AWS KMS (Key Management Service)

AWS KMS is a managed service that lets you create and control the encryption keys used to encrypt your data. It integrates with most AWS services (S3, EBS, RDS, Lambda, etc.) and provides an easy and secure way to handle encryption and decryption using keys without directly exposing them.

Key Features:

- Manages symmetric and asymmetric keys
- IAM integration for fine-grained access control
- Auditing of key usage via CloudTrail
- Central key usage through API calls (CLI/SDK)
- Supports automatic key rotation
- Available globally with multi-region key support

Use Case:
A financial application stores sensitive customer data in Amazon RDS. To ensure security and compliance, the application uses KMS to encrypt the database at rest. All access and key usage is logged and monitored using AWS CloudTrail for audit purposes.

KMS Key Types & Key Management Options

AWS KMS supports different types of cryptographic keys and management models to meet a variety of security needs:
Key Types:

1. Symmetric Keys (AES-256):
   a. One key used for both encryption and decryption. Most AWS services use symmetric KMS keys.
2. Asymmetric Keys (RSA or ECC):
   a. Public-private key pairs used for encrypt/decrypt or sign/verify operations. The public key can be downloaded, but the private key remains secure in KMS.

Key Management Models:

1. AWS Owned Keys (Free):
   Default keys used transparently for services like S3, SQS, DynamoDB with SSE-S3. No access or control by the user.
2. AWS Managed Keys (Free):
   Used by services like RDS, EBS, or Lambda. Named format: aws/service-name. Limited visibility but used for basic encryption needs.
3. Customer Managed Keys (CMK) ($1/month):
   User-created, with full control, including rotation, key policy, access permissions, and usage tracking.
4. Imported Keys:
   You bring your own key material into KMS. Used in highly regulated environments. Rotation is manual only.

Use Case:
A healthcare company must comply with HIPAA and needs full control and auditability of encryption keys. It uses customer-managed symmetric keys with CloudTrail logging and scheduled rotation enabled to encrypt patient records stored in Amazon S3 and Amazon RDS.

KMS Multi-Region Keys

KMS Multi-Region Keys are a special feature of AWS KMS that allows you to create cryptographically identical keys in multiple AWS Regions. These keys have the same key ID and key material, but are independently managed and replicated across regions (Primary + Replicas).

Key Features:

- Use the same key across multiple regions without cross-region API calls
- Encrypt data in one region and decrypt it in another
- Each region's key is independently managed and rotated
- Reduces latency and improves availability for multi-region architectures

Important Notes:

- These are not global keys — each copy is a distinct KMS key but behaves the same cryptographically
- Common use cases: Global DynamoDB tables, Global Aurora, and client-side encryption use cases

Use Case:
A SaaS company uses Global DynamoDB tables to store user data across us-east-1 and ap-southeast-2. It encrypts data client-side using a KMS multi-region primary key in us-east-1, and the replica key in ap-southeast-2 allows fast decryption without latency or dependency on the source region's KMS.

SSM Parameter Store

AWS Systems Manager Parameter Store is a secure, serverless storage service for storing configuration data and secrets like database connection strings, API keys, or environment variables. Parameters can be stored as plain text or encrypted with AWS KMS.

Key Features:

- Stores key-value pairs, either in plaintext or encrypted form
- Integration with AWS KMS for encryption at rest
- Version control for parameters
- IAM-based access control
- EventBridge integration for notifications
- Hierarchical storage path (/myapp/dev/db-password)

Two tiers:

1. Standard (free): up to 10,000 parameters, 4 KB max value
2. Advanced (paid): up to 100,000 parameters, 8 KB value, parameter policies

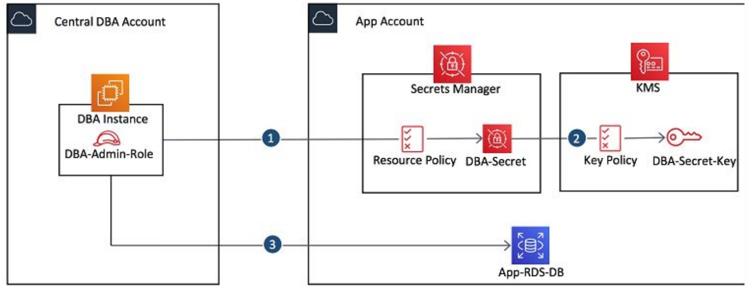
Use Case:
A development team uses SSM Parameter Store to securely store configuration values like DB passwords and API tokens. Lambda functions fetch these values at runtime using the GetParameter API, with decryption handled automatically via KMS.

AWS Secrets Manager

AWS Secrets Manager is a managed service designed for storing, rotating, and managing secrets such as database credentials, API keys, and other sensitive configuration data. It offers built-in rotation using AWS Lambda and integrates seamlessly with services like RDS, Redshift, and DocumentDB.

Key Features:

- Store secrets securely with KMS encryption
- Automate secret rotation (e.g., every X days)
- Native integration with Amazon RDS (MySQL, PostgreSQL, Aurora)
- Secrets can be fetched via SDK, CLI, or the console
- Supports multi-region replication of secrets for high availability
- Can trigger Lambda functions to generate new secrets on rotation



Use Case:
A company stores credentials for its RDS PostgreSQL database in AWS Secrets Manager. It configures automatic rotation every 30 days via a Lambda function, eliminating manual rotation tasks and reducing the risk of credential leakage.

