RUJUTA MEDHI **D15A-28**

EXPERIMENT:01

Aim: Utilize AWS CodePipeline to deploy a Sample Application on an EC2 instance using AWS CodeDeploy.

Theory:

Amazon Elastic Compute Cloud (Amazon EC2) offers flexible and scalable computing power in the Amazon Web Services (AWS) Cloud. By using Amazon EC2, businesses can reduce the need for physical hardware, thereby lowering costs and speeding up the development and deployment of applications. EC2 provides the ability to launch as many virtual servers as necessary, enabling the configuration of security settings, networking, and storage management. This scalability allows you to easily increase capacity (scale up) to accommodate compute-intensive tasks, such as high-traffic periods, and reduce capacity (scale down) during periods of lower demand.

An EC2 instance represents a virtual server running in the AWS Cloud. When launching an EC2 instance, the specified instance type dictates the available hardware resources. Each instance type is designed to offer a unique balance of compute power, memory, network performance, and storage options. For more details, refer to the Amazon EC2 Instance Types Guide.

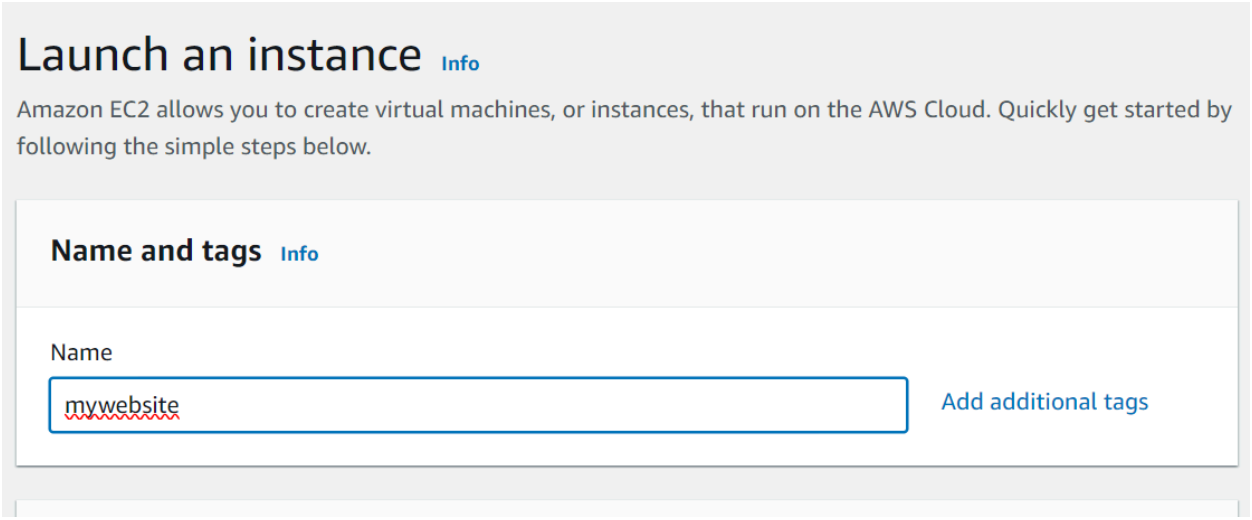
Key Features of Amazon EC2:

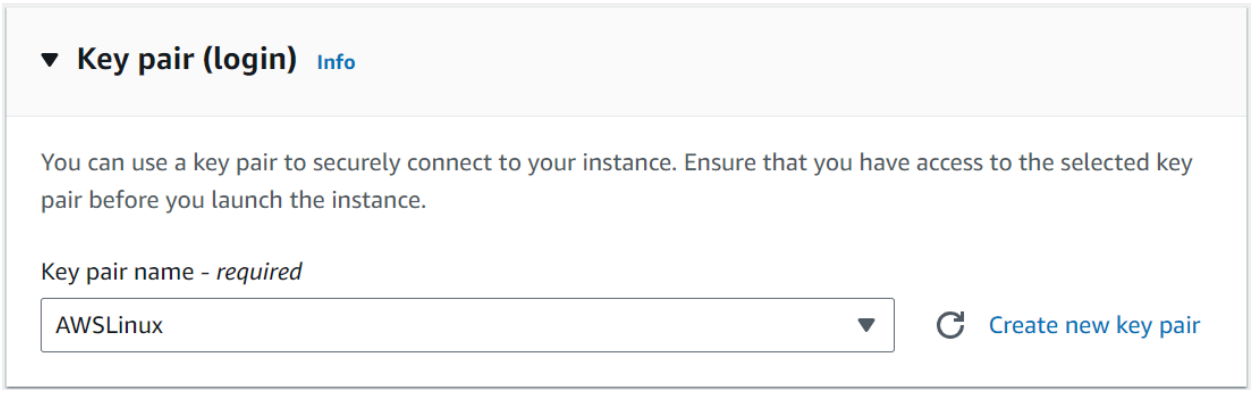
1. **Instances:**
   Virtual machines that run in the AWS Cloud.
2. **Amazon Machine Images (AMIs):**
   Prebuilt templates that define the necessary components for your instances, such as the operating system and additional software.
3. **Instance Types:**
   Different configurations of CPU, memory, storage, and networking capacity, allowing you to choose the optimal resources for your workload.
4. **Amazon EBS Volumes:**
   Persistent storage solutions using Amazon Elastic Block Store (Amazon EBS) that retain your data even when instances are stopped.
5. **Instance Store Volumes:**
   Temporary storage that is automatically deleted when an instance is stopped, hibernated, or terminated, suitable for ephemeral data.

