

Amazon Web Services – Virtual Private Cloud (VPC)



RICHARD FRISBY
JIMMY MCGIBNEY

Amazon Virtual Private Cloud (VPC)



- An Amazon VPC is an isolated portion of the AWS cloud. You use Amazon VPC to create a virtual network topology for your Amazon EC2 resources.
- 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 create a public-facing subnet for your webservers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access

Amazon Virtual Private Cloud (VPC)



Amazon
VPC

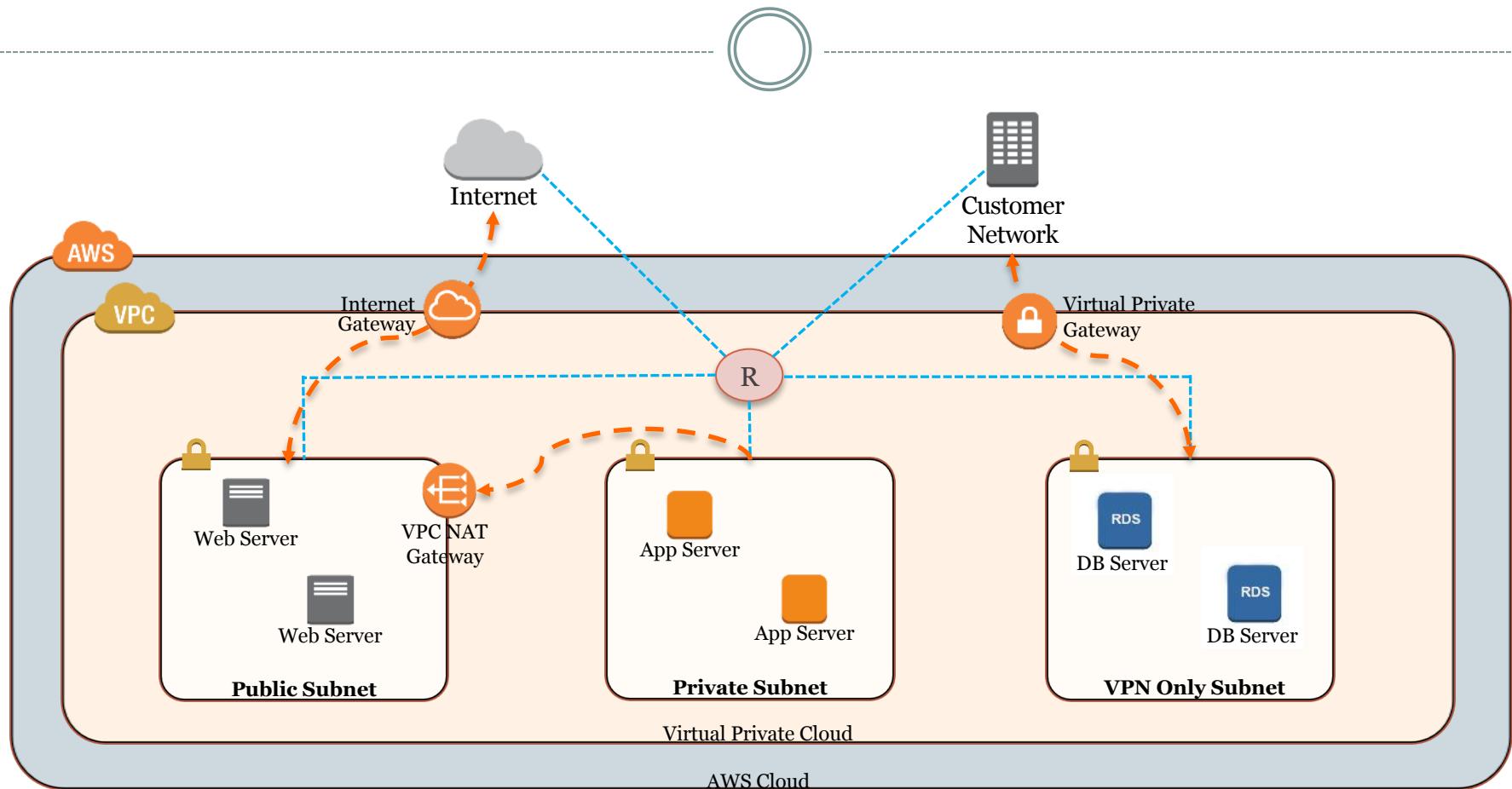
- Provision a **private, isolated virtual network** on the AWS cloud.
- Have complete control over your virtual networking environment.

VPCs and subnets



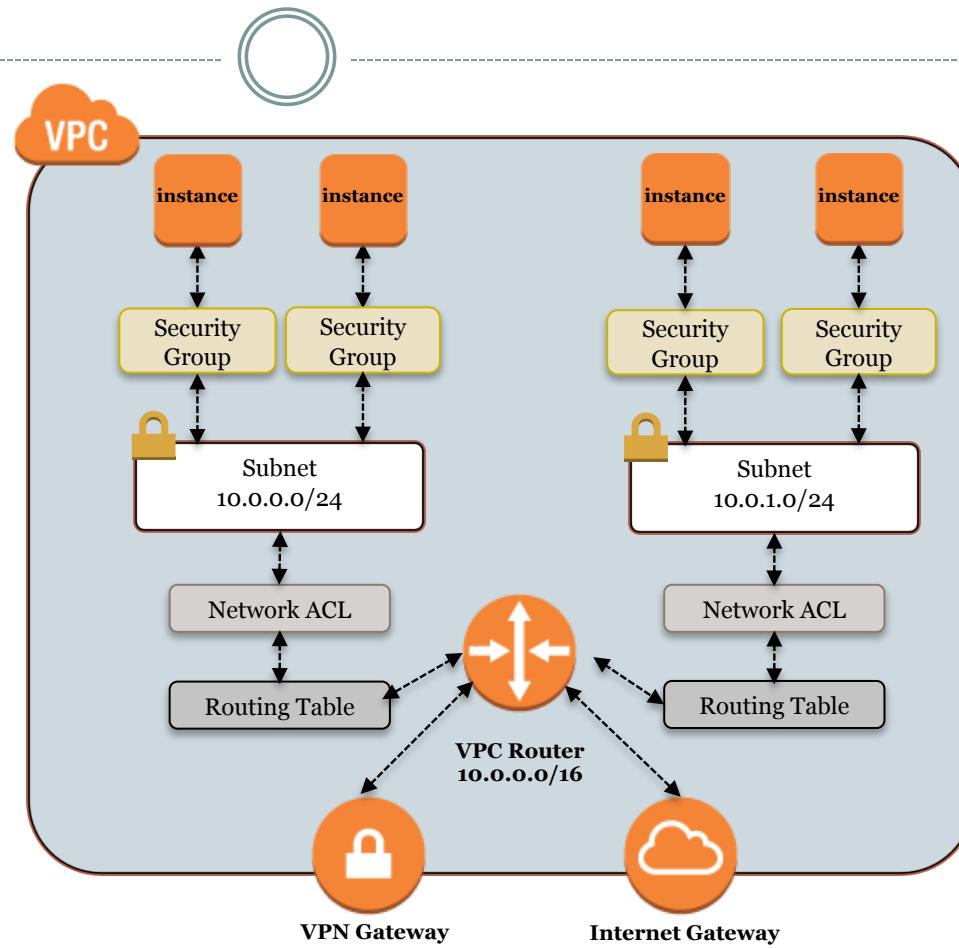
- A **subnet** defines a range of IP addresses in your VPC.
- You can launch AWS resources into a subnet that you select.
- A **private subnet** should be used for resources that won't be accessible over the Internet.
- A **public subnet** should be used for resources that will be accessed over the Internet.
- Each subnet must reside entirely within one Availability Zone and cannot span zones.

VPC example



Security in your VPC

- Security groups
- Network access control lists (ACLs)
- Key Pairs



VPN connections



VPN Connectivity option	Description
AWS Hardware VPN	You can create an IPsec hardware VPN connection between your VPC and your remote network.
AWS Direct Connect	AWS Direct Connect provides a dedicated private connection from a remote network to your VPC.
AWS VPN CloudHub	You can create multiple AWS hardware VPN connections via your VPC to enable communications between various remote networks.
Software VPN	You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a software VPN appliance .

