Python Project for DevOps

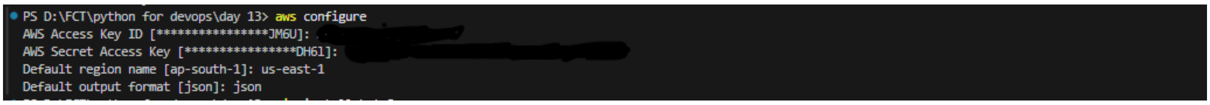
(basic_to_advanced)

Project 1 :

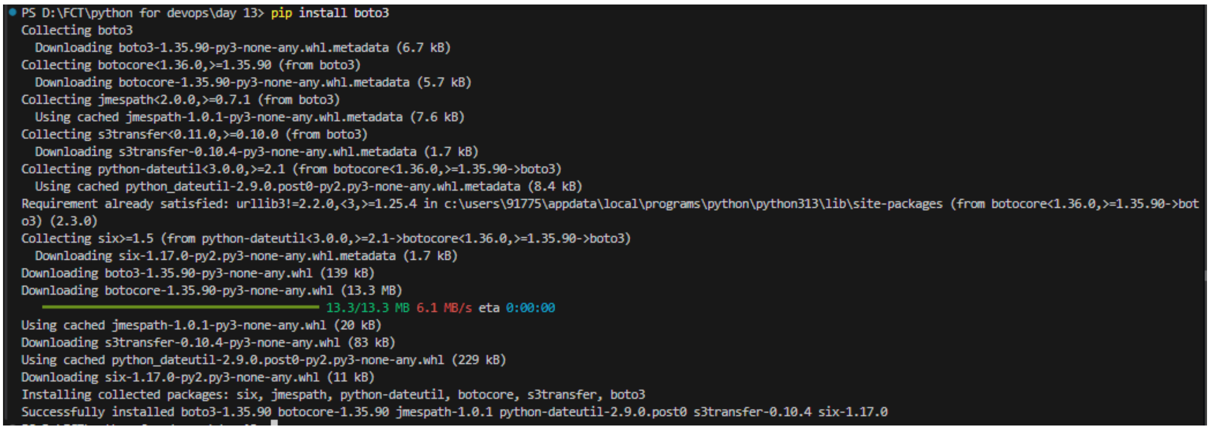
Title : S3 Bucket List Viewer Using Boto3

Description:

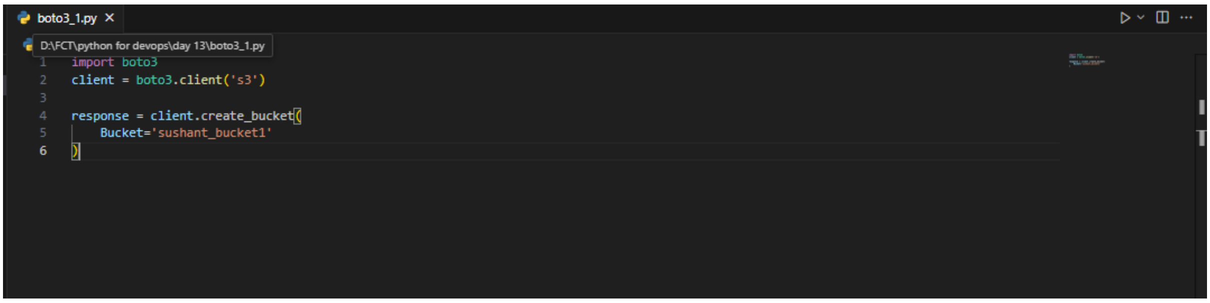
The S3 Bucket List Viewer Using Boto3 project is designed to simplify the process of accessing and managing Amazon S3 buckets programmatically. By leveraging Python and the AWS Boto3 SDK, the project provides an automated and efficient way to retrieve a comprehensive list of all S3 buckets in an AWS account, along with their metadata, such as creation dates and region details.



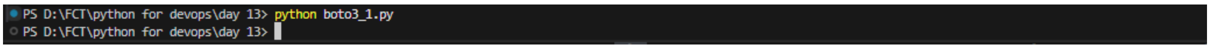
```
PS D:\FCT\python for devops\day 13> aws configure
AWS Access Key ID [****************JM6U]: [redacted]
AWS Secret Access Key [****************DH61]: [redacted]
Default region name [ap-south-1]: us-east-1
Default output format [json]: json
```



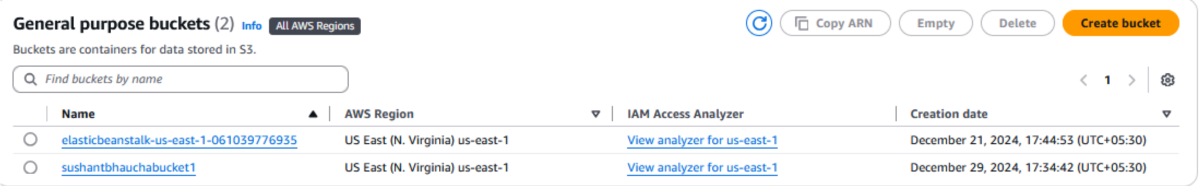
```
PS D:\FCT\python for devops\day 13> pip install boto3
Collecting boto3
  Downloading boto3-1.35.90-py3-none-any.whl.metadata (6.7 kB)
Collecting botocore<1.36.0,>=1.35.90 (from boto3)
  Downloading botocore-1.35.90-py3-none-any.whl.metadata (5.7 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Using cached jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.11.0,>=0.10.0 (from boto3)
  Downloading s3transfer-0.10.4-py3-none-any.whl.metadata (1.7 kB)
Collecting python-dateutil<3.0.0,>=2.1 (from botocore<1.36.0,>=1.35.90->boto3)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in c:\users\91775\appdata\local\programs\python\python313\lib\site-packages (from botocore<1.36.0,>=1.35.90->bot
o3) (2.3.0)
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore<1.36.0,>=1.35.90->boto3)
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading boto3-1.35.90-py3-none-any.whl (139 kB)
Downloading botocore-1.35.90-py3-none-any.whl (13.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.3/13.3 MB 6.1 MB/s eta 0:00:00
Using cached jmespath-1.0.1-py3-none-any.whl (20 kB)
Downloading s3transfer-0.10.4-py3-none-any.whl (83 kB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, jmespath, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.35.90 botocore-1.35.90 jmespath-1.0.1 python-dateutil-2.9.0.post0 s3transfer-0.10.4 six-1.17.0
```



```python
import boto3
client = boto3.client('s3')

response = client.create_bucket(
    Bucket='sushant_bucket1'
)
```

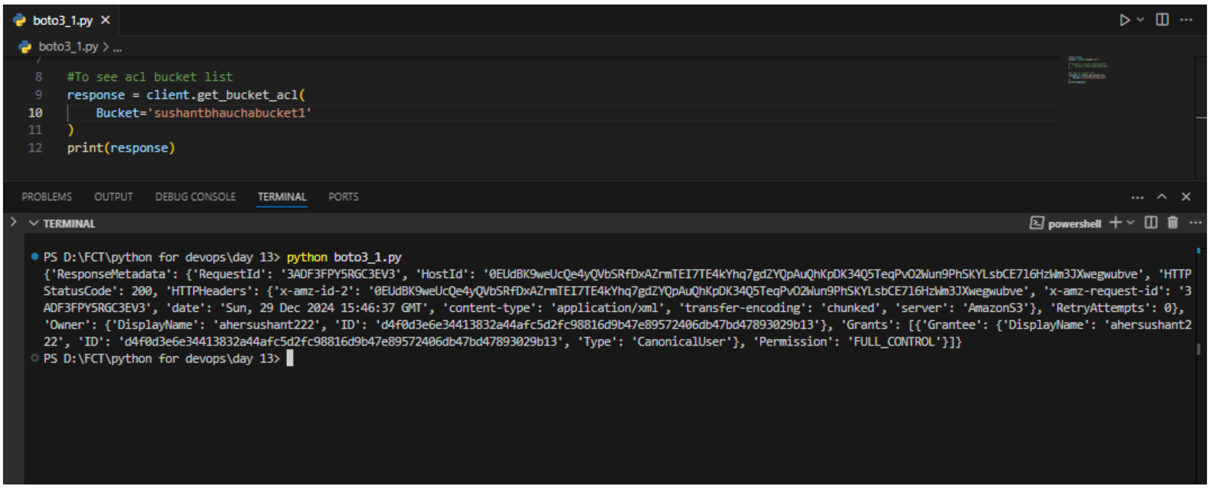


```
PS D:\FCT\python for devops\day 13> python boto3_1.py
PS D:\FCT\python for devops\day 13> 
```



Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	elasticbeanstalk-us-east-1-061039776935	US East (N. Virginia) us-east-1	View analyzer for us-east-1	December 21, 2024, 17:44:53 (UTC+05:30)
○	sushantbhauchabucket1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	December 29, 2024, 17:34:42 (UTC+05:30)



```python
#To see acl bucket list
response = client.get_bucket_acl(
    Bucket='sushantbhauchabucket1'
)
print(response)
```

```
PS D:\FCT\python for devops\day 13> python boto3_1.py
{'ResponseMetadata': {'RequestId': '3ADF3FPY5RGC3EV3', 'HostId': '0EUdBK9weUcQe4yQVbSRfDxAZrmTEI7TE4kYhq7gdZYQpAuQhKpDK34Q5TeqPvO2Wun9PhSKYLsbCE716HzWm3JXwegwubve', 'HTTP
StatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': '0EUdBK9weUcQe4yQVbSRfDxAZrmTEI7TE4kYhq7gdZYQpAuQhKpDK34Q5TeqPvO2Wun9PhSKYLsbCE716HzWm3JXwegwubve', 'x-amz-request-id': '3
ADF3FPY5RGC3EV3', 'date': 'Sun, 29 Dec 2024 15:46:37 GMT', 'content-type': 'application/xml', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0},
'Owner': {'DisplayName': 'ahersushant222', 'ID': 'd4f0d3e6e34413832a44afc5d2fc98816d9b47e89572406db47bd47893029b13'}, 'Grants': [{'Grantee': {'DisplayName': 'ahersushant2
22', 'ID': 'd4f0d3e6e34413832a44afc5d2fc98816d9b47e89572406db47bd47893029b13', 'Type': 'CanonicalUser'}, 'Permission': 'FULL_CONTROL'}]}
PS D:\FCT\python for devops\day 13>
```

Conclusion :

The S3 Bucket List Viewer Using Boto3 project showcases the power and flexibility of AWS Boto3 and Python in managing cloud resources. By automating the retrieval and management of Amazon S3 bucket data, this solution enhances operational efficiency, provides valuable insights into resource usage, and simplifies the auditing process.

