

BLACKBUCKS INTERNSHIP REPORT

**AN ARCHITECTURE OF AUTO-SCALING WITH LAUNCH TEMPLATE
AND LAUNCH CONFIGURATION**

SUBMITTED BY

V.VENKATA KAVERI(21B91A54I4)
V.POOJITHA(21B91A54I3)
V.SHYNIE PRAVALLIKA(21B91A54I7)

UNDER THE GUIDANCE OF MR.AASHU DEV

BLACKBUCKS INTERNSHIP REPORT

TEAM MEMBERS :

- V.VENKATA KAVERI(21B91A54I4)
- V.POOJITHA(21B91A54I3)
- V.SHYNIE PRAVALLIKA(21B91A54I7)

TITLE :

AN ARCHITECTURE OF AUTO-SCALING WITH LAUNCH TEMPLATE AND LAUNCH CONFIGURATION

ABSTRACT :

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes. AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.

TABLE OF CONTENTS

1. Introduction of AWS

2. Benefits of AWS

3. List of top AWS services

4. Implementation of the Architecture of Auto-Scaling with launch template and launch configuration

4.1. Introduction to Auto-scaling and benefits

4.2. Introduction to launch template

4.3. Introduction to launch configuration

4.4. Rough Architecture of Auto-scaling with launch template and launch config

4.5. Architecture of Auto-Scaling with launch template and launch config

5. Services used in creating Architecture

5.1. Service1:VPC

5.2. Service2:EC2

5.3. Service3:AMI

5.4. Service4:SNAPSHOT

5.5. CREATING LAUNCH TEMPLATE

5.6. Service5:AUTO-SCALING

6. Creating Auto-Scaling using launch template and launch configuration using services

7. Execution results

8. Termination process

Introduction of AWS

Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 200 AWS services are available. New services can be provisioned quickly, without the upfront fixed expense. This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements. This whitepaper provides you with an overview of the benefits of the AWS Cloud and introduces you to the services that make up the platform.

In 2006, Amazon Web Services (AWS) began offering IT infrastructure services to businesses as web services—now commonly known as cloud computing. One of the key benefits of cloud computing is the opportunity to replace upfront capital infrastructure expenses with low variable costs that scale with your business. With the cloud, businesses no longer need to plan for and procure servers and other IT infrastructure weeks or months in advance. Instead, they can instantly spin up hundreds or thousands of servers in minutes and deliver results faster.

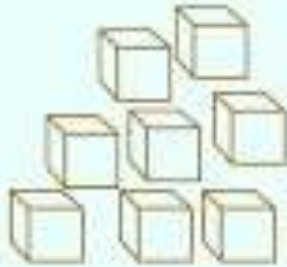


Today, AWS provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers hundreds of thousands of businesses in 190 countries around the world

Benefits of AWS

- **Trade fixed expense for variable expense** – Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources, and pay only for how much you consume.
- **Benefit from massive economies of scale** – By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay as-you-go prices.
- **Stop guessing capacity** – Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little capacity as you need, and scale up and down as required with only a few minutes' notice.
- **Stop spending money running and maintaining data centers** – Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers.










Benefits Of AWS



- **Stop spending money running and maintaining data centers** – Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers.
- **Go global in minutes** – Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide lower latency and a better experience for your customers at minimal cost.
- **Go global in minutes** – Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide lower latency and a better experience for your customers at minimal cost.

Popular AWS services

From sources across the web

- AWS lambda 
- Amazon Elastic Compute Cloud 
- Amazon RDS 
- Amazon S3 
- Amazon Virtual Private Cloud 
- Amazon DynamoDB 
- Amazon Simple Notification Service 
- Amazon CloudFront 
- AWS Elastic Beanstalk 



Implementation of the Architecture of Auto-Scaling with launch template and launch configuration

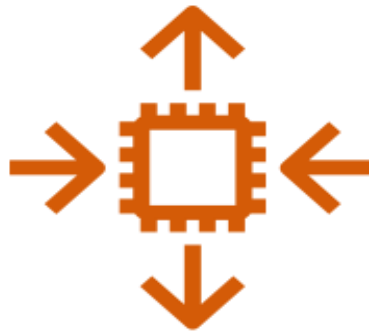
INTRODUCTION OF AUTO-SCALING

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including [Amazon EC2](#) instances and Spot Fleets, [Amazon ECS](#) tasks, [Amazon DynamoDB](#) tables and indexes, and [Amazon Aurora](#) Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using [Amazon EC2 Auto Scaling](#) to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

Benefits of Auto Scaling :

Auto Scaling your application leads to the following benefits:

- Better fault tolerance
- High availability of resources
- Better cost management
- High reliability of resources
- The high flexibility of resources



In Auto Scaling, creating a backup, and restoring the [data](#) is an essential part. This can be done by creating an EBS instance. EBS (elastic backup store) is responsible for creating volume backups. It consists of two backups, namely, snapshots and AMI.

Introduction to launch template

- A launch template is similar to a [launch configuration](#), in that it specifies instance configuration information. It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances. However, defining a launch template instead of a launch configuration allows you to have multiple versions of a launch template.
- With versioning of launch templates, you can create a subset of the full set of parameters. Then, you can reuse it to create other versions of the same launch template. For example, you can create a launch template that defines a base configuration without an AMI or user data script. After you create your launch template, you can create a new version and add the AMI and user data that has the latest version of your application for testing. This results in two versions of the launch template. Storing a base configuration helps you to maintain the required general configuration parameters. You can create a new version of your launch template from the base configuration whenever you want. You can also delete the versions used for testing your application when you no longer need them.

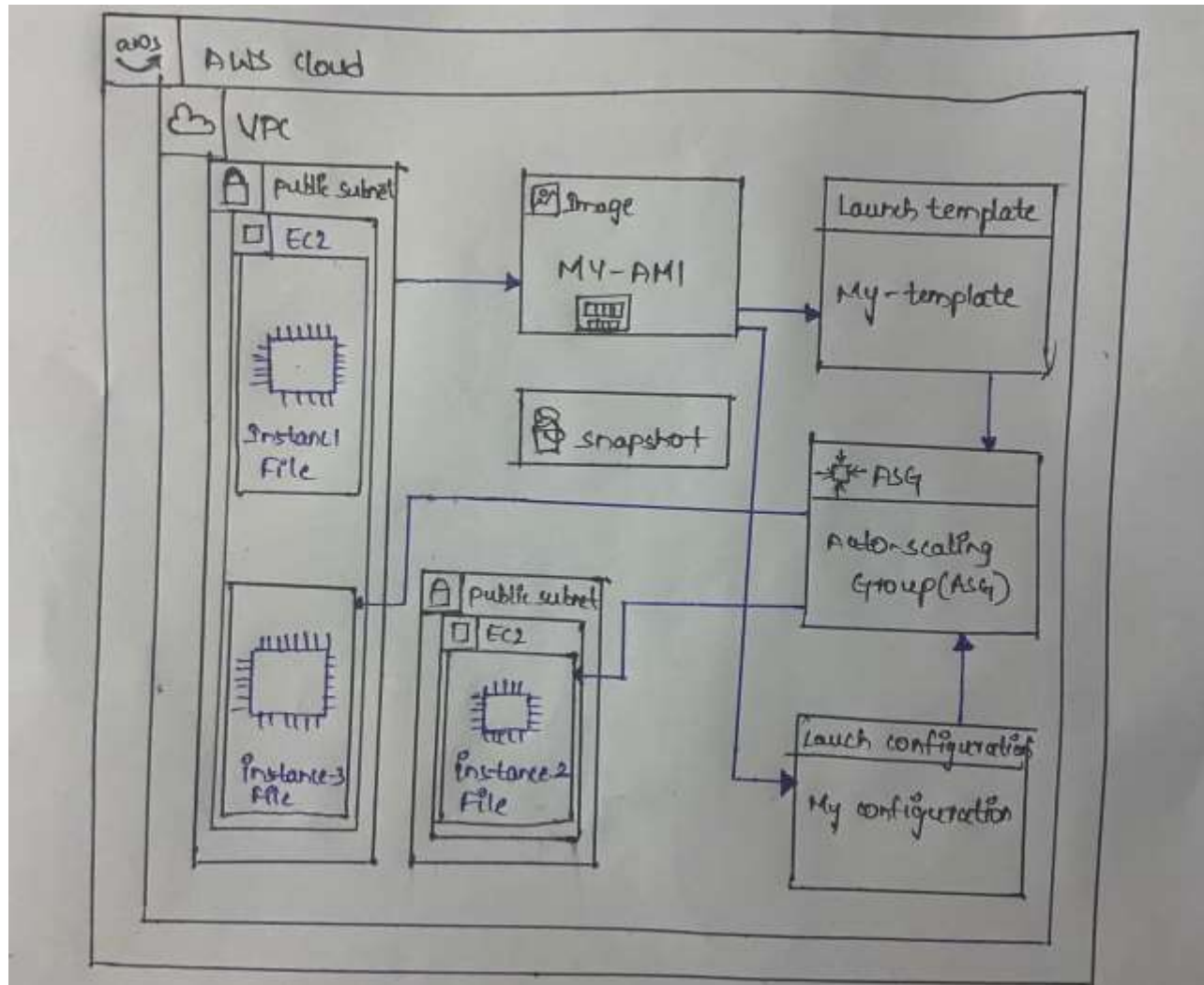
- We recommend that you use launch templates to ensure that you're accessing the latest features and improvements. Not all Amazon EC2 Auto Scaling features are available when you use launch configurations. For example, you cannot create an Auto Scaling group that launches both Spot and On-Demand Instances or that specifies multiple instance types. You must use a launch template to configure these features. For more information, see [Auto Scaling groups with multiple instance types and purchase options](#).
- With launch templates, you can also use newer features of Amazon EC2. This includes Systems Manager parameters (AMI ID), the current generation of EBS Provisioned IOPS volumes (io2), EBS volume tagging, [T2 Unlimited instances](#), Elastic Inference, and [Dedicated Hosts](#), to name a few. Dedicated Hosts are physical servers with EC2 instance capacity that are dedicated to your use. While Amazon EC2 [Dedicated Instances](#) also run on dedicated hardware, the advantage of using Dedicated Hosts over Dedicated Instances is that you can bring eligible software licenses from external vendors and use them on EC2 instances.
- When you create a launch template, all parameters are optional. However, if a launch template does not specify an AMI, you cannot add the AMI when you create your Auto Scaling group. If you specify an AMI but no instance type, you can add one or more instance types when you create your Auto Scaling group.

