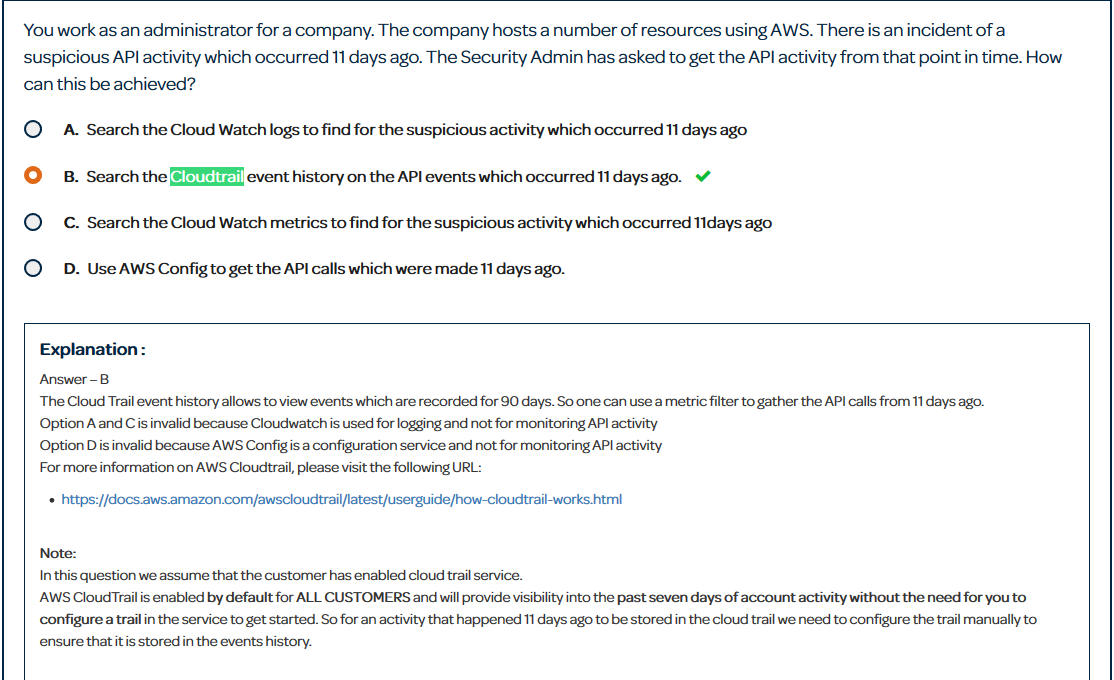
AWS Logging and Monitoring tools

Before I begin let me first explain how I will setup this document. First I will introduce you to different logging and monitoring tools that are important for the AWS Security Specialty Certification. Telling you what they do and why you might use them. I would also for most of them show sample questions of how they might come up on the exam. Then I will compare a few important and similar tools that may trip you up on the exam. I will lastly point out key features of each that might also help you like are the logs real time or not? Also what role if any do they need to work?

**CloudTrail:**

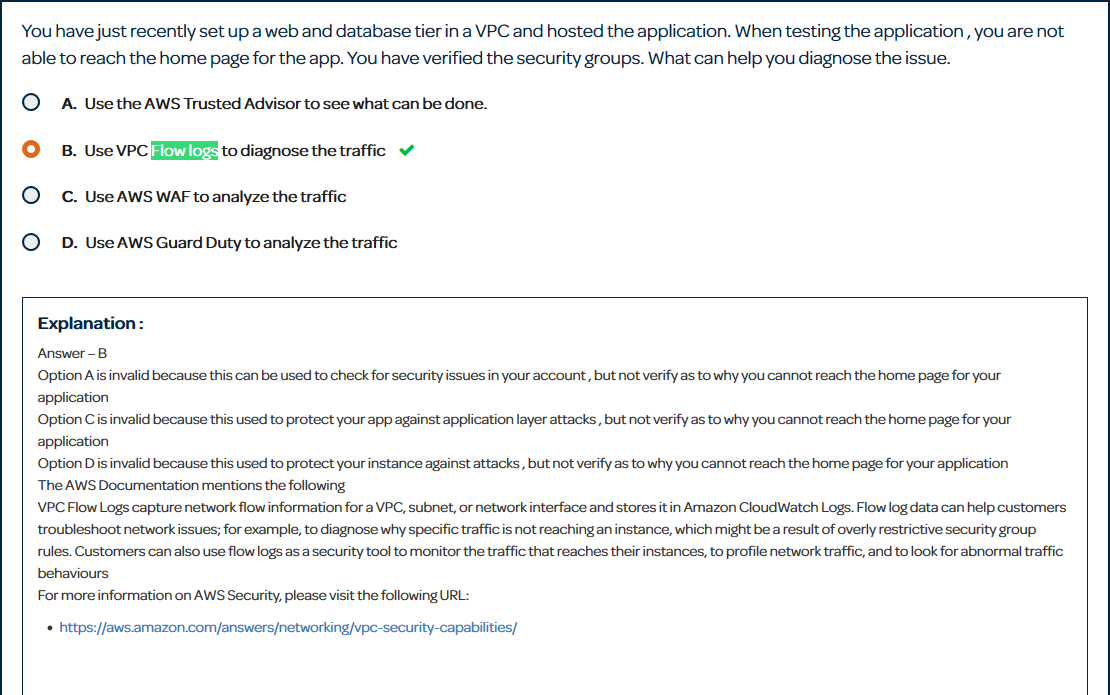
CloudTrail logging in AWS is a way to log API calls made to your account via the console, SDKs, and CLI. By default you have an event history that will logs API calls for up to 90 days, but it is always a better idea to turn it on. When you create a trail in your account the logs will get stored into an S3 bucket. You can also send the logs to CloudWatch logs if you want to run metric filters on them. With CloudTrail turned on you get PCI compliance and API history investigation.



