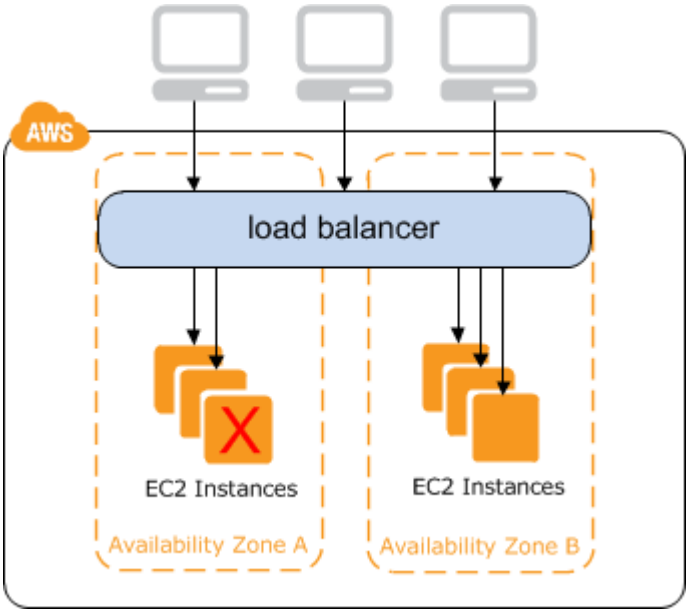
ELASTIC LOAD BALANCER (ELB)

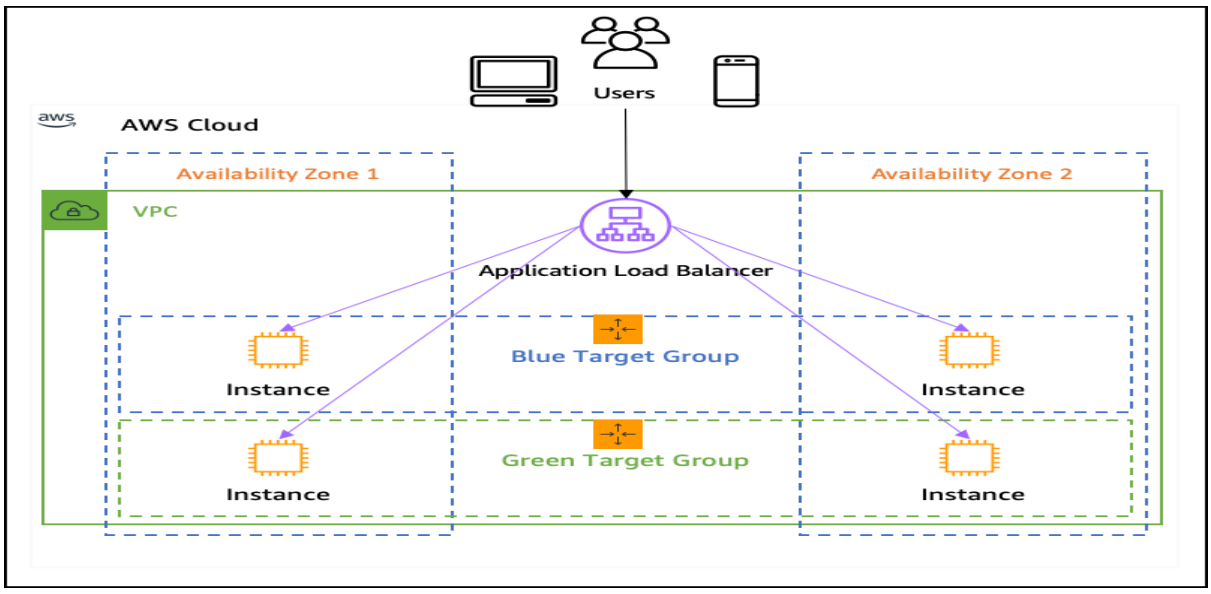
- Load balancer distributes the web traffic to the available server or Load balancing refers to efficient distributing incoming traffic across a group of backendserver.
- Load Balancer is of 4 types:

1. Classic Load Balancer
2. Application Load Balancer
3. Network Load Balancer
4. Gateway Load Balancer

1. Classic Load Balancer

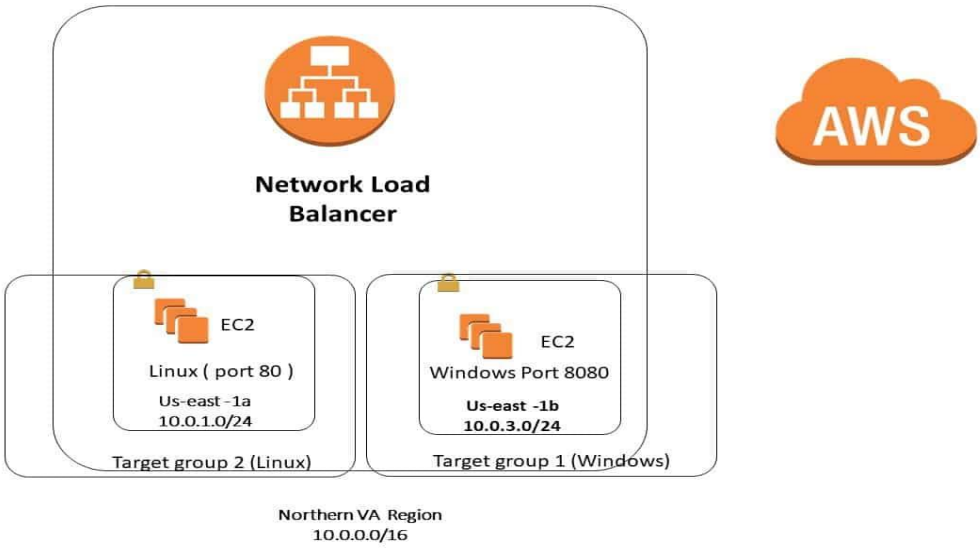


2. Application Load Balancer

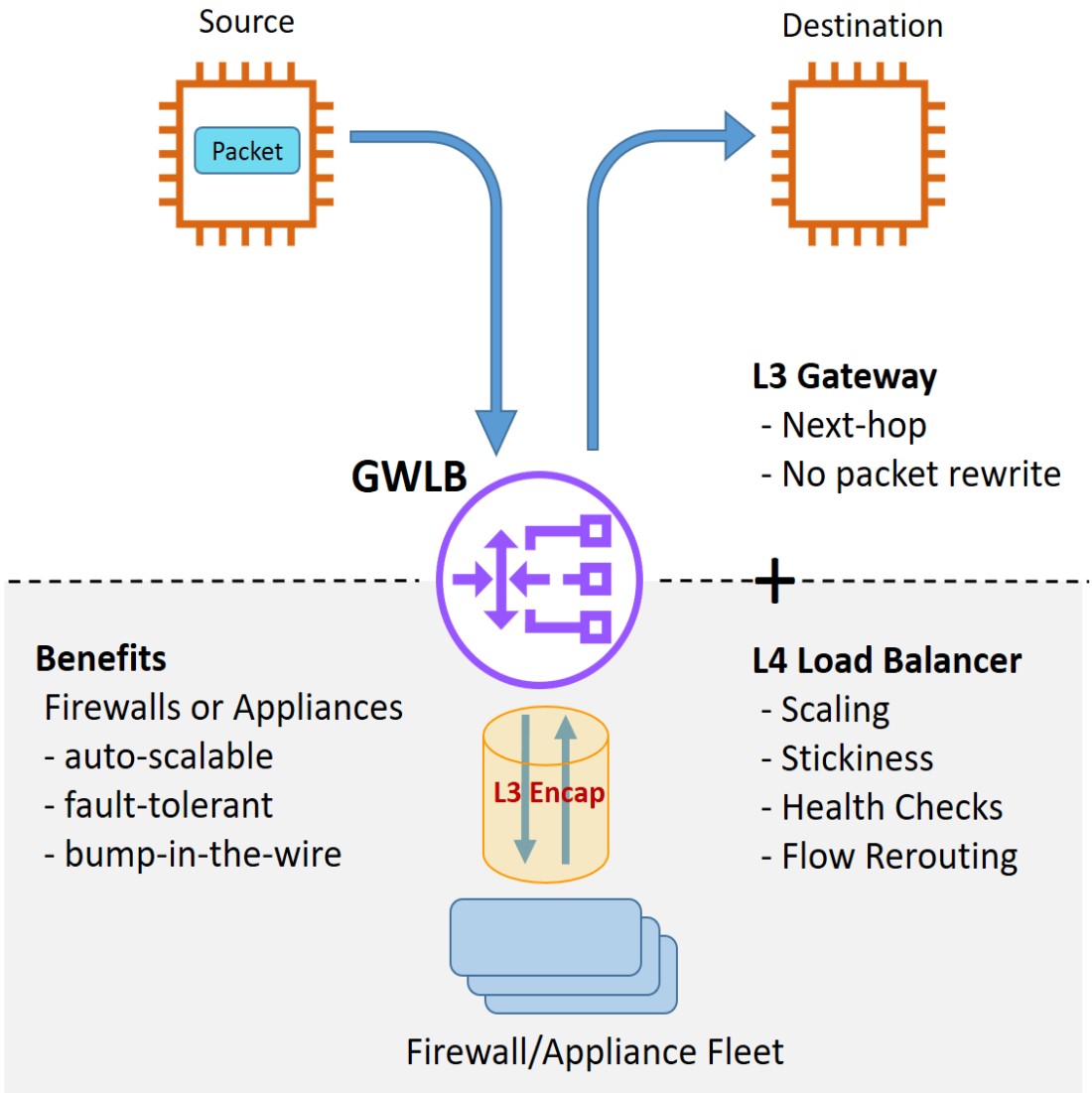


CREATED BY:- VIVEK GANDHI

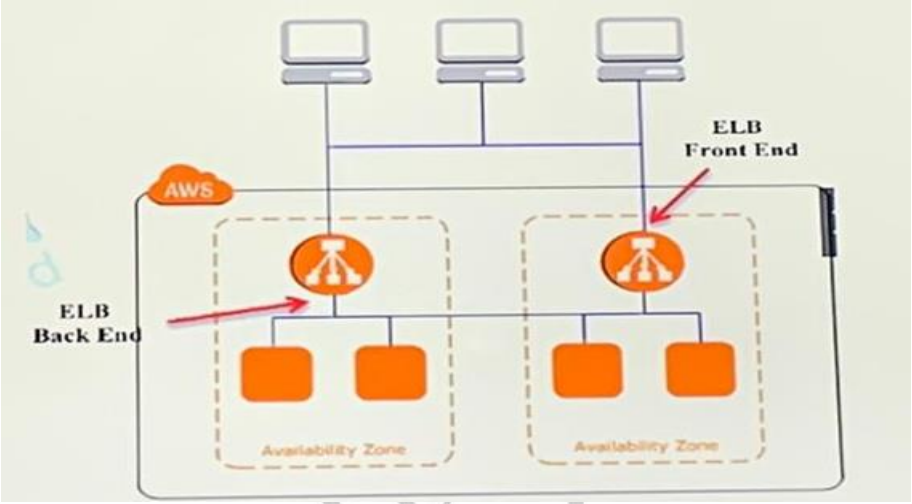
3. Network Load Balancer



4. Gateway Load Balancer

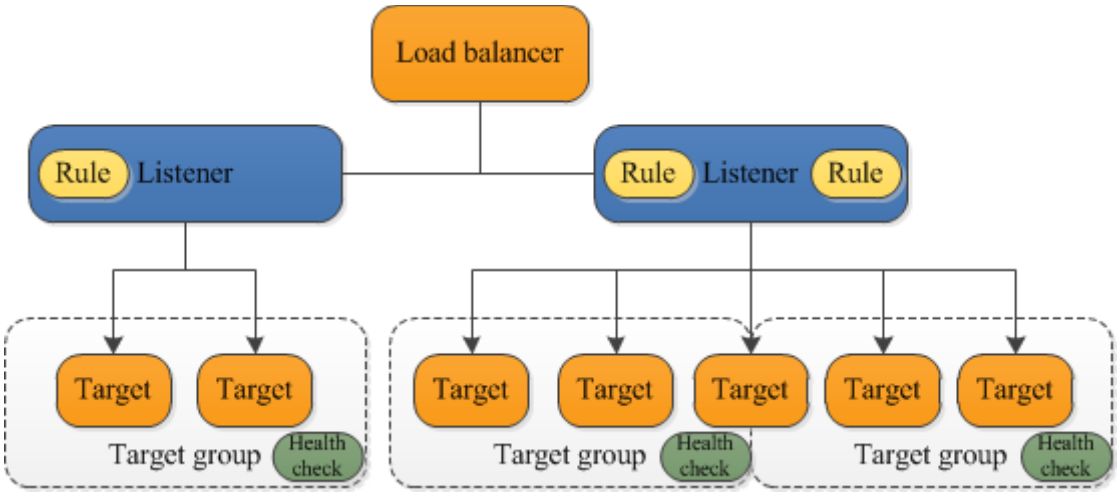


- An internet facing load balancer has a publicly resolvable DNS name.
- Domain names for content on the EC2 instances served by the ELB, is resolved by the internet DNS server to the ELB DNS name (and hence IP address).
- This is how traffic from the internet is directed to the ELB front-end.
- Classic load balancer service support: http, https, TCP, SSL.
- Protocols ports supported are 1-65535.
- It supports IPV4, IPV6 and Dual Stack.
- Application load balancer distributes incoming application traffic across multiple targets such as EC2 instances in multiple availability zone. This increases the availability of your application.
