

ULTIMATE Q/A AWS DOP-C01

DevOps Engineer Professional

Questions

Luca Cesarano
<https://lucacesarano.com>

Table of Contents

Table of Contents	2
Questions 100-149	3
Questions 150-199	22
Questions 200-249	40
Questions 250-299	59
Questions 300-349	75
Questions 350-399	94
Questions 400-449	113
Questions 450-499	132
Questions 500-END	150

Questions 100-149

Question #100

A DevOps Engineer responsible for a data analytics program must gather all application and Linux system logs from Amazon EC2 instances prior to termination for auditing, analytics, and troubleshooting reasons. The organization operates 10,000 instances in an Auto Scaling group on average. The business wants the ability to locate logs fast using instance IDs and date ranges.

Which approach is the MOST cost-effective?

- A. Create an EC2 Instance-terminate Lifecycle Action on the group, write a termination script for pushing logs into Amazon S3, and trigger an AWS Lambda function based on S3 PUT to create a catalog of log files in an Amazon DynamoDB table with the primary key being Instance ID and sort key being Instance Termination Date.
- B. Create an EC2 Instance-terminate Lifecycle Action on the group, write a termination script for pushing logs into Amazon CloudWatch Logs, create a CloudWatch Events rule to trigger an AWS Lambda function to create a catalog of log files in an Amazon DynamoDB table with the primary key being Instance ID and sort key being Instance Termination Date.
- **C. Create an EC2 Instance-terminate Lifecycle Action on the group, create an Amazon CloudWatch Events rule based on it to trigger an AWS Lambda function for storing the logs in Amazon S3, and create a catalog of log files in an Amazon DynamoDB table with the primary key being Instance ID and sort key being Instance Termination Date.**
- D. Create an EC2 Instance-terminate Lifecycle Action on the group, push the logs into Amazon Kinesis Data Firehose, and select Amazon ES as the destination for providing storage and search capability.

Commented [LC1]: <https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html>

You can only have 3 customer actions for the autoscaling group lifecycle hook. A, B, D. not actions listed in the link. The script is only for start script. So, C is the right answer

Question #101

How long can messages be saved in SQS?

- **A. 14 days**
- B. one month
- C. 4 days
- D. 7 days

Commented [LC2]:

Question #102

What information is captured by Amazon Inspector's interaction with CloudTrail for the List* and Describe* APIs?

- A. None. Amazon Inspector is an automated service and not monitored by CloudTrail.
- B. Both request and response information are logged.
- **C. Only request information is logged.**
- D. Request information is always logged. Response information is logged only for Completed assessment runs.

Commented [LC3]: For the Amazon Inspector integration with CloudTrail, for the List* and Describe* APIs, only the request information is logged.

Reference:
<https://docs.aws.amazon.com/inspector/latest/userguide/logging-using-cloudtrail.html>

Question #103

Which of the following setup or deployment techniques poses a danger to RDS's security?

- A. Storing SQL function code in plaintext
- B. Non-Multi-AZ RDS instance
- C. Having RDS and EC2 instances exist in the same subnet
- **D. RDS in a public subnet**

Commented [LC4]: Making RDS accessible to the public internet in a public subnet poses a security risk, by making your database directly addressable and spamable. DB instances deployed within a VPC can be configured to be accessible from the Internet or from EC2 instances outside the VPC. If a VPC security group specifies a port access such as TCP port 22, you would not be able to access the DB instance because the firewall for the DB instance provides access only via the IP addresses specified by the DB security groups the instance is a member of and the port defined when the DB instance was created.

Reference:
<http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.RDSSecurityGroups.html>

Question #104

A business uses Amazon EC2 for a variety of workloads. According to company policy, instances must be controlled centrally in order to standardize settings. Standard logging, metrics, security assessments, and weekly patching are all included in these settings.

How is the business going to satisfy these requirements? (Select three.)

- A. Use AWS Config to ensure all EC2 instances are managed by Amazon Inspector.
- B. Use AWS Config to ensure all EC2 instances are managed by AWS Systems Manager.
- C. Use AWS Systems Manager to install and manage Amazon Inspector, Systems Manager Patch Manager, and the Amazon CloudWatch agent on all instances.
- D. Use Amazon Inspector to install and manage AWS Systems Manager, Systems Manager Patch Manager, and the Amazon CloudWatch agent on all instances.
- E. Use AWS Systems Manager maintenance windows with Systems Manager Run Command to schedule Systems Manager Patch Manager tasks. Use the Amazon CloudWatch agent to schedule Amazon Inspector assessment runs.
- F. Use AWS Systems Manager maintenance windows with Systems Manager Run Command to schedule Systems Manager Patch Manager tasks. Use Amazon CloudWatch Events to schedule Amazon Inspector assessment runs.

