



Department	Computer Engineering and Application
Program	B.Tech CSE[DA]
University Roll No.	2315100006
Section	CE
Class Roll No.	06
Assignment No.	01
Subject Name	Applied mathematics II
Subject Code	BCSS 0152
Date of Submission	23-02-2024

Submitted To:

Mr. Sandeep Kumar Sir

Submitted By:

Gangadhar

REPORT

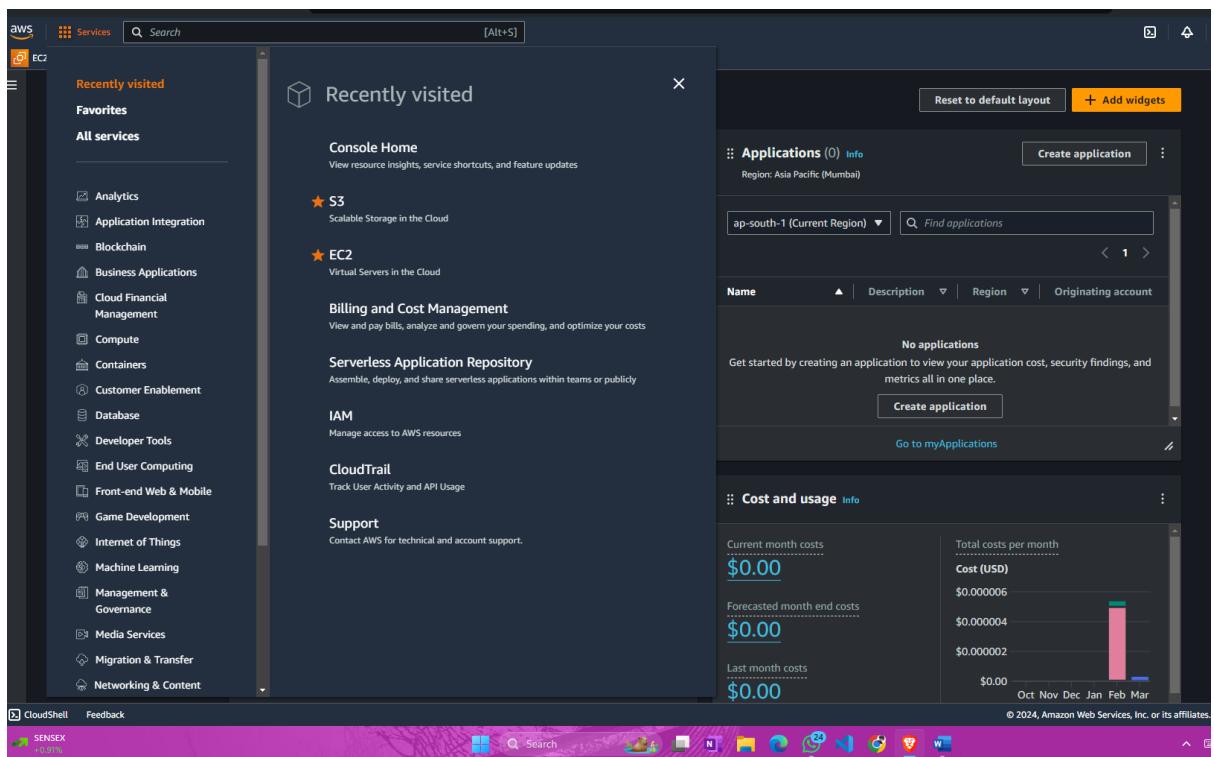
CLOUD COMPUTING

ELASTIC COMPUTE CLOUD (EC2)

-Launching an AWS Instance

Step 1) In this step,

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Compute EC2 as in the next step.



- Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the [AWS EC2](#) resources running.

The screenshot shows the AWS EC2 Dashboard for the Asia Pacific (Mumbai) Region. The left sidebar lists various EC2 resources: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 4, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 7, Snapshots 0, Volumes 4. Below this, there's a 'Launch instance' section with 'Launch instance' and 'Migrate a server' buttons, and a note that instances will launch in the Asia Pacific (Mumbai) Region. It also shows 'Instance alarms' (0 in alarm, 0 OK, 0 insufficient data) and 'Scheduled events'. The 'Service health' section indicates that the service is operating normally. The 'Zones' section lists three availability zones: ap-south-1a (Zone ID: aps1-az1), ap-south-1b (Zone ID: aps1-az3), and ap-south-1c (Zone ID: aps1-az2). The 'Account attributes' section shows the Default VPC (vpc-0eaa0d05a9ca33e51) and other settings like Data protection and security, Zones, EC2 Serial Console, Default credit specification, and Console experiments. The top right corner shows the AWS Region as Mumbai and the user's name as faizan.

Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

Here we are selecting Asia Pacific (Mumbai). AWS provides 10 Regions all over the globe

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with 'Resources' and a main panel with 'Launch instance' and 'Service health'. The 'Service health' section shows 'Asia Pacific (Mumbai)' is operating normally. On the right, a sidebar titled 'Offer usage (monthly)' lists various AWS regions and their respective Amazon Machine Image (AMI) IDs. The 'Amazon Linux 2 AMI (HVM)' entry is highlighted.

Region	Offered AMI ID
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
South America (São Paulo)	sa-east-1
Africa (Cape Town)	af-south-1

Step 3) In this step

- Once your desired Region is selected, come back to the EC2 Dashboard.
- Click on 'Launch Instance' button in the section of Create Instance (as shown below).

The screenshot shows the AWS EC2 Instances page. The 'Instances' section is active, and the 'Launch instances' button is highlighted with a red box. The page also includes a search bar, filter options, and a table header for columns like Name, Instance ID, Instance state, etc.

Step 4) On the Choose an Amazon Machine Image (AMI) screen, choose Amazon Linux 2 AMI (HVM), SSD Volume Type

Name and tags [Info](#)

Name
bastion [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

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S [Search](#)

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type [Free tier eligible](#)
ami-0b029b1931b347543 (64-bit (x86)) / ami-0335e6e62378b87c3 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2 Kernel 5.10 AMI 2.0.20230221.0 x86_64 HVM gp2

Architecture [AMI ID](#) [Verified provider](#)
64-bit (x86) ami-0b029b1931b347543

Step 5) On the Choose an Instance Type screen, select t2.micro as the instance type.

Instance type [Info](#)

Instance type
t2.micro [Free tier eligible](#)

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour

[Compare instance types](#)

Step 6) Create a key pair to access EC2 Instance. Select Create a new key pair, enter analytics-hol as the Key pair name, and click **Download Key Pair**. Save the Key Pair safely on your local machine and click **[Launch Instances]**. (EC2 Instance startup may take several minutes.)

