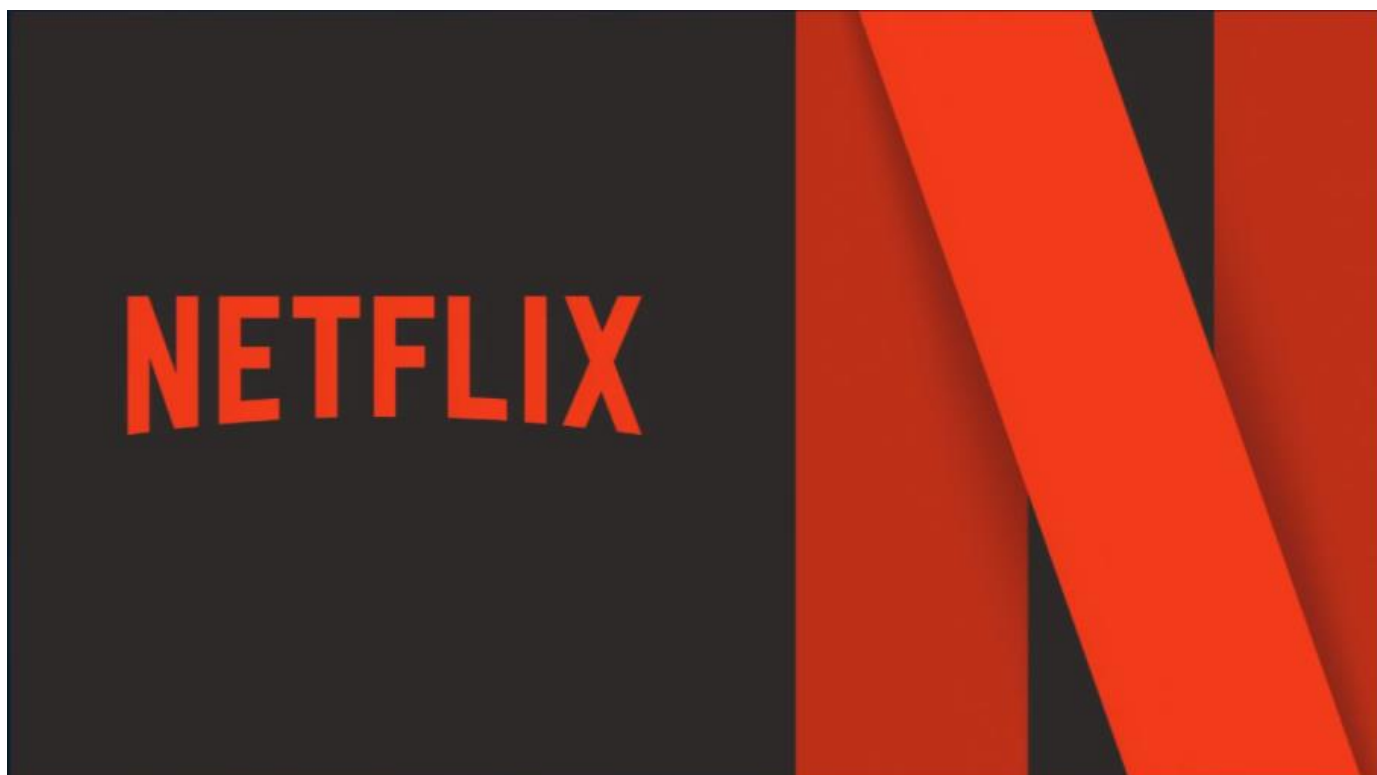
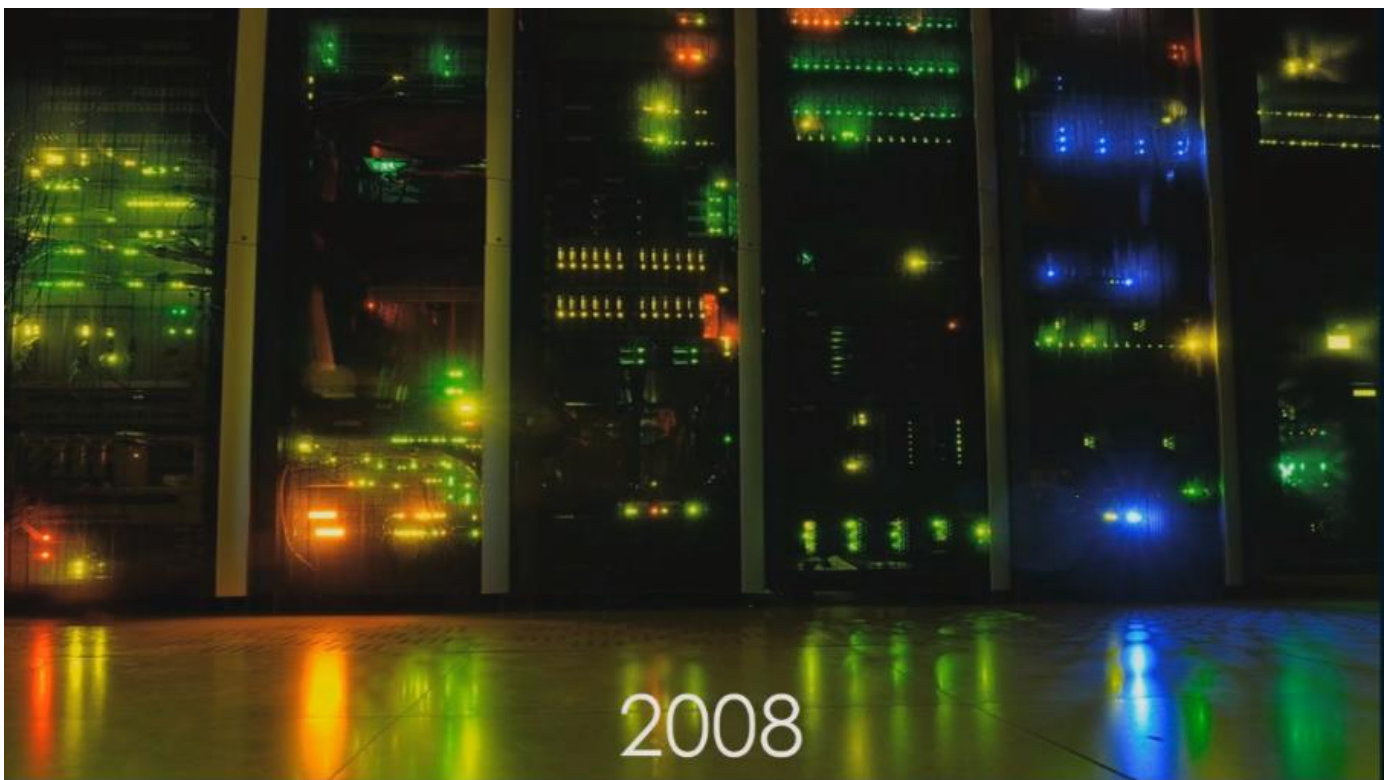
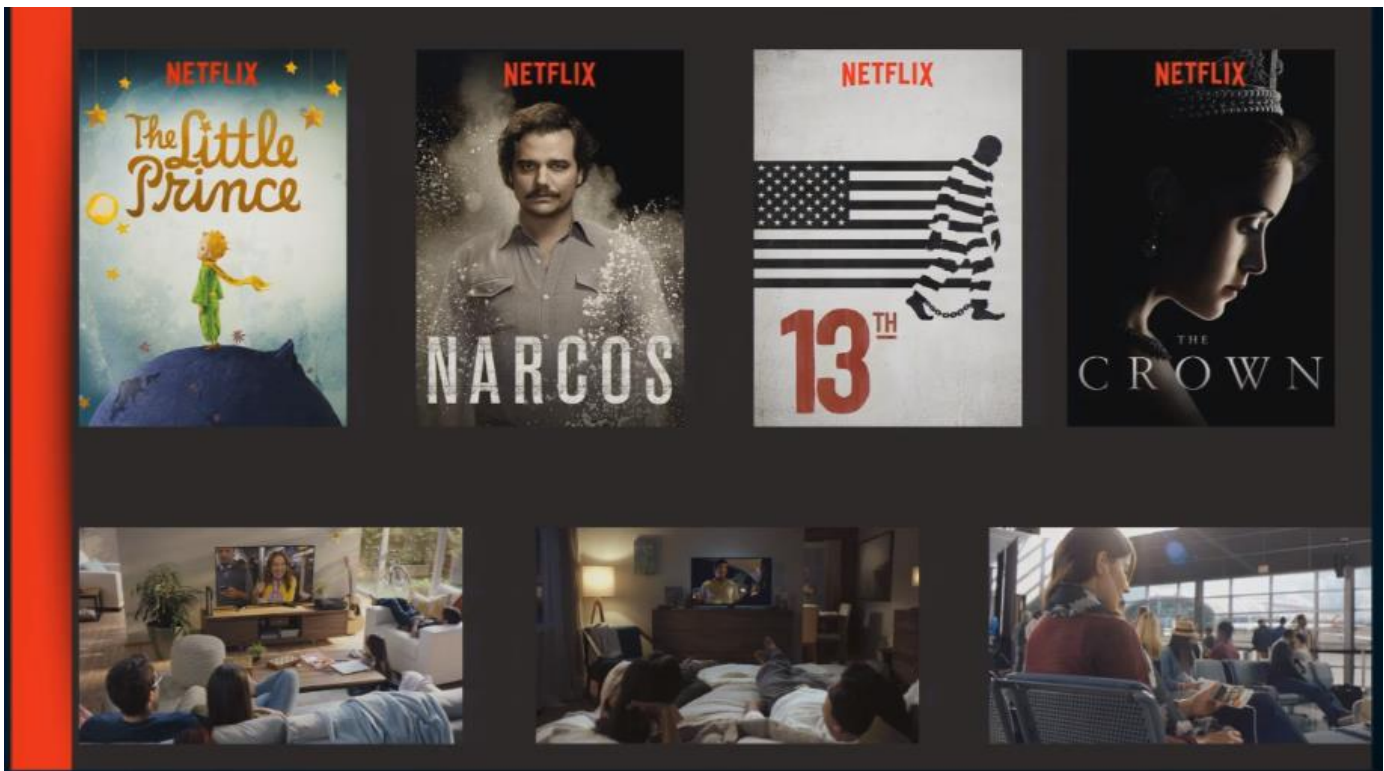


