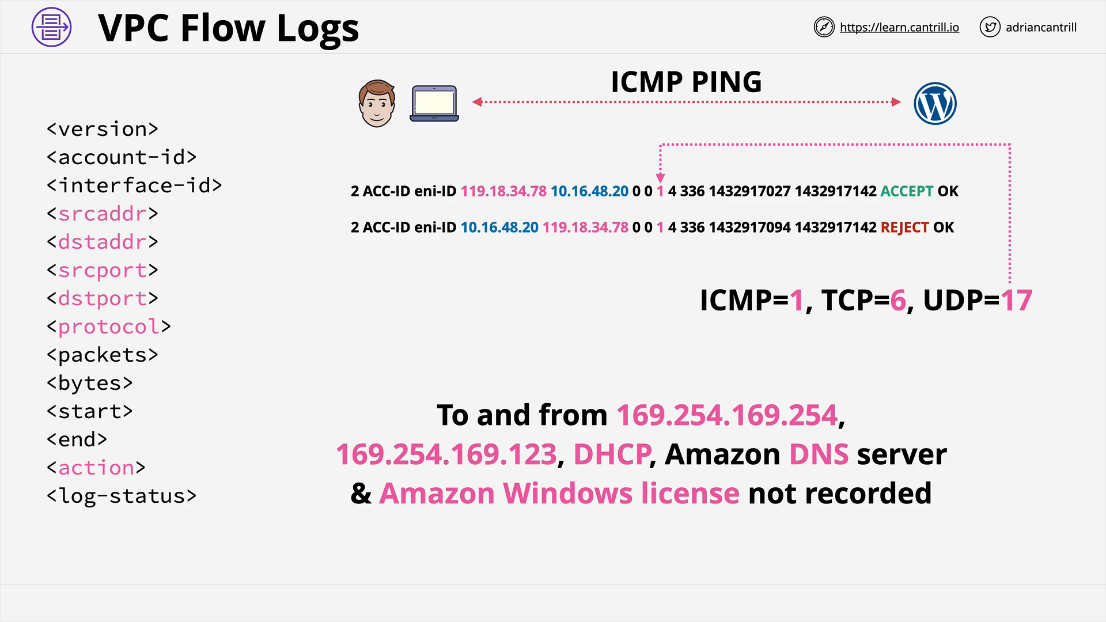
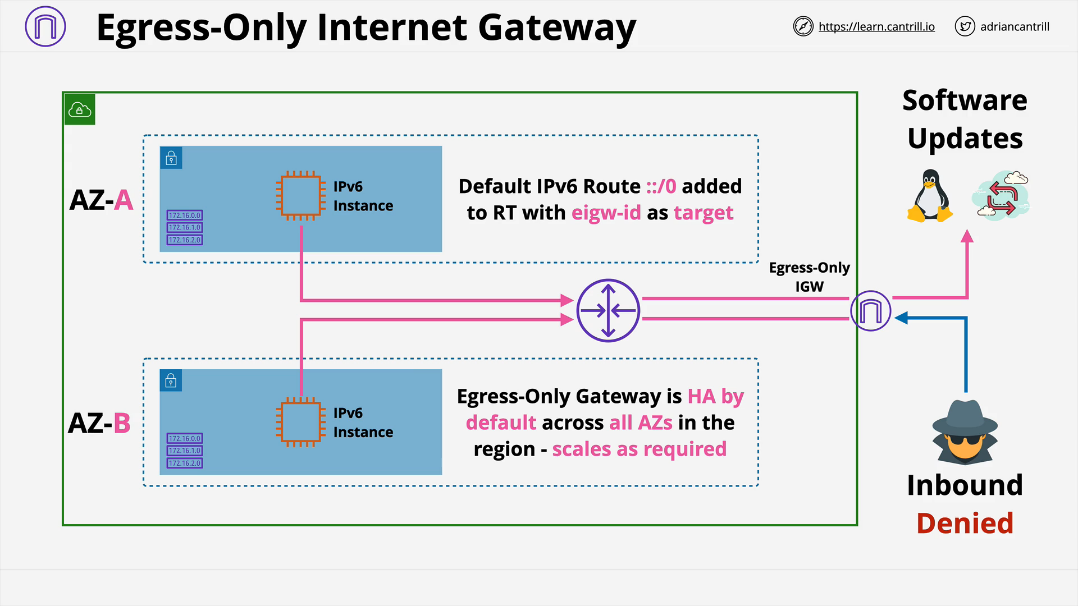
**Advanced-VPC**

