

Lab2: Deploy Website in AWS EC2 using the Linux OS

Name: Akilandeshwari Srinivasan

Student ID: 451036

Course: CLCM3102

Description:

Host the website in AWS EC2 with Linux OS.

Preparation:

Open aws learners lab and launch the aws management console, created an EC2 instance. Then copy the public DNS link in SSH client after connecting EC2 service and paste that in putty's Host name field and add the downloaded keypair document to putty. Install and start apache, download git code and unzip the file using linux command.

1) Why Linux OS considered as best?

- ❖ Open source and highly customizable when compared to other OS.
- ❖ Linux OS is well known for its stability
- ❖ It is more secure than windows and Mac.

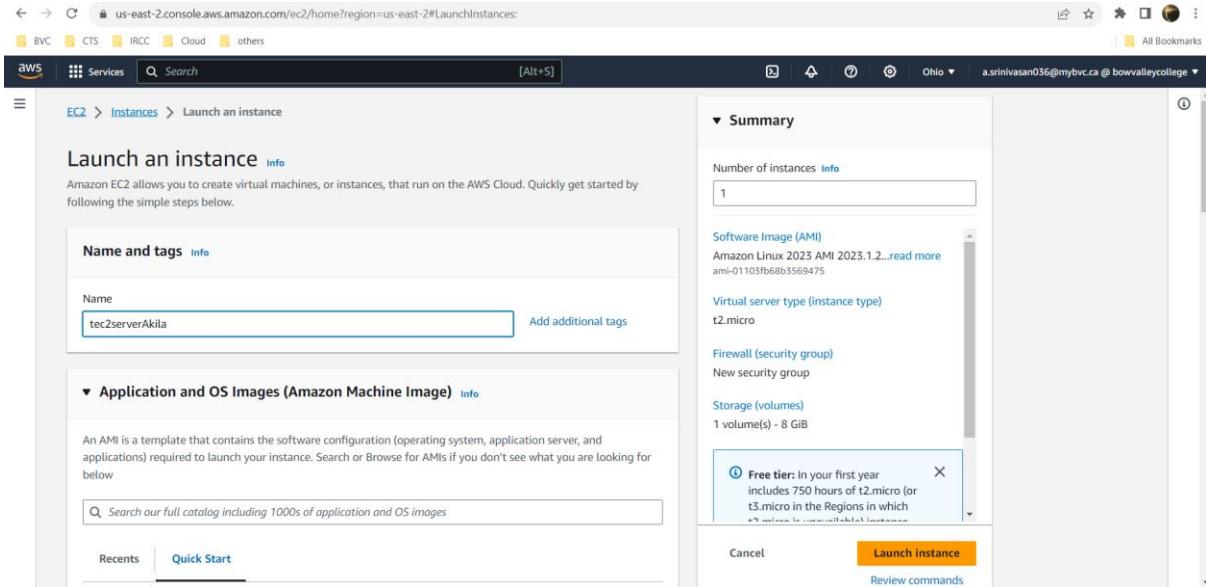
2) Advantages of hosting static website in AWS EC2 over AWS S3

AWS EC2 is more customizable when compared to AWS S3. Moreover AWS S3 is a storage bucket where

the computation have to done manually. On the other hand EC2 is flexible, can have more controls and allows you to run server in the cloud with minimal effort from our end.

Screenshots for Hosting website in AWS EC2:

- open aws management console and create a EC2 service.



- Have a unique name

The screenshot shows the AWS Lambda console interface. A new function named "HelloWorld" is being created. The "Code" tab is active, displaying the Lambda@Edge runtime and a sample code snippet. The "Test" tab shows a successful execution with the output "Hello World!". The "Overview" tab is also visible.

- Select the operating system which is required.

The screenshot shows the AWS Lambda console interface. A new function named "HelloWorld" is being created. The "Code" tab is active, displaying the Lambda@Edge runtime and a sample code snippet. The "Test" tab shows a successful execution with the output "Hello World!". The "Overview" tab is also visible.

- Select the instance type which may be nano, micro as per requirement.

The screenshot shows the AWS CloudShell interface. On the left, under 'Network settings', there's a section for creating a new security group. It includes fields for 'Create security group' (radio button selected) and 'Select existing security group'. Below these, it says 'We'll create a new security group called "launch-wizard-14" with the following rules:' followed by three checkboxes: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPS traffic from the internet (To set up an endpoint, for example when creating a web server)', and 'Allow HTTP traffic from the internet (To set up an endpoint, for example when creating a web server)'. A warning message at the bottom states: '⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' On the right, the 'Summary' section shows 'Number of instances: 1', 'Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type): t2.micro', 'Firewall (security group): New security group', and 'Storage (volumes): 1 volume(s) - 8 GiB'. A modal window titled 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you can't run t2.micro instances)' is open. At the bottom right are 'Cancel', 'Launch instance' (button), and 'Review commands'.

- Check all the checkboxes as we are allowing http, https
- Create a key pair and once you have created it will be downloaded to your local.
- This downloaded file will be used in putty.