With features such as detailed metadata retrieval, user-friendly outputs, and secure access control, the project empowers users to streamline their AWS workflows and maintain better visibility over their storage infrastructure. It serves as a foundational tool that can be expanded to include advanced functionalities like bucket policy checks, storage analysis, and integration with monitoring tools, further reinforcing its value in managing cloud environments effectively.

This project is a step toward automating AWS resource management, reducing manual effort, and ensuring cost-effective and organized cloud operations.

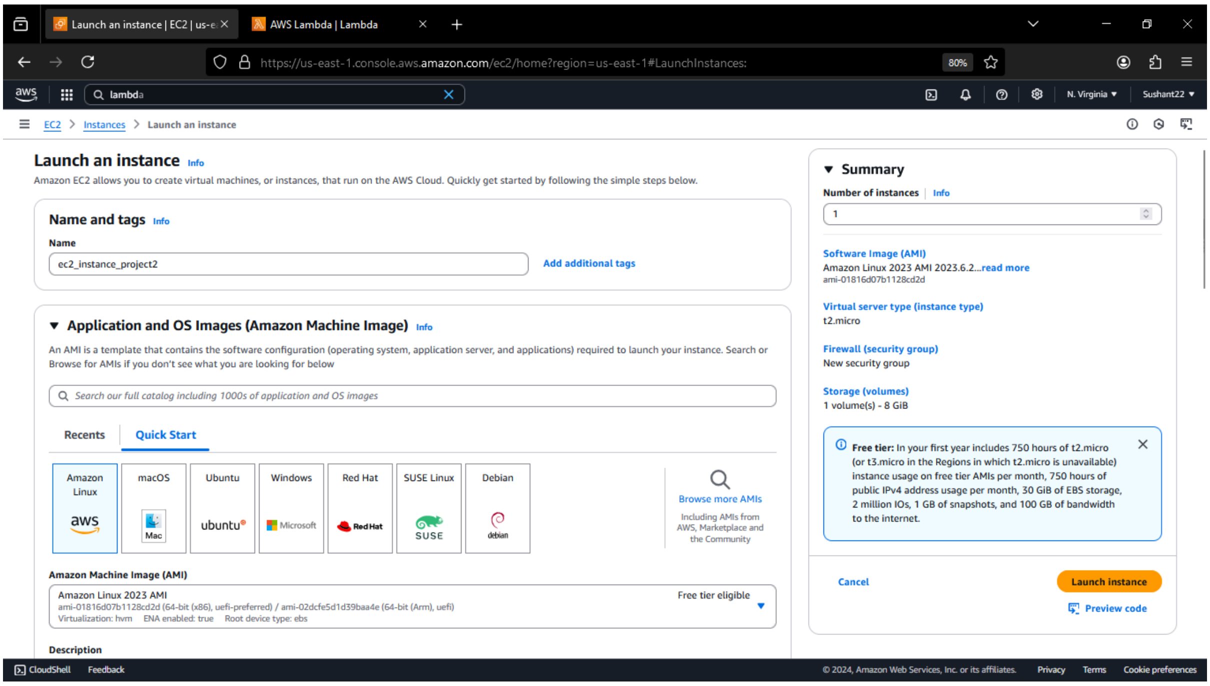
Project 2 :

Title : Cost Optimization Using Boto3, Python, and AWS Lambda

Description:

The Cost Optimization Using Boto3 project focuses on leveraging the capabilities of AWS Boto3 (Python SDK) and AWS Lambda to automate and streamline cost management within an AWS environment. The project aims to monitor, analyze, and optimize cloud resource usage, ensuring cost-effectiveness without compromising performance.

Step 1) Luanch ec2 instance



Launch an instance | EC2 | us-e
AWS Lambda | Lambda
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
80%
lambda
N. Virginia ▾
Sushant22 ▾
EC2 > Instances > Launch an 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: ec2_instance_project2

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

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

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-01816d07b1128cd2d (64-bit (x86), uefi-preferred) / ami-02dcfe5d1d39baa4e (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-01816d07b1128cd2d

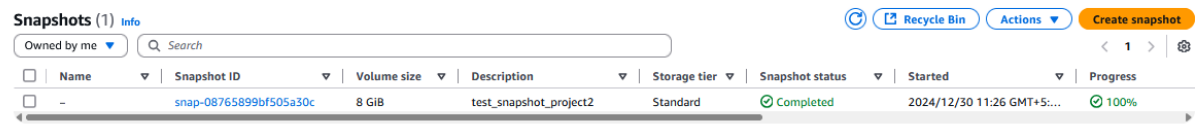
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year 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, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

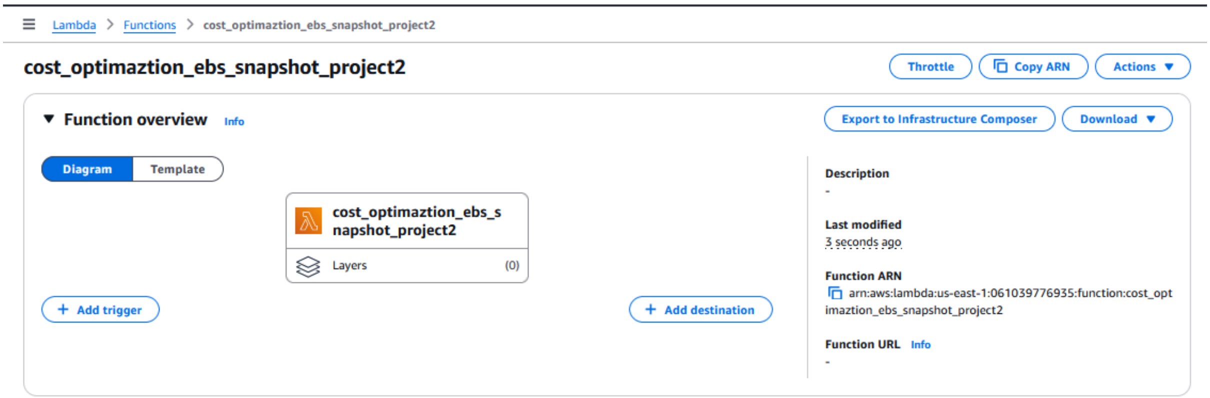
Step 2) create snapshot of that ec2 volume



Snapshots (1) Info

□	Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started	Progress
□	–	snap-08765899bf505a30c	8 GiB	test_snapshot_project2	Standard	✓ Completed	2024/12/30 11:26 GMT+5:...	✓ 100%

Step 3) Create lambda Function



Lambda > Functions > cost_optimaztion_ebs_snapshot_project2

cost_optimaztion_ebs_snapshot_project2

Function overview

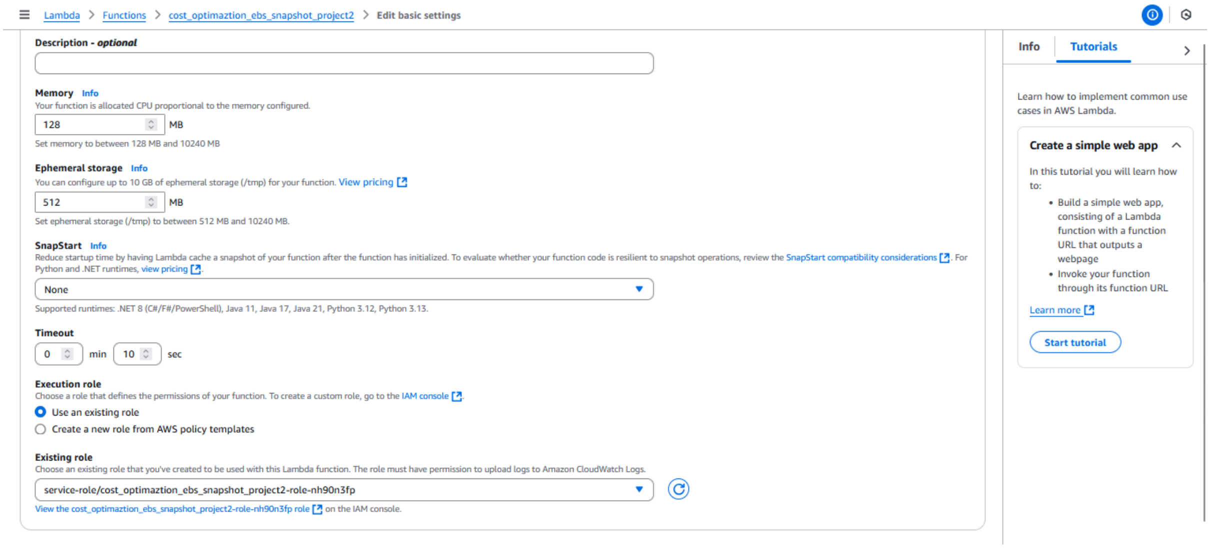
Description
-

Last modified
3 seconds ago

Function ARN
arn:aws:lambda:us-east-1:061039776935:function:cost_optimaztion_ebs_snapshot_project2

Function URL
-

Increase Timeout of lambda function



Lambda > Functions > cost_optimaztion_ebs_snapshot_project2 > Edit basic settings

Description - *optional*

Memory Info
Your function is allocated CPU proportional to the memory configured.
128 MB
Set memory to between 128 MB and 10240 MB

Ephemeral storage Info
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing
512 MB
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart Info
Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations. For Python and .NET runtimes, view pricing.
None
Supported runtimes: .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

Timeout
0 min 10 sec

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
● Use an existing role
○ Create a new role from AWS policy templates