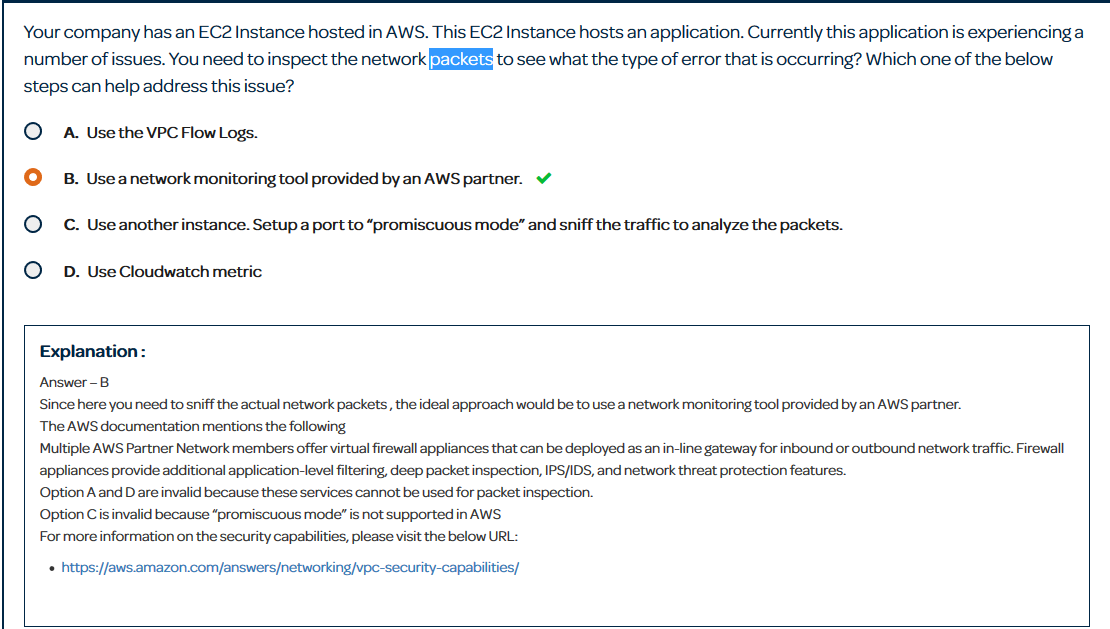
For CloudTrail questions the key word to look for is API. Remember that logs are stored in a bucket indefinitely by default. This allows for after the fact incident investigations.

**VPC Flow Logs:**

VPC Flow Logs monitor IP Meta data traffic in and out of your VPC. Keep in mind it will see things like source IP and port the traffic is coming from. If you need to inspect the actual traffic on an instance then VPC Flow Logs are not for you. These logs are stored in CloudWatch Logs by default.



With questions about monitoring web traffic in and out of a VPC/Subnet/ENI you should immediately think of VPC Flow Logs. Remember that if you need to inspect the actually network packets you should look for something on the AWS marketplace as there is no AWS tool at this moment that has that functionality.



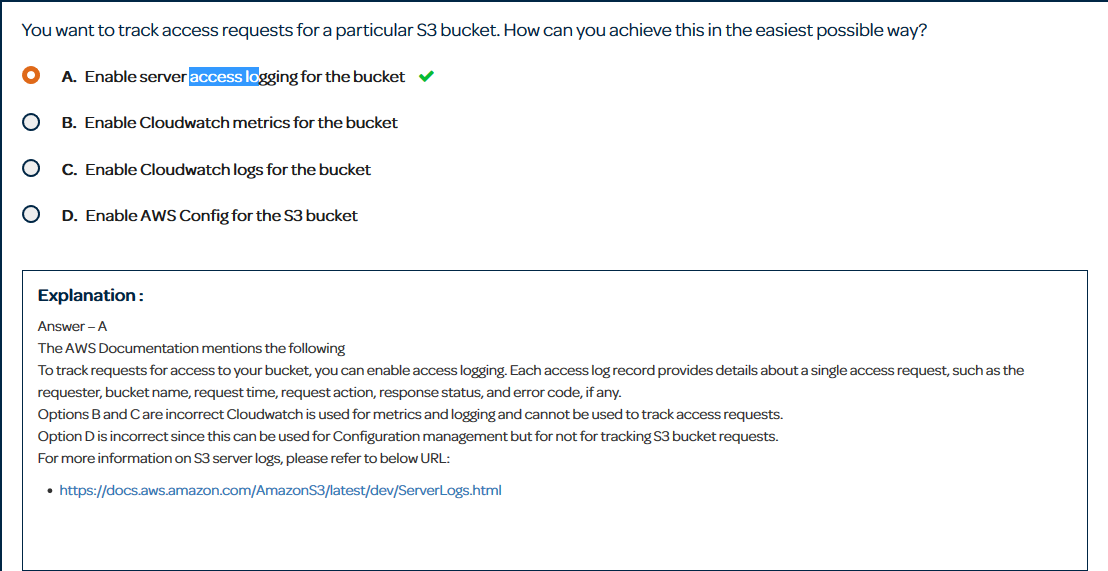
I will also quickly throw in that “promiscuous mode” is not supported in AWS.

**DNS Query Logging:**

DNS Query Logging is used to determine DNS errors in your applications. These logs are stored in CloudWatch Logs. It logs Route 53 name servers. I have not ran across any questions about Query Logging, but the Linux Academy guy doing the video stressed more about when you can use Query Logging. The requirements are stated as the name server must be Route 53 and it cannot be a private hosted zone. I have a documentation in this same folder with more about query logging and how to set it up if you want more info.

