



# Firewall Networks (FireNet)

ACE Solutions Architecture Team



# Firewalling in the Cloud

## Layered Security Model - Defense in depth

### Intra-Spoke Traffic – Distributed



- Best handled by L4 SG/NSGs (Aviatrix can do it)
- Lifecycle tied with compute instances (CI/CD)
- Provisioned/de-provisioned by automation

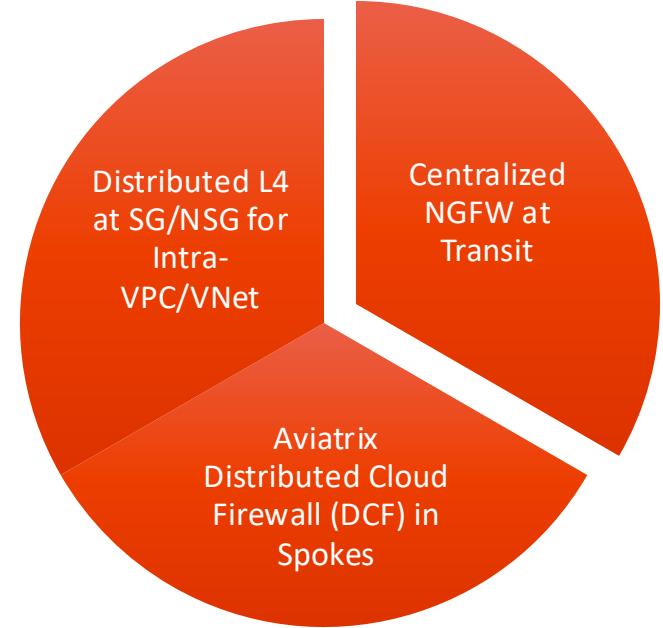
### E/W and N/S Inspection – Distributed



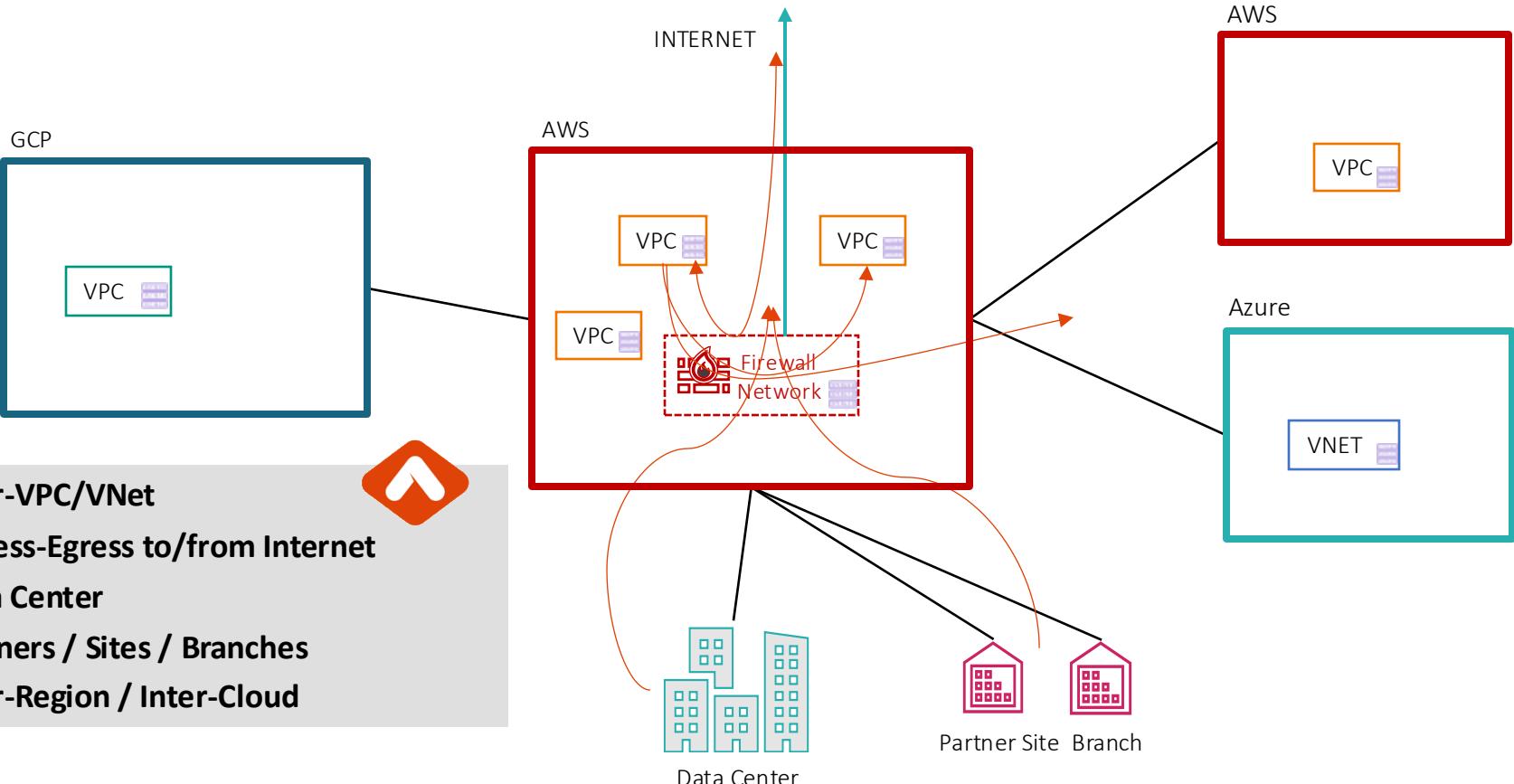
- Handle via Aviatrix L7 Firewall
- Relieves 3<sup>rd</sup> party Firewalls from processing
- Reduces cost by fewer egress charges and expensive firewalls
- Close to applications/workloads

