|  |  |
| --- | --- |
| Subject: Skill base Lab Course:  Cloud Computing | Course ID: CSL605 |
| Semester: VI | Course: AI & DS |
| Laboratory: 101 | Name of teacher: Prof. Devita Ghanekar |
| Name of Student: Ayush Gupta | Roll No: VU2S2324001 |

**EXPERIMENT NO. 2**

**Aim:**  
To study and Implement Identity and Access Management (IAM) practices on AWS/Azure cloud.

**Hardware:**

* **Device Name:** computer-ThinkCentre
* **Hardware Model:** Lenovo ThinkCentre neo 50t Gen 3
* **Memory:** 8.0 Gib
* **Processor:** 12th Gen Intel® Core™ i5-12400 × 12
* **Disk Capacity:** 256.1 GB

**Software:**

* **Operating System:** LINUX
* **Amazon AWS**

**Theory:**  
The objective is to understand the working of Identity and Access Management IAM in cloud computing and to demonstrate the case study based on Identity and Access Management (IAM) on AWS/Azure cloud platform.

Create a user from AWS Management Console Editor's note: AWS uses the names AWS Management Console and AWS Console interchangeably. To create a user in AWS, we fill out a form and receive an access ID and secret key. At this step, we create a user named cli-user with full access permissions and programmatical access. This user is how we will manage other users later.

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

**Algorithm:**

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which

AWS resources users can access. You use IAM to control **who is authenticated (signed in) and authorized (has permissions) to use resources**.

**Step 1:** Create a user from AWS Management Console Editor's note: AWS uses the names AWS

Management Console and AWS Console interchangeably.

**Step 2:** In AWS, we fill out a form and receive an access ID and secret key.

**Step 3:** At this step, we create a user named cli-user with full access permissions and

programmatical access. This user is how we will manage other users later.

**Step 4:** Open the IAM console of the AWS Console in a browser window.

* Sign in. Sign in as a root user. Provide username and password when prompted.

**Step 5:** Select the Users menu.

* Navigate to the Users screen. You'll find it in the IAM dashboard, under the Identity and Access Management (IAM) drop-down menu on the left side of the screen. Click on Users.

**Step 6:** Add a user.

* Click on Add User to navigate to a user detail form.
* Provide all details, such as the username and access type.
* In this tutorial, we use the name cli-user, and check the Programmatic access box under Access type.
* This option gives the user access to AWS development tools, such as the command line interface used later in this tutorial.
* Click on Next: Permissions to continue.

**Step 7:** Set the user permissions.

* Click Attach existing policies directly and then filter the policies by keyword: IAM.
* For this user, select IAMFullAccess from the list of available policies.
* The IAMFullAccess policy enables this user to create and manage user permissions in AWS.
* Later in the tutorial, this user will perform AWS IAM operations.

**Step 8:** Finish the user setup.

* For this tutorial, we will skip the tags section of user creation and go to the review page.
* Check the details of the username, AWS access type and permissions.
* Then, click Create user.

**Step 9:** Create the user after verifying the name, access type and permissions are correct.

**Step 10:** At this point in the tutorial, the user cli-user exists, with the chosen policies applied to the account.

* AWS provides this user an access key ID and secret access key.
* Download or copy these keys to a secure place to use later in this tutorial.

**Step 11:** Set up AWS user credentials in the CLI.

* Open source AWS CLI tool, available through the cloud provider can be used.
* AWS CLI enables an admin to use their favourite shell or CLI to interact with AWS services.
* You can choose any Linux distribution or shell. A Bash shell running on an Ubuntu Linux distribution can be used.

**Step 12:** Install the command line tool.

* First, install AWS CLI on your system using the following command in Bash terminal:  
  sudo apt install awscli

**Step 13:** Once the setup runs, verify the installation by checking the version:  
aws –version

**Step 14:** Configure the user with the keys.

* Run the aws configure command in the shell to quickly set up the access key ID and secret access key obtained from AWS when you created the new user in the IAM console.