Using One VPC



There are **limited** use cases where one VPC could be appropriate:

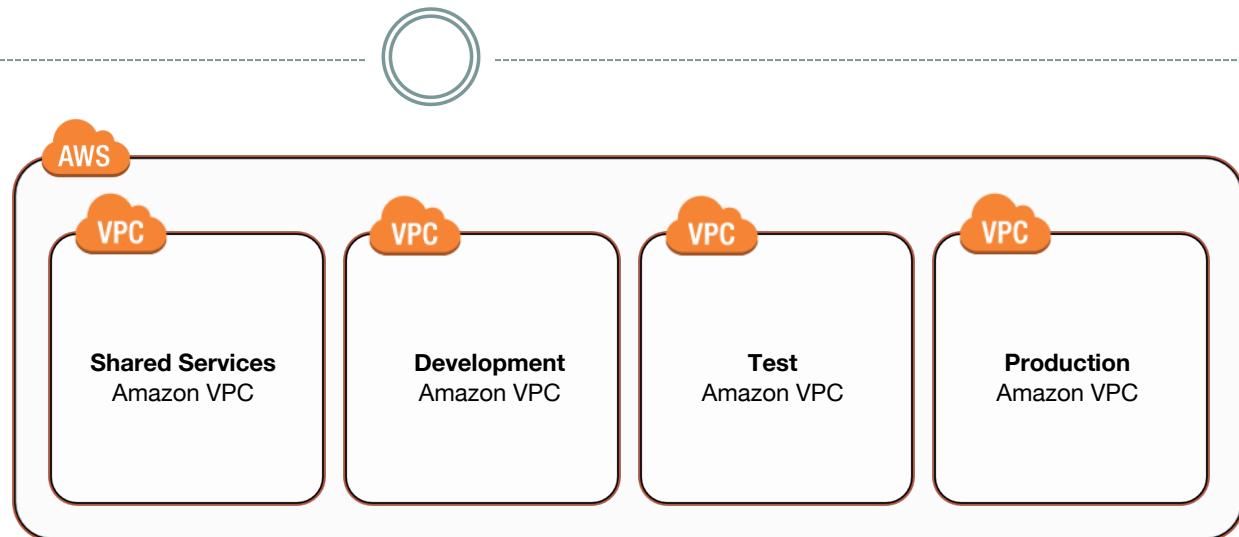
- High-performance computing
- Identity management
- Small, single applications managed by one person or very small team

For **most** use cases, there are two primary patterns for organizing your infrastructure:

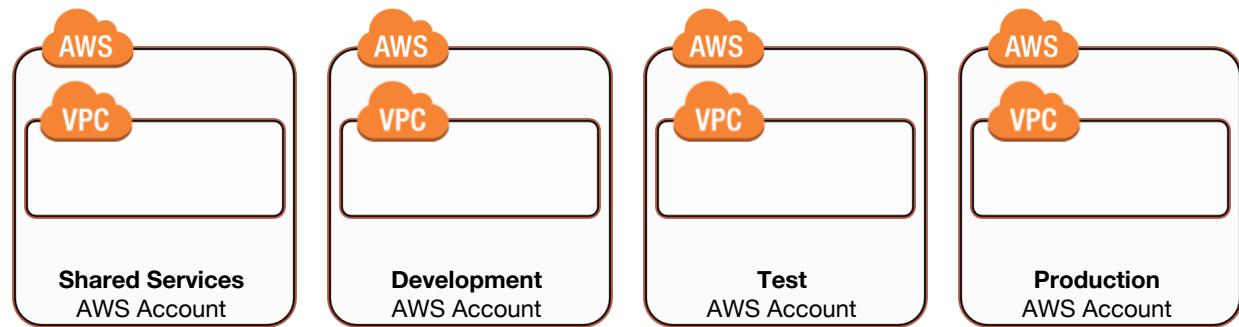
Multi-VPC and **Multi-Account**

AWS Infrastructure Patterns

VPC pattern



Account pattern



Choosing A Pattern



How do you know which pattern to use?

- The primary factors for determining this are the **complexity** of your organization and your **workload isolation** requirements:
 - Single IT team? **Multi-VPC**
 - Large organization with many IT teams? **Multi-account**
 - High workload isolation required? **Multi-account**

Other Important Considerations



The majority of AWS services **do not actually sit within a VPC.**

- For these services, a VPC **cannot provide any isolation outside of connectivity.**
- Network traffic between AWS Regions traverse the AWS global network backbone by default.
- Amazon S3 and DynamoDB offer **VPC endpoints** to connect without traversing the public Internet.

VPCs And IP Addresses

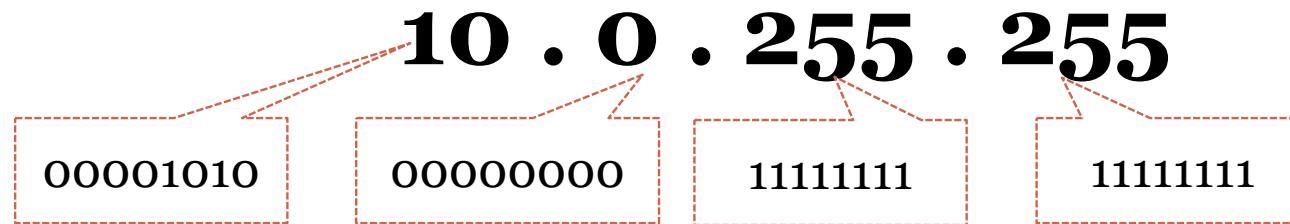
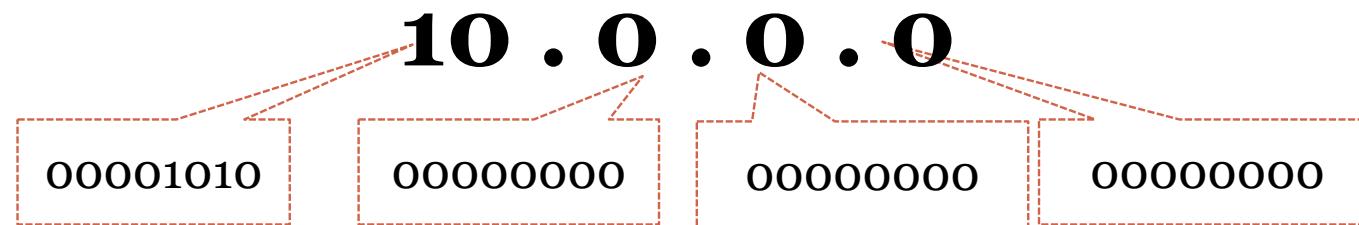


- When you create your VPC, you specify its set of IP addresses with CIDR notation
- Classless Inter-Domain Routing (CIDR) notation is a simplified way to show a specific range of IP addresses
- Example: $10.0.0.0/16$ = all IPs from $10.0.0.0$ to $10.0.255.255$
- **How does that work?** What does the **16** define?

IPs and CIDR



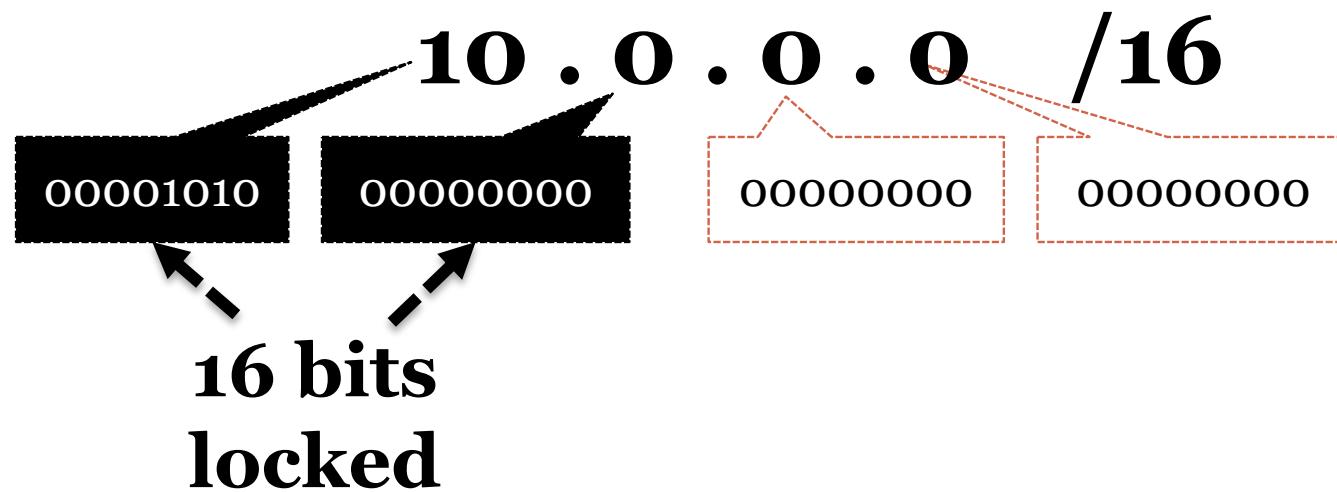
Every set of 4 digits in an IP address represents a set of 8 binary values (8 bits).



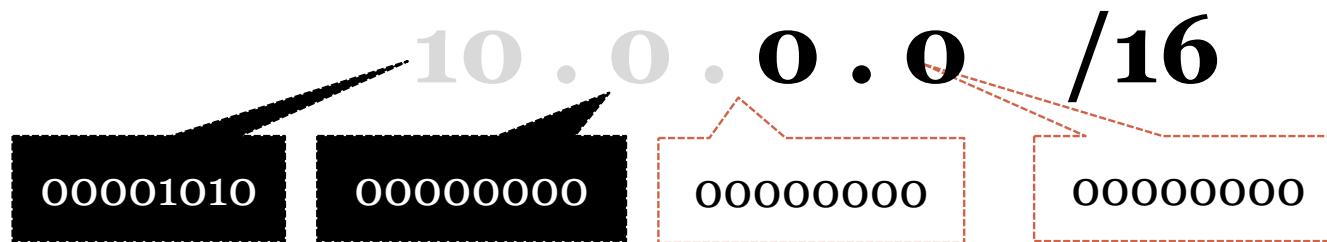
IPs and CIDR



The 16 in the CIDR notation example represents how many of those bits are "locked down" and cannot change.



