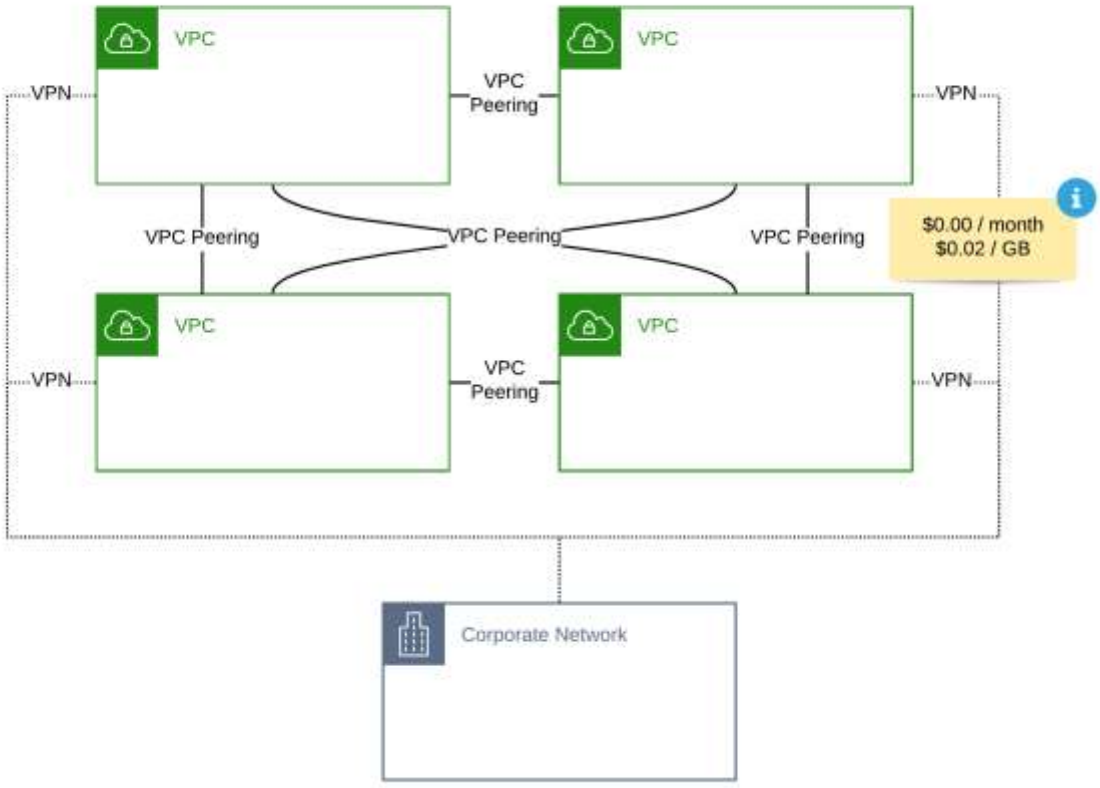
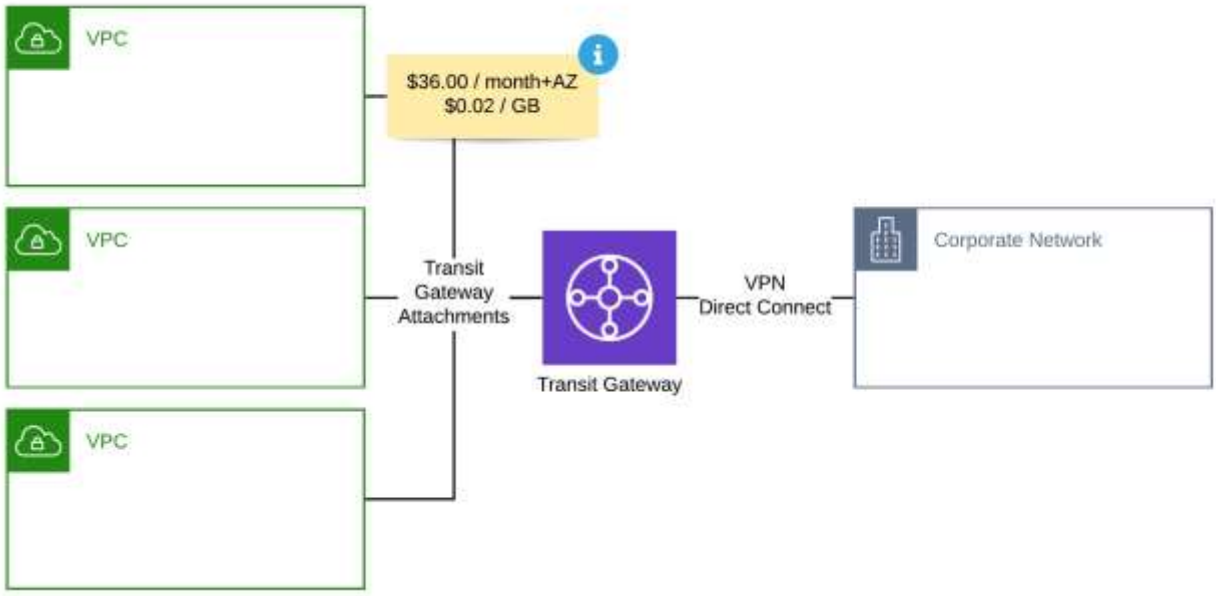
Advanced AWS Networking

AWS offers shiny and powerful networking services. However, you should know about the pitfalls when designing advanced networking architectures for AWS.

VPC Peering or Transit Gateway?

There are two different approaches to connect multiple VPCs. One option is to use VPC Peering. Creating a peering connection is simple: the owner of VPC A creates a peering request, and the owner of VPC B accepts the peering request. After the virtual peering is in place, all you need to do is to update the routing tables.





