Chapter 4

Designing Secure Workloads and Applications

This chapter covers the following topics:

- **Securing Network Infrastructure**
- **Amazon Cognito**
- **External Connections**
- **Amazon GuardDuty**
- **Amazon Macie**
- **Security Services for Securing Workloads**

This chapter covers content that's important to the following exam domain and task statement:

Domain 1: Design Secure Architectures

Task Statement 2: Design secure workloads and applications

Workload and application security at AWS refers to the measures and controls that are implemented to protect the data and associated cloud services used to process, store, and transmit data. This includes implementing security controls and practices to protect against potential threats and vulnerabilities that could compromise the security of workloads.

To properly secure workload network infrastructure in the cloud, organizations must design and deploy subnets and route tables, security groups (SG), and network access control lists (ACLs) to protect workload infrastructure hosted on subnets in each virtual private cloud (VPC). There are also security services to consider deploying at each edge location, including the AWS Web Application Firewall

(WAF), AWS Shield Standard, and AWS Shield Advanced, to help protect workloads that are exposed to the Internet.

Workload security can also utilize a combination of security controls provided by the AWS, such as utilizing Amazon Cognito, which provides authentication, authorization, and user management for your web and mobile applications. There are also several AWS security services that can assist in securing the associated workload, including Amazon Macie, Amazon GuardDuty, AWS CloudTrail, AWS Secrets Manager, Amazon Inspector, and AWS Trusted Advisor.

"Do I Know This Already?"

The "Do I Know This Already?" quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the "Exam Preparation Tasks" section. If you are in doubt about your answers to these questions or your assessment of your knowledge of the topics, read the entire chapter. **Table 4-1** lists the major headings in this chapter and their corresponding "Do I Know This Already?" quiz questions. You can find the answers in **Appendix A**, "**Answers to the 'Do I Know This Already?' Quizzes and Q&A Sections.**"

Table 4-1 "Do I Know This Already?" Section-to-Question Mapping

Foundation Topics Section	Questions
Securing Network Infrastructure	1, 2
Amazon Cognito	3, 4
External Connections	5, 6
Amazon GuardDuty	7, 8
Amazon Macie	9, 10

Foundation Topics Section	Questions
Security Services for Securing Workloads	11, 12

Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

---

1. Which route table is associated with each new subnet at creation?

1. Custom route table
2. None, because route tables are not automatically associated
3. The main route table
4. The network access control list

2. What is a security group's job?

1. To deny incoming and outgoing traffic
2. To control incoming and outgoing traffic
3. To block incoming and outgoing traffic
4. To explicitly deny incoming and outgoing traffic

3. What services are analyzed by Amazon GuardDuty?

1. Amazon EFS and FSx for Windows File Server logs
2. AWS Route 53 logs and Amazon VPC flow logs
3. Amazon Inspector logs and AWS Config rules
4. Amazon RDS logs and Amazon EC2 system logs

4. What AWS service is used to automate remediation of Amazon GuardDuty issues?

1. Amazon CloudTrail
2. Amazon CloudWatch
3. AWS Lambda
4. Amazon Route 53

5. What VPN service needs to be installed before connecting to a VPC?

1. AWS Direct Connect
2. AWS Customer Gateway
3. Virtual Private Gateway
4. AWS VPN Cloud Hub

6. What type of network connection is an AWS Direct Connect connection?

1. Public
2. Single-mode fiber
3. VPN
4. IPsec

7. What type of data records are analyzed by Amazon Macie?

1. Amazon S3 buckets
2. Amazon EFS
3. Amazon S3 Glacier
4. Amazon FSx for Windows File Server

8. What process is used by Amazon Macie to begin a data analysis?

1. Administrator
2. Schedule
3. Job
4. Task

9. What does AWS Cognito use to authenticate end users to a user pool?

1. Username/password
2. Username/SNS
3. Username/email/phone number
4. MFA and password

10. What does Amazon Cognito require to authenticate mobile application users?

1. User pool
2. Identity pool
3. Certificates
4. MFA

11. Which of the following can you use to retain AWS CloudTrail events permanently?

1. AWS Lambda function
2. A custom trail
3. AWS Step Function
4. None of these; events can be retained for only 90 days

12. Which of the following does Amazon Inspector evaluate?

1. Amazon S3 buckets
2. Amazon Elastic Container Service
3. Amazon EC2 instances
4. Amazon Relational Database Service

Foundation Topics

Securing Network Infrastructure

Workloads can be protected with a variety of AWS security services. Connections to workloads running at AWS can utilize both public

and private connections using one or more of the following AWS networking services (see **Figure 4-1**):

- **Internet connections (HTTPS/HTTP) to public-facing workloads:** An Internet gateway must be attached to the VPC hosting the workload, and the VPC's security groups and network ACLs must be configured to allow the appropriate traffic to flow through the subnets, load balancer, and web servers.
- **AWS Direct Connect:** Establish a dedicated network connection from your on-premises data center to your VPC (Virtual Private Cloud) at AWS.
- **An AWS VPN (Virtual Private Network) connection:** Provide a secure encrypted connection between an on-premises network and VPC at AWS. VPN connections allow access to your AWS resources and workloads as if they were on your on-premises network.
- **Edge locations for accessing cached data records using Amazon CloudFront, Amazon's content delivery network (CDN):** Edge locations are located at the edge of the AWS cloud and serve content to users more quickly and efficiently.
- **AWS Global Accelerator network:** Use Amazon's global network of edge locations to improve the performance of Internet applications routing traffic from users to the optimal AWS endpoint.

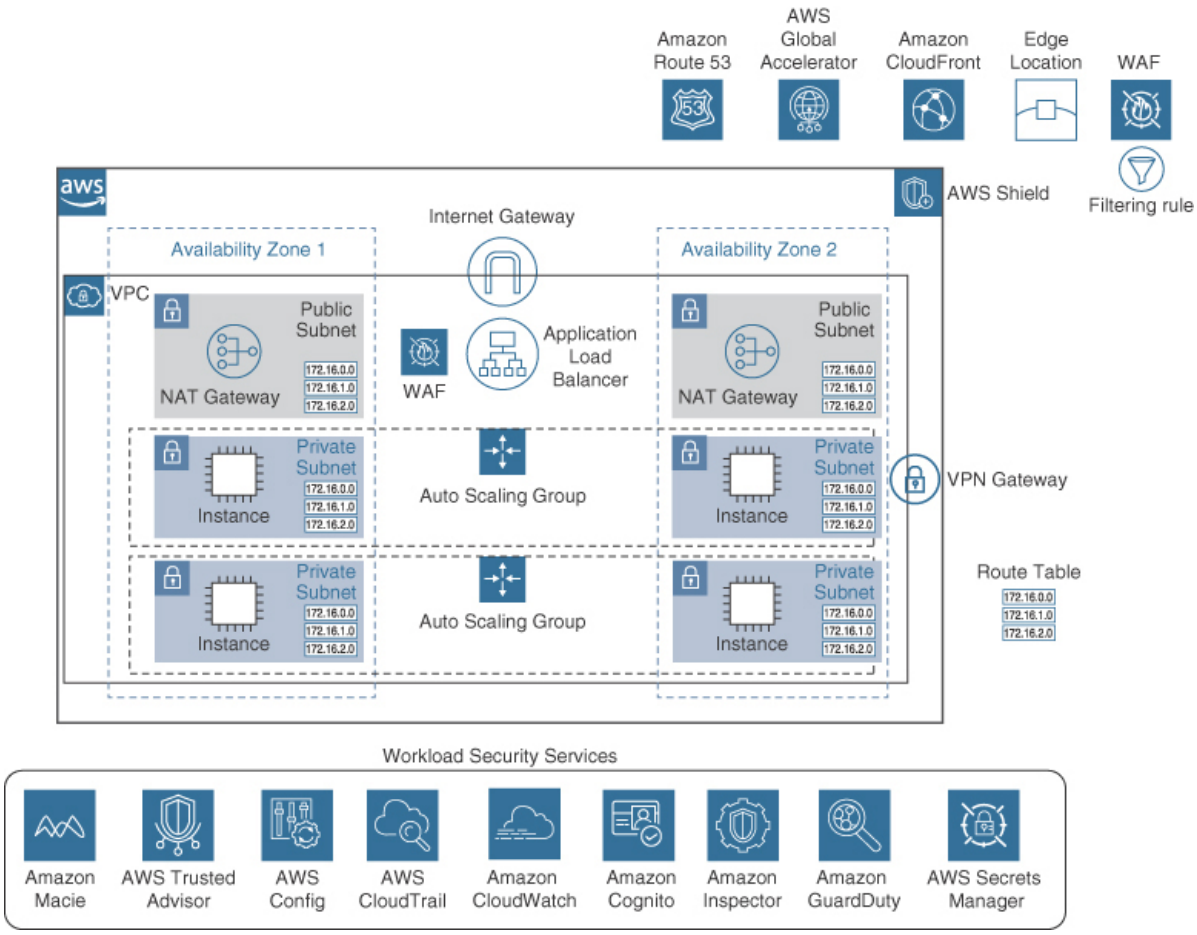


Figure 4-1 Connections and Security Services

Networking Services Located at Edge Locations

Edge locations are located at the edge of the AWS regions and are used to serve content to users more quickly and efficiently. These essential AWS services are located at each edge location:

- **Amazon Route 53:** Amazon-provided DNS services for resolving queries to Amazon CloudFront and AWS Global Accelerator deployments. Additional details on Route 53 operation can be found in **Chapter 7**, "**Designing Highly Available and Fault-Tolerant Architecture.**"
- **AWS Shield:** Provides protection against distributed denial of service attacks (DDoS).
- **AWS Web Application Firewall (WAF):** Protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. AWS WAF enables you to create rules that block, allow, or monitor web requests based on conditions.

- **Amazon CloudFront:** CloudFront serves cached static and dynamic content from edge locations rather than from the origin data location (S3 bucket or application server), which can reduce the amount of time it takes to access static, dynamic, or streaming content. Additional details on CloudFront operation can be found in **Chapter 11**, "**High-Performing and Scalable Networking Architecture.**"

AWS Shield (Standard and Advanced)

What if a malicious request, DDoS attack, or a malicious bot attempts to enter an AWS edge location and attack a public-facing application? AWS Shield Standard protection protects AWS infrastructure and ingress paths at each edge location for all AWS customers. AWS Shield runs at each edge location, providing basic DDoS protection for known Layer 3 and Layer 4 attacks using AWS Web Application Firewall rules deployed and managed by AWS.

If organizations don't have the required expertise needed to solve ongoing security exploits that are attacking workloads hosted at AWS, they can contract with AWS experts to assist with real-time custom protection, known as AWS Shield Advanced, a paid version of AWS Shield that utilizes an expert AWS DDoS response team with a 15-minute SLA response protecting your workload components (Amazon EC2 instances, AWS Elastic Load Balancers, Amazon CloudFront distributions, AWS Global Accelerator deployments, and Amazon Route 53 resources). After analyzing the situation, the response team creates and applies custom WAF filters to mitigate DDoS attacks and other security issues. All AWS Shield Advanced customers get access to a global threat dashboard (see **Figure 4-2**) that displays a sampling of current attacks.

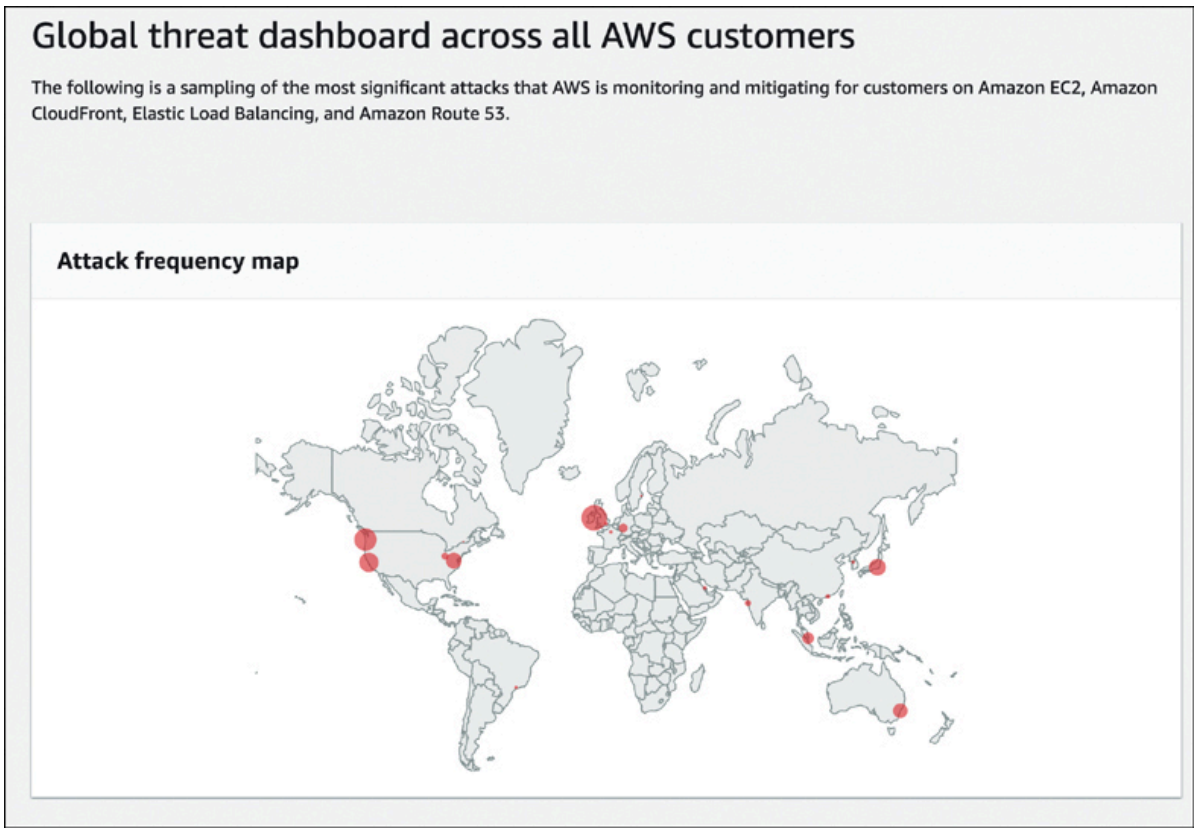


Figure 4-2 Global Threat Dashboard

AWS Shield Advanced also provides *cost protection*, which saves customers money when workload compute resources are required to scale due to illegitimate demand placed on the workload cloud services by a DDoS attack. AWS refunds the additional load balancer, compute, data transfer, and Route 53 query costs that accumulate during the DDoS attack for AWS Shield Advanced customers.

AWS Shield Advanced costs $3,000 a month with a one-year commitment. AWS WAF and AWS Firewall Manager are included with AWS Shield Advanced at no additional charge.

Note

Multiple WAF rules can be managed across multiple AWS accounts and workloads using AWS Firewall Manager.

AWS Web Application Firewall (WAF)

For custom workload protection at each edge location, the AWS Web Application Firewall (WAF) provides custom filtering of incoming (ingress) public traffic requests for IPv4 and IPv6 HTTP and HTTPS requests at each edge location, limiting any malicious request from gaining access to AWS cloud infrastructure. AWS WAF rules are created using the AWS Management Console or the AWS CLI. WAF rules are created using conditions combined into a web ACL. WAF rules allow or block, depending on the conditions, as shown in **Figure 4-3**. WAF rules can be applied to public-facing application load balancers, Amazon CloudFront distributions, Amazon API gateway hosted APIs, and AWS AppSync. To create a WAF rule, specify the conditions that will trigger the rule, such as the source IP address or the content of a web request. Next, specify the action that the rule should take when the conditions are met, such as blocking or allowing the request. Behaviors and conditions can be used to create custom rules that meet the specific security requirements for your web applications. WAF supports the following behaviors:

- **IP addresses:** Create rules that allow or block requests based on the source or destination IP address.
- **HTTP methods:** Create rules that allow or block requests based on the HTTP method used in the request, such as blocking all PUT requests.
- **Cookies:** Create rules that allow or block requests based on the presence or absence of cookies in the request, such as a specific cookie that is required for authentication.
- **Headers:** Create rules that allow or block requests based on the contents of the request header.
- **Query strings:** Create rules that allow or block requests based on the contents of the query string.

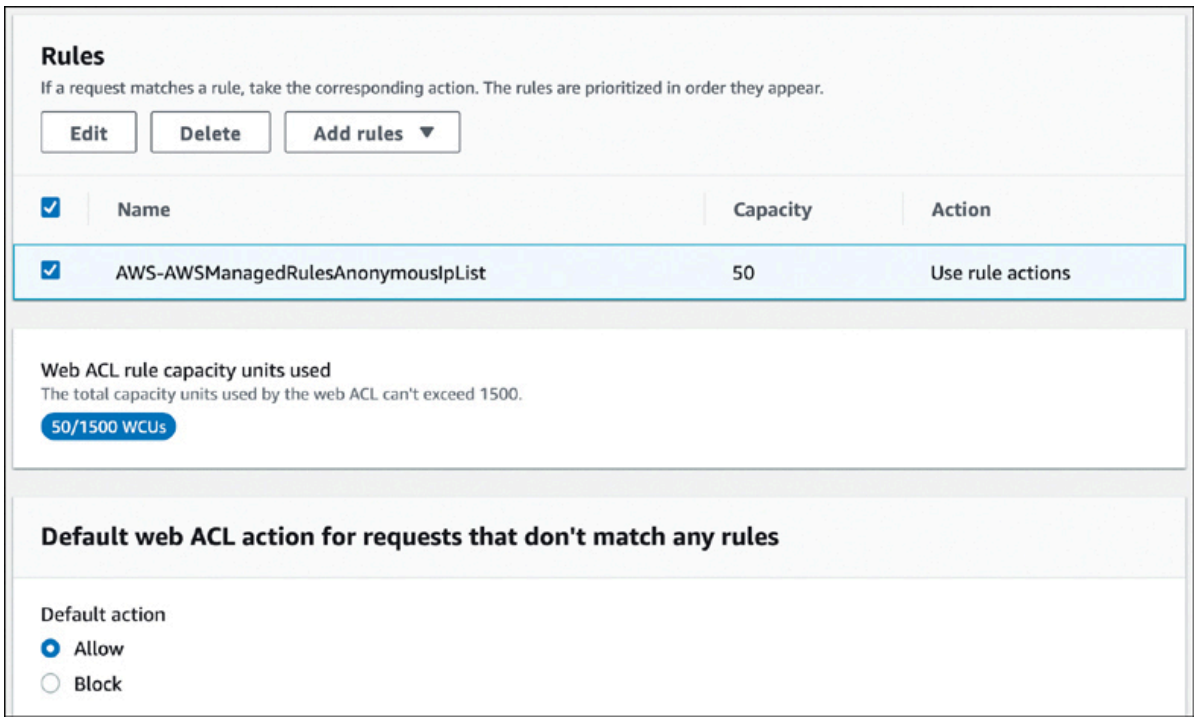


Figure 4-3 Web Application Firewall Rules

AWS WAF provides three types of rules:

- **Regular rules:** Used to specify conditions that must be met in order for the rule to be triggered. For example, you might create a regular rule that blocks requests from a specific IP address or that allows requests that contain a specific string in the query string.
- **Rate-based rules:** Used to limit the rate at which requests are allowed to be made to a web application. For example, you might create a rate-based rule that allows no more than 600 requests per second from a single IP address.
- **Group rules:** Used to group together multiple regular and rate-based rules, applying them as a single entity. For example, you might create a group rule that combines a regular rule that blocks requests from a specific IP address with a rate-based rule that limits the rate at which requests are allowed. This enables you to apply multiple rules to a web application with a single group rule.