Netflix was one of the earliest very large AWS customers. By 2014, we were running hundreds of applications in Amazon EC2. That was great, until we needed to move to VPC. Given our scale, uptime requirements, and the decentralized nature of how we manage our production environment, the VPC migration (still ongoing) presented particular challenges for us and for AWS as it sought to support our move. In this talk, we discuss the starting state, our requirements and the operating principles we developed for how we wanted to drive the migration, some of the issues we ran into, and how the tight partnership with AWS helped us migrate from an EC2-Classical platform to an EC2-VPC platform.





We now need to migrate all the EC2 infrastructure we have built over the years into VPC

Why.



What.

How.

Learnings.

## VPC Advantages.

- Security.
- Networking.
- Configurability.
- Diagnostics.



Configurability is being able to form our networks the way we really want them. We can also use VPC Flow Logs for diagnostics

## Netflix Ecosystem.

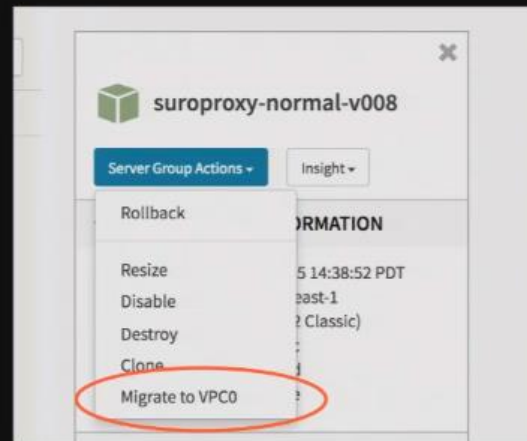
| Lots and lots.

**NETFLIX**

- 10s of critical tools.



# Delivery.



Spinnaker is the Netflix delivery system, it also needed to move into VPC without any outages and then provide other teams to move their applications also to VPC using a one-click process for moving the team pipelines, ASGs to VPC seamlessly, creating SGs and making sure they were aligned and working.

# Monitoring.



Netflix also has large real-time telemetry environment called Atlas that reports over 300 billion metrics/minute. Atlas will be used to monitor the migration effort using Dashboards telling us who is in VPC and who is not

- 10s of critical tools.
- 100s of databases & ELBs.
- 1,000s of services.
- 10,000s of instances.

## Migration Management.

If “plan A” didn’t work,  
the alphabet has 25 more letters.

**NETFLIX**

## Guiding Principles.

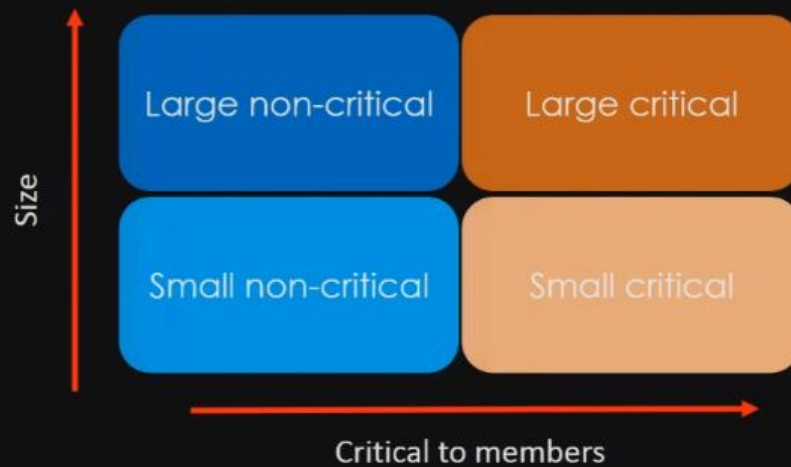
Seamless to engineers.



Velocity of innovation.

