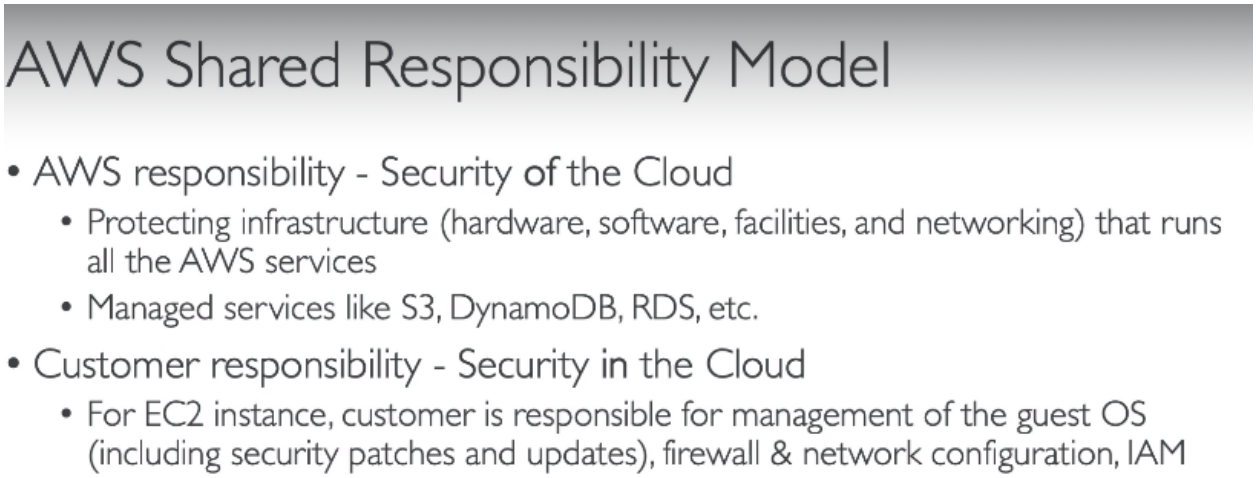
Shared Responsibility Model : Remainder



1.
2. So for example, as a customer, you are responsible for the security in the cloud. So in the instance of EC2 instance as a customer of aws we are responsible for the management of all the operating system that includes patching the operating system and making updates to it.
3. We have to configure firewalls for our ec2 instance..like NACL..and also you need to make sure that your EC2 instance has the correct IAM information through the use of, IAM instance role.

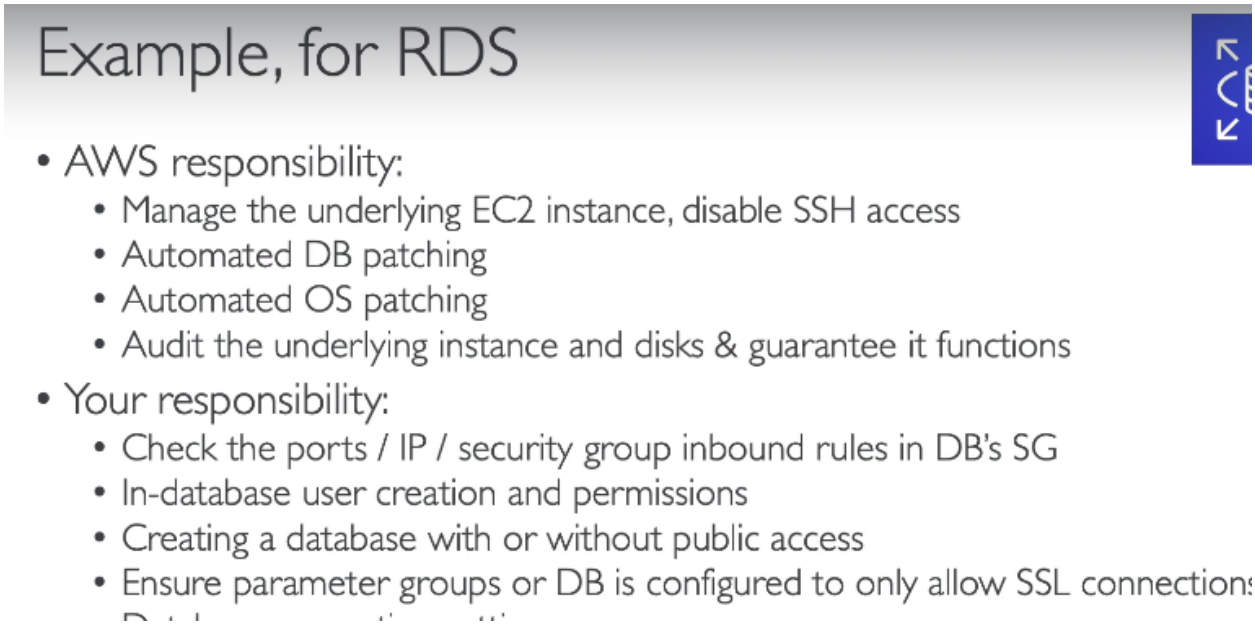
4.
5. For example, for awareness and training, AWS has to train their employees to use their facilities correctly, and to make sure they adhere to their security guidelines. And you have to make sure to train your employees correctly, to use the cloud, and doing this training is one of these ways, obviously.
6.

7. Shared controls in AWS are security controls that are jointly managed by AWS and its customers. These controls apply to both the infrastructure layer and the customer layer, but in completely separate contexts or perspectives.

   For example, AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest operating system and applications. AWS maintains the configuration of its infrastructure devices, but customers are responsible for configuring their own guest operating systems, databases, and applications.

   Shared controls can be a source of confusion for customers, as it can be difficult to determine which party is responsible for implementing and maintaining a particular control. However, AWS provides documentation and guidance on shared controls to help customers understand their responsibilities.

   Here are some examples of shared controls in AWS:

   - **Patch management:** AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest operating system and applications.
8. Lets take example of RDS
9. 

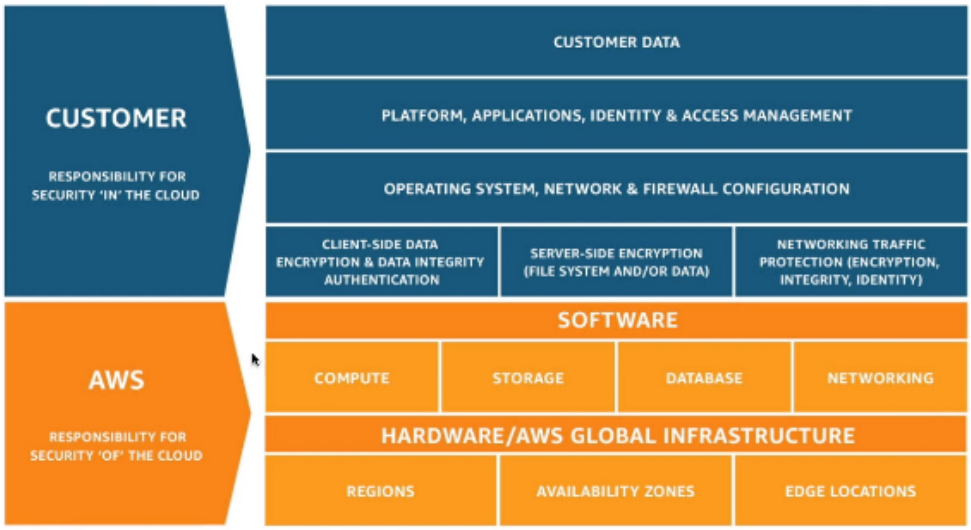
10. Example of s3

Example, for S3

- AWS responsibility:
  - Guarantee you get unlimited storage
  - Guarantee you get encryption
  - Ensure separation of the data between different customers
  - Ensure AWS employees can't access your data
- Your responsibility:
  - Bucket configuration
  - Bucket policy / public setting
  - IAM user and roles
  - Enabling encryption

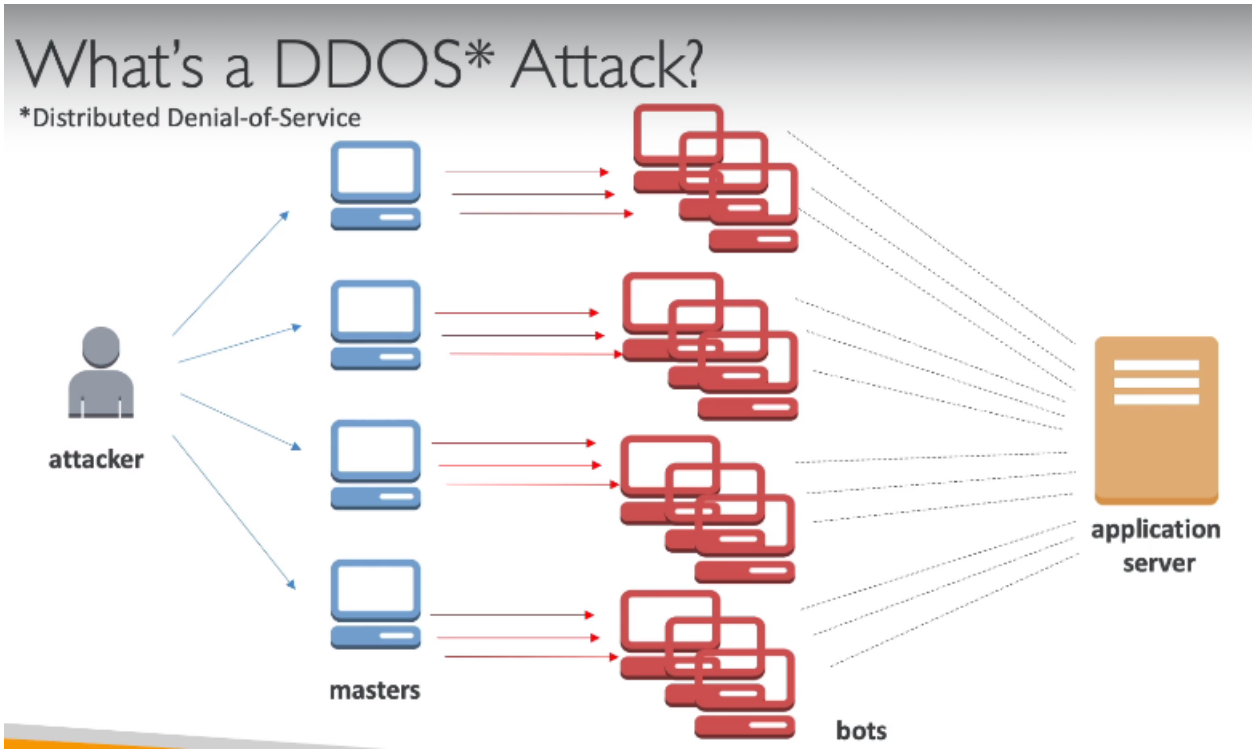
11.

