

INDEX

Sr.No.	Date	List of Experiment	Page	SIGN
1.		Navigate the AWS Management Console.	3-4	
2.		Create and manipulate Elastic Compute Cloud instances.	5-6	
3.		Create AWS EC2 Virtual Machine Using AWS Console.	7-9	
4.		Monitoring Virtual Resources in AWS.	10-11	
5.		Getting Started with S3 in Cloud.	12-13	
6.		Working with EBS in AWS.	14-16	
7.		Build a relational database server.	17-18	
8.		Create private cloud, Designing a Custom VPC (Virtual Private Cloud).	19-20	
9.		Create an IAM Group in Cloud.	21-24	
10.		Built a RESTful serverless API on AWS.	25-27	

Experiment 1

Aim: Navigate the AWS Management Console.

Task 1. To open AWS Management Console for a service:

Steps:

To open AWS Management Console for a service do one of the following:

In the search box on the navigation bar, enter all or part of the name of the service. Under Services, choose the service that you want from the list of search results.

Under Recently visited services, choose a service name. Under All services, choose a service name.

On the navigation bar, choose Services to open a full list of services.

Then choose a service under Recently visited or All services.

Task 2. To search for a service, feature, documentation, or AWS Marketplace product:

Steps:

In search box on the navigation bar of the AWS Management Console, enter all or part of your search terms. Do any of the following to refine your search and get more detail:

To narrow the results to the type of content that you want, choose one of the categories on the left. To see more results for a particular category, choose See all n results by each category heading.

To return to the main results list, choose Back in the top left corner.

To quickly navigate to popular features of a service, pause on the service name in the results and choose a link. To get more detail about a documentation or AWS Marketplace result, pause on the result title. Choose any link to navigate to your intended service, topic, or AWS Marketplace page.

Task 3. To launch AWS Cloudshell:

Steps:

You can launch AWS CloudShell from the AWS Management Console using either one of the following two methods:

Choose the AWS CloudShell icon on the console navigation bar.

Start typing "cloudshell" in the Find Services box and then choose the AWS CloudShell option.

Task 4. To get your billing information:

Steps:

On the navigation bar, choose your account name.

Choose Billing Dashboard.

Use the AWS Billing and Cost Management dashboard to find a summary and a breakdown of your monthly spending.

AWS

Services ▾ Resource Groups ▾ ★

Ireland ▾ Support ▾

AWS Management Console

AWS services


► All services

Build a solution

Get started with simple wizards and automated workflows.


Launch a virtual machine

With EC2
~2-3 minutes




Build a web app

With Elastic Beanstalk
~6 minutes




Build using virtual servers

With Lightsail
~1-2 minutes




Connect an IoT device

With AWS IoT
~5 minutes




Start a development project

With CodeStar
~5 minutes



Register a domain

With Route 53
~3 minutes




► See more

Learn to build

Learn to deploy your solutions through step-by-step guides, labs, and videos. [See all](#)


Websites and Web Apps

3 videos, 3 tutorials, 3 labs




Storage

3 videos, 3 tutorials, 3 labs




Databases

3 videos, 3 tutorials, 3 labs




DevOps

3 videos, 3 tutorials, 3 labs




Machine Learning

3 videos, 3 tutorials, 3 labs




Big Data

3 videos, 1 lab



[Build with SDKs](#)

Access resources on the go

 Access the Management Console using the AWS Console Mobile App. [Learn more](#)

Explore AWS

Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. [Learn more](#)

Run Serverless Containers with AWS Fargate

AWS Fargate runs and scales your containers without having to manage servers or clusters. [Learn more](#)


Scalable, Durable, Secure Backup & Restore with Amazon S3

Discover how customers are building backup & restore solutions on AWS that save money. [Learn more](#)

AWS Marketplace

Find, buy, and deploy popular software products that run on AWS. [Learn more](#)

Have feedback?

 [Submit feedback](#) to tell us about your experience with the AWS Management Console.

Feedback English (US)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Experiment 2

Aim: Create and manipulate Elastic Compute Cloud Instances.

Steps:

Task 1. Choose an Amazon Machine Image (AMI)

Choose the Services menu, locate the Compute services, and select EC2.

Choose the Launch instance button in the middle of the page, and then select Launch instance from the dropdown menu.

Task 2. Choose an instance type

For this activity, we used a t2.micro instance. This instance type has 1 virtual central processing unit (CPU) and 1 GiB of memory.

Task 3. Configure instance details

This page configures the instance to meet your requirements. This includes networking and monitoring settings. Select Next: Add Storage.

Task 4. Add storage

Amazon EC2 stores data on a network-attached virtual disk called Amazon Elastic Block Store (Amazon EBS).

