Your organization has a few million text documents in S3 that are stored in a somewhat random manner, and the amount of files is always growing. The developer that initially wrote the system in use stored everything with a random file name with some attempt at security through obscurity. Now your CEO and CFO both need to be able to search the contents of these documents, and they want to be able to do so quickly at a reasonable cost. What managed AWS services can assist with implementing a solution for your CEO and CFO, and what would the setup process involve?

1. Implement MongoDB
2. Create a search domain
3. Set up access policies
4. Configure your index
5. Implement ElasticSearch
6. Implement Amazon CloudSearch
7. Create a search index
8. Set up IAM roles
9. Configure your baseline

Answer: b, c, d, f

EXPLANATION:

CloudSearch by itself is enough to fulfill the requirements put forward here. CloudSearch is managed, scalable can very quick to configure and get online. In comparison it would take some time to set up EC2 and install ElasticSearch or any other search tool, and would be much more difficult to scale. This involves creating a search domain and configuring the index as required, and then setting up the access policies.

Your organisation utilises Kinesis Data Streams to ingest large amounts of streaming data. Each of the Kinesis Streams is consumed by a Java application running on multiple EC2 instances, utilising the KCL library. The CloudWatch logs for the KCL application normally only show 'INFO' entries, but you have noticed that some errors have started to appear recently, and you need to investigate. The errors are all of the form; "getShard - Cannot find the shard given the shardId" and include the ID of shard which still appears to exist. What may have caused these errors to appear in the logs?

1. These errors are seen when in Enhanced Fan-Out mode and more than 20 consumers are registered per stream.
2. These errors are caused by the shard iterator expiring before your code is ready to utilise it.
3. These errors are generated when migrating the Record Processor from version 1.x to version 2.x of the KCL.
4. These errors can be seen in the Kinesis Consumer log usually after a re-sharding event and are generated by the KinesisProxy.

Answer: d

EXPLANATION:

This error usually appears just after a re-sharding event has completed when the consumer code is using an older version of KCL. This appears to be caused when the KinesisProxy has a cache of shards which is not updated when leases change in the KCL DynamoDB table. Although this error appears to have no effect on the data within the shards, it can usually be resolved by restarting the KCL consumers.

Your application uses Kinesis Data Streams to process incoming streaming data. A colleague has noticed that GetRecords.IteratorAgeMilliseconds metric increases up to 10 minutes when traffic hits its peak during the day, and then reduces again when traffic gets less during the night. This is a problem as it means data processing is being delayed during the times when the most amount of data is being ingested. What changes could you make to reduce the time lag in the stream?

1. Increase parallelism by adding more shards per stream.
2. Rewrite code to handle all exceptions within processRecords.
3. Increase the amount of KPL producers putting data onto the shards.
4. Increase the retention time of the stream from 24 hours to 7 days.
5. Add more EC2 KCL consumers to allow each to process less shards per instance.

Answer: a, e

EXPLANATION:

An increased GetRecords.IteratorAgeMilliseconds metric means that either the KCL consumers cannot keep up processing the data from the Kinesis stream or there aren't enough shards in the stream. Choosing both of these options will satisfy the needs of the question.

Your company produces IoT enabled refrigerators and uses Kinesis Data Streams as part of its method to capture the streaming data coming back from all of the installed devices. Your team has written code using KPL to process, sanitise and enrich the data and then put it onto a Kinesis Stream. If the data cannot be added onto the stream after five tries, it pushes the data through a Kinesis Firehose and into an S3 bucket. During the last week they have started to see the following exception appearing: "Slow down. Service - AmazonKinesisFirehose; Status Code - 500; Error Code - ServiceUnavailableException". Why would you be seeing this error? Choose from the options below.

1. Verify that your Kinesis Data Firehose delivery stream is located in the same Region as your other services, as some services can only send messages to Firehose located in the same Region.
2. Firehose does not have the correct permissions to write to the S3 bucket. Change the permissions allocated to the Role and try again.
3. This message is shown when throughput limits for the delivery stream have been exceeded, they may be increased by contacting AWS Support.
4. This message means the data delivery to Firehose is stale. Check the 'DataFreshness' metric under the Monitoring tab to ensure that it is not increasing over time.

Answer: c

EXPLANATION:

Whenever a 'Slow down', 'Status Code 500' or 'ServiceUnavailableException' message is received from Kinesis Firehose, it's safe to assume that the error relates to a limit being reached. There are a number of limits which could affect why data isn't being successfully sent into Firehose, and you should examine 'Amazon Kinesis Data Firehose Limits' to see which case matches your issue and request an increase from AWS Support.

Your team has been planning to move functionality from an on-premises solution into AWS. New microservices will be defined using CloudFormation templates, but much of the legacy infrastructure is already defined in Puppet Manifests and they do not want this effort to go to waste. They have decided to deploy the Puppet-based elements using AWS OpsWorks but have received the following errors during configuration; "Not authorized to perform sts:AssumeRole" and also "The following resource(s) failed to create [EC2Instance]". The team have been unable to resolve these issues and have asked for your help. Identify the reasons why these errors occur from the options below.

1. You cannot use OpsWorks for managing Puppet based infrastructure. OpsWorks only operates and integrates with Chef.
2. Ensure that the 'AWSOpsWorksCMServerRole' policy is attached to the instance profile role.
3. Deploy in a VPC which is set to non-default tenancy with an instance type that supports dedicated tenancy.
4. Ensure that the EC2 instance has the AWS service agent is running, has outbound Internet access and has DNS resolution enabled.
5. Ensure there are no errors within the Puppet Manifests. Fix the syntax errors and restart the Puppet Master.

Answer: b, d

EXPLANATION:

There are two answers which would resolve the errors in the question. Any time a "not authorized" message is displayed, it is nearly always a permissions problem and in this case it can be resolved by attaching the AWSOpsWorksCMServiceRole policy to the instance profile role for EC2. opsworks-cm.amazonaws.com should also be listed in the Trust Relationships. For the second question, this error normally dictates that the EC2 instance doesn’t have sufficient network access, so we need to ensure that the instance has outbound Internet access, and that the VPC has a single subnet with DNS resolution and Auto-assign Public IP settings enabled. All other options will not resolve the errors.

You have an idea regarding your AWS account security. You would like to monitor your account for any possible attacks against your resources, such as port scans or brute force SSH and RDP attacks. If anything is detected, you want the report pushed to a Slack channel where anyone in your company can monitor and take action if it's their responsibility. How do you go about implemeting this?

1. Implement AWS Inspector. On detected events, trigger a Lambda function to post the information to your Slack channel.
2. Implement a Lambda function to monitor your VPC Flow Logs. For any odd requests, post the information to your Slack channel.
3. Implement Amazon GuardDuty. On detected events, trigger a Lambda function to post the information to your Slack channel.
4. Implement a Lambda function to monitor your CloudTrail file. For any odd API calls or requests, post the information to your Slack channel

Answer: c

EXPLANATION:

Amazon GuardDuty is the best way to implement this. It can trigger Lambda functions on events which can be used to post to a Slack channel.

Your company sells and supports a number of Internet-enabled printers and scanners. Every time these devices have an issue, the data is sent back to a global endpoint which ingests it and places it into a Kinesis stream for processing. Your sales team have have been meeting all of their targets and devices are sending more and more data back into your infrastructure. You have therefore made the decision to scale out the Kinesis stream. Recently a new Cloud Engineer has joined the Operations Team and he has been given the task of re-sharding. The task was completed successfully, but soon after he notices the following error in the logs; "Cannot get the shard for this ProcessTask, so duplicate KPL user records in the event of re-sharding will not be dropped during de-aggregation of Amazon Kinesis records". The Engineer has asked you what the error is and what they should do to resolve it. Chose your response from the following options.

1. To resolve these errors, re-shard the Kinesis stream to the the number of shards that were originally set before the errors occurred.
2. To resolve these errors, ensure the producers use multiple threads to write to the Kinesis Data Streams service at the same time.
3. To resolve these errors, either stop and start the KCL application or clear the lease entries from the DynamoDB lease table.
4. To resolve these errors, ensure that records use the the multi-record operation PutRecords, or are aggregated into a larger file before using the single-record operation PutRecord.

Answer: c

EXPLANATION:

These errors are generated from code that handles KPL messages. When the KCL consumers are restarted or the lease entries are cleared, the ListShards operation will be run which will resolve the KPL messages. None of the other resolutions listed above will resolve these messages.

You've spent weeks building a production Linux AMI that your AutoScaling group is using. You realise that there's one configuration change you need to perform on all new servers that are created due to scaling. You don't have time to build a new AMI until next month due to other work requirements. What's the fastest way to implement the change?

1. Switch to OpsWorks for Chef automate, write a cookbook which will implement your change and apply it to all servers and autoscaling groups.
2. Create a Lambda function to monitor CloudTrail EC2 NewInstance events. If one is detected SSH to the server and make the change.
3. Use a lifecycle hook to make the change during the creation of the new instance.
4. Create a cronjob to ping your subnet looking for new instances. If one is detected run a second bash script which will scp the configuration to the new server.

Answer: c

EXPLANATION:

A lifecycle hook will be the fastest way to implement the change until you are able to create a new AMI.

Your company develops an online shopping platform and would like to implement a way to recommend products to customers based on previous products they have looked at. To use this you want to record their clickstream, or the sequence of links they've clicked on as they navigate your website. At any one time, you can have thousands of users using your shopping platform. Which architecture will meet your requirements?

1. Ingest data with Kinesis Data Streams. Group user requests with Kinesis Data Analytics. Process and store the data with Lambda, Kinesis Firehose and S3.
2. Ingest data with Kinesis Data Analytics. Group user requests with Kinesis Data Firehose. Process and store the data with Lambda, Kinesis Streams and S3.
3. Ingest data with Kinesis Firehose. Group user requests with Kinesis Data Data Streams Groups. Process and store the data with EC2, Kinesis Streams and S3.
4. Ingest data with Kinesis Data Streams. Group user requests with Kinesis Data Analytics. Process and store the data with EC2, Kinesis Firehose and S3.

Answer: a

EXPLANATION:

Ingesting is done with Kinesis Data Streams, grouping user requests into sessions is done with Data Analytics. Data processing and storage is done with Lambda, Firehose and S3. Read the attached link and the high-level solution overview for more information.

Your CEO has heard how an ELK stack would improve your monitoring, troubleshooting and ability to secure your AWS environment. Before letting you explain anything about it, he demands you get one up and running as soon as possible using whatever AWS services you need to use. How do you go about it?

1. Install Elasticsearch, Logstash and Kibana on an EC2 instance.
2. Use the Amazon Elasticsearch Service.
3. Install CloudSearch, Logstash and Kibana on an EC2 instance.
4. Use CloudSearch, CloudWatch Logs and CloudKibana managed services to create your ELK stack.

Answer: b

EXPLANATION:

The Amazon Elasticsearch service will give you managed Elasticsearch, Logstash and Kibana without the requirement of installing, maintaning or scaling any of them and their associated infrastructure.

Your organisation is 75% through moving its core services from a Data centre and into AWS. The AWS stacks have been working well in their new environment but you have been told that the Data centre contract will expire in 3 months and therefore there is not enough time to re-implement the remaining 25% of services before this date. As they are already managed by Chef, you decide to move them into AWS and manage them using OpsWorks for Chef. However, when configuring OpsWorks you have noticed the following errors have appeared; "Not Authorized to perform sts:AssumeRole" and "FATAL Could not find pivotal in users or clients!". Choose the correct options to resolve the errors.