### E/W and N/S Inspection – Centralized

- Best handled via a 3<sup>rd</sup> Party Firewall
- Aviatrix provides Service Insertion, policy, and life-cycle management of 3<sup>rd</sup> party firewalls.
- Centralized security policy from the 3<sup>rd</sup> party firewall platform



# Centralized Firewall Traffic Patterns



# Challenges of Service Insertion in the Cloud

## Firewall Vendors

- Firewall vendors have repackaged on-prem solutions to cloud
- Not focused to solve cloud networking and challenges
- Expect customer to own routing traffic to and from FWs

## Cloud Provider

- Solution which might lack enterprise features you need
- Expect customer to figure out routing traffic to and from FWs
- Lack of visibility and troubleshooting tools

## Customer

- Manually figure out routing and troubleshooting
- Many components involved that require individual config/ops (LB, NAT GWs, routes, etc.)
- Lack of visibility and troubleshooting tools reduces efficiency and increases risk



# Aviatrix Transit FireNet

# Aviatrix Encrypted Transit Firewall Network



Scale out, multi-AZ FW deployments



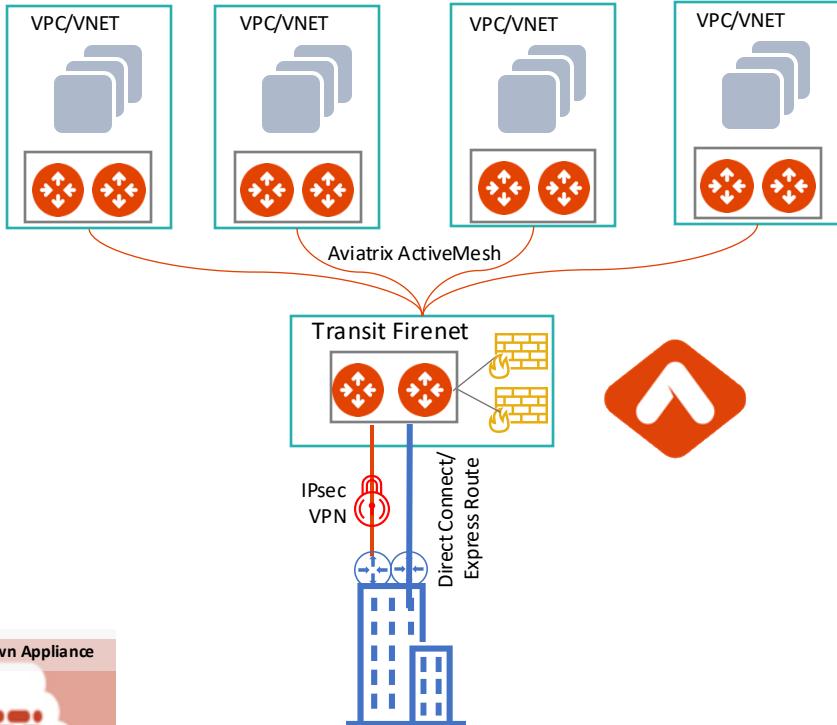
Automated route management  
Segmentation and Connection Policies



