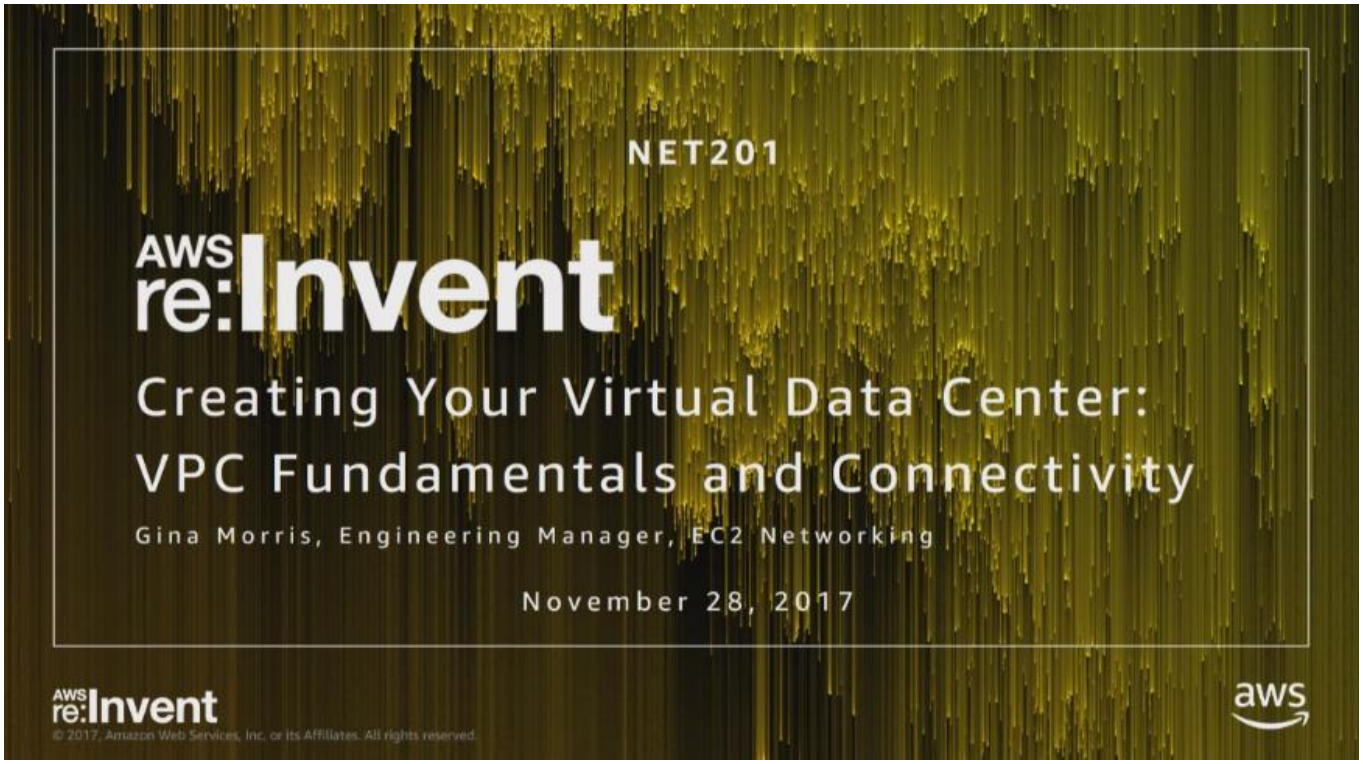


NET201

AWS re:Invent

Creating Your Virtual Data Center: VPC Fundamentals and Connectivity

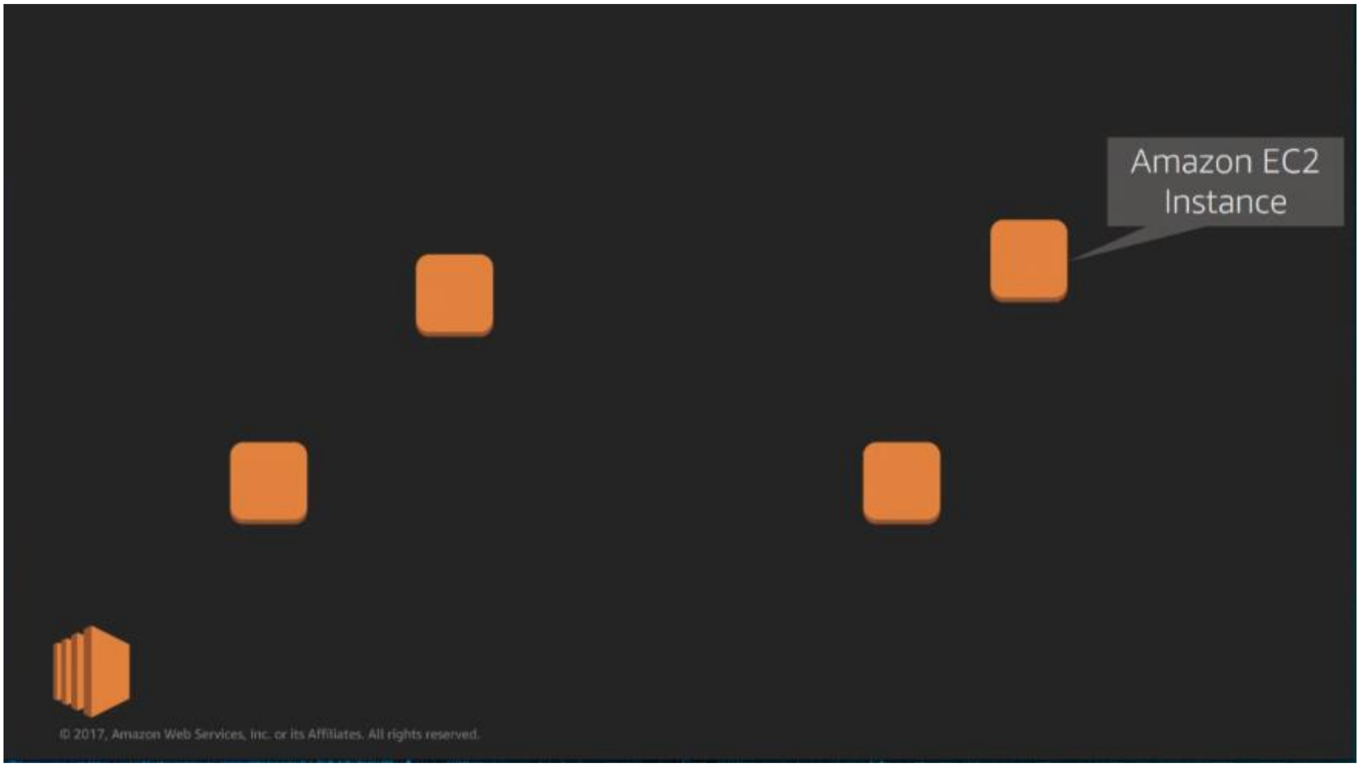
Gina Morris, Engineering Manager, EC2 Networking

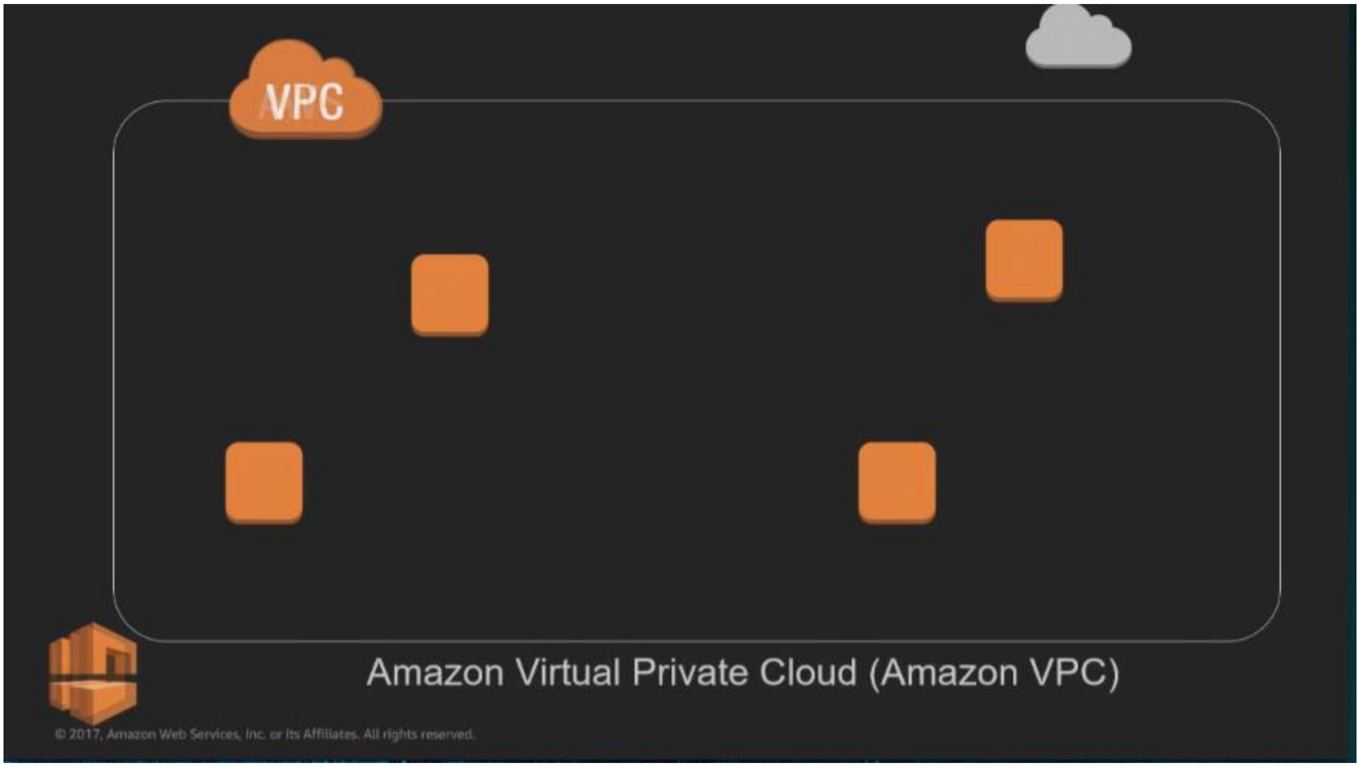
November 28, 2017

AWS re:Invent

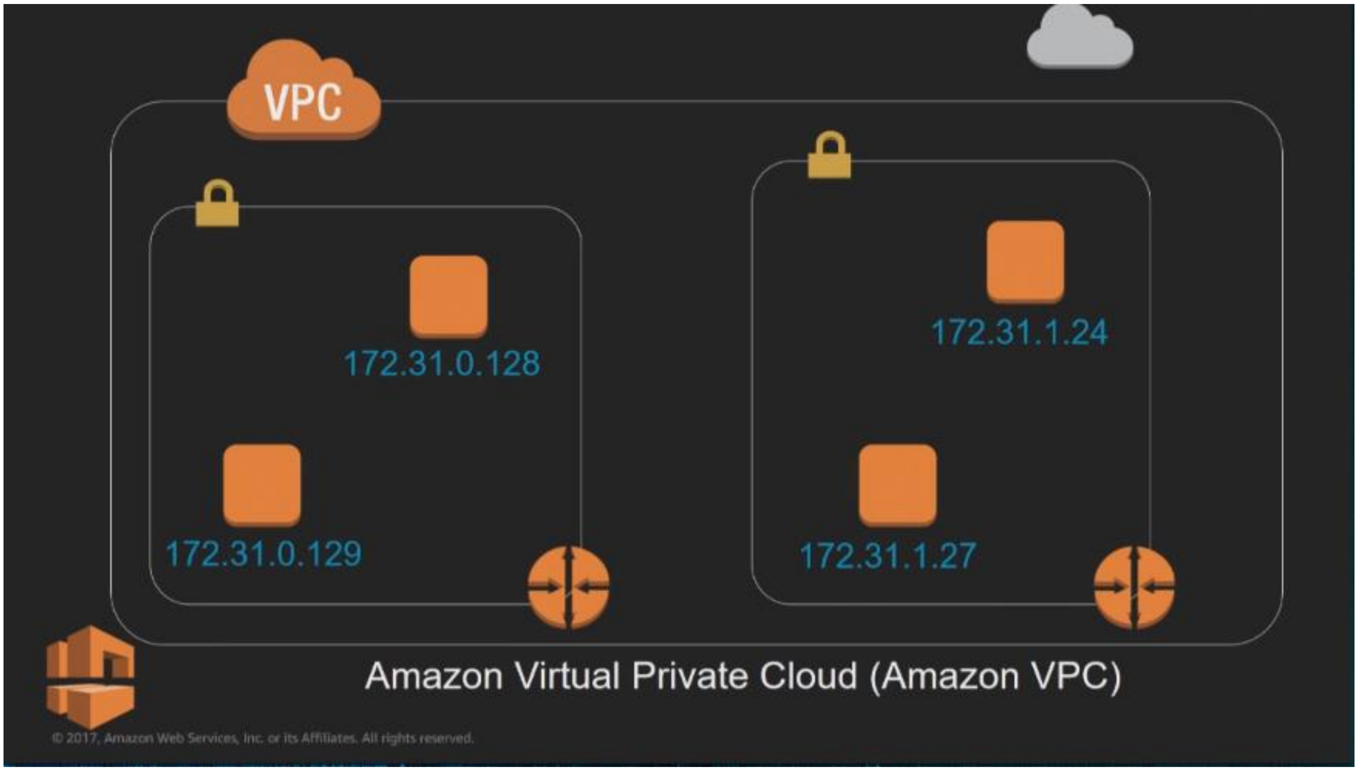
aws

Amazon VPC is how you do networking on AWS. In this session, we walk through the fundamentals of Amazon VPC. First, we cover build-out and design fundamentals for VPCs, including picking your IP space, subnetting, routing, security, NAT, and much more. We then transition into different approaches and use cases for optionally connecting your VPC to your physical data center with VPN or AWS Direct Connect. This ***mid-level architecture discussion is aimed at architects***, network administrators, and technology decision-makers interested in understanding the building blocks that AWS makes available with ***Amazon VPC***. Learn how you can connect VPCs with your offices and current data center footprint.

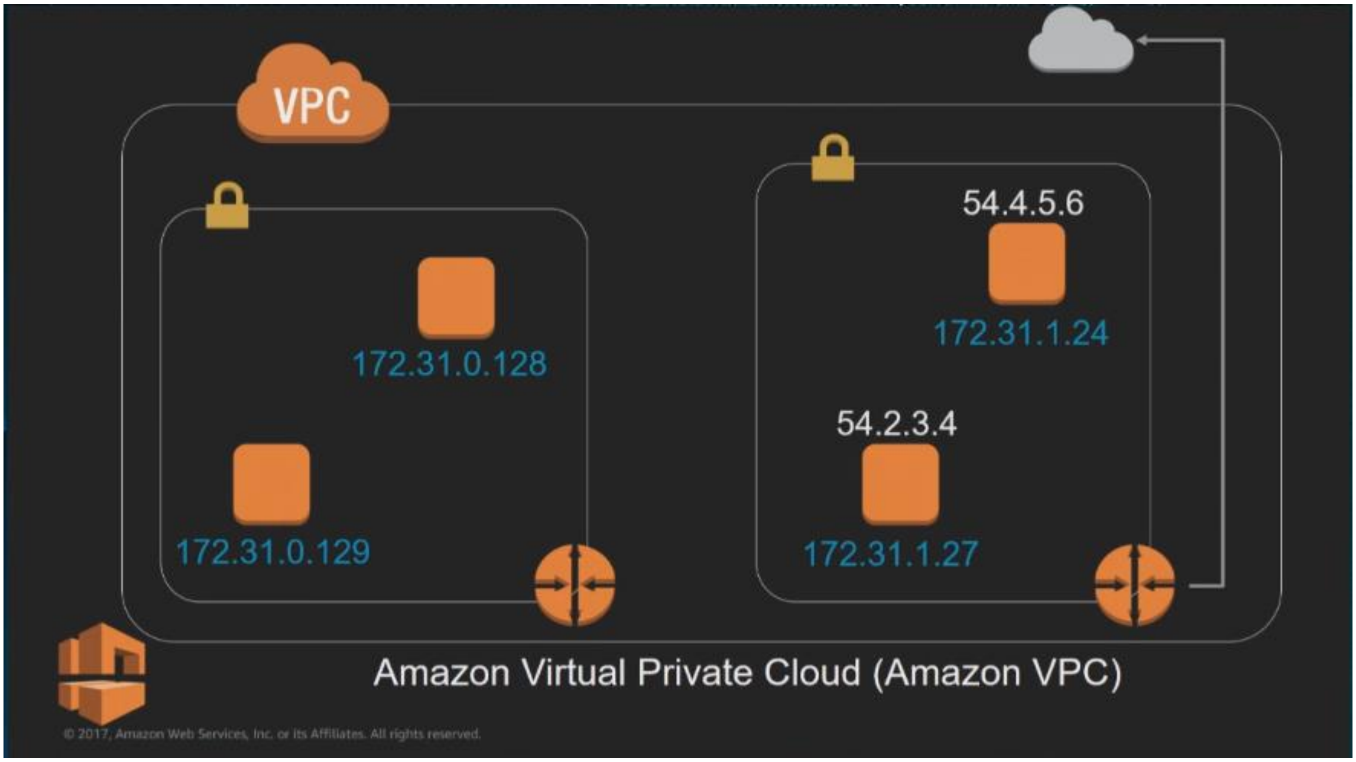



VPC
Amazon Virtual Private Cloud (Amazon VPC)
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

The EC2 instances can be run in your VPC, you have full control over the EC2 instances and can pick things for them like the IP address,


VPC
172.31.0.128
172.31.0.129
172.31.1.24
172.31.1.27
Amazon Virtual Private Cloud (Amazon VPC)
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

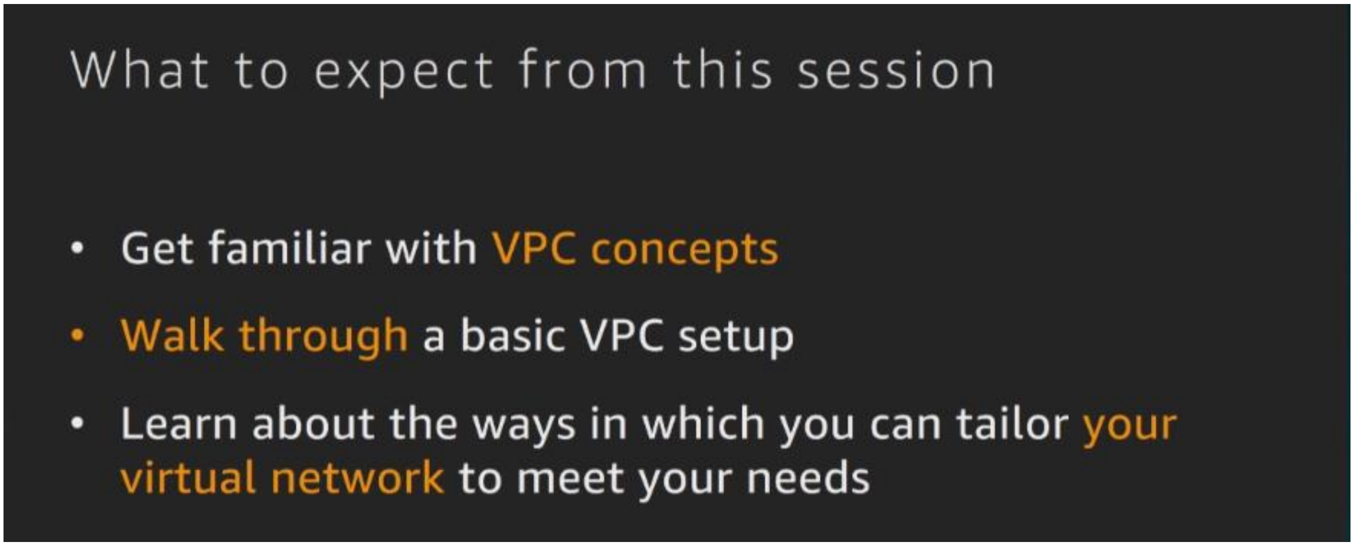
You are going to divide the VPC up into smaller sub-networks as above (for higher availability),



Amazon Virtual Private Cloud (Amazon VPC)

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

You can also choose what connectivity to use for certain parts of your network,



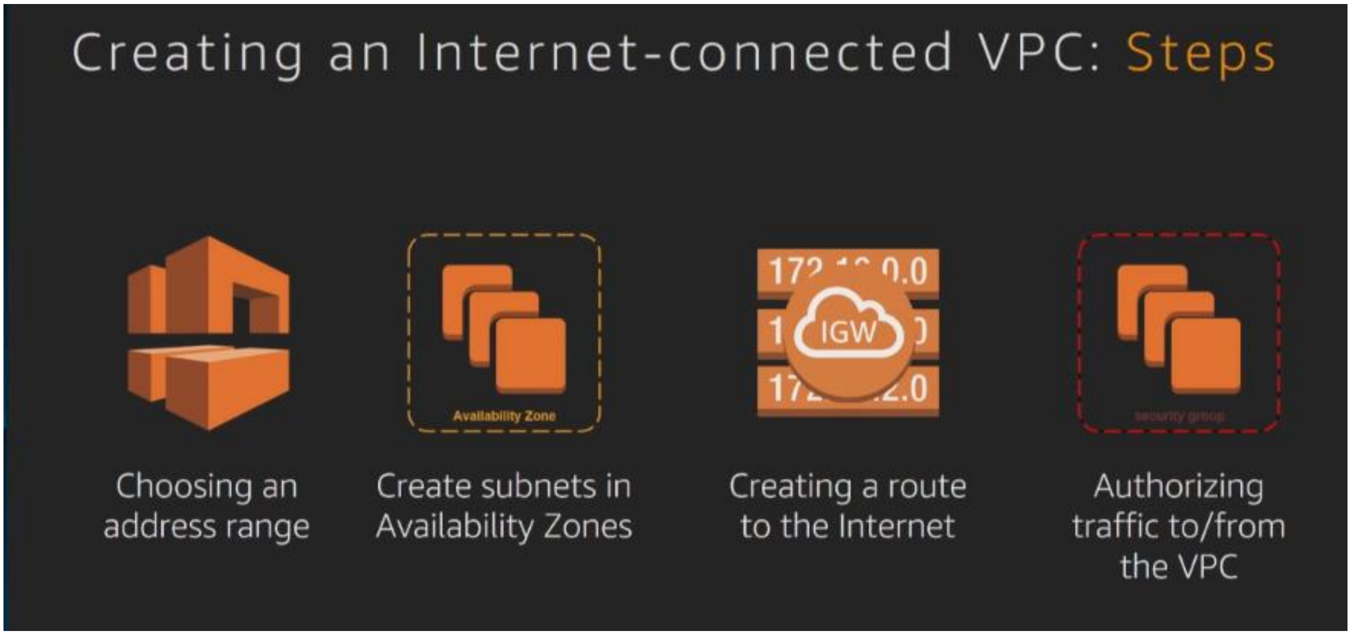
What to expect from this session

- Get familiar with VPC concepts
- Walk through a basic VPC setup
- Learn about the ways in which you can tailor your virtual network to meet your needs

Walkthrough:

Setting up an Internet-connected VPC

This is a VPC with internet access, it is called your ***default VPC***.



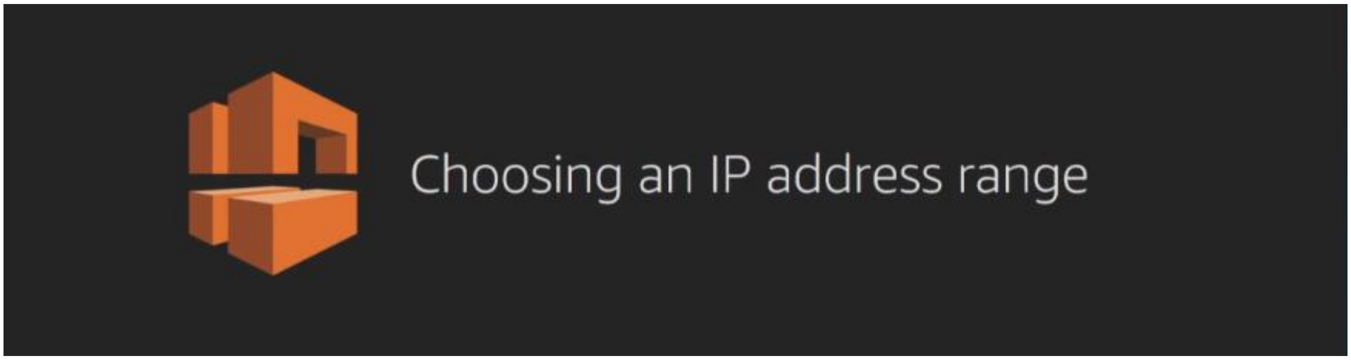
Creating an Internet-connected VPC: Steps

Choosing an address range

Create subnets in Availability Zones

Creating a route to the Internet

Authorizing traffic to/from the VPC



Choosing an IP address range

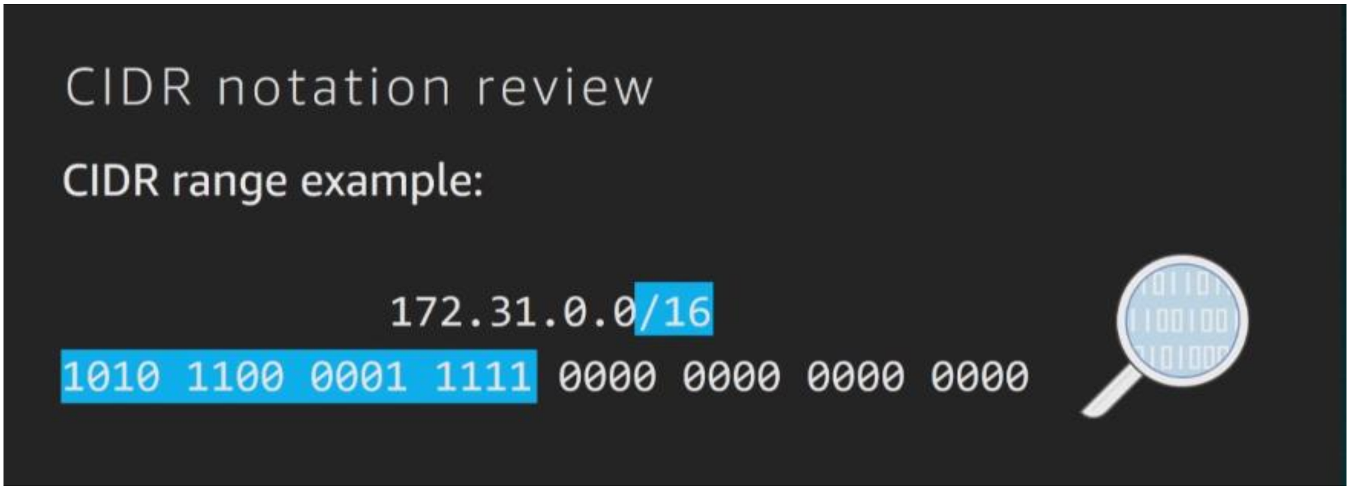
This is our private network, so we get to pick an IP address range to use within it.

CIDR notation review

CIDR range example:

172.31.0.0/16

This is Classless Inter-Domain Routing ***CIDR*** notation that we use a lot in VPC networks, it is an IPv4 address followed by a slash followed by a number.



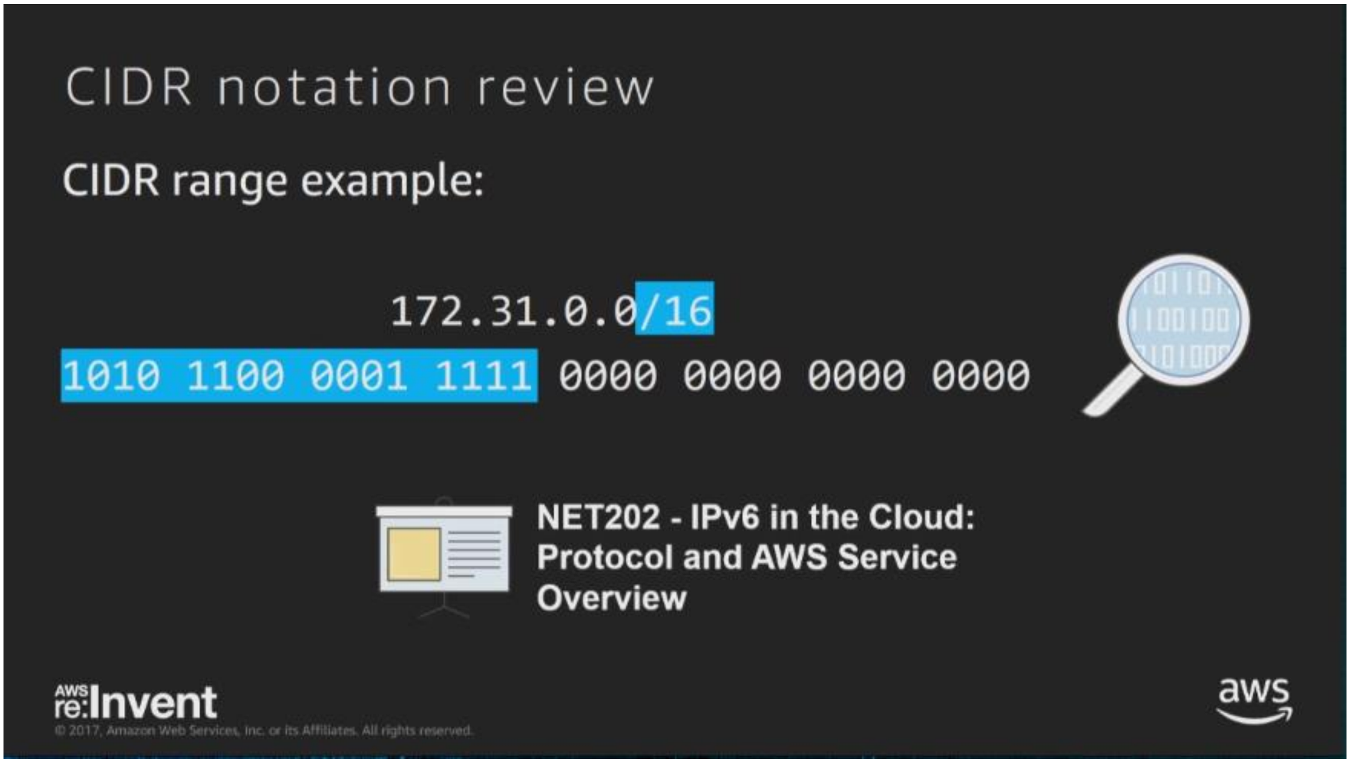
CIDR notation review

CIDR range example:

172.31.0.0/16

1010 1100 0001 1111 0000 0000 0000 0000

CIDR works by writing out the IPv4 address out in binary to get a 32-bit number as above, the ‘/16’ means that you hold the highest 16 bits steady and the lower 16 bits can vary. So, this gives us the IP range 172.31.x.x because all those possibilities can exist in this IP range above



CIDR notation review

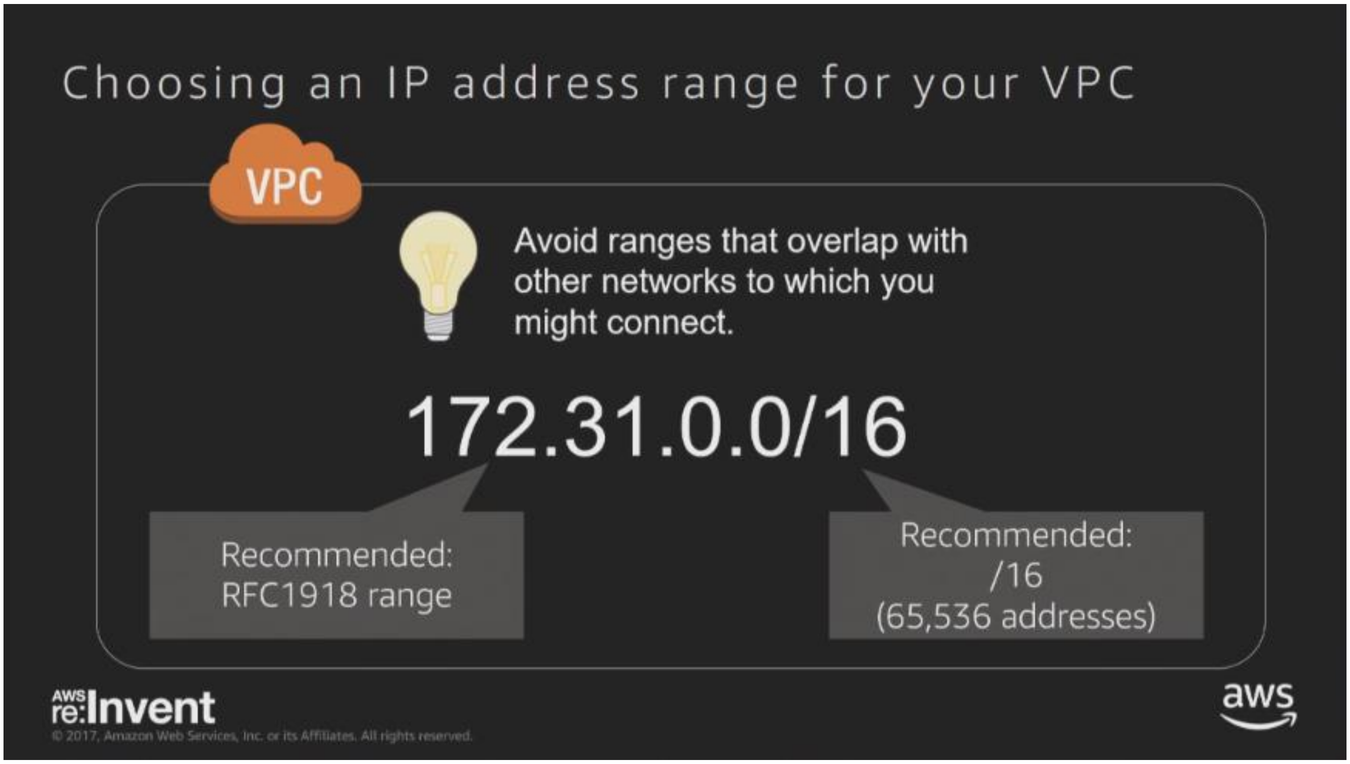
CIDR range example:

172.31.0.0/16

1010 1100 0001 1111 0000 0000 0000 0000

NET202 - IPv6 in the Cloud: Protocol and AWS Service Overview

VPC also supports IPv6 and there is a session later today about that



Choosing an IP address range for your VPC

VPC

Avoid ranges that overlap with other networks to which you might connect.