The screenshot shows the AWS CloudShell interface. On the left, under 'Key pair name - required', there's a 'Select' dropdown. Below it, under 'Network settings', there's a section for creating a new security group. It includes fields for 'Create security group' (radio button selected) and 'Select existing security group'. Below these, it says 'We'll create a new security group called "Launch-wizard-14" with the following rules:' followed by three checkboxes: 'Allow SSH traffic from Anywhere (0.0.0.0/0)', 'Allow HTTPS traffic from the Internet (To set up an endpoint, for example when creating a web server)', and 'Allow HTTP traffic from the Internet (To set up an endpoint, for example when creating a web server)'. A warning message at the bottom states: '⚠️ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more' (with a link). On the right, the 'Create key pair' dialog is open. It has fields for 'Key pair name' (with placeholder 'Enter key pair name') and 'Key pair type' (with options 'RSA' (selected) and 'ED25519'). Below that, it has 'Private key file format' (with options '.pem' and '.ppk' (selected)). A note below says: 'When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more'. At the bottom right of the dialog are 'Cancel' and 'Create key pair' (button).

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:

BVC CTS IRCC Cloud others

aws Services Search [Alt+S]

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
newKeyValPairAkila

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type
 RSA RSA encrypted private and public key pair
 ED25519 ED25519 encrypted private and public key pair

Private key file format
 .pem For use with OpenSSH
 .ppk For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more

Cancel Create key pair

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:

BVC CTS IRCC Cloud others

aws Services Search [Alt+S]

Instances (1/12) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Macserver	i-069b03d49e645334c	Stopped	t2.micro	-	No alarms +	us-east-2a	-
mymacserver	i-026433fceb5f5de5	Stopped	t2.micro	-	No alarms +	us-east-2a	-
-	i-0dca90331be517e1d	Stopped	t2.micro	-	No alarms +	us-east-2c	-
AppDBServerB...	i-0fd975875abd38dd2	Stopped	t2.micro	-	No alarms +	us-east-2c	-
AppDBServerB...	i-09c97a29cc457c1df	Stopped	t2.micro	-	No alarms +	us-east-2c	-
dynamodbbuc...	i-022fd5348c300138c	Stopped	t2.micro	-	No alarms +	us-east-2c	-
GafanaEc2	i-0795f025467f50eb	Stopped	t2.micro	-	No alarms +	us-east-2c	-
tec2serverAkila	i-0fed6e7b7fce7e44	Stopped	t2.micro	-	No alarms +	us-east-2c	-
instance9966	i-0304a60488bbad117	Running	t2.micro	2/2 checks passed	No alarms +	us-east-2c	ec2-18-224-72-1
Jay Patel Insta...	i-09987e258a75e9df2	Stopped	t2.micro	-	No alarms +	us-east-2b	-
Taniya	i-0d386ccdf53989601	Stopped	t2.micro	-	No alarms +	us-east-2b	-

Instance: i-0fed6e7b7fce7e44 (tec2serverAkila)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#InstanceDetails:instancetype=i-0fed6e7b7fce7e44

BVC CTS IRCC Cloud others

aws Services Search [Alt+S]

EC2 > Instances > i-0fed6e7b7fce7e44

Instance summary for i-0fed6e7b7fce7e44 (tec2serverAkila) Info

Updated less than a minute ago

Instance ID i-0fed6e7b7fce7e44 (tec2serverAkila)	Public IPv4 address 3.16.152.253 [open address]	Private IPv4 addresses 172.31.37.149
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-16-152-253.us-east-2.compute.amazonaws.com [open address]
Hostname type IP name: ip-172-31-37-149.us-east-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-37-149.us-east-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding User: arn:aws:iam::698668199773:user:a.srinivasan036@mybvc.ca is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action on
Auto-assigned IP address 3.16.152.253 [Public IP]	VPC ID vpc-a18d30ca	Retry
IAM Role -	Subnet ID subnet-b67309fa	Auto Scaling Group name -

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS EC2 instance details page for an instance with ID i-0fed6e7b7fce7e444. The instance has a public IP of 54.16.152.253 and is associated with a VPC subnet vpc-a18d30ca. The security tab is selected, displaying security group information. A tooltip on the right indicates a permission issue related to the IAM role.

- Once that done go to security tab and edit the inbound rules.

The screenshot shows the AWS Security Group details page for a group named launch-wizard-10. The inbound rules tab is selected, showing three entries. The first entry is for port 22 (SSH) on TCP, the second for port 80 (HTTP) on TCP, and the third for port 443 (HTTPS) on TCP. All three entries have the source set to "Anywhere (0.0.0.0/0)".

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-025436ff9e43030a8	IPv4	SSH	TCP	22
-	sgr-0c004febef0f05e88	IPv4	HTTP	TCP	80
-	sgr-02cc7dbf65c0298f8	IPv4	HTTPS	TCP	443

Change the source to Anywhere IPV4. So that we can access the page in any IP address.

