

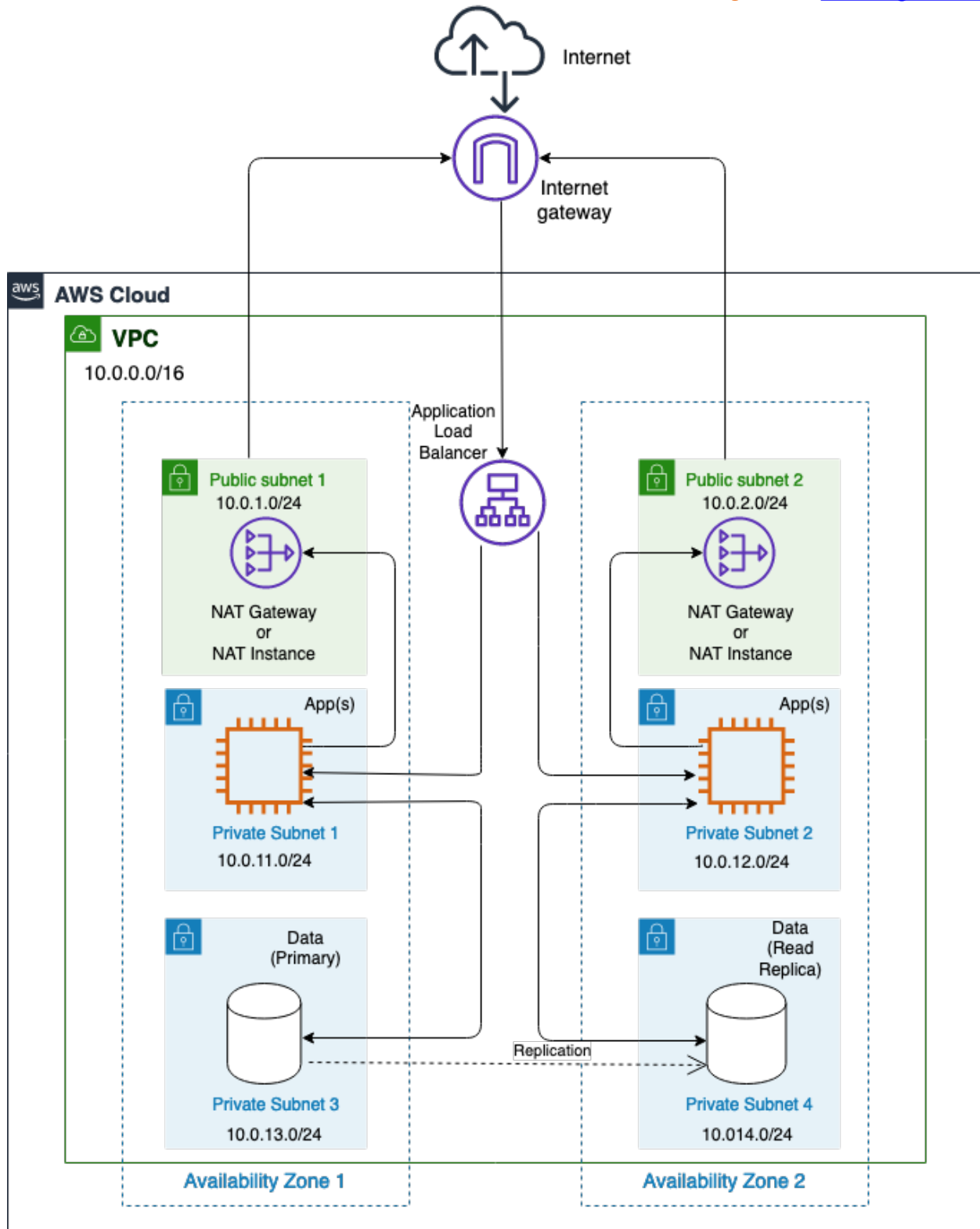


www.tngsllc.com info@tngsllc.com

Configuring and Deploying Amazon VPC for a 3-tier Web App

Lab Overview

This lab demonstrates how to set up a VPC infrastructure to support a basic 3-tier web application in your AWS Cloud environment. The VPC will be built across multiple Availability Zones so that your application is highly available. Let's get a quick overview of what a VPC is and how Elastic Load Balancing works.





www.tngsllc.com info@tngsllc.com

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

You can easily customize the network configuration of your Amazon VPC. For example, you can create a public-facing subnet for your web servers that have access to the internet. You can also place your backend systems, such as databases or application servers, in a private-facing subnet with no internet access. You can use multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

For more information about VPC you can view the AWS Documentation using the following link:

<https://docs.aws.amazon.com/vpc/latest/userguide/how-it-works.html>

Topics Covered

- Build a VPC
- Create public and private subnets
- Configure the internet gateway and NAT gateway

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic navigation of the AWS Management Console.