**Step 15:** This step saves your credentials in a local file at path:

* ~/.aws/credentials
* Region and output format configs at path: ~/.aws/config

**Step 16:** Now that cli-user with programmatic access is set up, we can use that account to create other users and give them policy-based access through AWS CLI.

**Step 17:** The next two sections follow these steps.

* Create a user and assign permissions.
* To create a user using IAM, run the following command:  
  aws iam create-user --user-name prateek
* It creates a new user and shows the user details in the bash console.

**Step 18.1:** The command creates a user with the name Prateek and shows details, such as the creation date and user ID.

Suppose this user needs to manage EC2 services.

To grant this new user EC2 admin rights, start by listing which EC2 policies we can grant.

**Step 18.2:** Use the command:

* aws iam list-policies | grep EC2FullAccess
* Identify the appropriate policy for the user's access level.
* In this case, it is AmazonEC2FullAccess.
* Pass the Amazon Resource Name (ARN) to the following command in --policy-arn Parameter:

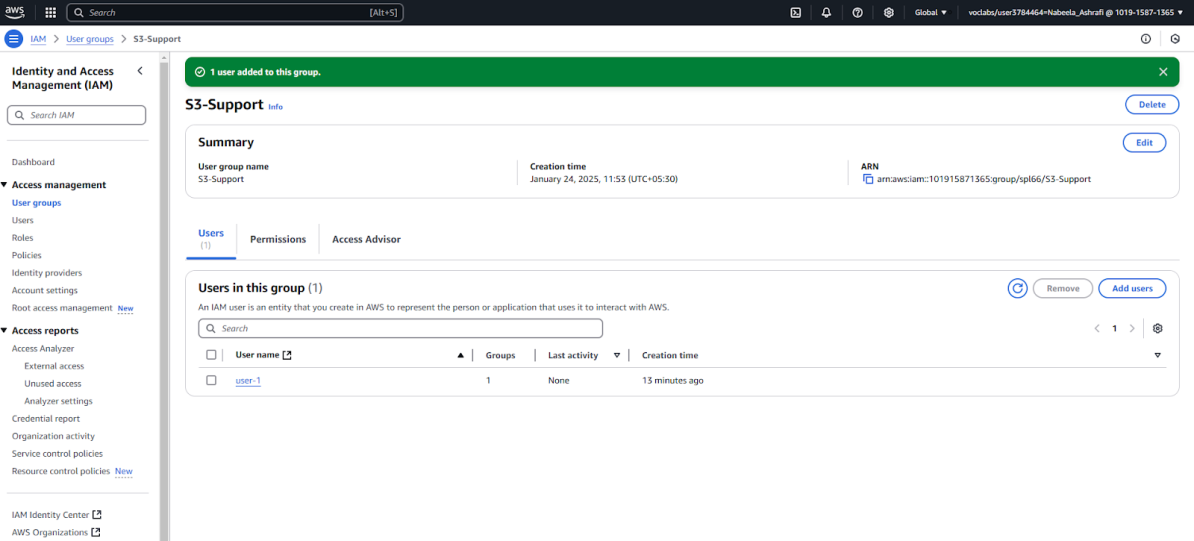
**Step 19:** Check user details and list user permissions.