1. Create a new service role and attach the AWSOpsWorksCMServiceRole policy to the role. Verify that the service role is associated with the Chef server and it has that policy attached.
2. Ensure you have at least one existing EIP address free by deleting unused addresses or by asking AWS Support for an increase.
3. Install the knife-opc command and then run the command; knife opc org user add default pivotal
4. Ensure that the EC2 instance has the AWS service agent running and has outbound Internet access with DNS resolution enabled.

Answer: a, c

EXPLANATION:

With the "Not Authorized to perform sts:AssumeRole" error, you can assume its policy/role related and therefore creating a role and attaching the AWSOpsWorksCMServiceRole policy should resolve this issue. Finally, any message which states that you 'cannot find a pivotal user', requires you to add one to the default location. All other answers will not resolve the problems listed.

You currently work for a local government department which has cameras installed at all intersections with traffic lights around the city. The aim is to monitor traffic, reduce congestion if possible and detect any traffic accidents. There will be some effort required to meet these requirements, as lots of video feeds will have to be monitored. You're thinking about implementing an application that will user Amazon Rekognition Video, which is a Deep learning video analysis service, to meet this monitoring requirement. However before you begin looking into Rekognition, which other AWS service is a key component of this application?

1. Amazon Kinesis Data Streams
2. Amazon Kinesis Video Streams
3. Amazon Kinesis Camera Streams
4. Amazon Kinesis Video Analytics

Answer: b

EXPLANATION:

Amazon Kinesis Video Streams makes it easy to capture, process and store video streams which can then be used with Amazon Rekognition Video.

Your team has been moving various legacy code projects over to AWS in the past few months. Every application has been analysed to see whether it is best hosted on an EC2 instance or as a Lambda function. One particular software project has been rewritten to store an object in an S3 bucket, and then this action triggers a Lambda function to compress the file as a new object in the same bucket, and then delete the original. Unfortunately, due to the way you have deployed the solution the Lambda function appears to be constantly invoking itself and is stuck in a continuous loop. How do you temporarily stop this from happening whilst you investigate?

1. Choose the 'Throttle' option on the function configuration page and then locate and resolve the error which caused the recursive invocation.
2. Remove the IAM User or Role associated with the Lambda function to cease operation. The Lambda function will stop and allow you to troubleshoot.
3. You can constrain the memory or processing power available to the function. The Lambda function will stop and allow you to troubleshoot.
4. There is no way to stop a Lambda function once in an infinite loop. You need to delete the function, re-create it and re-deploy the code.

Answer: a

EXPLANATION:

When architecting a solution, always ensure that you do not generate a recursive loop. This is when something in one part of the infrastructure will trigger something elsewhere, and in turn trigger the first part once again. In particular these scenarios are problematic when one AWS service triggers another AWS service and this in turn triggers the first. In the example, this can only be rectified by enabling the Throttle option, which sets the reserved concurrency to zero and will throttle all future invocations of this function. This action should only be used in case of emergencies.

You are building acatguru, which your organization describes as facebook for cats. As part of the signup process your users need to upload a full size profile image. You already have these photos being stored in S3, however you would like to also create thumbnails of the same image, which will be used throughout the site. How will you automate this process using AWS resources?

1. Create an S3 event trigger to execute a Lambda function when an object is created. The function will create the thumbnail from the source image and store it in a different S3 bucket.
2. Create an S3 bucket notification trigger to execute a Lambda function when an object is created. The function will create the thumbnail from the source image and store it in a different S3 bucket.
3. Configure S3 to publish its event stream to an SNS topic. Subscribe a Lambda function to the SNS topic which will trigger when a file is uploaded. The fuction will create the thumbnail from the source image and store it in a different S3 bucket.
4. Use CloudTrail to monitor PUT and POST calls sent to S3 and trigger a Lambda function when you identify an upload. The function will create the thumbnail from the source image and store it in a different S3 bucket.

Answer: a

EXPLANATION:

S3 triggering Lambda to create thumbnails is a perfect example of how to automate this process

You are discussing error scenarios and possible retry strategies for your Step Functions machine with your colleague. Which of her claims are incorrect?

1. A Retrier must contain the 'ErrorEquals' field which is a non-empty array of strings that match Error Names, e.g. 'States.Timeout'. When a state reports an error, Step Functions scans through the Retriers and, when the Error Name appears in this array, it implements the retry policy described in this Retrier.
2. When a state reports an error, the default course of action for AWS Step Functions is to log the error and perform a single retry after 1 second. If that doesn't succeed, AWS Step Functions will fail the execution entirely.
3. A Retry field with an 'IntervalSeconds' and 'MaxAttempts' value of 3 and 'BackoffRate' value of 1.5 will make three retry attempts after waits of 3, 4.5 and 6.75 seconds.
4. Any state can encounter runtime errors. Examples are when a Lambda function throws an exception or if a transient network issue exists. AWS Step Functions distinguishes these clearly from Failures such as state machine definition issues that are handled differently.

Answer: b, d

EXPLANATION:

Errors can arise because of state machine definition issues, task failures or because of transient issues. When a state reports an error, the default course of action for AWS Step Functions is to fail the execution entirely.

International Megawidgits Inc. is a manufacturing company which currently uses a DynamoDB table to store information relating to the status of a process. The Operations Team have been asked to design a solution so that whenever a message appears in the DynamoDB table, it informs another process that this has happened. As this process only polls an SQS queue, they have decided to transfer the DynamoDB files to SQS using an AWS Step Function. The team would like you to identify some Best Practices when designing the Step Functions, so that they don't run into any problems in the future. Choose which of the following are recognised Step Functions Best Practices.

1. Avoid Latency When Polling for Activity Tasks
2. Use ARNs Instead of Passing Large Payloads
3. Avoid defining Timeouts in state machine definitions
4. Avoid Reaching the History Limit
5. Ignore Lambda Service Exceptions

Answer: a, b, d

EXPLANATION:

All answers are Step Functions Best Practices, apart from "Ignoring Lambda Service Exceptions" and "Avoiding Defining Timeouts in State Machine Definitions". To avoid latency, separate polling threads and open at least 100 polls per activity ARN. Step functions have a hard 25,000 execution limit so ongoing work should be split across multiple workflow executions to prevent this. Passing large payloads (over 32KB) between states may terminate the execution so always store the data in an S3 bucket and pass the ARN instead of the raw data.

You have been ask to deploy a clustered application on a small number of EC2 instances. The application must be placed across multiple Availability Zones, have high speed, low latency communication between each of the nodes, and should also minimise the chance of underlying hardware failure. Which of the following options would provide this solution?

1. deploy the EC2 servers in a Spread Placement Group
2. The application should deployed as a service in ECS
3. Deploy the EC2 servers in a Cluster Placement Group
4. Create a new VPC with the tenancy type of host and deploy the instances in the VPC

Answer: a

EXPLANATION:

Spread Placement Groups are recommended for applications that have a small number of critical instances which need to be kept separate from each other. Launching instances in a Spread Placement Group reduces the risk of simultaneous failures that might occur when instances share the same underlying hardware. Spread Placement Groups provide access to distinct hardware, and are therefore suitable for mixing instance types or launching instances over time. In this case, deploying the EC2 instances in a Spread Placement Group is the only correct option.

You need to quickly test a proof of concept that your boss has given you. He's given you a zip file containing a php web application. You want to get it running in an AWS environment as fast a possible, however there's also a dependency on a library that must be installed as well. The library is available from a yum/apt repository. Which service do you choose and how do you ensure depedencies are installed?

1. AWS CloudFormation, install dependency with custom resources.
2. AWS OpsWorks for deployment, install dependency with chef.
3. AWS Elastic Beanstalk for deployment, install dependency with ebextensions.
4. AWS EC2 and apache2, install dependency with apt-get or yum.

Answer: c

EXPLANATION:

AWS Elastic Beanstalk is a quick way to test the proof of concept, as webserver configuration is not required. Required Libraries can be installed quickly and automatically using ebextensions.

Which of the following AWS services allow native encryption of data, while at rest?

1. Elastic Block Store (EBS)
2. ElastiCache for Memcached
3. S3
4. Elastic File System (EFS)

Answer: a, c, d

EXPLANATION:

EBS, S3 and EFS all allow the user to configure encryption at rest using either the AWS Key Management Service (KMS) or, in some cases, using customer provided keys. The exception on the list is ElastiCache for Memcached which does not offer a native encryption service, although ElastiCache for Redis does.

A company has created three Docker containers which need to be deployed. The first is a core application for the company which will need to accommodate thousands of Websocket connections every minute. The second is the main corporate Website which is based on a Node.js application running behind an Nginx sidecar. The final container is a small departmental application written by a single developer using the React framework. You have been asked to deploy each container on the most suitable, cost effective and reliable AWS platform from the options below.

1. Deploy the Websocket application on a managed ECS Cluster, the Corporate Website on Fargate and the Departmental application on Elastic Beanstalk.
2. Deploy the Websocket application on a managed ECS Cluster, the corporate Website in Fargate and install Docker on a single EC2 instance to run the departmental application.
3. Deploy all of the Docker containers in Fargate.
4. Deploy the Websocket application and the corporate Website on a managed ECS Cluster and deploy the departmental application in Fargate.
5. Deploy all three applications on a managed ECS Cluster.

Answer: a

EXPLANATION:

In order to make the decision which options are the best, we should start with the deployment that has the most constraints and that is the application using Websockets as these require many network connections and should be installed as multiple tasks across the ECS Cluster. With this constraint dealt with, we can eliminate all options that don't include the ECS Cluster. Putting all containers on one ECS Cluster would work technically, but wouldn't be cost effective and would mean an internal departmental application lives on the same cluster as the main core production applications. When we also exclude installing Docker directly on EC2 as there is no redundancy, this leaves the only option as using ECS for the core application, Fargate for the Website and Elastic Beanstalk for the internal application.

Your organization currently runs all of its applications on in-house virtual machines. Your CEO likes the direction AWS are taking the cloud industry and has suggested you look into what it would take to migrate your servers to the cloud. All of your servers are built and configured automatically using Chef. Which AWS services will you transition your first batch of servers to AWS in the FASTEST possible way while maintaining a centralised configuration management?

1. Use VM Import/Export to upload snapshots of your first batch of servers into AWS. Use OpsWorks for Chef Automate to deploy a new Chef server into your AWS account. Manually deploy the same recipes and cookbooks as you run on-prem, onto your new Chef server.
2. AWS OpsWorks with EC2 VM Import/Export
3. AWS CloudFormation and OpsWorks
4. Leverage the AWS Server Migration Service to migrate your instances across as AMIs into AWS. Use custom configuration script in the Server Migration Service console to register the instances to Chef.

Answer: d

EXPLANATION:

While AWS OpsWorks allows you to use your own custom chef cookbooks with your AWS resources, it is not as easy as simply importing your existing servers into OpsWorks. The fastest solution here is to use the AWS Server Migration Service to bring your existing Chef managed machines into EC2 and to manage them with Chef the same way you have been on your in-house system

In the Amazon States Language, InputPath, Parameters, ResultPath, and OutputPath filter and control the flow of JSON from state to state. Which of the following definitions are correct?

1. Parameters enable you to pass a collection of key-value pairs, where the values are either static values that you define in your state machine definition, or that are selected from the input using a path.
2. ResultPath can filter the JSON output to further limit the information that is passed to the output.
3. OutputPath selects what combination of the state input and the task result to pass to the output.
4. InputPath selects which parts of the JSON input to pass to the task of the Task state.

