

Karthik Pradeep Hegadi

2KE20CS032

Assignment 43

Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

AWS

Assignment: 9 Autoscaling

Autoscaling - Launch Configurations & Auto Scaling Groups

Instructions

Steps:

1. Creating launch configuration
2. Creating Auto Scaling with a load balancer
3. Apply load using meter and verify new instances are launched based on Autoscaling (meter script attached)

Pre-requests

1. Launch a new ec2 instance and install and run nginx server alone

The screenshot shows the AWS EC2 Instances Pre-requests page for an instance named 'i-0902b8f9089bcf2b4 (autoscaling-02)'. The instance is currently stopped. Key details visible include:

- Instance ID:** i-0902b8f9089bcf2b4 (autoscaling-02)
- Public IPv4 address:** -
- Private IP4 addresses:** 10.0.134.236
- Public IPv4 DNS:** -
- Private IP DNS name (IPv4 only):** ip-10-0-134-236.ap-south-1.compute.internal
- Instance type:** t2.micro
- Elastic IP addresses:** -
- VPC ID:** vpc-0e190ca43b317839f (my-vpc-01)
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
- Subnet ID:** subnet-0cd9caa8bc0c4598b (my-public-subnet-2)
- Auto Scaling Group name:** -
- IAM Role:** -
- IMDSv2:** Required

2. Create an image based on the above instance where you have nginx running

The screenshot shows the AWS EC2 AMI details page. At the top, there's a breadcrumb navigation: EC2 > AMIs > ami-06ea5cad146fdd69d. Below the title "Pre-requests" is a summary table for the AMI "ami-06ea5cad146fdd69d". The table includes columns for AMI ID, Image type, Platform details, Root device type; AMI name, Owner account ID, Architecture, Usage operation; Root device name, Status, Source, Virtualization type; Boot mode, State reason, Creation date, Kernel ID; Description, Product codes, RAM disk ID, Deprecation time; and Last launched time, Block devices. A note at the bottom states: "Image share permission Private".

Steps to create Launch configuration

1. Navigate to Auto scaling services -> Launch configurations -> create launch configurations

The screenshot shows the "Create launch template" page. At the top, there's a navigation bar with the AWS logo, Services, Search, and a [Option+S] button. The main heading is "Create launch template". A sub-instruction says: "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." The form starts with "Launch template name and description". The "Launch template name - required" field contains "autoscaling-LT". A note below it says: "Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.". The "Template version description" field contains "This is an autoscaling launch template." with a red circle containing the number "1" to its right. A note below it says: "Max 255 chars". The "Auto Scaling guidance" section has an "Info" link and a note: "Select this if you intend to use this template with EC2 Auto Scaling". A checked checkbox says: "Provide guidance to help me set up a template that I can use with EC2 Auto Scaling". At the bottom, there are two links: "Template tags" and "Source template".

2. Provide the Name, choose the AM that you have created in the pre-requests

▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

[Recents](#) [My AMIs](#) [Quick Start](#)

Owned by me Shared with me

[Browse more AMIs](#) 

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

autoscaling-image
ami-06ea5cad146fdd69d
2023-12-01T17:27:26.000Z Virtualization: **autoscaling-image** true Root device type: ebs

Description
this image provides the autoscaling image

Architecture AMI ID
x86_64 ami-06ea5cad146fdd69d

▼ Instance type [Info](#) | [Get advice](#) [Advanced](#)

Instance type

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0724 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

▼

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

▼

Create new key pair

3. Choose Instance type as t2.micro

4. Additional configuration, EBS volume as default settings, also enable public assign a public ip option
5. You can create or select existing security group
6. Choose from existing key pair
7. Create launch configuration

The screenshot shows the 'Network settings' section of an AWS Launch Configuration. It includes fields for Subnet (set to 'Don't include in launch template'), Firewall (security groups) (set to 'Select existing security group' with 'launch-wizard-44 sg-05654e480536849e4' selected), and Advanced network configuration.

Network settings

Subnet [Info](#)

Don't include in launch template [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group Create security group

Security groups [Info](#)

Select security groups

launch-wizard-44 sg-05654e480536849e4 [X](#)
VPC: vpc-0e190ca43b317839f

[Compare security group rules](#)

Advanced network configuration

Storage (volumes) [Info](#)

Create a Load Balancer for Auto scaling groups

1. Navigate to Load Balancers -> Target groups click create Target group
2. Target Type as Instance, provide Target group name, Protocol http : 80 (default)

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.

autosaling-LT



[Create a launch template](#)

Version

Default (1)



[Create a launch template version](#)

Description

This is a launch template for autoscaling.