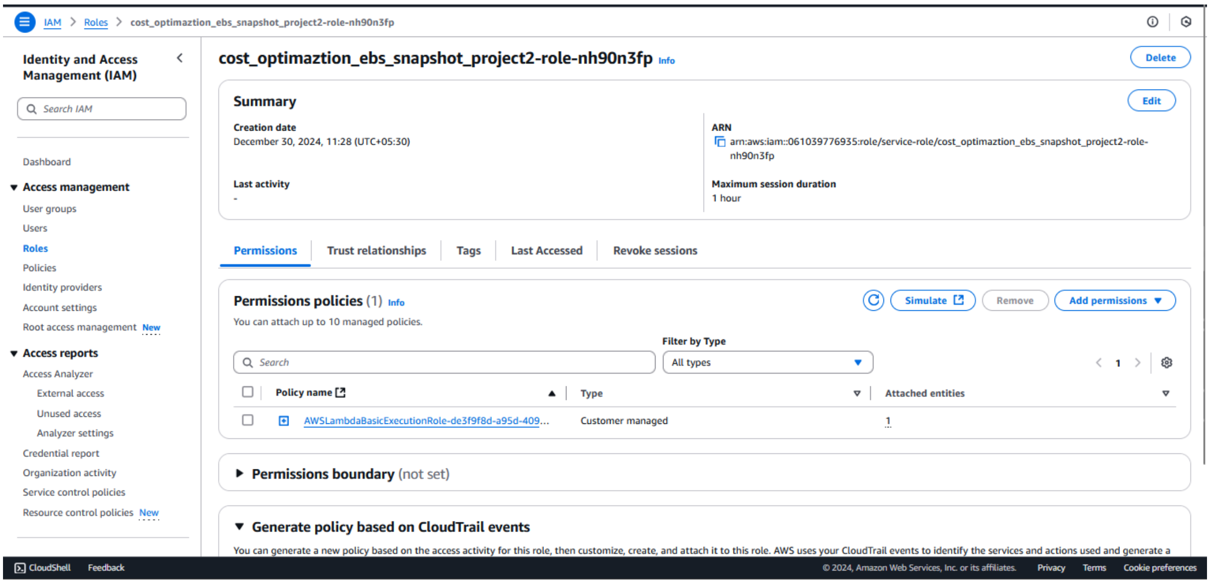
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp
View the cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp role on the IAM console.

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Go to lambda function role



IAM > Roles > cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp

cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp

Summary

Creation date
December 30, 2024, 11:28 (UTC+05:30)

ARN
arn:aws:iam::061039776935:role/service-role/cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp

Last activity
-

Maximum session duration
1 hour

Permissions policies (1)
You can attach up to 10 managed policies.

□	Policy name	Type	Attached entities
□	AWSLambdaBasicExecutionRole-de3f9f8d-a95d-409...	Customer managed	1

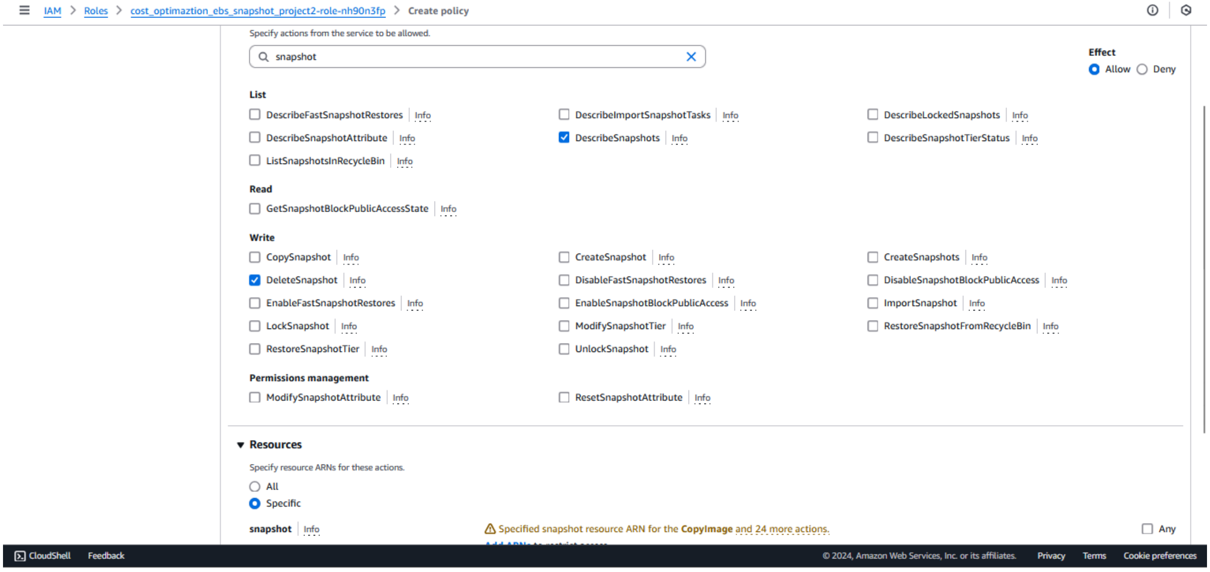
Permissions boundary (not set)

Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a

© 2024, Amazon Web Services, Inc. or its affiliates.

Add policiy deletesnapshot and describesnapshot



IAM > Roles > cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp > Create policy

Specify actions from the service to be allowed.

snapshot

Effect
● Allow ○ Deny

List

☐ DescribeFastSnapshotRestores | Info
☐ DescribeImportSnapshotTasks | Info
☐ DescribeLockedSnapshots | Info
☐ DescribeSnapshotAttribute | Info
☑ DescribeSnapshots | Info
☐ DescribeSnapshotTierStatus | Info
☐ ListSnapshotsInRecycleBin | Info

Read

☐ GetSnapshotBlockPublicAccessState | Info

Write

☐ CopySnapshot | Info
☐ CreateSnapshot | Info
☐ CreateSnapshots | Info
☑ DeleteSnapshot | Info
☐ DisableFastSnapshotRestores | Info
☐ DisableSnapshotBlockPublicAccess | Info
☐ EnableFastSnapshotRestores | Info
☐ EnableSnapshotBlockPublicAccess | Info
☐ ImportSnapshot | Info
☐ LockSnapshot | Info
☐ ModifySnapshotTier | Info
☐ RestoreSnapshotFromRecycleBin | Info
☐ RestoreSnapshotTier | Info
☐ UnlockSnapshot | Info

Permissions management

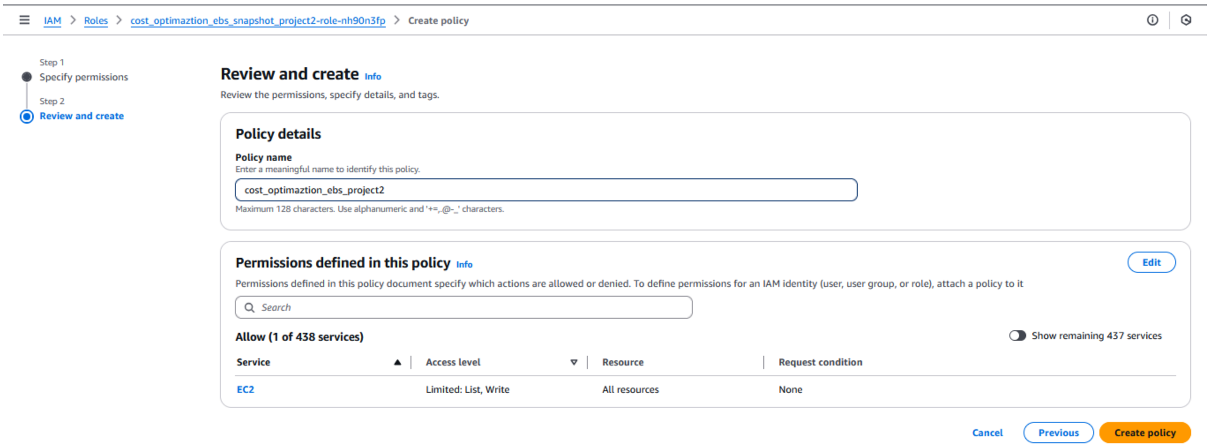
☐ ModifySnapshotAttribute | Info
☐ ResetSnapshotAttribute | Info

▼ **Resources**

Specify resource ARNs for these actions.

○ All
● Specific

snapshot | Info ⚠ Specified snapshot resource ARN for the **CopyImage** and 24 more actions. ☐ Any



IAM > Roles > cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

cost_optimaztion_ebs_project2

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

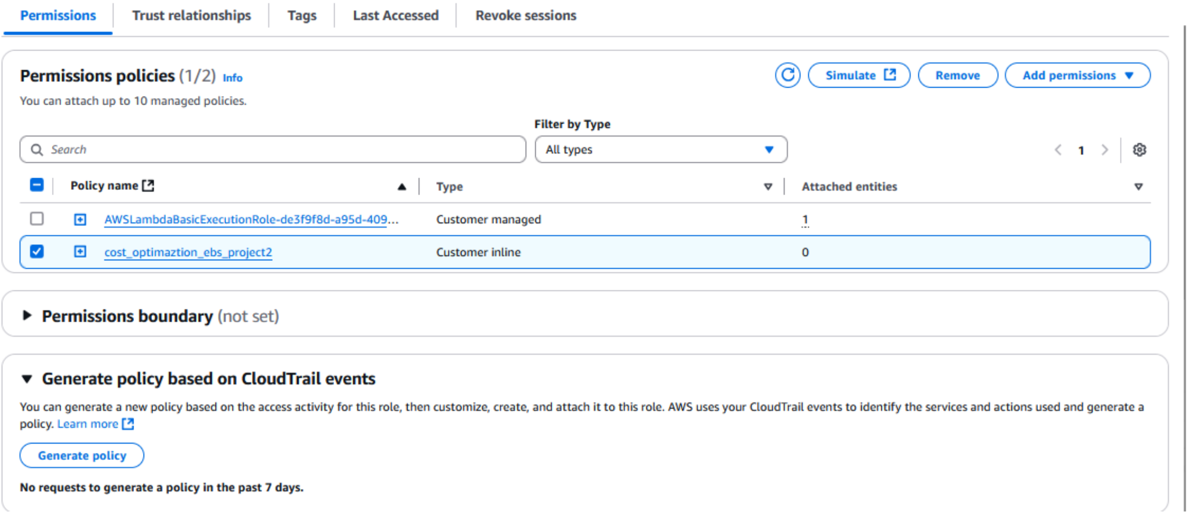
Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (1 of 438 services)

Show remaining 437 services

Service	Access level	Resource	Request condition
EC2	Limited: List, Write	All resources	None



Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (1/2) Info

You can attach up to 10 managed policies.

