

### TASK 3

#### **Task: Design a Multi-Cloud Architecture (AWS-Only Steps)**

**Focus:** Configure AWS services to enable interoperability with another cloud platform.

---

#### **Step-by-Step Guide**

**Region:** Use the **default AWS region** (e.g., us-east-1) for all steps.

---

#### **Step 1: Create a VPC for AWS Resources**

**Objective:** Design a secure network to host AWS services.

1. Go to **AWS Management Console > VPC Dashboard**.
2. Click **Create VPC**.
  - **Name:** MultiCloud-VPC
  - **IPv4 CIDR:** 10.0.0.0/16
  - Leave other settings as default.
3. Create **Public Subnets**:
  - Subnet 1: 10.0.1.0/24 (Name: Public-Subnet-1, Availability Zone: us-east-1a)
  - Subnet 2: 10.0.2.0/24 (Name: Public-Subnet-2, Availability Zone: us-east-1b)
4. Attach an **Internet Gateway**:
  - Create an Internet Gateway named MultiCloud-IGW and attach it to the VPC.
5. Configure **Route Tables**:
  - Edit the main route table to route 0.0.0.0/0 traffic to the Internet Gateway.

#### **Screenshot Heading:**

**"Step 1: AWS VPC Configuration Showing Subnets and Internet Gateway"**

**Where to Capture:** VPC Dashboard showing the created VPC, subnets, and Internet Gateway.

---

#### **Step 2: Launch an EC2 Instance for Frontend Application**

**Objective:** Host a web server to interact with the other cloud platform.

1. Go to **EC2 Dashboard > Launch Instance**.

2. Configure:

- **Name:** MultiCloud-WebServer
- **AMI:** Amazon Linux 2023
- **Instance Type:** t2.micro
- **Key Pair:** Create/use an existing key pair.
- **Network:** Select MultiCloud-VPC and Public-Subnet-1.
- **Auto-assign Public IP:** Enable.

3. Under **Security Groups**:

- Create a new security group: WebServer-SG
- Allow **HTTP (Port 80)**, **HTTPS (Port 443)**, and **SSH (Port 22)**.

4. Launch the instance.

**Screenshot Heading:**

**"Step 2: EC2 Instance Launch Configuration with Public IP"**

**Where to Capture:** EC2 instance summary page showing the public IP and security group.

---

**Step 3: Set Up RDS MySQL Database**

**Objective:** Create a database accessible from both AWS and the other cloud.

1. Go to **RDS Dashboard > Create Database**.

2. Configure:

- **Engine:** MySQL
- **Template:** Free Tier
- **DB Instance Identifier:** multicloud-db
- **Credentials:** Set admin username/password.
- **Network:** Select MultiCloud-VPC and place the DB in Public-Subnet-2 (for demo purposes).

3. Under **Security Group**, create a new SG: DB-SG

- Allow inbound traffic from WebServer-SG on **Port 3306**.

4. Disable **Public Access** (for production, but enable temporarily for the demo).

**Screenshot Heading:**

**"Step 3: RDS Database Configuration in Public Subnet"**

**Where to Capture:** RDS database connectivity & security tab showing the security group and subnet.

---

**Step 4: Configure IAM Roles for Cross-Cloud Access**

**Objective:** Grant EC2 permissions to access S3 (simulating cross-cloud storage).

1. Go to **IAM Dashboard > Roles > Create Role**.
2. Select **AWS Service > EC2 > Next**.
3. Attach the **AmazonS3FullAccess** policy (for demo purposes).
4. Name the role: EC2-S3-Access and create it.
5. Attach the role to the MultiCloud-WebServer EC2 instance.

**Screenshot Heading:**

**"Step 4: IAM Role with S3 Access Attached to EC2"**

**Where to Capture:** IAM role summary page showing the attached policies.

---

**Step 5: Deploy a Sample Web Application**

**Objective:** Test connectivity between EC2, RDS, and S3.

1. SSH into the EC2 instance.
2. Install Apache and PHP:

bash

Copy

```
sudo yum install -y httpd php mysql
```

```
sudo systemctl start httpd
```

3. Create a PHP file (/var/www/html/index.php) to connect to RDS:

php

Copy

```
<?php
```

```
$servername = "<RDS_ENDPOINT>";
```

## Intern ID: CT06DG1190

```
$username = "<DB_USER>";  
$password = "<DB_PASSWORD>";  
$conn = new mysqli($servername, $username, $password);  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
echo "Connected to RDS successfully!";  
?>
```

4. Test the page at [http://<EC2\\_PUBLIC\\_IP>/index.php](http://<EC2_PUBLIC_IP>/index.php).

### Screenshot Heading:

#### "Step 5: Web App Successfully Connecting to RDS"

**Where to Capture:** Browser showing "Connected to RDS successfully!".

---

## Step 6: Simulate Cross-Cloud Interoperability with S3

**Objective:** Use S3 as a mock service for the second cloud platform.

1. Create an S3 bucket:
  - Go to **S3 Dashboard > Create Bucket > Name:** multicloud-demo-bucket
2. On the EC2 instance, use the AWS CLI to upload a test file:

bash

Copy

```
echo "Multi-Cloud Demo" > test.txt
```

```
aws s3 cp test.txt s3://multicloud-demo-bucket/
```

### Screenshot Heading:

#### "Step 6: File Uploaded to S3 from EC2 Instance"

**Where to Capture:** S3 bucket contents showing the uploaded test.txt.

---

## Step 7: Document the Architecture

1. Use **AWS Architecture Tool** or draw.io to create a diagram showing:

**Intern ID: CT06DG1190**

- EC2, RDS, S3, and VPC components.
  - Label the other cloud platform as a placeholder (e.g., "External Cloud Provider").
2. Write a summary explaining how data flows between AWS and the other cloud.