Commented [LC5]:

Commented [LC6]:

Commented [LC7]:

Question #105 [SKIP]

AWS is experiencing a significant outage. EC2 is unaffected, however your EC2 instance deployment scripts in the area impacted by the outage ceased operating.

What may be the problem?

- A. The AWS Console is down, so your CLI commands do not work.
- B. S3 is unavailable, so you can't create EBS volumes from a snapshot you use to deploy new volumes.
- C. AWS turns off the `<code>DeployCode</code>` API call when there are major outages, to protect from system floods.
- D. None of the other answers make sense. If EC2 is not affected, it must be some other issue.

Question #106 [SKIP]

When you spend \$1000 or more on AWS, you need to be aware. What is the most convenient method for you to see the notification?

- A. AWS CloudWatch Events tied to API calls, when certain thresholds are exceeded, publish to SNS.
- B. Scrape the billing page periodically and pump into Kinesis.
- C. AWS CloudWatch Metrics + Billing Alarm + Lambda event subscription. When a threshold is exceeded, email the manager.
- D. Scrape the billing page periodically and publish to SNS.

Question #107

Customers have lately complained that your online application has ceased functioning unexpectedly. The team detected a significant fault in your new Java web application after a thorough dive into your logs. This problem results in a memory leak, which finally crashes the program. Your website is hosted on Amazon EC2 and was created using AWS CloudFormation.

Which strategies should you use to assist in identifying these issues more quickly and resolving the server's unresponsiveness? (Select two.)

- A. Update your AWS CloudFormation configuration and enable a CustomResource that uses `cfnsignal` to detect memory leaks.
- B. Update your CloudWatch metric granularity config for all Amazon EC2 memory metrics to support five-second granularity. Create a CloudWatch alarm that triggers an Amazon SNS notification to page your team when the application memory becomes too large.
- C. Update your AWS CloudFormation configuration to take advantage of Auto Scaling groups. Configure an Auto Scaling group policy to trigger off your custom CloudWatch metrics.
- D. Create a custom CloudWatch metric that you push your JVM memory usage to. Create a Cloudwatch alarm that triggers an Amazon SNS notification to page your team when the application memory usage becomes too large.
- E. Update your AWS CloudFormation configuration to take advantage of CloudWatch metrics Agent. Configure the CloudWatch Metrics Agent to monitor memory usage and trigger an Amazon SNS alarm.

Question #108

A business is deploying an application through AWS. The development team's deployments must be automated. After building the application using AWS CodeBuild, the team established an AWS CodePipeline pipeline to deploy it to Amazon EC2 instances using AWS CodeDeploy. The team want to include automated testing into the pipeline in order to validate the application's health prior to deploying it to the EC2 instances. Additionally, even if the tests are successful, the team needs human permission before the application can be deployed. Testing and approval must be carried out at the lowest possible cost and with the simplest management solution possible.

Which solution will satisfy these criteria?

- A. Create a manual approval action after the build action of the pipeline. Use Amazon SNS to inform the team of the stage being triggered. Next, add a test action using CodeBuild to perform the required tests. At the end of the pipeline, add a deploy action to deploy the application to the next stage.
- B. Create a test action after the CodeBuild build of the pipeline. Configure the action to use CodeBuild to perform the required test. If these tests are successful, mark the action as successful. Add a manual approval action that uses Amazon SNS to notify the team, and add a deploy action to deploy the application to the next stage.
- C. Create a new pipeline that uses a source action that gets the code from the same repository as the first pipeline. Add a deploy action to deploy the code to a test environment. Use a test action using AWS Lambda to test the deployment. Add a manual approval action by using Amazon SNS to notify the team, and add a deploy action to deploy the application to the next stage.
- D. Create a test action after the build action. Use a Jenkins server on Amazon EC2 to perform the required tests and mark the action as successful if the tests pass. Create a manual approval action that uses Amazon SQS to notify the team and add a deploy action to deploy the application to the next stage.

Commented [LC8]: <https://docs.aws.amazon.com/codebuild/latest/userguide/how-to-create-pipeline-add-test.html>

Question #109

A US-based internet retailer wants to expand into Europe and Asia over the next six months. Currently, the company's product is hosted on Amazon EC2 instances behind an Application Load Balancer. The instances are distributed across several Availability Zones through an Amazon EC2 Auto Scaling group. All data is stored in a single instance of the Amazon Aurora database. When a product is launched in many countries, the corporation desires a single product catalog across all regions, yet customer information and purchases must be retained in each zone for compliance reasons.