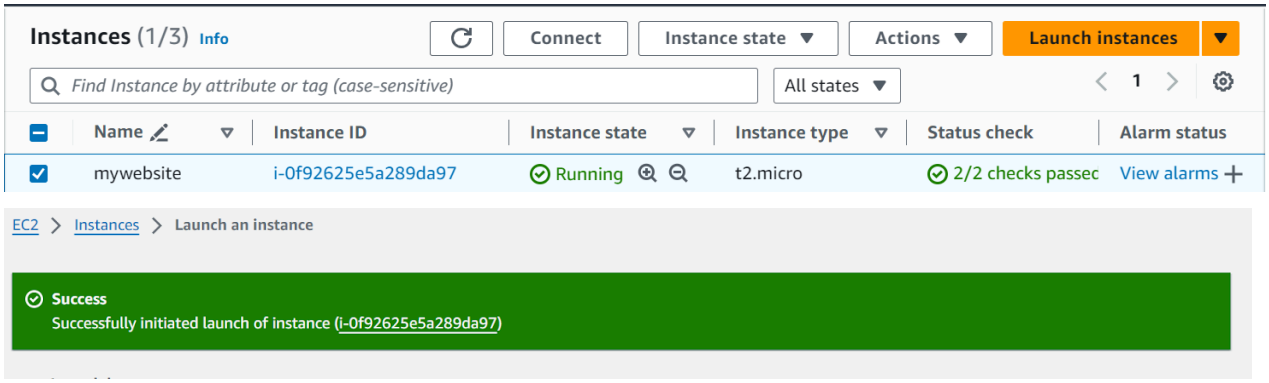
6. **Key Pairs:**
   A secure method for accessing your instances. AWS retains the public key, while you securely store the private key.
7. **Security Groups:**
   Virtual firewalls that enable you to control inbound and outbound traffic to your instances by specifying allowed protocols, ports, and IP address ranges.

Amazon EC2 also supports the secure processing, storage, and transmission of credit card information for merchants and service providers. It complies with the Payment Card Industry Data Security Standard (PCI DSS) and has been validated as PCI DSS compliant. For further information on PCI DSS and how to obtain the AWS PCI Compliance Package, refer to the PCI DSS Level 1 documentation.

Implementation:







1. Hosting a static website using EC2 instance

```
Complete!
[root@ip-172-31-62-13 ~]# systemctl status httpd
o httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-62-13 ~]# mkdir asw_assg3
[root@ip-172-31-62-13 ~]# cd aws_assg3
-bash: cd: aws_assg3: No such file or directory
[root@ip-172-31-62-13 ~]# cd asw_assg3
[root@ip-172-31-62-13 asw_assg3]# wget https://github.com/rujutamedhi22/infosys_html.git                https://github.com/rujutamedhi22/
infosys_html.git
--2024-08-17 15:29:04--  https://github.com/rujutamedhi22/infosys_html.git
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/rujutamedhi22/infosys_html [following]
--2024-08-17 15:29:04--  https://github.com/rujutamedhi22/infosys_html
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'infosys_html.git'

infosys_html.git                        [<=>                                                                    ]       0  --.-KB/s    infosys_
html.git                           [ <=>                                                                ] 266.79K  --.-KB/s    in 0.02s
```

```
2024-08-17 15:29:04 (15.9 MB/s) - 'infosys_html.git' saved [273193]

[root@ip-172-31-62-13 asw_assg3]# ls -lrt
total 268
-rw-r--r--. 1 root root 273193 Aug 17 15:29 infosys_html.git
[root@ip-172-31-62-13 asw_assg3]# wget https://github.com/rujutamedhi22/infosys_html/archive/refs/headshttps://github.com/rujutamedhi22/
infosys_html/archive/refs/heads/master.zip
--2024-08-17 15:35:07--  https://github.com/rujutamedhi22/infosys_html/archive/refs/heads/master.zip
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/rujutamedhi22/infosys_html/zip/refs/heads/master [following]
--2024-08-17 15:35:07--  https://codeload.github.com/rujutamedhi22/infosys_html/zip/refs/heads/master
Resolving codeload.github.com (codeload.github.com)... 140.82.112.10
Connecting to codeload.github.com (codeload.github.com)|140.82.112.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'master.zip'

master.zip                              [<=>                                                                    ]       0  --.-KB/s    master.z
ip                                 [ <=>                                                                ] 233.35K  --.-KB/s    in 0.01s

2024-08-17 15:35:07 (18.4 MB/s) - 'master.zip' saved [238946]

[root@ip-172-31-62-13 asw_assg3]# ls -lrt
```

