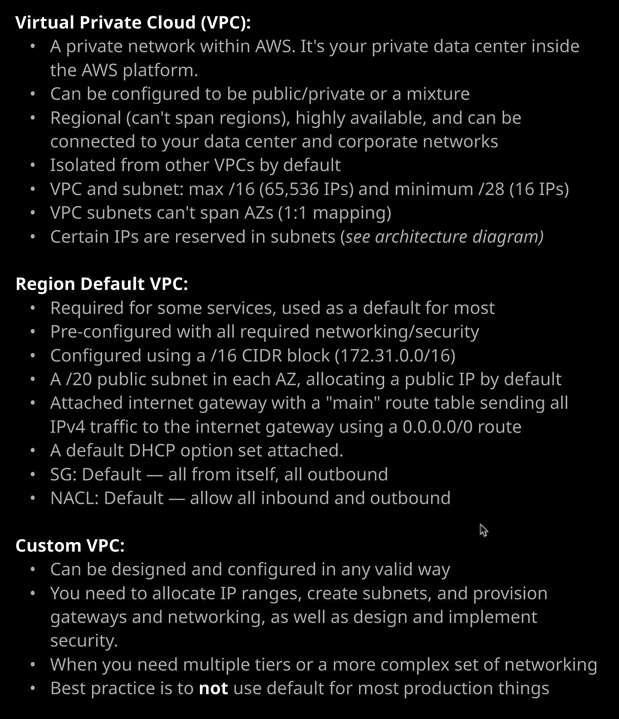
**AWS private networking**  
I want to introduce private AWS networking and that uses a product called **Virtual Private Cloud or VPC**. **VPCs are the way that you can define isolated networks inside of AWS**. So AWS is a public cloud platform. So you've got many customers accessing the same infrastructure/ Accounts provide some isolation, meaning that you can only see your own resources but **VPCs provide that networking isolation.** So one VPC is isolated from other VPCs and that includes other VPCs in the same region, other VPCs in different regions, and even VPCs in different accounts. **By default, unless you configure it otherwise VPCs cannot talk to anything outside themselves.** So any resources inside VPCs are isolated from anything beyond that VPC boundary unless you configure it otherwise. Now you might hear me refer to this as an **isolated network blast radius**. Essentially, if something happens in one VPC say a malware infection or an attacker than it's isolated to that VPC.



So I'm going to move across to the VPC area of the console. Now I've already mentioned how VPCs are virtual private clouds the private networking construct inside AWS. They essentially provide private networking, but they can be made to be public, depending on how you configure it. Now VPCs are regional based products, and you'll select the region with the region drop down or by setting it on the command line, and if you create a VPC, it will be created in the specific region that you're in. To do that, I'll go to your VPCs. Now **there are two types of VPC that you can create inside AWS, a custom VPC and a default VPC.** So you'll here me use both of those terms. A default VPC is something that exists by default in every region of a new AWS account. Now it can be deleted, which is what I've done here just to illustrate how they're created but by default, it's created by AWS and it's designed to remove the need to think about the networking too much when you're just starting to use services. So earlier in the course when we provisioned an EC2 instance the reason we didn't need to think about VPCs back then was because we utilized the default VPC. Default VPCs are needed for certain services so certain services have to be deployed into the default VPC, and historically certain services actually failed if the default VPC didn't exist. To make it more difficult, the default VPC was initially not something that you could create, but you could delete it. So if you deleted it, you could run into problems where certain services wouldn't launch and you needed to log an AWS support ticket on your behalf to get it recreated. More recently, though, you can create a default VPC yourself and you do that by going to actions, create default VPC, hit Create. It'll take a couple of seconds. You can hit Close, and that's it. Done.

**The default VPC is created in a very specific way.** It's configured using **a /16 CIDR block**, and it uses the **IP range of 172.31.00/16**. So that's a CIDR block that includes over **65,000 IP addresses** that are available for use. Inside that default VPC, it also creates a public subnet inside every availability zone in that region and these subnets use a **/20 prefix**, which includes **4,091 available IP addresses**. So this is what the default VPC comes with, and additionally, it's also got an attached internet gateway, which we'll talk about in upcoming lesson in this topic of the course, it's got a default DHCP option set created, which again we'll talk about towards the end of this lesson and it comes with a default security group and a default network ACL. Both of which you're going to expand on inside this topic and that's the default VPC. So if you do have one in your account and you launch any resources, it will be the default. So we see too if you don't choose to select anything else, it will use this default VPC to launch that instance into and certain other services such as data pipeline or other services whilst you can't specify VPC if you don't do they use the default one.

Now beyond that, you can create a custom VPC and custom **VPCs can be configured however you want, but you are responsible for every step of the configuration,** so VPCs are regional based and to create one it's a simple as hitting create VPC. Now VPCs support both IP version four and IP version six **but IP version four is the only mandatory version,** so you have to pick an IP version four CIDR block to allocate to the VPC, and there are all limits. **The maximum size of a VPC is a /16 which gives 65,000 IP addresses, and the minimum size of the VPC is /28 which gives 16 IP addresses** but be careful of this because this can be too small to support many services so you need to pay specific attention to what CIDR block you give to a VPC. Now a VPC can also be provided with an IP version six range. It's a /56 range, which gives you over four trillion trillion IP version six addresses, and they're all public. So you do have the option of **using IP version six but this note that it's not fully supported across all AWS products and services yet**.

You also need to decide on a **VPC tenancy. Now, options here are either default or dedicated.** Now be really careful here because picking **dedicated locks this VPC on dedicated hardware**. If you pick **default then it uses shared hardware, which you can later change so if you've got a shared hardware VPCs so you used default you can lead to choose to deploy resources into it that use dedicated hardware for you.** But if you pick dedicated, it does lock that to dedicated hardware, and you can't later choose to launch services into it that use a shared hardware. So unless you absolutely know that you need dedicated tenancy, then you should always default to default. The main choice here is to pick the IP version four CIDR. In the fundamentals part of the course, I talked about subnetting that's taking a big IP range and breaking it into small ones. Picking your VPC CIDR is important because it's this range that you'll be breaking up for your subnets, and you need to make sure that your subnets themselves have enough IP addresses to support whatever services you'll be deploying into them, and you'll need to break this down in a certain way based on how many availability zones you'll be using and then how many application tiers your VPC will have. So this is things like a public tier, an app tier, and a database tier. So for this VPC that I'm creating to demonstrate this, I'm going to choose not to allocate an IP version six CIDR lock and I'm going to call it VPC demo to keep things simple and then for the IP version four CIDR I'm going to pick 10.0.0.0/16 which will give me over 65,000 IP addresses. So once I've set that I'll hit Create follow up with close and that's my custom VPC created. So that's this one VPC demo and if I scroll across to the right, I'll be able to see that the initial VPC I created is the default VPC and this one is set to no. So this is the not default VPC this is a custom VPC. Now something to think about when making VPCs is this CIDR range that you use not just because of breaking it up for subnets, but to save yourself hassle later on. There are lots of networking features which really don't like the same CIDR block. So where possible**, you've got to ensure that the VPCs, your corporate network, and any other VPCs that you work with any partners and vendors that you interact with have non overlapping CIDR blocks.** It will just make things a lot easier further down line.