**S3 Access Logging:**

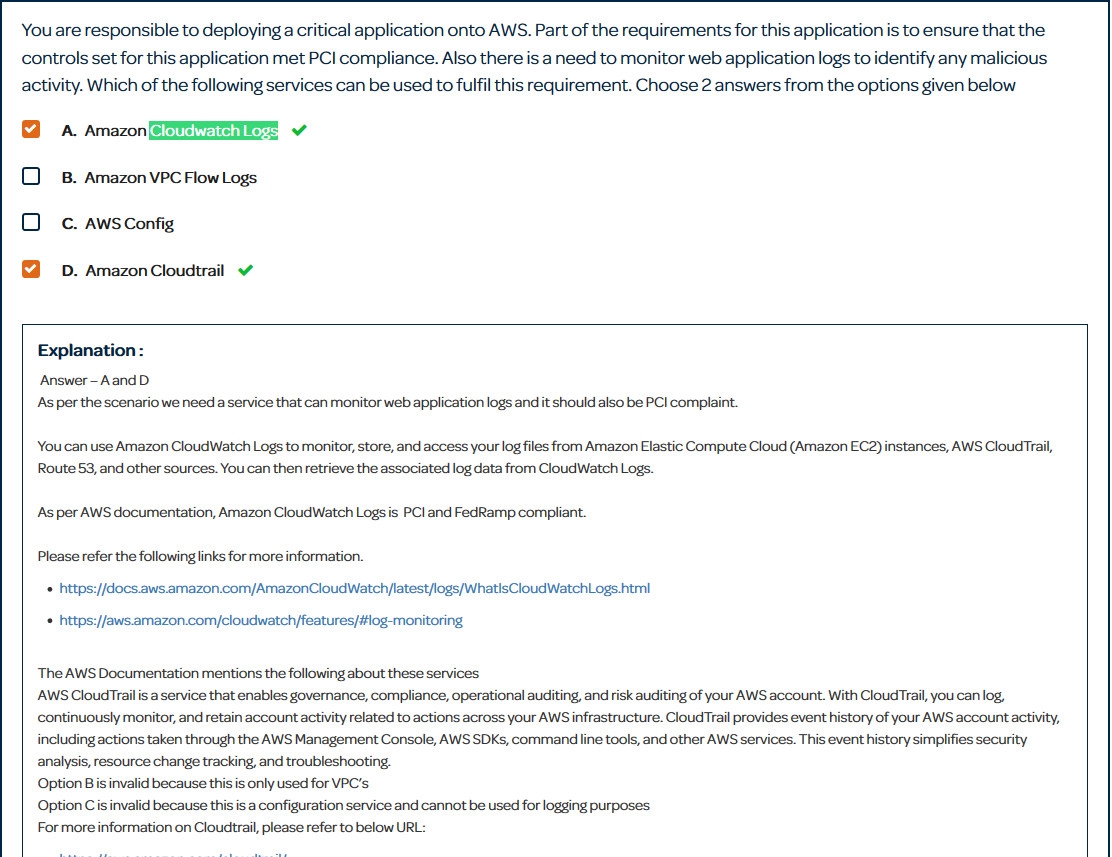
S3 Access Logging is used to log access to an S3 bucket. Logs for this are stored in a separate S3 bucket in the same region. This was mainly used to track access to a bucket hosting a static website.



The biggest thing with access logging to remember is that it is not near real time. It actually is the slowest logging tool in AWS usually taking an hour to send logs. If you want more info about access logging I have a documentation with more notes and how to set it up in this same folder.

**EC2 Application and System Logs:**

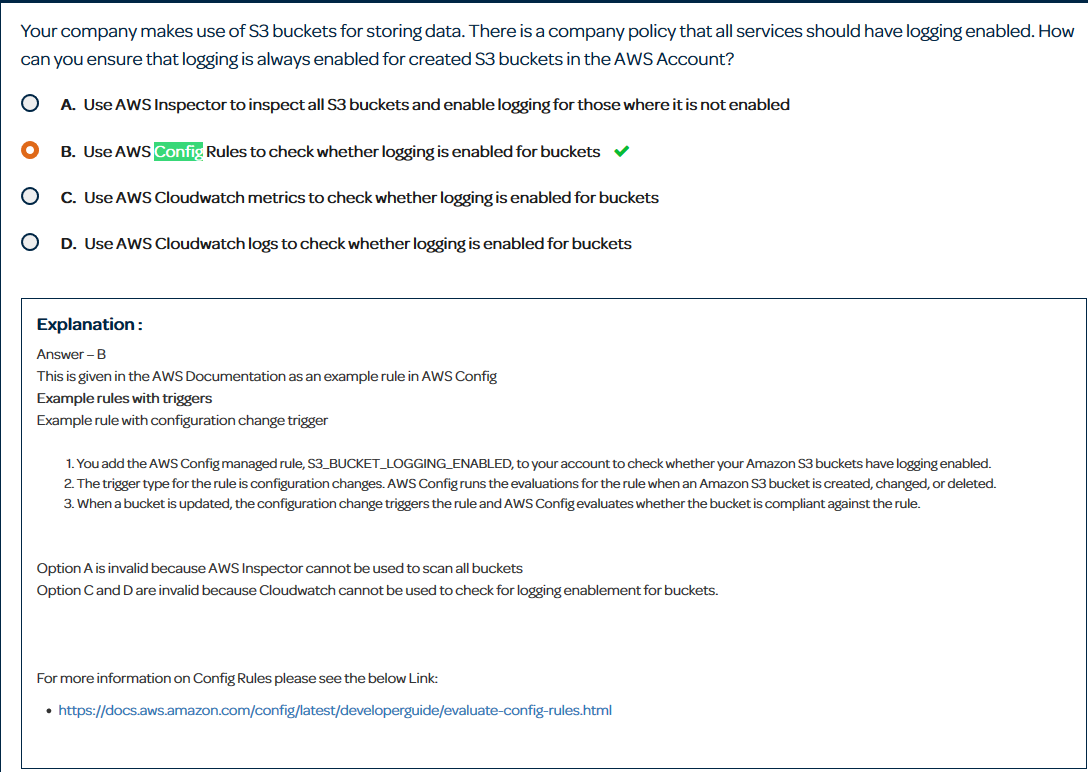
EC2 app and sys logs are brought up to CloudWatch Logs. This is done by installing the CloudWatch Logs agent and giving EC2 an appropriate role to send logs up. You define the types of logs you want to send to CloudWatch. This can range from apache logs, to memory and disk-use percentages.



Key things to remember for choosing CloudWatch Logs. Web application logs are to be assumed on EC2 and there is no other way to logs EC2 instances without using a 3rd party tool. Also look out for system logs on EC2 instances as that is another time to use CloudWatch Logs. One last thing with CloudWatch Logs is almost any AWS service can send logs to here. Most logging services besides CloudTrail send logs here by default.