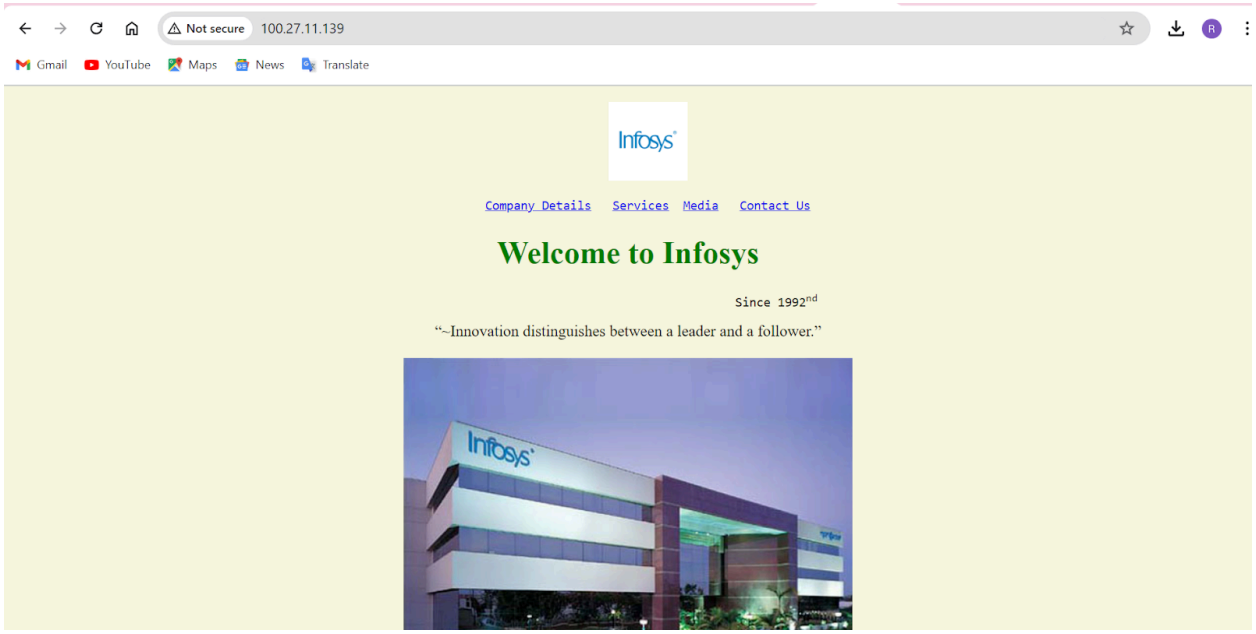
```
[root@ip-172-31-62-13 asw_assg3]# ls -lrt
total 504
-rw-r--r--. 1 root root 273193 Aug 17 15:29 infosys_html.git
-rw-r--r--. 1 root root 238946 Aug 17 15:35 master.zip
[root@ip-172-31-62-13 asw_assg3]# unzip master.zip
Archive:  master.zip
703660532cf867469a131d058181d873f4603d49
   creating: infosys_html-master/
  inflating: infosys_html-master/Infosys.jpg
  inflating: infosys_html-master/consultant.jpeg
  inflating: infosys_html-master/digital.jpeg
  inflating: infosys_html-master/facebook.png
  inflating: infosys_html-master/index.html
  inflating: infosys_html-master/infosys-logo.jpg
  inflating: infosys_html-master/twitter.png
  inflating: infosys_html-master/world-wide-signal.png
[root@ip-172-31-62-13 asw_assg3]# ls -lrt
total 504
drwxr-xr-x. 2 root root    178 Aug  4 05:08 infosys_html-master
-rw-r--r--. 1 root root 273193 Aug 17 15:29 infosys_html.git
-rw-r--r--. 1 root root 238946 Aug 17 15:35 master.zip
[root@ip-172-31-62-13 asw_assg3]# cd ^C
[root@ip-172-31-62-13 asw_assg3]# cd https://github.com/rujutamedhi22/infosys_html/archive/refs/headshttps://github.com/rujutame
wget https://github.com/rujutamedhi22/infosys_html/archive/refs/heads/master.zip                  ls -lrt^C
[root@ip-172-31-62-13 asw_assg3]# cd infosys_html-master
[root@ip-172-31-62-13 infosys_html-master]# ls -lrt
```

```
-rw-r--r--. 1 root root   4840 Aug  4 05:08 index.html
-rw-r--r--. 1 root root  15105 Aug  4 05:08 facebook.png
-rw-r--r--. 1 root root   8443 Aug  4 05:08 digital.jpeg
-rw-r--r--. 1 root root 103234 Aug  4 05:08 consultant.jpeg
-rw-r--r--. 1 root root  38350 Aug  4 05:08 Infosys.jpg
[root@ip-172-31-62-13 infosys_html-master]# mv * /var/www/html/
[root@ip-172-31-62-13 infosys_html-master]# cd /var/www/html
[root@ip-172-31-62-13 html]# ls -lrt
total 284
-rw-r--r--. 1 root root  10975 Aug  4 05:08 world-wide-signal.png
-rw-r--r--. 1 root root  23628 Aug  4 05:08 twitter.png
-rw-r--r--. 1 root root  67699 Aug  4 05:08 infosys-logo.jpg
-rw-r--r--. 1 root root   4840 Aug  4 05:08 index.html
-rw-r--r--. 1 root root  15105 Aug  4 05:08 facebook.png
-rw-r--r--. 1 root root   8443 Aug  4 05:08 digital.jpeg
-rw-r--r--. 1 root root 103234 Aug  4 05:08 consultant.jpeg
-rw-r--r--. 1 root root  38350 Aug  4 05:08 Infosys.jpg
[root@ip-172-31-62-13 html]# systemctl status httpd
o httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-62-13 html]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-62-13 html]# systemctl start httpd
[root@ip-172-31-62-13 html]#
```

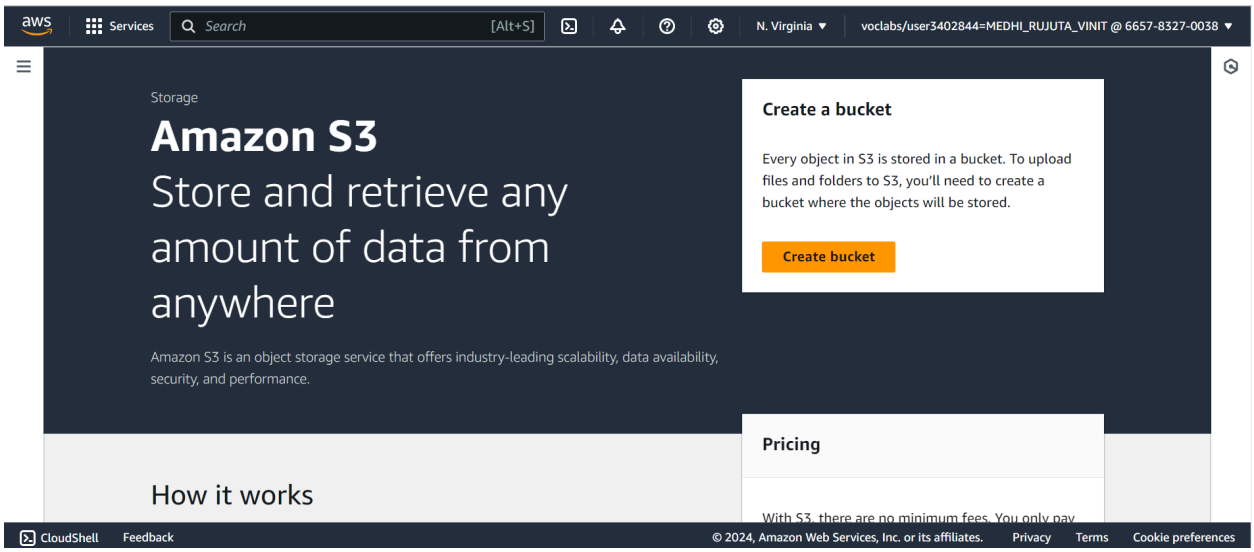
i-0f92625e5a289da97 (mywebsite)