Deep visibility and operational capabilities

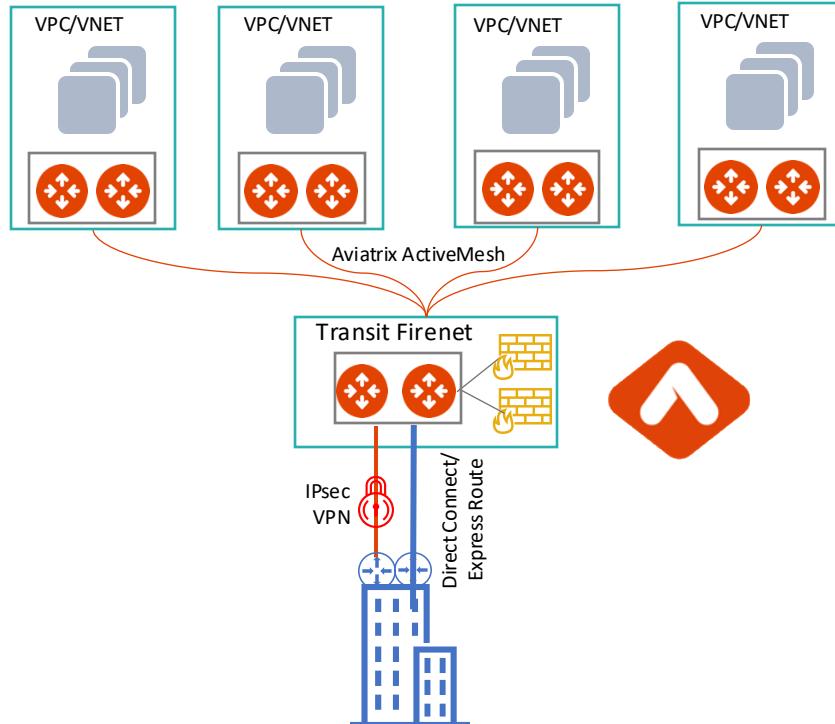


Repeatable architecture, across regions  
and clouds



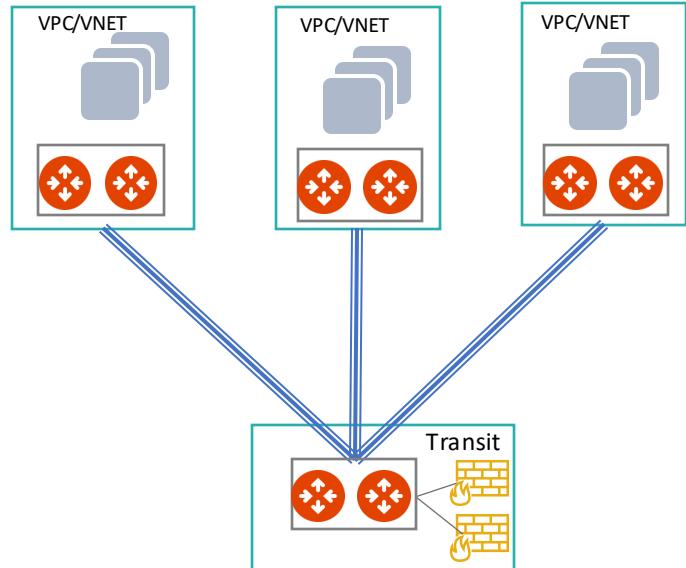
# Aviatrix Encrypted Transit Firewall Network

1. From the Aviatrix CoPilot, you enable FireNet (or through TF)
2. Aviatrix Controller deploys the Firewalls.
3. Aviatrix Controller configures the interfaces and routing entries on the firewalls.
4. No SNAT, IPsec, BGP, or other elements are required to insert the FWs into the traffic path.
5. Aviatrix Controller ensures all the Spokes and other connected networks marked for inspection have their traffic inspected by the FW.
6. Aviatrix Controller monitors the health of the FW instances and ensures the traffic is only forwarded to “up” Firewalls.
7. Aviatrix GWs or a native cloud LB incorporated into the overall design ensures the traffic is correctly load-balanced to all available FWs while maintaining the session stickiness.