Introduction to launch configuration

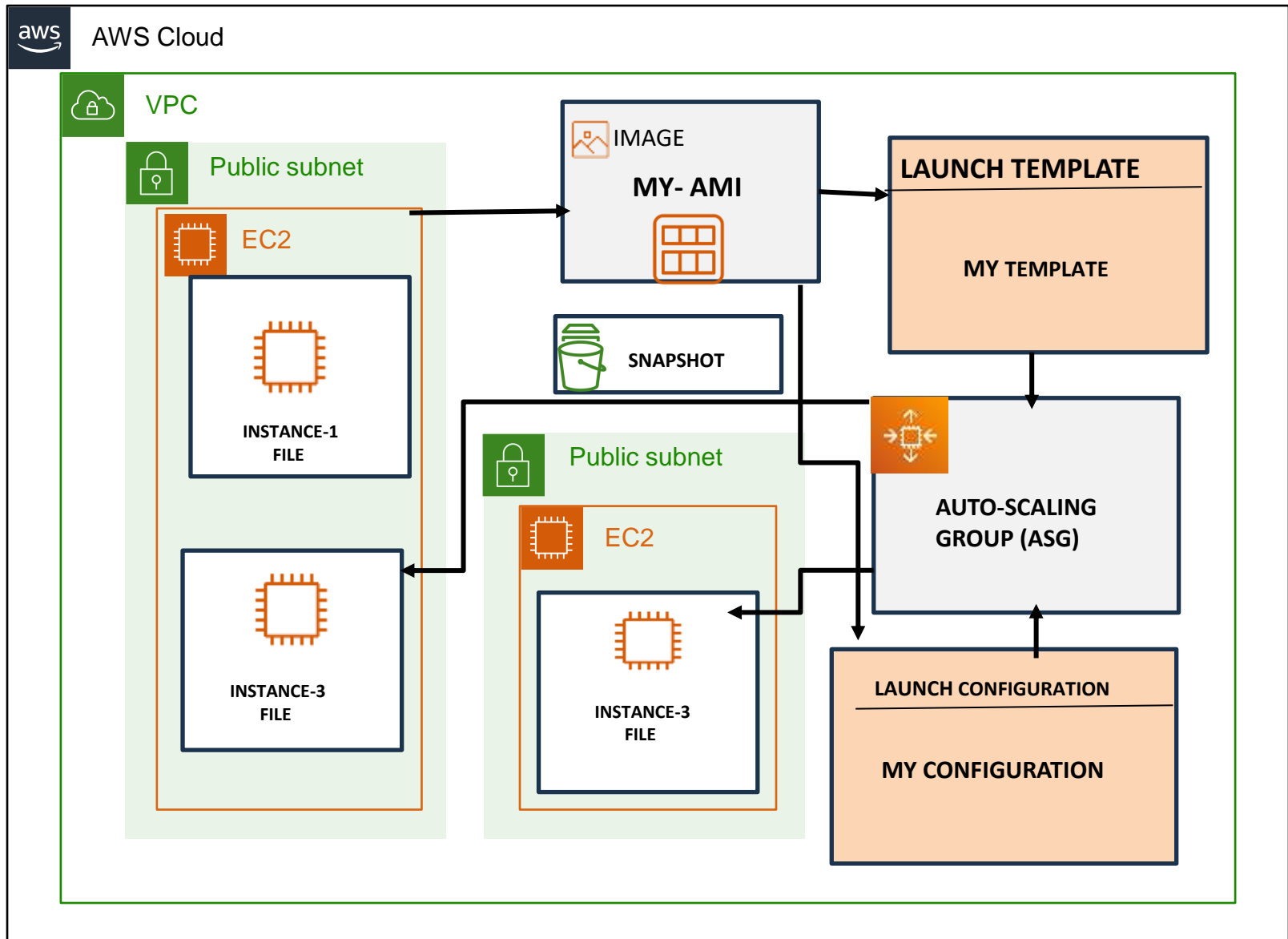
A *launch configuration* is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.

You can specify your launch configuration with multiple Auto Scaling groups. However, you can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. To change the launch configuration for an Auto Scaling group, you must create a launch configuration and then update your Auto Scaling group with it.

ROUGH ARCHITECTURE AUTO-SCALING WITH LAUNCH TEMPLATE AND LAUNCH CONFIGURATION



ARCHITECTURE OF AUTO-SCALING WITH LAUNCH TEMPLATE & LAUNCH CONFIGURATION



Services used in Architecture

Service1:VPC



Amazon VPC is the Virtual Private Cloud, which is an isolated cloud resource. It controls the virtual networking environment, such as resource placement, connectivity, and security. And it allows you to build and manage compatible VPC networks across cloud AWS resources and on-premise resources. Here, it improves security by applying rules for inbound and outbound connections. Also, it monitors VPC flow logs delivered to Amazon S3 and Amazon Cloudwatch to gain visibility over network dependencies and traffic patterns. Amazon VPC also detects anomalies in the patterns, prevents data leakage, and troubleshoots network connectivity and configuration issues.

Creating a VPC

To create a VPC with the name "my-vpc1," follow these steps:

- Sign into the AWS Management Console.
- Open the Virtual Private Cloud Service.
- Click on the "Create VPC" option.
- Give the VPC name as my-vpc1 and CIDR value as 10.0.0.0/24.
- Click on create VPC.

VPC > Your VPCs > Create VPC

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

☒ VPC only

☐ VPC and more

Name tag - optional

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

my-vpc1

IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input

☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/24

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ IPAM-allocated IPv6 CIDR block

VPC dashboard

EC2 Global View

Filter by VPC:
Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Your VPCs (2)

Find resources by attribute or tag

Name

VPC ID

State

IPv4 CIDR

IPv6 CIDR

DHCP o

☐

-

vpc-07654b3e9e9b4e9f5

Available

172.31.0.0/16

-

dopt-03

☐

my-vpc1

vpc-0ba6d5cd14bb39831

Available

10.0.0.0/24

-

dopt-03

Select a VPC above

- Go to route table and edit the name of route table as rt1-vpc1 that is created along with my-vpc1.

The screenshot shows the AWS Management Console interface. On the left, the 'Virtual private cloud' section is expanded, showing 'Route tables'. The main content area displays a list of route tables. One route table is selected, and an 'Edit Name' modal dialog is open, showing the new name 'rt1-vpc1' entered in the text field. Below the list, the details for the selected route table 'rtb-045654804a5432582' are visible, including tabs for 'Details', 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. A notification banner at the bottom of the details section says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button.

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC
-	rtb-06355f566e316e407	-	-	Yes	vpc-07654b3e9b4e9f5
rt1-vpc1	rtb-045654804a5432582	-	-	Yes	vpc-0ba6d5cd14bb39831

- Go to subnets and create a subnet and name it as subnet-1. Select the availability zone as south-1a, CIDR as 10.0.0.0/25 and select create subnet.

The screenshot displays the AWS Management Console interface for creating a new subnet. The top navigation bar includes the AWS logo, a search bar, and a list of services (IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, AWS Glue). The 'VPC' service is selected, and the 'Subnets' page is shown. A dropdown menu at the top left displays 'vpc-0ba6d5cd14bb39831 (my-vpc1)'. Below this, the 'Associated VPC CIDRs' section shows 'IPv4 CIDRs' with the value '10.0.0.0/24'. The 'Subnet settings' section is active, with the instruction 'Specify the CIDR blocks and Availability Zone for the subnet.' Below this, the 'Subnet 1 of 1' section contains the following fields:

- Subnet name:** A text input field containing 'subnet-1'. Below the field is a hint: 'Create a tag with a key of 'Name' and a value that you specify.' and a note: 'The name can be up to 256 characters long.'
- Availability Zone:** A dropdown menu showing 'Asia Pacific (Mumbai) / ap-south-1a'. A link for 'Info' is next to the label.
- IPv4 CIDR block:** A text input field containing '10.0.0.0/25'. A link for 'Info' is next to the label.

