## Author: Madhurima Rawat

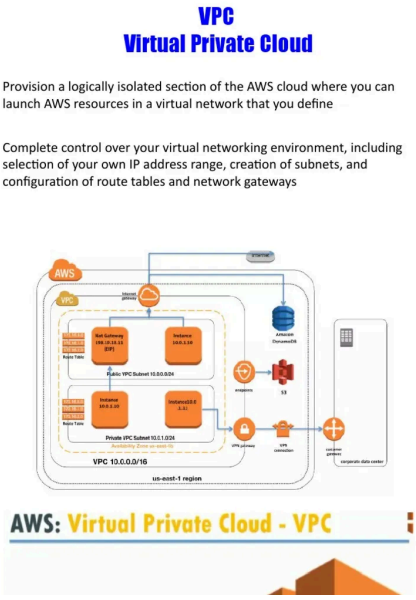
## Setting Up and Configuring Cloud Networking

**This experiment explores configuring and managing cloud networking services, including setting up Virtual Private Clouds (VPCs) and subnets. It utilizes AWS CLI and LocalStack to simulate cloud networking environments.**

## 1. Overview of Virtual Private Cloud (VPC)

A **VPC** is a customizable virtual network within the AWS cloud. It allows you to manage networking resources securely and privately.

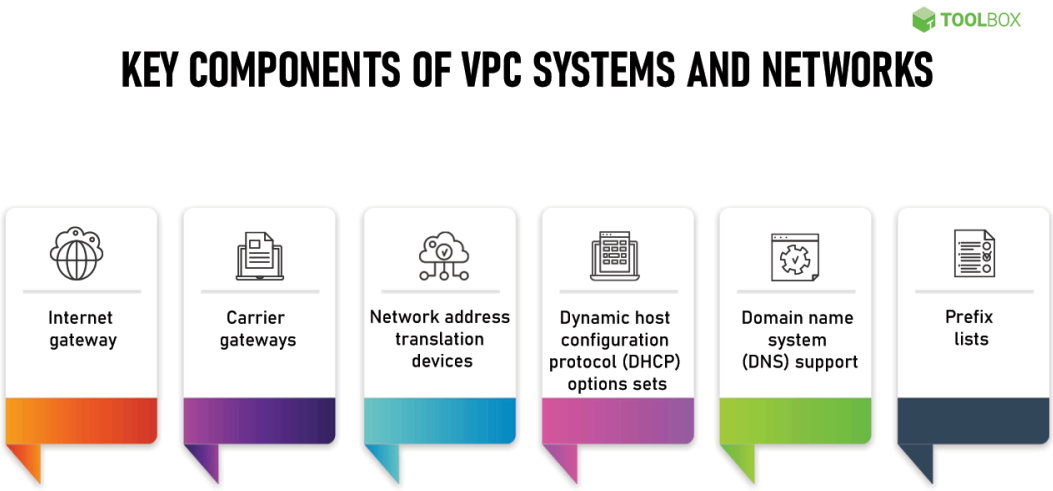
- Allows the creation of isolated networks in the cloud.
- Provides control over IP address ranges and routing.
- Enables creating public and private subnets.
- Facilitates secure, private communication between instances.

## 2. Key Components of a VPC

The VPC is made up of several essential components for creating a robust networking environment:

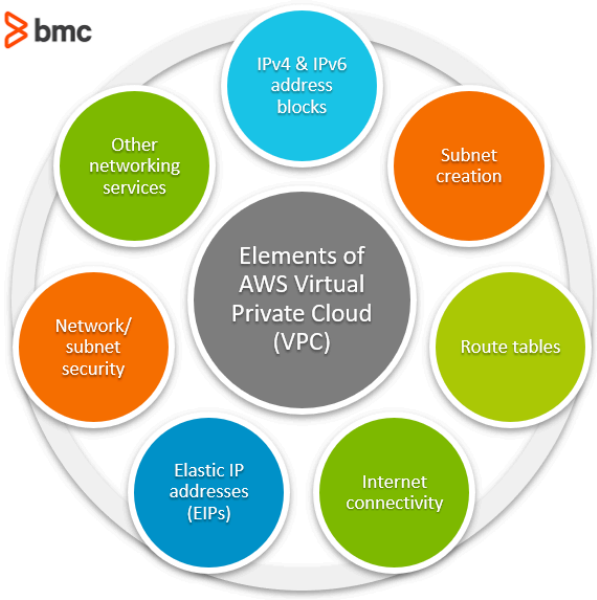
- **Subnets**: Segment the network into smaller, manageable sections.
- **Internet Gateway (IGW)**: Provides internet access for public-facing resources.
- **Route Tables**: Direct traffic within the VPC and between the internet.
- **NAT Gateway**: Allows private subnet instances to access the internet.
- **Security Groups & Network ACLs**: Define inbound and outbound traffic permissions for resources.