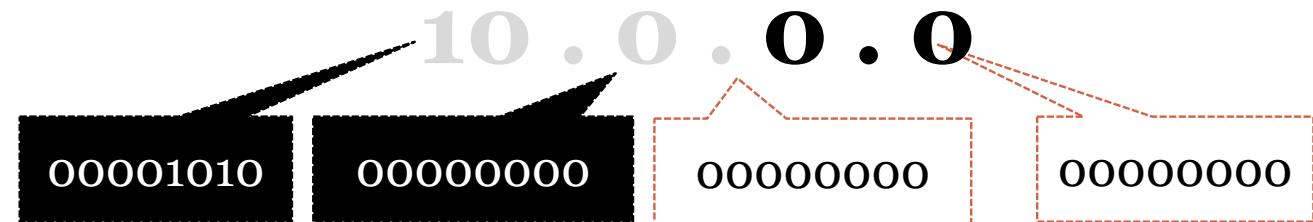
IPs and CIDR



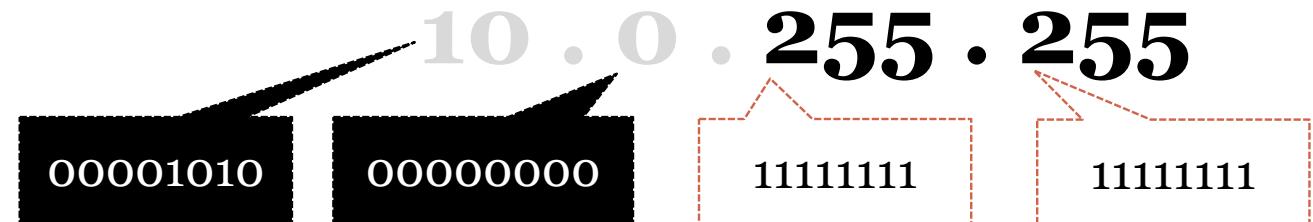
The unlocked bits can change between 1 and 0, allowing the full range of possible values.

CIDR Example: 10.0.0.0/16

**Lowest
possible
IP**



**Highest
possible IP**



VPCs and IP Addresses



- AWS VPCs can use CIDR ranges between /16 and /28.
- For every one step a CIDR range increases, the total number of IPs is cut in half:

CIDR / Total IPs						
/16	/17	/18	/19	/20	/21	/22
65,536	32,768	16,384	8,192	4,096	2,048	1,024
/23	/24	/25	/26	/27	/28	
512	256	128	64	32	16	

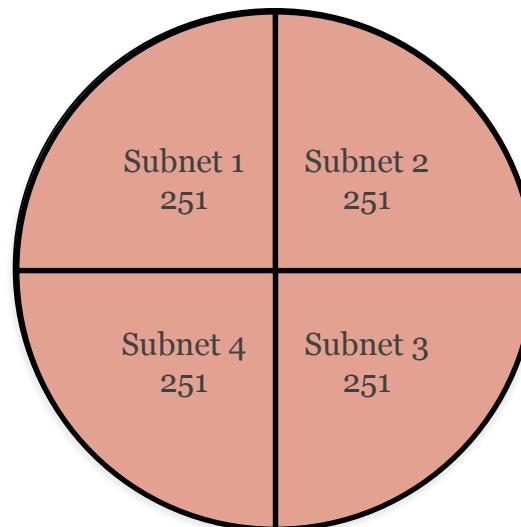
What Are Subnets?



Subnets are **segments or partitions** of a network, divided by **CIDR range**.

Example:

A VPC with **CIDR /22**
includes 1,024 total IPs



Note: In every subnet,
the first four and last
one IP addresses are
reserved for AWS use.

How to Use Subnets



Recommendation: Use subnets to define Internet accessibility.

Public subnets

Include a routing table entry to an **Internet gateway** to support inbound/outbound access to the public Internet.

Private subnets

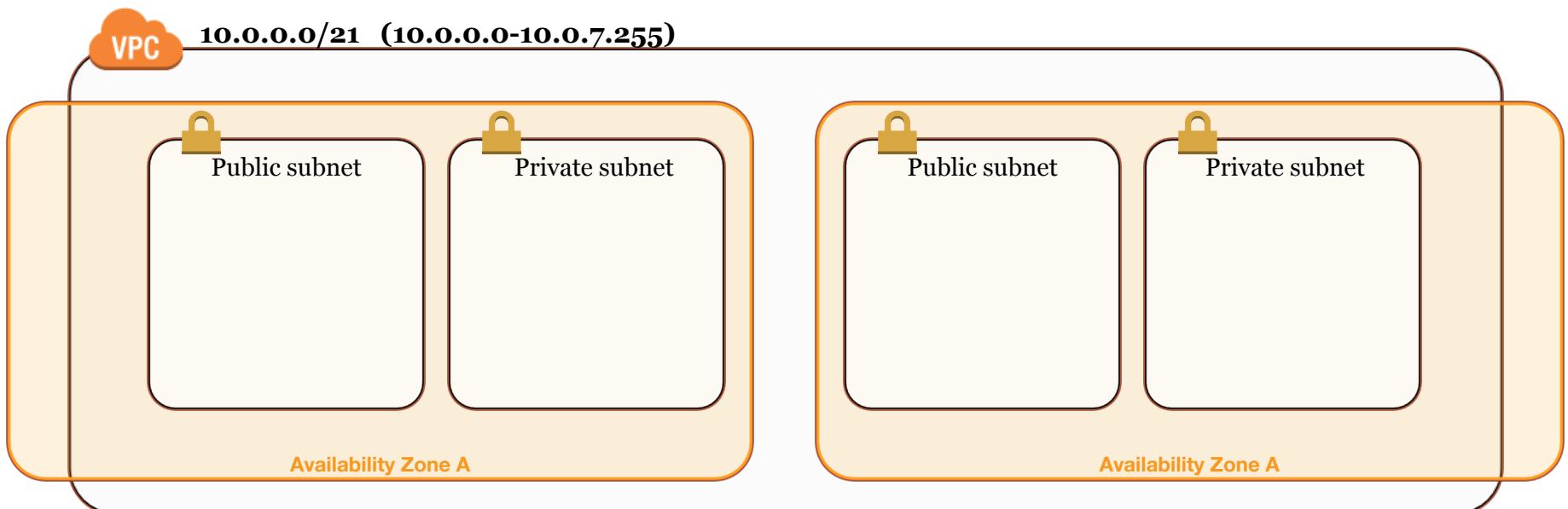
Do not have a routing table entry to an Internet gateway and are **not directly accessible** from the public Internet.

Typically use a "jump box" (NAT/proxy/bastion host) to support restricted, **outbound-only** public Internet access.

Subnets



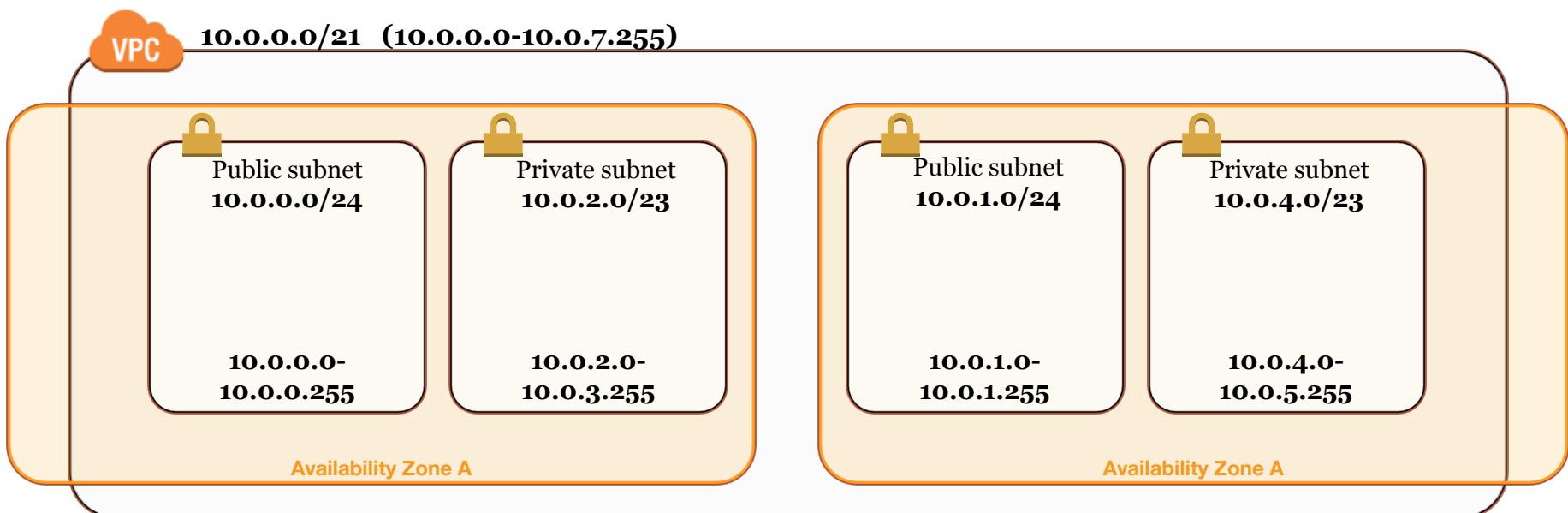
Recommendation: Start with **one public** and **one private** subnet per Availability Zone.



Subnets



Recommendation: Start with **one public** and **one private** subnet per Availability Zone.