At the bottom of the 'Subnet settings' section, there is a link for 'Tags - optional'.

The footer of the console shows 'CloudShell', 'Feedback', 'Language', and copyright information: '© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences'.

Services

Search

[Alt+S]

IAM

EC2

VPC

S3

RDS

DynamoDB

Lambda

EFS

Cloud9

Elastic Beanstalk

AWS Glue

Mumbai

Kaveri

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

You have successfully created 1 subnet: subnet-0f9441b4d3225cd2d

Subnets (1/1)

Find resources by attribute or tag

Subnet ID = subnet-0f9441b4d3225cd2d

Clear filters

Name	Subnet ID	State	VPC	IPv4 CIDR
subnet-1	subnet-0f9441b4d3225cd2d	Available	vpc-0ba6d5cd14bb39831 my-...	10.0.0.0/25

subnet-0f9441b4d3225cd2d / subnet-1

Details

Flow logs

Route table

Network ACL

CIDR reservations

Sharing

Tags

Details

Subnet ID	Subnet ARN	State	IPv4 CIDR
subnet-0f9441b4d3225cd2d	arn:aws:ec2:ap-south-	Available	10.0.0.0/25

CloudShell

Feedback

Language

© 2023, Amazon Web Services India Private Limited or its affiliates.

Privacy

Terms

Cookie preferences

- Go to route tables, select the route table(rt1-vpc1) and go to subnet associations and select(subnet-1) and click on save associations.

The screenshot displays the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and a list of services. The left sidebar shows the 'Virtual private cloud' section with options like 'Your VPCs', 'Subnets', 'Route tables', and 'Internet gateways'. The main content area is titled 'Route tables (1/2)' and shows a table of route tables. The 'rt1-vpc1' route table is selected. Below the table, the 'Subnet associations' tab is active, showing 'Explicit subnet associations (0)'. A message states 'No subnet associations. You do not have any subnet associations.' The bottom of the console shows 'Subnets without explicit associations (1)'.

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC
-	rtb-06355f566e316e407	-	-	Yes	vpc-07654b3e9e9b4e9f5
rt1-vpc1	rtb-045654804a5432582	-	-	Yes	vpc-0ba6d5cd14bb39831

rtb-045654804a5432582 / rt1-vpc1

Details Routes Subnet associations Edge associations Route propagation Tags

Explicit subnet associations (0)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations			

You do not have any subnet associations.

Subnets without explicit associations (1)

VPC > Route tables > rtb-045654804a5432582 > Edit subnet associations

Edit subnet associations


Change which subnets are associated with this route table.

Available subnets (1/1)

< 1 > 

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	subnet-1	subnet-0f9441b4d3225cd2d	10.0.0.0/25	-	Main (rtb-045654804a5432582 / rt1-v...

Selected subnets

subnet-0f9441b4d3225cd2d / subnet-1 

Cancel

Save associations

- Create an internet gateway with the name igw1-vpc1

The screenshot shows the AWS Management Console interface for creating an internet gateway. The top navigation bar includes the AWS logo, a search bar, and a list of services. The main heading is 'Create internet gateway' with an 'Info' link. Below this is a descriptive paragraph about internet gateways. The 'Internet gateway settings' section contains a 'Name tag' field with the value 'igw1-vpc1'. The 'Tags - optional' section shows a single tag with the key 'Name' and the value 'igw1-vpc1'. At the bottom, there are 'Cancel' and 'Create internet gateway' buttons.

aws Services Search [Alt+S]

Services: IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, AWS Glue

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.

igw1-vpc1

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	
Name	igw1-vpc1	Remove

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create internet gateway](#)

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- Attach custom IGW to custom VPC.

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a list of services including IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. A green notification banner at the top states: "The following internet gateway was created: igw-0c6c660e7612271c1 - igw1-vpc1. You can now attach to a VPC to enable the VPC to communicate with the internet." with an "Attach to a VPC" button.

The left sidebar shows the navigation menu with categories like "Virtual private cloud" (containing VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections) and "Security".

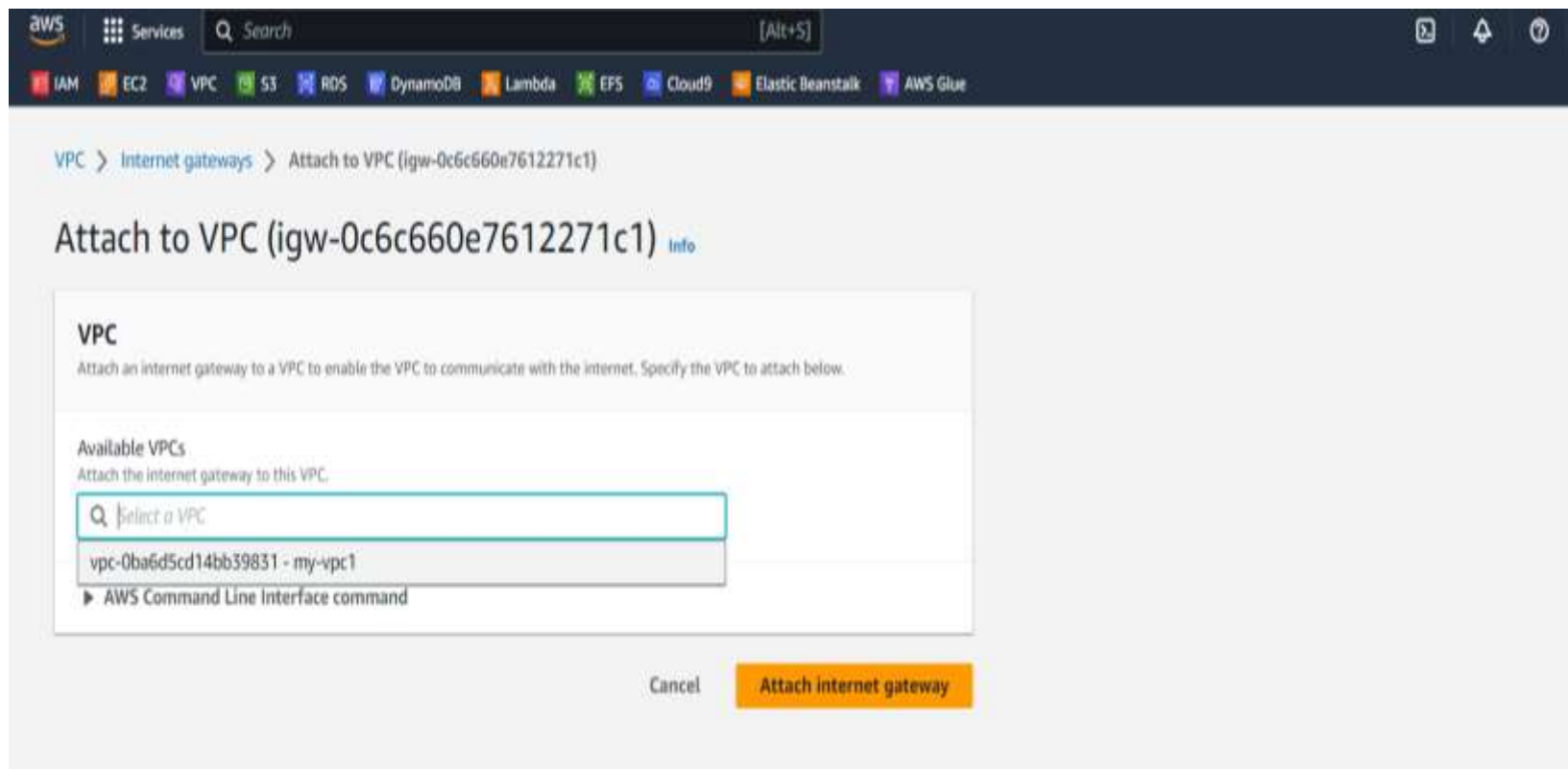
The main content area displays the details for the Internet Gateway **igw-0c6c660e7612271c1 / igw1-vpc1**. The "Details" section shows the following information:

Internet gateway ID	State	VPC ID	Owner
igw-0c6c660e7612271c1	Detached	-	543985770747

The "Tags" section shows a search bar and a table with one tag:

Key	Value
Name	igw1-vpc1

At the bottom of the console, there's a footer with "CloudShell", "Feedback", "Language", and copyright information: "© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences".



- Internet Gate Way is successfully attached to VPC.
- Now go to route tables, select rt1-vpc1, click on edit routes, go to add routes, and add 0.0.0.0/0 at destination and select our internet gateway in destination and save it.

