

12-Aug-2024

Internship Day - 20 Report:

Remote Access Guide for Ubuntu Linux

Steps to Remotely Access One Ubuntu PC from Another

1) Ping Google to Check Connectivity:

```
ping www.google.com
```

2) Install Curl:

```
sudo apt install curl
```

3) Test Curl with Google

```
curl google.com
```

4) Switch to Super User

```
sudo su
```

5) Update Package List

```
apt update
```

6) Install Firewall

```
apt install firewall -y
```

7) Install Net Tools

```
apt install net-tools -y
```

8) Install OpenSSH Server

```
apt install openssh-server -y
```

9) Check SSH Status

```
systemctl status ssh
```

10) Identify Network Type

- If the ping is successful, both systems are on the same network.
- If not, they are on different networks.

11) Check Current User

```
whoami
```

12) SSH Into Another Ubuntu Machine

```
ssh <username>@<another_ubuntu_ip>
```

13) Enter Password For the Remote Machine

14) Optional Add a Rich Rule to Firewall

- `firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" destination address="175.176.187.102" reject' --permanent`

15) List Current Rich Rules

```
firewall-cmd --list-rich-rules
```

13-Aug-2024

Internship Day - 21 Report:

Amazon VPC: Purpose and Benefits in Steps

What is Amazon VPC?

Definition: Amazon Virtual Private Cloud (VPC) is a service that allows you to create a logically isolated virtual network within the AWS cloud.

Why Use Amazon VPC?

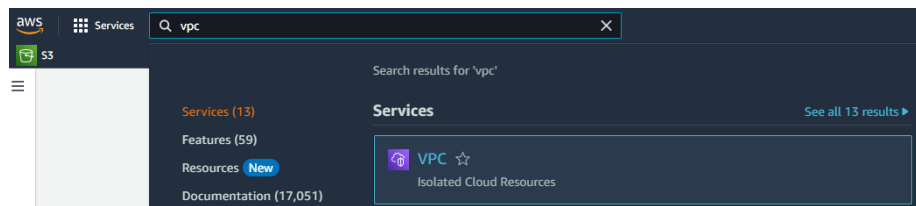
1. **Enhanced Security:** Provides a secure environment by isolating resources from other AWS users.
2. **Flexibility and Scalability:** Allows customization of network architecture to meet specific needs and easily scale resources.
3. **Cost-Effective:** No additional charges for creating a VPC; only pay for specific components like NAT gateways.
4. **Integration with AWS Services:** Seamlessly integrates with other AWS services (e.g., RDS, ELB) while maintaining network control.

Key Features of Amazon VPC:

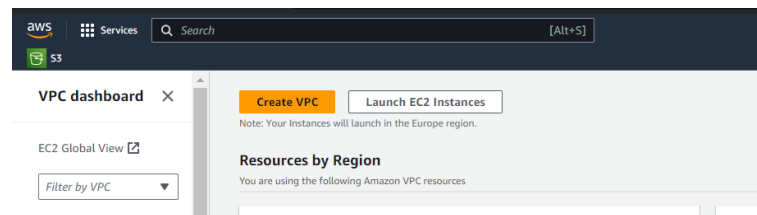
- **Isolation and Control:** Complete control over your virtual network, including IP address range and subnet configuration.
- **Customizable Network Configuration:** Create public and private subnets tailored to your application needs.
- **Security:** Utilize security groups and network ACLs to manage traffic, with VPC flow logs for monitoring.
- **Connectivity Options:** Connect to the internet via Internet Gateway or to on-premises networks using VPN; supports VPC peering.

Steps to Create a VPC and Launch Instances in AWS:

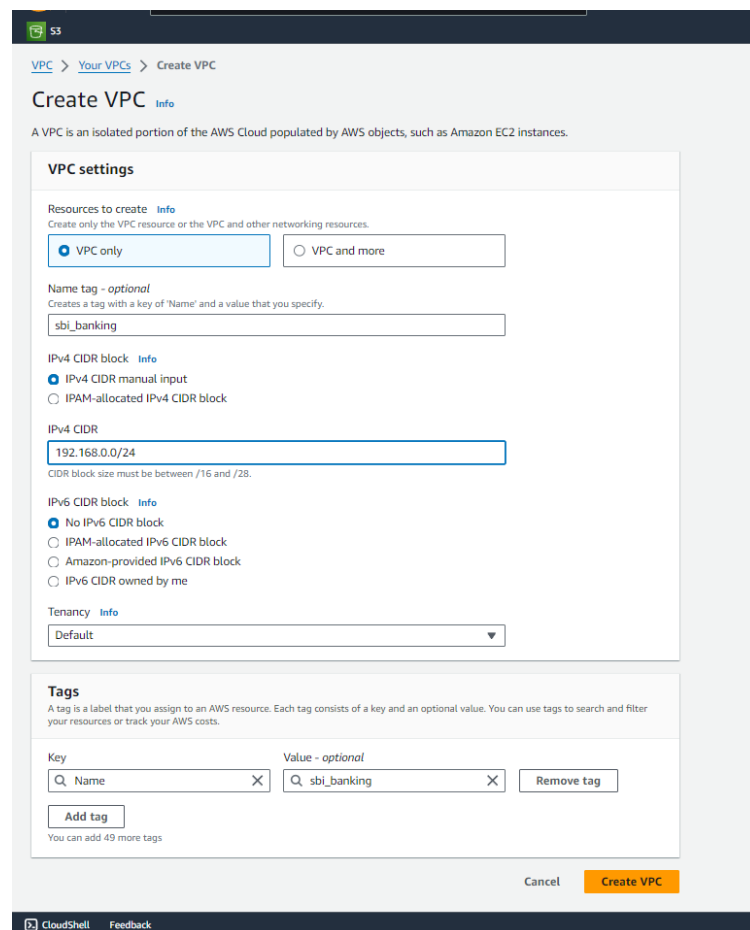
- 1) **Search for VPC:** In the AWS Management Console, search for "VPC" and select it.



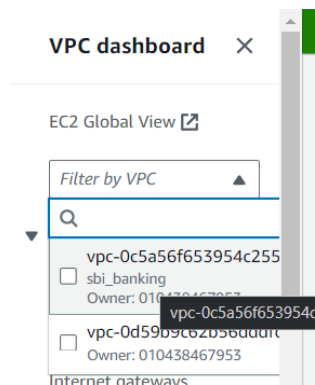
- 2) **Create VPC:** Click on "Create VPC".



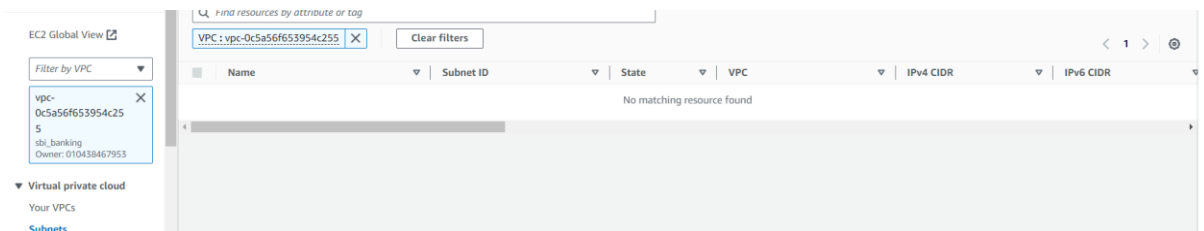
- 3) **Configure VPC:** Select "VPC only", enter a name, and specify the IPv4 CIDR block, then click "Create VPC".



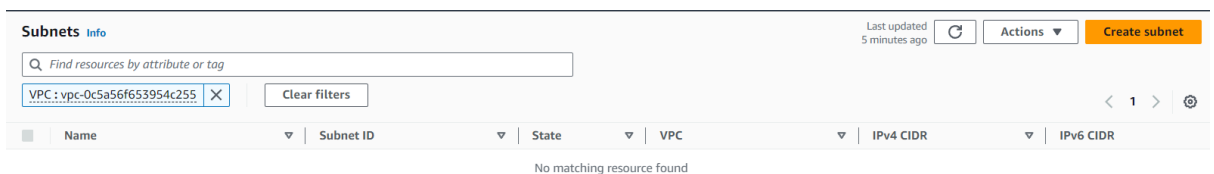
4) **Select VPC:** In the left-hand sidebar, select the VPC you just created.



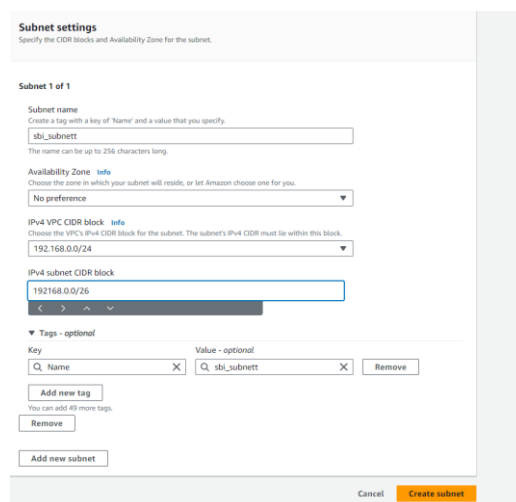
5) **Access Subnets:** In the left-hand sidebar, click on "Subnets".



