

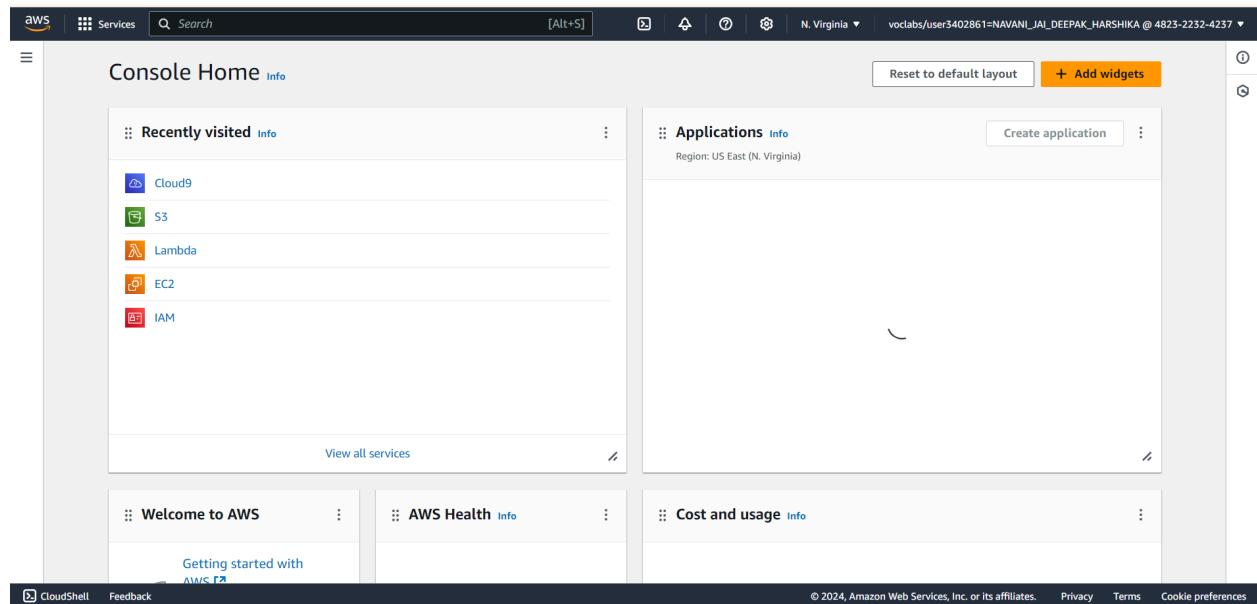
Name:Jai Deepak Navani
Rollno:31
Advance Deveops

Study case(11):

Simple Cloud-Based Application Deployment

- Concepts Used: AWS Cloud9, S3, and EC2.
- Problem Statement: "Develop a simple HTML page using AWS Cloud9 and deploy it to an S3 bucket for static website hosting. Then, set up an EC2 instance to serve as a backup server for the website."
- Tasks:
 - Create a basic HTML page using AWS Cloud9.
 - Deploy the HTML page to an S3 bucket and enable static website hosting.
 - Launch an EC2 instance and configure it to serve the same HTML page as a backup.

Sign-in to the console:



Search cloud9 and create environment:

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237

Developer Tools

AWS Cloud9

A cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

New AWS Cloud9 environment

Create environment

How it works

Create an AWS Cloud9 development environment on a new Amazon EC2 instance or connect it to your own Linux server through SSH. Once you've created an AWS Cloud9 environment, you will have immediate access to a rich code editor, integrated debugger, and built-in terminal with pre-configured AWS CLI – all within your browser.

Using the AWS Cloud9 dashboard, you can create and switch between many different AWS Cloud9 environments, each one containing the custom tools, runtimes, and files for a specific project.

Getting started

[Before you start](#) (2 min read)

[Create an environment](#) (2 min read)

[Working with environments](#) (15 min read)

[Working with the IDE](#) (10 min read)

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237

Network settings [Info](#)

Connection

How your environment is accessed.

☐ AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).

☒ Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.