Launch template

[autosaling-LT](#)

lt-0a9ef642ca09d0675

Instance type

t2.micro

AMI ID

ami-06ea5cad146fdd69d

Security groups

-

Request Spot Instances

No

Key pair name

key

Security group IDs

[sg-05654e480536849e4](#)

Additional details

Storage (volumes)

Date created

3. Select your VPC

In the Advanced health check settings you can give your custom values in the traffic port. (you can leave as default if you don't wish to change it)

5. Add tag and click Next

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

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

vpc-0e190ca43b317839f (my-vpc-01)
10.0.0.0/16



[Create a VPC](#)

Availability Zones and subnets

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

Select Availability Zones and subnets



ap-south-1a | subnet-0f3a2030b8047392a (my-public-subnet-1)
10.0.1.0/24



ap-south-1b | subnet-0cd9caa8bc0c4598b (my-public-subnet-2)
10.0.128.0/17



[Create a subnet](#)

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

autoscaling-ALB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme can't be changed after the load balancer is created.

Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

Select the type of IP addresses that your subnets use.

IPv4

Recommended for internal load balancers.

Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping Info

VPC Info

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. This load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

my-vpc-01

vpc-0e190ca43b317839f

IPv4: 10.0.0.0/16



Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones where the load balancer or the VPC are not available for selection.

ap-south-1a (aps1-az1)

Subnet

subnet-0f3a2030b8047392a

my-public-subnet-1 ▾

IPv4 address

Assigned by AWS

ap-south-1b (aps1-az3)

Subnet

subnet-0cd9caa8bc0c4598b

my-public-subnet-2 ▾

IPv4 address

Assigned by AWS

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups



default

sg-025e3c77b92134dee VPC: vpc-0e190ca43b317839f



SG_mattermost_LB-01

sg-0d9085f82835e6578 VPC: vpc-0e190ca43b317839f



7. Click 'Create Target group' and it will be created successfully

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP	▼	80	▲
1-65535			

IP address type

8. Create a load balancer by selecting Application Load Balancer

9. Provide the load balancer name, select Internet -facing, and IPv4 address type

autoscaling-TG

Details

arn:aws:elasticloadbalancing:ap-south-1:405819896469:targetgroup/autoscaling-TG/07d88165782a4084

Target type
Instance

Protocol : Port
HTTP: 80

Protocol version
HTTP1

IP address type
IPv4

Load balancer
[None associated](#)

10. In the network mapping select your VPC

11. In the subnet mapping select the availability zones and select public subnets that

you have created from the dropdown

12. Select the security group from existing that you have created for previous Load Balancer assignment

13. In Listeners and Routing select the target group that you have created above

14. Provide the Tags and create the load balancer

Steps to Create Auto Scaling group

1. Navigate to Auto Scaling group -> Create an Auto scaling group
2. Provide Auto scaling group name
3. In the Launch template option click switch to launch configuration
4. Select the Launch configuration that you have created and click Next
5. In the Network, choose your pc, choose the different availability zones of public subnet
6. Click Next
7. Now select attach to existing Load Balancer
8. Select the target group that you have created for Autoscaling
9. Leave Health checks and Additional settings as default
10. Click Next
11. In the Configure group size set as below 2,1,2
12. In the Scaling policies, select Target scaling policy and provide the options as below
13. Click Next
14. In Add notification click on create a topic, provide your email address
15. Provide the Tags for new instances

16. Verify all the configurations and click on 'create auto scaling groups'
17. Auto Scaling group will be created as below

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how to its registered targets.

▼ Listener HTTP:80

Protocol	Port	Default action
HTTP	: 80 1-65535	Forward to autoscaling-TG Target type: Instance, IPv4 Create target group

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

You can add up to 50 more tags.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▾ ✖ 

autoscaling-TG | HTTP X
Application Load Balancer: autoscaling-ALB

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks
 Always enabled

Additional health check types - optional 

Turn on Elastic Load Balancing health checks Recommended
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

Health check grace period 
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.
150 ▼ seconds

(Step 11)

Group size 
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
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.