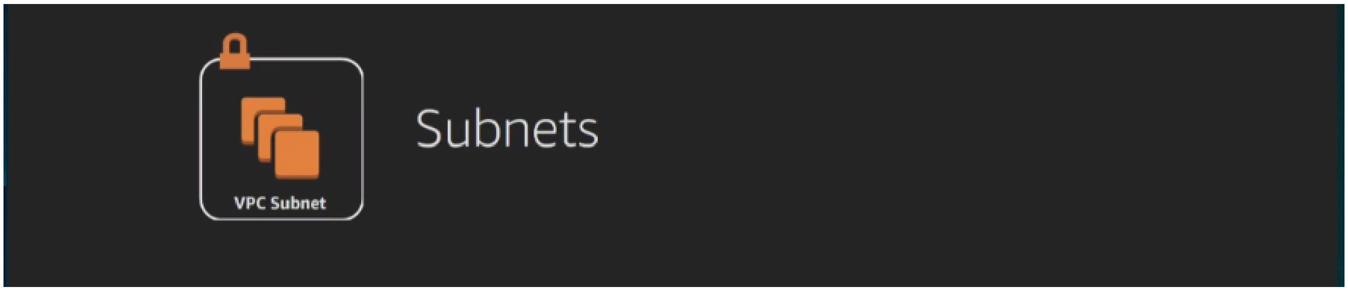
172.31.0.0/16

Recommended:
RFC1918 range

Recommended:
/16
(65,536 addresses)

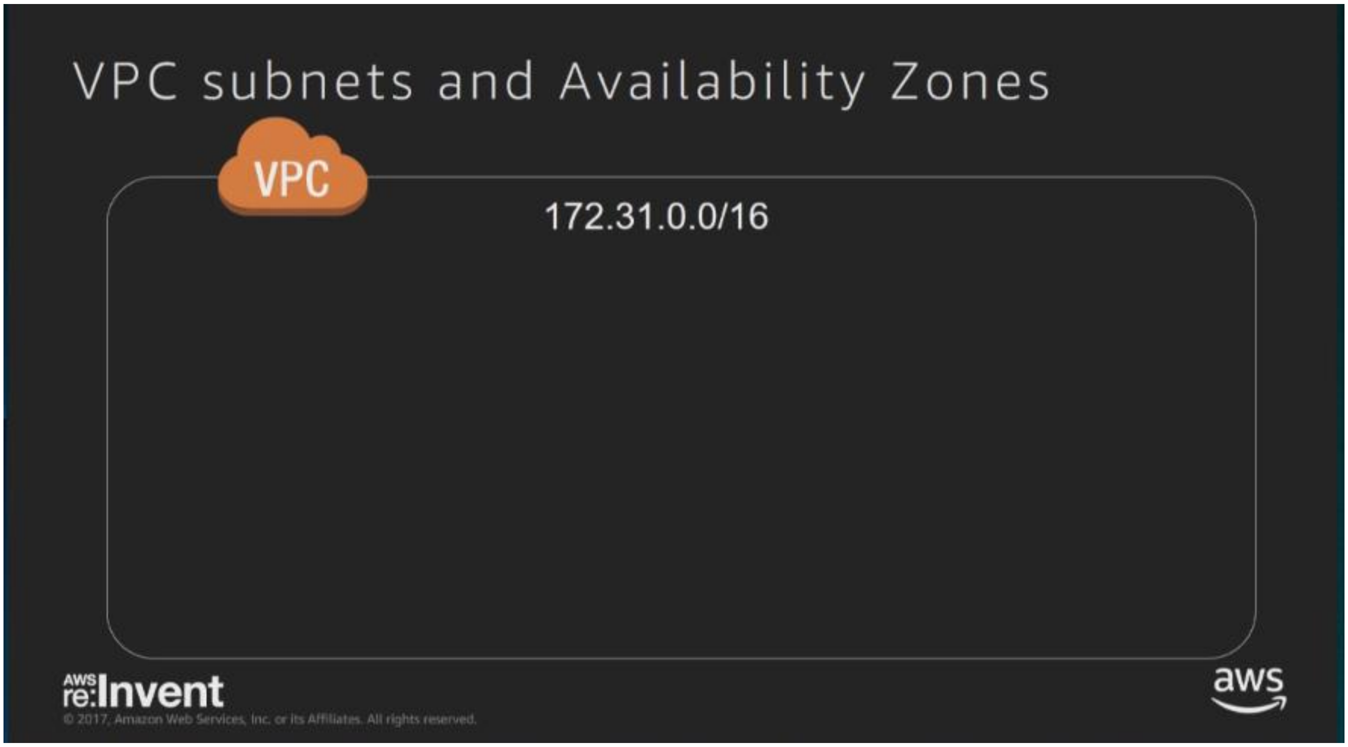
AWS re:Invent
aws

In RFC1918, ***172.x.x.x is a recommendation for private networks IP addresses***, the ***'/16' gives us over 65,536 IP address*** possibilities to use within that VPC for our instances. If you have other VPCs or data centers that you will like to connect this VPC to, you need to pick ranges of IP addresses that do not overlap each other to prevent conflicts. You can start with a smaller VPC IP ranges to be safe



Subnets

Subnets are a sub-range of that IP address space we created for the VPC. ***Subnets are how you use the VPC to deploy high availability applications***.



VPC subnets and Availability Zones

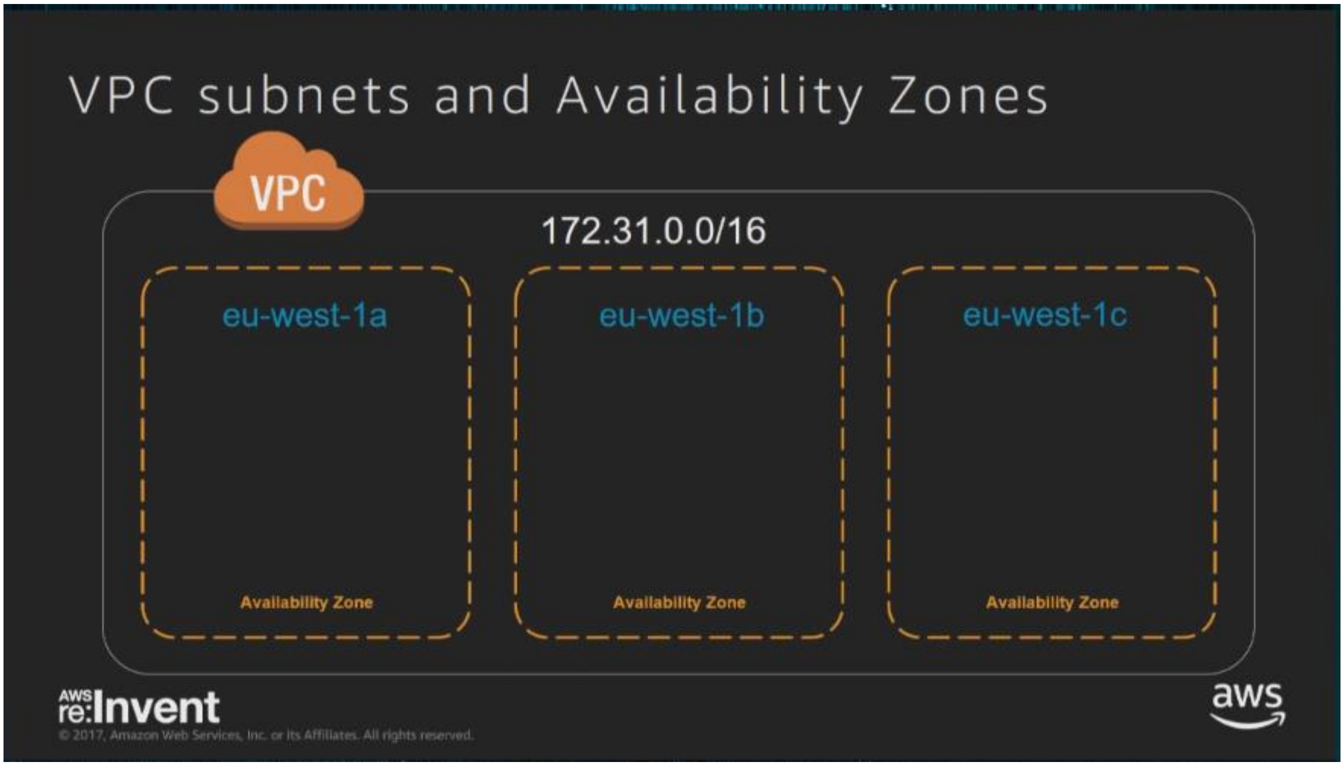
VPC

172.31.0.0/16

AWS re:Invent

aws

AWS has 16 Regions and a VPC is within a Region.



VPC subnets and Availability Zones

VPC

172.31.0.0/16

eu-west-1a

Availability Zone

eu-west-1b

Availability Zone

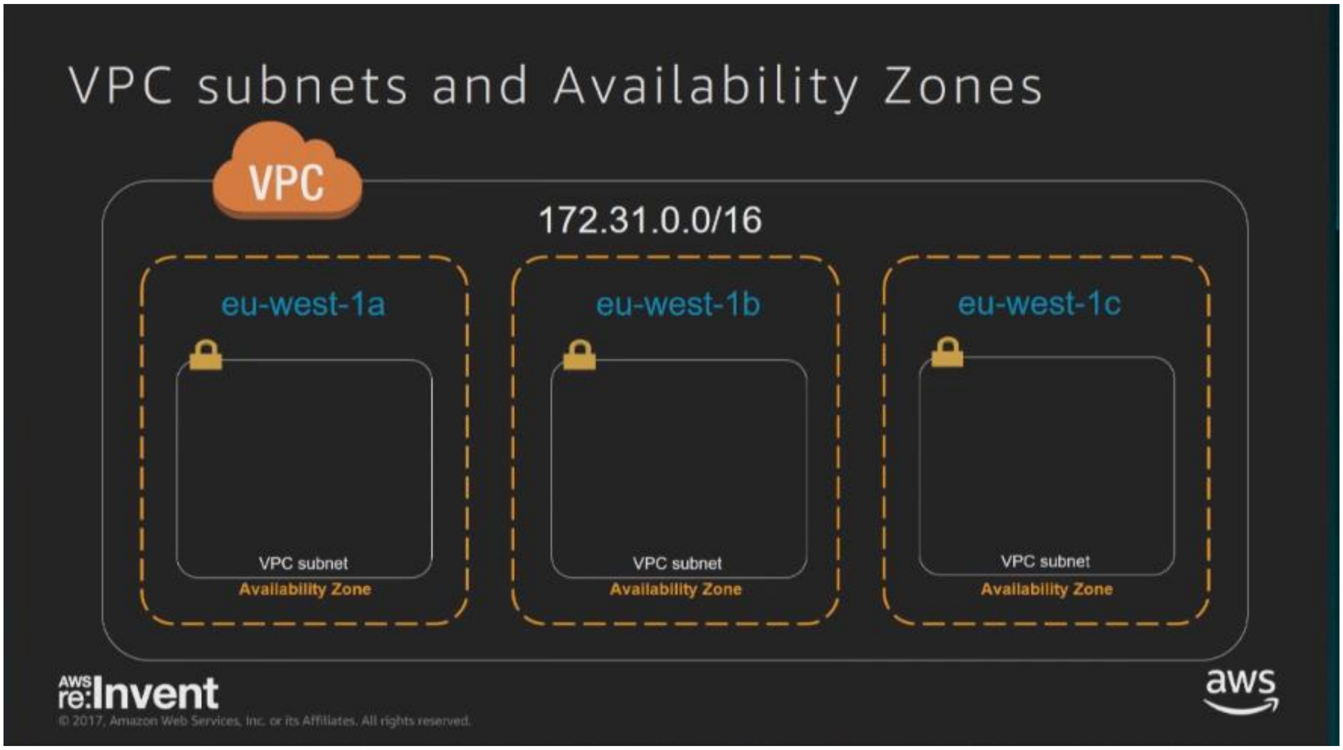
eu-west-1c

Availability Zone

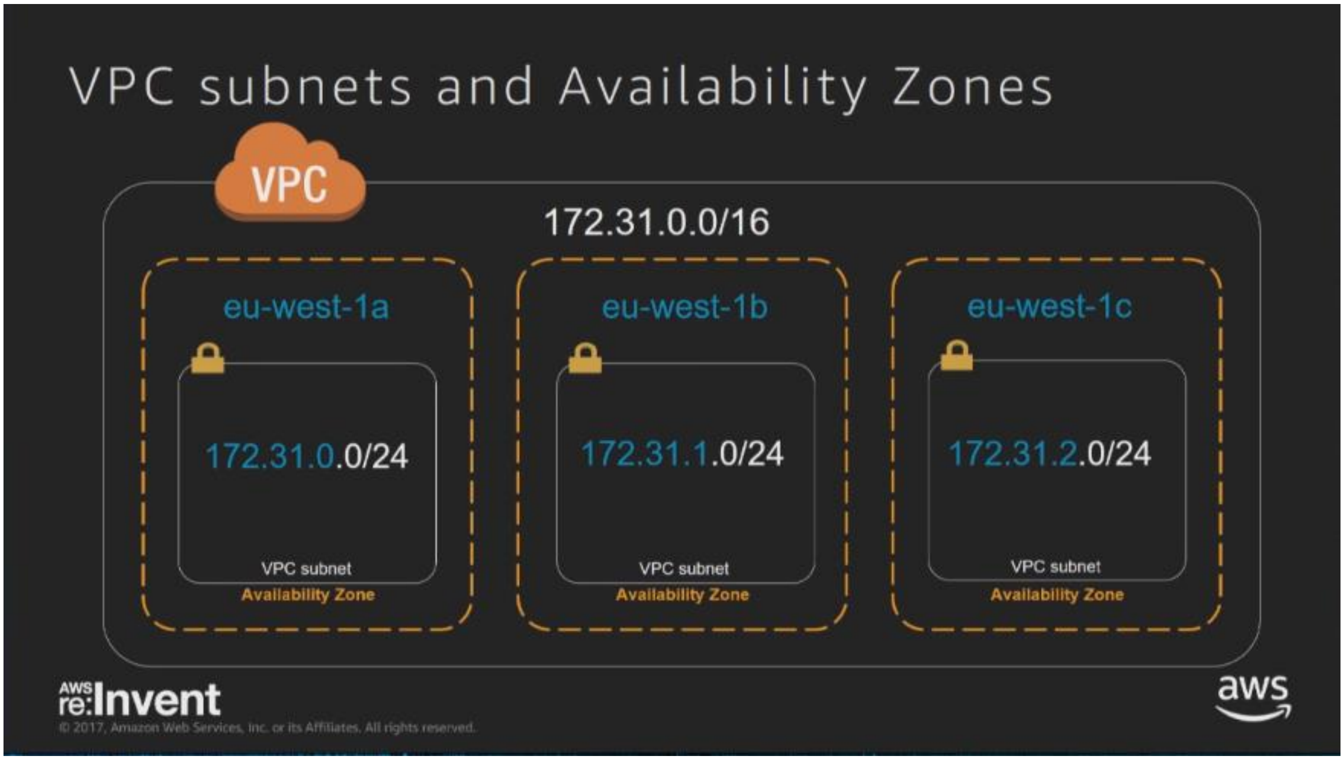
AWS re:Invent

aws

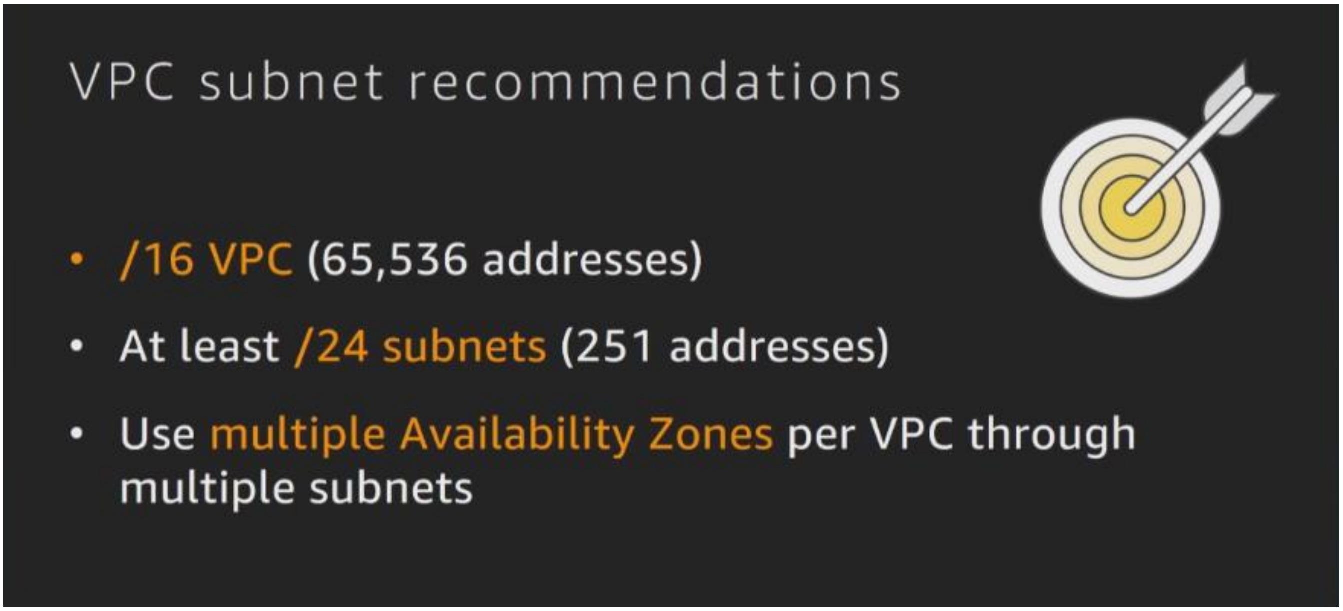
A Region is divided into AZs as above. ***An AZ is like a data center with its own redundant power and networking for isolation***. You generally want to build your apps across multiple AZs.



A Subnet is within an AZ and you need subnets because when you create certain AWS resources like EC2 instances, you will have to specify the subnets that you want those resources deployed in. ***Having subnets is how you get your AWS resources into specific AZs***. You can add or remove subnets within an AZ and even have multiple subnets within an AZ.



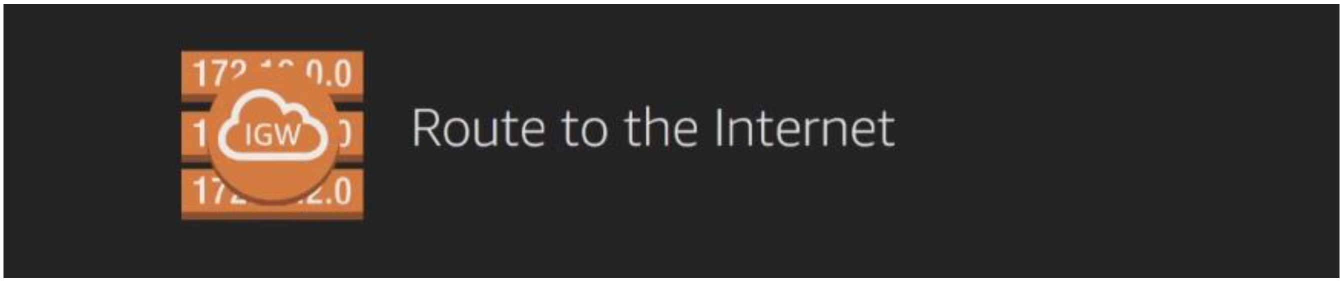
We also ***pick IP address ranges for our subnet from within the VPCs IP address range*** as above.



VPC subnet recommendations

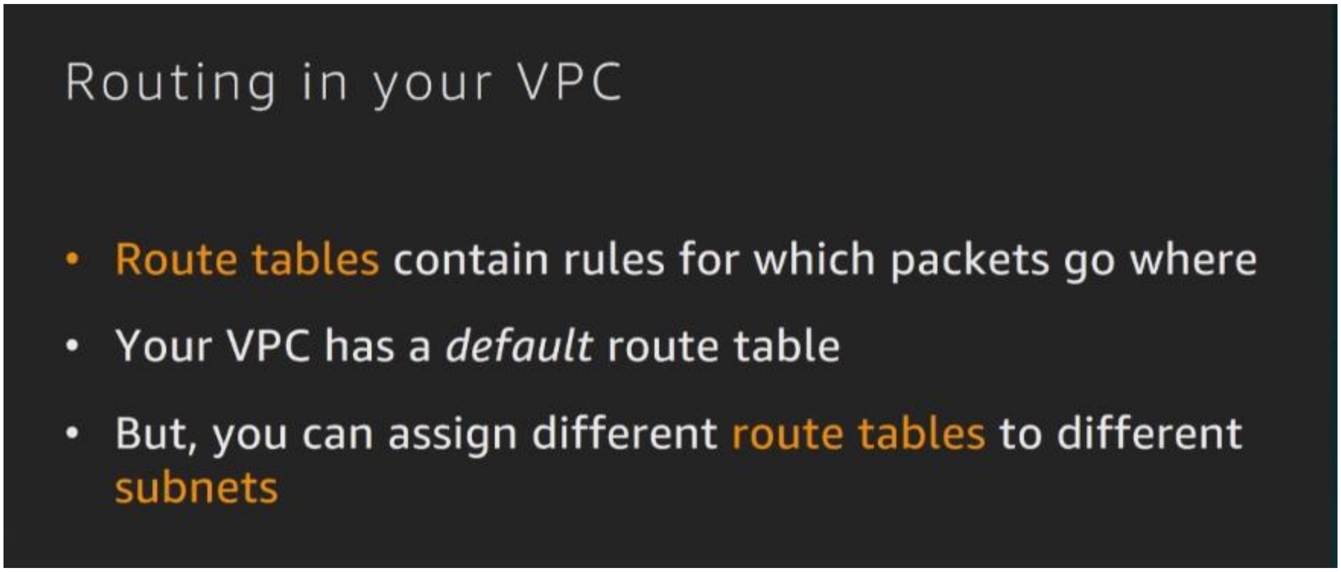
- /16 VPC (65,536 addresses)
- At least /24 subnets (251 addresses)
- Use multiple Availability Zones per VPC through multiple subnets

5 IP addresses are reserved in a subnet's possible 256 addresses to give us the 251 address range above.



Route to the Internet

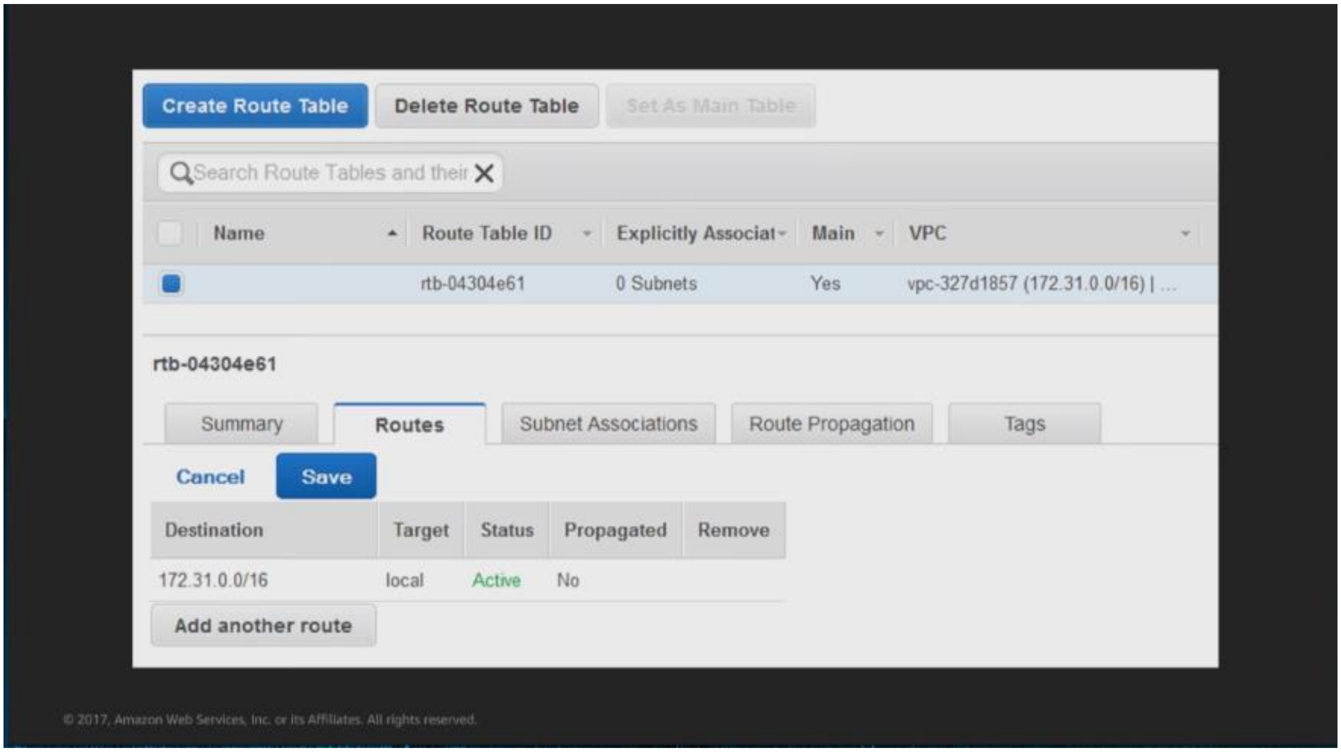
How do we send traffic to the internet?



Routing in your VPC

- Route tables contain rules for which packets go where
- Your VPC has a *default* route table
- But, you can assign different route tables to different subnets

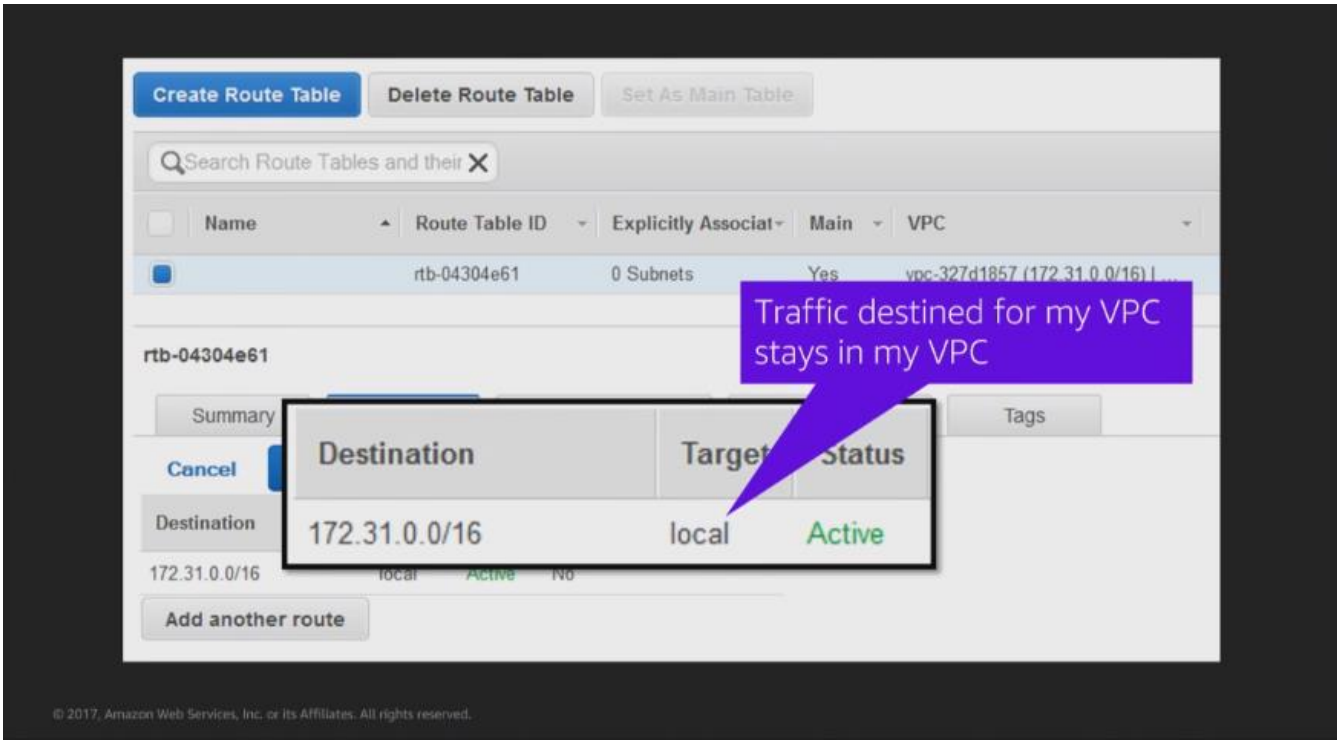
Routing in your VPC is done with Route Tables, which are a simple, easy to read list of rules that specifies which traffic should be sent to which gateway. You can override routing on a subnet by subnet basis.



Name	Route Table ID	Explicitly Associat	Main	VPC
	rtb-04304e61	0 Subnets	Yes	vpc-327d1857 (172.31.0.0/16) | …

rtb-04304e61

Destination	Target	Status	Propagated	Remove
172.31.0.0/16	local	Active	No	



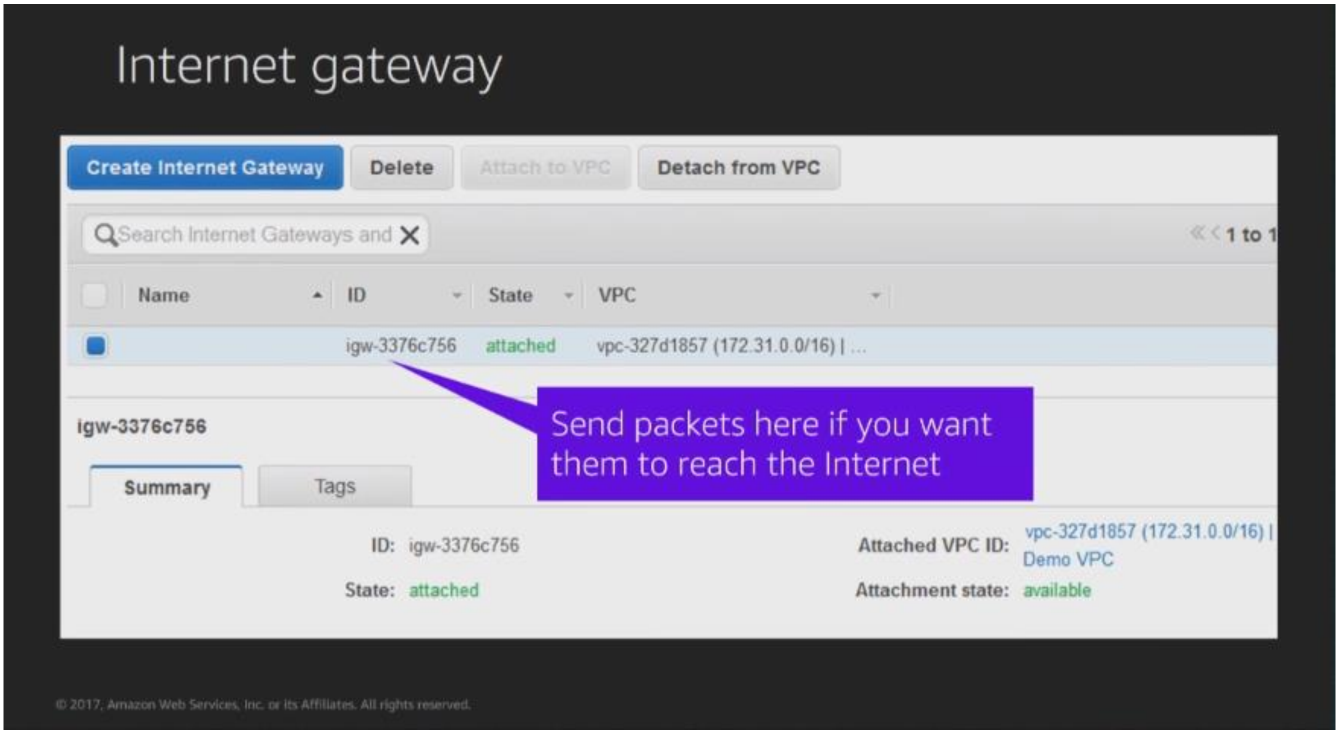
Name	Route Table ID	Explicitly Associat	Main	VPC
	rtb-04304e61	0 Subnets	Yes	vpc-327d1857 (172.31.0.0/16) | …

rtb-04304e61

Traffic destined for my VPC stays in my VPC

Destination	Target	Status
172.31.0.0/16	local	Active

There is only one rule here for traffic destined for anywhere within this VPC to stay get routed to destination within this VPC, other traffic going elsewhere we just get DROPPED!



Internet gateway

Name	ID	State	VPC
	igw-3376c756	attached	vpc-327d1857 (172.31.0.0/16) | ...

Send packets here if you want them to reach the Internet

igw-3376c756

ID: igw-3376c756

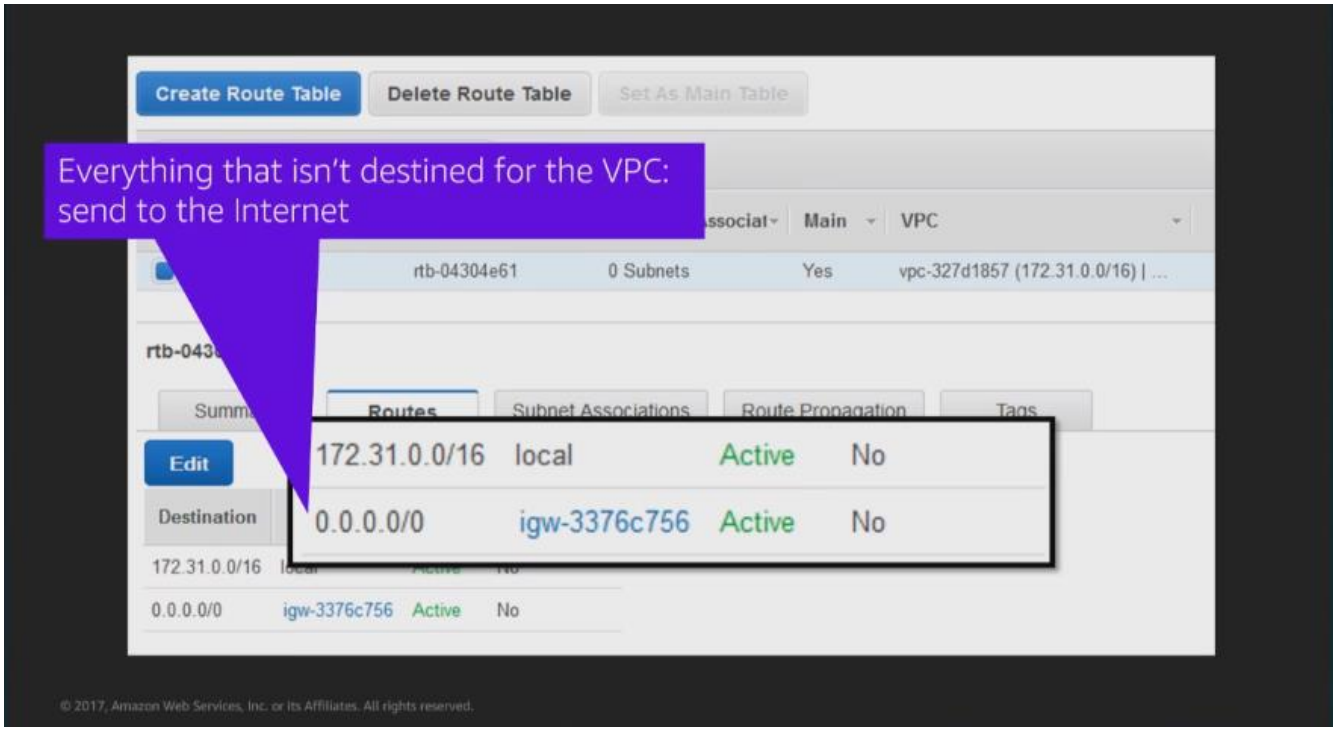
State: attached

Attached VPC ID: vpc-327d1857 (172.31.0.0/16) | Demo VPC

Attachment state: available

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

To send traffic to the internet you need to create an internet gateway IGW, and you attach the IGW to your VPC.



Everything that isn't destined for the VPC: send to the Internet

	ssociat	Main	VPC
rtb-04304e61	0 Subnets	Yes	vpc-327d1857 (172.31.0.0/16) | ...

rtb-043

172.31.0.0/16	local	Active	No
0.0.0.0/0	igw-3376c756	Active	No

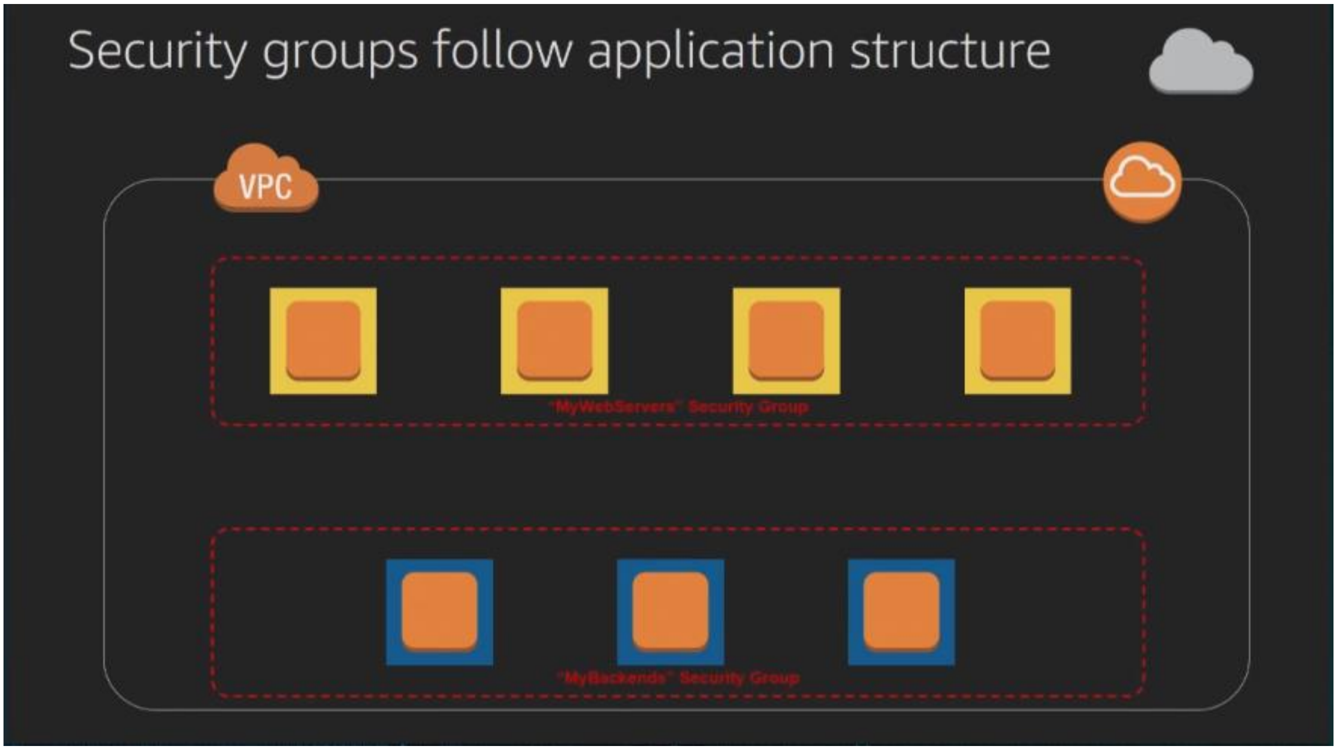
Destination			
172.31.0.0/16	local	Active	No
0.0.0.0/0	igw-3376c756	Active	No

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

We add a new route/rule that says anything that matches ***0.0.0.0/0 (CIDR notation for 'match all traffic')*** should be sent to the IGW for the internet. The IGW is an abstraction backed by something HA.