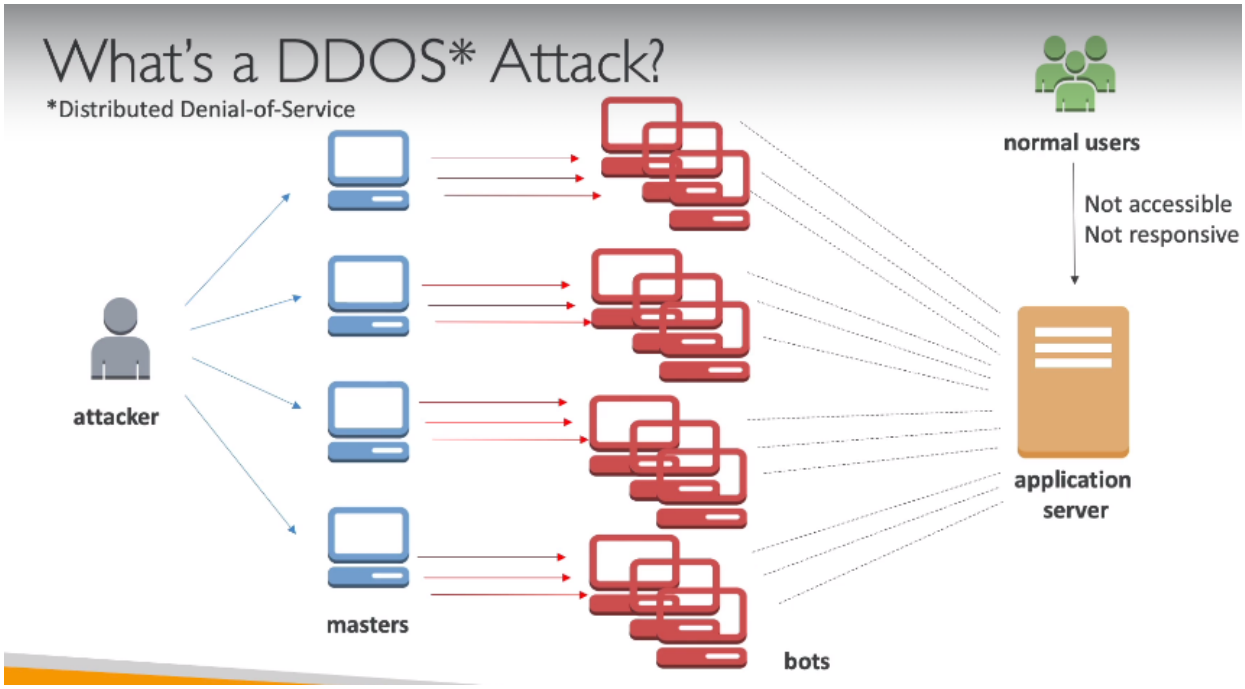
Shared Responsibility Model diagram



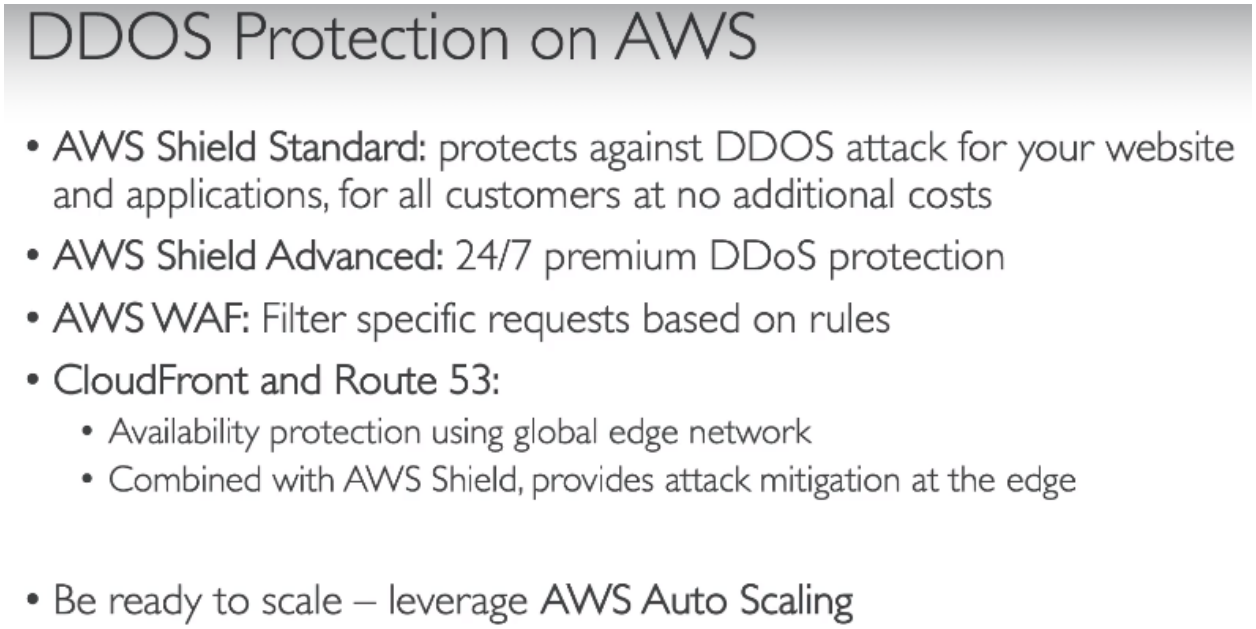
12.

DDoS Protection:WAF & shield

1. 

2. 

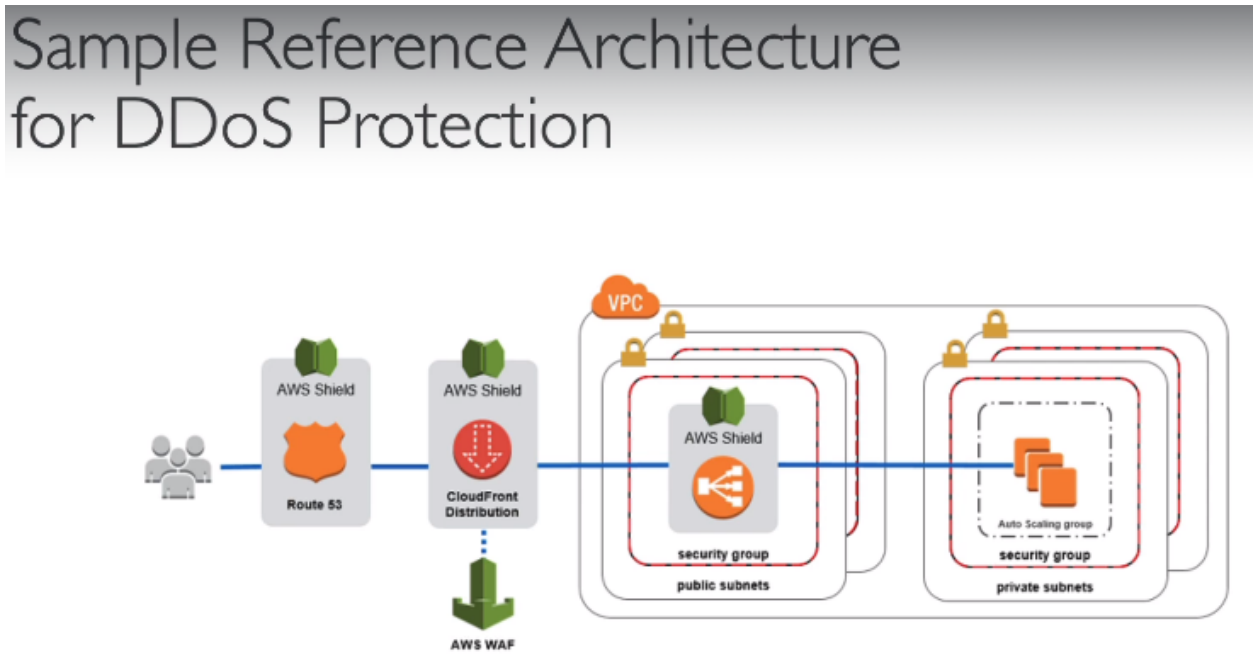
3. and therefore any normal user trying to connect to our application server, will see that our server is not accessible or not responsive,effectively making our application down.



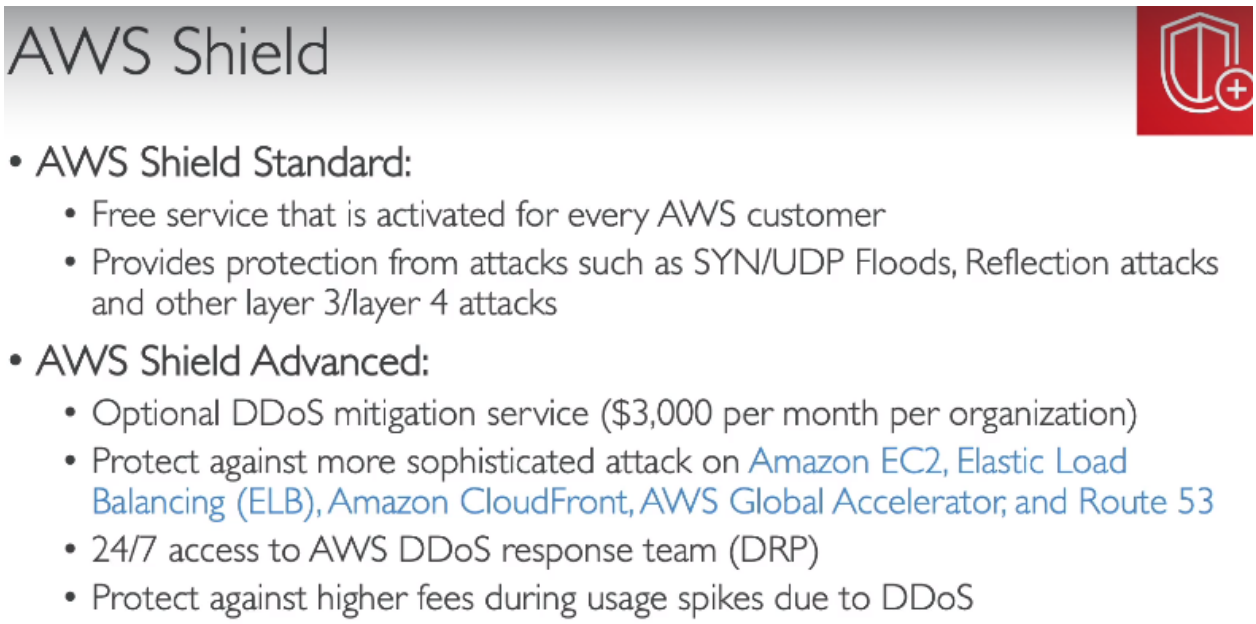
4.
5. Even if we getting attacked…. we need to be ready to scale..to avoid crashing and giving users to access our application

6. AWS offers a variety of services to help protect against DDoS attacks, including:

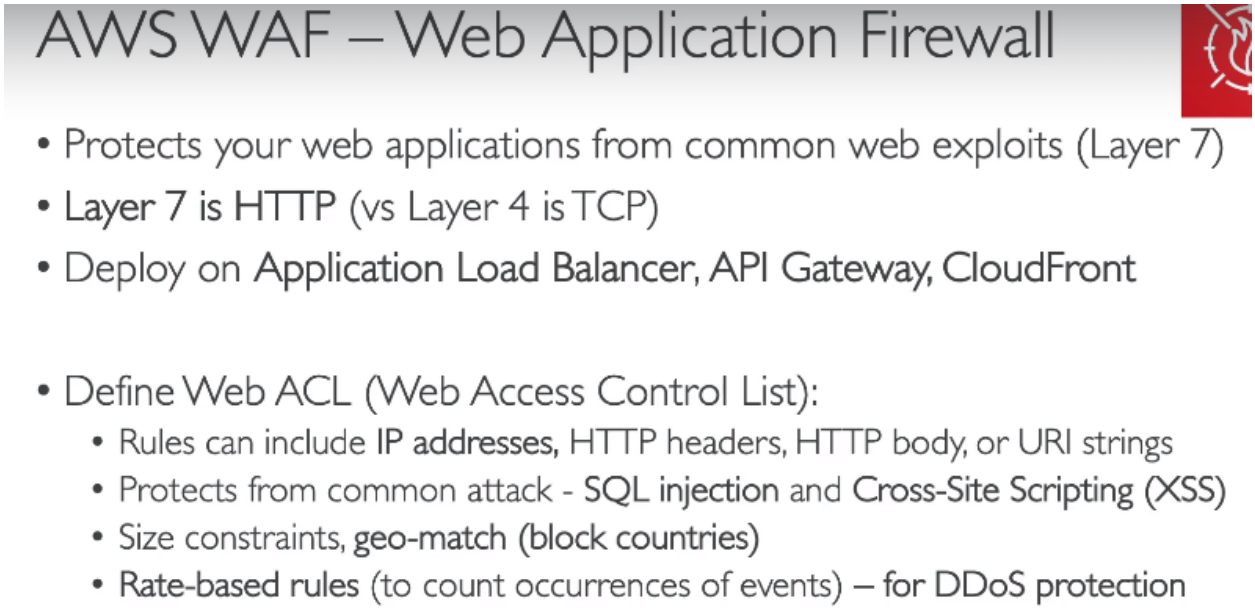
   - **AWS Shield Standard:** This is a free service that provides basic DDoS protection for all AWS customers. Shield Standard uses a combination of network flow monitoring, signature-based detection, and anomaly detection to identify and mitigate DDoS attacks.
   - **AWS Shield Advanced:** This is a paid service that provides more comprehensive DDoS protection than Shield Standard. Shield Advanced includes all of the features of Shield Standard, as well as additional features such as:
     - Application layer DDoS protection
     - Custom detection rules
     - 24/7 support from AWS experts
   - **AWS WAF:** This is a web application firewall (WAF) that can be used to protect web applications from a variety of attacks, including DDoS attacks. WAF can be used to block malicious traffic, filter traffic, and rate limit traffic.
   - **AWS CloudFront:** This is a content delivery network (CDN) that can be used to distribute your content to users around the world. CloudFront can help to protect your content from DDoS attacks by caching your content in multiple locations and by using a variety of techniques to mitigate DDoS attacks.

7. 

8. So we have our users
9. and they will be routed through the DNS on Route 53 which is protected by shield so your DNS is safe from DDoS attack, then you should use a CloudFront distribution to make sure your content is cached at the edge and then it is also protected by shield
10. and in case you need to filter and protect from an attack, you can use the web application firewall, then to serve that application you can use a load balancer in the public subnet that will scale for you
11. and finally behind the load balancer you should use EC2 instances in an auto scaling group to be able to scale to the higher demand.
12. AWS Shield



13.

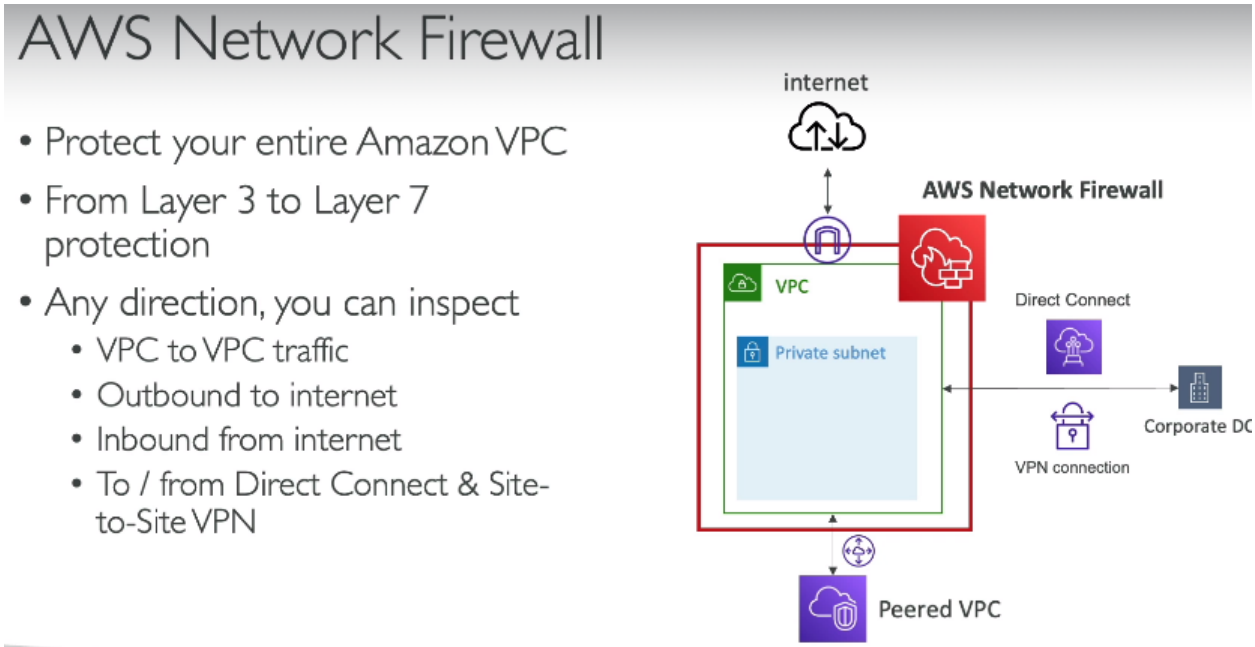
14. 