- Network load balancer has ability to handle volatile workloads and scale to millions of requests per seconds.



❖ ELB – Listeners

- An ELB listener is the process that checks for connection request.
- You can configure the protocol/ port number on which your ELB listener listen for connection request.
- Fronted listeners check for traffic from client to the listener.
- Backend listeners are configured with protocol/port to check for traffic from the ELB to the EC2 instances.



 CREATED BY:- VIVEK GANDHI

❖ ELB

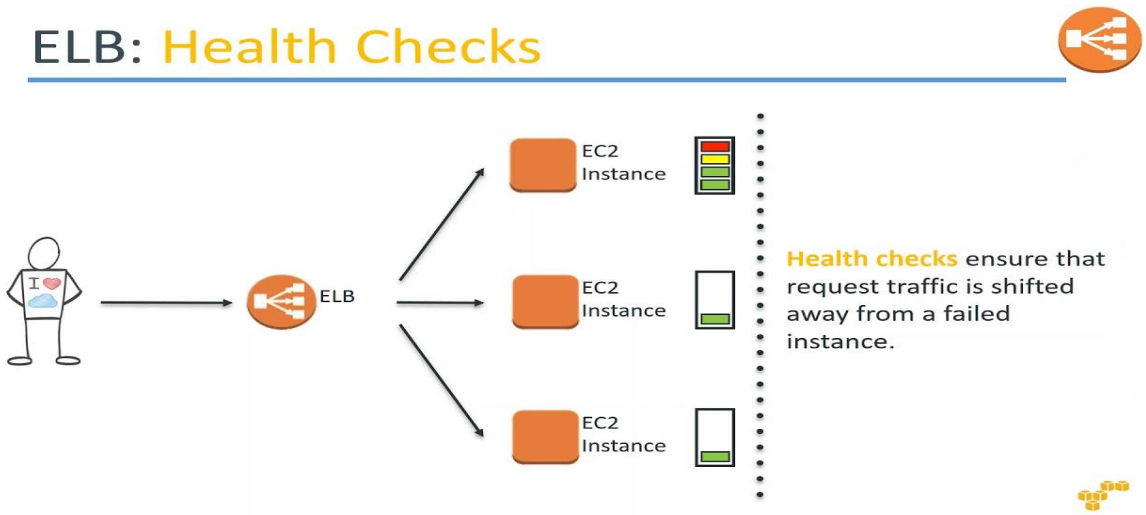
- It may take some time for the registration of the EC2 instances under the ELB to complete.
- Registered EC2 instances are those are defined under the ELB.
- ELB has nothing to do with the outbound traffic that is initiated/generated from the registered EC2 instances destined to the internet or to any other instances within the VPC.
- ELB only has to do with inbound traffic destined to the EC2 registered instances (as the destination) and the respective return traffic.
- You start to be charged hourly (also for partial hours) once your ELB is active.
- If you do not want to be charged as you so not need the ELB anymore, you can delete it.
- Before you delete the ELB, it is recommended that you point the Route53 to somewhere else other than ELB.
- Deleting the ELB does not affect or delete the EC2 instance registered with it.
- ELB forwards traffic to "eth0" of your registered instances.
- In case the EC2 registered instances has multiple IP address on eth0, ELB will route the traffic to its primary IP address.
- Elastic load balancer supports IPV4 address only in VPC.
- To ensure that the ELB service can scale ELB nodes in each AZ, ensure that the subnet defined for the load balancer is at least /27 in scale size and has at least 8 available IP address the ELB nodes can use to scale.