Answer: a, d

EXPLANATION:

The definition of OutputPath and ResultPath are swapped around, i.e. an OutputPath can filter the JSON output to further limit the information that is passed to the output while the ResultPath selects what combination of the state input and the task result to pass to the output

You're developing a Node.js application that uses the AWS Node.js API. You interact with lots of different AWS services, and would like to determine how long your API requests are taking in an effort to make your application as efficient as possible. It would also be useful to detect any issues that may be arising and give you an idea about how to fix them. Which AWS service can assist in this task and how would you go about achieving it?

1. Use Amazon QuickSight, inspect the service map, trace the request path, determine the bottlenecks.
2. Use Amazon QuickSight, inspect the client map, trace the segment path, determine the bottlenecks.
3. AWS AWS X-Ray, inspect the client map, trace the segment path, determine the bottlenecks.
4. Use AWS X-Ray, inspect the service map, trace the request path, determine the bottlenecks.

Answer: d

EXPLANATION:

AWS X-Ray will produce an end to end view of requests made from your application where you can analyze the requests made as they pass through your application.

You work for a company that uses a serverless architecture to process up to 4,000 events per second for its usage analytics service. At its core, it consists of a multitude of Lambdas that have been given various resource configurations, i.e. memory/CPU settings. For each function invocation, you want to monitor that allocation against the actual memory usage. Select the simplest feasible approach to achieve that.

1. Log entries written into the log group associated with a Lambda function don't include profiling info such as memory usage. You need to use AWS X-Ray for that.
2. You can set up a custom CloudWatch metric filter using a pattern that includes 'REPORT', 'MAX', 'Memory' and 'Used:'.
3. AWS Lambda provides a built-in CloudWatch metric for memory usage.
4. Use the 'Monitoring' tab in the AWS Lambda console and add the 'Resource' monitoring graph to your dashboard.

Answer: b

EXPLANATION:

AWS Lambda doesn’t provide a built-in metric for memory usage but you can set up a CloudWatch metric filter. The AWS Lambda console provides monitoring graphs for Invocations, Duration, Error count and success rate (%), Throttles, IteratorAge and DeadLetterErrors.

Your customers have recently reported that your Java web application stops working sometimes. Your developers have researched the issue and noticed that there appears to be a memory leak which causes the software to eventually crash. They have fixed the issue, but your CEO wants to ensure it never happens again. Which AWS services could help you detect future leaks so you're able to fix the issue before the application crashes?

1. Create a CloudWatch dashboard to monitor the memory usage of your app from a custom metric you are pushing to CloudWatch.
2. Push your memory usage to CloudTrail, have a lambda function monitor it and alert a SNS queue if it crosses a threshold.
3. Push your memory usage to CloudWatch logs, have a lambda function monitor it and alert a SNS queue if it crosses a threshold.
4. Push your memory usage to a custom CloudWatch metric, set it to alert your developers if it crosses a threshold.

Answer: d

EXPLANATION:

Pushing the custom cloudwatch metric is a good idea, you could add it to a dashboard but that wont alert your developers unless they're actively checking it, which you can't rely on.

Your organisation would like to implement autoscaling of your servers, so you can spin up new servers during times of high demand and remove servers during the quiet times. Your application load mainly comes from memory usage, and so you have chosen that as your scaling metric. What do you have to do next?

1. Publish a custom memory utilization metric to CloudWatch, as there isn't one by default.
2. Configure the Auto Scaling Group to use the default metric type 'Average Memory Utilization'
3. Configure the Auto Scaling Group with scaling policy with steps, use the default metric type 'Average Memory Utilization'
4. Configure the Auto Scaling Group with a target tracking scaling policy, use the default metric type 'Average Memory Utilization'

Answer: a

EXPLANATION:

Memory utilization is not a default CloudWatch metric, you will have to publish a custom metric first.

You have three EC2 instances running a core application, which has been performing sub-optimally since yesterday. One of your colleagues said that they remember that the system appeared to perform in a similar way about 18 months ago, but they can't remember what the issue was. You need to perform an indepth investigation of the current issue and you will need to view graphs of that period, with granular metrics. Reading the logs from when the issue originally occurred would also help troubleshooting. Which of the following options would give you the best chance of resolving the issue?

1. View the Cloudwatch logs from 18 months ago and view the Cloudwatch graphs from the same time, setting the granularity at 15 minutes.
2. View the Cloudtrail logs from 18 months ago and view the Cloudwatch graphs from the same time, setting the granularity at 1 minute.
3. View the Cloudwatch logs from 18 months ago and view the Cloudwatch graphs from the same time, setting the granularity at 60 minutes.
4. View the Cloudwatch graphs from 18 months ago, setting the granularity at 1 minute. Unfortunately Cloudwatch logs cannot be kept for that length of time.
5. View the Cloudwatch logs from 18 months ago and use the API to retrieve the datapoints and store in a local Database for analysis.

Answer: e

EXPLANATION:

You can immediately disregard any option with Cloudtrail as these will only contain API logs and will not record application issues. For the remaining options, it's important to note that Cloudwatch Logs are available indefinitely by default, so any option stating that logs can't be kept can also be excluded. Now we have to factor in the length of time in the past we are investigating. 15 months is the maximum amount of time we can retrieve metrics from the Console, this means that we need to retrieve the data from the API and process it locally.

Your company has been moving its core application from a monolith to an ECS based, microservices architecture. Although functionally the application is operational in its new environment, you randomly see spikes of latency throughout the day and you are concerned with the overall performance of the application. How can you rapidly gain information about the requests each microservice is serving?

1. Enable Enhanced Cloudwatch functionality in each microservice, push all performance information into Cloudwatch Logs and use Log Insights to identify poorly performing APIs.
2. Utilise AWS X-Ray by deploying a Docker image containing the X-Ray daemon to ECS, which will then gather segment data. Then examine the traces generated to track the path of a request through the application and use Filter Expressions to find traces relating to specific paths.
3. Utilise the AWS SDK to communicate with the AWS X-Ray API. These will generate the JSON segments into an S3 bucket and the use SQL statements within AWS Athena to search through the traces stored in the bucket.
4. Deploy the X-Ray daemon to all ECS Services and view the traces and performance data that is generated, within the X-Ray console.

Answer: b

EXPLANATION:

X-Ray is the AWS service that can deal with this sort of scenario, therefore we can discount any option that doesn't include it. The best solution would be to deploy a Docker image running the X-Ray agent as an ECS service and this could ingest data from all sources. You could then track the path of the application and filter on this data to find the cause.

Your organization has been using AWS for 12 months. Currently, you store all of your custom metrics in CloudWatch. Per company policy, you must retain your metrics for 3 years before it is OK to discard or delete them. Is CloudWatch suitable?

1. No, CloudWatch only retains metrics for 15 months. You will have to use the API to pull all metrics and store them somewhere else.
2. Yes, CloudWatch will retain custom metrics for 3 years.
3. No, CloudWatch only retains metrics for 24 months. You will have to use the API to pull all metrics and store them somewhere else.
4. Yes, CloudWatch will retain custom metrics for 5 years.

Answer: a

EXPLANATION:

CloudWatch will retain metrics for 15 months, after which they will expire. If you need to keep metrics for longer you must pull them out of CloudWatch using their API and store them in a database somewhere else.

Your organization runs a large amount of workloads in AWS and has automated many aspects of its operation, including logging. As the in-house devops engineer, you've received a ticket asking you to log every time an EC2 instance state changes. Normally you would use CloudWatch events for something like this, but CloudWatch logs aren't a valid target in CloudWatch Events. How will you solve this?

1. Use CloudWatch Events, but use a Lambda function target. Write a Lambda function which will perform the logging for you.
2. Create a Lambda function and run it on a schedule with CloudWatch Events. Make the Lambda function parse your CloudTrail logs for EC2 instance state changes and log them to another CloudWatch Logs log.
3. Parse the EC2 CloudWatch changelog with a Lambda function each minute, and log the results to a separate log for instance state changes
4. Use a CloudWatch dashboard, which will log EC2 state changes to CloudWatch logs if you create a text widget.

Answer: a

EXPLANATION:

CloudWatch Events can use a Lambda function as a target, which will solve this issue.

You have a large amount of infrastructure, and monitoring has been neglected since it was provisioned. You want to monitor your AWS resources, your on-premises resources, applications and services. You would like to be able to retain your system and application logs, graph metrics and be alerted to critical events. Which AWS services and features will assist you in meeting this requirement?

1. Amazon QuickSight for graphing, Amazon CloudWatch Logs for logging, Amazon CloudWatch Alarms for alerts.
2. Amazon CloudTrail for graphing and logging, Amazon CloudWatch Alarms for alerts
3. Amazon CloudWatch Metrics for graphing, Amazon CloudWatch Logs for logging, Amazon CloudWatch Alarms for alerts.
4. Amazon CloudWatch Metrics for graphing, Amazon CloudLog for logging, Amazon CloudWatch Alarms for alerts.

Answer: c

EXPLANATION:

Amazon CloudWatch Metrics is a suitable service for graphing, Amazon CloudWatch Logs will allow you to log both AWS and on-premises resources, and Amazon CloudWatch Alarms will be suitable for alerts and notifications.

Your organization deals with petabytes of data which needs to be shared with a vendor. They require full access to your private S3 buckets to perform their development work. They have extensive AWS experience already. How will you give them access?

1. Grant permission to the vendors AWS ID in the S3 bucket policy
2. Create a cross-account IAM Role
3. Create an IAM Role
4. Grant the role permission to the bucket
5. Edit the bucket policy to allow the vendor AWS ID read access
6. Enable cross region replication
7. Grant permission to vendor AWS ID to use the role

Answer: b, d, g

EXPLANATION:

The best way to accomplish this is to create a cross-account IAM role with permission to access the bucket, and grant the vendor's AWS ID permission to use the role.

Your organization currently runs a hybrid cloud environment with servers in AWS and in a local data center. You currently use a cronjob and some bash scripts to compile your application and push it out to all of your servers via SSH. It's both difficult to log, maintain and extend to new servers when they are provisioned. You've been considering a move to CodePipeline to manage everything for you. Will it suit your requirements?

1. Yes, CodeDeploy can deploy to any servers that can run the CodeDeploy agent
2. Yes, CodePipeline will be able to interface with any servers that can run the CodePipeline agent
3. No, CodePipeline won’t be able to deploy to the non-AWS managed servers in your data center
4. No, CodeDeploy won’t be able to deploy to the non-AWS managed servers in your data center

Answer: a

EXPLANATION:

CodeBuild is able to deploy to any server that can run the CodeDeploy agent, whether in AWS or in your own data centre.

You have been asked to deploy two, small internal applications on Elastic Beanstalk. One is a Python application which requires Nginx, and the other is a Java application with the need to have special java parameters passed to it. Which Platform option should you use as the Base Configuration to run each application? Choose the option that is the quickest and easiest to deploy in each case.

1. Create the Python application as a Docker container, and on the 'Platform' drop down, choose 'Generic - Docker' and upload the container and for the Java application, carry out the same process.
2. Create the Python application as a Docker container, and on the 'Platform' drop down, choose 'Generic - Docker' and upload the container. For the Java application, choose 'Preconfigured - Node.js' and upload the code.
3. Create the Python application as a Docker container, and on the 'Platform' drop down, choose 'Generic - Docker' and upload the container. For the Java application, choose 'Preconfigured - Java' and upload the code.
4. On the 'Platform' drop down, choose 'Preconfigured - Python' for the Python application and then upload the code. For the Java application, choose 'Preconfigured - Java' and upload the code.