VPC Networking Services for Securing Workloads

Each VPC provides a number of security features designed to protect workloads running in the AWS cloud. Features covered in this section include route tables, security groups, and network ACLs. It's im-

portant to realize that these security features are assigned to a specific VPC when they are created.

Route Tables

Each route table is used to control subnet traffic using a set of rules, called routes, that determine where network traffic is directed within each VPC. Workloads running on virtual servers or containers are hosted on EC2 instances located on subnets contained within a specific VPC. Subnets are associated with a specific availability zone within each AWS region.

Each subnet must be associated with a route table. If no specific route table association has been configured, the subnet will use the default route table that was created when the VPC was first created, called the main route table (discussed next), containing a default route that allows instances within the VPC to communicate with each other. Multiple subnets can be associated and controlled with a single route table that is assigned to multiple subnets. You might have multiple private subnets that need routes to the same service, such as a route to the NAT gateway service enabling resources on private subnets to get updates from a public location on the Internet, or a route to the virtual private gateway (VGW) for VPN connections from external locations.

The Main Route Table

Each VPC has a default route table called the main route table that provides local routing services throughout each VPC and across all defined availability zones (AZs), as shown in **Figure 4-4**. The main route table is associated with a VPC after it is first created. The main route table also defines the routing for all subnets that are not explicitly associated with any other custom route table. The main route

table cannot be deleted; however a custom route table can be associated with a subnet, replacing main route table association.

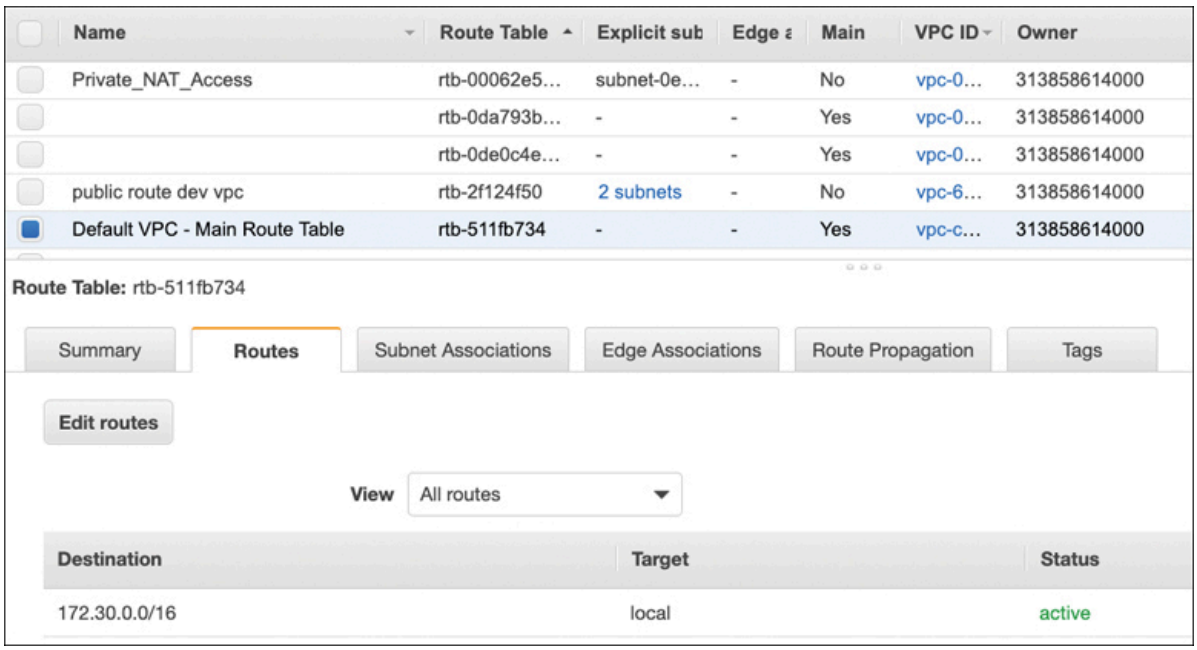


Figure 4-4 Main Route Table

Each custom or default route table has an entry containing the VPC's initial CIDR designations, and a local route used to provide access to the VPC's resources.

As mentioned earlier, you cannot delete the local route entry in a subnet route table, but you can change the local entry to point to another verified target such as a NAT gateway, network interface, or Gateway Load Balancer endpoint.

Custom Route Tables

A custom route table is a user-defined routing table that enables custom routes to direct traffic to specific destinations or implement more complex network architectures.

For example, suppose an organization named Terra Firma is considering starting with a two-tier design for the human resources customer relationship management (CRM) application hosted at AWS.

For the production workload network, Terra Firma has decided to use two AZs within the VPC hosting the CRM servers to provide high availability and failover for the application and database servers. The following tasks must be carried out to create the required infrastructure for the CRM workload:

- Create public subnets in each AZ.
- Add Elastic Load Balancing (ELB) and NAT services to the public subnets.
- Create separate private subnets for the EC2 instances hosting the CRM application servers and the Amazon RDS MySQL database servers (see **Figure 4-5**).

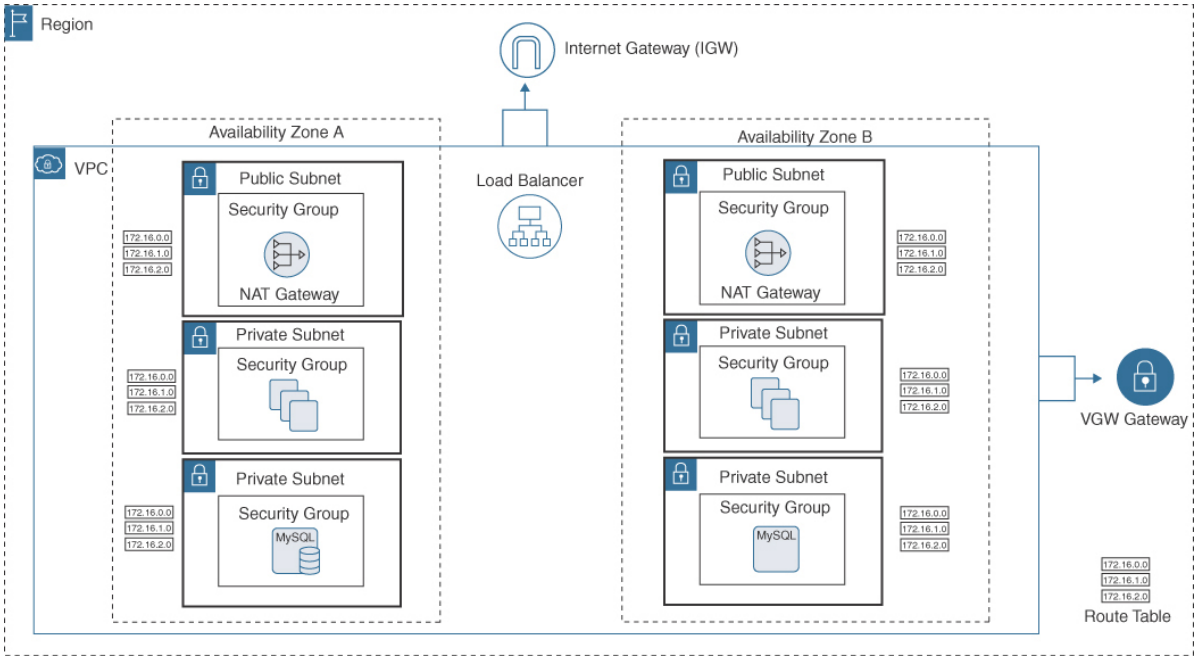


Figure 4-5 Proposed Two-Tier VPC Subnet Design

The RDS database servers use synchronous replication to ensure database records remain up to date. When synchronous replication is enabled for an RDS instance, the primary and standby instances are continuously connected, and all data changes made to the primary instance are immediately replicated to the standby instance. This ensures that the standby instance is always up to date and can be quickly switched over to if the primary instance fails.

For the initial infrastructure design, after the subnets have been created, Terra Firma's network administrators must create custom

route tables for the following subnet groupings (see Figure 4-6):

- **Public subnets and custom route tables:** Public subnets host the AWS ELB load balancer and the AWS NAT gateway service. A custom route table will be created and associated with the public subnets, adding a route table entry for the AWS Internet Gateway service. Internet gateway routes are usually set with a destination route of 0.0.0.0/0 as client queries will typically come from multiple source locations across the public Internet.
- **Private subnets and custom route tables:** The application servers are hosted on private subnets within each AZ. The primary and standby database instances will be deployed and managed using the Amazon Relational Database Service (RDS). Separate AWS NAT gateway services with associated Elastic IP addresses will be ordered and attached to the public subnets in each AZ, enabling the application servers hosted on private subnets to connect to the NAT gateway service and receive any required updates from the Internet. Custom route tables and route table entries pointing to the NAT gateway service must be defined in each private subnet's route table.

Figure 4-6 Using Custom Route Tables

Note

A single route table can be assigned to multiple subnets within the same VPC.

Route Table Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of route tables:

- Each VPC has a main route table that provides local routing throughout each VPC.
- Each subnet, when created using the VPC dashboard, is implicitly associated with the main route table.
- Don't add additional routes to a main route table. Leaving the main route table in its default state ensures that if the main route table remains associated to a subnet by mistake, the worst that can happen is that local routing is enabled. If additional routes are added to the main route table, the additional routes will be available from each new subnet due to the default association with the main route table.
- The main route table cannot be deleted; however, it can be ignored and will remain unassigned if you do not associate it with any subnets within the VPC.
- Create and assign a custom route table for custom routes required by a subnet.
- Subnet destinations are matched with the most definitive route within the route table that matches the traffic request.

Security Groups

A **security group (SG)** is a virtual software firewall that controls the incoming and outgoing network traffic for one or more EC2 instances hosted in a VPC. Security groups enable you to specify the protocols, ports, and source IP ranges that are allowed to reach your instances. Every attached elastic network interface (ENI) is protected by a security group. Each security group is associated with a specific VPC and has a set of inbound and outbound rules that designate the port(s) and protocol(s) allowed into and out of each network interface, as shown in **Figure 4-7**.

Note

It might not be obvious when you start creating VPC networking components but all components created using the VPC console are associated. Each subnet, route table, security group, network interface, and network ACL is assigned to a specific VPC during creation.

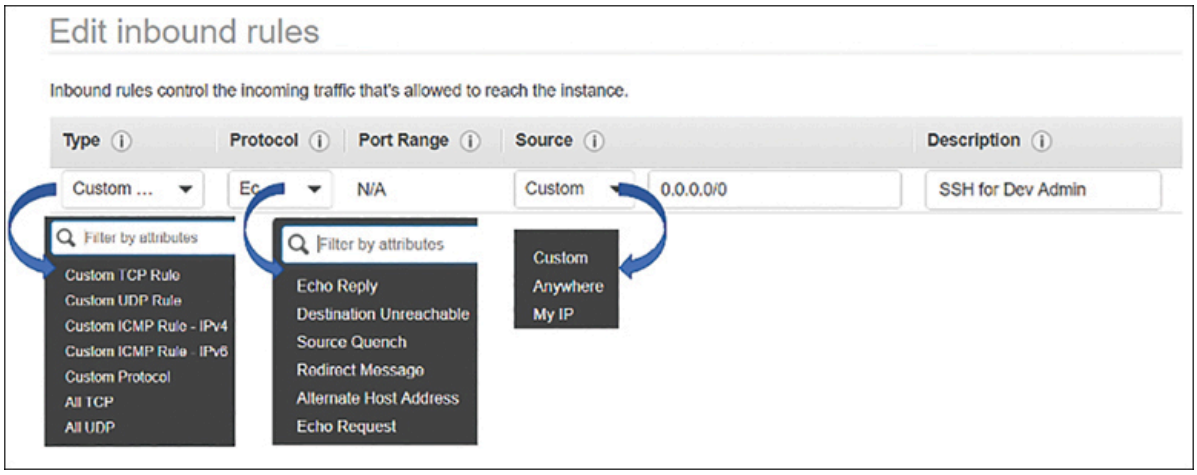


Figure 4-7 Security Group Details

After security groups have been assigned to an EC2 instance, changes made to security groups attached to an EC2 instance while the instance is online usually take effect within seconds.

The initial service quota limit for the number of security groups applied to an EC2 instance is five; you can request an increase using the Service Quotas utility. In addition, every custom security group

can have up to 50 inbound and 50 outbound IPv4 or IPv6 rules. You can also increase the number of rules per security group.

Think of each security group as a reusable security template assigned to a particular VPC. Once a security group has been created, it can be assigned multiple times within the VPC where it was created to protect one or many EC2 instances.

One important concept to grasp about security groups is that they don't *deny* traffic flow. Instead, their job is to *allow* traffic flow. Another equally important concept is the direction of traffic flow both inbound and outbound each security group allows.

Security groups are defined as stateful, which means that if traffic is allowed in one direction, the security group automatically allows the traffic in the opposite direction. For example, if you allow incoming traffic on port 80 (HTTP), the security group will automatically allow outgoing traffic on port 80.

A defined inbound port request does not usually use the same port number for the outbound response. For example, if there's a rule defined allowing inbound HTTP traffic across port 80, the outbound traffic response is allowed out; however, the outbound traffic will not use port 80 for outbound communication. Outbound traffic uses a dynamically assigned port called an *ephemeral port*, determined by the operating system of the server making the response (you will learn more about ephemeral ports later in this chapter, in the upcoming section, "**Network ACLs**").

When a VPC is first created, a default security group is also created. Note that the default security group allows outbound traffic, but any inbound traffic is implicitly denied as inbound rules have not been defined (see **Figure 4-8**).

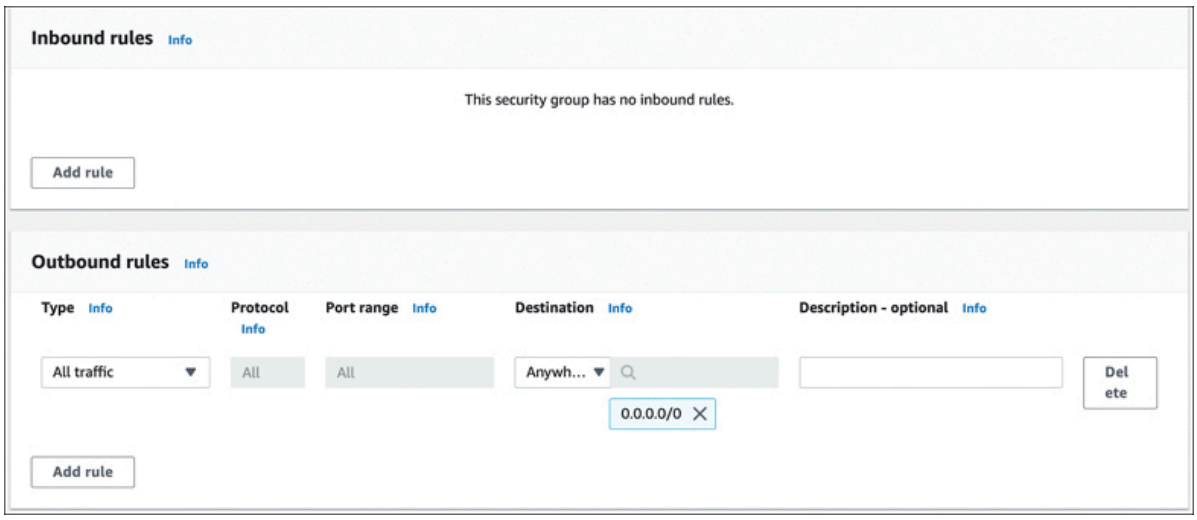


Figure 4-8 Default Security Group Rules

The default security group allows all outbound traffic but denies all inbound traffic. EC2 instances in a VPC associated with just the default security group can initiate outbound communications, but all inbound communication is blocked. The default security groups can't be deleted; however, the default security groups can be removed and custom security groups can be created and used instead.

Here are some additional details about security groups to know:

- A security group is always associated with a single VPC.
- Each elastic network interface assigned to an EC2 instance hosted within a VPC can be associated with up to five security groups by default.
- Security groups allow traffic; however, a security group cannot explicitly deny traffic. If either inbound or outbound access is not specifically allowed, it is implicitly denied.
- When a new security group is created, all outbound traffic is allowed if you don't review and change the default outbound rules before saving.
- Specify which protocols, ports, IP ranges, and security groups are allowed to access your instances.
- Outbound rules can be changed to direct outbound traffic to a specific outbound destination. For example, you could decree that all outbound traffic from a public-facing load balancer can only flow to a security group protecting the web tier.