	VPC Peering	Transit Gateway
Create a network mesh with minimal configuration effort.	✗	✓
Reuse the same VPN connection for multiple VPCs.	✗	✓
Full flexibility to configure routing between networks?	✓	✓
Bandwith Limitation	No	50 Gbps (burst)
Connect networks across AWS accounts?	✓	✓
Connect networks across AWS regions?	✓	✓
Baseline Costs (per connected VPC and month)	$0.00	$36.00
Traffic Costs (per GB)	$0.02	$0.02

A pricing example:

- Connect 4 VPCs with each other
- 2 VPCs connected with the on-premises network via VPN
- 1,000 GB traffic between VPCs
- 500 GB outgoing traffic via VPN

	VPC Peering	Transit Gateway	
VPC Peerings / TGW attachments		$0.00	$180.0
VPN connections		$72.00	$36.0
Traffic between VPCs		$20.00	$20.0
Traffic across VPN		$45.00	$45.0
Total Monthly Costs		**$137.00**	**$281.0**

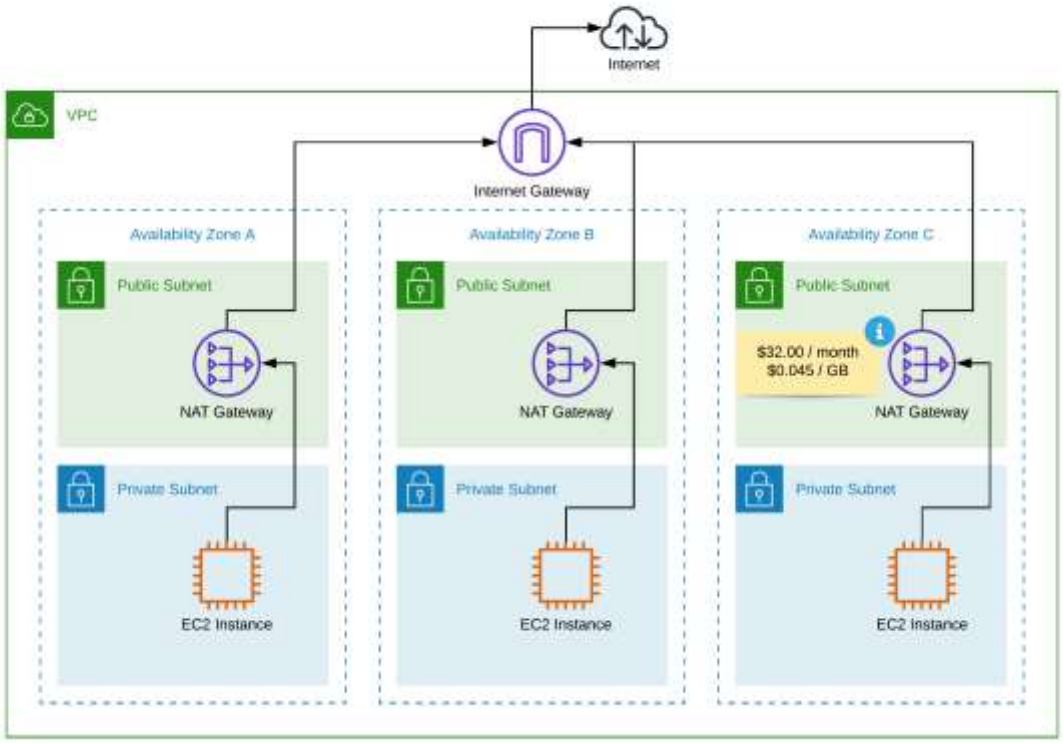
So using Transit Gateway doubles monthly costs for your networking infrastructure. Let's have a look at the pricing details.

- Attaching a VPC to a Transit Gateway costs $36.00 per month.
- A VPN connection costs $36.00 per month.
- Traffic costs are the same for VPC Peering and Transit Gateway.

NAT Gateway or Public Subnet?

AWS advocates to divide your VPC into public and private subnets. A public subnet comes with a rout to the Internet Gateway. Therefore, a public subnet enables incoming and outgoing connections to the Internet. A private subnet is neither accessible from the Internet, nor is it possible to establish a connection to the Internet.

However, it is quite common that EC2 instances need to connect to a resource via the Internet. This is where the NAT Gateway comes in. The NAT Gateway enables outgoing Internet connectivity for a private subnet. It is important to note that you need to create a NAT Gateway for every Availability Zone that you have created private subnets to achieve high availability.



The described network architecture consisting of public subnets, private subnets, and NAT gateways works fine but comes with two downsides.

If keeping costs to a minimum is essential, the baseline costs of $32.00 per month per NAT Gateway could be a show stopper. When using three AZs, you will pay $96.00 per month for three NAT Gateways.

The NAT Gateway also increases costs for outbound traffic. You have to pay a premium of $0.045 per GB flowing from a private subnet to the Internet. That's raising the costs for outgoing traffic by 50%.

Let's have a look at a pricing example.