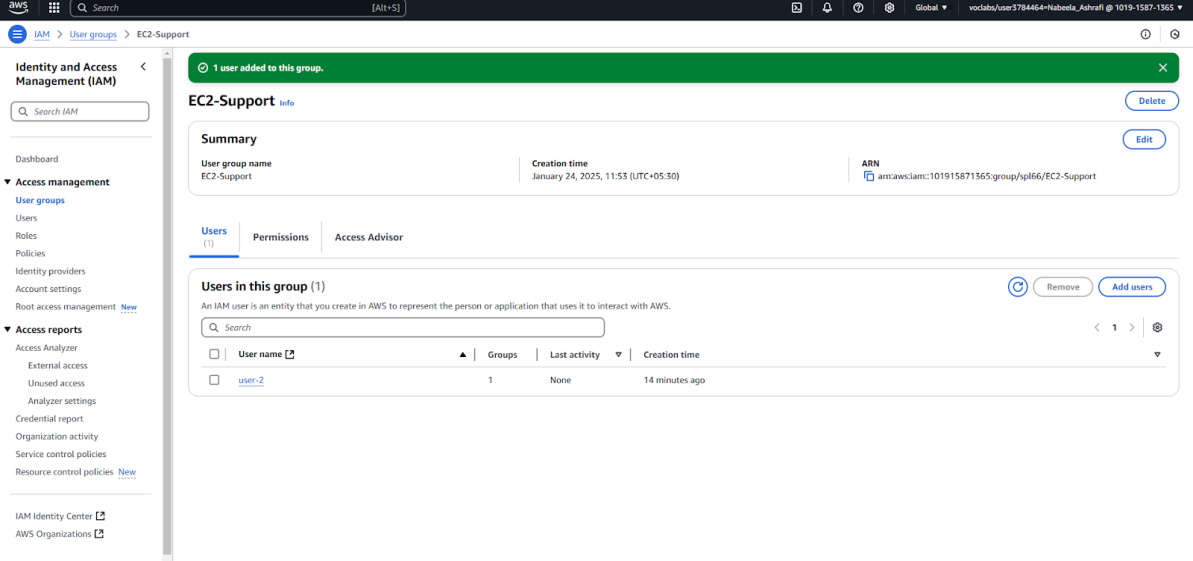
* Once you create the user and attach the appropriate user policy, verify that AWS assigned the appropriate policy by checking the user details.
* To check the list of IAM users, run:  
  aws iam list-users

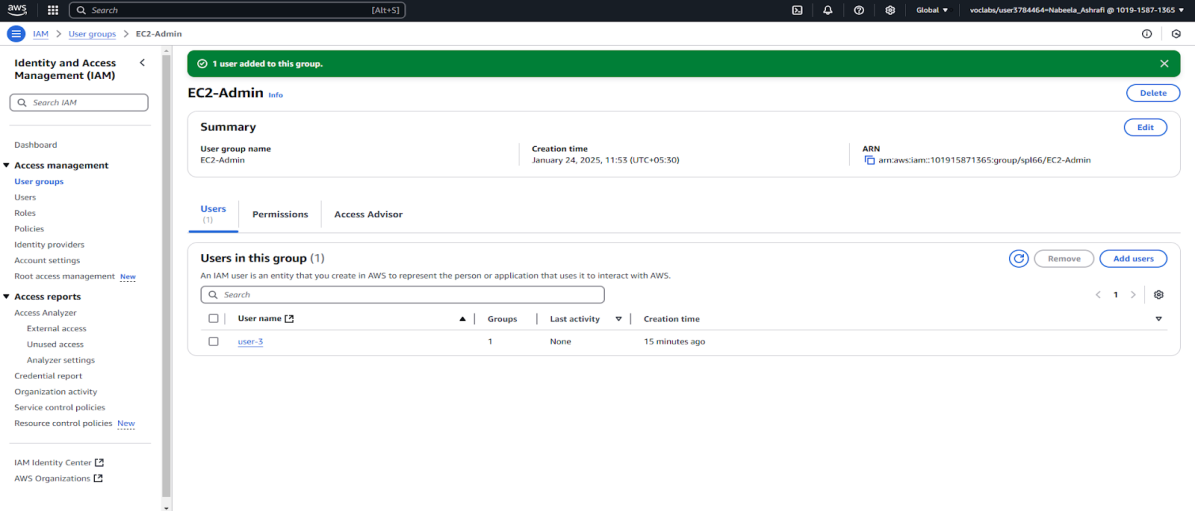
**Conclusion:**  
Thus, we have studied **Identity and Access Management (IAM) practices on AWS**.