Security group rules are defined as *stateful*. This means that inbound traffic flowing through a security group is tracked, logging the traffic allowed in, and any allowed inbound traffic is always allowed outbound. This process is called *connection tracking*; connections to a security group are automatically tracked to ensure valid replies. Some AWS services that can be associated with a security group include

- **Amazon Elastic Compute Cloud (EC2):** Security groups can be used to control network traffic to and from EC2 instances.
- **Amazon Elastic Kubernetes Service (EKS):** Security groups can be used to control network traffic to and from EKS clusters.
- **Amazon Elastic Container Service (ECS):** Security groups can be used to control network traffic to and from ECS tasks and services.
- **Amazon Relational Database Service (RDS):** Security groups can be used to control network traffic to and from RDS instances.
- **Amazon Elastic Load Balancer (ELB):** Security groups can be used to control network traffic to and from ELB load balancers.

Security Groups Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of security groups:

- A security group acts like a firewall at the EC2 instance, protecting all attached network interfaces.
- Security groups support both IPv4 and IPv6 traffic.
- A security group controls both outgoing (egress) and incoming (ingress) traffic.
- For each security group, rules control the inbound traffic that is allowed to reach the associated EC2 instances.

- Separate sets of rules control both the inbound and the outbound traffic.
- Each security group includes an outbound rule that allows all outbound traffic by default. Outbound rules can be modified and, if necessary, deleted.
- Security groups allow traffic based on protocols and port numbers.
- Security groups define allow rules. (It is not possible to create rules that explicitly deny access.)
- Security group rules allow you to direct traffic outbound from one security group inbound to another security group within the same VPC.
- Changes made to a security group take effect immediately.
- Security groups don't deny traffic explicitly; instead, they deny traffic implicitly by defining only *allowed* traffic.
- Security groups are stateful; for requests that are allowed in, their response traffic is allowed out, and vice versa.
- For each rule, you define the protocol, the port or port range, and the source inbound and output destination for the traffic.
- The protocols allowed with security groups are TCP, UDP, or ICMP.

Note

It is impossible to block specific IP addresses by using a security group; instead, use a network access control list to block a range of IP addresses. Further details are provided in the section "**Network ACLs**" later in this chapter.

Custom Security Groups

When a custom security group is created, it is associated with a specific VPC. By default, a custom security group allows no inbound traffic but allows all outbound traffic. To create inbound rules and outbound rules, select the security group properties and use the following tabs to make changes (see **Figure 4-9**):

- **Inbound rules:** Define the source of the traffic—that is, where it is coming from—and what the destination port or port range is. The traffic source could be a single IP address (IPv4 or IPv6), a range of addresses, or from another security group.
- **Outbound rules:** Define the destination of the outbound traffic. The destination of the traffic could be a single IP address (IPv4 or IPv6), a range of addresses, or from another security group; EC2 instances that are associated with one security group can access EC2 instances associated with another security group.

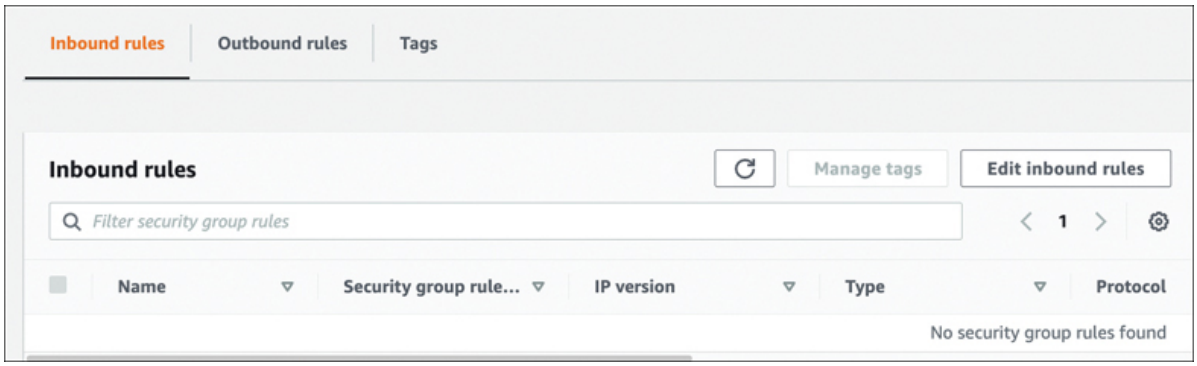


Figure 4-9 Default Security Group Inbound and Outbound Tabs

When designing security groups, the best practice is to minimize the ports allowed. Open only the specific ports that are needed for the services and applications running on your load balancer. The following sections provide some examples to consider for your security group configurations and setup.

Note

Security groups "allow" access; however, security groups can also be said to "deny by default." If an incoming port is not allowed, access is denied.

Web Server Inbound Ports

Web server security group rules need to allow inbound HTTP or HTTPS traffic access from the Internet (see **Table 4-2**). Other rules

and inbound ports can also be allowed, depending on workload requirements.

Table 4-2 Web Server Security Group Options

Web Server Inbound Ports	
Port	**Details**
80 (HTTP)	Inbound IPv4 (0.0.0.0) Inbound IPv6 (::0)
443	HTTPS
25	SMTP
53	DNS
22	SSH

Database Server Inbound Ports

Several database server engines are available when deploying Amazon Relational Database Server database instances. The default port address assigned during deployment is based on the database engine's default port, which can be changed to a custom port number for additional security. **Table 4-3** lists the default RDS security group database port options that are assigned per database engine during installation.

Table 4-3 RDS Database Inbound Ports

Port	Database Engine
3306	Microsoft SQL Server
3306	Amazon Aurora/MySQL
5432	Amazon Aurora PostgreSQL
1521	Oracle
27017	MongoDB

Administration Access

Connecting to an EC2 instance to perform direct administration requires associating a security group with an elastic network interface with inbound rules allowing either Secure Shell (SSH) or Remote Desktop Protocol (RDP) access, depending on the host's operating system (see **Table 4-4**). Deploying an EC2 instance as a bastion host on a public subnet would allow administrators to first authenticate to the bastion host and then "jump" to the associated EC2 instance in the private subnet.

A bastion host is a special purpose EC2 instance or third-party software appliance hosted on a public subnet exposed to the Internet, serving as a secure gateway or "jump box" for remote access to instances hosted on private subnets in the VPC without exposing a web or database server directly to the Internet (see **Figure 4-10**). Common security group settings for a bastion host could include the following:

- Allow incoming traffic on port 22 for SSH access.
- Allow incoming traffic on port 443 for HTTPS access.

- Allow outgoing traffic to allow the bastion host to access other machines on the network.
- Allow outgoing traffic linking specific security groups to allow the bastion host to access specific machines on the network.
- Set the source IP range for incoming traffic to a restricted range, such as the IP addresses of your office or trusted administrators.

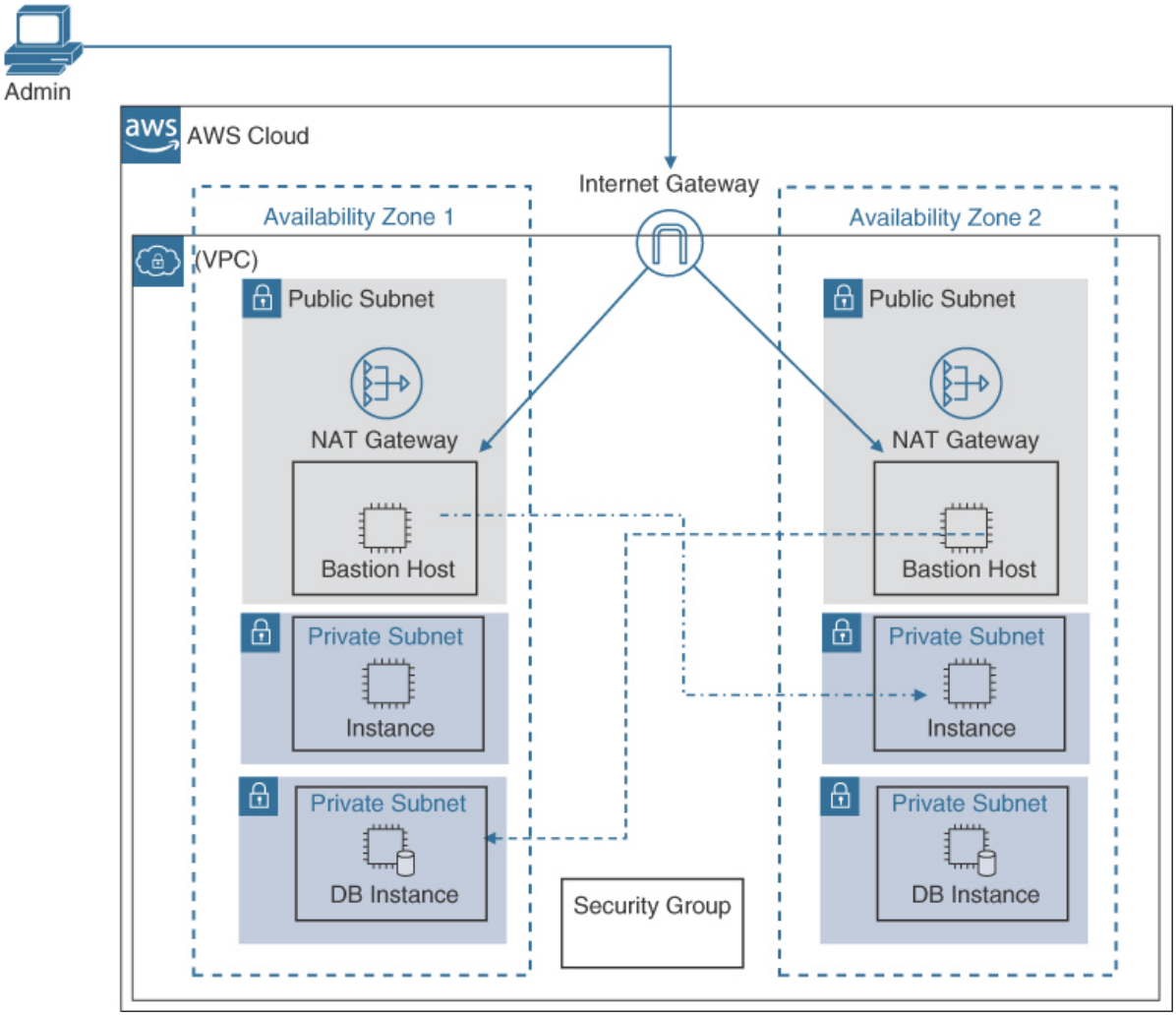


Figure 4-10 Bastion Host Solution

Table 4-4 Security Groups Inbound Ports for Administrative Access

Port	Operating System
22	Linux
3389	Windows

Understanding Ephemeral Ports

When configuring network security settings, such as a security group or network access control list, you need to allow for ephemeral port ranges for outbound communication from your instances or network services, such as load balancers.

Network design needs to consider where the traffic originated for both inbound and outbound traffic requests. Return traffic from an EC2 instance hosted in a VPC to a destination across the Internet communicates using a dynamic outbound or inbound *ephemeral port*.

Ephemeral ports are temporary, short-lived ports that are typically used by client applications for outbound communications from a predefined range of port numbers and are used for the duration of a communication session. When the session is complete, the port is released and can be used by another application.

TCP/IP communications don't utilize the same inbound and outbound ports; instead, the client or server's operating system defines the range of ports that will be dynamically selected for the return communication—that is, the outbound communication. Network connections require two endpoints: a source and a destination. Each source and destination endpoint has an IP address and an associated port number.

When a client system connects to a server, several components are employed: the server IP address, the server port, the client IP address, and the client port. The ephemeral port is a temporary port assigned by the computer's TCP/IP stack. The TCP/IP implementation chooses the port number based on the host operating system. In the case of Windows Server 2016 and above, the ephemeral port range is

from 49152 to 65535. If Linux is the operating system, the ephemeral port range is from 32768 to 61000, as shown in **Table 4-5**. Different operating system versions may use slightly different ephemeral ranges; check what your operating system uses for ephemeral ports.

When communication is carried out from a source service to its destination, the traffic typically uses the named port for the destination traffic, such as port 22 on a Linux box accepting SSH connections. However, for the return traffic from the server to the client, an ephemeral port is typically used for the return traffic. An ephemeral port can be defined as a dynamically assigned port from a range of assumed available port addresses. Outbound packets travel through an outbound port allowed by the existing security group using an allowed ephemeral port.

Outbound communication from an EC2 instance hosted on a VPC must have an allowed outbound range of ephemeral ports. These ports remain available only during the communication session; each dynamically assigned port is released after the TCP connection terminates. If custom security groups or NACLs are deployed, ephemeral rules need to appear in both the inbound and outbound rules to cover the dynamic requirements of communication using ephemeral ports. **Table 4-5** lists some common inbound port numbers that are typically used. The outbound port 443 is the exception as it answers outbound, using port 443.

Table 4-5 Inbound Port Numbers

Port #	Service	Protocol	Description	Port Type
20	FTP	TCP/UDP	File transfer data	Dynamic

Port #	Service	Protocol	Description	Port Type
21	FTP	TCP/UDP	File transfer control	Dynamic
22	SSH	TCP/UDP/SCTP	Secure Shell	Dynamic
25	SMTP	TCP/UDP	Simple mail transfer	Dynamic
67	BOOTPS	UDP	Bootstrap (BOOTP/DHCP) server	Dynamic
68	BOOTPC	UDP	Bootstrap (BOOTP/DHCP) client	Dynamic
69	TFTP	UDP	Trivial file transfer	Dynamic
80	HTTP	TCP	Hypertext Transfer Protocol	Dynamic
88	Kerberos	TCP	Kerberos	Dynamic
123	NTP	UDP	Network time	Dynamic
443	HTTPS	TCP	HTTP over TLS/SSL	443

Port #	Service	Protocol	Description	Port Type
143	Microsoft-ds	IMAP	Internet Message Access Protocol	Dynamic

Security Group Planning

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of security group design:

- Create a security group for your public-facing application load balancer that accepts inbound traffic from the Internet (port 80 or port 443) and sends outbound traffic to your web tier security group, as shown in **Figure 4-11**.
- Create separate security groups for administrative tasks.
- Create a security group for your application tier that only accepts inbound traffic from the web tier and sends outbound traffic to your database tier security group.
- Create a security group for your database tier that only accepts inbound traffic from the application tier.
- Deploy a test application and test communication on a test VPC before deploying to production.

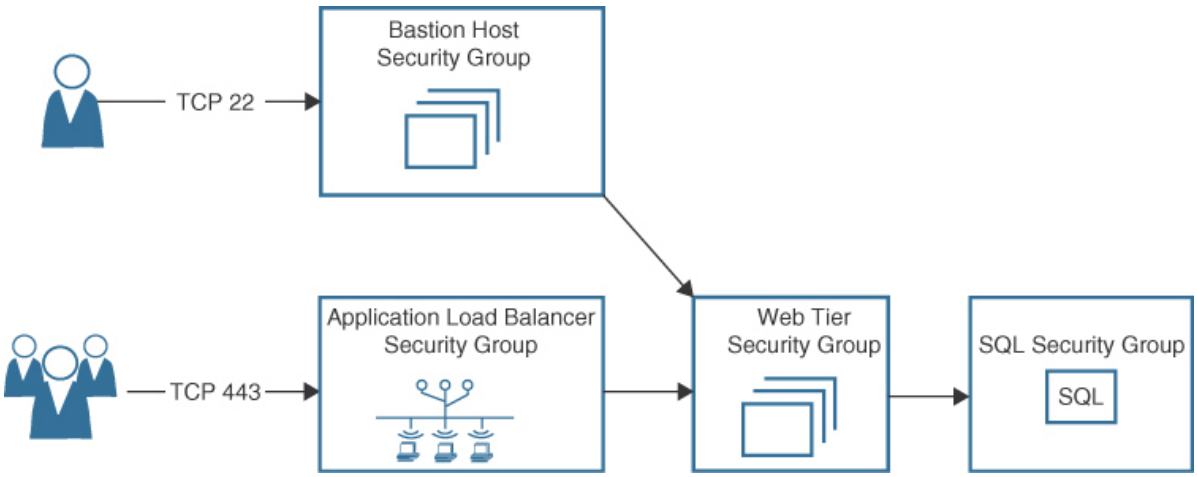


Figure 4-11 Security Group Design

Network ACLs

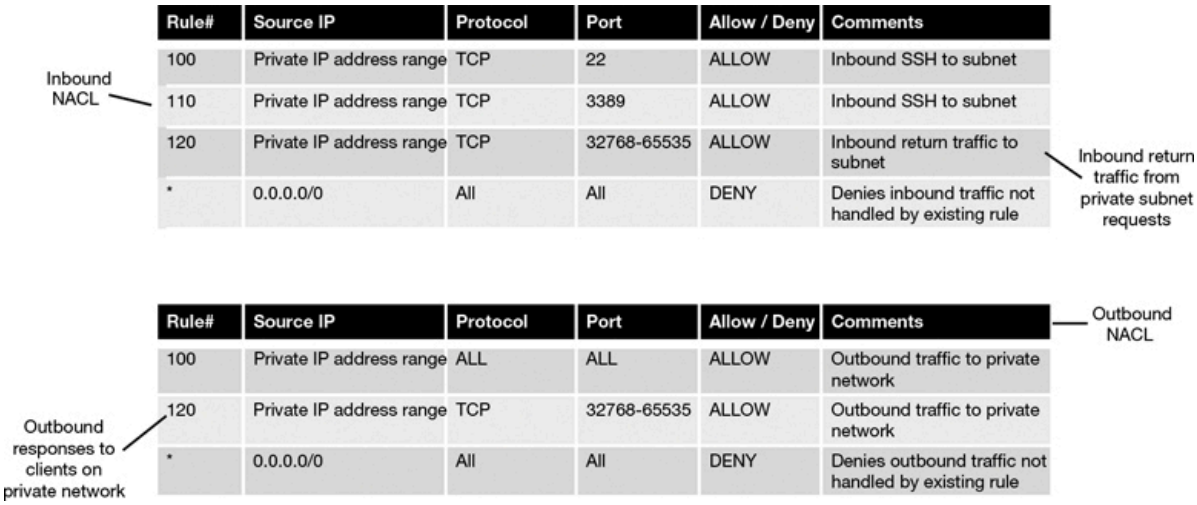
A **network access control list (NACL)** is an optional software firewall that controls inbound and outbound traffic for each subnet within a VPC. A NACL is a set of rules that allows or denies traffic based on the source and destination IP addresses, ports, and protocols. Both the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are supported by network ACLs.