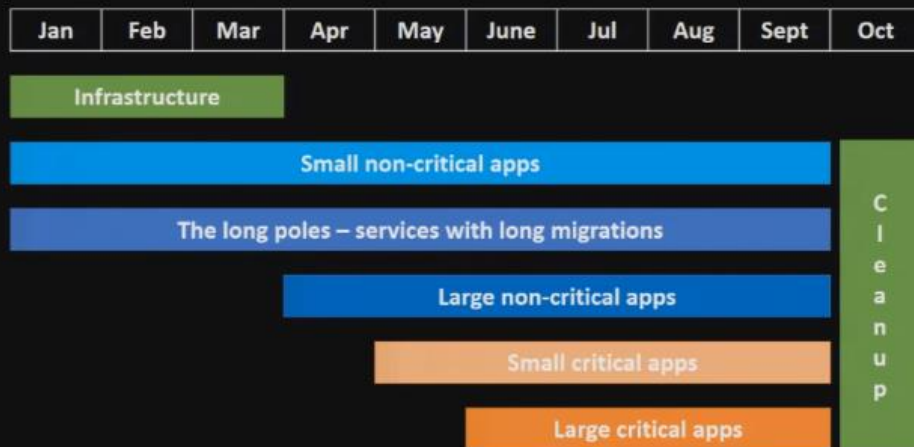
Opportunistic improvements.

# Service Classification.



An application is considered **large** if it has over 200 instances running, a service is considered **critical** if it causes a customer from being able to stream.

# 2016 – The Migration.



We cannot migrate any service during the holiday season.

# VPC Exploration.

EXPECTATIONS.  
SURPRISES.

NETFLIX

## Primary Goals.

Identify environmental differences.



Alignment on desired VPC end state.



Develop migration strategy.



## Technical Challenges.

Network Routing.



DNS.



Security Groups.



## Account Rationale.

Security compartmentalization.



Administrative Domain.



Rate Limit Restrictions.



Capacity Constraints.



# Account Classifiers.

Business Purpose.



Operational Model.

User Access.



# Regional Routing.

PACKET KUNG-FU.

**NETFLIX**

# ClassicLink.



ClassicLink is a feature that allows EC2-Classic instances the ability to communicate directly with instances in a single VPC in the same region.



# IP Addressing Allocation.

Amazon AWS utilizes 10.0.0.0/8.



Globally non-overlapping IP addresses.

Network Size.

## RFC 6598.



100.64.0.0/10 network is a reserved block to facilitate Carrier Grade Network Address Translation (CGN).

This provides over 4 million usable IP addresses that we can use for our instances, but this IP address space is not resolvable using EC2 DNS.

## IP Addressing Reservation.

Cloud IP (VPC EIP API).



ENI Auto-attach.

We then built some tools listed above to help with IP addressing reservation for teams.

## VPC Subnet Layout.

External Subnets.



Internal Subnets.

Partner Subnets.

## VPC Subnet Layout.

External Subnets.



Internal Subnets.

Partner Subnets.

/16

The largest subnet we had available was a /16 subnet.

## VPC Subnet Layout.

External Subnets.



Internal Subnets.

Partner Subnets.

/18

/18

/18

/18

The internal subnets were allocated /18 address spaces (subnet) within the larger /16 subnet.

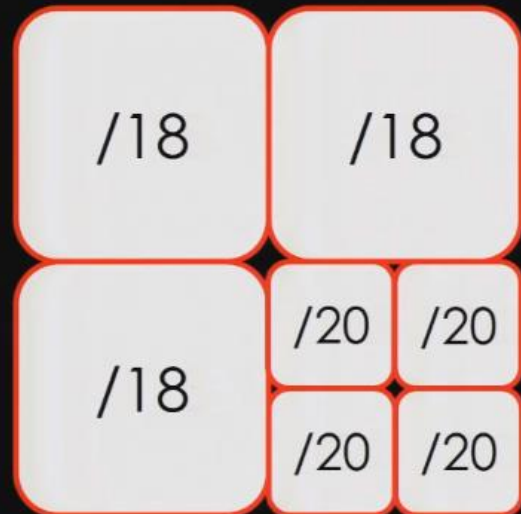
## VPC Subnet Layout.

External Subnets.



Internal Subnets.

Partner Subnets.



We then allocate /20 subnets to our external subnets with the /18 subnets

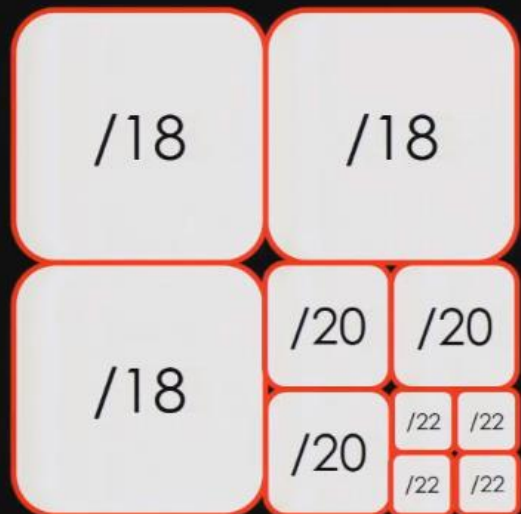
## VPC Subnet Layout.

External Subnets.



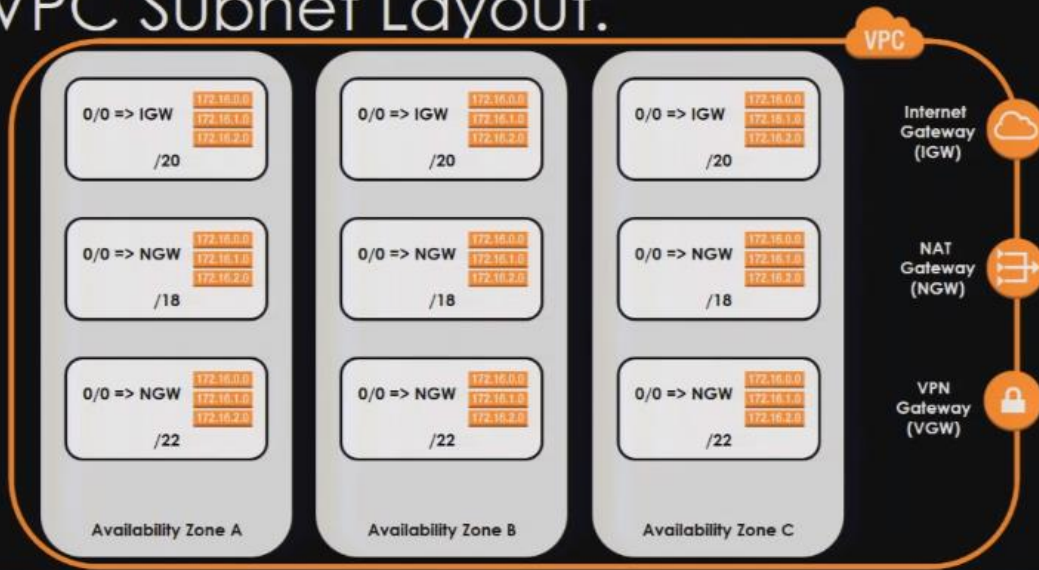
Internal Subnets.

Partner Subnets.