Task 5. Add tags

Tags help in categorizing the AWS resources in different ways; for example, by purpose, owner, or environment. Select Add Tag, and then configure:

Key:

Enter

Value:

Enter

Task 6. Configure the security group

Keep the default selection Create a new security group. Security group name: Clear the text

Description: Clear the text and enter

Task 7. Review the instance and launch

This is the final step to launch your EC2 instance. The Review page displays the configurations selected for the instance you are about the launch.

Select View

Instances. Wait for

the following:

Instance state: *Running*

Status check: *2/2 checks passed*

Task 8. Access your EC2 instance

From the Details tab, copy the Public IPv4 address value of your instance to the clipboard.

Open a new tab in your web browser, paste the public IP address you just copied, and press Enter. The webpage does not load. You must update the security group to be able to access the page.

Task 9. Update the security group

You are not able to access your web server because the security group is not permitting inbound traffic on port 80, which is used for HTTP web requests. In this task, you update the security group.

Return to the EC2 Management Console browser tab.

In the left navigation pane, under Network & Security, choose Security Groups.

Select the Web Server security group, which you created when launching your EC2 instance. In the lower pane, choose the Inbound rules tab.

Task 10. Create a rule

Create an inbound rule.

Choose Edit inbound rules, and then choose Add rule.

Configure the following:

Type: HTTP

Source:

Anywhere

Choose Save

rules

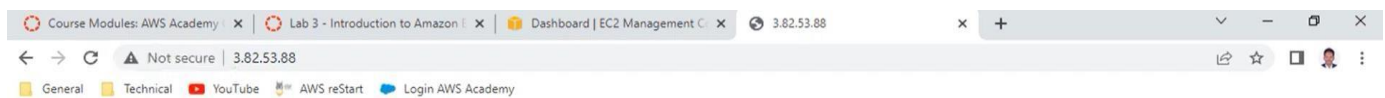
The new inbound HTTP rule creates an entry for IPv4 IP (0.0.0.0/0) and IPv6 IP addresses (:::/0).

Task 11. Test your rule

Return to the tab that you used to try to connect to the web server.

Refresh the page.

The page should display the message *Hello From Your Web Server!*



Experiment 3

Aim: Create AWS EC2 Virtual Machine using AWS Console.

Steps:

```
$ aws ec2 run-instances --image-id ami-173d747e --count 1 --instance-type t1.micro -  
-key-name MyKeyPair -- security-groups my-sg { c  
  "OwnerId": "123456789012",  
  "ReservationId": "r-5875ca20",  
  "Groups": [  
    {  
      "GroupName": "my-sg",  
      "GroupId": "sg-903004f8"  
    }  
  ],  
  "Instances": [  
    {  
      "Monitoring":  
      {  
        "State": "disabled"  
      },  
      "PublicDnsName": null,  
      "Platform": "windows",  
      "State": {  
        "Code": 0,  
        "Name": "pending"  
      },  
      "EbsOptimized": false,  
      "LaunchTime": "2013-07-19T02:42:39.000Z",  
      "ProductCodes": [],  
      "InstanceId": "i-5203422c",  
      "ImageId": "ami-173d747e",  
      "PrivateDnsName": null,  
      "KeyName": "MyKeyPair",  
      "SecurityGroups": [  
        {
```

```
"GroupName": "my-sg",

  "GroupId": "sg-903004f8"
},
"ClientToken": null,
"InstanceType": "t1.micro",
  "NetworkInterfaces": [],
"Placement": {
"Tenancy": "default",
  "GroupName": null,

  "AvailabilityZone": "us-west-2b"
},
"Hypervisor": "xen",
  "BlockDeviceMappings": [
    {
"DeviceName": "/dev/sda1",
  "Ebs": {
    "Status": "attached",
    "DeleteOnTermination": true,
    "VolumeId": "vol-877166c8",
    "AttachTime": "2013-07-19T02:42:39.000Z"
  }
    },
  ],
"Architecture": "x86_64",
  "StateReason":
{
  "Message": "pending",
  "Code": "pending"
},
"RootDeviceName": "/dev/sda1",
  "VirtualizationType": "hvm",
  "RootDeviceType": "ebs",
  "Tags": [
    {
"Value": "MyInstance",
```

```

"Key": "Name"
}
],
"AmiLaunchIndex": 0

}

]
}

```

The screenshot shows the AWS Management Console interface. On the left, there's a navigation menu with options like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images', and 'AMIs'. The main area displays the 'Instances (1/1)' page. At the top, there's a search bar and buttons for 'Connect', 'Instance state', 'Actions', and 'Launch Instances'. Below this is a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 D. The table shows one instance, 'My First Instance', with ID 'i-051664a7b9b8959cb', in a 'Running' state, of type 't2.micro', with no alarms, in the 'ap-south-1a' availability zone, and public IP 'ec2-13-232-104-235.ap-south-1.compute.amazonaws.com'. Below the table, the details for 'My First Instance' are shown, including its public IP address (13.232.104.235), private IP address (172.31.46.149), instance state (Running), and instance type (t2.micro). The console also shows the 'Instance summary' section with details like 'Instance ID', 'Public IPv4 address', 'Private IPv4 addresses', 'Instance state', 'Public IPv4 DNS', 'Private IP DNS name (IPv4 only)', 'Hostname type', and 'IP name'.

