ELB - Elastic load balancing

ELB

- distribute traffic across servers
- Supprots HTTP/S, SSL and TCP distribution
- Provides signle CNAME for DNS

Benifits

- Managed load balancing scales-in and scales-out
- Does health check. Distributed to only healhty instances. Achives High availability.
- Integrates with Auto scale to scale-in and out instances
    - Since EC2 counts are dynamically changing, ELB provides single point of entry for consumers.
- Not only public facing. Internal facing manages DB or app layer.
- SSL termination and Certificate management (check how this works!)

Types of load balancers

Internet facing

- Receives traffic from client across internet.
- Have public DNS name
- Only use DNS and dont use IP address
- Web tier of app

Internal

- Receives traffic from ELB / EC2 of public subnet
- DB or App tier of app
- EC2 instances behind these balancers are in private subnet

HTTPS load balancers

- Enables traffic encryption b/w ELB & Clients
- Enables traffic encryption b/w ELB & Backend hosts
- Install SSL cert in ELB, terminates SSL traffic at ELB
    - ELB then decrypts messsage and forward to EC2
- Mulitple websites served behind single ELB. Need to include Server Alternative Name for each website.

Listeners

- To setup ELB , you need a listener process.
- Listener process configured with protocol & port (for client and backend)
    - HTTP/ HTTPS/ TCP / SSL
    - These protocols are used by listener process

- Layer 4
    - ELB forwards message without modifying headers
    - Back-end EC2 wont know whats originating client
    - Use proxy-server for this
- Layer 7
    - ELB forwards message without modifying HTTP headers
    - Backend EC2 wont know. So you x-forward-for headers

Configuring ELB

Idle connection timeout

- Set to 60 seconds.
- Close connection to backend or client if no traffic received
- Customizable
- HTTP/ HTTPS listeners
    - Recommendation is to use keep-alive settings in webserver or kernel level of back-end

Cross zone load balancing

- ELB by default load balances across AZ's evenly. So keep equal EC2s in each AZs
- If you can't do that then enable Cross zone load balancing which will balance traffic across all EC2

Proxy protocol

- Enable proxy protocol so that EC2 backend knows who is originating client

Sticky sessions

- Send traffic to same EC2 for that session
- Key- how long session should be bound
    - Application based stickiness - Configure ELB to use your application session cookie
        - Specify application cookie name console or CLI while enabling this option
        - ELB uses special cookie that follows application cookie's lifetime
    - Duration based stickiness - Or Let AWS create cookie - AWSELB
        - enable this option and specify seconds

Health checks

- Ping , html page