15. Sure. AWS WAF (Web Application Firewall) is a web application firewall that helps protect web applications from a variety of attacks, including common web exploits and bots that can affect availability, compromise security, or consume excessive resources. AWS WAF can be used to block malicious traffic, filter traffic, and rate limit traffic.

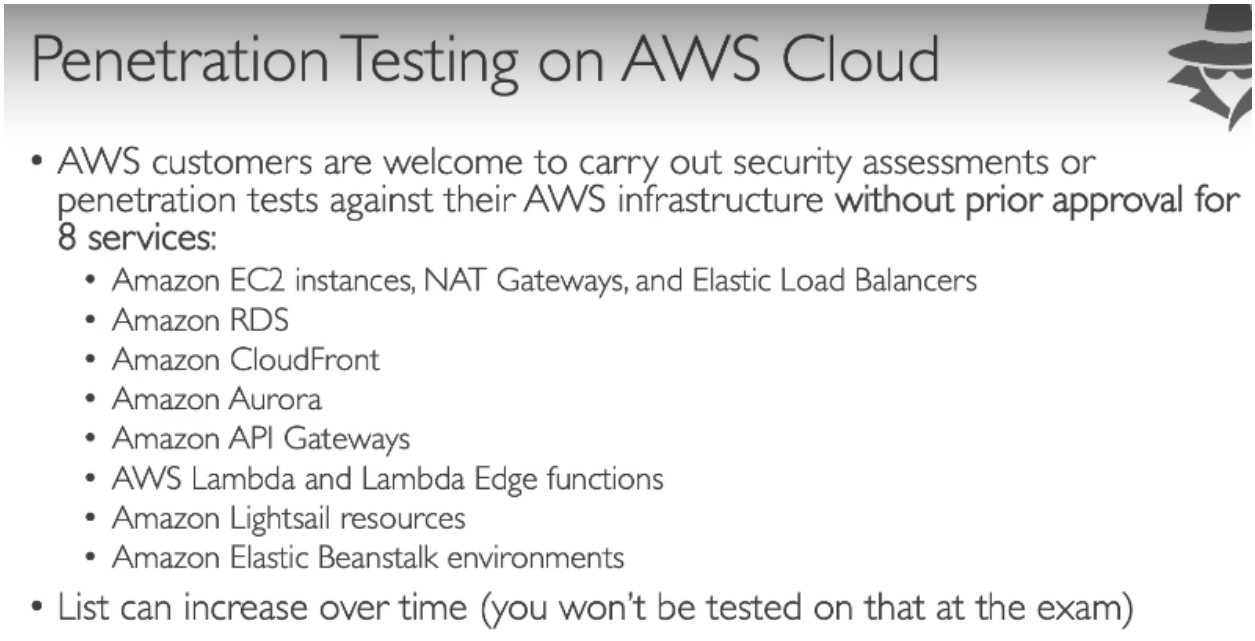
    Here is a practical example of how AWS WAF can be used to protect a web application from a DDoS attack. Let's say you have a web application that is hosted on Amazon Elastic Compute Cloud (EC2) instances. You can use AWS WAF to create a rule that blocks traffic from any IP address that sends more than 100 requests per second to your web application. This will help to protect your web application from a DDoS attack that is trying to overwhelm your EC2 instances with traffic.

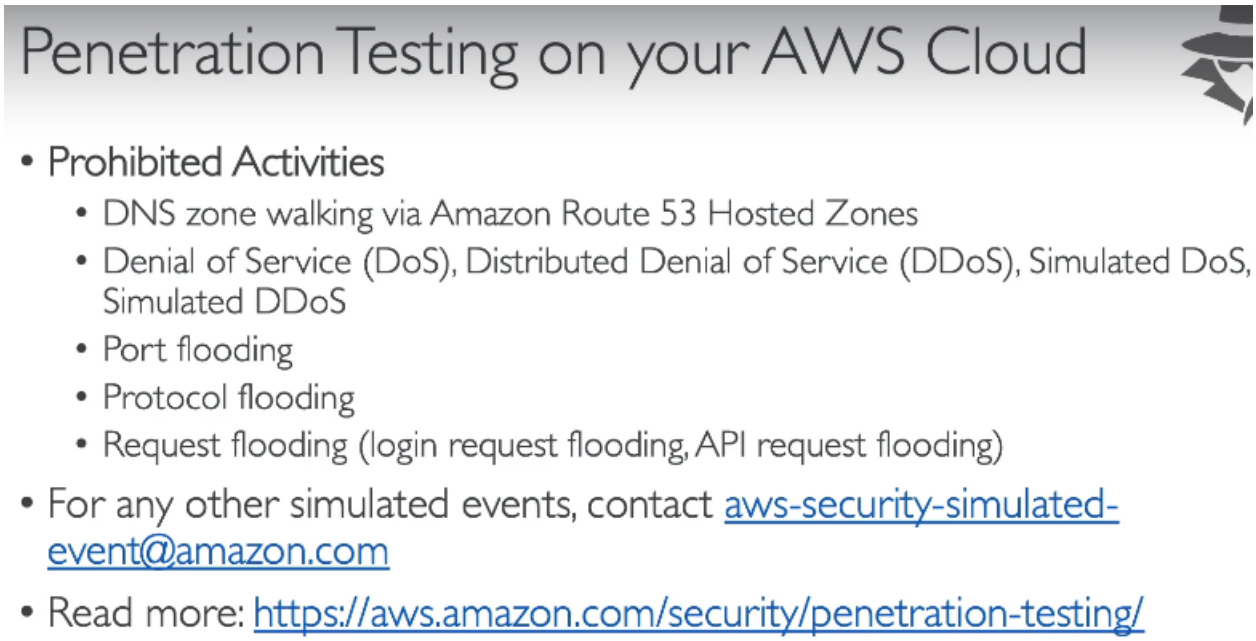
16. Sure. A web access control list (Web ACL) is a collection of rules that you define to control the HTTP(S) traffic that your web application receives. AWS WAF uses Web ACLs to inspect and filter web requests before they reach your application.

AWS Network FireWall

1. 

Penetration testing

1. 

2. 

3. DNS zone walking is a type of attack that attempts to retrieve all of the records in a DNS zone. This can be done by issuing a series of DNS queries, starting with the zone apex (the domain name itself) and then recursively querying for all of the subdomains.

In AWS, DNS zone walking can be performed using the Route 53 API or the Route 53 console. To perform a zone walk using the API, you would use the following command:

Code snippet

```
aws route53 list-resource-record-sets --hosted-zone-id Z123ABCDE890ABCDE
```

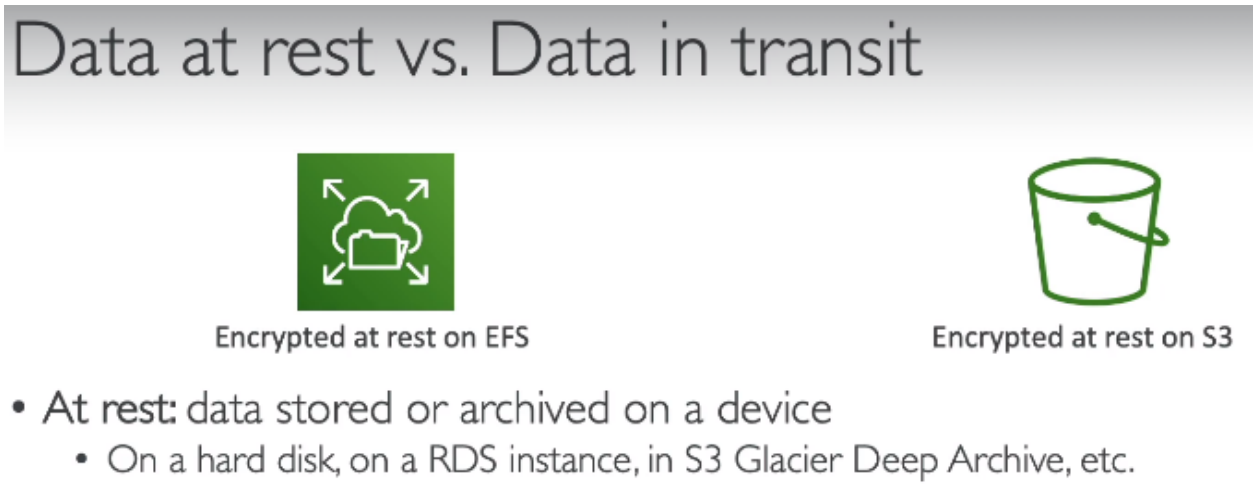
Use code with caution. Learn more

This command will return a list of all of the resource record sets in the specified hosted zone.

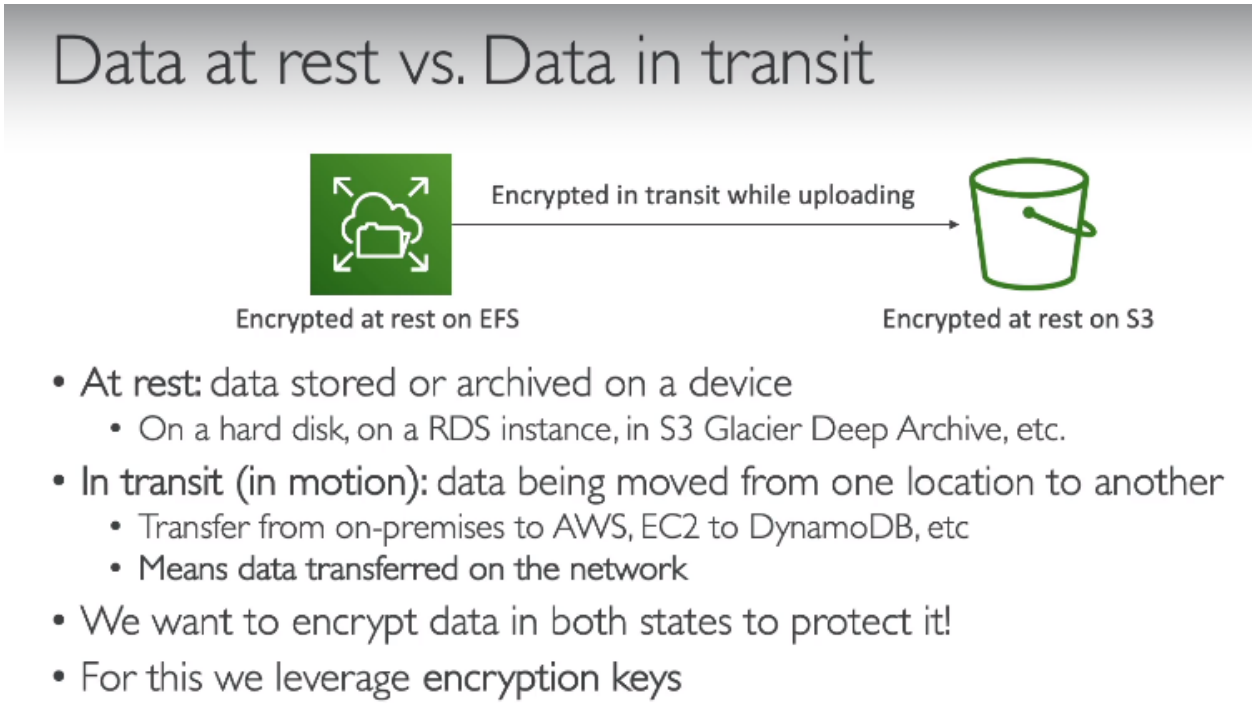
To perform a zone walk using the console, you would go to the Route 53 console and select the hosted zone that you want to walk. Then, you would click on the "Resource record sets" tab. This will display a list of all of the resource record sets in the hosted zone.

Encryption with KMS and cloudHSM

1. Here we have two types of encryption…which is data at rest & data in transit



2.
3. Here we can see data is encrypted and stores in EFS and S3



4.

5. Leveraging encryption keys in AWS allows you to protect your data by encrypting it with a key that is stored in AWS Key Management Service (KMS). This ensures that your data is only accessible to authorized users and that it cannot be decrypted without the key.

   Here are some of the benefits of leveraging encryption keys in AWS:

   - **Data security:** Encryption keys help to protect your data from unauthorized access, even if your data is stored in the cloud.
   - **Compliance:** Encryption keys can help you to comply with data security regulations, such as HIPAA and PCI DSS.
   - **Auditability:** Encryption keys can help you to track who has access to your data and when they accessed it.
   - **Scalability:** Encryption keys can be used to encrypt data stored in any AWS service, so you can easily scale your security as your needs grow.