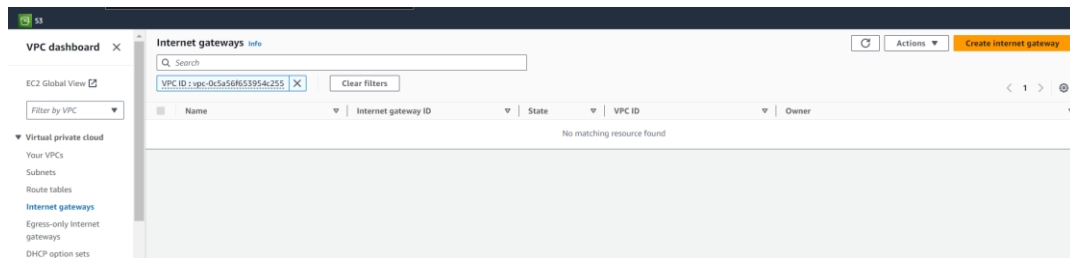
6) **Create Subnet:** Click on "Create Subnet", select the VPC ID of your created VPC.



7) **Configure Subnet:** Enter the CIDR block, select an availability zone, enter a name, and click "Create Subnet".



- 8) **Access Internet Gateway:** In the left-hand sidebar, click on "Internet Gateways" and then click "Create Internet Gateway".



- 9) **Name Internet Gateway:** Enter a name for the internet gateway and click "Create Internet Gateway".

VPC > Internet gateways > Create internet gateway

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional

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.

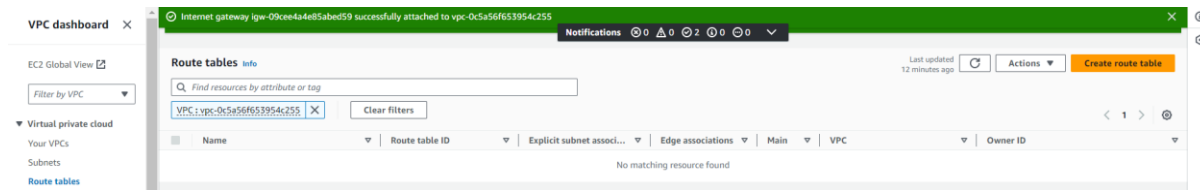
Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="my gateway"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

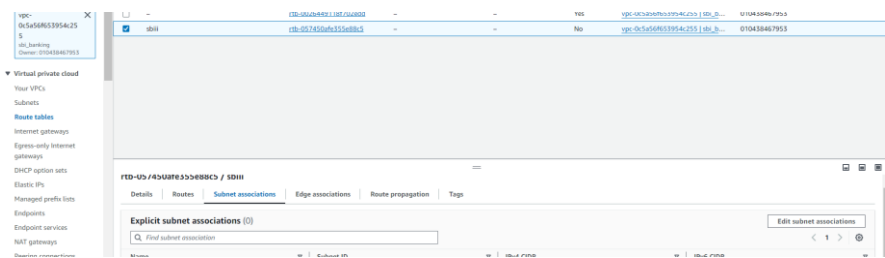
- 10) **Attach Internet Gateway:** Click on "Attach to VPC", select your VPC, and click "Attach Internet Gateway".



- 11) **Access Route Tables:** In the left-hand sidebar, select "Route Tables" and click "Create Route Table".

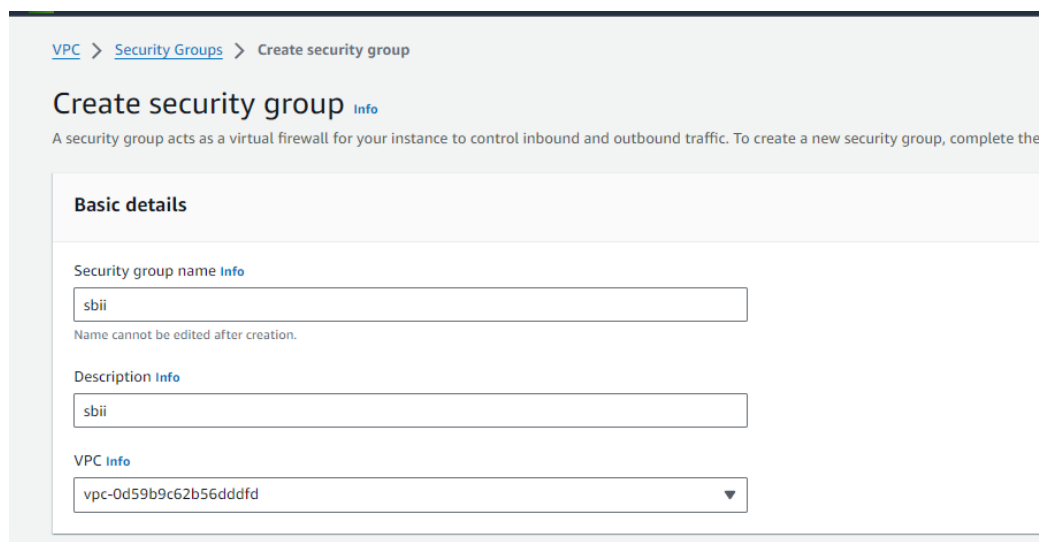


- 12) **Configure Route Table:** Enter a name, select your VPC, and click "Create Route Table".