The screenshot shows two side-by-side windows. On the left is a 'Key pair (login)' configuration page with a dropdown menu set to 'Select' and a red box around the 'Create new key pair' button. On the right is a 'Create key pair' dialog box. It contains instructions about key pairs, a 'Key pair name' field with 'analytics-hol' typed in, and a note about character limits. It also includes sections for 'Key pair type' (set to RSA), 'Private key file format' (set to .pem), and a 'Create key pair' button at the bottom.

Step 7) After creating key pair ,under network setting you can allow http and https traffic based on your requirements.

The screenshot shows the 'Network settings' configuration page for an EC2 instance. It includes sections for 'Network' (vpc-0eea0d05a9ca33e51), 'Subnet' (No preference), 'Auto-assign public IP' (Enable), and 'Firewall (security groups)' (Create security group selected). A note indicates additional charges apply outside of free tier allowance. The 'Create security group' section shows three rules: 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 note at the bottom states: 'We'll create a new security group called 'launch-wizard-8' with the following rules:' followed by the three listed rules.

Step 8) Ignore the rest of the settings and click **Launch instance**.

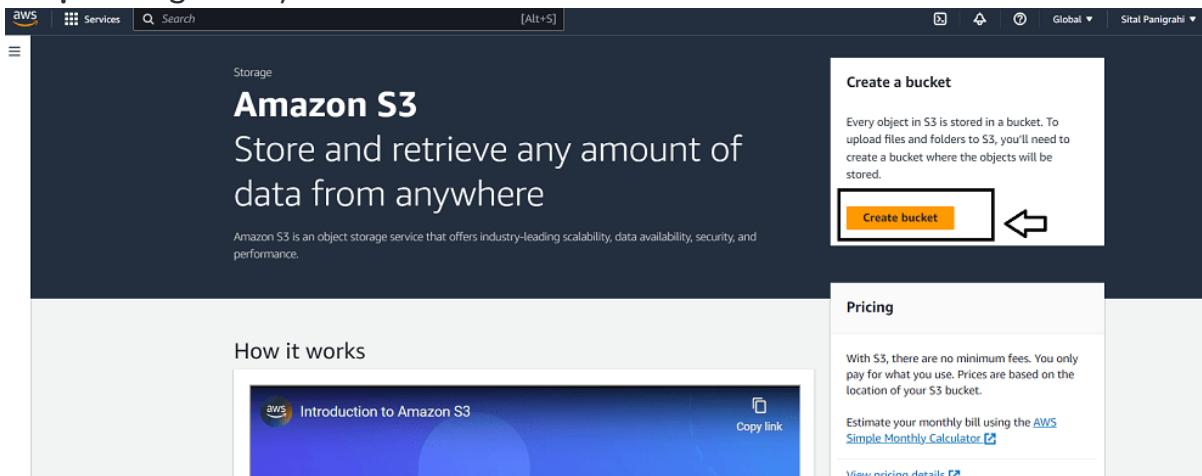
The screenshot shows the AWS Launch Wizard Step 8 configuration screen. It includes the following sections:

- Network:** Shows three checked options: "Allow SSH traffic from Anywhere (0.0.0.0/0)", "Allow HTTPS traffic from the internet", and "Allow HTTP traffic from the internet". A warning message 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."
- Configure storage:** Shows 1x 8 GiB gp3 Root volume (Not encrypted). A note says: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage".
- Advanced details:** Shows 0 x File systems.
- Summary:** Shows 1 instance, Software Image (AMI) as Amazon Linux 2023 AMI 2023.4.2..., Virtual server type (instance type) as t2.micro, Firewall (security group) as New security group, and Storage (volumes) as 1 volume(s) - 8 GiB.
- Free tier information:** A callout box details the free tier benefits: "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."
- Buttons:** Includes "Cancel" and "Launch instance" (which is circled in red).