6. Sure. Here is an example of how you can leverage encryption keys in AWS to protect your data:

   Let's say you have a database of customer records that you need to store in the cloud. You can use AWS KMS to create an encryption key and then use that key to encrypt the data in your database. This will ensure that your data is only accessible to authorized users and that it cannot be decrypted without the key.

   To do this, you would first need to create an encryption key in AWS KMS. You can do this using the AWS KMS console or the AWS KMS API. Once you have created an encryption key, you can then use it to encrypt your data.
7. AWS KMS
8. So the encryption service at the center of AWS is called KMS, key management service. So anytime you hear encryption for a service it's most likely going to be KMS.
9. And so with KMS we don't have access to the keys. AWS will manage the keys for us and we just define who can access these keys.

AWS KMS (Key Management Service)

- Anytime you hear "encryption" for an AWS service, it's most likely KMS
- KMS = AWS manages the encryption keys for us
- Encryption Opt-in:
  - EBS volumes: encrypt volumes
  - S3 buckets: Server-side encryption of objects
  - Redshift database: encryption of data
  - RDS database: encryption of data
  - EFS drives: encryption of data
- Encryption Automatically enabled:
  - CloudTrail Logs
  - S3 Glacier
  - Storage Gateway

10.
11. We have an option to choose the encryption for like EBS,S3 etc..
12. CloudHSM

CloudHSM

- KMS => AWS manages the software for encryption
- CloudHSM => AWS provisions encryption **hardware**
- Dedicated Hardware (HSM = Hardware Security Module)
- You manage your own encryption keys entirely (not AWS)
- HSM device is tamper resistant, FIPS 140-2 Level 3 compliance

Sample HSM device

13.

14. Sure. Here is a practical example of how you can use CloudHSM in AWS:

Let's say you have a web application that stores customer credit card information. You want to store this information in a secure way, so you decide to use CloudHSM.

You would first need to create a CloudHSM cluster. You can do this using the AWS Management Console or the AWS CLI. Once you have created a CloudHSM cluster, you would need to create a Customer Master Key (CMK) in AWS KMS. You can do this using the AWS Management Console or the AWS CLI.

Once you have created a CMK, you would need to use the CloudHSM client to encrypt the customer credit card information. You can do this using the CloudHSM API or the CloudHSM CLI.

Once the customer credit card information is encrypted, you would need to store it in your web application. You can store the encrypted data in any AWS service that supports encryption, such as Amazon S3 or Amazon RDS.

To access the customer credit card information, you would need to use the CloudHSM client to decrypt the data. You can do this using the CloudHSM API or the CloudHSM CLI.

15. A customer master key (CMK) is a cryptographic key that you create and manage in AWS Key Management Service (KMS). You can use a CMK to encrypt your data and to perform cryptographic operations on your data.

    CMKs are stored in AWS KMS, which is a highly secure service that uses FIPS 140-2 Level 3 validated hardware and software to protect your keys. AWS KMS also provides detailed audit logs that you can use to track who has accessed your keys and when they accessed them.

    There are two types of CMKs:

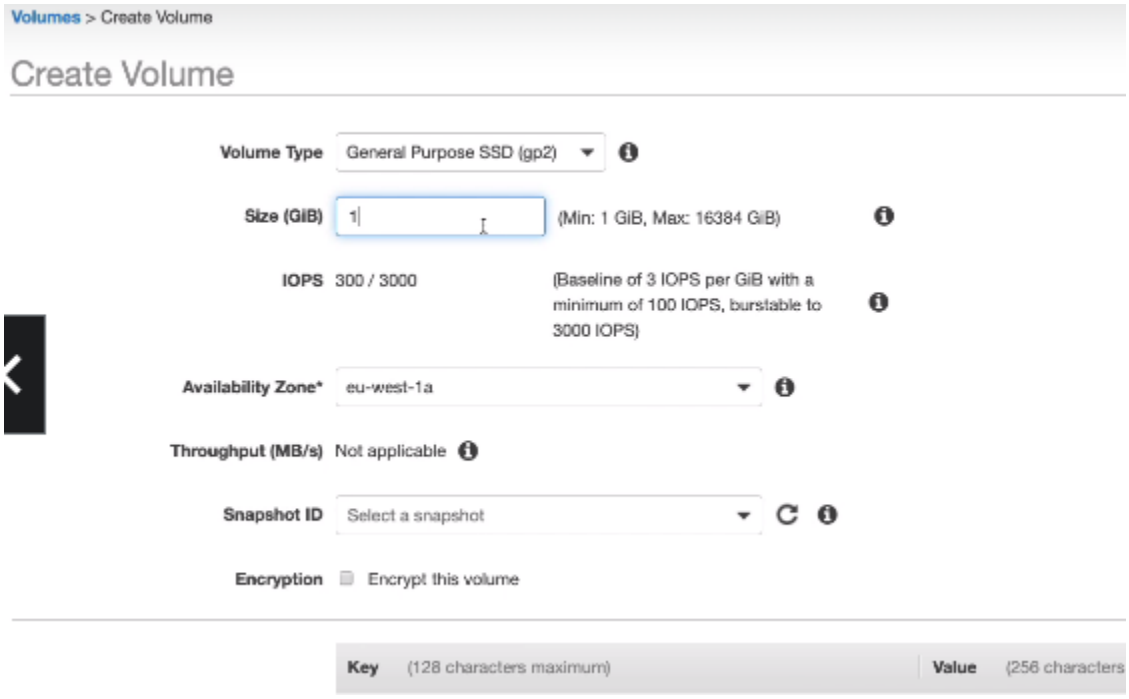
    - **Customer managed CMKs:** These are CMKs that you create and manage in AWS KMS. You have full control over these CMKs, including their key policies, IAM policies, and grants.
    - **AWS managed CMKs:** These are CMKs that are created and managed by AWS. AWS managed CMKs are used by AWS services to encrypt your data. You cannot directly use AWS managed CMKs, but you can use them indirectly by using the AWS services that use them.

16.
17. So this customer managed CMK are the keys that we create, manage and use ourselves, AWS users.

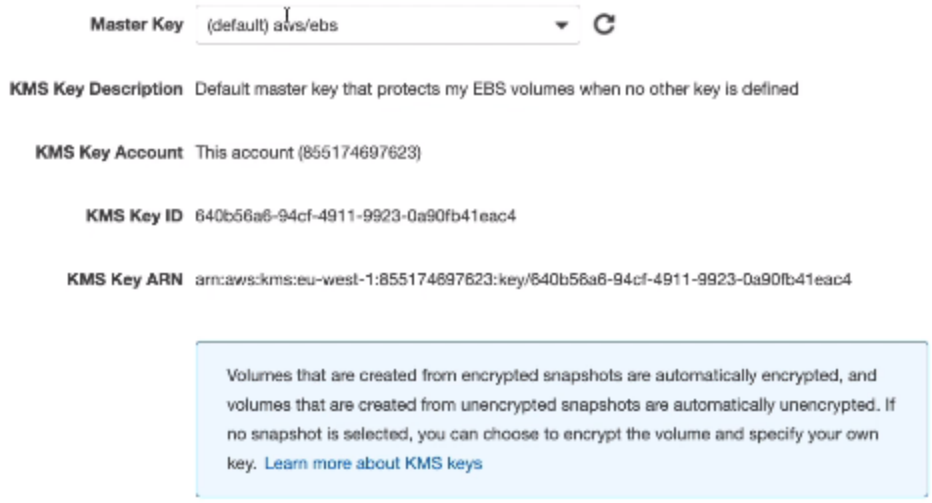
18. And we can create them, enable them or disable them. We can define a rotation policy for these keys, for example a new key generated every year while the old keys of course preserved. And we can bring our own key.

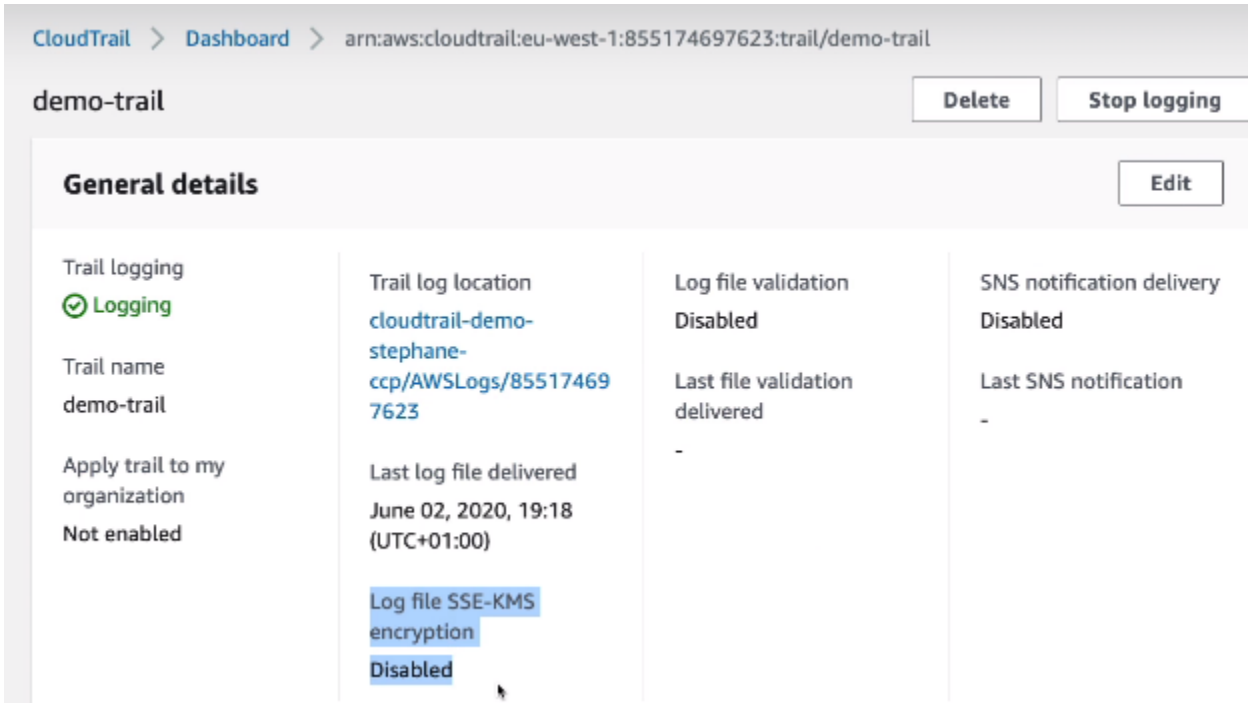
AWS KMS & cloudHSM hands on

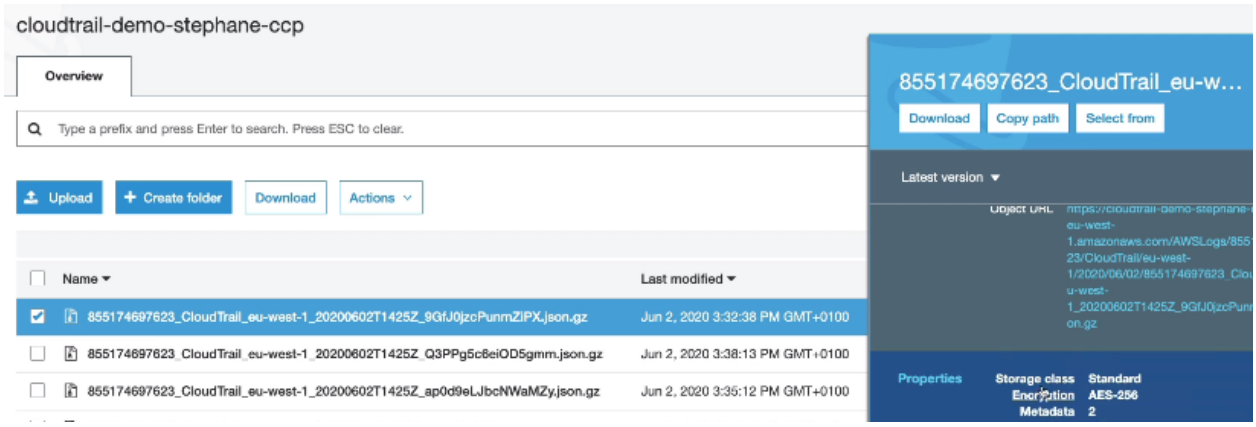
1. Here first we have create a volume for our ec2 instance and have to opt in encryption and gave protection to it using aws/ebs