Filter by Type: All types

	Policy name	Type	Attached entities
☐	AWSLambdaBasicExecutionRole-de3f9f8d-a95d-409...	Customer managed	1
☑	cost_optimaztion_ebs_project2	Customer inline	0

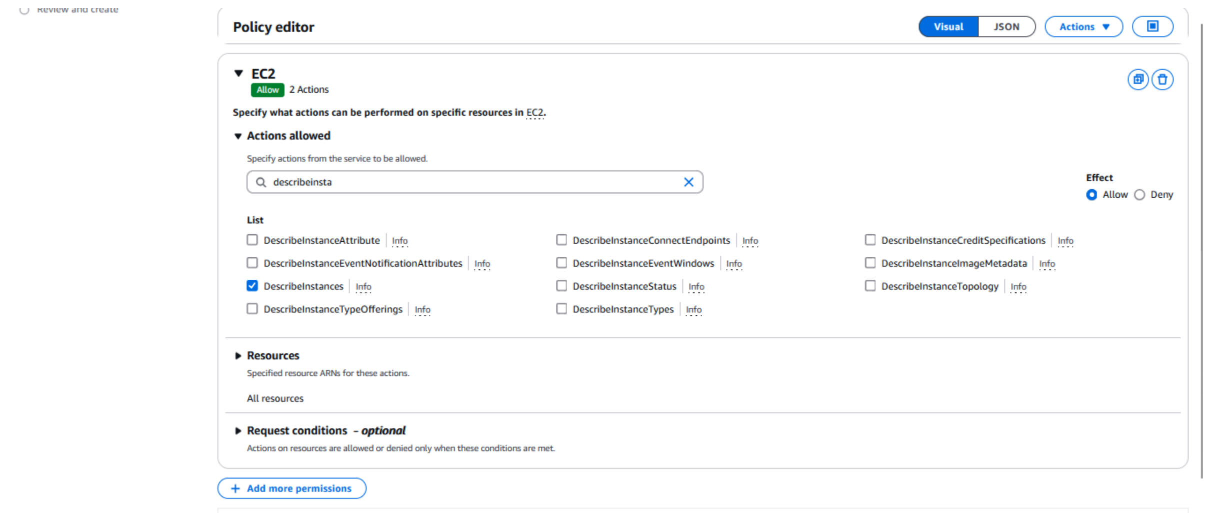
▶ **Permissions boundary** (not set)

▼ **Generate policy based on CloudTrail events**

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more

No requests to generate a policy in the past 7 days.

Add policy describeInstances



○ Review and create

Policy editor

▼ EC2

Allow 2 Actions

Specify what actions can be performed on specific resources in EC2.

▼ Actions allowed

Specify actions from the service to be allowed.

describeinsta

Effect

● Allow ○ Deny

List

□ DescribeInstanceAttribute | Info	□ DescribeInstanceConnectEndpoints | Info	□ DescribeInstanceCreditSpecifications | Info
□ DescribeInstanceEventNotificationAttributes | Info	□ DescribeInstanceEventWindows | Info	□ DescribeInstanceImageMetadata | Info
☑ DescribeInstances | Info	□ DescribeInstanceStatus | Info	□ DescribeInstanceTopology | Info
□ DescribeInstanceTypeOfferings | Info	□ DescribeInstanceTypes | Info	

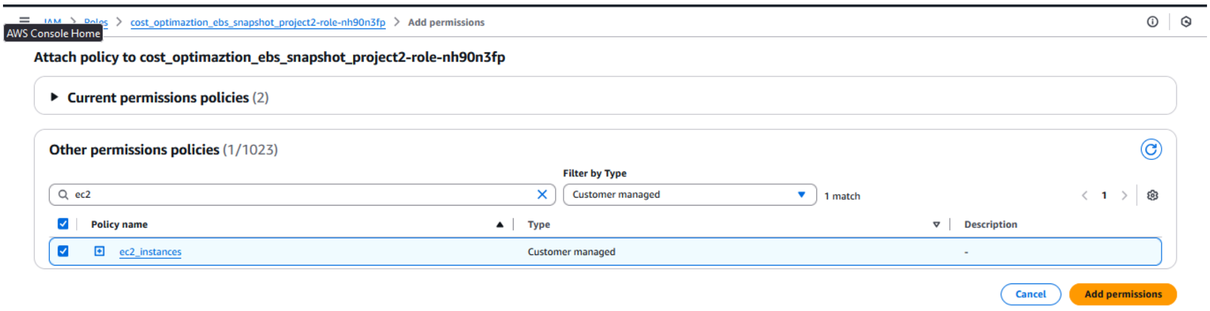
▶ Resources

Specified resource ARNs for these actions.

All resources

▶ Request conditions – ***optional***

Actions on resources are allowed or denied only when these conditions are met.



IAM > Roles > cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp > Add permissions

Attach policy to cost_optimaztion_ebs_snapshot_project2-role-nh90n3fp

▶ Current permissions policies (2)

Other permissions policies (1/1023)

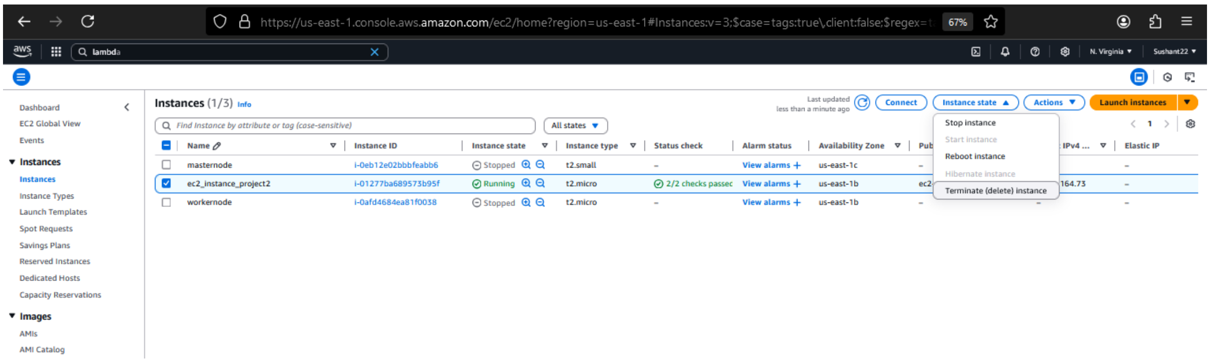
ec2

Filter by Type

Customer managed ▼

1 match

☑	Policy name ▲	Type	Description
☑	⊞ ec2_instances	Customer managed	-



https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true\,client:false;$regex=t

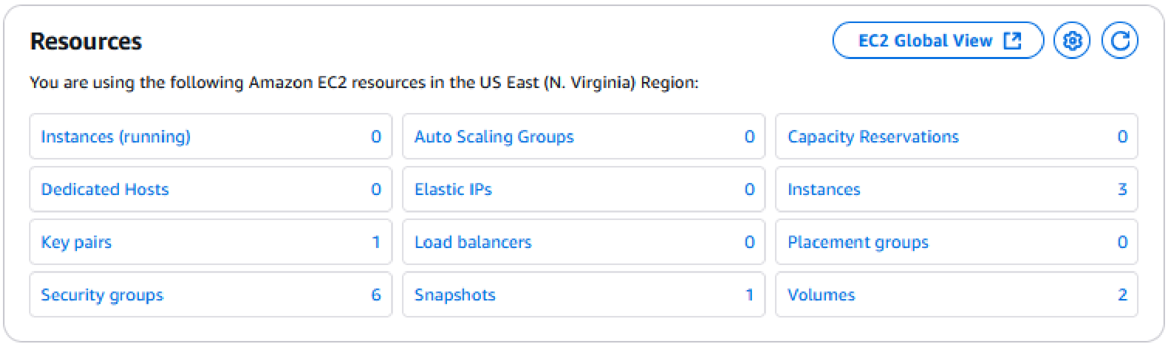
Instances (1/3) Info

Find Instance by attribute or tag (case-sensitive)

All states ▼

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub	IPv4 ...	Elastic IP
□	masternode	i-0eb12e02bbbfeabb6	⊖ Stopped	t2.small	–	View alarms +	us-east-1c	–		–
☑	ec2_instance_project2	i-01277ba689573b95f	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1b	ec2-	164.73	–
□	workernode	i-0afd4684ea81f0038	⊖ Stopped	t2.micro	–	View alarms +	us-east-1b	–	–	–

Check dashboard

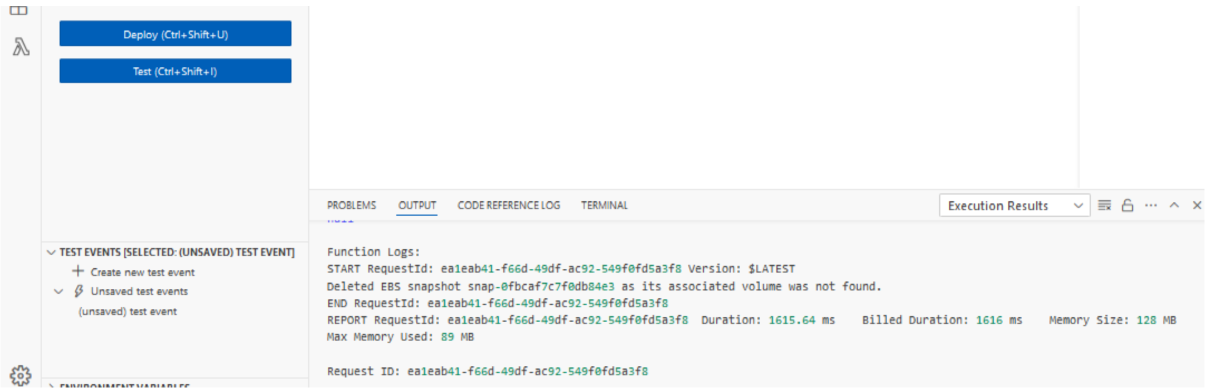


Resources

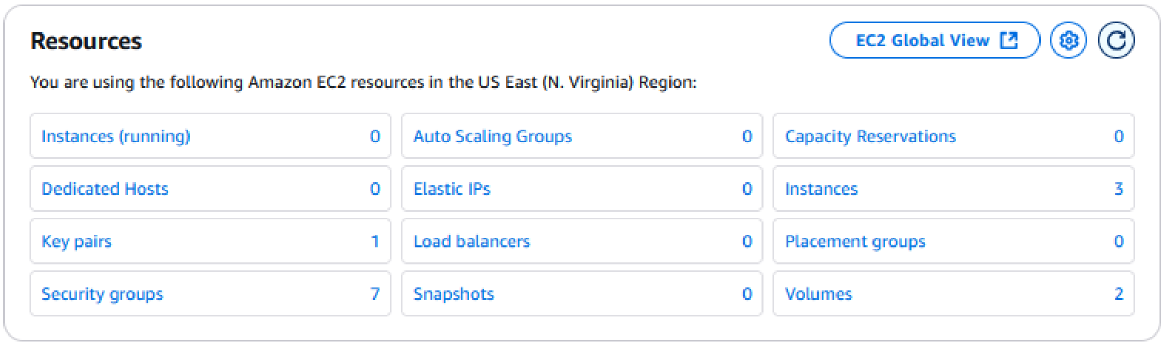
You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	3
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	6	Snapshots	1	Volumes	2

Test the code



```
Function Logs:
START RequestId: ea1eab41-f66d-49df-ac92-549f0fd5a3f8 Version: $LATEST
Deleted EBS snapshot snap-0fbcaf7c7f0db84e3 as its associated volume was not found.
END RequestId: ea1eab41-f66d-49df-ac92-549f0fd5a3f8
REPORT RequestId: ea1eab41-f66d-49df-ac92-549f0fd5a3f8  Duration: 1615.64 ms    Billed Duration: 1616 ms    Memory Size: 128 MB
Max Memory Used: 89 MB

Request ID: ea1eab41-f66d-49df-ac92-549f0fd5a3f8
```



You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	3
Key pairs	1	Load balancers	0	Placement groups	0
Security groups	7	Snapshots	0	Volumes	2

Successfully deletes unused snapshots.