Experiment 4

Aim: Monitoring Virtual Resources in AWS.

Steps:

Task 1. Create and subscribe to an SNS topic

Choose the Services menu, locate the Application Integration section, and choose Simple Notification Service. In the left navigation pane, select Topics.

Select Create topic.

For Type, select Standard.

For Name, enter `MoneyAlert`

Choose Create topic.

In the Subscriptions section, choose Create subscription. For Topic ARN, select the `MoneyAlert` topic you created. For Protocol, select Email to get an email alert.

For Endpoint, enter your email address or phone number, depending on what you selected for Protocol. Select Create subscription.

Task 2. Create a CloudWatch alarm

Now you will create the billing alert.

Choose the Services menu, locate the Management & Governance section, and choose CloudWatch. In the left navigation pane, select Alarms.

Choose Create

alarm. Choose

Select metric.

Select Billing.

Select Total Estimated Charge.

Check the box for

EstimatedCharges. Choose Select metric.

In the Notification section, configure the following: For Alarm state trigger, choose In alarm.

For Select an SNS topic, choose Select an existing SNS topic.

For Send a notification to..., choose the `MoneyAlert` topic.

For Alarm name, enter `MoneyAlertAlarm`

Review the settings. Scroll down and choose Create alarm.

Congratulations! You have set up a CloudWatch alarm that uses Amazon SNS to send an alert when the cost of services reaches \$100.

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user2721088=FAIZ_UL_HAQ @ 6038-5395-9892

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

Security Groups

Elastic IPs

Placement Groups

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

Alarm recommendations

1h3h12h1d3d1w

UTC timezone

Manage detailed monitoring

Add to dashboard

CPU utilization (%)

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Status check failed (any)...

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Status check failed...

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Status check failed (syst...

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Network in (bytes)

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Network out (bytes)

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Network packets in (count)

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Network packets out (co...

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Disk reads (bytes)

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Disk read operations (op...

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Disk writes (bytes)

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

Disk write operations (o...

No unitNo data available.
Try adjusting the dashboard time range.

0.5

017:2518:20

CPU credit usage (count)

CPU credit balance (count)

CloudShell

Feedback

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

11

Experiment 5

Aim: Getting started with S3 in Cloud.

Steps:

Task 1. Create an S3 bucket

Choose the Services menu, locate the Storage services, and select S3. Select Create bucket on the right side of the page.

For Bucket name, enter a unique Domain Name System (DNS)-compliant name for your new bucket.

For Region, choose the AWS Region where you want the bucket to reside.

The new bucket appears in the Buckets list.

Task 2. Add a bucket policy to make the content publicly available

Choose the link for your bucket's name, and then select the Permissions tab and make it public.

Task 3. Upload an HTML document

In the console, choose the Objects tab. Upload the html file to your bucket.

Choose Upload.

Drag and drop the index.html file onto the upload page. Expand the Permissions section.

Under Predefined ACLs, select Grant public-read access.

Expand the Properties section.

Ensure that the Standard storage class is selected. At the bottom of the page, choose Upload.

Choose Close.

The html file appears in the Objects list.

Task 4. Test your website

Select the Properties tab, and scroll down to the Static website hosting section.

Choose Edit.

Select Enable.

In the Index document text box, enter `html file name`

Select "Save changes"

Scroll down to the Static website hosting section again, and copy the Bucket website endpoint URL to your clipboard.

Open a new tab in your web browser, paste the URL you just copied, and press Enter.

The Hello World webpage should display.



Experiment 6

Aim: Working with EBS in AWS.

Steps:

Task 1. Choose an Amazon Machine Image (AMI) and an instance type

Choose the Services menu, locate the Compute services, and select EC2.

Choose the Launch instance button in the middle of the page, and then select Launch instance from the dropdown menu.

To the right of the Amazon Linux 2 AMI (HVM), SSD Volume Type AMI name, choose Select.