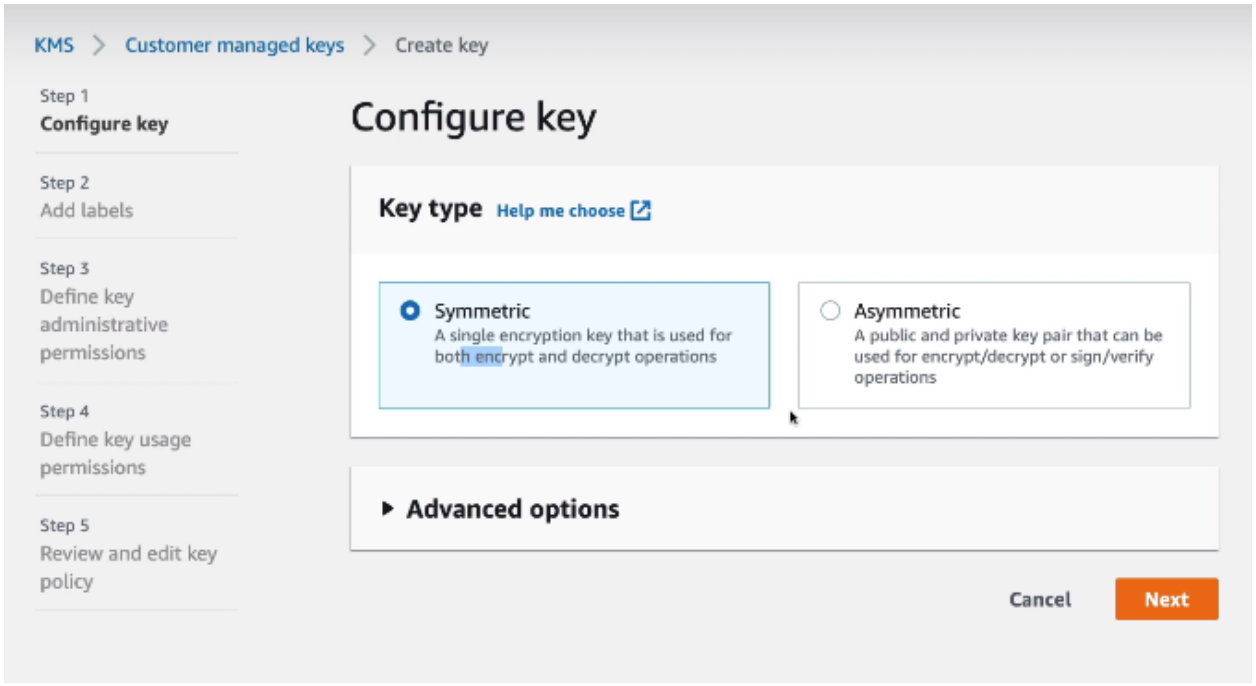


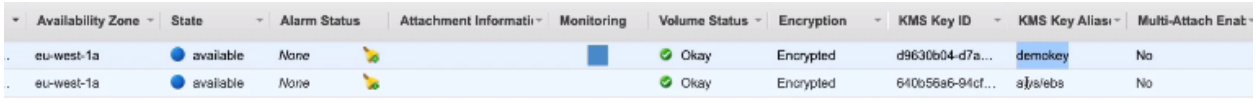
2.



3.
4. Then click on create volume
5. And now we have created an encrypted volume. So it is properly protected against for example, attacks if they're trying to decrypt it.
6. where this is a voluntary opt-in for the encryption.But if you remember, I said that for example, for CloudTrail or for Glacier S3 encryption was enabled by default.
7. 

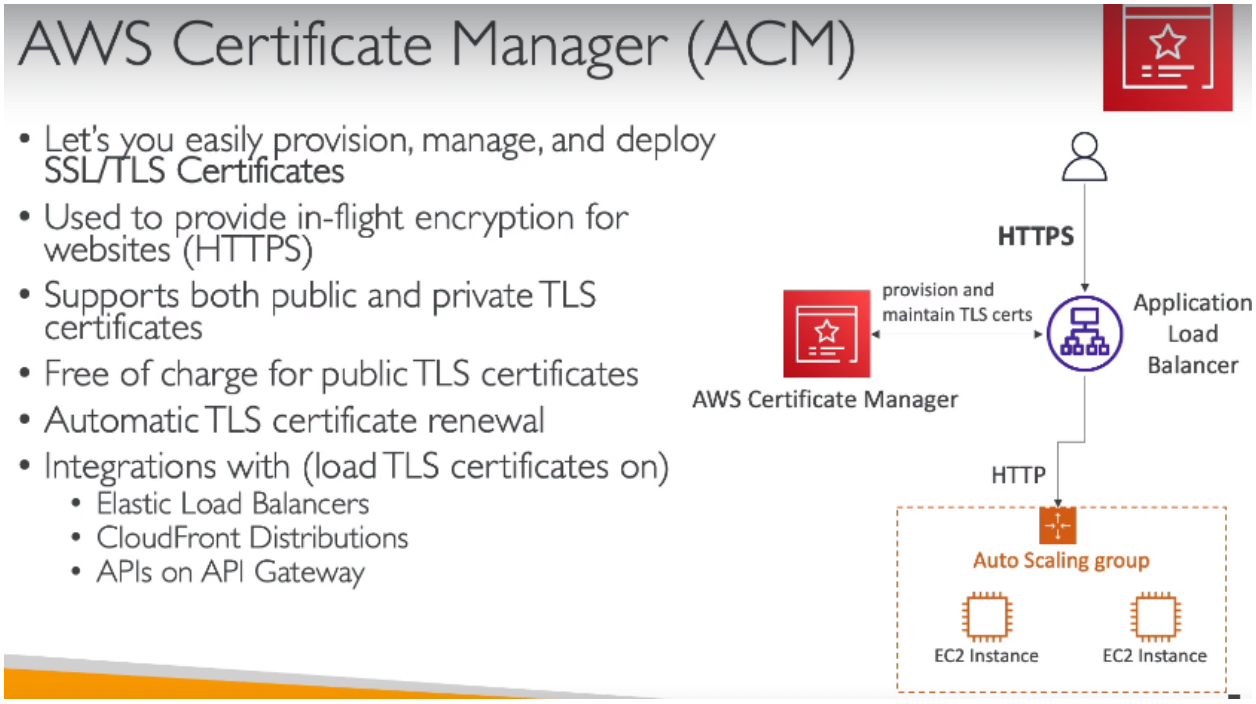
8. 

9. even though it said disabled here for the encryption, it actually was enabled in Amazon S3.
10. In customer managed keys
11. 

12. If we create a CMS..then we get an option to choose encryption for our volume
13. 

14. Here we have created 2 volumes with diff types of encryptions ..

AWS Certificate Manager(ACM)

1. 

2. Sure. AWS Certificate Manager (ACM) is a fully managed service that makes it easy to request, deploy, and manage SSL/TLS certificates for your website or application. ACM can be used to provision certificates for a wide variety of use cases, including:

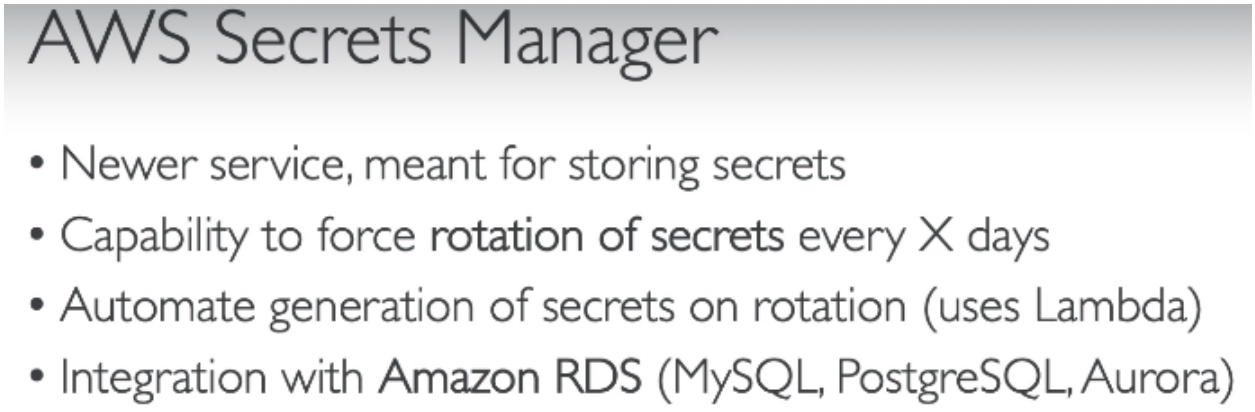
   - **Ecommerce websites:** ACM can be used to provision certificates for ecommerce websites, which helps to protect sensitive data, such as credit card numbers, from being intercepted by unauthorized third parties.
   - **Banking websites:** ACM can also be used to provision certificates for banking websites, which helps to protect customers' financial information from being intercepted by unauthorized third parties.
   - **Email:** ACM can also be used to provision certificates for email servers, which helps to protect the confidentiality of email messages.
   - **Remote access:** ACM can also be used to provision certificates for remote access servers, which helps to protect sensitive data from being accessed by unauthorized users.
3. So anytime you see what service can help us do in-flight encryption and generates these certificates then think ACM, that's it.

4. Inflight encryption in AWS Certificate Manager (ACM) is a feature that encrypts all traffic between your website or application and your users' browsers. This helps to protect sensitive data, such as credit card numbers and passwords, from being intercepted by unauthorized third parties.

   Inflight encryption works by using a technique called Transport Layer Security (TLS). TLS is a cryptographic protocol that encrypts data in transit. When a user visits your website or application, their browser will request a TLS connection from your server. Your server will then send back a TLS certificate, which contains the public key for your website or application. The user's browser will then use the public key to encrypt data that it sends to your server. Your server will then decrypt the data using its private key.

   This process ensures that all traffic between your website or application and your users' browsers is encrypted and cannot be intercepted by unauthorized third parties.

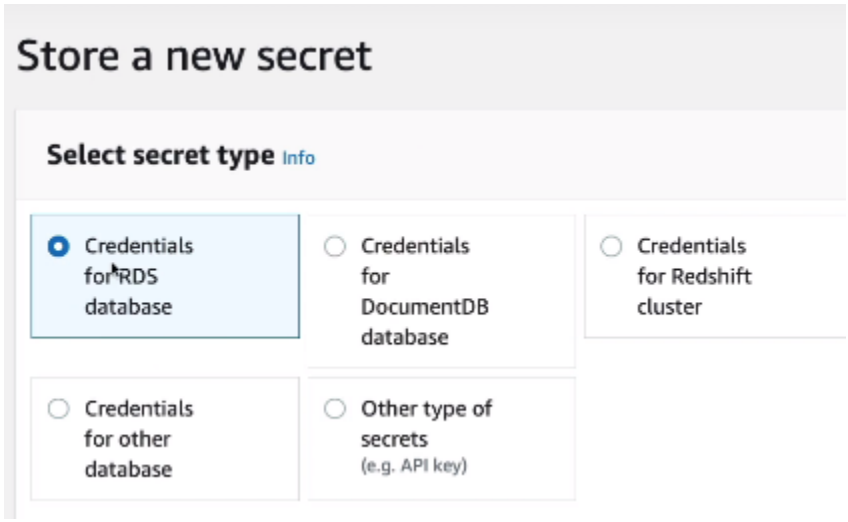
AWS Secrets Manager

1. 

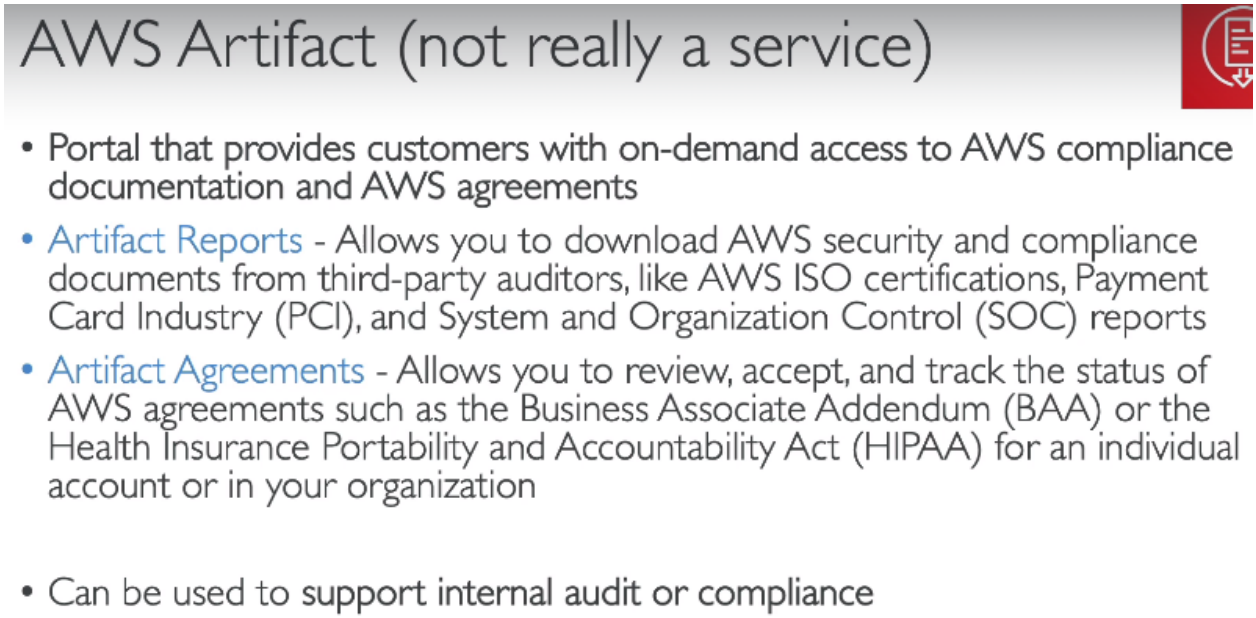
2. And it has an integration with Amazon RDS. So using the Secrets Manager, we can create the passwords for Amazon RDS automatically.
3. The secrets are going to be encrypted using KMS, that we just saw from before, automatically.
4. And so from an exam perspective, anytime you see a secret to be managing in RDS and to be rotated, you have to think about Secrets Manager.

5. • Secrets are encrypted using KMS

   • Mostly meant for RDS integration

6. 