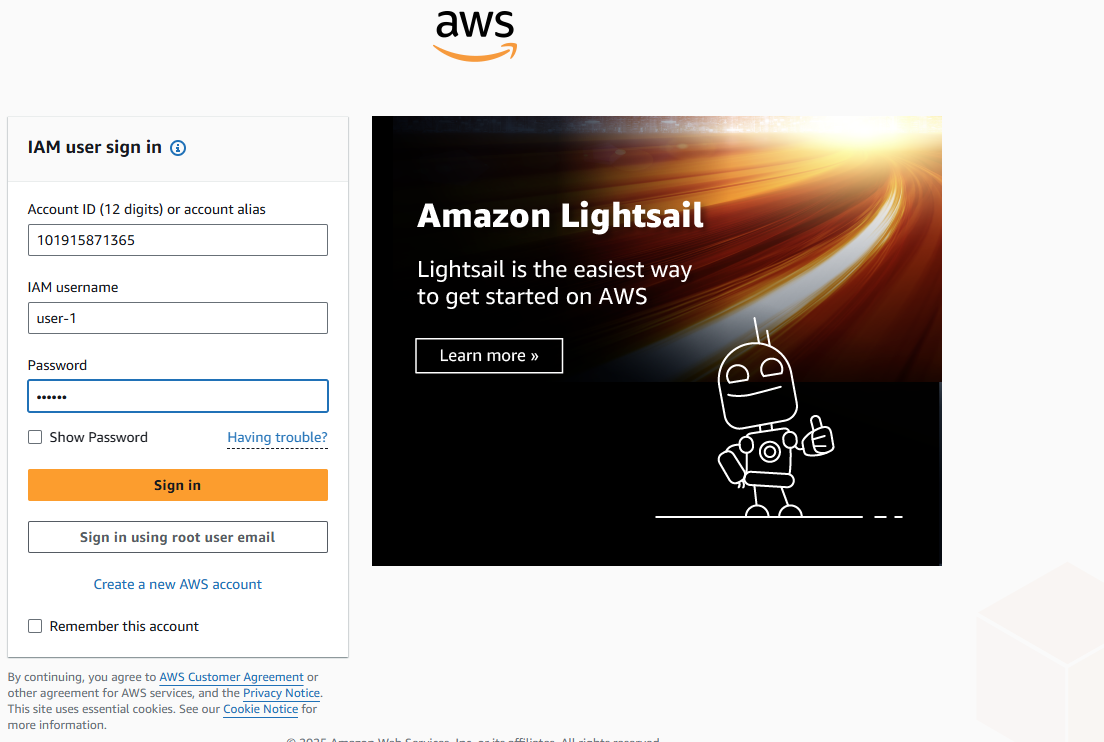


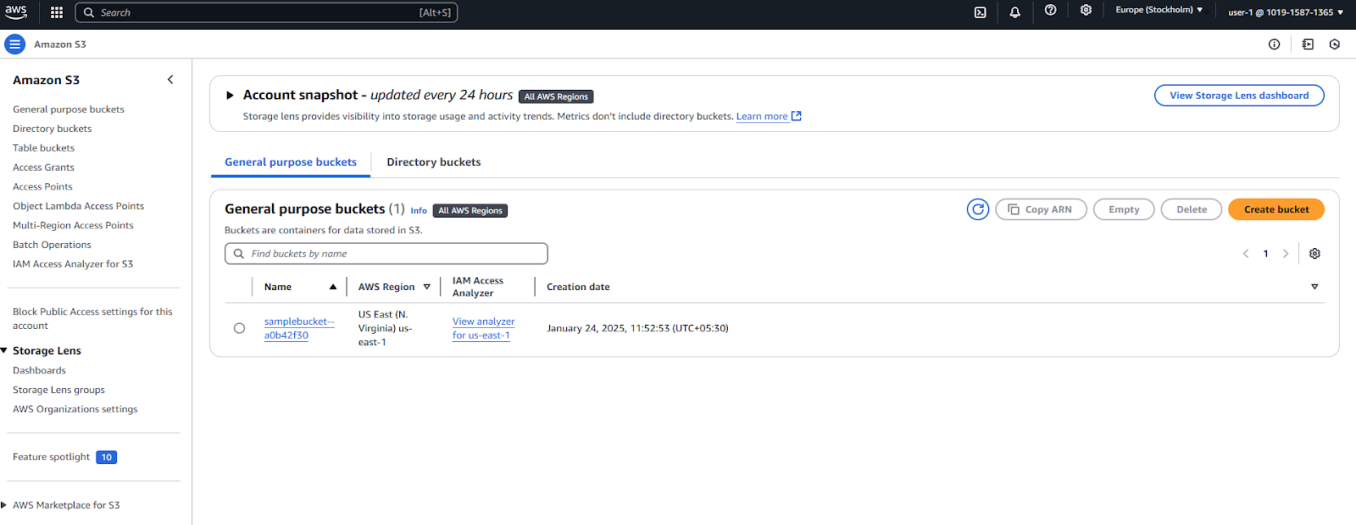