**AWS Config:**

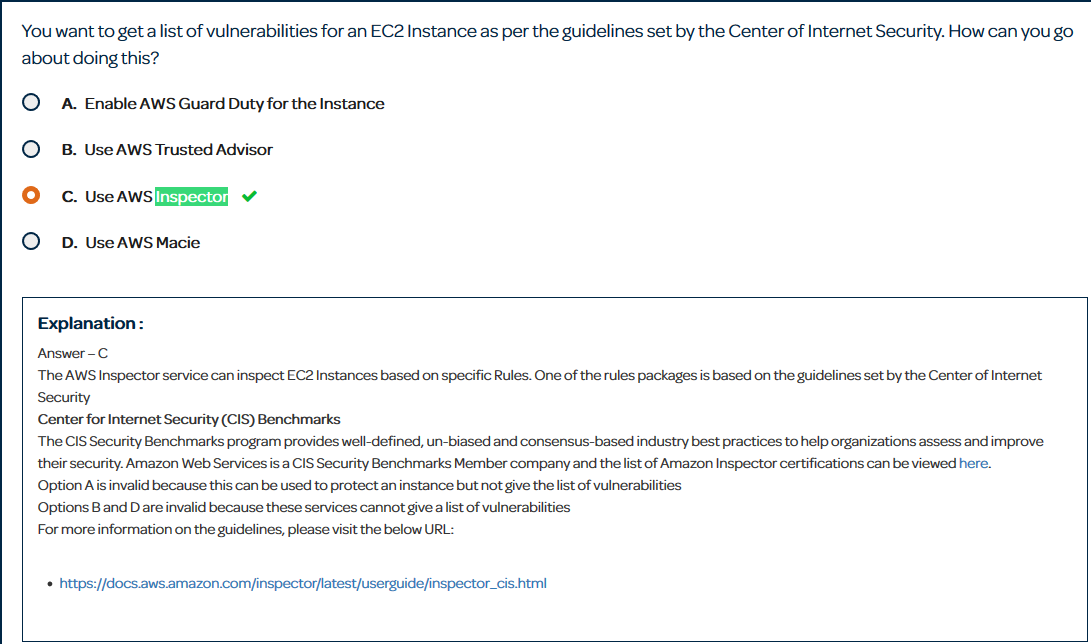
AWS Config is a detailed view of your AWS resources configuration. From the moment you turn it on it gives you a full history of configurations in your account. This comes in handy for auditing and compliance. You can also define configuration rules which act as a compliance check in your AWS account.



With config look for AWS configuration policies. In this example you always want to make sure S3 has the logging enabled. Config will react if this is turned off.

**AWS Inspector:**

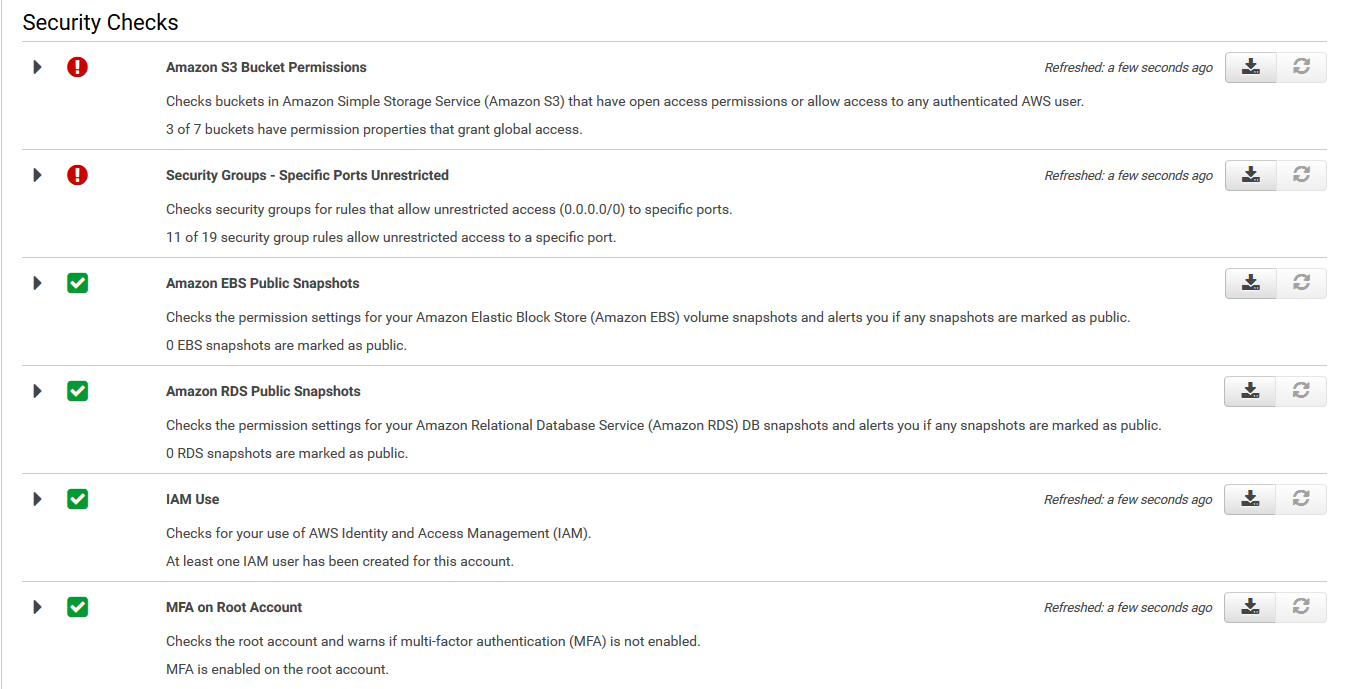
Inspector is a tool similar to AWS Config. The main difference is that while Config looks at your AWS account, Inspector looks on your EC2 instances for security vulnerabilities. You define assessment templates and rule packages for inspector to run on your instances.



Inspector questions are arguably the easiest questions on the exams in my opinion. The main key you are looking for is checking for vulnerabilities on EC2 instances. They are the service that checks EC2 instances. It also helps to know all the rules packages and what they do.