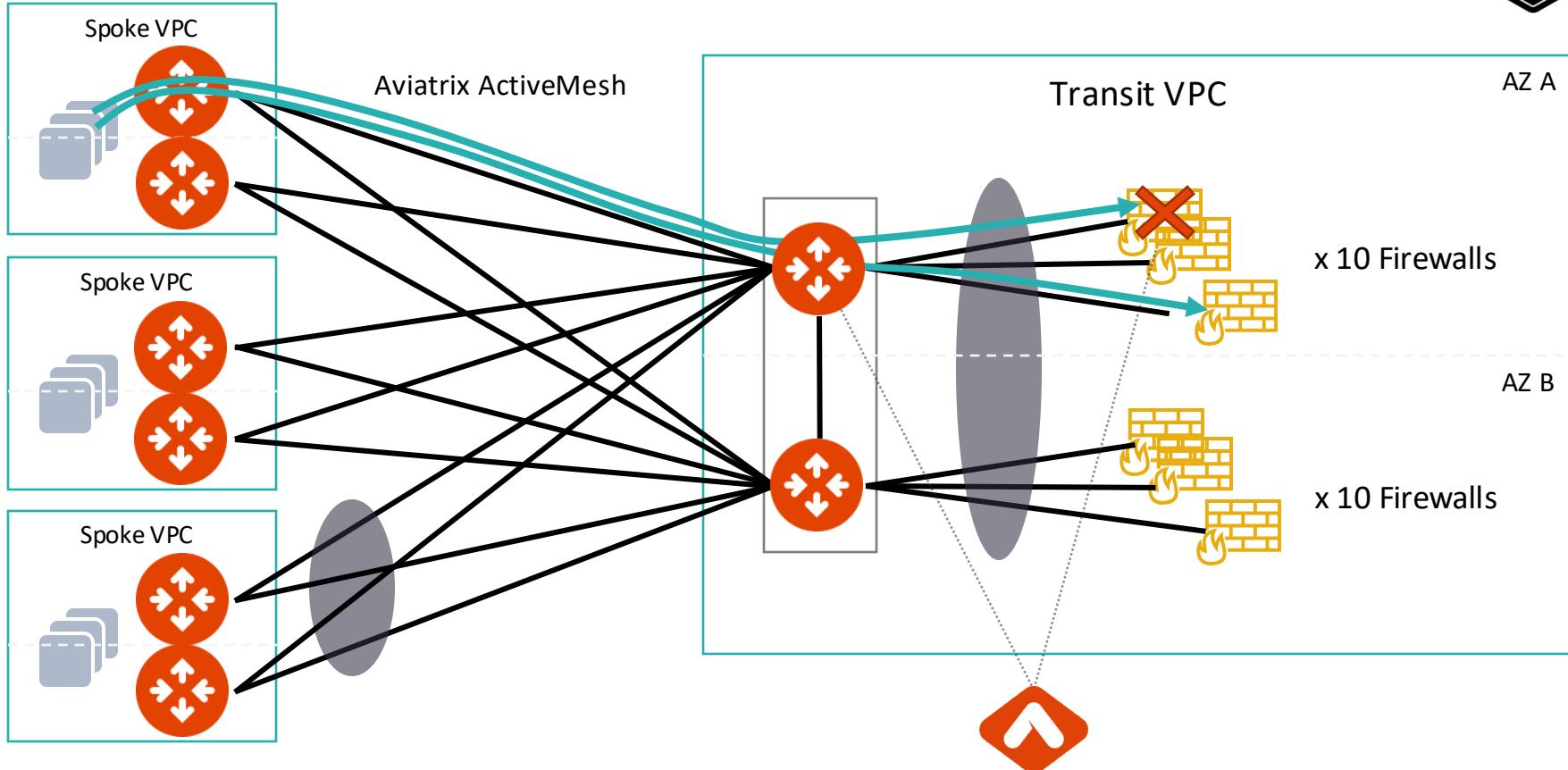


# Aviatrix Transit FireNet Load Balancing and Failover

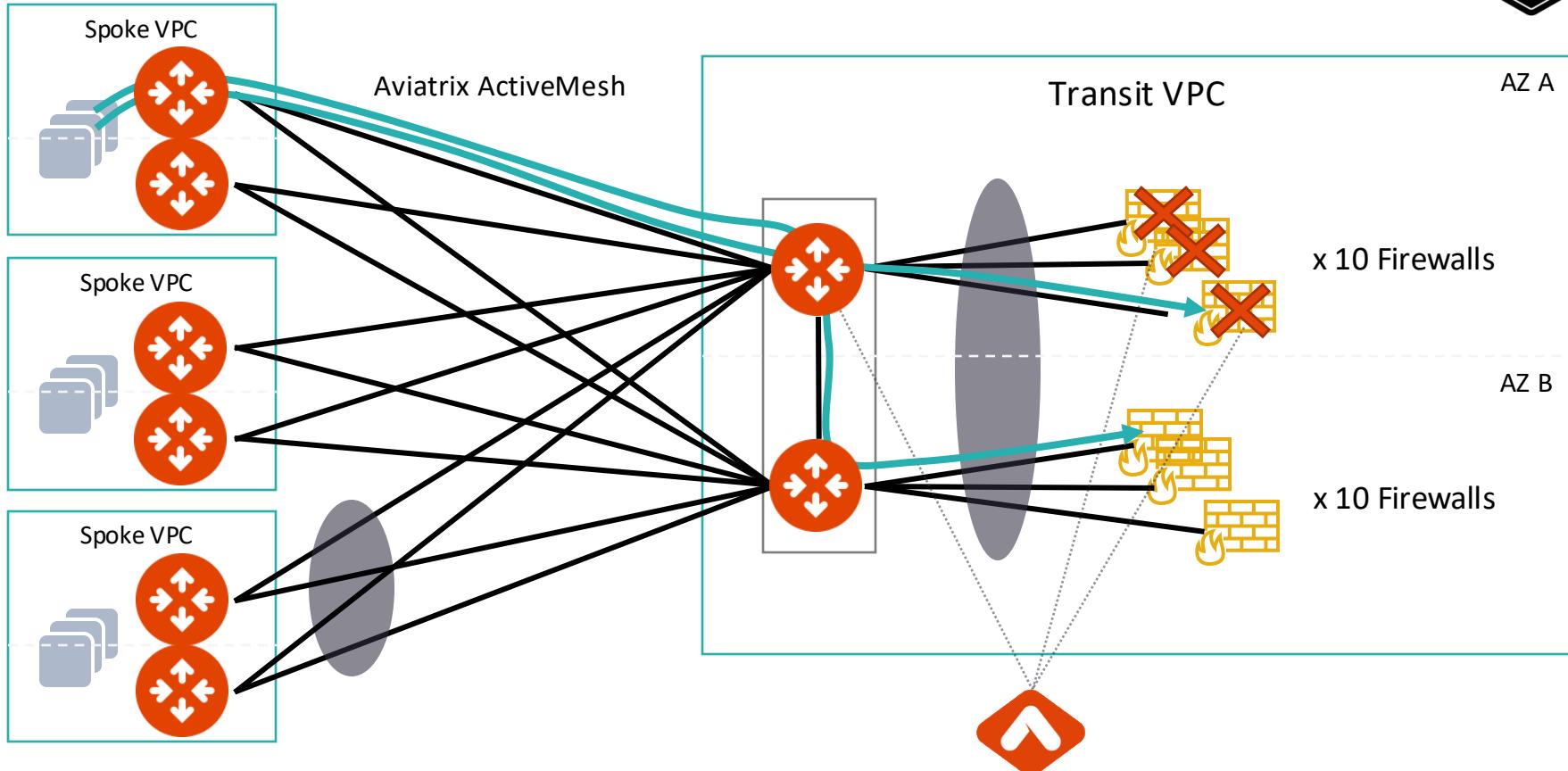
- The Aviatrix Controller monitors the health of the Firewalls
- Controller periodically checks Firewall instance health
- **Session stickiness** is maintained
  - Existing sessions on working firewalls are not disturbed
  - AKA resilient hashing