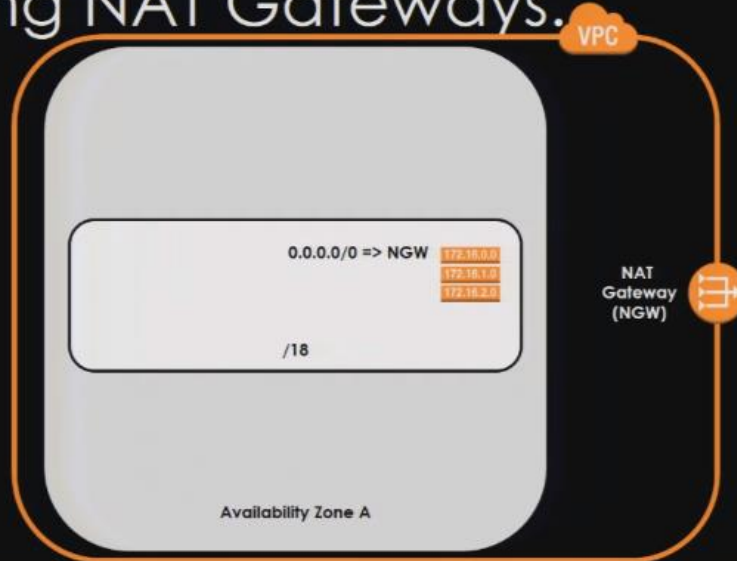
We then allocate /22 subnets to our partner subnets. This is how we maximize the use of the IP address space using our 3 AZ model

# VPC Subnet Layout.



This is a sample of our 3 AZ model. But we need to think about how we push data out from the subnet using our NAT gateways because some NATs push well over 500GB out

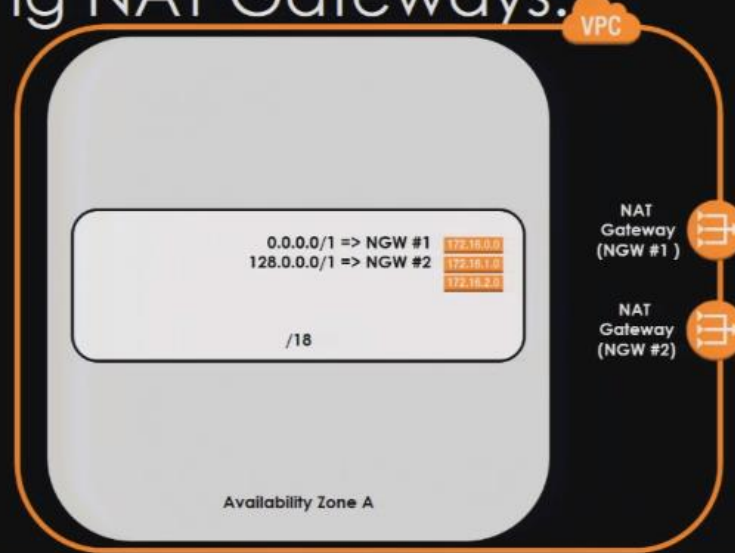
# Scaling NAT Gateways.



Can we shard the subnet?

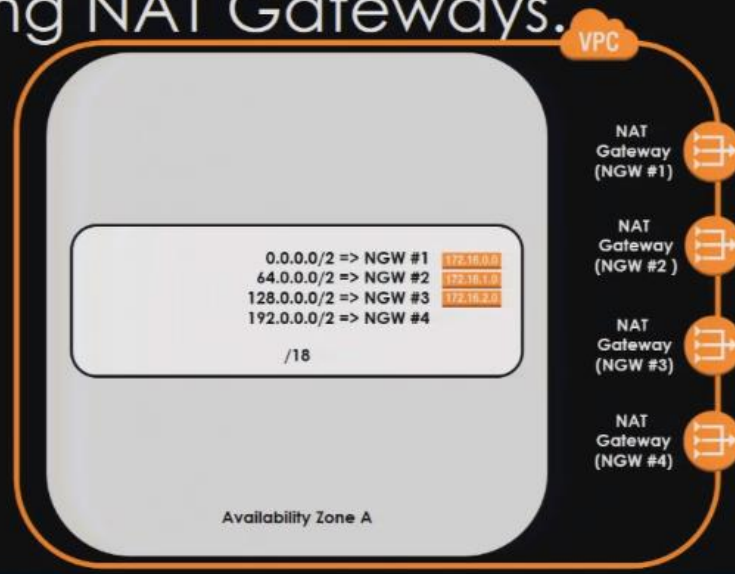


# Scaling NAT Gateways.



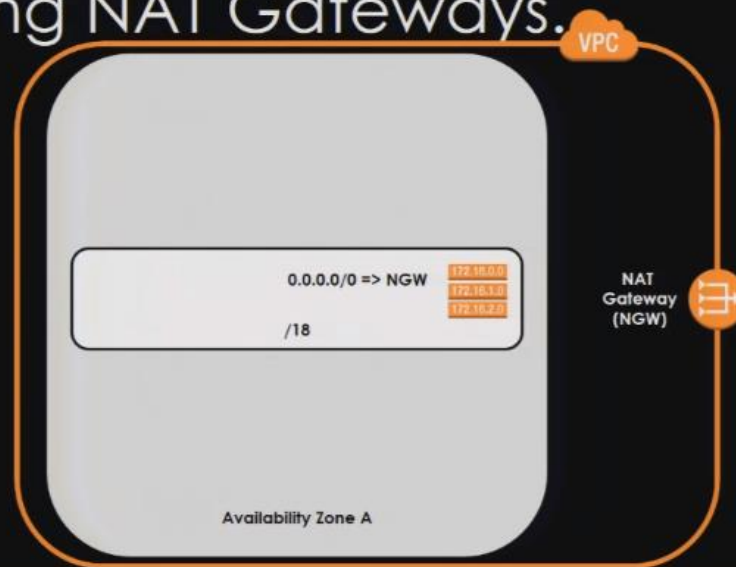
This is an approach where we are sharding the subnet by using 2 NATs with different routes out of the subnet.

# Scaling NAT Gateways.



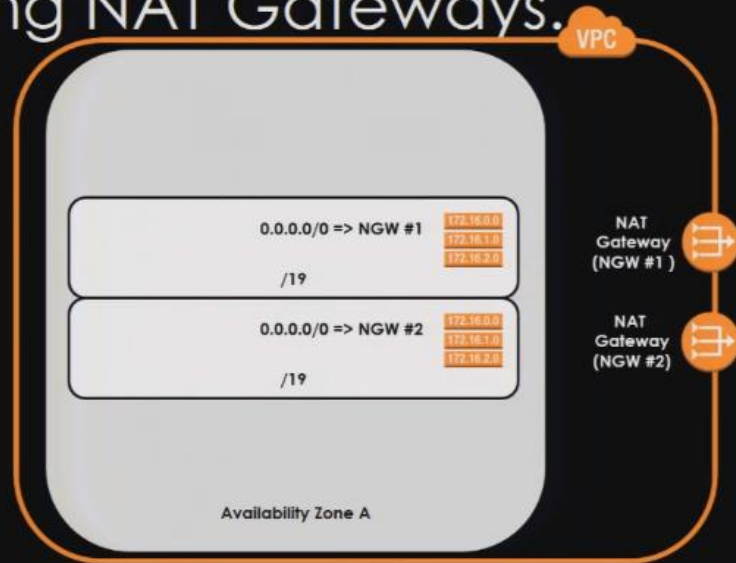
We can then add more NATs when needed to get traffic out of a heavy traffic subnet. Then we can also build some automation around this to monitor traffic. But this can create hotspot NATs.