Task 1: Creating a VPC

In this task, you create a new VPC in the AWS Cloud.

- In the AWS Management Console, choose **Services** and select **VPC**. (You may also type VPC in the search bar and choose VPC.)
- Choose **Your VPCs** on the left navigation menu.

Note: You will see a default VPC (one is created whenever an AWS account is created). To learn more about default VPC, go to [Default VPC and default subnets](#)

- Choose **Create VPC** on the right side of the console.
- Select **VPC only**.
- Under **VPC settings** enter the following values.
- **Name tag:** Enter
- **IPv4 CIDR block:** Enter

Note: This VPC will not have an IPv6 CIDR block, and we will leave it with default tenancy.

- Choose **Create VPC**

A VPC with the specified CIDR block has been created. Now, let's create the subnets.

Task 2: Creating subnets

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet and a private subnet for resources that won't be connected to the internet. To protect the AWS resources in each subnet, you can



use multiple layers of security, including security groups and network access control lists (ACL).

In this task, you create two public subnets and four private subnets in the Lab VPC (as shown in the Lab Overview's architecture diagram above).

Create public subnets

The public subnets will be for internet-facing resources.

- In the left navigation pane, choose **Subnets**.

Note: You will see subnets for the default VPC. You can ignore them and go to the next step.

- Choose **Create subnet** and configure it with the following details:
- **VPC ID:** Select *LabVPC*
- **Subnet name:** Enter
- **Availability Zone:** Select the first Availability Zone in the list
- **IPv4 CIDR block:** Enter
- Choose **Create subnet**

Note: When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses). The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC) or a subset of the CIDR block for the VPC (for multiple subnets). If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap.

Tools are available on the internet to help you calculate and create IPv4 subnet CIDR blocks; for example, [IPv4 Address Planner](#).

One public subnet has been created. Now, create the other public subnet with the following details:

- **VPC ID:** Select
- **Subnet name:** Enter
- **Availability Zone:** Select the second Availability Zone in the list



- **IPv4 CIDR block:**

Note: Even though we named these subnets PublicSubnet, they are not yet public. A public subnet must have an internet gateway, which you will create and attach later in the lab.

Create private subnets

The private subnets are for resources that remain isolated from the internet. In this lab, we create two private subnets for the EC2 instances and two private subnets for the RDS instances.

- Create the four private subnets with the following details:
- **PrivateSubnet1:**
 1. **VPC ID:** Select **LabVPC**
 2. **Subnet name:** Enter
 3. **Availability Zone:** Select the first Availability Zone in the list
 4. **IPv4 CIDR block:** Enter
- **PrivateSubnet2:**
 5. **VPC ID:** Select **LabVPC**
 6. **Subnet name:** Enter
 7. **Availability Zone:** Select the second Availability Zone in the list
 8. **IPv4 CIDR block:** Enter
- **PrivateSubnet3:**
 9. **VPC ID:** Select **LabVPC**
 10. **Subnet name:** Enter
 11. **Availability Zone:** Select the first Availability Zone in the list
 12. **IPv4 CIDR block:** Enter
- **PrivateSubnet4:**
 13. **VPC ID:** Select **LabVPC**
 14. **Name tag:** Enter



15. **Availability Zone:** Select the second Availability Zone in the list

16. **IPv4 CIDR block:** Enter

Your VPC now has six subnets in total: two public subnets for the Application Load Balancer in two different Availability Zones, two subnets for the EC2 instances in two different Availability Zones, and two subnets for the RDS instances in two different Availability Zones.

Task 3: Creating an internet gateway

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in a VPC and the internet. An internet gateway does not impose availability risks or bandwidth constraints on network traffic.

An internet gateway serves two purposes:

- Provide a target in route tables to connect to the internet
- Perform network address translation (NAT) for instances that have been assigned public IPv4 addresses

In this task, you create an internet gateway so that internet traffic can access the public subnets.

- In the left navigation pane, choose **Internet Gateways**.

Note: A default internet gateway was created with the default VPC. You can ignore this and proceed with the next step.

- Choose **Create internet gateway** and configure:
- **Name tag:** Enter
- Choose **Create internet gateway**



Once it's created, you need to attach the internet gateway to your Lab VPC.