AWS Certificate Manager (ACM)

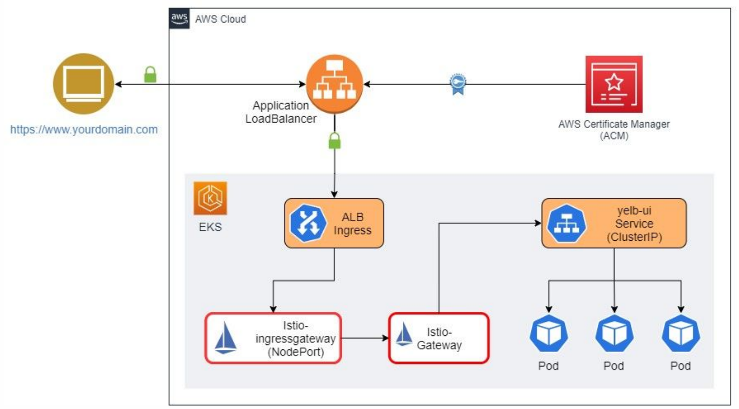
AWS Certificate Manager (ACM) is a service that simplifies the provisioning, management, and deployment of TLS/SSL certificates for securing network communications and protecting websites. ACM supports both public and private certificates and handles automatic renewal for certificates it issues.

Key Features:

- Provision public or private certificates for your domain names
- Automatic renewal of ACM-issued public certs
- Validation options: DNS (recommended) or Email
- Certificates can't be exported (not usable on EC2 directly)
- Certificate expiration notifications via EventBridge

Integrated with services like:

- Elastic Load Balancers (ALB/NLB/CLB)
- API Gateway
- CloudFront


AWS Cloud
https://www.yourdomain.com
Application LoadBalancer
AWS Certificate Manager (ACM)
EKS
ALB Ingress
yelb-ui Service (ClusterIP)
Istio-ingressgateway (NodePort)
Istio-Gateway
Pod
Pod
Pod

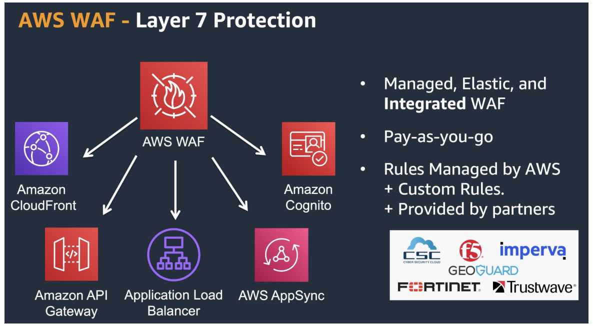
Use Case:
A company hosts a public web application behind an Application Load Balancer. Using ACM, they provision a public certificate for www.example.com, enable automatic renewal, and ensure HTTPS termination at the ALB without manually managing certificates.

AWS WAF (Web Application Firewall)

AWS WAF is a web application firewall that protects applications from common web exploits and bots. It allows you to create customizable rules to block, allow, or monitor (count) HTTP and HTTPS requests based on conditions like IP addresses, HTTP headers, query strings, URI paths, and more.

Key Features:

- Protects ALB, API Gateway, CloudFront, and App Runner
- Define rules using Web ACLs (Access Control Lists)
- Supports managed rule groups (AWS & 3rd-party via AWS Marketplace)
- Rate limiting and bot control features
- Real-time metrics with Amazon CloudWatch
- Integrated with AWS Firewall Manager for centralized rule management


AWS WAF - Layer 7 Protection
AWS WAF
Amazon CloudFront
Amazon Cognito
Amazon API Gateway
Application Load Balancer
AWS AppSync
Managed, Elastic, and Integrated WAF
Pay-as-you-go
Rules Managed by AWS + Custom Rules. + Provided by partners
CSC
f5
imperva
GEOGUARD
FORTINET
Trustwave

Use Case:

An e-commerce company uses AWS WAF in front of its CloudFront distribution. It blocks requests from known bad IPs and applies rate limiting to prevent brute-force login attempts, ensuring application availability and reducing malicious traffic.

AWS Shield

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service. It comes in two tiers:

1. Shield Standard (free): Automatically enabled for all AWS customers, protecting against most common network and transport layer (Layer 3/4) attacks such as SYN/UDP floods or reflection attacks.

2. Shield Advanced (paid): Provides enhanced detection and mitigation for large and sophisticated attacks across services including EC2, ELB, CloudFront, Global Accelerator, and Route 53. It also offers 24/7 access to the AWS DDoS Response Team (DRT), cost protection for scaling during an attack, and automatic web ACL tuning via WAF.

Use Case:
A media streaming company experiences periodic large-scale DDoS attempts targeting its CloudFront distribution. By subscribing to Shield Advanced, the company gets 24/7 support from AWS DRT, automatic mitigation of both network- and application-layer DDoS attacks, and financial protection against scaling charges during an attack.

AWS Firewall Manager

AWS Firewall Manager is a security management service that centralizes the creation, management, and enforcement of firewall rules across your AWS Organization. It integrates with AWS WAF, AWS Shield Advanced, Security Groups, AWS Network Firewall, and Route 53 Resolver DNS Firewall.

Key Features:

- Define security policies for:
- WAF rules (ALB, API Gateway, CloudFront)
- Shield Advanced protections (ELB, CloudFront, Global Accelerator, Route 53)
- Security Groups (EC2, ALB, ENI)
- AWS Network Firewall (VPC-level)
- Route 53 Resolver DNS Firewall
- Automated policy enforcement on new and existing resources across all accounts
- Centralized audit and compliance reporting
- Policies are managed per region but can cover all accounts in the Organization

Use Case:

A global enterprise needs consistent WAF and Shield settings across 50 AWS accounts. Using Firewall Manager, the security team defines a WAF policy with custom rules and a Shield Advanced policy, and ensures they're automatically applied to all current and future CloudFront distributions and load balancers in every account.

WAF vs. Firewall Manager vs. Shield

1. AWS WAF: Provides granular, application-layer (Layer 7) protection through customizable Web ACLs. Best when you need to block specific HTTP/HTTPS patterns, SQL injection, or XSS attacks on individual resources.

2. AWS Shield: Focuses on DDoS protection at the network/transport layer (Layer 3/4) and, with Shield Advanced, at the application layer (Layer 7) via automatic WAF rule tuning. It offers rapid mitigation of volumetric attacks and 24/7 DDoS response support.