The screenshot shows the AWS Management Console interface for modifying inbound security group rules. The URL in the address bar is <https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#ModifyInboundSecurityGroupRulessecurityGroupId=sg-084a36c94ad14238a>. The page title is "Inbound rules". The main content area displays three existing rules:

Security group rule ID	Type	Protocol	Port range	Source	Description
sgr-025436ff9e43030a8	SSH	TCP	22	Anywhere	0.0.0.0/0
sgr-0c004efebef905e88	HTTP	TCP	80	Anywhere	0.0.0.0/0
sgr-02cc7dbf65c0298f8	HTTPS	TCP	443	Anywhere	0.0.0.0/0

At the bottom of the table, there is a "Add rule" button. Below the table, there are "Cancel", "Preview changes", and "Save rules" buttons.

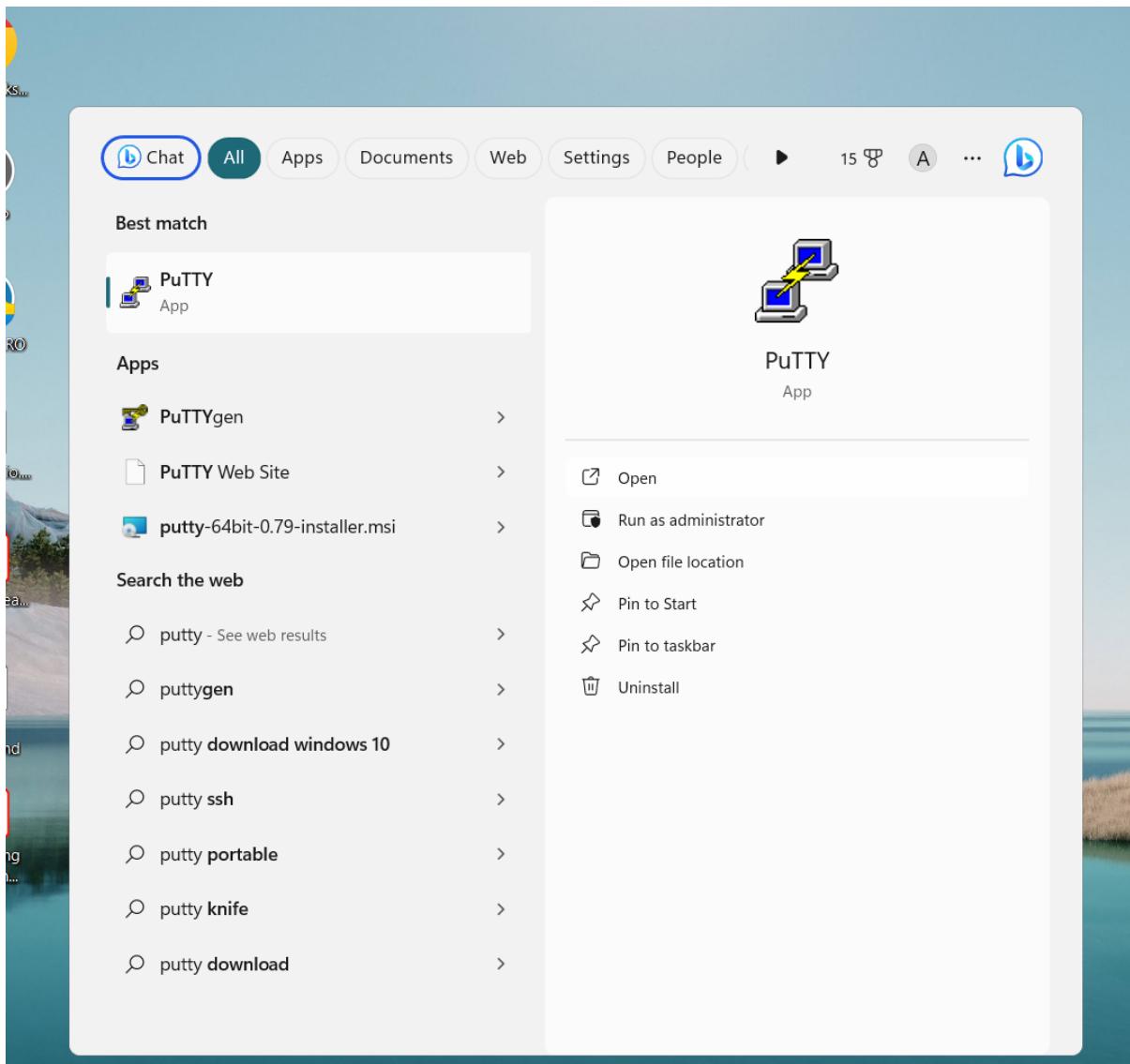
Post that open a github and create a new repository and push the code which you in your local.