Subnet Sizes



Recommendation: Consider larger subnets over smaller ones (/24 and larger).

Simplifies workload placement:

Choosing where to place a workload among 10 small subnets is more complicated than with one large subnet.

Less likely to waste or run out of IPs:

If your subnet runs out of available IPs, you can't add more to that subnet.

Example: If you have 251 IPs in a subnet that's using only 25 of them, you can't share the unused 226 IPs with another subnet that's running out.

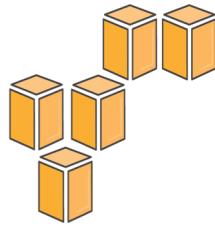
Subnet Types



Which subnet type (public or private) should you use for these resources ?

	Public	Private
Datastore instances	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Batch processing instances	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Back-end instances	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Web application instances	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

How do you control your VPC traffic?



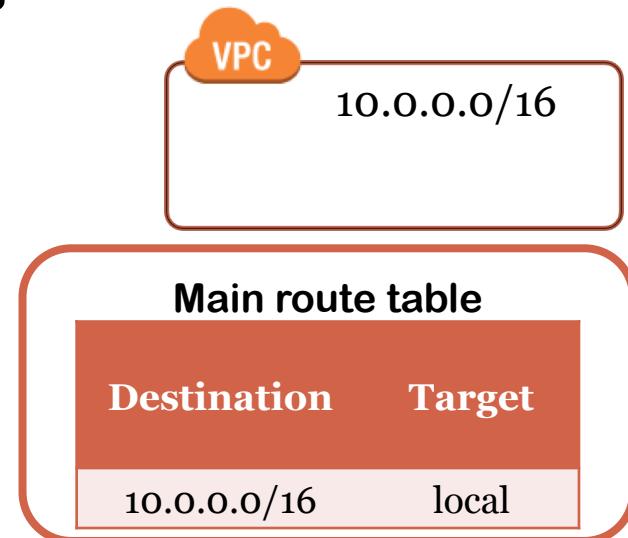
- Route tables
- Security groups
- Network ACLs
- Internet gateways

Route Tables



Directing Traffic Between VPC Resources

- Determine where network traffic is routed
- Main and custom route tables
- VPC route table: *Local route*
- Only one route table per subnet



Best practice:

Use custom route tables for each subnet to enable granular routing for destinations.

Security Groups



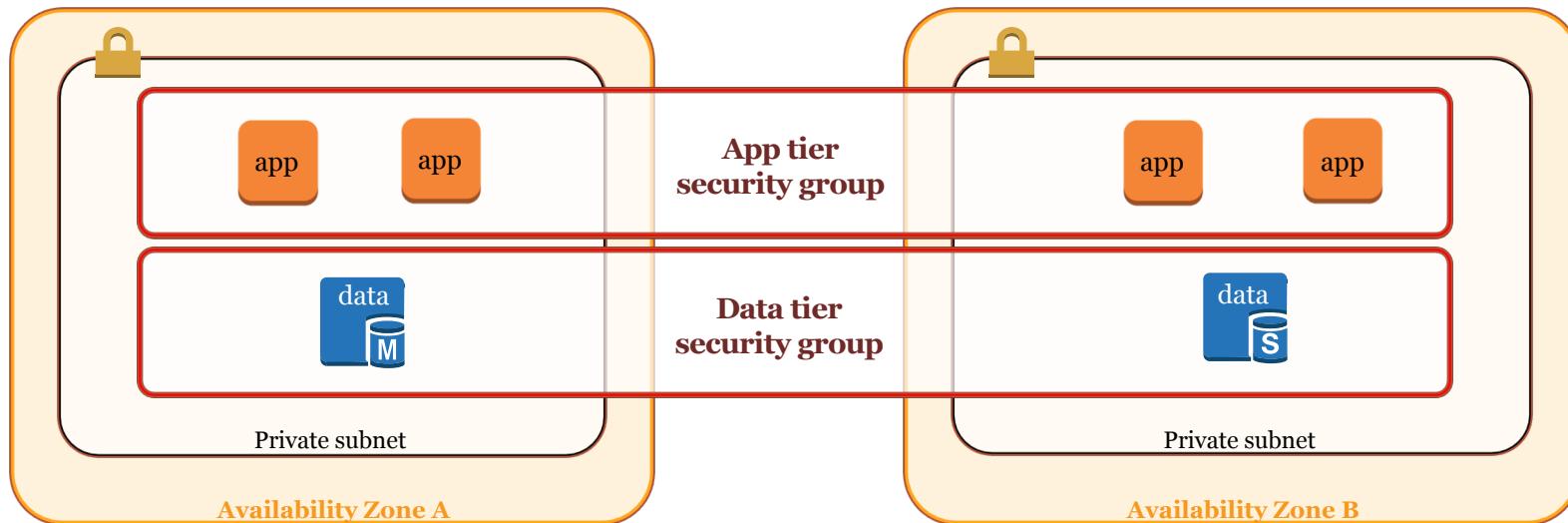
Securing VPC Traffic With Security Groups

- Are **virtual firewalls** that control inbound and outbound traffic for one or more instances.
- Deny all incoming traffic by default and use *allow* rules that can filter based on TCP, UDP, and ICMP protocols.
- Are ***stateful***, which means that if your inbound request is allowed, the outbound response does not have to be inspected/tracked, and vice versa.
- Can define a **source/target** as either a **CIDR block** or **another security group** to handle situations like auto scaling.

Security Groups



Use security groups to control traffic **into, out of, and between** resources.



How Security Groups Are Configured

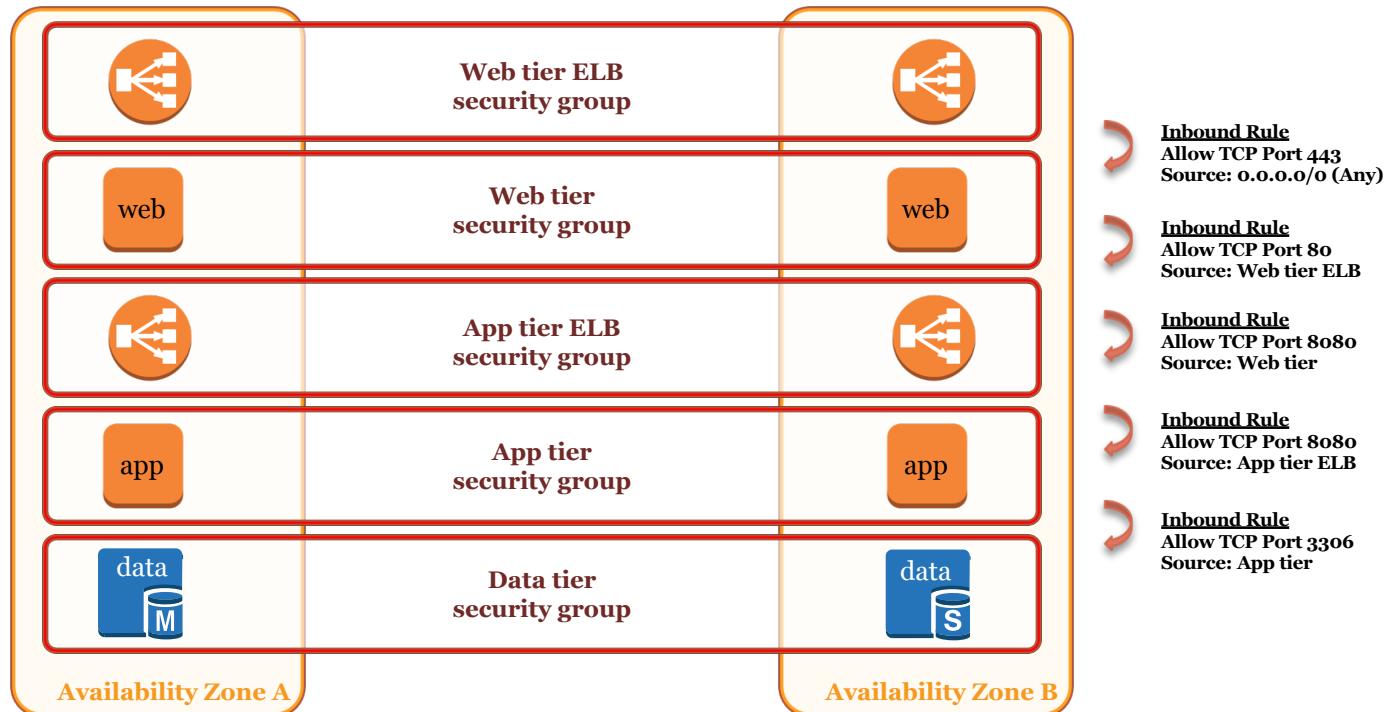


- By default, all newly created security groups **allow all outbound traffic** to all destinations.
Modifying the default outbound rule on security groups increases complexity and is not recommended unless required for compliance.
- Most organizations create security groups with **inbound rules** for **each functional tier** (web/app/data/etc.) within an application.