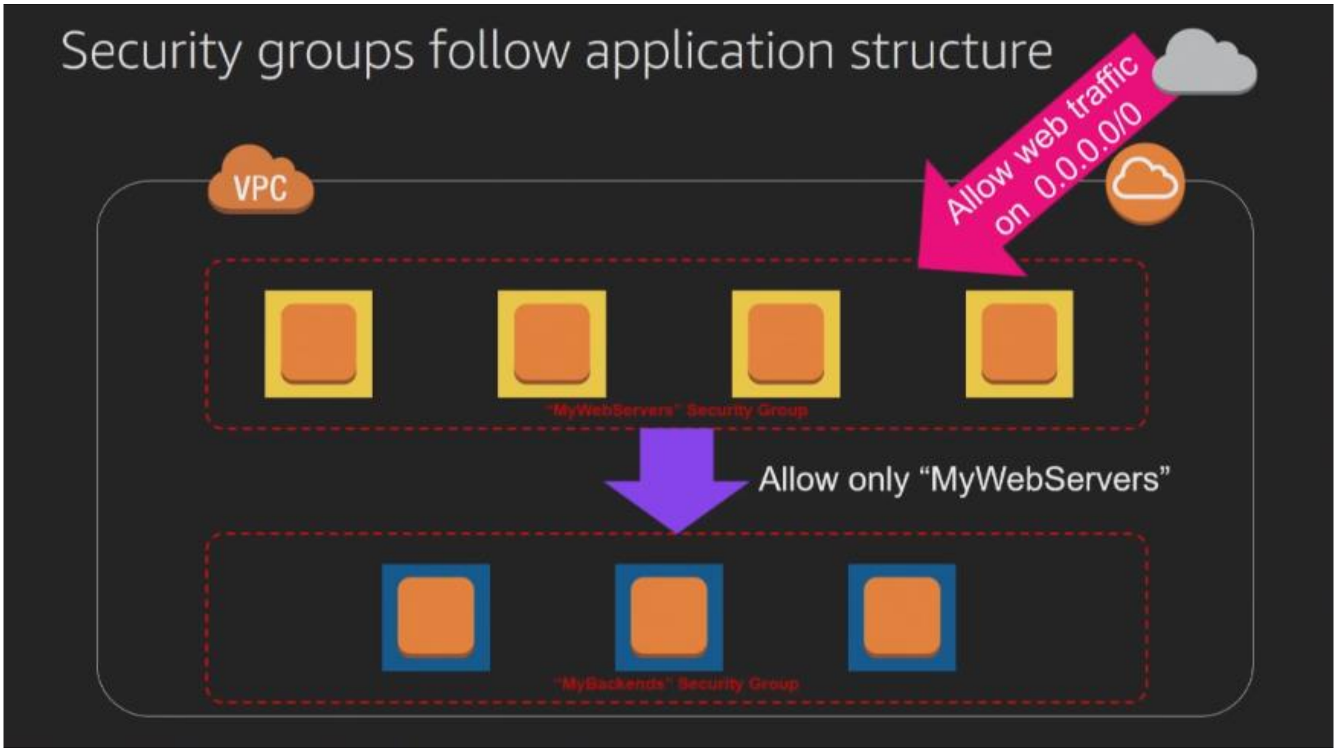
Network security in your VPC:
Security groups

We don’t want to just receive traffic from anywhere, this is why we need network security for our VPC. AWS VPC has tools for network security (in place of using firewalls in traditional data centers DCs).



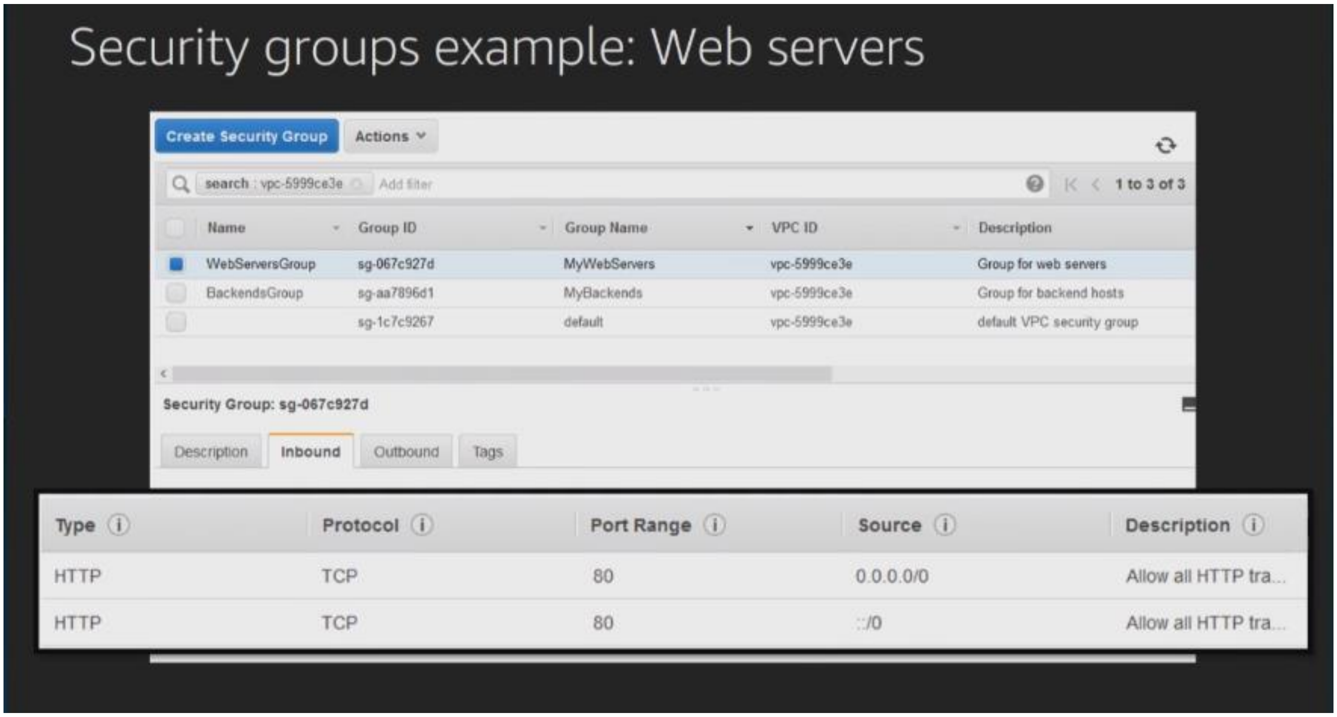
Security groups follow application structure

Assuming we have the yellow web services within our VPC that received requests from the internet, then they in turn make requests to the blue backend services for some data to fulfill those requests. They are in different security groups SGs with different rules applying to them as below



Security groups follow application structure

VPC
Allow web traffic on 0.0.0.0/0
"MyWebServers" Security Group
Allow only "MyWebServers"
"MyBackends" Security Group



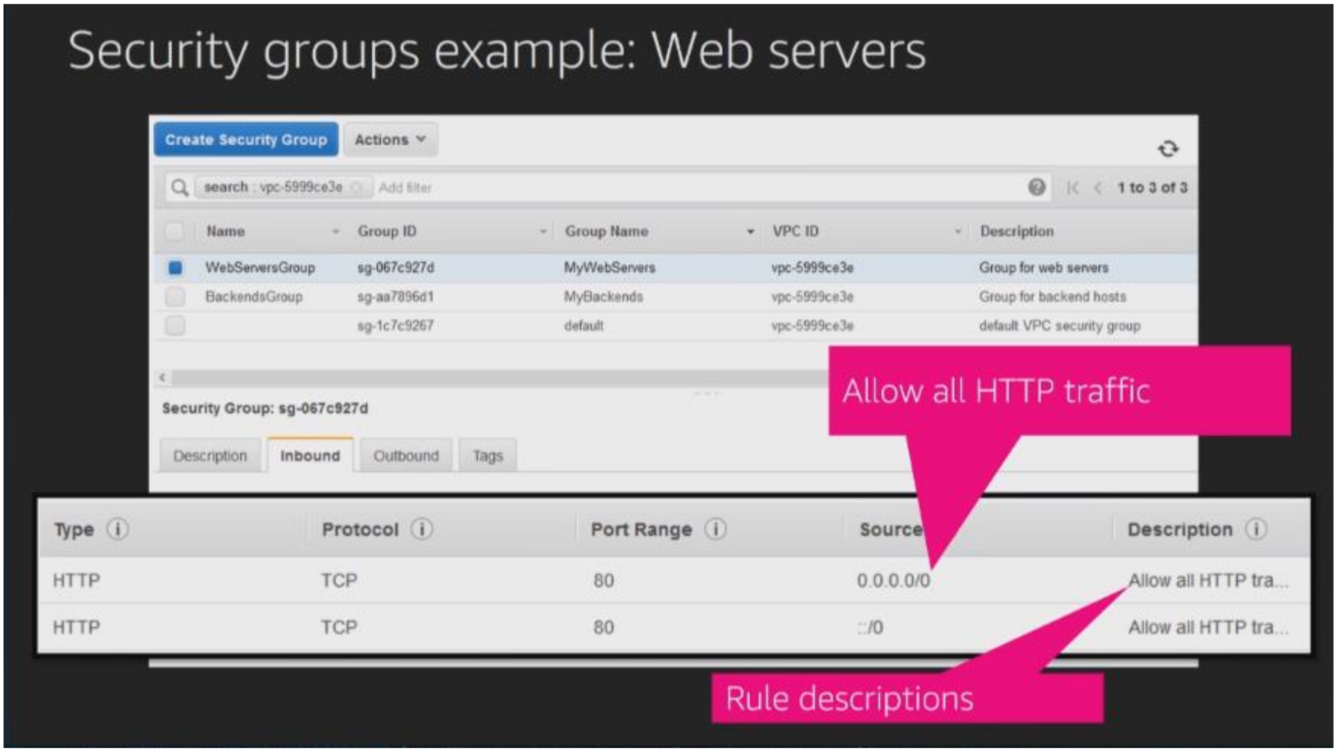
Security groups example: Web servers

	Name	Group ID	Group Name	VPC ID	Description
■	WebServersGroup	sg-067c927d	MyWebServers	vpc-5999ce3e	Group for web servers
□	BackendsGroup	sg-aa7896d1	MyBackends	vpc-5999ce3e	Group for backend hosts
□		sg-1c7c9267	default	vpc-5999ce3e	default VPC security group

Security Group: sg-067c927d

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow all HTTP tra...
HTTP	TCP	80	::/0	Allow all HTTP tra...

This is what it would look like in the EC2 console, we have created SGs to do the specific things we need and attached the instances to the relevant SGs



Security groups example: Web servers

Name	Group ID	Group Name	VPC ID	Description
WebServersGroup	sg-067c927d	MyWebServers	vpc-5999ce3e	Group for web servers
BackendsGroup	sg-aa7896d1	MyBackends	vpc-5999ce3e	Group for backend hosts
	sg-1c7c9267	default	vpc-5999ce3e	default VPC security group

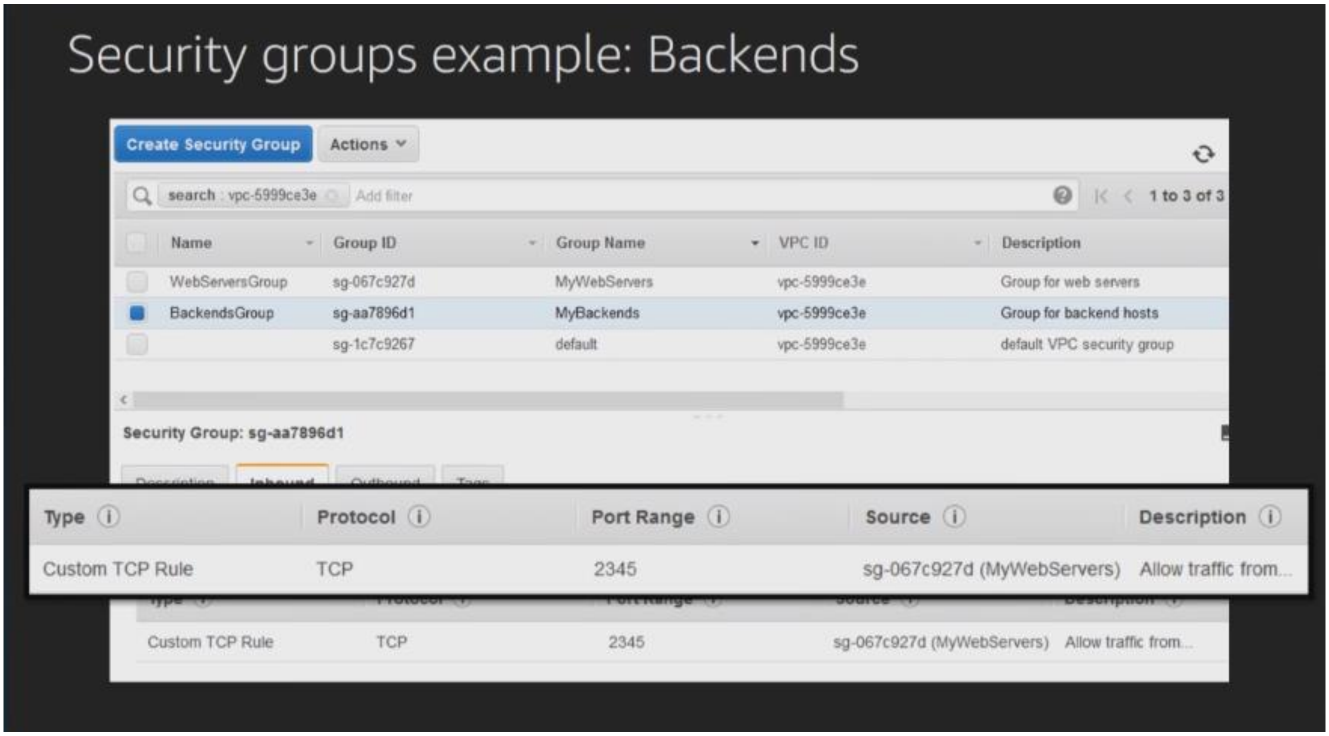
Security Group: sg-067c927d

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow all HTTP tra...
HTTP	TCP	80	::/0	Allow all HTTP tra...

Allow all HTTP traffic

Rule descriptions

SGs allow you to specify the Type of traffic like HTTP, the Protocol like TCP, the Port like 80, the Source IPs allowed like 0.0.0.0/0 and a description for each rule.



Security groups example: Backends

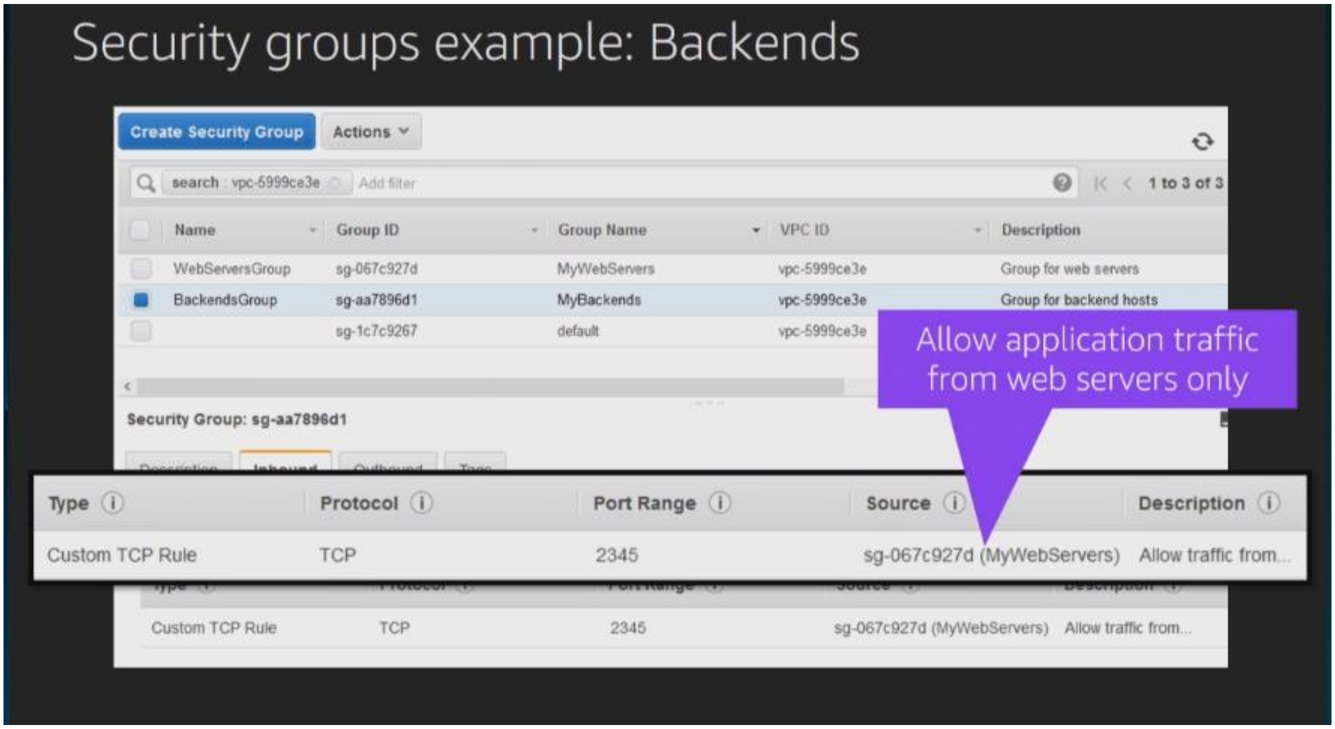
Name	Group ID	Group Name	VPC ID	Description
WebServersGroup	sg-067c927d	MyWebServers	vpc-5999ce3e	Group for web servers
BackendsGroup	sg-aa7896d1	MyBackends	vpc-5999ce3e	Group for backend hosts
	sg-1c7c9267	default	vpc-5999ce3e	default VPC security group

Security Group: sg-aa7896d1

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	2345	sg-067c927d (MyWebServers)	Allow traffic from...

Custom TCP Rule TCP 2345 sg-067c927d (MyWebServers) Allow traffic from...

For the backends SG that are not allowing all traffic



Security groups example: Backends

	Name	Group ID	Group Name	VPC ID	Description
□	WebServersGroup	sg-067c927d	MyWebServers	vpc-5999ce3e	Group for web servers
■	BackendsGroup	sg-aa7896d1	MyBackends	vpc-5999ce3e	Group for backend hosts
□		sg-1c7c9267	default	vpc-5999ce3e	

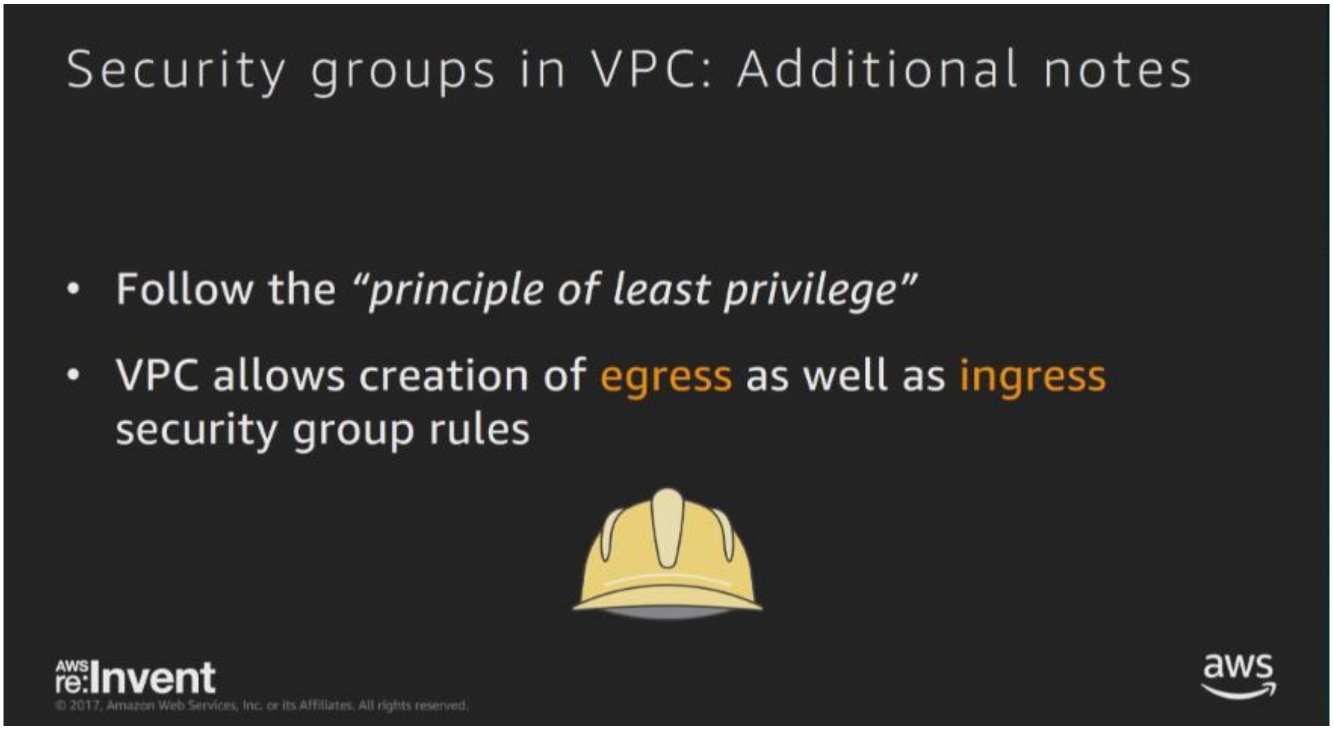
Security Group: sg-aa7896d1

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	2345	sg-067c927d (MyWebServers)	Allow traffic from...

Custom TCP Rule TCP 2345 sg-067c927d (MyWebServers) Allow traffic from...

Allow application traffic from web servers only

The Source is not a range of IPs but another SG, they will only allow traffic coming from any EC2 instances belonging to the MyWebServers security group.



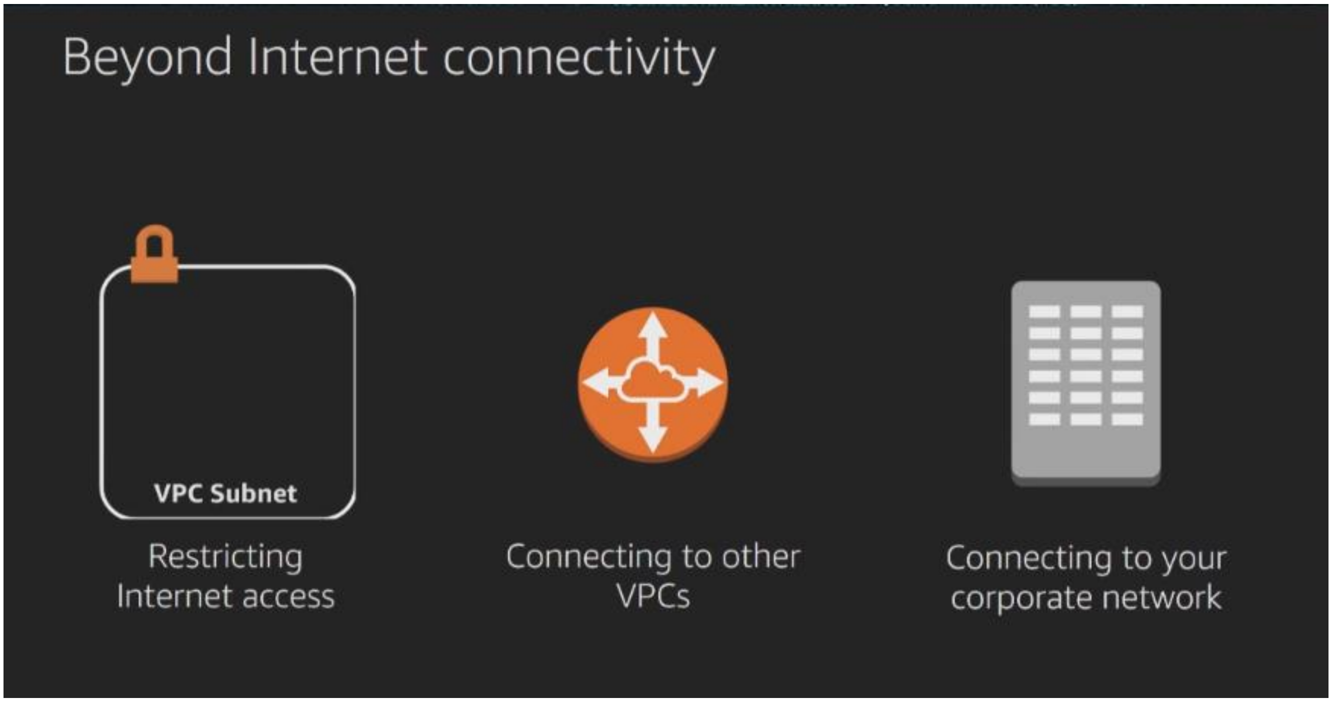
Security groups in VPC: Additional notes

- Follow the *"principle of least privilege"*
- VPC allows creation of egress as well as ingress security group rules

AWS re:Invent
aws

Ingress is incoming traffic, Egress is for outgoing traffic. We can create rules for outgoing traffic also even though the default is to allow all outgoing traffic.

Connectivity options for VPCs



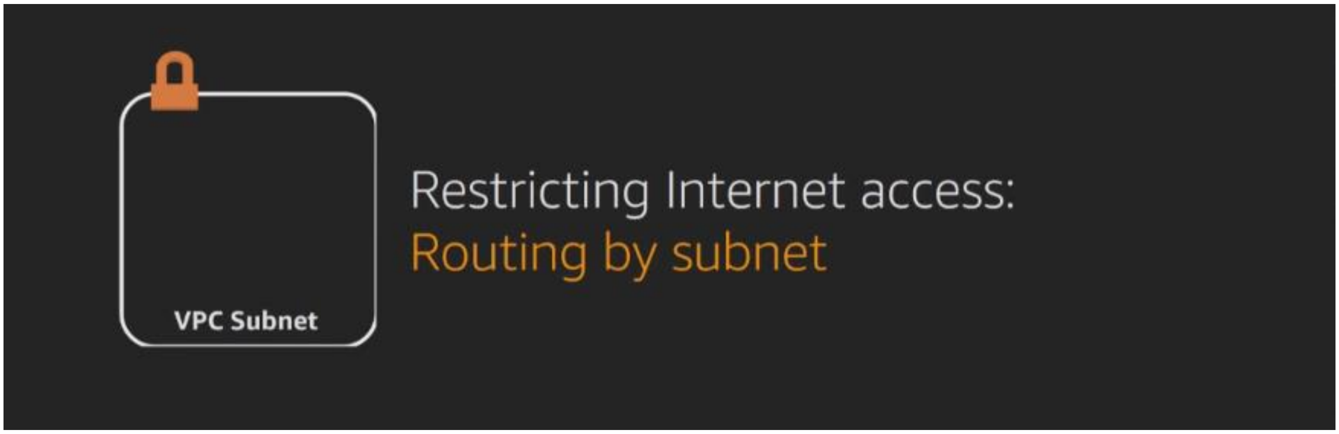
Beyond Internet connectivity

Restricting Internet access

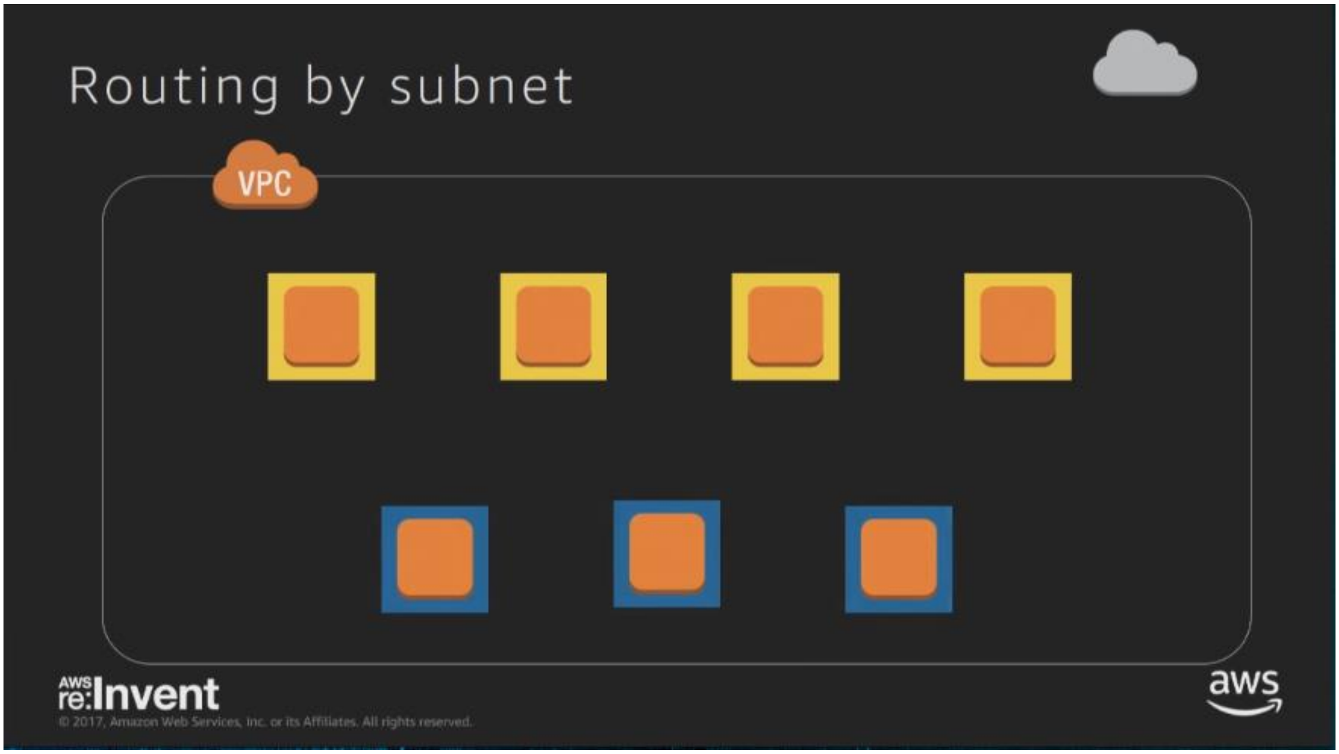
Connecting to other VPCs

Connecting to your corporate network

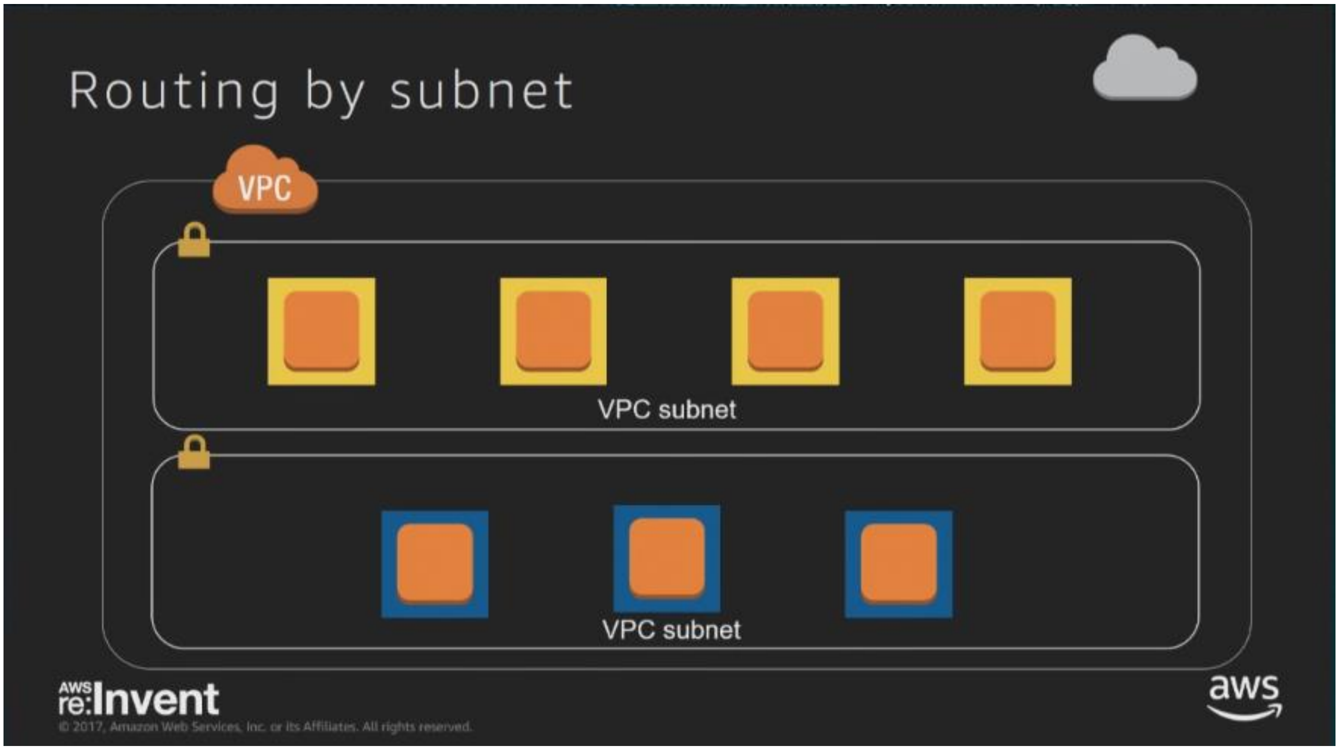
You can restrict access to your VPC, connect your VPCs together using VPC peering, even connect your corporate data center network to your VPC like during migration to the cloud or running a hybrid infrastructure.



Restricting Internet access:
Routing by subnet

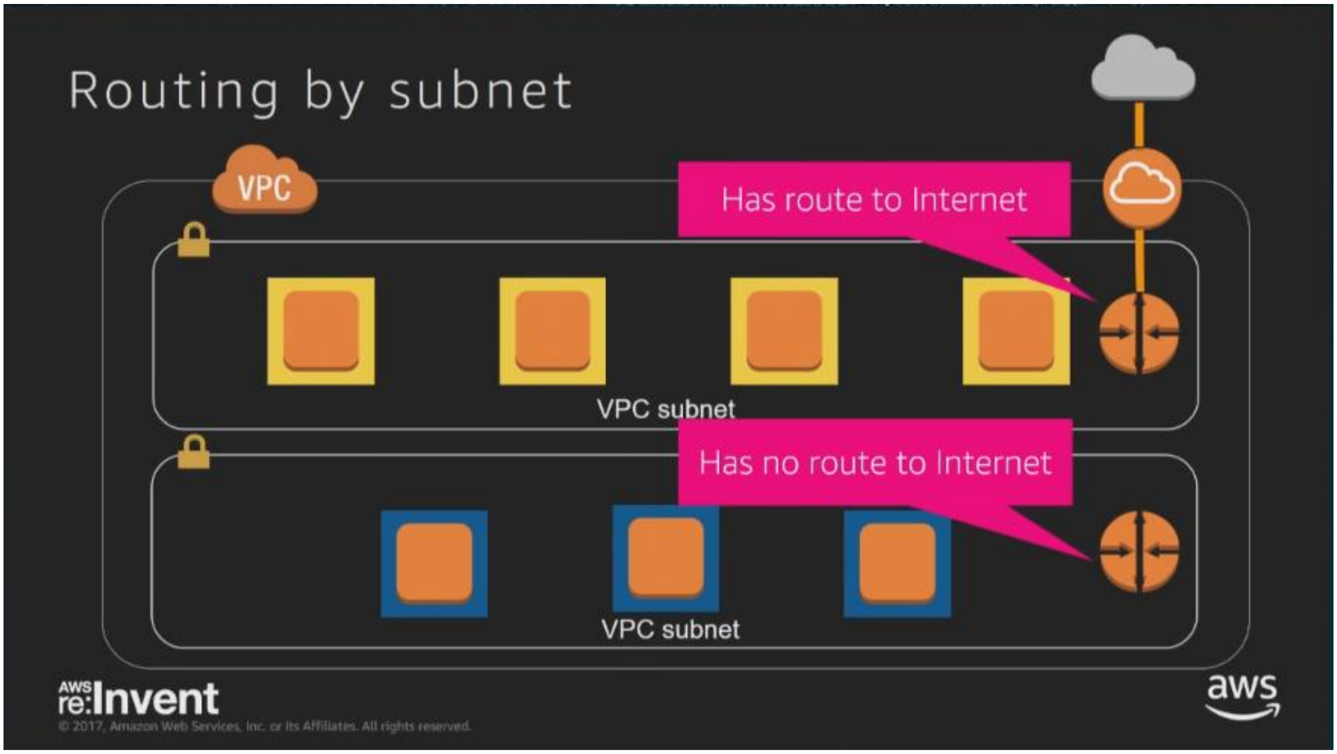


Routing by subnet



Routing by subnet

You can put each group of instances into their own subnet as above, because ***the subnet is the unit that we can do routing on***.