The screenshot shows a GitHub repository page for "3102Lab". The URL in the address bar is <https://github.com/Akila-19/3102Lab>. The repository has 1 branch and 0 tags. The code listing shows files added via upload:

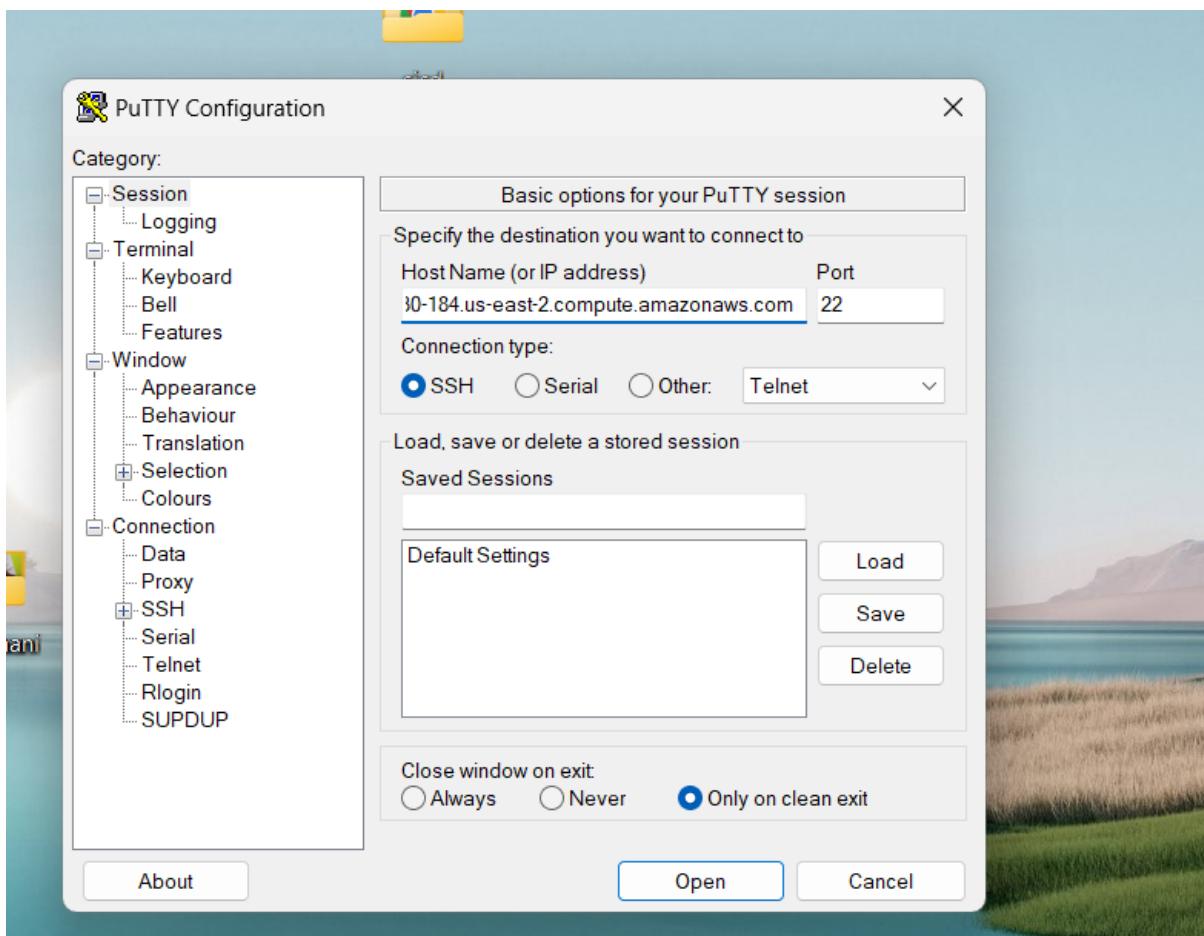
File	Last Commit
assets/images	yesterday
images	yesterday
index.html	yesterday
soeul.html	yesterday
styles.css	yesterday
venice.html	yesterday
venice.mp4	yesterday

The repository has 2 commits and 0 stars. It also includes sections for About, Releases, Packages, and Languages.

