



NextWork.org

# Launching VPC Resources



mallangiharinathreddy727@gmail.com

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. On the left, the 'VPC settings' section includes fields for 'Name tag auto-generation' (set to 'Auto-generated'), 'IPv4 CDR block' (set to '10.0.0.0/16'), 'IPv6 CDR block' (set to 'No IPv6 CDR block'), and 'Tenancy' (set to 'Default'). Below these are sections for 'Number of Availability Zones (AZs)' (set to 2) and 'Number of public subnets' (set to 2). On the right, the 'Preview' section displays a hierarchical tree of resources. It starts with 'Your AWS virtual network' containing 'NextWork-vpc'. This leads to 'Subnets (6)' under 'ap-southeast-1a' and 'ap-southeast-1b', each with three subnets: 'public1-ap-southeast-1a', 'private1-ap-southeast-1a', and 'private2-ap-southeast-1a'. These subnets are connected to 'Route tables (5)' via 'Network-rtb-publicC', 'Network-rtb-private1-ap-southeast-1a', 'Network-rtb-private2-ap-southeast-1b', 'Network-rtb-private3-ap-southeast-1a', and 'Network-rtb-private4-ap-southeast-1b'. Finally, these route tables connect to 'Network-igw' and 'Network-spine-s1' under 'Network connections (2)'.

# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC is a service that lets you create a private network within the AWS cloud. It's like having your own isolated section of AWS where you can launch and manage resources, such as EC2 instances, with full control over your network settings.

## How I used Amazon VPC in this project

I used VPC to create my own vpc with its resources.

## One thing I didn't expect in this project was...

The Resource map of our entire VPC.

## This project took me...

This project take 2.5 hours to complete including live demo.

# Setting Up Direct VM Access

Directly accessing a virtual machine means connecting to your EC2 instance without any intermediaries. This usually involves using tools like SSH (for Linux) or Remote Desktop (for Windows) to log in and interact with the instance

## SSH is a key method for directly accessing a VM

Secure Shell, is the protocol we use for this secure access to a remote machine. When you connect to the instance, SSH verifies you possess the correct private key corresponding to the public key on the server, ensuring only authorized users can access

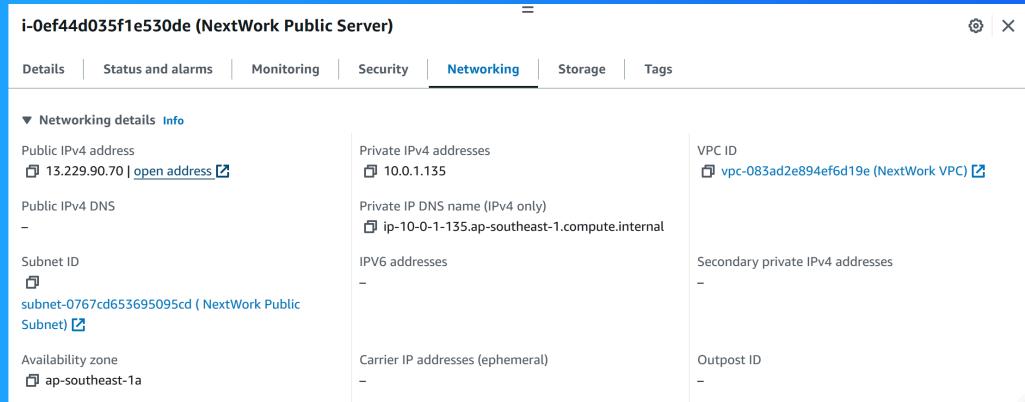
## To enable direct access, I set up key pairs

Key pairs are tools that helps developers/engineers authenticate themselves when trying to get direct access to virtual machine.eg EC2 instance.

A private key's file format means the file type that my keys stored in. My private key's file format was .pem which is widely accepted file format that most servers will be able to read/use.

# Launching a public server

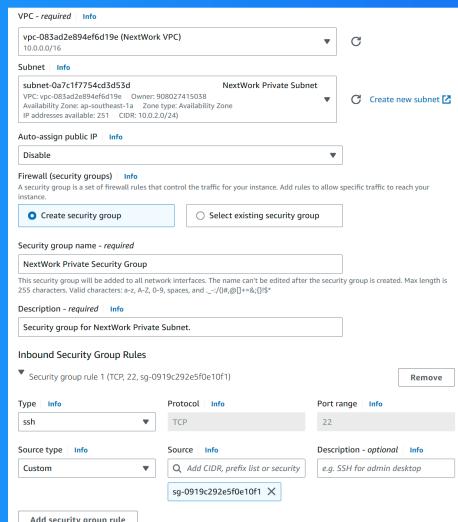
I had to change my EC2 instance's networking settings by changing the VPC



# Launching a private server

My private server has its own dedicated security group because the Nextwork public security group allows in ALL HTTP traffic -which would leave our private server much more vulnerable to security attacks/risks.