- 13) **Edit Route Table:** In the route table, click on "Subnet Associations", then "Edit subnet associations", select your subnet, and click "Save".

- 14) **Access Security Groups:** In the left-hand sidebar, click on "Security Groups" and click "Create Security Group".



- 15) **Configure Security Group:** Enter a name, select your VPC, and click "Create Security Group".

- 16) **Add Rules:** Add a rule for "All Traffic" and select "Custom" for the source security group; add another rule for "RDP" with the source set to "Anywhere", then save the rules.

- 17) **Launch Instances:** Create multiple Windows instances, selecting your VPC, enabling automatic IP assignment, and choosing the existing security group, then click "Launch Instance".

- 18) **Name Instances:** Assign different names to the instances, ensuring one is named "Server".

- 19) **Connect to Instances:** Connect to the instances to access the machines.

- 20) **Adjust Firewall Settings:** Turn off firewall settings on the instances.

- 21) **Ping Instances:** Ping the IP addresses of all instances to ensure connectivity.

- 22) **Set Up Web Server:** On one instance, search for "Server Manager", select "Add Features and Roles", then proceed through the prompts to select "Web Server (IIS)"

and click "Next" until done.

- 23) **Access Server IP:** Use the server's IP address to access the website from different PCs or instances.

14-Aug-2024

Internship Day - 22 Report:

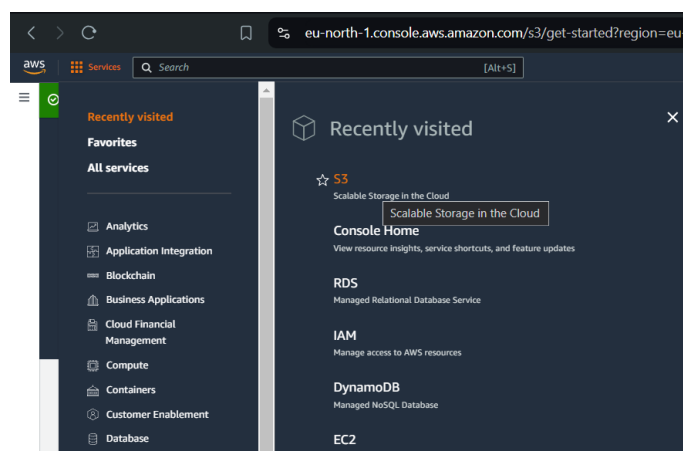
What is an S3 Bucket in AWS?

- **Definition:** An S3 bucket is a container used to store objects in Amazon Simple Storage Service (S3), which is a scalable cloud storage solution.
- **Purpose:** Buckets serve as a repository for data, allowing users to store, retrieve, and manage various types of files such as documents, images, and videos.
- **Unique Naming:** Each S3 bucket must have a globally unique name, ensuring that no two buckets across all AWS accounts can share the same name.
- **Access Control:** Users can configure access permissions for their buckets using policies and access control lists (ACLs), providing security and control over who can access the data.
- **Features:** S3 buckets support features like versioning, lifecycle management, and different storage classes to optimize cost and performance based on usage needs.
- **Use Cases:** Common use cases for S3 buckets include data backup, content distribution, and hosting static websites, making them versatile for various applications in the cloud.

Steps to Create an S3 Bucket for Static Website Hosting and Link with Route 53 in AWS

Step 1: Create an S3 Bucket

- 1) **Search for S3:** In the AWS Management Console, search for "S3" and select it.



- 2) **Create a Bucket:** Click on "Create bucket".

- 3) **Enter Bucket Name:** Provide a unique bucket name (e.g., school-website) which should match your domain name.

Create bucket [info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type [info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [info](#)

gne

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional
Only this bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

- 4) **Configure ACLs:** Under "Block Public Access settings for this bucket", uncheck "Block all public access".