❖ ELB – Health Checks

- For fault tolerance it is recommended that you distribute your registered EC2 instances across multiple AZ with in the VPC region.
- If possible, try to allocate same number of registered instances in each AZ.
- The load balancer also monitors the health of its registered instances and ensures that it routes traffic only to healthy instances.
- A healthy instance shows as healthy under the ELB.
- When the ELB detects an unhealthy instance, it stops routing traffic to that instance.
- An unhealthy instance shows as unhealthy under the ELB.
- By default, AWS console uses ping http (port 80) for healthy check.
- Registered instances must respond with an http "200 OK" message within the timeout period else it will be considered as unhealthy.
- AWS API uses ping TCP (port-80) for health check.
- Response time-out is 5 seconds (range is 2-60 sec).
- Health check internet.
- Period of time between health check (default 30 and range is 5 to 300 sec)

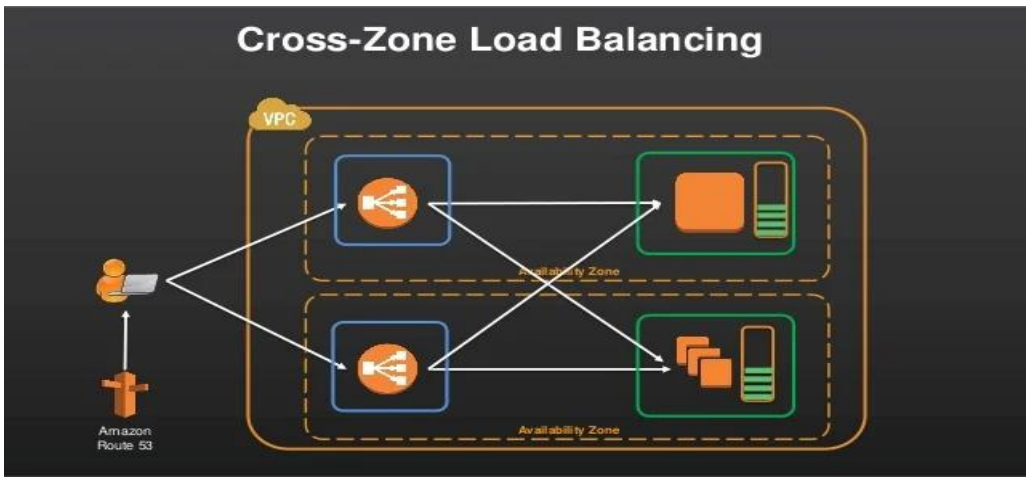
- **Unhealthy Threshold:** number of consecutive failed health check that should occur before the instance is declared unhealthy.
  - Range is 2 to 10, Default is 2

- **Healthy Threshold:** number of consecutive successful health checks that must occur before the instance considered unhealthy.
  - Range is 2 to 10, Default is 10
  - By default, the ELB distributes traffic evenly between the AZ, it is defined in without consideration to the number of registered EC2 instances in each AZ.



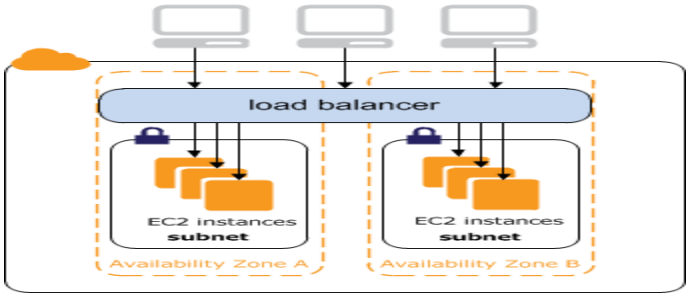
❖ Cross Zone Load Balancing

- Disabled by default.
- When enabled, the ELB will distribute traffic evenly between registered EC2 instances.
- If you have 7 EC2 instances in one AZ, and 3 in another AZ, and you enabled cross zone
- Load balancing each registered EC2 instances will be getting the same amount of traffic load from the ELB.
- ELB, name you choose must be unique within the account.
- ELB is region specific, so all registered EC2 instances must be in the same region, but can be in different AZs.
- To define your ELB in an AZ you can select one subnet in that AZ. Subnet can be public or private.
- Only one subnet can be defined for the ELB in an AZ.
- If you try and select another one in the same AZ, it will replace the former one.
- If you register instance in an AZ with ELB but do not define a subnet in that AZ for the ELB, these instances will not receive traffic form the ELB.
- ELB should always be accessed using DNS and not IP.