Answer: c

EXPLANATION:

To choose the quickest and easiest deployment mechanism, you must understand what options the preconfigured solutions give you. In the case of Python, it comes with Apache HTTP by default, but we required Nginx, so we cannot choose a preconfigured option for the Python application and we must build it as a contaniner. For the Java application however, it will happliy be deployed as the preconfigured option with the java parameters specified in the 'Environment properties' section.

About a dozen people collaborate on a company-internal side project using CodeCommit. The developer community is spread across multiple timezones and relies on repository notifications via email. Initially, it was configured for all event types but this resulted in a lot of emails and team members complained about too much 'noise'. Since then, commit comment notifications have been turned off. How can you improve this situation?

1. Assign Amazon SNS subscription filter policies to the 'commit comment' topic subscriptions so that team members receive only a subset of the messages.
2. For each team member, create an individual SNS topic and a CodeCommit trigger that uses a Lambda function to filter out notifications not authored by that developer. The remaining ones are send to that SNS topic. This allows users to selectivly subscribe to specific persons to follow their activities.
3. Ask all team members to sign in to CodeCommit as IAM users.
4. Create a CodeCommit trigger that invokes a Lambda function when a new comment is added to a commit. Configure it so that its 'Custom data' field is populated with the email address of the user who authored the original commit, i.e. ${commit.author.email}. In the function, use SNS to send the notification to that address.

Answer: a, c

EXPLANATION:

You can create up to 10 triggers for each CodeCommit repository. The customData field is of type string and is used for information that you want included in the Lambda function such as the IRC channel used by developers to discuss development in the repository. It cannot be used to pass any dynamic parameters. This string is appended as an attribute to the CodeCommit JSON returned in response to the trigger. You can comment on an overall commit, a file in a commit, or a specific line or change in a file. For best results, use commenting when you are signed in as an IAM user. The commenting functionality is not optimized for users who sign in with root account credentials, federated access, or temporary credentials. By default, an Amazon SNS topic subscriber receives every message published to the topic. To receive a subset of the messages, a subscriber must assign a filter policy to the topic subscription

Your organization has been using CodePipeline to deploy software for a few months now and it has been smoothly for the majority of releases, but when something breaks during the build process it requires lots of man hours to determine what the problem is, roll back the deployment and fix it. This is frustrating both management and your customers. Your supervisor would like to assign one developer to test the build works successfully before your CodePipeline proceeds to the deploy stage so you don't encounter this issue again. How would you implement this?

1. Configure SES to email the assigned developer when CodePipeline has deployed to production. This will provide an immediate notification so they can check if there are any errors in testing. Then they can push the changes to the production git branch.
2. Configure SQS to email the assigned developer when CodePipeline has deployed to production. This will provide an immediate notification so they can check if there are any errors in testing. Then they can push the changes to the production git branch.
3. Create a test deploy stage as well as a manual approval stage in CodePipeline. Once the assigned developer checks the testing deploy worked, they can authorize the pipeline to continue and deploy to production.
4. Ask the assigned developer to run a local build first to test all changes, and then commit it to the repository which is then deployed to production only when they know there are no errors.

Answer: c

EXPLANATION:

CodePipeline allows for manual approval steps to be implemented for exactly this reason

You want to use Jenkins as the build provider in your CI/CD Pipeline. Is this possible, and if so how would you implement it?

1. Yes it's possible. You can use a CodePipeline plugin for Jenkins and can configure a build stage which connects to your Jenkins instance.
2. Yes it's possible. CodeBuild will let you select Jenkins as a source provider when you are creating your build project
3. No it's not possible.
4. Yes it's possible. CodePipeline will let you select Jenkins as a destination build provider when you are creating your pipeline.

Answer: a

EXPLANATION:

You can select Jenkins an action provider when creating a build stage in CodePipeline. You cannot select it as a source provider within CodeBuild.

You have multiple teams of developers and at the moment they all have the ability to start and stop any EC2 instance that they can see in the EC2 console, which is all of them. You would really like to implement some security measures so they can only start and stop the instances based on their cost center. What AWS features would you use to achieve this?

1. Implement tags and restrict access by comparing the iam:PrincipalTag and the aws:ResourceTag in a policy attached to your developer role and seeing if they match.
2. Implement tags and restrict access by comparing the aws:PrincipalTag and the iam:ResourceTag in a policy attached to your developer role and seeing if they match.
3. Implement EC2 policies which you can assign to each resource which will allow a developer to start or stop the instance if they are also assigned to it.
4. Implement roles which you can assign to each resource which will allow a developer to start or stop the instance if they are also assigned to it.

Answer: b

EXPLANATION:

You can simplify user permissions to resources by using tags and policies attached to roles. the aws:PrincipalTag is a tag that exists on the user or role making the call, and an iam:ResourceTag is a tag that exists on an IAM resource. In this case we want the CostCenter tag on the resource to match the ConstCenter tag assigned to the developer.

You are using the AWS Serverless Application Model to build a serverless application on AWS. You've just completed the development of a new feature that you wish to roll out to production. Despite proper testing, you want to gradually shift customer traffic to the updated Lambda version in increments of 10% with 10 minutes between each increment until you are satisfied that it's working as expected. How can you achieve this?

1. In the AWS SAM template, specify the Canary10Percent10Minutes Deployment Preference Type.
2. Define a DeploymentPreference of type 'Linear10PercentEvery10Minutes' in your AWS SAM template.
3. Set up a DeploymentPreference alarm that flips, after the traffic shifting has completed, the Lambda alias back to the old version if the CloudWatch Alarm goes to the Alarm state.
4. Configure a post-traffic hook Lambda function to run a sanity test that is invoked by CodeDeploy after traffic shifting completes.

Answer: b, d

EXPLANATION:

When a canary deployment preference is used, traffic will be shifted in two increments. Various available options specify the percentage of traffic that's shifted to the updated Lambda function version in the first increment, and the interval, in minutes, before the remaining traffic is shifted in the second increment. During traffic shifting, if any of the CloudWatch Alarms go to Alarm state, CodeDeploy will immediately flip the Alias back to old version and report a failure to CloudFormation.

Your senior developer wants to be able to access any past version of the binaries that are being built as part of your CI/CD pipeline. You are using CodeBuild with CodePipeline to automate your build process. How will you achieve this?

1. Nothing needs to change, by default all artifacts are given a unique filename in S3.
2. Change the artifact namespace type to Build ID, which will insert the build ID into path of the output folder in S3.
3. Implement a CodeBuild lambda trigger which will copy each build artifact to S3 with a unique ID.
4. Change the artifacts packaging to zip, which will append a version number to each build.

Answer: b

EXPLANATION:

CodeBuild has an optional 'Namespace type', which will insert the build ID into the path to the build output zip file or folder, giving you a unique directory and binary artfact for each build that runs.

Your CI/CD pipeline generally runs well, but your manager would like a report of some CodeBuild metrics, such as how many builds were attempted, how many builds were successful and how many builds failed in an AWS account over a period of time. How would you go about gathering the data for you manager?

1. Configure CodeBuild to log builds to CloudWatch Logs, and then write a metric filter which will graph the data points your manager requires.
2. CloudWatch metrics will report these metrics by default. You can view them in the CloudWatch console.
3. Configure a CloudWatch custom metric to track the build information, and create custom graphs in the CloudWatch console.
4. Implement a lambda function to poll the CodeBuild API to gather the data and store it in CloudWatch Logs. Write a metric filter to graph the data and generate your report.

Answer: b

EXPLANATION:

These are default CloudWatch metrics that come with CodeBuild.

Your organization is currently using CodeDeploy to deploy your application to 20 EC2 servers which sit behind a load balancer. It's making use of the CodeDeployDefault.OneAtATime deployment configuration. Your manager has decided to speed up deployment by deploying to as many servers as possible at once, as long as at least five of them remain in service at any one time. How do you achieve this, ensuring that it will scale?

1. Create a custom deployment configuration, specifying the maximum healthy host as a "Number" and set it to 5
2. Create a custom deployment configuration, specifying the minimum healthy host as a "Percentage" and set it to 25%
3. Create a custom deployment configuration, specifying the maximum healthy host as a "Percentage" and set it to 25%
4. Create a custom deployment configuration, specifying the minimum healthy host as a "Number" and set it to 5

Answer: d

EXPLANATION:

Setting the minimum healthy host as a Number and 5 will work as desired. A percentage set to 25% will work for the current 20 servers, but it will not scale down if any servers are removed.

In your organization, your recently departed DevOps engineer configured CodeDeploy to use Blue/green deployment in your CodePipeline. This has worked brilliantly in the past, but tighter budgets have unfortunately caused management to direct your department to only run one set of servers and to update everything at the same time, despite your objections. How will you revert to this obviously inferior deployment method in the simplest way?

1. Create a new CodeDeploy application deployment group with the deployment type set to "In-place", and modify your application to use the new deployment group configuration and delete the old one.
2. Edit your CodePipeline deployment group and change the deployment type from Blue/Green to In-place.
3. Edit your CodeDeploy application deployment group and change the deployment type from Blue/Green to In-place.
4. Create a new CodePipeline application deployment group with the deployment type set to "In-place", and modify your application to use the new deployment group configuration and delete the old one.

Answer: c

EXPLANATION:

You are able to edit the application deployment group in CodeDeploy, you don't have to create a new one.

The world wide cat news powerhouse, Meow Jones, has hired you as a DevOps Database consultant. They're currently using legacy in-house PostgreSQL databases which cost a considerable amount to maintain the server fleet, as well as operational costs for staff, and further hardware costs for scaling as the industry grows. You are tasked in finding an AWS solution which will meet their requirements. They require high throughput, push button scaling, storage auto-scaling and low latency read replicas. Any kind of automatic monitoring and repair of databases instances will also be appreciated. Which AWS service(s) do you suggest?

1. A cluster of Amazon RDS PostgreSQL instances, AWS Database Migration Service.
2. An autoscaled, load balanced EC2 fleet running PostgreSQL with data shared via EFS volumes.
3. Keep the current PostgreSQL databases and implement an ElastiCache to cache common queries and reduce load on your in-house databases to save on upgrade costs.
4. Amazon Aurora, AWS Database Migration Service

Answer: d

EXPLANATION:

Amazon Aurora will fit the needs perfectly, and the Database Migration Service can assist with the migration.

Due to the design of your application, your EC2 servers aren't treated as cattle as advised in the cloud world, but as pets. As such, you need DNS entries for each of them. Managing each DNS entry is taking a long time, especially when you have lots of servers, some of which may last a day, a week or a month. You don't want your Route53 records to be messy, and you would prefer some kind of automation to add and remove them. Which method would you choose to solve this in the best way?

1. Tag your instance with the DNS record required. Deploy a Lambda function which can add or remove DNS records in Route53 based on the DNS tag. Use a CloudWatch Events rule to monitor when an instance is started or stopped and trigger the Lambda function.
2. Make your instance ID the DNS record required. Deploy a Lambda function which can add or remove DNS records in Route53 based on the DNS tag. Use a CloudWatch Events rule to detect when an instance is started or stopped and trigger the Lambda function.
3. Make your instance ID the DNS record required. Deploy a Lambda function which can add or remove DNS records in Route53 based on the DNS tag. Use CloudTrail API call logs to detect when an instance is started or stopped and trigger the Lambda function.
4. Tag your instance with the DNS record required. Deploy a Lambda function which can add or remove DNS records in Route53 based on the DNS tag. Use CloudTrail API call logs to monitor when an instance is started or stopped and trigger the Lambda function.