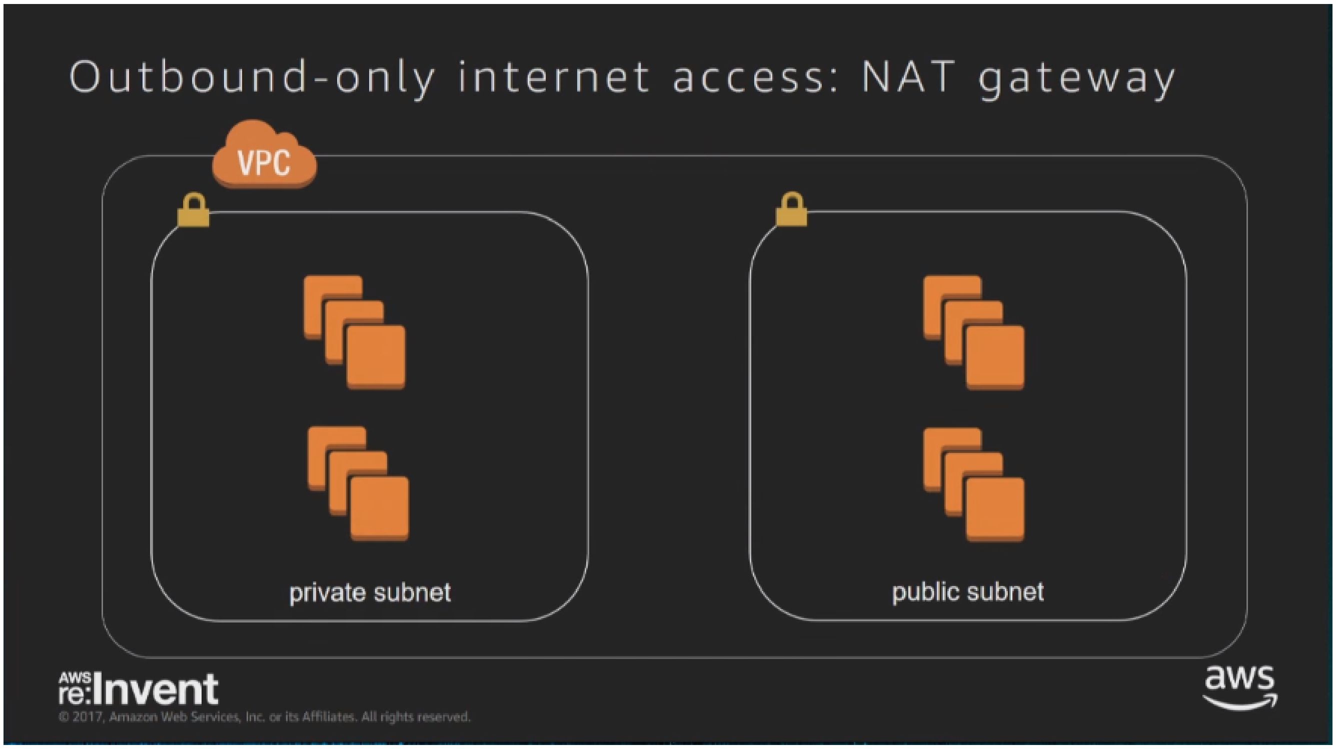


Routing by subnet

Has route to Internet

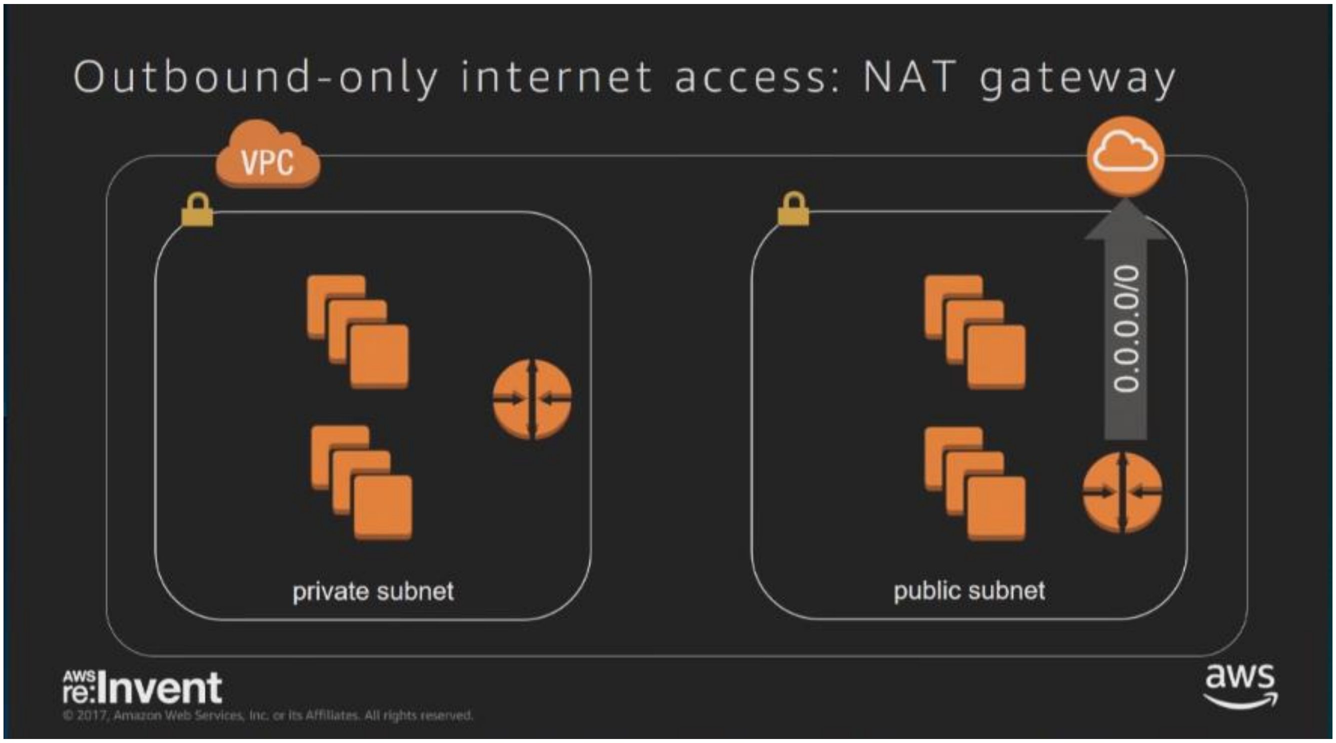
Has no route to Internet

We can then say that the web servers have a route table that has a route to the internet through an IGW, but the backend servers don't have an IGW and cannot reach the internet. This is useful for compliance and audit requirements to show that some servers are not reachable by the internet. ***A subnet that has a route to the internet through an IGW is called a Public Subnet***, while ***a subnet with no route to an IGW to the internet is called a Private Subnet***.



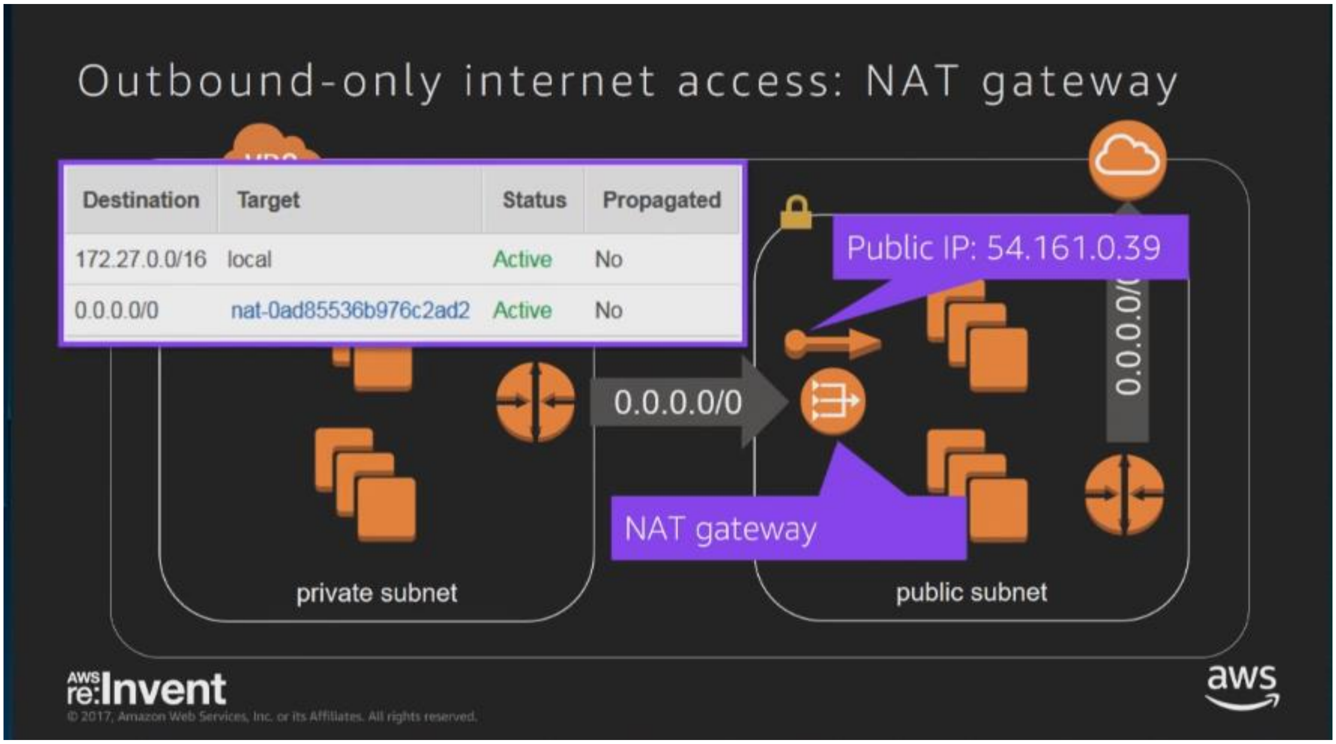
Outbound-only internet access: NAT gateway

An instance in a private subnet cannot get yum repo updates from the internet because it has no route to the internet. You can use ***Network Address Translation*** NAT instance in this case to run a NAT instance in the public subnet to proxy traffic going out to the internet



Outbound-only internet access: NAT gateway

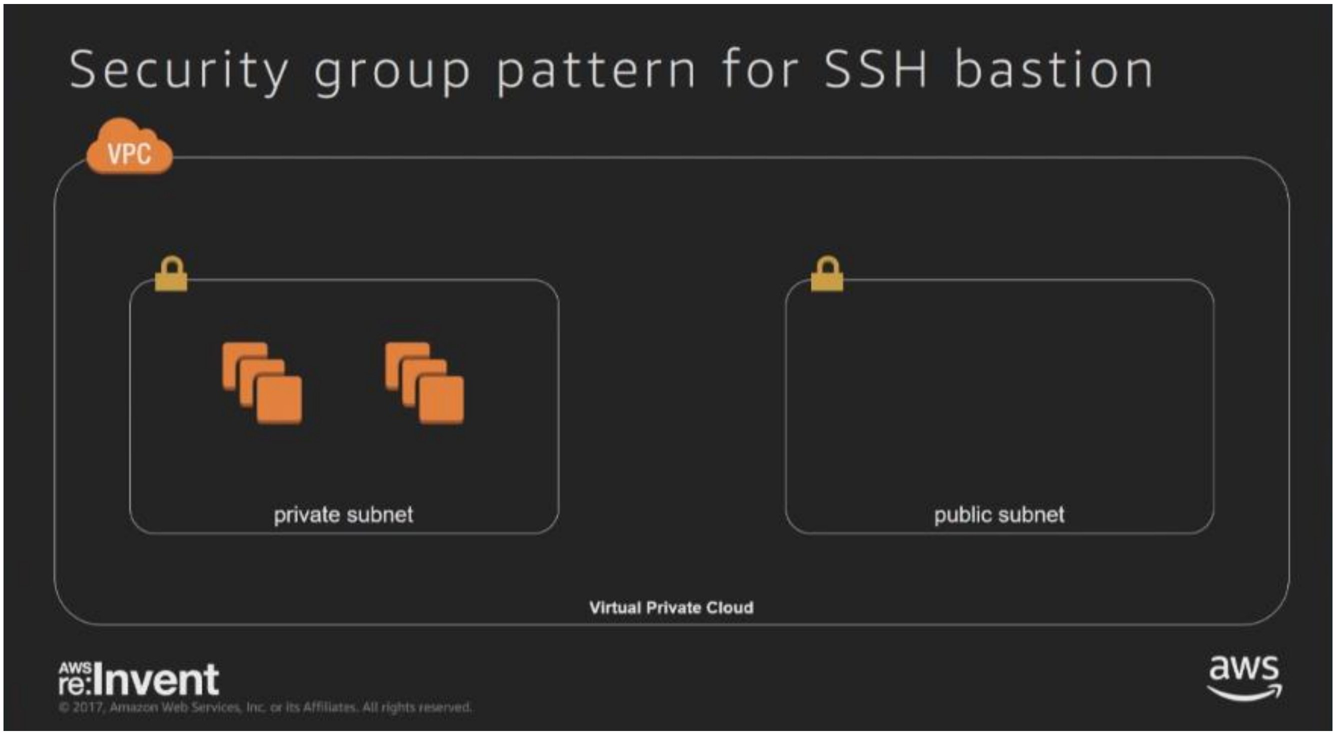
A NAT helps translate IPs on packets out to the internet and bac again to the original calling instance in the private subnet.



Outbound-only internet access: NAT gateway

Destination	Target	Status	Propagated
172.27.0.0/16	local	Active	No
0.0.0.0/0	nat-0ad85536b976c2ad2	Active	No

A NAT gateway is a gateway that you send traffic to when you want that traffic to be NATed and have NAT. you create this in the EC2 console by specifying an Elastic IP address that you control. Then all traffic that you send to that NAT gateway for the internet is going to appear on the internet as if it were coming from the Public IP. It works exactly like the rules approach.



Security group pattern for SSH bastion

VPC

private subnet

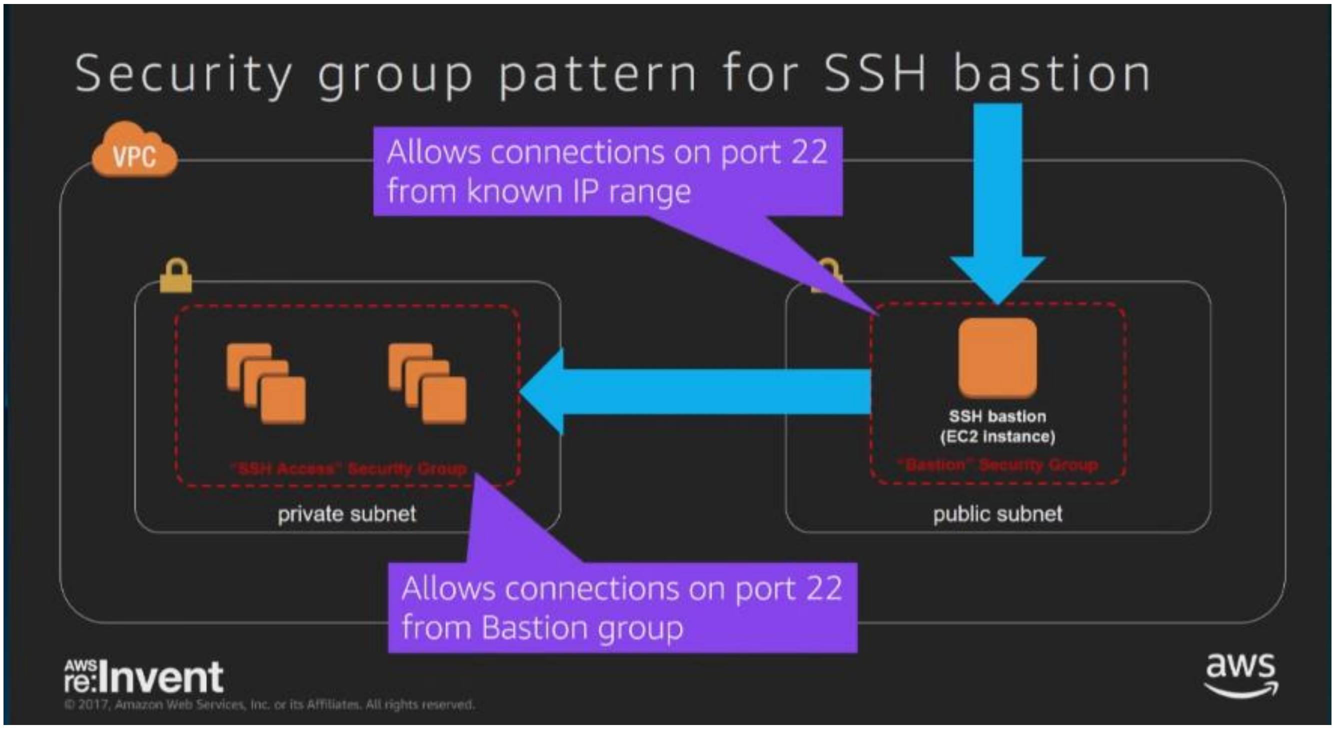
public subnet

Virtual Private Cloud

AWS re:Invent

aws

We want to be able to SSH into the private subnet instances but we can’t get to them since we have no route to the internet.



Security group pattern for SSH bastion

VPC

Allows connections on port 22 from known IP range

SSH bastion (EC2 instance)

"Bastion" Security Group

"SSH Access" Security Group

private subnet

public subnet

Allows connections on port 22 from Bastion group

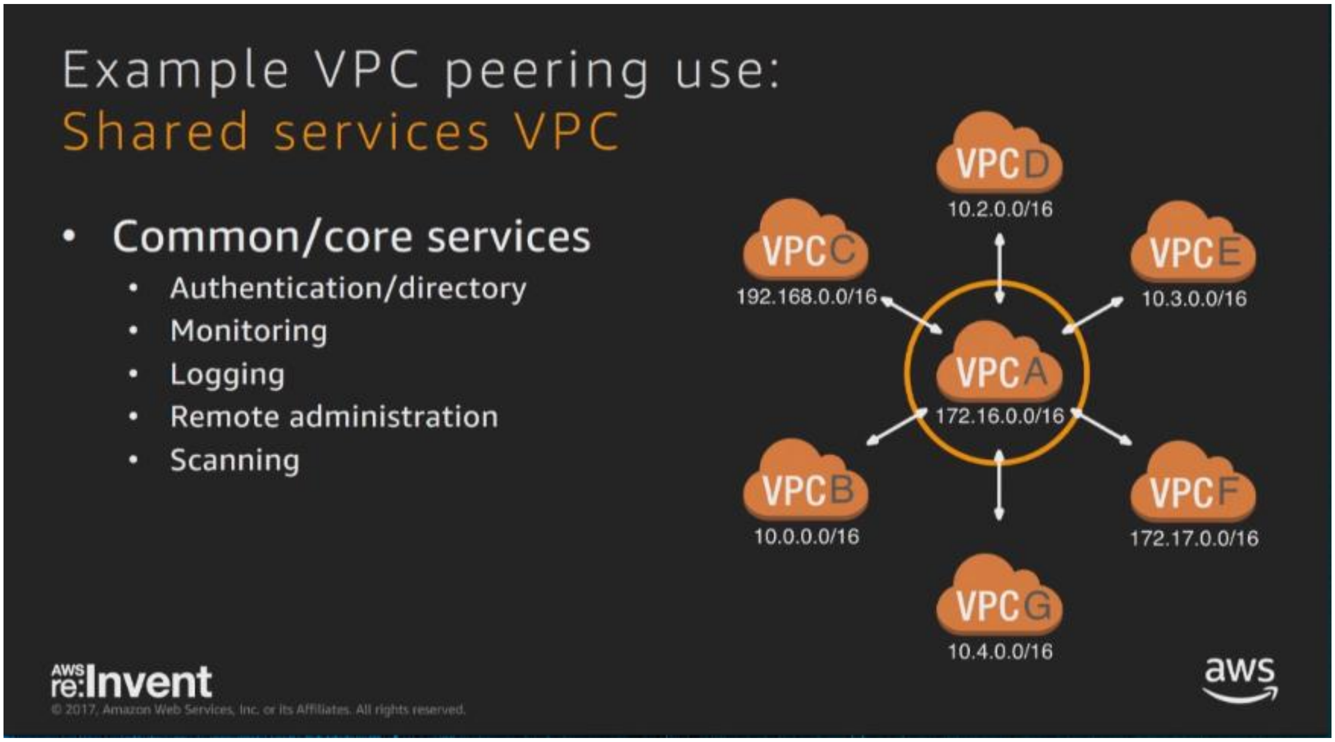
AWS re:Invent

aws

We can create an EC2 instance that is an SSH bastion in the public subnet, this will act like a proxy for our SSH connections. The SSH bastion instance is put in a SG that has only its SSH port 22 opened and only allows incoming traffic from IPs in a known IP address range like IP addresses used within our data center. We can then SSH into instances in our private subnet using the SSH bastion instance.

Inter-VPC connectivity:
VPC peering

For architectures having multiple VPCs or multiple accounts, we can use VPC peering to share resources or services between instances in different VPCs instead of using traffic over the internet.

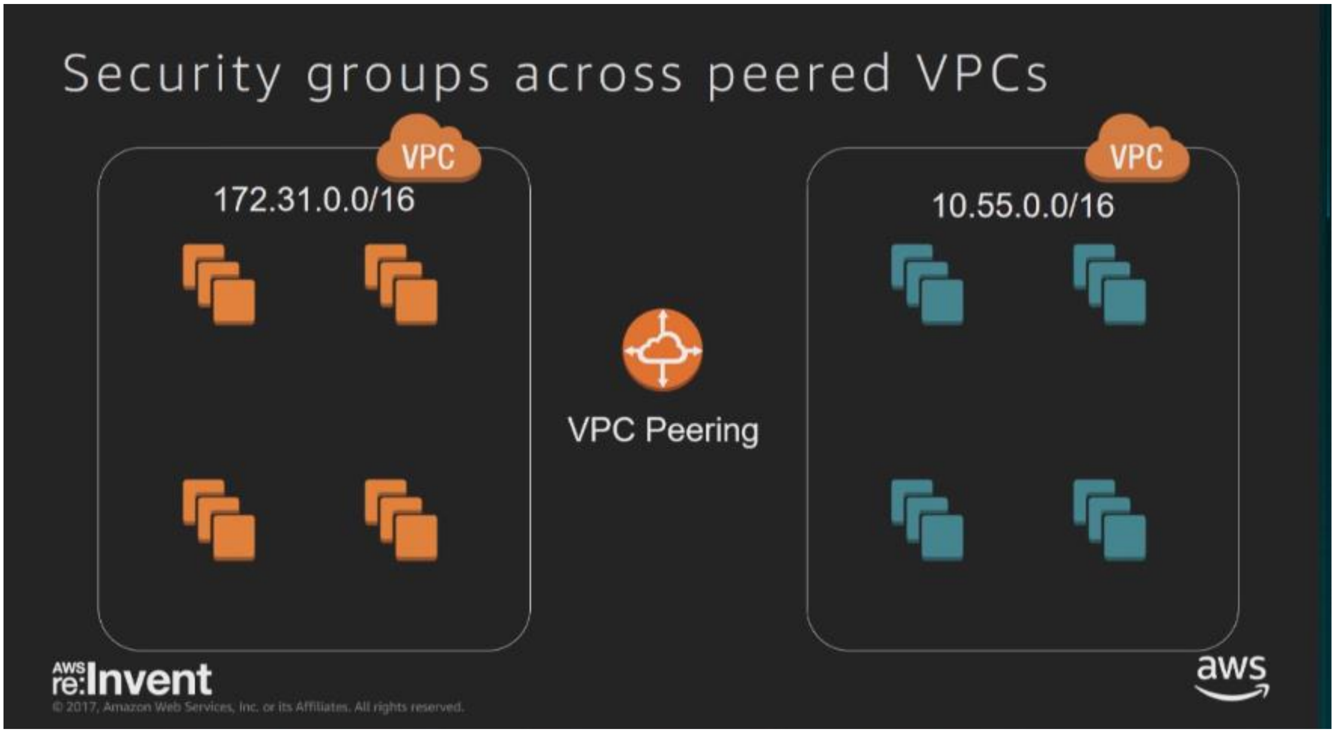


Example VPC peering use:
Shared services VPC

- Common/core services
  - Authentication/directory
  - Monitoring
  - Logging
  - Remote administration
  - Scanning

AWS re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
aws

VPC Peering gives you full private non-IP connectivity between 2 non-overlapping VPCs*. *This allows you to build a spoke-and-wheel architecture like above.



Security groups across peered VPCs

VPC

172.31.0.0/16

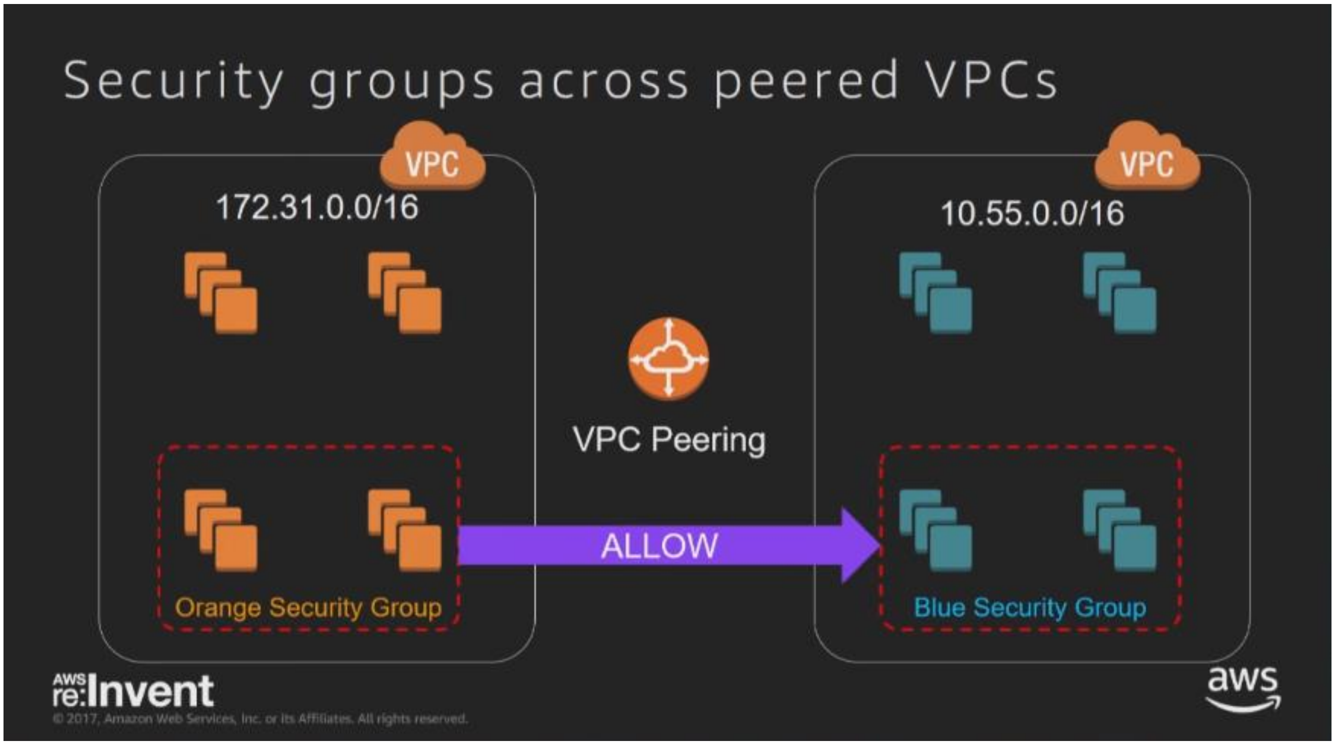
VPC

10.55.0.0/16

VPC Peering

AWS re:Invent

aws



Security groups across peered VPCs

VPC

172.31.0.0/16

VPC

10.55.0.0/16

VPC Peering

ALLOW

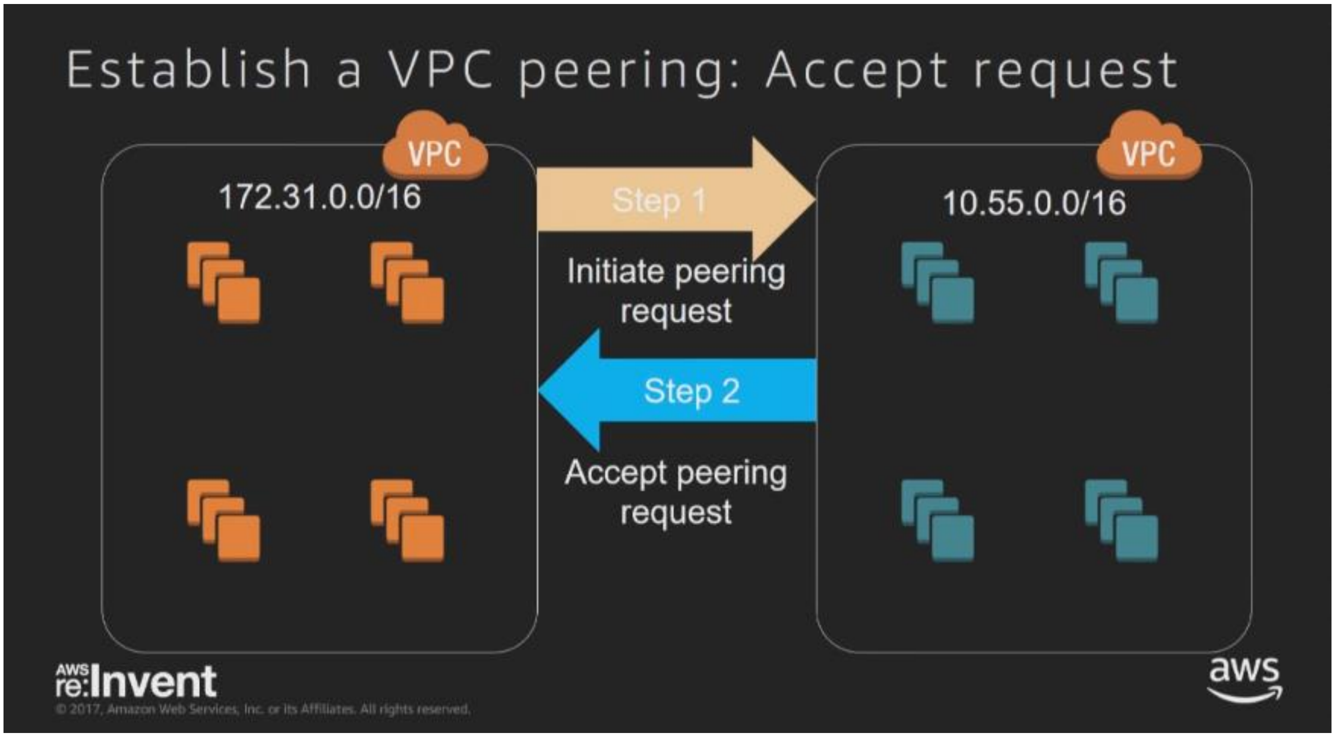
Orange Security Group

Blue Security Group

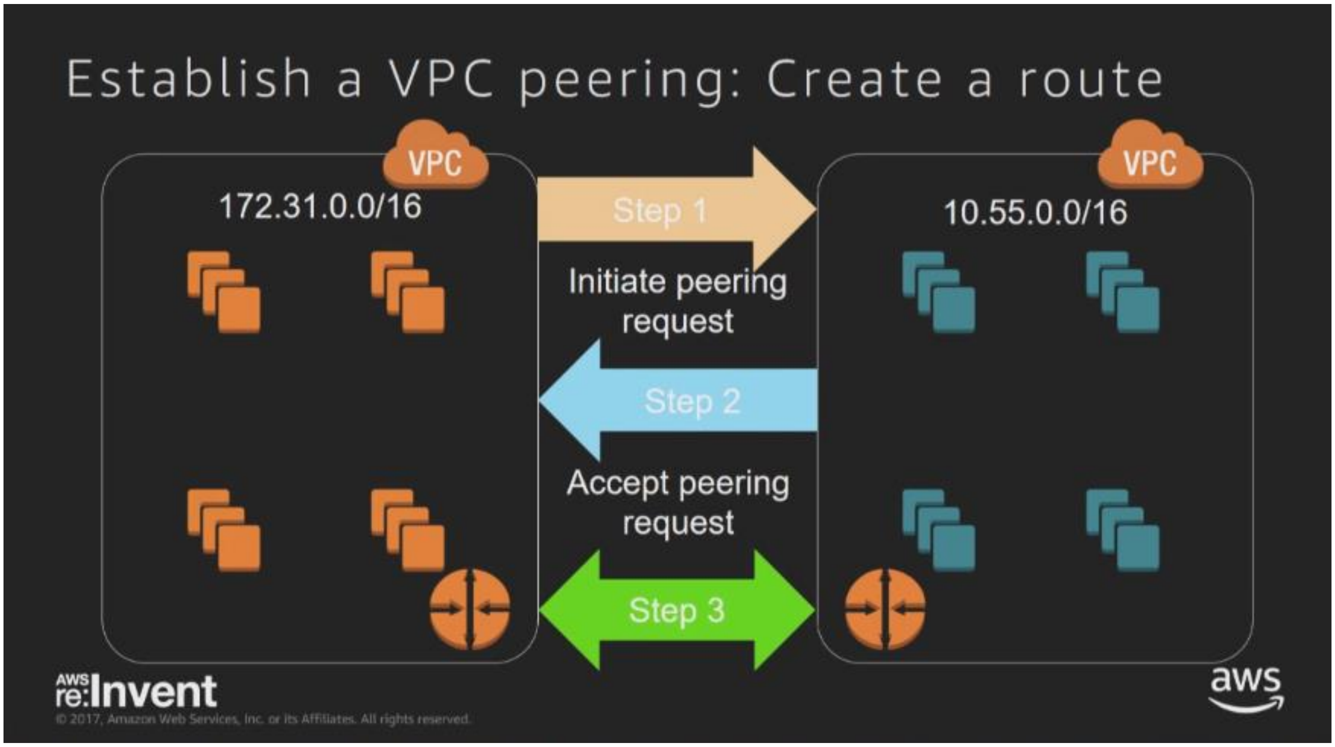
AWS re:Invent

aws

Your SGs still work exactly the same way

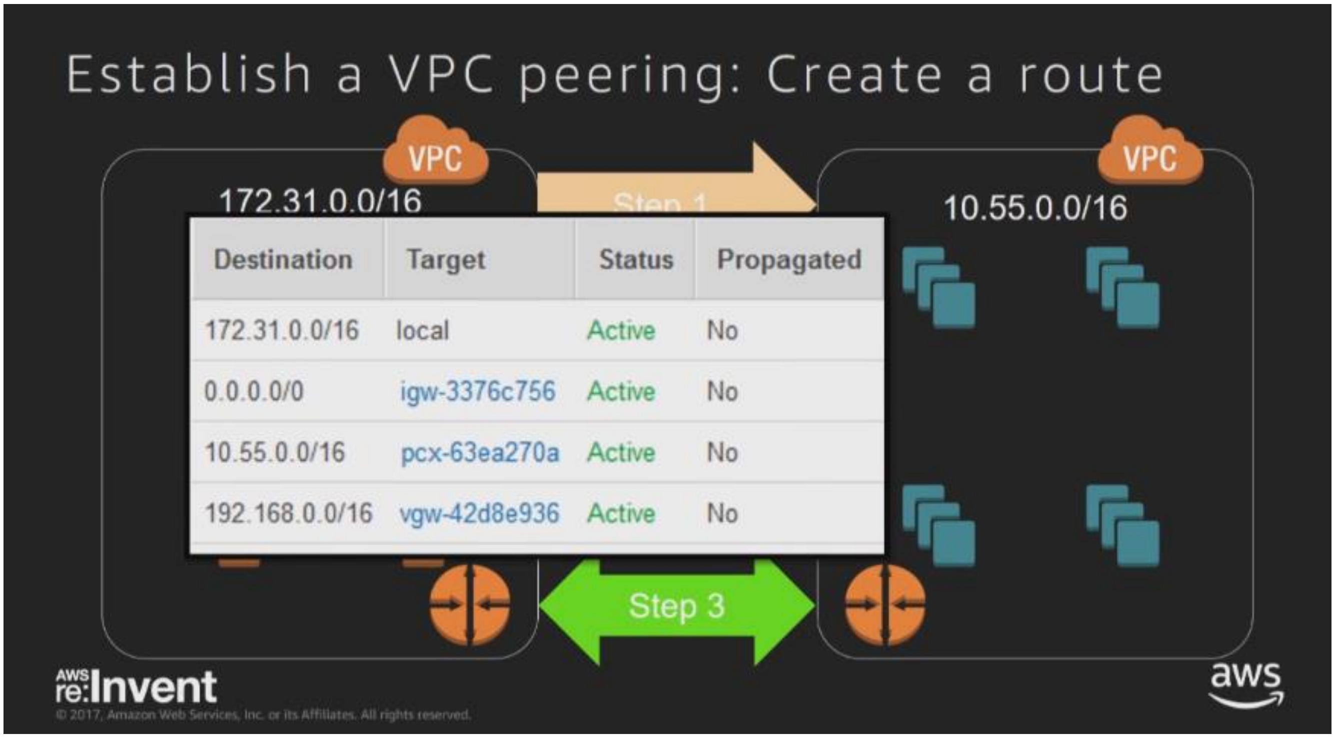


Establish a VPC peering: Accept request



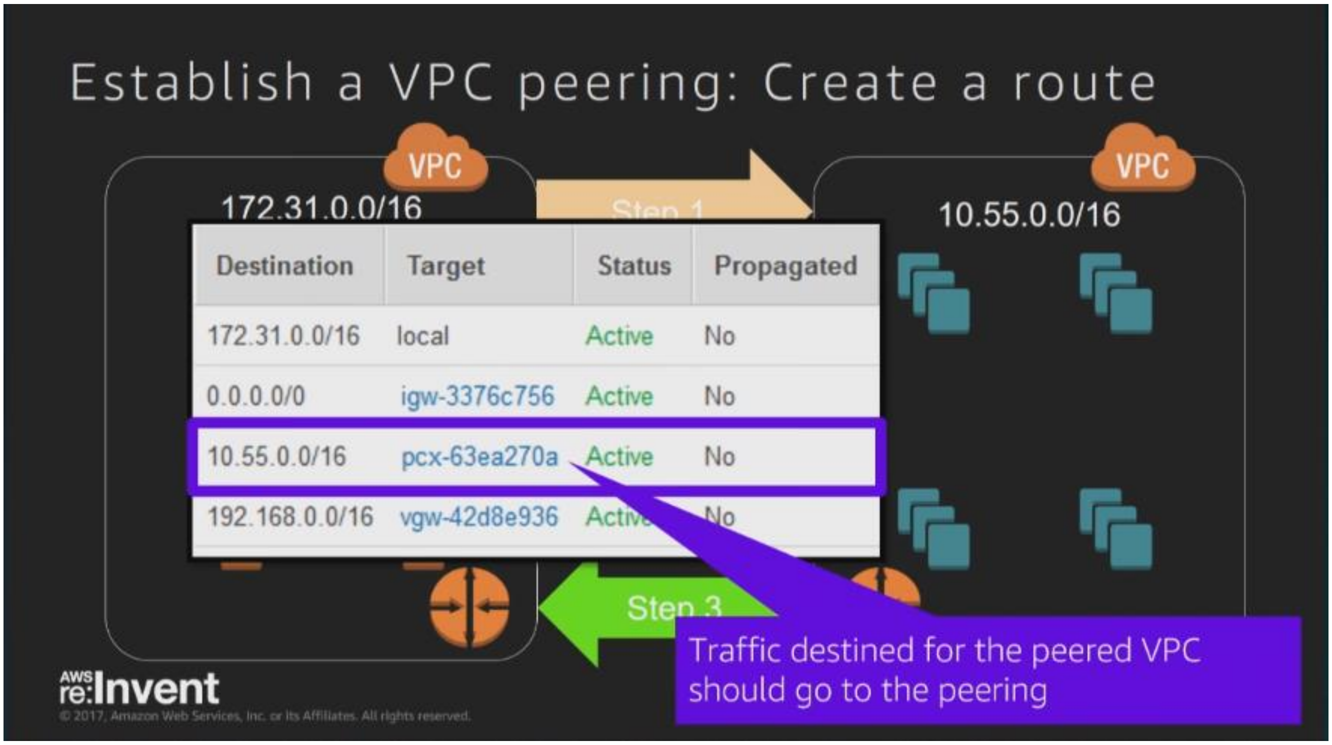
Establish a VPC peering: Create a route

Then you have a peering connection but no traffic is going to be sent yet until you add route rules as below