**VPC Flow Logs**

* Capture packet metadata, not packet contents.
  + Things like source IP
  + Destination IP
  + Packet size
  + Anything which could be observed from the outside of the packet.
* Capture data at various different monitoring points.
  + VPC: all interfaces in that vpc
  + Subnets: interfaces in that subnet
  + Interface directly
* VPC flow logs are not realtime
* Destination can be **S3 or CloudWatch logs**
* Flow log inheritance is downwards starting at the VPC.
* RDS can use VPC flow logs
* The packet will always have source, then destination, then response.

**Egress-Only Internet Gateway**

* IPv4 addresses are private or public
* NAT allows private IPs to access public networks and receive responses.
* NAT will **not allow externally initiated connections** IN.
* Using IPv6, all IPs are public.
  + Internet Gateway (IPv6) allows all IPs **in** and **out**
* Egress-only is **outbound only** for IPv6. It is exactly the same as NAT, only outbound only.
* To configure the Egress-only gateway, you must add default IPv6 route ::/0 added to RT with eigw-id as target.

**VPC Endpoints (Gateway)**

Allow a private only resource inside a VPC or any resource inside a private only VPC access to **S3 and DynamoDB.**

Normally when you want to access a public service through a VPC, you need infrastructure. You would create an IGW and attach it to the VPC. Resources inside need to be granted IP address or implement one or more NAT gateways which allow instances with private IP addresses to access these public services.

When you allocate a gateway endpoint to a subnet, a **prefix list** is added to the route table. The target is the gateway endpoint. Any traffic destined for S3, goes **via the gateway endpoint**.

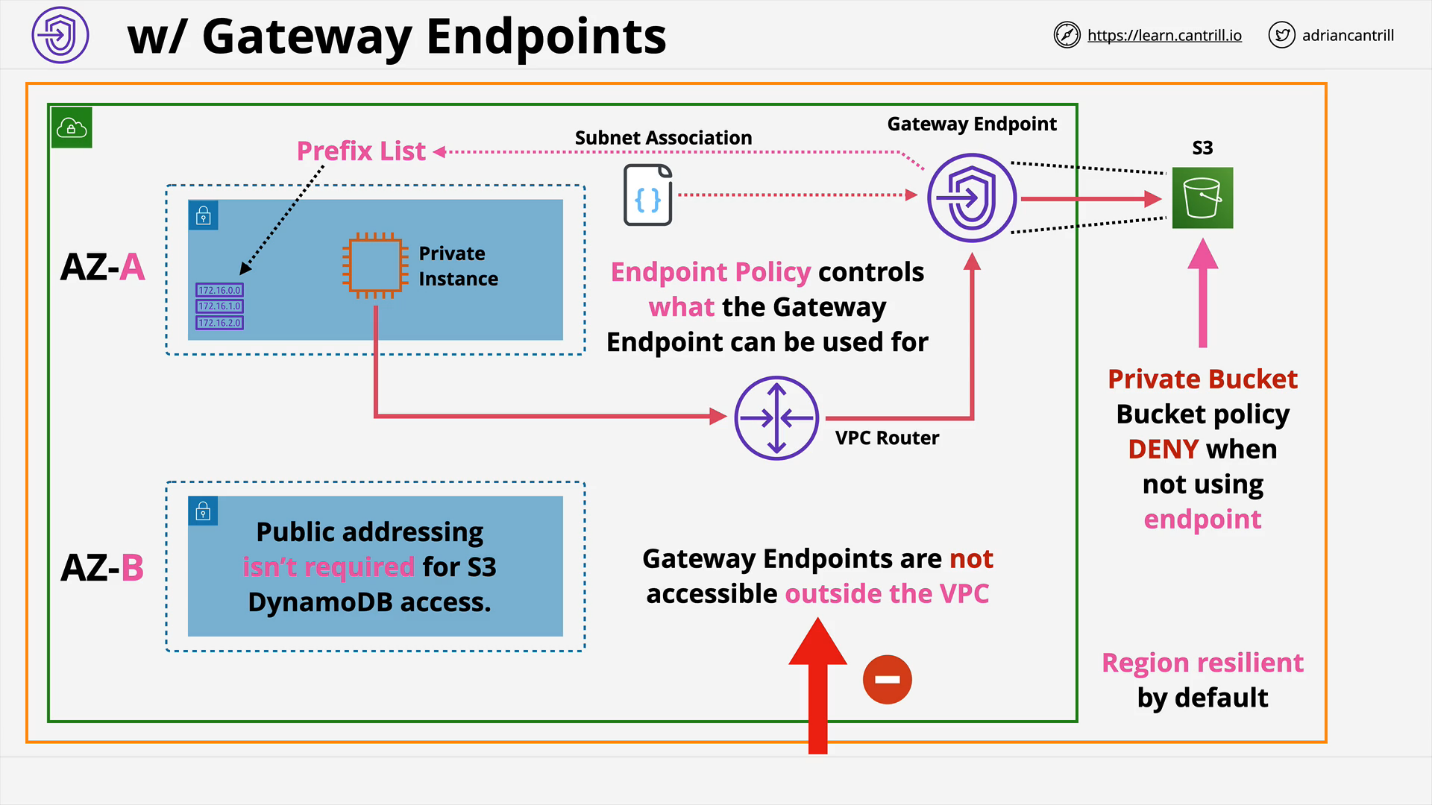
The gateway endpoint is **highly available** for all AZs in a region by default.