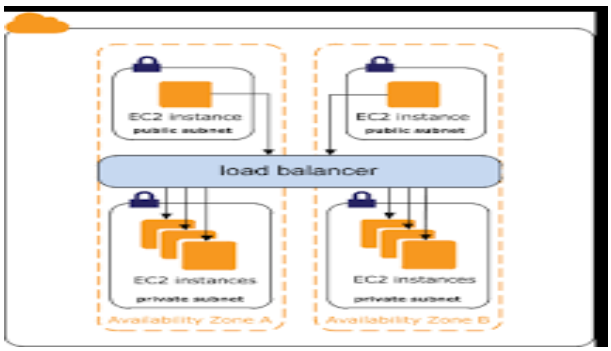


❖ An ELB can be internet facing or internal ELB:

- **Internet Facing:**
  - ELB nodes will have public IP address.
  - DNS will resolve the ELB DNS name to these IP address.
  - If routes traffic to the private IP address of your registered EC2 instances.
  - You need one public subnet in each AZ where the internet facing ELB will bedefined such that the ELB will be able to route internet traffic.
  - Format of the public ELB DNS name of internet facing ELB: name-1234567890.region.elb.amazonaws.com



- **Internal LB:**
  - ELB Nodes will have private IP Address to which the DNS resolves ELB DNS Name
  - It routes traffic to the private IP address of your registered EC2 instances.
  - Format of the ELB DNS name for internal ELB
  - Internal name **name.123456789.region.elb.amazonaws.com**
  - An ELB listener is the process that checks for connection request.
  - Each network load balancer needs at least one listener to accept traffic.
  - You must assign a security group to your ELB. This will control traffic that can reach your ELB front end listeners.



- **Target Group:**
  - Logical grouping of targets behind the load balancer.
  - Target groups can be existed independently from the load balancer.
  - Target group can be associated with an auto scaling group.
  - Target group can contain up to 200 targets.

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer	Classic Load Balancer
Load Balancer type	Layer 7	Layer 4	Layer 3 Gateway + Layer 4 Load Balancing	Layer 4/7
Target type	IP, Instance, Lambda	IP, Instance	IP, Instance	
Terminates flow/proxy behavior	Yes	Yes	No	Yes
Protocol listeners	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP	TCP, SSL/TLS, HTTP, HTTPS
Reachable via	VIP	VIP	Route table entry	
Layer 7				
Redirects	✔			
Fixed Response	✔			
Desync Mitigation Mode	✔			
HTTP header based routing	✔			
HTTP2/gRPC	✔			

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer	Classic Load Balancer
Slow start	✔			
Outpost support	✔			
Local Zone	✔			
IP address - Static, Elastic		✔		
Connection draining (deregistration delay)	✔	✔	✔	✔
Configurable idle connection timeout	✔			✔
PrivateLink Support		✔ (TCP, TLS)	✔ (GWLBE)	
Zonal Isolation		✔	✔	
Session resumption	✔	✔		
Long-lived TCP connection		✔	✔	
Load Balancing to multiple ports on the same instance	✔	✔	✔	
Load Balancer deletion protection	✔	✔	✔	
Preserve Source IP address	✔	✔	✔	
WebSockets	✔	✔	✔	
Supported network/Platforms	VPC	VPC	VPC	EC2-Classic, VPC
Cross-zone Load Balancing	✔	✔	✔	✔
IAM Permissions(Resource, Tag based)	✔	✔	✔	✔ (Only resource based)
Flow Stickiness (All packets of a flow are sent to one target, and return traffic comes from same target)	Symmetric	Symmetric	Symmetric	Symmetric
Target Failure behavior	Fail close on targets, unless all targets are unhealthy(fail open)	Fail close on targets, unless all targets are unhealthy(fail open)	Existing flows continue to go to existing target appliances, new flows are rerouted to healthy target appliances.	
Health Checks	HTTP, HTTPS, gRPC	TCP, HTTP, HTTPS	TCP, HTTP, HTTPS	TCP, SSL/TLS, HTTP, HTTPS