**AWS Trusted Advisor:**

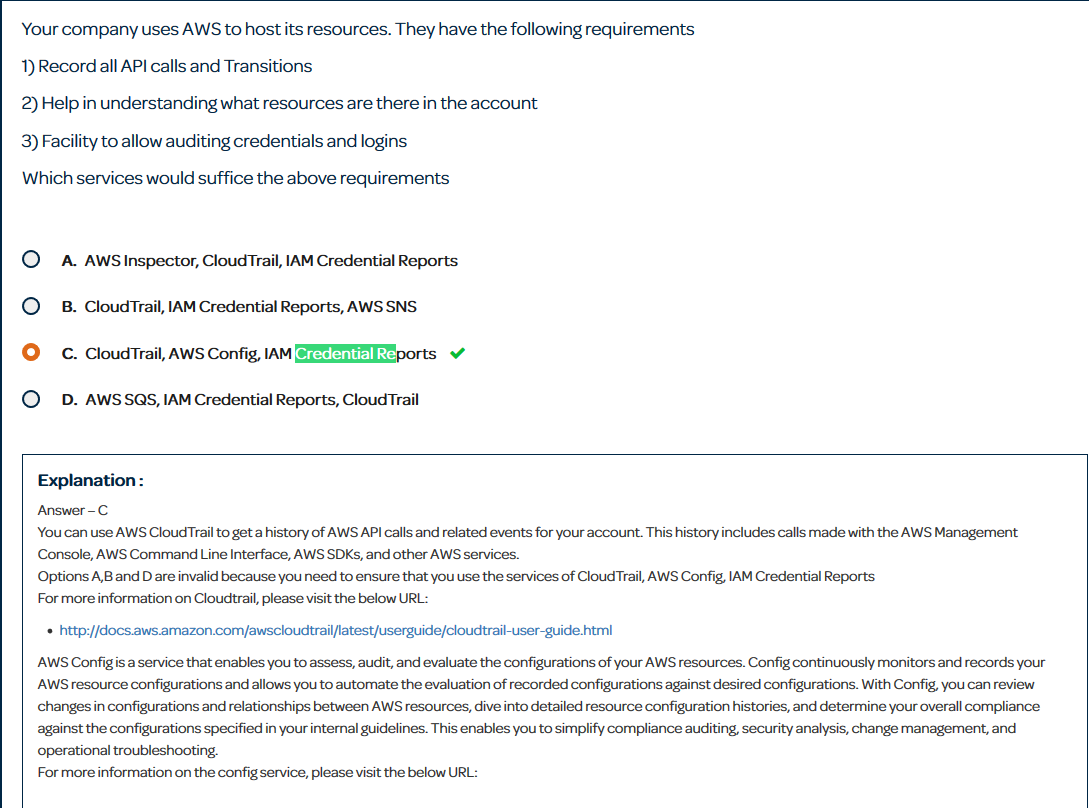
AWS Trusted Advisor is more of a management tool for AWS. It basically advises you on how to improve your environment. Most of it is locked away unless you buy a business or enterprise support plan, but the security portion has a lot of free functionalities to it.



Those are all the free security check you get. The main one to focus on is the security group’s one. I am going to have 1 section of this paper dedicated to the differences between Config, Inspector, and Trusted Advisor because they have similar functionalities. There will also be example question of when to use each in that section.

**IAM Credential Reports:**

The IAM Credential Report allows you to list all users in your account and their various credentials. Credentials that you can list include passwords, access keys, and MFA devices. The use case would be to assist in auditing and compliance with a specific emphasis on the lifecycle requirements of your passwords and access keys. Auditors can check when you rotated each of these last.

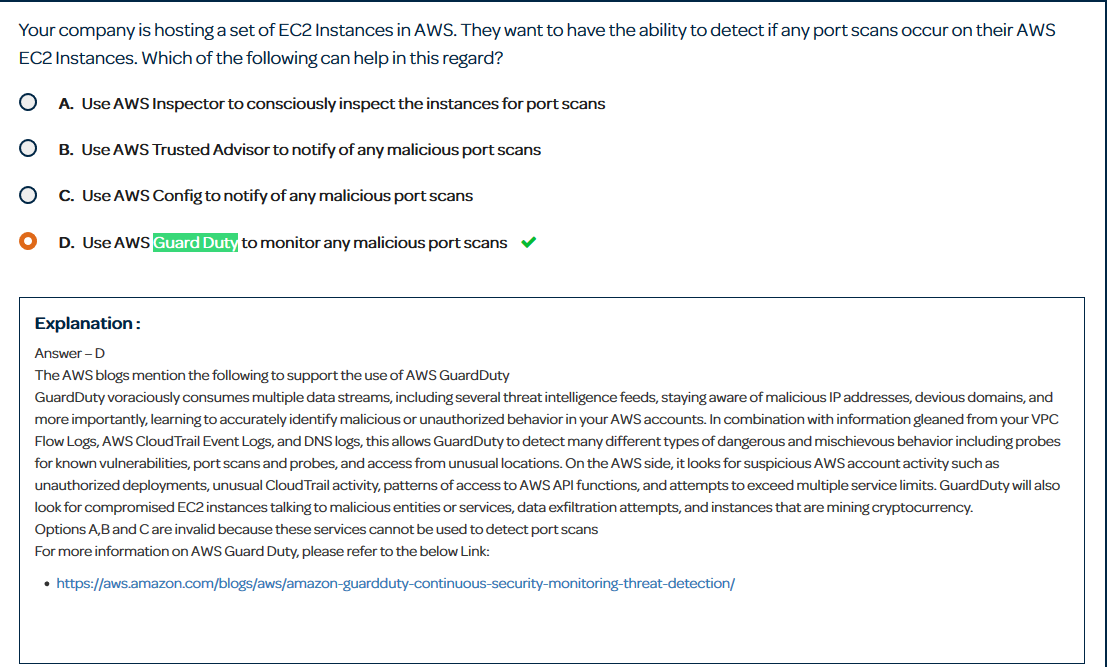


**AWS Access Advisory:**

The access advisory reports about the last time that an IAM entity (user or role) attempted to access a service.

**AWS Guard Duty:**

AWS Guard Duty is a continuous security monitoring service that analyzes and processes the following data sources: VPC Flow Logs, CloudTrail Event Logs, DNS Logs. It uses threat intelligent feeds and machine learning to identify unexpected and potentially unauthorized/malicious activity within your AWS environment. This may come up in your exam, but neither Linux Academy nor ACloudGuru has a video on it in the security course. The only reason I bring it up is because of this question in a whiz labs quiz.