3. AWS Firewall Manager: Orchestrates and automates the deployment of WAF rules and Shield Advanced protections (as well as Security Groups and AWS Network Firewall policies) across all accounts in an AWS Organization. Ideal for large environments that require consistent, centralized security policy enforcement.

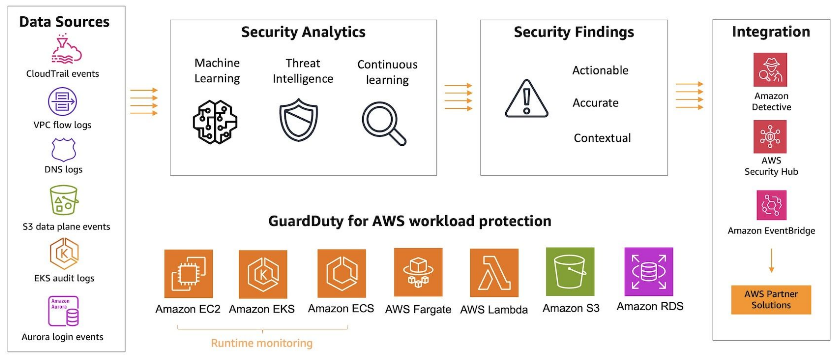
Use Case:
A media-streaming company uses WAF to block malicious HTTP requests, Shield Advanced to absorb large DDoS floods, and Firewall Manager to enforce those protections across all its AWS accounts for its streaming platform.

Amazon GuardDuty

Amazon GuardDuty is a continuous security monitoring service that uses machine learning, anomaly detection, and integrated threat intelligence to identify suspicious activity and unauthorized behavior in your AWS environment. It analyzes data sources like:

- CloudTrail management and data events (unauthorized API calls, suspicious deployments)
- VPC Flow Logs (unusual network traffic)
- DNS query logs (compromised instances communicating over DNS)

Use Case:
A fintech company enables GuardDuty across all accounts to automatically detect anomalous API calls and compromised EC2 instances, then routes findings via EventBridge to an automated Lambda remediation workflow.

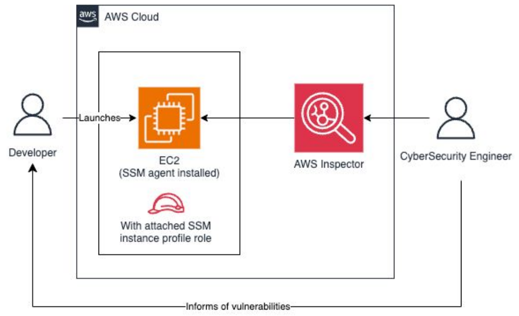


AWS Inspector

Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. It assesses applications for vulnerabilities or deviations from security best practices by analyzing:

- Network configurations (security group rules, open ports)
- Operating system configurations (CVE vulnerabilities in EC2 AMIs)
- Runtime behavior (with Inspector Classic 2.0 agents for containerized workloads)
- Findings are scored and prioritized, with detailed remediation guidance.

Use Case:
A healthcare provider runs Inspector scans weekly on its EC2 instances and container images to detect missing OS patches and CVEs, then integrates findings into its DevOps pipeline for automated patch deployments.



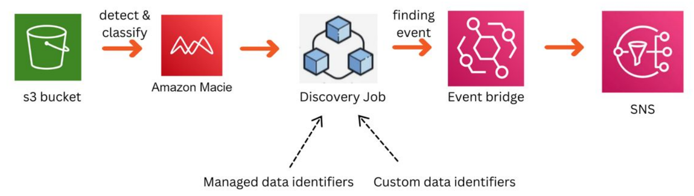
Amazon Macie

Amazon Macie is a data security and privacy service that uses machine learning and pattern matching to discover, classify, and protect sensitive data stored in Amazon S3. It automatically recognizes PII (e.g., names, credit card numbers), assesses data risk, and generates detailed dashboards and alerts for anomalous access or unintended public exposure.

Key Features:

- Sensitive data discovery (PII, PHI)
- Automatic and scheduled S3 bucket assessments
- Customizable data classification jobs
- Alerting via CloudWatch Events or EventBridge
- Detailed dashboards on data visibility and access patterns

Use Case:
A retail company uses Macie to automatically scan its S3 data lake nightly, classify customer PII, and alert the security team if any bucket policy change exposes sensitive files publicly.



FlipTheScript

Security Exam Questions

A healthcare startup stores medical images in an S3 bucket and must ensure that the images are encrypted at rest with a customer-managed key that rotates every 90 days. They also want to minimize cross-region latency by using the same key in two regions. Which combination of services and configurations meets these requirements?

A. Use SSE-S3 with AWS owned keys and enable cross-region replication on the bucket.
B. Create a customer-managed symmetric CMK in KMS with automatic 90-day rotation, enable it as a multi-region primary key in the two regions, and configure S3 SSE-KMS encryption with that CMK.
C. Use Secrets Manager to store the encryption key and have the application fetch it at upload time, then use SSE-C in S3.
D. Use AWS managed KMS keys (aws/s3) and set a lifecycle rule to rotate the bucket every 90 days.

A global e-commerce platform experiences occasional Layer 7 attacks and periodic large-scale DDoS attempts. They need application-layer protection, network-layer DDoS mitigation, and centralized policy management across 20 AWS accounts. Which combination of services should they implement?