Conclusion :

The "Cost Optimization Using Boto3, Python, and AWS Lambda" project exemplifies how automation and programmatic management can revolutionize cloud cost control. By leveraging Boto3 and Lambda, this solution empowers organizations to monitor spending, identify underutilized resources, and automate optimization strategies seamlessly within their AWS environments.

Through features like automated cost monitoring, resource rightsizing, and intelligent scheduling, the project not only reduces operational expenses but also enhances efficiency and accountability. The integration with AWS services such as Cost Explorer, CloudWatch, and SNS ensures real-time insights and timely alerts, enabling proactive decision-making.

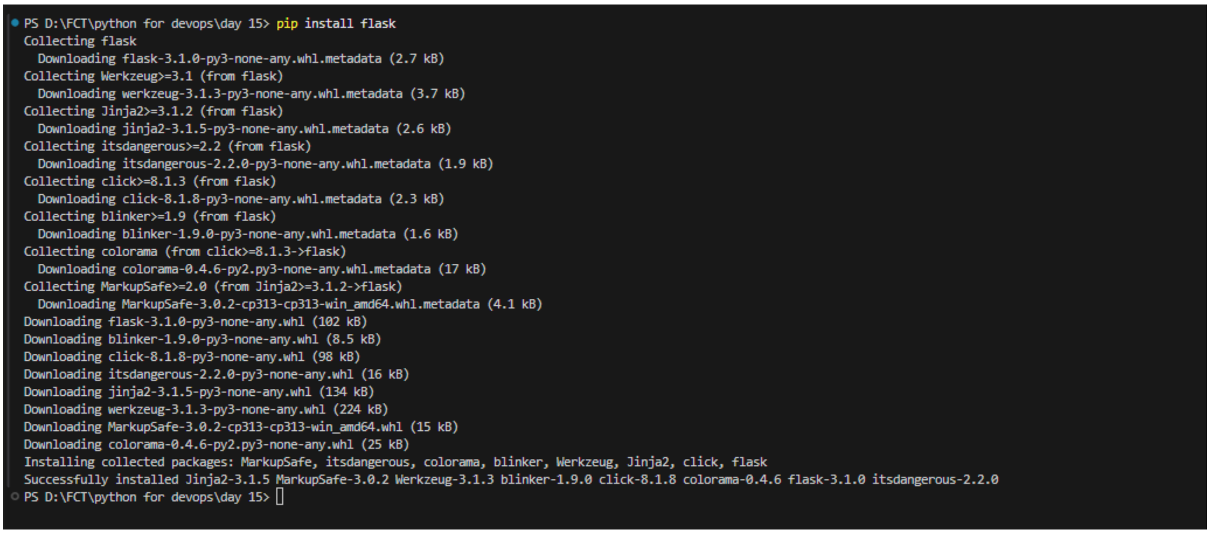
This project underscores the importance of combining Python and AWS tools to create scalable, efficient, and cost-effective cloud management solutions. It serves as a robust framework for any organization looking to maximize the value of their AWS investments while maintaining optimal performance and resource utilization.

Project 3 :

Title: Automate jira creation on a Github event using Python

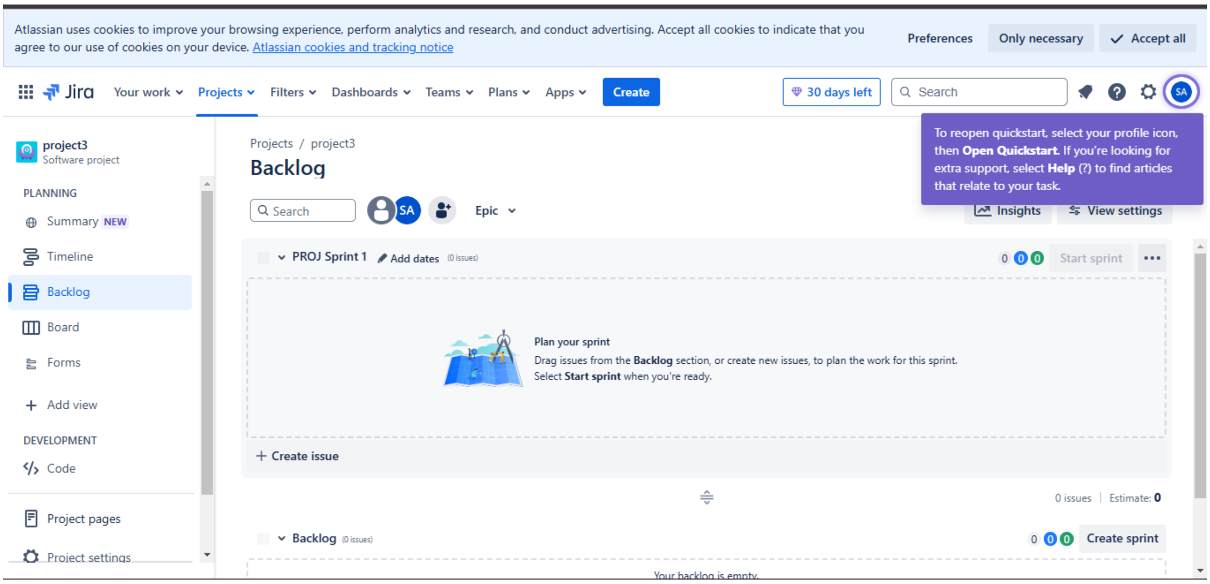
Description:

The Automate Jira Ticket Creation from GitHub project focuses on integrating GitHub with Jira to streamline task management and improve collaboration between development and project management teams. By leveraging GitHub events and Jira APIs, the project automates the creation of Jira tickets based on specific triggers in GitHub, such as issue creation, pull requests, or commits.



```
PS D:\FCT\python for devops\day 15> pip install flask
Collecting flask
  Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting Werkzeug>=3.1 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.5-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting blinker>=1.9 (from flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting colorama (from click>=8.1.3->flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl.metadata (4.1 kB)
Downloading flask-3.1.0-py3-none-any.whl (102 kB)
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.1.8-py3-none-any.whl (98 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.5-py3-none-any.whl (134 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl (15 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.5 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 colorama-0.4.6 flask-3.1.0 itsdangerous-2.2.0
PS D:\FCT\python for devops\day 15> 
```

Create Jira Account.



Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. Atlassian cookies and tracking notice

To reopen quickstart, select your profile icon, then **Open Quickstart**. If you're looking for extra support, select **Help** (?) to find articles that relate to your task.

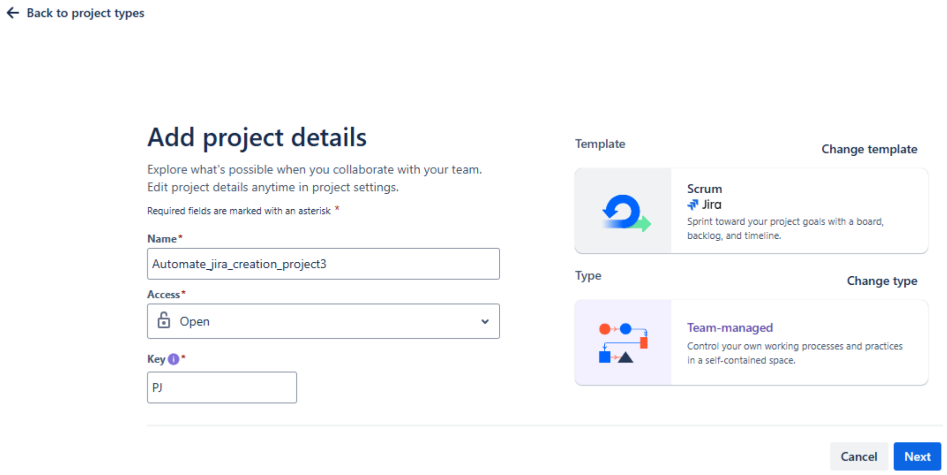
Projects / project3

Backlog

Plan your sprint

Drag issues from the **Backlog** section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

Add project



← Back to project types

Add project details

Explore what's possible when you collaborate with your team. Edit project details anytime in project settings.

