

Capture Moments - AWS-Powered Photographer Booking System

Project Description:

Capture Moments is a cloud-native photographer booking platform that connects professional photographers with clients in real time. Built using scalable AWS infrastructure, the system enables users to explore photographer portfolios, filter by category (wedding, portrait, event, etc.), and book appointments securely and efficiently.

The system is built using **Amazon EC2** for hosting the application backend, ensuring reliable performance and full control over the server environment. **Amazon DynamoDB** is used database to manage user data, bookings, and photographer listings with high availability. **AWS IAM** is implemented to manage secure access between services, define user roles (admin, photographer, client), and enforce fine-grained permission control.

Scenario 1:

Riya is planning a wedding and needs to book a professional photographer. She visits the Capture Moments platform, logs into her account, and begins her search. On the home page, she sees a search bar and filtering options such as: Photography Type (Wedding, Event, Portrait, etc.), Location, Availability Dates, Ratings etc. Riya selects the "Wedding" category and filters photographers available in her city.

The platform displays a list of photographers in a card layout. Each photographer's card includes:

-  Profile Picture
-  Photographer Name (e.g., *Amit Lensman*)
-  Specialties (e.g., Wedding, Portrait)
-  Location (e.g., Hyderabad, India)
-  Average Rating (e.g., ★★★★☆)
-  Years of Experience (e.g., 5+ years)
-  Mini Portfolio Preview (3–4 sample images)

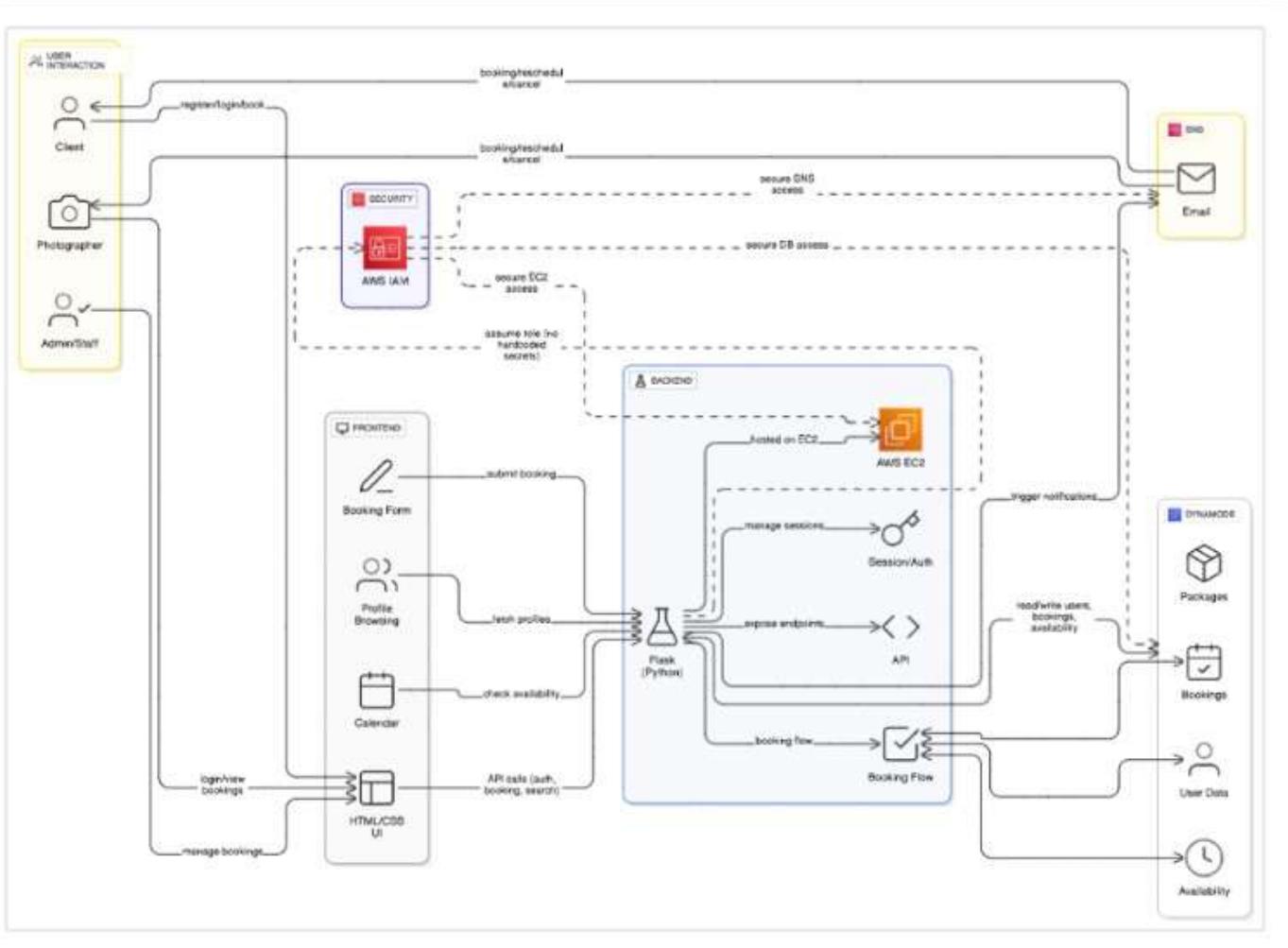
Scenario 2:

A client (like Riya) visits the platform and searches for wedding photographers. She sees Amit's profile in the results. On the webpage, his photographer card includes:

-  Name: *Amit Lensman*
-  Specialties: Wedding, Portrait
-  Rating: 4.8/5
-  Location: Hyderabad
-  3–4 sample portfolio image

AWS ARCHITECTURE

This AWS-based architecture powers a scalable and secure web application using Amazon EC2 for hosting the backend, with a lightweight framework like Flask handling core logic. Application data is stored in Amazon DynamoDB, ensuring fast, reliable access, while user access is managed through AWS IAM for secure authentication and control. Real-time alerts and system and the communication and user engagement.



Project Workflow:

1. AWS Account Setup and Login

Activity 1.1: Set up an AWS account if not already done.

Activity 1.2: Log in to the AWS Management Console

2. DynamoDB Database Creation and Setup

Activity 2.1: Create a DynamoDB Table.

Activity 2.2: Configure Attributes for User Data and Book Requests.

3. IAM Role Setup

Activity 3.1: Create IAM Role

Activity 3.2: Attach Policies

4. EC2 Instance Setup

Activity 4.1: Launch an EC2 instance to host the Flask application.

Activity 4.2: Configure security groups for HTTP, and SSH access.

5. Deployment on EC2

Activity 5.1: install

dependencies

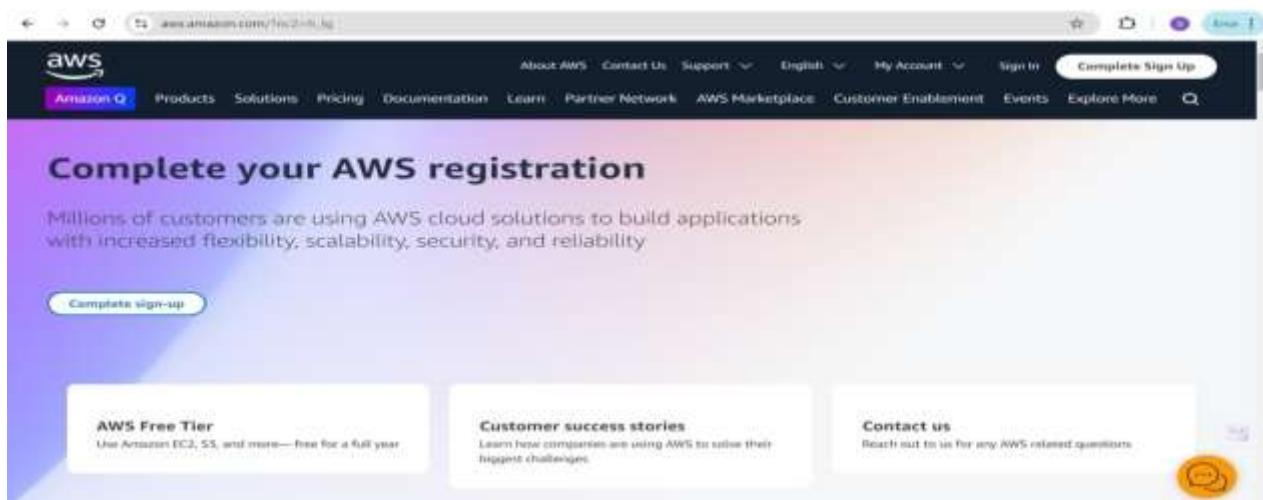
Milestone 6: Testing and Deployment

Activity 6.1 Conduct functional testing to verify booking a photographer and view photographers:

Milestone 1: AWS Account Setup and Login

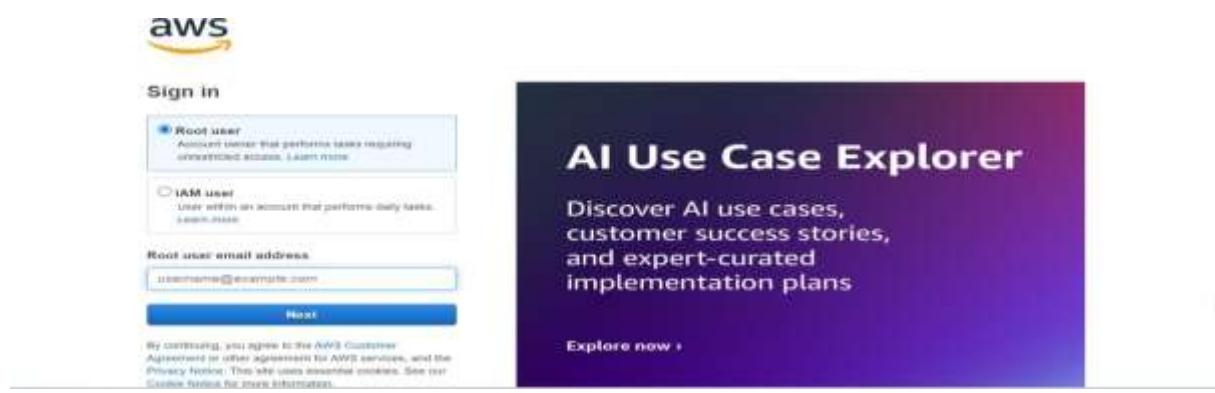
Activity 1.1: Set up an AWS account if not already done.