Establish a VPC peering: Create a route

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
0.0.0.0/0	igw-3376c756	Active	No
10.55.0.0/16	pcx-63ea270a	Active	No
192.168.0.0/16	vgw-42d8e936	Active	No



Establish a VPC peering: Create a route

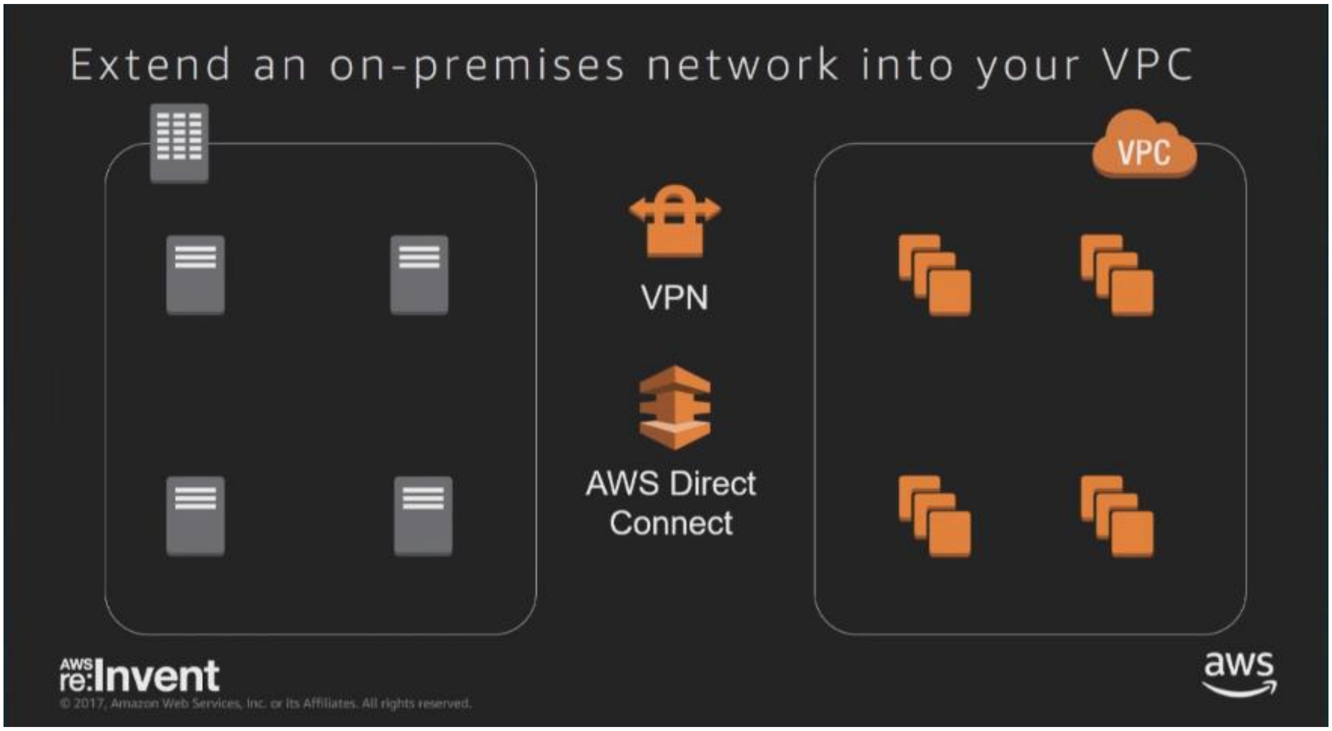
Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
0.0.0.0/0	igw-3376c756	Active	No
10.55.0.0/16	pcx-63ea270a	Active	No
192.168.0.0/16	vgw-42d8e936	Activ	No

Traffic destined for the peered VPC should go to the peering

From one of the VPCs, you send traffic for the peering VPC over to the peering connection. The ***VGW*** is a ***Virtual Private Gateway*** discussed next.

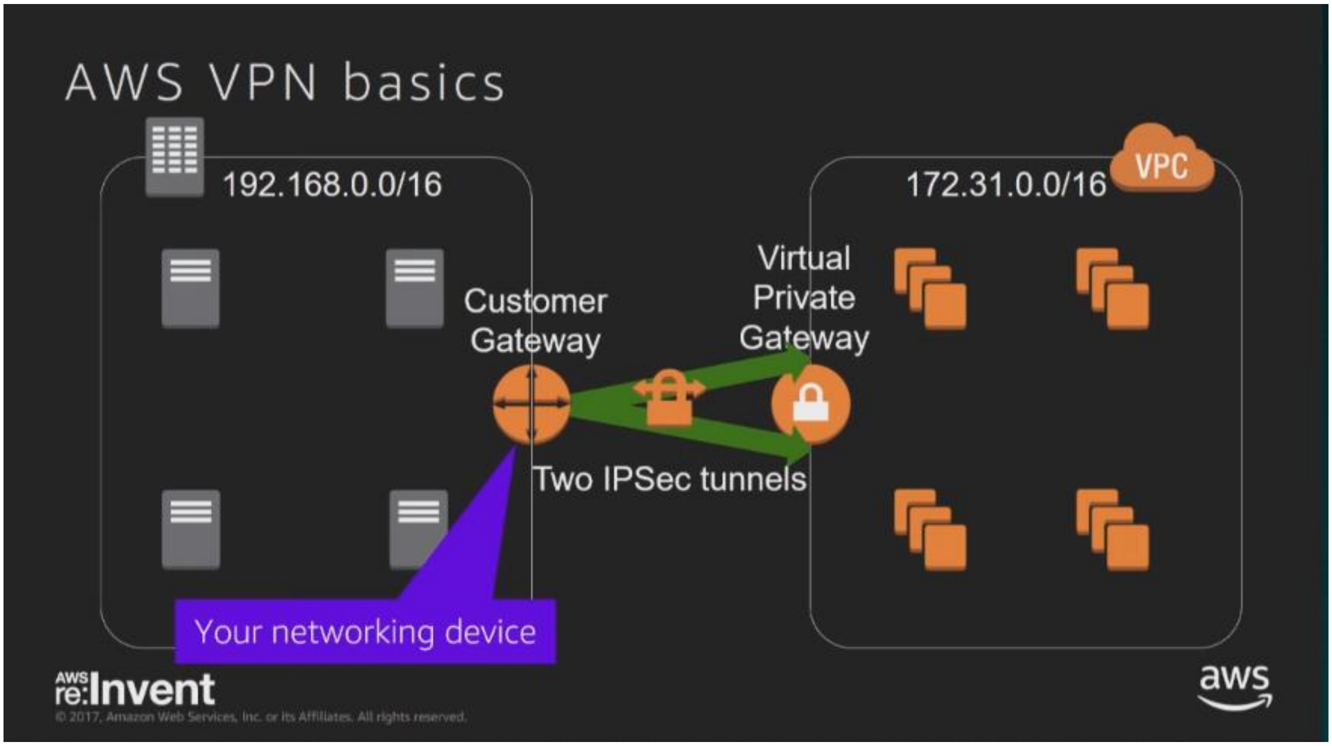
Connecting to on-premises networks:
AWS Virtual Private Network
and AWS Direct Connect

This is good for hybrid networks.

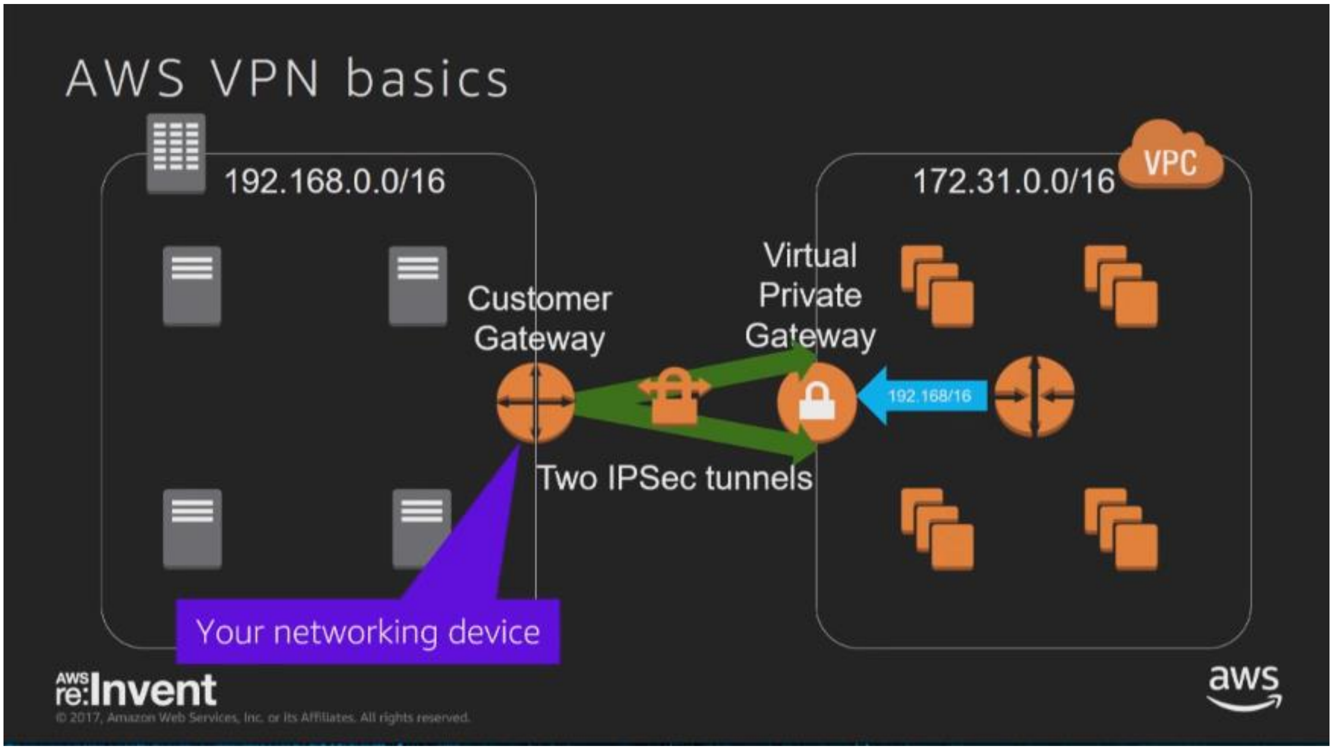


Extend an on-premises network into your VPC

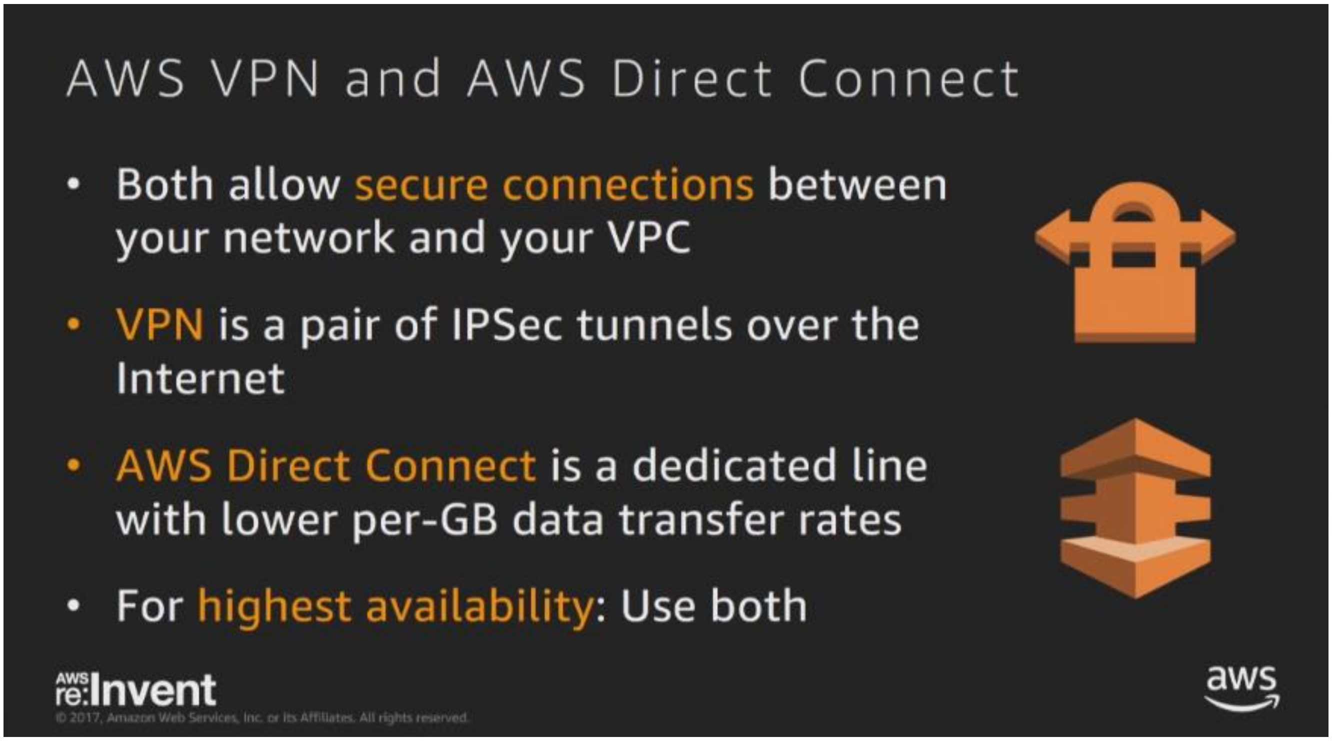
There are 2 technologies that we can use to connect your EC2 resources to other resources running in your own data center DC or corporate networks, ***VPC*** and ***AWS DirectConnect***.



In your own ***DC*** you will need to configure one of your own devices like a ***router*** and use instructions to set this up as your ***Customer Gateway*** and give AWS the statically routable address of the Customer Gateway device. Then on the VPC side, you create your Virtual Private Gateway VGW and you are going to get 2 IPSec tunnels so that you can send encrypted traffic from your on-premises DC to your VPC over the internet.



You need to configure the IPSec tunnels further because they terminate in 2 separate AZs that allow you to take advantage of the AWS global infrastructure. The last thing to do is to add a route to your route table to route traffic that looks destined for your on-premises network to your VGW to forward via the IPSec tunnels.



AWS VPN and AWS Direct Connect

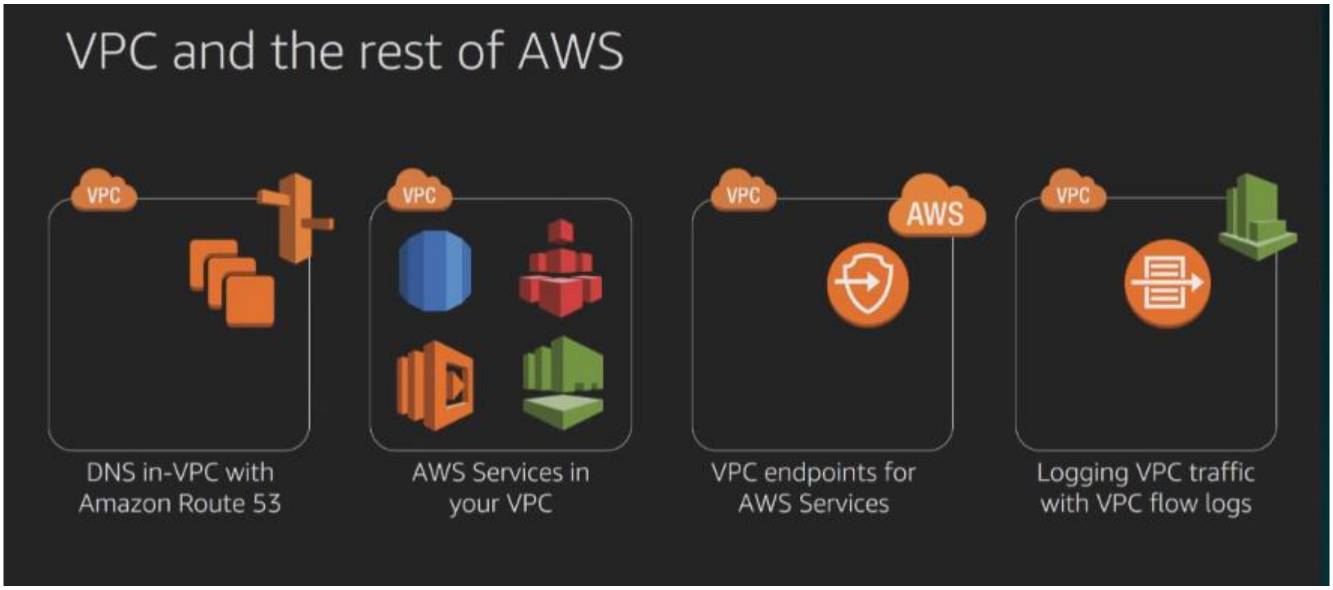
- Both allow secure connections between your network and your VPC
- VPN is a pair of IPSec tunnels over the Internet
- AWS Direct Connect is a dedicated line with lower per-GB data transfer rates
- For highest availability: Use both

AWS re:Invent

aws

AWS DirectConnect is literally plugging a cable from your equipment into AWS equipment at one of the available collocation centers around the world for very high data transfer needs.

VPC and the rest of AWS



VPC and the rest of AWS

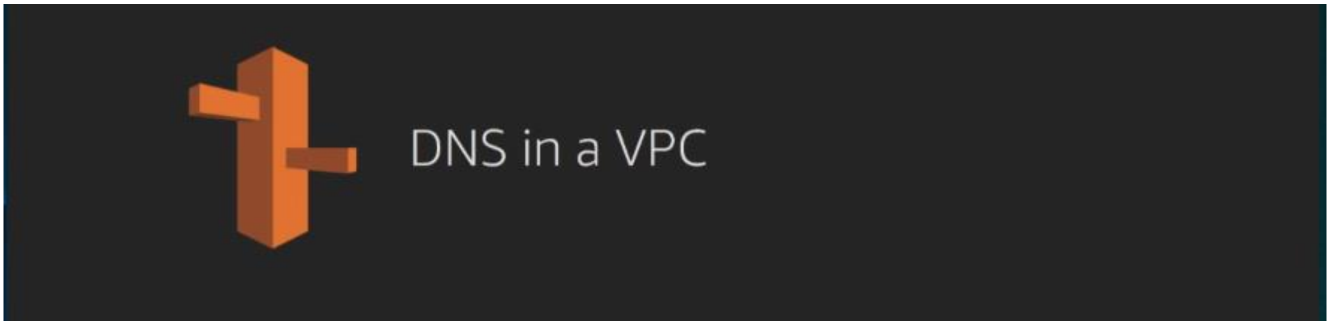
DNS in-VPC with Amazon Route 53

AWS Services in your VPC

VPC endpoints for AWS Services

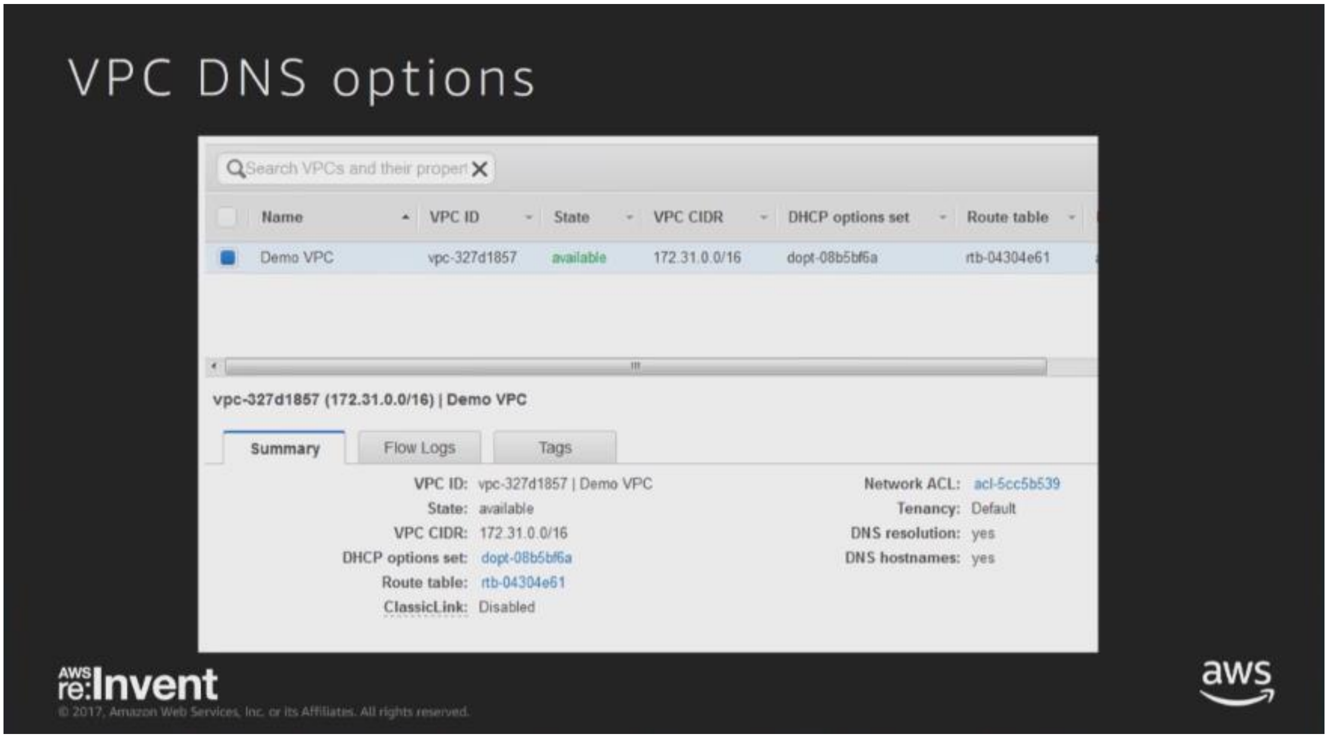
Logging VPC traffic with VPC flow logs

You can do some interesting things within your VPC with DNS.



DNS in a VPC

DNS is a very basic service of a network, DNS allows you to do domain name resolution. In VPC, you can do splitterizing DNS



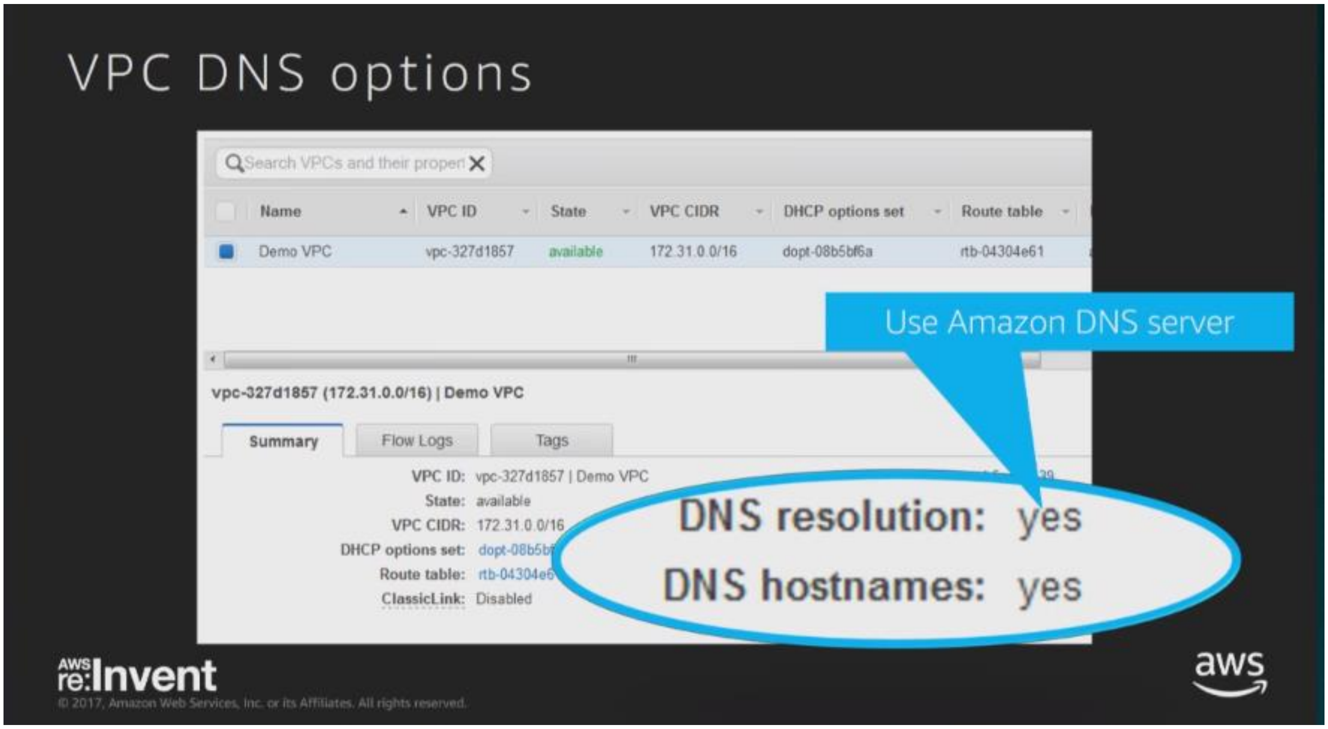
VPC DNS options

Name	VPC ID	State	VPC CIDR	DHCP options set	Route table
Demo VPC	vpc-327d1857	available	172.31.0.0/16	dopt-08b5bf6a	rtb-04304e61

vpc-327d1857 (172.31.0.0/16) | Demo VPC

VPC ID: vpc-327d1857 | Demo VPC
State: available
VPC CIDR: 172.31.0.0/16
DHCP options set: dopt-08b5bf6a
Route table: rtb-04304e61
ClassicLink: Disabled

Network ACL: acl-5cc5b539
Tenancy: Default
DNS resolution: yes
DNS hostnames: yes



VPC DNS options

Name	VPC ID	State	VPC CIDR	DHCP options set	Route table
Demo VPC	vpc-327d1857	available	172.31.0.0/16	dopt-08b5bf6a	rtb-04304e61

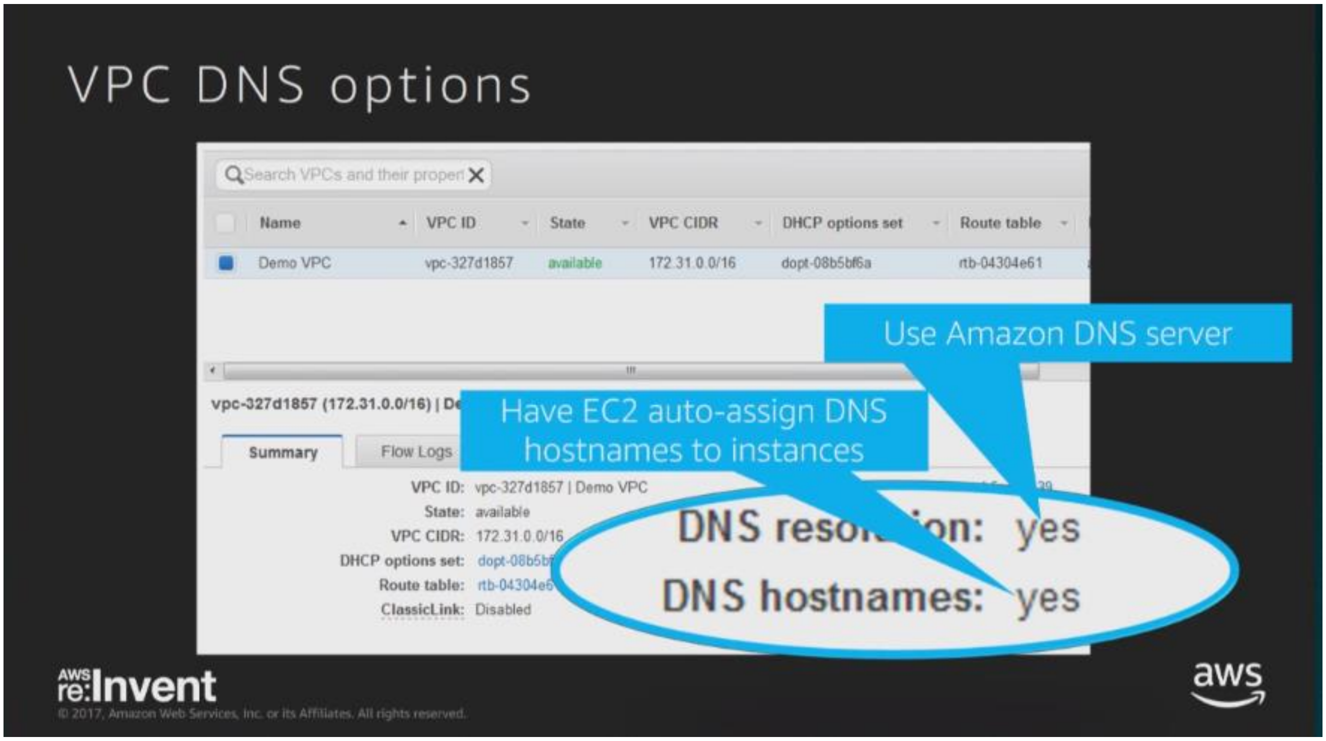
vpc-327d1857 (172.31.0.0/16) | Demo VPC

VPC ID: vpc-327d1857 | Demo VPC
State: available
VPC CIDR: 172.31.0.0/16
DHCP options set: dopt-08b5bf
Route table: rtb-04304e6
ClassicLink: Disabled

Use Amazon DNS server

DNS resolution: yes
DNS hostnames: yes

There are 2 DNS options available in the VPC console and you want to say YES to both options because it gives you some nice features. ***DNS resolution*** option is if you want Amazon to take care of DNS for you instead of you having to run your own DNS server.



VPC DNS options

Name	VPC ID	State	VPC CIDR	DHCP options set	Route table
Demo VPC	vpc-327d1857	available	172.31.0.0/16	dopt-08b5bf6a	rtb-04304e61

vpc-327d1857 (172.31.0.0/16) | De

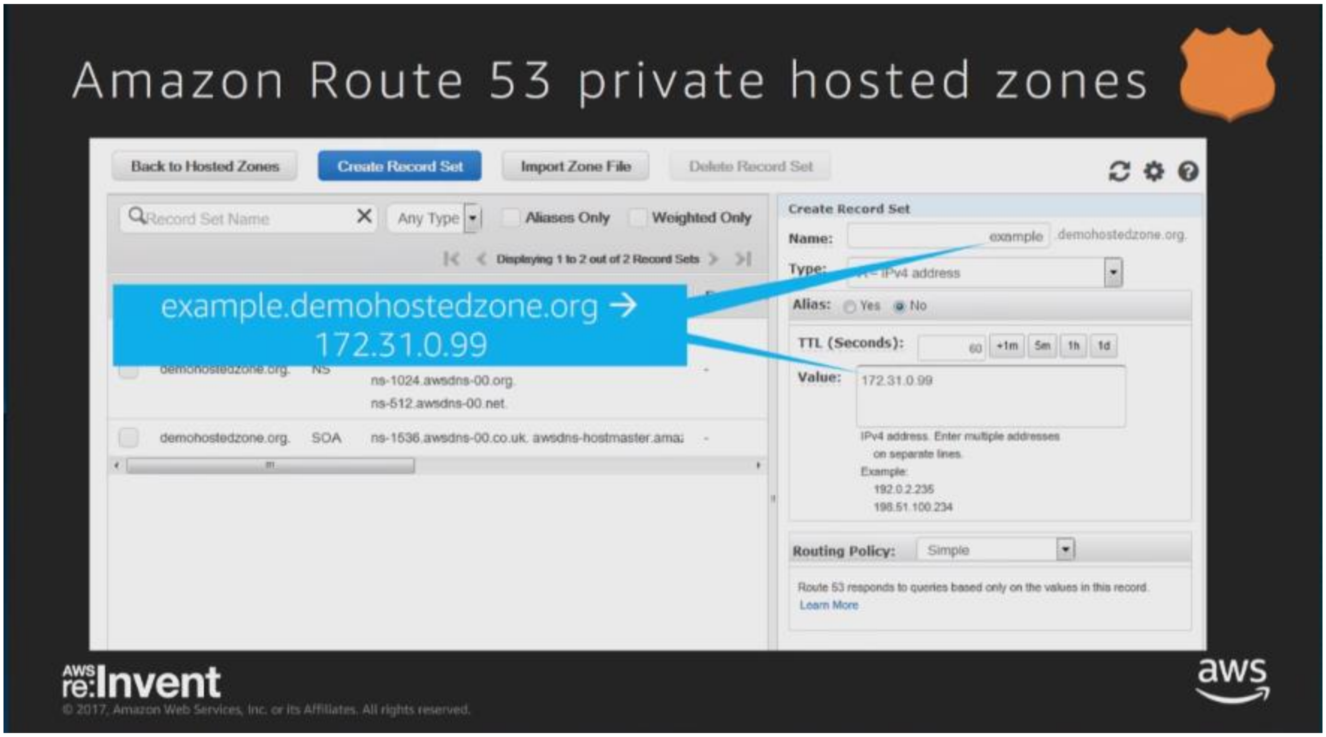
VPC ID: vpc-327d1857 | Demo VPC
State: available
VPC CIDR: 172.31.0.0/16
DHCP options set: dopt-08b5bf
Route table: rtb-04304e6
ClassicLink: Disabled

DNS resolution: yes
DNS hostnames: yes

Use Amazon DNS server

Have EC2 auto-assign DNS hostnames to instances

The ***DNS hostnames*** option means that every time you launch an EC2 instance, you will get a hostname.