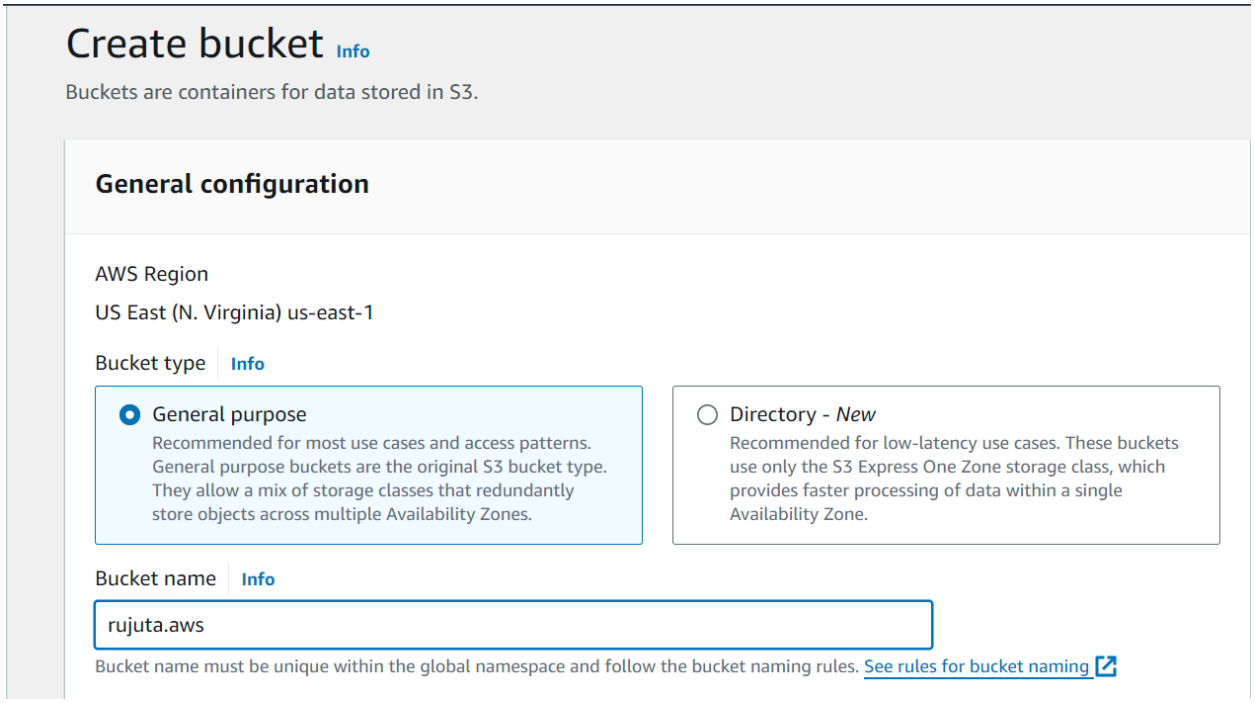
Auto-assigned IP address

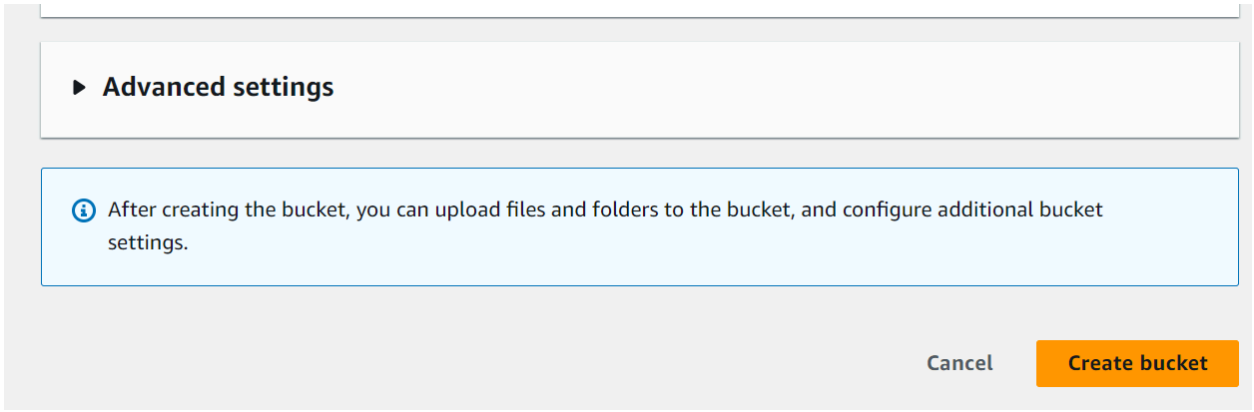
100.27.11.139 [Public IP]