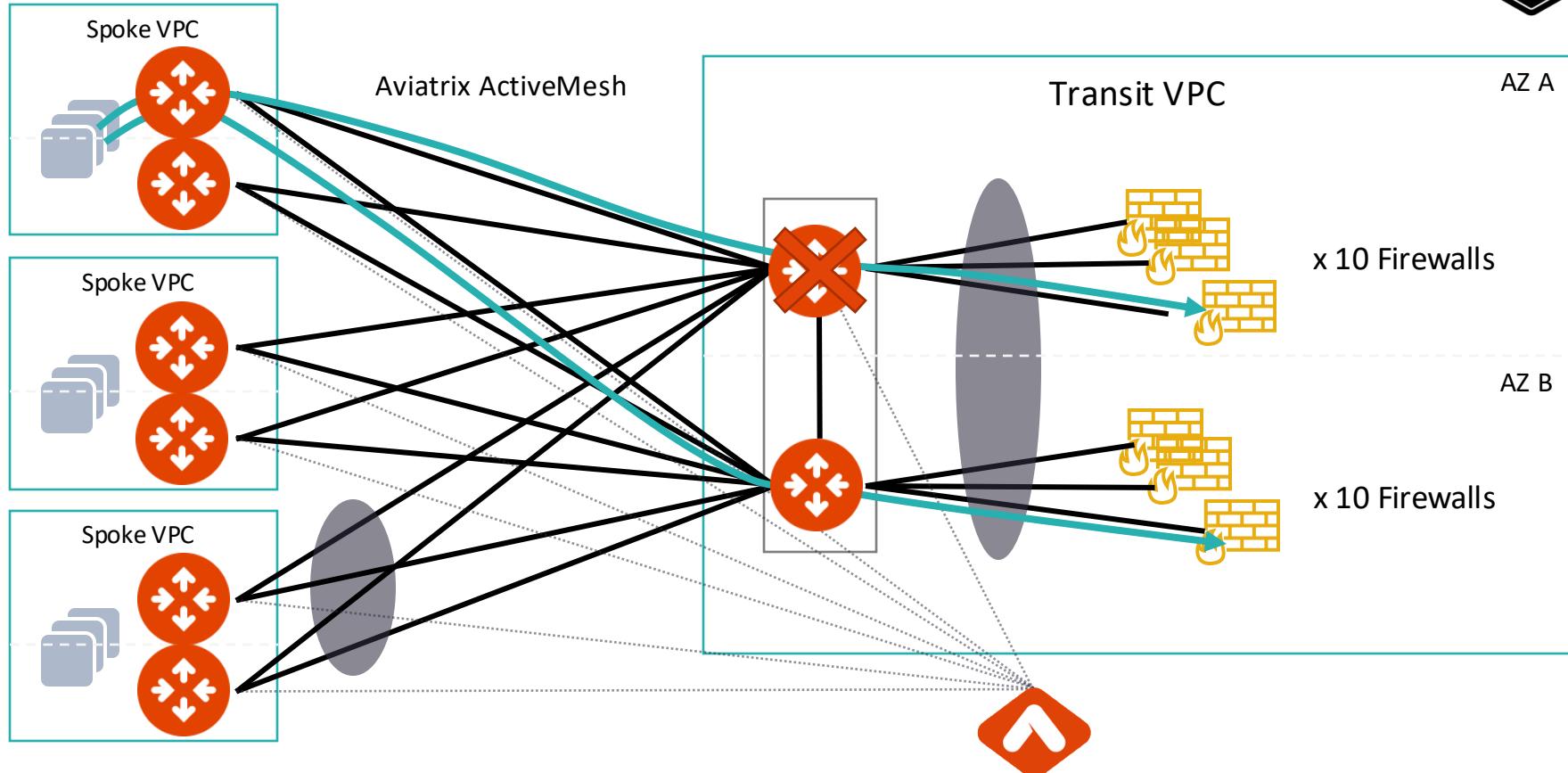
# Aviatrix Transit FireNet Failover



# Aviatrix Transit FireNet Failover

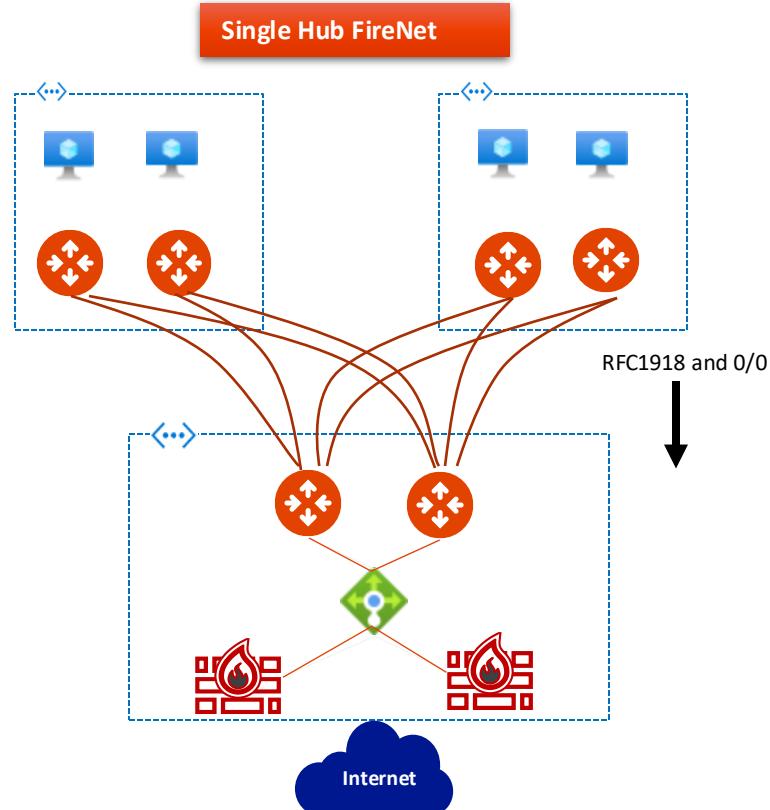
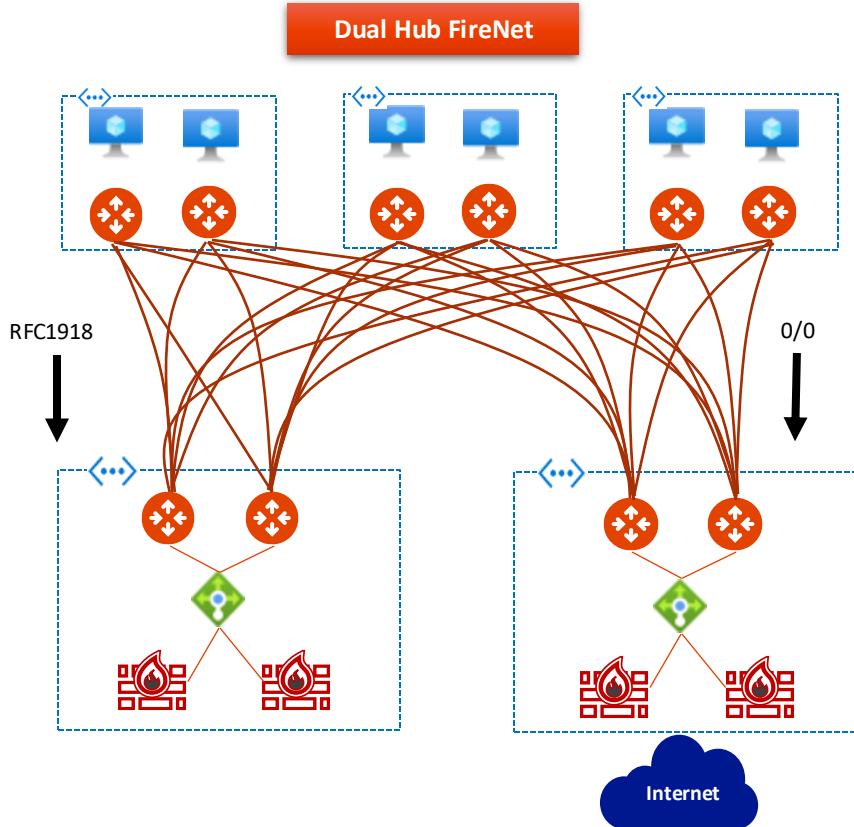