With a **gateway endpoint you set which subnet will be used with it and it will configure automatically.** A gateway endpoint is a VPC gateway object. **Endpoint policy controls what things can be connected to by that endpoint.**

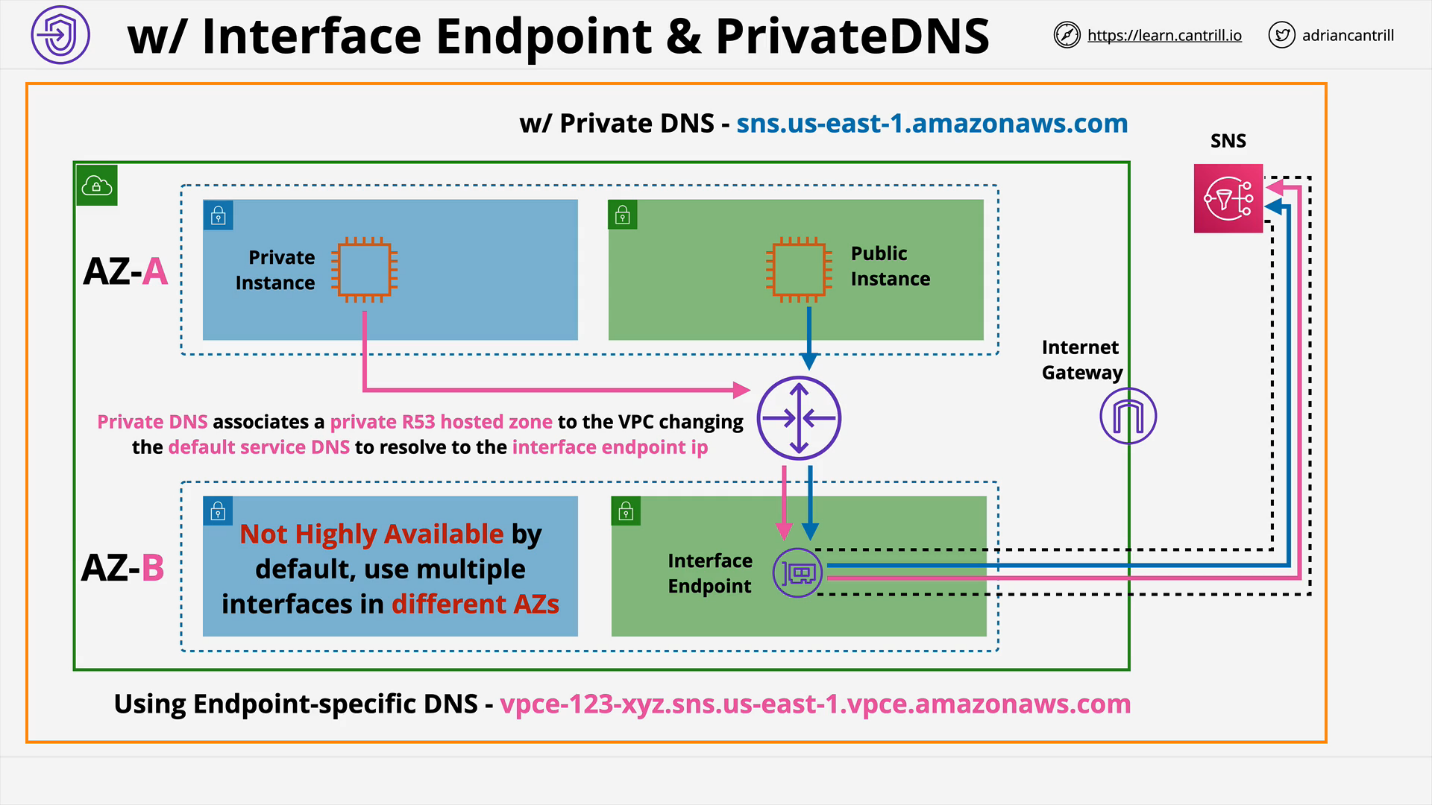
Gateway endpoints can only be used to access services in the same region. Can't access **cross-region services.**