Security Group Chaining Diagram



Security group rules per application tier



Network ACLs

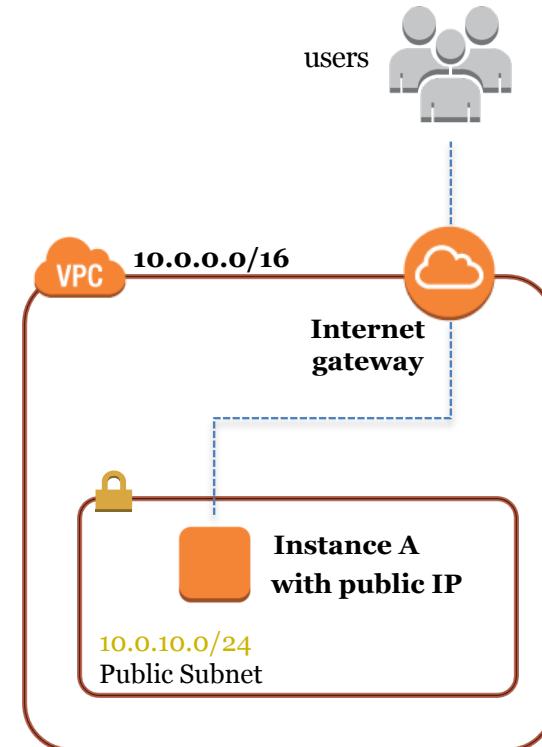


- Are **optional virtual firewalls** that control traffic in and out of a subnet.
- **Allow all** incoming/outgoing traffic by default and use **stateless** rules to allow or deny traffic.
"Stateless rules" inspect all inbound and outbound traffic and do not keep track of connections.
- Enforce rules only at the **boundary of the subnet**, not at the instance-level, like security groups.

Internet gateways

Directing Traffic To Your VPC

- Allow communication between instances in your VPC and the Internet.
- Are horizontally scaled, redundant, and highly available by default.
- Provide a target in your VPC route tables for Internet-routable traffic.



Directing Traffic To Your VPC



To enable access to or from the Internet for instances in a VPC subnet, you must:

- Attach an Internet gateway to your VPC
- Ensure that your subnet's route table points to the Internet gateway
- Ensure that instances in your subnet have public IP addresses or Elastic IP addresses
- Ensure that your NACLs and security groups allow the relevant traffic to flow to and from your instance

What About Outbound Traffic From Private Instances?



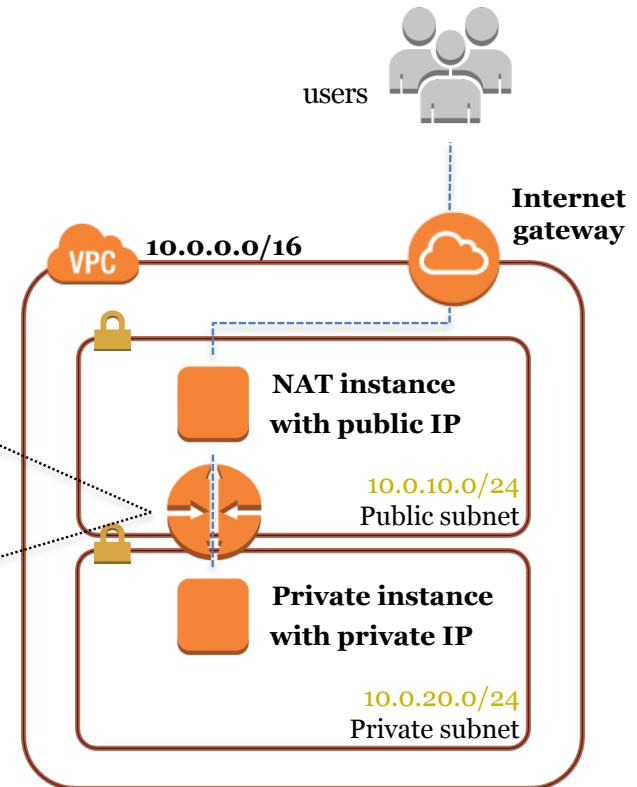
Network Address Translation services:

- Enable instances in the private subnet to initiate outbound traffic to the Internet or other AWS services.
- Prevent private instances from receiving inbound traffic from the Internet.

Two primary options:

- Amazon EC2 instance set up as a NAT in a public subnet
- VPC NAT Gateway

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	NAT



What About Outbound Traffic From Private Instances ?

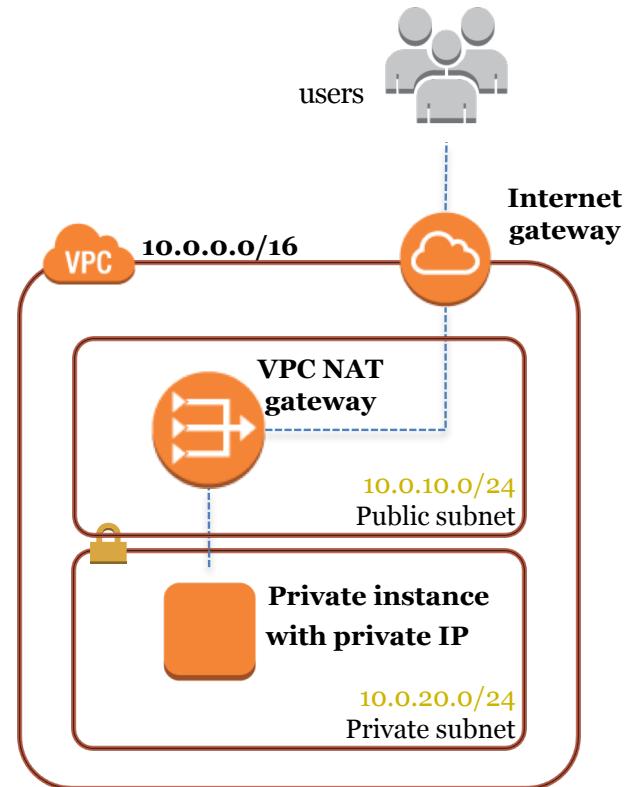


Network Address Translation services:

- Enable instances in the private subnet to initiate outbound traffic to the Internet or other AWS services.
- Prevent private instances from receiving inbound traffic from the Internet.

Two primary options:

- Amazon EC2 instance set up as a NAT in a public subnet
- VPC NAT Gateway

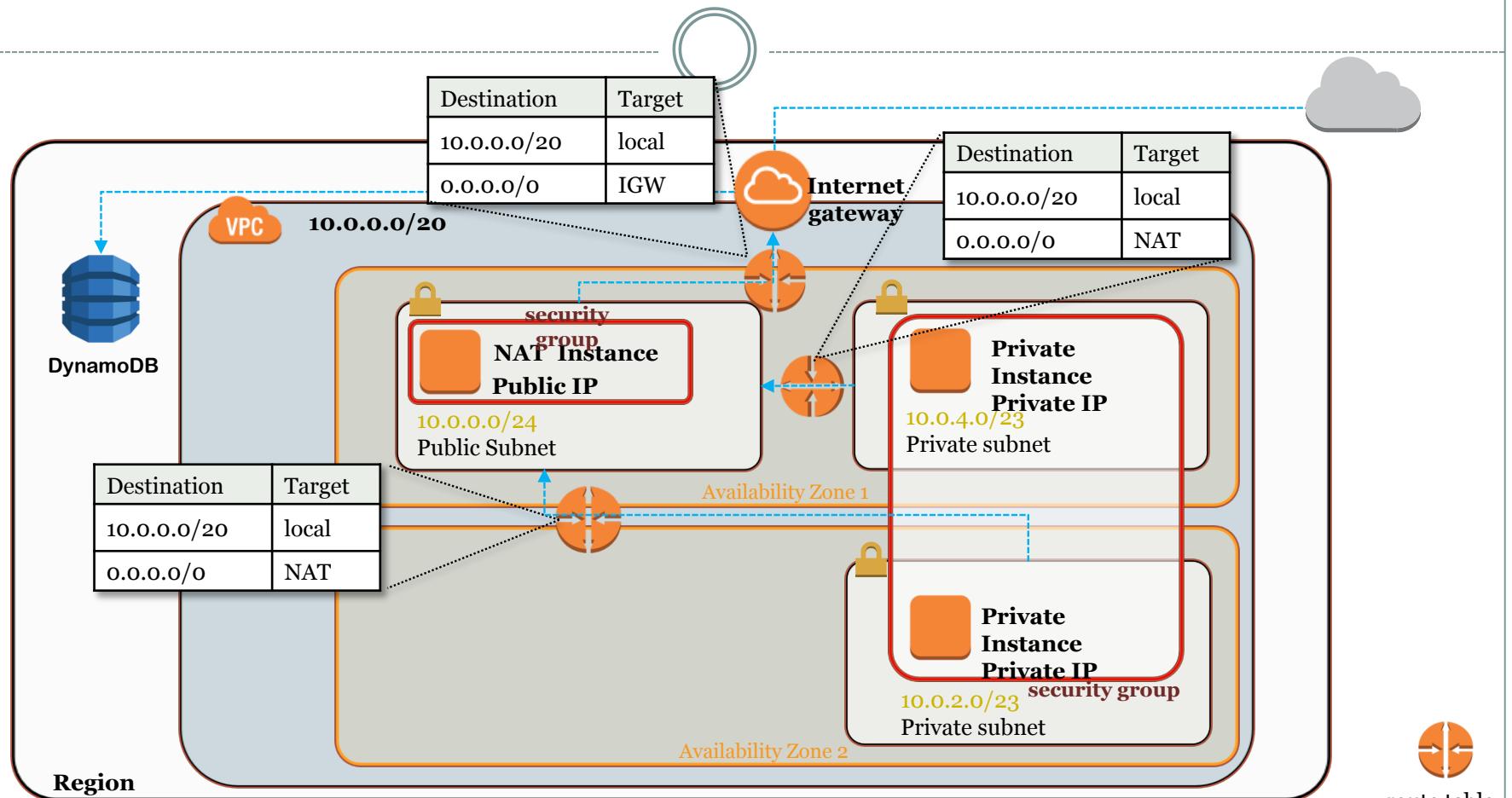


VPC NAT Gateways vs. NAT Instances On Amazon EC2



	VPC NAT gateway	NAT instance
Availability	Highly available by default	Use script to manage failover
Bandwidth	Bursts to 10 Gbps	Based on bandwidth of instance type
Maintenance	Managed by AWS	Managed by you
Security	NACLs	Security groups and NACLs
Port forwarding	Not supported	Supported