- Sign up for an AWS account and configure billing settings.
 -



● Activity 1.2: Log in to the AWS Management Console

- After setting up your account, log in to the [AWS Management Console](#)



Milestone 2: DynamoDB Database Creation and Setup

- **Activity 2.1: Create a DynamoDB Table.**

- In the AWS Console, navigate to DynamoDB and click on create tables.

The screenshot shows the AWS Services search results for 'dyn'. The top result is 'DynamoDB' under the 'Services' category. Other services listed include Amazon DocumentDB, CloudFront, and Athena. Below the services, there are sections for 'Features' (Settings, Clusters) and 'Clusters' (DynamoDB feature).

The screenshot shows the DynamoDB Dashboard. On the left, there's a sidebar with links like Dashboard, Tables, Alarms, PartiQL editor, Backups, Imports from S3, Imports from FS, Integrations, Reserved capacity, and Settings. The main area is titled 'Dashboard' and contains sections for 'Alarms' (0), 'DAX clusters' (0), and 'What's new'. On the right, there's a 'Create resources' section with a prominent 'Create table' button. A note about DAX clusters is also present.



- **Activity 2.2: Configure Attributes for User Data and Book Request.**

- Create photographers table with partition key "photographer_id" with type String and click on create tables.

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

1 to 255 characters and case sensitive.

Table class	DynamoDB Standard	Yes
Capacity mode	Provisioned	Yes
Provisioned read capacity	5 RCU	Yes
Provisioned write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Deletion protection	Off	Yes
Resource-based policy	Not active	Yes

Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

[Add new tag](#)

You can add 50 more tags.

[Cancel](#)

[Create table](#)

The screenshot shows the AWS DynamoDB Tables page. At the top, there's a green success message: "The photographers table was created successfully." Below this, the "Tables (1)" section is displayed. A single table named "photographers" is listed, showing its status as "Active", partition key as "photographer_id (\$)", and other details like replication regions and deletion protection. The left sidebar includes links for Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. The bottom navigation bar includes CloudShell, Feedback, and links to AWS terms and conditions.

- Create booking table with partition key "booking_id" with type String and click on create tables.

DynamoDB > Tables > Create table

Create table

Table details [Edit](#)

DynamoDB is a fast, fault-tolerant database that requires only a table name and a primary key when you create the table.

Table name [Create](#) [Import](#) [Export](#) [Copy table](#)

Booking
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.)

Partition key [Edit](#)
The partition key is part of the standard primary key in a table that identifies the row or item. You can use this key to search across your entire table with a single query or scan operation.

booking_id [Edit](#)
String
1–255 characters, must start with a letter, and can contain numbers, underscores (_), hyphens (-), and periods (.)

Sort key - optional [Edit](#)
You can add a sort key to the second part of a table's primary key. This will allow you to sort or search by more than one dimension.
String

Table settings

Default settings	Customize settings	
The fastest way to create your table. DynamoDB automatically provisions read and write capacity after your table has been created. To modify these settings later, see Table settings .	Low latency, increased read/write capacity, provisioned write capacity for your needs.	
Table class	DynamoDB Standard	Yes
Capacity mode	Provisioned	Yes
Provisioned read capacity	5 RCU	Yes
Provisioned write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Deletion protection	Off	Yes
Resource-based policy	Not active	Yes

Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

[Add new tag](#)

You can add 50 more tags.

[Cancel](#) [Create table](#)

Screenshot of the AWS DynamoDB console showing the 'photographers' table settings.

Left Sidebar:

- DynamoDB
- Dashboard
- Tables
- Explore items
- PartiQL editor
- Backups
- Exports to S3
- Imports from S3
- Integrations New
- Reserved capacity
- Settings

Top Bar:

- Search
- [Alt+Shift+S]
- Notifications
- Help
- Asia Pacific (Mumbai)

Table Settings View:

Tables (1)

- Any tag key
- Any tag value
- Find tables

photographers (Selected)

Actions: C A B D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C D E F G H I J K L M N O <span

- Create items with data and click on create items

DynamoDB > Explore Items: photographers > Create item

Attributes

Attribute name	Value	Type	Action
Name	Alex John	String	<button>Remove</button>
Email ID	alexjohn@gmail.com	String	<button>Remove</button>
Skills	Portrait, Wedding, Birthdays	String	<button>Remove</button>
Photo	https://images.pexels.com/photos/1264210/pexels-photo-1264210.jpeg?cs=srgb&dl=pexels-andre-furtado-43594-1264210.jpg&fm=jpg	String	<button>Remove</button>
Available Dates	Empty value	String	<button>Remove</button>

Add new attribute Cancel Create item

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

Milestone 3: IAM Role Setup

Activity 3.1: Create IAM Role

- go to search bar and search the iam and click on the iam below given figure

Dynamo

Services

- IAM Manage access to AWS resources
- IAM Identity Center Manage workforce user access to multiple AWS accounts and cloud applications
- Resource Access Manager Share AWS resources with other accounts or AWS Organizations

Features

- Groups
- AWS Lambda Functions

Were these results helpful?

DynamoD

- Dashboard
- Tables
- Explore items
- PartiQL editor
- Backups
- Exports to S3
- Imports from
- Integrations
- Reserved capacity
- Settings

Show more

Region

Activity 3.2: Attach Policies

- Click on the iam and select the roles and create the roles with step by step with given figure

The screenshot shows the AWS IAM Roles page. The left sidebar includes sections for Identity and Access Management (IAM), Access management, and Access reports. The main content area displays a table titled 'Roles (3)'. The table has columns for Role name, Trusted entities, and Last activity. Three roles are listed:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support@service-Linker	
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisorService	
capturements_ec2	AWS Service: ec2	

Below the table, there are sections for 'Roles Anywhere' (with options for 'Access AWS from your non AWS workloads' and 'X.509 Standard'), and 'Temporary credentials' (with a note about using temporary credentials for enhanced security).

The screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. The left sidebar shows steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area is titled 'Select trusted entity' and contains the following sections:

- Trusted entity type:** A group of radio buttons:
 - AWS service: Allows AWS services like EC2, Lambda, or others to perform actions in this account.
 - AWS account: Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
 - Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 Federation:** Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy:** Creates a custom trust policy to enable others to perform actions in this account.
- Use case:** Allows an AWS service like EC2, Lambda, or others to perform actions in this account.
- Service or use case:** A dropdown menu with the placeholder 'Choose a service or use case'.

At the bottom right are 'Cancel' and 'Next' buttons.

- In this step we select the `amazondynamodbfullaccess` and go to next steps

Khit_3rd_year_aws_Mentoring_sessions

SWS Q Search [Alt+S] IAM > Roles > Create role

Step 1 Select trusted entity Step 2 Add permissions Step 3 Name, review, and create

Add permissions Info

Permissions policies (1/1057) View
Choose one or more policies to attach to your new role.

Filter by Type All types 6 matches

Policy name	Type	Description
<input checked="" type="checkbox"/>  AmazonDynamoDBFullAccess	AWS managed	Provides full access to Amazon Dynam...
<input type="checkbox"/>  AmazonDynamoDBFullAccess_v2	AWS managed	Provides full access to Amazon Dynam...
<input type="checkbox"/>  AmazonDynamoDBFullAccessWithDataPipeline	AWS managed	This policy is on a deprecation path. S...
<input type="checkbox"/>  AmazonDynamoDBReadOnlyAccess	AWS managed	Provides read only access to Amazon D...
<input type="checkbox"/>  AWSLambdaDynamoDBExecutionRole	AWS managed	Provides list and read access to Dynam...
<input type="checkbox"/>  AWSLambdaDynamoDBInvocationRole	AWS managed	Provides read access to DynamoDB Str...

▶ Set permissions boundary - optional

Q Search [Alt+S] IAM > Roles > Create role

Name, review, and create

Enter a meaningful name to identify this role:
capturescreens_ec2
Maximum 64 characters. Use alphanumeric and '-' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: -, +, @, /, \, !, %, ^, &, =.