NACLs are used to supplement the security provided by security groups, which are associated with individual Amazon EC2 instances. Whereas security groups control traffic to and from specific instances, network ACLs control traffic at the subnet level, allowing you to set rules for the subnet.

Each VPC is associated with a default NACL that is merely a placeholder as the default network ACL allows all inbound and outbound traffic at the subnet level. Custom NACLs can and should be created, just like security groups. NACLs, once created, can also be associated with one or multiple subnets within the associated VPC.

Each NACL contains a set of inbound and outbound subnet traffic rules, from a starting lowest-numbered rule to the highest-numbered

rule, as shown in **Figure 4-12**. Rules are processed in order to determine whether traffic is allowed or denied inbound or outbound on each subnet.



Rule#	Source IP	Protocol	Port	Allow / Deny	Comments
100	Private IP address range	TCP	22	ALLOW	Inbound SSH to subnet
110	Private IP address range	TCP	3389	ALLOW	Inbound SSH to subnet
120	Private IP address range	TCP	32768-65535	ALLOW	Inbound return traffic to subnet
*	0.0.0.0/0	All	All	DENY	Denies inbound traffic not handled by existing rule

Rule#	Source IP	Protocol	Port	Allow / Deny	Comments
100	Private IP address range	ALL	ALL	ALLOW	Outbound traffic to private network
120	Private IP address range	TCP	32768-65535	ALLOW	Outbound traffic to private network
*	0.0.0.0/0	All	All	DENY	Denies outbound traffic not handled by existing rule

Figure 4-12 NACL Design

NACLs are located at the perimeter of each subnet and provide an additional layer of defense. A single NACL can protect multiple application servers at the subnet level. Rules can target an entire subnet or a block of IP addresses.

Note

Security protection provided by a NACL is at the subnet level. Blocked network traffic denied at the subnet level cannot get anywhere near your EC2 instances.

Network ACL Implementation Details

Both inbound and outbound rules should be numbered in an organized fashion with some separation between the numbers so that you can make changes if necessary in the future. It is best practice to number your inbound and outbound rules by 10s—10 for the first rule, 20 for the second rule, and so on.

NACL rules are *stateless*; this means that inbound and outbound NACL rules are independent from each other.

- Outbound rules are processed separately without any regard to the defined inbound rules.
- Inbound rules are processed without any regard to the outbound rules that have been defined.

Network ACL Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of NACLs:

- A NACL is an optional security control for subnets.
- Each VPC is assigned a default NACL that allows all inbound and outbound traffic across all subnets.
- NACLs are stateless in design; inbound and outbound rules are enforced independently.
- Each NACL is a collection of deny or allow rules for both inbound and outbound traffic.
- The default NACL can be modified.
- A NACL has both allow and deny rules.
- A NACL applies to both ingress and egress subnet traffic; it does not apply to traffic within the subnet.
- You can create custom NACLs and associate them to any subnet in a VPC.
- A custom NACL can be associated with more than one subnet.
- A subnet can be associated with only one NACL.
- A NACL is a first line of defense at the subnet level; a security group is a second line of defense at the instance.

Network ACL Rule Processing

Both inbound and outbound rules are evaluated, starting with the lowest-numbered defined rule. Once a rule matches the traffic re-

quest, it is applied; there is no additional comparison with higher-numbered rules that may also match. A misconfigured lower-numbered rule that also matches the same traffic request could cause problems. If you designated a higher-numbered rule to deal with specific traffic, but instead a lower-numbered rule matched the traffic request, the higher-numbered rule would never be used, as shown in **Table 4-6**.

Table 4-6 NACL Rules with Incorrect Order

Rule Number	Source	Protocol	Port Number	Allow/Deny	Comment
100	0.0.0.0/0	TCP	22	Allow	Inbound SSH is allowed
110	0.0.0.0/0	TCP	3389	Allow	Inbound RDP is allowed
120	0.0.0.0/0	TCP	3389	Deny	Inbound RDP deny rule will not be evaluated
*	0.0.0.0/0	All	All	Deny	Denies all traffic not defined by any other rule

* All undefined traffic is blocked.

When inbound packets appear at the subnet level, they are evaluated against the incoming (ingress) rules of the network ACL. For example, the request is for port 443. Starting with the first rule, numbered 100, there is not a match because the first rule has been defined for port 80 HTML traffic (see **Table 4-7**). The second rule, numbered 110, has been defined for allowing HTTPS traffic. Therefore, HTTP traffic is allowed onto the subnet. All other traffic is denied access if it doesn't match any of the inbound allow rules. If the inbound communication is from the Internet, the source is defined as 0.0.0.0/0 because the traffic could come from any location.

Outbound or egress traffic also must be matched with an outbound rule for the traffic to be allowed to exit the subnet. The outbound rule for HTTPS traffic also uses port 443; the destination is 0.0.0.0/0 because the destination could be anywhere across the Internet. In this case, both the inbound and the outbound rules for HTTPS traffic is set to allow. A rule for the required range of dynamic ports allows outbound responses.

Table 4-7 Custom NACL Setup

Inbound Network ACL					
Rule	Source Address	Protocol	Port Number	Allow/Deny	Details
100	0.0.0.0/0	TCP	80	Allow	Allows inbound HTTP traffic from any

Inbound Network ACL					
					IPv4 address on the Internet
110	0.0.0.0/0	TCP	443	Allow	Allows inbound HTTPS traffic from any IPv4 address on the Internet
120	IPv4 address range for administration	TCP	22	Allow	Allows inbound SSH traffic for administrators
130	IPv4 address range for administration	TCP	3389	Allow	Allows inbound RDP traffic

Inbound Network ACL					
					for administrators
*	0.0.0.0/0	All	All	Deny	Denies all traffic not defined by any other rule

Outbound Network ACL

Rule	Destination IP Address	Protocol	Port	Allow/Deny	Details
100	0.0.0.0/0	TCP	80	Allow	Allows outbound HTTP traffic from the public subnet to the Internet
110	0.0.0.0/0	TCP	443	Allow	Allows outbound

Inbound Network ACL					
					HTTPS traffic from the sub-net to the Internet
120	0.0.0.0/0	TCP	32768–65535	Allow	Allows out-bound re-sponses to clients across the Internet
*	0.0.0.0/0	All	All	Deny	Denies all traf-fic not defined by any other rule

* All undefined traffic is blocked.

VPC Flow Logs

VPC flow logs enable you to capture information about the IP traffic going to and from a VPC. Flow logs can be used to monitor, troubleshoot, and analyze the network traffic in your VPC.

VPC flow logs can be enabled at the VPC, subnet, or elastic network interface level, capturing traffic flowing in and out of the specified resource. Flow logs record the IP traffic flowing in and out of your VPC, including information about the source and destination IP addresses, ports, protocols, and packet and byte counts.

Network traffic can be captured for analysis or to diagnose communication problems at the level of the elastic network interface, subnet, or entire VPC. When each flow log is created, define the type of traffic that will be captured—accepted traffic, rejected traffic, or all traffic. AWS does not charge for creating a flow log but will impose charges for log data storage.

Flow logs can be stored either as CloudWatch logs or directly in an S3 bucket, as shown in **Figure 4-13**. If VPC flow logs are stored as CloudWatch logs, AWS IAM roles must be created that define the permissions allowing the CloudWatch monitoring service to publish the flow log data to the CloudWatch log group. Once a log group has been created, you can publish multiple flow logs to the same log group.

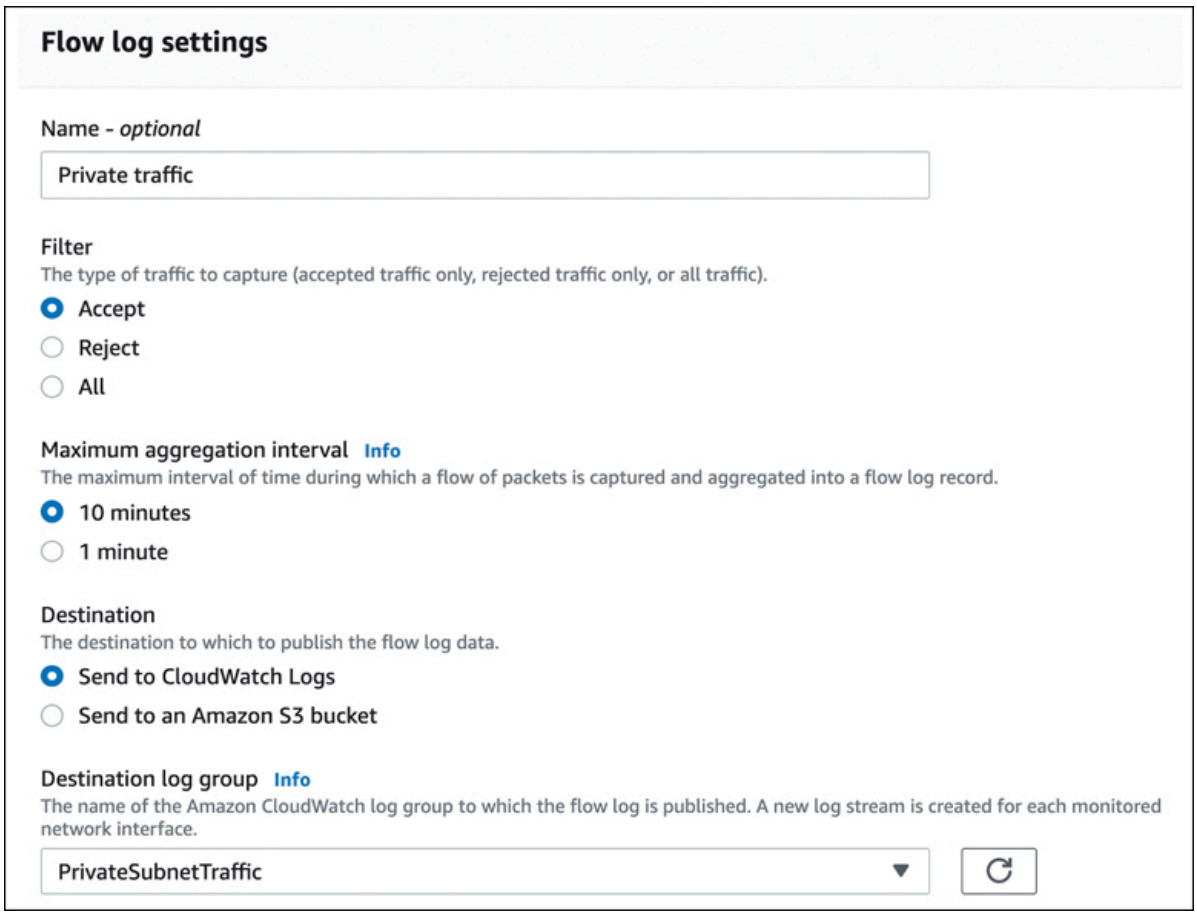


Figure 4-13 Flow Log Storage Location Choices

If you create a flow log for a subnet, or VPC, each network interface in the subnet or VPC is monitored. Launching additional EC2 instances into a subnet with an attached flow log results in new log streams for each new network interface and network traffic flows.

Not all traffic is logged in a flow log. Examples of traffic that is not logged in flow logs include AWS Route 53 server traffic, Windows license activation traffic, EC2 instance metadata requests, Amazon Time Sync Service traffic, reserved IP address traffic, and DHCP traffic.

Any EC2 instance elastic network interface can be tracked with flow logs. Here are several examples of where VPC flow could be useful:

- **Amazon Elastic Compute Cloud (EC2):** VPC flow logs can be enabled for EC2 instances to capture traffic flowing to and from the instances.
- **Amazon Elastic Load Balancer (ELB):** VPC flow logs can be enabled for ELB load balancers to capture traffic flowing to and from

the load balancer.

- **Amazon Elastic Kubernetes Service (EKS):** VPC flow logs can be enabled for EKS clusters to capture traffic flowing to and from the cluster.
- **Amazon Elastic Container Service (ECS):** VPC flow logs can be enabled for ECS tasks and services to capture traffic flowing to and from the tasks and services.
- **Amazon Route 53:** VPC flow logs can be enabled for Route 53 to capture traffic flowing to and from Route 53.

NAT Services

At AWS, the purpose of network address translation (NAT) services is to provide an indirect path for EC2 instances hosted on private subnets that need Internet access to obtain updates, licensing, or other external resources. NAT is a networking technique that enables private network resources to access the Internet while hiding their true IP addresses. Several AWS services provide NAT capabilities:

- **Amazon Virtual Private Cloud (VPC) NAT Gateway:** Enables instances in a private subnet to access the Internet without exposing their private IP addresses.
- **AWS Transit Gateway NAT:** Enables instances in a VPC or on-premises network to access the Internet without exposing their private IP addresses.
- **AWS PrivateLink NAT Gateway:** Enables instances in a VPC to access resources in another VPC or on-premises network without exposing their private IP addresses.

NAT Gateway Service

Amazon VPC NAT Gateway is a service that provides NAT capabilities for Amazon VPC, allowing instances in private subnets to indirectly

access the Internet. The NAT gateway translates the private IP addresses of the EC2 instance requesting access to its own public IP address, allowing EC2 instances hosted on private subnets to access the Internet without exposing their private IP addresses.

The **NAT gateway service** is hosted in a public subnet configure with an Elastic IP address (a static public IP address), as shown in **Figure 4-14** for Internet communication. For multi-availability redundancy, Amazon recommends placing a NAT gateway in each availability zone. Route table entries need to be added to each private subnet's route table, allowing EC2 instances with a path to access the NAT gateway service.

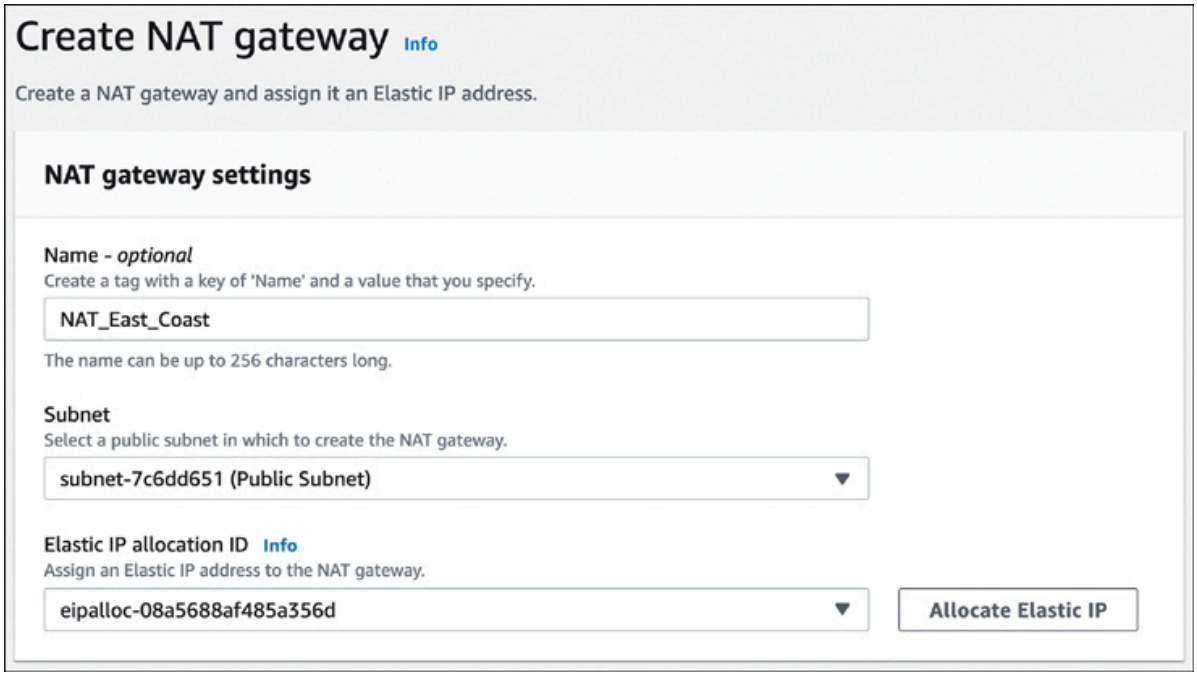


Figure 4-14 Creating a NAT Gateway

Note

The AWS NAT Gateway service initially supports up to 5 Gbps of bandwidth throughput and can scale up to 50 Gbps, as required.

NAT Instance

A NAT gateway third-party software appliance could also be deployed in a public subnet to allow EC2 instances in a private subnet to connect to the Internet and receive updates as necessary. However, you must configure and manage each NAT instance that is deployed. If you decide to use a third-party solution to provide NAT services, Amazon recommends that you create a high-availability pair of NAT instances for redundancy. **Table 4-8** compares the NAT gateway service and the NAT instance.

Table 4-8 NAT Gateway and NAT EC2 Comparison

Parameter	NAT Gateway Service	NAT Instance
Management	By AWS	By the customer
Bandwidth	Up to 50 Gbps	Depends on the EC2 instance size
Maintenance	By AWS	By the customer
Public IP address	Elastic IP address	Elastic IP address
Security groups	Not supported	Required
Port forwarding	Not supported	Supported
Bastion host	Not supported	Supported

Note

When deploying a NAT instance, source/destination checks must be disabled on the EC2 instance. By default, source/destination checks are enabled for all EC2 instances, which means that the instance can send and receive traffic

only if the source and destination IP addresses match the private IP address of the instance.

AWS NAT Gateway Service Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of the NAT Gateway service:

- An AWS NAT gateway must be hosted in a public subnet.
- An AWS NAT gateway uses an Elastic IP address as its static public IP address.
- The AWS NAT gateway service does not support security groups.
- The AWS NAT gateway service does not support port forwarding.

Amazon Cognito

Amazon Cognito provides authentication, authorization, and user management for web and mobile applications. Amazon Cognito enables users to sign into applications hosted at AWS using popular identity providers, such as Amazon, Facebook, and Google, without having to create new credentials. End users sign in using either a user pool or federated identity provider (see **Figure 4-15**).