[VPC settings](#) [Info](#)

[Tags - optional](#) [Info](#)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

i

The following IAM resources will be created in your account

- AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

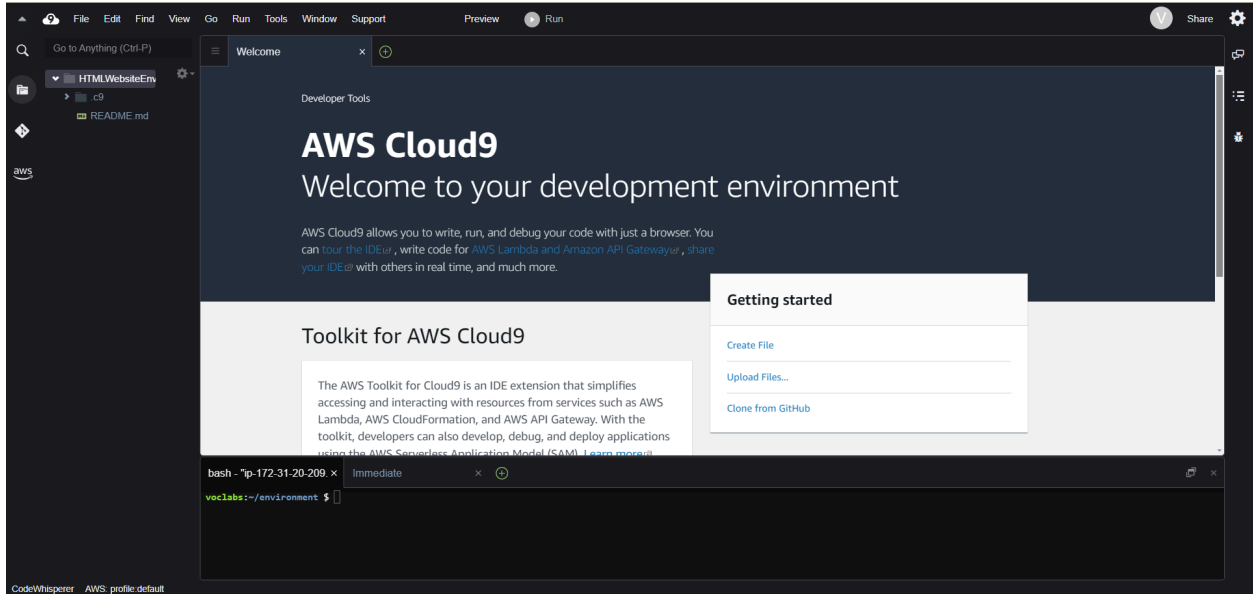
Cancel

Create

CloudShell

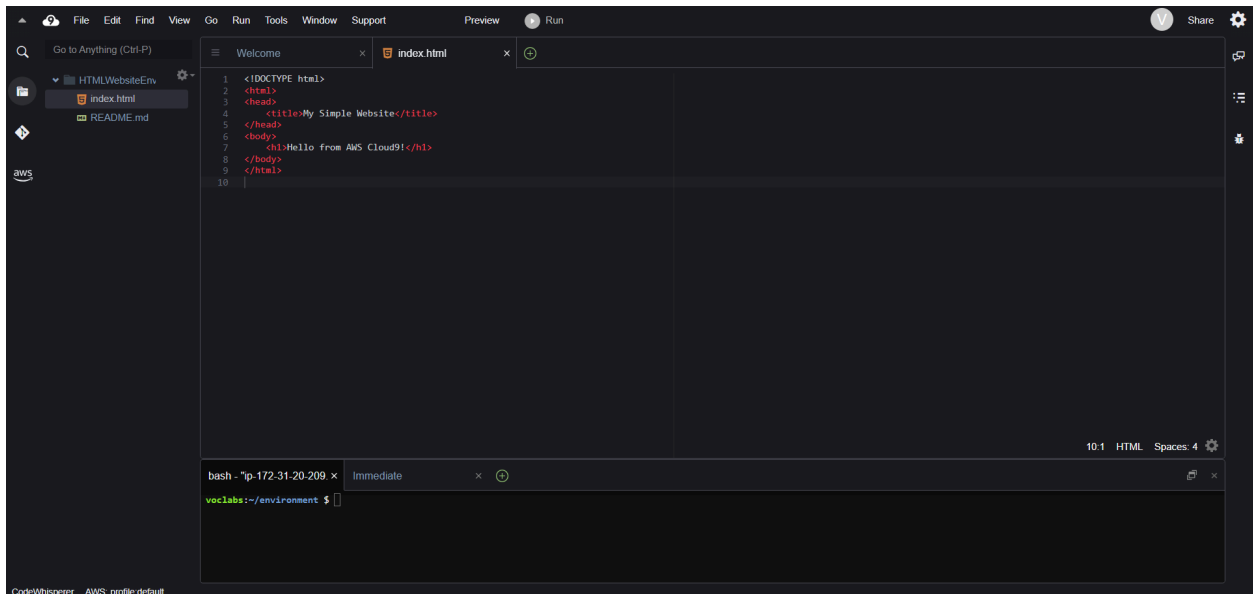
Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