## 3. Networking Flow and Communication

The VPC enables seamless communication between instances and secure internet access.

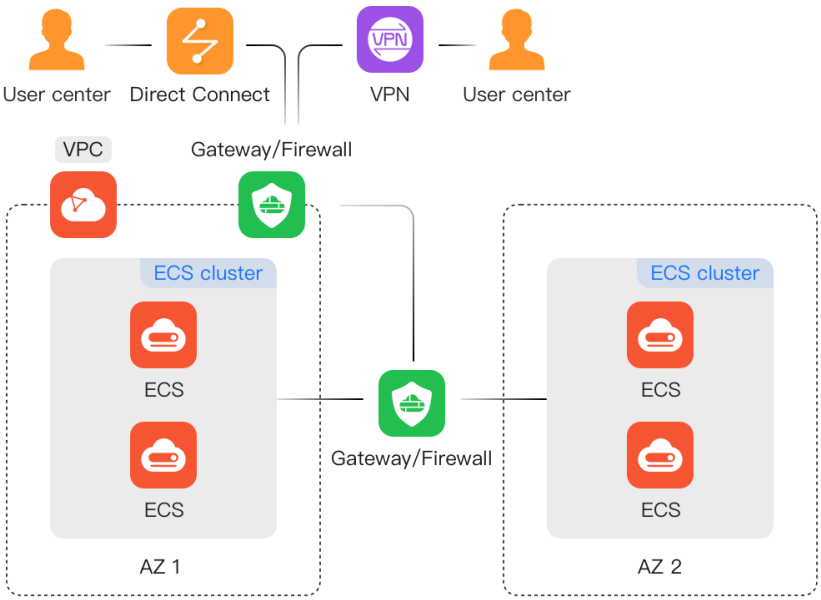
- **Public Subnets**: These subnets have internet access via an IGW, making them suitable for web servers.
- **Private Subnets**: Isolated subnets that don't directly connect to the internet; they access the internet via a NAT Gateway.
- **Route Tables**: Manage the routing of network traffic within the VPC, to/from the internet or between subnets.
- **Security Layers**: Using **Security Groups** and **Network ACLs** to filter traffic and secure resources.