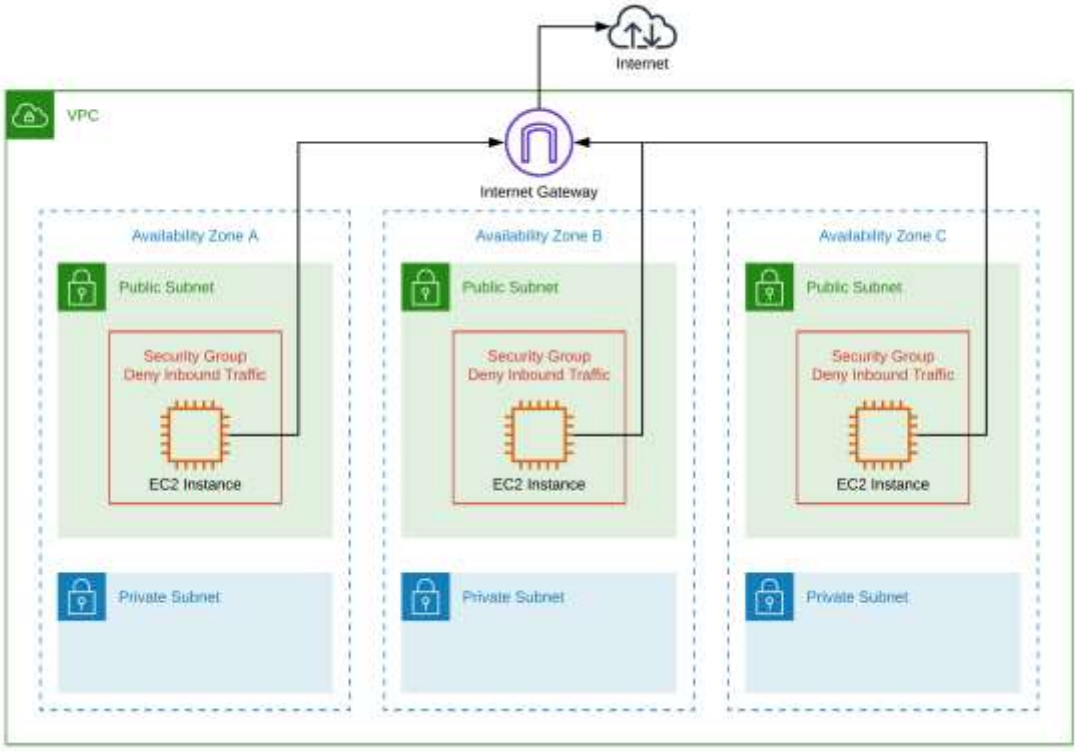
Outgoing Traffic	100 GB	1,000 GB	10,000 GI
With NAT Gateway	$13.50	$135.00	$1350.0
Without NAT Gateway	$9.00	$90.00	$900.0

In summary, following the standard network architecture on AWS can become quite costly.

Keep in mind that the NAT Gateway bandwidth can scale up to 45 Gbps. Check out our CloudFormation templates to learn how to monitor this resource limit!

What's the alternative? Place your workload - most likely, your EC2 instances - into the public subnets. By doing so, the NAT Gateways are no longer needed. Compared to the example with NAT Gateway from above, you will save $141.00 per month.

Keep in mind that your EC2 instances are now reachable from the Internet via their public IP addresses. Make sure you are blocking unwanted incoming traffic by making use of Security Groups and Network Access Control Lists.