# Aviatrix Transit FireNet Failover



# FireNet Architecture Options (Azure Example)

Each firewall set can scale independently based on need



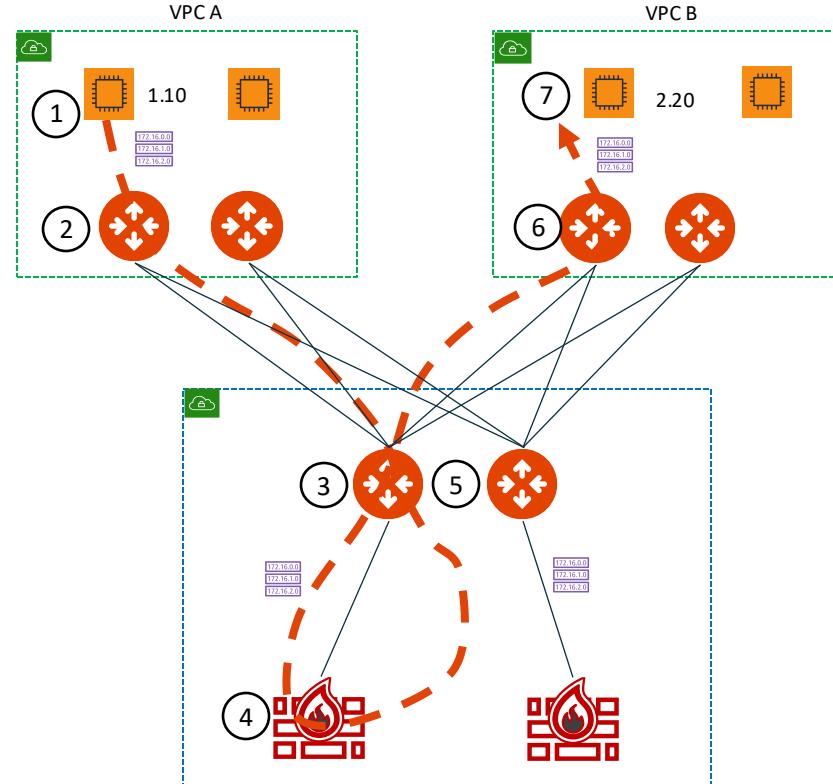


# Aviatrix Transit FireNet Packet Walk

# FireNet Packet Walkthrough – AWS Example

A Host 1.10 communicating with 2.20 with VPC A inspected via FireNet

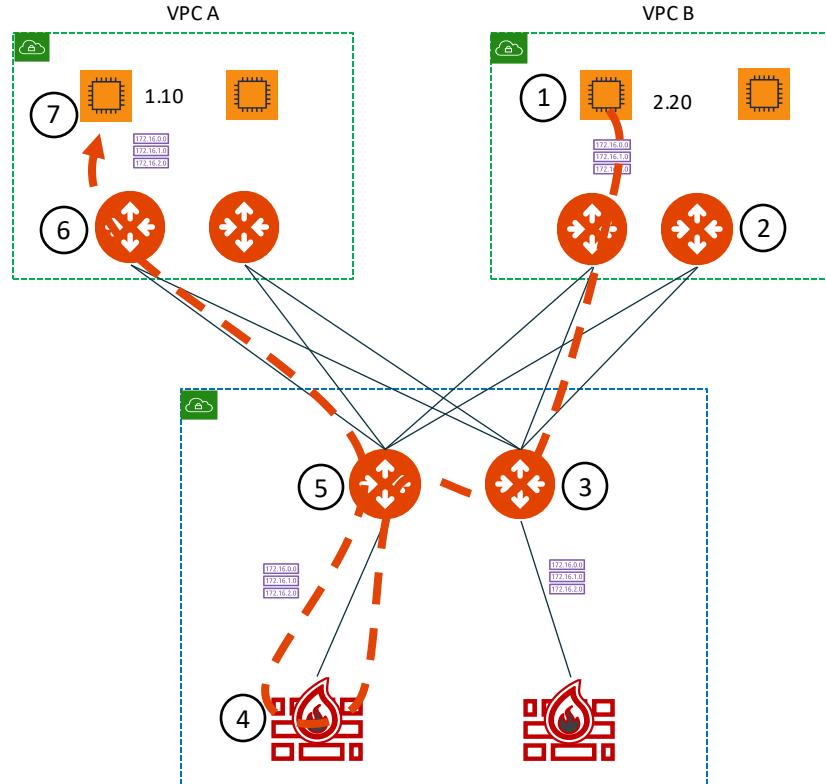
1. The local route table for 1.10 has RFC1918 routes pointed to its local gateway
2. The local Aviatrix spoke gateway will ECMP traffic with 5-tuple hash to one of the Aviatrix Transit Gateways
3. The Aviatrix Transit Gateway receiving the flow will check PBR rules to determine if either source or destination requires FireNet. If a match, traffic is redirected to the one of the available FWs (it can be in the same AZ or a different AZ – when it's in a different AZ, Transit GW sends the traffic first to the other Transit GW).
4. The Firewall selected will process the packet and send the traffic back to its local Transit Gateway.
5. The Aviatrix Transit Gateway will receive the processed packet and PBR this traffic back into the egress interface and ECMP traffic with 5-tuple hash towards the destination spokes.