Block Public Access settings for this bucket

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 this bucket and its 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 this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

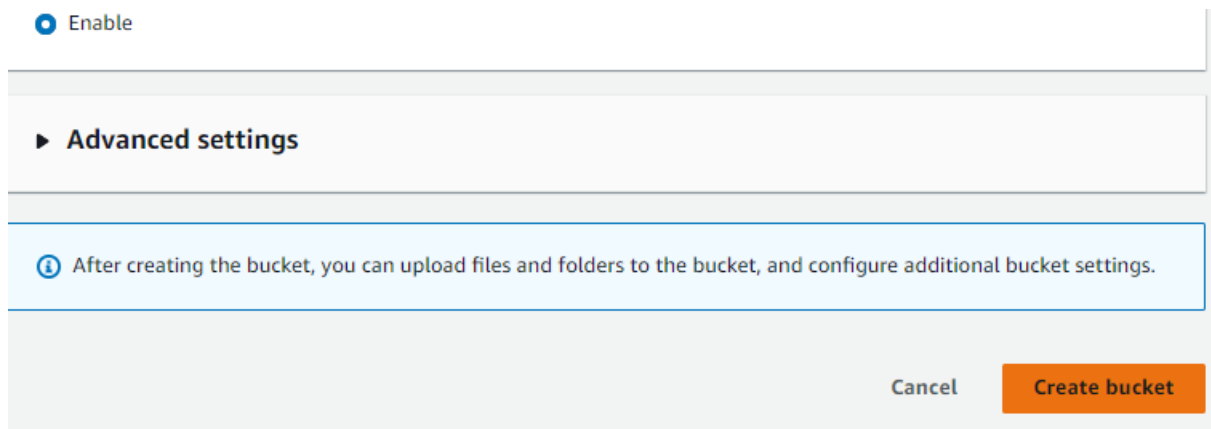
☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

- 5) **Acknowledge Settings:** Check the acknowledgment box and click "Create bucket".



Step 2: Enable Static Website Hosting

- 1) **Access Bucket Properties:** Click on the bucket you just created to open its settings.
- 2) **Scroll to Properties:** Navigate to the "Properties" tab.



- 3) **Enable Static Website Hosting:** Scroll down and click the "Edit" button.
 - Configure Hosting Settings:
 - Select "Enable" for Static Website Hosting
 - Enter the main file name (e.g., index.html).
- 4) If applicable, enter an error file name (e.g., error.html).
- 5) **Save Changes:** Click "Save changes".



Step 3: Set Bucket Policy for Public Access

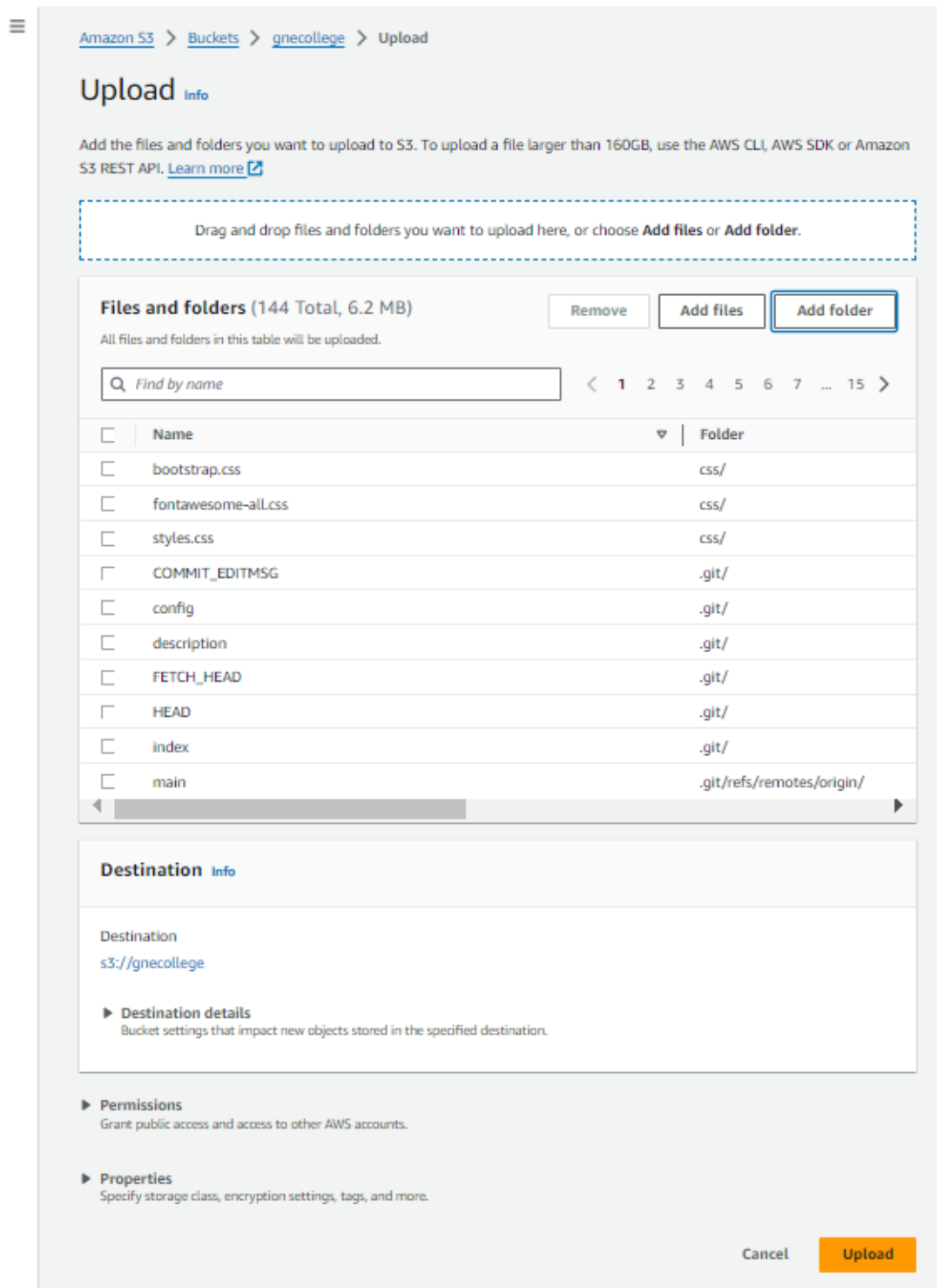
- 1) **Go to Permissions:** Click on the "Permissions" tab.
- 2) **Edit Bucket Policy:** Click on "Bucket Policy" and edit the policy.
- 3) **Copy and Modify Policy:** Use the following policy template, replacing Bucket-Name with your actual bucket name:

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "PublicReadGetObject",
6       "Effect": "Allow",
7       "Principal": "*",
8       "Action": [
9         "s3:GetObject"
10      ],
11      "Resource": [
12        "arn:aws:s3:::school-website/*"
13      ]
14    }
15  ]
16 }
```

4) **Save Changes:** Click "Save changes".

Step 4: Upload Website Files

- 1) **Upload Files:** Click on the "Upload" button in the bucket.
- 2) **Select Files:** Upload all your website files and folders one by one.



3) **Complete Upload:** After selecting all files, click the "Upload" button.

Step 5: Access Your Website

- 1) **Copy Website Link:** Go to the "Properties" tab, scroll down to the "Static website hosting" section, and copy the endpoint link.
- 2) **Test Your Website:** Paste the link in your browser to ensure the website is online.

Step 6: Configure Route 53 for Domain

1. **Search for Route 53:** In the AWS Management Console, search for "Route 53" and select it.
2. **Create Hosted Zone:** Click on "Hosted zones" and then "Create hosted zone".
3. **Enter Domain Name:** Provide your domain name (e.g., example.com) and click "Create hosted zone".
4. **Create Record:** Click on "Create record".
5. **Configure Alias Settings:**
 - Set "Alias" to "Yes".
 - Choose the "Endpoint" as "S3 Website".
 - Select the S3 website you created earlier.
6. **Save Changes:** Click "Create records" to save the settings.

16-Aug-2024

Internship Day - 23 Report:

Introduction

AWS Identity and Access Management (IAM) allows you to manage access to AWS services and resources securely. With IAM, you can create multiple users and groups, assign different permissions, and ensure that users have the access they need to perform their jobs while following the principle of least privilege. This document outlines the steps to create IAM users, groups, and policies to manage permissions effectively.

Why Use IAM?

- **Security:** IAM helps to secure your AWS environment by allowing you to control who can access your resources.
- **Granular Permissions:** You can assign specific permissions to users or groups based on their job roles.
- **Management:** Easily manage access for multiple users and groups from a single console.
- **Audit and Compliance:** IAM provides features to monitor user activity and maintain compliance with security policies.