Your EC2 instance is launched

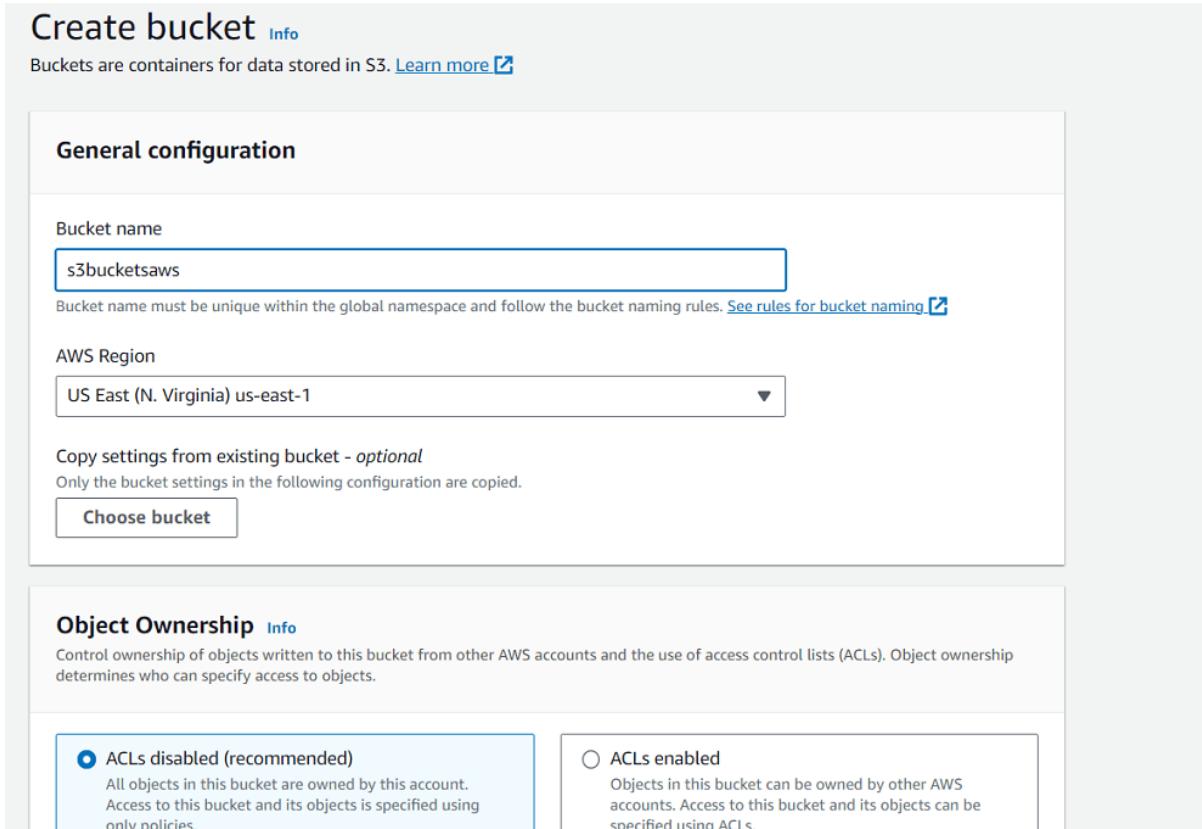
CONNECTING EC2 WITH S3

Step 1: Log in to your Console and search S3 → Click on create bucket



The screenshot shows the AWS S3 service page. At the top right, there is a prominent orange "Create bucket" button. To the left of the main content area, there's a section titled "How it works" with a video thumbnail labeled "Introduction to Amazon S3". On the right side, there's a "Pricing" section with a note about no minimum fees and a link to the AWS Simple Monthly Calculator.

Step 2: Give it a unique name(follow S3 naming convention) → Choose region(Keep other things as it is)→ Click on create



The screenshot shows the "Create bucket" configuration page. The "General configuration" tab is active. In the "Bucket name" field, the value "s3bucketsaws" is entered. Below the field, a note states that the bucket name must be unique within the global namespace and follow the bucket naming rules, with a link to "See rules for bucket naming". The "AWS Region" dropdown is set to "US East (N. Virginia) us-east-1". Under "Copy settings from existing bucket - optional", there is a "Choose bucket" button. The "Object Ownership" tab is also visible at the bottom.

Now open the bucket→ Click on Upload → Choose on Add file→ select file from system →Click on Upload.

The screenshot shows the AWS S3 console interface for a bucket named "s3bucketsaws".

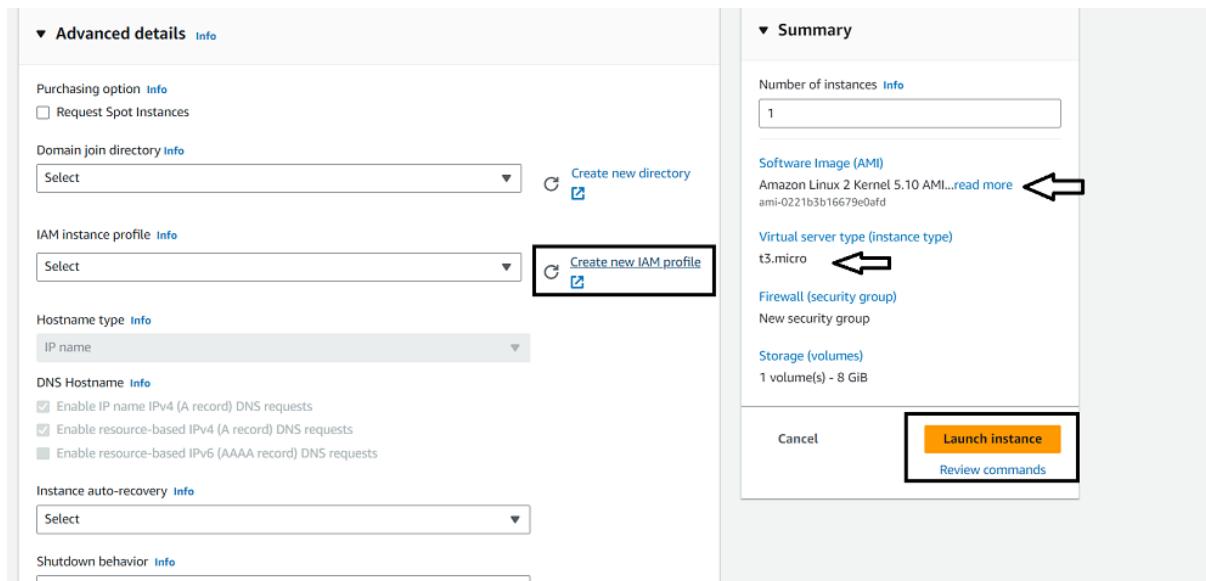
Step 1: The "Objects (0)" section shows a list of objects. A yellow box highlights the "Upload" button in the top right corner of the toolbar, with a black arrow pointing to it.

Step 2: A modal window titled "Files and folders (1 Total, 11.0 MB)" is open. It lists the uploaded file "Working Capital.docx". The "Add files" button is highlighted with a yellow box and a black arrow pointing to it.

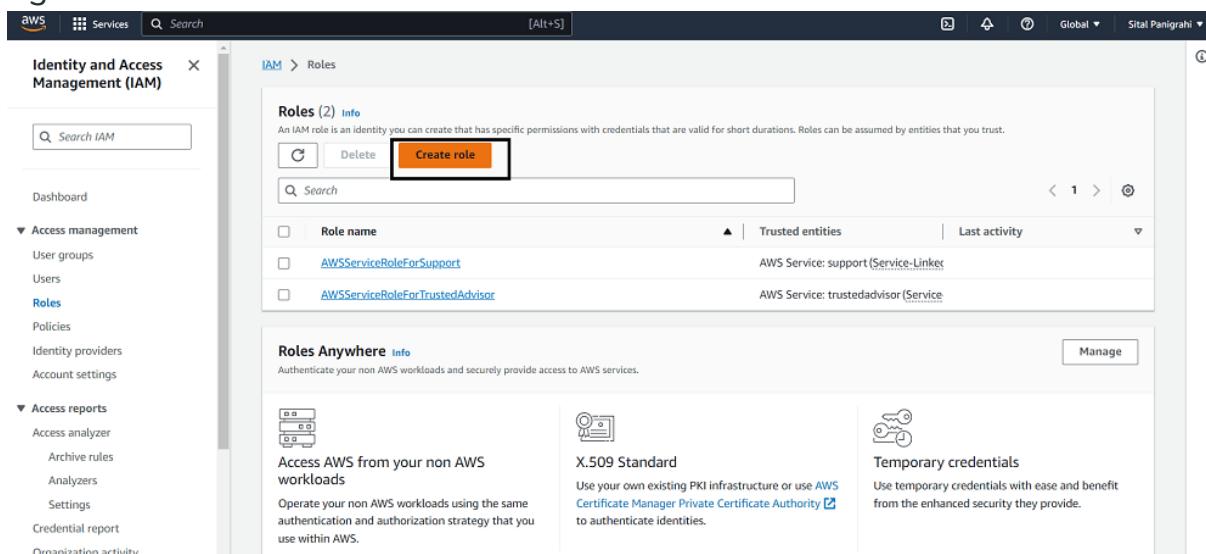
Step 3: The "Destination" section is shown, indicating the destination is "s3://s3bucketsaws". The "Upload" button in the bottom right corner of the modal is highlighted with a yellow box and a black arrow pointing to it.