Units (number of instances) ▾

Desired capacity
Specify your group size.
2 ▼

Scaling 
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	Max desired capacity
1 ▼	2 ▼
Equal or less than desired capacity	Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy | [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name

Target Tracking Policy

Metric type | [Info](#)

Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Application Load Balancer request count per target

Target group

autoscaling-TG

Target value

50

Instance warmup | [Info](#)

300

seconds

18. Once Auto scaling group created you can see the instance will be launched automatically as per the minimum capacity value. You can check the target details in your target group

19. Also you can see the instances list, where you can see Auto scaled instance will

be up and running

Load balancers (3)					
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.					
<input type="text"/> Filter load balancers					
	Name	DNS name	State	VPC ID	
<input type="checkbox"/>	NLBmattermost	NLBmattermost-c927335121...	Active	vpc-0e190ca43b3178...	
<input type="checkbox"/>	ALBmattermost	ALBmattermost-898366186...	Active	vpc-0e190ca43b3178...	
<input type="checkbox"/>	autoscaling-ALB	autoscaling-ALB-411914057...	Active	vpc-0e190ca43b3178...	

Test the Auto scaling by applying Loads using meter(install in MAC)

1. Download and install Apache meter 5.5 from this url

<https://imeter.apache.org/download/jmeter.cgi>

2. Run meter batch file inside bin folder to open the console

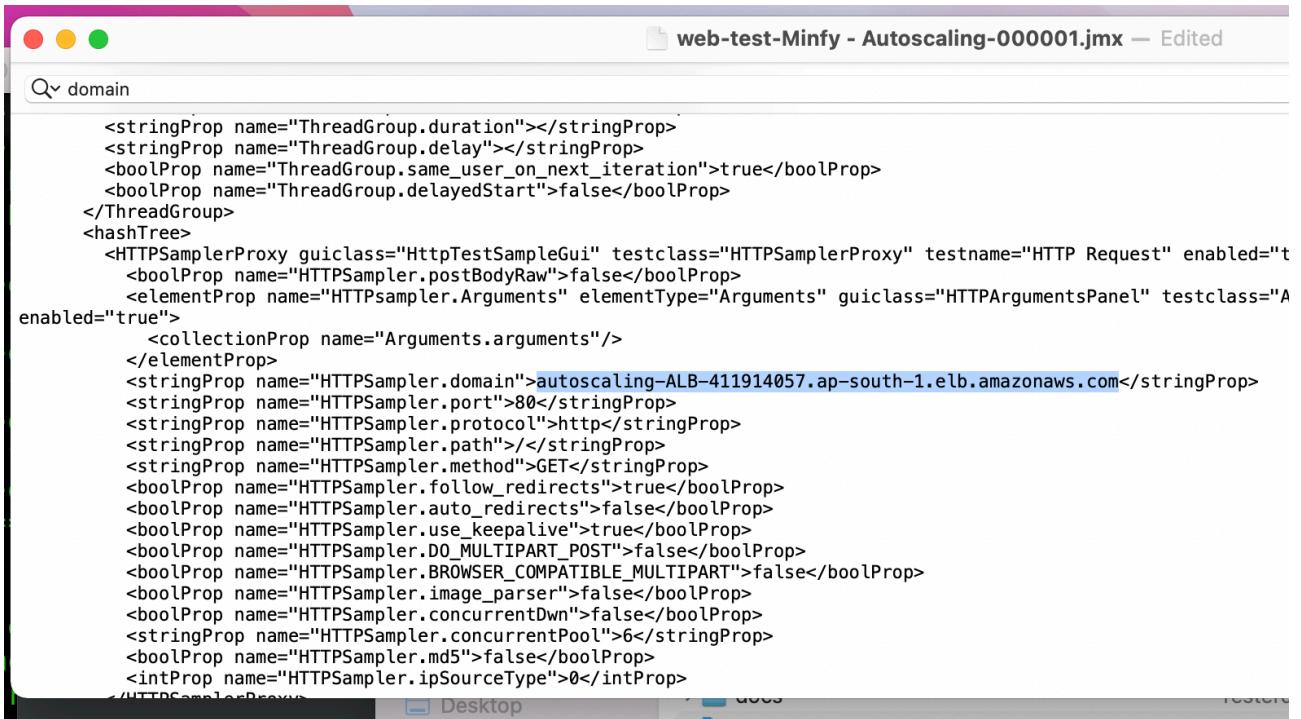
3. Open web-test.jmx in the below link your_Load_Balancer url update your url and save changes