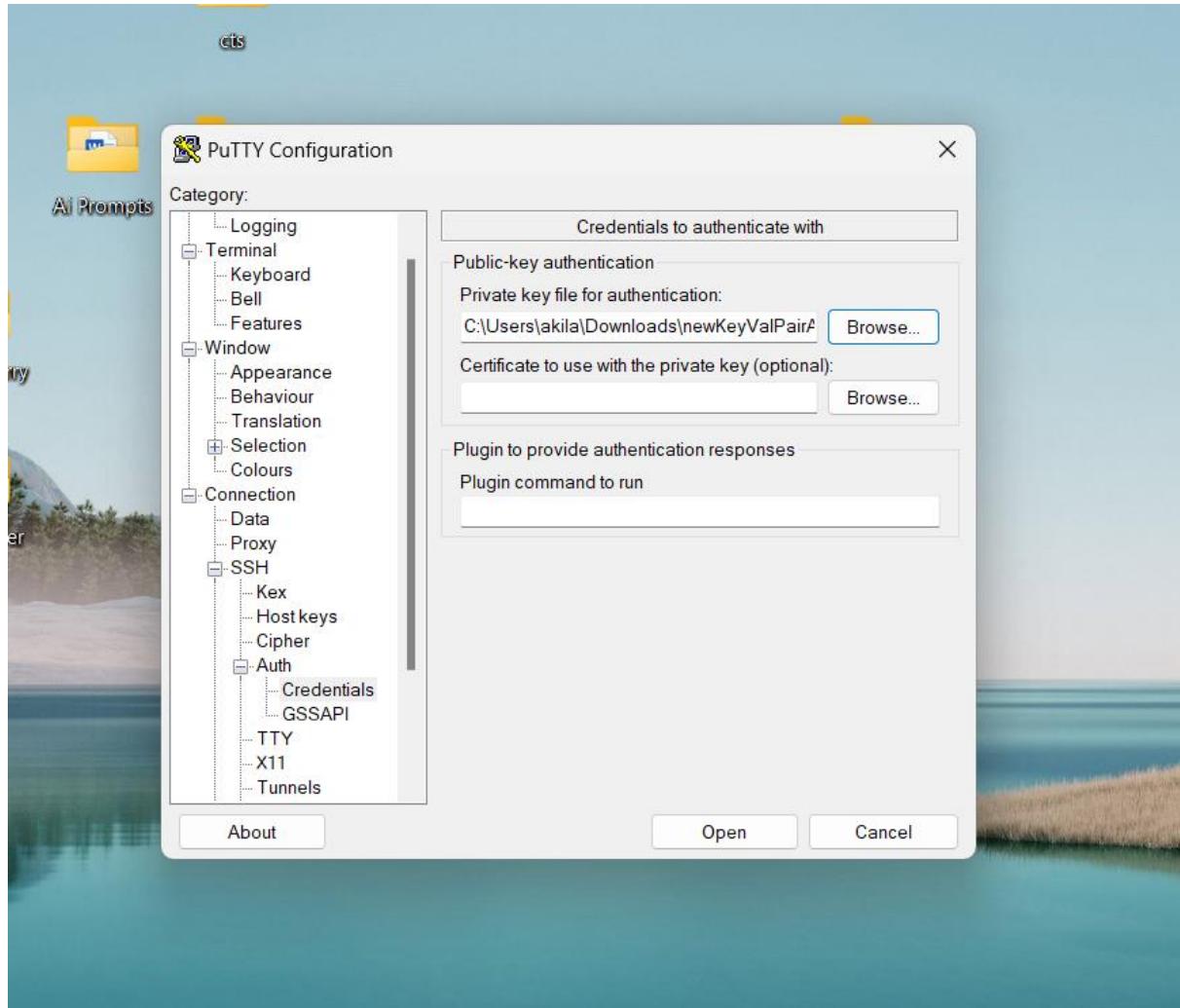
Once code pushed to git copy the git zip file link (It is one of the ways). open the putty.



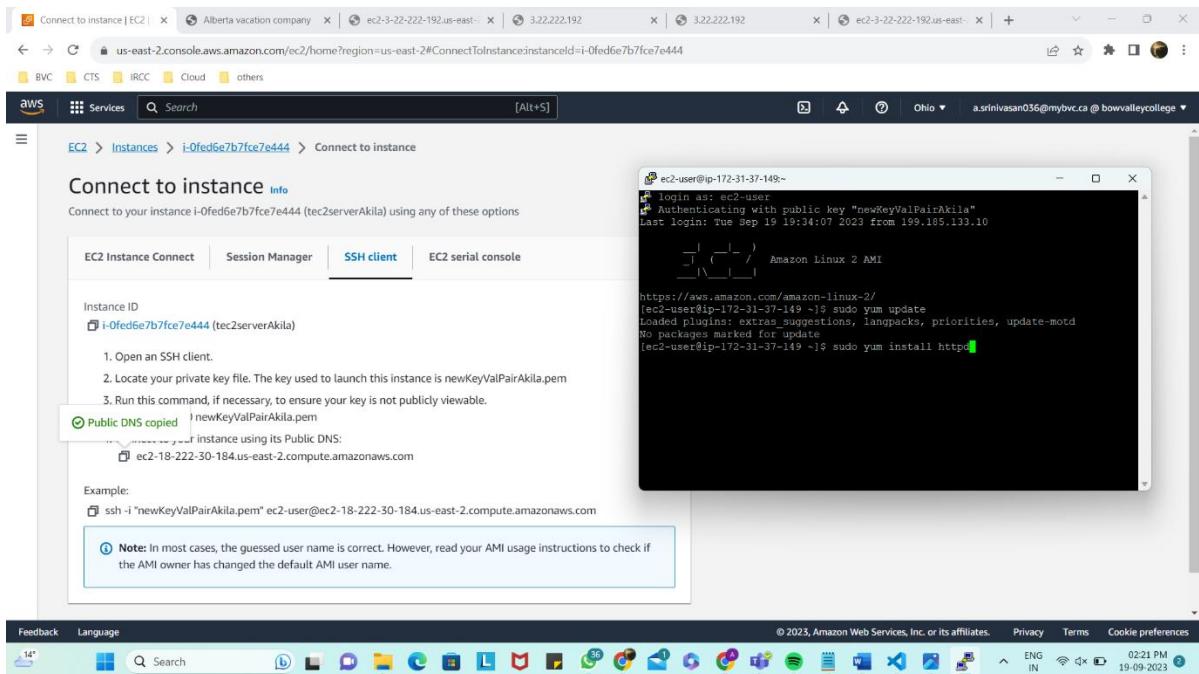
Paste the public DNS code which is copied in EC2 SSH client to the host name



- In the side navbar expand SSH ->Auth ->Credentials.
- Click the browse button and navigate to download key pair file.