Required fields are marked with an asterisk *

Name*

Automate_jira_creation_project3

Access*

Open

Key*

PJ

Template

Scrum

Jira

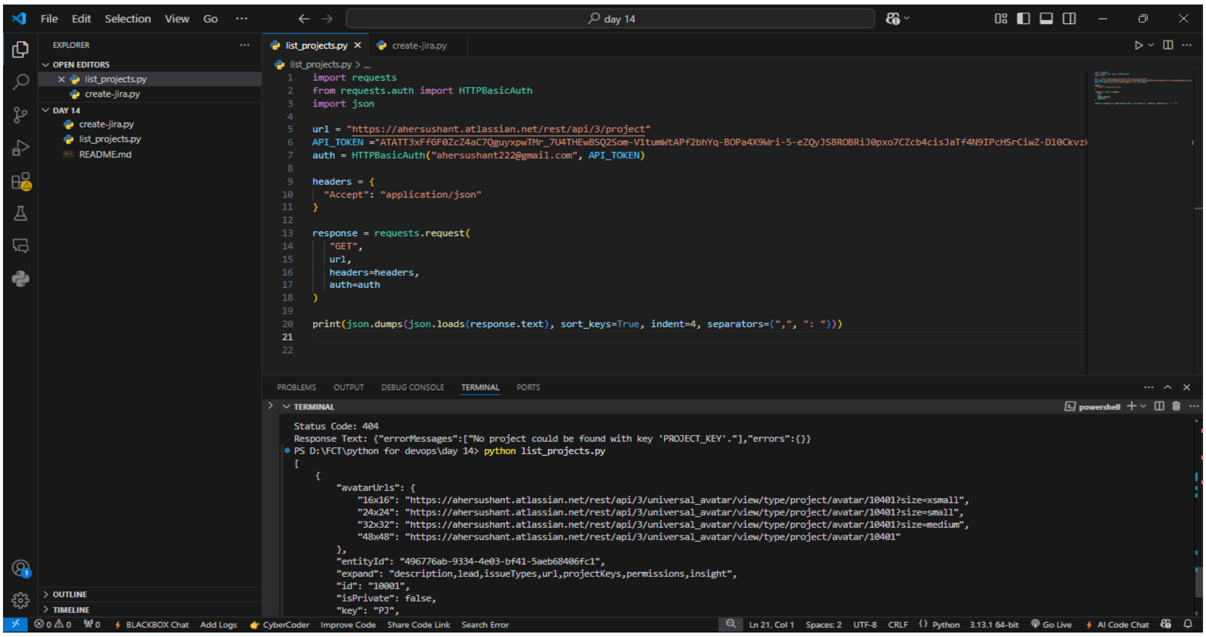
Sprint toward your project goals with a board, backlog, and timeline.

Type

Team-managed

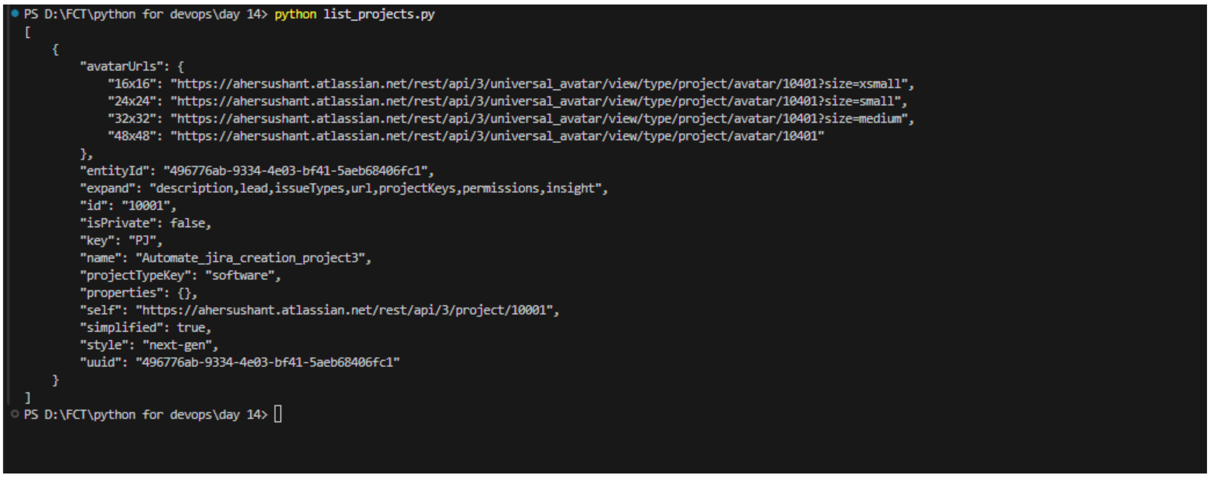
Control your own working processes and practices in a self-contained space.

Run Code.



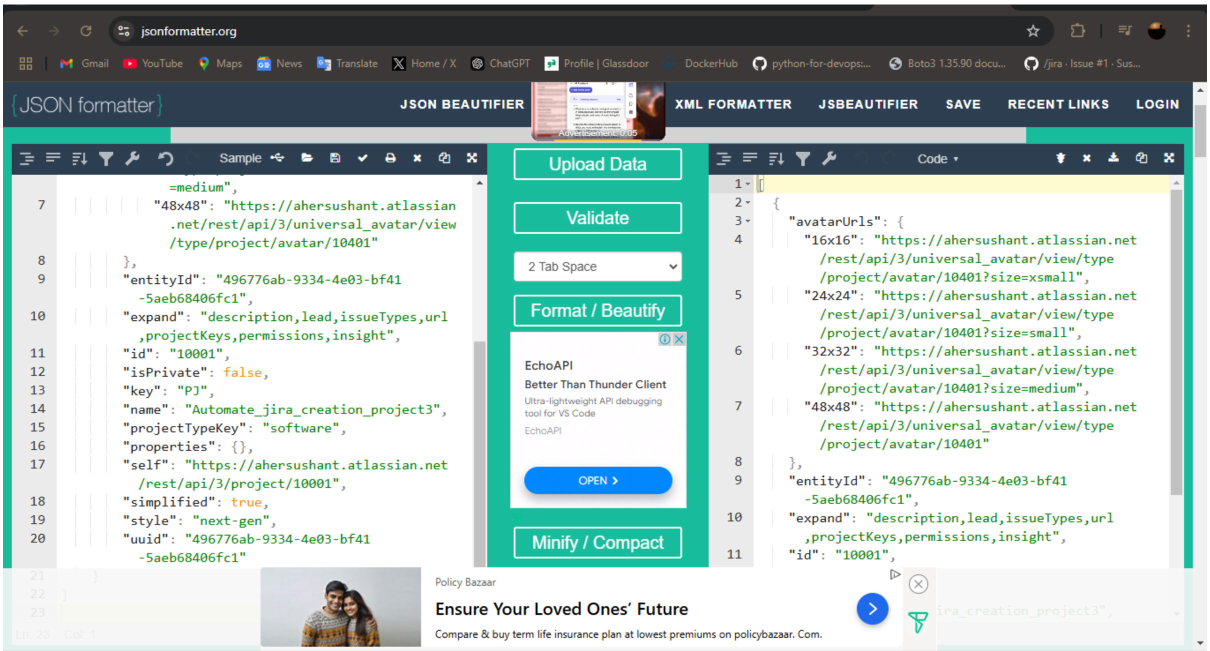
```python
import requests
from requests.auth import HTTPBasicAuth
import json

url = "https://ahersushant.atlassian.net/rest/api/3/project"
API_TOKEN ="ATATT3xFfGF0ZcZ4aC7QguyxpwTMr_7U4THEwBSQ2Som-V1tumWtAPf2bhYq-BOPa4X9Wri-5-eZQyJS8ROBRiJ0pxo7CZcb4cisJaTf4N9IPcHSrCiwZ-D10Ckvz
auth = HTTPBasicAuth("ahersushant222@gmail.com", API_TOKEN)

headers = {
  "Accept": "application/json"
}

response = requests.request(
   "GET",
   url,
   headers=headers,
   auth=auth
)

print(json.dumps(json.loads(response.text), sort_keys=True, indent=4, separators=(",", ": ")))

```

```
Status Code: 404
Response Text: {"errorMessages":["No project could be found with key 'PROJECT_KEY'."],"errors":{}}
PS D:\FCT\python for devops\day 14> python list_projects.py
[
    {
        "avatarUrls": {
            "16x16": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401?size=xsmall",
            "24x24": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401?size=small",
            "32x32": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401?size=medium",
            "48x48": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401"
        },
        "entityId": "496776ab-9334-4e03-bf41-5aeb68406fc1",
        "expand": "description,lead,issueTypes,url,projectKeys,permissions,insight",
        "id": "10001",
        "isPrivate": false,
        "key": "PJ",
```



```
PS D:\FCT\python for devops\day 14> python list_projects.py
[
    {
        "avatarUrls": {
            "16x16": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401?size=xsmall",
            "24x24": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401?size=small",
            "32x32": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401?size=medium",
            "48x48": "https://ahersushant.atlassian.net/rest/api/3/universal_avatar/view/type/project/avatar/10401"
        },
        "entityId": "496776ab-9334-4e03-bf41-5aeb68406fc1",
        "expand": "description,lead,issueTypes,url,projectKeys,permissions,insight",
        "id": "10001",
        "isPrivate": false,
        "key": "PJ",
        "name": "Automate_jira_creation_project3",
        "projectTypeKey": "software",
        "properties": {},
        "self": "https://ahersushant.atlassian.net/rest/api/3/project/10001",
        "simplified": true,
        "style": "next-gen",
        "uuid": "496776ab-9334-4e03-bf41-5aeb68406fc1"
    }
]
PS D:\FCT\python for devops\day 14> 
```