Task 2. Configure instance details

Scroll down, and expand Advanced Details. A field for User data appears.

write the following code and paste it into the User data field.



```
#!/bin/bash
yum update
-y
yum -y install httpd
systemctl enable
httpd systemctl
start httpd
echo '<html><h1>Hello World!</h1></html>' > /var/www/html/index.html
```

This script does the following:

Updates the server

Installs an Apache web server (httpd)

Configures the web server to automatically start on

boot Activates the web server

Creates a simple webpage

Select Next: Add Storage.

Task 3. Add storage and tags

Amazon EC2 stores data on a network-attached virtual disk called Amazon Elastic Block Store (Amazon EBS).

You will launch the Amazon EC2 instance using a default 8 GiB disk volume.

Choose Next: Add Tags.

Select Add Tag, and then configure:

KeyName

ValueWeb Server

Select Next: Configure Security Group.

Task 4. Configure the security group

Configure a new security group:

Keep the default selection Create a new security group. Select Review and Launch.

Task 5. Review the instance and launch

This is the final step to launch your EC2 instance.

When you are finished reviewing the launch details, select Launch.

Select Launch Instances.

Select View Instances.

The instance will appear in the *Pending* state, which means it is being launched. It will then change to *Running*, which indicates that the instance has started booting.

Task 6. Access your EC2 instance

When you launched your EC2 instance, you provided a script that installed a web server and created a simple webpage. In this task, you will try to access the content from the web server.

Select the Web Server instance, choose the Details tab, and copy the Public IPv4 address value of your instance to your clipboard.

Open a new tab in your web browser, paste the public IP address you just copied, and press Enter. The webpage does not load. You must update the security group to be able to access the page.

Task 7. Update the security group

Return to the EC2 Management Console browser tab.

In the left navigation pane, under Network & Security, choose Security Groups.

Select the Web Server security group, which you created when launching your EC2 instance. In the lower pane, choose the Inbound rules tab.

Create an inbound rule.

Choose Edit inbound rules, and then choose Add rule.

Configure the following:

Type: HTTP

Source:

Anywhere

Choose Save

rules

Return to the tab that you used to try to connect to the web server, and refresh the page. The page should display the message *Hello World!*

Task 8. Attach an EBS volume to your EC2 instance

Return to the EC2 Management Console browser tab.

In the left navigation pane, under Elastic Block Store, select Volumes. Select Create Volume.

For Size, enter 1 to create a volume with 1 GiB.

For Availability Zone, select the same Availability Zone that your EC2 instance is running in.

Select Create Volume.

Select Close.

The new volume appears in the volumes list with a state of *available*. Select the new volume. Then, choose Actions, and Attach Volume.

Select the Instance drop-down menu, and then select your EC2 instance. The device will automatically populate. Select Attach.

The state of the volume changes to *in-use*. The new volume is now attached to your EC2 instance.

The screenshot shows the AWS Academy lab interface. The browser tabs include 'Course Modules: AWS Academy', 'Lab 4 - Working with EBS', and 'Volumes | EC2 Management Console'. The URL is 'awsacademy.instructure.com/courses/3781/modules/items/462292'. The main content area is divided into a terminal window and a file explorer.

Terminal Window:

```
ec2-user@ip-10-1-11-154:~$  
ec2-user@ip-10-1-11-154:~$ sudo mkdir /mnt/data-store2  
ec2-user@ip-10-1-11-154:~$ sudo mount /dev/sdg /mnt/data-store2  
ec2-user@ip-10-1-11-154:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        484M   0  484M   0% /dev  
tmpfs           492M   0  492M   0% /dev/shm  
tmpfs           492M  416K  491M   1% /run  
tmpfs           492M   0  492M   0% /sys/fs/cgroup  
/dev/xvda1      8.0G  1.5G   6.5G  19% /  
tmpfs           99M    0   99M   0% /run/user/1000  
/dev/xvdf       975M   60K  924M   1% /mnt/data-store  
/dev/xvdg       975M   64K  924M   1% /mnt/data-store2  
ec2-user@ip-10-1-11-154:~$ ll /mnt/data-store2/  
total 20  
-rw-r--r-- 1 root root  27 Feb 22 07:45 file.txt  
drwx----- 2 root root 16384 Feb 22 07:44 lost+found  
ec2-user@ip-10-1-11-154:~$ cp /mnt/data-store2/file.txt /mnt/data-store  
cp: cannot create regular file '/mnt/data-store/file.txt': Permission denied  
ec2-user@ip-10-1-11-154:~$ sudo cp /mnt/data-store2/file.txt /mnt/data-store  
ec2-user@ip-10-1-11-154:~$ cat /mnt/data-store/file.txt  
some text has been written  
ec2-user@ip-10-1-11-154:~$
```

File Explorer:

The file explorer shows a list of files with a '1' next to 'file.txt'. A message says: '1 Choose a file to load from the Files menu on the left'.