- Give open button; a modal popup will show then give accept it will open a terminal.
- Then give the command “`ec2-user`” and update using “`sudo yum update`”



- Then install apache usning the command “ **`sudo yum install httpd`** ”
- Navigate to `/var/www/html/`.
- Then start the apache “ **`systemctl start httpd`** ”
- We can either create a html file and code manually or push code to git and give the git link. So here we are giving git link using the cmd “ **`wget repositoryLink`** ”
- And unzip the main.zip


```

ec2-user@ip-172-31-37-149:~/var/www/html$ 
Length: unspecified [application/zip]
main.zip: Permission denied

Cannot write to 'main.zip' (Success).
[ec2-user@ip-172-31-37-149 html]$ ls
[ec2-user@ip-172-31-37-149 html]$ ll
total 0
[ec2-user@ip-172-31-37-149 html]$ wget https://github.com/Akila-19/3102Lab/archive/refs/heads/main.zip
--2023-09-19 20:25:11-- https://github.com/Akila-19/3102Lab/archive/refs/heads/main.zip
Resolving github.com (github.com) ... 140.82.112.4
Connecting to github.com (github.com) 140.82.112.4:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/Akila-19/3102Lab/zip/refs/heads/main [following]
--2023-09-19 20:25:11-- https://codeload.github.com/Akila-19/3102Lab/zip/refs/heads/main
Resolving codeload.github.com (codeload.github.com) ... 140.82.113.10
Connecting to codeload.github.com (codeload.github.com) 140.82.113.10:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
main.zip: Permission denied

Cannot write to 'main.zip' (Success).
[ec2-user@ip-172-31-37-149 html]$ ls
[ec2-user@ip-172-31-37-149 html]$ sudo wget https://github.com/Akila-19/3102Lab/archive/refs/heads/main.zip
--2023-09-19 20:26:09-- https://github.com/Akila-19/3102Lab/archive/refs/heads/main.zip
Resolving github.com (github.com) ... 140.82.114.3
Connecting to github.com (github.com) 140.82.114.3:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'main.zip'

[          =>          ] 118,176,186 25.3MB/s  in 4.5s

2023-09-19 20:26:14 (25.2 MB/s) - 'main.zip' saved [118176186]

[ec2-user@ip-172-31-37-149 html]$ ls
main.zip
[ec2-user@ip-172-31-37-149 html]$ sudo unzip main.zip
Archive: main.zip
390b684ad2d9ef782adc0b7d0311ba756324de5
  creating: 3102Lab-main/assets/
  creating: 3102Lab-main/assets/images/
  inflating: 3102Lab-main/assets/images/home1.jpg
  inflating: 3102Lab-main/assets/images/home2.jpg

14" Search ENG IN 02:32 PM 19-09-2023

ec2-user@ip-172-31-37-149:~/var/www/html$ 
[          =>          ] 118,176,186 25.3MB/s  in 4.5s

2023-09-19 20:26:14 (25.2 MB/s) - 'main.zip' saved [118176186]

[ec2-user@ip-172-31-37-149 html]$ ls
main.zip
[ec2-user@ip-172-31-37-149 html]$ sudo unzip main.zip
Archive: main.zip
390b684ad2d9ef782adc0b7d0311ba756324de5
  creating: 3102Lab-main/assets/
  creating: 3102Lab-main/assets/images/
  inflating: 3102Lab-main/assets/images/home1.jpg
  inflating: 3102Lab-main/assets/images/home2.jpg
  inflating: 3102Lab-main/assets/images/home3.jpg
  inflating: 3102Lab-main/assets/images/home4.jpg
  inflating: 3102Lab-main/assets/images/home5.jpg
  inflating: 3102Lab-main/assets/images/seoul.jpg
  creating: 3102Lab-main/assets/images/seoul/
  inflating: 3102Lab-main/assets/images/seoul/s1.jpg
  inflating: 3102Lab-main/assets/images/seoul/s2.jpg
  inflating: 3102Lab-main/assets/images/seoul/s3.jpg
  inflating: 3102Lab-main/assets/images/seoul/s4.jpg
  inflating: 3102Lab-main/assets/images/seoul/s5.jpg
  inflating: 3102Lab-main/assets/images/seoul/s6.jpg
  inflating: 3102Lab-main/assets/images/seoul/s7.jpg
  inflating: 3102Lab-main/assets/images/seoul/s8.jpg
  inflating: 3102Lab-main/assets/images/venice.jpg
  creating: 3102Lab-main/assets/images/venice/
  inflating: 3102Lab-main/assets/images/venice/v1.jpg
  inflating: 3102Lab-main/assets/images/venice/v2.jpg
  inflating: 3102Lab-main/assets/images/venice/v3.jpg
  inflating: 3102Lab-main/assets/images/venice/v4.jpg
  inflating: 3102Lab-main/assets/images/venice/v5.jpg
  inflating: 3102Lab-main/assets/images/venice/v6.jpg
  inflating: 3102Lab-main/assets/images/venice/v7.jpg
  inflating: 3102Lab-main/assets/images/venice/v8.jpg
  creating: 3102Lab-main/images/
  inflating: 3102Lab-main/images/home1.jpg
  inflating: 3102Lab-main/images/home2.jpg
  inflating: 3102Lab-main/images/home3.jpg
  inflating: 3102Lab-main/images/home4.jpg
  inflating: 3102Lab-main/images/home5.jpg
  inflating: 3102Lab-main/images/seoul.jpg
  creating: 3102Lab-main/images/seoul/
  inflating: 3102Lab-main/images/seoul/s1.jpg
  inflating: 3102Lab-main/images/seoul/s2.jpg
  inflating: 3102Lab-main/images/seoul/s3.jpg

14" Search ENG IN 02:32 PM 19-09-2023