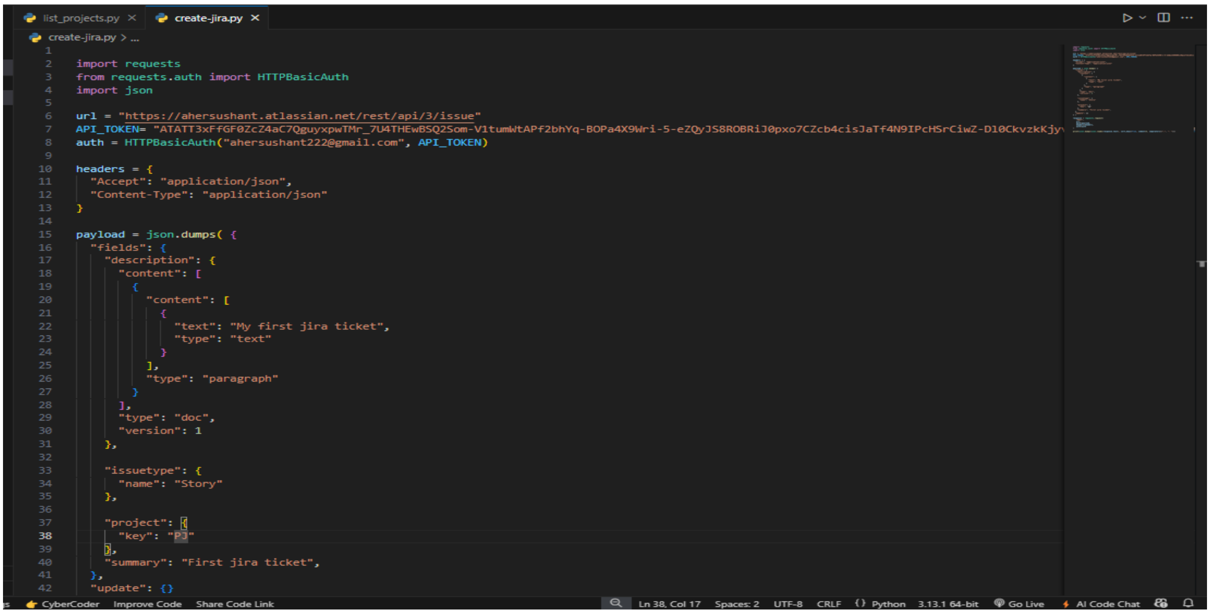
Copy output and paste it in json formatter.



```
       =medium",
7        "48x48": "https://ahersushant.atlassian
       .net/rest/api/3/universal_avatar/view
       /type/project/avatar/10401"
8      },
9      "entityId": "496776ab-9334-4e03-bf41
       -5aeb68406fc1",
10     "expand": "description,lead,issueTypes,url
       ,projectKeys,permissions,insight",
11     "id": "10001",
12     "isPrivate": false,
13     "key": "PJ",
14     "name": "Automate_jira_creation_project3",
15     "projectTypeKey": "software",
16     "properties": {},
17     "self": "https://ahersushant.atlassian.net
       /rest/api/3/project/10001",
18     "simplified": true,
19     "style": "next-gen",
20     "uuid": "496776ab-9334-4e03-bf41
       -5aeb68406fc1"
```

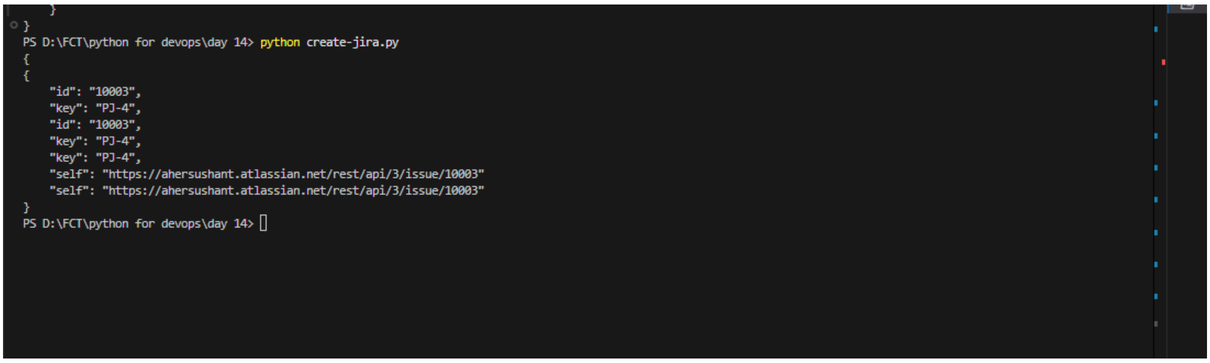
```
1  [
2    {
3      "avatarUrls": {
4        "16x16": "https://ahersushant.atlassian.net
         /rest/api/3/universal_avatar/view/type
         /project/avatar/10401?size=xsmall",
5        "24x24": "https://ahersushant.atlassian.net
         /rest/api/3/universal_avatar/view/type
         /project/avatar/10401?size=small",
6        "32x32": "https://ahersushant.atlassian.net
         /rest/api/3/universal_avatar/view/type
         /project/avatar/10401?size=medium",
7        "48x48": "https://ahersushant.atlassian.net
         /rest/api/3/universal_avatar/view/type
         /project/avatar/10401"
8      },
9      "entityId": "496776ab-9334-4e03-bf41
       -5aeb68406fc1",
10     "expand": "description,lead,issueTypes,url
       ,projectKeys,permissions,insight",
11     "id": "10001",
```

Create create_jira.py



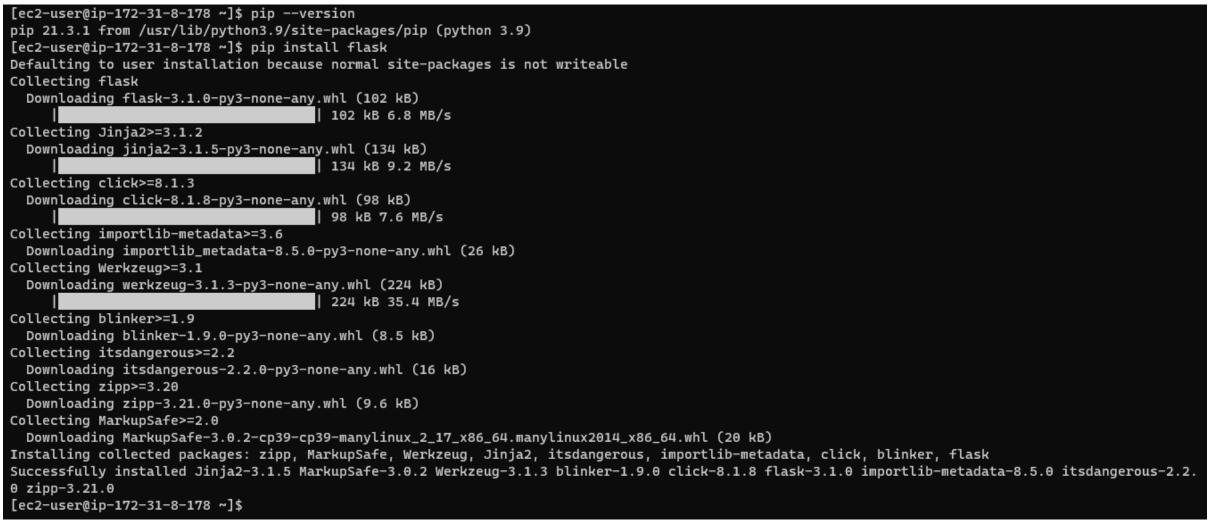
```python

import requests
from requests.auth import HTTPBasicAuth
import json

url = "https://ahersushant.atlassian.net/rest/api/3/issue"
API_TOKEN= "ATATT3xFfGF0ZcZ4aC7QguyxpwTMr_7U4THEwBSQ2Som-V1tumWtAPf2bhYq-BOPa4X9Wri-5-eZQyJS8ROBRiJ0pxo7CZcb4cisJaTf4N9IPcHSrCiwZ-D10CkvzkKjy
auth = HTTPBasicAuth("ahersushant222@gmail.com", API_TOKEN)

headers = {
  "Accept": "application/json",
  "Content-Type": "application/json"
}

payload = json.dumps( {
  "fields": {
    "description": {
      "content": [
        {
          "content": [
            {
              "text": "My first jira ticket",
              "type": "text"
            }
          ],
          "type": "paragraph"
        }
      ],
      "type": "doc",
      "version": 1
    },

    "issuetype": {
      "name": "Story"
    },

    "project": {
      "key": "PJ"
    },
    "summary": "First jira ticket",
  },
  "update": {}
```

Test.



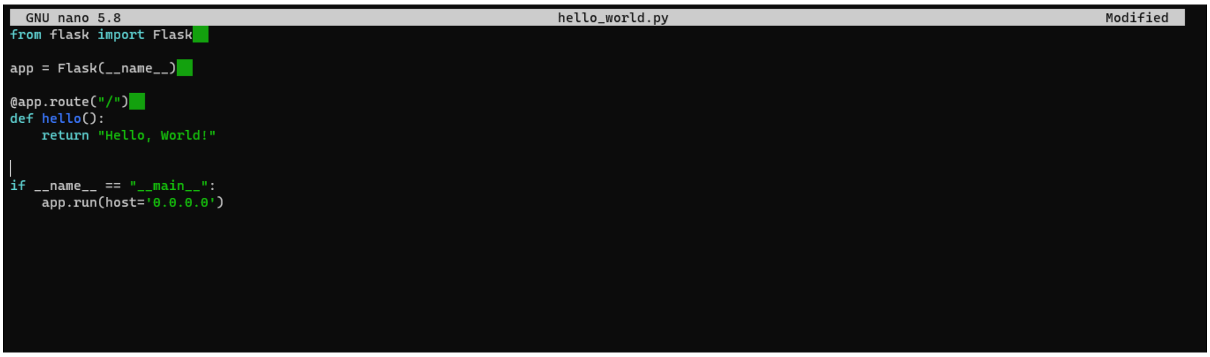
```
    }
}
PS D:\FCT\python for devops\day 14> python create-jira.py
{
{
    "id": "10003",
    "key": "PJ-4",
    "id": "10003",
    "key": "PJ-4",
    "key": "PJ-4",
    "self": "https://ahersushant.atlassian.net/rest/api/3/issue/10003"
    "self": "https://ahersushant.atlassian.net/rest/api/3/issue/10003"
}
PS D:\FCT\python for devops\day 14> 
```

Installing pip to ec2_instance



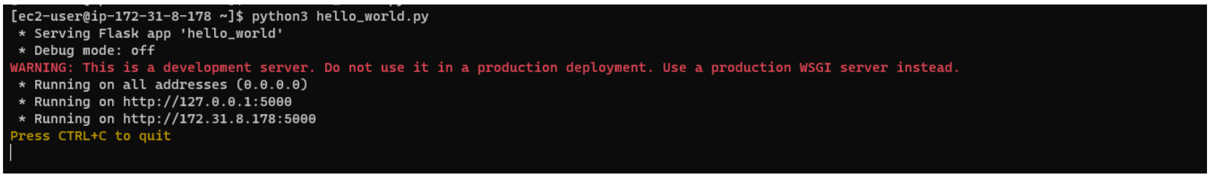
```
[ec2-user@ip-172-31-8-178 ~]$ pip --version
pip 21.3.1 from /usr/lib/python3.9/site-packages/pip (python 3.9)
[ec2-user@ip-172-31-8-178 ~]$ pip install flask
Defaulting to user installation because normal site-packages is not writeable
Collecting flask
  Downloading flask-3.1.0-py3-none-any.whl (102 kB)
     |████████████████████████████████| 102 kB 6.8 MB/s
Collecting Jinja2>=3.1.2
  Downloading jinja2-3.1.5-py3-none-any.whl (134 kB)
     |████████████████████████████████| 134 kB 9.2 MB/s
Collecting click>=8.1.3
  Downloading click-8.1.8-py3-none-any.whl (98 kB)
     |████████████████████████████████| 98 kB 7.6 MB/s
Collecting importlib-metadata>=3.6
  Downloading importlib_metadata-8.5.0-py3-none-any.whl (26 kB)
Collecting Werkzeug>=3.1
  Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
     |████████████████████████████████| 224 kB 35.4 MB/s
Collecting blinker>=1.9
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Collecting itsdangerous>=2.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting zipp>=3.20
  Downloading zipp-3.21.0-py3-none-any.whl (9.6 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-3.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (20 kB)
Installing collected packages: zipp, MarkupSafe, Werkzeug, Jinja2, itsdangerous, importlib-metadata, click, blinker, flask
Successfully installed Jinja2-3.1.5 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 flask-3.1.0 importlib-metadata-8.5.0 itsdangerous-2.2.
0 zipp-3.21.0
[ec2-user@ip-172-31-8-178 ~]$
```

Create hello_world.py for testing.