Step 4: Now let's create an instance, for that

Click on “launch instance”-> Give it a name-> Choose AMI(Here I have taken **Linux 2 kernel**)>Choose t3.micro as instance type->Go to advanced, In IAM instance profile-> Click on “Create new IAM profile”



Step 5: Click on Create role->Click on AWS service-> in use case select EC2-> Now in permission search and select “S3fullaccess”->Click Next->give it a Name and Create



Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Choose a use case for the specified service.
Use case

EC2 Allows EC2 instances to call AWS services on your behalf.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Add permissions [Info](#)

Permissions policies (1/876) [Info](#)

Choose one or more policies to attach to your new role.

Filter by Type

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the ...

▶ Set permissions boundary - optional

Cancel Previous **Next**

Step 6: Now in launch template-> In IAM instance profile-> refresh and select newly created IAM role.

Step 7: Now, after launching SSH into the server.

```
ec2-user@ip-172-31-35-3:~ x + ^

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sital\OneDrive\Desktop>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\sital\OneDrive\Desktop> ssh -i mykeypair.pem ec2-user@13.51.200.78
The authenticity of host '13.51.200.78 (13.51.200.78)' can't be established.
ED25519 key fingerprint is SHA256:JhkDiEwp1dD6NORxpSYs+79zP1Adnd6XKSJx5zk2Ts0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.51.200.78' (ED25519) to the list of known hosts.

      _ _|_ _|- )
      _| ( _ /   Amazon Linux 2 AMI
     ___|\_\_||__|_|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 36 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-35-3 ~]$ |
```

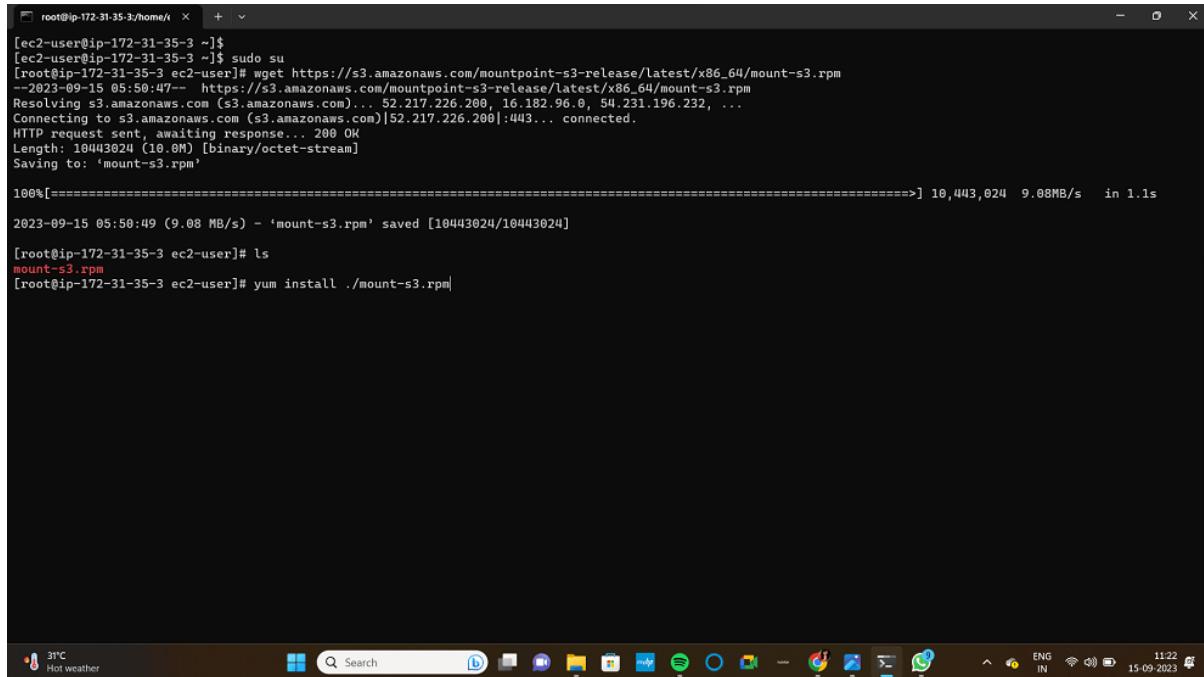
Step 8: Run following commands

```
# sudo su  
# wget https://s3.amazonaws.com/mountpoint-s3-  
release/latest/x86_64/mount-s3.rpm
```

```
[root@ip-172-31-35-3 ~]#
[ec2-user@ip-172-31-35-3 ~]$ sudo su
[root@ip-172-31-35-3 ~]# wget https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
```

Install the mount-s3

```
# yum install ./mount-s3.rpm
```



```
[root@ip-172-31-35-3 ~]$ [ec2-user@ip-172-31-35-3 ~]$ sudo su [root@ip-172-31-35-3 ec2-user]# wget https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm --2023-09-15 05:50:47-- https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.226.200, 16.182.96.0, 54.231.196.232, ... Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.226.200|:443... connected. HTTP request sent, awaiting response... 200 OK Length: 10443024 (10.0M) [binary/octet-stream] Saving to: 'mount-s3.rpm' 100%[=====] 10,443,024 9.08MB/s in 1.1s 2023-09-15 05:50:49 (9.08 MB/s) - 'mount-s3.rpm' saved [10443024/10443024] [root@ip-172-31-35-3 ec2-user]# ls mount-s3.rpm [root@ip-172-31-35-3 ec2-user]# yum install ./mount-s3.rpm
```

Now, use this command to list of all the buckets present in your AWS S3

```
# aws s3 ls
```

```
[root@ip-172-31-35-3 ec2-user]# aws s3 ls
2023-09-15 05:31:40 s3bucketsaws
[root@ip-172-31-35-3 ec2-user]# |
```