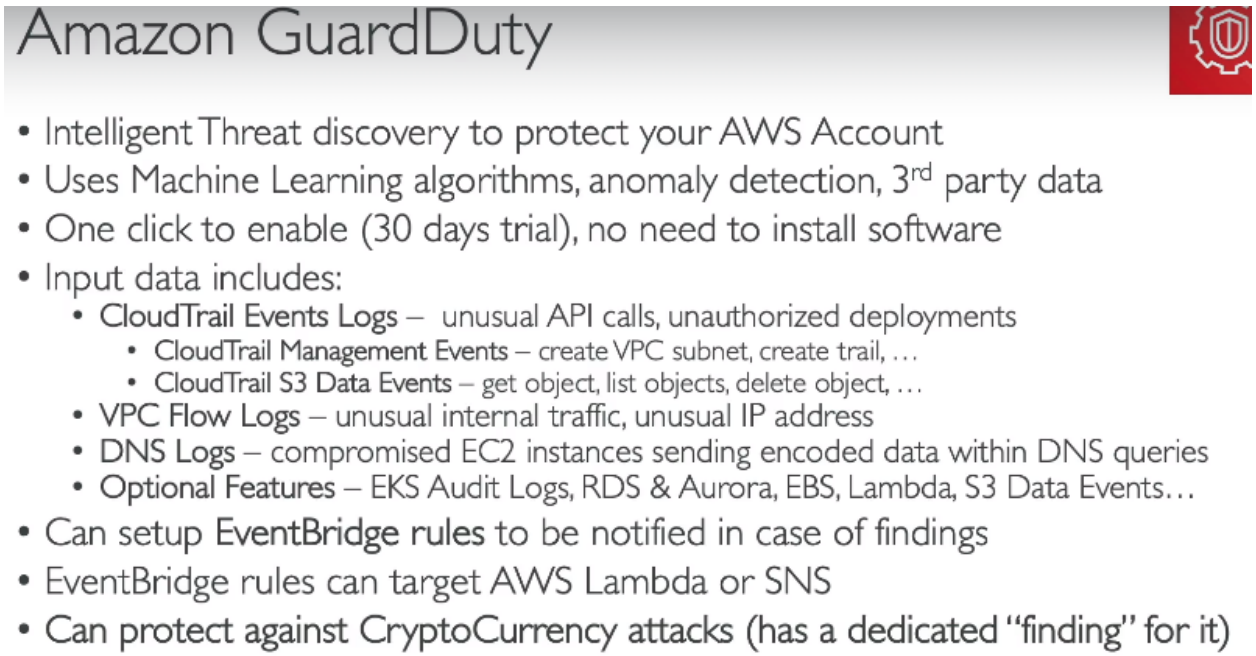
AWS ArtiFact



1.
2. Can be used to download reports regarding security and compliances etc
3. It's not really a service. It's a way for you to download compliance documents,

Amazon GuardDuty

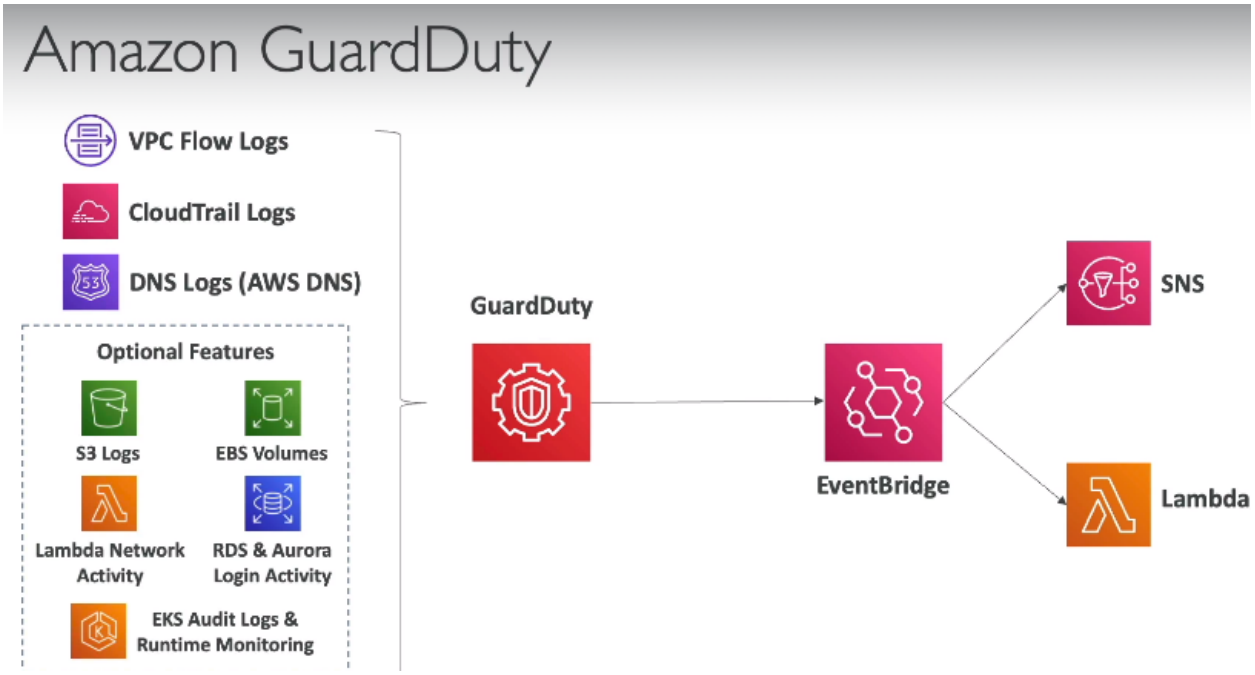
1.



2. Sure. Amazon GuardDuty is a continuous security monitoring service that helps to identify and investigate potential security threats to your AWS resources. GuardDuty uses machine learning and anomaly detection to identify unusual activity in your AWS environment, and it can alert you to potential threats even before they cause damage.

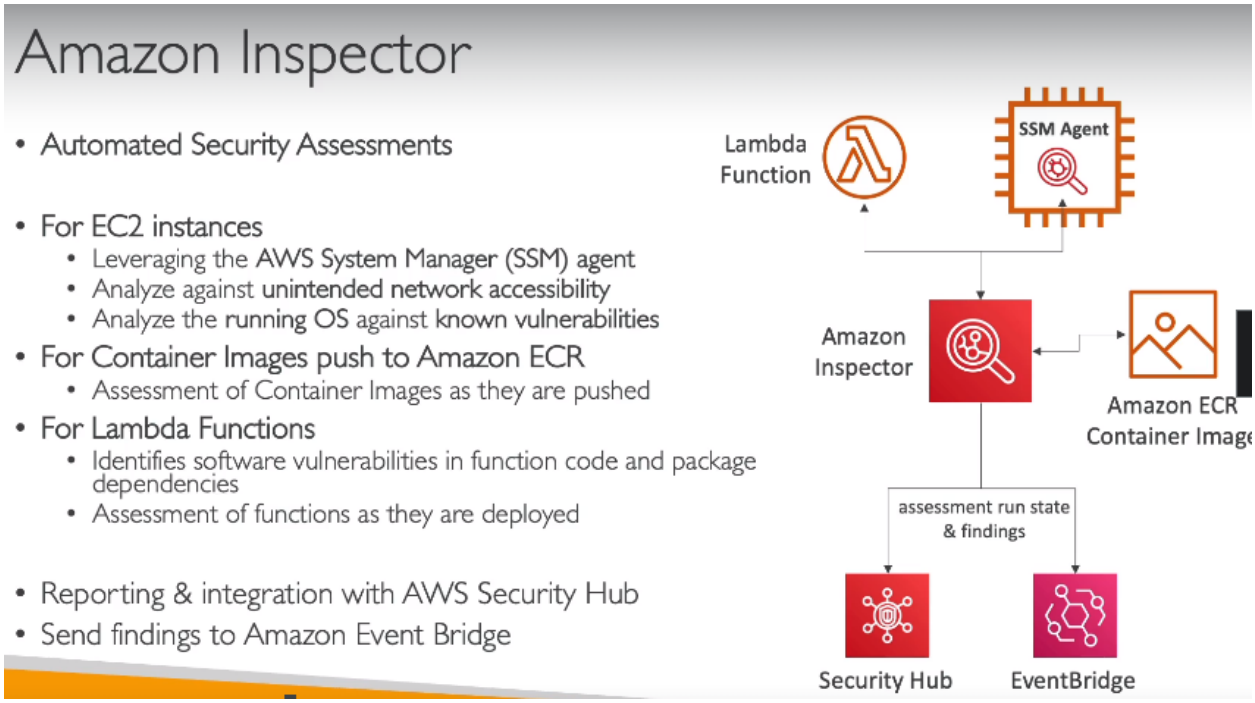
   Here is an example of how you can use GuardDuty to protect your AWS resources:

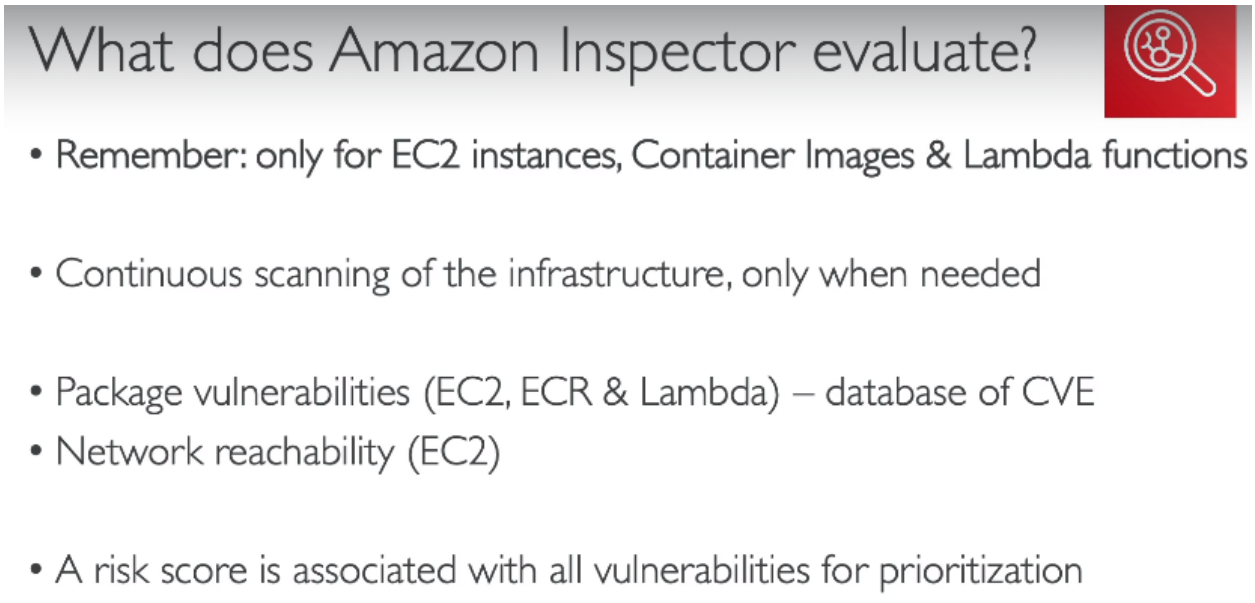
   - Let's say you have an EC2 instance that is running a web server. GuardDuty can monitor the instance for unusual activity, such as a large number of failed login attempts or a sudden increase in network traffic. If GuardDuty detects any unusual activity, it will alert you so that you can investigate the issue.
   - GuardDuty can also monitor your VPC for unauthorized access. If GuardDuty detects any unauthorized access, it will alert you so that you can take action to block the unauthorized access.
   - GuardDuty can also monitor your AWS CloudTrail logs for suspicious activity. If GuardDuty detects any suspicious activity, it will alert you so that you can investigate the issue.

3. 

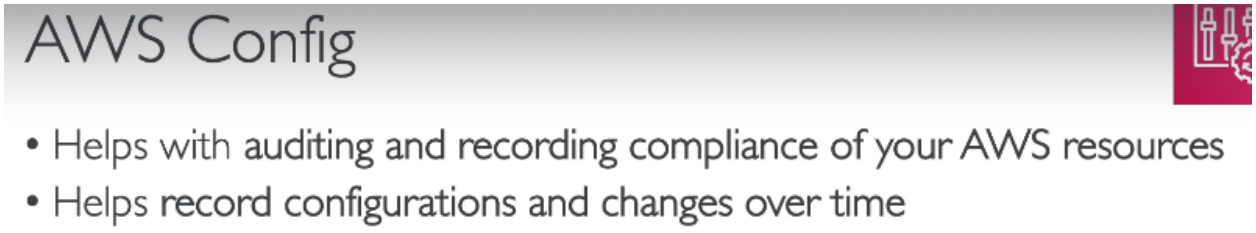
4. If any finding are detected..then it creates an event bridge..with that we can notify the customers

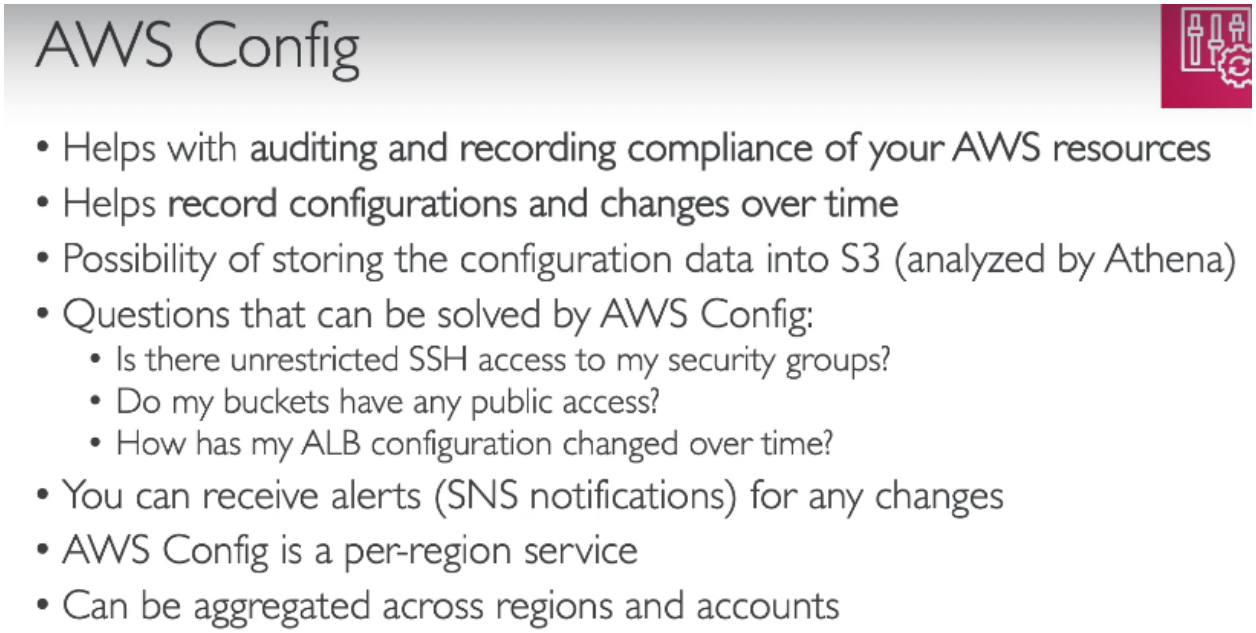
Amazon Inspector

1. 