Step 1: Select trusted entities Edit

Trust policy

```

1* [
2*   "Version": "2012-10-17",
3*   "Statement": [
4*     {
5*       "Effect": "Allow",
6*       "Action": [
7*         "sts:AssumeRole"
8*       ],
9*       "Principal": [
10*         "ec2.amazonaws.com"
11*       ]
12*     }
13*   ]
14* ]
15* ]

```

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

- successfully created iam role

The screenshot shows the AWS IAM Roles page. A green success message at the top left says "Role capturescreenshots_ec2 created." Below it, the "Roles [3]" section lists three roles: "AWSLambdaRoleForSupport", "AWSServiceRoleForTrustAdvisor", and "capturescreenshots_ec2". The "capturescreenshots_ec2" role is associated with "AWS Service: ec2". On the right side, there are sections for "Roles Anywhere", "Access AWS from your non AWS workloads", "X.509 Standard", and "Temporary credentials". The bottom right corner includes standard AWS footer links: "© 2025, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

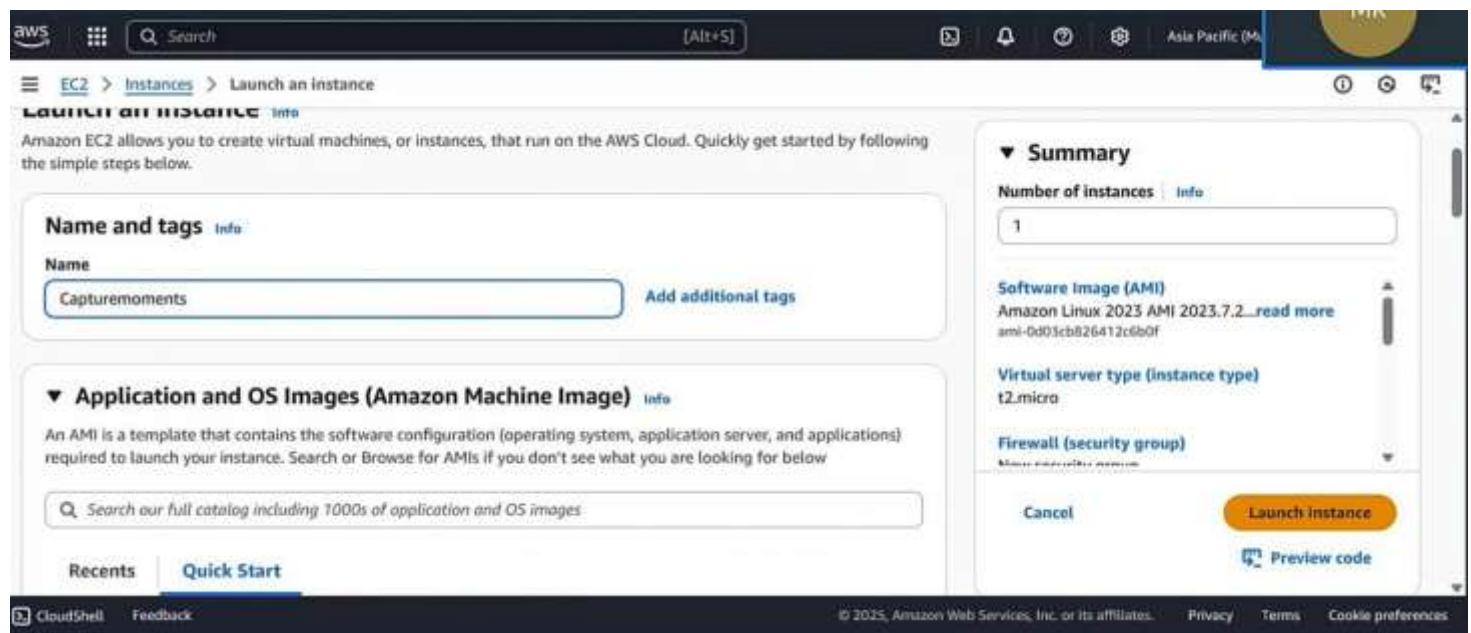
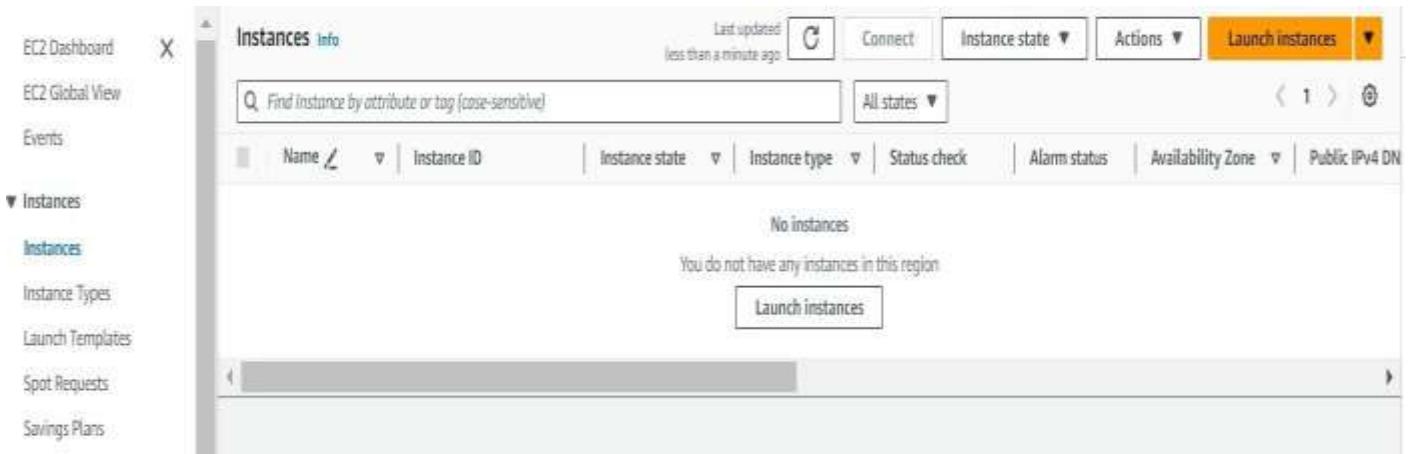
Milestone 4: EC2 Instance Setup

Activity 4.1: Launch **6** an EC2 instance to host the Flask application.

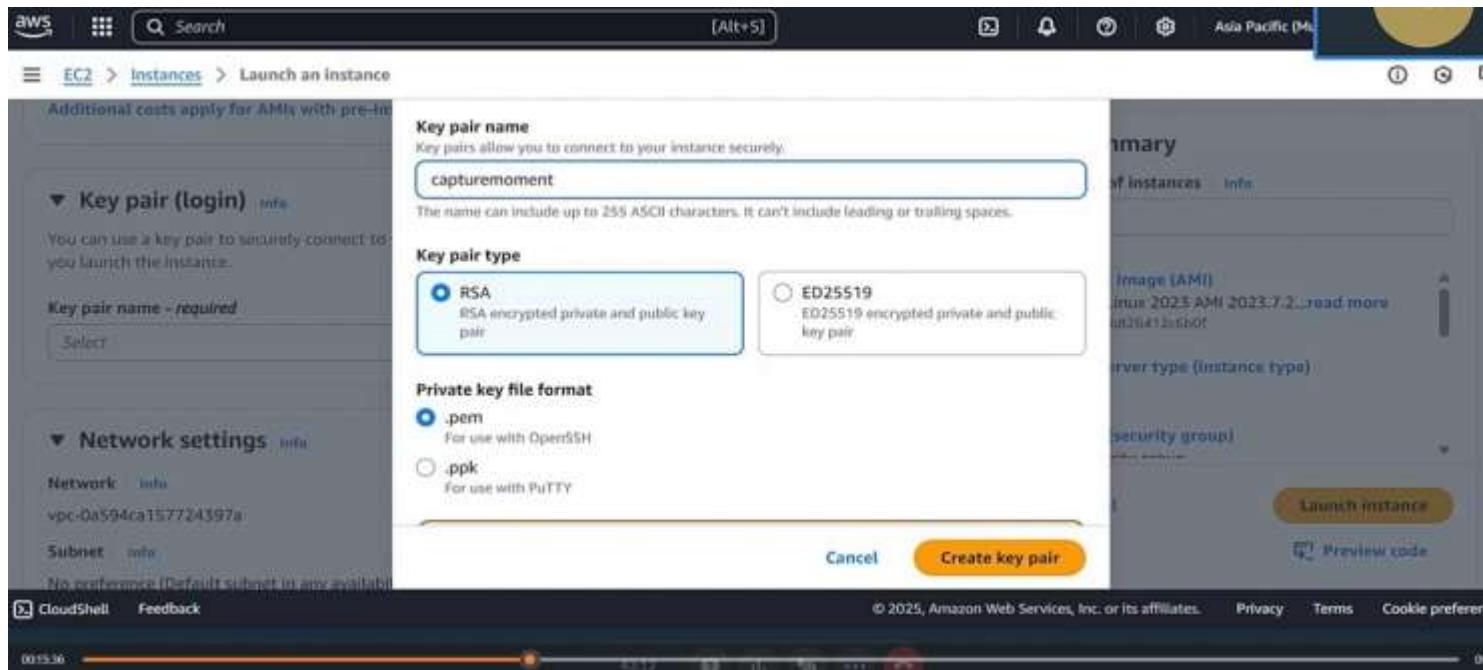
- go to search bar and search the iam and click on the iam below given figure

The screenshot shows the AWS Services search results for "ec2". The search bar at the top has "ec2" typed into it. The main content area is titled "Search results for 'ec2'" and contains a "Services" section. It lists several services: "EC2" (Virtual Servers in the Cloud), "EC2 Image Builder" (A managed service to automate build, customize and deploy OS images), and "Recycle Bin" (Protect resources from accidental deletion). To the left of the main content is a sidebar with "Services" and "Features" sections, including "Resources New", "Documentation", "Knowledge articles", "Marketplace", "Blog posts", "Events", and "Tutorials". The "Resources New" link is highlighted in blue.

- Launch the instance and create the instance.