Answer: a

EXPLANATION:

Tagging your instance with the required DNS record is a great way to help you automate the creation of Route53 records. A Lambda function can be triggered from a CloudWatch Events EC2 start/stop event and can add and remove the Route53 records on your behalf. This will meet your requirements and automate the creation and cleanup of DNS records.

You run a load balanced, autoscaled website in EC2. Your CEO informs you that due to an upcoming public offering, your website must not go down, even if there is a region failure. What's the best way to achieve this?

1. Deploy CloudFront in front of your instances. It will cache requests even if a region goes down and your users will not notice.
2. Deploy your load balancers and autoscaled website in two different availability zones. Create a Route53 GeoProximity Routing Record. Point the record to each of your Elastic LoadBalancers.
3. Deploy your load balancers and autoscaled website in two different availability zones. Create a Route53 weighted Routing Record. Point the record to each of your Elastic LoadBalancers.
4. Deploy your load balancers and autoscaled website in two different regions. Create a Route53 Latency Based Routing Record. Point the record to each of your Elastic LoadBalancers.

Answer: d

EXPLANATION:

A latency based routing policy will keep your website as fast as possible for your customers, and will act as redundancy should one of the regions go down.

You currently run an autoscaled application which is database read heavy. Due to this, you are making use of a read replica for all application reads. It's currently running at around 60% load with your user base but you expect your company growth to double the amount of users every 6 months. You need to forward plan for this and determine methods to ensure the database doesn't become a bottleneck while still maintaining some redundancy. What is the best way to approach this issue?

1. Create more read replicas. Use a Route53 weighted routing policy to ensure the load is spread across your read replicas evenly.
2. Monitor your database read replica usage in CloudWatch alerts. When it's close to 90% capacity perform an online resize to a larger instance type.
3. Deploy an additional Multi-AZ RDS read replica and modify your application to use it instead.
4. Create another read replica and deploy a second autoscaled version of your application. Point it at your second read replica.

Answer: a

EXPLANATION:

More read replicas will ease the load on your current ones, and load balancing them with a weighted routing policy will ensure they're not a single point of failure for your application.

You're assisting a developer working on a very large and read-heavy application which uses an Amazon Aurora database cluster. The feature currently being worked on requires reading but no writing, however it will be called by the application frequently and from multiple different servers so the reads need to be load balanced. Additionally your reporting team need to make ad-hoc, expensive queries which need to be isolated so that reads for production are not affected by reporting. Which Aurora configuration fulfils both needs with minimal extra configuration?

1. Use the Custom endpoint for your Production system to make its reads against a single high-capacity replica, and create another custom endpoint pointing to a dedicated reporting replica for isolation of ad-hoc reporting.
2. Use the Reader endpoint for your Production system to make its load-balanced reads against high-capacity read replicas, and create a custom endpoint pointing to a separate replica for isolation of ad-hoc reporting.
3. Use the Cluster endpoint for your Production system to make its reads against high-capacity read replicas, and create an Aurora reporting endpoint pointing to a separate replica for isolation of ad-hoc reporting.
4. Use the Aurora Instance endpoint for your Production system to make load-balaned reads against a read replicas, and create another custom endpoint pointing to a dedicated reporting replica for isolation of ad-hoc reporting.

Answer: b

EXPLANATION:

The Reader endpoint is appropriate in this situation. The reader endpoint provides load balanced support for read only connections to the database cluster. A custom endpoint can be used to connect to an isolated replica for report generation or ad hoc (one-time) querying,

Your CEO wants you to start future proofing your AWS environment, so he's asked you to look into IPv6 compatibility of your existing Load Balanced EC2 stack. You make use of both Application (ALB) and Network (NLB) load balancers in your EC2-VPC. What are your findings?

1. Application Load balancers do not support IPv6, Network Load balancers do.
2. Application Load balancers support IPv6, Network Load Balancers do not.
3. Application and Network Load Balancers both support IPv6.
4. No Load Balancers in EC2 support IPv6.

Answer: b

EXPLANATION:

At this moment in time only the Application Load Balancer supports IPv6 in the EC2-VPC environment. Classic Load balancers only support IPv6 if you are using an EC2-Classic environment.

You currently have a lot of IoT weather data being stored in a DynamoDB database. It stores temperature, humidity, wind speed, rainfall, dew point and air pressure. You would like to be able to take immediate action on some of that data. In this case, you want to trigger a new high or low temperature alert and then send a notification to an interested party. How can you achieve this in the most efficient way?

1. Use a DynamoDB stream and Lambda trigger only on a new temperature reading. Send a SNS notification if a record is breached.
2. Modify your IoT devices to also log their data to Kinesis Data Firehose and trigger a Lambda function which will check for new high or low temperatures. Send a SNS notification.
3. Write an application to use a DynamoDB scan and select on your Sort key to determine the maximum and minimum temperatures in the table. Compare them to the existing records and send an SNS notification if they are breached. Run the application every minute.
4. Use CloudWatch custom metrics to plot your temperature readings and generate an event alert if it breaches your high and low thresholds.

Answer: a

EXPLANATION:

Using a DynamoDB stream is the most efficient way to implement this. It allows you to trigger the lambda function only when a temperature record is created, thus saving Lambda from triggering when other records are created, such as humidity and wind speed.

Your company utilizes EC2 and various other AWS services for its workloads. As the DevOps engineer it is your responsibility to ensure all company policies are implemented. You have noticed that while you are using S3 for data archival and backups, your company policy is that your backups need to reside on company owned servers, such as those you run in your local Equinix data center. You also have another company policy that backup and archival cannot traverse the internet. What do you do?

1. Provision an AWS Direct Connect connection to your local router in your data center and your local VPC. Push backups via the Direct Connect connection.
2. Implement a Site to site VPN with a company owned server in your data center. Push backups to your data center backup NAS through SSH via vpn.
3. Implement an AWS Client VPN with a company owned server in your data center. Push backups to your data center backup NAS through SSH via vpn.
4. Install the EFS agent on your data center backup NAS, mount the volume on an EC2 server and copy the backups to the volume.

Answer: a

EXPLANATION:

AWS Direct Connect is the only way to access your AWS resources from a Data Center without traversing the internet, despite the encryption offered by the other solutions.

You work for a medical imaging company, dealing with X-rays, MRI's, CT scans and so on. The images and other related patient reports and documents are stored in various S3 buckets in the US West region. Your organization is very security conscious and wants to ensure that while the S3 buckets are locked down, there's no other way that the documents are being shared internally or externally other than the approved methods already in place. Audits are also important, so whatever methods of data protection are in place must work together with this, as well as providing actionable alerts if there any observed issues. How do you best achieve this? Which AWS services can help?

1. Don't store sensitive data in S3, the public cloud is not secure enough. Look into moving storage in-house.
2. Write a lambda function to monitor CloudTrail API calls to S3 and trigger an SNS notification if anything out of the ordinary is detected.
3. Write a lambda function which is triggered when new data is uploaded into your S3 buckets to apply an S3 policy to ensure the data is secure.
4. Implement Amazon Macie across your S3 buckets.

Answer: d

EXPLANATION:

Amazon Macie is a security service that uses machine learning to discover personally identifiable information in your S3 buckets. It also provides you with dashboards and alerts that show how your private data is being accessed.

Your fleet of Windows EC2 instances are running well. They're automatically built from an AMI and generally don't require much interaction, except for when they need to be patched with the latest Windows updates or have new software installed or updated. Ideally, you'd like to automate the process to install or update applications and apply windows updates and patches during the quiet hours when your customers are sleeping, which means around 3am. How would you best automate this process?

1. Implement AWS Systems Manager Run Commands to execute the PowerShell scripts to run updates at 3am
2. Implement AWS Systems Manager Patch Manager and AWS Systems Manager Maintenance Windows
3. Implement a Lambda function to run PowerShell update commands on a schedule using CloudWatch Events rules for each server
4. Implement AWS Systems Manager Patch manager and AWS Systems Manager Run Commands

Answer: b

EXPLANATION:

Patch Manager combined with Maintenance Windows in AWS Systems manager is the recommended way to automate this requirement.

Your company is preparing to become ISO 27001 certified and your manager has asked you to propose a comprehensive solution to log configuration and security changes in a separate audit account. Specifically, the solution should ensure IAM users have MFA enabled, identify S3 buckets which aren't encrypted and enforce the addition of specific tagging. Identify which option solves the problem.

1. Enable AWS Config and create three default rules to check whether IAM users have MFA enabled, S3 buckets have server side encryption and tagging is added to resources.
2. Enable AWS Config and AWS Cloudtrail to track changes in all resources and to identify IAM user API calls with MFA enabled.
3. Enable AWS Cloudtrail to check for MFA enabled IAM users, configure Server access logging in S3 to view the encryption status and use a CloudFormation Templates to add tagging.
4. Enable Enhance Logging in AWS Cloudwatch to track all security and configuration changes and view these using Cloudwatch Logs Insights.

Answer: a

EXPLANATION:

AWS Config is the only service that will meet all of the requirements in the question as it records configuration changes and snapshots the configuration at regular intervals set by you. Data aggregation means that AWS Config data from multiple accounts can be stored in a single account. The following built in rules; s3-bucket-server-side-encryption-enabled and iam-user-mfa-enabled identify any S3 buckets not encrypted and any IAM accounts that do not have MFA enabled. Tagging is also available for AWS Config resources that describe AWS Config rules.

You work for a global service provider, deploying critical software 24 hours a day. You have 3 AWS Accounts; 'POC' allows for Developers to stand up new technology and try out new ideas on an adhoc basis. 'QA' allows for automated builds and testing to be carried out as part of your CI/CD Pipeline and any problems here mean that software will not get pushed into production, so it's important that any issues are resolved quickly. The final account is 'Live' which is the main Production account. It's the most critical and requires the best response times. You need to choose the most appropriate and cost effective Support Plan which will satisfy your support needs but also allow for the full range of AWS Trusted Advisor Checks and Recommendations.

1. The 'POC' Account should be allocated the Basic Support Plan, 'QA' the Business Support Plan and 'Live' the Enterprise Support Plan.
2. The 'POC', 'QA' and 'Live' Accounts should all be allocated the Business Support Plan.
3. The 'POC' and 'QA' Accounts should use the Basic Support Plan, while 'Live' should use the Enterprise Support Plan.
4. The 'POC' Account should be allocated the Basic Support Plan and both 'QA' and 'Live' the Business Support Plan.
5. The 'POC' Account should be allocated the Developer Support Plan, 'QA' the Business Support Plan and 'Live' the Enterprise Support Plan.

Answer: b

EXPLANATION:

The important phrase in the question is that any choice must "...allow for the full range of AWS Trusted Advisor Checks and Recommendations", only two support plans have all the Trust Advisor checks and these are Business and Enterprise. Anything that contains the Basic or Developer plans will not be correct. Also, it specifies that the 'Live' account is "critical and requires the best possible response times", so we could use the Enterprise plan for this, but we'll only get better response time for mission critical business applications such as Microsoft, SAP and Oracle tools, which is not the software we are deploying. Therefore taking all of this into account, along with cost, 'POC', 'QA' and 'Live' should all be allocated the Business plan.