It’s always safe to know more about AWS going into an exam than less. Guard Duty may pop up in an answer bank so look to use it to monitor logging sources for malicious port scans.

**Amazon Macie:**

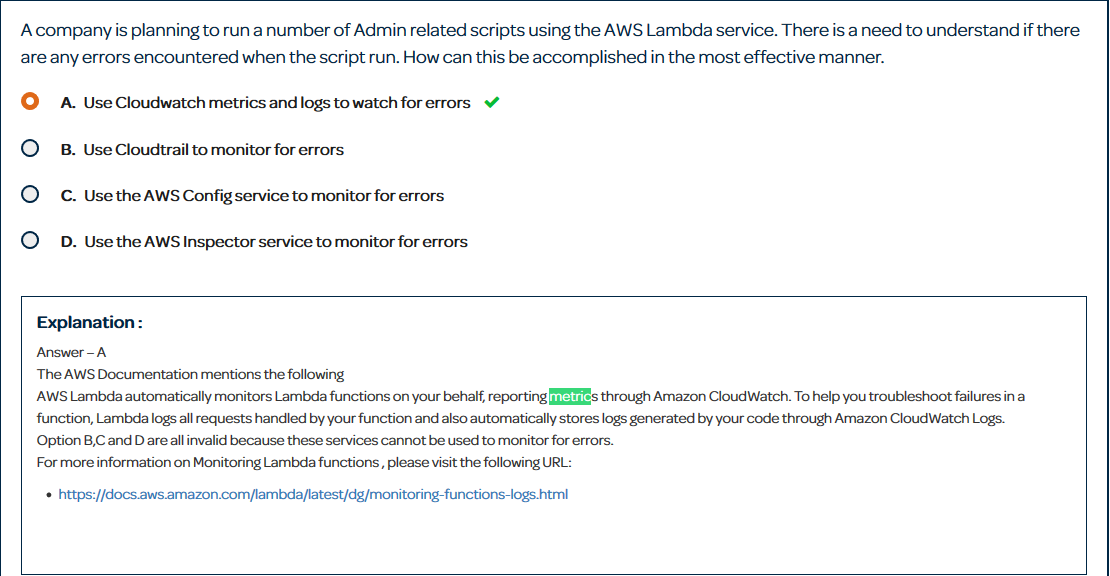
Amazon Macie is a security service that uses machine learning to automatically discover, classify, and protect sensitive data. There is no questions on that so just know what it is at a high level.

**Time for logs:**

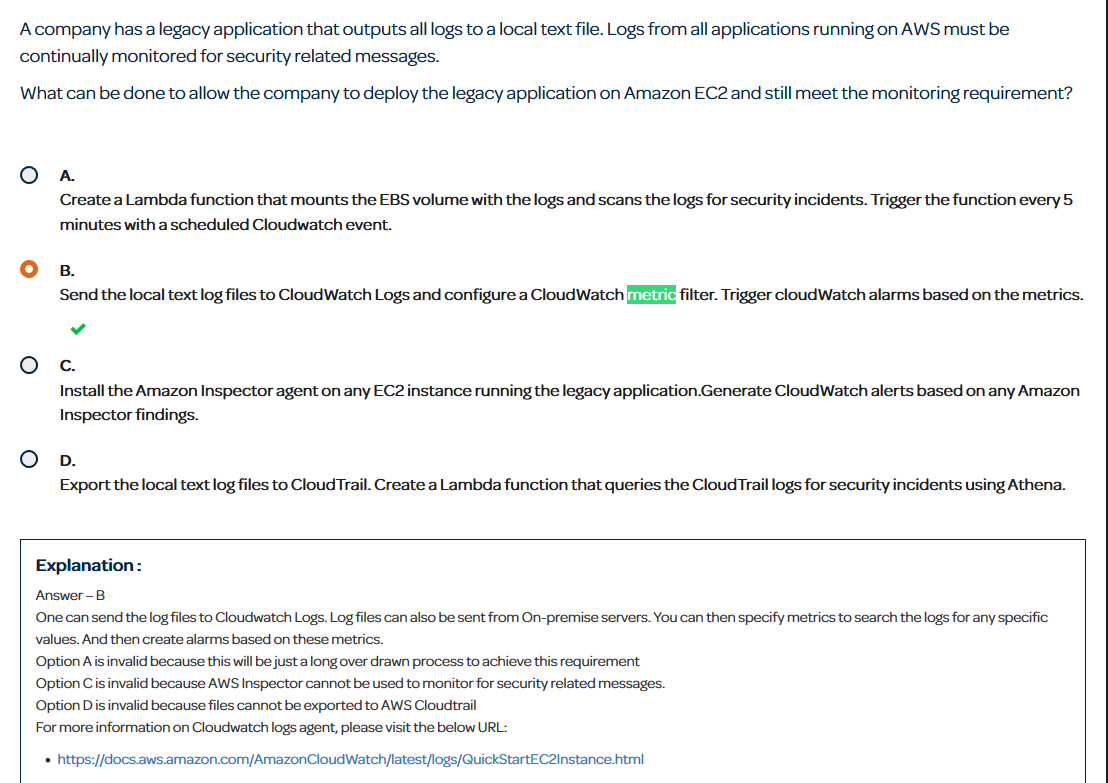
The contents of this paragraph will explain whether the logs are near-real-time or not near-real-time. Most of this information will help you with answering automation questions. CloudTrail is the fastest logging tool. It produces log files about every 5 minutes. Every CloudWatch Logs integration are delivered every 15 minutes. This includes VPC Flow Logs, DNS query logs, CloudTrail logs (remember that CloudTrail by default sends to S3, but you can also configure them to send to CloudWatch Logs as well), and EC2 app or sys logs. S3 Access Logs take the longest taking about an hour to propagate.