VPC > Route tables > rtb-045654804a5432582 > Edit routes

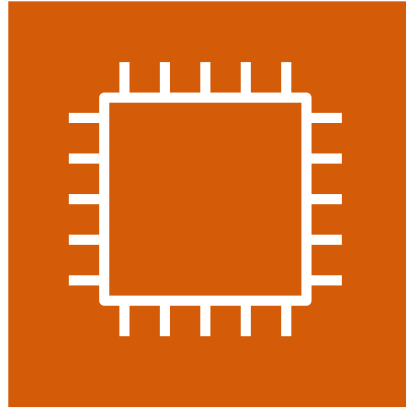
Edit routes

Destination	Target	Status	Propagated
10.0.0.0/24	<input type="text" value="local"/>	Active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="igw-"/> <input type="text" value="igw-0c6c660e7612271c1 (igw1-vpc1)"/>	-	No
<input type="button" value="Add route"/>			

Remove

Cancel Preview

SERVICE2:EC2



Amazon EC2 is one of the fastest-growing [cloud computing](#) AWS services, which offers virtual servers to manage any kind of workload. It facilitates the computing infrastructure with the best suitable processors, networking facilities, and storage systems. As a result, it supports adapting to the workloads precisely. Amazon EC2 provides a highly secure, reliable, performing computing infrastructure meeting business demands. And, it helps you to access resources quickly and dynamically scale capacities as per demands.

Creating an EC2 instance in a public subnet :

- Name your instance as instance-1.
- Select “Amazon Linux 2 AMI”.
- Instance type “t2. micro”.
- Select your existing key pair.
- Select your custom VPC, public subnet and enable the auto-assign public IP.
- In the security group section, select availability zone as south-1a.
 - create a new security group, add security groups that supports SSH and all the traffic.
- Configure storage gp2
- Launch your instance



Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

instance-1

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [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

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

[Quick Start](#)

▼ Summary

Number of instances [Info](#)

1

[Software Image \(AMI\)](#)

Amazon Linux 2023 AMI 2023.1.2...[read more](#)
ami-072ec2f4ea4a6f2cf

[Virtual server type \(instance type\)](#)

t2.micro

[Firewall \(security group\)](#)

New security group

[Storage \(volumes\)](#)

1 volume(s) - 8 GiB

[Free tier](#): In your first year

[Cancel](#)

[Launch instance](#)

[Review commands](#)

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour

☒ All generations

Compare instance types

▼ 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 - *required*

Kaveri_vegi

 [Create new key pair](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

```
vpc-0ba6d5cd14bb39831 (my-vpc1)
10.0.0.0/24
```

Subnet Info

subnet-0f9441b4d3225cd2d

subnet-1

VPC: vpc-0ba6d5cd14bb39831 Owner: 543965770747

[Create new subnet](#)

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...[read more](#)
ami-072ec8f4ea4a6f2cf

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

i Free tier: In your first year

Cancel

Launch instance

Review commands

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

ssh

TCP

22

Source type [Info](#)

Source [Info](#)

Description - optional [Info](#)

Anywhere

Q Add CIDR, prefix list or security

0.0.0.0/0 X ::/0 X

e.g. SSH for admin desktop

▼ Security group rule 2 (All, All, Multiple sources)

Remove

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

All traffic

All

All

Source type [Info](#)

Source [Info](#)

Description - optional [Info](#)

Anywhere

Q Add CIDR, prefix list or security

0.0.0.0/0 X ::/0 X

e.g. SSH for admin desktop

⚠ 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.

X

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.1.2...[read more](#)

ami-072ec8f4ea4a6f2cf

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year

X

Cancel

Launch instance

[Review commands](#)

The screenshot displays the AWS Management Console interface. At the top, the navigation bar includes the AWS logo, a search bar, and a list of services: IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. The left-hand sidebar contains a 'New EC2 Experience' notification and a menu with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', 'AMI Catalog', 'Elastic Block Store', and 'Volumes'. The main content area is titled 'Instances (1/4) info' and features a search bar with the placeholder text 'Find instance by attribute or tag (case-sensitive)'. Below the search bar is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. A single instance, 'instance-1', is listed with ID 'i-0ee10aa2ada64240b', state 'Running', type 't2.micro', and status 'Initializing'. Below the table, a detailed view for 'Instance: i-0ee10aa2ada64240b (instance-1)' is shown, with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The 'Details' tab is active, displaying an 'Instance summary' with the following information:

Instance summary		
Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ee10aa2ada64240b (instance-1)	13.127.129.234 open address	10.0.0.42
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-
Hostname type	Private IP DNS name (IPv4 only)	
IP name: in-10.0.0.42 ap-south-1 compute internal	in-10.0.0.42 ap-south-1 compute internal	

The bottom of the console shows a footer with 'CloudShell', 'Feedback', 'Language', and copyright information for Amazon Web Services India Private Limited or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

- Now “instance-1” is successfully launched
- Now connect instance-1

Connect to your instance i-0ee10aa2ada64240b (instance-1) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID
i-0ee10aa2ada64240b (instance-1)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
13.127.129.234

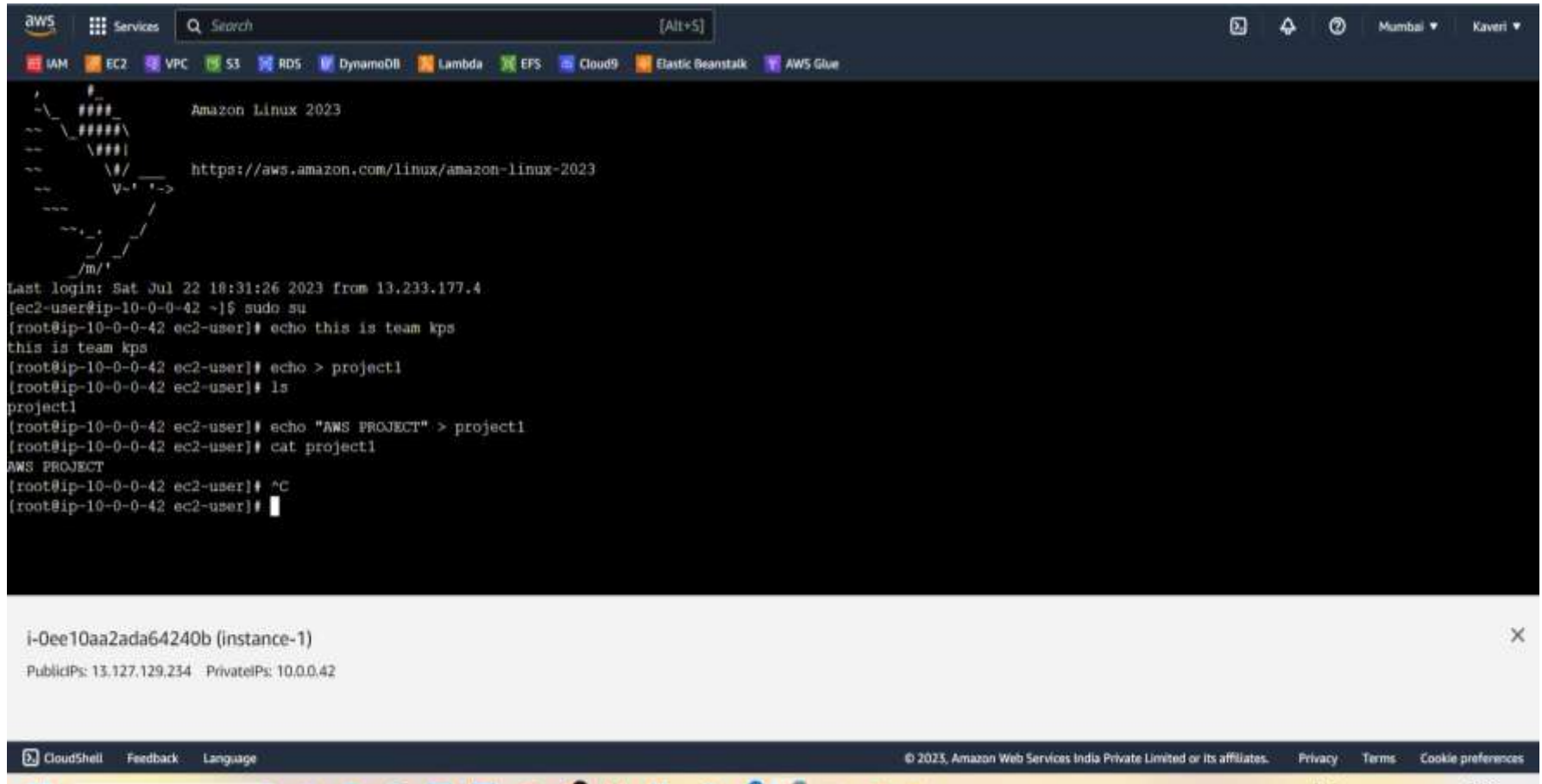
User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel Connect

- Perform the commands as shown.