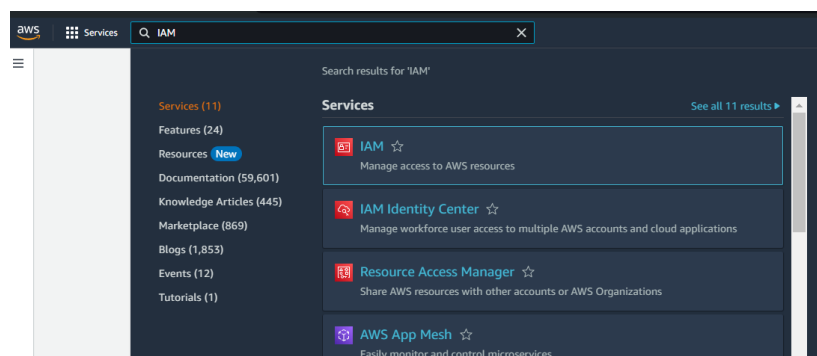
Steps to Create IAM Users and Groups

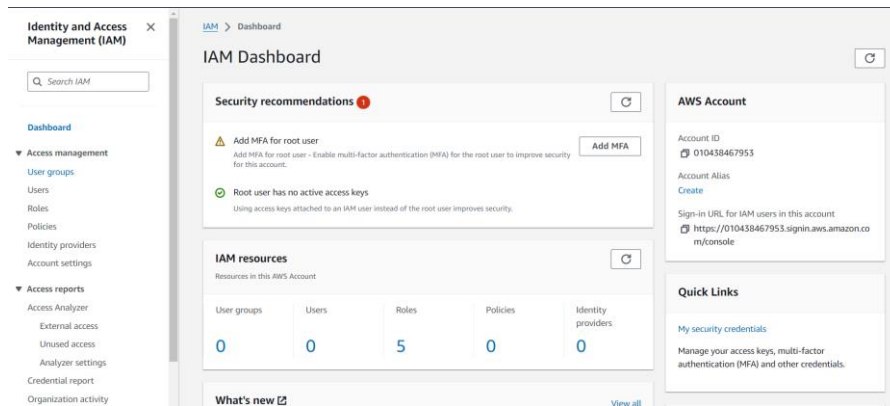
Step 1: Sign in to the AWS Management Console

1. **Log In:** Sign in to the AWS Management Console using your root user credentials.

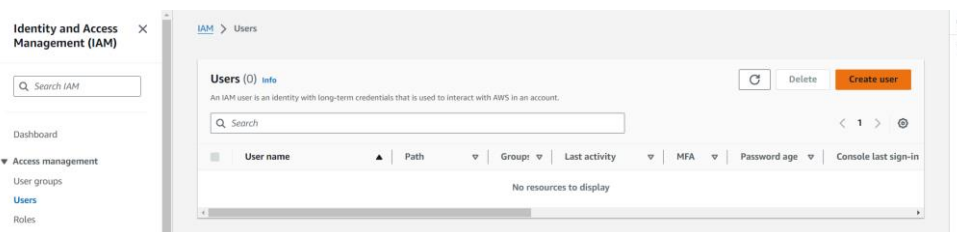
Step 2: Create a Normal User

1. **Access IAM:** In the AWS Management Console, search for "IAM" and select it.

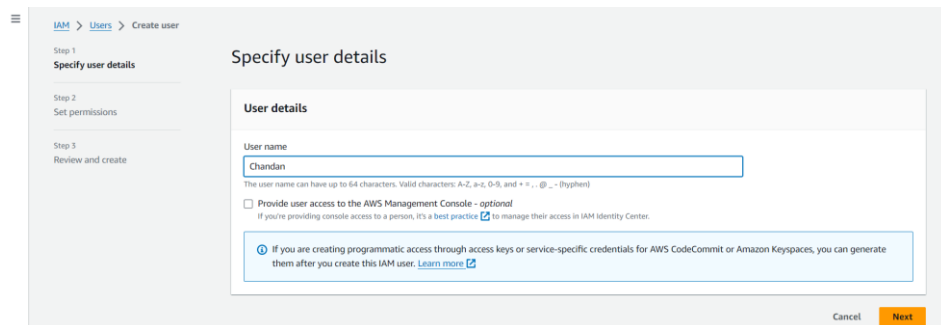




- 2. Create User:** Click on "Users" in the left sidebar and then click the "Add user" button.

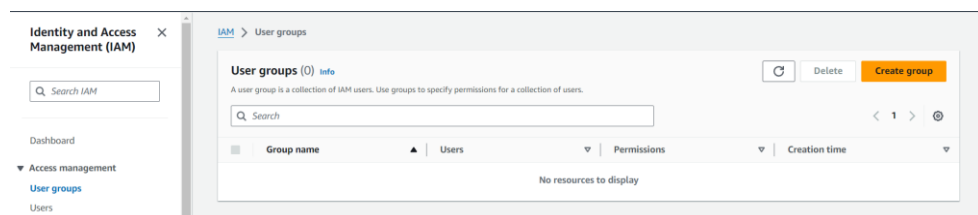


- 3. Enter User Name:** Input a username for the new user and click the "Next: Permissions" button.



Step 3: Create a User Group

- 1. Create Group:** Click on the "Create group" button to create a new group.



2. **Enter Group Name:** Provide a name for the group (e.g., "RDS-Users") and click "Create group".

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

IAM > User groups > Create user group

Create user group

Name the group

User group name
Enter a meaningful name to identify this group.

RDS_Team

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

Step 4: Assign User to Group

1. **Select Group:** In the "Set permissions" step, select the group you just created.

IAM > Users > Create user

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

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.

User groups (1/1)

Search

Group name	Users	Attached policies	Created
RDS_Team	0	-	2024-09-06 (1 minute ago)

Set permissions boundary - optional

Cancel Previous **Next**

2. **Click Next:** Click the "Next: Tags" button.

Step 5: Review and Create User

1. **Review Settings:** Review the user details and permissions.

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

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 Chandan	Console password type None	Require password reset No
----------------------	-------------------------------	------------------------------

Permissions summary

Name	Type	Used as
RDS_Team	Group	Permissions group

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

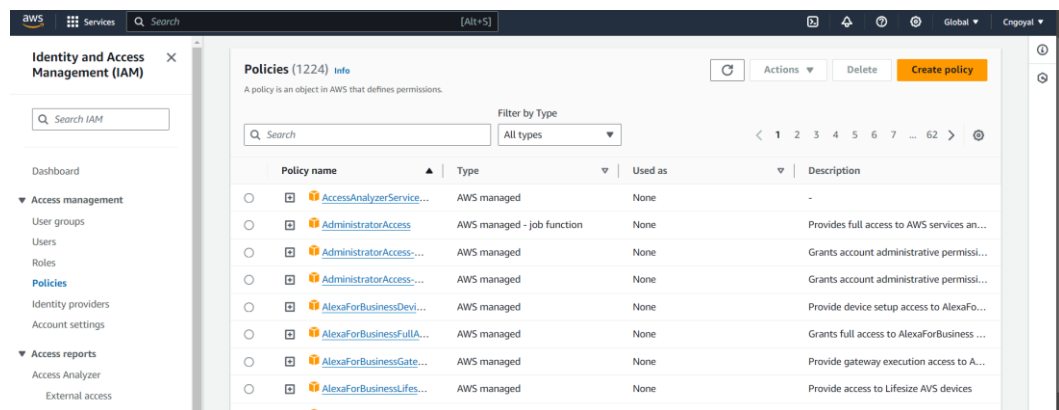
You can add up to 50 more tags.

Cancel Previous **Create user**

2. **Create User:** Click the "Create user" button to finalize the process.

Step 6: Create a Policy