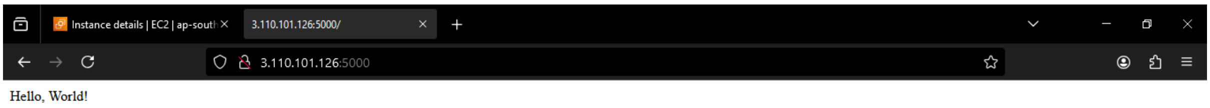
```
  GNU nano 5.8                          hello_world.py                          Modified
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, World!"


if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

Run python File.



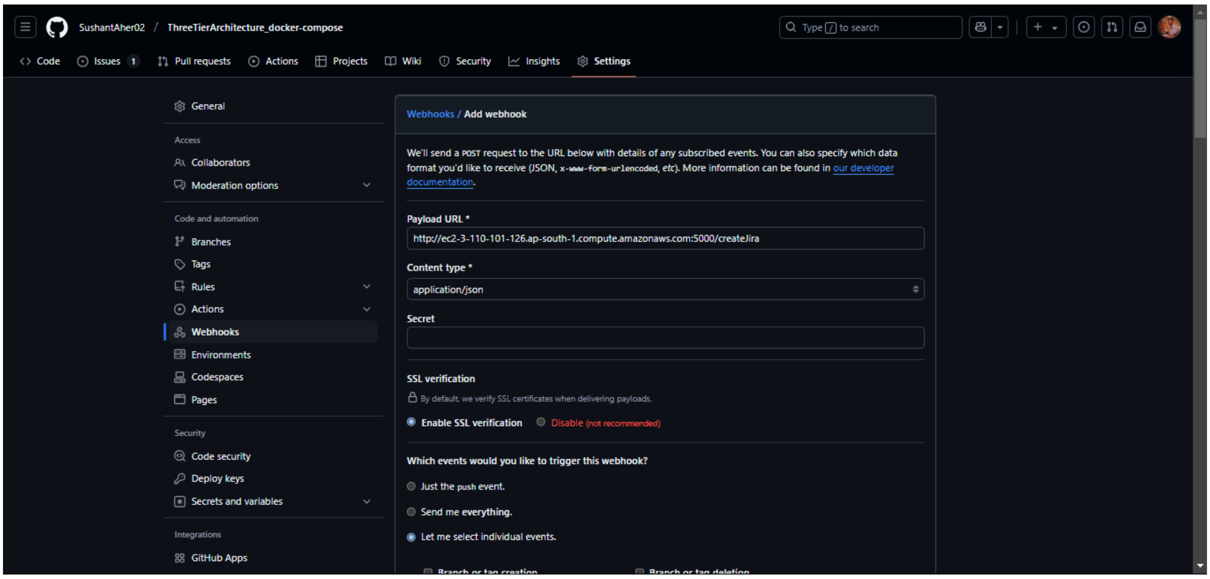
```
[ec2-user@ip-172-31-8-178 ~]$ python3 hello_world.py
 * Serving Flask app 'hello_world'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.8.178:5000
Press CTRL+C to quit
```

Test python file.



Hello, World!

Some Github Settings.



Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *
http://ec2-3-110-101-126.ap-south-1.compute.amazonaws.com:5000/createJira

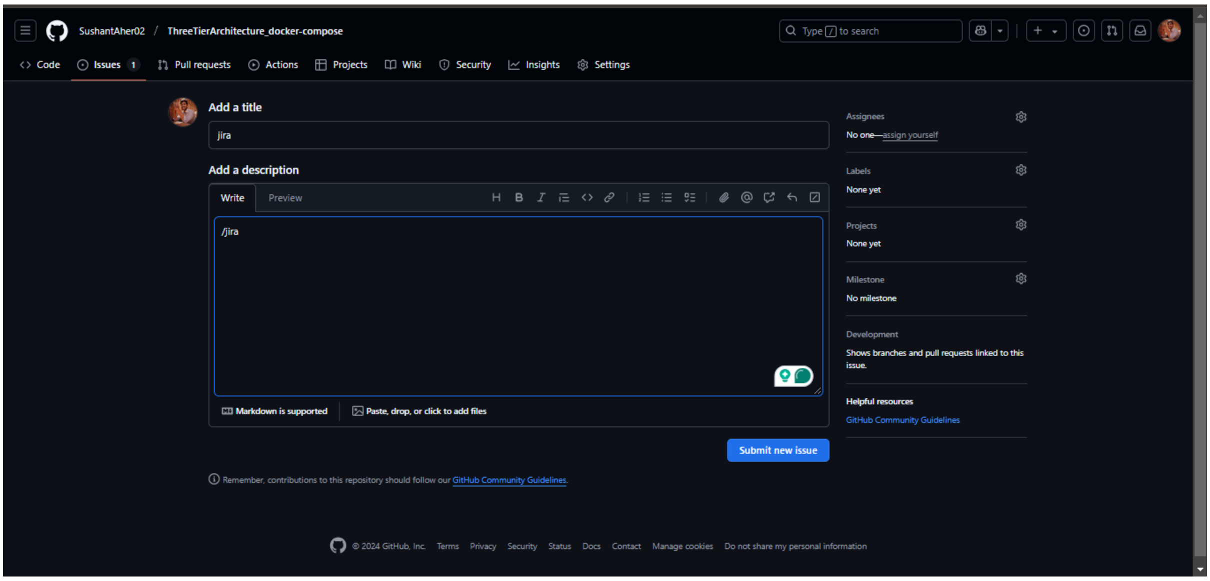
Content type *
application/json

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.
● Enable SSL verification ○ Disable (not recommended)

Which events would you like to trigger this webhook?
○ Just the push event.
○ Send me everything.
● Let me select individual events.

□ Branch or tag creation □ Branch or tag deletion


SushantAher02 / ThreeTierArchitecture_docker-compose
Type / to search
Code
Issues 1
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Write
Preview
Markdown is supported
Paste, drop, or click to add files
Submit new issue
© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Add a title
jira

Add a description
/jira

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

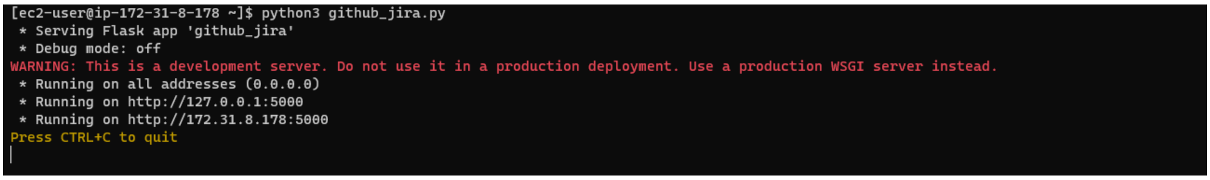
Milestone
No milestone

Development
Shows branches and pull requests linked to this issue.

Helpful resources
GitHub Community Guidelines

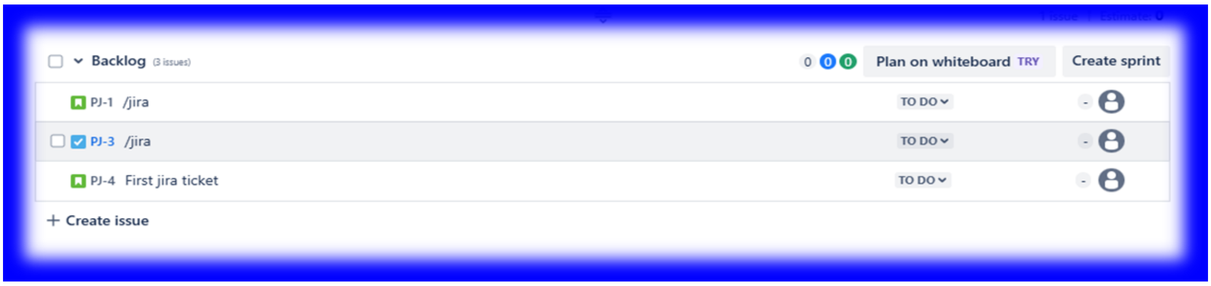
Remember, contributions to this repository should follow our GitHub Community Guidelines.

Run newly created github_jira.py file.



```
[ec2-user@ip-172-31-8-178 ~]$ python3 github_jira.py
 * Serving Flask app 'github_jira'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.8.178:5000
Press CTRL+C to quit
```

Automatically created issuetype in Jira.



□ Backlog (3 issues) 0 0 0 Plan on whiteboard TRY Create sprint

PJ-1	/jira	TO DO	-
☑ PJ-3	/jira	TO DO	-
PJ-4	First jira ticket	TO DO	-

+ Create issue

Conclusion :

The Automate Jira Ticket Creation from GitHub project highlights the power of automation in bridging development and project management workflows. By integrating GitHub and Jira, this solution streamlines the process of creating Jira tickets based on GitHub events, such as issue creation, pull requests, or code commits.

This automation eliminates manual effort, reduces errors, and ensures that all tasks and development activities are tracked seamlessly in Jira. With features like customizable ticket templates, event-driven triggers, and secure API interactions, the project enhances collaboration between development and management teams.

By leveraging tools like GitHub Actions or Python scripts with Jira APIs, this project demonstrates how to optimize workflows, save time, and maintain a centralized system for task management. It sets the stage for a more connected and efficient development lifecycle, enabling teams to focus on delivering value rather than administrative overhead.