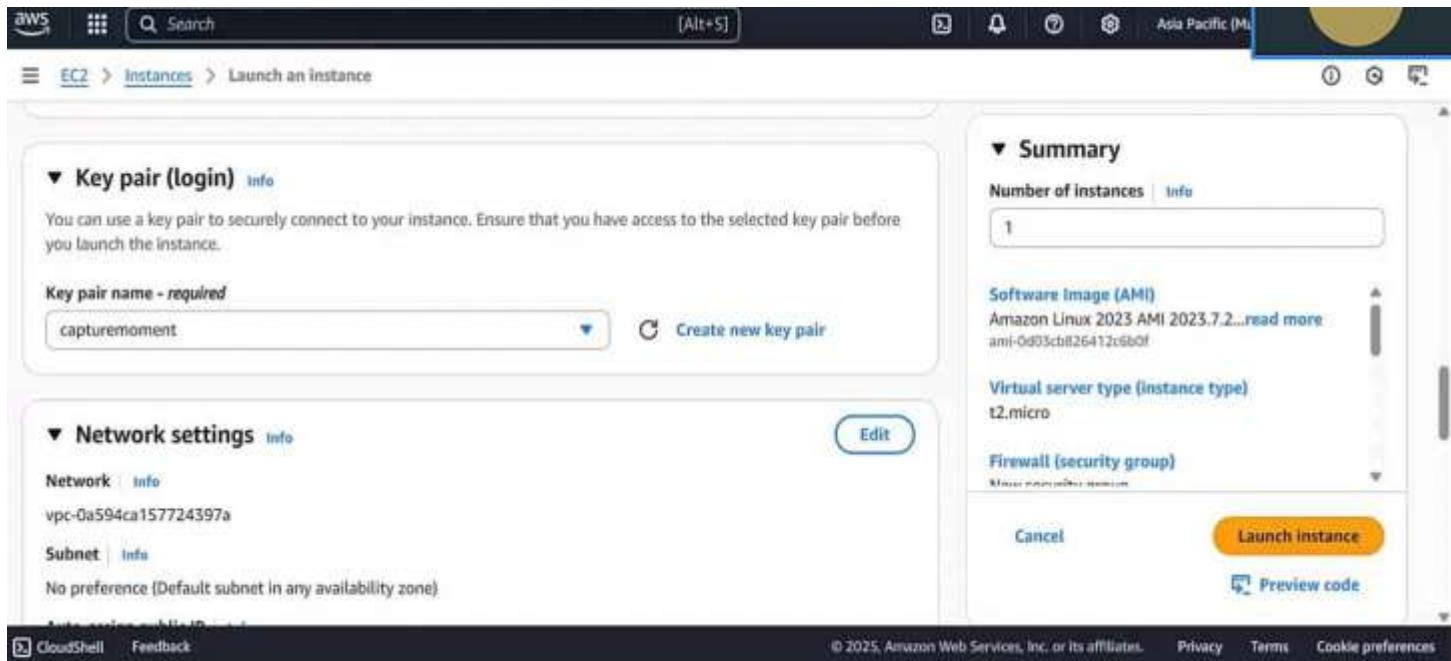


- Create the key pair name and download the key pair.



Activity 4.2: Configure security groups for HTTP, and SSH access.

- Go to network settings and click on the edit and to down and click the network security rule.



The screenshot shows the AWS Next Step dashboard with a green header bar. Below it, there's a search bar and a 'Next Step' section. The main area contains several service cards:

- Create billing and free tier usage alerts**: Manage costs and avoid surprises with AWS CloudWatch Metrics and CloudWatch Metrics Insights.
- Connect to your instance**: Use your browser to connect, log in to EC2 instances from your local computer.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to easily query data between them.
- Create EBS snapshot policy**: Create a policy that automatically creates, retains, and deletes EBS snapshots.
- Manage detailed monitoring**: Break down detailed monitoring for EC2 instances. You can enable detailed monitoring, the Amazon EC2 CloudWatch Metrics Monitoring graphs with a 1-minute period.
- Create Load Balancer**: Create a load balancer, automatically scaling up or down based on traffic.
- Create AWS budget**: AWS Budgets allows you to create budgets. Action (Spending) and Like actions are pre-defined and change from a single location.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance.
- Disable recovery for your instances**: Disable the instances you just identified in a different availability zone or a different Region using AWS Lambda (Serverless Function).
- Monitor for suspicious runtime activities**: Amazon CloudWatch enables you to continuously monitor for malicious network activity, bad snapshots and behaviors, and real-time visibility into air-time activities occurring across your Amazon EC2 workloads.
- Get instance screenshots**: Capture a screenshot from the instance and save it as an image. This is useful for providing evidence of a problematic instance.
- User streaming**: Use the instance's streaming capability to upload content.

At the bottom right, there's a yellow button labeled 'View all services'.

Description:

To connect to EC2 using EC2 Instance Connect, start by ensuring that an IAM role is attached to your EC2 instance. You can do this by selecting your instance, clicking on Actions, then navigating to Security and selecting Modify IAM Role to attach the appropriate role. After the IAM role is connected, navigate to the EC2 section in the AWS Management Console. Select the EC2 instance you wish to connect to. At the top of the EC2 Dashboard, click the Connect button. From the connection methods presented, choose EC2 Instance Connect. Finally, click Connect again, and a new browser-based terminal will open, allowing you to access your EC2 instance directly from your browser.

The screenshot shows the AWS Instances dashboard. At the top, there are filters for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', 'Alarm status', 'Availability Zone', 'Public IPv4 DNS', 'Public IPv4', 'Static IP', 'IP6s', 'Monitoring', and 'Security'. The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4	Static IP	IP6s	Monitoring	Security
InstantLibrary... > i-001861022fbcac290	i-001861022fbcac290	Stopped	t2.micro	-	-	-	-	-	-	-	Detailed	Launched

Below the table, there's a detailed view for the instance i-001861022fbcac290:

Instance summary for i-001861022fbcac290 (InstantLibraryApp)

Details for this instance are as follows:

- Instance ID:** i-001861022fbcac290
- IPv4 address:** -
- Hardware type:** t2.micro
- IP name:** ip-172-31-5-5.ap-south-1.compute.internal
- Amazon private resource DNS name:** ip-172-31-5-5.ap-south-1.compute.internal
- IPv6 ID:** -
- Auto-assigned IP address:** -
- AMI Role:** arn:aws:iam::123456789012:role/lambdaBasicExecutionRole
- Root FS:** Required
- Public IPv4 address:** -
- Instance state:** Stopped
- Public IPv4 DNS:** ip-172-31-5-5.ap-south-1.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-001861022fbcac290
- Subnet ID:** subnet-001861022fbcac290
- Interface ARN:** arn:aws:ec2:ap-south-1:524064601318:instance/i-001861022fbcac290

At the bottom, there are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

i-001861022fbcac290 (InstantLibraryApp)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

sns_Dynamodb_role



Create new IAM role

Cancel

Update IAM role

Connect to instance Info

Connect to your instance i-001861022fbcac290 (InstantLibraryApp) using any of these options

[EC2 Instance Connect](#)

[Session Manager](#)

[SSH client](#)

[EC2 serial console](#)



Port 22 (SSH) is open to all IPv4 addresses

Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 13.233.177.0/29. [Learn more](#).

Instance ID

i-001861022fbcac290 (InstantLibraryApp)

Connection Type

[Connect using EC2 Instance Connect](#)

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 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 IPv4 address](#)

13.200.229.59

[IPv6 address](#)

Username

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

ec2-user



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

Cancel

Connect

Milestone 5: Deployment on EC2

Activity 5.1: install dependencies



```
A newer release of "Amazon Linux" is available.
Version 2023.6.20241010!
Run "/usr/bin/dnf check-release-update" for full release and version update info
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Tue Oct 16 04:17:59 2024 from 13.233.177.3
[ec2-user@ip-172-31-3-2 ~]$
```

This will download your project to the EC2 instance. Now install dependencies, so type following commands.