1. **Access Policies:** In the IAM dashboard, click on "Policies" in the left sidebar.



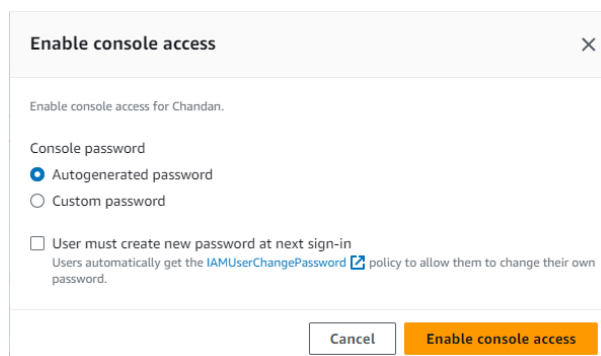
2. **Create Policy:** Click the "Create policy" button.
3. **Select Services:** Choose the services you want to grant permissions for (e.g., select "RDS" to give RDS permissions).
4. **Set Permissions:** Choose "All actions" to grant all permissions for the selected service.
5. **Next and Review:** Click "Next: Tags", then "Next: Review".
6. **Create Policy:** Provide a name for the policy (e.g., "RDS-Full-Access") and click "Create policy".

Step 7: Attach Policy to User Group

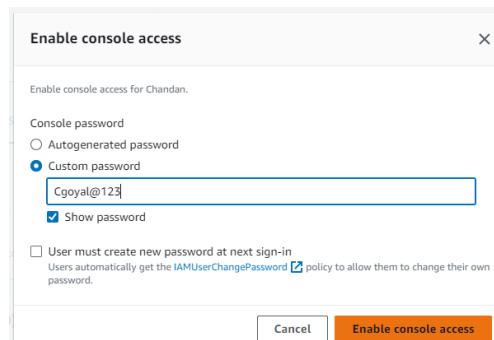
1. **Go to User Groups:** Return to the "User groups" section in IAM.
2. **Select Group:** Click on the group you created earlier.
3. **Attach Policy:** Click on the "Permissions" tab and then "Attach policies".
4. **Search and Attach:** Search for the policy you created (e.g., "RDS-Full-Access") and attach it to the group.

Step 8: Enable Console Access for Users

1. **Return to Users:** Go back to the "Users" section.
2. **Select User:** Click on the username of the user you created.
3. **Enable Console Access:** Click on the "Security credentials" tab and select "Manage" next to "Console password".



4. **Set Custom Password:** Choose "Custom password" and enter a password for the user. Enable console access.



The screenshot shows a dialog box titled "Enable console access" with a close button (X) in the top right corner. The text "Enable console access for Chandan." is displayed. Under the "Console password" section, there are two radio buttons: "Autogenerated password" (unselected) and "Custom password" (selected). Below the "Custom password" radio button is a text input field containing the password "Cgoyal@123". A checkbox labeled "Show password" is checked. At the bottom, there is an unchecked checkbox labeled "User must create new password at next sign-in" with a subtext: "Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password." At the bottom right, there are two buttons: "Cancel" and "Enable console access".

5. **Save Changes:** Click "Save changes".

Step 9: Share Access Details

1. **Copy Access Link:** Copy the link provided for console access.
2. **Send to Users:** Share the link along with the username and password with the user(s) for access.