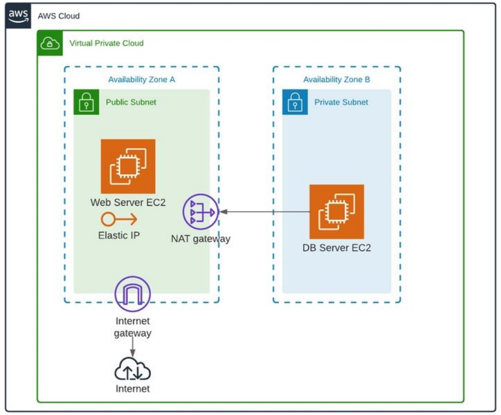
A. Deploy AWS WAF on their ALBs, enable Shield Standard, and manually replicate WAF configurations per account.
B. Use AWS WAF for HTTP filtering, subscribe to Shield Advanced for DDoS protection, and configure AWS Firewall Manager to centrally enforce WAF and Shield policies across all accounts.
C. Use AWS Network Firewall in each VPC, enable Inspector for runtime checks, and store rules in SSM Parameter Store.
D. Configure GuardDuty for threat detection, ACM for certificate management, and Secrets Manager for storing firewall rules.

Amazon VPC (Virtual Private Cloud)

A Virtual Private Cloud (VPC) is your own isolated network within AWS. You define its IPv4 (and optionally IPv6) address range as a CIDR block, then create one or more subnets mapped to Availability Zones. Within a VPC, you control:

- Subnet layout: public, private, and/or isolated subnets
- Routing: via custom route tables to direct traffic between subnets, gateways, and on-premises networks
- Gateways: Internet Gateways for public traffic and NAT devices or gateways for outbound-only internet access
- Security: Security Groups (stateful) and Network ACLs (stateless) to filter traffic at the instance and subnet levels

A VPC gives you full control of your network topology—just as if you were running your own data center, but with AWS's scalability and availability.



FlipTheScript

Subnets

A subnet is a segment of your VPC’s IP address range where you place groups of resources. Each subnet maps to a single Availability Zone and can be designated as:

Public Subnet: Route table points to an Internet Gateway (IGW), allowing instances to receive direct internet traffic.

Private Subnet: No direct route to IGW; instances can initiate outbound internet via NAT but are not directly reachable from the internet.

Isolated Subnet: No internet connectivity at all, used for highly sensitive workloads.

When you define a subnet, you specify a CIDR block carved from the VPC’s range. AWS supports IPv4 subnet sizes from /16 (65,536 addresses) down to /28 (16 addresses). Note that AWS reserves 5 IP addresses in each subnet for networking purposes, so effective available addresses are 11 for a /28.

Use Case:
An e-commerce application places web servers in a public subnet, application servers in a private subnet (using a NAT Gateway for updates), and database servers in an isolated subnet with no internet access.

Internet Gateway (IGW)

An Internet Gateway (IGW) is a fully managed VPC component that provides a bidirectional path between your VPC and the public internet. It is horizontally scaled, redundant across Availability Zones, and designed for high availability and fault tolerance. Key characteristics include:

- Attachment and Association: Each VPC can have exactly one IGW, which you "attach" in the VPC settings. You then configure your subnet route tables to send traffic destined for 0.0.0.0/0 (all IPv4 addresses) or ::/0 (all IPv6 addresses) to the IGW. Only subnets whose routes point to the IGW become public subnets.
- NAT for Public IPs: For IPv4, the IGW performs network address translation (NAT) so that instances with a public IP address can communicate with the internet. For IPv6, the IGW provides direct end-to-end addresses without translation.
- Scalability & Availability: Being a managed service, the IGW automatically scales to handle your bandwidth needs and remains operational even if individual Availability Zones experience disruptions.
- No Route Propagation: Unlike VPN or Direct Connect attachments, the IGW does not support route propagation; all routes must be explicitly defined in each route table.
- Security Controls: While the IGW itself does not filter traffic, you secure the traffic at the instance level with Security Groups and at the subnet level with Network ACLs.

Use Case:
A public-facing web tier uses an IGW to allow EC2 instances in a public subnet to both initiate outbound HTTPS calls (for API calls or updates) and receive inbound HTTP/HTTPS requests from end users, while keeping backend services in private subnets unreachable directly from the internet.

Route Tables

A Route Table is a set of rules ("routes") that control how network traffic is directed within your VPC. Each route specifies a destination CIDR block and a target (such as an Internet Gateway, NAT Gateway, VPC peering connection, or virtual private gateway). Key points:

- Main vs. Custom: Every VPC has a single main route table which all subnets use by default, unless you explicitly associate a subnet with a custom route table.
- Route priority: Routes are evaluated in order of longest prefix match, so a more specific CIDR (e.g., 10.0.1.0/24) takes precedence over a broader one (e.g., 10.0.0.0/16).
- Common Targets:
    - Internet Gateway (igw-…): for public subnets
    - NAT Gateway/Instance (nat-…/eni-…): for private subnets to access the internet
    - Peering Connection (pcx-…): for VPC peering
    - Virtual Private Gateway (vgw-…): for Site-to-Site VPN
- No Implicit Routes: You must manually add routes for any destination outside the local VPC CIDR; the local VPC CIDR is automatically routed to local.

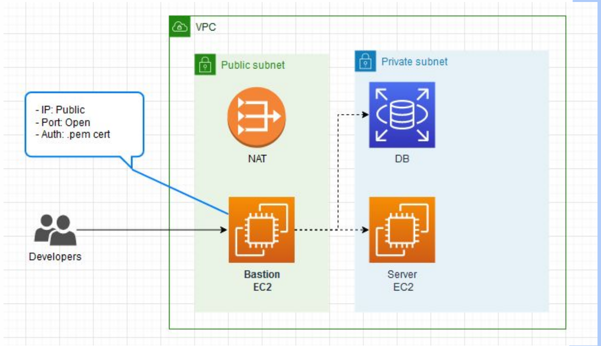
Use Case:
An application's private subnet has a custom route table with two routes: one to the NAT Gateway for 0.0.0.0/0 (internet access) and one to a VPC peering connection for 10.1.0.0/16 (communication with a shared services VPC).