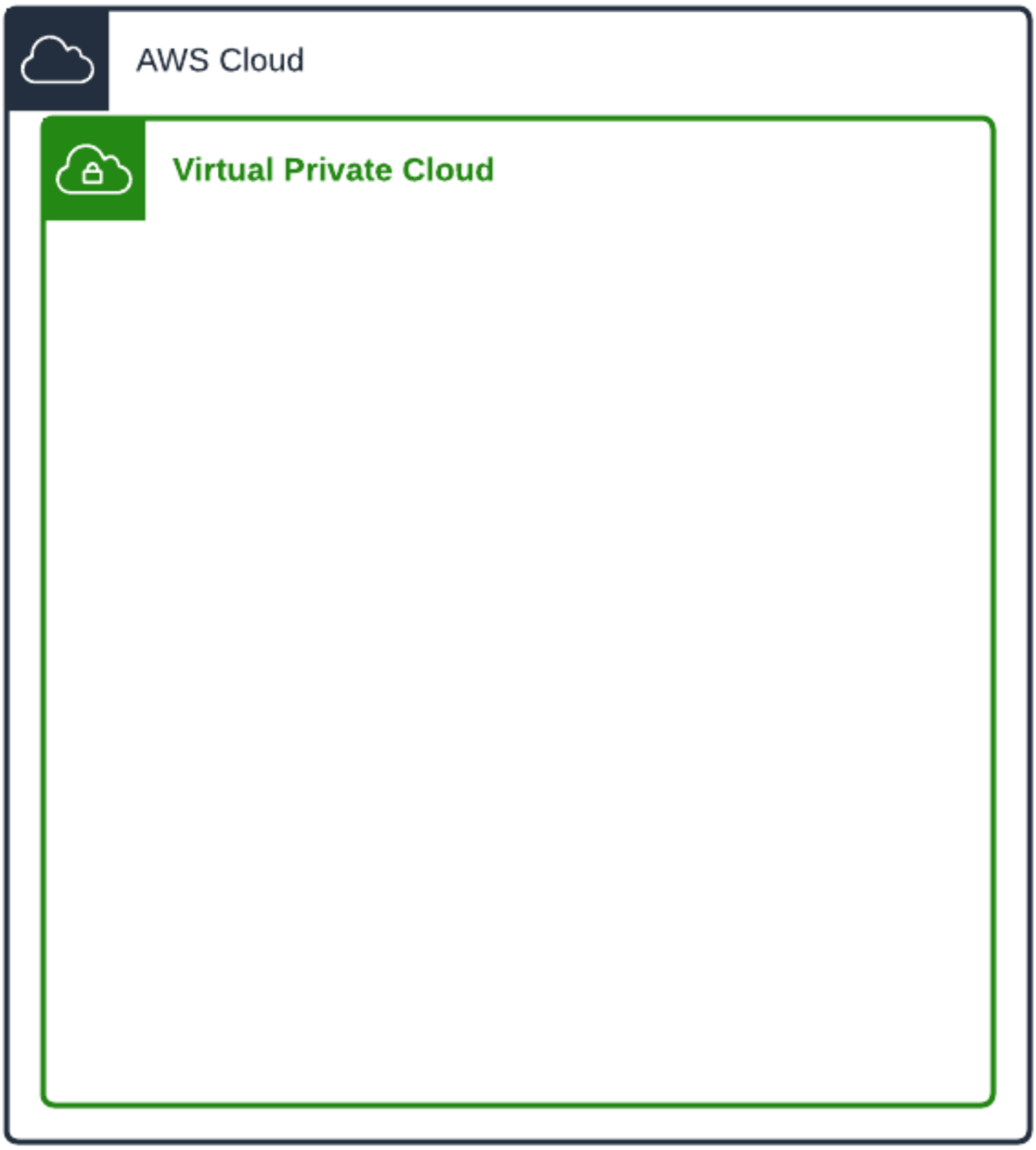


When designing the network architecture, keep the costs for the NAT Gateways in mind. I have to mention at well, that deviation from the standard can cause troubles. For example, an external auditor is most likely not happy with placing an EC2 instance into a public subnet.

High-Level Overview

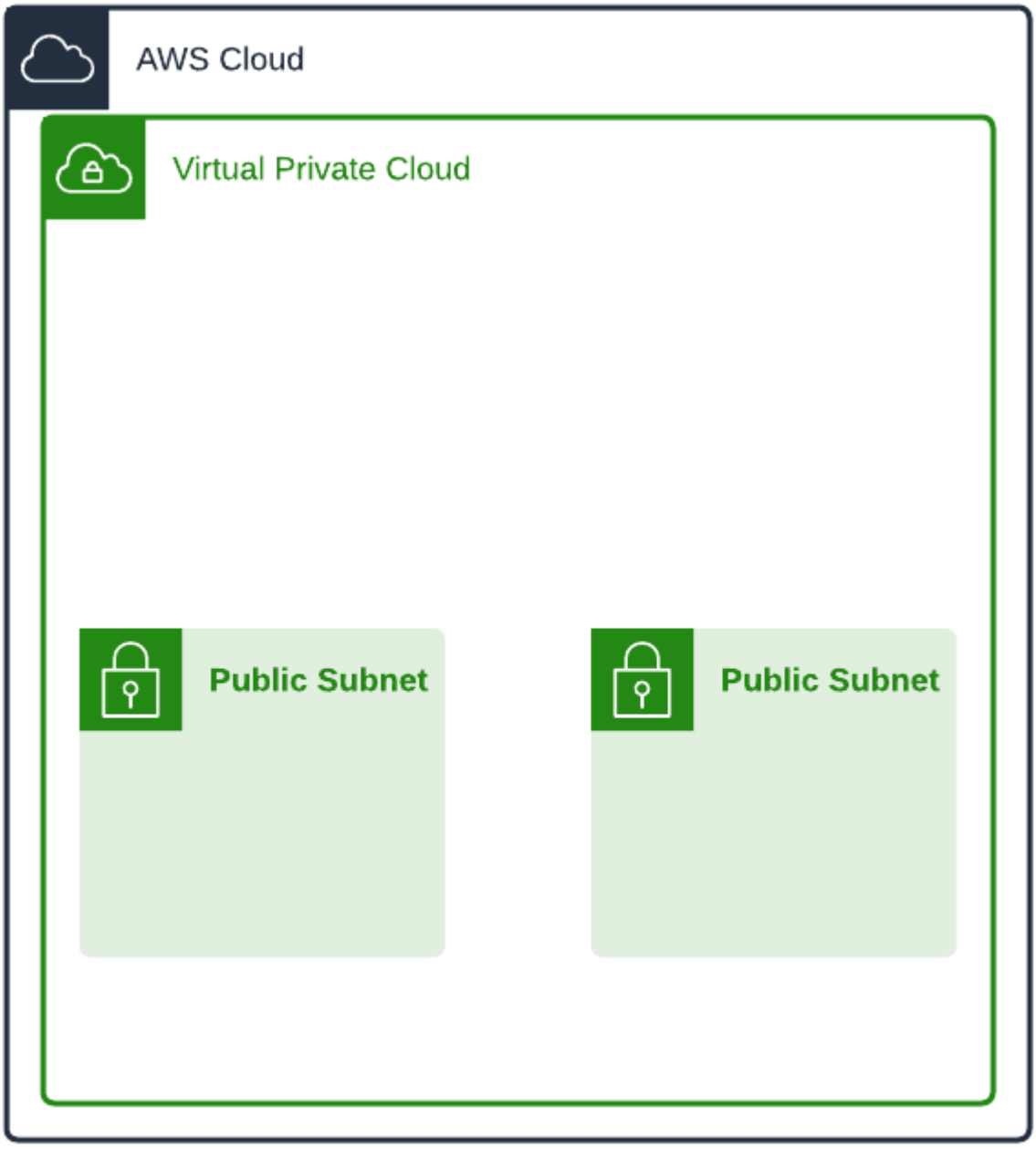
A computer on the cloud is called an **instance**. An instance in the cloud needs to be surrounded by several logical parts.

A VPC (Virtual Private Cloud) to exist inside of. The VPC acts as a big network foundation you put your resources into. You can have multiple VPCs on your AWS account, but we will be dealing with only one.