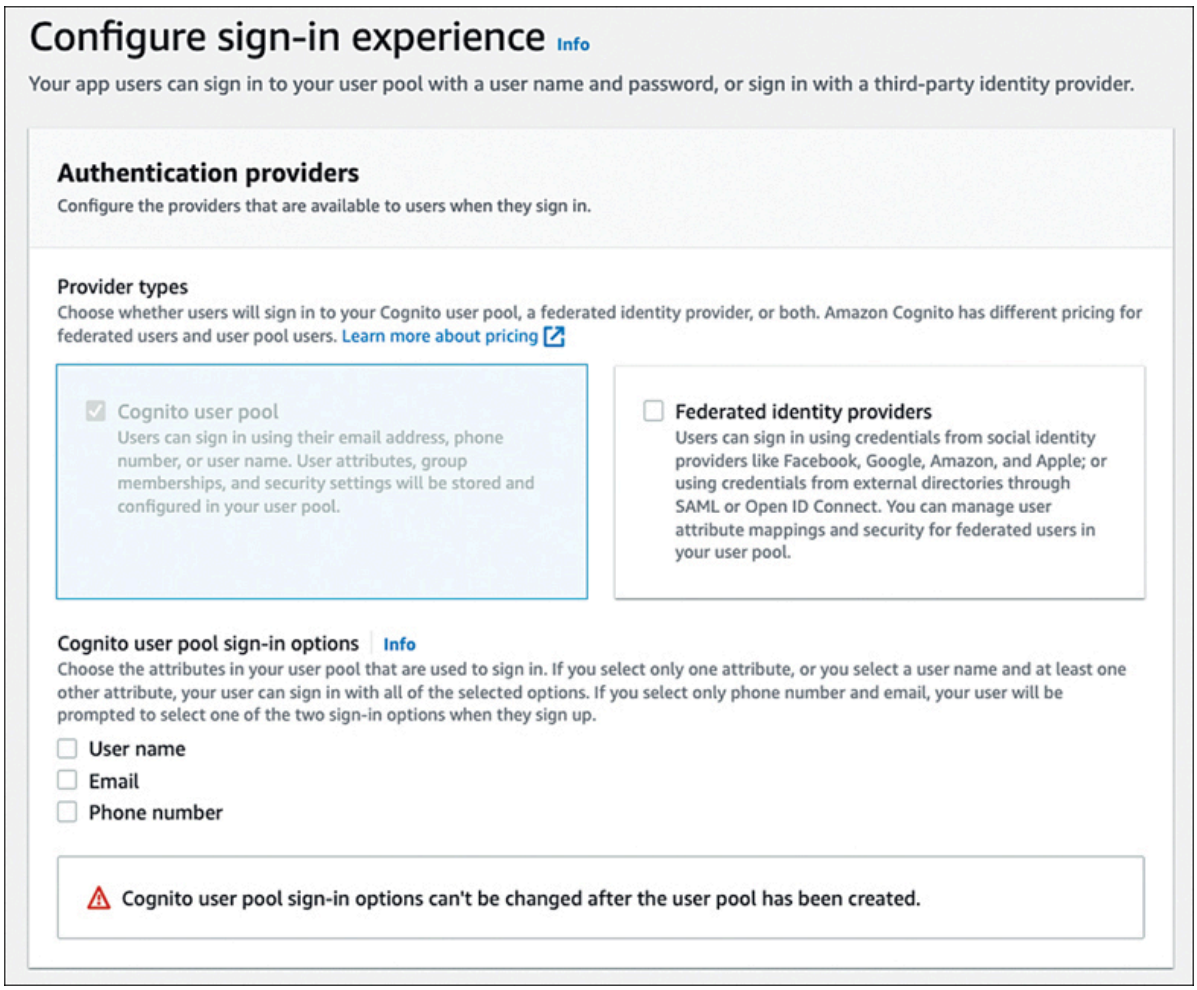


Figure 4-15 AWS Cognito Authentication Options

User Pool

Amazon Cognito user pools are a fully managed user directory that enables you to create and manage user accounts for your application. User pools provide sign-up and sign-in options for your users, as well as user profile management and security features such as multi-factor authentication and password policies.

A member of a user pool can sign into a web application with a username, phone number, or email address. Multi-factor authentication (see **Figure 4-16**) is supported during the sign-in process using an authenticator app such as Authy or Google Authenticator or an SMS message for the time-based one-time password (TOTP).

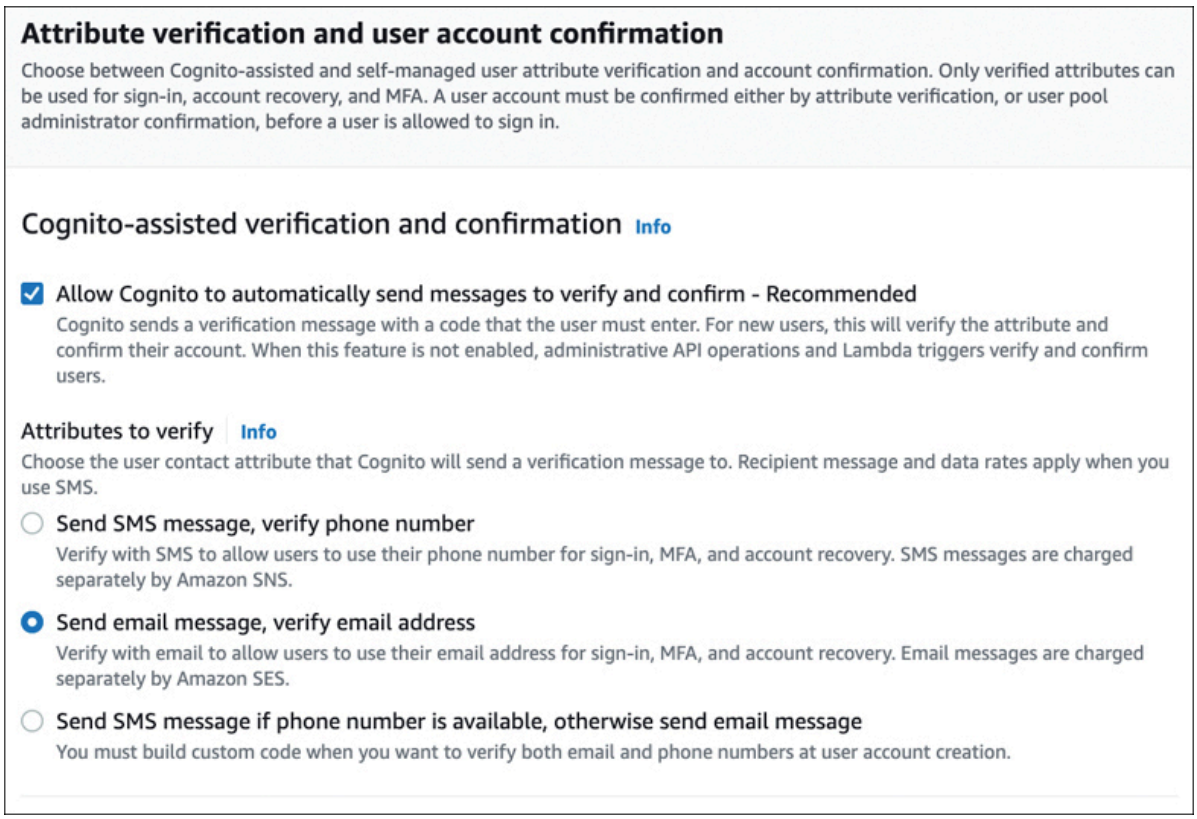


Figure 4-16 Multi-Factor Authentication Options

After an end user has been successfully authenticated using Amazon Cognito, a JSON Web Token (JWT) is issued to secure API communications or to be exchanged for temporary credentials allowing access to on-premises resources or AWS resources such as the S3 storage services used by the web or mobile application (see **Figure 4-17**).

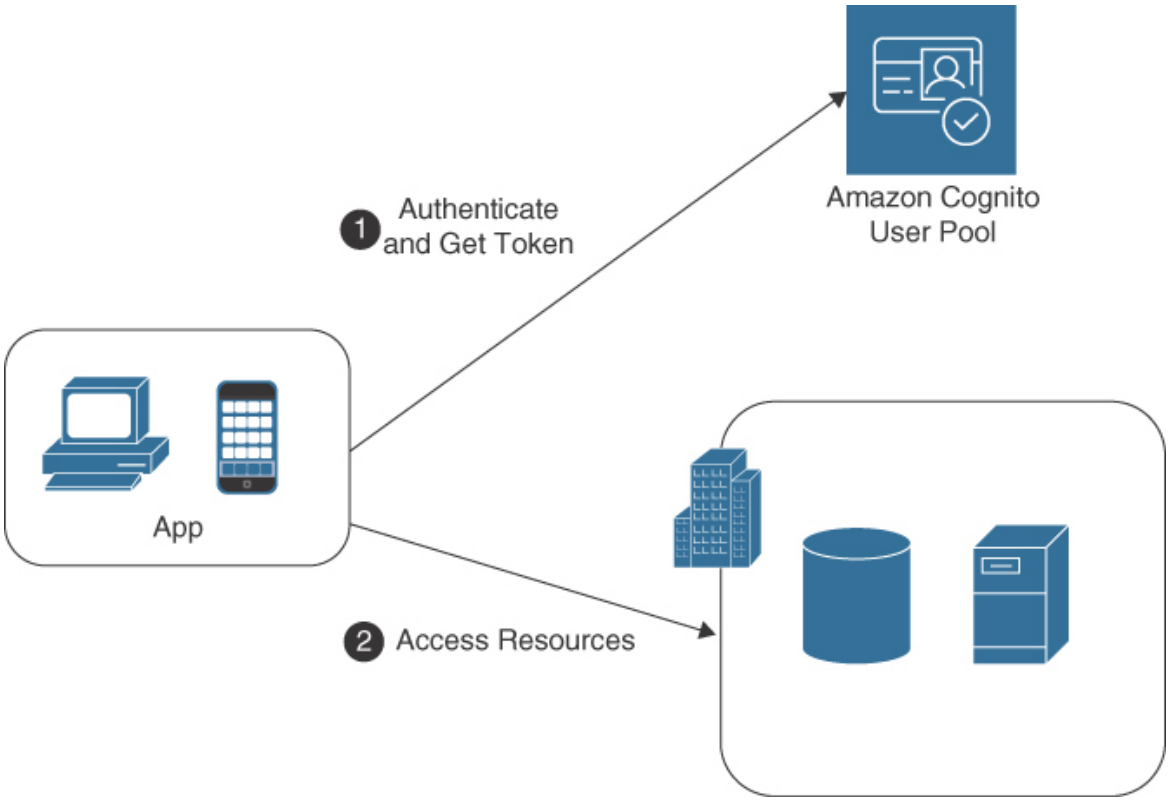


Figure 4-17 User Pool Sign-in

Federated Identity Provider

Users can authenticate to a web or mobile app using a social identity provider such as Google, Facebook, or Apple, or using a Security Association Markup Language (SAML) provider such as Active Directory Federation Services or OpenID Connect (OIDC). After a successful user pool authentication, the user pool tokens are forwarded to the AWS Cognito identity pool, which provides temporary access to AWS services (see **Figure 4-18**). Amazon Cognito identity pools enable you to grant your users access to AWS services, such as Amazon S3 and Amazon DynamoDB. Identity pools enable your users to sign in to your application and use AWS resources without having to create AWS credentials.

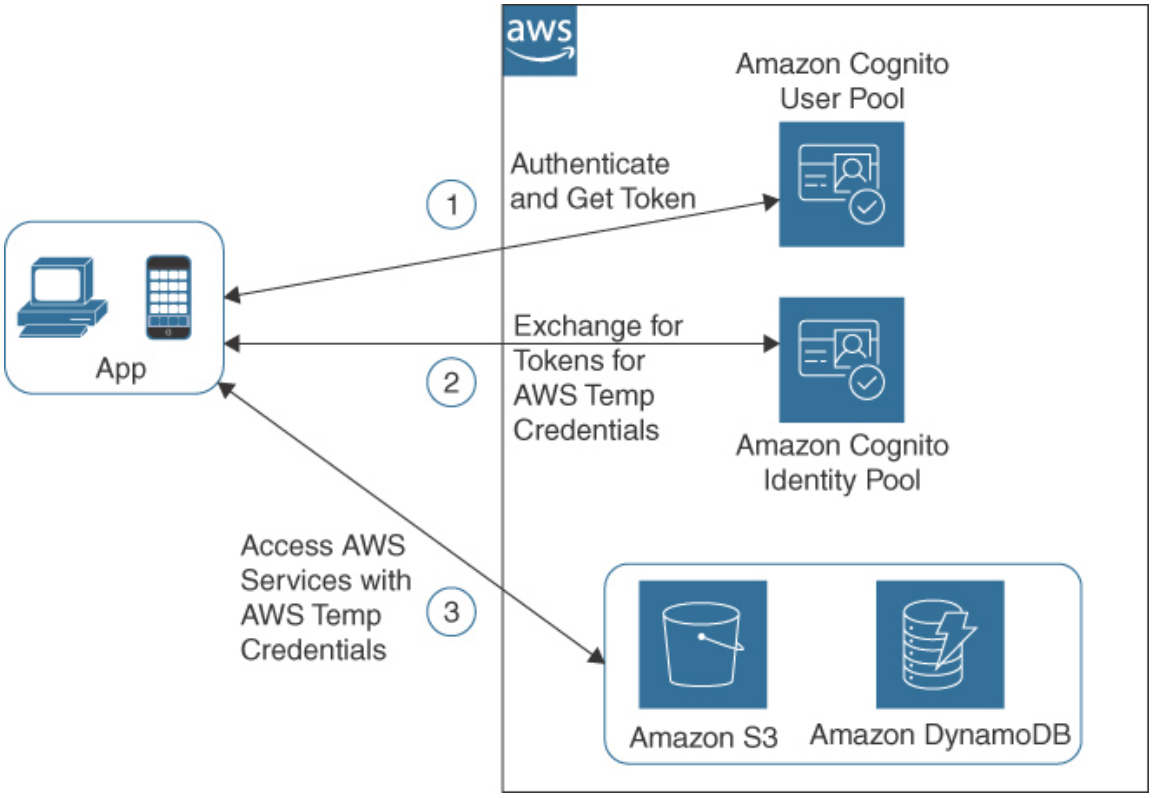


Figure 4-18 User Pool and Federated Identity Pool

Note

The AWS Amplify framework can be used to create an end-user application that integrates with Amazon Cognito.

External Connections

Many companies design solutions using a private hybrid design, where the corporate data center is securely connected to AWS using an AWS VPN connection. Using an IPsec VPN connection to connect to your VPC provides a high level of security.

Before a VPN connection can be set up and connected from your corporate data center to a VPC from your remote network, you need a *virtual private gateway (VPG)* directly attached to the VPC where access is required.

Note

Both an Internet gateway (IGW) and a VPG are directly attached to the VPC and not subnets. Gateway devices require route table entries on each subnet where access is required.

Routing types supported are either static or dynamic routes using Border Gateway Protocol (BGP). A VPN connection with a single static route is shown in **Figure 4-19**.

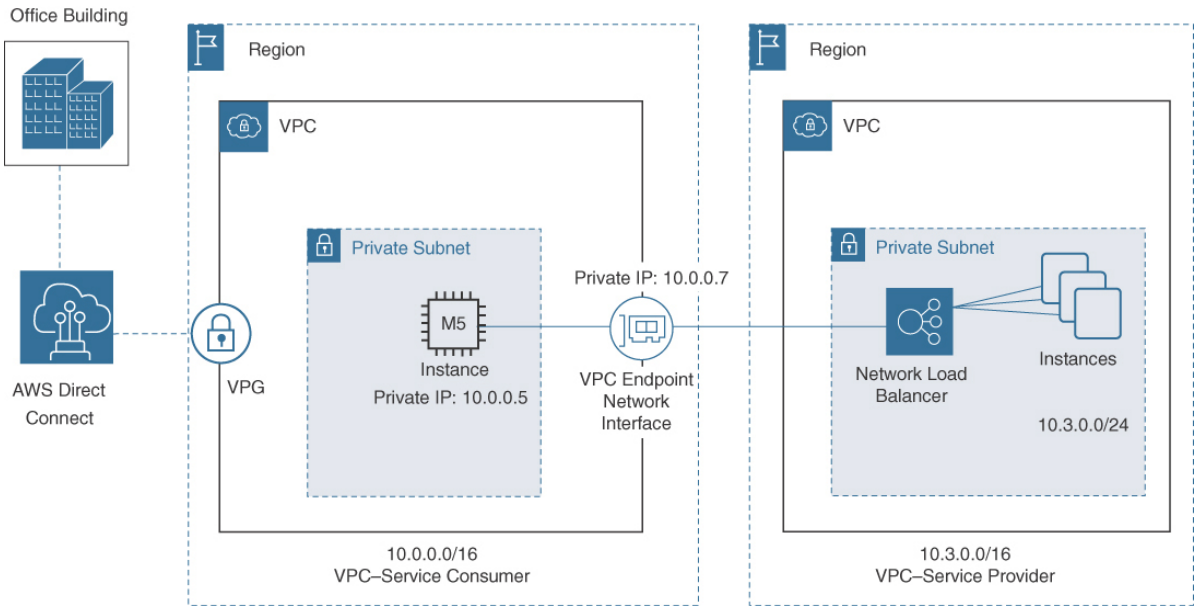


Figure 4-19 External Private Connection Choices

Each VPN connection at AWS is created with two endpoints; each endpoint is connected to a separate availability zone and assigned a unique public IP address.

Virtual Private Gateway

A virtual private gateway is the VPN concentrator on the AWS VPC. The virtual private gateway uses Internet Protocol Security (IPsec) to encrypt the data transmitted between the on-premises network and the VPC. When creating a site-to-site VPN connection, create a virtual private gateway on the AWS side of the connection and a customer gateway on the customer side of the connection.

Several AWS components are required to be set up and configured for an AWS VPN connection. **Figure 4-20** shows the common components: the VPG, the customer gateway (CGW), and the VPN connection.