S3 buckets can be set to **private only by allowing access ONLY from a gateway endpoint.** For anything else, the implicit deny will apply.

They are only accessible from inside that specific VPC.

**VPC Endpoints (Interface)**

* Provide private access to AWS Public Services.
  + Anything EXCEPT S3 and DynamoDB
* These are **not HA by default** and are added to **specific subnets.**
  + For HA, add **one endpoint, to one subnet, per AZ used in the VPC**
  + Must add **one endpoint for one subnet per AZ**
* Network access controlled via security groups.
* You can **use Endpoint policies to restrict what can be accessed with the endpoint**.
* ONLY TCP and IPv4 at the moment.
* Behind the scenes, it uses **PrivateLink**.
* Endpoint provides a **NEW** service endpoint DNS
  + e.g. vpce-123-xyz.sns.us-east-1.vpce.amazonaws.com
* **Regional DNS** is one single DNS name that works whatever AZ you're using to access the interface endpoint. Good for simplicity and HA.
* **Zonal DNS** resolved to that one specific interface in that one specific AZ.
* Either of those two points of endpoints can be used by applications to directly and immediately utilize interface endpoints.
* **PrivateDNS associates R53 private hosted zone with your VPC. This private hosted zone carries a replacement DNS record for the default service endpoint DNS name. It overrides the default service DNS with a new version that points at your interface endpoint. Enabled by default.**



**Gateway Endpoints vs Interface Endpoints**

**Gateway endpoints** work using **prefix lists and route tables** so they **do not need changes to the applications**. The application thinks it's communicating directly with S3 or DynamoDB and all we're doing by using a **gateway endpoint is influencing the route that the traffic flow uses**. Instead of using IGW, it goes via gateway endpoint and can use private IP addressing. **highly available**

**Interface Endpoints** uses DNS and a private IP address for the interface endpoint. You can either use the endpoint specific DNS names or you can **enable PrivateDNS which overrides the default and allows unmodified applications to access the services using the interface endpoint.** This doesn't use routing and **only DNS.** **not highly available**

**VPC Peering**

**Direct encrypted network** link between two and only two VPCs. Peering connection can be in the same or cross region and in the same or across accounts.

When you create a VPC peer, you **can enable an option so that public hostnames of services in the peered VPC resolve to the private internal IPs.** You can use the **same DNS names if its in peered VPCs or not**. If you attempt to resolve the **public DNS hostname of an EC2 instance, it will resolve to the private IP address of the EC2 instance**.

VPCs in the **same region can reference** each other by **using security group id**. You can do the same efficient referencing and nesting of security groups that you can do if you're inside the same VPC. This is a feature that only works with VPC peers inside the same region.

In different regions, you can **utilize security groups**, but you'll need to **reference IP addresses or IP ranges**. If VPC peers are in the same region, then you can do the logical referencing of an entire security group.

VPC peering connects **ONLY TWO**

VPC Peering does not support **transitive peering**. If you want to connect 3 VPCs, you need 3 connections. You can't route through interconnected VPCs.

VPC **Peering Connections CANNOT be created with overlapping VPC CIDRs.**