Bastion Hosts

A Bastion Host is a hardened EC2 instance placed in a public subnet that serves as the sole entry point for SSH or RDP access to instances in private subnets. Key considerations:

- Security: Lock down the Bastion with a strict Security Group—only allow SSH/RDP from known IPs.
- Auditability: Enable detailed logging (CloudTrail, Session Manager/IAM) to track who accessed which private instances.
- High Availability: You can deploy multiple Bastions across AZs behind an Elastic Load Balancer or use AWS Systems Manager Session Manager to remove the need for Bastions entirely.

Use Case:
An enterprise restricts all administrative SSH/RDP to database servers in private subnets by requiring operators to first connect to a Bastion Host in the public subnet, ensuring no direct internet access to critical resources.

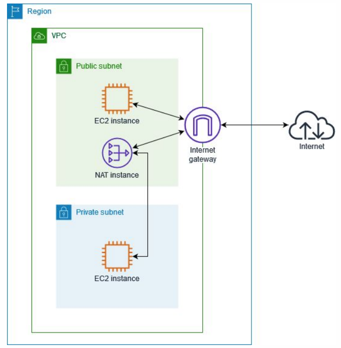


NAT Instances

A NAT Instance is a self-managed EC2 instance configured to enable outbound internet access from private subnets. It performs network address translation (NAT) by routing traffic from private instances through its public IP. Key points:

- Configuration: Launch a Linux AMI, enable IP forwarding, and assign an Elastic IP.
- Scalability & Availability: User must manage Auto Scaling, health checks, and replace single points of failure manually.
- Cost: Lower hourly cost than NAT Gateway but operational overhead for maintenance and patching.
- Security: Control access via Security Groups on the NAT instance.

Use Case:
A startup uses a NAT Instance to allow private-app servers to download OS updates and patches from the internet, while maintaining inbound restrictions—until they migrate to a managed NAT Gateway for simpler scaling.

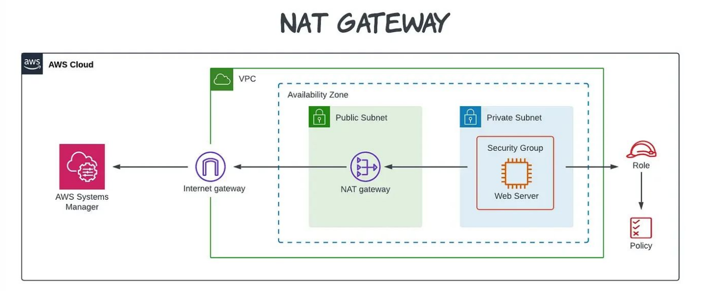


NAT Gateway

A NAT Gateway is a fully managed AWS service that provides scalable, highly available outbound internet access for instances in private subnets. Key aspects:

- Managed Scaling: AWS automatically scales throughput (up to 45 Gbps) and handles redundancy across multiple Availability Zones.
- Simplicity: No maintenance, patching, or instance management; you pay per hour and per GB of data processed.
- High Availability: Create one NAT Gateway per AZ for resilience; if you only deploy one, a failure in that AZ cuts off internet access for that subnet.
- Elastic IP: Each NAT Gateway requires an Elastic IP; route tables direct 0.0.0.0/0 traffic from private subnets to the NAT Gateway.

Use Case:
A fintech application's private compute nodes use a NAT Gateway to fetch security updates and call external payment APIs without exposing them to inbound internet traffic.



Security Groups (SG) & Network ACLs (NACL)

Security Groups: Virtual firewalls applied at the instance level. They are stateful: allowed inbound traffic automatically permits response traffic outbound, and vice versa. Rules specify protocols, ports, and CIDR ranges. Instances can belong to multiple SGs, and SG changes apply immediately.

Network ACLs: Stateless filters applied at the subnet level. You must explicitly allow inbound and outbound traffic; return traffic does not automatically flow. NACLs process rules in ascending order by rule number and have separate rule sets for ingress and egress. Useful for broad subnet-level controls and denial rules (e.g., blacklisting).

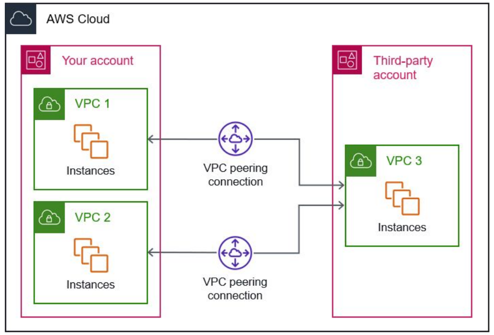
Use Case:
A web tier uses SGs to allow HTTP/HTTPS from anywhere and SSH from a bastion IP. The associated private subnet has a NACL that blocks all outbound traffic to known malicious IP ranges as an extra layer of defense.

VPC Peering

VPC Peering establishes a private, point-to-point network connection between two VPCs, enabling resources in either VPC to communicate using private IPs—without traversing the public internet. Peering works across AWS accounts and Regions (inter-region peering), is non-transitive (A↔B and B↔C doesn't imply A↔C), and requires adding corresponding route table entries in each VPC.

- Non-transitive: You need direct peer connections for each pair.
- Security: Traffic stays on the AWS global network; controlled by security groups and NACLs—no additional gateway required.
- Cost: Peering traffic is charged at data transfer rates (inter-AZ or inter-region fees may apply).

Use Case:
A development VPC peers with a shared-services VPC so application servers can securely access a central logging and metrics cluster over private IPs, avoiding public endpoints.



FlipTheScript

VPC Endpoints

VPC Endpoints enable private connections between your VPC and AWS services or VPC resources without requiring an Internet Gateway, NAT device, VPN, or AWS Direct Connect. Traffic stays within the AWS network.