How can the organization achieve these criteria with the fewest application modifications possible?

- A. Use Amazon Redshift for the product catalog and Amazon DynamoDB tables for the customer information and purchases.
- B. Use Amazon DynamoDB global tables for the product catalog and regional tables for the customer information and purchases.
- C. Use Aurora with read replicas for the product catalog and additional local Aurora instances in each region for the customer information and purchases.
- D. Use Aurora for the product catalog and Amazon DynamoDB global tables for the customer information and purchases.

Commented [LC9]: It's C - all the others require application changes to accommodate a different DB, which is undesirable since the question is asking for minimal application changes. It's worth adding that in the real world things are often much more complicated than this, so you'd want to revisit the business requirements, validate them, forecast future requirements and make a decision. Having regions of US, Asia and Europe is a gross simplification, because what you really need to consider is the data protection regulations in specific jurisdictions, not continents. e.g. Indonesia is in Asia and up until Oct 2019 required customer data to be held in-country, but there is no AWS data centre there. What do you do?

Question #110

Changes to the contents of objects within production Amazon S3 buckets that contain encrypted secrets should be done only by a trusted group of administrators, according to Information Security policy.

How should a DevOps Engineer implement this need using real-time, automated checks?

- A. Create an AWS Lambda function that is triggered by Amazon S3 data events for object changes and that also checks the IAM user's membership in an administrator's IAM role.
- B. Create a periodic AWS Config rule to query Amazon S3 Logs for changes and to check the IAM user's membership in an administrator's IAM role.
- C. Create a metrics filter for Amazon CloudWatch logs to check for Amazon S3 bucket-level permission changes and to check the IAM user's membership in an administrator's IAM role.
- D. Create a periodic AWS Config rule to query AWS CloudTrail logs for changes to the Amazon S3 bucket-level permissions and to check the IAM user's membership in an administrator's IAM role.

Commented [LC10]: Answer is A:
You can send S3 events to lambda:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html>

Answer D is a periodic answer and you want it to be as fast as possible which can happen with events and lambda's.

Question #111

A business is using Amazon EC2 instances to launch a new application. The organization wants to keep a consolidated application and Amazon API logs accessible through a single tool or service.

Which solution will satisfy these criteria?

- A. Use the Amazon CloudWatch agent to send logs from the Amazon EC2 instances to CloudWatch. Configure AWS CloudTrail to deliver the API logs to CloudWatch and use Amazon Athena to query both log sets in CloudWatch.
- B. Use the Amazon CloudWatch agent to send logs from the Amazon EC2 instances to CloudWatch. Configure an Amazon Kinesis Data Firehose log group subscription to send those logs to Amazon S3. Use AWS CloudTrail to deliver the API logs to Amazon S3. Use Amazon Athena to query both log sets in Amazon S3.
- C. Use the Amazon CloudWatch agent to send logs from the Amazon EC2 instances to Amazon Kinesis. Configure AWS CloudTrail to deliver the API logs to Kinesis. Use Amazon to load the data into Amazon Redshift and use Amazon Redshift to query both log sets.
- D. Use the Amazon CloudWatch agent to send logs from the Amazon EC2 instances to Amazon S3. Use Amazon CloudTrail to deliver the API logs to Amazon S3 and use Amazon Redshift to query both log sets in Amazon S3.

Commented [LC11]: Before Nov 2019 the answer would have been B.
I suspect this question is older than that.
After Nov 2019 the answer can be A, because of

<https://aws.amazon.com/blogs/big-data/query-any-data-source-with-amazon-athenas-new-federated-query/>

Question #112

A business must get user approval to a privacy agreement. With a user base of between 20 and 30 million, an application is deployed in six AWS Regions, two in North America, two in Europe, and two in Asia. The organization must read and write data pertaining to each user's answer, as well as ensuring that the replies are accessible in all six Regions.

Which solution satisfies these objectives while keeping latency to a minimum?

- A. Implement Amazon Aurora Global Database in each of the six Regions.
- B. Implement Amazon DocumentDB (with MongoDB compatibility) in each of the six Regions.
- C. Implement Amazon DynamoDB global tables in each of the six Regions.
- D. Implement Amazon ElastiCache for Redis replication group in each of the six Regions.

Commented [LC12]: I'll go with C

Reference:

<https://aws.amazon.com/blogs/database/how-to-use-amazon-dynamodb-global-tables-to-power-multiregion-architecture>

Question #113

The staging website of a business is hosted on an Amazon EC2 instance that is backed up by Amazon EBS storage. The organization wishes to recover fast and with little data loss if the EC2 instance's network connection or power fails.