6. The spoke gateway will receive the traffic and route the traffic out its local interface to the VPC route table. Note that this GW may not be in the same AZ as the destination instance.
7. The destination will receive the original traffic and see this as native VPC communication flow.



# FireNet Packet Walkthrough – AWS Example

## Return Flow: 1.10 communicating with 2.20 with VPC A inspected via FireNet

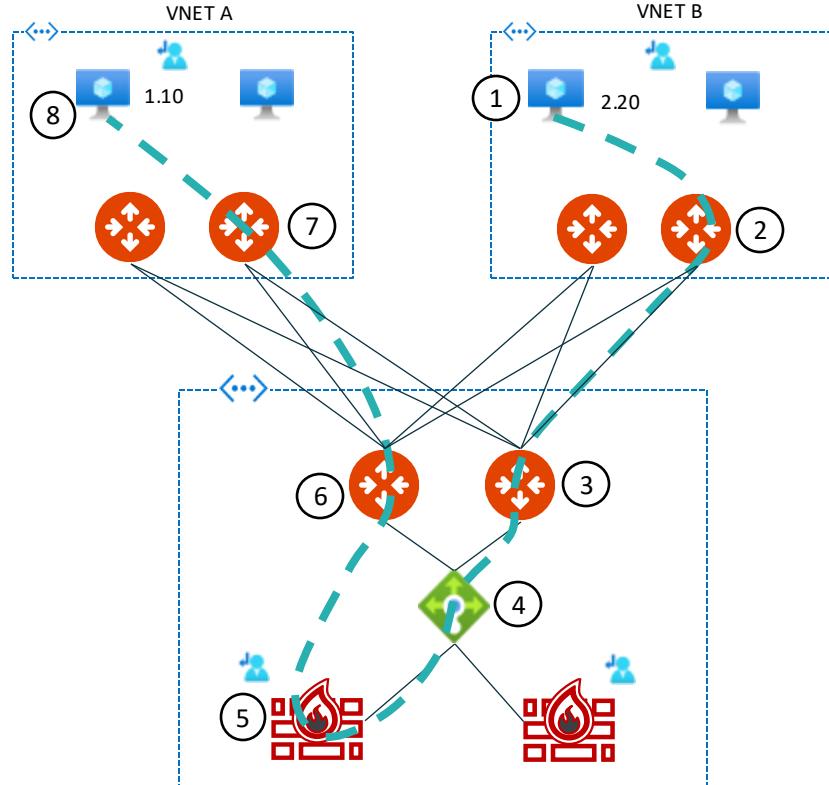
1. The local route table for 2.20 has RFC1918 routes pointed to its local spoke gateway for return traffic.
2. The local Aviatrix spoke gateway will ECMP traffic with 5-tuple hash to one of the Aviatrix Transit Gateways
3. The Aviatrix Transit Gateway receiving the traffic will pass the traffic to the the same FW which handled the initial flow to maintain symmetry (directly or via another Transit GW).
4. The stateful Firewall will process the return traffic and route the traffic back to its designated gateway.
5. The Aviatrix gateway will ECMP traffic with 5-tuple hash to one of the destination spoke gateways.
6. The destination spoke gateway will route this traffic out its local interface to the native VPC route table.
7. The original source will receive the return traffic and see this as native VPC communication flow.



