

1. Create a VPC (Virtual Private Cloud) – Name: Govardhan-vpc

AWS uses software-defined networks to offer a small network that is secure from others called a Virtual Private Cloud (VPC).

1. Find the VPC Service using the search at the top
2. Start the **VPC Wizard**. (**This is an important step** because the VPC Wizard sets up other necessary components of the VPC, such as the Internet Gateway. Please double check that you used the VPC Wizard and not just the “Create VPC” option.)
3. On the default tab, “VPC with a Single Public Subnet”, click Select. (The subnet is a subset of the whole AWS network that is available.)
4. On “Step 2”, you’ll need to change a few bits of information:
 - a. **IPv4 CIDR Block:** 10.0.0.0/26
 - b. **VPC Name:** *my-name-vpc*
 - c. **Public Subnet’s IPv4 CIDR:** 10.0.0.0/28
 - d. Everything else can be left as the default
1. The click “Create VPC”

Step 2: VPC with a Single Public Subnet

IPv4 CIDR block:* 10.0.0.0/26 (59 IP addresses available)

IPv6 CIDR block: ☒ No IPv6 CIDR Block
☐ Amazon provided IPv6 CIDR block

VPC name: david-vpc

Public subnet's IPv4 CIDR:* 10.0.0.0/28 (11 IP addresses available)

Availability Zone:* No Preference ↕

Subnet name: Public subnet

You can add more subnets after AWS creates the VPC.

Service endpoints

Enable DNS hostnames:* ☒ Yes ☐ No

Hardware tenancy:* Default ↕

Note for the curious: If you're unfamiliar with CIDR notation, the number after the '/' indicates the size of the network. For example, the 28 in the subnet specifies that the first 28 bits of the 32-bit IP address are the same, meaning the last 4 bits are variable. This means the subnet is made of all the IP addresses between 10.0.0.0 and 10.0.0.15. AWS reserves 5 of the IPs for networking (e.g. a router, DNS), leaving your subnet with 11 available IP addresses for instances.

You've now configured a secure VPC and subnet - use them whenever you spin up nodes. Let's setup budgets next to you can track your expenses in AWS.

2. Create Security Groups:

Created 3 security groups initially:

1. One to access ssh from home IP address
2. One to allow downloads
3. One to allow internal communication between machines with this security group

Follow this video: <https://www.youtube.com/watch?v=3taULsvuZUM&feature=youtu.be>

3. Create EC2 instances:

Created 3 EC2 instances – t2.small for Zookeepers and then SSH into the instance with the pem keypair file

Zookeepers:

10-0-0-4 – server 1
10-0-0-9 – server 2
10-0-0-7 – server 3

ssh -i ~*location of the pem file* ubuntu@*Ec2instancePublicDNS*

Create 3 EC2 instances – m4.large for PulsarBrokersand Bookie and then SSH again

Pulsar Brokers:

10-0-0-10
10-0-0-14
10-0-0-08

Install JAVA 8 on all the machines.

Sudo apt update

Sudo apt install openjdk-8-jdk

4. Follow <https://pulsar.apache.org/docs/en/deploy-bare-metal/> for installation of pulsar.

- Install the Pulsar binary package
- After that install tiered storage offloaders in all the broker systems
- Have to deploy a zookeeper cluster

Use :

```
echo server.1=ec2-34-221-137-48.us-west-2.compute.amazonaws.com:2888:3888 >>
./conf/zookeeper.conf
```

to add server to talk to. Make sure these are your zookeeper instance's IP

5. Enable the Pulsar functions by following the steps in:

<https://pulsar.apache.org/docs/en/functions-worker/>

6. After setting up all the nodes, need to create 3 more nodes for pulsar sql. Please follow the below link:

<https://pulsar.apache.org/docs/en/sql-deployment-configurations/>

7. Setup Tiered Storage by following the below link:

<https://pulsar.apache.org/docs/en/cookbooks-tiered-storage/>