



## Day 12

# “CLOUD SECURITY”

### Utilizing Separate VPCs to Isolate Infrastructure

#### 1. Purpose & Benefits

- Amazon VPC provides logically isolated virtual networks within AWS to separate workloads or organizational units.
- Enhances security, network control, and resource management in the public cloud.

#### 2. Isolation & Configuration

- Use subnets to separate application tiers (e.g., web, app, DB).
- Private subnets prevent direct internet access to sensitive instances.
- Administrators control IP ranges, DHCP, routing, and access policies.

#### 3. Security Features

- Security Groups (SGs): Act as virtual firewalls to control traffic to/from instances.
- Access Control Lists (ACLs): Define specific IPs or applications allowed to access the VPC.
- Prevents DDoS and unauthorized access by isolating instances from public exposure.

#### 4. Result

- Ensures granular access control, layer-3 Internet isolation, and multi-tenant data separation in AWS.

### Creating a VPC:

Steps:

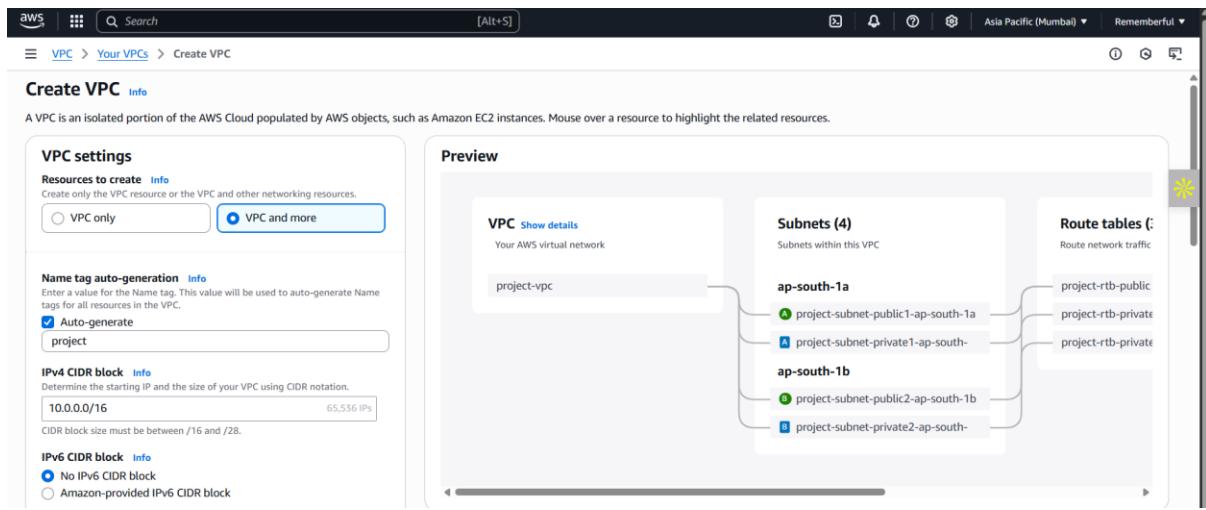
Open the AWS, and enter VPC in the search bar, and press enter. Following Screen will appear:

The screenshot shows the AWS VPC dashboard. At the top, there are buttons for 'Create VPC' and 'Launch EC2 Instances'. Below this, a section titled 'Resources by Region' displays the following data:

Category	Region	Count
VPCs	Mumbai	2
Subnets	Mumbai	6
NAT Gateways	Mumbai	0
VPC Peering Connections	Mumbai	0

On the left sidebar, under 'Virtual private cloud', the following resources are listed: Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, and Default VPCs. On the right side, there are sections for 'Service Health', 'Settings' (with options for Block Public Access, Zones, and Console Experiments), and 'Additional Information' (with links to VPC Documentation and AWS Best Practices).

Click on the “Create VPC” button: following screen will appear



Specify the details and then click on the “create VPC” button at the bottom.

### **Best practices for creating a secure and well-architected VPC in AWS:**

#### **1. VPC Design**

- Choose a non-overlapping CIDR block (e.g., 10.0.0.0/16).
- Create separate public and private subnets.
- Use multiple Availability Zones (AZs) for high availability.
- Isolate application tiers (web, app, DB) using subnets.

#### **2. Security Best Practices**

- Use Security Groups (SGs):
  - Allow only required ports/IPs.
  - Avoid 0.0.0.0/0 unless strictly necessary.
- Use Network ACLs (NACLs) for subnet-level filtering.
- Enable VPC Flow Logs for traffic monitoring and auditing.

#### **3. Internet & Routing**

- Attach an Internet Gateway (IGW) to VPC for internet access.
- Use NAT Gateway or NAT instance for secure internet access from private subnets.
- Configure Route Tables correctly for each subnet.

#### **4. Private Connectivity**

- Use VPC Endpoints (Interface/Gateway) for private AWS service access.
- Use AWS PrivateLink for secure private connectivity to third-party services.

#### **5. Network Management**

- Enable DNS resolution and hostnames.
- Use DHCP options set for custom domain names.
- Use Tags for organizing and managing resources.

#### **6. Access & Control**

- Use Bastion Host (Jump Box) for SSH access to private instances.
- Enforce IAM roles and policies for least-privilege access.
- Use multi-factor authentication (MFA) for management access.

#### **7. Monitoring & Compliance**

- Enable AWS CloudTrail to log all API actions.
- Use AWS Config for continuous compliance checks.
- Enable Amazon GuardDuty for threat detection.

## **8. Encryption**

- Use AWS KMS for encrypting data at rest (EBS, S3).
- Enforce TLS for data in transit.

## **9. High Availability**

- Deploy resources across multiple AZs.
- Use Elastic Load Balancing for distributing traffic.
- Set up Auto Scaling for resilience.