- sudo yum update -y
- sudo yum install python3 git -y
- pip3 install --user flask boto3
- git clone https://github.com/<your-username>/<repo-name>.git
- cd <repository-name>



```
ec2-user@ip-172-31-3-2:~$ sudo yum update -y
[sudo] password for ec2-user: 
Loaded plugins: priorities, update-motd, upgrade-helper
Transaction Test succeeded.
Running Transaction Test
Transaction Test succeeded.

Transaction Summary
  Installed: libcrypt-compat-4.4.33-7.amzn2023.x86_64
  Installing : python3-pip-21.3.1-2.amzn2023.0.11.noarch
  Running scriptlets: python3-pip-21.3.1-2.amzn2023.0.11.noarch
  Verifying  : libcrypt-compat-4.4.33-7.amzn2023.x86_64
  Verifying  : python3-pip-21.3.1-2.amzn2023.0.11.noarch

Installed:
  libcrypt-compat-4.4.33-7.amzn2023.x86_64
```



```
ec2-user@ip-172-31-3-2:~$ python3 -m pip install --user flask boto3
Collecting Flask
  Downloading flask-3.1.1-py3-none-any.whl (103 kB)
    100% |██████████| 103 kB 38.0 kB/s
Collecting Boto3
  Downloading boto3-1.29.1-py3-none-any.whl (329 kB)
    100% |██████████| 329 kB 28.7 kB/s
Collecting werkzeug==3.1.0
  Downloading werkzeug-3.1.0-py3-none-any.whl (234 kB)
    100% |██████████| 234 kB 48.8 kB/s
Collecting uritemplate==3.0.0
  Downloading uritemplate-3.0.0-py3-none-any.whl (8.5 kB)
Collecting importlib-metadata==3.6.0
  Downloading importlib_metadata-3.6.0-py3-none-any.whl (27 kB)
Collecting itsdangerous==2.2.0
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting Jinja2==3.1.2
  Downloading Jinja2-3.1.2-py3-none-any.whl (124 kB)
    100% |██████████| 124 kB 47.0 kB/s
Collecting click==8.1.3
  Downloading click-8.1.3-py3-none-any.whl (56 kB)
    100% |██████████| 56 kB 38.1 kB/s
Collecting werkzeug-sessions[redis]==2.1.1
  Downloading WerkzeugSession-redis-2.1.1-py3-manylinux_2_17_x86_64.manylinux2018_x86_64.whl (20 kB)
Collecting astroid==3.1.10
  Downloading astroid-3.1.10-py3-none-any.whl (85 kB)
Collecting cattrs==0.17.1-py3-none-any.whl (85 kB)
  Downloading cattrs-0.17.1-py3-none-any.whl (85 kB)
Collecting attrs==2.40.0-py3.10.0
  Downloading attrs-2.40.0-py3.10.0.whl (13 kB)
Collecting botocore==1.39.0
  Downloading botocore-1.39.0-py3-none-any.whl (13.8 kB)
  100% |██████████| 13.8 kB 38.1 kB/s
Requirement already satisfied: jmespath<=1.0.1 in /usr/lib/python3.9/site-packages (from botocore) (1.0.1)
Requirement already satisfied: python-dateutil<3.0.0,>=2.3.1 in /usr/lib/python3.9/site-packages (from botocore) (2.3.1)
Requirement already satisfied: urllib3<1.27,>=1.25.1 in /usr/lib/python3.9/site-packages (from botocore<1.40.0,>=1.39.1>botocore) (1.25.1)
Collecting rfc3986==2.0.0
  Downloading rfc3986-2.0.0-py3-none-any.whl (10 kB)
Requirement already satisfied: sibyl<1.5 in /usr/lib/python3.9/site-packages (from botocore<1.40.0,>=1.39.1>botocore) (1.15.0)
Requirement already satisfied: werkzeug<3.1.1,!=3.1.0,!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,!=3.0.4,!=3.0.5,!=3.0.6,!=3.0.7,!=3.0.8,!=3.0.9,!=3.0.10,!=3.0.11,!=3.0.12,!=3.0.13,!=3.0.14,!=3.0.15,!=3.0.16,!=3.0.17,!=3.0.18,!=3.0.19,!=3.0.20,!=3.0.21,!=3.0.22,!=3.0.23,!=3.0.24,!=3.0.25,!=3.0.26,!=3.0.27,!=3.0.28,!=3.0.29,!=3.0.30,!=3.0.31,!=3.0.32,!=3.0.33,!=3.0.34,!=3.0.35,!=3.0.36,!=3.0.37,!=3.0.38,!=3.0.39,!=3.0.40,!=3.0.41,!=3.0.42,!=3.0.43,!=3.0.44,!=3.0.45,!=3.0.46,!=3.0.47,!=3.0.48,!=3.0.49,!=3.0.50,!=3.0.51,!=3.0.52,!=3.0.53,!=3.0.54,!=3.0.55,!=3.0.56,!=3.0.57,!=3.0.58,!=3.0.59,!=3.0.60,!=3.0.61,!=3.0.62,!=3.0.63,!=3.0.64,!=3.0.65,!=3.0.66,!=3.0.67,!=3.0.68,!=3.0.69,!=3.0.70,!=3.0.71,!=3.0.72,!=3.0.73,!=3.0.74,!=3.0.75,!=3.0.76,!=3.0.77,!=3.0.78,!=3.0.79,!=3.0.80,!=3.0.81,!=3.0.82,!=3.0.83,!=3.0.84,!=3.0.85,!=3.0.86,!=3.0.87,!=3.0.88,!=3.0.89,!=3.0.90,!=3.0.91,!=3.0.92,!=3.0.93,!=3.0.94,!=3.0.95,!=3.0.96,!=3.0.97,!=3.0.98,!=3.0.99,!=3.0.100,!=3.0.101,!=3.0.102,!=3.0.103,!=3.0.104,!=3.0.105,!=3.0.106,!=3.0.107,!=3.0.108,!=3.0.109,!=3.0.110,!=3.0.111,!=3.0.112,!=3.0.113,!=3.0.114,!=3.0.115,!=3.0.116,!=3.0.117,!=3.0.118,!=3.0.119,!=3.0.120,!=3.0.121,!=3.0.122,!=3.0.123,!=3.0.124,!=3.0.125,!=3.0.126,!=3.0.127,!=3.0.128,!=3.0.129,!=3.0.130,!=3.0.131,!=3.0.132,!=3.0.133,!=3.0.134,!=3.0.135,!=3.0.136,!=3.0.137,!=3.0.138,!=3.0.139,!=3.0.140,!=3.0.141,!=3.0.142,!=3.0.143,!=3.0.144,!=3.0.145,!=3.0.146,!=3.0.147,!=3.0.148,!=3.0.149,!=3.0.150,!=3.0.151,!=3.0.152,!=3.0.153,!=3.0.154,!=3.0.155,!=3.0.156,!=3.0.157,!=3.0.158,!=3.0.159,!=3.0.160,!=3.0.161,!=3.0.162,!=3.0.163,!=3.0.164,!=3.0.165,!=3.0.166,!=3.0.167,!=3.0.168,!=3.0.169,!=3.0.170,!=3.0.171,!=3.0.172,!=3.0.173,!=3.0.174,!=3.0.175,!=3.0.176,!=3.0.177,!=3.0.178,!=3.0.179,!=3.0.180,!=3.0.181,!=3.0.182,!=3.0.183,!=3.0.184,!=3.0.185,!=3.0.186,!=3.0.187,!=3.0.188,!=3.0.189,!=3.0.190,!=3.0.191,!=3.0.192,!=3.0.193,!=3.0.194,!=3.0.195,!=3.0.196,!=3.0.197,!=3.0.198,!=3.0.199,!=3.0.200,!=3.0.201,!=3.0.202,!=3.0.203,!=3.0.204,!=3.0.205,!=3.0.206,!=3.0.207,!=3.0.208,!=3.0.209,!=3.0.210,!=3.0.211,!=3.0.212,!=3.0.213,!=3.0.214,!=3.0.215,!=3.0.216,!=3.0.217,!=3.0.218,!=3.0.219,!=3.0.220,!=3.0.221,!=3.0.222,!=3.0.223,!=3.0.224,!=3.0.225,!=3.0.226,!=3.0.227,!=3.0.228,!=3.0.229,!=3.0.230,!=3.0.231,!=3.0.232,!=3.0.233,!=3.0.234,!=3.0.235,!=3.0.236,!=3.0.237,!=3.0.238,!=3.0.239,!=3.0.240,!=3.0.241,!=3.0.242,!=3.0.243,!=3.0.244,!=3.0.245,!=3.0.246,!=3.0.247,!=3.0.248,!=3.0.249,!=3.0.250,!=3.0.251,!=3.0.252,!=3.0.253,!=3.0.254,!=3.0.255,!=3.0.256,!=3.0.257,!=3.0.258,!=3.0.259,!=3.0.260,!=3.0.261,!=3.0.262,!=3.0.263,!=3.0.264,!=3.0.265,!=3.0.266,!=3.0.267,!=3.0.268,!=3.0.269,!=3.0.270,!=3.0.271,!=3.0.272,!=3.0.273,!=3.0.274,!=3.0.275,!=3.0.276,!=3.0.277,!=3.0.278,!=3.0.279,!=3.0.280,!=3.0.281,!=3.0.282,!=3.0.283,!=3.0.284,!=3.0.285,!=3.0.286,!=3.0.287,!=3.0.288,!=3.0.289,!=3.0.290,!=3.0.291,!=3.0.292,!=3.0.293,!=3.0.294,!=3.0.295,!=3.0.296,!=3.0.297,!=3.0.298,!=3.0.299,!=3.0.300,!=3.0.301,!=3.0.302,!=3.0.303,!=3.0.304,!=3.0.305,!=3.0.306,!=3.0.307,!=3.0.308,!=3.0.309,!=3.0.310,!=3.0.311,!=3.0.312,!=3.0.313,!=3.0.314,!=3.0.315,!=3.0.316,!=3.0.317,!=3.0.318,!=3.0.319,!=3.0.320,!=3.0.321,!=3.0.322,!=3.0.323,!=3.0.324,!=3.0.325,!=3.0.326,!=3.0.327,!=3.0.328,!=3.0.329,!=3.0.330,!=3.0.331,!=3.0.332,!=3.0.333,!=3.0.334,!=3.0.335,!=3.0.336,!=3.0.337,!=3.0.338,!=3.0.339,!=3.0.340,!=3.0.341,!=3.0.342,!=3.0.343,!=3.0.344,!=3.0.345,!=3.0.346,!=3.0.347,!=3.0.348,!=3.0.349,!=3.0.350,!=3.0.351,!=3.0.352,!=3.0.353,!=3.0.354,!=3.0.355,!=3.0.356,!=3.0.357,!=3.0.358,!=3.0.359,!=3.0.360,!=3.0.361,!=3.0.362,!=3.0.363,!=3.0.364,!=3.0.365,!=3.0.366,!=3.0.367,!=3.0.368,!=3.0.369,!=3.0.370,!=3.0.371,!=3.0.372,!=3.0.373,!=3.0.374,!=3.0.375,!=3.0.376,!=3.0.377,!=3.0.378,!=3.0.379,!=3.0.380,!=3.0.381,!=3.0.382,!=3.0.383,!=3.0.384,!=3.0.385,!=3.0.386,!=3.0.387,!=3.0.388,!=3.0.389,!=3.0.390,!=3.0.391,!=3.0.392,!=3.0.393,!=3.0.394,!=3.0.395,!=3.0.396,!=3.0.397,!=3.0.398,!=3.0.399,!=3.0.400,!=3.0.401,!=3.0.402,!=3.0.403,!=3.0.404,!=3.0.405,!=3.0.406,!=3.0.407,!=3.0.408,!=3.0.409,!=3.0.410,!=3.0.411,!=3.0.412,!=3.0.413,!=3.0.414,!=3.0.415,!=3.0.416,!=3.0.417,!=3.0.418,!=3.0.419,!=3.0.420,!=3.0.421,!=3.0.422,!=3.0.423,!=3.0.424,!=3.0.425,!=3.0.426,!=3.0.427,!=3.0.428,!=3.0.429,!=3.0.430,!=3.0.431,!=3.0.432,!=3.0.433,!=3.0.434,!=3.0.435,!=3.0.436,!=3.0.437,!=3.0.438,!=3.0.439,!=3.0.440,!=3.0.441,!=3.0.442,!=3.0.443,!=3.0.444,!=3.0.445,!=3.0.446,!=3.0.447,!=3.0.448,!=3.0.449,!=3.0.450,!=3.0.451,!=3.0.452,!=3.0.453,!=3.0.454,!=3.0.455,!=3.0.456,!=3.0.457,!=3.0.458,!=3.0.459,!=3.0.460,!=3.0.461,!=3.0.462,!=3.0.463,!=3.0.464,!=3.0.465,!=3.0.466,!=3.0.467,!=3.0.468,!=3.0.469,!=3.0.470,!=3.0.471,!=3.0.472,!=3.0.473,!=3.0.474,!=3.0.475,!=3.0.476,!=3.0.477,!=3.0.478,!=3.0.479,!=3.0.480,!=3.0.481,!=3.0.482,!=3.0.483,!=3.0.484,!=3.0.485,!=3.0.486,!=3.0.487,!=3.0.488,!=3.0.489,!=3.0.490,!=3.0.491,!=3.0.492,!=3.0.493,!=3.0.494,!=3.0.495,!=3.0.496,!=3.0.497,!=3.0.498,!=3.0.499,!=3.0.500,!=3.0.501,!=3.0.502,!=3.0.503,!=3.0.504,!=3.0.505,!=3.0.506,!=3.0.507,!=3.0.508,!=3.0.509,!=3.0.510,!=3.0.511,!=3.0.512,!=3.0.513,!=3.0.514,!=3.0.515,!=3.0.516,!=3.0.517,!=3.0.518,!=3.0.519,!=3.0.520,!=3.0.521,!=3.0.522,!=3.0.523,!=3.0.524,!=3.0.525,!=3.0.526,!=3.0.527,!=3.0.528,!=3.0.529,!=3.0.530,!=3.0.531,!=3.0.532,!=3.0.533,!=3.0.534,!=3.0.535,!=3.0.536,!=3.0.537,!=3.0.538,!=3.0.539,!=3.0.540,!=3.0.541,!=3.0.542,!=3.0.543,!=3.0.544,!=3.0.545,!=3.0.546,!=3.0.547,!=3.0.548,!=3.0.549,!=3.0.550,!