**"Setting Up and Configuring Nginx on EC2: A Step-by-Step Guide"**

1. Create an EC2 instance, ensuring that you download a key pair. Also, configure network settings to allow HTTP requests from anywhere.

2. Connect to the server using Git Bash with SSH: ssh -i "tingu-server.pem" ubuntu@ec2-65-2-140-82.ap-south-1.compute.amazonaws.com

3. Update the Ubuntu package list with: sudo apt-get update

4. Install Nginx with the command: sudo apt install nginx

5. Check the status of the Nginx web server: sudo systemctl status nginx

6. Use the public IP address of the EC2 instance to perform a Google search.

7. In a new Git Bash session, copy the HTML file to the server using scp -i "tingu-server.pem" tingu.html ubuntu@ec2-65-2-140-82.ap-south-1.compute.amazonaws.com:/tmp/

8. Move the HTML file from /tmp to /var/www/html with: sudo mv /tmp/tingu.html /var/www/htm

9. To access the site, use the public IP address of the instance, like 65.2.140.82/tingu.html

10. To make tingu.html load by default when the IP address is entered, edit the Nginx default configuration: sudo vim /etc/nginx/sites-available/default and change the index from index.html to tingu.html

11. Verify the Nginx configuration with: sudo nginx -t

12. Restart the Nginx server: sudo systemctl restart nginx

13. You're all set! Your changes are now in effect.

**Load Balancers**

Both Application Load Balancers (ALBs) and Network Load Balancers (NLBs) are used to distribute traffic to different servers, with ALB focusing on application-specific routing and NLB focusing on lower-level transport protocol routing.

**VPC**

If you don't already know amazon has a massive global footprint with regions and availability zones all around the world.

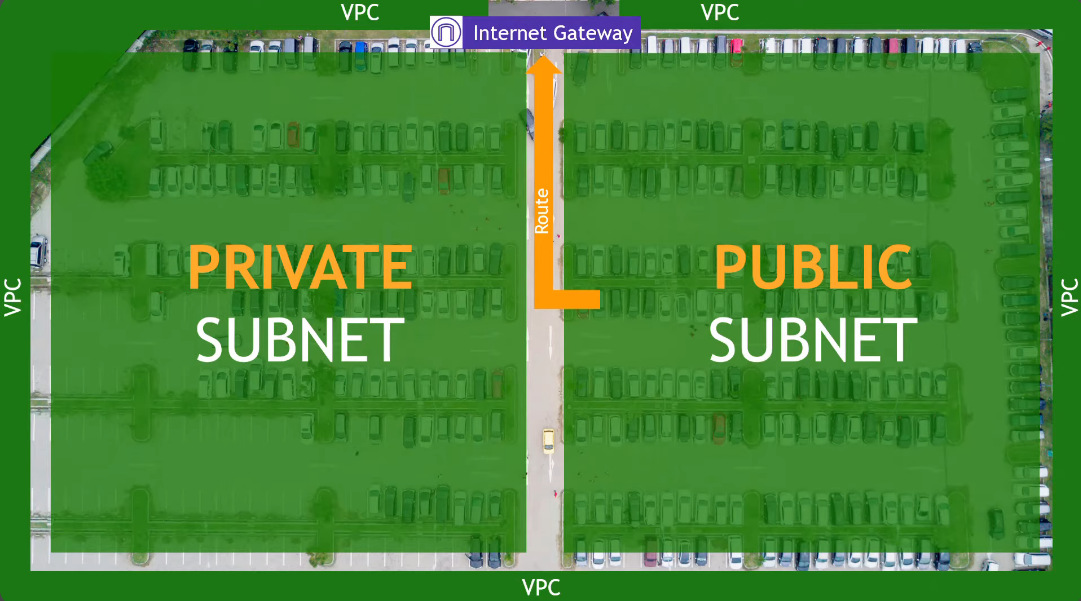
There are hundreds of thousands even millions of customers using AWS every day. They're using ec2 instances and s3 buckets, databases and so on.