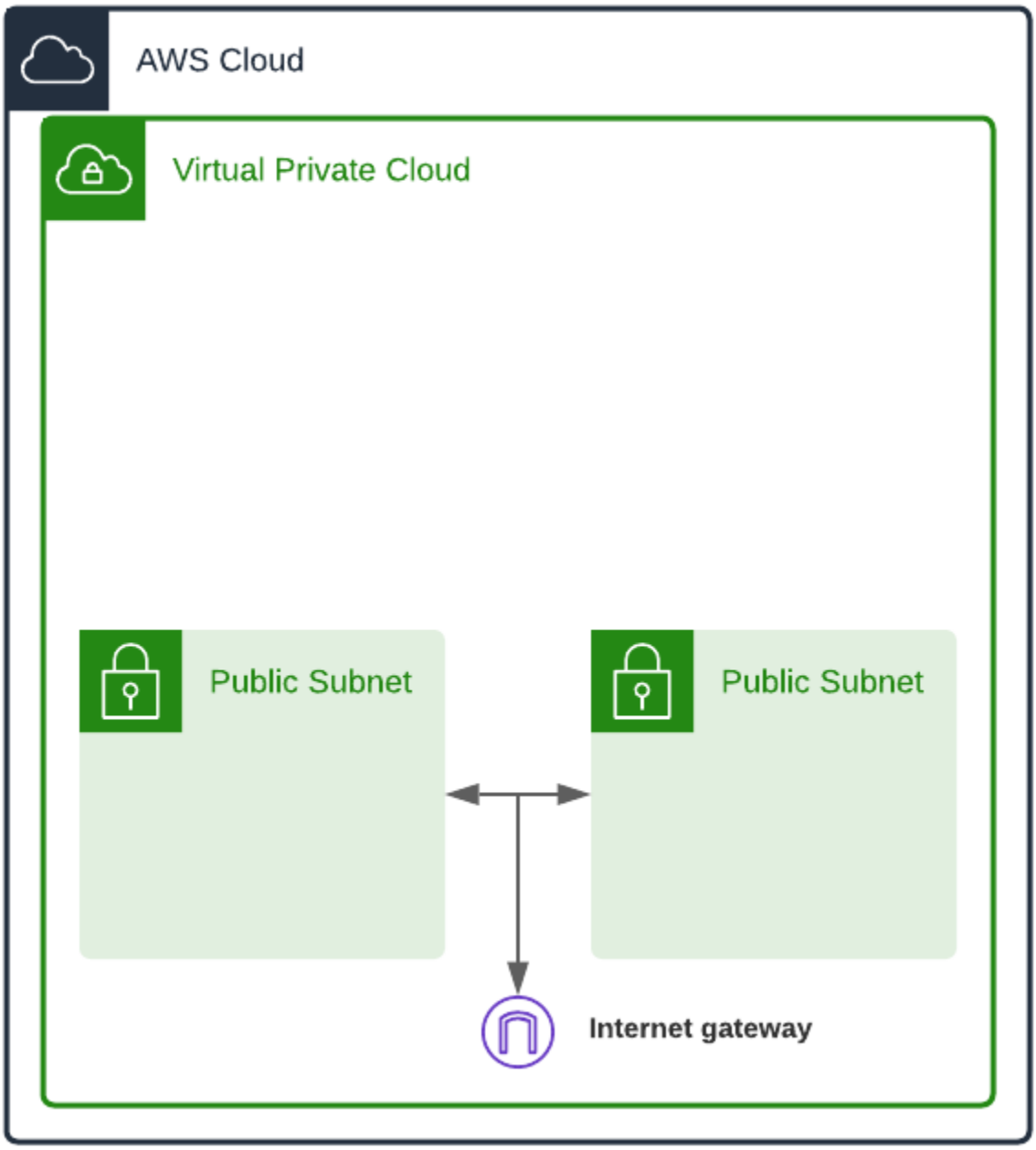
Computers also need **a subnet** to also exist inside of. Subnets logically divide up your VPC networks. You can have over 200 subnets in your VPC, but ours will have 2 public subnets, and 2 private subnets.

Public subnets are networks that are set up in a way that makes them capable of sending outbound connections and receiving inbound connections. We place resources such as instances in a public subnet only when we want them to be publicly accessible. Resources in a public subnet have public IP addresses, but still need to communicate via gateways.

