instance management, Instance refresh, Activity, and Monitoring. The Details tab is selected. Under 'Launch template', there is another 'Edit' button. The launch template details are:

Launch template	AMI ID	Instance type	Owner
lt-002f647743edf32cd Website-LT	ami-0109c2c6454dd3675	t2.micro	arn:aws:ssts::771923946960:assumed-role/vclabs/user4088122=harshithagalla00@gmail.com

At the bottom of the page are links for CloudShell, Feedback, and footer information: © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 7: Validate Setup

1. Open **ALB DNS** in a browser → website should load.
2. Stop one EC2 instance manually → ALB still serves traffic from remaining instance.
3. Check **Auto Scaling Events** → new instance launched to maintain desired capacity.

Screenshot of the AWS EC2 Instances page showing two instances selected for monitoring.

Instances (2/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Auto-Scaling&...	i-0b1b5127e2e616d15	Shutting-d...	t3.micro	3/3 checks passed	View alarms +	us-east-1a
	i-091d1acf1366df45d	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	us-east-1a

Monitoring

2 instances selected

Investigate with AI - new

CPU utilization (%)	Network in (bytes)	Network out (bytes)	Network packets in (count)
Various units 7.37	Various units 10.5M	Various units 254k	Various units 3.67k

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Instances page showing three instances listed, with the middle one running.

Instances (2/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Auto-Scaling&...	i-0b1b5127e2e616d15	Terminated	t3.micro	-	View alarms +	us-east-1a
	i-02b6b910f7b0ff1af	Running	t2.micro	Initializing	View alarms +	us-east-1a
	i-091d1acf1366df45d	Terminated	t2.micro	-	View alarms +	us-east-1a

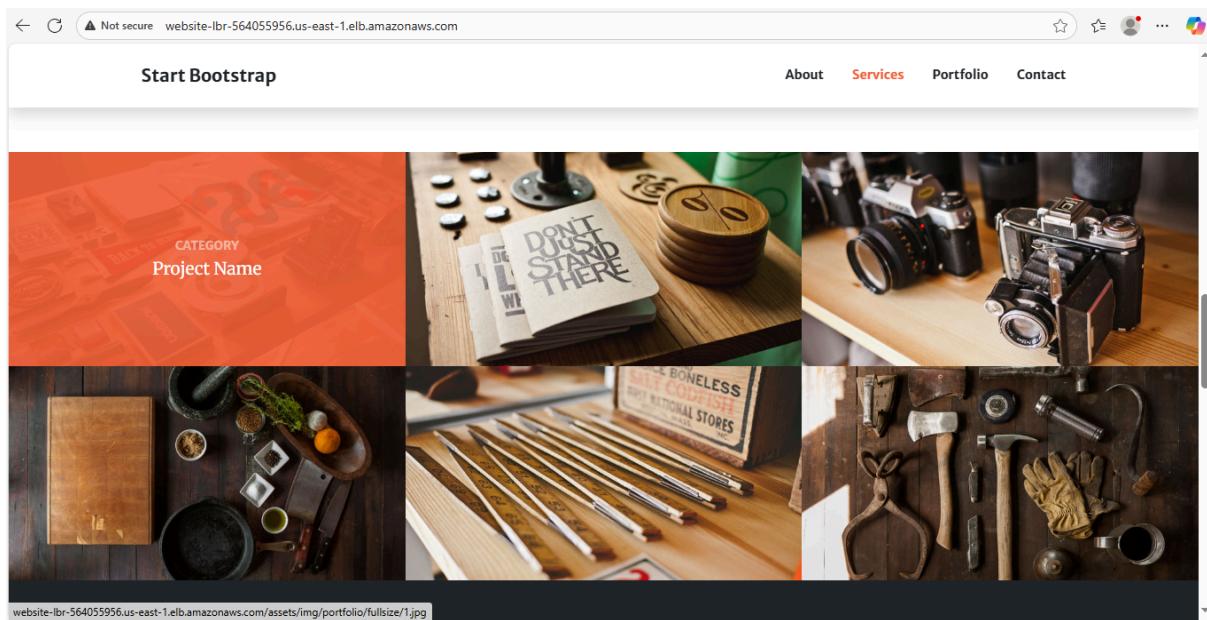
Monitoring

2 instances selected

Investigate with AI - new

CPU utilization (%)	Network in (bytes)	Network out (bytes)	Network packets in (count)
Various units 7.37	Various units 10.5M	Various units 254k	Various units 3.67k

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Outcome

- Achieved **high availability** with ALB distributing traffic.
- Configured **fault-tolerant and scalable infrastructure** using ASG.
- Prepared a production-like **AWS deployment environment** for static websites.