<https://drive.google.com/drive/folders/1aHiNZsEJDKLUAgcCgfVzGHmf4qZejLwS>

4. In the jmeter console open the web-test.jmx file and click play button and it starts to push the load to your Load Balancer

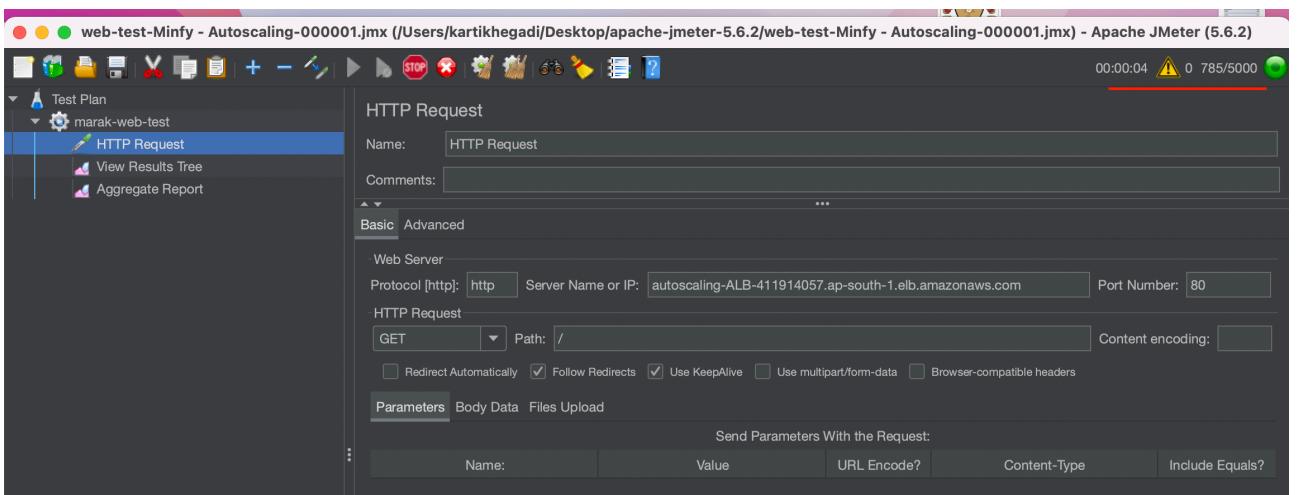
```
jmeter.log                                     utility.groovy
jmeter.properties
kartikhegadi@Kartiks-MacBook-Air bin % ./jmeter.sh*
WARNING: package sun.awt.X11 not in java.desktop
WARN StatusConsoleListener The use of package scanning to locate and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate and will be removed in a future release
=====
Don't use GUI mode for load testing !, only for Test creation
.
For load testing, use CLI Mode (was NON GUI):
  jmeter -n -t [jmx file] -l [results file] -e -o [Path to we
& increase Java Heap to meet your test requirements:
  Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMet
the jmeter batch file
Check : https://jmeter.apache.org/usermanual/best-practices.html#batch
```

(Step 3)



The screenshot shows a code editor window with the title "web-test-Minfy - Autoscaling-000001.jmx — Edited". The content is a snippet of JMeter configuration XML. It includes properties for Thread Group duration, delay, and iteration settings. A specific section for an HTTP sampler is highlighted in blue, showing the domain as "autoscaling-ALB-411914057.ap-south-1.elb.amazonaws.com", port as "80", protocol as "http", and method as "GET". Other properties like "Follow Redirects" and "KeepAlive" are also defined.

```
<stringProp name="ThreadGroup.duration"></stringProp>
<stringProp name="ThreadGroup.delay"></stringProp>
<boolProp name="ThreadGroup.same_user_on_next_iteration">true</boolProp>
<boolProp name="ThreadGroup.delayedStart">false</boolProp>
</ThreadGroup>
<hashTree>
    <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="HTTP Request" enabled="t
        <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
        <elementProp name="HTTPSampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="A
enabled="true">
            <collectionProp name="Arguments.arguments"/>
        </elementProp>
        <stringProp name="HTTPSampler.domain">autoscaling-ALB-411914057.ap-south-1.elb.amazonaws.com</stringProp>
        <stringProp name="HTTPSampler.port">80</stringProp>
        <stringProp name="HTTPSampler.protocol">http</stringProp>
        <stringProp name="HTTPSampler.path">/</stringProp>
        <stringProp name="HTTPSampler.method">GET</stringProp>
        <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
        <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
        <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
        <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
        <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
        <boolProp name="HTTPSampler.image_parser">false</boolProp>
        <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
        <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
        <boolProp name="HTTPSampler.md5">false</boolProp>
        <intProp name="HTTPSampler.ipSourceType">0</intProp>
    </HTTPSamplerProxy>
```