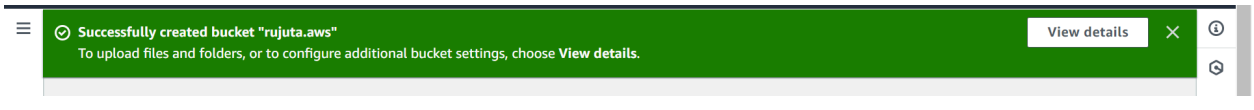


2. Hosting using S3 bucket

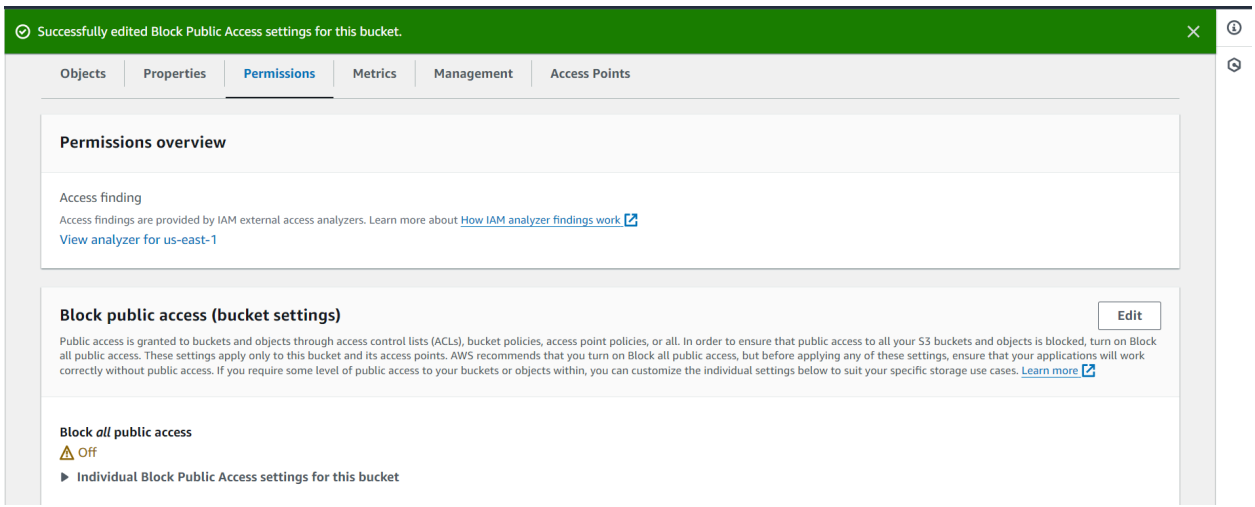


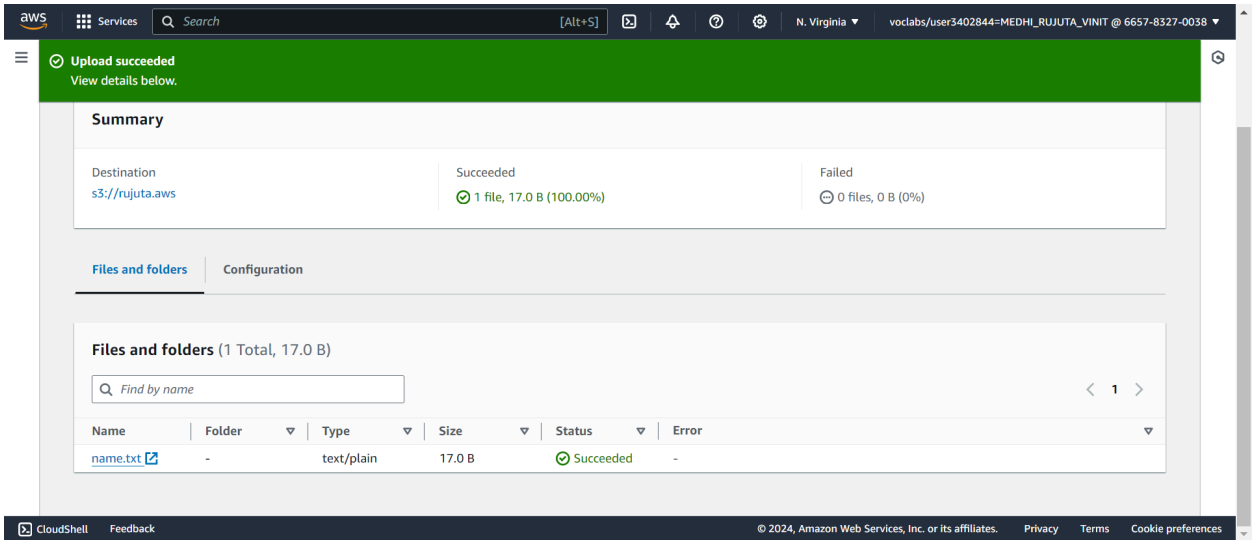


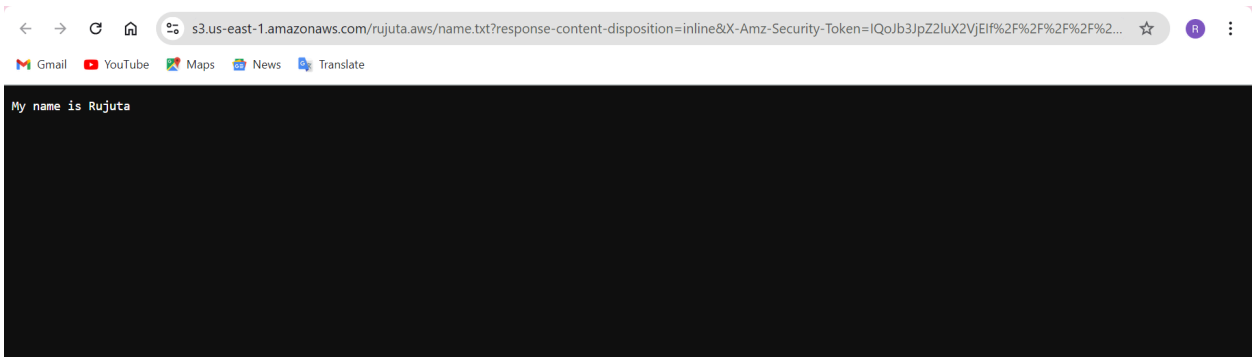

Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket


Successfully created bucket "rujuta.aws"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details