=3.0.551,!=3.0.552,!=3.0.553,!=3.0.554,!=3.0.555,!=3.0.556,!=3.0.557,!=3.0.558,!=3.0.559,!=3.0.560,!=3.0.561,!=3.0.562,!=3.0.563,!=3.0.564,!=3.0.565,!=3.0.566,!=3.0.567,!=3.0.568,!=3.0.569,!=3.0.570,!=3.0.571,!=3.0.572,!=3.0.573,!=3.0.574,!=3.0.575,!=3.0.576,!=3.0.577,!=3.0.578,!=3.0.579,!=3.0.580,!=3.0.581,!=3.0.582,!=3.0.583,!=3.0.584,!=3.0.585,!=3.0.586,!=3.0.587,!=3.0.588,!=3.0.589,!=3.0.590,!=3.0.591,!=3.0.592,!=3.0.593,!=3.0.594,!=3.0.595,!=3.0.596,!=3.0.597,!=3.0.598,!=3.0.599,!=3.0.600,!=3.0.601,!=3.0.602,!=3.0.603,!=3.0.604,!=3.0.605,!=3.0.606,!=3.0.607,!=3.0.608,!=3.0.609,!=3.0.610,!=3.0.611,!=3.0.612,!=3.0.613,!=3.0.614,!=3.0.615,!=3.0.616,!=3.0.617,!=3.0.618,!=3.0.619,!=3.0.620,!=3.0.621,!=3.0.622,!=3.0.623,!=3.0.624,!=3.0.625,!=3.0.626,!=3.0.627,!=3.0.628,!=3.0.629,!=3.0.630,!=3.0.631,!=3.0.632,!=3.0.633,!=3.0.634,!=3.0.635,!=3.0.636,!=3.0.637,!=3.0.638,!=3.0.639,!=3.0.640,!=3.0.641,!=3.0.642,!=3.0.643,!=3.0.644,!=3.0.645,!=3.0.646,!=3.0.647,!=3.0.648,!=3.0.649,!=3.0.650,!=3.0.651,!=3.0.652,!=3.0.653,!=3.0.654,!=3.0.655,!=3.0.656,!=3.0.657,!=3.0.658,!=3.0.659,!=3.0.660,!=3.0.661,!=3.0.662,!=3.0.663,!=3.0.664,!=3.0.665,!=3.0.666,!=3.0.667,!=3.0.668,!=3.0.669,!=3.0.670,!=3.0.671,!=3.0.672,!=3.0.673,!=3.0.674,!=3.0.675,!=3.0.676,!=3.0.677,!=3.0.678,!=3.0.679,!=3.0.680,!=3.0.681,!=3.0.682,!=3.0.683,!=3.0.684,!=3.0.685,!=3.0.686,!=3.0.687,!=3.0.688,!=3.0.689,!=3.0.690,!=3.0.691,!=3.0.692,!=3.0.693,!=3.0.694,!=3.0.695,!=3.0.696,!=3.0.697,!=3.0.698,!=3.0.699,!=3.0.700,!=3.0.701,!=3.0.702,!=3.0.703,!=3.0.704,!=3.0.705,!=3.0.706,!=3.0.707,!=3.0.708,!=3.0.709,!=3.0.710,!=3.0.711,!=3.0.712,!=3.0.713,!=3.0.714,!=3.0.715,!=3.0.716,!=3.0.717,!=3.0.718,!=3.0.719,!=3.0.720,!=3.0.721,!=3.0.722,!=3.0.723,!=3.0.724,!=3.0.725,!=3.0.726,!=3.0.727,!=3.0.728,!=3.0.729,!=3.0.730,!=3.0.731,!=3.0.732,!=3.0.733,!=3.0.734,!=3.0.735,!=3.0.736,!=3.0.737,!=3.0.738,!=3.0.739,!=3.0.740,!=3.0.741,!=3.0.742,!=3.0.743,!=3.0.744,!=3.0.745,!=3.0.746,!=3.0.747,!=3.0.748,!=3.0.749,!=3.0.750,!=3.0.751,!=3.0.752,!=3.0.753,!=3.0.754,!=3.0.755,!=3.0.756,!=3.0.757,!=3.0.758,!=3.0.759,!=3.0.760,!=3.0.761,!=3.0.762,!=3.0.763,!=3.0.764,!=3.0.765,!=3.0.766,!=3.0.767,!=3.0.768,!=3.0.769,!=3.0.770,!=3.0.771,!=3.0.772,!=3.0.773,!=3.0.774,!=3.0.775,!=3.0.776,!=3.0.777,!=3.0.778,!=3.0.779,!=3.0.780,!=3.0.781,!=3.0.782,!=3.0.783,!=3.0.784,!=3.0.785,!=3.0.786,!=3.0.787,!=3.0.788,!=3.0.789,!=3.0.790,!=3.0.791,!=3.0.792,!=3.0.793,!=3.0.794,!=3.0.795,!=3.0.796,!=3.0.797,!=3.0.798,!=3.0.799,!=3.0.800,!=3.0.801,!=3.0.802,!=3.0.803,!=3.0.804,!=3.0.805,!=3.0.806,!=3.0.807,!=3.0.808,!=3.0.809,!=3.0.810,!=3.0.811,!=3.0.812,!=3.0.813,!=3.0.814,!=3.0.815,!=3.0.816,!=3.0.817,!=3.0.818,!=3.0.819,!=3.0.820,!=3.0.821,!=3.0.822,!=3.0.823,!=3.0.824,!=3.0.825,!=3.0.826,!=3.0.827,!=3.0.828,!=3.0.829,!=3.0.830,!=3.0.831,!=3.0.832,!=3.0.833,!=3.0.834,!=3.0.835,!=3.0.836,!=3.0.837,!=3.0.838,!=3.0.839,!=3.0.840,!=3.0.841,!=3.0.842,!=3.0.843,!=3.0.844,!=3.0.845,!=3.0.846,!=3.0.847,!=3.0.848,!=3.0.849,!=3.0.850,!=3.0.851,!=3.0.852,!=3.0.853,!=3.0.854,!=3.0.855,!=3.0.856,!=3.0.857,!=3.0.858,!=3.0.859,!=3.0.860,!=3.0.861,!=3.0.862,!=3.0.863,!=3.0.864,!=3.0.865,!=3.0.866,!=3.0.867,!=3.0.868,!=3.0.869,!=3.0.870,!=3.0.871,!=3.0.872,!=3.0.873,!=3.0.874,!=3.0.875,!=3.0.876,!=3.0.877,!=3.0.878,!=3.0.879,!=3.0.880,!=3.0.881,!=3.0.882,!=3.0.883,!=3.0.884,!=3.0.885,!=3.0.886,!=3.0.887,!=3.0.888,!=3.0.889,!=3.0.890,!=3.0.891,!=3.0.892,!=3.0.893,!=3.0.894,!=3.0.895,!=3.0.896,!=3.0.897,!=3.0.898,!=3.0.899,!=3.0.900,!=3.0.901,!=3.0.902,!=3.0.903,!=3.0.904,!=3.0.905,!=3.0.906,!=3.0.907,!=3.0.908,!=3.0.909,!=3.0.910,!=3.0.911,!=3.0.912,!=3.0.913,!=3.0.914,!=3.0.915,!=3.0.916,!=3.0.917,!=3.0.918,!=3.0.919,!=3.0.920,!=3.0.921,!=3.0.922,!=3.0.923,!=3.0.924,!=3.0.925,!=3.0.926,!=3.0.927,!=3.0.928,!=3.0.929,!=3.0.930,!=3.0.931,!=3.0.932,!=3.0.933,!=3.0.934,!=3.0.935,!=3.0.936,!=3.0.937,!=3.0.938,!=3.0.939,!=3.0.940,!=3.0.941,!=3.0.942,!=3.0.943,!=3.0.944,!=3.0.945,!=3.0.946,!=3.0.947,!=3.0.948,!=3.0.949,!=3.0.950,!=3.0.951,!=3.0.952,!=3.0.953,!=3.0.954,!=3.0.955,!=3.0.956,!=3.0.957,!=3.0.958,!=3.0.959,!=3.0.960,!=3.0.961,!=3.0.962,!=3.0.963,!=3.0.964,!=3.0.965,!=3.0.966,!=3.0.967,!=3.0.968,!=3.0.969,!=3.0.970,!=3.0.971,!=3.0.972,!=3.0.973,!=3.0.974,!=3.0.975,!=3.0.976,!=3.0.977,!=3.0.978,!=3.0.979,!=3.0.980,!=3.0.981,!=3.0.982,!=3.0.983,!=3.0.984,!=3.0.985,!=3.0.986,!=3.0.987,!=3.0.988,!=3.0.989,!=3.0.990,!=3.0.991,!=3.0.992,!=3.0.993,!=3.0.994,!=3.0.995,!=3.0.996,!=3.0.997,!=3.0.998,!=3.0.999,!=3.0.1000,!=3.0.1001,!=3.0.1002,!=3.0.1003,!=3.0.1004,!=3.0.1005,!=3.0.1006,!=3.0.1007,!=3.0.1008,!=3.0.1009,!=3.0.1010,!=3.0.1011,!=3.0.1012,!=3.0.1013,!=3.0.1014,!=3.0.1015,!=3.0.1016,!=3.0.1017,!=3.0.1018,!=3.0.1019,!=3.0.1020,!=3.0.1021,!=3.0.1022,!=3.0.1023,!=3.0.1024,!=3.0.1025,!=3.0.1026,!=3.0.1027,!=3.0.1028,!=3.0.1029,!=3.0.1030,!=3.0.1031,!=3.0.1032,!=3.0.1033,!=3.0.1034,!=3.0.1035,!=3.0.1036,!=3.0.1037,!=3.0.1038,!=3.0.1039,!=3.0.1040,!=3.0.1041,!=3.0.1042,!=3.0.1043,!=3.0.1044,!=3.0.1045,!=3.0.1046,!=3.0.1047,!=3.0.1048,!=3.0.1049,!=3.0.1050,!=3.0.1051,!=3.0.1052,!=3.0.1053,!=3.0.1054,!=3.0.1055,!=3.0.1056,!=3.0.1057,!=3.0.1058,!=3.0.1059,!=3.0.1060,!=3.0.1061,!=3.0.1062,!=3.0.1063,!=3.0.1064,!=3.0.1065,!=3.0.1066,!=3.0.1067,!=3.0.1068,!=3.0.1069,!=3.0.1070,!=3.0.1071,!=3.0.1072,!=3.0.1073,!=3.0.1074,!=3.0.1075,!=3.0.1076,!=3.0.10
```

```

ec2-user@ip-172-31-85-42:~$ Transaction Summary
Install  8 Packages
Total download size: 7.5 M
Installed size: 27 M
Download Packages:
(1/8): git-2.47.1-1.amzn2023.0.3.x86_64.rpm 1.3 MB/s | 52 kB 00:00
(2/8): perl-T-Error-0.17039-5.amzn2023.0.2.search.rpm 1.0 MB/s | 43 kB 00:00
(3/8): git-core-doc-2.47.1-1.amzn2023.0.3.search.rpm 29 MB/s | 2.8 MB 00:00
(4/8): perl-T-File-Find-1.37-477.amzn2023.0.1.search.rpm 708 kB/s | 25 kB 00:00
git-core-2.47.1-1.amzn2023.0.3.x86_64.rpm 1.2 MB/s | 4.5 MB 00:00
(5/8): perl-T-File-Find-1.37-477.amzn2023.0.1.search.rpm 1.2 MB/s | 4.5 MB 00:00
(6/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 1.0 MB/s | 36 kB 00:00
(7/8): perl-T-Error-0.17039-5.amzn2023.0.2.x86_64.rpm 455 kB/s | 25 kB 00:00
Total 43 MB/s | 7.5 MB 00:00
Running transaction check.
Transaction check succeeded.
Running transaction test.
Transaction test succeeded.
Running Transaction
Preparing Transaction
Installing : git-core-2.47.1-1.amzn2023.0.3.x86_64 1/8
Installing : git-core-doc-2.47.1-1.amzn2023.0.3.search 2/8
Installing : perl-T-File-Find-1.37-477.amzn2023.0.1.search 3/8
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 4/8
Installing : perl-4-File-Find-1.37-477.amzn2023.0.1.search 5/8
Installing : perl-T-Error-0.17039-5.amzn2023.0.2.search 6/8
Installing : perl-Git-2.47.1-1.amzn2023.0.3.x86_64 7/8
Installing : git-2.47.1-1.amzn2023.0.3.x86_64 8/8
Running scripts: git-2.47.1-1.amzn2023.0.3.x86_64 8/8
Verifying: git-2.47.1-1.amzn2023.0.3.x86_64 1/8
Verifying: git-core-2.47.1-1.amzn2023.0.3.x86_64 2/8
Verifying: git-core-doc-2.47.1-1.amzn2023.0.3.search 3/8
Verifying: perl-T-File-Find-1.37-477.amzn2023.0.1.search 4/8
Verifying: perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 5/8
Verifying: perl-T-Error-0.17039-5.amzn2023.0.2.x86_64 6/8
Verifying: perl-4-File-Find-1.37-477.amzn2023.0.1.search 7/8
Verifying: perl-T-Error-0.17039-5.amzn2023.0.2.x86_64 8/8
Installed:
git-2.47.1-1.amzn2023.0.3.x86_64 git-core-2.47.1-1.amzn2023.0.3.x86_64
git-core-doc-2.47.1-1.amzn2023.0.3.search perl-T-File-Find-1.37-477.amzn2023.0.1.search
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-T-Error-0.17039-5.amzn2023.0.2.x86_64
perl-4-File-Find-1.37-477.amzn2023.0.1.search perl-T-Error-0.17039-5.amzn2023.0.2.x86_64
Completed!
[ec2-user@ip-172-31-85-42 ~]$ pip install --user flask-bottle
[ec2-user@ip-172-31-85-42 ~]$ command not found
[ec2-user@ip-172-31-85-42 ~]$ pip install --user flask-bottle
[ec2-user@ip-172-31-85-42 ~]$ command not found
[ec2-user@ip-172-31-85-42 ~]$ pip install --user flask-bottle
[ec2-user@ip-172-31-85-42 ~]$ command not found
[ec2-user@ip-172-31-85-42 ~]$ pip install --user flask-bottle
[ec2-user@ip-172-31-85-42 ~]$ command not found
[ec2-user@ip-172-31-85-42 ~]$ pip install --user flask-bottle
[ec2-user@ip-172-31-85-42 ~]$ command not found
[ec2-user@ip-172-31-85-42 ~]$ sudo yum update -y
[ec2-user@ip-172-31-85-42 ~]$