Subnets, Gateways, and Routes



Logging VPC Traffic



Amazon VPC Flow Logs

- Captures traffic flow details in your VPC
Accepted and rejected traffic
- Can be enabled for VPCs, subnets, and ENIs
- Logs published to CloudWatch Logs

Use cases:

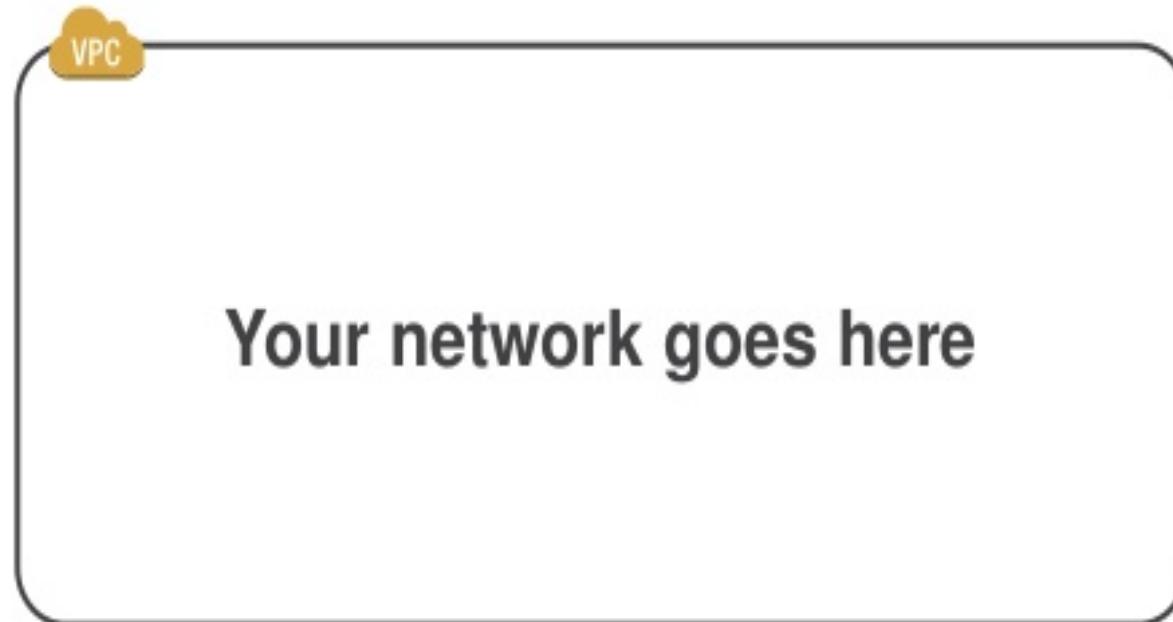
- Troubleshoot connectivity issues.
- Test network access rules.
- Monitor traffic.
- Detect and investigate security incidents.

Amazon Virtual Private Cloud (VPC)



VPC Overview

- Bring your own network

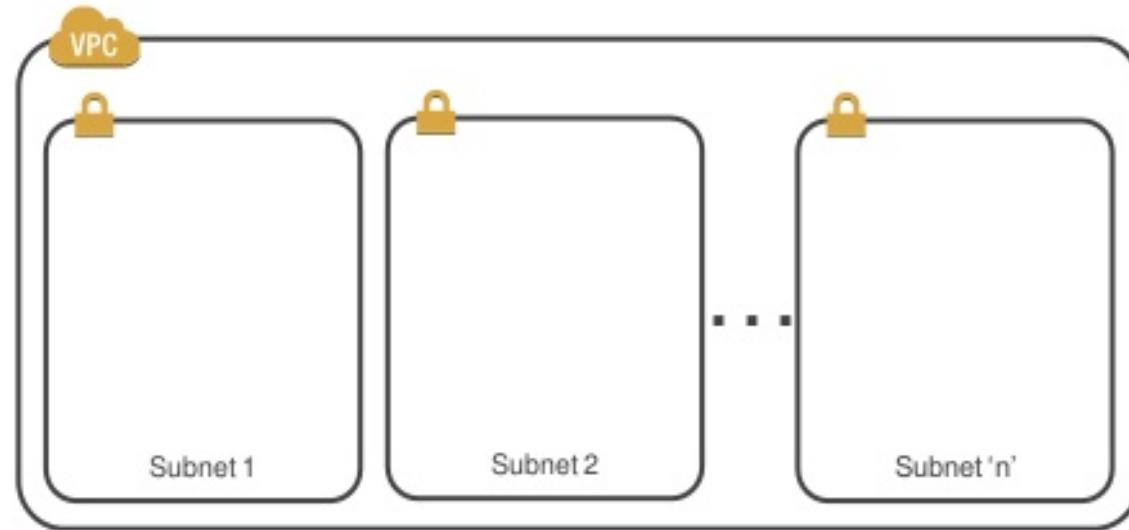


Amazon Virtual Private Cloud (VPC)



VPC Overview

- Bring your own network
- Create your own subnets

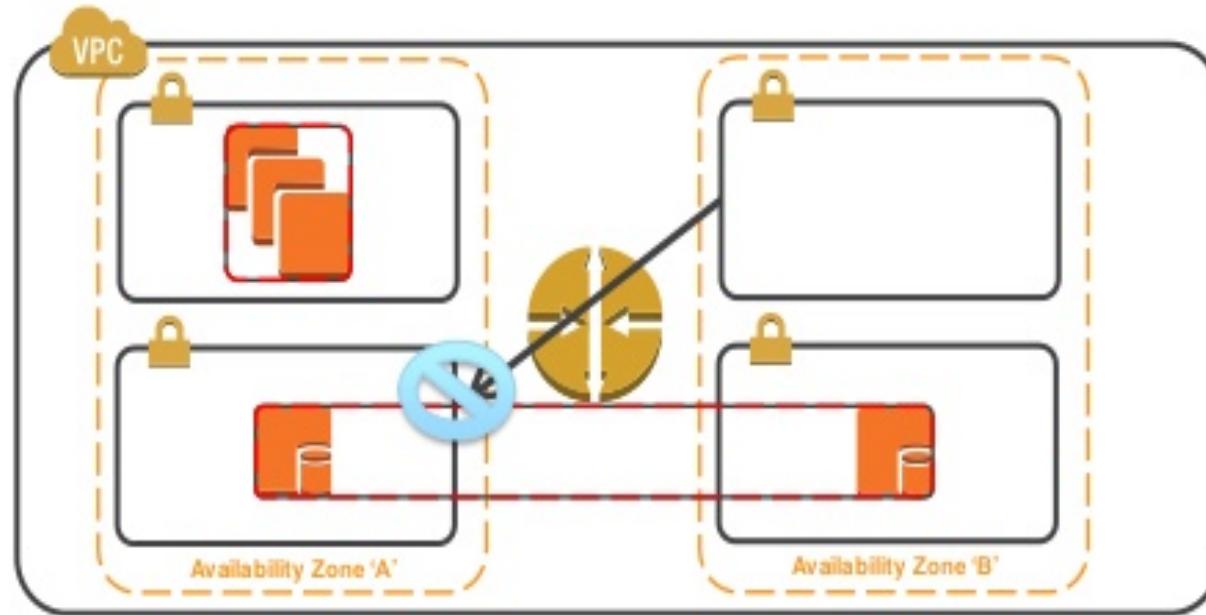


Amazon Virtual Private Cloud (VPC)