You don't want to just throw your resources out there with all the other millions of resources. Instead, you want your own little private slice of the overall AWS pie and that's basically what a virtual private cloud or VPC is.

It's your own private cloud or network within the cloud within AWS and it's how your resources are isolated from everybody else's resources.

When you create your AWS account you get a default VPC and then of course you can create others.

(The default VPC in AWS is a pre-configured VPC that comes with a public subnet in each Availability Zone, an internet gateway, and settings to enable DNS resolution. This makes it easy to get started with AWS and to launch resources quickly. However, the default VPC is not as secure or customizable as a custom VPC. For example, the default VPC has a security group that allows all traffic from the internet to all resources in the VPC. This is not a good security practice)



Imagine there is a parking lot and outside of the parking lot is the whole world.

The fence or border around the parking lot can be thought of as your VPC. It separates your resources from the rest of the resources in AWS.

Up at the top, we have traffic coming in from the outside world and traffic going out to the outside world. In AWS networking terms this would be an internet gateway. That's how your VPC talks to the internet.

Inside the VPC we have logically separated areas. For our example we're going to say they're the left and the right of the parking lot. These are called your subnets in AWS networking terms.

Now let's say that one subnet has a route to get out to the internet. If that's the case this is a public subnet. The other side, the subnet that doesn't have the route to the internet that's a private subnet

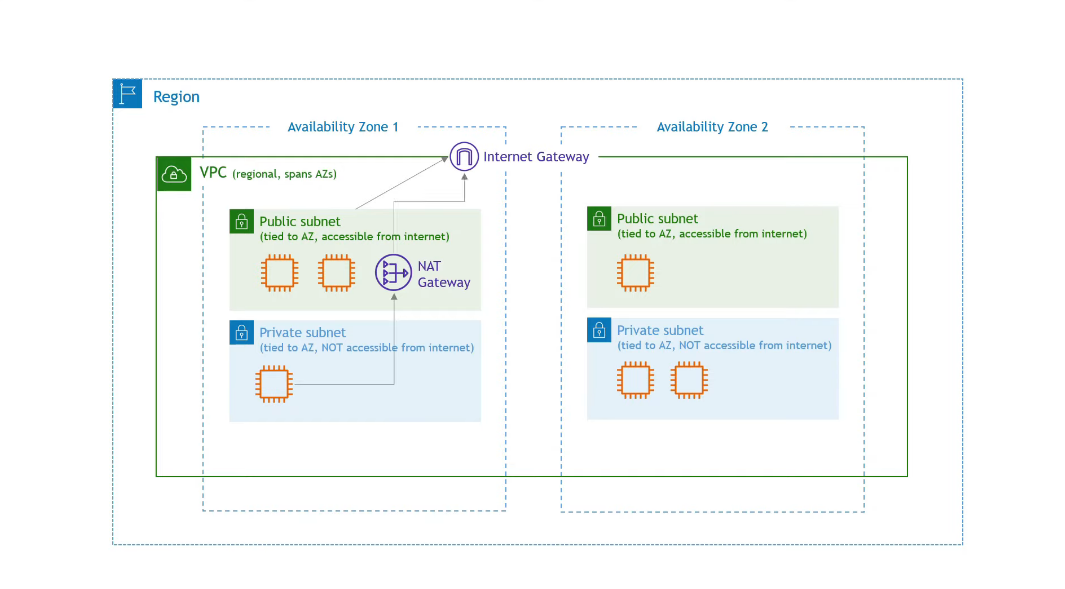
So maybe on the public subnet you put servers that would host public-facing web pages and in the private subnet you would put your databases so there's that extra layer of security protecting from the outside world.

Within the subnets is where your actual resources live like your ec2 instances and so on.