# Scaling NAT Gateways.



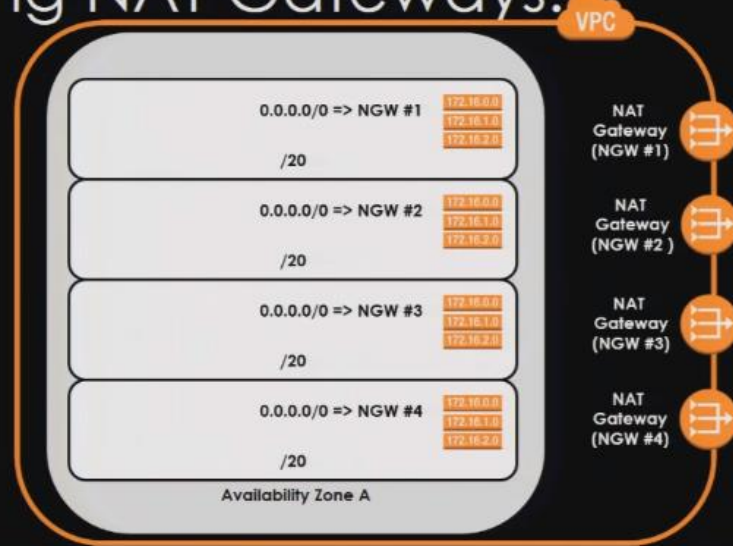
Instead of using multiple NATs for a subnet by sharding the network route for different NATs, we tried a different approach

# Scaling NAT Gateways.



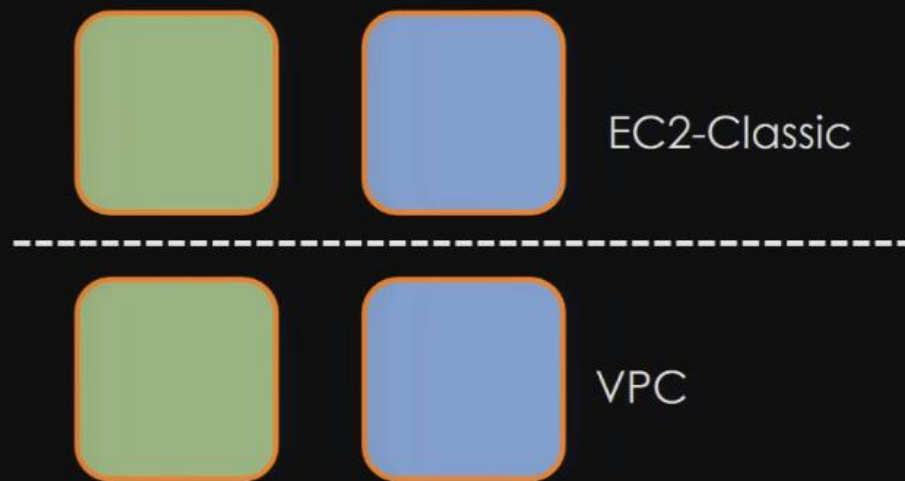
We could instead split up the subnet into 2 smaller subnets

# Scaling NAT Gateways.



Our even split it up further as above. Each subnet now has its own route table and a dedicated NAT gateway.

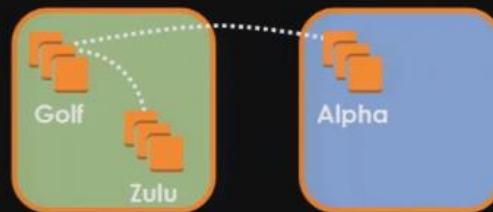
# ClassicLink.



The green and blue accounts have both a classic deployment and an VPC deployment with the same region.

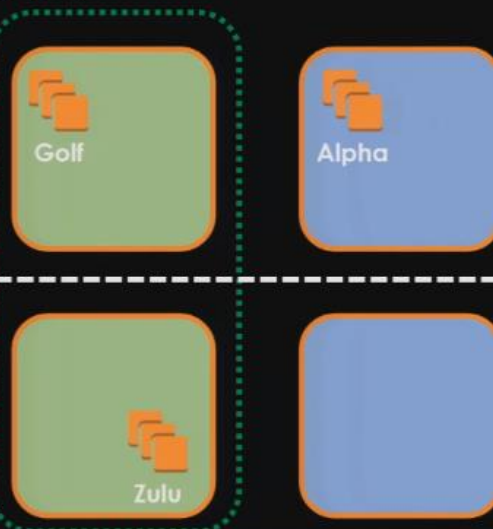
# ClassicLink.

```
gethostname(zulu.public)  
10.0.0.100  
  
gethostname(alpha.public)  
10.0.0.200
```



All service to service communication within a region in Classic are private and we can easily have communication between a green service in a green account and a blue service in a blue account as above. When you make a public DNS call to the other account, you actually get back a private address. This allows you to think about this as one big routing address domain like a 10.0.0.0/8 network

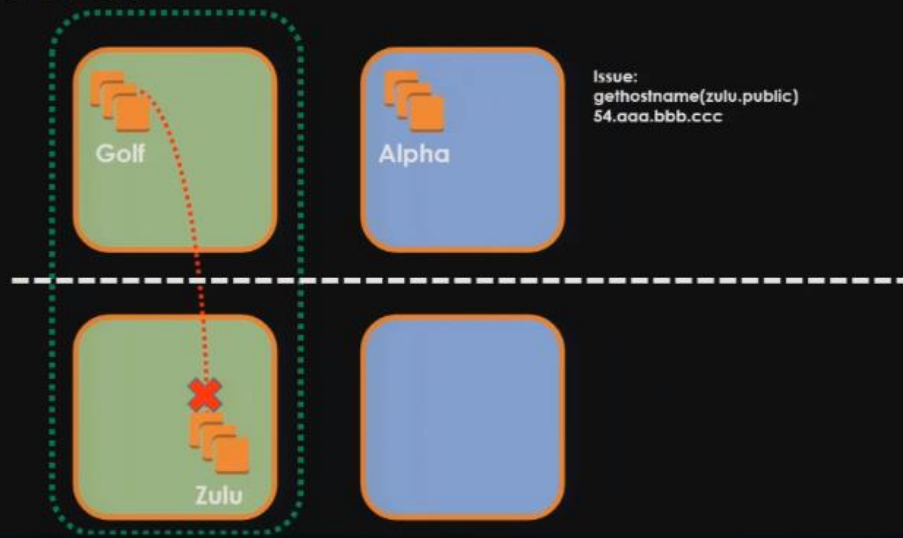
# ClassicLink.



So, when we migrate a system from classic to VPC, we expect to still be able to talk to it.



## ClassicLink.



The system wants to communicate with a system that has been migrated into VPC but got back a public address. The calling system had to go out through external DNS and was not allowed into the VPC of the migrated service because it is now external traffic.