Interface Endpoints:

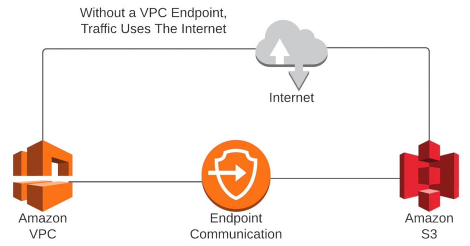
- Powered by AWS PrivateLink.
- Elastic Network Interfaces (ENIs) in subnets.
- Use for services like S3, DynamoDB (as interface), Kinesis, SNS, SageMaker, and partner services.

Gateway Endpoints:

- Target specified in route tables.
- Currently only for S3 and DynamoDB.
- Appear as a gateway in route table entries.

Gateway Load Balancer Endpoints:

- For integrating third-party virtual appliances behind a Gateway Load Balancer.

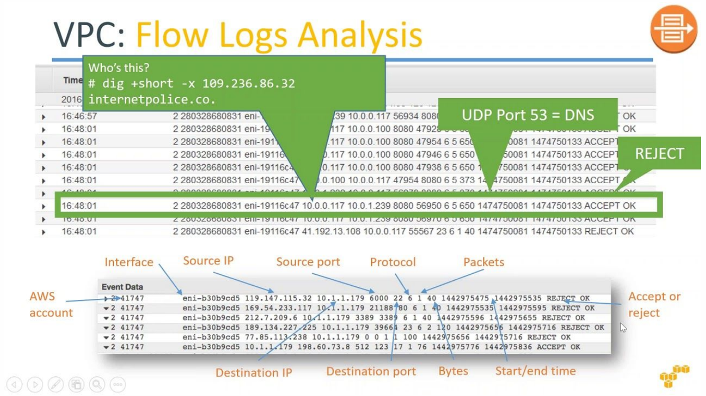


Use Case:
An analytics VPC creates a gateway endpoint for S3 so EMR clusters can read/write data without traversing NAT or IGW, and an interface endpoint for Kinesis to securely ingest streaming data.

VPC Flow Logs

VPC Flow Logs capture metadata about the IP traffic going to and from network interfaces in your VPC. You can publish logs to CloudWatch Logs or S3 for analysis. Flow Logs record fields like source/destination IP, ports, protocol, packet and byte counts, timestamps, and accept/reject actions based on security groups/NACLs.

Levels:
Network Interface (ENI)
Subnet
VPC



```
# dig +short -x 109.236.86.32
internetpolice.co.
```

```
Time
2016
16:46:57    2 280328680831 eni-          39 10.0.0.117 56934 8080                       T OK
16:48:01    2 280328680831 eni-19        117 10.0.0.100 8080 47928                      T OK
16:48:01    2 280328680831 eni-191       .117 10.0.0.100 8080 47954 6 5 650    0081 1474750133 ACCEPT
16:48:01    2 280328680831 eni-19116     0.117 10.0.0.100 8080 47946 6 5 650    50081 1474750133 ACCEPT
16:48:01    2 280328680831 eni-19116c4   0.117 10.0.0.100 8080 47938 6 5 650 1  750081 1474750133 ACCEPT
16:48:01    2 280328680831 eni-19116c47  0.0.100 10.0.0.117 47954 8080 6 5 373 14 4750081 1474750133 ACCEPT
16:48:01    2 280328680831 eni-19116c47 10.0.0.117 10.0.1.239 8080 56950 6 5 650 1474750081 1474750133 ACCEPT OK
16:48:01    2 280328680831 eni-19116c47 41.192.13.108 10.0.0.117 55567 23 6 1 40 1474750081 1474750133 REJECT OK
```

```
Event Data
▸ 2 41747   eni-b30b9cd5 119.147.115.32 10.1.1.179 6000 22 6 1 40 1442975475 1442975535 REJECT OK
▾ 2 41747   eni-b30b9cd5 169.54.233.117 10.1.1.179 21188 80 6 1 40 1442975535 1442975595 REJECT OK
▾ 2 41747   eni-b30b9cd5 212.7.209.6 10.1.1.179 3389 3389 6 1 40 1442975596 1442975655 REJECT OK
▾ 2 41747   eni-b30b9cd5 189.134.227.225 10.1.1.179 39664 23 6 2 120 1442975656 1442975716 REJECT OK
▾ 2 41747   eni-b30b9cd5 77.85.113.238 10.1.1.179 0 0 1 1 100 1442975656 1442975716 REJECT OK
▾ 2 41747   eni-b30b9cd5 10.1.1.179 198.60.73.8 512 123 17 1 76 1442975776 1442975836 ACCEPT OK
```

Use Cases for Analysis:
Troubleshooting connectivity issues
Auditing security group and NACL rule effects
Identifying unexpected or malicious traffic patterns
Cost allocation by tracking data transfer volumes

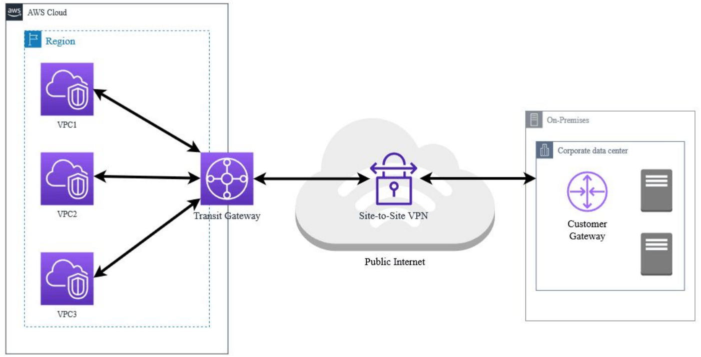
Use Case:
A security operations team enables VPC Flow Logs at the VPC level and sends them to a centralized S3 bucket. They use Athena to query rejected traffic patterns, quickly identifying misconfigured NACL rules.

Site-to-Site VPN

A Site-to-Site VPN creates an encrypted IPsec tunnel between your on-premises network (via a Customer Gateway, CGW) and your VPC (via a Virtual Private Gateway, VGW). AWS manages one end (VGW) and you configure the CGW on your firewall/router. You can have two tunnels for redundancy, each with unique IP addresses and pre-shared keys.