The screenshot displays the AWS CloudShell interface. At the top, there's a navigation bar with the AWS logo, a 'Services' menu, a search bar, and a list of services including IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. The main terminal area shows the following content:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sat Jul 22 18:31:26 2023 from 13.233.177.4
[ec2-user@ip-10-0-0-42 ~]$ sudo su
[root@ip-10-0-0-42 ec2-user]# echo this is team kps
this is team kps
[root@ip-10-0-0-42 ec2-user]# echo > project1
[root@ip-10-0-0-42 ec2-user]# ls
project1
[root@ip-10-0-0-42 ec2-user]# echo "AWS PROJECT" > project1
[root@ip-10-0-0-42 ec2-user]# cat project1
AWS PROJECT
[root@ip-10-0-0-42 ec2-user]# ^C
[root@ip-10-0-0-42 ec2-user]#
```

Below the terminal, a metadata box for instance `i-0ee10aa2ada64240b (instance-1)` is visible, showing Public IPs: 13.127.129.234 and Private IPs: 10.0.0.42. The bottom of the interface includes a footer with 'CloudShell', 'Feedback', 'Language', and copyright information for Amazon Web Services India Private Limited or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Service3: AMI



An Amazon Machine Image (AMI) is a supported and maintained image provided by AWS that provides the information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you require multiple instances with the same configuration. You can use different AMIs to launch instances when you require instances with different configurations.

An AMI includes the following:

- One or more Amazon Elastic Block Store (Amazon EBS) snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

- Go to custom instance(instance-1) actions.
- Create image my-ami

The screenshot shows the AWS Management Console interface for creating a new Amazon Machine Image (AMI). The breadcrumb navigation at the top indicates the path: EC2 > Instances > i-0ee10aa2ada64240b > Create image. The main heading is 'Create image' with an 'info' link. Below the heading is a descriptive paragraph: '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.' The form contains several sections: 'Instance ID' with a dropdown menu showing 'i-0ee10aa2ada64240b (instance-1)'; 'Image name' with a text input field containing 'my-ami' and a note 'Maximum 127 characters. Can't be modified after creation.'; 'Image description - optional' with a text input field containing 'ami' and a note 'Maximum 255 characters.'; 'No reboot' with an unchecked 'Enable' checkbox; and 'Instance volumes' which is a table with columns: Storage type, Device, Snapshot, Size, Volume type, IOPS, Throughput, Delete on termination, and Encrypted.

EC2 > Instances > i-0ee10aa2ada64240b > Create 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-0ee10aa2ada64240b (instance-1)

Image name

my-ami

Maximum 127 characters. Can't be modified after creation.

Image description - optional

ami

Maximum 255 characters.

No reboot

☐ Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
--------------	--------	----------	------	-------------	------	------------	-----------------------	-----------

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

- Check AMI and rename the custom AMI image and check availability status.

The screenshot displays the AWS Management Console interface for Amazon Machine Images (AMIs). The left sidebar shows navigation options like EC2 Dashboard, Instances, and Images. The main content area is titled 'Amazon Machine Images (AMIs) (1/1)'. A table lists the AMIs, with one entry named 'my-ami' having ID 'ami-02ca1de7278103e38'. An 'Edit Name' dialog box is open over the table, showing the current name 'my-ami' in a text input field, with 'Cancel' and 'Save' buttons. Below the table, the 'Details' tab for the selected AMI is shown, displaying various attributes:

AMI ID: ami-02ca1de7278103e38			
Details			
AMI ID	ami-02ca1de7278103e38	Image type	machine
AMI name	my-ami	Platform details	Linux/UNIX
Root device name	/dev/xvda	Architecture	x86_64
Boot mode		Source	543985770747/my-ami
Owner account ID	543985770747	Usage operation	RunInstances
Status	Pending	Virtualization type	hvm
State reason		Creation date	
		Kernel ID	

Service4:Snapshot

- Stores AMI image by creating automatically snapshot



Creating launch template

- Create launch template
- Name template (my-template-asg)
- Description as webservice

- Automatically my-AMI is attached
- Choose t2-micro as instance type
- Choose keypair existing one
- Select subnet ap-south-1a
- Click on create launch template

The screenshot shows the AWS Management Console interface for creating a launch template. The top navigation bar includes the AWS logo, a search bar, and a list of services (IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, AWS Glue). The breadcrumb trail indicates the path: EC2 > Launch templates > Create 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 &, %, ^, etc.

Template version description

Max 128 chars

Auto Scaling guidance [help](#)
Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Summary

- Software Image (AMI)
ami-02ca1de7278103e38
- Virtual server type (instance type)
t2.micro
- Firewall (security group)
-
- Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year, includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.