Navigation buttons: 'Previous', 'Next', 'Show all'.

Experiment 7

Aim: Build a relational database server.

Steps:

Task 1: Create a Security Group for the RDS DB Instance

In the AWS Management Console, on the Services menu, choose VPC. In the left navigation pane, choose Security Groups.

Choose Create security group and then configure:

Security group name: DB Security Group

Description: Permit access from Web Security Group

VPC: *Lab VPC*

Task 2: Create a DB Subnet Group

On the Services menu, choose RDS.

In the left navigation pane, choose Subnet groups.

Name: DB-Subnet-Group

Description: DB Subnet Group

VPC: *Lab VPC*

Scroll down to the Add Subnets section.

Choose Create

Task 3: Create an Amazon RDS DB Instance

In the left navigation pane, choose Databases.

Choose Create database

Select MySQL.

Under Storage, configure:

Storage type: *General Purpose (SSD)*

Allocated storage: *20*

Under Connectivity, configure:

Virtual Private Cloud (VPC): *Lab VPC*

Under Existing VPC security groups, from the dropdown list:

Choose *DB Security Group*.

Choose Create database

Task 4: Interact with Your Database

Open a new web browser tab, paste the *WebServer* IP address and press Enter.

The web application will be displayed, showing information about the EC2 instance. Choose the RDS link at the top of the page.

Configure the following settings:

Endpoint: Paste the Endpoint you copied to a text editor earlier

Database: *lab*

Username: *main*

Password: lab-password

Choose Submit

The screenshot shows a web browser window with the following details:

- Browser Tabs:** "Lab 5 - Build a Database Server", "RDS | us-east-1", "AWS Technical Essentials v4.1".
- Address Bar:** "Not secure | 44.202.208.107/rds.php?mode=add".
- Page Header:** "aws Load Test RDS".
- Page Content:**
 - Address Book**
 - Add Contact**
 - Form:** Fields for "Last Name:", "First Name:", "Phone:", and "Email:". A "Submit" button is below the "Email" field.
 - Table:** A table with 5 columns: "Last name", "First name", "Phone", "Email", and "Admin". It contains 4 rows of contact data.

The Windows taskbar at the bottom shows the date and time as "20:26 25-09-2023".

Last name	First name	Phone	Email	Admin
Doe	Jane	010-110-1101	jane.d@someotheraddress.org	Edit Remove
Haq	Faizul	7753858243	imfaizulhaq@gmail.com	Edit Remove
Johnson	Roberto	123-456-7890	robertoj@someaddress.com	Edit Remove
Rai	Aman	89765132487	amanrai@gmail.com	Edit Remove

Experiment 8

Aim: Create a private cloud - Designing a custom VPC (Virtual Private Cloud).

Steps:

Task 1: Create your VPC

In the AWS Management Console, on the Service menu, choose VPC. Choose Launch VPC Wizard

VPC name: Lab VPC

Availability Zone: Select the *first* Availability Zone

Public subnet name: Public Subnet 1

Availability Zone: Select the *first* Availability Zone

Private subnet name: Private Subnet 1

Choose Create VPC

Task 2: Create Additional Subnets

In the left navigation pane, choose Subnets.

Choose create subnet then configure:

VPC ID: Lab VPC

Subnet name: Public Subnet

Availability Zone: Select the *second* Availability Zone

Task 3: Create a VPC Security Group

In the left navigation pane, choose Security Groups.

Choose Create security groups and then configure:

Security group name: Web Security Group

Description: Enable HTTP access

VPC: Lab VPC

In the Inbound rules pane, choose "Add Rule"

Configure the following settings:

Type: HTTP

Source: Anywhere-IPv4

Description: Permit web

Scroll to the bottom of the page and choose create security group

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user2721088=FAIZ_UL_HAQ @ 6038-5395-9892

You have successfully modified the settings for vpc-0e55fc42c928f8662 / My First VPC.

VPC > Your VPCs > vpc-0e55fc42c928f8662

vpc-0e55fc42c928f8662 / My First VPC

Actions

Details

Info

VPC ID

vpc-0e55fc42c928f8662

Tenancy

Default

Default VPC

No

Network Address Usage metrics

Enabled

State

Available

DHCP option set

dopt-0a82fe24638fd8b7d

IPv4 CIDR

10.0.0.0/17

Route 53 Resolver DNS Firewall rule groups

Failed to load rule groups

DNS hostnames

Enabled

Main route table

rtb-0835c4f4cd66abd13

IPv6 pool

-

Owner ID

603853959892

DNS resolution

Enabled

Main network ACL

acl-0a1b1495ce3c188c0

IPv6 CIDR (Network border group)

-

Resource map

New

CIDRs

Flow logs

Tags

Integrations

Resource map

Info

VPC

Show details

Your AWS virtual network

My First VPC

Subnets (2)

Subnets within this VPC

us-east-1d

subnet_1

subnet_2

Route tables (1)

Route network traffic to resources

rtb-0835c4f4cd66abd13

Was the resource map helpful

CloudShell

Feedback

© 2023, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Experiment 9

Aim: Create an IAM Group in Cloud.