# FireNet Packet Walkthrough – Azure Example

**Return Flow: 1.10 communicating with 2.20 with VNET A inspected via FireNet**

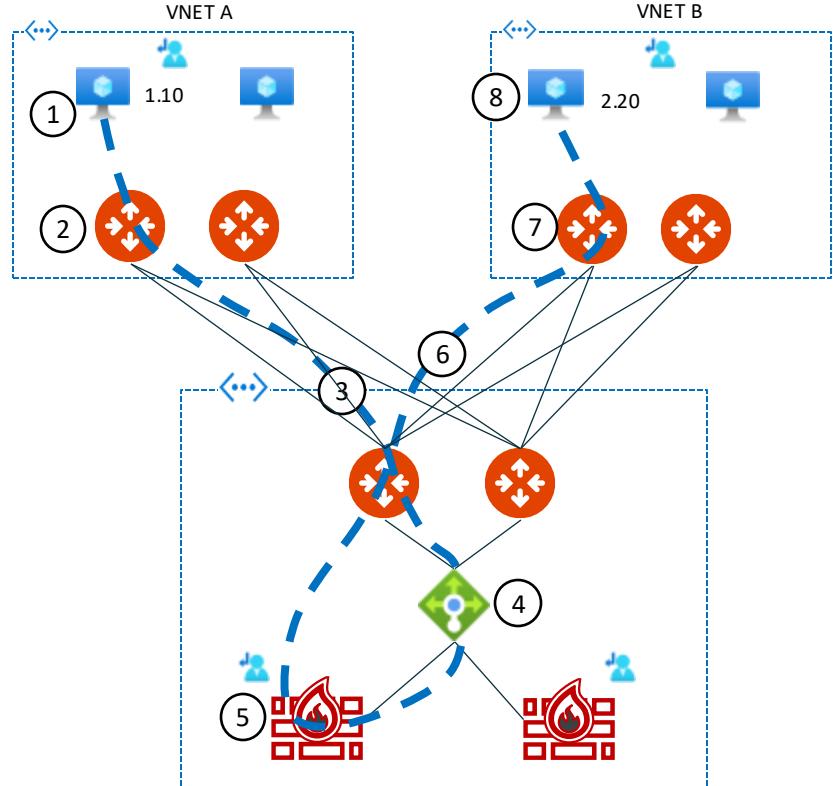
1. The local route table for 2.20 has RFC1918 routes pointed to its local spoke gateway for return traffic.
2. The local Aviatrix spoke gateway will ECMP traffic with 5-tuple hash to one of the Aviatrix Transit Gateways
3. The Aviatrix Transit Gateway receiving the traffic will pass the traffic to the ILB. The gateway will PBR the traffic back to the ILB for FireNet.
4. The Azure load balancer will hash the traffic however, the reverse flow hash will match the initial flow to ensure symmetry.
5. The stateful Firewall will process the return traffic and route the traffic back to its designated gateway.
6. The Aviatrix gateway will ECMP traffic with 5-tuple hash to one of the destination spoke gateways.
7. The destination spoke gateway will route this traffic out its local interface to the native Azure route table.
8. The original source will receive the return traffic and see this as native Azure communication flows.



# FireNet Packet Walkthrough – Azure Example

## A Host 1.10 communicating with 2.20 with VNET A inspected via FireNet

1. The local route table for 1.10 has RFC1918 routes pointed to its local gateway
2. The local Aviatrix spoke gateway will ECMP traffic with 5-tuple hash to one of the Aviatrix Transit Gateways
3. The Aviatrix Transit Gateway receiving the flow will check PBR rules to determine if either source or destination requires FireNet. If a match, traffic is redirected to Azure ILB.
4. The Azure ILB will perform a 5-tuple hash to send the traffic to one of the backend pool members.
5. The Firewall selected will process the packet and send the traffic back to its defined Transit Gateway.
6. The Aviatrix Transit Gateway will receive the processed packet and PBR this traffic back into the egress interface and ECMP traffic with 5-tuple hash towards the destination spokes.
7. The spoke gateway will receive the traffic and route the traffic out its local interface to the Azure VNET route table.
8. The destination will receive the original traffic and see this as native Azure communication flows.





Next: Lab 6 - FireNet