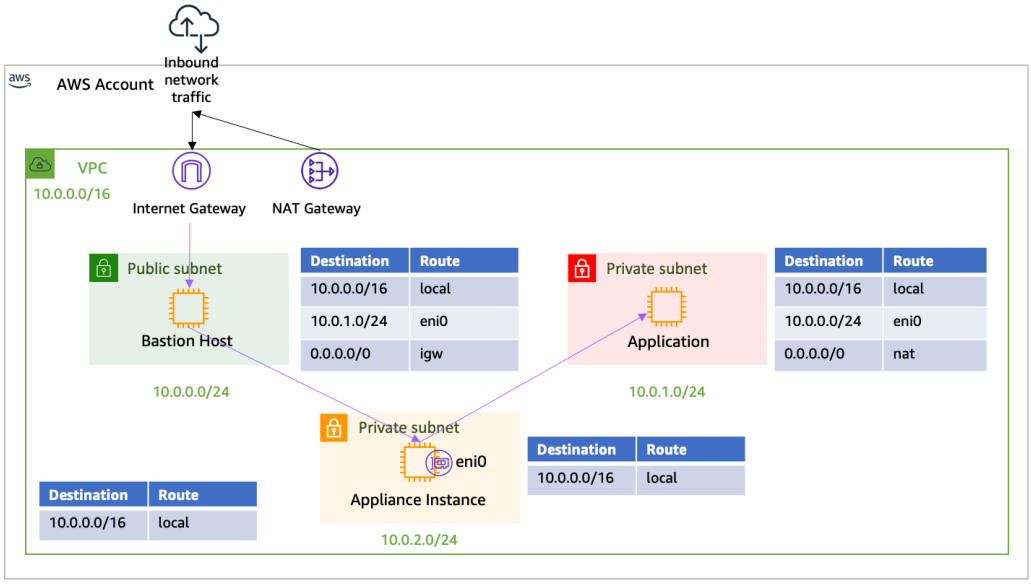



## 4. LocalStack for VPC Simulation

LocalStack simulates AWS services locally, allowing you to test cloud configurations without needing actual AWS resources.

- **Simulates AWS Cloud Services**: Enables local testing for cloud networking and services like VPC.
- **Supports Full AWS Environment**: Includes EC2, S3, VPC, and more.
- **Custom Endpoints**: Interact with LocalStack through custom endpoint URLs, mimicking real AWS behavior.
- **Ideal for Development and Testing**: Provides a cost-effective solution for developers to test network setups.

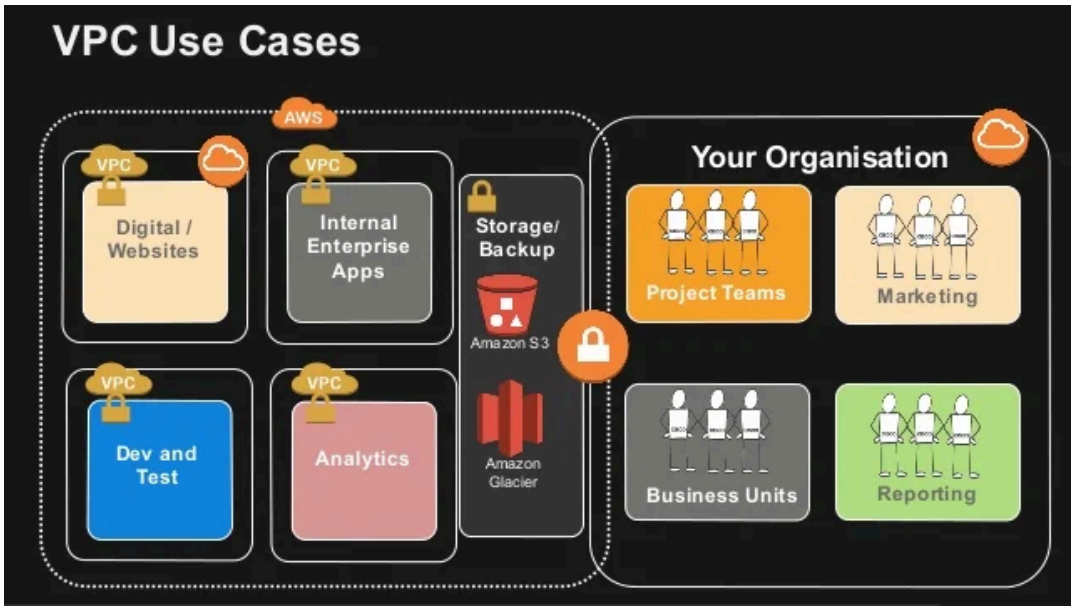





## 5. Use Cases of VPCs

VPCs can be used for various networking purposes to create secure, scalable architectures.

- **Web Hosting**: Deploy secure web applications with public-facing and private backends.
- **Hybrid Cloud**: Extend on-premise networks into the cloud via VPN or Direct Connect.
- **Big Data & Machine Learning**: Isolate workloads for processing and analysis.
- **Multi-Tier Architectures**: Separate frontend, application, and database tiers for better security.




# Step-by-Step Guide: Setting Up a VPC in LocalStack

# Step 1: Start LocalStack

Start LocalStack using one of the following methods:

### Using the CLI

```
localstack start
```

### Using Docker

```
docker run --rm -it -p 4566:4566 localstack/localstack
```

### Ensure Docker is Running

- Open **Docker Desktop** and wait until it displays "**Docker is running.**"
- LocalStack will simulate AWS services on **port 4566**, allowing local development without needing an actual AWS account.