An error has occurred in one of the applications that your team looks after and you have traced it back to a DB connection issue. There have been no network outages and the database is up and running. A colleague tells you that all credentials are now stored in AWS Secrets Manager and suspects that the problem might be caused by a recent change in that area. Select all possible reasons for this.

1. The AWS credentials that are used for the call to Secrets Manager in the client-side component embedded in the application to retrieve the database password don't have the secretsmanager:DescribeSecret permission on that secret.
2. The DB credentials and connection details in Secrets Manager have been encrypted using the default Secrets Manager CMK for the account. The application and secret are in different AWS accounts though and no cross-account access has been granted.
3. The GetSecretValue API call in the application didn't include the version of the secret to return.
4. When secret rotation is configured in Secrets Manager, it causes the secret to rotate once as soon as you store the secret. This can lead to a situation where the old credentials are not usable anymore after the initial rotation. It is possible that the team forgot to update the application to retrieve the secret from Secrets Manager.

Answer: a, b, d

EXPLANATION:

Although you typically only have one version of the secret active at a time, multiple versions can exist while you rotate a secret on the database or service. GetSecretValue has an optional VersionId parameter that specifies the unique identifier of the version of the secret that you want to retrieve. If you don't specify either a VersionStage or VersionId then the default is to perform the operation on the version with the VersionStage value of AWSCURRENT.

Your fleet of Windows EC2 instances are running well. They're automatically built from an AMI and generally don't require much interaction, except for when they need to be patched with the latest Windows updates or have new software installed or updated. Ideally, you'd like to automate the process to install or update applications and apply windows updates and patches during the quiet hours when your customers are sleeping, which means around 3am. How would you best automate this process?

1. Implement AWS Systems Manager Run Commands to execute the PowerShell scripts to run updates at 3am
2. Implement a Lambda function to run PowerShell update commands on a schedule using CloudWatch Events rules for each server
3. Implement AWS Systems Manager Patch manager and AWS Systems Manager Run Commands
4. Implement AWS Systems Manager Patch Manager and AWS Systems Manager Maintenance Windows

Answer: d

EXPLANATION:

Patch Manager combined with Maintenance Windows in AWS Systems manager is the recommended way to automate this requirement.

Your organization has multiple weather stations installed around your state. You would like to move the data from your old relational database to a newer and hopefully faster NoSQL database. Which AWS solution would you choose for storing and migrating the data, and which keys do you use for your weather station ID and the timestamp of the reading?

1. Use Amazon DMS to migrate data to an Amazon ElasticCache table. Select your RDS instance as the source and ElasticCache as destination. Create a task with an object mapping rule to copy the required relational database tables to DynamoDB. Within your object-mapping set your weather station ID as the Primary Key of your table and timestamp of the reading as the Sort key.
2. Use Amazon DMS to migrate data to an Amazon DynamoDB table. Select your RDS instance as the source and DynamoDB as destination. Create a task with an object mapping rule to copy the required relational database tables to DynamoDB. Within your object-mapping set your weather station ID as the Primary Key of your table and timestamp of the reading as the Sort key.
3. Use Amazon DMS to migrate data to an Amazon DataTables table. Select your RDS instance as the source and DataTables as destination. Create a task with an object mapping rule to copy the required relational database tables to DynamoDB. Within your object-mapping set your weather station ID as the Partition Key of your table and timestamp of the reading as the Sort key.
4. Use Amazon DMS to migrate data to an Amazon DynamoDB table. Select your RDS instance as the source and DynamoDB as destination. Create a task with an object mapping rule to copy the required relational database tables to DynamoDB. Within your object-mapping set your weather station ID as the Partition Key of your table and timestamp of the reading as the Sort key.

Answer: d

EXPLANATION:

DynamoDB is the AWS NoSQL database. The Weather Station ID will be your Partition Key and the timestamp will be your Sort key.

You have a new website design you would like to test with a small subset of your users. If the test is successful, you would like to increase the amount of users accessing the new site to half your users. If that's successful and your infrastructure is able to scale up correctly, you would like to completely roll over to the new design and then decommission the servers hosting the old design. Which of these methods do you choose?

1. Install the new website design in a new AutoScaling group. Use a Weighted Routing policy in Route53 and use it to choose the percentage of users you would like during different testing phases. Start with 5%, then 50%, and end with 100% of traffic going to the new AutoScaling group if tests are successful. Decommission the old EC2 servers.
2. Install the new website design in a new AutoScaling group. Use an A/B Test Routing policy in Route53 and use it to choose the percentage of users you would like during different testing phases. Start with 5%, then 20%, and end with 100% of traffic going to the new AutoScaling group if tests are successful. Decommission the old EC2 servers.
3. Install the new website design in a new AutoScaling group. Use a Weighted Routing policy in Route53 and use it to choose the percentage of users you would like during different testing phases. Start with 5%, then 20%, and end with 100% of traffic going to the new AutoScaling group if tests are successful. Decommission the old EC2 servers.
4. Install the new website design in a new AutoScaling group. Create a Lambda function to modify your Route53 apex record to use the new AutoScaling group for 5% of the day. If that's successful then modify the function to change the apex record for half the day, and end with 100% of traffic going to the new AutoScaling group if tests are successful. Decommission the old EC2 servers.

Answer: a

EXPLANATION:

A weighted routing policy combined with an autoscaling group will meet your requirements and will continue to scale if your tests are successful and you completely roll over to the new design.

The security officer of the company you work for has mandated that from now on all production database credentials are to be changed every 90 days. You've decided to automate this for a cluster of Db2 instances and want to use AWS Secrets Manager. This database is used by several of your applications but owned by an external party who has given you three separate DB users for different environments (DEV, TEST, PROD). You can change their passwords yourself but not create new users. Select the correct answer that describes the best way to proceed.

1. Update the apps to retrieve the DB credentials from AWS Secrets Manager. You will also need to configure Secrets Manager with a custom Lambda function that is called several times by it when rotation is triggered, each time with different parameters. It is expected to perform several tasks throughout the process of rotating a secret. The task to be performed for each request is specified by the 'Step' parameter in the request. Every step is invoked with a unique 'clientTokenRequest' parameter.
2. You can implement the DB credentials change with an AWS Lambda function that changes the DB user's password and updates the application's DB connection configurations. Use CloudWatch events to trigger this every 3 months.
3. AWS Secrets Manager allows you to automatically rotate secrets for some Amazon RDS databases, Redshift and DocumentDB. Db2 is not a supported RDS database engine and therefore, you cannot use AWS Secrets Manager to rotate your secrets and should use the AWS Systems Manager Parameter Store instead.
4. You should ask the external party for a DB user with at least two credential sets or the ability to create new users yourself. Otherwise, you might encounter client sign-on failures. The risk is because of the time lag that can occur between the change of the actual password and - when using Secrets Manager - the change in the corresponding secret that tells the client which password to use.

Answer: d

EXPLANATION:

Secrets Manager already natively knows how to rotate secrets for supported Amazon RDS databases. However, it also can enable you to rotate secrets for other databases or third-party services. It provides a secure API that enables the programmatic retrieval of secrets to ensure that it can't be compromised by someone examining your code and configurations stored in a version control system. Secrets Manager invokes the secrets rotation Lambda function each time with the same secretId and clientTokenRequest. Only the Step parameter changes with each call. This helps prevent you from having to store any state between steps.

Your CEO loves serverless. He won’t stop talking about how your entire company is built on serverless architecture. He attends Serverlessconf to talk about it. Now, it's up to you to actually build the web application he's been talking about for 6 months. Which AWS services do you look at using to both create the application and orchestrate your components?

1. Create your application using AWS Lambda for compute functions within the application. Data storage can be provided using Amazon DynamoDB for a NoSQL database. File storage can be provided using Amazon S3. AWS Serverless Application Framework can orchestrate workflows and AWS Glacier will allow you to archive old files cheaply.
2. Create your application using AWS Lambda for compute functions within the application. Data storage can be provided using Amazon DynamoDB for a NoSQL database. File storage can be provided using Amazon S3. AWS Step Functions can orchestrate workflows and AWS Glacier will allow you to archive old files cheaply.
3. Create your application using AWS Elastic Beanstalk for compute functions within the application. Data storage can be provided using Amazon DynamoDB for a NoSQL database. File storage can be provided using Amazon S3. AWS Step Functions can orchestrate workflows and AWS Cold Store will allow you to archive old files cheaply.
4. Create your application using AWS Elastic Compute for compute functions within the application. Data storage can be provided using Amazon DynamoDB for a NoSQL database. File storage can be provided using Amazon S3. AWS Step Functions can orchestrate workflows and AWS Glacier will allow you to archive old files cheaply.

Answer: b

EXPLANATION:

AWS Lambda is the correct choice for compute in a serverless environment, as is DynamoDB for NoSQL databases and S3 for storage. AWS Step Functions are used to orchestrate your components, such as your lambda functions.

You are the only developer in your new team, and you are developing a PHP application with a postgres database. You would like to just focus on your code and spend a minimal amount of time in the AWS console. You would also like any new developer who joins your team to be able to deploy to the environment in a simple way, as they are traditional developers and not cloud engineers. It also needs to support rolling deployments as you grow, giving you the ability to deploy your software to existing servers without taking them all offline, as well as rolling back if required. Which AWS service do you choose to meet these needs, and how do you roll back if required?

1. Build and deploy the application using AWS CloudFormation and AWS Autoscaling. Deploy using CloudFormation Update Stacks, and roll back on failure to the using the previous template automatically stored in S3
2. Build and deploy your application using the AWS CodeDeploy service. Deploy to your servers with CodeDeploy Rolling Deploy. Initiate a roll back if required using CodeDeploy Rollback.
3. Build and deploy your application using the AWS Elastic Beanstalk service. Deploy to your servers with All-at-once deployment and initiate a roll back if required using manual redeploy.
4. Build and deploy your application using the AWS Elastic Beanstalk service. Deploy to your servers with Rolling deployment and initiate a roll back if required using manual redeploy.

Answer: d

EXPLANATION:

AWS Elastic Beanstalk is the best candidate for this requirement. You can quickly deploy code without having to learn about the infrastructure that runs those applications. Rolling deployments are possible, and a manual redeploy of the old version of your code will be required to roll back.

ADogGuru is developing a new Node.js application which will require some servers, a mySQL database and a load balancer. You would like to deploy and maintain this in the easiest way possible, have the ability to configure it with basic configuration management code, which is a new concept for ADogGuru. You also need to allow non-technical people to deploy software to servers and for managers to administer access. A key component of this setup will also be the ability to rollback deploys should you need to. Which service(s) and configuration management system will you choose to achieve this with the LEAST operational overhead?

1. Use AWS OpsWorks to create an application using pre-built layer templates to create your servers, mySQL RDS instances and load balancer. Use recipes running in Chef solo for configuration management. Grant your non-technical staff the 'deploy' permission level, and the administrator the 'Manage' permission level.
2. Use AWS OpsWorks to create an application using pre-built layer templates to create your servers, mySQL RDS instances and load balancer. Use OpsWorks for Puppet Enterprise for configuration management. Grant your non-technical staff the 'deploy' permission level, and the administrator the 'Manage' permission level.
3. Use AWS OpsWorks to create an application using pre-built layer templates to create your servers, mySQL RDS instances and load balancer. Also deploy OpsWorks for Chef Automate and run recipes for configuration management. Grant your non-technical staff the 'deploy' permission level, and the administrator the 'Manage' permission level.
4. Use AWS Elastic Beanstalk to create an application using pre-built layer templates to create your servers, mySQL RDS instances and load balancer. Use the OpsWorks Puppet Enterprise server for configuration management. Grant your non-technical staff the 'deploy' permission level, and the administrator the 'Manage' permission level.