**Automation and why time is important:**

In AWS there are 3 automation techniques which are CloudWatch Events, Metric Filters, and S3 Events. Often time’s questions on the exam will ask which type of logging tool should you use and which type of automation you should use. CloudWatch Events are real time and best used off of API calls often triggering a Lambda function to respond to the event. Metric Filters are broader and because they are based off of CloudWatch Logs take up to 15 minutes to trigger. Metric filters also contain only 3 responses and you are usually going to use email with them. So already if you need real time you use CloudWatch Logs, but sometimes services do not integrate with CloudWatch Logs. For example EC2 instances logging data into CloudWatch would be more suitable for a metric filter of the sorts. S3 Events are real time and only work in S3. They have 3 targets which are SNS, SQS, and Lambda functions. When dealing with S3, S3 events as long as they trigger one of the above 3 are always something to consider. S3 events did not appear on my test and are not in any whiz lab questions. Let me point out how the other 2 might come up.

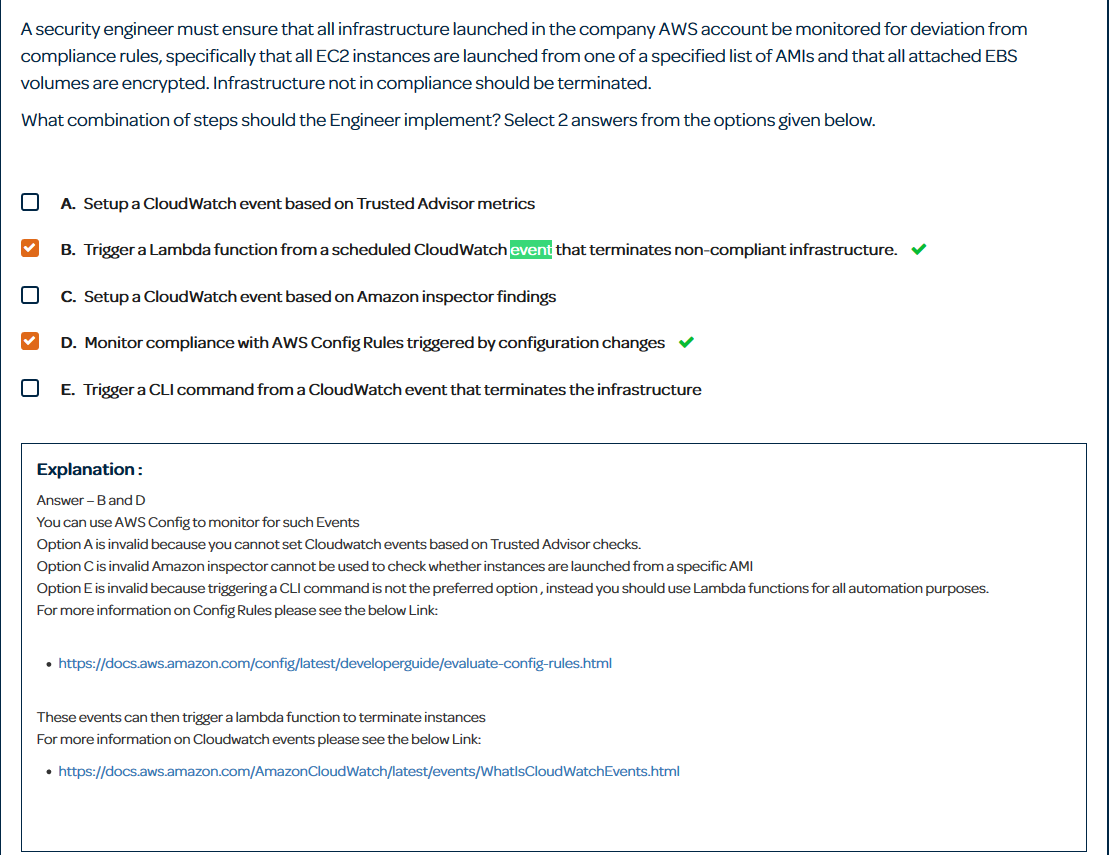


First with Metric Filters remember that they are more suitable based on different AWS service logs. Lambda logs, to CloudWatch logs and if there is any error with Lambda it does a FailedInvocation call whenever it is unable to run. This is a good metric filter to put for all your lambda functions.

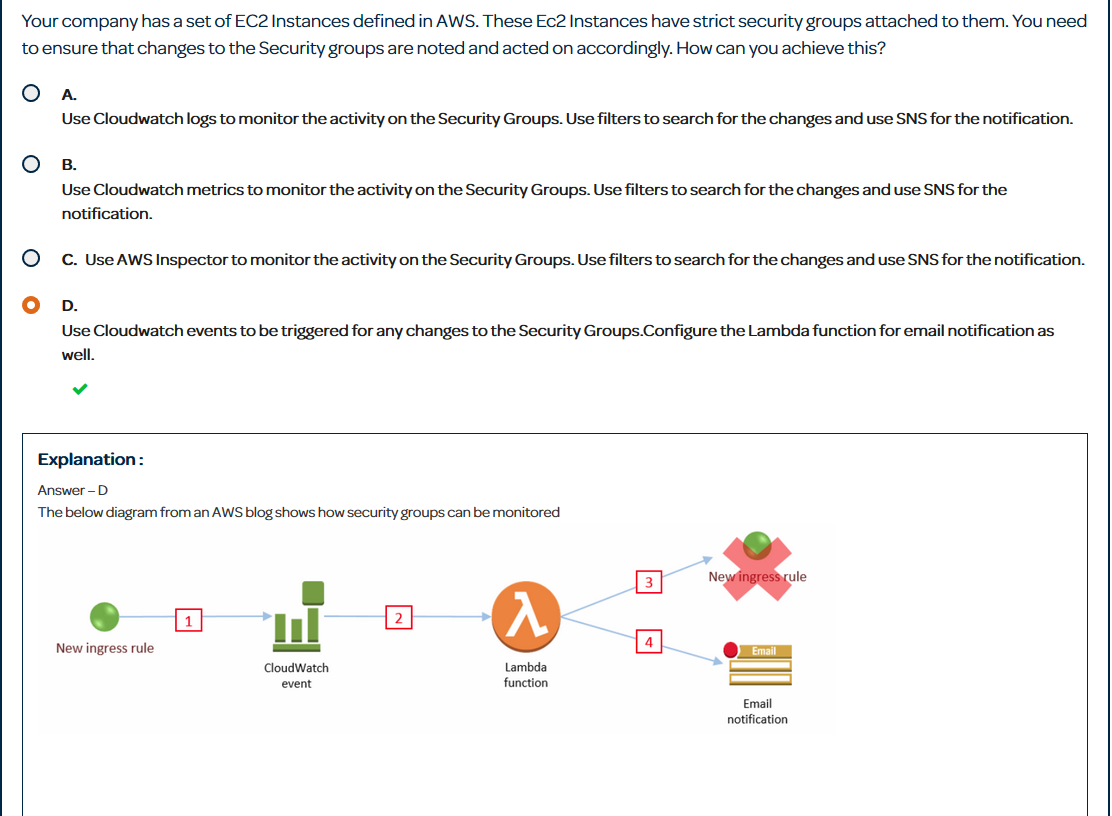


For this question remember that if you are sending data up to CloudWatch from EC2, then using Metric Filters is probably the best option.

