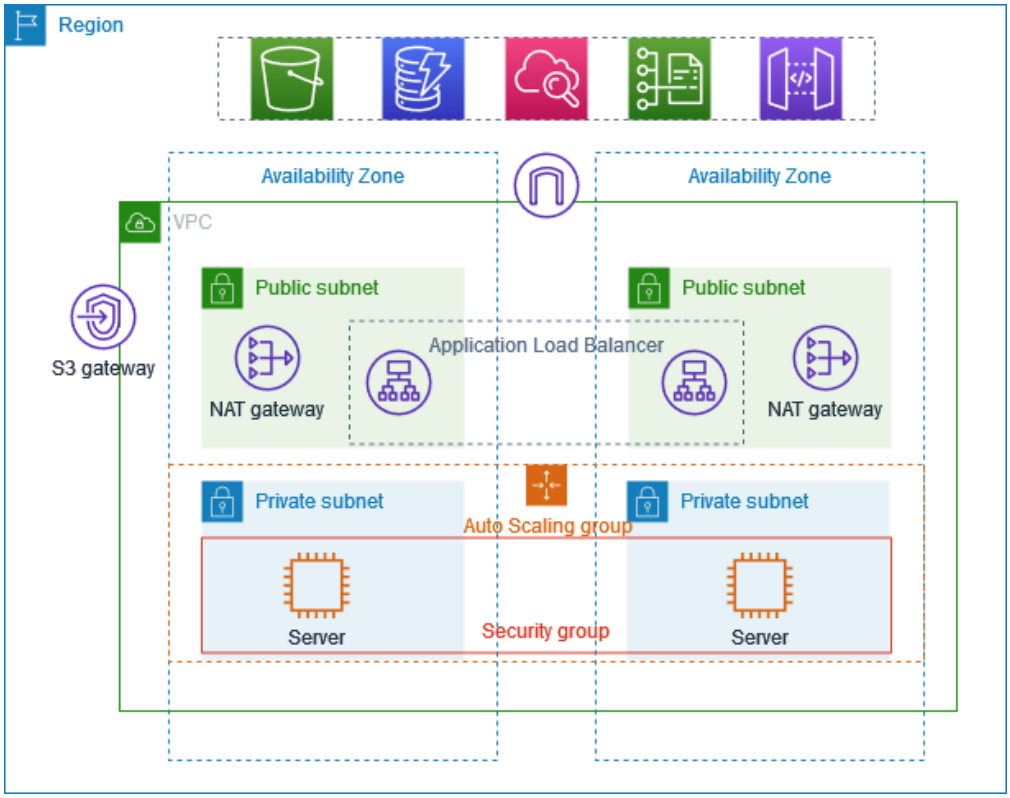
VPC-with-servers-in-private-subnets-and-NAT

This demonstrates how to create a VPC that you can use for servers in a production environment. To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both Availability Zones.

Contents

Overview

The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway. The servers can connect to Amazon S3 by using a gateway VPC endpoint.



Routing

When you create this VPC by using the Amazon VPC console, we create a route table for the public subnets with local routes and routes to the internet gateway. We also create a route table for the private subnets with local routes, and routes to the NAT gateway, egress-only internet gateway, and gateway VPC endpoint.

The following is an example of the route table for the public subnets, with routes for both IPv4 and IPv6. If you create IPv4-only subnets instead of dual stack subnets, your route table includes only the IPv4 routes.

Destination	**Target**
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	*igw-id*
::/0	*igw-id*

The following is an example of a route table for one of the private subnets, with routes for both IPv4 and IPv6. If you created IPv4-only subnets, the route table includes only the IPv4 routes. The last route sends traffic destined for Amazon S3 to the gateway VPC endpoint.

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	*nat-gateway-id*
::/0	*eigw-id*
s3-prefix-list-id	*s3-gateway-id*

Security

The following is an example of the rules that you might create for the security group that you associate with your servers. The security group must allow traffic from the load balancer over the listener port and protocol. It must also allow health check traffic.

Inbound			
Source	**Protocol**	**Port range**	**Comments**
ID of the load balancer secur	*listener protocol*	*listener port*	Allows inbound traffic from the load

Inbound			
Source	**Protocol**	**Port range**	**Comments**
ity group			balancer on the listener port
ID of the load balancer security group	*health check protocol*	*health check port*	Allows inbound health check traffic from the load balancer

---

Create the VPC

Use the following procedure to create a VPC with a public subnet and a private subnet in two Availability Zones, and a NAT gateway in each Availability Zone.

To create the VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the dashboard, choose **Create VPC**.
3. For **Resources to create**, choose **VPC and more**.
4. **Configure the VPC**
    a. For **Name tag auto-generation**, enter a name for the VPC.

    b. For **IPv4 CIDR block**, you can keep the default suggestion, or alternatively you can enter the CIDR block required by your application or network.
    c. If your application communicates by using IPv6 addresses, choose **IPv6 CIDR block**, **Amazon-provided IPv6 CIDR block**.
5. **Configure the subnets**
    a. For **Number of Availability Zones**, choose **2**, so that you can launch instances in multiple Availability Zones to improve resiliency.
    b. For **Number of public subnets**, choose **2**.
    c. For **Number of private subnets**, choose **2**.
    d. You can keep the default CIDR block for the public subnet, or alternatively you can expand **Customize subnet CIDR blocks** and enter a CIDR block. For more information, see Subnet CIDR blocks.
6. For **NAT gateways**, choose **1 per AZ** to improve resiliency.
7. If your application communicates by using IPv6 addresses, for **Egress only internet gateway**, choose **Yes**.
8. For **VPC endpoints**, if your instances must access an S3 bucket, keep the **S3 Gateway** default. Otherwise, instances in your private subnet can't access Amazon S3. There is no cost for this option, so you can keep the default if you might use an S3 bucket in the future. If you choose **None**, you can always add a gateway VPC endpoint later on.
9. For **DNS options**, clear **Enable DNS hostnames**.
10. Choose **Create VPC**.

Deploy your application

Ideally, you've finished testing your servers in a development or test environment, and created the scripts or images that you'll use to deploy your application in production.

You can use Amazon EC2 Auto Scaling to deploy servers in multiple Availability Zones and maintain the minimum server capacity required by your application.

To launch instances by using an Auto Scaling group

1. Create a launch template to specify the configuration information needed to launch your EC2 instances by using Amazon EC2 Auto Scaling. For step-by-step directions, see Create a launch template for your Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.

2. Create an Auto Scaling group, which is a collection of EC2 instances with a minimum, maximum, and desired size. For step-by-step directions, see Create an Auto Scaling group using a launch template in the *Amazon EC2 Auto Scaling User Guide*.
3. Create a load balancer, which distributes traffic evenly across the instances in your Auto Scaling group, and attach the load balancer to your Auto Scaling group. For more information, see the Elastic Load Balancing User Guide and Use Elastic Load Balancing in the *Amazon EC2 Auto Scaling User Guide*.

Test your configuration

After you've finished deploying your application, you can test it. If your application can't send or receive the traffic that you expect, you can use Reachability Analyzer to help you troubleshoot. For example, Reachability Analyzer can identify configuration issues with your route tables or security groups. For more information, see the Reachability Analyzer Guide.

Clean up

When you are finished with this configuration, you can delete it. Before you can delete the VPC, you must delete the Auto Scaling group, terminate your instances, delete the NAT gateways, and delete the load balancer. For more information, see Delete your VPC.