Cantrill Review – Advanced VPC Networking

* Only capture packet metadata (NOT CONTENTS)
  + To capture packet contents, you need a package sniffer – typically installed on an EC2 instance
* Attach virtual monitors
  + VPC – All ENI’s in that VPC
  + Subnet – All ENIs in that Subnet
  + Single ENI
* VPC Flow Logs are NOT real-time
* Supported Log Destinations
  + S3
    - Able to access Log files directly – integrate with third party monitoring system
  + CloudWatch Logs
    - Integrate with other products, stream data to different locations, access pragmatically or via console
* Use Athena to query logs in S3 – only pay for data read
  + Schema on read architecture
* Flow logs capture metadata from the capture point downwards
  + Ex. Flow Log attached to VPC
    - Logs packet metadata in VPC => Subnets within VPC => all ENI’s in VPC
* VPC Flow Logs can capture –
  + ACCEPTED connections
  + REJECTED connections
  + ALL connections
* Captures Flow Log Records
* VPC Flow Logs
  + Src/dest address
  + Src/dest port
  + Protocol
    - ICMP – 1
    - TCP – 6
    - UDP - 17
  + Action
    - Accepted or Rejected
* VPC Flow Log excluded
  + To/From 169.254.169.254
  + 169.254.169.123
  + DHCP
  + Amazon DNS Server
  + Amazon Windows License

Egress – Only Internet Gateway

* Allow outbound (and response) only access to the Public AWS Services and Public Internet for IPv6 enabled instances or other VPC based services
* With IPv4 addresses are Private or Public
* NAT allows Private IPv4 IPs to access public networks + Public Internet
  + Without allowing externally initiated connections (IN)
* With IPv6 all IPs are Public
* Internet Gateway (IPv6) allows all IPs IN and OUT
* Egress-Only is outbound-only for IPv6
* Attached at the VPC level
  + 1 per VPC – used by all AZs
  + Add default IPv6 outbound route for Egress-Only Gateway
    - ::// added to RT with eigw-id as target
  + Stateful devices
  + All inbound is denied
    - Return traffic from an outbound connection is allowed
* Egress-Only Gateway is HA by default across all AZs in the region – scales as required

VPC Endpoints (Gateway)

* allow access to S3 and DynamoDB without using public addressing
* Gateway endpoints add ‘prefix lists’ to route table, allowing the VPC router to direct traffic flow to the public services via the gateway endpoint
* Provide private access to S3 and DynamoDB
* Prefix List added to route table => Gateway Endpoint
* Highly Available (HA) across all AZ’s in a region by default
* Associated with a VPC
  + Specify which subnets will be using it
  + Automatically configures the route table to include the prefix list for S3 and DynamoDB
* Endpoint Policy
  + Used to control what it can access
    - Ex. particular subset of S3 buckets
* Regional – can’t access cross-region services
* Use cases –
  + Private VPC allow access to public resources (S3 or DynamoDB)
  + Prevent Leaky Buckets – S3 buckets can be set to private only by allowing access ONLY from a Gateway Endpoint
* Primary Benefit –
  + Public addressing isn’t required for S3 or DynamoDB access
    - For maximum security, configure the Bucket Policy to DENY all traffic that doesn’t come from the GW endpoint
  + Region resilient by default
  + Not accessible outside the VPC
* Endpoint Policy – controls WHAT the GW Endpoint can be used for
* Gateway Endpoints deploy instantly after creation

VPC Endpoints (Interface)

* Provide private access to AWS Public Services
* Historically – anything NOT S3 and DDB
  + But S3 is now supported
* Added to specific subnets – an ENI
  + Single ENI
  + Not Highly Available
  + Best practice
    - 1 Interface Endpoint (ENI) per AZ to ensure full HA
* Network access controlled via Security Groups
* Endpoint Policies
  + Restrict what can be done (accessed) with the endpoint
* Supported protocols
  + TCP
  + IPv4
* Use PrivateLink
  + Allow AWS or External services to be injected to your VPC
    - Assigned network interfaces in VPC
  + Useful for heavily regulated industries
* Interface Endpoint provides a NEW service endpoint DNS (upon creation)
  + Resolves to private address of the service
* DNS Endpoints
  + Endpoint Regional DNS
  + Endpoint Zonal DNS
  + Private DNS (default)
* Applications can optionally use the Regional or Zonal endpoints or leverage PrivateDNS.
* Problem (what is solved using Interface endpoints)
  + If private resources attempt to access SNS via the endpoint, no access allowed without a public IP address
* Solution
  + Traffic flows from Private instance in private subnet to Interface endpoint in Public subnet then to the service
* Private DNS associated a private R53 Hosted Zone to the VPC changing the default service DNS to resolve to the interface endpoint IP
* Can only have 1 interface in each AZ
* Interface Endpoints take a few minutes to deploy after creation

VPC Peering

* Direct encryption network link between two VPCs
* Works same/cross-region and same/cross-account
* Optional – Public Hostnames resolve to private IPs
* Same Region VPC Peers – SGs can reference peer SGs
  + Cross region – SGs must reference IP or IP Range
* VPC Peering does NOT support transitive peering
  + Creates a logical gateway inside each VPC
* Routing Configuration is needed – SGs & NACLs can filter
  + Ensure traffic is able to flow between the two VPCs
* Route Tables at both sides of the peering connection are needed, directing traffic flow for the remote CIDR at the peer gateway object
  + Configure on all subnets on both sides of the peering connection
    - For open communication only
* VPC Peering connections cannot be created where there is overlap in the VPC CIDRs
  + Ideally NEVER use the same address ranges in multiple VPCs
* Any data transferred between VPCs is encrypted
  + Cross region peered VPCs use the AWS Global network