- Traffic Routing: Update your VPC route table to point the on-premises CIDR to the VGW.
- High Availability: AWS maintains both tunnels; you configure your customer side to failover automatically.
- Bandwidth/Performance: Limited by your on-premises VPN device and internet link.

Use Case:
A retail chain sets up a Site-to-Site VPN so its store locations securely access centralized inventory databases in a VPC, using two tunnels for resilience.



VPN CloudHub

VPN CloudHub allows you to securely connect multiple on-premises sites to a single VPC using a hub-and-spoke model. Each site's Customer Gateway (CGW) establishes a Site-to-Site VPN to the same Virtual Private Gateway (VGW). The VGW routes traffic between spokes, enabling inter-site communication over the AWS backbone. All tunnels use separate pre-shared keys for security.

Hub-and-Spoke: One VGW as the hub, multiple CGWs as spokes.
Transitive Routing: Unlike VPC Peering, CloudHub supports transitive traffic between on-prem sites via the VGW.
Simplicity: No need for additional transit appliances; uses existing VPN infrastructure.

Use Case:
A distributed enterprise with three regional offices uses VPN CloudHub to connect all sites to a central VPC, allowing workloads at each location to securely exchange data with each other through the AWS network.

AWS Direct Connect

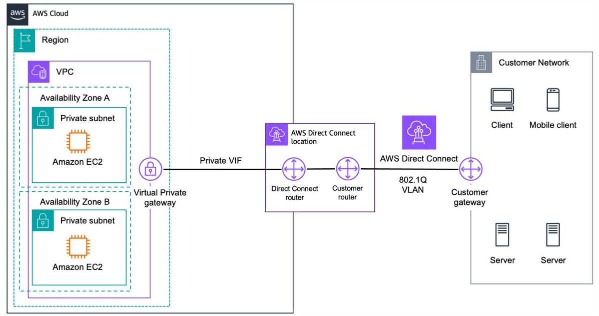
AWS Direct Connect provides a dedicated, private network connection between your on-premises data center and AWS. It bypasses the public internet, offering more consistent network performance, lower latency, and potentially reduced bandwidth costs. You order a port at a Direct Connect location, then configure a virtual interface (public for AWS services or private for your VPC via a Virtual Private Gateway or Transit Gateway).

Connection Types:

1. Dedicated Connection: Physical port (1 Gbps, 10 Gbps) on AWS router.
2. Hosted Connection: Partner-provisioned sub-port (50 Mbps–10 Gbps).

Virtual Interfaces:

- Private VIF: Connects to a VPC.
- Public VIF: Accesses AWS public services (S3, DynamoDB).
- Transit VIF: Used with Transit Gateway for multi-VPC connectivity.



Resilience: Support for active-active or active-passive connections by using multiple ports or locations.

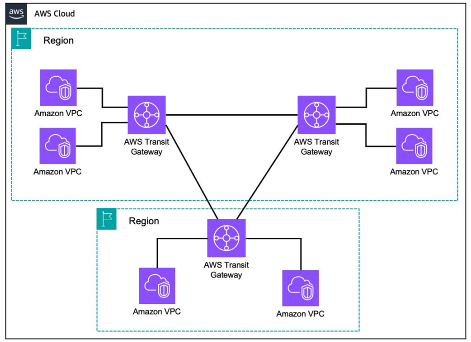
Use Case:
A financial institution establishes AWS Direct Connect to a Direct Connect location near its data center, then sets up a private VIF to its production VPC, ensuring high-throughput, low-latency connectivity for trading applications.

AWS Transit Gateway

AWS Transit Gateway is a scalable, central hub that connects VPCs and on-premises networks through a single gateway. It simplifies network management by consolidating hundreds of VPC and VPN connections into a single managed resource. Transit Gateway supports:

- Hub-and-Spoke Model: VPCs and VPNs attach to the Transit Gateway, which routes traffic between attachments.
- Route Tables: Multiple TGW route tables allow segmentation of traffic (e.g., prod vs. dev).
- Inter-Region Peering: Transit Gateways in different regions can be peered.
- Multicast Support: For applications like media distribution.
- Bandwidth: Up to 50 Gbps per attachment (varies by region).

Use Case:
A global SaaS provider uses Transit Gateway to interconnect 20 VPCs across three regions and three on-premises sites, centralizing route management and reducing complexity compared to multiple VPC peering connections.

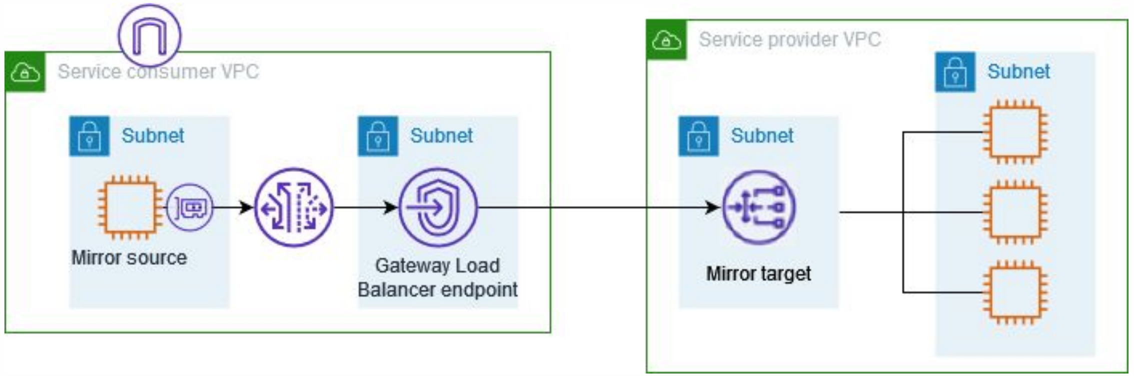


VPC Traffic Mirroring

VPC Traffic Mirroring lets you capture and inspect network traffic at the packet level from ENIs in your VPC. You mirror inbound and/or outbound traffic from a source ENI to a target (an IP address, Network Load Balancer, or an EC2 monitoring appliance) for deep packet inspection, forensics, or threat detection.

