DevOps Shack

Ports, SSH, SSH Key Pair, and Security Groups in Linux

[Click Here To Enrol To Batch-6 | DevOps & Cloud DevOps]

Ports in Linux

Ports are logical endpoints for communication used by network protocols to manage data exchange between devices over a network. Each port is identified by a number, ranging from 0 to 65535, and is associated with a specific protocol and service.

Common Port Numbers

- **HTTP:** Port 80
- **HTTPS:** Port 443
- **SSH:** Port 22
- **FTP:** Port 21

Ports below 1024 are known as "well-known ports" and are reserved for system or well-known services. Ports from 1024 to 49151 are "registered ports," and ports from 49152 to 65535 are "dynamic or private ports."

SSH (Secure Shell)

SSH is a cryptographic network protocol used for secure data communication, remote command-line login, and other secure network services between two networked computers. SSH operates on port 22 by default and uses strong encryption to protect the data transmitted over the network.

Key Features of SSH:

- **Encrypted Communication:** Ensures that data sent over the network is encrypted and secure.
- **Authentication:** Uses public key authentication, making it more secure than password-based login.
- **Port Forwarding:** Allows secure forwarding of ports from the client machine to the remote server.

SSH Key Pair

SSH key pairs are a set of cryptographic keys used to authenticate a user in SSH. They consist of a private key (kept secret) and a public key (shared with the remote server).

Generating SSH Key Pair

To generate an SSH key pair in Ubuntu:

ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

This command generates a public-private key pair:

- **Public Key:** Stored in ~/.ssh/id_rsa.pub
- **Private Key:** Stored in ~/.ssh/id_rsa

You will be prompted to enter a file path to save the key and an optional passphrase for additional security.

Using SSH Key Pair

1. **Add the Public Key to the Remote Server:**

ssh-copy-id user@remote_host

This command appends the public key to the ~/.ssh/authorized_keys file on the remote server.

2. **Connect to the Remote Server Using SSH Key:**

ssh user@remote_host

Example in Ubuntu

1. **Generate SSH Key Pair:**

ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

2. **Add the Public Key to the Remote Server:**

ssh-copy-id user@192.168.1.100

3. **Connect to the Remote Server:**

ssh user@192.168.1.100

Security Groups, Inbound Rules, and Outbound Rules

Security groups act as virtual firewalls for instances to control inbound and outbound traffic. They are commonly used in cloud environments (e.g., AWS, Azure, Google Cloud) to manage access to resources.

Inbound Rules

Inbound rules control the incoming traffic to an instance. These rules specify which ports and protocols are allowed to receive traffic and from which sources (IP addresses).

Outbound Rules

Outbound rules control the outgoing traffic from an instance. These rules specify which ports and protocols are allowed to send traffic and to which destinations.

Example Configuration

1. **Inbound Rule:**
    - Allow SSH access on port 22.
    - Allow HTTP access on port 80.

```
Inbound Rule:
+------------+---------+---------+
| Port Range | Protocol| Source  |
+------------+---------+---------+
| 22         | TCP     | 0.0.0.0/0|
| 80         | TCP     | 0.0.0.0/0|
+------------+---------+---------+
```

2. **Outbound Rule:**
    - Allow all outgoing traffic.

```
Outbound Rule:
+------------+---------+------------+
| Port Range | Protocol| Destination|
+------------+---------+------------+
| All        | All     | 0.0.0.0/0  |
+------------+---------+------------+
```

Textual Diagram

Detailed Explanation

Ports

Ports are essential for network communication, allowing different services to run simultaneously on a single machine. For example, a web server running on port 80 (HTTP) can coexist with an SSH server running on port 22.

SSH and SSH Key Pair

SSH is widely used for secure remote login and command execution. Instead of passwords, SSH keys offer stronger security. The private key remains on the user's machine, while the public key is placed on the remote server. When the user attempts to connect, the server verifies the user's identity by matching the public key with the private key.

Security Groups, Inbound, and Outbound Rules

Security groups manage traffic to and from instances. Inbound rules define what traffic is allowed to reach the instance, while outbound rules define what traffic is allowed to leave. For example, allowing inbound SSH access on port 22 lets users remotely manage the instance, while allowing HTTP access on port 80 enables the instance to serve web pages.

This structure provides a clear and secure way to manage network traffic and access control in a Linux environment, ensuring both functionality and security.