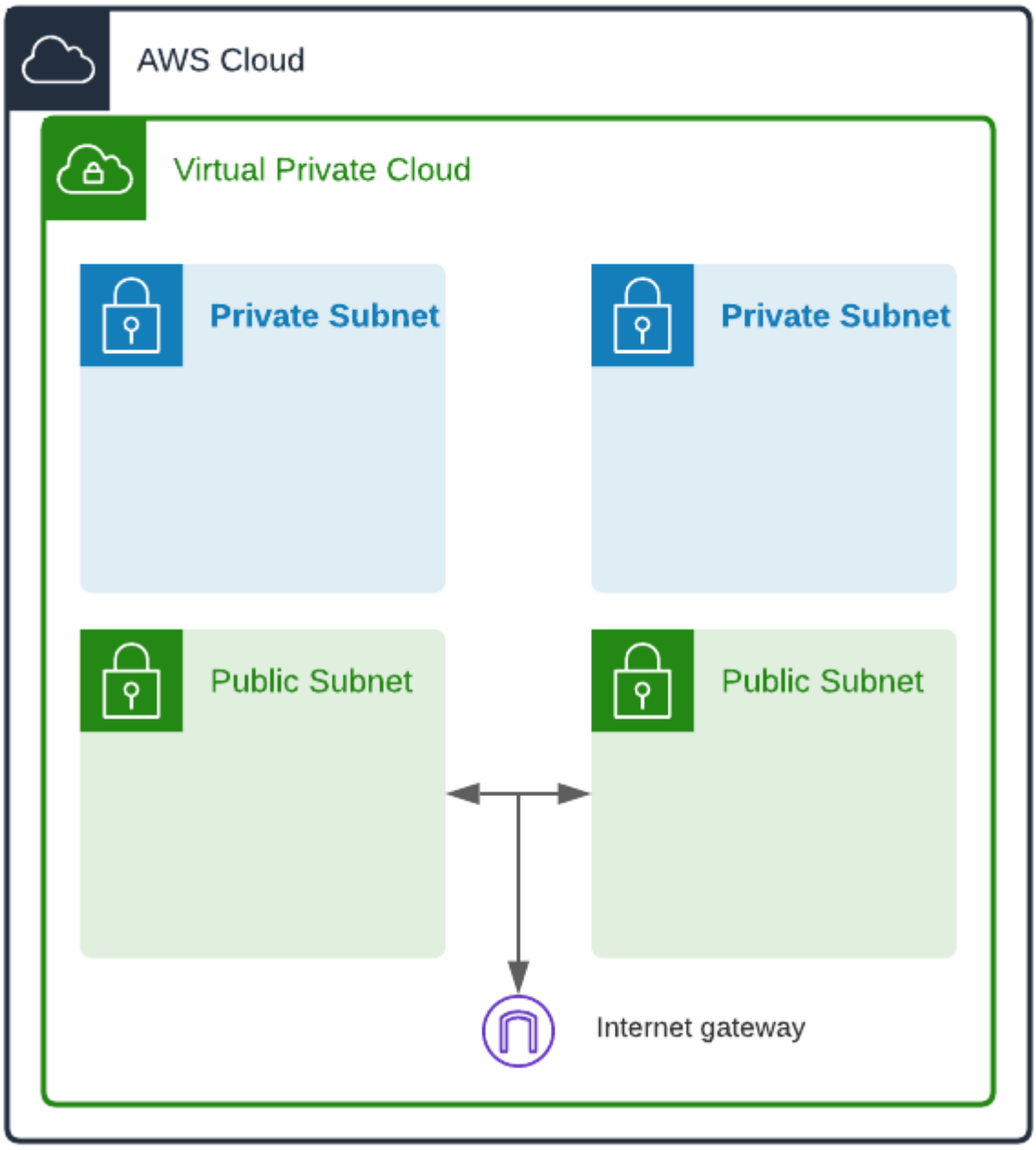


,

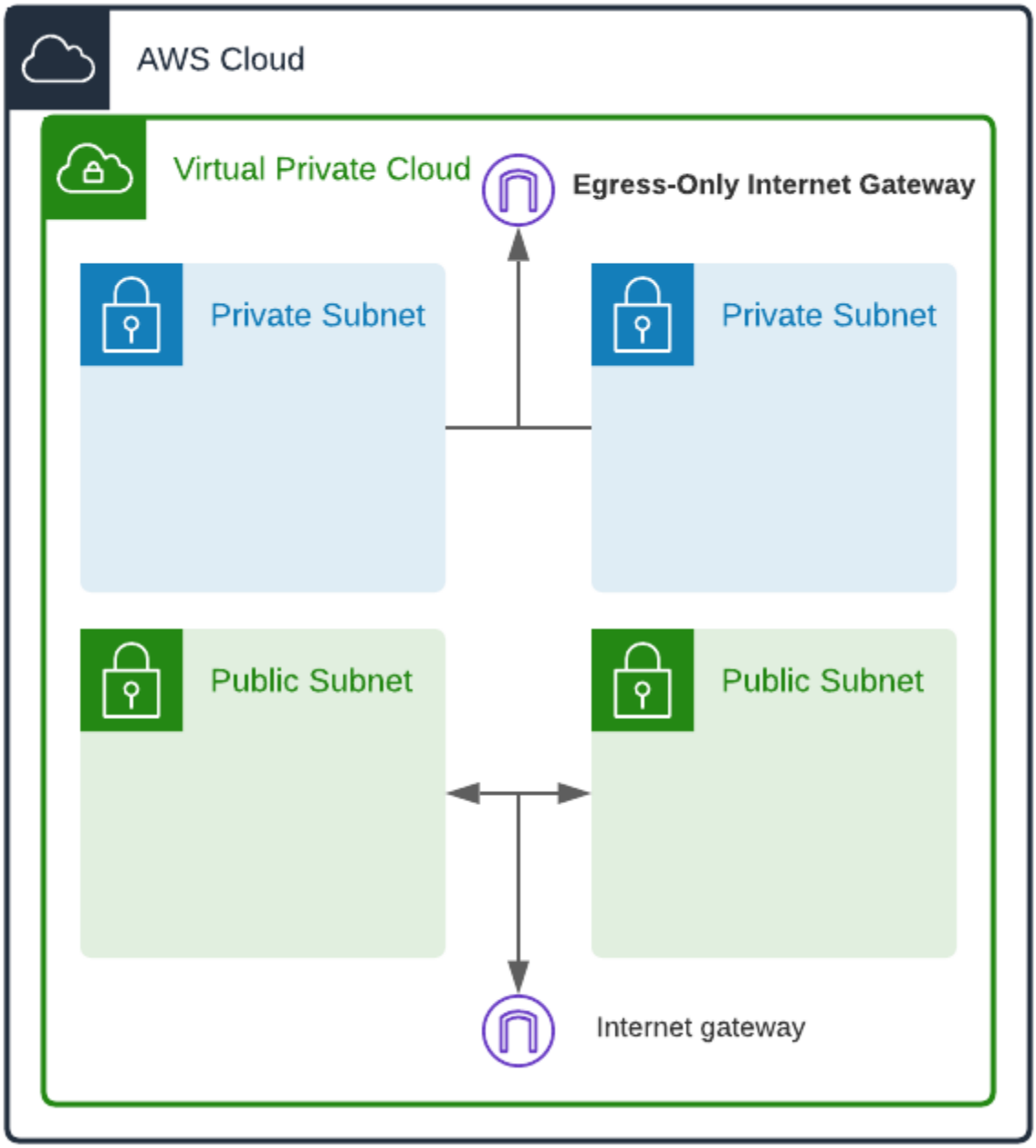
Internet Gateways (IGW) allow communication between your VPC and the internet. Any messages going out or coming into your VPC go through internet gateways. The IGW in our example will allow resources in our public subnets to communicate in and out of the VPC.