Now create directory

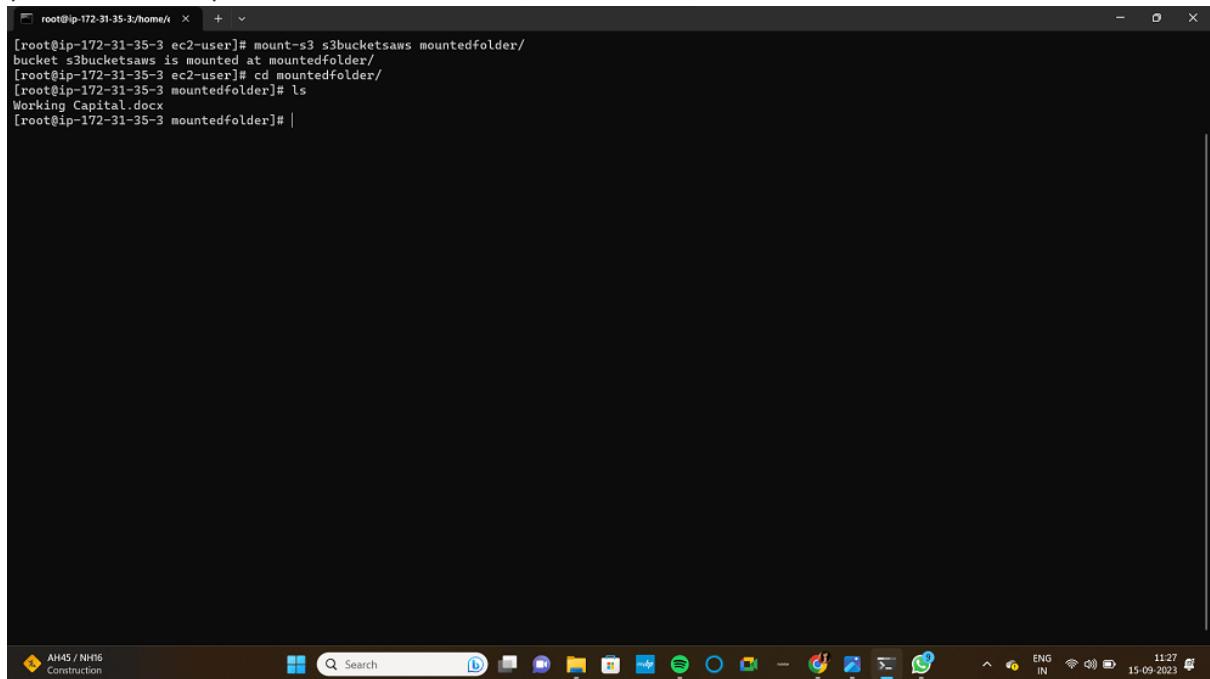
```
# mkdir {name of directory}
```

```
[root@ip-172-31-35-3 ec2-user]# aws s3 ls
2023-09-15 05:31:40 s3bucketsaws
[root@ip-172-31-35-3 ec2-user]# clear
[root@ip-172-31-35-3 ec2-user]# mkdir mountedfolder
[root@ip-172-31-35-3 ec2-user]# ls
mountedfolder mount-s3.rpm
[root@ip-172-31-35-3 ~]
```

Now to mount the required bucket

```
# mount-s3 s3bucketsaws[name of the bucket] mountedfolder/[directory where to be mounted]
```

Now, go to the directory and do “ls”
you can find your files.



A screenshot of a Windows desktop environment. At the top, there is a dark-themed terminal window titled "root@ip-172-31-35-3:/home/r". The window contains the following command-line session:

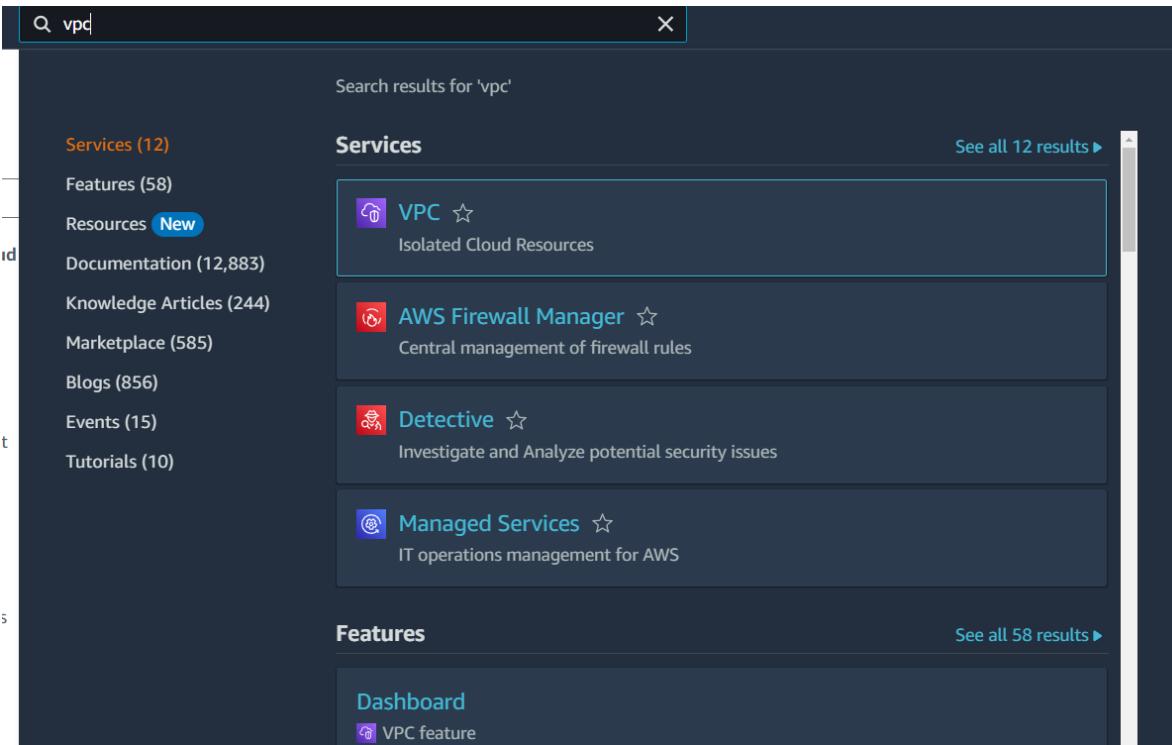
```
[root@ip-172-31-35-3 ec2-user]# mount-s3 s3bucketsaws mountedfolder/
bucket s3bucketsaws is mounted at mountedfolder/
[root@ip-172-31-35-3 ec2-user]# cd mountedfolder/
[root@ip-172-31-35-3 mountedfolder]# ls
Working Capital.docx
[root@ip-172-31-35-3 mountedfolder]# |
```

The taskbar below the terminal window shows several pinned icons, including Microsoft Edge, File Explorer, Task View, and others. On the right side of the taskbar, there are system status icons for battery level (11:27), language (ENG IN), signal strength, and date/time (15-09-2023).