```

```

ec2-user@ip-172-31-37-149:~/var/www/html
infating: 3102Lab-main/assets/images/venice.jpg
creating: 3102Lab-main/assets/images/venice/
infating: 3102Lab-main/assets/images/venice/v1.jpg
infating: 3102Lab-main/assets/images/venice/v2.jpg
infating: 3102Lab-main/assets/images/venice/v3.jpg
infating: 3102Lab-main/assets/images/venice/v4.jpg
infating: 3102Lab-main/assets/images/venice/v5.jpg
infating: 3102Lab-main/assets/images/venice/v6.jpg
infating: 3102Lab-main/assets/images/venice/v7.jpg
infating: 3102Lab-main/assets/images/venice/v8.jpg
creating: 3102Lab-main/images/
infating: 3102Lab-main/images/home1.jpg
infating: 3102Lab-main/images/home2.jpg
infating: 3102Lab-main/images/home3.jpg
infating: 3102Lab-main/images/home4.jpg
infating: 3102Lab-main/images/home5.jpg
infating: 3102Lab-main/images/seoul.jpg
Creating: 3102Lab-main/images/venice/
infating: 3102Lab-main/images/venice/v1.jpg
infating: 3102Lab-main/images/venice/v2.jpg
infating: 3102Lab-main/images/venice/v3.jpg
infating: 3102Lab-main/images/venice/v4.jpg
infating: 3102Lab-main/images/venice/v5.jpg
infating: 3102Lab-main/images/venice/v6.jpg
infating: 3102Lab-main/images/venice/v7.jpg
infating: 3102Lab-main/images/venice/v8.jpg
infating: 3102Lab-main/index.html
infating: 3102Lab-main/seoul.html
infating: 3102Lab-main/styles.css
infating: 3102Lab-main/venice.html
infating: 3102Lab-main/venice.mp4
[ec2-user@ip-172-31-37-149 html]$ ls
3102Lab-main main.zip
[ec2-user@ip-172-31-37-149 html]$ pwd
/Volumes/www/html
[ec2-user@ip-172-31-37-149 html]$ sudo mv 3102Lab-main/* .
[ec2-user@ip-172-31-37-149 html]$ ls
3102Lab-main assets images index.html main.zip seoul.html styles.css venice.html venice.mp4
[ec2-user@ip-172-31-37-149 html]$ █

```

ec2-user@ip-172-31-37-149 html]\$ ls
3102Lab-main main.zip
[ec2-user@ip-172-31-37-149 html]\$ pwd
/Volumes/www/html
[ec2-user@ip-172-31-37-149 html]\$ sudo mv 3102Lab-main/* .
[ec2-user@ip-172-31-37-149 html]\$ ls
3102Lab-main assets images index.html main.zip seoul.html styles.css venice.html venice.mp4
[ec2-user@ip-172-31-37-149 html]\$ █

- Again go to the management console and click the open address it open tab in chrome , just remove secure in url which is “https” to “http”. Now we have hosted the website using AWS EC2.

The screenshot shows the AWS Management Console interface. On the left, there's a navigation sidebar with links like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances' (selected), 'Images', 'Elastic Block Store', and 'CloudWatch Metrics'. The main content area is titled 'EC2 > Instances > i-0fed6e7b7fce7e444'. It displays an 'Instance summary' for the instance 'i-0fed6e7b7fce7e444 (tec2serverAkila)'. Key details shown include:

- Public IPv4 address:** 3.16.152.253 [open address]
- Private IP4 addresses:** 172.31.37.149
- Public IPv4 DNS:** ec2-3-16-152-253.us-east-2.compute.amazonaws.com [open address]
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-172-31-37-149.us-east-2.compute.internal
- Instance type:** t2.micro
- VPC ID:** vpc-a18d30ca
- IAM Role:** -
- Subnet ID:** subnet-b67309fa

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2023, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and 'Cookie preferences'.