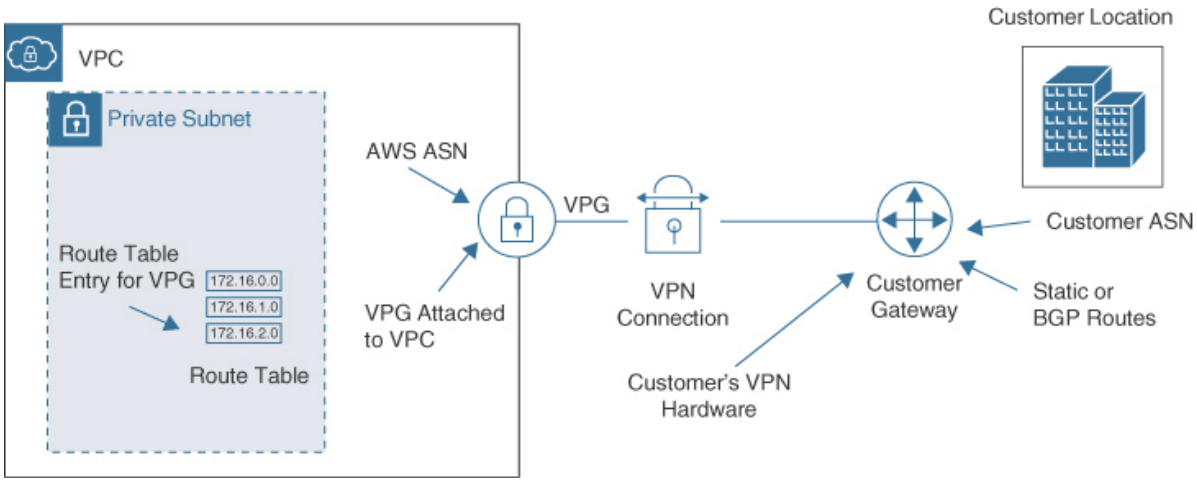


Figure 4-20 VPG Connection Components Choices

Customer Gateway

The customer gateway is the VPN concentrator on the customer side of a site-to-site VPN connection.

The customer gateway provides the VPN endpoint for your on-premises network and uses IPsec to encrypt the data transmitted be-

tween the on-premises network and the VPC. The customer gateway device provided must be compatible with AWS VPN connections. Customers use hardware or virtual devices for their customer gateway devices. AWS provides configuration steps for most of the popular customer hardware options. Examples of devices that AWS supports include Cisco, Check Point, Fortinet, Juniper, and Palo Alto.

During installation, you will be prompted to download the configuration file that matches your customer gateway device. Information contained in this document includes device details and tunnel configuration.

When creating a customer gateway, enter the public IP address or the private certificate of your customer gateway device and indicate the type of routing to be used: static or dynamic. If you choose dynamic routing, enter your private autonomous system number (ASN) for border gateway patrol (BGP) communications. When connections are completed on both the customer and AWS sides, traffic requests from the customer side of the AWS VPN connection initiate the VPN tunnel, as shown in **Figure 4-21**.

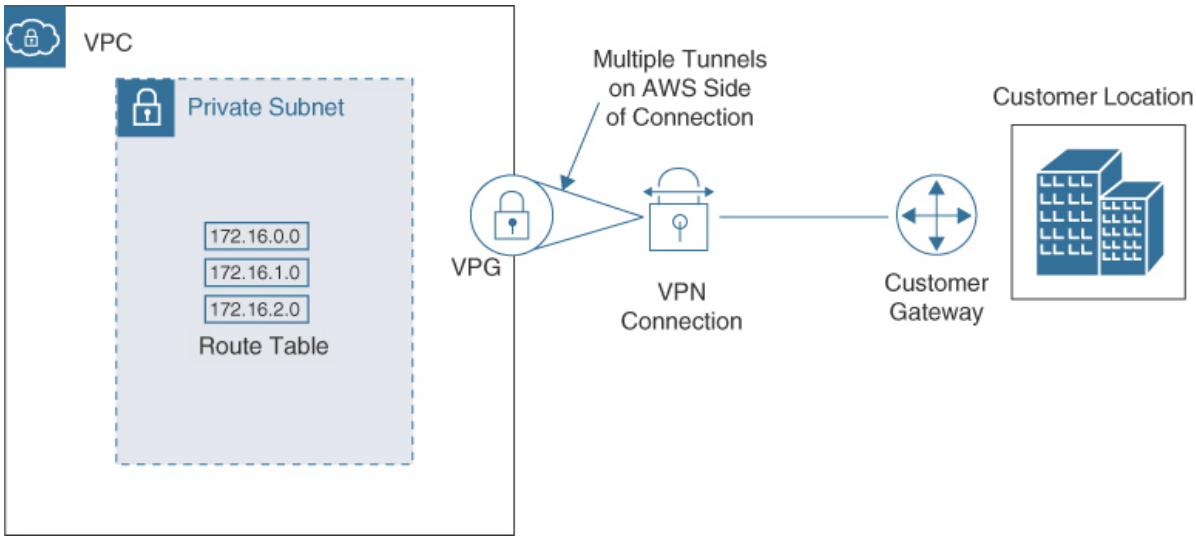


Figure 4-21 AWS VPN Tunnel Connections Choices

Note

During the configuration of an AWS VPN connection, you can accept the ASN provided by AWS or specify your cus-

tom ASN number.

Some of the most common routing options for AWS VPN connections include

- **Static routing:** Static routing enables you to specify routes for traffic over a VPN connection. With static routing, specify the IP address ranges and destinations for your traffic, and the VPN connection will use this information to route traffic.
- **Dynamic routing:** With dynamic routing, the AWS VPN connection will automatically add and remove routes as needed, based on the traffic paths available.

AWS Managed VPN Connection Options

Common solutions for AWS VPN connections include

- **AWS Site-to-Site VPN:** Enables you to create a secure connection between your on-premises network and your VPC. This type of VPN connection uses IPsec to encrypt the data transmitted between the on-premises network and the VPC.
- **AWS Client VPN:** Enables you to create a secure, encrypted connection between your VPC and your remote users. This type of VPN connection uses the OpenVPN protocol and is typically accessed using a client application installed on the user's device.
- **AWS Transit Gateway VPN:** Enables you to create a secure connection between your VPC and an on-premises network, as well as connections between multiple VPCs and on-premises networks. This type of VPN connection uses IPsec to encrypt the data transmitted between the networks.
- **AWS VPN CloudHub:** With CloudHub, multiple remote sites can communicate with the VPC and each other. CloudHub design follows the traditional hub-and-spoke model.

Deploying AWS VPN CloudHub, on the AWS side, there is a single VPG; however, there are multiple customer gateways required as

there are multiple connection paths from multiple physical sites (see Figure 4-22). Each customer gateway requires a unique BGP ASN to distinguish its location. The maximum bandwidth of each AWS VPN connection at AWS is 1.25 Gbps.

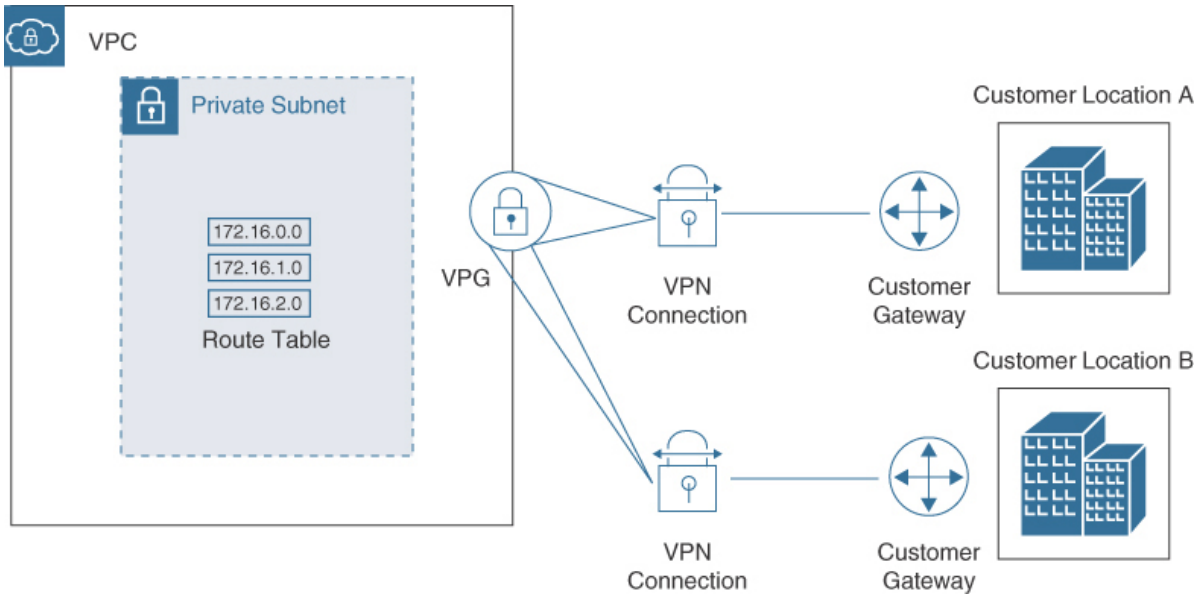


Figure 4-22 VPN CloudHub Design Choices

Understanding Route Propagation

After route table entries have been created to allow VPN connections from the customer gateway, you can enable the automatic provisioning of the available routes through *route propagation*. To enable automatic route propagation, choose the Route Propagation tab from the properties of the route table and then select the VPG to assign to the route table. Route propagation allows a virtual private gateway to automatically propagate routes to the route tables, ensuring efficient communications.

Each AWS VPN connection created in AWS has two tunnels for failover on the Amazon side of the connection. Each tunnel has a unique security association (SA) that identifies each tunnel's inbound and outbound traffic. If static routes are available, when an AWS VPN connection is activated, the static addresses for your customer data center and the CIDR ranges for the connected VPC are automatically added to the route table.

AWS Direct Connect

AWS Direct Connect is a service provided by AWS that enables you to establish a dedicated network connection from your on-premises data center to AWS. It offers two types of connections:

- **Dedicated connection:** This type of connection provides a dedicated, single-tenant network connection between your on-premises network and your VPC. The dedicated connection uses a physical network connection with a capacity of from 1 to 100 Gbps.
- **Hosted connection:** This type of connection enables you to establish a connection to AWS Direct Connect over the public Internet. The hosted connection uses a virtual interface with a capacity of 50 Mbps, 100 Mbps, or 200 Mbps.

Each AWS Direct Connect dedicated connection ordered is a single dedicated connection from your organization's routers to an AWS Direct Connect router. Virtual interface connections can be created to connect directly to AWS services or VPCs. A virtual public interface enables access to Amazon cloud services; a private virtual interface enables access to a VPC.

AWS Direct Connect dedicated connections support 1000BASE-LX or 10GBASE-LR connections over single-mode fiber using Ethernet transport and 1310 nm connectors.

A hosted connection is provided by an AWS Direct Connect partner from the customer data center to the facility where AWS Direct Connect connections can be made. The connection speeds available from the selected Amazon partner can range from 1 to 100 Gbps. To sign up for AWS Direct Connect, open the AWS Direct Connect Dashboard and complete the following steps:

Step 1. Request a connection, specifying the port speed and the Direct Connect location where the connection will be terminated. If

your port speed required is less than 1 Gbps, you must contact a registered AWS Direct Connect vendor that is a member of the Amazon Partner Network (APN) in your geographic location and order a hosted connection at the bandwidth you desire. When this connection is complete, the setup of Direct Connect can continue in the console.

Step 2. When AWS has approved your connection, download a Letter of Authorization-Connecting Facility Assignment (LOA-CFA) and present it to your provider as authorization to create a cross-connect network connection to AWS.

Step 3. Create virtual interfaces for the required connections to either a VPC or a public AWS service.

Step 4. After virtual interfaces have been created, download the router configuration file containing detailed router configuration information to successfully connect to the virtual interfaces.

There are many considerations for AWS Direct Connect, including your location, the AWS region you are operating in, the level of redundancy required, the number of VPCs, public AWS services, or AWS Direct Connect gateways that you connect (see **Figure 4-23**).

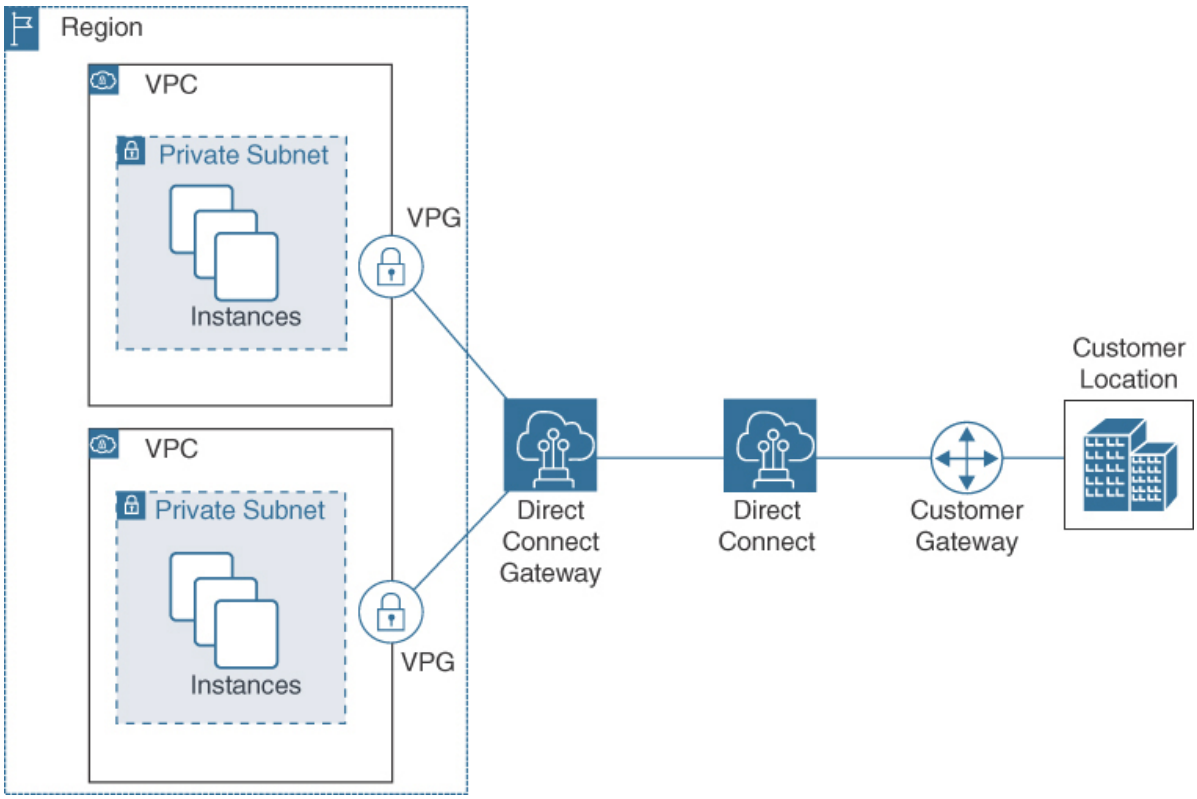


Figure 4-23 Direct Connect Choices

AWS Direct Connect Gateway

A Direct Connect Gateway is a component of AWS Direct Connect that enables you to connect multiple virtual private clouds to a single AWS Direct Connect connection. A Direct Connect gateway acts as a central hub for the VPCs that are connected to it, enabling the routing of traffic between the VPCs.

AWS Direct Connect Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of Direct Connect:

- You can configure an AWS Direct Connect connection with one or more virtual interfaces (VIFs).
- Public VIFs allow access to services such as Amazon S3 buckets and Amazon DynamoDB tables.

- Private VIFs allow access only to VPCs.
- An AWS Direct Connect connection allows connections to all availability zones within the region where the connection has been established.
- You are charged for AWS Direct Connect connections based on data transfer and port hours used.
- AWS Direct Connect dedicated connections are available at 1 Gbps up to 100 Gbps speeds.
- You can order speeds of 50 Mbps up to 200 Mbps through a hosted connection through AWS Direct Connect partners.
- An AWS Direct Connect gateway allows you to connect to multiple VPCs.
- An AWS Direct Connect gateway can connect to virtual private gateways and private virtual interfaces owned by the same AWS account.
- An AWS Direct Connect gateway can be associated with AWS Transit Gateway, extending an organization's private network.
- An AWS Direct Connect connection can also be used with an IPsec VPN connection for additional security.

Amazon GuardDuty

Amazon GuardDuty is a threat detection service that continuously monitors and protects your AWS account, EC2 instances, container applications Amazon Aurora databases, and data stored in S3 buckets (see **Figure 4-24**). It uses machine learning and anomaly detection to identify potentially malicious activity in your AWS environment, such as unauthorized access or unusual behavior. GuardDuty provides alerts for any suspicious activity it detects, allowing organizations to take appropriate action to protect AWS resources. AWS GuardDuty also supports AWS Organizations.

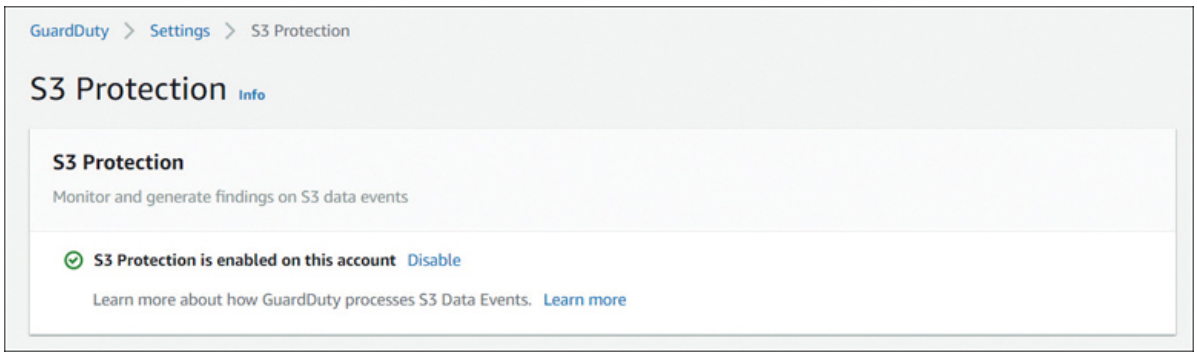


Figure 4-24 GuardDuty Settings

Amazon GuardDuty relies on machine learning anomaly detection, network monitoring and malicious file detection using AWS, and third-party security knowledge to analyze AWS security services, including CloudTrail events, VPC flow logs, Amazon Elastic Kubernetes Service audit logs, and DNS query logs. Amazon GuardDuty performs near-real-time analysis and actions can be automated using AWS Lambda or Amazon EventBridge. Amazon GuardDuty is helpful when deployments are too extensive for organizations to adequately manage and protect AWS resources.