## Service Discovery.

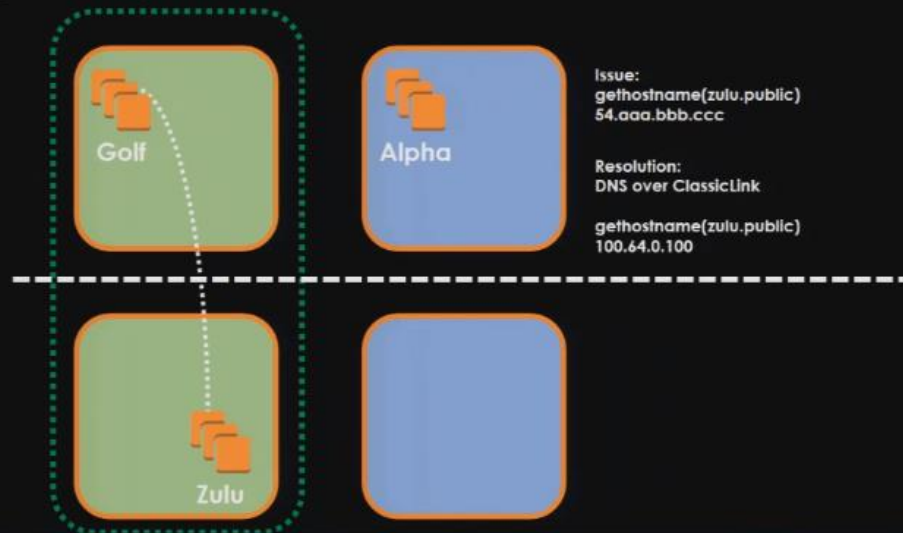


### Registration

- `hostname.public.`
- `hostname.private.`
- `ipaddress.public.`
- `ipaddress.private.`

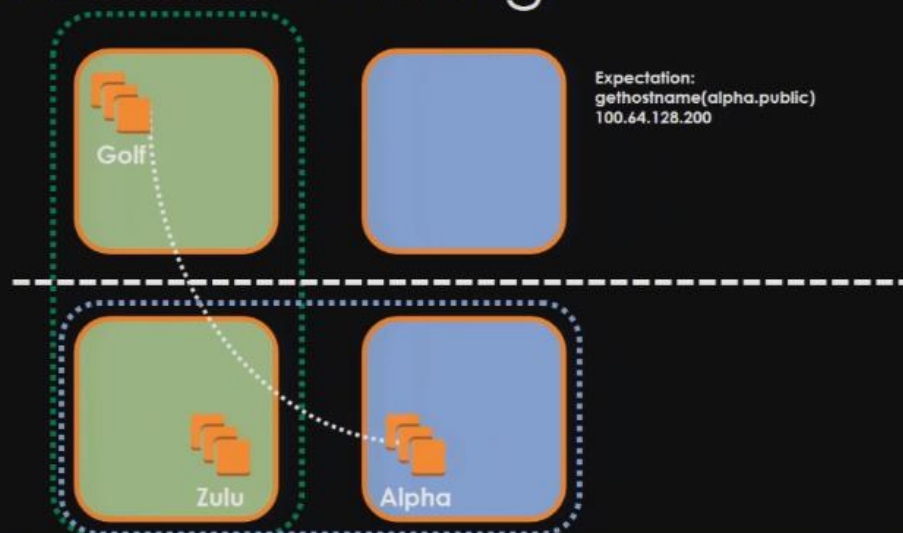
The way we address this is via service discovery. Most Netflix services use Eureka for service discovery, Eureka has the ability to capture 4 different attributes listed above. What we did was that for systems that are being migrated to VPC, rather than deploy with their public hostname into the VPC, we have them register to Eureka with their private address within the VPC. That way if a service inside of Classic attempts to talk to the service migrated to VPC, it actually gets back the private address directly.

# ClassicLink.

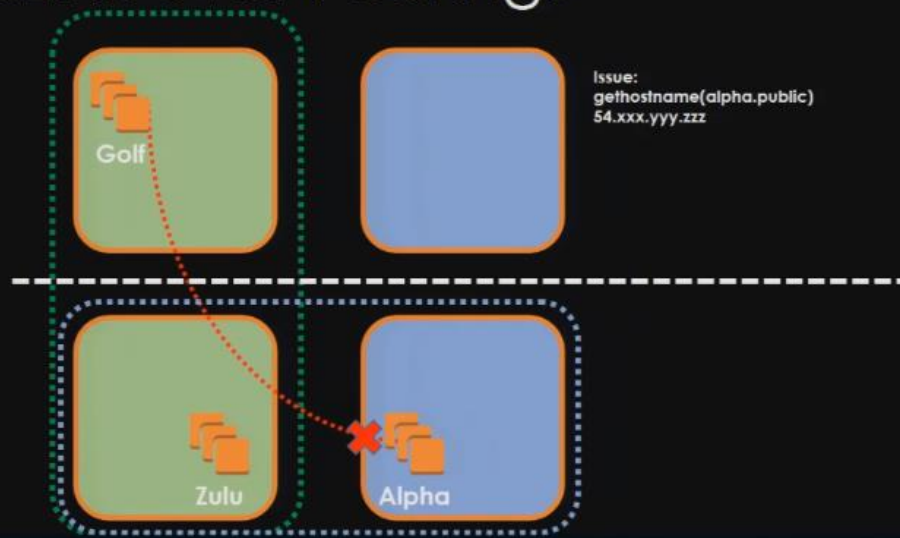


Amazon then built this feature called DNS-Over-Classic-Link

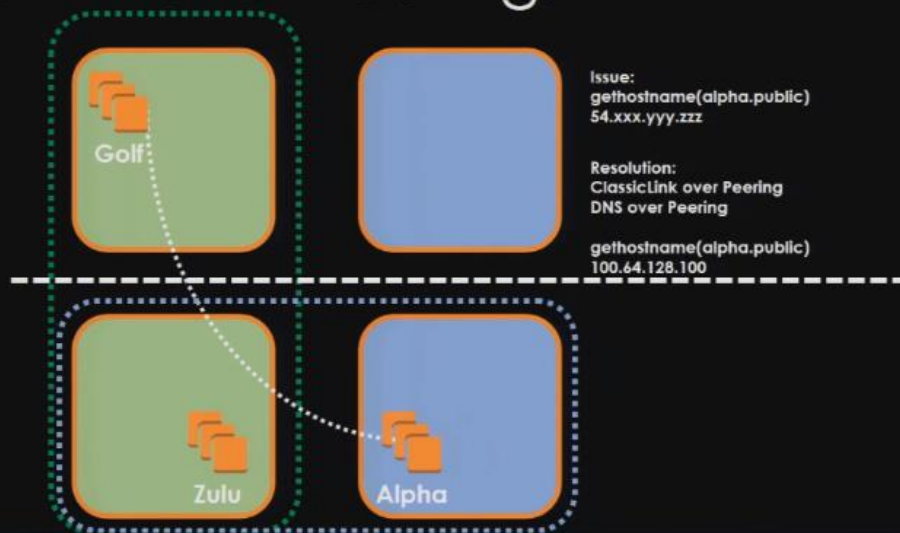
# ClassicLink over Peering.