2. 

AWS Config

1. 

2. So any time we've been doing some manual changes of the configuration in AWS, we did not have a list of all the changes that happened, but we can have this using Config.
3. 

AWS Config Resource

- View compliance of a resource over time

- View configuration of a resource over time

- View CloudTrail API calls if enabled

4.
5. And finally, you can view who made these changes to the resources based on CloudTrail if you have enabled CloudTrail in your accounts.
6.

Sure. AWS Config is a service that helps you to track and manage the configurations of your AWS resources. You can use AWS Config to:

- **Record the current configuration of your AWS resources:** AWS Config records the current configuration of your AWS resources in a central repository. This allows you to track changes to your configurations over time and to restore your resources to a previous configuration if necessary.
- **Identify misconfigurations:** AWS Config can identify misconfigurations in your AWS resources. This can help you to improve the security and compliance of your AWS environment.
- **Audit your AWS environment:** AWS Config can be used to audit your AWS environment to ensure that your resources are configured in accordance with your security policies and procedures.

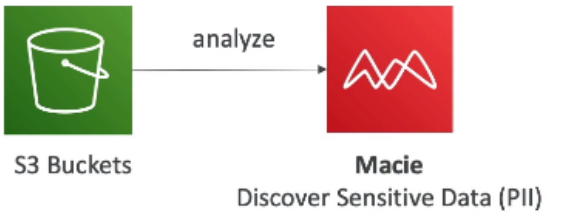
7. Here is an example of how you can use AWS Config to track and manage the configurations of your AWS resources:

   - Let's say you have an EC2 instance that is running a web server. You can use AWS Config to record the current configuration of the EC2 instance, including the ports that are open, the permissions that are granted to users, and the software that is installed.
   - If you make a change to the configuration of the EC2 instance, AWS Config will record the change. This allows you to track changes to the configuration of the EC2 instance over time and to restore the EC2 instance to a previous configuration if necessary.
   - AWS Config can also identify misconfigurations in the EC2 instance. For example, if you open a port that should not be open, AWS Config will identify the misconfiguration. This can help you to improve the security of the EC2 instance.
   - You can also use AWS Config to audit your AWS environment to ensure that your resources are configured in accordance with your security policies and procedures. For example, you can use AWS Config to ensure that all of your EC2 instances have strong passwords and that all of your software is up to date.

AWS Macie

Amazon Macie

- Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data in AWS.
- Macie helps identify and alert you to sensitive data, such as personally identifiable information (PII)



1.
2. So very simply, your PII data will be in your S3 buckets and it will be analyzed by Macie which will discover what data can be classified as PII.

3.

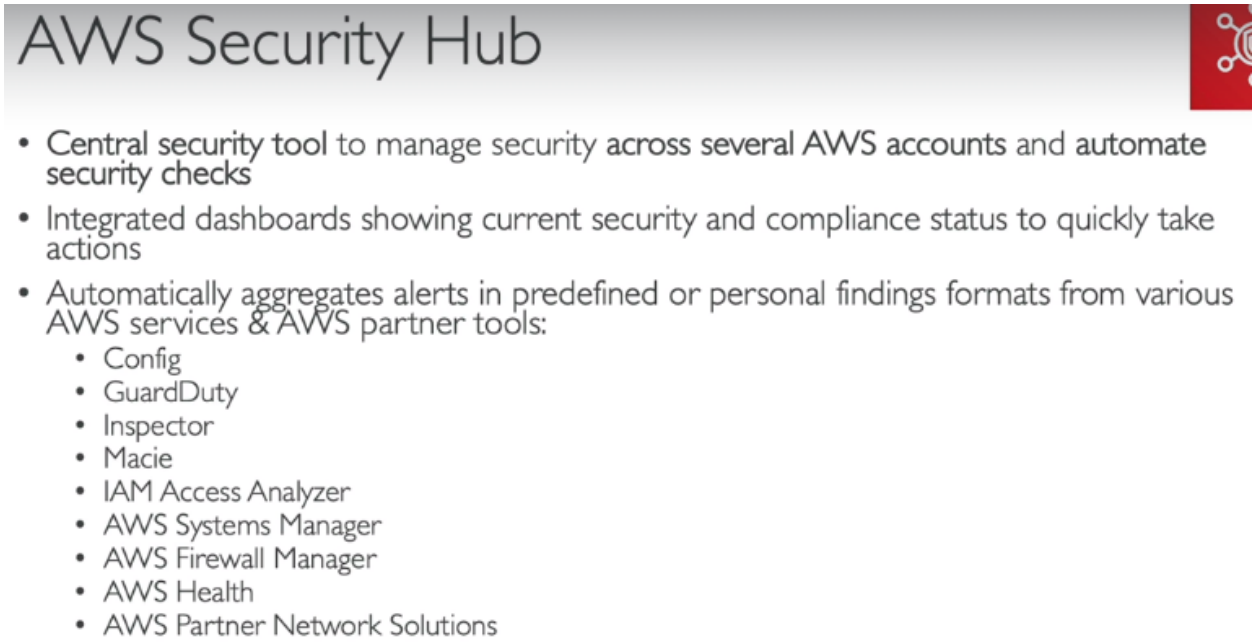
4. Sure. Amazon Macie is a fully managed data security and compliance service that uses machine learning to discover, classify, and protect sensitive data in AWS. Macie can be used to:

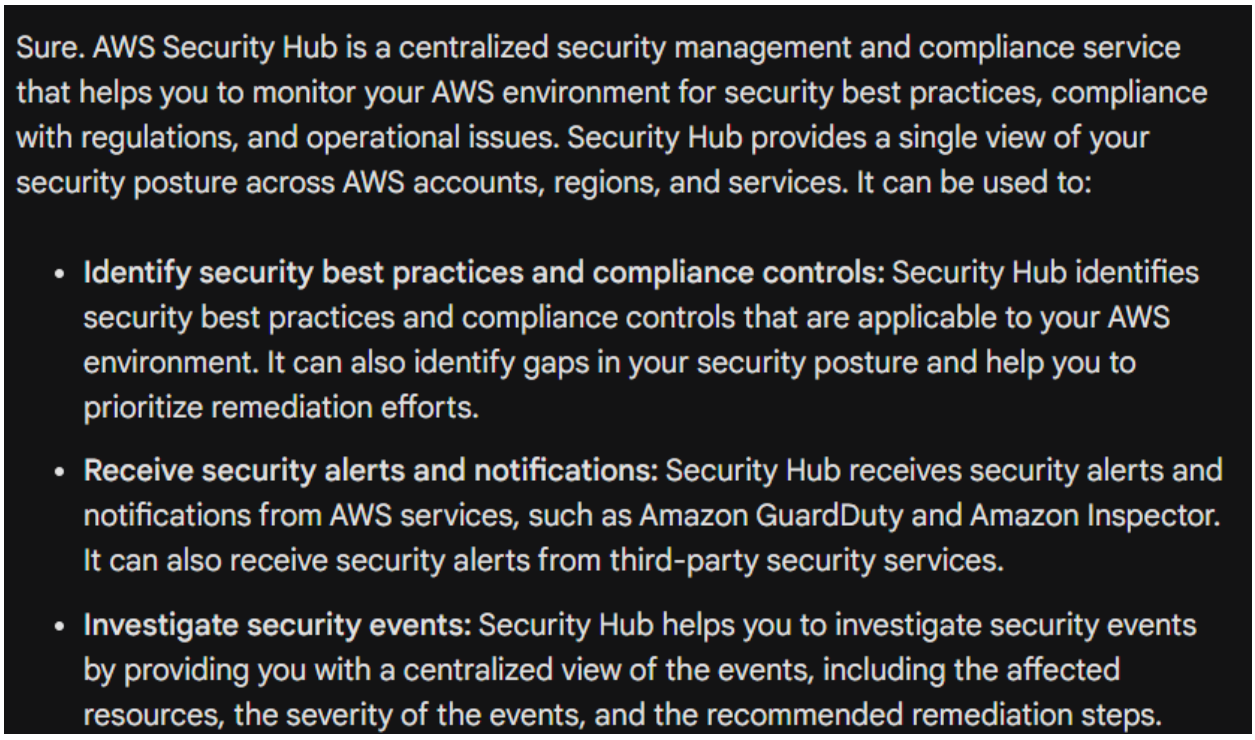
   - **Discover sensitive data:** Macie uses machine learning to discover sensitive data in your AWS environment, including personal identifiable information (PII), financial data, and intellectual property.
   - **Classify sensitive data:** Macie classifies sensitive data according to a variety of classification schemes, such as industry standards and regulatory requirements.
   - **Protect sensitive data:** Macie can be used to protect sensitive data by encrypting it, tagging it, and setting access controls.
   - **Audit your AWS environment:** Macie can be used to audit your AWS environment to ensure that your sensitive data is properly classified and protected.

5. Here is an example of how you can use Amazon Macie to protect sensitive data:

   - **Let's say you have a database that contains PII data. You can use Macie to discover the database and classify the PII data. Macie can then be used to encrypt the PII data and set access controls to restrict access to the data.**
   - **Macie can also be used to audit the database to ensure that the PII data is properly classified and protected. For example, Macie can be used to check if the PII data is encrypted and if access to the data is restricted to authorized users.**

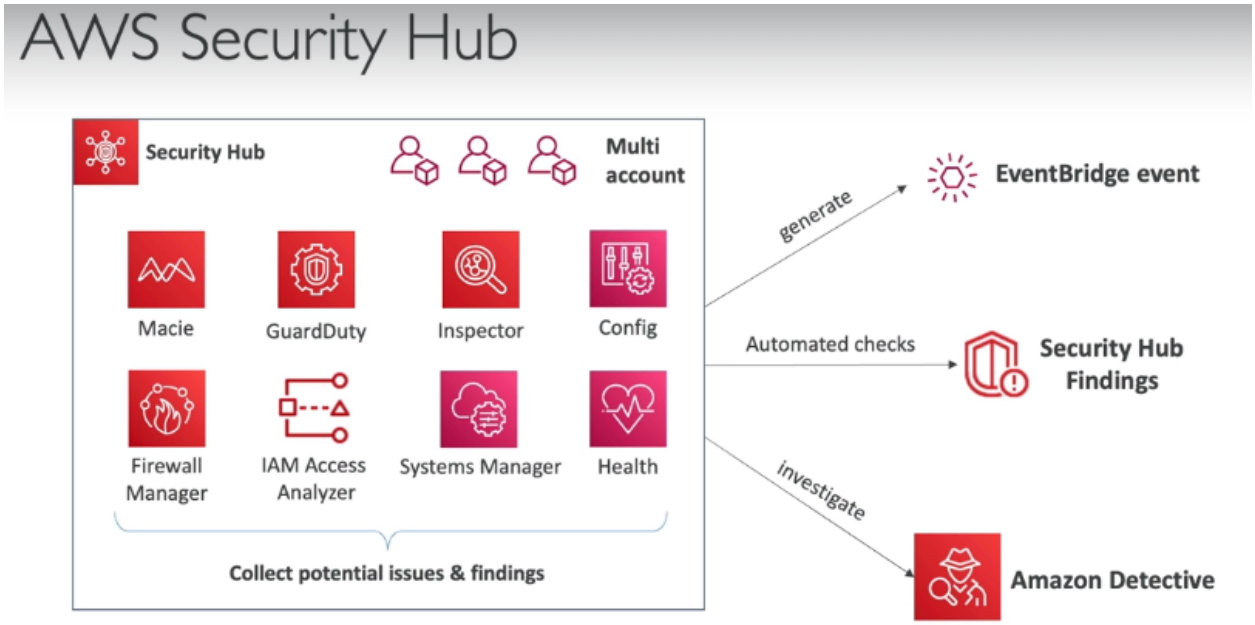
AWS Security Hub

1. 

2. 