Which solution will satisfy these criteria?

- A. Add the instance to an EC2 Auto Scaling group with the minimum, maximum, and desired capacity set to 1.
- B. Add the instance to an EC2 Auto Scaling group with a lifecycle hook to detach the EBS volume when the EC2 instance shuts down or terminates.
- C. Create an Amazon CloudWatch alarm for the StatusCheckFailed_System metric and select the EC2 action to recover the instance.
- D. Create an Amazon CloudWatch alarm for the StatusCheckFailed_Instance metric and select the EC2 action to reboot the instance.

Commented [LC13]: <https://docs.aws.amazon.com/AWSAmazonCloudWatch/latest/monitoring/UsingAlarmActions.html#AddingRecoverActions>

Question #114

A business has an application that makes use of an Amazon Aurora Multi-AZ DB cluster that supports MySQL. For catastrophe recovery reasons, a cross-Region read replica has been built. A DevOps engineer wishes to automate the promotion of the replica to primary status in the case of a database loss.

Which approach is most likely to do this?

- A. Configure a latency-based Amazon Route 53 CNAME with health checks so it points to both the primary and replica endpoints. Subscribe an Amazon SNS topic to Amazon RDS failure notifications from AWS CloudTrail and use that topic to trigger an AWS Lambda function that will promote the replica instance as the master.
- B. Create an Aurora custom endpoint to point to the primary database instance. Configure the application to use this endpoint. Configure AWS CloudTrail to run an AWS Lambda function to promote the replica instance and modify the custom endpoint to point to the newly promoted instance.
- C. Create an AWS Lambda function to modify the application's AWS Cloud Formation template to promote the replica, apply the template to update the stack, and point the application to the newly promoted instance. Create an Amazon CloudWatch alarm to trigger this Lambda function after the failure event occurs.
- **D. Store the Aurora endpoint in AWS Systems Manager Parameter Store. Create an Amazon EventBridge (Amazon CloudWatch Events) event that detects the database failure and runs an AWS Lambda function to promote the replica instance and update the endpoint URL stored in AWS Systems Manager Parameter Store. Code the application to reload the endpoint from Parameter Store if a database connection fails.**

Commented [LC14]: <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.Endpoints.html#Aurora.Overview.Endpoints.HA>

Question #115

Which of the following strategies offers the quickest rollback timeframes feasible in the case of a botched deployment?

- A. Rolling; Immutable
- B. Rolling; Mutable
- C. Canary or A/B
- **D. Blue-Green**

Commented [LC15]: AWS specifically recommends Blue-Green for super-fast, zero-downtime deploys - and thus rollbacks, which are redeploying old code. You use various strategies to migrate the traffic from your current application stack (blue) to a new version of the application (green). This is a popular technique for deploying applications with zero downtime.

Reference:
<https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-onaws.pdf>

Question #116

A configuration file is required to execute an application on a collection of Amazon EC2 machines in an Auto Scaling group. AWS CloudFormation is used to build and manage the instances. A DevOps engineer wants to ensure that instances are launched with the most recent configuration file and that changes to the configuration file are reflected on all instances with the least amount of delay possible when the CloudFormation template is changed. According to company policy, application configuration files must be kept in source control alongside AWS infrastructure configuration files.

Which approach is most likely to do this?

- A. In the CloudFormation template, add an AWS Config rule. Place the configuration file content in the rule's InputParameters property, and set the Scope property to the EC2 Auto Scaling group. Add an AWS Systems Manager Resource Data Sync resource to the template to poll for updates to the configuration.
- B. In the CloudFormation template, add an EC2 launch template resource. Place the configuration file content in the launch template. Configure the cfn-init script to run when the instance is launched, and configure the cfn-hup script to poll for updates to the configuration.
- C. In the CloudFormation template, add an EC2 launch template resource. Place the configuration file content in the launch template. Add an AWS Systems Manager Resource Data Sync resource to the template to poll for updates to the configuration.
- **D. In the CloudFormation template, add Cloud Formation init metadata. Place the configuration file content in the metadata. Configure the cfn-init script to run when the instance is launched, and configure the cfn-hup script to poll for updates to the configuration.**

Commented [LC16]: I'll go with D

Use the `AWS::CloudFormation::Init` type to include metadata on an Amazon EC2 instance for the cfn-init helper script. If your template calls the cfn-init script, the script looks for resource metadata rooted in the `AWS::CloudFormation::Init` metadata key.

