Cantrill – Regional and Global Architecture

Global –

* Global Service Location & Discovery
* Content Delivery (CDN) and optimization
* Global health checks & Failover

Regional –

* Regional entry point
* Scaling & Resilience
* Application services & components

ELB Evolution

* 3 types of Load Balancers (ELB) available within AWS
* Split between v1 (avoid/migrate) and v2 (prefer)
* Classic Load Balancer (CLB) – v1 – introduced in 2009
  + Not really Layer 7, lacking features, 1 SSL per CLB
* Application Load Balancer (ALB) – v2 0 HTTP/S/WebSocket
* Network Load Balancer (NLB) – v2 – TCP, TLS, UDP
* V2 = faster, cheaper, support target groups and rules

Elastic Load Balancer (ELB) Architecture

* 2 types of ELB
  + Internet-facing
  + Internal Only
* Each ELB is configured with an (A) record DNS name. This resolves to the ELB Nodes
* Configured to run in 2+ AZs. 1+ Nodes are placed into a subnet in each AZ and scale with load
* Load Balancers (Nodes) are configured with listeners which accept traffic on a port & protocol & communicate with targets on a port and protocol.
* ELB uses 8 Ips per subnet and requires a /27 (or/28\*) or larger subnet to allow for scale
* Internet facing Nodes have public IPs
* Internal Only Nodes have private IPs
* Internal load balancers can be used to allow scaling between application tiers
  + Users connect to the Load balancer via (A) record DNS name to the nodes in public subnets (or private) then traffic is distributed to the public (or private) EC2 web server. To allow for independent scaling, traffic from the EC2 web server is routed to the internal-only ELB, traffic is then routed to the private EC2 instances or other resources in the private subnet.

ELB Key Facts

* ELB is a DNS A Record pointing at 1+ Nodes per AZ
* Nodes (in one subnet) per AZ can scale
* Internet-facing means nodes have public IPv4 IPs
* Internal is private only IPS
* EC2 doesn’t need to be public to work with a LB
  + EC2 doesn’t require a public IP to receive traffic from an internet-facing LB
* Listener Configuration controls WHAT the LB does
* 8+ Free IPs per subnet required, and /27 (or /28\*) subnet to allow scaling

Application and Network Load Balancer (ALB vs NLB)

* Load Balancer Consolidation
  + CLBs (v1) don’t scale – every unique HTTPs name requires an individual DLB because SNI isn’t supported.
  + V2 load balancers support rules and target groups. Host based rules using SNI and an ALB allows consolidation

Application Load Balancer (ALB) –

* Layer 7 load balancer – listens on HTTP and/or HTTPS
* No other layer 7 protocols supported – SMTP, SSH, Gaming
* No TCP, UDP, or TLS Listeners
* L7 content type, cookies, custom headers, user location, and app behavior – advanced traffic inspection capabilities/visibility (rule config)
* HTTP HTTPS (SSL/TLS) always terminated on the ALB – NO BROKEN SSL
  + A new connection is made to the application
  + Security risk – no end to end inflight encryption capabilities
* ALB MUST have SSL certs if HTTPS is used
* ALBs are slower than NLB – more levels of the network stack to process
* Health checks evaluate application health – layer 7

ALB Rules –

* Rules direct connections which arrive at a listener
* Processed in priority order
* Default rule = catchall
* Rule Conditions: host-header, http-header, http-request-method, path-pattern, query-string, and source-ip
* Actions: forward, redirect, fixed-response, authenticate-oidc, and authenticate-cognito

NLB –

* Layer 4 Load Balancer – TCP, TLS, UDP, TCP\_UDP
* No visibility or understanding of HTTP or HTTPS
* No headers, cookies, or session-stickiness
* Really really really fast – millions of requests per second (mps), 25% of ALB latency
* Also supports SMTP, SSH, Game Servers, Financial apps (not http/s)
* Health checks JUST check ICMP / TCP Handshake – not app (layer 7) aware
* NLBs can have static IP’s – useful for whitelisting
* Forward TCP to instances (end to end) unbroken encryption
* Used with private link to provide services to other VPCs

ALB vs NLB –

* Unbroken encryption – NLB
* Static IP for whitelisting – NLB
* The fastest performance – NLB (millions rps)
* Protocols not HTTP or HTTPs – NLB
* Private Link – NLB
* Otherwise – ALB

Launch Configurations and Launch Templates – EC2 feature

* LC and LT Key Concepts
  + Allow you to define the configuration of an EC2 instance in advance
  + AMI, Instance Type, Storage & Key Pair
  + Networking and Security Groups
  + User data & IAM Role
  + Both are NOT editable – defined once. LT has versions
  + LT provide newer features – including T2/T3 Unlimited, Placement Groups, Capacity Reservations, and Elastic Graphics
  + LC can only be used to configure ASGs
  + LT can be used to configure ASGs and EC2
    - LT can be used to save time when provisioning EC2 instances from the console UI or CLI

Gateway Load Balancer (GWLB)

* Helps you run and scale 3rd party appliances
  + Firewalls, intrusion detection, and prevention systems
* Inbound and Outbound traffic (transparent inspection and protection)
* GWLB Endpoints – traffic enters/leaves via these endpoints
* The GWLB balances across multiple backend appliances
* Traffic and metadata is tunneled using GENEVE protocol
* Traffic remains unaltered
  + Original packets remain unaltered encapsulated through to appliances and back
* Load balances across security appliances
* Flow stickiness
  + One flow will always use the appliance
* Client traffic flows in via ingress gateway route table. RT directs any traffic destined for the ALB subnets at the GW Endpoint in the same AZ. Traffic flows from the GW Endpoint to the GWLB (Security VPC) then through to the backend security appliances. Traffic is returned to the GWLB (GENEVE encapsulation) through to the GW Endpoint through ALB through to application. Return traffic is sent to ALB. ALB sends traffic to GW Endpoint, then through to the GWLB (Security VPC) then to the backend security appliances. From there traffic is routed through GWLB to the GW Endpoint and out to the IGW back to the client.
  + This allows for original source and destination persistence.
  + Transparent, inline network security, done in a scalable, resilient, and abstract way.