## Step 2: Create a Virtual Private Cloud (VPC)

Create a **VPC (Virtual Private Cloud)** to define a private network:

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16
--endpoint-url=%AWS_ENDPOINT_URL%
```

### What This Does:

- Defines a **private network (10.0.0.0/16)** for AWS resources.
- Returns a **VPC ID** (e.g., `"VpcId": "vpc-123456"`), which is required for the next steps.

## Step 3: Create a Subnet

A **subnet** allows instances to communicate within the VPC.

```
aws ec2 create-subnet --vpc-id vpc-123456
--cidr-block 10.0.1.0/24 --endpoint-url=%AWS_ENDPOINT_URL%
```

### What This Does:

- Creates a **smaller network (10.0.1.0/24)** within the VPC for organizing resources.

- A **Subnet ID** (e.g., `"SubnetId": "subnet-123456"`) is returned, which is needed for later steps.

## Step 4: Create and Attach an Internet Gateway

An **Internet Gateway (IGW)** allows resources in the subnet to access the internet.

### Create the Internet Gateway:

```
aws ec2 create-internet-gateway --endpoint-url=%AWS_ENDPOINT_URL%
```

### Attach the Internet Gateway to the VPC:

```
aws ec2 attach-internet-gateway --internet-gateway-id
igw-123456 --vpc-id vpc-123456 --endpoint-url=%AWS_ENDPOINT_URL%
```

### What This Does:

- Enables instances in the VPC to communicate with the internet.
- Returns an **Internet Gateway ID** (e.g., `"InternetGatewayId": "igw-123456"`).

## Step 5: Configure Routing for Internet Access

To allow traffic to and from the internet, a **Route Table** must be created and updated.

### Create a Route Table for the VPC:

```
aws ec2 create-route-table --vpc-id vpc-123456
--endpoint-url=%AWS_ENDPOINT_URL%
```

### Add a Default Route to Enable Internet Access:

```
aws ec2 create-route --route-table-id rtb-123456
--destination-cidr-block 0.0.0.0/0 --gateway-id
igw-123456 --endpoint-url=%AWS_ENDPOINT_URL%
```

### Associate the Route Table with the Subnet:

```
aws ec2 associate-route-table --route-table-id rtb-123456
--subnet-id subnet-123456 --endpoint-url=%AWS_ENDPOINT_URL%
```

### What This Does:

- Defines a **default route ( `0.0.0.0/0` )**, allowing traffic to flow to the internet.
- Links the route table to the **subnet** so instances can use the IGW.

## Step 6: Verify the Configuration

### List All VPCs:

```
aws ec2 describe-vpcs --endpoint-url=%AWS_ENDPOINT_URL%
```

### List All Subnets:

```
aws ec2 describe-subnets --endpoint-url=%AWS_ENDPOINT_URL%
```

### What This Does:

- Confirms the VPC and subnets have been successfully created and configured.
- Ensures all components (VPC, subnet, IGW, and routing) are correctly set up.

## Useful Resources for Learning about Virtual Private Cloud (VPC)

1. **Spiceworks - What is Virtual Private Cloud?**
   A comprehensive guide explaining the concept of Virtual Private Cloud (VPC), its benefits, and how it enhances network security and control over cloud resources.

2. **BMC - AWS VPC: Virtual Private Cloud**
   This article provides an in-depth overview of AWS VPC, covering key features, use cases, and how to set up and manage VPCs effectively on AWS.

3. **Medium - Deep Dive into Amazon Virtual Private Cloud (VPC)**
   A detailed exploration of the core features of Amazon VPC, focusing on security, scalability, and flexibility in network configurations.

4. **Medium - Understanding AWS VPC: A Comprehensive Guide from Basics to Best Practices**
   This guide offers an extensive look at AWS VPC, from its basic concepts to best practices for

optimal usage in various scenarios, including security and architecture considerations.

5. **Medium - AWS VPC: Virtual Private Cloud**
   A beginner-friendly article that explains the fundamental concepts of AWS VPC, its components, and how to leverage VPC for secure cloud networking.