Internal DNS Setup to Connect EC2 Servers

This document demonstrates how to configure internal DNS using AWS Route 53 to allow EC2 servers to communicate using hostnames instead of IP addresses.

Step 1: Create EC2 Servers

Launch three EC2 Linux instances and name them Linux1, Linux2, and Linux3.

Step 2: Create a User and Configure Sudo Access

```
useradd linux
passwd linux

# Add to sudoers
visudo
linux   ALL=(ALL)       NOPASSWD:ALL
```

Alternatively, you can safely add sudo privileges using:

```
echo "linux ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/linux
chmod 440 /etc/sudoers.d/linux
```

Step 3: Enable Password Authentication for SSH

```
sudo vi /etc/ssh/sshd_config
PasswordAuthentication yes
sudo systemctl restart sshd
```

Step 4: Configure Private DNS in Route 53

In Route 53, create a private hosted zone and add A-records with private IPs of the EC2 instances.

Record Name	Type	Value
linux1.nagendrappa.xyz	A	172.31.xx.xx
linux2.nagendrappa.xyz	A	172.31.xx.xx
linux3.nagendrappa.xyz	A	172.31.xx.xx

Step 5: Update Hostname

```
sudo hostnamectl set-hostname linux1.nagendrappa.xyz
hostname
```

Step 6: Verify Connectivity

```
ping linux1.nagendrappa.xyz
```

You should see successful ICMP responses confirming internal DNS resolution.

Step 7: Verify Inter-Server Connectivity

Once DNS and hostnames are configured, you can connect all three servers with each other using SSH or ping commands.

```
ping linux2.nagendrappa.xyz
```

```
ping linux3.nagendrappa.xyz
ssh linux@linux2.nagendrappa.xyz
ssh linux@linux3.nagendrappa.xyz
```

All three servers (Linux1, Linux2, Linux3) should be able to communicate internally using their DNS hostnames.

Optional Enhancements

• Add fallback entries to /etc/hosts if DNS resolution fails. • Use 'dig' or 'nslookup' to verify DNS records. • Repeat the steps for Linux2 and Linux3.