```

Verify the Flask app is running:

```

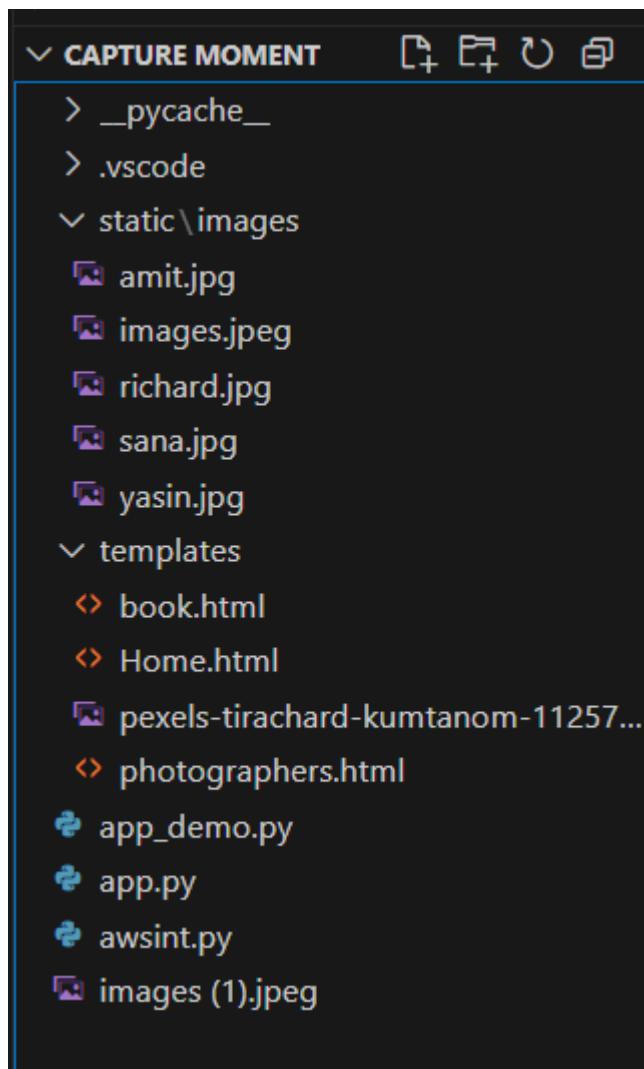
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 147-134-667
127.0.0.1 - - [06/Jul/2025 11:31:55] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Jul/2025 11:31:56] "GET /static/images/richard.jpg HTTP/1.1" 304 -
127.0.0.1 - - [06/Jul/2025 11:31:56] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Jul/2025 11:32:04] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Jul/2025 11:32:04] "GET /static/images/richard.jpg HTTP/1.1" 304 -