Now let’s look at some questions that leads to CloudWatch Events being the correct answer.

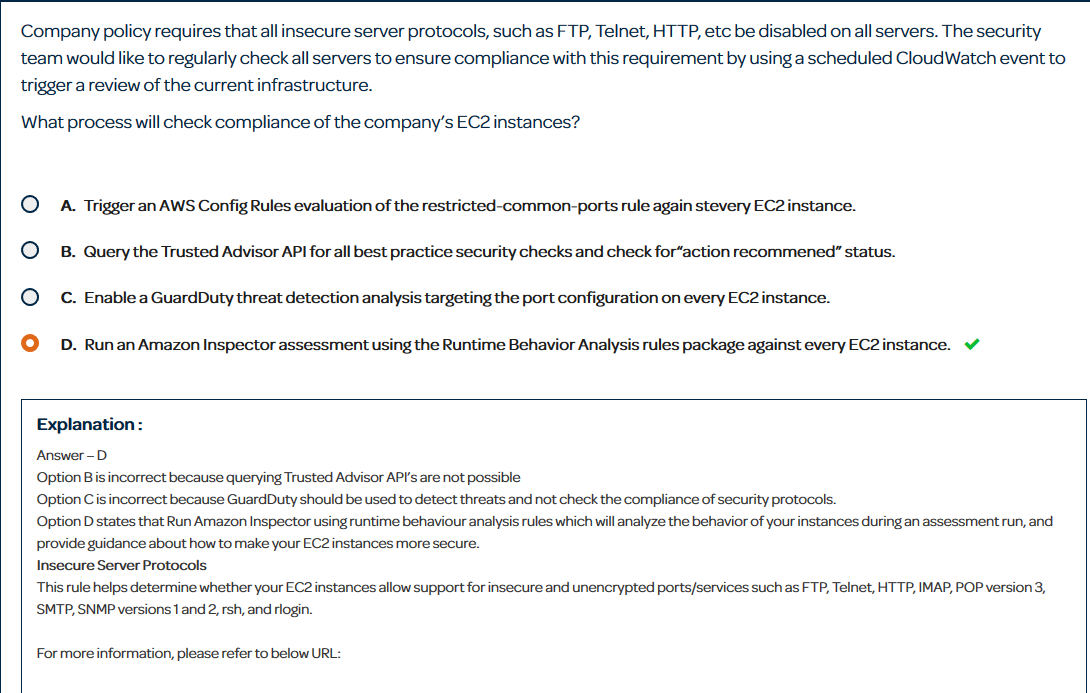


In this scenario CloudWatch events is being triggered by AWS Config and no complaint infrastructure. Already a metric filter cannot be used as Config does not store logs in CloudWatch. Also notice that they want you to react to the non-complaint infrastructure automatically. They suggest you do that by implementing Lambda to terminate non-compliant infrastructure.

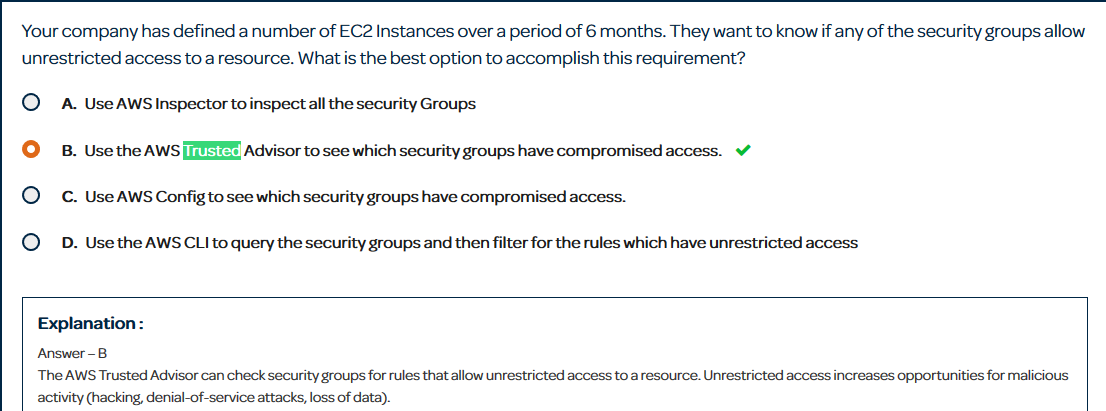


This one is a better example. It states that it wants to be notified which fits metric filters, but it also states that they want it to be acted accordingly on. This is a functionality of CloudWatch Events.

The last thing I want to better explain is something I find sometimes confusing. When checking for ports which should I use? AWS Config, Inspector, or Trusted Advisor. It took me awhile to fully understand this, with that being said lets eliminate the easy one… Inspector.



This question is a question in which it helps to know what the Runtime Behavior Analysis rules package does for inspector. It does exactly what the question wants. It checks your instances for insecure ports and protocols. Just look for protocols and look if the question is asking to examine instances.



This question is very similar to the one above. The main difference is that it asks to check the security groups. This functionality is not found in inspector. AWS trusted advisor has basic security checks one of which being common ports being opened to the world.

I could not find the question I was looking for but let me explain it. With AWS Config you can have Config rules. These rules define how you want your environment setup and if it does not match than it triggers an alarm. There are custom rules that you code and managed rules provided by AWS. There is a specific managed rule in which you specify security ports and rules in which you don’t allow. So if a question needs more specific ports other than just looking at common admin ports you should think about using config.