Reference:
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-init.html>

Question #117

Your team has established via meter analysis that the company's website is having longer response times than intended during peak hours. You presently depend on Auto Scaling to ensure that your environment is scaled appropriately during peak periods.

How can you optimize your Auto Scaling strategy to decrease this lag time? (Select two.)

- A. Push custom metrics to CloudWatch to monitor your CPU and network bandwidth from your servers, which will allow your Auto Scaling policy to have better fine-grain insight.
- B. Increase your Auto Scaling group's number of max servers.
- C. Create a script that runs and monitors your servers; when it detects an anomaly in load, it posts to an Amazon SNS topic that triggers Elastic Load Balancing to add more servers to the load balancer.
- D. Push custom metrics to CloudWatch for your application that include more detailed information about your web application, such as how many requests it is handling and how many are waiting to be processed.
- E. Update the CloudWatch metric used for your Auto Scaling policy, and enable sub-minute granularity to allow auto scaling to trigger faster.

Question #118

A business uses AWS Lambda to execute microservices that read data from Amazon DynamoDB. Developers manually deploy Lambda code after successful testing. The organization now requires automated testing and deployments that operate in the cloud. Additionally, traffic should be slowly transferred to the updated versions of each microservice upon deployment.

Which solution satisfies all needs while maintaining the highest developer velocity possible?

- A. Create an AWS CodePipeline configuration and set up a post-commit hook to trigger the pipeline after tests have passed. Use AWS CodeDeploy and create a Canary deployment configuration that specifies the percentage of traffic and interval.
- B. Create an AWS CodeBuild configuration that triggers when the test code is pushed. Use AWS CloudFormation to trigger an AWS CodePipeline configuration that deploys the new Lambda versions and specifies the traffic shift percentage and interval.
- C. Create an AWS CodePipeline configuration and set up the source code step to trigger when code is pushed. Set up the build step to use AWS CodeBuild to run the tests. Set up an AWS CodeDeploy configuration to deploy, then select the `CodeDeployDefault.LambdaLinear10PercentEvery3Minutes` option.
- D. Use the AWS CLI to set up a post-commit hook that uploads the code to an Amazon S3 bucket after tests have passed. Set up an S3 event trigger that runs a Lambda function that deploys the new version. Use an interval in the Lambda function to deploy the code over time at the required percentage.

Commented [LC17]: A - this mentions after tests have passed. Tests need to be automated.
B - do we really need AWS CloudFormation?
C - looks correct
D - s3 / lambda/cli may not be needed here.

Question #119

Before migrating to AWS, a corporation must guarantee that an application running on Amazon Linux behaves consistently across its corporate ecosystem. The organization already has an automated server build mechanism in place that utilizes VMware. The objective is to show the application's functionality and requirements on the new target operating system. The DevOps Engineer must develop a server image using the current corporate server pipeline and virtualization technologies. The server image will be validated on-premises to ensure that it closely resembles the build on Amazon EC2.

How is this possible?

- A. Download and integrate the latest ISO of CentOS 7 and execute the application deployment on the resulting server.
- B. Launch an Amazon Linux AMI using an AWS OpsWorks deployment agent onto the on-premises infrastructure, then execute the application deployment.
- C. Build an EC2 instance with the latest Amazon Linux operating system, and use the AWS Import/Export service to export the EC2 image to a VMware ISO in Amazon S3. Then import the resulting ISO onto the on-premises system.
- D. Download and integrate the latest ISO of Amazon Linux 2 and execute the application deployment on the resulting server. Confirm that operating system testing results are consistent with EC2 operating system behavior.

Commented [LC18]: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/amazon-linux-2-virtual-machine.html>

Question #120

You're developing an application that will keep incredibly private financial data. At rest and in transit, every data in the system must be encrypted.

Which of the following is a violation of our policy?

- A. ELB SSL termination.
- B. ELB Using Proxy Protocol v1.
- C. CloudFront Viewer Protocol Policy set to HTTPS redirection.
- D. Telling S3 to use AES256 on the server-side.

Commented [LC19]: Terminating SSL terminates the security of a connection over HTTP, removing the S for "Secure" in HTTPS. This violates the "encryption in transit" requirement in the scenario.

Reference:

<http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elb-listener-config.htm>

Question #121

A DevOps Engineer is tasked with the responsibility of designing and implementing a backup strategy for Amazon EFS. The following criteria are made of the Engineer:

- ☞ Backups should be scheduled.
- ☞ If the backup window expires, the backup should be terminated.
- ☞ If the backup completes before the backup window closes, the backup should be terminated.
- ☞ Backup logs should be preserved for analytical purposes.
- ☞ The design should be fault-tolerant and highly available.
- ☞ Backup metadata should be sent to administrators.