Once enabled, Amazon GuardDuty starts analyzing account, network, data activity, and AWS services enabled for analysis in near real time. Amazon GuardDuty monitors for many security issues, including the following:

- **Reconnaissance:** Amazon GuardDuty scans for unusual API activity, failed database login attempts using CloudTrail management event logs, and suspicious ingress and egress network traffic using VPC flow logs (see **Figure 4-25**).
- **Global events:** Amazon GuardDuty monitors CloudWatch global events for malicious IAM user behaviors, AWS Security Token Service, unauthorized Amazon S3 access, Amazon CloudFront, and Amazon Route 53 for malicious usage across AWS regions.
- **Amazon EC2 instance compromise:** Amazon GuardDuty monitors network protocols, inbound and outbound communication, and compromised EC2 credentials. Amazon GuardDuty Malware

Protection, when enabled, scans EBS volumes attached to EC2 instances and container workloads.

- **Amazon EKS Protection:** Amazon GuardDuty monitors Amazon EKS cluster control plane activity by analyzing Amazon EKS audit logs for issues.
- **Amazon RDS Protection:** Amazon GuardDuty monitors access attempts to existing and new Aurora databases.
- **Amazon S3 Bucket compromise:** Amazon GuardDuty monitors for suspicious data patterns by analyzing AWS CloudTrail management events and Amazon S3 data events, including **Get**, **Put**, **List**, and **Delete** object API operations from a remote host or unauthorized S3 access from known malicious IP addresses.
- **Amazon Route 53 DNS logs:** GuardDuty monitors Amazon Route 53 request and response logs for security issues.

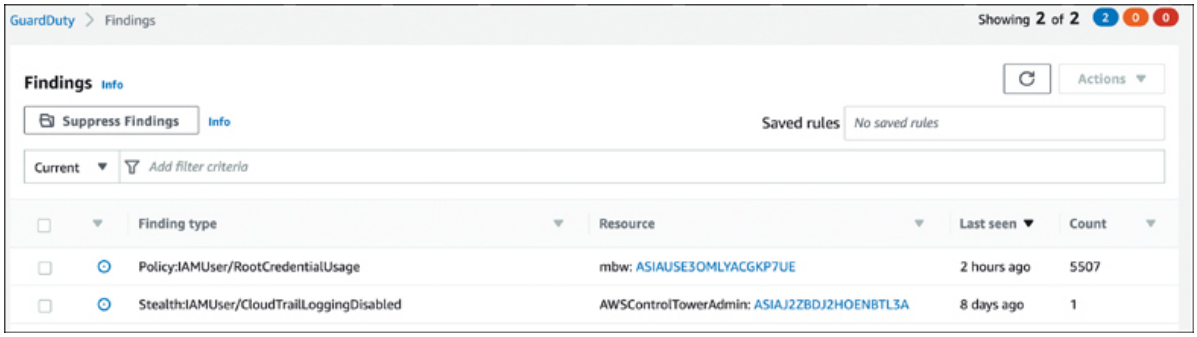


Figure 4-25 GuardDuty Findings

Amazon GuardDuty Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of GuardDuty:

- Amazon GuardDuty can also be deployed with AWS Organizations (AWS recommended deployment).
- When Amazon GuardDuty Malware Protection finds issues with EBS volumes, it creates replica snapshots of the affected EBS volumes.
- Amazon GuardDuty can also be integrated with AWS Security Hub and Amazon Detective Services to perform automated actions.

Amazon Macie

Amazon Macie is a security service provided by AWS that uses machine learning to automatically discover, classify, and protect sensitive data stored in S3 buckets. Amazon Macie helps you secure your data and prevent unauthorized access or accidental data leaks.

Amazon Macie uses machine learning and pattern matching to discover and protect sensitive data, such as personally identifiable information (PII) and intellectual property (IP). Discovered issues are presented as detailed findings for sensitive data, review, and remediation.

Amazon Macie runs data discovery jobs on a schedule or a one-time basis (see **Figure 4-26**), which starts the automated discovery, logging, and reporting of any security and privacy issues that are discovered. Each job selects the S3 bucket(s) and bucket criteria (name, account ID, effective permissions, shared access, and tags). Up to 1000 Amazon S3 buckets and AWS accounts can be selected per discovery job. The following sensitive data types are identified using data identifiers:

- Credential data such as private keys or AWS secret access keys
- Credit card and bank account numbers
- Personal information, health insurance details, passports, and medical IDs
- Custom identifiers consisting of regular expressions (regex) per organization, such as employee IDs, or internal data identifiers

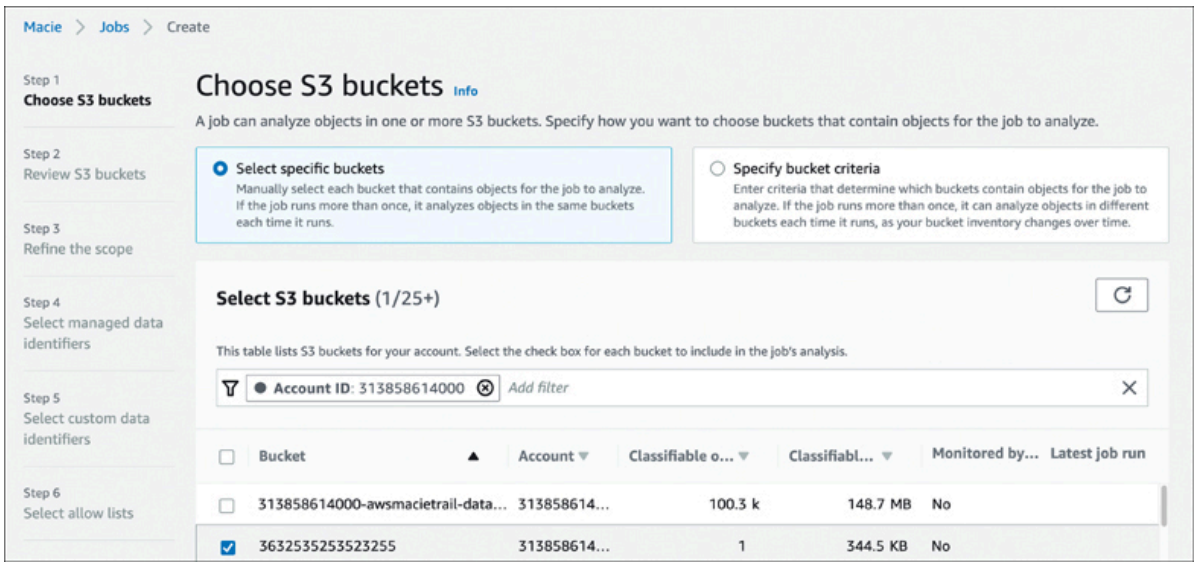


Figure 4-26 Amazon Macie Job Configuration

Amazon Macie data findings are published to the Amazon Macie console. Amazon EventBridge events can be configured that call a custom AWS Lambda function to perform automated remediation tasks.

Amazon Macie Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of Amazon Macie:

- AWS Organizations uses multiple Amazon Macie accounts: an Administrator account that manages the Amazon Macie accounts for the organization and member accounts.
- Sensitive data can be identified using a custom data identifier or keyword.
- Amazon Macie can publish sensitive data policy findings automatically to Amazon EventBridge as events.
- A policy finding provides a detailed report of a potential policy violation (for example, unexpected access to S3 bucket), including a severity rating, detailed information, and when the issue was found.
- Amazon Macie publishes near-real-time logging data to CloudWatch logs.

- Amazon Macie can analyze encrypted objects with the exception of objects encrypted with customer-provided keys (SSE-C).

Security Services for Securing Workloads

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the use cases for the following AWS security tools for monitoring and managing hosted workloads:

- AWS CloudTrail
- AWS Secrets Manager
- Amazon Inspector
- AWS Trusted Advisor
- AWS Config

AWS CloudTrail

AWS CloudTrail records all AWS API calls carried out within each AWS account related to actions across your AWS infrastructure. AWS CloudTrail also logs all account authentications and event history for your AWS account, including actions taken through the AWS Management Console, AWS SDKs, the AWS CLI command prompt, and other AWS service activity. AWS CloudTrail is enabled by default for all AWS accounts, and events in the default CloudTrail trail are available for the last 90 days free of charge. Organizations can also create custom trails storing all activity indefinitely in an Amazon S3 bucket or Amazon CloudWatch log group. Amazon CloudWatch events are logged in AWS CloudTrail within 15 minutes of each API request. The following are common tasks that CloudTrail is useful for:

- Review event history and insights for resource management, compliance, and operational and risk auditing.

- Review event history in AWS CloudTrail for information on successful and unsuccessful authentication requests.
- Review API calls carried out in an AWS account.

Figure 4-27 shows the details of an AWS management console logon listing Amazon S3 Buckets by IAM user Mark, including the AWS account ID, username, time, source, and region.

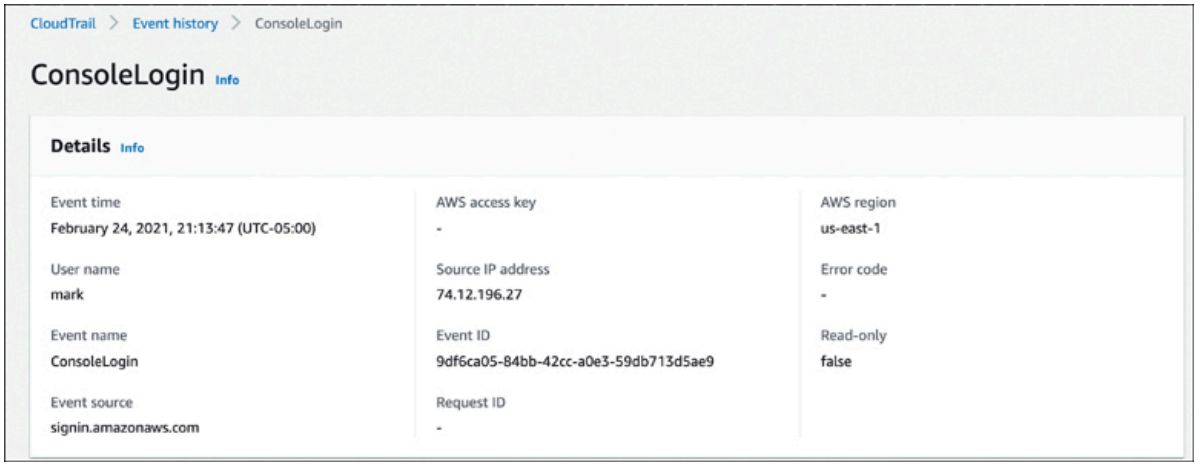


Figure 4-27 Detailed CloudTrail Event

AWS CloudTrail is a regional service with a global reporting reach because the default trail automatically creates separate trails in each active AWS region. AWS CloudTrail events for each AWS region can be viewed using the AWS CloudTrail console and manually switching to the desired AWS region. IAM policies can be created using the AWS Identity and Access Management service to control which IAM users can create, configure, or delete AWS CloudTrail trails and events.

Creating an AWS CloudWatch Trail

To store AWS CloudTrail events longer than the default 90-day time frame, create a custom trail that stores the AWS CloudTrail event information in an Amazon S3 bucket or Amazon CloudWatch log group. Management read/write events, or just read-only or write-only events, can be added to your custom trail, as shown in Figure 4-28. Optionally, you can also create an AWS Simple Notification topic

to receive notifications when specific events have been delivered to the Amazon CloudWatch log group.

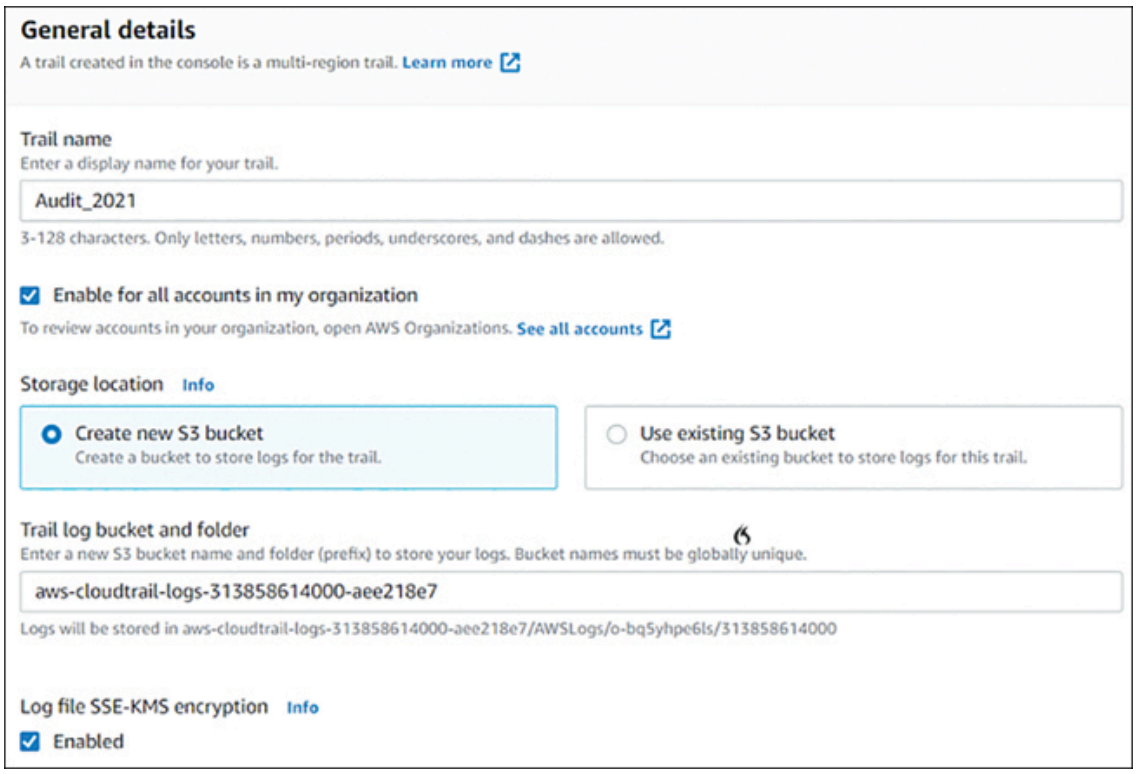


Figure 4-28 Creating a CloudTrail Trail

After data has been logged and stored in a custom trail, analysis and possible remediation can be performed using these methods:

- **Amazon S3 bucket:** API activity for the S3 bucket can trigger a notification to an AWS SNS topic or trigger an AWS Lambda custom function.
- **AWS Lambda function:** Custom AWS Lambda functions can respond to selected AWS CloudTrail data events.
- **AWS CloudTrail Insights:** CloudTrail Insights can be used to detect unusual activity for individual CloudTrail write management events within an AWS account.
- **AWS CloudTrail events:** A CloudTrail event can display a specific pattern, such as authentication as the root user, as shown in Figure 4-29.

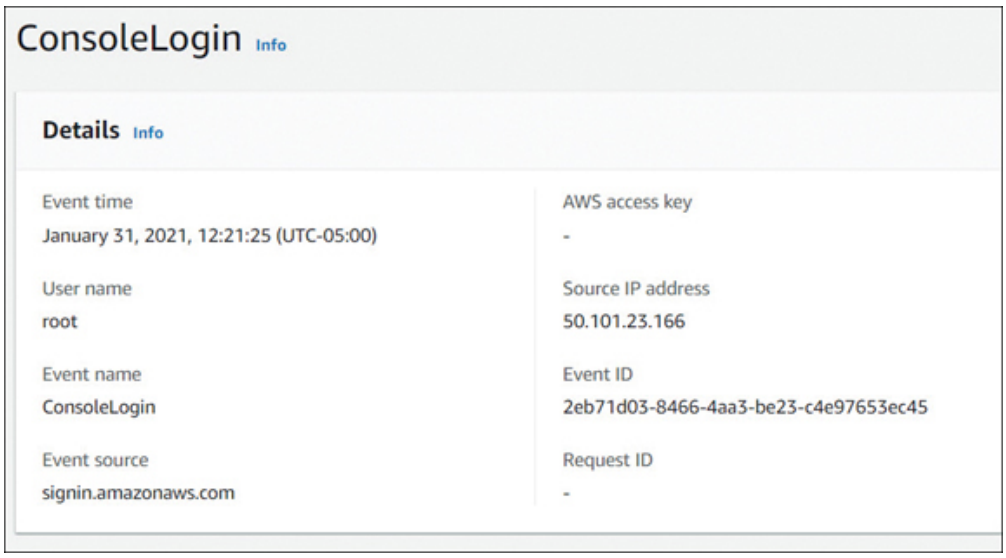


Figure 4-29 CloudTrail Authentication Event

AWS CloudTrail Cheat Sheet

For the AWS Certified Solutions Architect – Associate (SAA-C03) exam, you need to understand the following critical aspects of AWS CloudTrail:

- AWS CloudTrail records all activity on an AWS account, including API calls and authentications.
- Custom AWS CloudWatch trails can deliver events to an S3 bucket or a CloudWatch log group.
- AWS CloudTrail events can be used for auditing AWS account activity.
- AWS CloudTrail reports activity for each AWS account.
- AWS CloudTrail can be integrated with an AWS Organization.
- AWS CloudTrail tracks both data and management events.
- AWS CloudTrail records can be encrypted using S3 server-side encryption.

AWS Secrets Manager

AWS Secrets Manager is a service that enables you to store, rotate, and manage organizational secrets used to access your applications, services, and IT resources. With Secrets Manager, you can securely store and manage secrets, such as database credentials and API keys, helping reduce the risk of secrets being compromised and meet compliance requirements. AWS Secrets Manager enables you to secure and manage secrets for SaaS applications, SSH keys, RDS databases, third-party services, and on-premises resources (see **Figure 4-30**). You can also store credentials for MySQL, PostgreSQL, and Amazon Aurora, and Oracle databases hosted on EC2 instances and OAuth refresh tokens used when accessing third-party services and on-premises resources.