# ClassicLink over Peering.



# ClassicLink over Peering.



## ClassicLink Everywhere.

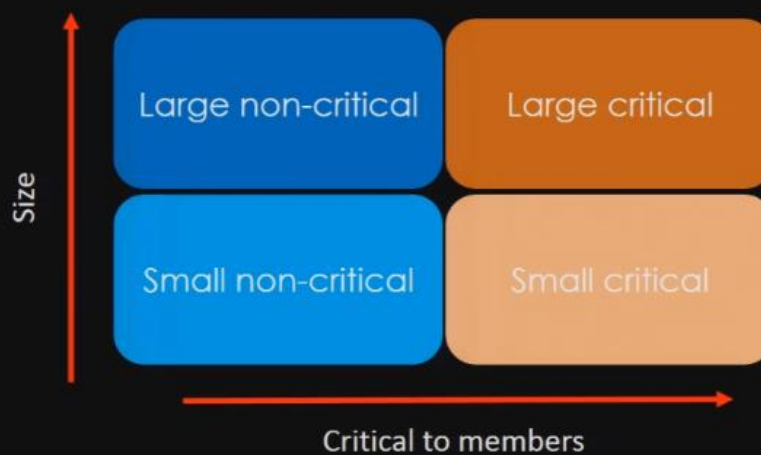


We ended up creating a fully routable mesh as a single domain for all possible communication flows

## ClassicLink at Scale.



## Service Classification.





# Dependency Mappings.

Flow Collection.



IP Metadata.

Flow Analysis.

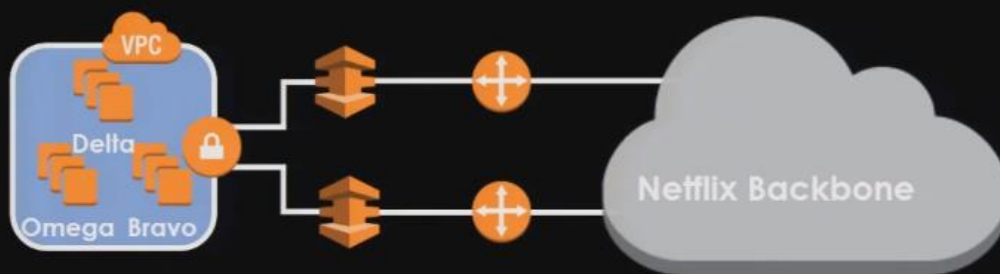


## Global Routing

MORE PACKET KUNG-FU.

NETFLIX

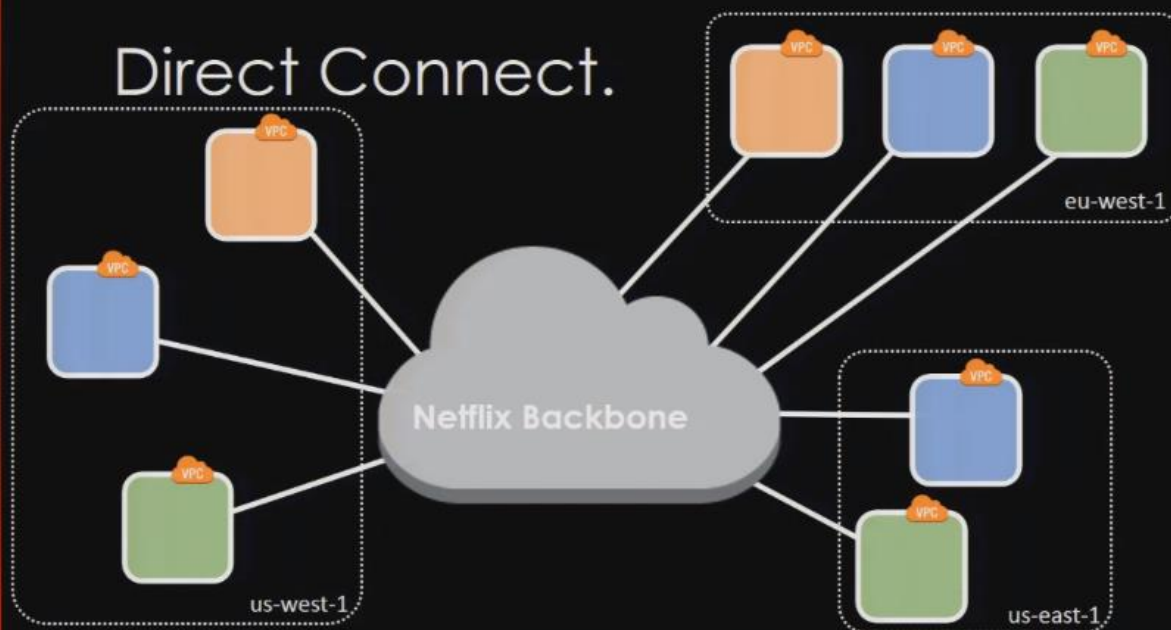
## AWS Direct Connect.



## Global Backbone.



## Direct Connect.

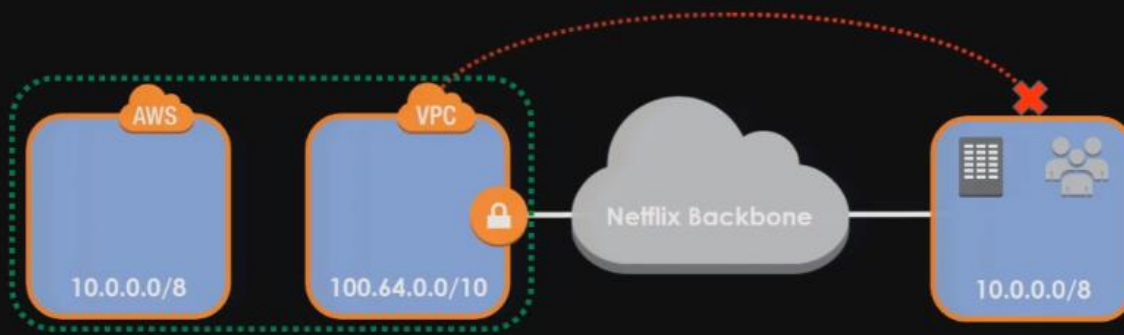


## Backbone Traffic.



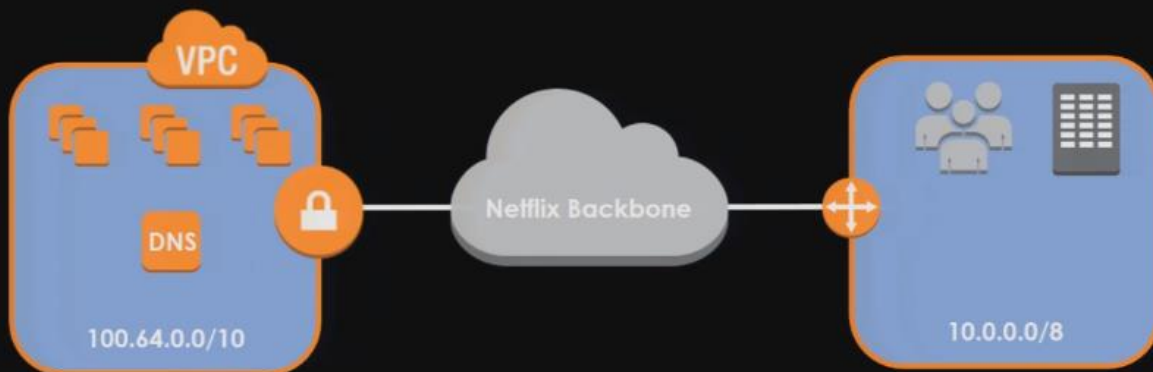
We have labs that use services in AWS as above