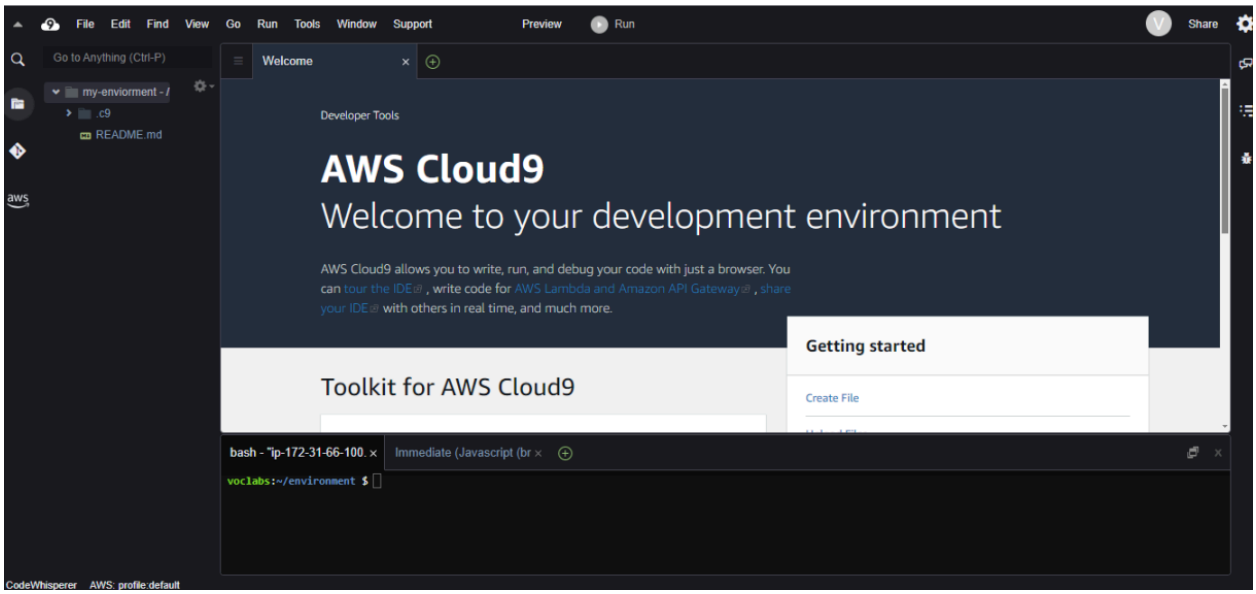

General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
IAM Access Analyzer
Creation date
rujuta.aws
US East (N. Virginia) us-east-1
View analyzer for us-east-1
August 22, 2024, 20:09:18 (UTC+05:30)

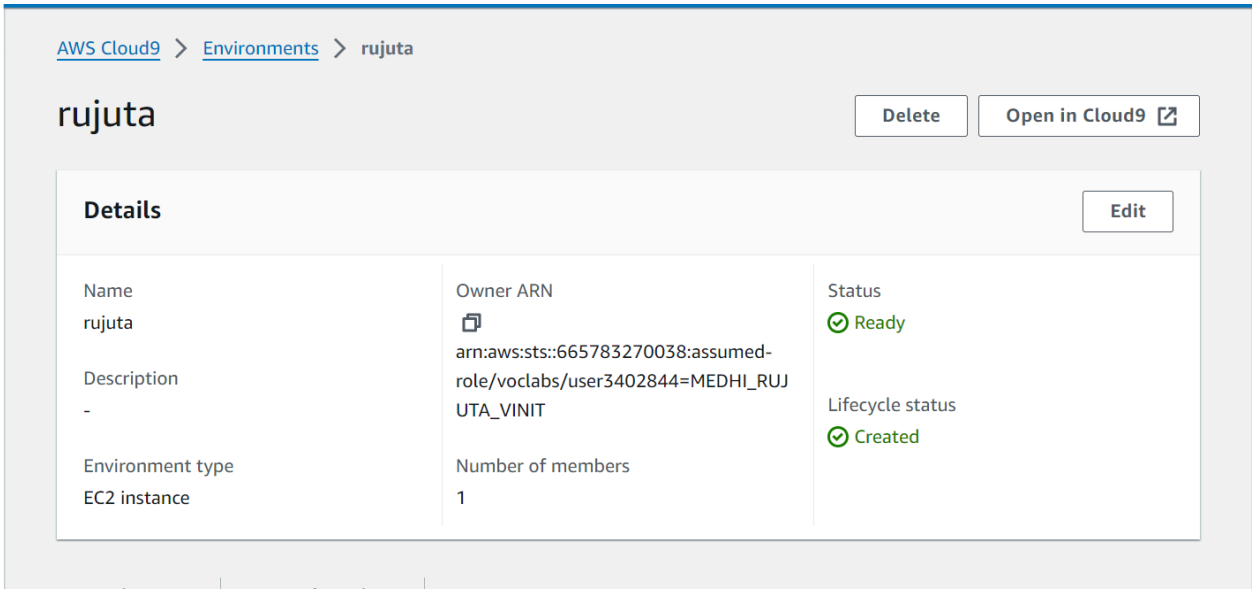

Successfully edited Block Public Access settings for this bucket.
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for us-east-1
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Off
Individual Block Public Access settings for this bucket