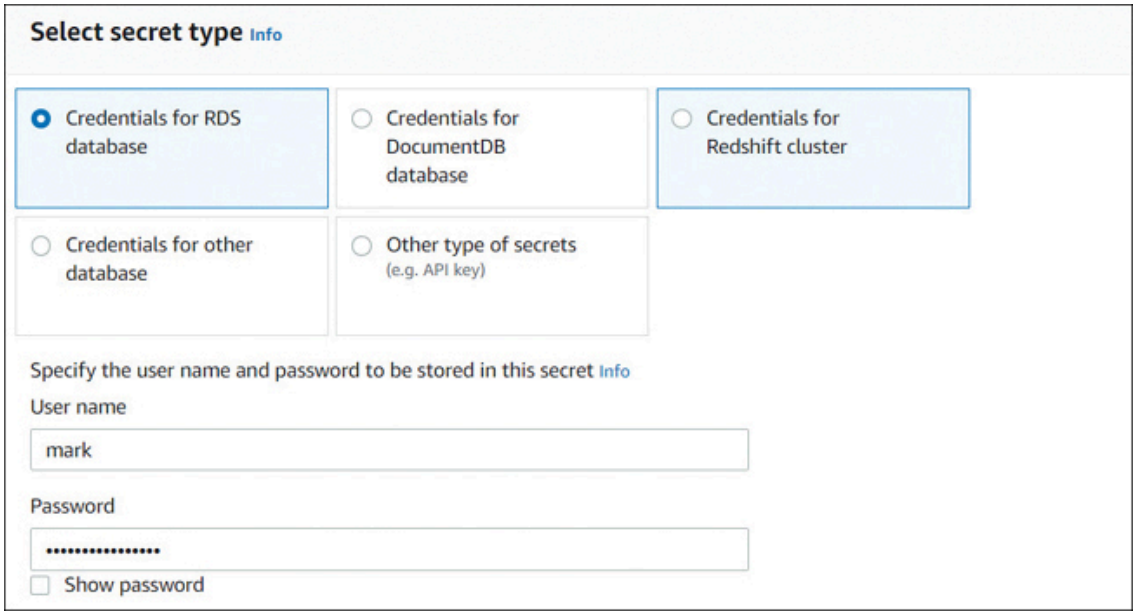


Figure 4-30 Storing RDS Credentials as a Secret

When database secrets are stored in AWS Secrets Manager, the rotation of database credentials can be automatically configured. Secrets are encrypted at rest using encryption keys stored in AWS Key Management Service. You can either specify customer master keys (CMKs) to encrypt secrets or use the default AWS KMS encryption keys provided for your AWS account.

Note

> Use of the term *master* is ONLY in association with the official terminology used in industry specifications and/or standards, and in no way diminishes Pearson's commitment to promoting diversity, equity, and inclusion, and challenging, countering, and/or combating bias and stereotyping in the global population of the learners we serve.

Using the AWS Secrets Manager APIs, developers can replace any hard-coded secrets used in their applications with secrets retrieved from Secrets Manager. Access to secrets is controlled by the IAM policy, which defines the access permissions of users and applications when retrieving secrets.

Applications that are running on EC2 instances hosted within a VPC can use a private interface endpoint to connect directly to AWS Secrets Manager across the AWS private network.

Amazon Inspector

Amazon Inspector allows you to test the security levels of instances you have deployed. After you define an assessment target for Amazon Inspector, which is a group of tagged EC2 instances, Amazon Inspector evaluates the state of each instance by using several rule packages.

Amazon Inspector uses two types of rules: network accessibility tests that don't require the Inspector agent to be installed, and host assessment rules that require the Inspector agent to be installed (see **Figure 4-31**). Amazon Inspector performs security checks and assessments against the operating systems and applications hosted on Linux and Windows EC2 instances by using an optional Inspector agent installed on the operating system associated with the EC2 instance.

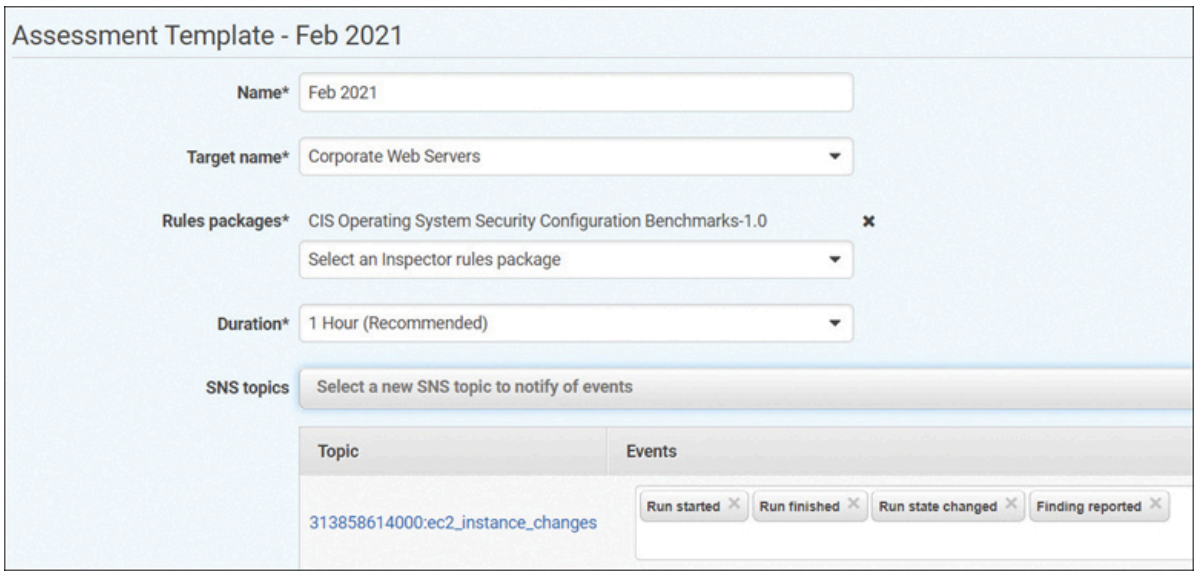


Figure 4-31 Amazon Inspector Options

Assessment templates check for any security issues on targeted EC2 instances. The choices for rule packages comply with industry standards. They include Common Vulnerabilities and Exposure (CVE) checks, Center for Internet Security (CIS) checks, operating system configuration benchmarks, and other security best practices. Current supported levels of CVE checks can be found at **https://nvd.nist.gov/general**. The Amazon Inspector Network Reachability rules package allows you to identify ports and services on your EC2 instances that are reachable from outside the VPC. Amazon Inspector gathers the current network configuration, including security groups, network access control lists, and route tables, and analyzes the accessibility of the instance.

Amazon Inspector rules are assigned severity levels of medium and high based on the defined assessment target's confidentiality, integrity, and availability. Amazon Inspector also integrates with Amazon Simple Notification Service (SNS), which sends notifications when failures occur. An AWS SNS notification can, in turn, call an AWS Lambda function, which can carry out any required task; AWS Lambda can call any AWS API. Amazon Inspector can alert you when security problems are discovered on web and application servers, including insecure network configurations, missing patches, and potential vulnerabilities in the application's runtime behavior.

AWS Trusted Advisor

AWS Trusted Advisor is a built-in management service that executes several essential checks against your AWS account resources (see Figure 4-32). Every AWS account has access to several core AWS Trusted Advisor checks, and access to the AWS Personal Health Dashboard, which alerts you when specific resources you are using at AWS are having issues. The following are core AWS Trusted Advisor checks:

- Security checks include permission checks for EBS and RDS snapshots that are marked public, S3 buckets that have open access, checking for the creation of at least one IAM user, and root accounts that don't have MFA enabled.
- AWS Trusted Advisor checks various AWS services and alerts when usage is greater than 80% of the current service quota limits in force, including IAM users, Amazon S3 buckets created, VPCs created, and Auto Scaling groups.

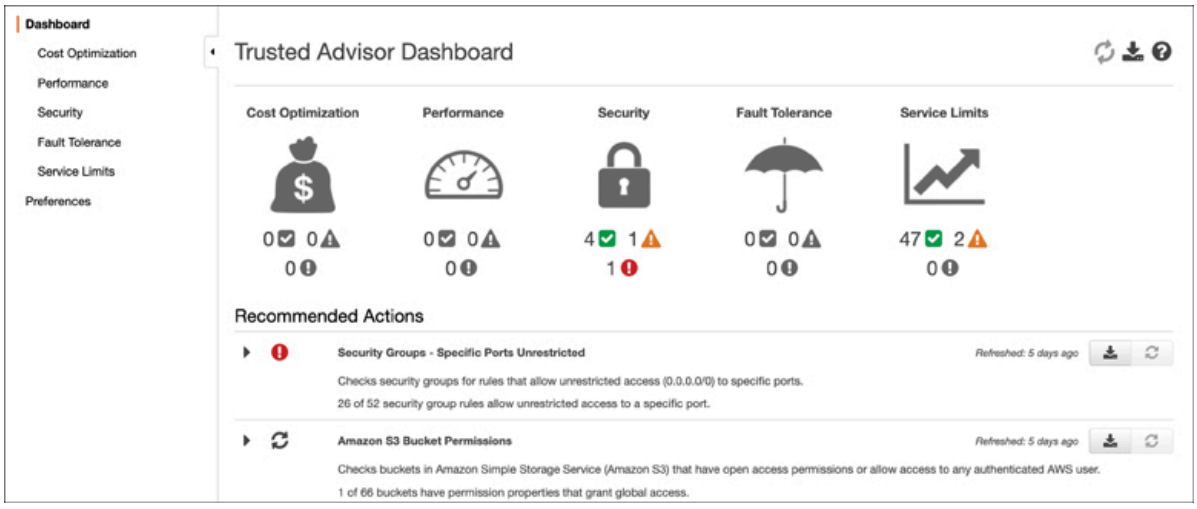


Figure 4-32 Trusted Advisor Security Checks

AWS Trusted Advisor can also provide additional checks if your organization has purchased Business or Enterprise support. Full AWS Trusted Advisor checks provide recommendations for improving performance, security, fault tolerance, and cost-effectiveness. AWS Trusted Advisor is useful to run against AWS account resources to re-

view current security issues and any flagged service quotas (see Figure 4-33).

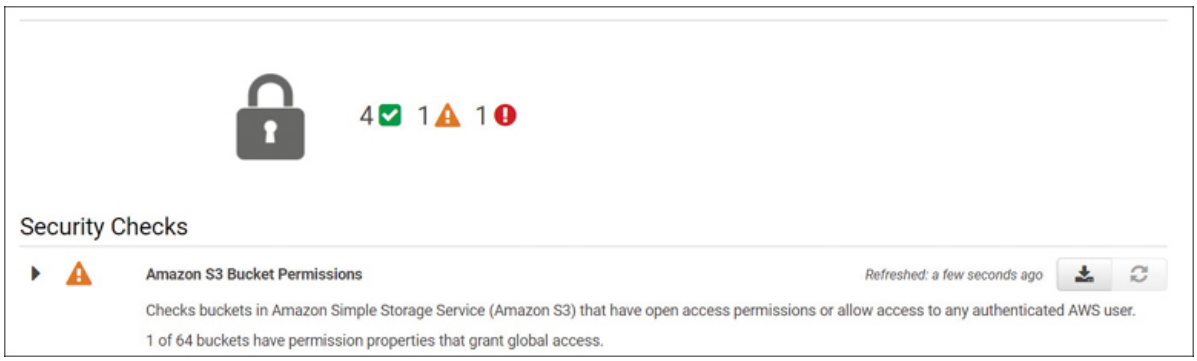


Figure 4-33 Trusted Advisor Security Checks Results

Once Business and Enterprise support has been purchased, AWS Trusted Advisor can alert you about issues to check, including the following:

- **Reserved Amazon EC2 instances:** AWS Trusted Advisor can calculate the optimized number of partial upfront reserved instances required based on an analysis of your usage history for the past month.
- **AWS ELB load balancers:** AWS Trusted Advisor checks current AWS ELB load balancing usage.
- **EBS volume check:** AWS Trusted Advisor warns if AWS EBS volumes in your AWS account are unattached or have low access rates.
- **Elastic IP addresses:** AWS Trusted Advisor warns if any Elastic IP addresses assigned to your account have not been associated. (Charges apply if Elastic IP addresses in your account are not used.)
- **Amazon RDS instances:** AWS Trusted Advisor checks for idle AWS RDS database instances.
- **Amazon Route 53 records:** AWS Trusted Advisor checks whether the creation of latency record sets has been properly designed to replicate end-user requests to the best AWS region.
- **Reserved reservation expiration check:** AWS Trusted Advisor warns you if your current reserved reservation is scheduled to ex-

pire within the next month. (Reserved reservations do not automatically renew.)

AWS Config

AWS Config enables customers to monitor, audit, and evaluate the deployed configurations of deployed IaaS resources, including EC2 instances, VPCs and components, IAM permissions, and S3 buckets deployed in a single AWS account or AWS accounts managed by an AWS organization. AWS Config provides detailed records of resource inventory, configuration history, and changes. Configuration data collected by AWS Config is stored in Amazon S3 buckets and Amazon DynamoDB.

The following are features of AWS Config:

- **Resource inventory:** Up-to-date inventory of selected AWS resources is recorded on an automated schedule.
- **Configuration History:** Configuration changes to AWS resources are tracked and stored, providing a historical view of changes over time.
- **Configuration Compliance:** Resources can be evaluated against predefined or custom rules, assessing the compliance of deployed AWS infrastructure components.
- **Management of Resources:** Centrally storing AWS resources helps an organization manage compliance and security standards.
- **Rules:** Managed rules created by AWS (see **Figure 4-34**) and custom rules can be used to evaluate resource configurations against predefined or custom criteria. Organizations can create their own custom AWS Config rules based on specific governance requirements such as security policies, compliance standards, or adhering to best practices. Custom rules are created using the AWS Lambda functions. Resource-specific rules could be created to evaluate the capacity of EC2 instances, the configuration of S3 buckets, or the configuration of a VPC.

- **Event Management:** SNS events can be generated when resource configurations change.

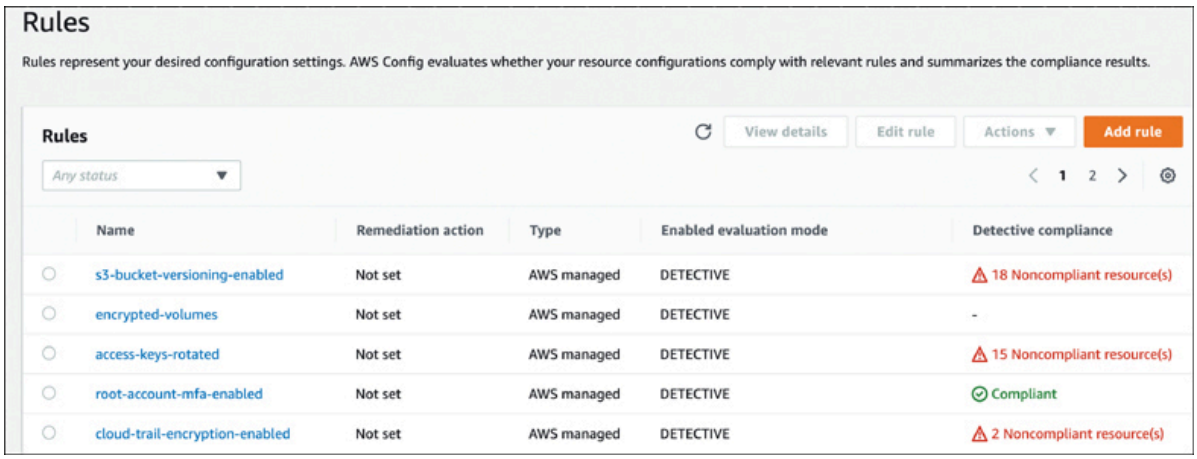


Figure 4-34 AWS Config Managed Rules

Exam Preparation Tasks

As mentioned in the section "**How to Use This Book**" in the Introduction, you have a couple of choices for exam preparation: the exercises here, **Chapter 16**, "**Final Preparation**," and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the margin of the page. **Table 4-9** lists these key topics and the page number on which each is found.

Table 4-9 Chapter 4 Key Topics

Key Topic Element	Description	Page Number
Figure 4-1	Connections and Security Services	150
Section	AWS Shield (Standard and Advanced)	151
Section	AWS Web Application Firewall (WAF)	152

Key Topic Element	Description	Page Number
Section	The Main Route Table	155
Section	Custom Route Tables	155
Section	Route Table Cheat Sheet	158
Section	Security Groups	158
Section	Security Groups Cheat Sheet	161
Section	Understanding Ephemeral Ports	165
Figure 4-11	Security Group Design	168
Section	Network ACLs	168
Section	Network ACL Cheat Sheet	169
Section	VPC Flow Logs	172
Section	NAT Services	174
Section	AWS NAT Gateway Service Cheat Sheet	176
Section	Amazon Cognito	176
Section	External Connections	180
Section	AWS Direct Connect Cheat Sheet	187
Section	Amazon GuardDuty	187

Key Topic Element	Description	Page Number
Section	Amazon Macie	189
Section	Security Services for Securing Workloads	191
Section	AWS CloudTrail Cheat Sheet	194
Section	AWS Secrets Manager	194

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

AWS Direct Connect

security group (SG)

network access control list (NACL)

NAT gateway service

Q&A

The answers to these questions appear in **Appendix A**. Use the Pearson Test Prep Software Online for more practice with exam format questions.

1. What AWS networking services can replace existing hardware devices?

2. What can network ACLs do that a security group cannot do?

3. What is the benefit of using CloudTrail trails for all AWS regions?

4. What is the benefit of using AWS Secrets Manager?

5. What type of artificial intelligence is used to operate GuardDuty?

6. How can Direct Connect help with high-speed connections to multiple VPCs?

7. For what assessments are you not required to have the Amazon Inspector agent installed?

8. How do you enable all checks for Trusted Advisor?