Amazon Route 53 private hosted zones

Displaying 1 to 2 out of 2 Record Sets

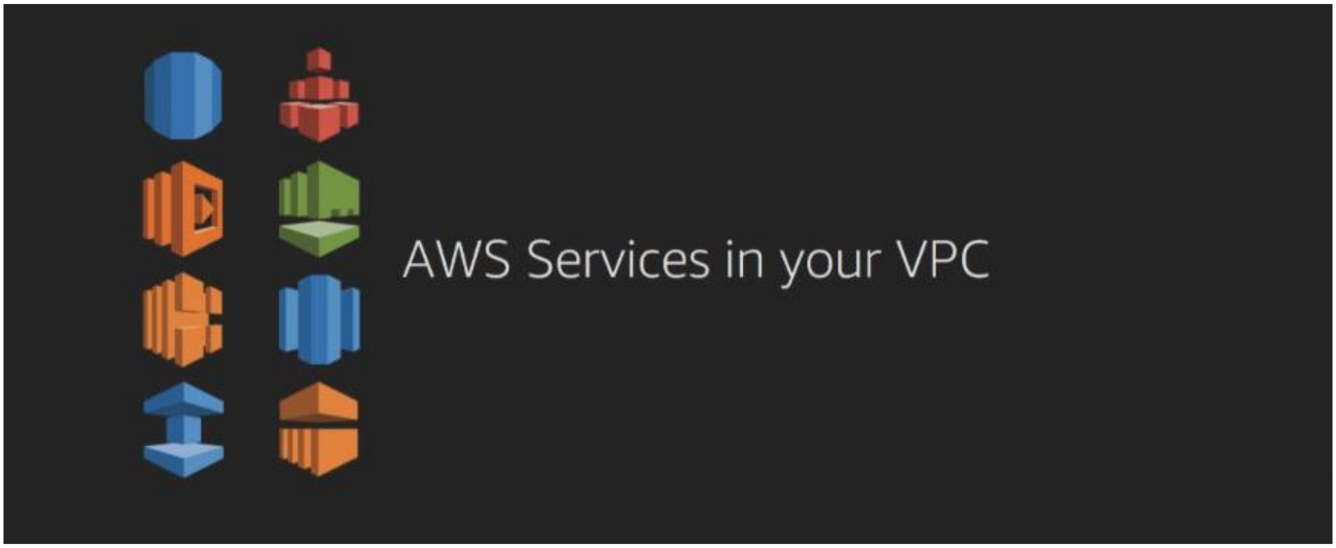
demohostedzone.org.	NS	ns-1024.awsdns-00.org. ns-512.awsdns-00.net.
demohostedzone.org.	SOA	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.ama:

Create Record Set
Name: example .demohostedzone.org.
Type: A – IPv4 address
Alias: ○ Yes ● No
TTL (Seconds): 60
Value: 172.31.0.99
IPv4 address. Enter multiple addresses on separate lines.
Example:
192.0.2.235
198.51.100.234
Routing Policy: Simple
Route 53 responds to queries based only on the values in this record.
Learn More

example.demohostedzone.org → 172.31.0.99

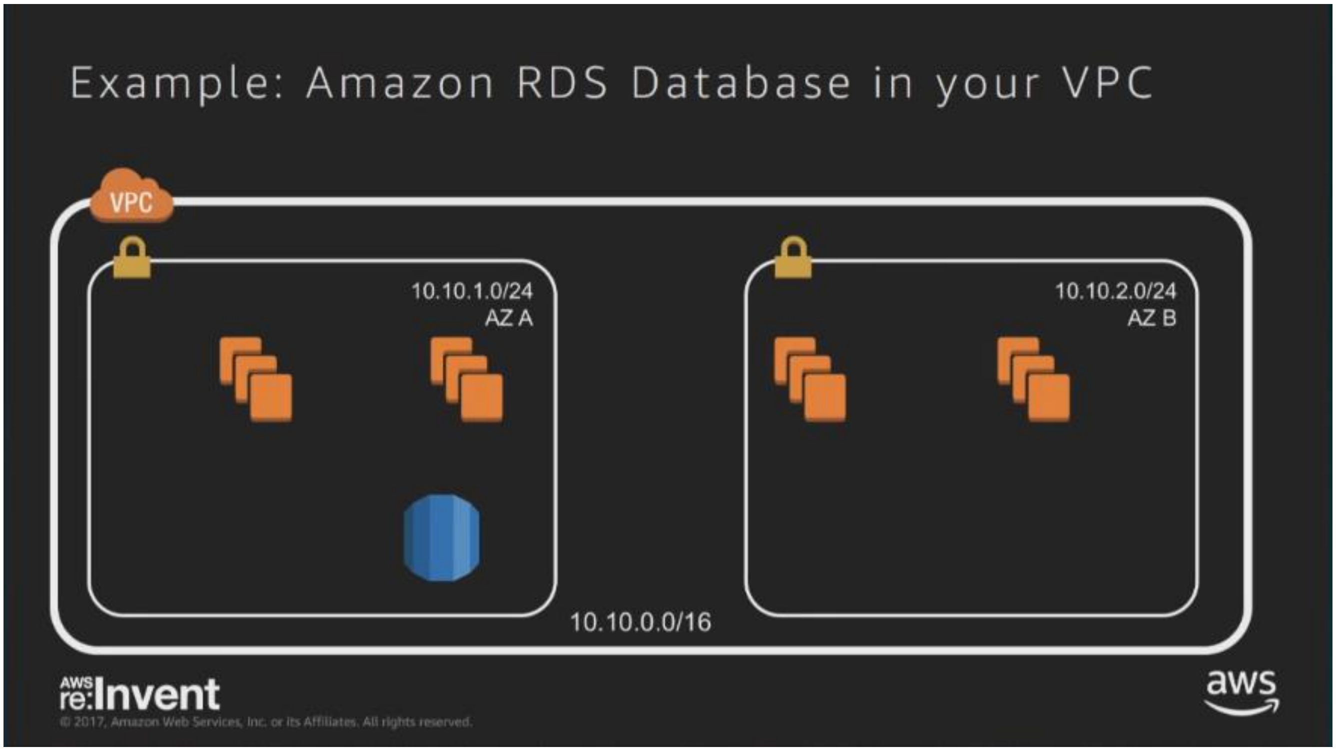
This is a look at Amazon Route53 console, Route53 is Amazon's managed DNS service mostly used for public facing DNS. You can also use a Private Hosted Zone which allows you to take over a zone inside of your VPC. The above shows that ***we have taken over the demohostedzone.org zone even though we do not own that domain name, but within our VPC we can control what DNS does and make demohostedzone.org do whatever we want***. In this case, ***we have pointed***

the domain name example.demohostedzone.org to point to the IP address 172.31.0.99 which is a ***private IP*** within our VPC's Private Subnet.



AWS Services in your VPC

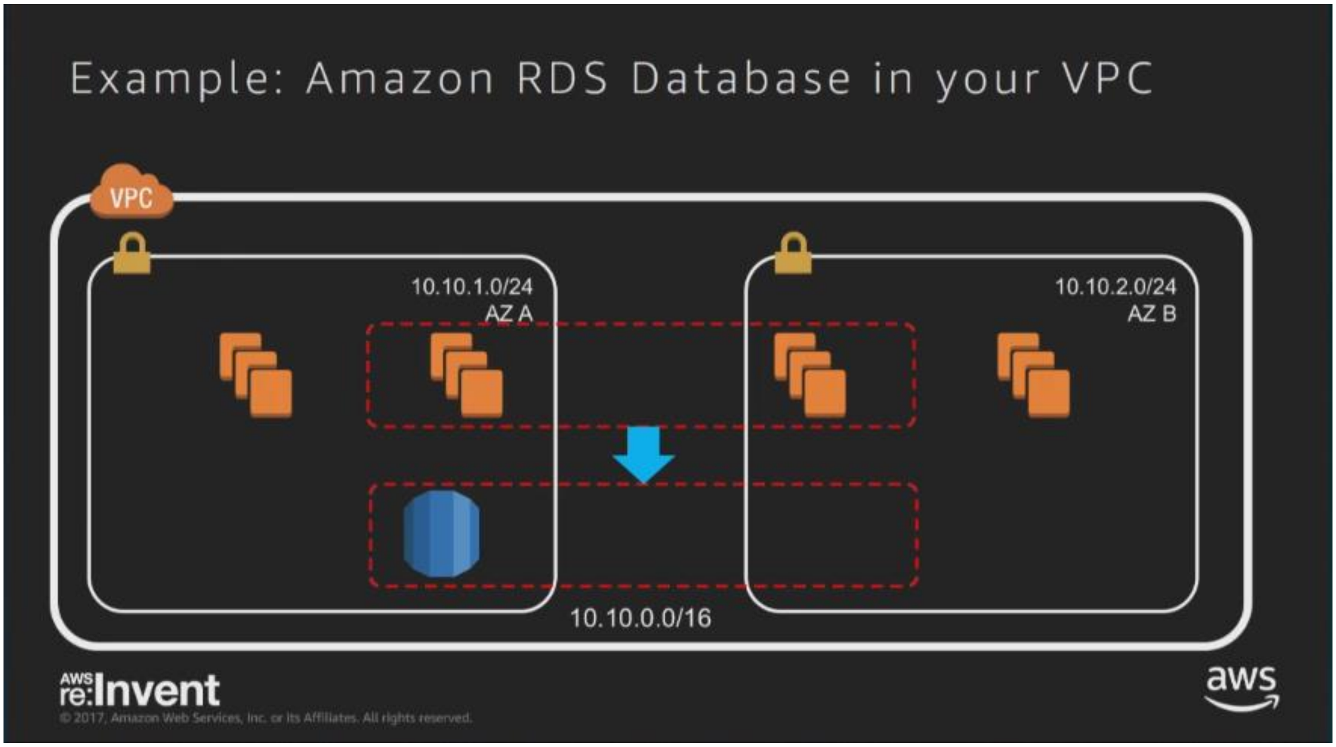
Lots of AWS services are running infrastructure on your behalf, so we can give you the choice to run it inside your own network or VPC. Let us see 2 patterns that get used a lot, the patterns exist to take advantage of the HA of the AWS infrastructure.



Example: Amazon RDS Database in your VPC

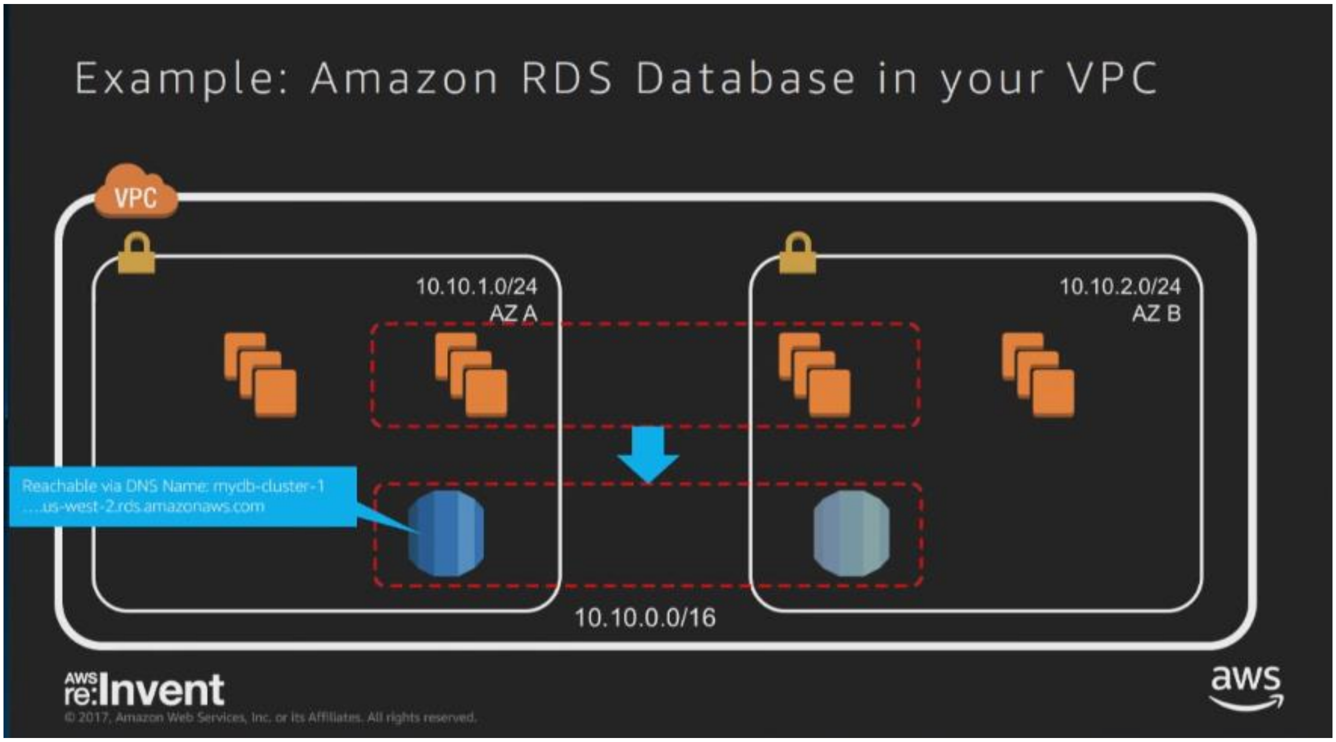
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

When you create an RDS database you are given the option of running the RDS database within your VPC,



Example: Amazon RDS Database in your VPC

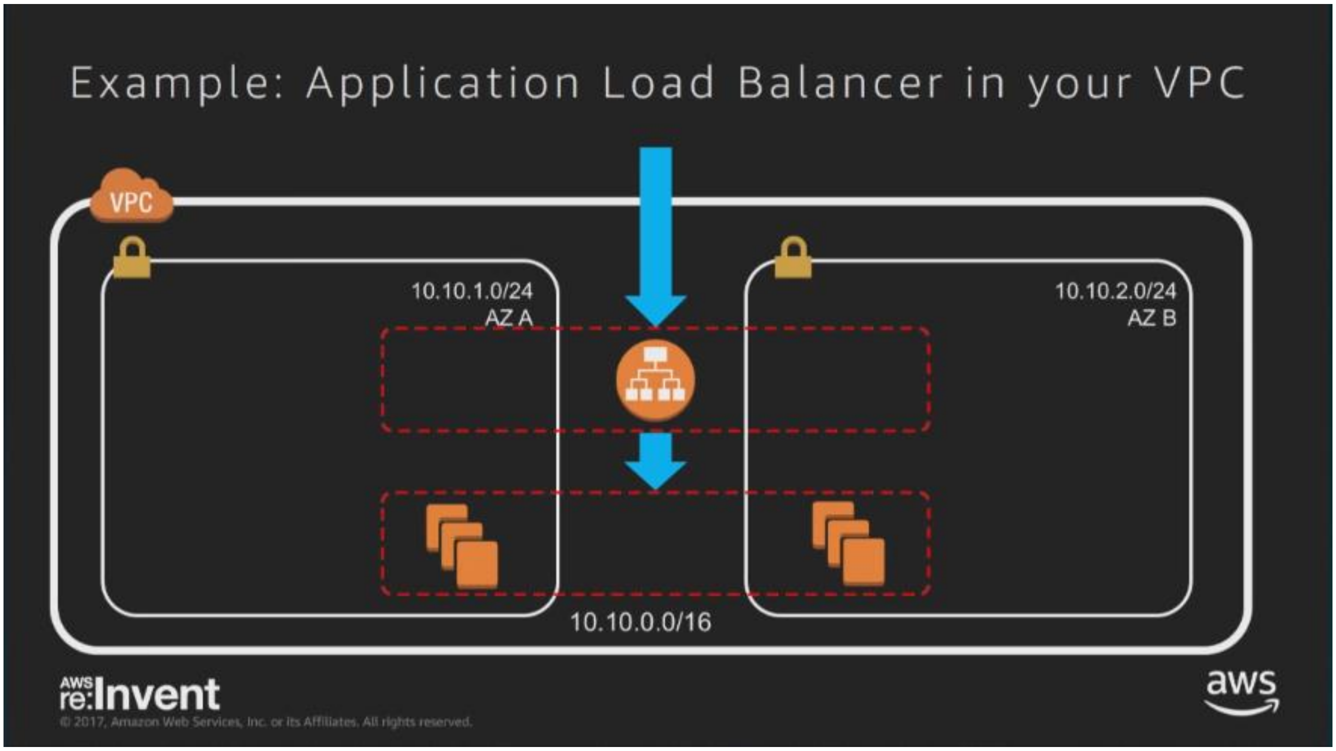
The first thing you are asked about is your SGs and your Subnets for your AZs,



Example: Amazon RDS Database in your VPC

Reachable via DNS Name: mydb-cluster-1 ....us-west-2.rds.amazonaws.com

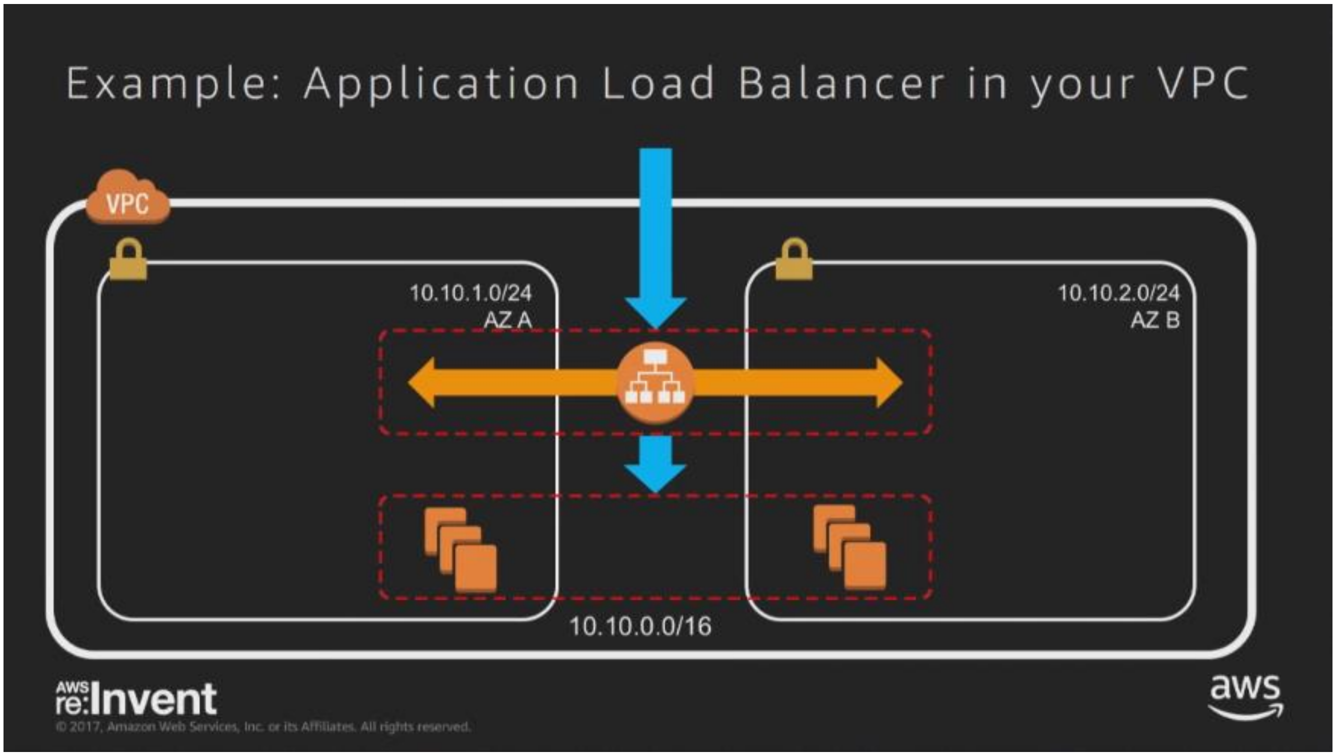
When you specify subnet spread over different AZs, you get a master RDS database in one subnet in an AZ and also get a failover candidate on standby in a 2nd AZ. This is an ***Active-Standby pattern***. You also get a DNS name that always point to the active database.



Example: Application Load Balancer in your VPC

VPC
10.10.1.0/24
AZ A
10.10.2.0/24
AZ B
10.10.0.0/16
AWS re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
aws

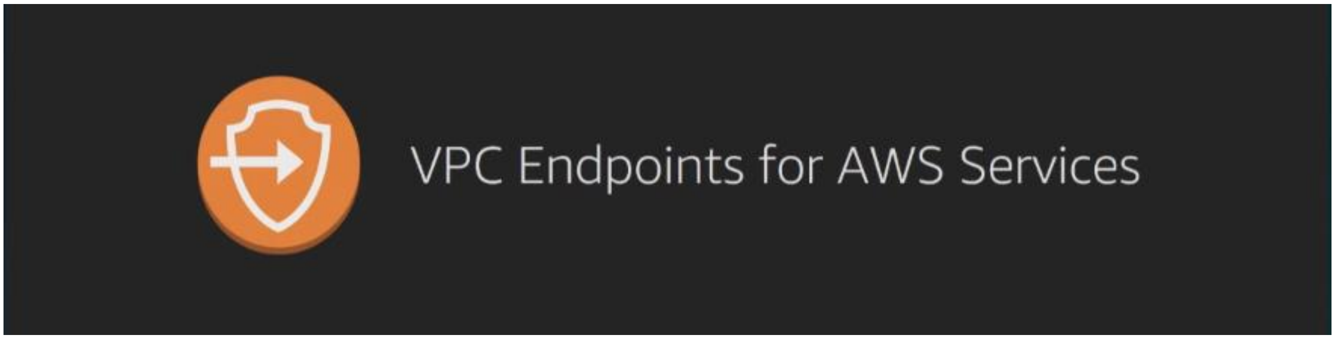
The 2nd example is the ***Application Load Balancer*** ALB, this is a Layer 7 or HTTP load balancer run by ELB.



Example: Application Load Balancer in your VPC

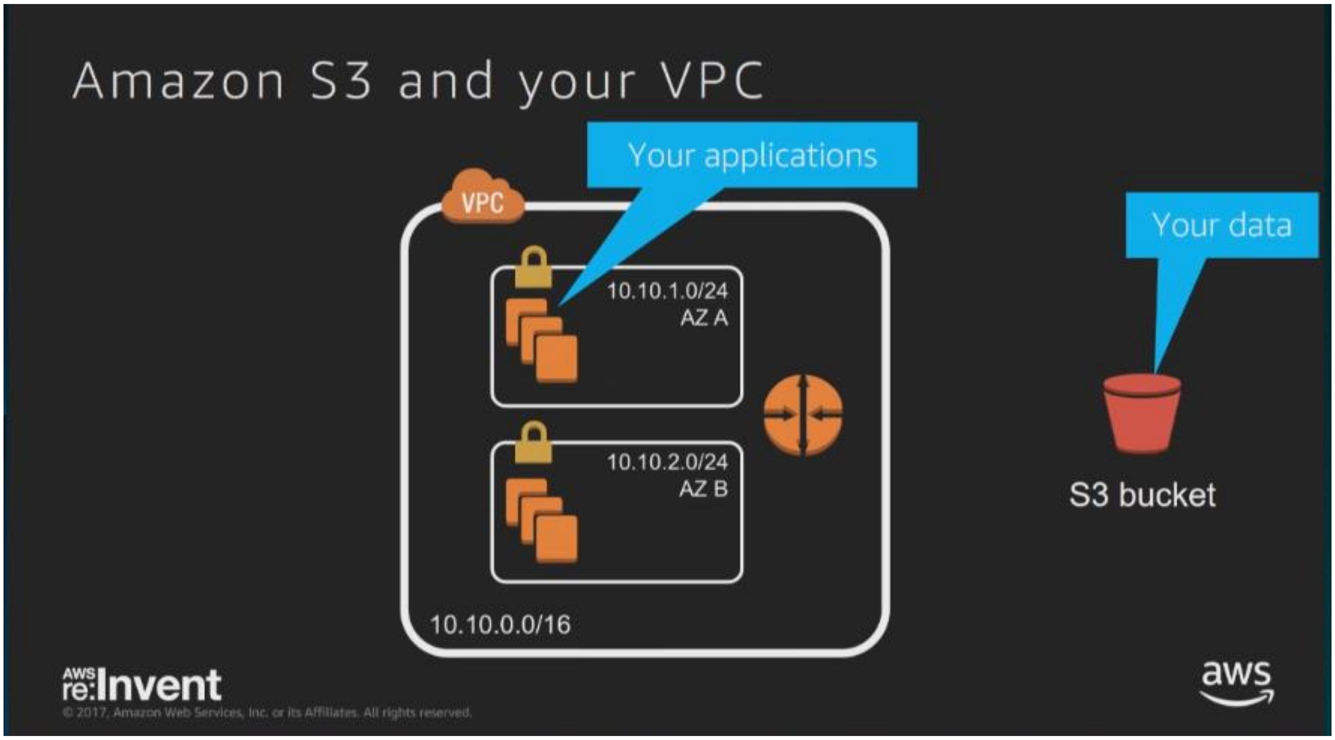
VPC
10.10.1.0/24
AZ A
10.10.2.0/24
AZ B
10.10.0.0/16
AWS re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
aws

You are again asked to specify Subnets that will define your AZs. Your ALB is then going to create active nodes in multiple AZs for constant load balancing and HA.



VPC Endpoints for AWS Services

VPC endpoints give a direct private connection to AWS services from within your VPC, it allows you to continue to practice the principle of least privilege. You also get some nice tools for access management.



Amazon S3 and your VPC

Your applications

VPC

10.10.1.0/24
AZ A

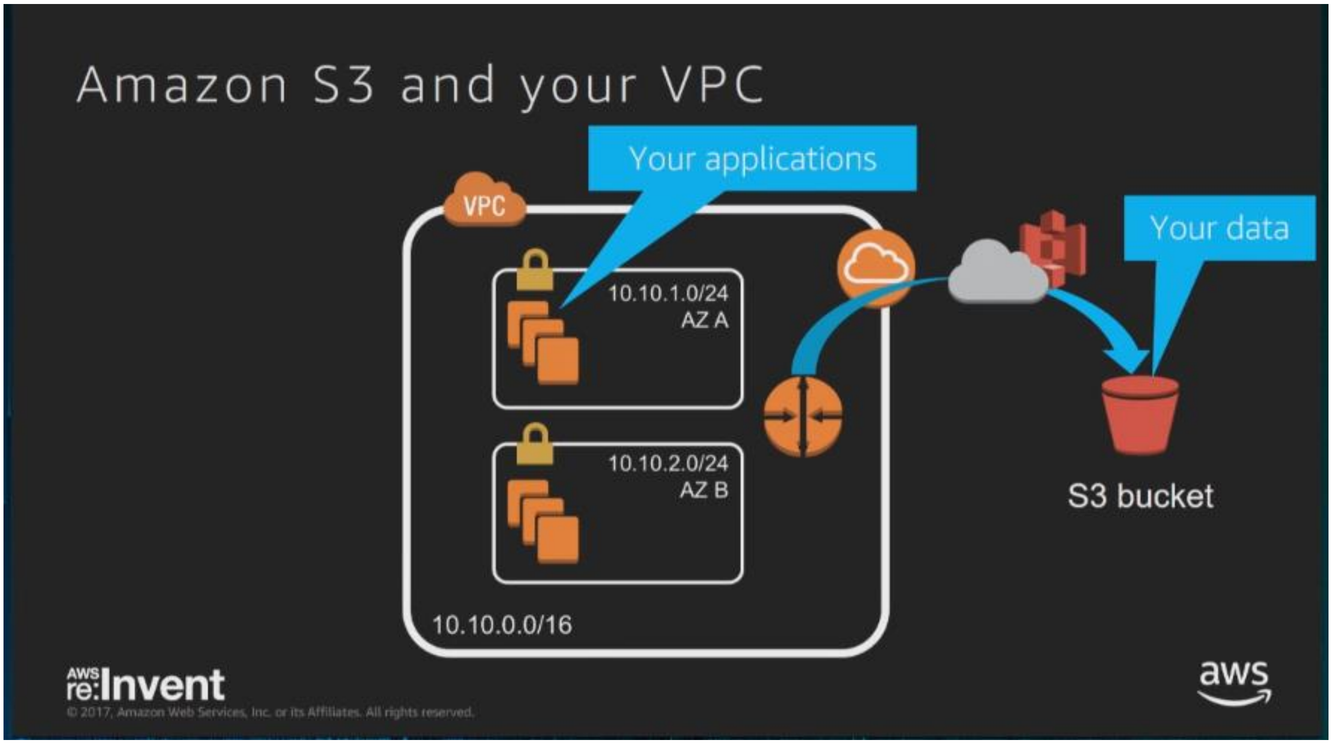
10.10.2.0/24
AZ B

10.10.0.0/16

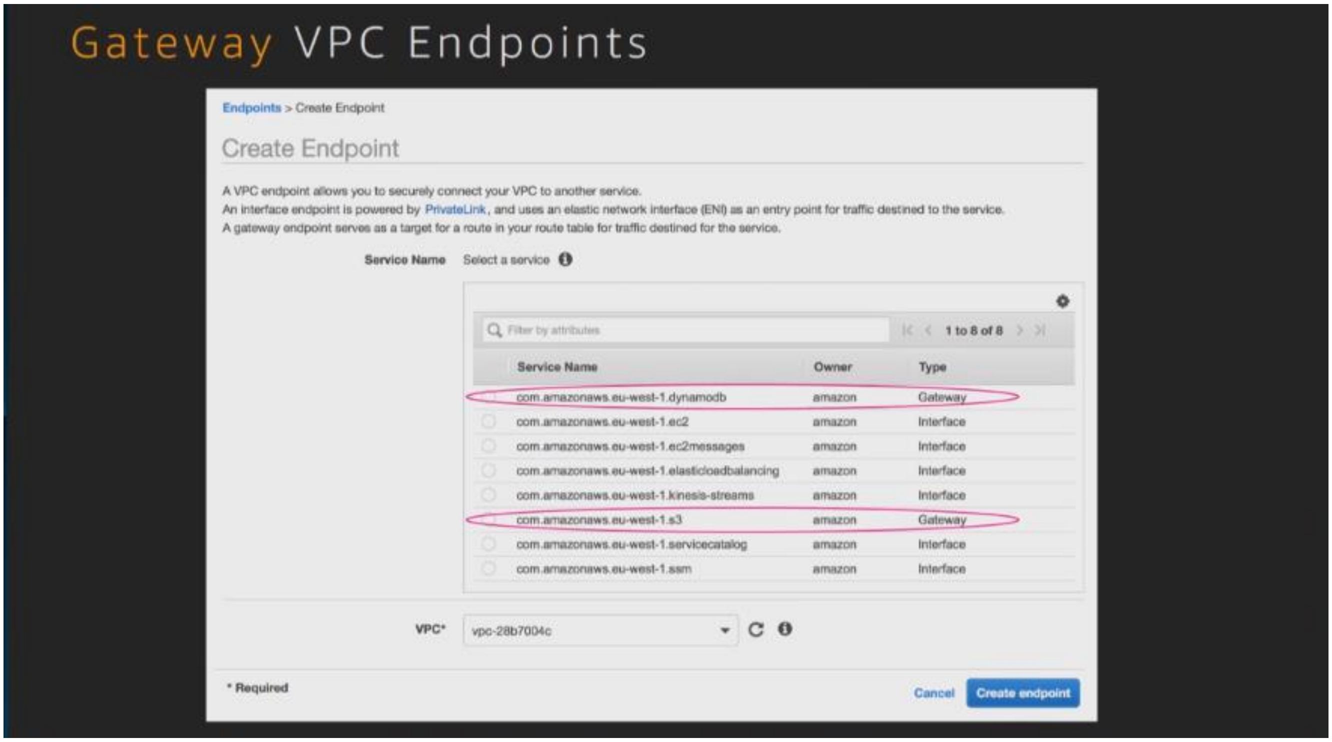
Your data

S3 bucket

You have your apps running in your VPC and you have your data stored in S3 buckets, but your data is like a part of your applications.



When you resolve the S3 bucket's DNS name, you are going to get a public IP address. So, if you want to access your data in S3, then you have to create a way to go over the internet to get that access using the bucket's public DNS name using an IGW or a NAT gateway.



Gateway VPC Endpoints

Endpoints > Create Endpoint

Create Endpoint

A VPC endpoint allows you to securely connect your VPC to another service.
An interface endpoint is powered by PrivateLink, and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.
A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

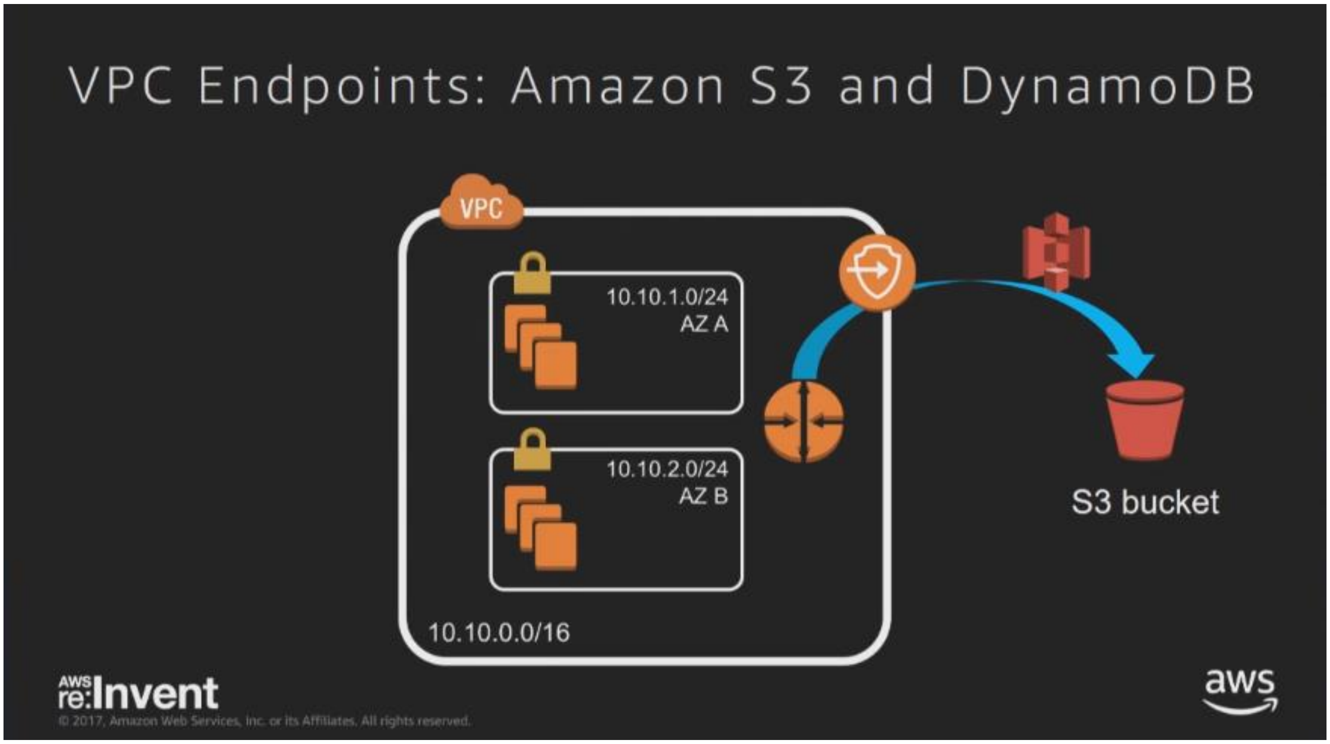
Service Name: Select a service

Service Name	Owner	Type
com.amazonaws.eu-west-1.dynamodb	amazon	Gateway
com.amazonaws.eu-west-1.ec2	amazon	Interface
com.amazonaws.eu-west-1.ec2messages	amazon	Interface
com.amazonaws.eu-west-1.elasticloadbalancing	amazon	Interface
com.amazonaws.eu-west-1.kinesis-streams	amazon	Interface
com.amazonaws.eu-west-1.s3	amazon	Gateway
com.amazonaws.eu-west-1.servicecatalog	amazon	Interface
com.amazonaws.eu-west-1.ssm	amazon	Interface

VPC*: vpc-28b7004c

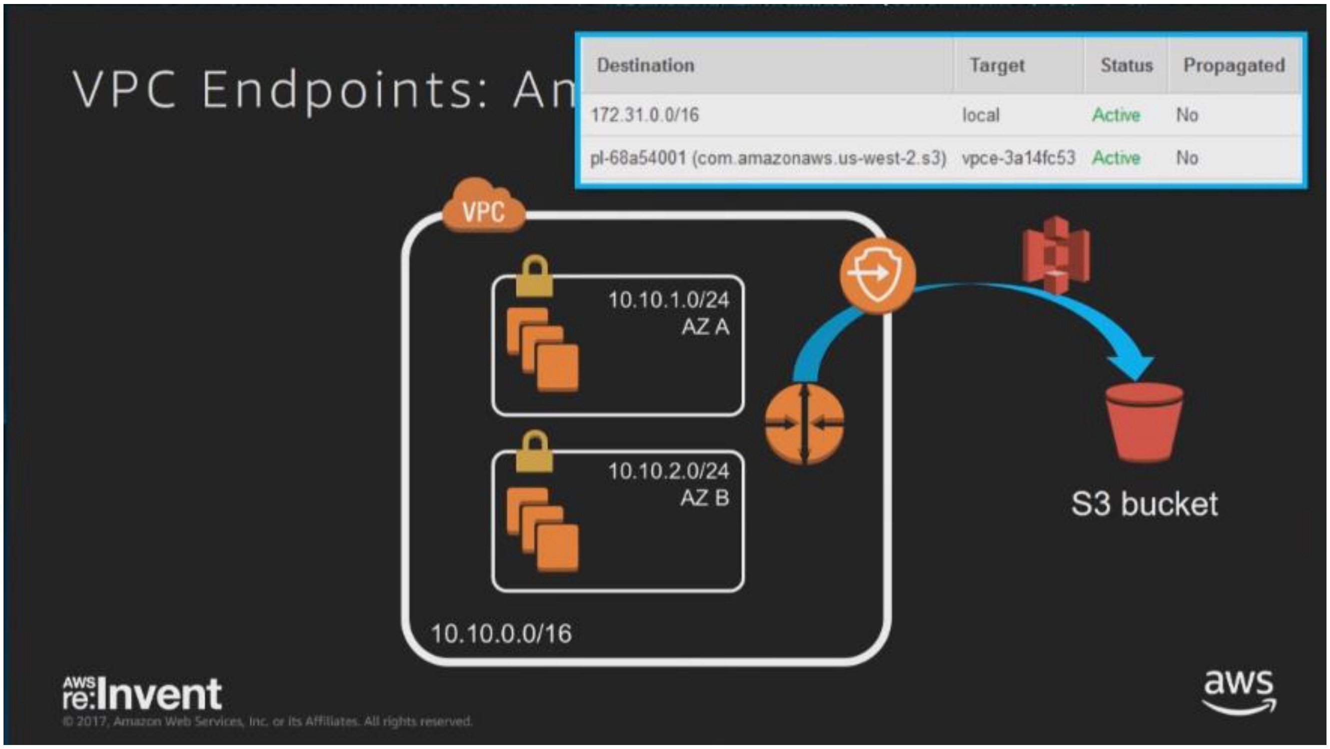
* Required

VPC Endpoints helps solve the problem of going over the internet to get access to your S3 data. The ***2 types of VPC endpoints*** are the ***Gateway VPC Endpoints*** and the ***Interface VPC Endpoints***.