Steps:

Task 1. Explore the users and groups

First, note the Region that you are in.

Choose the Services menu, locate the Security, Identity, & Compliance services, and choose IAM. In the navigation pane on the left, choose Users.

The following IAM users have already been created:

user-1

user-2

user-3

Choose the name of user-1.

This brings you to a summary page for user-1.

The Permissions tab will be displayed.

Notice that user-1 does not have any permission.

Choose the Groups tab.

Notice that user-1 also is not a member of any of the groups.

Choose the Security credentials tab.

Notice that user-1 is assigned a Console password.

This allows the user to access the AWS Management Console.

In the navigation panel on the left, choose User groups. The following groups have already been created:

EC2-Admin

EC2-Support

S3-Support

Choose the name of the EC2-Support group.

This brings you to the summary page for the EC2-Support group.

Choose the Permissions tab.

Under Policy Name, choose the link for the AmazonEC2ReadOnlyAccess policy.

Choose the {} JSON tab.

Statements in an IAM policy have the following basic

structure: Effect says whether to *Allow* or *Deny* the permissions.

Action specifies the API calls that can be made against an AWS service.

Resource defines the scope of entities covered by the policy rule.

In the navigation pane on the left, choose User groups. Choose the name of the S3-Support group. Choose the Permissions tab. The S3-Support group has the AmazonS3ReadOnlyAccess policy attached. Under Policy Name, choose the link for the AmazonS3ReadOnlyAccess policy. Choose the {} JSON tab.

This policy has permissions to *Get* and *List* for *all* resources in Amazon S3. In the navigation pane on the left, choose User groups. Choose the name of the EC2-Admin group. Choose the Permissions tab.

Under Policy Name, choose the name of the EC2-Admin-Policy policy. Choose the JSON tab. This policy grants permission to *Describe* information about Amazon EC2 instances, and also the ability to *Start* and *Stop* instances. At the bottom of the screen, choose Cancel to close the policy.

Task 2. Add users to groups

You want to add *user-1* to *S3-Support* group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy.

Add user-1 to the S3-Support group

In the left navigation pane, choose User groups.

Choose the name of the S3-Support group.

On the Users tab, choose Add users. Select user-1, and choose Add users.

On the Users tab, notice that *user-1* has been added to the group.

Add user-2 to the EC2-Support group

You want to add *user-2* to *EC2-Support* group so that they inherit the necessary permissions via the attached *AmazonEC2ReadOnlyAccess* policy.

Use what you learned from the previous steps to add *user-2* to the *EC2-Support* group.

user-2 should now be part of the *EC2-Support* group.

Add user-3 to the EC2-Admin group

You want to add *user-3* to the *EC2-Admin* group so that they inherit the necessary permissions via the attached *EC2-Admin-Policy*.

Use what you learned from the previous steps to add *user-3* to the *EC2-Admin* group.

user-3 should now be part of the *EC2-Admin* group. In the navigation pane on the left, choose User groups.

Task 3. Sign in and test users

Get the console sign-in URL

In the navigation pane on the left, choose Dashboard.
Notice the Sign-in URL for IAM users in this account section at the top of the page. Copy the sign-in link to a text editor.

Test user-1 permissions

Open a private or incognito window in your browser.
Paste the sign-in link into the private browser, and press ENTER.
You will now sign-in as *user-1*, who has been hired as your Amazon S3 storage support staff. Sign in with the following credentials:
IAM user name: user-1
Password: Lab-Password1
Choose the Services menu, and choose S3.
Choose the name of one of your buckets, and browse the contents.

Because this user is part of the *S3-Support* group in IAM, they have permissions to view a list of Amazon S3 buckets and their contents.

Now, test whether the user has access to Amazon EC2. Choose the Services menu, and choose EC2.

In the left navigation pane, choose Instances.

You cannot see any instances. Instead, an error message says *you are not authorized to perform this operation*. This user has not been assigned any permissions to use Amazon EC2.

You will now sign in as *user-2*, who has been hired as your Amazon EC2 support person.

First, sign out *user-1* from the console:

In the upper-right corner of the page, choose user-1.

Choose Sign Out.

Test user-2 permissions

Paste the sign-in link into the private browser again, and press ENTER. Sign in with the following credentials:

IAM user name: user-2

Password: Lab-Password2

Choose the Services menu, and choose EC2.