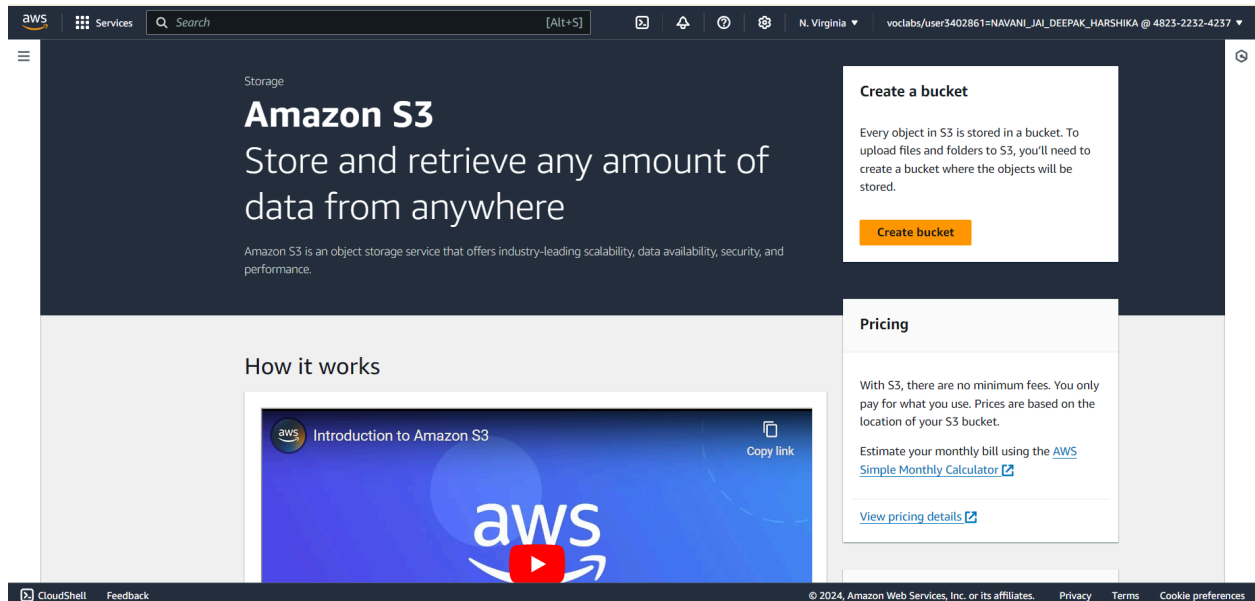


Make index.html file and save the code inside the file:

```
<!DOCTYPE html>
<html>
<head>
  <title>My Simple Website</title>
</head>
<body>
  <h1>Hello from AWS Cloud9!</h1>
</body>
</html>
```



Now create s3 bucket:



Now after creating the bucket upload the cloud9 file into the s3 bucket using simple commands:
aws --version

- aws --version
- aws s3 cp index.html s3://my-website-bucke/

```
bash - "ip-172-31-20-209. x Immediate x (+)
voclabs:~/environment $ aws --version
aws-cli/2.18.1 Python/3.12.6 Linux/6.1.109-118.189.amzn2023.x86_64 exe/x86_64.amzn.2023
voclabs:~/environment $ aws s3 cp index.html s3://my-website-bucke/
upload: ./index.html to s3://my-website-bucke/index.html
voclabs:~/environment $
```

Check the uploaded file in s3 bucket:

Amazon S3 > Buckets > my-website-bucke

my-website-bucke [Info](#)

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (1) [Info](#)