Which design will satisfy these criteria?

- A. Use AWS Lambda with an Amazon CloudWatch Events rule for scheduling the start/stop of backup activity. Run backup scripts on Amazon EC2 in an Auto Scaling group. Use Auto Scaling lifecycle hooks and the SSM Run Command on EC2 for uploading backup logs to Amazon S3. Use Amazon SNS to notify administrators with backup activity metadata.
- B. Use Amazon SWF with an Amazon CloudWatch Events rule for scheduling the start/stop of backup activity. Run backup scripts on Amazon EC2 in an Auto Scaling group. Use Auto Scaling lifecycle hooks and the SSM Run Command on EC2 for uploading backup logs to Amazon Redshift. Use CloudWatch Alarms to notify administrators with backup activity metadata.
- C. Use AWS Data Pipeline with an Amazon CloudWatch Events rule for scheduling the start/stop of backup activity. Run backup scripts on Amazon EC2 in a single Availability Zone. Use Auto Scaling lifecycle hooks and the SSM Run Command on EC2 for uploading the backup logs to Amazon RDS. Use Amazon SNS to notify administrators with backup activity metadata.
- D. Use AWS CodePipeline with an Amazon CloudWatch Events rule for scheduling the start/stop of backup activity. Run backup scripts on Amazon EC2 in a single Availability Zone. Use Auto Scaling lifecycle hooks and the SSM Run Command on Amazon EC2 for uploading backup logs to Amazon S3. Use Amazon SES to notify admins with backup activity metadata.

Commented [LC20]:

Question #122

In a development environment, a corporation operates a database on a single Amazon EC2 instance. Separate Amazon EBS volumes are used to store the data, which are tied to the EC2 instance. A 53-mile Amazon Route 53 record pointing to the EC2 instance has been generated and configured. The organization wishes to automate database instance recovery in the event that an instance or Availability Zone (AZ) fails. Additionally, the corporation wishes to keep its expenses low. RTO is four hours while RPO is twelve hours.

Which DevOps solution should a DevOps Engineer adopt in order to satisfy these requirements?

- A. Run the database in an Auto Scaling group with a minimum and maximum instance count of 1 in multiple AZs. Add a lifecycle hook to the Auto Scaling group and define an Amazon CloudWatch Events rule that is triggered when a lifecycle event occurs. Have the CloudWatch Events rule invoke an AWS Lambda function to detach or attach the Amazon EBS data volumes from the EC2 instance based on the event. Configure the EC2 instance UserData to mount the data volumes (retry on failure with a short delay), then start the database and update the Route 53 record.
- B. Run the database on two separate EC2 instances in different AZs with one active and the other as a standby. Attach the data volumes to the active instance. Configure an Amazon CloudWatch Events rule to invoke an AWS Lambda function on EC2 instance termination. The Lambda function launches a replacement EC2 instance. If the terminated instance was the active node, then the function attaches the data volumes to the standby node. Start the database and update the Route 53 record.
- C. Run the database in an Auto Scaling group with a minimum and maximum instance count of 1 in multiple AZs. Create an AWS Lambda function that is triggered by a scheduled Amazon CloudWatch Events rule every 4 hours to take a snapshot of the data volume and apply a tag. Have the instance UserData get the latest snapshot, create a new volume from it, and attach and mount the volume. Then start the database and update the Route 53 record.
- D. Run the database on two separate EC2 instances in different AZs. Configure one of the instances as a master and the other as a standby. Set up replication between the master and standby instances. Point the Route 53 record to the master. Configure an Amazon CloudWatch Events rule to invoke an AWS Lambda function upon the EC2 instance termination. The Lambda function launches a replacement EC2 instance. If the terminated instance was the active node, the function promotes the standby to master and points the Route 53 record to it.

Commented [LC21]:

Question #123 [SKIP]

If Erin has three clusters of different server kinds, all of which are controlled by Ansible, and she wants to supply each cluster with the proper NTP server addresses,






What is the best technique for Erin to manage this with Ansible?

- A. Write a task that scans the network in the target hosts' region for the NTP server, register the resulting address so that the next task can write the NTP configuration.
- B. Break down the hosts by region in the Ansible inventory file and assign an inventory group variable the NTP address value for the respective region. The playbook can contain just the single play referencing the NTP variable from the inventory.
- C. Create a playbook for each different region and store the NTP address in a variable in the play in the event the NTP server changes.
- D. Create three plays, each one has the hosts for their respective regions and set the NTP server address in each task.