In the navigation pane on the left, choose Instances.

You are now able to see an EC2 instance. However, you cannot make any changes to Amazon EC2 resources because you have read-only permissions.

Select the EC2 instance.

Choose the Instance state menu, and then choose Stop instance. To confirm that you want to stop the instance, choose Stop.

An error message appears and says that *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information without making changes.

Next, check if *user-2* can access Amazon S3.

Choose the Services menu, and choose S3.

An error message says *You don't have permissions to list buckets* because *user-2* does not have permissions to use Amazon S3.

You will now sign-in as *user-3*, who has been hired as your Amazon EC2 administrator. First, sign out *user-2* from the console:

In the upper-right corner of the page, choose *user-2*.

Choose Sign Out.

Test *user-3* permissions

Paste the sign-in link into the private browser again, and press ENTER. Sign in with the following credentials:

IAM user name *user-3*

Password: *lab-Password3*

Choose the Services menu, and choose EC2.

In the navigation pane on the left, choose Instances.

An EC2 instance is listed. As an Amazon EC2 Administrator, this user should have permissions to *Stop* the EC2 instance.

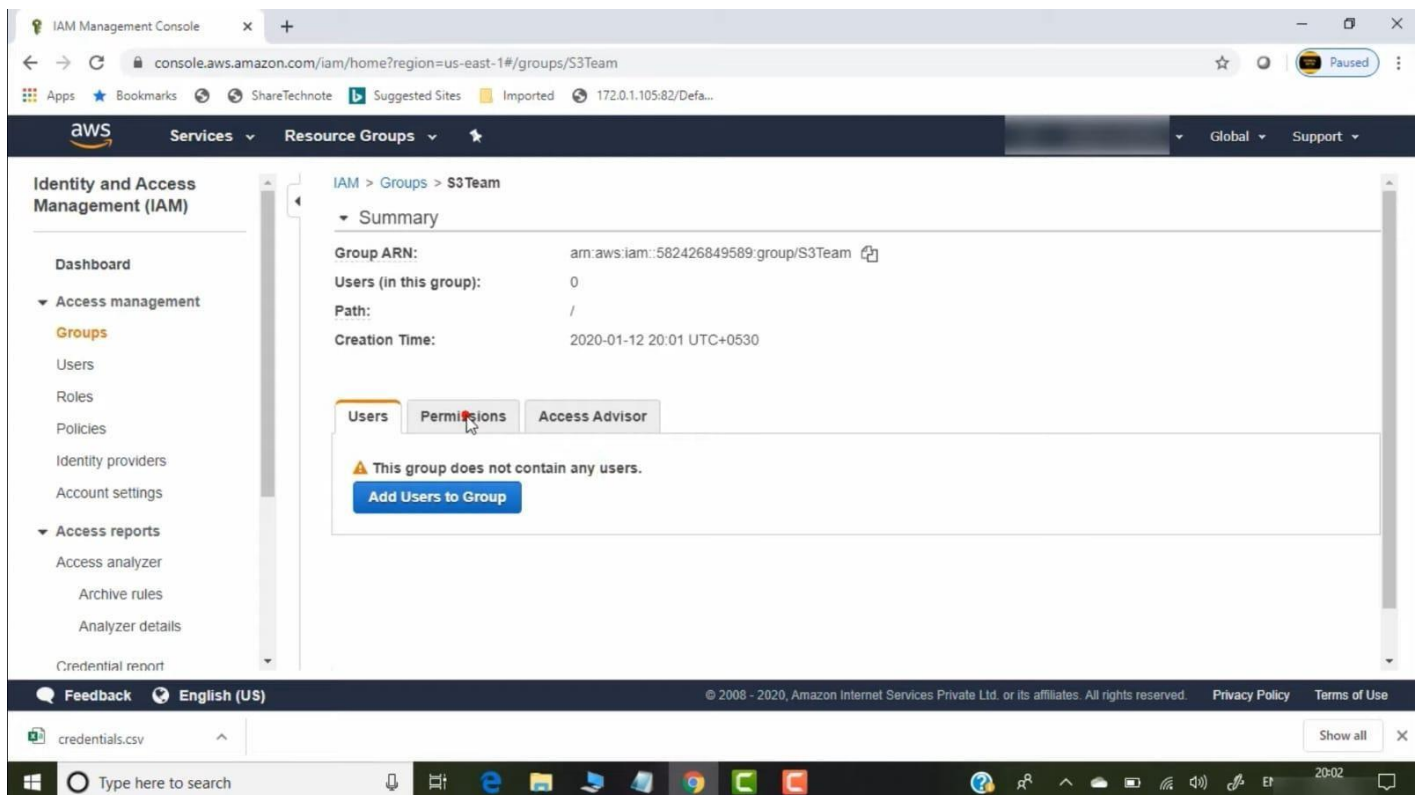
Select the EC2 instance.