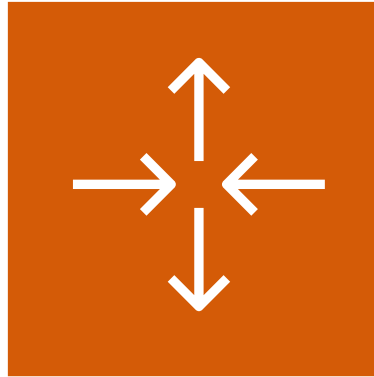
- Launch template is created

The screenshot shows the AWS Management Console interface for the 'Launch templates' page. The top navigation bar includes the AWS logo, a search bar, and various service icons like IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. The left sidebar shows the 'New EC2 Experience' section with links to EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates (highlighted), Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, and Volumes. The main content area is titled 'Launch templates (1)' and includes a search bar with the placeholder text 'Filter by tags or properties or search by keyword'. Below the search bar is a table with the following data:

Launch template ID	Launch template name	Default version	Latest version
lt-0f5e8b7a21e771c36	my-template-asg	1	1

Below the table, there is a section titled 'Select a launch template' with a search bar and a list of launch templates. The bottom of the console shows the 'CloudShell' button, 'Feedback', 'Language', and the copyright notice '© 2023 Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences'.

SERVICE5:Auto-scaling



This AWS service scales computing capacity to meet the demands accurately. And it is achieved by adding or removing EC2 instances automatically. There are two types of scaling such as dynamic scaling and predictive scaling. Here, dynamic scaling responds to the presently changing demands, whereas predictive scaling responds based on predictions. Through Amazon EC2 Auto-scaling, you can identify the unhealthy EC2 instances, terminate them, and replace them with new instances.

Benefits of Auto-scaling

- Better fault tolerance. Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it. You can also configure Amazon EC2 Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Amazon EC2 Auto Scaling can launch instances in another one to compensate.
- Better availability. Amazon EC2 Auto Scaling helps ensure that your application always has the right amount of capacity to handle the current traffic demand.
- Better cost management. Amazon EC2 Auto Scaling can dynamically increase and decrease capacity as needed. Because you pay for the EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

- Create AUTO –SCALING
- Open Auto scaling groups
- Enter Name name of Auto-scaling group (ASG1)
- Launch template choose the custom (my-template-asg)

The screenshot shows the AWS Management Console interface for creating an Auto Scaling Group. The top navigation bar includes links to IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. The left sidebar shows a step-by-step process: Step 1: Choose launch template (selected), Step 2: Choose instance launch options, Step 3 - optional: Configure advanced options, Step 4 - optional: Configure group size and scaling policies, Step 5 - optional: Add notifications, Step 6 - optional: Add tags, and Step 7: Review.

The main content area is titled "Choose launch template" with an "info" link. Below the title is a description: "Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group."

The "Name" section contains the label "Auto Scaling group name" and the instruction "Enter a name to identify the group." Below this is a text input field containing "ASG1". A note below the field states: "Must be unique to this account in the current Region and no more than 255 characters."

The "Launch template" section has an "info" link. A blue information box contains the following text: "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." Below this box, the "Launch template" section includes the instruction "Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups." Below this instruction is a dropdown menu showing "my-template-asg" and a refresh button.

The footer of the console includes "CloudShell", "Feedback", "Language", and copyright information: "© 2023, Amazon Web Services India Private Limited or its affiliates." It also includes links for "Privacy", "Terms", and "Cookie preferences".

- Next
- Choose network vpc custom(my-vpc1)
- Select subnets
- Load balancer attach to new load balancer

The screenshot displays the AWS Management Console interface during the configuration of an Auto Scaling group. The left sidebar shows the progress through seven steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (optional, Configure advanced options), Step 4 (optional, Configure group size and scaling policies), Step 5 (optional, Add notifications), Step 6 (optional, Add tags), and Step 7 (Review). The main content area is titled 'Configure advanced options - optional' and includes a description: '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.'

The 'Load balancing' section is currently active, with the instruction: 'Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.' Three radio button options are presented:

- ☐ 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.

Below this, the 'Attach to a new load balancer' section is visible, with the instruction: 'Define a new load balancer to create for attachment to this Auto Scaling group.' The 'Load balancer type' section shows two options:

- ☒ Application Load Balancer: HTTP, HTTPS
- ☐ Network Load Balancer: TCP, UDP, TLS