[Refresh](#)
[Copy S3 URI](#)
[Copy URL](#)
[Download](#)
[Open](#)
[Delete](#)
[Actions](#)
[Create folder](#)
[Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	index.html	html	October 14, 2024, 19:04:03 (UTC+05:30)	134.0 B	Standard

Now go the properties of the bucket and edit static website enable from the disable:

Static website hosting [Edit](#)

Use this bucket to host a website or redirect requests. [Learn more](#)

S3 static website hosting
Disabled

Static website hosting [Edit](#)

Use this bucket to host a website or redirect requests. [Learn more](#)

S3 static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://my-website-bucke.s3-website-us-east-1.amazonaws.com>

Go to the permission and write the bucket policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
```

```


    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::my-website-bucke/*"
  }
]
}

```

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by

Bucket ARN

 arn:aws:s3:::my-website-bucke

Policy

```

1 ▼ {
2   "Version": "2012-10-17",
3 ▼  "Statement": [
4 ▼    {
5      "Sid": "PublicReadGetObject",
6      "Effect": "Allow",
7      "Principal": "*",
8      "Action": "s3:GetObject",
9      "Resource": "arn:aws:s3:::my-website-bucke/*"
10   }
11 ]
12 }

```

Now go to the properties, then to the static hosting and see the URL and check whether the HTML file is hosted or not:

Hello from AWS Cloud9!

Now launch the instance:

aws

Services

Search

[Alt+S]

📧 🔔 ⚙

N. Virginia

voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237

EC2 Dashboard

×

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Resources

EC2 Global View

⚙

🔄

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

Instances (running)

Auto Scaling Groups

0

Capacity Reservations

Dedicated Hosts

Elastic IPs

Instances

Key pairs

Load balancers

Placement groups

Security groups

Snapshots

Volumes

Launch instance

To get started, launch an Amazon EC2 Instance, which is a virtual server in the cloud.

Launch instance

▼

Migrate a server

🔗

Note: Your instances will launch in the US East (N. Virginia) Region

Service health

AWS Health Dashboard

🔗

🔄

Region

US East (N. Virginia)

Status

🟢 This service is operating normally.

Zones

Account attributes

🔄

⌵

Settings

Data protection and security

Zones

EC2 Serial Console

Default credit specification

EC2 console preferences

Explore AWS

×

Amazon GuardDuty Malware Protection

GuardDuty now provides agentless malware detection in Amazon EC2 & EC2 container workloads. [Learn more](#)

Enable Best Price-Performance with AWS Graviton2

AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. [Learn more](#)

CloudShell

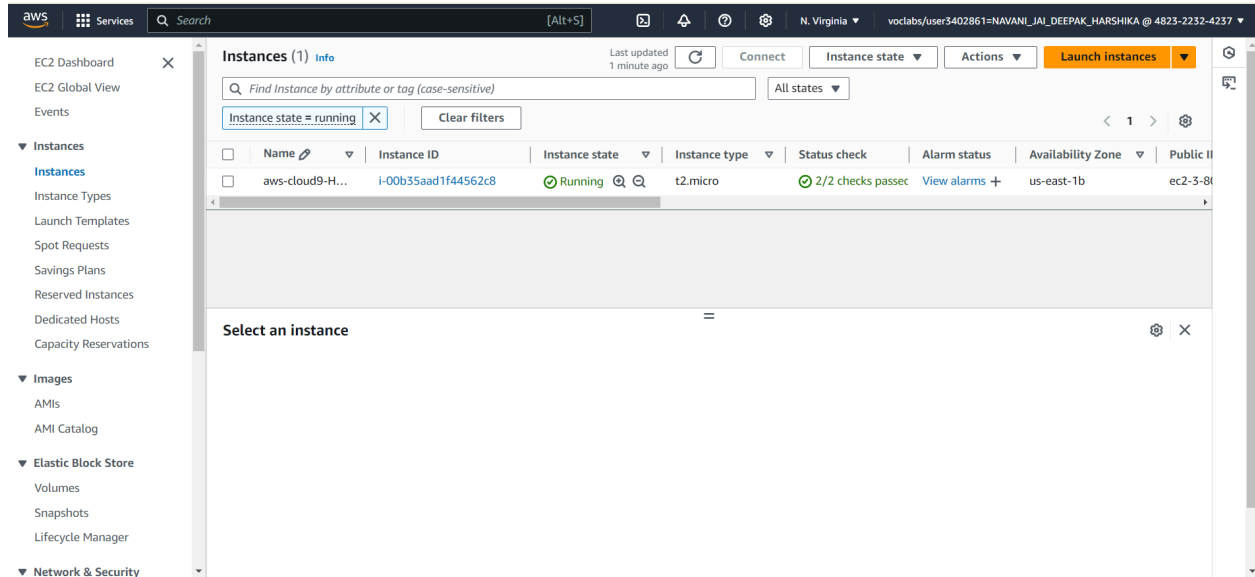
Feedback

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

Privacy

Terms

Cookie preferences



Now connect and add this commands:

`sudo yum update -y`