Choose the Instance state menu, and then choose Stop instance.

To confirm that you want to stop the instance, choose Stop.

This time, the action is successful because *user-3* has permissions to stop EC2 instances. The Instance state changes to *Stopping* and starts to shut down.

Close your private browser window



Experiment 10

Aim: Build a RESTful serverless API on AWS.

Steps:

Task 1. Create a New REST API

In the AWS Management Console, click Services then select API Gateway under Application Services. Choose Create API.

Select New API and enter WildRydes for the API Name.

Keep Edge optimized selected in the Endpoint Type dropdown. Note: Edge optimized are best for public services being accessed from the Internet. Regional endpoints are typically used for APIs that are accessed primarily from within the same AWS Region.

Choose Create API

Task 2. Create a Cognito User Pools Authorizer

Under your newly created API, choose Authorizers.

Chose Create New Authorizer.

Enter WildRydes for the Authorizer name. Select Cognito for the type.

In the Region drop-down under Cognito User Pool, select the Region where you created your Cognito user pool.

Enter WildRydes (or the name you gave your user pool) in the Cognito User Pool input. Enter Authorization for the Token Source.

Choose Create.

Verify your authorizer configuration

Open a new browser tab and visit /ride.html under your website's domain.

If you are redirected to the sign-in page, sign in with the user you created in the last module. You will be redirected back to /ride.html.

Copy the auth token from the notification on the /ride.html,

Go back to previous tab where you have just finished creating the Authorizer Click Test at the bottom of the card for the authorizer.

Paste the auth token into the Authorization Token field in the popup dialog.

Task 3. Create a new resource and method

In the left nav, click on Resources under your WildRydes API.

From the Actions dropdown select Create Resource.

Enter ride as the Resource Name.

Ensure the Resource Path is set to ride.

Select Enable API Gateway CORS for the resource.

Click Create Resource.

With the newly created /ride resource selected, from the Action dropdown select Create Method.

Working with EBS in AWS.

Build a relational database server.

Create a private cloud - Designing a custom VPC (Virtual Private Cloud).

Create an IAM Group in Cloud.

Build a RESTful serverless API on AWS.

Select POST from the new dropdown that appears, then click the checkmark.

Select Lambda Function for the integration type.

Check the box for Use Lambda Proxy integration.

Select the Region you are using for Lambda Region.

Enter the name of the function you created in the previous module, RequestUnicorn, for Lambda Function.

Choose Save.

Task 4. Deploy Your API

In the Actions drop-down list select Deploy API.

Select [New Stage] in the Deployment stage drop-down list.

Enter prod for the Stage Name.

Choose Deploy.

Note the Invoke URL. You will use it in the next section.

Task 5. Update the Website Config

visit `/js/config.js` under the base URL for your website and choose File, then choose Save Page As from your browser.

Open the config.js file in a text editor.

Update the `invokeUrl` setting under the `api` key in the config.js file. Set the value to the Invoke URL for the deployment stage you created in the previous section.

An example of a complete `config.js` file is included below. Note, the actual values in your file will be different.

```
window._config = {
  cognito: {
    userPoolId: 'us-west-2_uXboG5pAb', // e.g. us-east-2_uXboG5pAb
    userPoolClientId: '25ddkmj4v6hfsfvrupfi7n4hv', // e.g. 25ddkmj4v6hfsfvrupfi7n4hv region: 'us-west-2' // e.g. us-east-2
  },
  api: {
    invokeUrl: 'https://rc7nyt4tql.execute-api.us-west-2.amazonaws.com/prod' // e.g. https://rc7nyt4tql.execute-api.us-west-2.amazonaws.com/prod,
  }
};
```

Save your changes locally.

In the AWS Management Console, choose Services then select S3 under Storage.

Navigate to the website bucket and then browse to the `js` key prefix.

Choose Upload.

Choose Add files, select the local copy of `config.js` and then click Next.

Choose Next without changing any defaults through the **Set permissions** and **Set properties** sections.

Choose Upload on the **Review** section.



NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY
GREATER NOIDA

CLOUD COMPUTING LAB

BMCA0351

Master of Computer Applications

(Session:2024-2025)

Affiliated to



Dr. A.P.J. Abdul Kalam Technical University, Lucknow (U.P.)

Under Guidance of:

Mr. ABHISHEK KUMAR

(Asst. Professor)

(Department of Computer Applications)

Submitted by:

Student Name: Asheesh Rathore

ROLL No.: 2301330140035