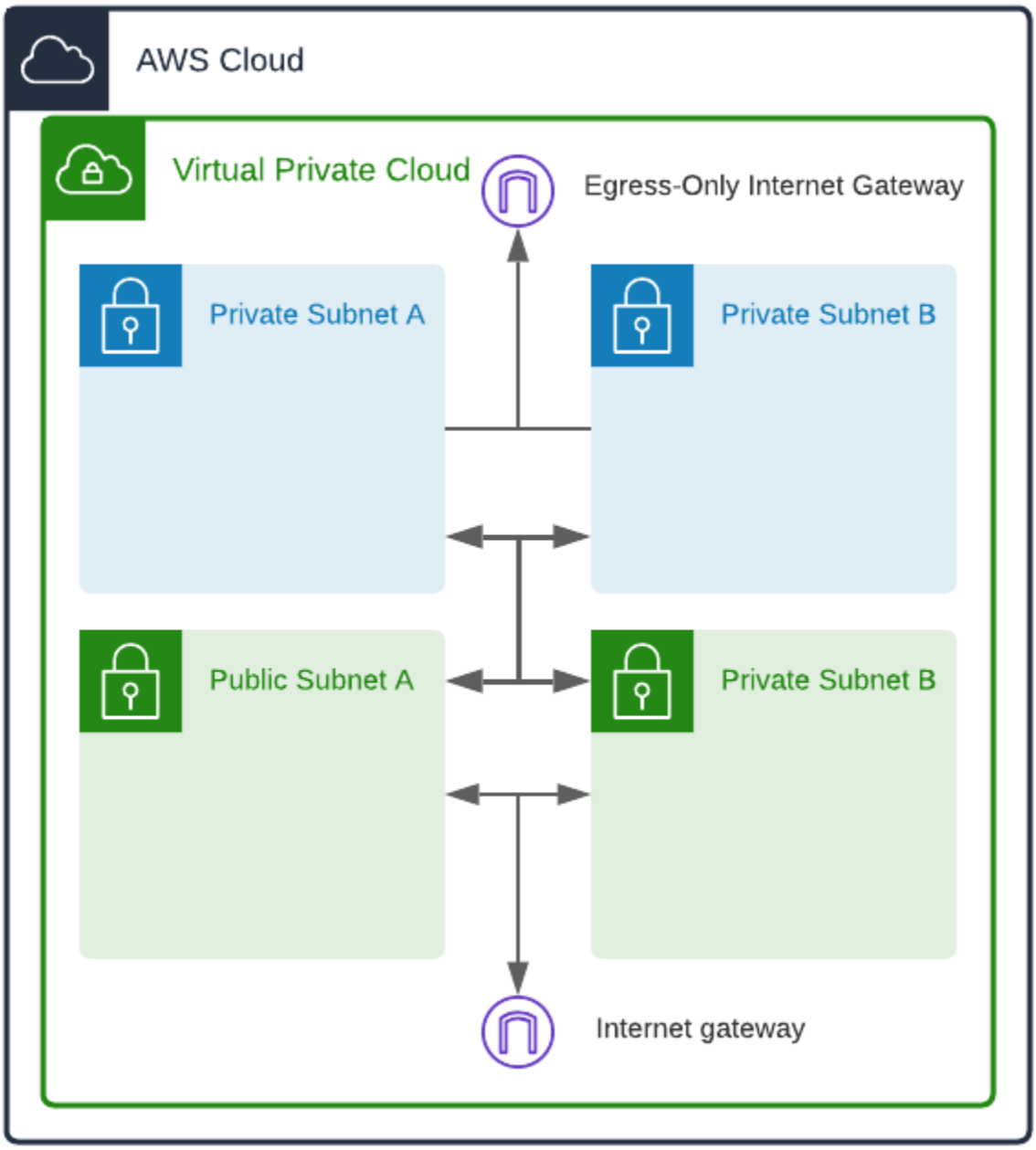
Private subnets are locked-down networks for more security-sensitive resources. We usually customize these subnets to not allow any outbound or inbound internet traffic at all. Private subnets usually only allow incoming traffic originating from inside the VPC. Resources in private subnets have private IP addresses, and as such can not directly communicate with entities outside their own subnets.



Egress-only internet gateways are IGWs that only allow outgoing connections. They're perfect for resources in private subnets that need to send data out, but restrict data coming in. Our Egress-only IGW will allow resources in our private subnets to communicate out.


AWS Cloud
Virtual Private Cloud
Egress-Only Internet Gateway
Private Subnet
Private Subnet
Public Subnet
Public Subnet
Internet gateway