VPC Overview

- Control instance placement and traffic
 - Security Groups & NACLs

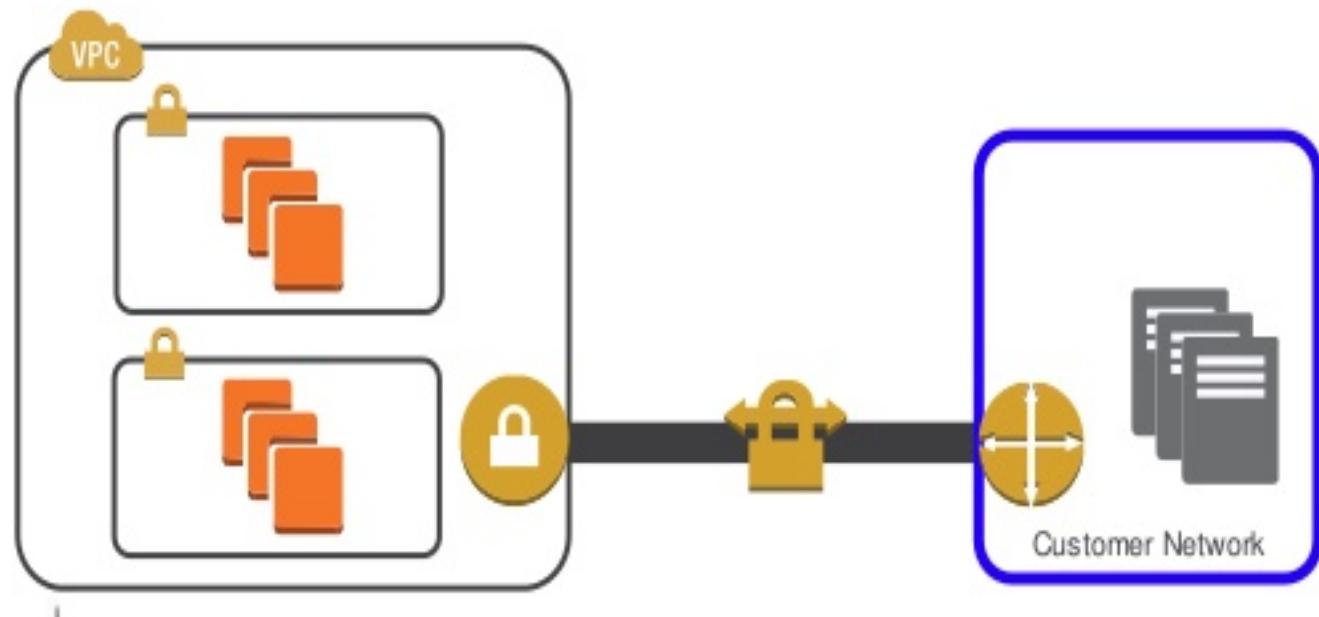


Amazon Virtual Private Cloud (VPC)



VPC Overview

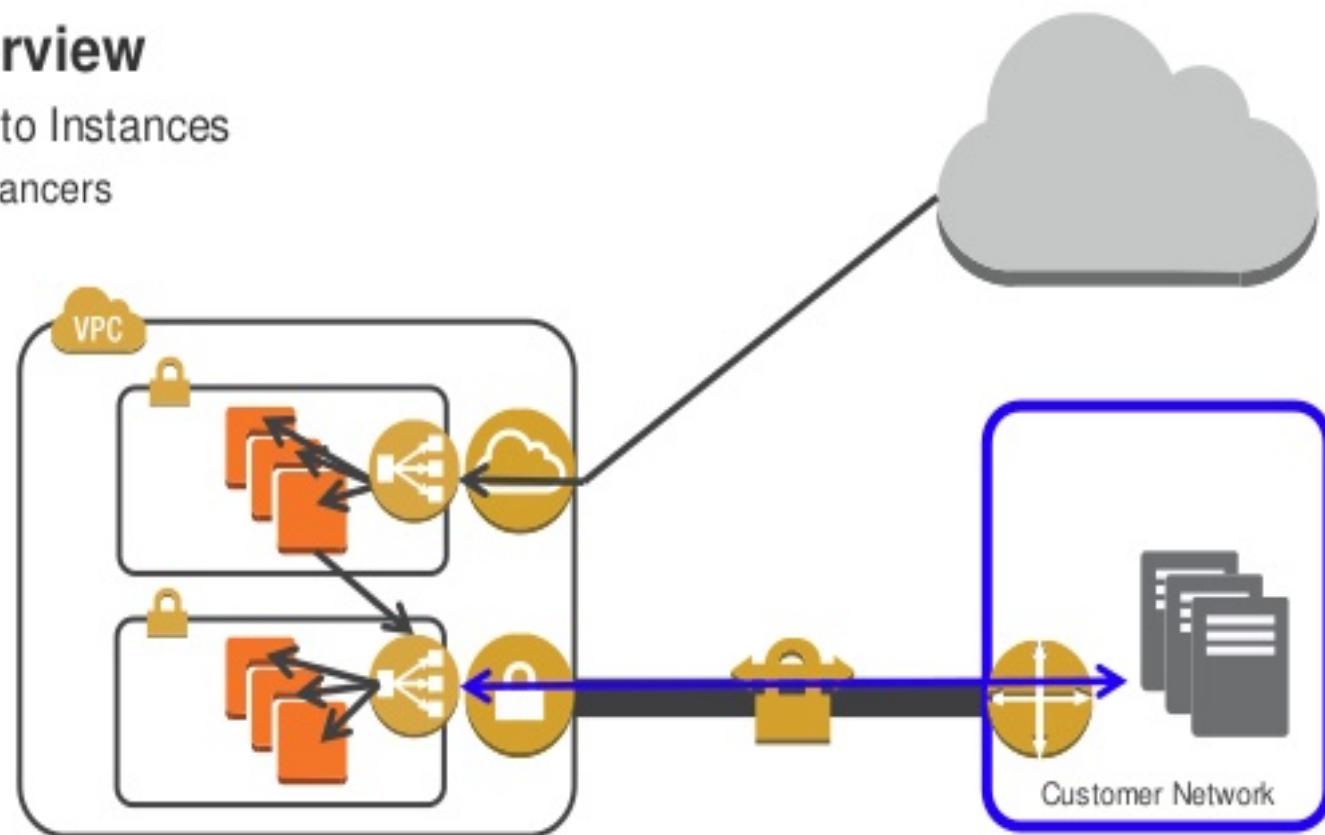
- Virtual Private Gateway
 - IPSEC VPN



Amazon Virtual Private Cloud (VPC)

VPC Overview

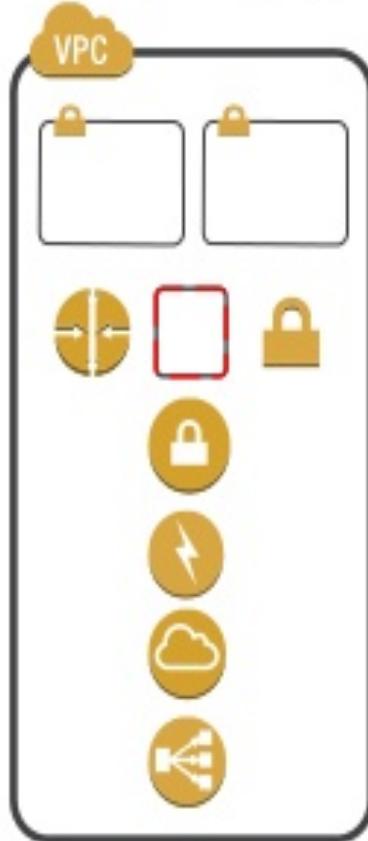
- Connecting to Instances
 - Load Balancers



Amazon Virtual Private Cloud (VPC)

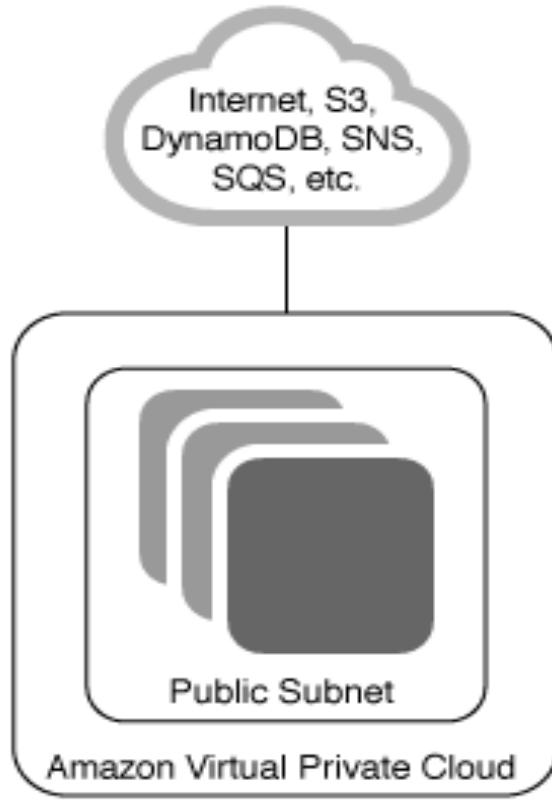


VPC Building Blocks Summary



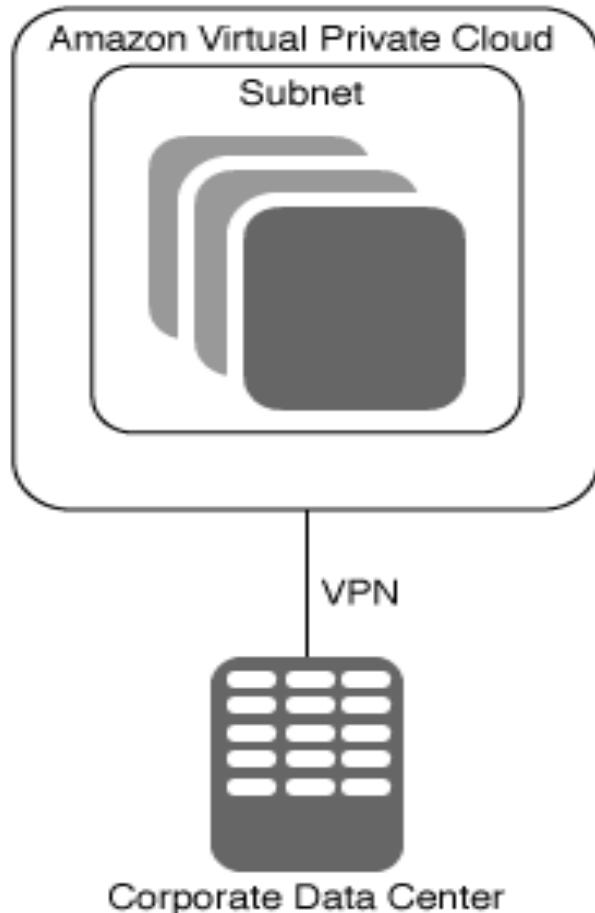
Virtual Private Cloud
Subnets
Route Tables, Security Groups, NACLs
Virtual Private Gateway
AWS Direct Connect
Internet Gateway
Elastic IPs and Load Balancers