My private server's security group's source is my Nextwork public security group ,which means only SSH traffic coming from resources associated with that security group would be allowed.

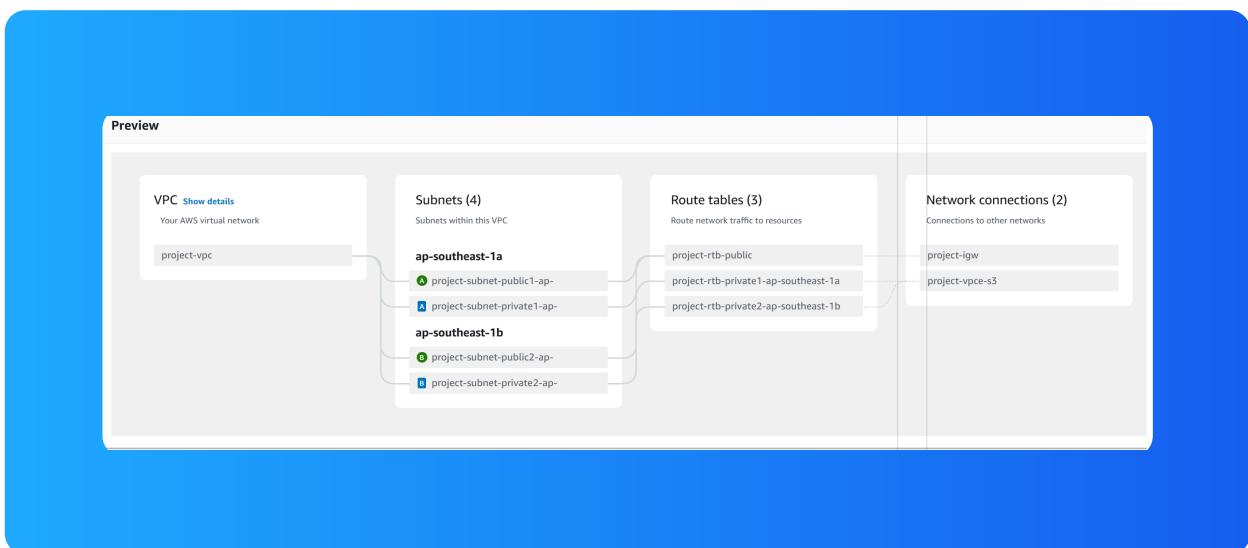


# Speeding up VPC creation

I used an alternative way to set up an Amazon VPC! This time, I used the 'VPC and more' option, which gives me a resource map to use when creating the VPC and all of its components. eg: security groups

A VPC resource map is a visual diagram that maps out my VPC components and the relationships/connections between them.

My new VPC has a CIDR block of 10.0.0.0/16 It is possible for my new VPC to have the same IPv4 CIDR block as my existing VPC because VPC's are already isolated from each other. still, this is not best practice if we need VPC peering.



# Speeding up VPC creation

## Tips for using the VPC resource map

When determining the number of public subnets in my VPC, I only had two options: either none or one in each availability zone for my VPC. This was because<sup>1</sup> it is best practice(improve redundancy and high availability) to have atleast subnet/AZ.

The set up page also offered to create NAT gateways, which are connectors/gateways that will let resources in my private subnet get access to the internet(eg: for security updates )while still blocking off incoming traffic from the internet.

The screenshot shows the 'Create VPC' setup page on the AWS Management Console. On the left, there are several configuration sections: 'VPC settings' (with 'VPC only' selected), 'Name tag auto-generation' (with 'Auto-generate' checked), 'IPv4 CIDR block' (set to 10.0.0.0/16), 'Tenancy' (set to 'Default'), 'Number of Availability Zones (AZs)' (set to 2), 'Number of public subnets' (set to 2), and 'Number of private subnets' (set to 4). On the right, a 'Preview' section displays a hierarchical diagram of the VPC resources. It shows a central 'Your AWS virtual network' node connected to 'Subnets (6)', 'Route tables (5)', and 'Network connections (2)'. The 'Subnets (6)' node is further expanded to show subnets grouped by availability zone: 'ap-southeast-1a' and 'ap-southeast-1b'. Each AZ contains two public subnets (green icons) and two private subnets (blue icons). The 'Route tables (5)' node lists route tables for each AZ, and the 'Network connections (2)' node lists connections to other networks named 'Nextwork-ipw' and 'Nextwork-vpc-e-s3'.



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