```
[ec2-user@ip-172-31-44-199 ~]$ sudo yum update -y
Last metadata expiration check: 0:01:04 ago on Mon Oct 14 13:53:59 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

`sudo yum install httpd -y`

```
[ec2-user@ip-172-31-44-199 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:01:44 ago on Mon Oct 14 13:53:59 2024.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
-----
Installing:
httpd                                   x86_64            2.4.62-1.amzn2023  amazonlinux         48 k
Installing dependencies:
apr                                     x86_64            1.7.2-2.amzn2023.0.2  amazonlinux         129 k
apr-util                               x86_64            1.6.3-1.amzn2023.0.1  amazonlinux         98 k
generic-logos-httpd                   x86_64            18.0.0-12.amzn2023.0.3  amazonlinux         19 k
httpd-core                             x86_64            2.4.62-1.amzn2023    amazonlinux         1.4 M
httpd-filesystem                       noarch            2.4.62-1.amzn2023    amazonlinux         14 k
httpd-tools                            x86_64            2.4.62-1.amzn2023    amazonlinux         81 k
libbrotli                               x86_64            1.0.9-4.amzn2023.0.2  amazonlinux         315 k
mailcap                                noarch            2.1.49-3.amzn2023.0.3  amazonlinux         33 k
Installing weak dependencies:
apr-util-openssl                       x86_64            1.6.3-1.amzn2023.0.1  amazonlinux         17 k
Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.62-1.amzn2023.x86_64  httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch  httpd-tools-2.4.62-1.amzn2023.x86_64  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.27-1.amzn2023.0.3.x86_64  mod_lua-2.4.62-1.amzn2023.x86_64
Complete!
```

`sudo systemctl start httpd`

```
[ec2-user@ip-172-31-44-199 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-44-199 ~]$
```



```
[ec2-user@ip-172-31-44-199 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

Now run this command in the cloud9 to connect the server to the ec2 instance:

```
ssh -i wind-key.pem ec2-user@54.89.150.198
```

The screenshot displays the AWS CloudShell interface. On the left, a file explorer shows the directory structure: `HTMLWebsiteEm`, `index.html`, `README.md`, and `wind-key.pem`. The main area is split into two panes. The top pane shows the contents of `index.html`, which is a simple HTML document with a title "My Simple Website" and a body containing "Hello from AWS Cloud9!". The bottom pane is a terminal window titled `ec2-user@ip-172-31-39-1x Immediate`. It shows the following commands and their outputs:

```
Connection closed
voclabs:~/environment $ chmod 400 wind-key.pem
chmod: cannot access 'wind-key.pem': No such file or directory
voclabs:~/environment $ ls
README.md index.html
voclabs:~/environment $ find -/ -name "wind-key.pem"
mv: cannot stat '/path/to/wind-key.pem ~/environment/': No such file or directory
voclabs:~/environment $ chmod 400 wind-key.pem
voclabs:~/environment $ scp -i wind-key.pem index.html ec2-user@54.89.150.198:/var/www/html/
dest open("/var/www/html/index.html"): Permission denied
failed to upload file index.html to /var/www/html/index.html
voclabs:~/environment $ ssh -i wind-key.pem ec2-user@54.89.150.198
```

The terminal output then shows the SSH connection details for Amazon Linux 2023, including the public key fingerprint and the URL `https://aws.amazon.com/linux/amazon-linux-2023`. At the bottom, it indicates the last login time: `Last login: Mon Oct 14 14:06:49 2024 from 18.206.107.29 [ec2-user@ip-172-31-39-190 ~]$`.

Hello from AWS Cloud9!