**Screenshot Heading:**

**"Step 7: Multi-Cloud Architecture Diagram"**

**Where to Capture:** Final architecture diagram with AWS and placeholder components.

**AWS VPC Configuration Showing Subnets and Internet Gateway"**

## Intern ID: CT06DG1190

The screenshot shows the AWS VPC dashboard. The left sidebar contains navigation links for 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main content area displays 'Your VPCs (2)' with a table listing two VPCs: 'Public-Subnet-1' and 'Public-Subnet-2'. Both are in an 'Available' state. Below the table, there is a 'Select a VPC above' prompt.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option se
Public-Subnet-1	vpc-037966befba7837e8	Available	Off	172.31.0.0/16	-	dopt-0d05831e
Public-Subnet-2	vpc-0869fa51b971d4685	Available	Off	10.0.0.0/16	-	dopt-0d05831e

The screenshot shows the AWS VPC dashboard. The left sidebar contains navigation links for 'Virtual private cloud', 'Security', and 'PrivateLink and Lattice'. The main content area displays 'Subnets (2)' with a table listing two subnets: 'Public-Subnet-1' and 'Public-Subnet-2'. Both are in an 'Available' state. Below the table, there is a 'Select a subnet' prompt.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
Public-Subnet-1	subnet-0ca2a2c4669f119f0	Available	vpc-0869fa51b971d4685   Mult...	Off	10.0.1.0/24	-
Public-Subnet-2	subnet-0502293f5c17082de	Available	vpc-0869fa51b971d4685   Mult...	Off	10.0.2.0/24	-

VPC Dashboard showing the created VPC, subnets, and Internet Gateway.

Search

[Alt+S]

United States (N. Virginia)

CH Pavan

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Security

Network ACLs

Security groups

PrivateLink and Lattice

Getting started

Endpoints

Route tables (1/2)

Find resources by attribute or tag

Route table ID

Explicit subnet associ...

Edge associations

Main

VPC

Owner ID

rtb-0d3ebd0b95764f423

rtb-0a2c64fae2f9965db

Yes

Yes

vpc-037966befba7837e8

vpc-0869fa51b971d4685

952216971402

952216971402

rtb-0d3ebd0b95764f423

Details

Routes

Subnet associations

Edge associations

Route propagation

Tags

Details

Route table ID

rtb-0d3ebd0b95764f423

Main

Yes

Explicit subnet associations

-

Edge associations

-

VPC

vpc-037966befba7837e8

Owner ID

952216971402

Search

[Alt+S]

United States (N. Virginia)

CH Pavan

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Security

Network ACLs

Security groups

PrivateLink and Lattice

Getting started

Endpoints

Internet gateways (2)

Search

Internet gateway ID

State

VPC ID

Owner

igw-04160c61d654b952e

igw-03f5e1be5153453f

Attached

Attached

vpc-037966befba7837e8

vpc-0869fa51b971d4685

952216971402

952216971402

Select an internet gateway above

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

## Launch instance

Success

Successfully initiated launch of instance (i-0f3d82db7128939a0)

Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Create billing alerts

Connect to your instance

Once your instance is running, log into it from your local computer.

Connect to instance

Learn more

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Connect an RDS database

Create a new RDS database

Learn more

Create EBS snapshot policy

Create a policy that automates the creation, retention, and deletion of EBS snapshots

Create EBS snapshot policy

Manage detailed monitoring

Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays console statistics with a 1-minute resolution.

Create Load Balancer

Create an application, network gateway or classic Elastic Load Balancing

Create Load Balancer

Create AWS budget

AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.

Manage CloudWatch alarms

Create or update Amazon CloudWatch alarms for the instance.

Manage CloudWatch alarms

Auto-assign public IP

Whether a public IP address is automatically assigned to the primary network interface of the instance

Get more networking guidance

Pricing

Public IPv4 addresses (including Amazon-provided public IPv4 addresses) incur an hourly charge. The AWS Free Tier for EC2 includes 750 hours of public IPv4 address usage per month for the first 12 months.

Learn more

Amazon VPC Pricing

AWS Free Tier

What's new: Public IPv4 charges

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Instances (1)

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
<input type="checkbox"/>	MultiCloud-W...	i-0f3d82db7128939a0	Running	t2.micro	Initializing	View alarms +	us-east-1a	-	3.83.2:

Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



## Creating Database

The screenshot shows the AWS Management Console interface for Amazon RDS. A blue notification banner at the top states: "Creating database multicloud-db. Your database might take a few minutes to launch. You can use settings from multicloud-db to simplify configuration of suggested database add-ons while we finish creating your DB for you." Below the notification, the "Databases (1)" section shows a table with one entry: "multicloud-db" with a status of "Creating". The left sidebar contains navigation links for Amazon RDS, including Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update.

DB identifier	Status	Role	Engine	Region	Size	Recommendations	CPU
multicloud-db	Creating	Instance	MySQL Co...	-	db.t4g.micro	-	-

The screenshot shows the AWS Management Console interface for Amazon RDS. A green notification banner at the top states: "Successfully created database multicloud-db. You can use settings from multicloud-db to simplify configuration of suggested database add-ons while we finish creating your DB for you." Below the notification, the "Databases (1)" section shows a table with one entry: "multicloud-db" with a status of "Backin...". The left sidebar contains navigation links for Amazon RDS, including Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update.

DB identifier	Status	Role	Engine	Region	Size	Recommendations	CPU
multicloud-db	Backin...	Instance	MySQL Co...	us-east-1a	db.t4g.micro	-	-

**Intern ID: CT06DG1190**