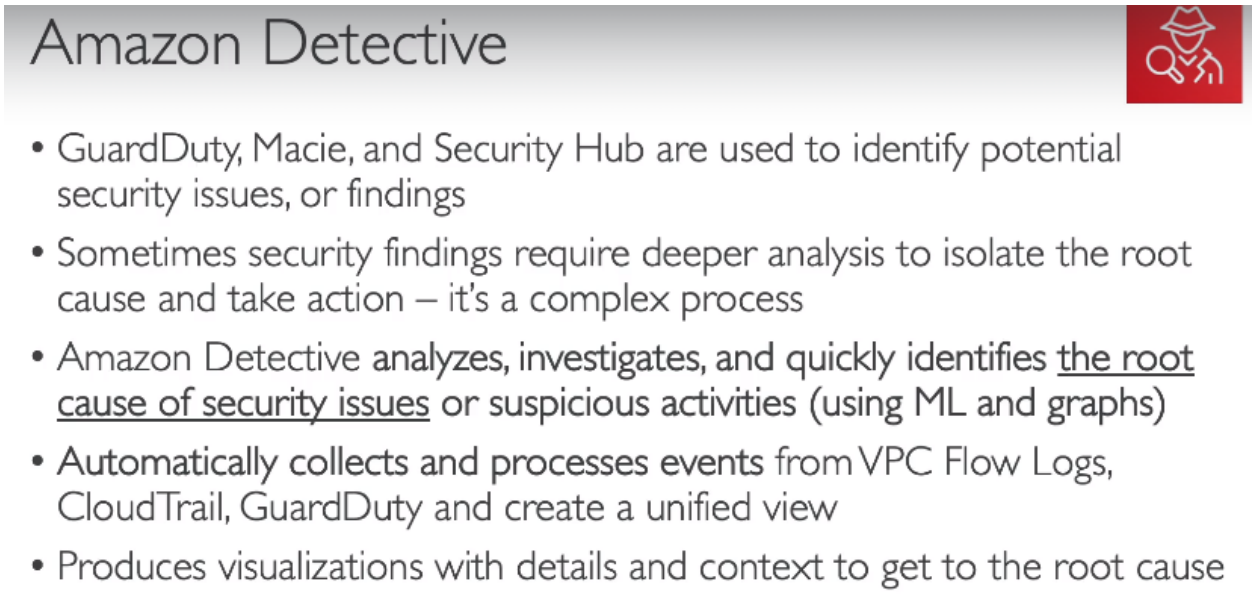
3. • Must first enable the AWS Config Service
4. And so for Security Hub to work, first of all, you must enable the AWS Config service.

5.

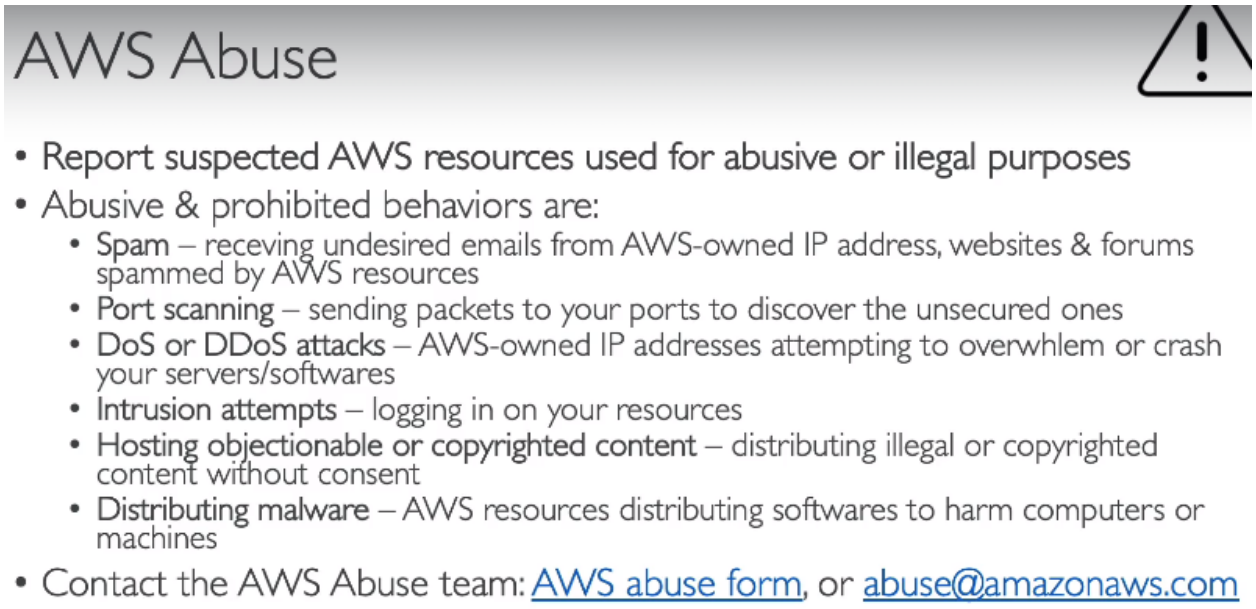


Amazon Detective

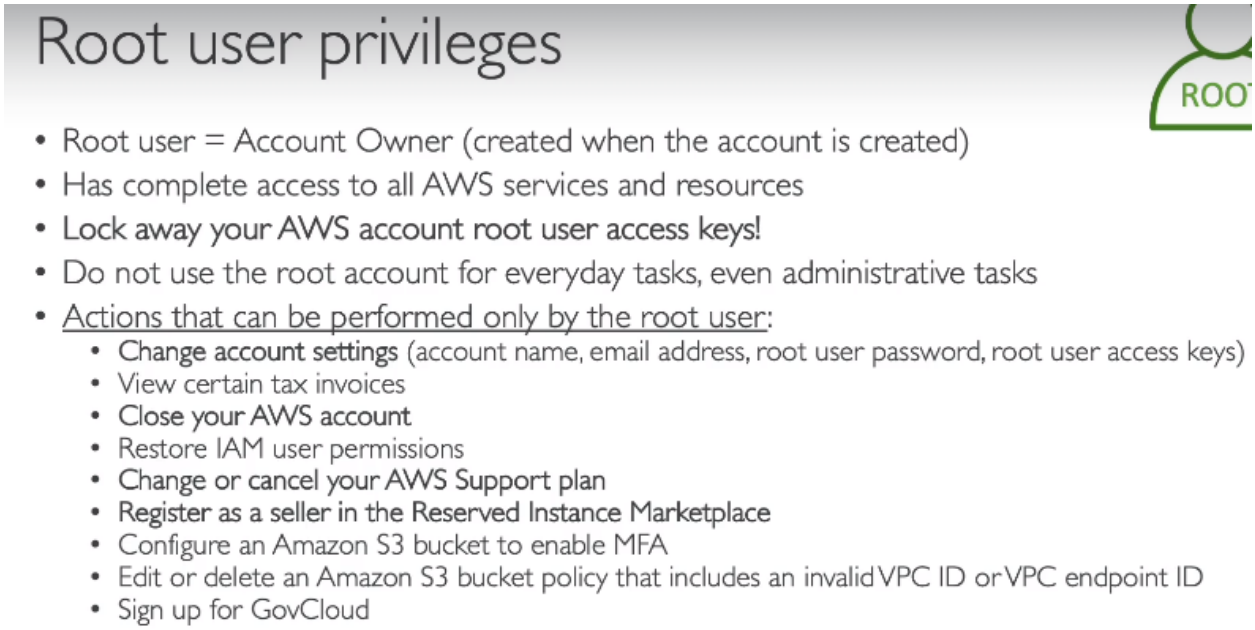
1.



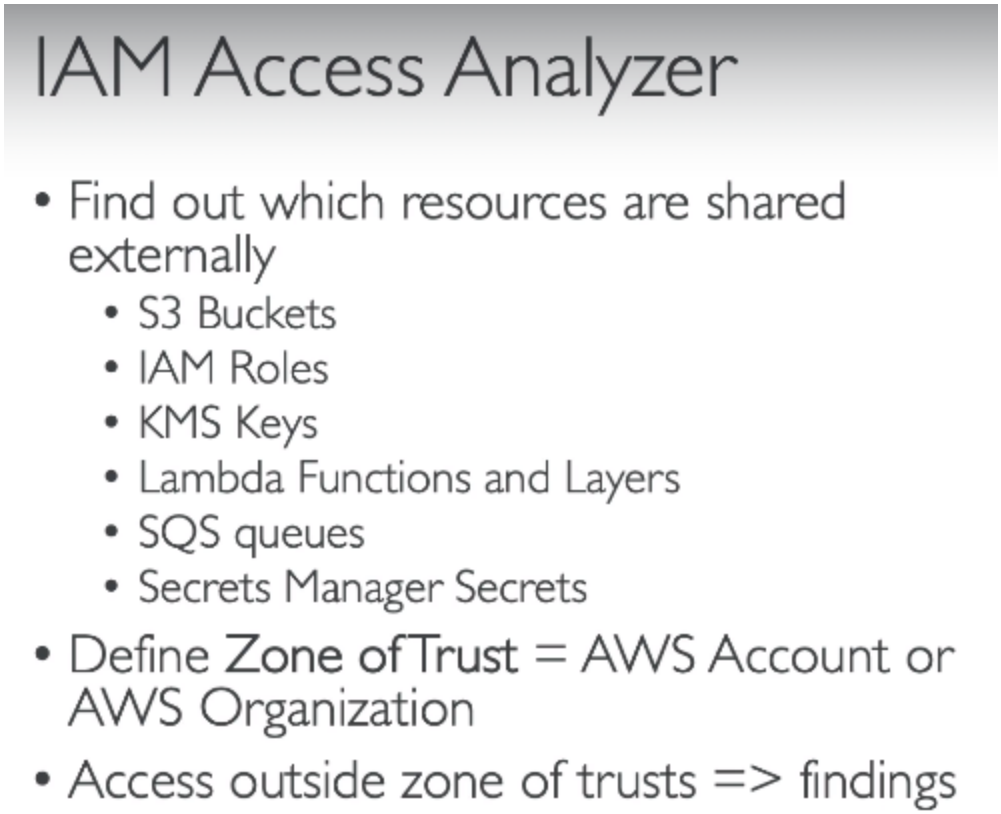
AWS Abuse

1. 

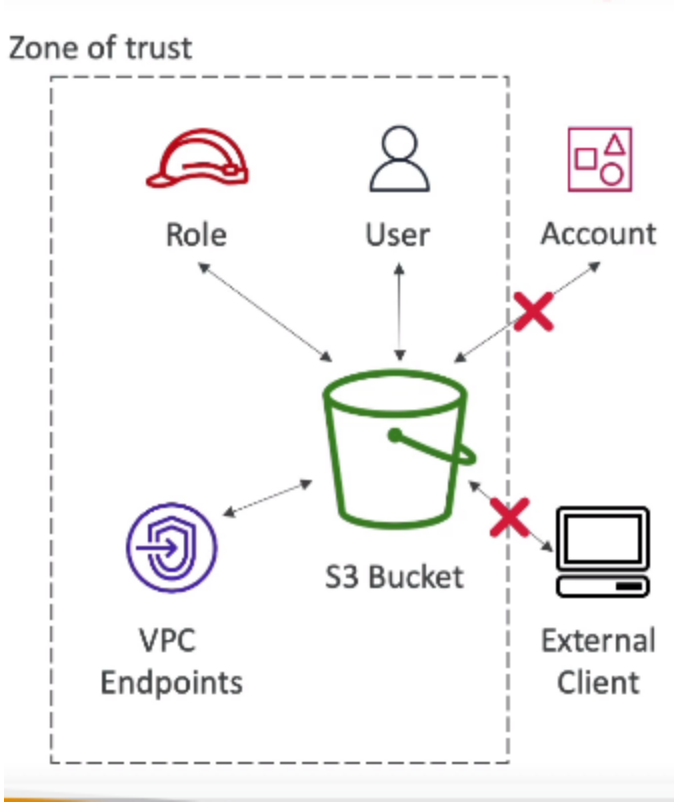
Root user privileges

1. 

IAM Access Analyzer



1.
2. and then anything outside your zone of trust that has access to the resources said above are going to be reported as findings.



3.
4. Here the account and an external client can access the s3 bucket…and if Account and external client will try to access it..then access analyzer will notify us

Summary

1.

Section Summary: Security & Compliance

- **Shared Responsibility on AWS**
- **Shield:** Automatic DDoS Protection + 24/7 support for advanced
- **WAF:** Firewall to filter incoming requests based on rules
- **KMS:** Encryption keys managed by AWS
- **CloudHSM:** Hardware encryption, we manage encryption keys
- **AWS Certificate Manager:** provision, manage, and deploy SSL/TLS Certificates
- **Artifact:** Get access to compliance reports such as PCI, ISO, etc…
- **GuardDuty:** Find malicious behavior with VPC, DNS & CloudTrail Logs
- **Inspector:** find software vulnerabilities in EC2, ECR Images, and Lambda functions
- **Network Firewall:** Protect VPC against network attacks

2.

Section Summary: Security & Compliance

- **Config:** Track config changes and compliance against rules
- **Macie:** Find sensitive data (ex: PII data) in Amazon S3 buckets
- **CloudTrail:** Track API calls made by users within account
- **AWS Security Hub:** gather security findings from multiple AWS accounts
- **Amazon Detective:** find the root cause of security issues or suspicious activities
- **AWS Abuse:** Report AWS resources used for abusive or illegal purposes
- **Root user privileges:**
  - Change account settings
  - Close your AWS account
  - Change or cancel your AWS Support plan
  - Register as a seller in the Reserved Instance Marketplace
- **IAM Access Analyzer:** identify which resources are shared externally