Website hosted in AWS EC2



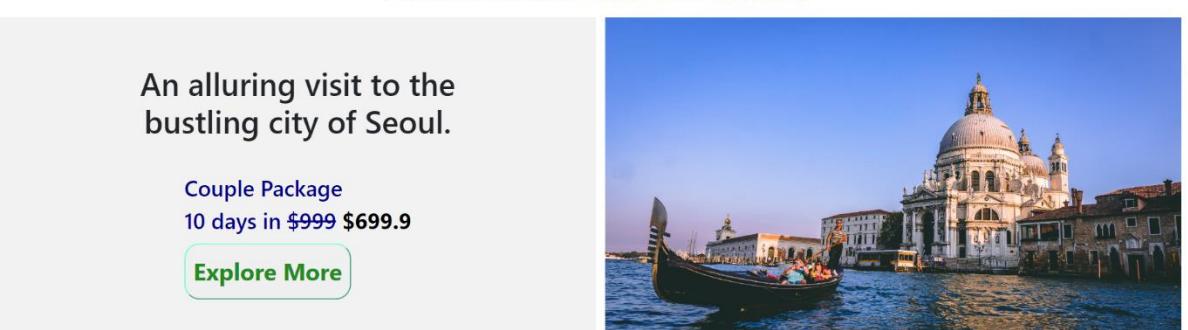
Let us have look for **Soeul** and **Venice**



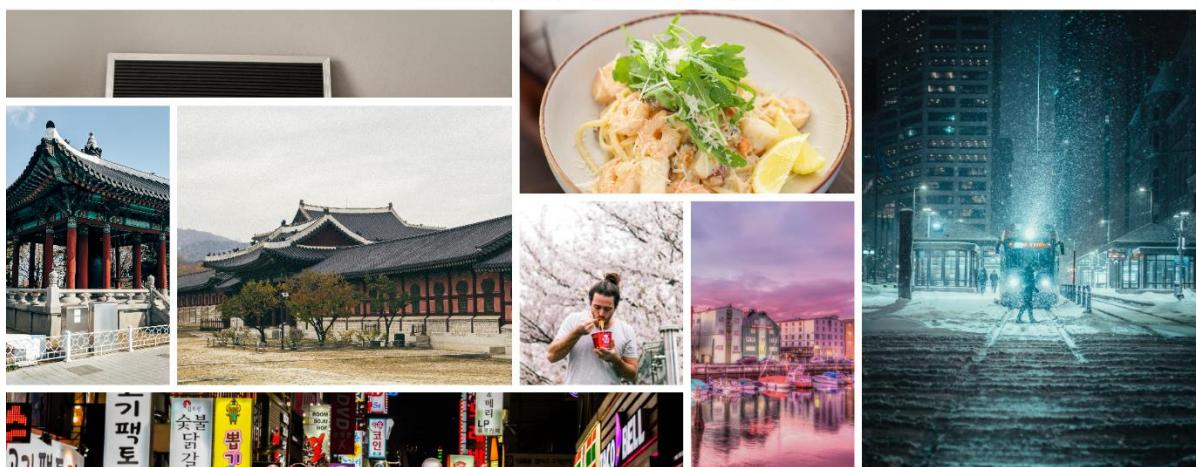
Let us have look for **Soeul** and **Venice**



Let us have look for **Soeul** and **Venice**



WELCOME TO SOEUL CITY



Seoul, the vibrant capital of South Korea, is a thrilling destination for tourists. This dynamic metropolis seamlessly blends modernity and tradition, offering visitors a diverse array of experiences. From its bustling markets and mouthwatering street food to its historic palaces and cutting-edge technology, Seoul promises an unforgettable journey into the heart of Korean culture

and innovation.

Please fill the form to book a plan

First name

Last name

Username

City

State

Zip

Agree to terms and conditions

[Back to Home Page](#)

WELCOME TO VENICE CITY



Venice, a captivating Italian city, is renowned for its intricate network of canals and exquisite Gothic, Renaissance, and Byzantine architecture. Its iconic Piazza San Marco, adorned with St. Mark's Basilica and the Campanile bell tower, is a symbol of its grandeur. With a rich cultural heritage that has inspired artists, musicians, and writers for centuries, Venice stands as a timeless and enchanting destination.

destination.

Please fill the form to book a plan

First name

Mark

Last name

Otto

Username

@ Username

City

City

State

State

Zip

Zip

Agree to terms and conditions

Submit form

Observation :

Initially I felt the bit hard while working in putty (with the commands). Later after doing it second time I have understand the meaning of commands like first install, then start, etc., Overall now I am confident in deploying website in AWS EC2.

Reference:

AWS Official documentation site.