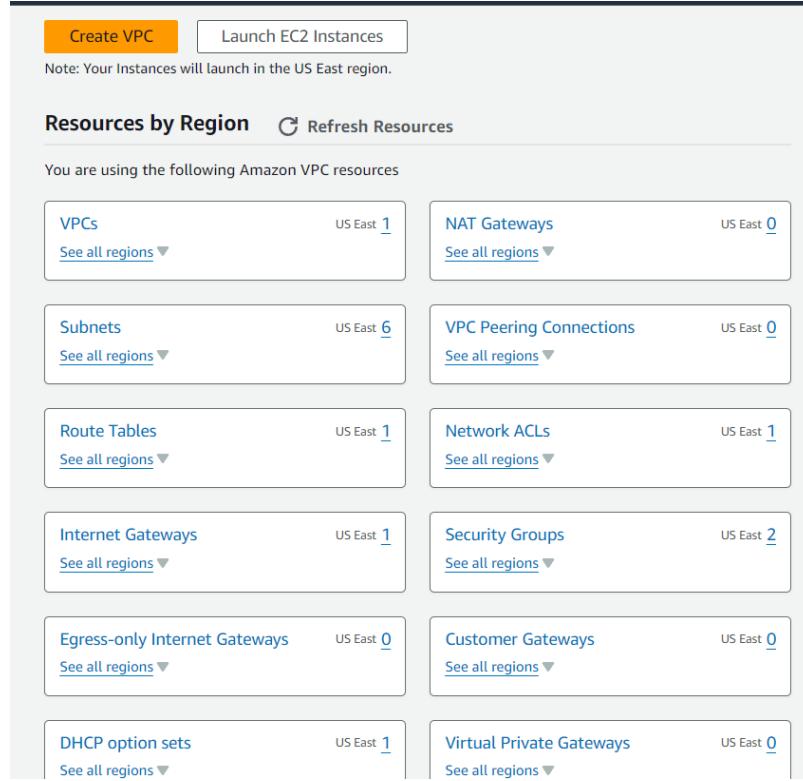
Amazon VPC (Virtual Private Cloud) Working

Follow the Steps Mentioned Below to Configure Virtual Private Cloud (VPC)

Step 1: Login into AWS Console and navigate to the VPC as shown below



Step 2: After navigating to the AWS VPC know click on create VPC



Step 3: Configure all the details required to create as shown in the image below. Some of the most required settings to configure VPC was as follows

Name of the Network.

IPv4 CIDR.

And tags of VPC after that click on create VPC.

AWS Services Search [Alt+S]

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
my-vpc-01

IPv4 CIDR block Info
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block
10.0.0.0/24
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me
IPv6 CIDR owned by me

Tenancy Info
Default

Tags

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.

Key	Value - <i>optional</i>
<input type="text"/> Name	<input type="text"/> vivaan-vpc

Add tag Remove tag
You can add 49 more tags

Cancel Create VPC

Step 4: Virtual Private Cloud Created successfully with the required setting to us.

vpc-03a292c14607698c3 / vivaan-vpc

Actions ▾

Details Info

VPC ID vpc-03a292c14607698c3	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0dc62d6913301e9c2	Main route table rtb-01be9c1b63c65552a	Main network ACL acl-05b04736729f55203
Default VPC No	IPv4 CIDR 10.0.0.0/24	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 637423304993	

CLOUDCRAFT

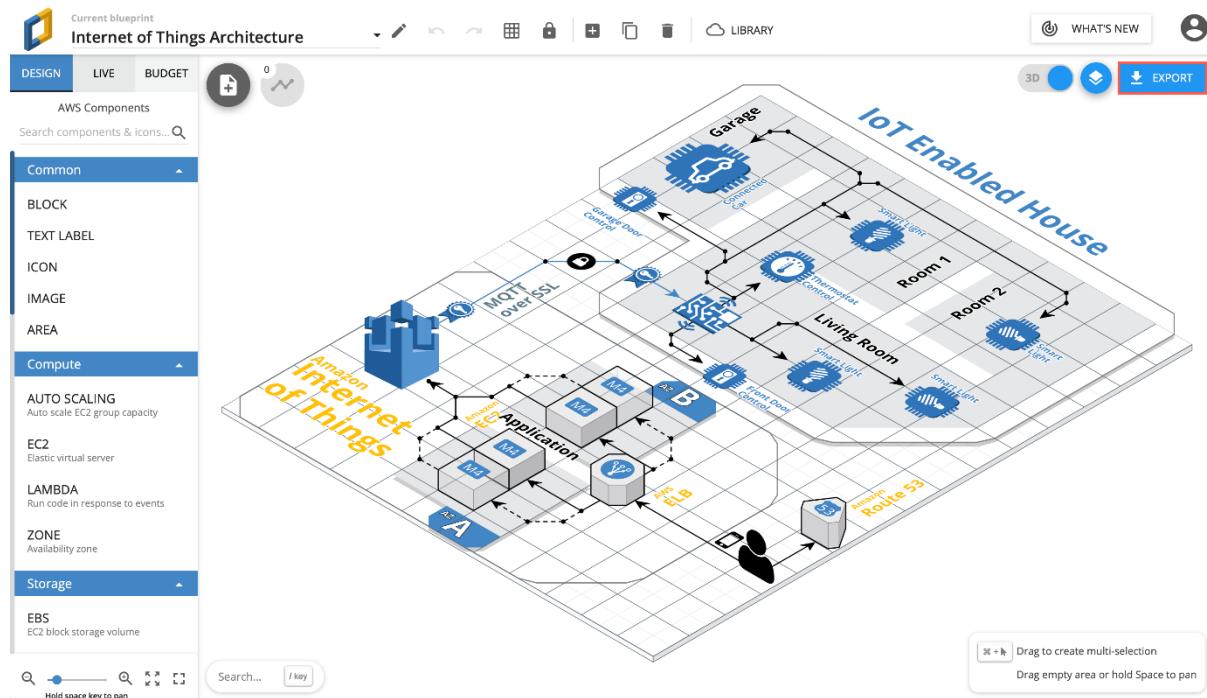
Cloudfactory is a software tool specifically designed to help manage and visualize cloud infrastructure. It caters primarily to users of cloud platforms like Amazon Web Services (AWS) and Microsoft Azure.

Cloudfactory offers features like:

- Automatic visualization of existing cloud environments on AWS or Azure.
- Designing cloud architectures from scratch with drag-and-drop functionality.
- Generating cost estimates (budgets) as you design your cloud architecture.
- Creating professional-quality cloud architecture diagrams for documentation and collaboration.

