LATHIKA LJ **212223220050**

Exp.No.	04	Study of setting up a Private Cloud in an AWS	Year/Sem	2nd/4th
Date	12/04/2025		Branch	IT

Aim:

To Study of setting up a Private Cloud in an AWS.

Procedure:

1. Create a Virtual Private Cloud (VPC)

- Log in to the AWS Management Console and navigate to the "VPC" service.
- Click on "Create VPC" and assign it a name (e.g., "MyPrivateCloud").
- Define an IP address range using CIDR notation, such as 10.0.0.0/16, which provides 65,536 IP addresses for your VPC.

2. Set Up Subnets

- Subnets are subdivisions of your VPC that allow you to segment resources.
- Create both public and private subnets:
  - Public subnets allow internet access for resources like web servers.
  - Private subnets are isolated from the internet for sensitive workloads.
- Assign each subnet a CIDR block within the VPC's IP range (e.g., 10.0.1.0/24 for public and 10.0.2.0/24 for private).

3. Configure Routing

- Set up route tables to define how traffic is directed within the VPC:
  - Attach an internet gateway to the public subnet for outbound internet access.
  - Use a NAT gateway or NAT instance for private subnets to securely access the internet.

4. Create Security Groups

- Security groups act as virtual firewalls for resources in your VPC:
  - Define inbound rules (e.g., allow HTTP traffic on port 80 or SSH traffic on port 22).
  - Define outbound rules as needed.

5. Launch EC2 Instances

- Navigate to the EC2 dashboard and launch instances into your configured subnets:
  - Public-facing instances (e.g., web servers) should reside in public subnets.
  - Backend services or databases should be deployed in private subnets.
- Assign Elastic IPs to instances in public subnets if external access is required.

6. Test Connectivity

- Verify that resources in public subnets can access the internet.
- Test connectivity between instances in private subnets using their private IP addresses.

CONCLUSION:

- Monitoring and Scaling: Use AWS CloudWatch to monitor resource usage and set up auto-scaling groups for dynamic scaling.
- Security Best Practices: Implement IAM roles, encrypt data at rest and in transit, and regularly audit security configurations.
- Multi-VPC Architecture: For larger setups, consider creating multiple VPCs connected via peering or AWS Transit Gateway to segregate workloads