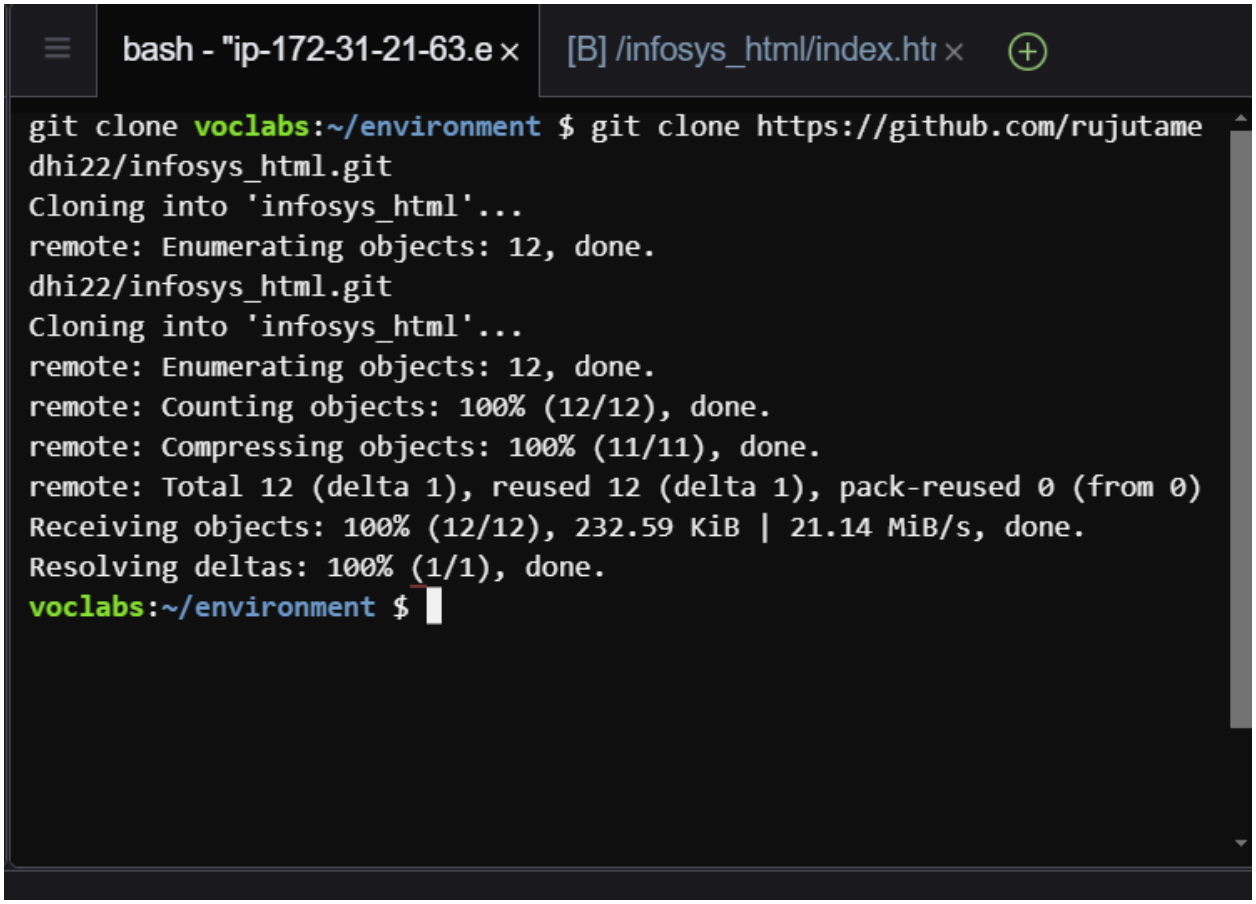

aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402844=MEDHI_RUJUTA_VINIT @ 6657-8327-0038
Upload succeeded
View details below.
Summary
Destination
s3://rujuta.aws
Succeeded
1 file, 17.0 B (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 17.0 B)
Find by name
1
Name
Folder
Type
Size
Status
Error
name.txt
-
text/plain
17.0 B
Succeeded
-
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

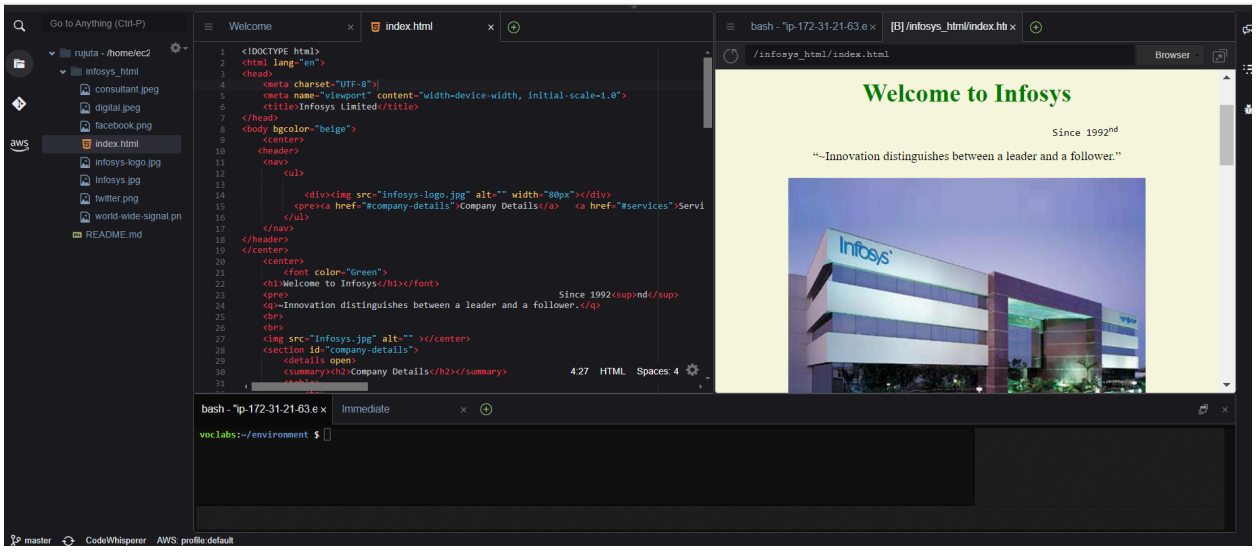

s3.us-east-1.amazonaws.com/rujuta.aws/name.txt?response-content-disposition=inline&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEIf%2F%2F%2F%2F%2...
Gmail
YouTube
Maps
News
Translate
My name is Rujuta

3. Hosting using Cloud 9






bash - "ip-172-31-21-63.e
[B] /infosys_html/index.ht
git clone voclabs:~/environment $ git clone https://github.com/rujutame
dhi22/infosys_html.git
Cloning into 'infosys_html'...
remote: Enumerating objects: 12, done.
dhi22/infosys_html.git
Cloning into 'infosys_html'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 1), reused 12 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 232.59 KiB | 21.14 MiB/s, done.
Resolving deltas: 100% (1/1), done.
voclabs:~/environment $


Welcome
index.html
bash - "ip-172-31-21-63.e
[B] /infosys_html/index.ht
/infosys_html/index.html
Browser
Welcome to Infosys
Since 1992nd
"~Innovation distinguishes between a leader and a follower."
Infosys
Immediate
voclabs:~/environment $

Specify user details

User details

User name

Rujuta

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

- Add user to group
  Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
  Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly
  Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Get started with groups

Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn

Create group

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more

User group name

Enter a meaningful name to identify this group.

group

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

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more

User group name

Enter a meaningful name to identify this group.

group

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

Permissions policies (950)

Create policy

Filter by Type

Search | All ty... | 1 2 3 4 5 6 7 ... 48

Policy name	Type	Use...	Description
AdministratorAccess	AWS managed ...	None	Provides full access to AWS service

Cancel | Create user group

User groups (1)

Create group

Search | 1

Group name	Users	Attached policies	Created
group	0	-	2024-08-22 (1...