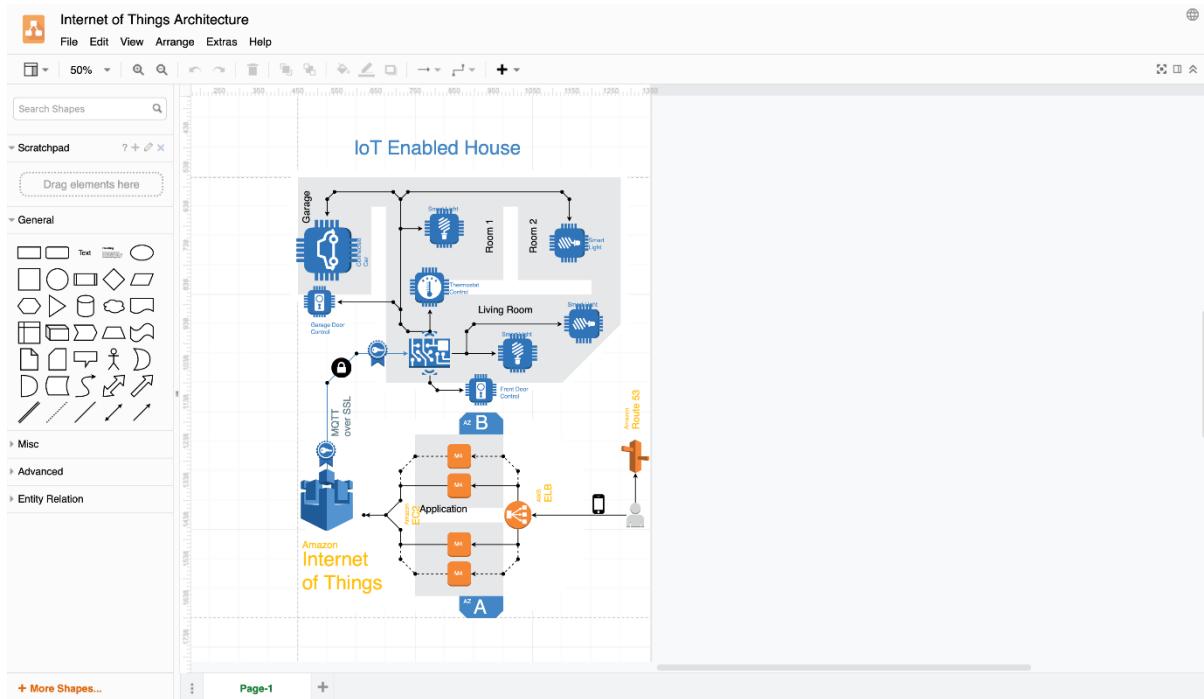
❖ How to export from Cloudfactory:

1. In your Cloudfactory account, in the top right of the drawing area, click *Export*.

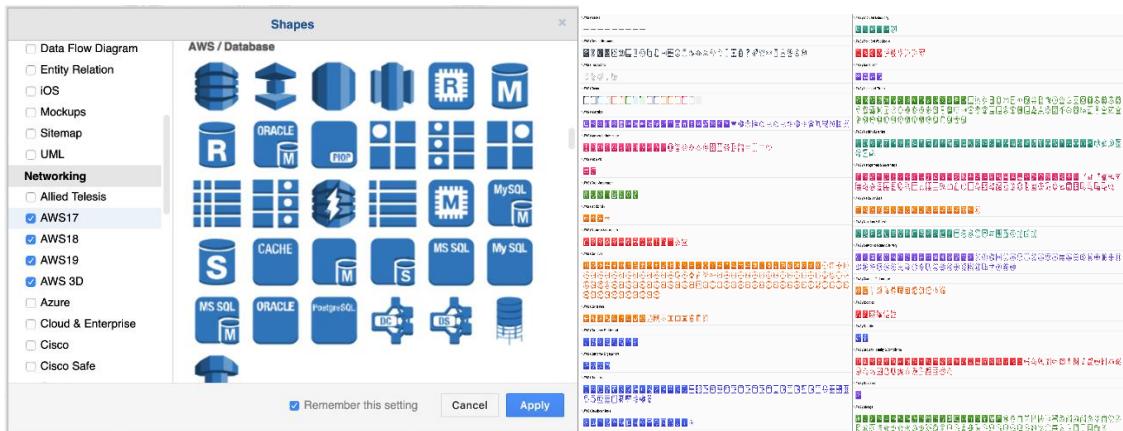


- You can now save this diagram file to your version control system, cloud storage platform, or whichever location you have chosen to maintain a record of your infrastructure.

❖ Internet of architecture:



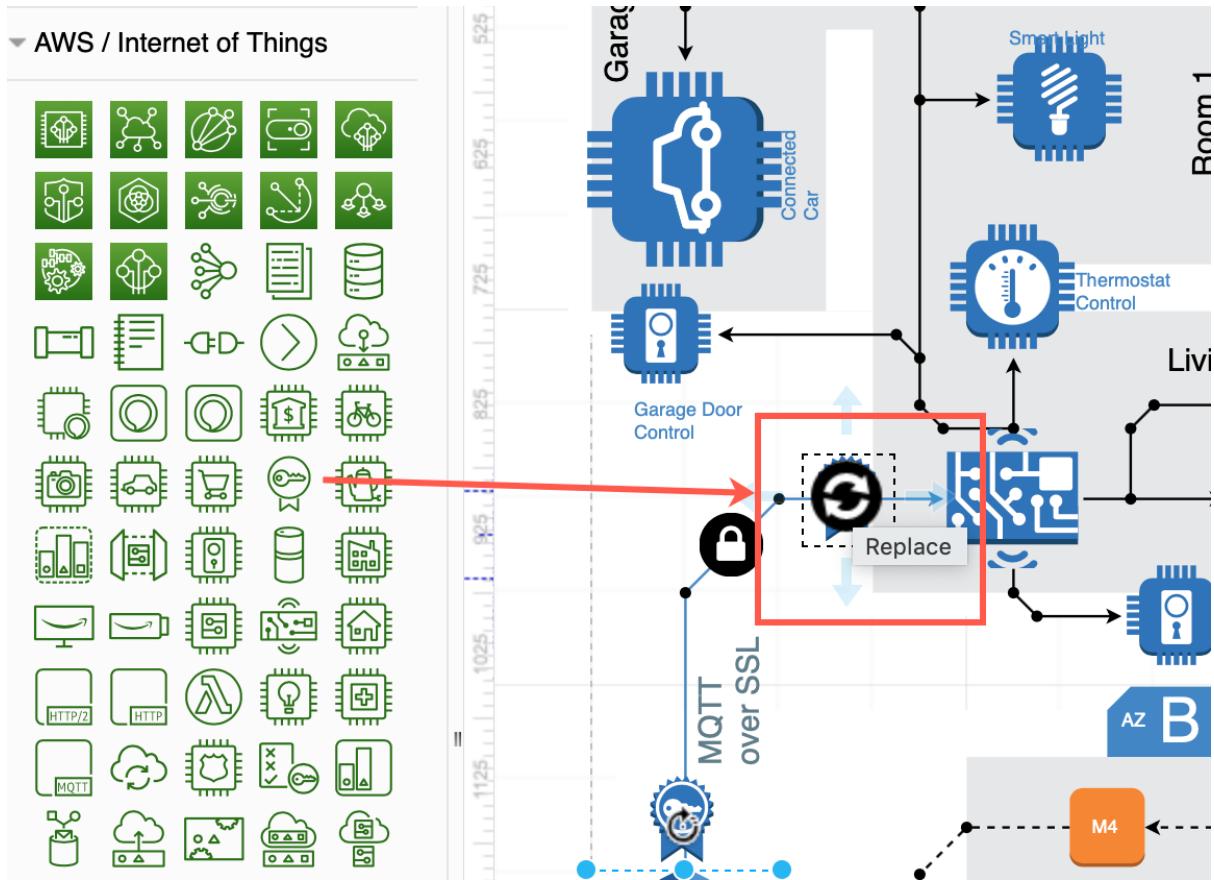
❖ Editing AWS diagrams :