AWS VPC (Single Public Subnet)



Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

AWS VPC (Single Private Subnet H/W VPN)

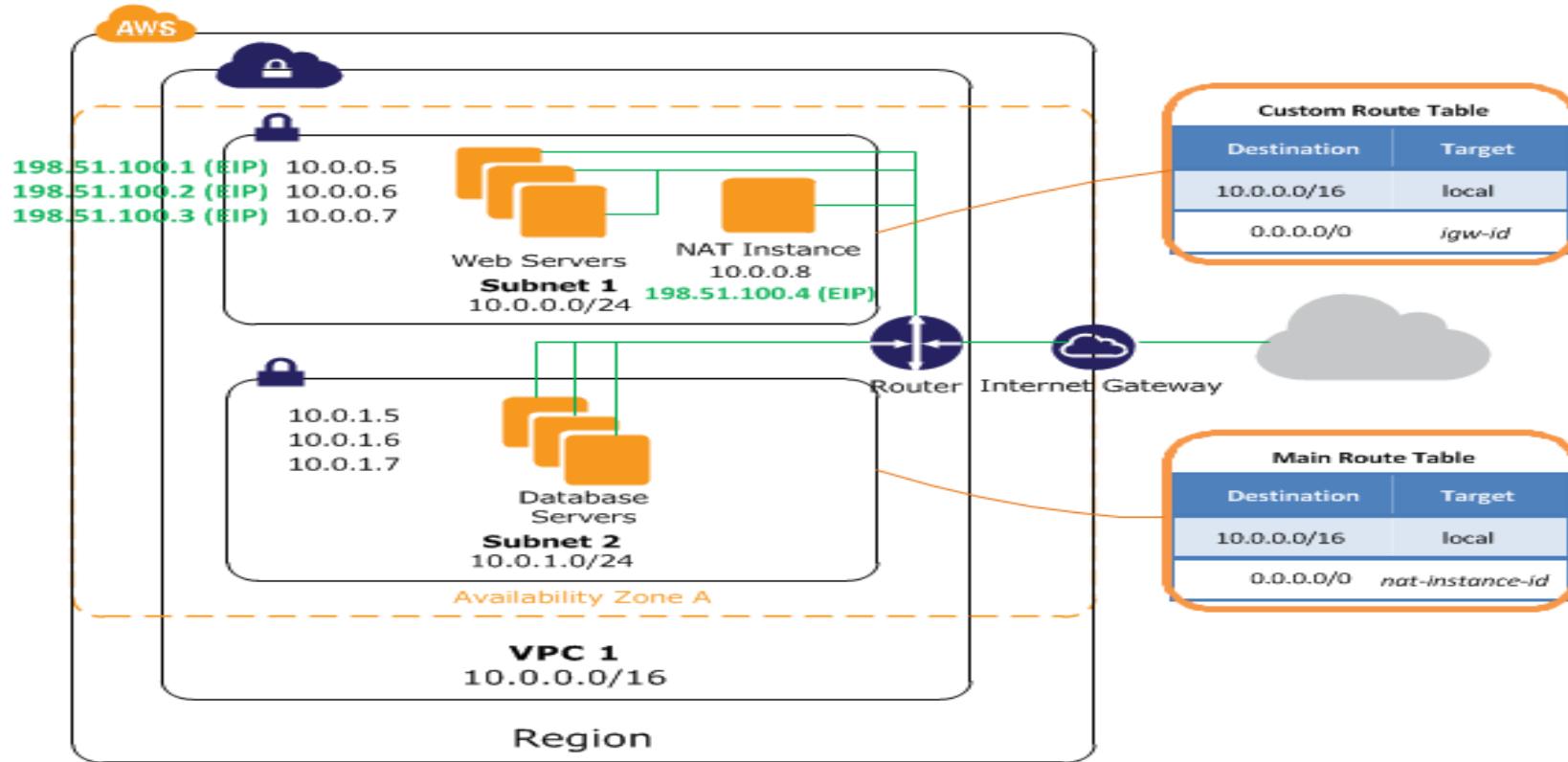


Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate data center via an IPsec Virtual Private Network (VPN) tunnel.

AWS VPC



- This is a diagram of a typical scenario you can create full details can be found [here](#).



AWS VPC

VPC Dashboard

Filter by VPC:
None

Virtual Private Cloud

Your VPCs

Resources

Start VPC Wizard Launch EC2 Instances

Note: Your Instances will launch in the EU West (Ireland) region.

You are using the following Amazon VPC resources in the EU West (Ireland) region:

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

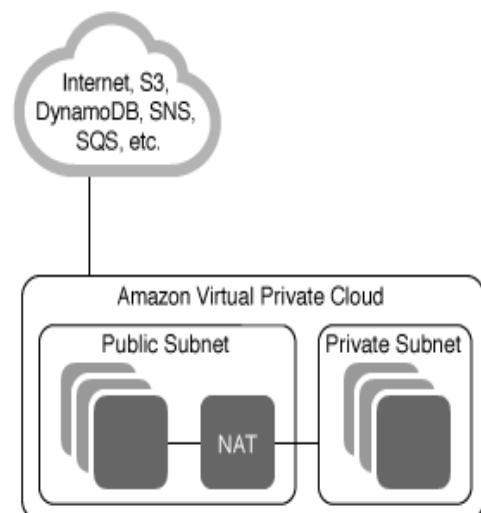
VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply.)

Select



AWS VPC



You will need to create the following security groups

- WebServerSG—For the web servers in the public subnet
- DBServerSG—For the database servers in the private subnet

AWS VPC



- From the Your VPCs screen note the details for your VPC – VPC ID, DHCP Options set, Main Route table, Default Network ACL.
- Also note the Subnets, Internet Gateways and Elastic IPs that have been created for your VPC. You should clearly name your VPC resources.

The screenshot shows the AWS VPC console interface. At the top, there's a table titled 'Your VPCs' with columns: Name, VPC ID, State, VPC CIDR, and DHCP options set. One row is selected, showing 'RFVPC' with VPC ID 'vpc-39ad555c', State 'available', VPC CIDR '10.200.0.0/16', and DHCP options set 'dopt-5038083b'. Below this, a modal window is open for the selected VPC ('vpc-39ad555c (10.200.0.0/16) | RFVPC'). The modal has tabs for 'Summary' (selected), 'Tags', and 'Edit'. The 'Edit' tab is currently active. Under 'Edit', there are two columns of settings:

VPC ID:	Network ACL:
vpc-39ad555c RFVPC	acl-bae928df
State:	Tenancy:
available	Default
VPC CIDR:	DNS resolution:
10.200.0.0/16	yes
DHCP options set:	DNS hostnames:
dopt-5038083b	yes
Route table:	
rtb-cd3bfba8	

AWS VPC



- You can choose yourself whether you want to work with Windows or Linux machines or a mixture of both.
- Launch a web server in the Public subnet in the VPC. Make sure you enable Auto-Assign Public IP address.
- You should put in some meaningful details in the Instance details tags key – value screen e.g. RFwebserver
- Launch the server in the relevant Security Group e.g. RFWebServerSG
- You will see both the Private and Public IP addresses assigned to this server. You can configure a webserver and connect to the Public IP address from your own desktop.

AWS VPC



- Now you can launch a Linux instance – you can choose a basic AMI - this instance must be launched in the private. This Server should be launched into the DBServerSG.
- You **DO NOT** want to Auto-Assign a Public IP address to this server.
- If you enable ssh from the WebServerSG to the DBServerSG you will be able to login from the Server in the Public subnet to the server in the Private subnet.
- Once you ssh from your webserver instance to your dbinstance you can check your public IP address using **wget <http://ipinfo.io/ip> -qO -**
- What is the Public IP address of the server in your Private Network ? What does it correspond with?

AWS VPC



- When you have investigated this VPC Scenario you can terminate your instances in the Public and Private subnets.
- In this exercise you created your own VPC with Public and Private subnets.
- Note you can delete your VPC and all associated resources (NAT gateway, instances, Elastic IPs, etc.)

References



- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenario2.html
- How to securely manage AWS credentials
 - <https://blogs.aws.amazon.com/security/post/Tx3D6U6WSFGOK2H/A-New-and-Standardized-Way-to-Manage-Credentials-in-the-AWS-SDKs>
- How to login securely to Linux AMI in VPC Private subnet using ssh agent forwarding
 - <https://blogs.aws.amazon.com/security/post/Tx3N8GFK85UN1G6/Securely-connect-to-Linux-instances-running-in-a-private-Amazon-VPC>