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

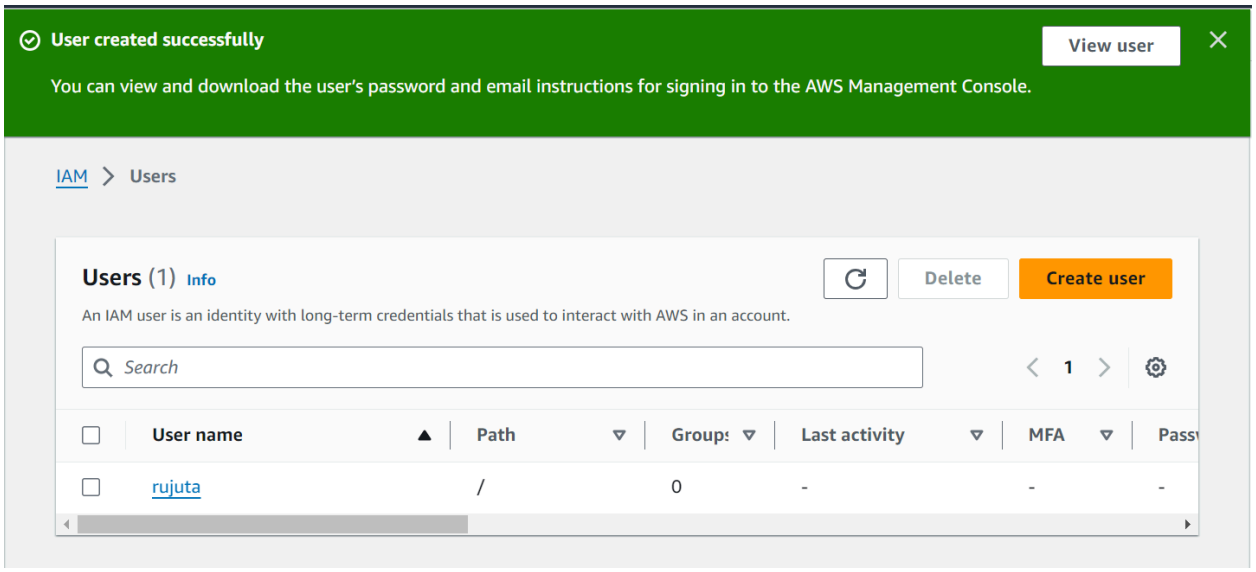
User details

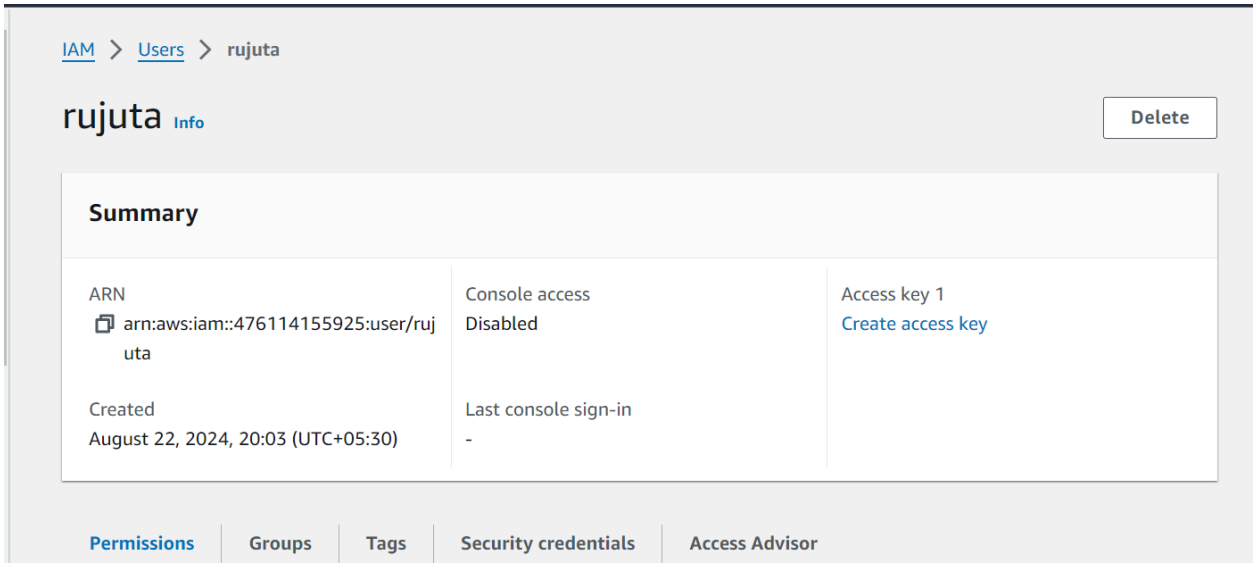
User name	Console password type	Require password reset
rujuta	None	No

Permissions summary

1

Name	Type	Used as


User created successfully
View user
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
IAM > Users
Users (1) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Groups
Last activity
MFA
Pass
rujuta
/
0
-
-
-


IAM > Users > rujuta
rujuta Info
Delete
Summary
ARN
arn:aws:iam::476114155925:user/rujuta
Console access
Disabled
Access key 1
Create access key
Created
August 22, 2024, 20:03 (UTC+05:30)
Last console sign-in
-
Permissions
Groups
Tags
Security credentials
Access Advisor