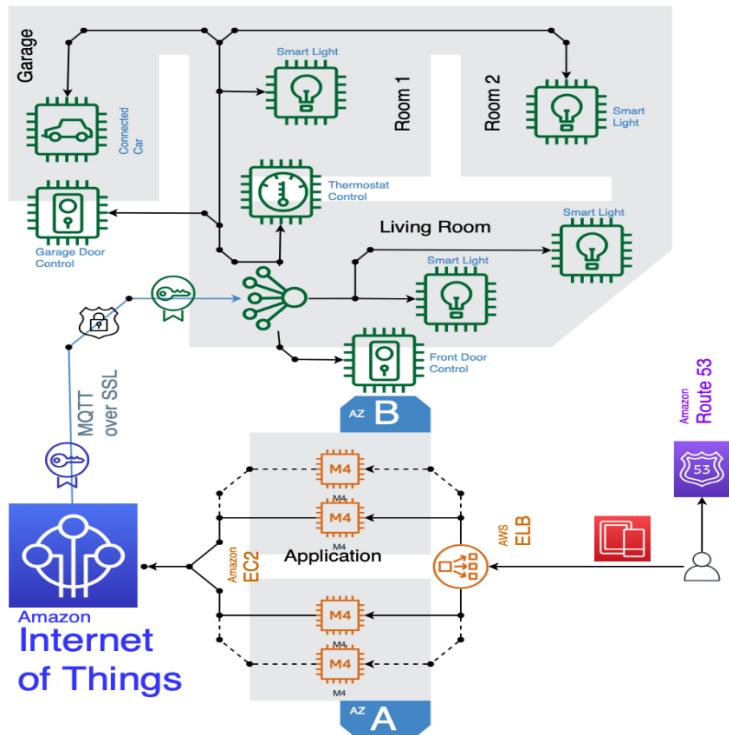
2. First, enable the AWS libraries. Click *More Shapes* at the bottom of the left panel, enable one or more AWS libraries under Networking, then click *Apply*.

3. There are many AWS shapes available to visualise your AWS infrastructure. When you enable just one of the shape libraries above, you'll see a number of sub-categories of shapes appear in the left panel.

❖ Replace shapes , Copy styles ,Copy sizes:



IoT Enabled House



Launching a website on the Ubuntu server:

Step 1: Launching an EC2 Instance:

Begin by launching an EC2 instance on AWS, opting for the Ubuntu image to serve as the foundation for your web server.

Step 2: Connecting to EC2 Instance:

Establish a secure connection to your EC2 instance using EC2 Instance Connect, ensuring a seamless and secure connection.

Step 3: Installing Apache Web Server:

```
bash                                     Copy code
sudo apt update && sudo apt upgrade

bash                                     Copy code
sudo apt install apache2
```

Step 4: Verifying Apache Installation:

Check if Apache is successfully installed by running commands such as which apache2 or apache2 -v.

Step 5: Starting and Checking Apache Service:

Initiate the Apache service using :

- sudo service apache2 start.

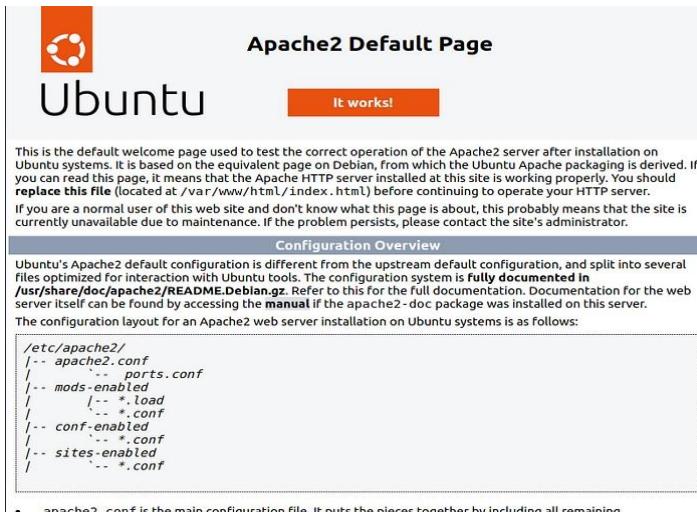
Verify the status of the Apache service with:

- sudo service apache2 status.

Step 6: Viewing Default Apache Page:

Copy the public IP address of your EC2 instance and paste it into a browser.

You should see the default Apache welcome page.



- apache2.conf is the main configuration file. It puts the pieces together by including all remaining

Step 7: Customizing the Website:

- Navigate to the directory where web content is stored:
- cd /var/www/html.
- Remove the default index.html file:
- sudo rm index.html.
- Create and edit a new index.html file with your own content.

Step 8: Restarting Apache Service:

- After customization, restart the Apache service with
- sudo service apache2 restart

Step 9: Viewing Customized Website:

- Copy the public IP address of your EC2 instance again.
- paste it in the browser.
- You should now see your customized website content.