The bottom of the console shows the footer with 'CloudShell', 'Feedback', 'Language', and copyright information for Amazon Web Services India Private Limited.

- Choose internet-facing
- Create a target group (default routing) select ASG1-1

The screenshot shows the AWS Management Console interface for configuring a new load balancer. The top navigation bar includes the AWS logo, a search bar, and a list of services: IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. The main content area is titled 'Load Balancing' and features two tabs: 'Internal' and 'Internet-facing', with the latter being selected. Below the tabs, the 'Network mapping' section explains that the load balancer will use the same VPC and Availability Zone as the Auto Scaling group. The 'VPC' section shows a selected VPC: 'vpc-07654b3e9e9b4e9f5'. The 'Availability Zones and subnets' section lists three Availability Zones with their corresponding subnets: 'ap-south-1a' (subnet-0368165a2f326ba1b), 'ap-south-1c' (subnet-0de5fd91ca1a44a1b), and 'ap-south-1b' (subnet-0f7e4b6996ee39f42). The 'Listeners and routing' section shows a listener configured for 'HTTP' on port '80', routing to 'ASG1-1 | HTTP'. Finally, the 'Tags - optional' section provides information about adding tags to the load balancer.

Services Search [Alt+S]

IAM EC2 VPC S3 RDS DynamoDB Lambda EFS Cloud9 Elastic Beanstalk AWS Glue

Internal Internet-facing

Network mapping
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
vpc-07654b3e9e9b4e9f5

Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

Availability Zone	Subnet
ap-south-1a	subnet-0368165a2f326ba1b
ap-south-1c	subnet-0de5fd91ca1a44a1b
ap-south-1b	subnet-0f7e4b6996ee39f42

Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	ASG1-1 HTTP

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

- Next
- Edit Configure group size and scaling policies as show

The screenshot displays the AWS Management Console interface for creating an Auto Scaling group. The top navigation bar includes the AWS logo, a search bar, and a list of services: IAM, EC2, VPC, S3, RDS, DynamoDB, Lambda, EFS, Cloud9, Elastic Beanstalk, and AWS Glue. The breadcrumb trail indicates the current path: EC2 > Auto Scaling groups > Create Auto Scaling group.

The left sidebar shows a sequence of steps for the creation process:

- Step 1: Choose launch template
- Step 2: Choose instance launch options
- Step 3 - optional: Configure advanced options
- Step 4 - optional: **Configure group size and scaling policies** (currently active)
- Step 5 - optional: Add notifications
- Step 6 - optional: Add tags
- Step 7: Review

The main content area is titled "Configure group size and scaling policies - optional" with an "Info" link. Below the title, a descriptive text states: "Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group."

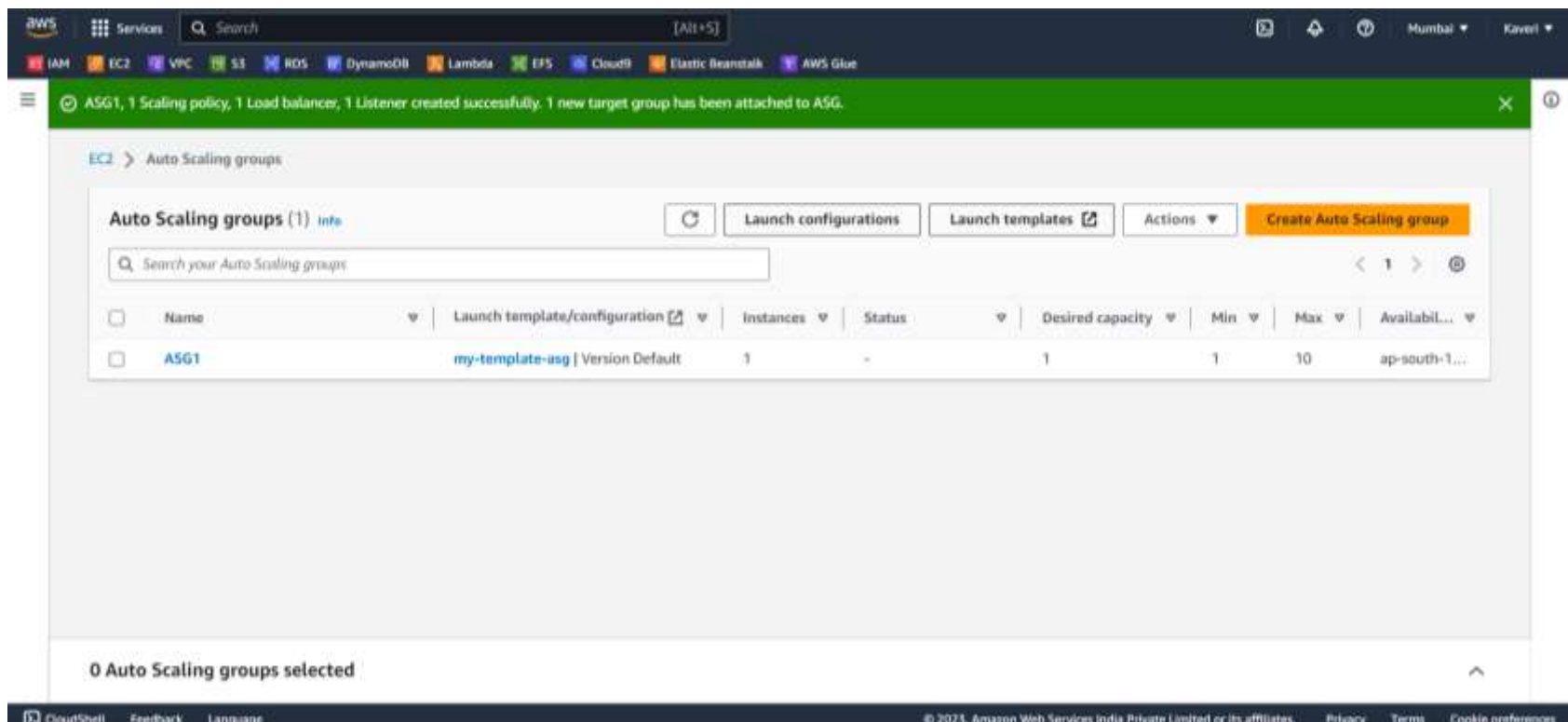
The "Group size - optional" section contains the following configuration fields:

- Desired capacity:** A text input field containing the value "1".
- Minimum capacity:** A text input field containing the value "1".
- Maximum capacity:** A text input field containing the value "10", accompanied by a small up/down arrow icon.

Below the group size section, the "Scaling policies - optional" section is visible but currently empty.

The footer of the console includes links for CloudShell, Feedback, and Language, along with the copyright notice "© 2023, Amazon Web Services India Private Limited or its affiliates." and links for Privacy, Terms, and Cookie preferences.

- Add notifications –optional
- Add tag name(name) and value(ASG-VMs)
- Create Auto-scaling group



- After creating Auto-scaling group
- New instance is create automatically
- Now connect the new instance naming it as (ASG-VMs)

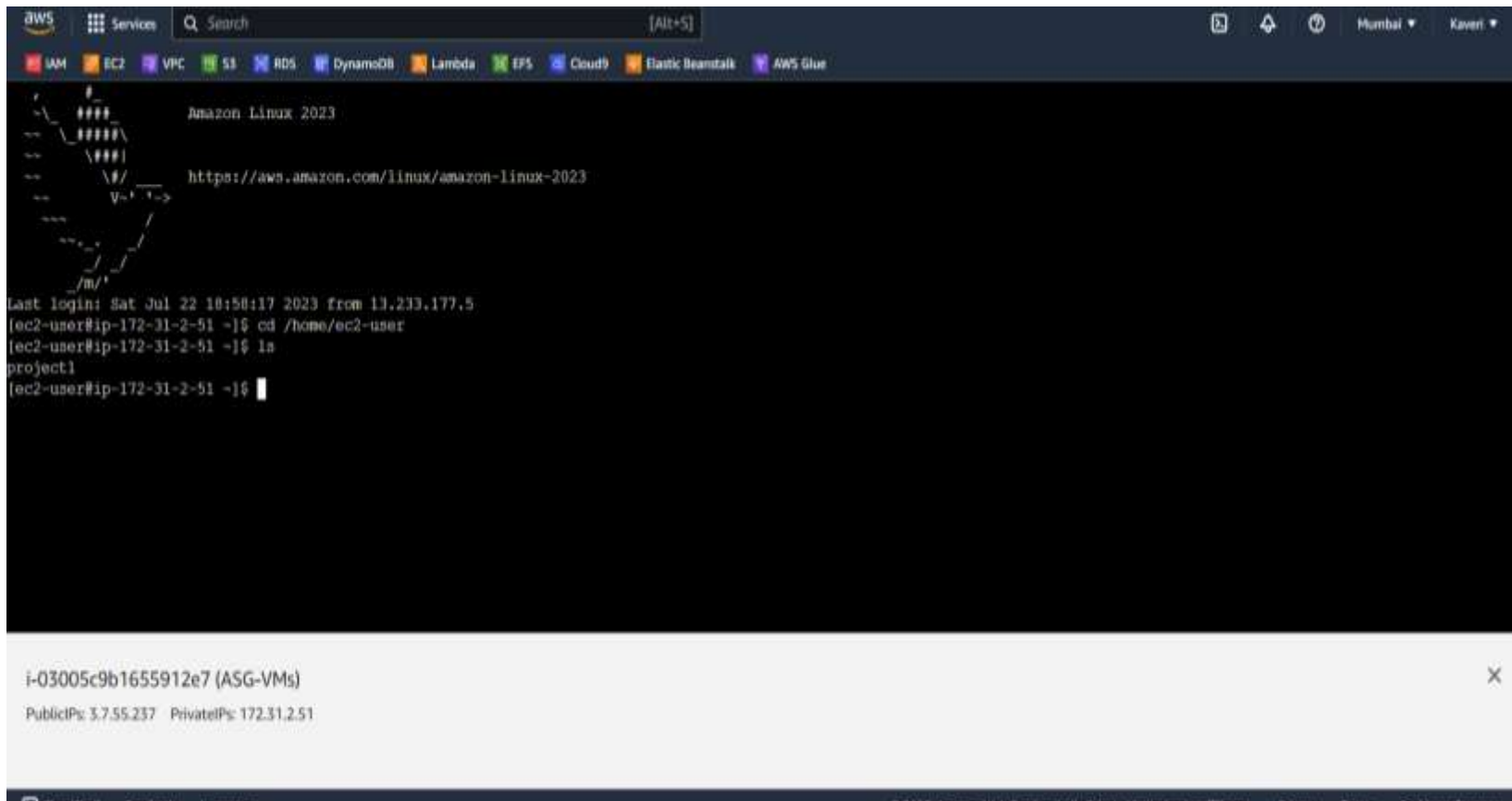
The screenshot displays the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, a search bar, and various service icons. The left sidebar shows the 'Instances' section selected. The main content area shows a list of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. One instance, 'ASG-VMs', is highlighted, and an 'Edit Name' dialog box is open, showing the name 'ASG-VMs' in a text input field. Below the list, the 'Instance summary' for the selected instance is shown, including its ID, public IPv4 address, private IPv4 address, and instance state (Running).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Myapplication...	i-032f9fbad8a091d74	Terminated	t3.micro	-	No alarms	ap-south-1c	-
instance-1	i-0ee10xaa2ada64240b	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	-
Myapplication...	i-04cc06a09fdeabe24	Terminated	t3.micro	-	No alarms	ap-south-1b	-
ASG-VMs	i-03005c9b1655912e7	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-7-55-237.ap-south-1.compute.amazonaws.com

Instance summary

- Instance ID: i-03005c9b1655912e7 (ASG-VMs)
- Public IPv4 address: 3.7.55.237 | open address
- Private IPv4 addresses: 172.31.2.51
- Instance state: Running
- Public IPv4 DNS: ec2-3-7-55-237.ap-south-1.compute.amazonaws.com | open address

- Change user name to ec2-user
- After connecting the instance(ASG-VMs)
- Give command `cd /home/ec2-user`



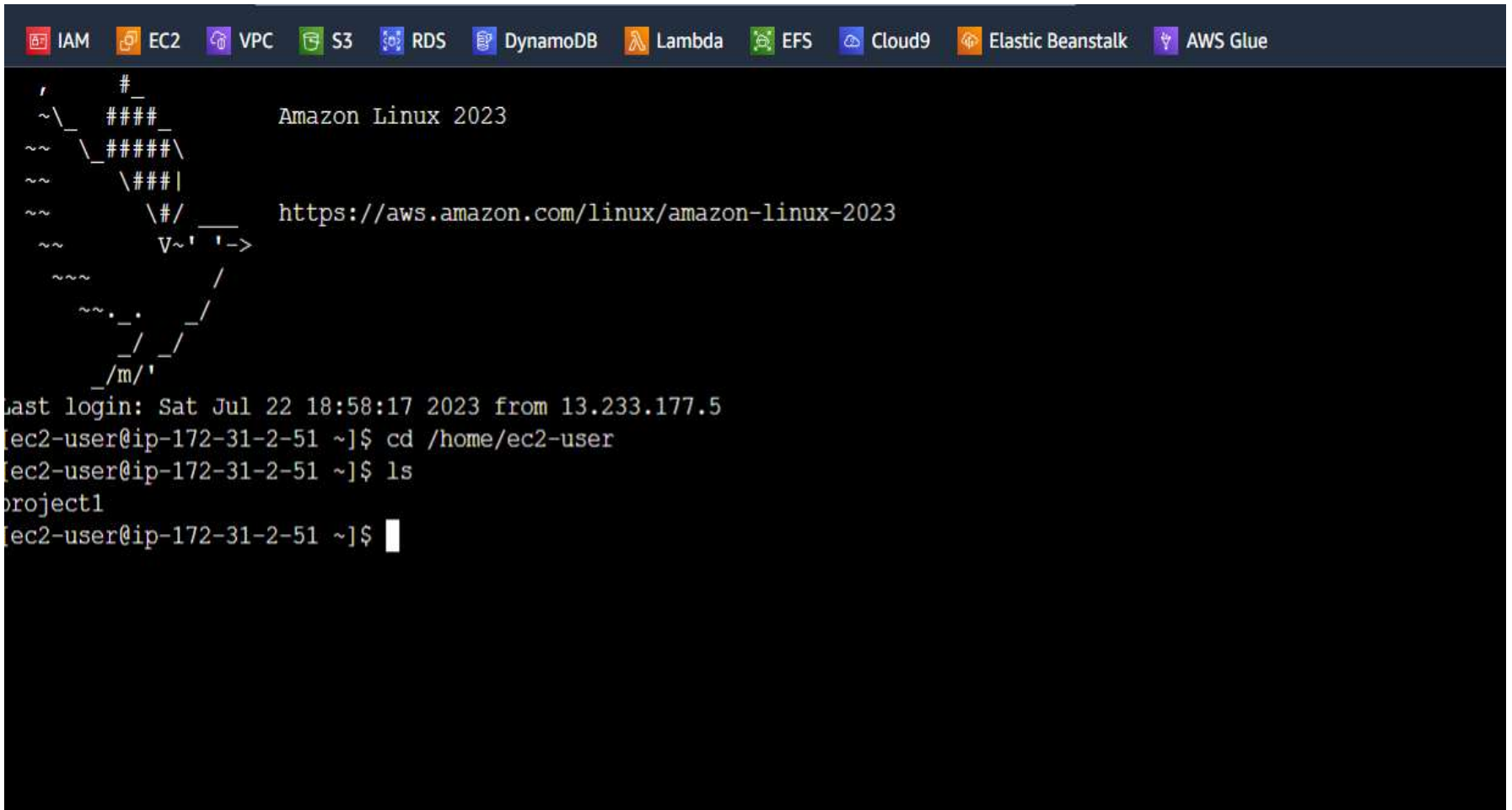
```
aws Services Search [Alt+S]
IAM EC2 VPC S3 RDS DynamoDB Lambda EFS Cloud9 Elastic Beanstalk AWS Glue