Answer: a

EXPLANATION:

OpsWorks can achieve all of stated requirements, deployments are possible with a few clicks in the UI, and can be rolled back. It also supports Chef solo as a built-in configuration management system, which is the recommended solution for those not already using Puppet or Chef. Running a separate Chef Automate instance is unnecessary overhead and cost when Chef Solo will suffice.

You have an application built on docker. You would like to not only launch your docker instances at scale but load balance them as well. Your director has also enquired as to whether it is possible to build and store the containers programatically using AWS API calls in future or if additional products are required. Which AWS services will enable you to store and run the containerised application and what do you tell the director?

1. Build EC2 instances with docker using EC2 and AutoScaling and store them using AWS ECR (Elastic Container Registry). Run the containers on docker desktop on EC2. Yes automated container builds are possible with AWS CodeBuild pushing to ECR and using ECS APIs.
2. Build containers and store them using AWS ECR (Elastic Container Registry). Run the containers using Amazon ECS. Yes automated container builds are possible with AWS CodeBuild pushing to ECR and using ECS APIs.
3. Build containers and store them using AWS ECR (Elastic Container Registry). Run the containers using Amazon ECS. Automated container builds are not possible with AWS services and APIs however third-party build engines such as Jenkins and BuildKite will achieve your director's goal.
4. Build EC2 instances with docker using EC2 and AutoScaling and store them using AWS ECR (Elastic Container Registry). Run the containers using Autoscaling EC2. Automated container builds are not possible with AWI services and APIs however third-party build engines such as Jenkins and BuildKite will achieve your director's goal.

Answer: b

EXPLANATION:

Amazon ECS and ECR are the Amazon Elastic Container Service and Registry, which will meet all of the requirements specified. It also has an API which can be used to implement your requirements.

You are developing an augmented reality mobile game. Initially, you want to launch this in California but plan to expand rapidly to other areas if the pilot runs successfully. For operational purposes, you need to keep track of the number of players in each area and decided to use a custom CloudWatch metric called 'Population' for this. To enable a basic level of analytics, you also want to capture more than a dozen mobile device specific values such as battery level and network signal strength. During the pilot, you specify a data collection frequency of 6 times per minute. How could you achieve this? Select all correct statements.

1. For the initial launch, create a separate CloudWatch namespace for your 'Population' metric for each individual area such as 'NewEGamesLtd-ARGame>MountainView'.
2. Standard resolution is for data having a one-minute granularity. When you publish a high-resolution custom metric, CloudWatch stores it with a resolution of 1 second, and you can read and retrieve it with a period of 1 second, 5 seconds, 10 seconds, 30 seconds, or any multiple of 60 seconds. If you choose to use one or more custom CloudWatch metrics for your analytics requirements, define these as high-resolution.
3. Every 10 seconds, make a 'PutMetricData' call from within the app to publish the data to a new 'Analytics' CloudWatch custom metric. Specify the individual device values as separate dimensions, e.g. --dimensions BatteryLevel=42,SignalStrength=72, etc.
4. Assign CloudWatch dimensions to your 'Population' metric that specify the player's current area, i.e. Country=USA, State=California, City=MountainView, etc. At any time, you can then get the total number of active players on any geographical level by aggregating across the relevant dimensions.
5. If you choose to use one or more custom high-resolution CloudWatch metrics for your analytics requirements, your data points with a period of less than 60 seconds are available for only 3 hours. After that, the data is still available but is aggregated together for long-term storage with a 1-minute resolution. After 15 days the data is aggregated again and is retrievable only with a resolution of 5 minutes.

Answer: b, e

EXPLANATION:

Namespace names must contain valid XML characters. Possible characters are: alphanumeric characters (0-9A-Za-z), period (.), hyphen (-), underscore (\_), forward slash (/), hash (#), and colon (:). CloudWatch does not aggregate across dimensions for your custom metrics and you can only assign up to 10 dimensions to a metric.

Your companies Security Officer just mandated a policy whereby all in use encryption keys within your organisation must be rotated every 120 days. How does that affect your CodePipelines Artifact store?

1. Your artifact store can either be a CodeCommit, GitHub or an Amazon ECR repository. It can also be an S3 source bucket where your source code is stored. Every 120 days, you need to generate new access keys and change the connection details to your repository if you are not using S3 as your artifact store. No further action is required if you use S3.
2. AWS recommends to use the same default global artifact store for all your CodePipelines. This is created for you when you create your first pipeline in any region. By default, CodePipeline uses server-side encryption with the AWS KMS-managed keys (SSE-KMS) using the default key for Amazon S3 (the aws/s3 key). This key is created and stored in your AWS account. When artifacts are retrieved from the artifact store, CodePipeline uses the same SSE-KMS process to decrypt the artifact. Change that key every 120 days.
3. CodePipeline Artifact stores are regional. When you use the CodePipeline console in a specific region to create a pipeline and choose 'Default location' when asked for an Artifact store, a new default one will be created for you in that region if none existed beforehand. CodePipeline creates default AWS-managed SSE-KMS encryption keys when you create a pipeline using the Create Pipeline wizard. The master key is encrypted along with object data and managed by AWS. However, you can also create and manage your own customer-managed SSE-KMS keys in AWS KMS to encrypt or decrypt artifacts in your artifact store and rotate these keys as necessary.
4. You can use any already existing artifact store as long as it is in the same Region as your pipeline. When you create a pipeline using AWS CloudFormation or the CLI, you must configure server-side encryption for your artifact store manually. Use an appropriate bucket policy and then create your own customer-managed SSE-KMS encryption keys. Instead of using the default Amazon S3 key, choose to use your own keys so that you can rotate these every 120 days as per your organisations security requirements.
5. You must have a separate artifact store for each CodePipeline. When using the AWS CLI, these are automatically created during the creation of your pipelines. CodePipeline also configures default AWS-managed SSE-KMS encryption keys for your artifact store. If you enable automatic key rotation and specify a refresh rate of 120 days, AWS KMS generates new cryptographic material and saves the key's older cryptographic material so that it can be used to decrypt data that it encrypted.

Answer: c, d

EXPLANATION:

An artifact store for your pipeline, such as an Amazon S3 bucket, is not the source bucket for your source code and is required for each pipeline. When you create or edit a pipeline, you must have an artifact bucket in the pipeline Region, and then you must have one artifact bucket per AWS Region where you are running an action. Unlike the console, running the create-pipeline command in the AWS CLI does not create an Amazon S3 bucket for storing artifacts. The bucket must already exist. When you enable automatic key rotation for a customer managed CMK, AWS KMS generates new cryptographic material for the CMK every year.

You are part of a development team that has decided to compile release notes directly out of a CodeCommit repository, the version control system in use. This step is to be automated as much as possible. Standard GitFlow is used as the branching model with a fortnightly production deploy at the end of a sprint and occasional hotfixes. Select the best approach.

1. Create a trigger for your CodeCommit repository using the 'Push to existing branch' event and apply that to any release and hotfix branch. Add an Amazon SNS topic as the target and have a Lambda listen to it. In that function, filter out specific commit type changes such as style, refactor and test that are not relevant for release notes. Store all other commit messages in a DynamoDB table and, at release time, run a query to collate the release notes.
2. Configure a trigger by choosing the 'Delete branch or tag' repository event that invokes a Lambda function when development for a sprint is finished, i.e. the last feature-\* branch has been deleted. In that Lambda, retrieve the latest git merge commit message before the deletion and append it to the release notes text file stored in an S3 bucket.
3. Setup up an Amazon CloudWatch Event rule to match CodeCommit repository events of type 'CodeCommit Repository State Change'. Look for 'referenceCreated' events with a 'tag' referenceType that are created when a production release is tagged after a merge into 'master'. In a Lambda function, use the CodeCommit API to retrieve that release commit message and store it in a static website hosting enabled S3 bucket.
4. Use the 'generate release notes' feature of CodeCommit by running the 'create-release-notes' command with the --from <datetime> (use the start of the sprint) or --previousTag <tagName> option in the AWS CLI. Create a Lambda to execute this on a regular schedule (i.e. every 2 weeks) using CloudWatch Events with a cron expression.

Answer: c

EXPLANATION:

Following GitFlow's standard release procedures, a release branch is merged into master. That commit on master must be tagged for easy future reference to this historical version. Both release and hotfix branches are temporary branches and would require ongoing updates of the CodeCommit trigger. Feature branches are used to develop new features for the upcoming or a distant future release and might be discarded (e.g. in case of a disappointing experiment). CodeCommit does not provide a generate release notes feature.

You are contracting for APetGuru, an online pet food store. Their website works fine, but you really want to update the look and feel to be more modern. They have given you strict guidelines that their website can not be offline at all or they will lose sales. You are thinking of using a rolling deployment so some of their servers are always online during the rollout. Just before you trigger the roll out, you receive a phone call asking if you can only install your updates on a few servers first for testing purposes. It is suggested that a few customers can be redirected to the updated site and everyone else can still use the old site until you confirm the new one is working properly. Once you're happy with the operation of the updated website, you can complete the rollout to all servers. What do you do?

1. Use a Blue/Green deployment. This allows you to install the new website to the blue servers, and once you're happy with it working you can finalise the install on the green servers. If there's an issue you can roll back the blue installation.
2. Use an In-Place Deployment. You can deploy to the "In-Place" servers first for testing, then once testing is verified you can continue the deployment to the "Out-Place" externally facing servers.
3. Use a Canary Deployment. This allows deployment to a few servers where you can observe how the website is running while still receiving a small amount of customers. If there's an error it can be rolled back. Otherwise, the rollout will continue on new servers.
4. Use a Batch Deployment. This allows you to install your new website to the small batch which traffic will be directed to, and will allow you to roll back the smnall batch if there's an error. Otherwise, you can complete the rollout on the large batch.

Answer: c

EXPLANATION:

A canary deployment will allow you to complete the roll out in the way you want to.

CodeBuild has been configured as the Action provider for your Integration Test Action which has been added to your CodePipelines' 'Test' stage. These tests connect to your RDS test database that is isolated on a private subnet and because executing that stage alone takes nearly 2 hours, you want to run it during the night before developers come into the London office in the morning. How might you go about this?

1. Amazon VPC access in your CodeBuild project needs to be enabled. Specify your VPC ID, subnets, and security groups in your build project and set up a rule in Amazon CloudWatch Events to start the pipeline on a schedule. Use the following command for that: aws events put-rule --schedule-expression 'cron(10 3 ? \* MON-FRI \*)' --name IntegrationTests
2. By default, CodeBuild projects can access VPCs in the same account. Use the AWS CLI CodeBuild command 'start-build' with the --repeat=true, --hours=3, --minutes=10 and --frequency=weekdays option.
3. To allow CodeBuild access to your database, you have to specify the VPC ID and the ID of the subnet of your private RDS instance. Then create a CloudWatch events rule to schedule a CodeBuild project build using the following cron expression: 3 10 \* \* ? \*
4. CodeBuild supports assigning elastic IP addresses to the network interfaces that it creates. Therefore, you need to configure your build project with the allocated EIP and VPC details of your RDS instance. Call the AWS CodePipeline 'StartPipelineExecution' API with {'stage': 'Test', 'schedule': '5 23 ? \* SUN-THU \*' }.