**Screenshots:**

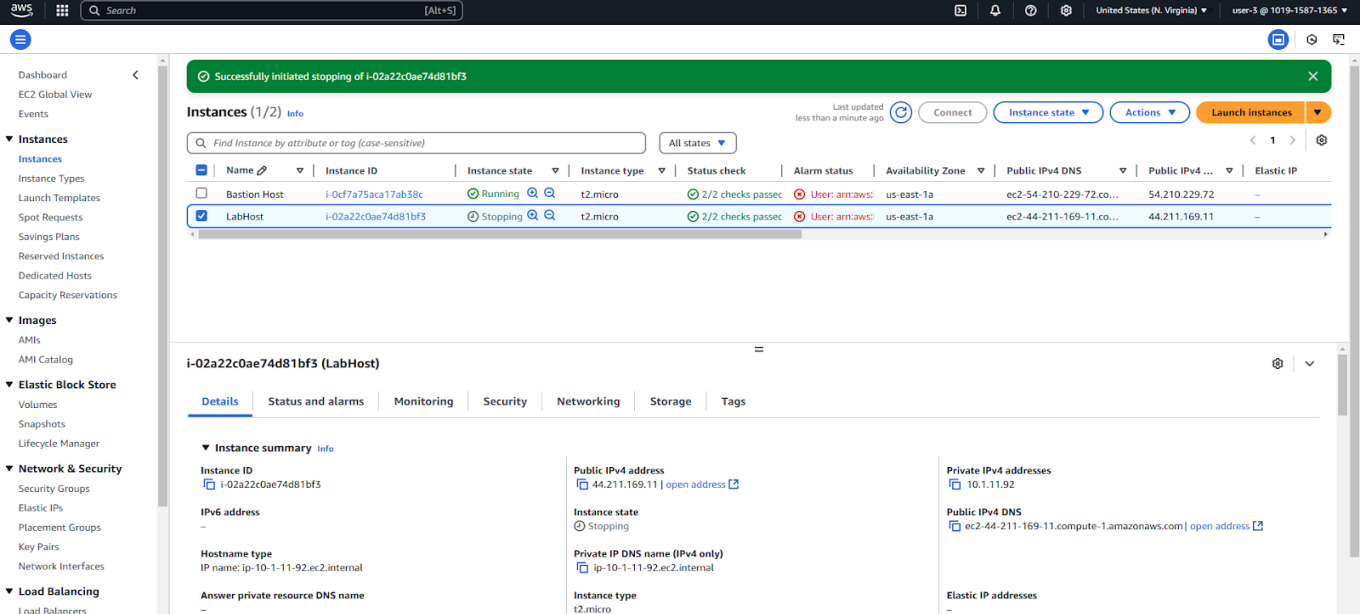






****

****

****