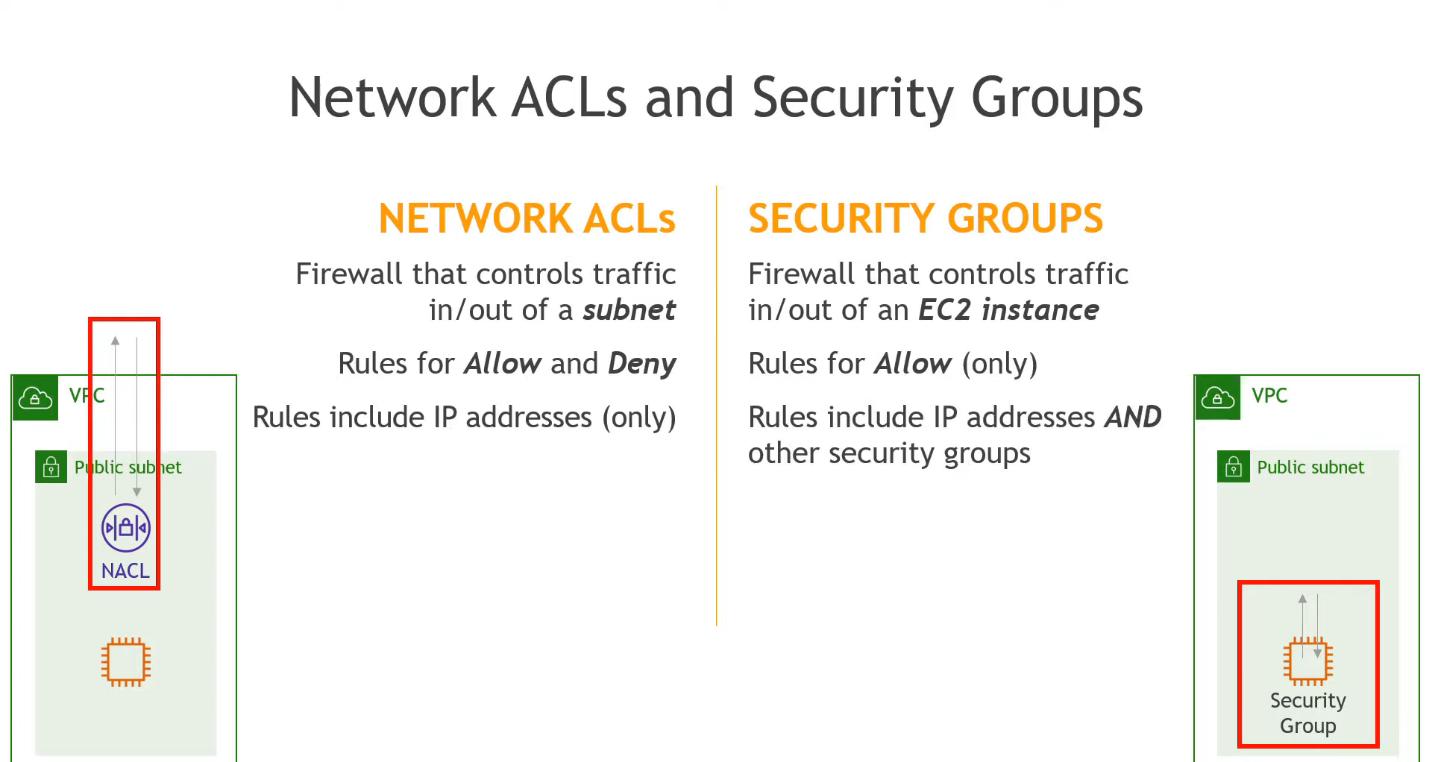
Remember, internet gateway is at the VPC level and the fact that a public subnet has a route to it is what makes it public. It's accessible to and from the internet. The private subnet however doesn't have that route. You can't get to it from the internet and you can't get out to the internet.

But what if you do need to get to the internet for something? Maybe to do updates or patching or to download files? But at the same time, you don't want the outside world getting in.

In this case you want to use what's called a NAT gateway. NAT stands for network address translation.



Now the question is how to secure everything we have created? The answer is using network ACLs or access control lists and security groups.



To summarize, Amazon VPC is like a big box inside which you can create your own private network environment. A VPC is a region-level resource, and subnets must be created within an Availability Zone.