Answer: a

EXPLANATION:

Enabling Amazon VPC access in your CodeBuild project requires the ID of the VPC, subnets, and security groups. You cannot use the internet gateway instead of a NAT gateway or a NAT instance because CodeBuild does not support assigning elastic IP addresses to the network interfaces that it creates, and auto-assigning a public IP address is not supported by Amazon EC2 for any network interfaces created outside of Amazon EC2 instance launches. The ordered cron expression fields are: minutes, hours, day of month, month, day of week, year.

Your manager wants to implement a CI/CD pipeline for your new cloud-native project using AWS services, and would like you to ensure that it is performing the best automated tests that it can. He would like fast and cheap testing, where bugs can be fixed quickly. He suggests starting with individual units of your software and wants you to test each one, ensuring they perform how they are designed to perform. What kind of tests do you suggest implementing, and what part of your CICD pipeline will you implement them with?

1. Start by creating a code repository in AWS CodeCommit for your software team to perform source-control. Build some compliance tests for current code base and ensure that your developers produce component tests as early as possible for software as it is built. Implement the execution of unit testing using AWS CodeBuild
2. Start by creating a code repository in AWS CodeCommit for your software team to perform source-control. Build some unit tests for the existing code base and ensure that your developers produce component tests as early as possible for software as it is built. Implement the execution of unit testing using AWS CodeCommit
3. Start by creating a code repository in AWS CodeCommit for your software team to perform source-control. Build source tests for current code base and ensure that your developers produce source tests as early as possible for software as it is built. Implement the execution of unit testing using AWS CodeCommit
4. Start by creating a code repository in AWS CodeCommit for your software team to perform source-control. Build some unit tests for the existing code base and ensure that your developers produce unit tests as early as possible for software as it is built. Implement the execution of unit testing using AWS CodeBuild

Answer: d

EXPLANATION:

Unit tests are built to test individual units of your software and quickly identify bugs. These can be implemented with CodeBuild.

Your current application uses an Aurora database, however the speeds aren't as blazingly fast as you would like for your bleeding edge website. You are too deep into developing your application to be able to change the database you are using or to implement faster or larger read replicas. Your application is read-heavy, and the team has identified there are a number of common queries which take a long time to be returned from Aurora. What recommendations would you make to the development team in order to increase your read performance and optimise the application to use Aurora?

1. You should tell your team to optimise their application by ensuring that where possible they engineer the application to make a large number of concurrent queries and transactions as this is one area that Aurora is optimised for. In addition they should Implement ElastiCache between your application and the database.
2. You should tell your team to optimise their application by ensuring that where possible they engineer the application to make a large number of concurrent queries and transactions as this is one area that Aurora is optimised for. In addition they should switch to Aurora Serverless.
3. You should tell your team to optimise their application by ensuring that where possible they engineer the application to make a large number of concurrent queries and transactions as this is one area that Aurora is optimised for. In addition they should create a read-optimised replica and redirect all application reads to that endpoint.
4. You should tell your team to optimise their application by ensuring that where possible they engineer the application to make a large number of concurrent queries and transactions as this is one area that Aurora is optimised for. In addition they should increase the database instance size to a low-latency instance.

Answer: a

EXPLANATION:

ElastiCache will cache common queries by holding them in memory instead of on disk, and will speed up your application considerably

Your team is excited about embarking upon their first greenfield AWS project after months of lift-and-shift migration from your old datacenter into AWS. This will be your first true infrastructure as code project. Your team consists of eight people who all will be making changes to infrastructure over time. You would like to use native AWS tooling for writing the infrastructure templates however you are concerned about how team member changes will actually affect the resources you have running already. You don't want to accidentally destroy important AWS resources due to a developer or engineer changing a CloudFormation property whose update property requires replacement. How can you ensure that engineers are aware of the impact of their updates before they implement them, and protect important stateful resources such as EBS volumes and RDS instances against accidental deletion?

1. Get the team to use AWS CloudFormation to build the infrastructure as code. Mandate that the team include the UpdateReplacePolicy property on important resources such as RDS. Use an UpdateReplacePolicy of Retain in order to retain the old physical resource or snapshot in the AWS account, but remove it from AWS CloudFormation's scope. When your team wants to update the infrastructure templates, advise they use CloudFormation Modify Set to compare their old and new templates. This will will allow them to preview the effects of their changes to see whether resources will be replaced by the CloudFormation service.
2. Get the team to use AWS CloudFormation to build the infrastructure as code. Mandate that the team include the UpdateReplacePolicy property on important resources such as RDS. Use an UpdateReplacePolicy of Retain in order to retain the old physical resource or snapshot in the AWS account, but remove it from AWS CloudFormation's scope. When your team wants to update the infrastructure templates, advise they use CloudFormation Drift Check to compare their old and new templates. This will will allow them to preview the effects of their changes to see whether resources will be replaced by the CloudFormation service.
3. Get the team to use AWS CloudFormation to build the infrastructure as code. Mandate that the team include the UpdateReplacePolicy property on important resources such as RDS. Use an UpdateReplacePolicy of Retain in order to retain the old physical resource or snapshot in the AWS account, but remove it from AWS CloudFormation's scope. When your team wants to update the infrastructure templates, advise they first create a CloudFormation Change Set. This will will allow them to preview the effects of their changes to see whether resources will be replaced by the CloudFormation service.
4. Get the team to use AWS CloudFormation to build the infrastructure as code. Mandate that the team include the UpdateReplacePolicy property on important resources such as RDS. Use an UpdateReplacePolicy of Retain in order to retain the old physical resource or snapshot in the AWS account, but remove it from AWS CloudFormation's scope. When your team wants to update the infrastructure templates, advise they use diff in the Linux terminal to compare their old and new templates. This will will allow them to preview the effects of their changes to see whether resources will be replaced by the CloudFormation service.

Answer: c

EXPLANATION:

CloudFormation Change sets will let you submit your modified stack template, it will compare it for you and show you which stack settings and resources will change. You can then execute that change set if you are happy with the changes that will occur.

Your current workload with DynamoDB is extremely latency bound, and you need it to be as fast as possible. You do not have time to look at other AWS services but instead have been instructed to use features and configuration changes of the services you are currently using. What do you do?

1. Minimise the number of separate attributes in your table by grouping for example into JSON blobs to optimise scan times. Implement a DAX cluster for caching of common queries and to reduce latency on common queries. Consider replacing certain sequential slow scan operations with parallel scans of separate segments of the table, and suggest the development team use the AWS SDK to make available X-Ray tracing of their DynamoDB calls.
2. Minimise the number of separate attributes in your table by grouping for example into JSON blobs to optimise scan times. Implement Elasticache Memcached for caching of common queries and to reduce latency on common queries. Consider replacing certain sequential slow scan operations with parallel scans of separate segments of the table, and suggest the development team use the AWS SDK to make available X-Ray tracing of their DynamoDB calls.
3. Minimise the number of separate attributes in your table by grouping for example into JSON blobs to optimise scan times. Implement Elasticache Redis for caching of common queries and to reduce latency on common queries. Consider replacing certain sequential slow scan operations with parallel scans of separate segments of the table, and suggest the development team use the AWS Insights to make available Insights tracing of their DynamoDB calls.
4. Minimise the number of separate attributes in your table by grouping for example into JSON blobs to optimise scan times. Implement DynamoDB Cached for caching of common queries and to reduce latency on common queries. Consider replacing certain sequential slow scan operations with parallel scans of separate segments of the table, and suggest the development team use the AWS SDK to make available X-Ray tracing of their DynamoDB calls.
5. Minimise the number of separate attributes in your table by grouping for example into JSON blobs to optimise scan times. Implement a DAX cluster for caching of common queries and to reduce latency on common queries. Consider replacing certain sequential slow scan operations with parallel scans of separate segments of the table, and suggest the development team use the AWS Insights to make available Insights tracing of their DynamoDB calls.

Answer: a

EXPLANATION:

Implementing a DAX cluster is the ideal solution here. It meets the requirement of using the existing DynamoDB service feature, while having the ability to reduce latency from milliseconds to microseconds. DynamoDB Scan times can also be optimised by reducing the number of attributes in your table and grouping attributes as JSON blobs within a single attribute.

Your application has a multi-region architecture and database reading and writing is becoming an issue. You are currently storing flat file keypairs on a shared EFS volume across all of your application servers but it is simply too slow to handle the growth your company is experiencing. Additionally, latency of static files delivered to your customers from S3 has been noted as an issue. Which solution will be not only fast but scalable as you move forward?

1. Utilise Amazon CloudFront to optimise delivery of your static S3 content to your users, and use a multi-region read-replica RDS configuration to create a multi-master, cross-region data store for your application's back-end. DynamoDB streams will propagate changes between the replicas so that users will have high-performant and consistent application experience regardless of from where they access your services.
2. Utilise Amazon CloudFront to optimise delivery of your static S3 content to your users, and use a multi-region write replica database for your application's back-end. DynamoDB streams will propagate changes between the replicas so that users will have high-performant and consistent application experience regardless of from where they access your services.
3. Utilise Amazon CloudFront to optimise delivery of your static S3 content to your users, and use multi-region rw-replica RDS configuration to create a multi-master, cross-region data store for your application's back-end. DynamoDB streams will propagate changes between the replicas so that users will have high-performant and consistent application experience regardless of from where they access your services.
4. Utilise Amazon CloudFront to optimise delivery of your static S3 content to your users, and use Amazon DynamoDB Global Tables to create a multi-master, multi-region data store for your application's back-end. DynamoDB streams will propagate changes between the replicas so that users will have high-performant and consistent application experience regardless of from where they access your services.

Answer: d

EXPLANATION:

DynamoDB is the only option that supports multi-region replication and multi-master writes, and it does this using Global Tabl

You work for a company who sells various branded products via their popular Website. Every time a customer completes a purchase, streaming data is immediately sent back to your endpoint and is placed onto a Kinesis Data Stream. Most of the year, traffic remains static and no scaling of the Stream is necessary. However, during the Black Friday period, scaling is required and this is accomplished by increasing the KCL instances and also re-sharding the Kinesis Stream. This year, when sharding some of the streams, it was noticed that an extra shard was left after the operation finished, and this meant that if an even number of shards was requested, the number of open shards became odd. You have been asked to troubleshoot the issue and find the cause and resolution from the list below.

1. This occurs when the width of a shard is very small in size in relation to other shards in the stream. This is resolved by merging with any adjacent shard.
2. This occurs when an unhandled exception is thrown in the KCL from the processRecords code, and a record is skipped. This is resolved by handling all exceptions within processRecords appropriately.
3. This occurs if a shard iterator expires immediately before you can use it. This may be resolved by ensuring the DynamoDB storing the lease information has enough capacity to store this data, by increasing the write capacity assigned to the shard table.
4. This occurs when a producer application writes to an encrypted stream without permissions on the KMS master key. This is resolved by assigning the correct permissions to an application to access a KMS key.

Answer: a

EXPLANATION:

This is an issue which occurs from time to time when re-sharding. The difference between the StartingHashKey and EndingHashKey values is normally large (depending on the number of shards you have in the stream). Occasionally, the difference can be a very low value such as 1 and this causes the UpdateShardCount to end up with an extra shard. This can usually be resolved by finding the ShardID with next adjacent Hash Key value, and merging the small shard with the larger shard.

AWS Certified Developer - Associate

AWS Certified DevOps Engineer – Professional

A close up of a telephone

Description automatically generated