VPC Endpoints: Amazon S3 and DynamoDB

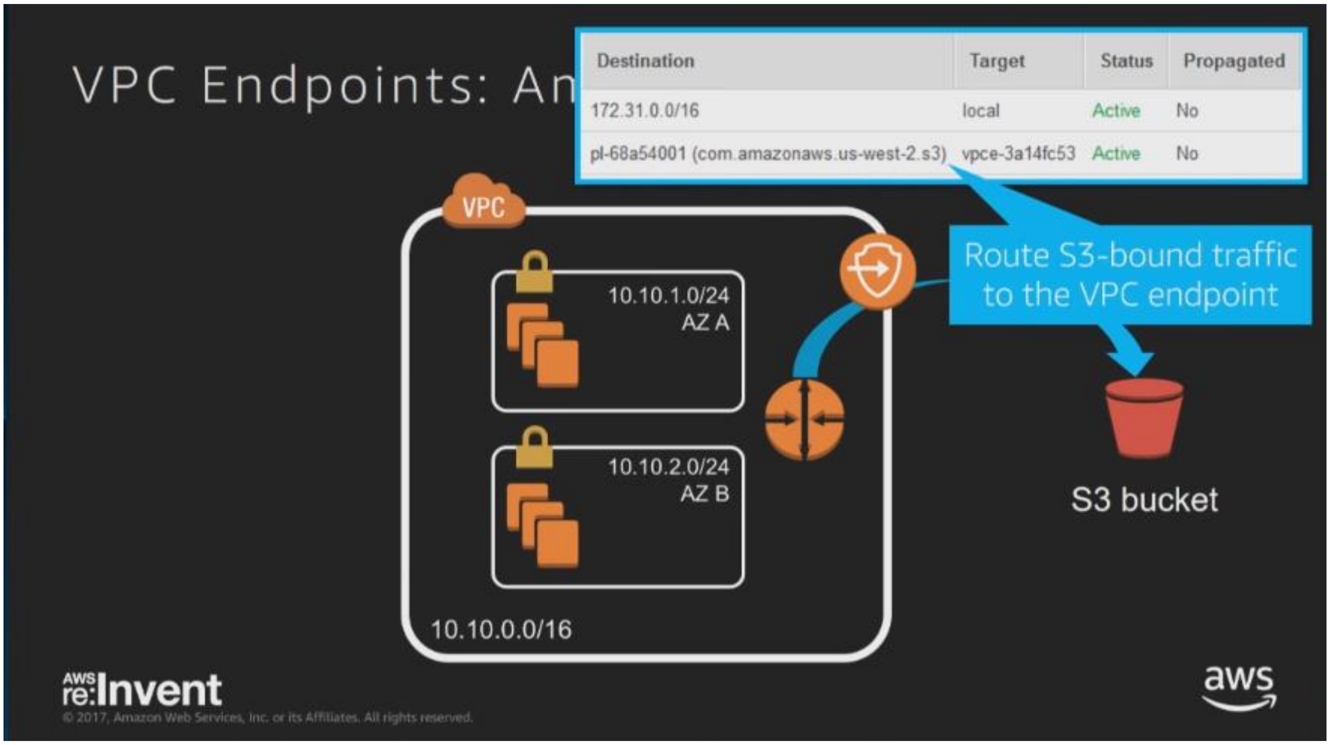
Gateway VPC Endpoints are supported for S3 and DynamoDB.



VPC Endpoints: An

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
pl-68a54001 (com.amazonaws.us-west-2.s3)	vpce-3a14fc53	Active	No

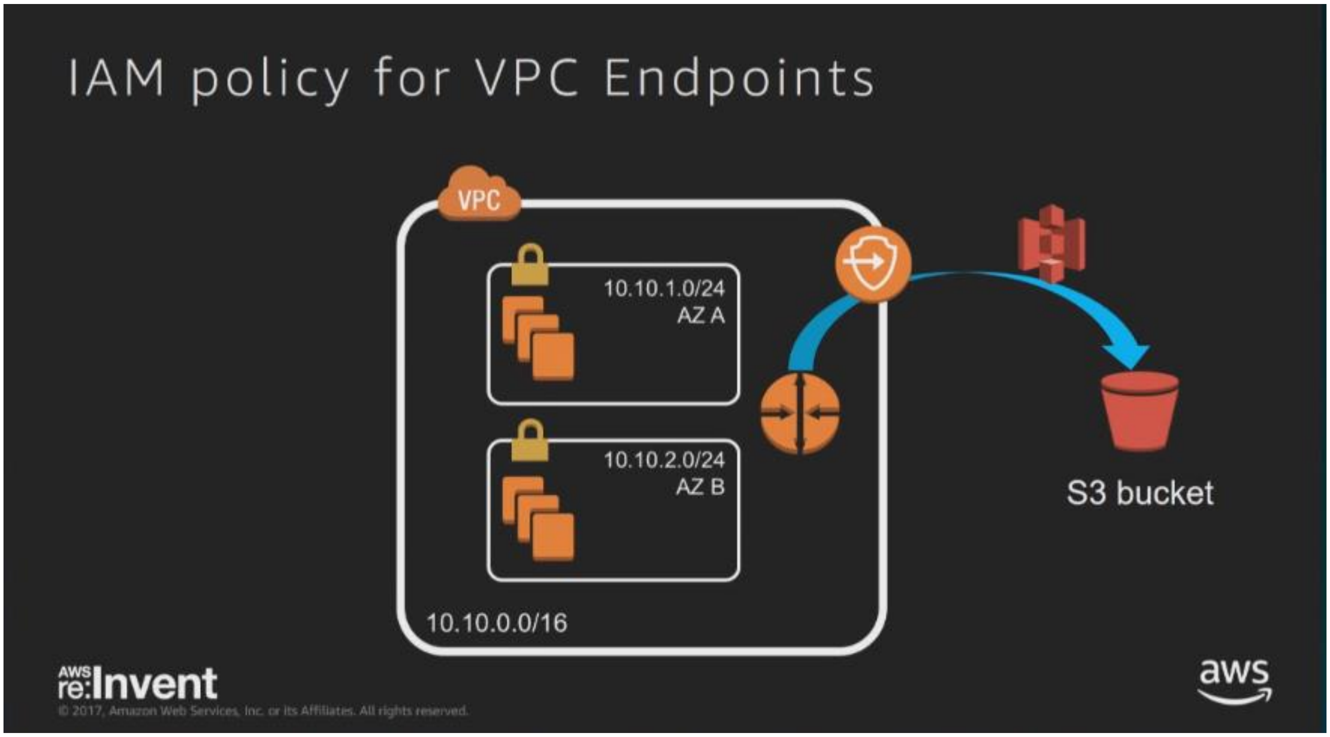
You simply create a ***Gateway VPC Endpoint*** and then create a route in your route table that behave like the other gateways and takes traffic destined for S3 bucket should be sent to the VPC Endpoint.



VPC Endpoints: An

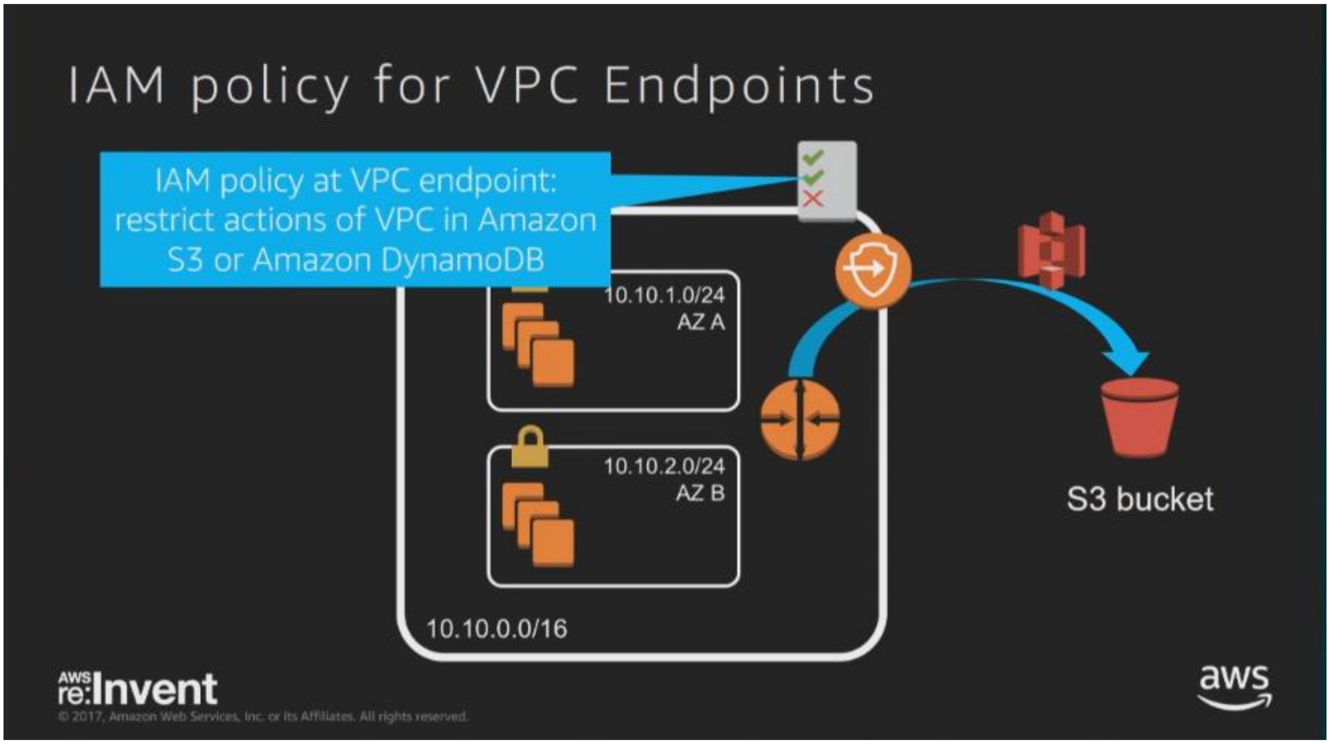
Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
pl-68a54001 (com.amazonaws.us-west-2.s3)	vpce-3a14fc53	Active	No

Route S3-bound traffic to the VPC endpoint



IAM policy for VPC Endpoints

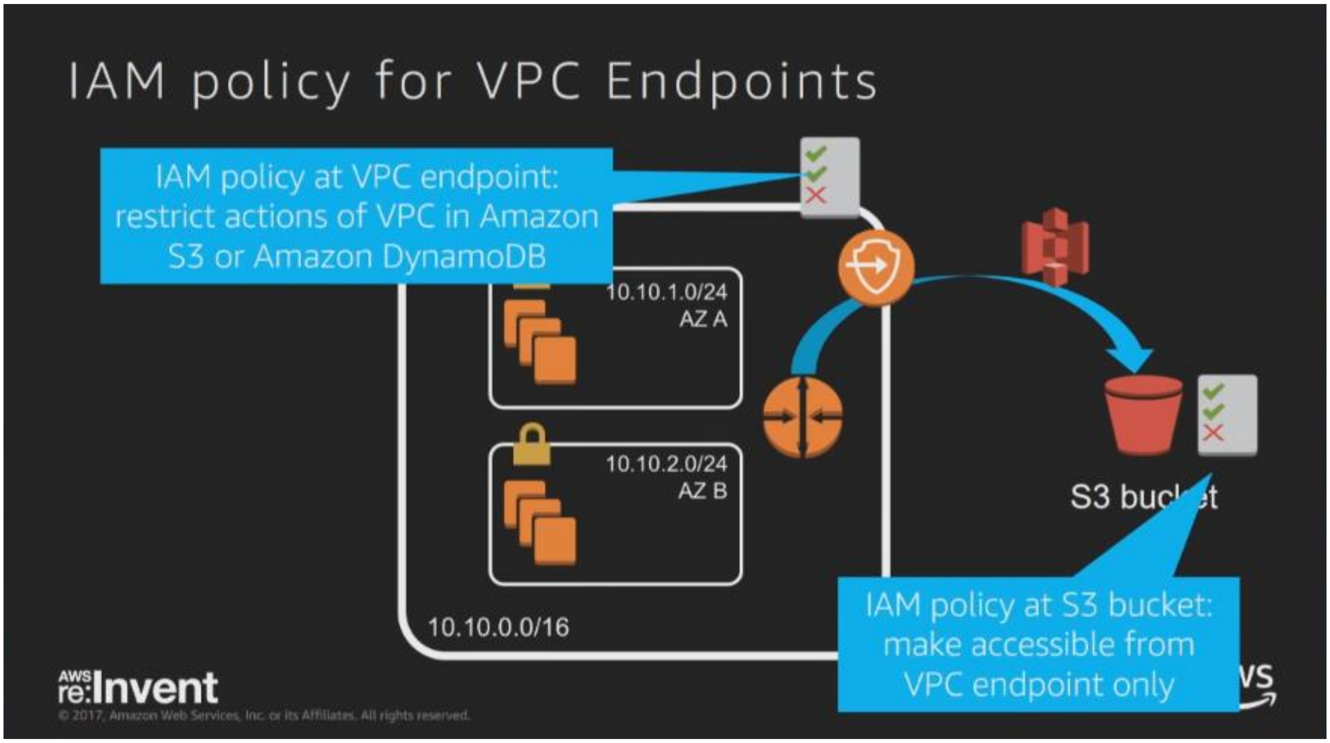
Using VPC Endpoints, you get the ability to add IAM policies on the VPC Endpoint.



IAM policy for VPC Endpoints

IAM policy at VPC endpoint: restrict actions of VPC in Amazon S3 or Amazon DynamoDB

This applies directly to the VPC Endpoint, so you can say exactly what the VPC can do and cannot do with each service that it is talking to using the IAM policies.

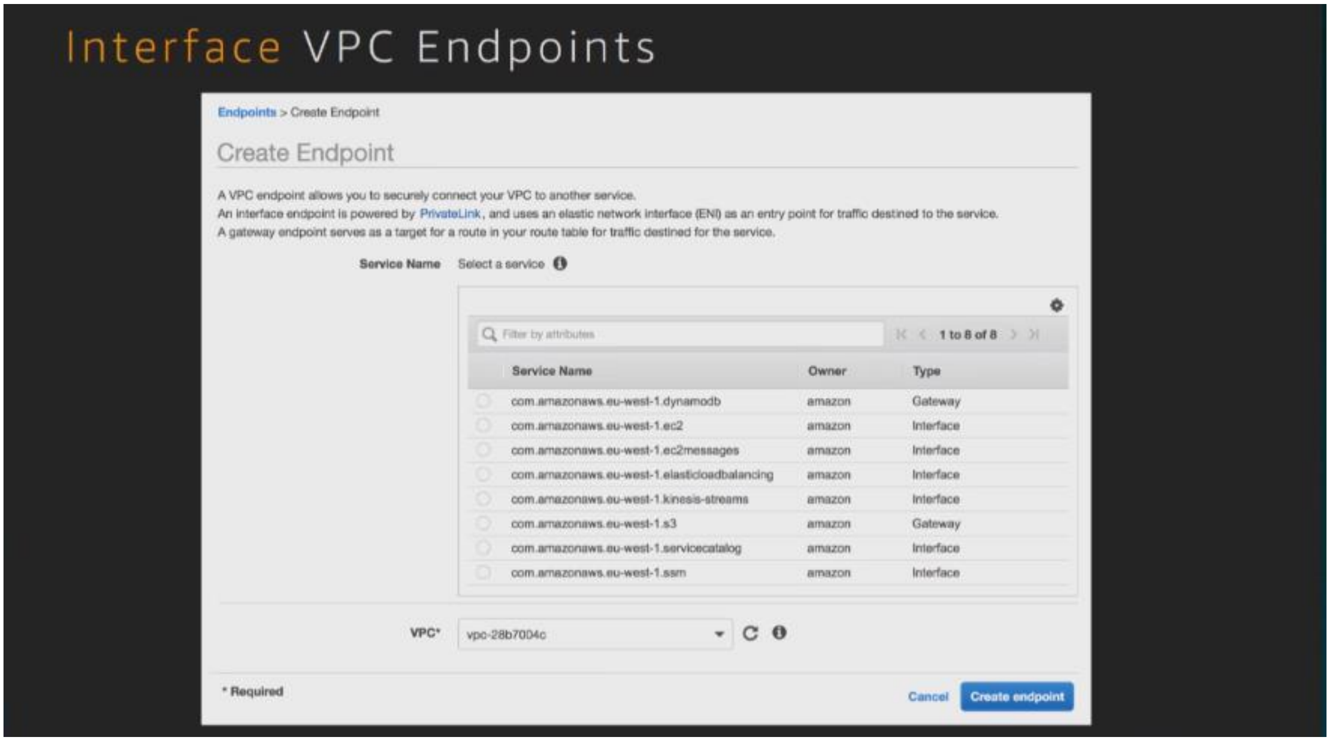


IAM policy for VPC Endpoints

IAM policy at VPC endpoint: restrict actions of VPC in Amazon S3 or Amazon DynamoDB

IAM policy at S3 bucket: make accessible from VPC endpoint only

With S3 you also have the ability to add a policy on the S3 bucket, this means that we can lock down the S3 bucket to only allow access from traffic coming from that particular VPC Endpoint.



Interface VPC Endpoints

Create Endpoint

A VPC endpoint allows you to securely connect your VPC to another service.
An interface endpoint is powered by PrivateLink, and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.
A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

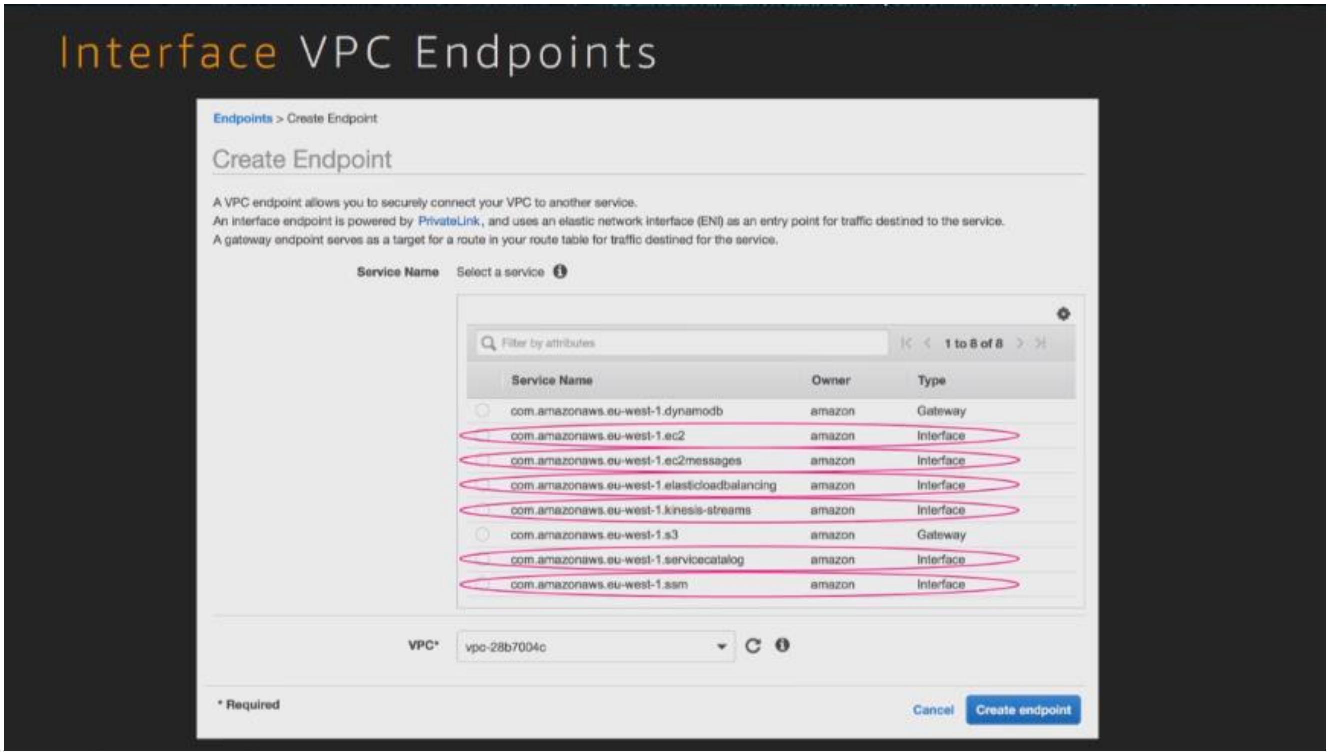
Service Name: Select a service

	Service Name	Owner	Type
○	com.amazonaws.eu-west-1.dynamodb	amazon	Gateway
○	com.amazonaws.eu-west-1.ec2	amazon	Interface
○	com.amazonaws.eu-west-1.ec2messages	amazon	Interface
○	com.amazonaws.eu-west-1.elasticloadbalancing	amazon	Interface
○	com.amazonaws.eu-west-1.kinesis-streams	amazon	Interface
○	com.amazonaws.eu-west-1.s3	amazon	Gateway
○	com.amazonaws.eu-west-1.servicecatalog	amazon	Interface
○	com.amazonaws.eu-west-1.ssm	amazon	Interface

VPC*: vpc-28b7004c

* Required

The 2nd type of VPC Endpoint is the ***Interface VPC Endpoint***, this is also called ***AWS PrivateLink for AWS Services***.



Interface VPC Endpoints

Endpoints > Create Endpoint

Create Endpoint

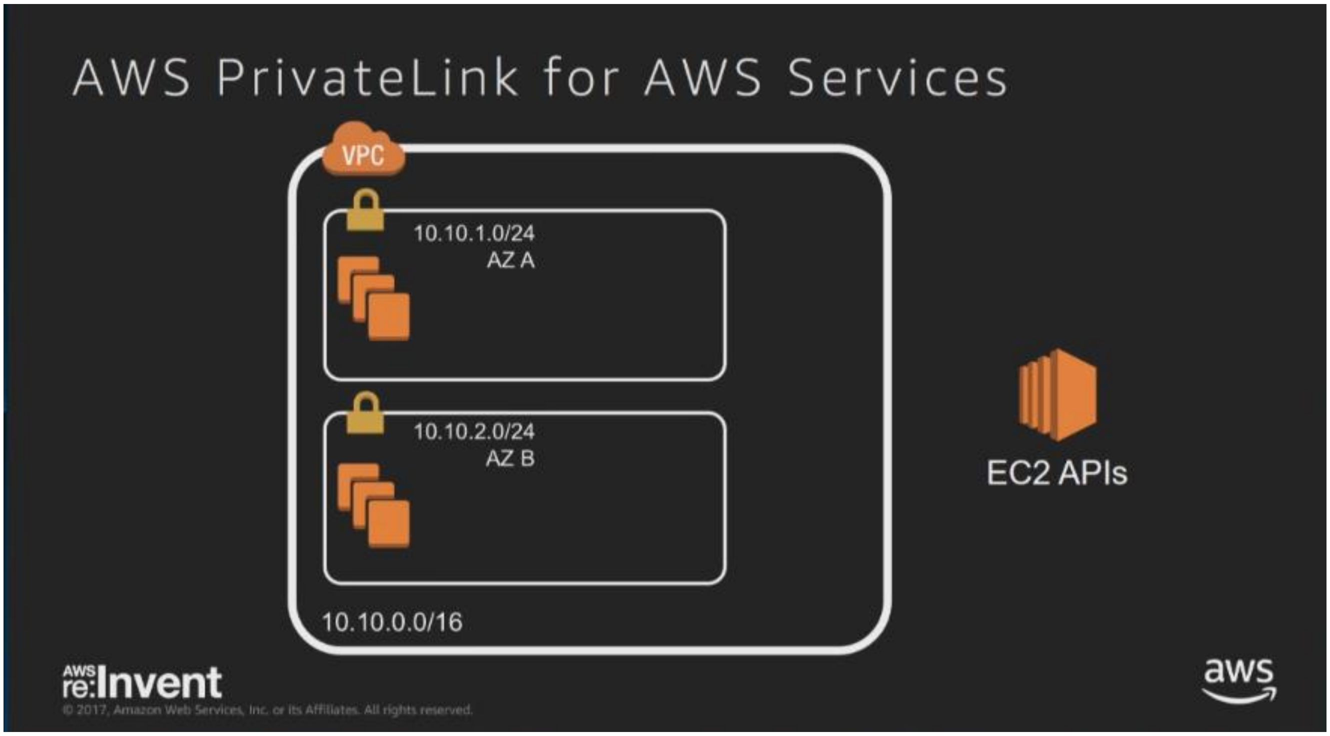
A VPC endpoint allows you to securely connect your VPC to another service.
An interface endpoint is powered by PrivateLink, and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.
A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

Service Name: Select a service

	Service Name	Owner	Type
○	com.amazonaws.eu-west-1.dynamodb	amazon	Gateway
○	com.amazonaws.eu-west-1.ec2	amazon	Interface
○	com.amazonaws.eu-west-1.ec2messages	amazon	Interface
○	com.amazonaws.eu-west-1.elasticloadbalancing	amazon	Interface
○	com.amazonaws.eu-west-1.kinesis-streams	amazon	Interface
○	com.amazonaws.eu-west-1.s3	amazon	Gateway
○	com.amazonaws.eu-west-1.servicecatalog	amazon	Interface
○	com.amazonaws.eu-west-1.ssm	amazon	Interface

VPC*: vpc-28b7004c

* Required



AWS PrivateLink for AWS Services

VPC

10.10.1.0/24
AZ A

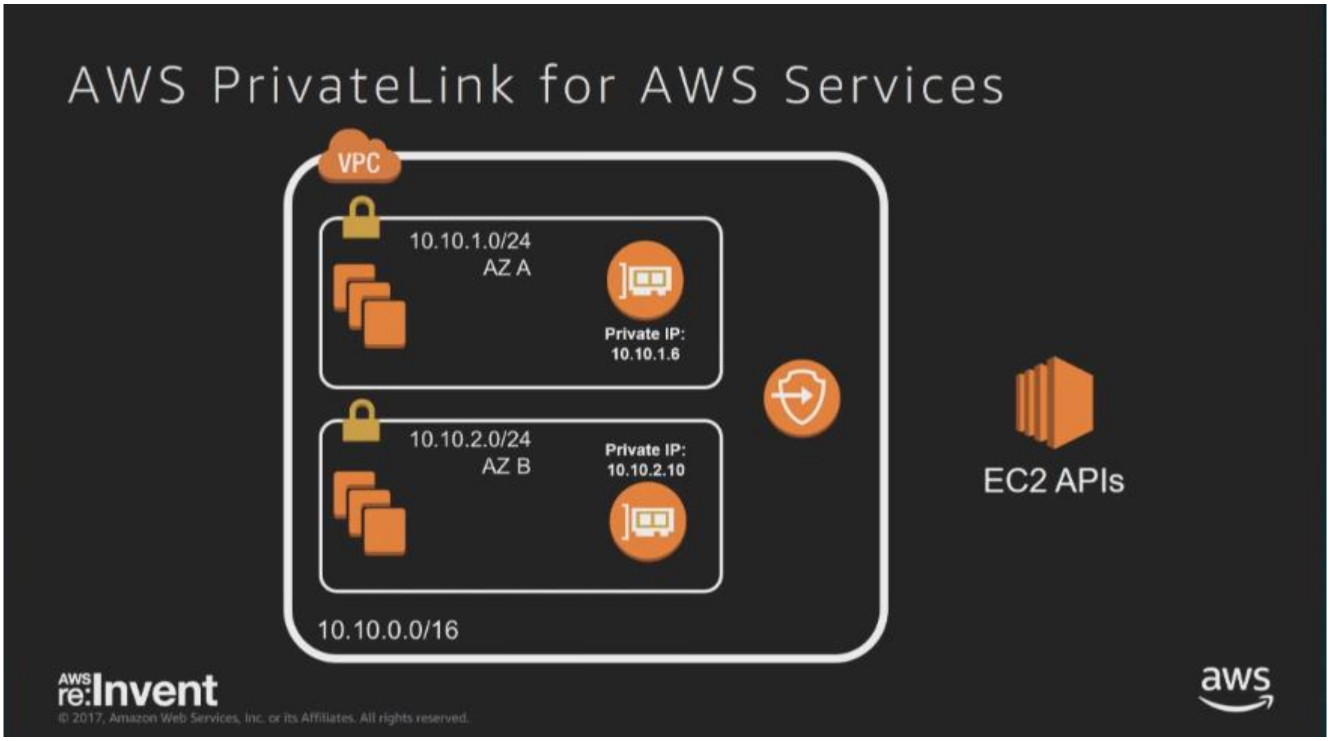
10.10.2.0/24
AZ B

10.10.0.0/16

EC2 APIs

re:Invent

AWS PrivateLink is a VPC endpoint



AWS PrivateLink for AWS Services

VPC

10.10.1.0/24
AZ A

Private IP:
10.10.1.6

10.10.2.0/24
AZ B

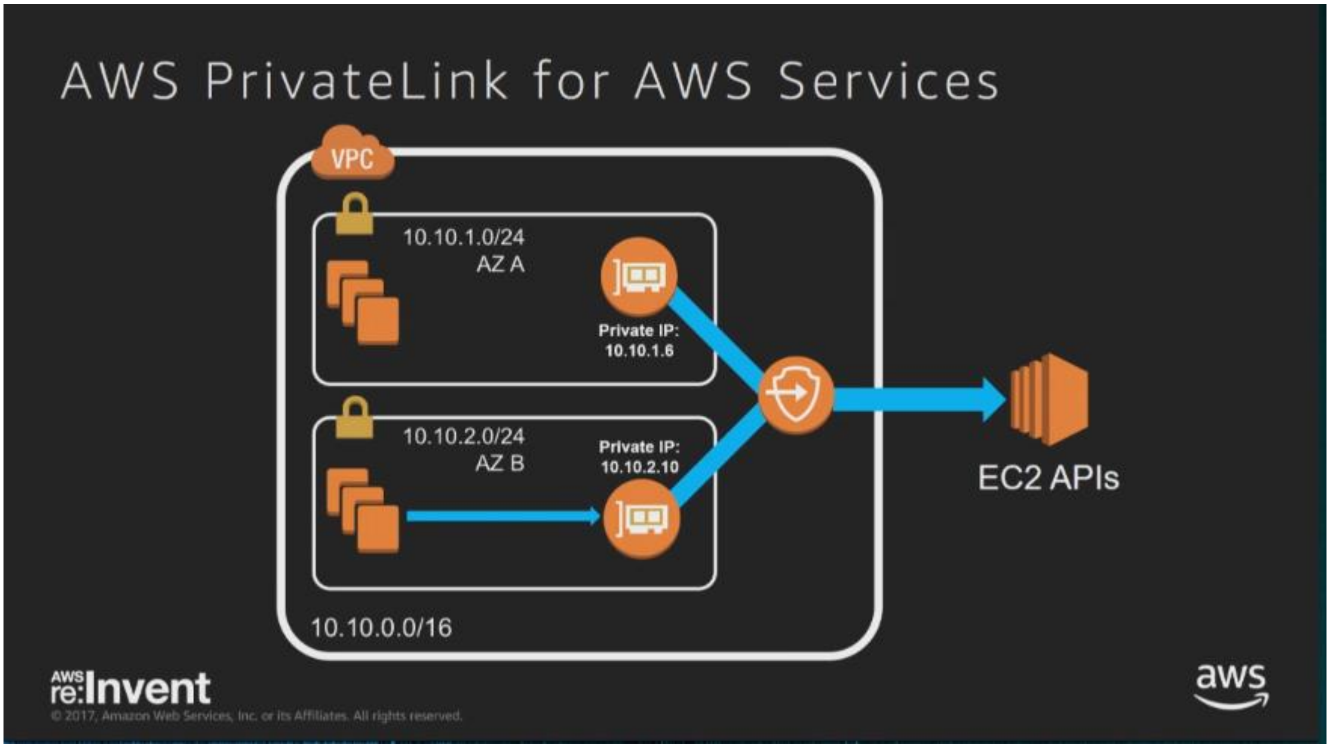
Private IP:
10.10.2.10

10.10.0.0/16

EC2 APIs

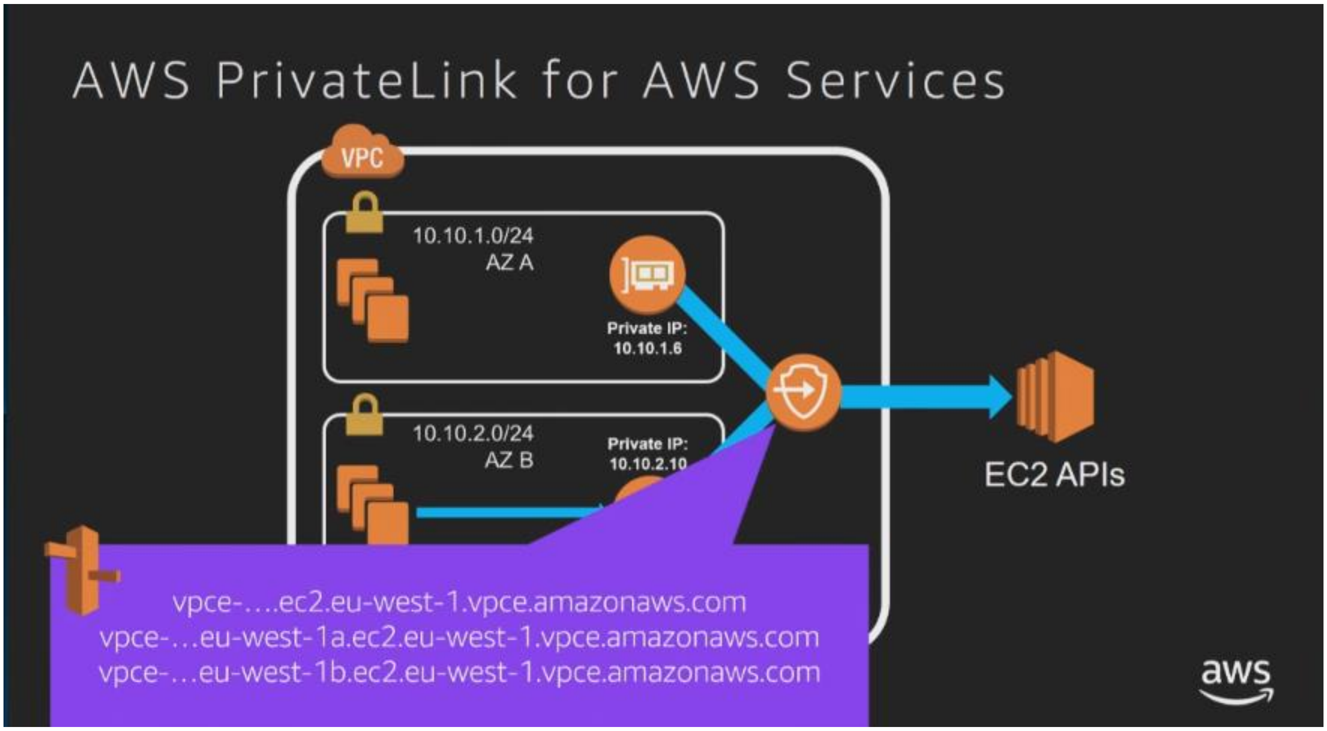
re:Invent

When you create a VPC endpoint, you specify subnets and it creates an ***Elastic Network Interface*** ENI in each of the subnets that you specify.