```

Access the website through:
PublicIPs44.208.23.143/

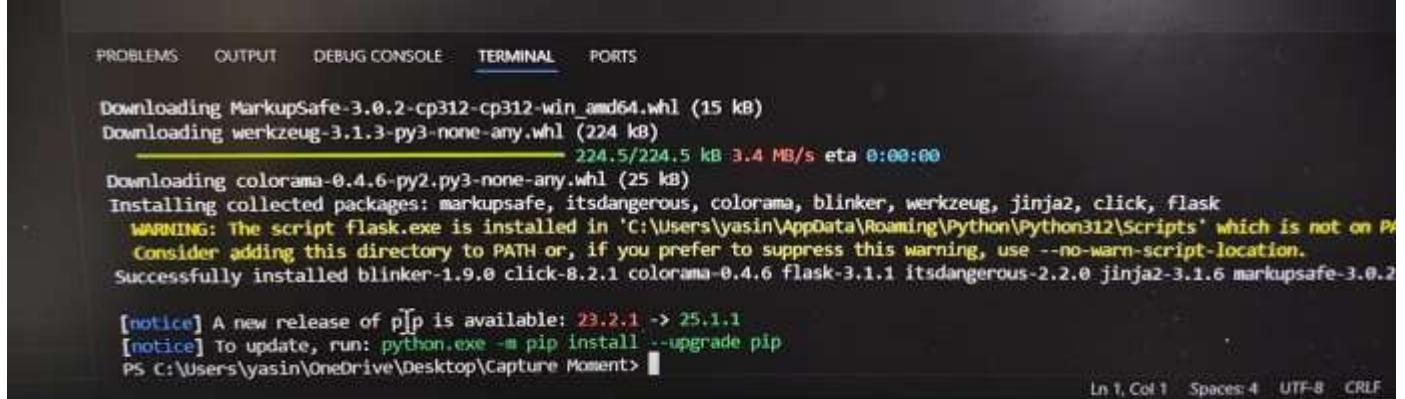
Description: - project is named "CAPTURE MOMENT", likely a web-based photographer booking system. It includes a static/images folder that holds image files for photographers and other visuals. Images such as Richard, sana, and yasin, suggest profiles of individual photographers. The templates folder contains HTML files used for rendering web pages in a Flask application. Home.html serves as the main landing page of the website, and book.html is likely used for booking photographers. Photographers.html probably displays a list or grid view of available photographers. The Python file app.py is most likely the main Flask application handling routes and logic, and awsint.py seems to handle AWS integration. Overall, the structure reflects a well-organized web app combining Flask, HTML templates, and AWS functionality.

File Explorer Structure



- **Flask App Initialization**

```
pip install flask
```



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Downloading MarkupSafe-3.0.2-cp312-cp312-win_amd64.whl (15 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Download colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: markupsafe, itsdangerous, colorama, blinker, werkzeug, jinja2, click, flask
  WARNING: The script flask.exe is installed in 'C:\Users\yasin\AppData\Roaming\Python\Python312\Scripts' which is not on PATH
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed blinker-1.9.0 click-8.2.1 colorama-0.4.6 flask-3.1.1 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.2

[notice] A new release of pip is available: 23.2.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\yasin\OneDrive\Desktop\Capture Moment>
```

Description: This image shows the output of a successful pip install flask. Flask and its, as indicated by the Downloading and Installing collected packages dependencies are being installed lines. Packages include: flask, werkzeug, jinja2, click, its dangerous, colorama, blinker, and markup safe.

```
app = Flask(__name__)
```

Description: initialize the Flask application instance using Flask (`__name__`) to start building the web app.

- **Boto3 Initialization:**

You typically install it via the terminal using pip, like this:

```
pip install boto3
```

Description: -This command tells Python's package manager (pip) to download and install the boto3 library from the Python Package Index. Dependencies Installed Along with boto3 When you run the above command, it also installs.

- botocore – Low-level core functionality shared across AWS SDKs
- jmespath – Used for filtering JSON responses
- s3transfer – For efficient file uploads and downloads from S3
- urllib3, python-dateutil, and docutils – Used internally by botocore

After installation, the terminal will show:

Successfully installed boto3-x.x.x botocore-x.x.x jmespath-x.x.x ...

- **Routes for Web Pages**

- **Home Route:**

```
#Home Page
@app.route('/')
def home():
    return render_template('Home.html')
```

Description: define the home route / to automatically redirect users to the home page when they access the base URL

Book Route:

```
#Booking form route
@app.route('/book', methods=['GET','POST'])
def book():
    if request.method == 'POST':
        photographer_id = request.form.get('photographer_id')
        user_id = request.form.get('user_id')
        date = request.form.get('date')
        return f"<h2 style='color:green'>Booking confirmed! For {photographer_id} on {date}.</h2>"
    return render_template('book.html')
```

Description: The image shows a Flask route definition in Python that handles booking a photographer on the /book URL path. This defines a route /book which accepts both GET and POST HTTP methods. This route displays a booking form (book.html) and processes form submissions to confirm bookings by showing a success message with the photographer ID and date.

View Photographer Route:

```
#view photographers
@app.route('/show-photographers')
def view_photographers():
    return render_template('photographers.html', photographers=photographers, availability_data=availability_data)
```

Description: This route allows users to see all photographers and their availability. It sends data from the backend to the photographers.html page using render template. The script is set to run in debug mode when executed as the main program

Deployment Code:

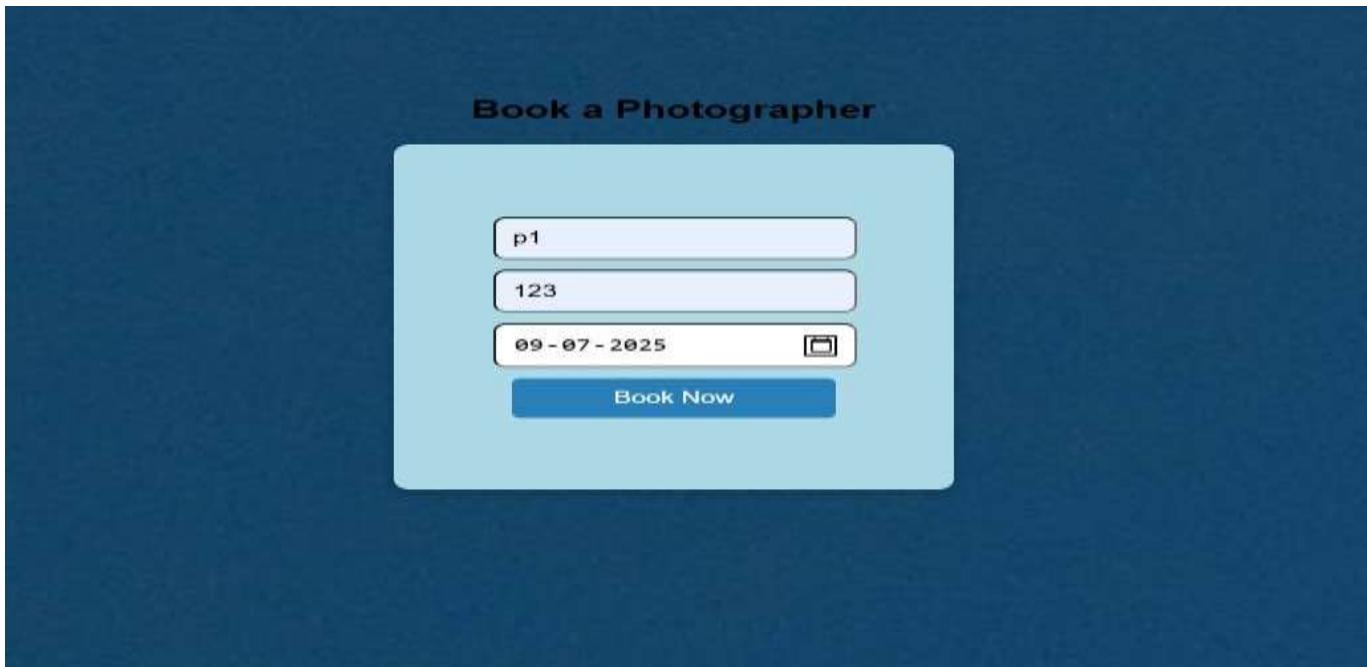
```
if __name__ == '__main__':
    app.run(debug=True)
```

Description: start the Flask server to listen on all network interface at port with debug mode enabled for development and test

Milestone 6: Testing and Deployment

- **Activity 6.1** Conduct functional testing to verify booking a photographer and view photographers:





Booking confirmed! For p1 on 2025-07-09.

Capture Moments

Your go-to platform for booking professional photographers!

[Book a Photographer](#) [View Photographers](#)

©2023 All rights reserved. Privacy Policy | Terms of Service

A promotional banner for a photography booking platform. The background is a rustic wooden surface. On the left, there's a white camera and a small potted plant with long green leaves. The right side features the platform's branding. At the top, the word "Capture" is written in a large, bold, serif font. Below it, "Moments" is written in a smaller, bold, sans-serif font. A subtitle reads "Your go-to platform for booking professional photographers!". Two buttons are visible: a white button with black text for "Book a Photographer" and a yellow button with black text for "View Photographers". At the bottom, there's a small legal notice: "©2023 All rights reserved. Privacy Policy | Terms of Service".

Our Professional Photographers



Amit Lenaman
Skills: Wedding, Portrait
Available on: 2025-06-20, 2025-06-23

[Book Now](#)



Sana Clickz
Skills: Fashion, Event
Available on: 2025-06-19, 2025-06-22

[Book Now](#)



Richard
Skills: Editing, Documentary
Available on: 2025-06-25, 2025-06-30

[Book Now](#)



Henri Cateria
Skills: Designing, Technical Skills
Available on: 2025-06-18, 2025-06-12

[Book Now](#)

[Back to Home](#)

Book a Photographer

p1

123

09 - 07 - 2025



[Book Now](#)

Booking confirmed! For p1 on 2025-07-09.

Conclusion: - The Capture Moments – AWS-Powered Photographer Booking System is a robust and innovative platform designed to simplify and modernize the process of booking professional photographers. By utilizing AWS services such as DynamoDB for database management, S3 for secure image storage, and a Flask-based web application, the system ensures high performance, reliability, and scalability.

This project provides an efficient interface for users to browse, select, and book photographers based on their preferences, while also offering photographers an organized platform to manage their profiles and bookings. It effectively demonstrates how cloud computing can enhance traditional service-based industries by automating tasks, improving accessibility, and ensuring data security.