- Filters: Define traffic to mirror by protocol, ports, and CIDR ranges.
- Targets: Mirror sessions can send to a monitoring EC2 instance or a Load Balancer in another VPC or on-premises.
- Performance: Capable of handling tens of Gbps per source ENI; you pay per GB mirrored.

Use Case:
A security team mirrors traffic from all database ENIs to a packet-inspection appliance to detect SQL injection attempts and anomalous queries in real time.



IPv6 in AWS

AWS VPCs support dual-stack addressing—each VPC can be assigned an Amazon-provided IPv6 CIDR block (commonly a /44) or you can Bring Your Own IP (BYOIP).

Within the VPC:

- Subnets inherit a /64 from the VPC's /44 and can be public or private.
- Instances receive both an IPv4 and an IPv6 address.
- Routing uses route tables—add a ::/0 route to an Internet Gateway for public subnets, or to an Egress-Only IGW for IPv6-only egress from private subnets.
- Security Groups and NACLs support IPv6 rules alongside IPv4, allowing filtering by IPv6 CIDR, ports, and protocols.
- Because IPv6 space is abundant, you generally eliminate the need for NAT on IPv6 traffic, enabling direct end-to-end connectivity that reduces latency and simplifies address management.

Use Case:
A global gaming service assigns an Amazon-provided /44 IPv6 CIDR to its production VPC. It creates multiple /64 subnets—public for game servers, private for analytics—and allows players on IPv6-only networks to connect directly to game servers without NAT, improving performance and scalability.

Egress-Only Internet Gateway

An Egress-Only Internet Gateway is a VPC component designed specifically for IPv6 traffic. Unlike a standard Internet Gateway, which allows bidirectional IPv6 communication, an EOIGW only permits outbound connections from your VPC to the internet on IPv6 (::/0) and automatically blocks any inbound-initiated traffic. This is ideal for scenarios where you want your instances to access external resources without exposing them to incoming requests.

To use an EOIGW, you:

1. Attach one EOIGW to your VPC.
2. Update the route table for each private subnet by adding an IPv6 default route (::/0) that points to the EOIGW.
3. Ensure Security Groups and NACLs allow the required outbound traffic—no additional rules are needed to block inbound IPv6, as the EOIGW enforces it at the gateway level.

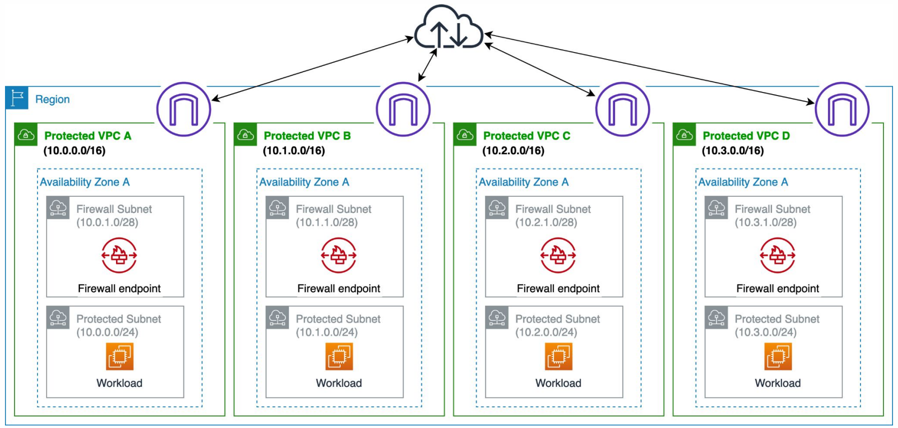
- Because IPv6 addresses are globally routable and abundant, the EOIGW removes the need for NAT for IPv6 traffic, reduces network complexity, and prevents accidental exposure of private workloads.

Use Case:
A microservices platform uses an EOIGW so its dual-stack services can download container image updates over IPv6 while remaining completely unreachable from the public internet via IPv6.

AWS Network Firewall

AWS Network Firewall is a fully managed, stateful network firewall service for your VPC that protects both inbound and outbound traffic at the subnet level. You define firewall policies—composed of stateful and stateless rule groups—that can perform signature-based intrusion prevention (IPS), domain list filtering, and TLS deep inspection. The service scales automatically across Availability Zones to handle tens of gigabits per second, and integrates with CloudWatch or S3 for centralized logging of alerts and flow data.

Use Case:
A financial services firm routes all east-west traffic within its production VPC through Network Firewall to block known malicious IPs, enforce intrusion-prevention signatures, and maintain detailed logs of dropped packets for compliance.



Networking Exam Questions

You must allow HTTP/HTTPS from anywhere, SSH only from a management IP, and block outbound traffic to malicious IPs. Which setup is correct?

A. Security Group for HTTP/HTTPS (0.0.0.0/0) and SSH (management IP), plus a Network ACL denying outbound to the malicious IPs.

B. Security Group for HTTP/HTTPS and SSH (0.0.0.0/0), plus a Network ACL denying outbound to the malicious IPs.

C. Network ACL for HTTP/HTTPS (0.0.0.0/0) and SSH (management IP), plus a Security Group blocking outbound.

D. Security Group for HTTP/HTTPS and SSH (management IP) only.

A company needs a private network connection between its on-premises data center and AWS that bypasses the public internet. Which solution should it choose?

A. A Site-to-Site VPN over the internet.

B. An AWS Transit Gateway attachment.

C. An AWS Direct Connect dedicated connection with a Private VIF to the VPC.

D. A NAT Gateway in a public subnet.