AWS PrivateLink for AWS Services

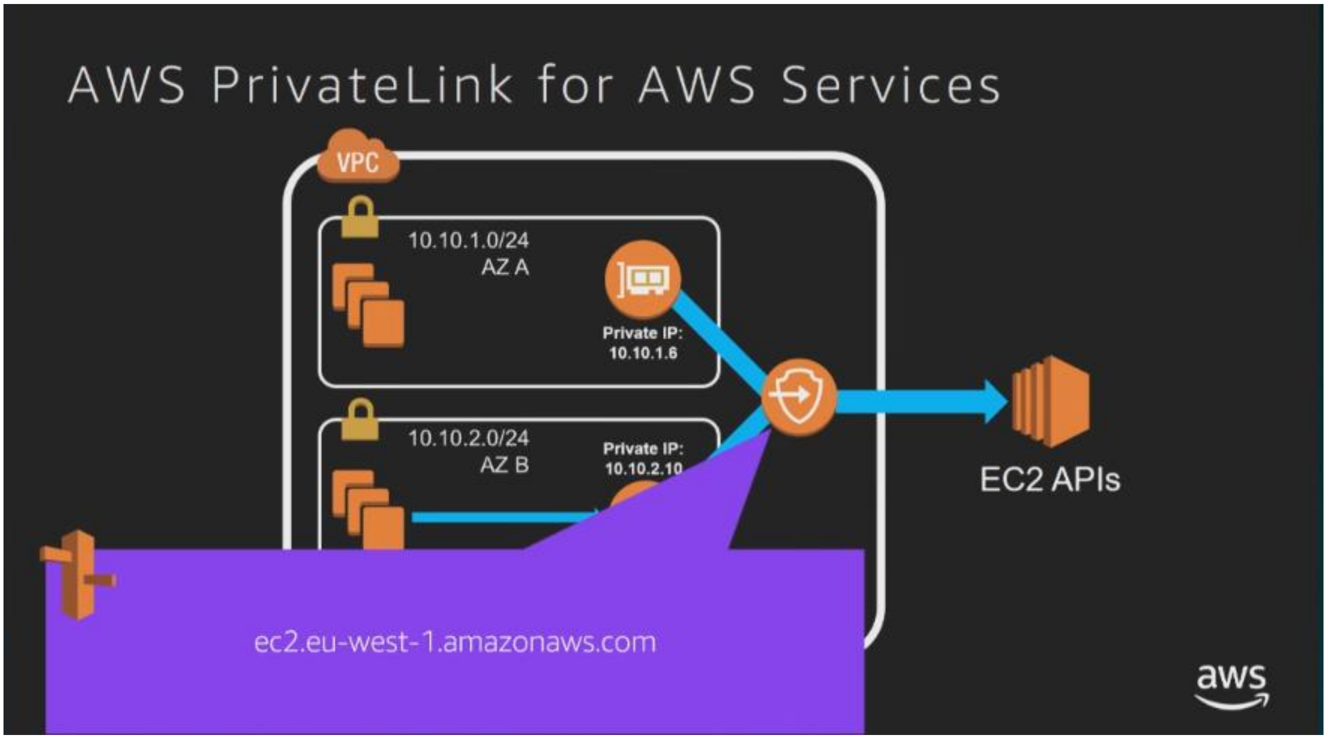
The ENI has a Private IP and when you send traffic to the Private IP, it goes to the VPC endpoint and directly to the service. This gives you Private IP connectivity to AWS services from within your VPC.



AWS PrivateLink for AWS Services

vpce-….ec2.eu-west-1.vpce.amazonaws.com
vpce-…eu-west-1a.ec2.eu-west-1.vpce.amazonaws.com
vpce-…eu-west-1b.ec2.eu-west-1.vpce.amazonaws.com

Because this is not a gateway and you don't use routing tables to send traffic to it, you get given some DNS name. You get one DNS name that will have all of the IPs, and you also get a zonal DNS name for each specific IP so that you can keep traffic within a zone if you want to.

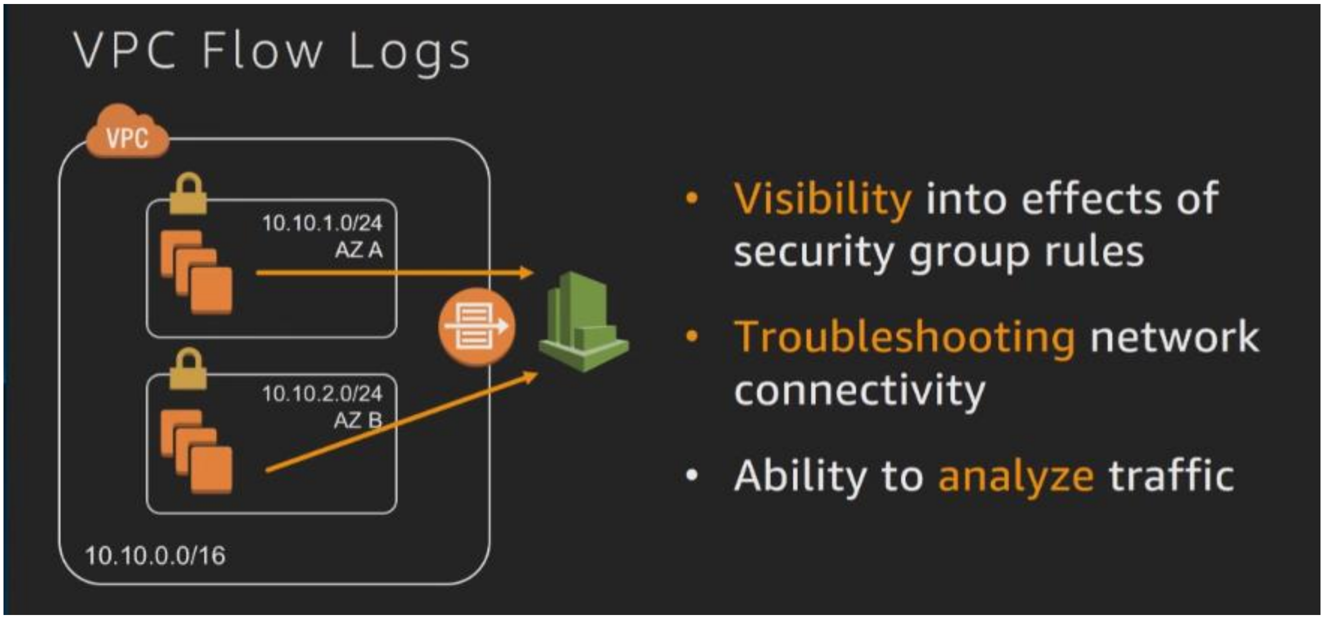


AWS PrivateLink for AWS Services

You can also choose to let AWS manage a DNS name in your VPC that looks exactly like the DNS name of the service outside of your VPC. So that when you resolve that DNS inside your VPC, you will go your VPC endpoint and if you were outside you will go to the public endpoint.

VPC Flow Logs:
VPC traffic metadata in Amazon CloudWatch Logs

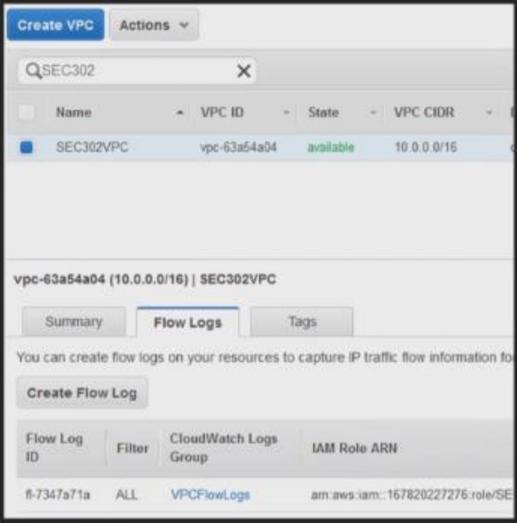
VPC Flow Logs gives you visibility into your VPC, you don't have to manage any infrastructure for this to work



VPC Flow Logs

- Visibility into effects of security group rules
- Troubleshooting network connectivity
- Ability to analyze traffic

VPC Flow Logs: Setup



Name	VPC ID	State	VPC CIDR
SEC302VPC	vpc-63a54a04	available	10.0.0.0/16

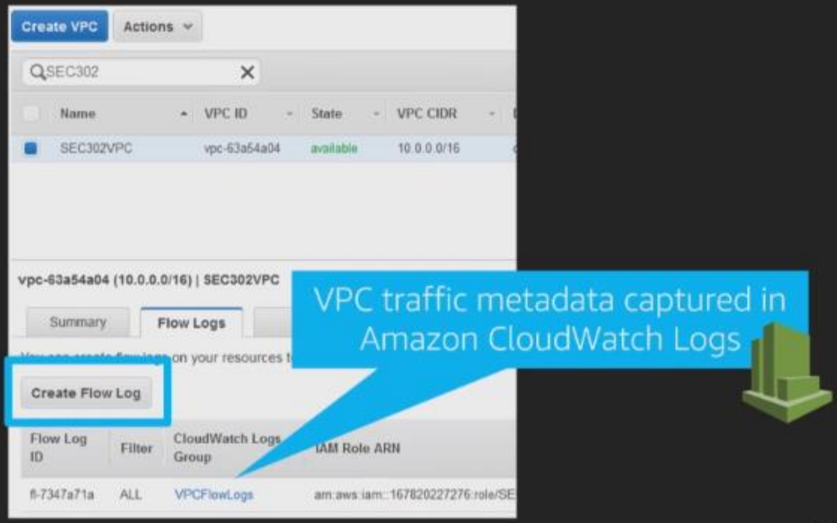
vpc-63a54a04 (10.0.0.0/16) | SEC302VPC

You can create flow logs on your resources to capture IP traffic flow information fo

Flow Log ID	Filter	CloudWatch Logs Group	IAM Role ARN
fl-7347a71a	ALL	VPCFlowLogs	arn:aws:iam::167820227276:role/SE

re:Invent

VPC Flow Logs: Setup



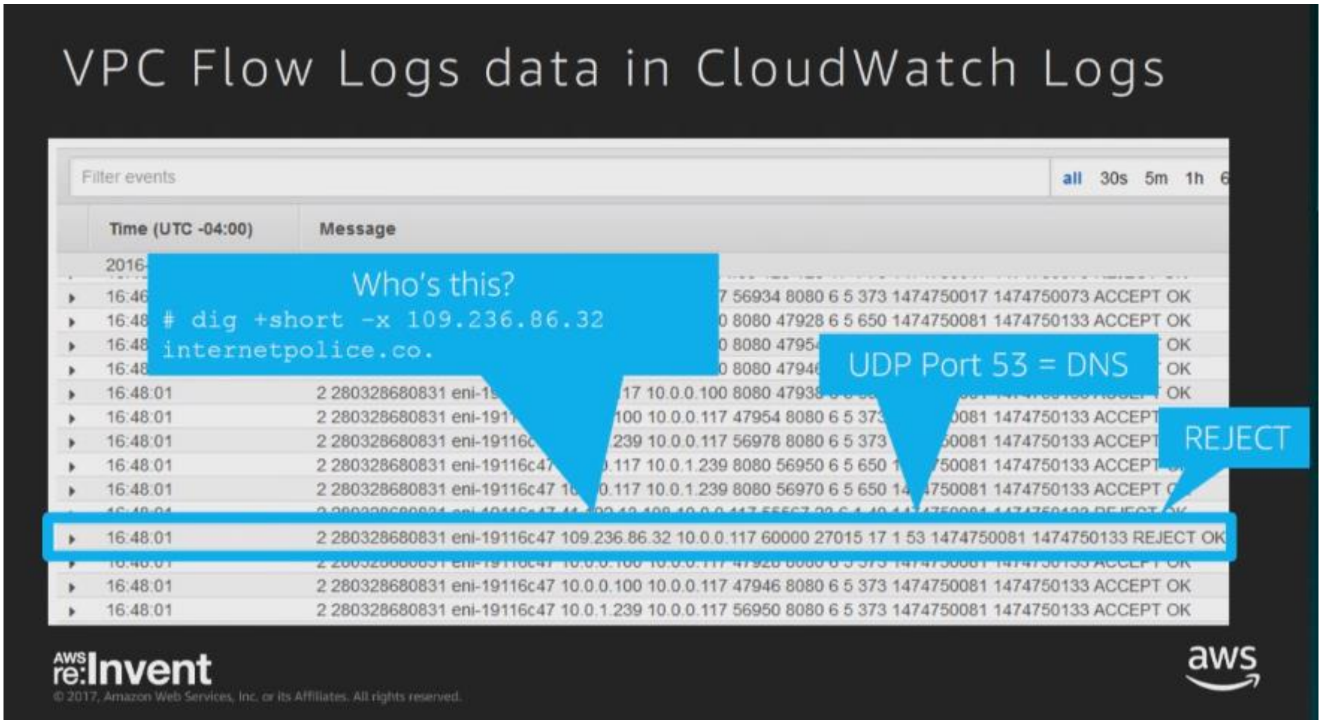
Name	VPC ID	State	VPC CIDR
SEC302VPC	vpc-63a54a04	available	10.0.0.0/16

vpc-63a54a04 (10.0.0.0/16) | SEC302VPC

Flow Log ID	Filter	CloudWatch Logs Group	IAM Role ARN
fl-7347a71a	ALL	VPCFlowLogs	arn:aws:iam::167820227276:role/SE

VPC traffic metadata captured in Amazon CloudWatch Logs

re:Invent



VPC Flow Logs data in CloudWatch Logs

```
Time (UTC -04:00)  Message
2016-
16:46              7 56934 8080 6 5 373 1474750017 1474750073 ACCEPT OK
16:48              0 8080 47928 6 5 650 1474750081 1474750133 ACCEPT OK
16:48              0 8080 4795                                    OK
16:48              0 8080 4794                                    OK
16:48:01           2 280328680831 eni-19    17 10.0.0.100 8080 47938          OK
16:48:01           2 280328680831 eni-191    100 10.0.0.117 47954 8080 6 5 37    0081 1474750133 ACCEPT
16:48:01           2 280328680831 eni-19116c    239 10.0.0.117 56978 8080 6 5 373    50081 1474750133 ACCEPT
16:48:01           2 280328680831 eni-19116c47    .117 10.0.1.239 8080 56950 6 5 650 1    750081 1474750133 ACCEPT
16:48:01           2 280328680831 eni-19116c47 1    .117 10.0.1.239 8080 56970 6 5 650 14    750081 1474750133 ACCEPT
16:48:01           2 280328680831 eni-19116c47 109.236.86.32 10.0.0.117 60000 27015 17 1 53 1474750081 1474750133 REJECT OK
16:48:01           2 280328680831 eni-19116c47 10.0.0.100 10.0.0.117 47946 8080 6 5 373 1474750081 1474750133 ACCEPT OK
16:48:01           2 280328680831 eni-19116c47 10.0.1.239 10.0.0.117 56950 8080 6 5 373 1474750081 1474750133 ACCEPT OK
```

Who's this?

```
# dig +short -x 109.236.86.32
internetpolice.co.
```

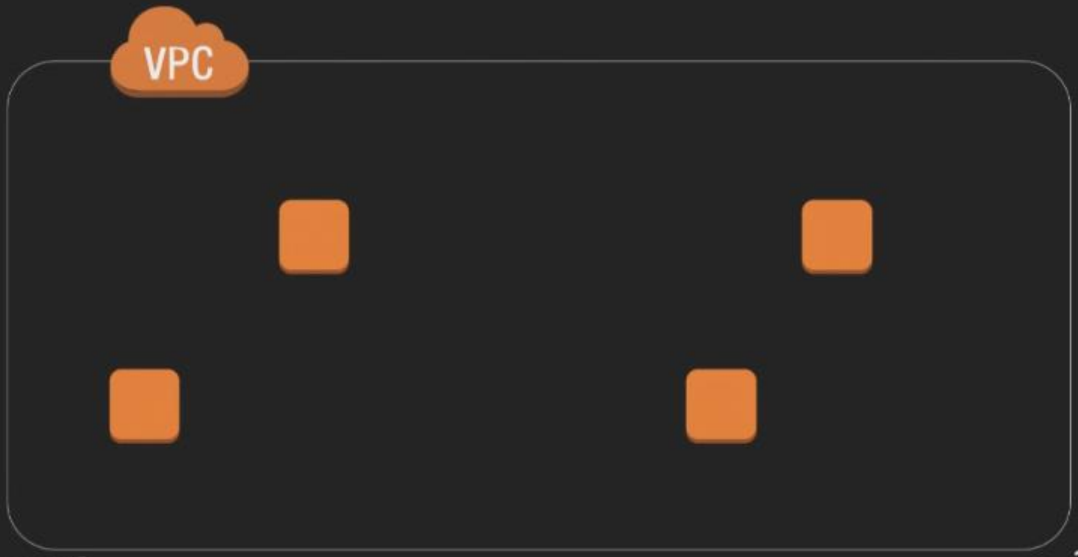
UDP Port 53 = DNS

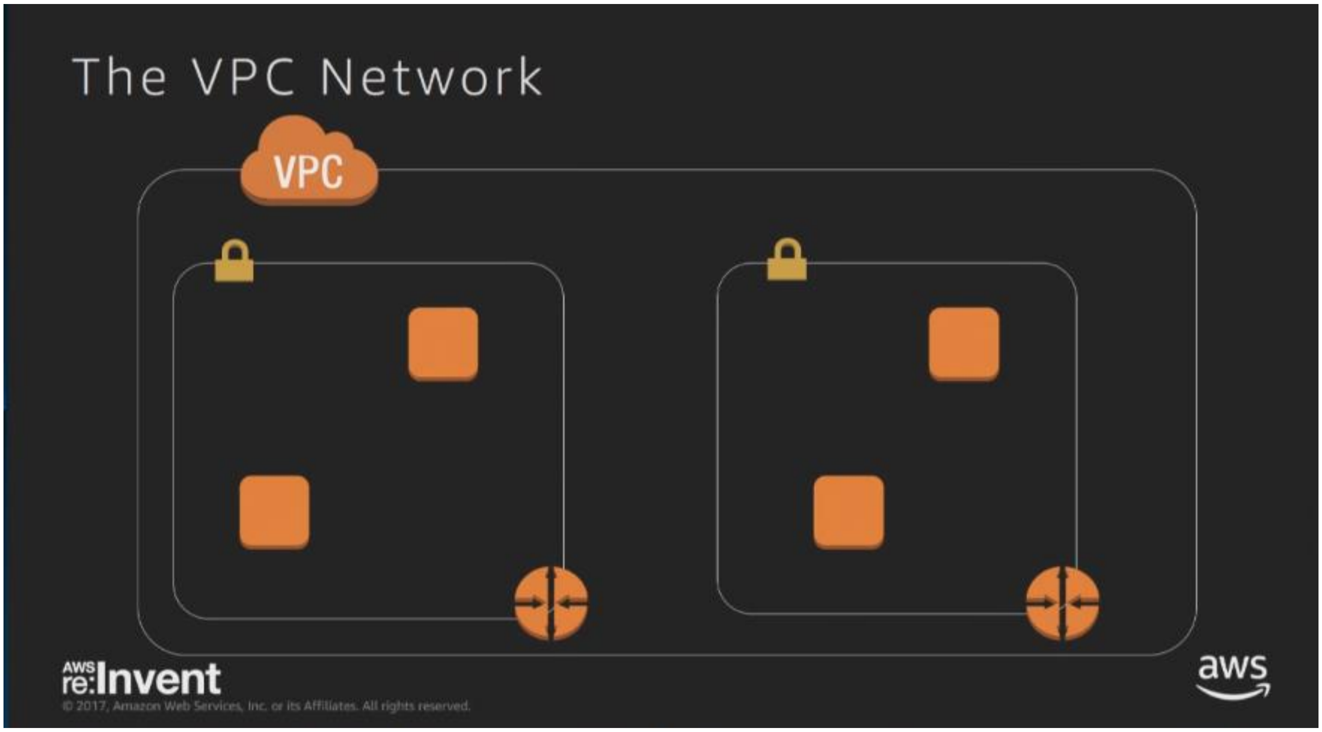
REJECT

This entry is a reject traffic log, we can do a reverse lookup to see who was scanning traffic on our instance

VPC: Your Private Network in AWS

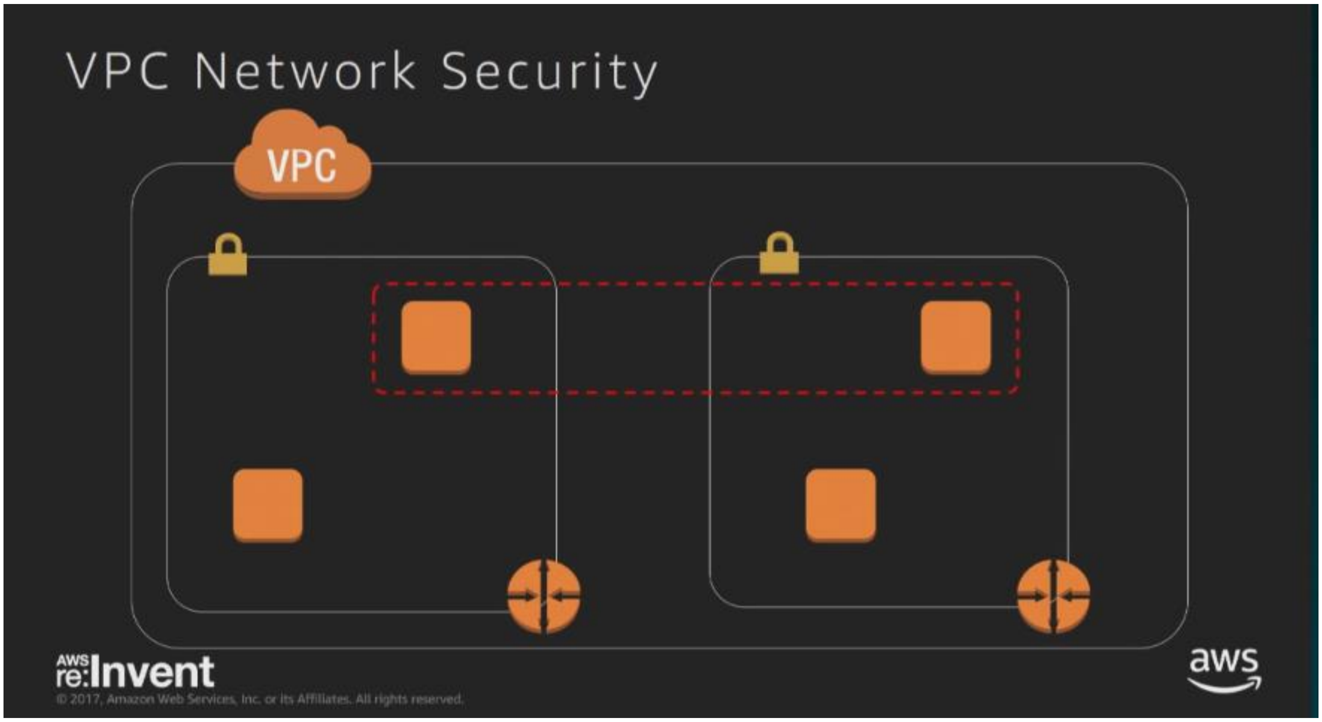
The VPC Network





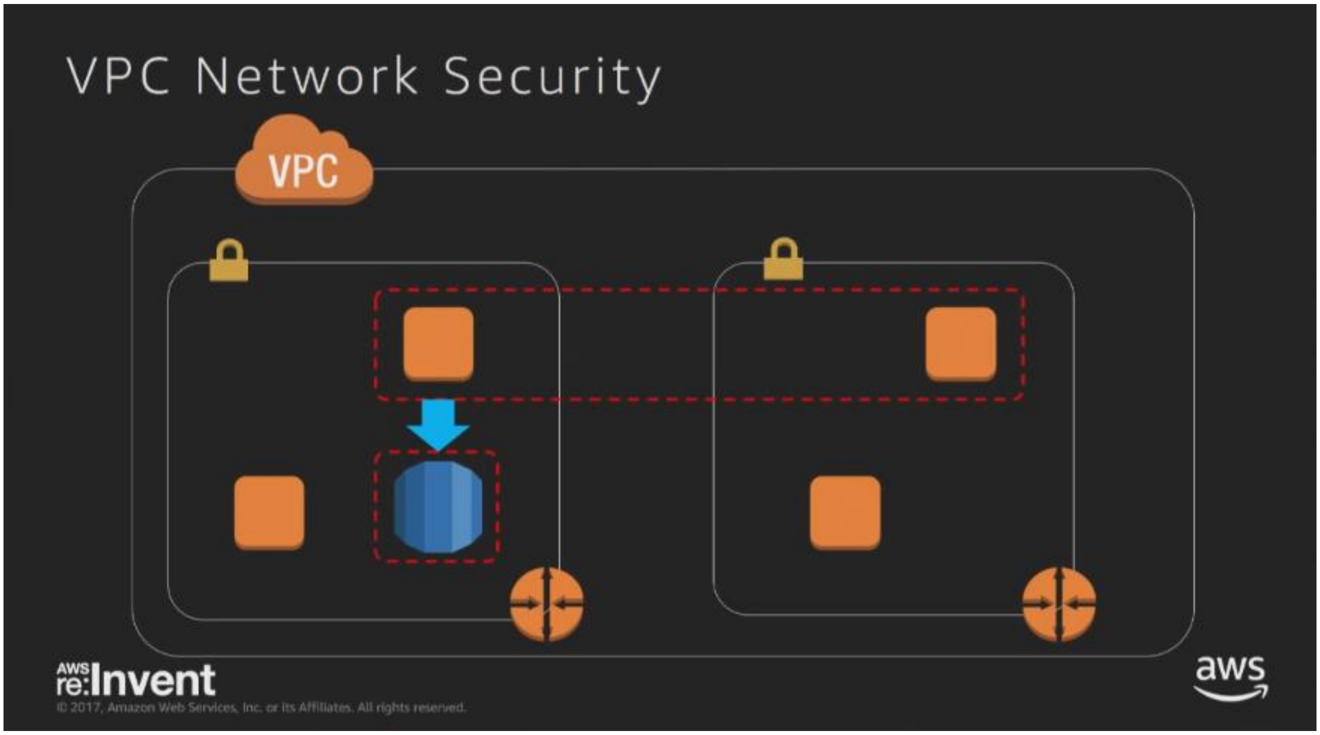
The VPC Network

Use subnets for HA applications



VPC Network Security

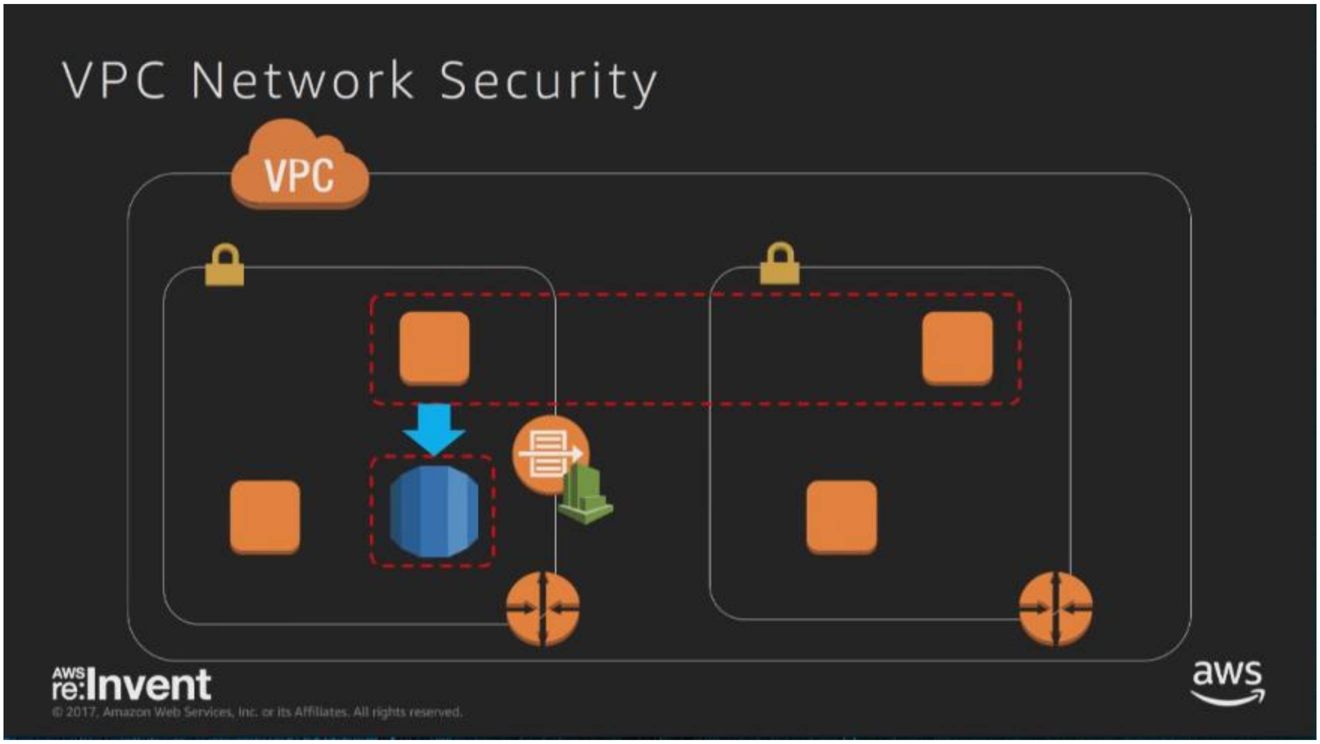
Security groups help us control who has access to talk to whom



VPC Network Security

VPC
AWS re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
aws

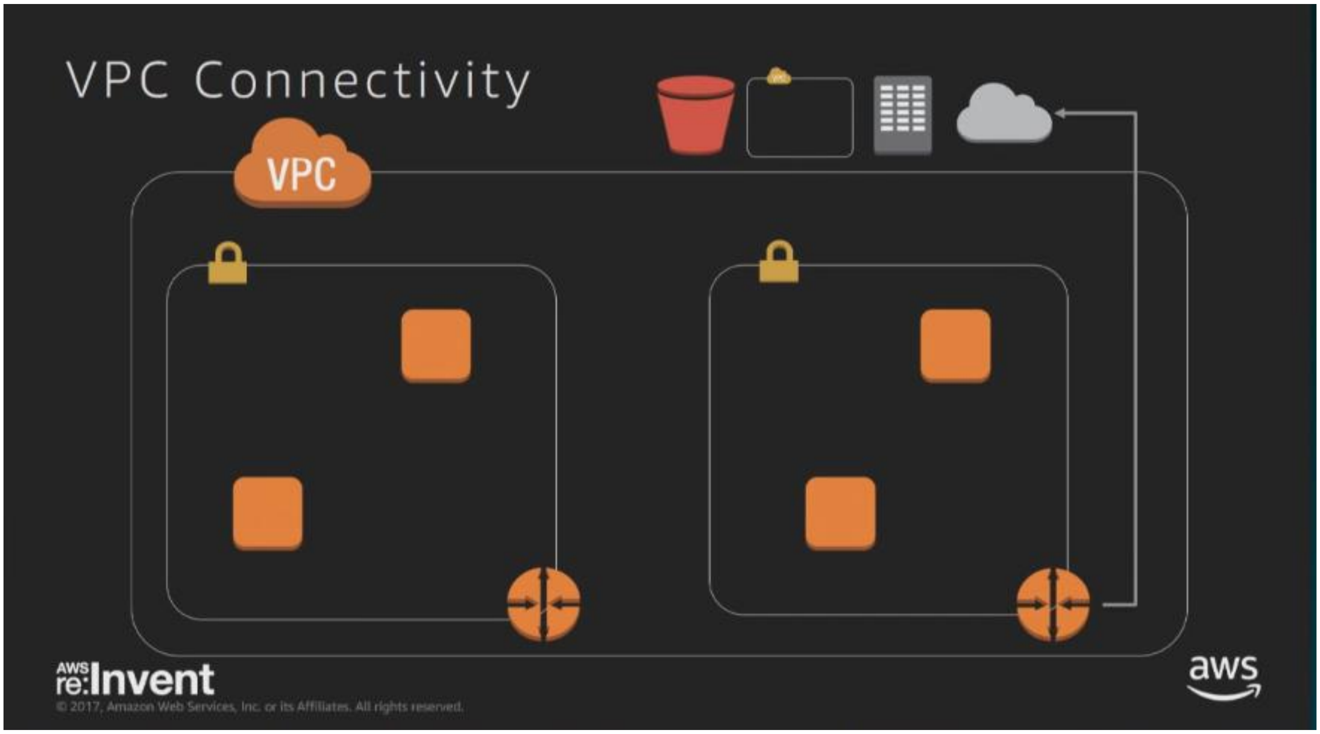
We can also run AWS services inside our VPC



VPC Network Security

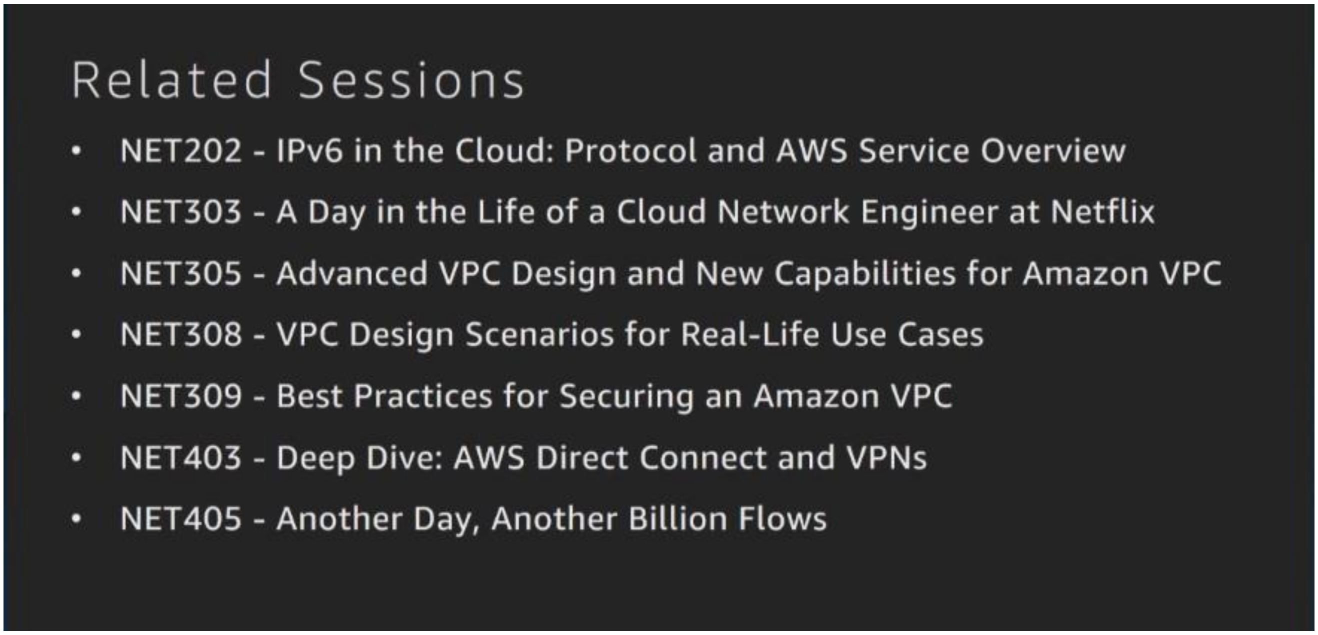
VPC
AWS re:Invent
© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
aws

We can also get visibility into our VPC using AWS VPC Flow Logs.



VPC Connectivity

We can get connectivity to the internet using ***IGW***, connectivity to your on-premises DC and networks using ***VPN*** and ***AWS DirectConnect***, to other VPCs using VPC Peering, and to other AWS services like S3 using ***VPC Endpoints***.



Related Sessions

- NET202 - IPv6 in the Cloud: Protocol and AWS Service Overview
- NET303 - A Day in the Life of a Cloud Network Engineer at Netflix
- NET305 - Advanced VPC Design and New Capabilities for Amazon VPC
- NET308 - VPC Design Scenarios for Real-Life Use Cases
- NET309 - Best Practices for Securing an Amazon VPC
- NET403 - Deep Dive: AWS Direct Connect and VPNs
- NET405 - Another Day, Another Billion Flows

AWS re:Invent

Thank you!