Question #124

A business wants to use AWS CloudFormation to automatically recreate its infrastructure as part of its quality assurance (QA) workflow. Each QA run requires the creation of a new VPC in a single account, the deployment of resources into the VPC, and the execution of tests against the newly formed infrastructure. To enable centralized logging, the corporate policy requires that all VPCs be peering with a central management VPC. The business already has CloudFormation templates in place for deploying its VPC and related resources.

Which combination of actions will accomplish the task in an automated and repeatable manner? (Select two.)

- A. Create an AWS Lambda function that is invoked by an Amazon CloudWatch Events rule when a `CreateVpcPeeringConnection` API call is made. The Lambda function should check the source of the peering request, accept the request, and update the route tables for the management VPC to allow traffic to go over the peering connection.
- B. In the CloudFormation template:  Invoke a custom resource to generate unique VPC CIDR ranges for the VPC and subnets.  Create a peering connection to the management VPC.  Update route tables to allow traffic to the management VPC.
- C. In the CloudFormation template:  Use the `Fn::Cidr` function to allocate an unused CIDR range for the VPC and subnets.  Create a peering connection to the management VPC. Update route tables to allow traffic to the management VPC. ■
- D. Modify the CloudFormation template to include a `mappings` object that includes a list of /16 CIDR ranges for each account where the stack will be deployed.
- E. Use CloudFormation StackSets to deploy the VPC and associated resources to multiple AWS accounts using a custom resource to allocate unique CIDR ranges. Create peering connections from each VPC to the central management VPC and accept those connections in the management VPC.

Commented [LC22]:

Commented [LC23]:

Question #125

A federal organization uses an encrypted Amazon S3 bucket to store extremely secret material. The agency has setup federated access and restricted access to this bucket to a single on-premises Active Directory user group. The agency wishes to keep audit records and to automatically identify and undo any unintended modifications administrators make to the IAM rules used to provide this limited federated access.

Which of the following choices satisfies these criteria the FASTEST?

- A. Configure an Amazon CloudWatch Events Event Bus on an AWS CloudTrail API for triggering the AWS Lambda function that detects and reverts the change.
- B. Configure an AWS Config rule to detect the configuration change and execute an AWS Lambda function to revert the change.
- C. Schedule an AWS Lambda function that will scan the IAM policy attached to the federated access role for detecting and reverting any changes.
- D. Restrict administrators in the on-premises Active Directory from changing the IAM policies.

Commented [LC24]:

Question #126

A DevOps engineer used AWS CodePipeline to automate the following steps:

- ⇒ AWS CodeBuild builds and tests the deployment artifact.
- ⇒ AWS CodeDeploy distributes the web service to Amazon EC2 instances in the staging environment using an AWS CodeDeploy deployment group.

The web service is deployed to EC2 instances in the production environment using a CodeDeploy deployment group.

Before the build artifact is deployed to the production environment, the quality assurance (QA) team has requested authorization to examine it. The QA team wishes to conduct some manual tests using an internal automated penetration testing tool (invoked through a REST API request).

Which action combination will satisfy this request? (Select two.)

- **A. Insert a manual approval action between the test and deployment actions of the pipeline.**
- B. Modify the buildspec.yml file for the compilation stage to require manual approval before completion.
- C. Update the CodeDeploy deployment group so it requires manual approval to proceed.
- D. Update the pipeline to directly trigger the REST API for the automated penetration testing tool.
- **E. Update the pipeline to invoke a Lambda function that triggers the REST API for the automated penetration testing tool.**

Commented [LC25]: A and E are correct.
B: there is no approval process in CodeBuild
C: there is no approval process in CodeDeploy (remember, approvals = CodePipeline)
D: CodePipeline does not call rest API endpoint, you need a lambda (so E is correct)

Commented [LC26]:

Question #127

You use AWS EBS to manage a clustered NoSQL database on AWS EC2. You must decrease the delay associated with database response times. Performance, not availability, is the primary consideration. You did not execute the original setup; someone with less AWS understanding did, and you have no way of knowing whether everything was configured correctly.

Which of the following is NOT a potential cause of increasing latency?

- A. The EC2 instances are not EBS Optimized.
- **B. The database and requesting system are both in the wrong Availability Zone.**
- C. The EBS Volumes are not using PIOPS.
- D. The database is not running in a placement group.