## Backbone Traffic.



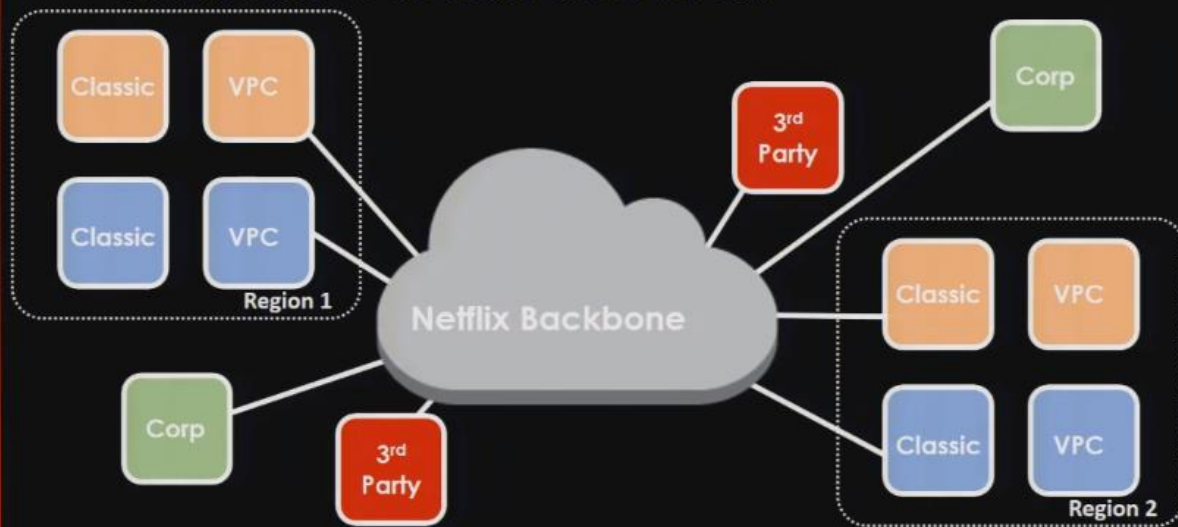
The Labs can no longer communicate with migrated services

## Backbone Traffic.



What we did was to modify the incoming Lab requests

## Global Infrastructure.



# Retrospective.


MULLIGANS.  
SECOND CHANCES.

NETFLIX

## Lessons Learned.

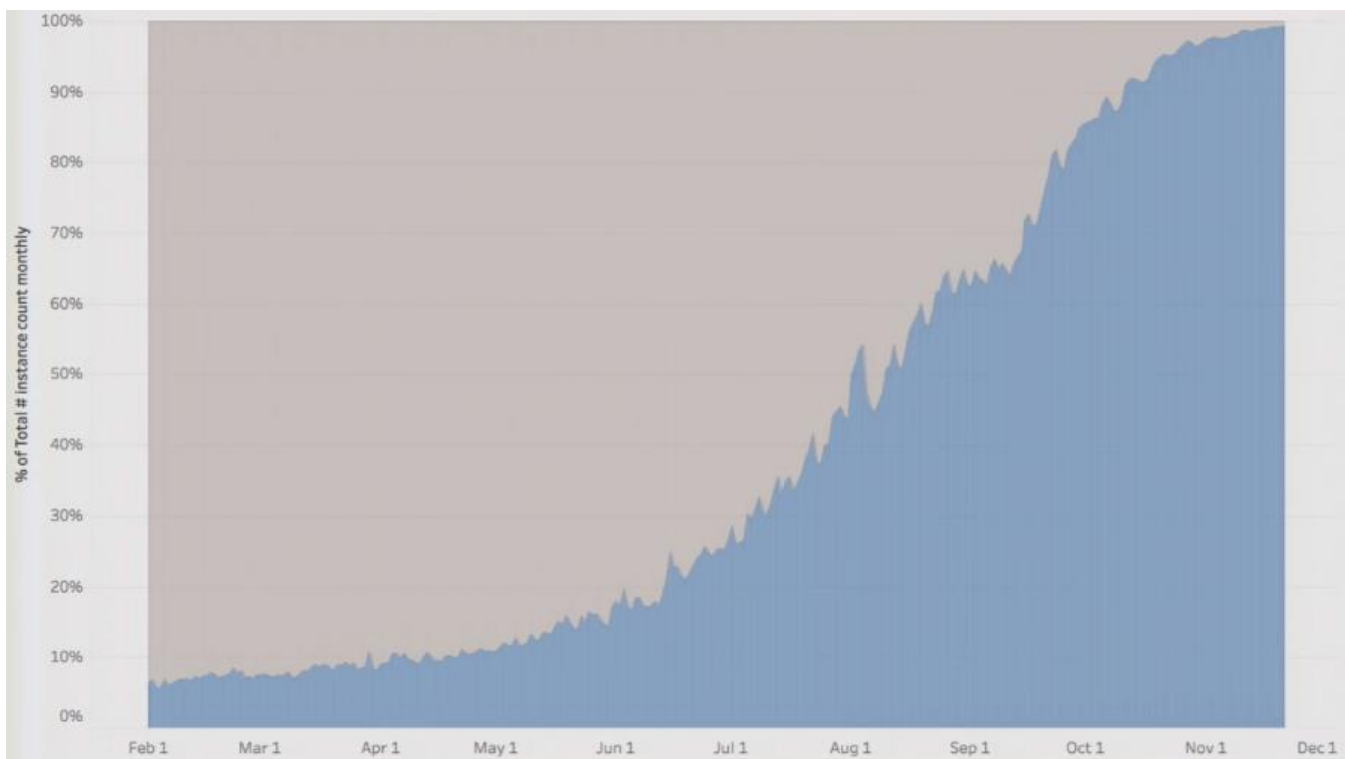
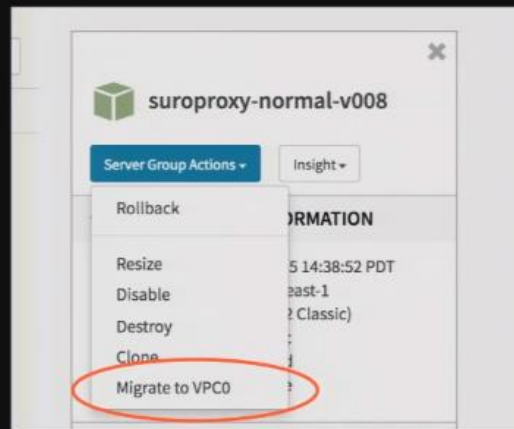
- 
- IP address scheme.
  - Traffic patterns.
  - Partner engagement.
  - Technical debt.

## Features.

- 
- ClassicLink.
  - ClassicLink over Peering.
  - DNS over Peering.
  - EC2 DNS for non-RFC 1918.



# Delivery.



AWS  
re:Invent

Thank you!