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sat Jul 22 10:50:17 2023 from 13.233.177.5
[ec2-user@ip-172-31-2-51 ~]$ cd /home/ec2-user
[ec2-user@ip-172-31-2-51 ~]$ ls
project1
[ec2-user@ip-172-31-2-51 ~]$
```

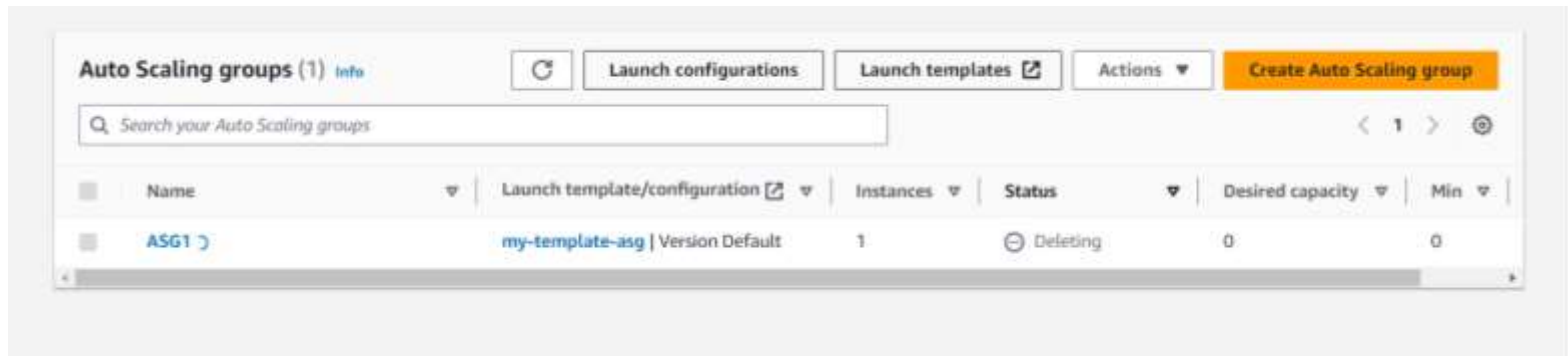
i-03005c9b1655912e7 (ASG-VMs)
PublicIP: 3.7.55.237 PrivateIP: 172.31.2.51

Execution results

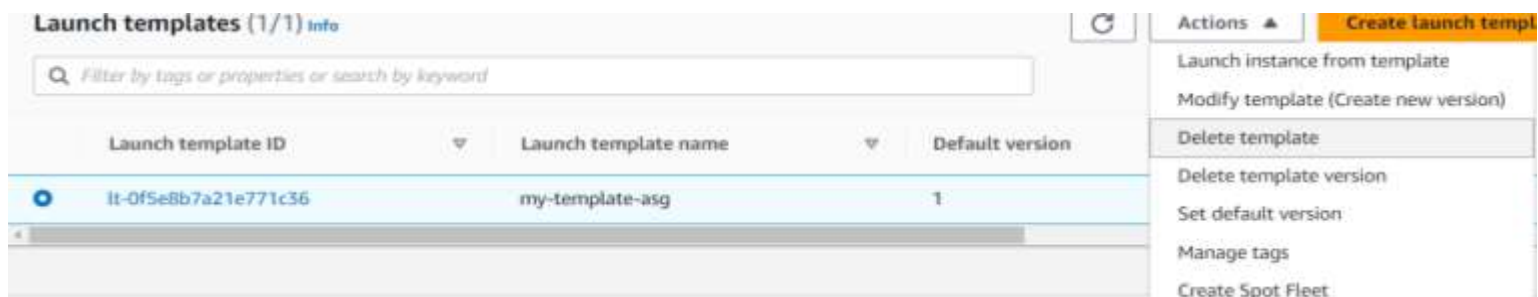


Termination process

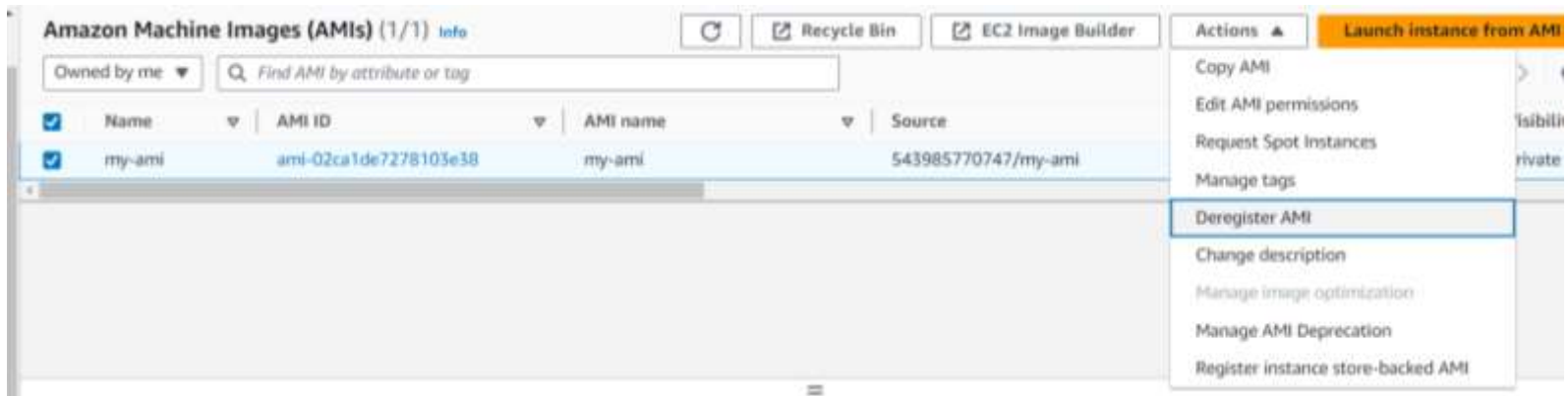
- Delete Auto scaling group



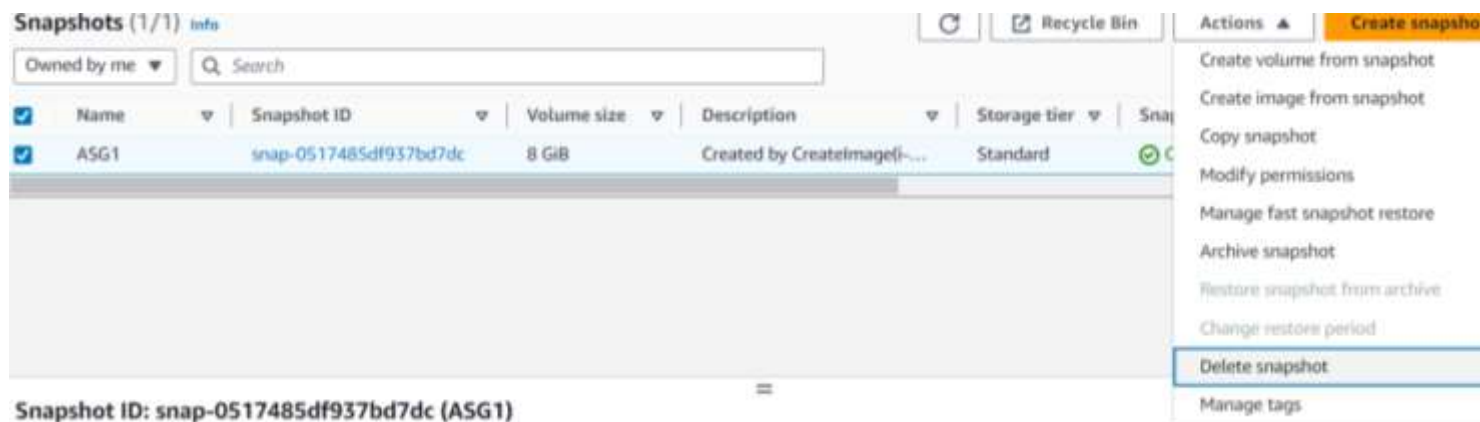
- Delete launch template



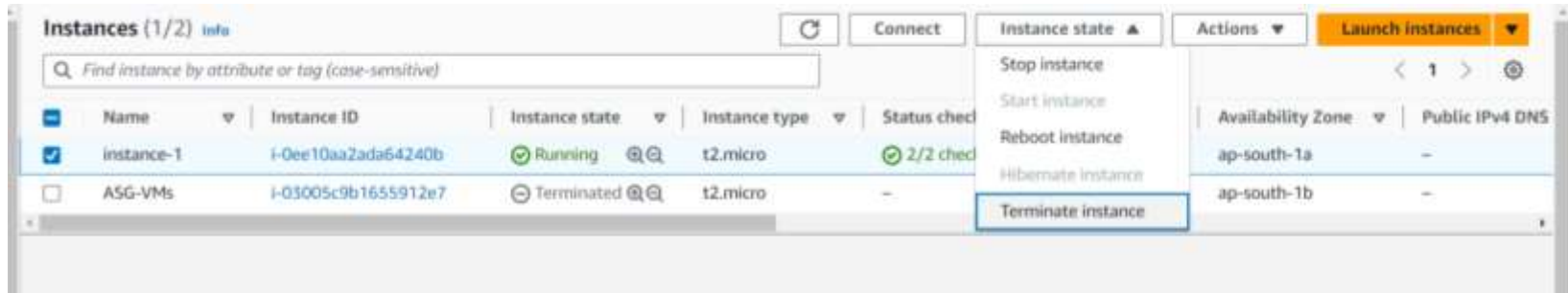
- Deregister AMI image



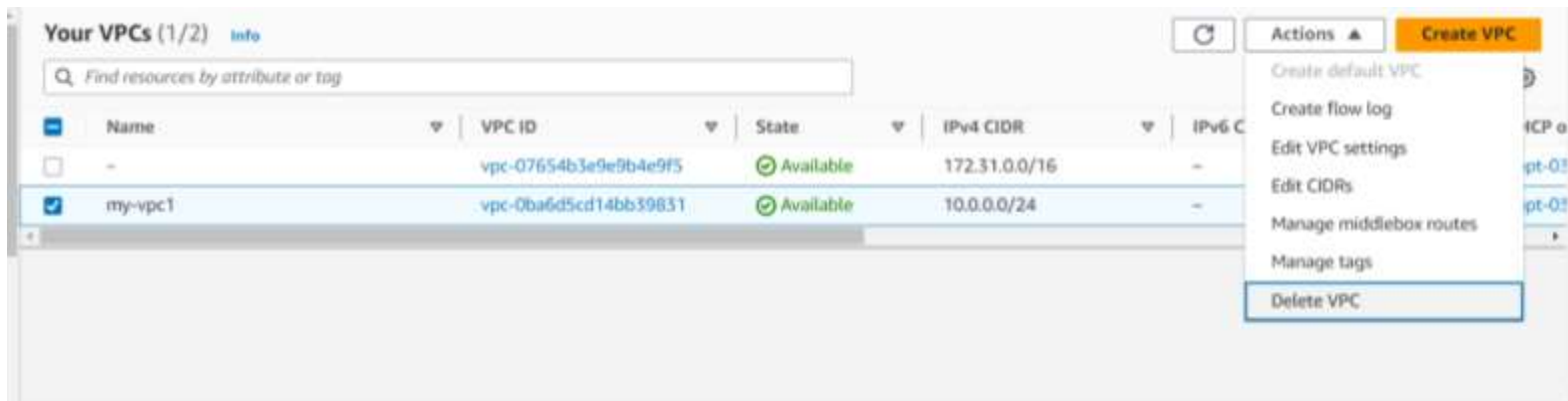
- Delete snapshot created due to AMI image



- ASG-VMs instance is terminated automatically after deletion of ASG
- Terminate the instance(instance-1)



- Delete the my-vpc1 in VPC service



- Automatically subnet and internetgateway and routetable are cleared after deleting VPC



THANK you

TEAM KPS

BRANCH:AIDS

V.VENKATA KAVERI(21B91A54I4)

V.POOJITHA(21B91A54I3)

V.SHYNIE PRAVALLIKA(21B91A54I7)