- Choose **Actions > Attach to VPC**.
- For **VPC**, select **LabVPC**
- Choose **Attach internet gateway**

The internet gateway is now attached to your Lab VPC. You have created an internet gateway and attached it to your VPC, and you must now configure the route table of the public subnets to use the internet gateway. You'll do that after you create your NAT gateway.

Task 4: Creating a NAT gateway

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services. The NAT gateway also prevents the internet from initiating a connection with those instances. For more information, see [NAT](#).

In this task, you create a NAT gateway so that the private subnets can connect to the internet.

- In the left navigation pane, choose **NAT Gateways**.
- Choose **Create NAT gateway** and configure:
 - **Name:** Enter
 - **Subnet:** Select *PublicSubnet1*
 - **Connection Type:** **Public**
 - **Elastic IP allocation ID:** Choose **Allocate Elastic IP**

This will generate an Elastic IP address and will allocate it to the NAT gateway.



www.tngsllc.com info@tngsllc.com

An Elastic IP address is a public IPv4 address that is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet. For example, this allows you to connect to your instance from your local computer.

- Choose **Create NAT gateway**

The NAT gateway has been created. You must now configure the route table of the private subnets to use the NAT gateway.

Task 5: Configuring route tables

A route table contains a set of rules, called *routes*, used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table, which controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

In this task, you configure two route tables, one for the public subnets and one for the private subnets.

Configure public route table.

To use an internet gateway, a subnet's route table must contain a route that directs internet-bound traffic to the internet gateway. This subnet is called a *public subnet*.

You will:



- Create a public route table for internet-bound traffic
- Add a route to the route table to direct internet-bound traffic to the internet gateway
- Associate the public subnets with the new route table
- In the left navigation pane, choose **Route Tables**.

Several route tables are displayed, but there is only one route table associated with the Lab VPC. This is the main route table.

- Choose **Create route table** and configure:
 - **Name tag:** Enter
 - **VPC:** Select *LabVPC*
 - Choose **Create** and then choose **Close**
- Select **PublicRouteTable** and choose the **Routes** tab.
- Choose **Edit routes**

Now, add a route to direct internet-bound traffic (0.0.0.0/0) to the internet gateway.

- Choose **Add route** and configure:
 - **Destination:** Enter
 - **Target:** Select **Internet Gateway** and *LabVPCInternetGateway*.
 - Choose **Save routes** and then choose **Close**

The last step is to associate this new route table with the public subnets.

- Choose the **Subnet Associations** tab.
- Choose **Edit subnet associations**
- On the Edit subnet associations page, select the rows with *PublicSubnet1* and *PublicSubnet2*.
- Choose **Save**

The PublicSubnet is now public because it has a route table entry that sends traffic to the internet via the internet gateway.



Configure private route table

To use a NAT gateway, you must update the route table associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. This enables instances in your private subnets to communicate with the internet, mainly for any upgrades that needs to be installed.

You will:

- Create a private route table for private subnets
- Add a route to the route table for the NAT gateway
- Associate the private subnets with the new route table
- Choose **Create route table** and configure it with the following details:
- **Name tag:** Enter
- **VPC:** Select *LabVPC*
- Choose **Create** and then choose **Close**
- Select **PrivateRouteTable** and choose the **Routes** tab.
- Choose **Edit routes**

In the next step, you add a route to direct internet-bound traffic (0.0.0.0/0) to the NAT gateway.

- Choose **Add route** and configure:
- **Destination:** Enter
- **Target:** Select *NAT Gateway* and *LabVPCNATGateway*.
- Choose **Save routes** and then choose **Close**

The last step is to associate this new route table with the private subnets.

- Choose the **Subnet Associations** tab.
- Choose **Edit subnet associations**
- On the Edit subnet associations page, select all of the rows with *PrivateSubnet*.



www.tngsllc.com info@tngsllc.com

Note: You may need to expand the Subnet ID column to see their full names.

- Choose **Save**

You have successfully created a route table and associated the private subnets to the NAT gateway in the public subnet so that the incoming traffic can reach the private subnets.

Note: If you have resources in multiple Availability Zones that share one NAT gateway, and if the NAT gateway's Availability Zone is down, resources in the other Availability Zones lose internet access. To create an AZ-independent architecture, create a NAT gateway in at least two Availability Zones that Amazon CloudWatch can monitor and configure your routing to ensure resources use the NAT gateway in the same Availability Zone.

Conclusion

Congratulations! You now have successfully:

- Created a VPC
- Created public and private subnets
- Created an internet gateway and a NAT gateway
- Configured route tables and associated them to subnets