The screenshot shows the Apache JMeter interface with a "Test Plan" tree on the left. A "HTTP Request" item under a "marak-web-test" folder is selected. The main panel displays the "HTTP Request" configuration dialog. The "Name" field is set to "HTTP Request". Under the "Basic" tab, the "Protocol [http]" is set to "http", "Server Name or IP" is "autoscaling-ALB-411914057.ap-south-1.elb.amazonaws.com", and "Port Number" is "80". The "Method" is set to "GET" and the "Path" is "/". The "Content encoding" dropdown is empty. Below these fields, there are checkboxes for "Redirect Automatically", "Follow Redirects", "Use KeepAlive", "Use multipart/form-data", and "Browser-compatible headers". The "Parameters" tab is selected, showing a table with columns for "Name", "Value", "URL Encode?", "Content-Type", and "Include Equals?".

VERIFICATION

autoscaling-TG

Details

arn:aws:elasticloadbalancing:ap-south-1:405819896469:targetgroup/autoscaling-TG/07d88165782a4084

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0e190ca43b317839f
IP address type IPv4	Load balancer autoscaling-ALB		
2 Total targets	2 Healthy	0 Unhealthy	0 Unused
	0 Anomalous		0 Initial
			0 Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (2) Info

① Anomaly mitigation: Not applicable [C](#) [Deregister](#) [Register targets](#)

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.

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Anomaly detection result
<input type="checkbox"/>	i-0d1f9e3bcc57a725a		80	ap-south-1a	Healthy	-	Normal
<input type="checkbox"/>	i-03fa65c914a85e807		80	ap-south-1b	Healthy	-	Normal

Instances (10) Info

[Find Instance by attribute or tag \(case-sensitive\)](#) [C](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instance](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	public-inst(nginx)-2	i-05a4e1f88e7ed4668	Stopped	t2.micro	-	No alarms	+ ap-south-1b
<input type="checkbox"/>	private-inst(appserver)-2	i-0afa5896198d39468	Stopped	t2.micro	-	No alarms	+ ap-south-1b
<input type="checkbox"/>	autoscaling-02	i-0902b8f9089bcf2b4	Stopped	t2.micro	-	No alarms	+ ap-south-1b
<input type="checkbox"/>		i-0d26aaaae8af5872d	Terminated	t2.micro	-	No alarms	+ ap-south-1b
<input type="checkbox"/>		i-03fa65c914a85e807	Running	t2.micro	Initializing	No alarms	+ ap-south-1b
<input type="checkbox"/>	public-inst(nginx)-1	i-059f3bb15a76aad7d	Stopped	t2.micro	-	No alarms	+ ap-south-1a
<input type="checkbox"/>	private-inst(appserver)-1	i-0e7568e59b5b24589	Stopped	t2.micro	-	No alarms	+ ap-south-1a
<input type="checkbox"/>	private-inst(dbserver)	i-0a6d610ce310ab3bf	Stopped	t2.micro	-	No alarms	+ ap-south-1a
<input type="checkbox"/>		i-0d1f9e3bcc57a725a	Running	t2.micro	Initializing	No alarms	+ ap-south-1a
<input type="checkbox"/>		i-07abc3046cf8b281e	Terminated	t2.micro	-	No alarms	+ ap-south-1a

Auto Scaling group: [Autoscaling-alb-411914057.ap-south-1.elb.amazonaws.com](#) Instances | EC2 | [Create launch temp:](#) [Load balancers](#) [Target group details](#) [Load balancers](#) Data Representation Dev

Now we can see our static-website is visible!

Line Chart

Category	Value
Category A	15
Category B	25
Category C	28
Category D	20
Category E	10

Radar Chart

Category	Value
Category A	30
Category B	28
Category C	26
Category D	24
Category E	22