Putting our Pieces Together

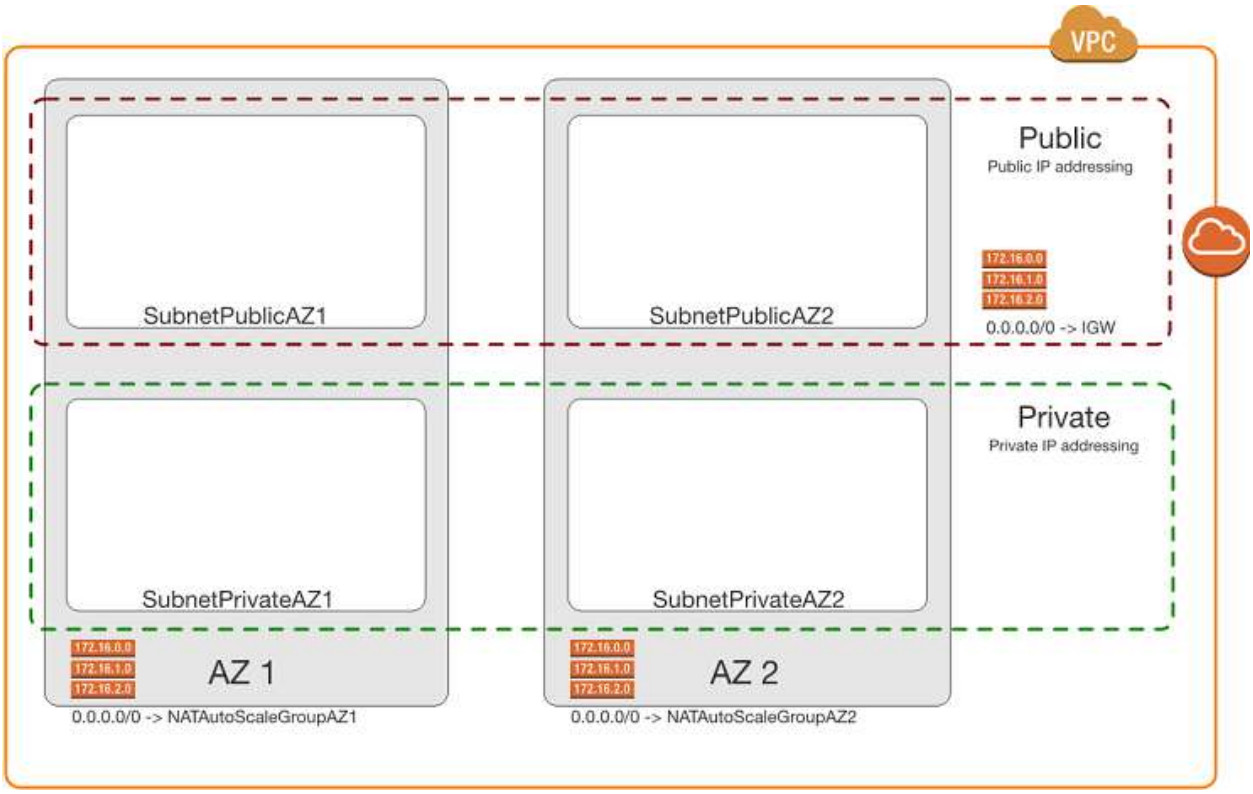


AWS Subnet Recommendations

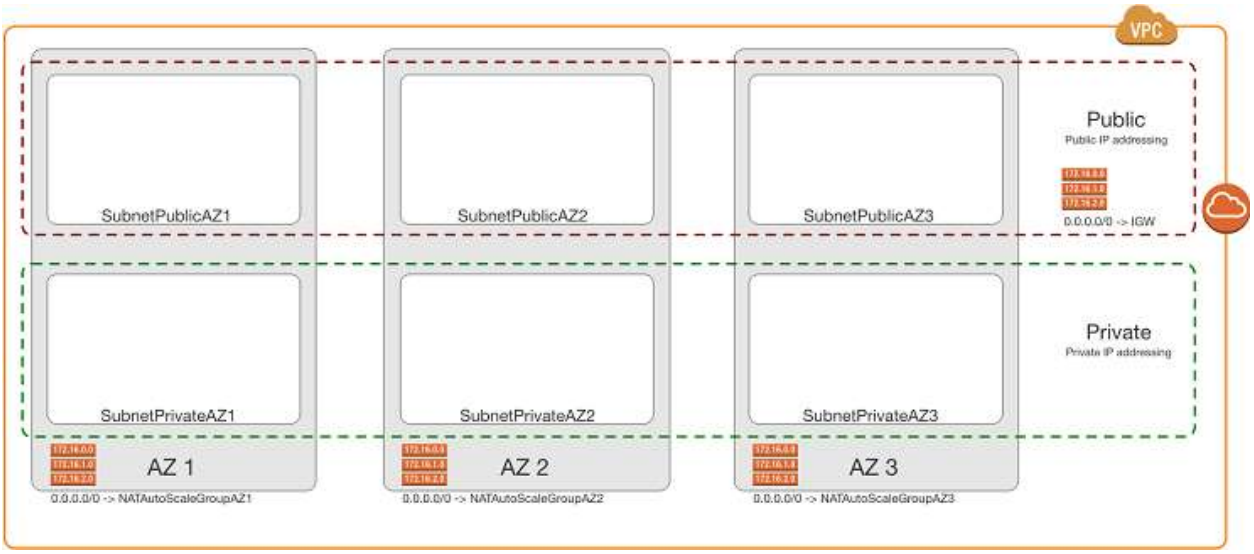
- EC2 instances in Public Subnets have public IP addresses associated with them and have a direct route to an AWS Internet Gateway (IGW), thus having the capability (if required) to access or be accessed by the Internet.

- EC2 instances in Private Subnets only have private IP addresses and cannot be accessed by the Internet. These EC2 instances have the capability to access the Internet via a NAT Gateway in the Public subnets

  Assuming a typical two AZ deployment, four subnets would be required (two for Public and two for Private).



  In situations where a third AZ is required (e.g. MongoDB servers in the Private subnets) then six subnets would be required (three for Public and three for Private).



It is important to note that within each tier, all the subnets will have the same network mask to simplify the operational processes (e.g. /22 for all Public subnets and /21 for all Private subnets).

Unlike traditional networking segmentation approaches that requires separate subnets (VLANs) for web, batch, application, and data tiers, AWS's use of Security Groups allows you to leverage just the Public and Private subnets, applying specific Security Groups to each tier

- Public Subnets
  - Bastion servers
  - NAT servers (if not using a NAT Gateway)
  - VPN servers (if not using a Virtual Private Gateway)
  - Web servers not behind any ELB
- Private Subnets
  - Web servers behind an ELB
  - Batch-tier instances
  - App-tier instance

- Data-tier instances

Differentiating between Azure Virtual Network (VNet) and AWS Virtual Private Cloud (VPC)