Commented [LC27]: For the highest possible performance, all instances in a clustered database like this one should be in a single Availability Zone in a placement group, using EBS optimized instances, and using PIOPS SSD EBS Volumes. The particular Availability Zone the system is running in should not be important, as long as it is the same as the requesting resources.

Reference:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

Question #128

A secondary index in DynamoDB is a data structure that includes a subset of the characteristics from a database, as well as an other key to facilitate additional _____ operations.

- A. None of the above
- **B. Both**
- C. Query
- D. Scan

Commented [LC28]: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SecondaryIndexes.html>

You can create one or more secondary indexes on a table and issue Query or Scan requests against these indexes.

Question #129 [SKIP]

Ansible has many techniques for configuring how and when a job is executed.

Which of the following is an appropriate way for using a loop to control a task?

- A. - with: <value>
- B. - with_items: <value>
- C. - only_when: <conditional>
- D. - items: <value>

Question #130

You must authorize access to your AWS account to a vendor. They must have the ability to view protected communications stored in a private S3 bucket at their convenience. Additionally, they make advantage of AWS.

What is the most effective method for doing this?

- A. Create an IAM User with API Access Keys. Grant the User permissions to access the bucket. Give the vendor the AWS Access Key ID and AWS Secret Access Key for the User.
- B. Create an EC2 Instance Profile on your account. Grant the associated IAM role full access to the bucket. Start an EC2 instance with this Profile and give SSH access to the instance to the vendor.
- C. Create a cross-account IAM Role with permission to access the bucket, and grant permission to use the Role to the vendor AWS account.
- D. Generate a signed S3 PUT URL and a signed S3 GET URL, both with wildcard values and 2 year durations. Pass the URLs to the vendor.

Commented [LC29]:

Question #131 [SKIP]

If Ansible meets a resource that does not fulfill the play's criteria, it modifies the resource accordingly; however, if the resource is already in the intended condition, Ansible does nothing.

Which technique is exemplified by this?

- A. Idempotency
- B. Immutability
- C. Convergence
- D. Infrastructure as Code

Question #132

Which of the following solutions for monitoring your AWS OpsWorks stacks does not natively support AWS OpsWorks?

- A. AWS Config
- B. Amazon CloudWatch Metrics
- C. AWS CloudTrail
- D. Amazon CloudWatch Logs

Commented [LC30]:

Question #133

A DevOps Engineer handles video files for a video production firm. The application is deployed on Amazon EC2 instances and is load balanced through an ELB Application Load Balancer. The instances are distributed across several Availability Zones in an Auto Scaling group. The database is hosted on Amazon RDS PostgreSQL Multi-AZ and the video files are saved in an Amazon S3 bucket. On a normal day, the S3 bucket receives 50 GB of new video. The Engineer is responsible for implementing a multi-region disaster recovery strategy that minimizes data loss and recovery time. AWS CloudFormation has previously been used to define the present application architecture.

Which deployment strategy should the Engineer use in order to satisfy the system's uptime and recovery objectives?

- A. Launch the application from the CloudFormation template in the second region, which sets the capacity of the Auto Scaling group to 1. Create an Amazon RDS read replica in the second region. In the second region, enable cross-region replication between the original S3 bucket and a new S3 bucket. To fail over, promote the read replica as master. Update the CloudFormation stack and increase the capacity of the Auto Scaling group.
- B. Launch the application from the CloudFormation template in the second region, which sets the capacity of the Auto Scaling group to 1. Create a scheduled task to take daily Amazon RDS cross-region snapshots to the second region. In the second region, enable cross-region replication between the original S3 bucket and Amazon Glacier. In a disaster, launch a new application stack in the second region and restore the database from the most recent snapshot.
- C. Launch the application from the CloudFormation template in the second region, which sets the capacity of the Auto Scaling group to 1. Use Amazon CloudWatch Events to schedule a nightly task to take a snapshot of the database, copy the snapshot to the second region, and replace the DB instance in the second region from the snapshot. In the second region, enable cross-region replication between the original S3 bucket and a new S3 bucket. To fail over, increase the capacity of the Auto Scaling group.
- D. Use Amazon CloudWatch Events to schedule a nightly task to take a snapshot of the database and copy the snapshot to the second region. Create an AWS Lambda function that copies each object to a new S3 bucket in the second region in response to S3 event notifications. In the second region, launch the application from the CloudFormation template and restore the database from the most recent snapshot.

Commented [LC31]:

Question #134

Can you put an IfExists condition at the end of a Null condition in an IAM policy?

- A. Yes, you can add an IfExists condition at the end of a Null condition but not in all Regions.
- B. Yes, you can add an IfExists condition at the end of a Null condition depending on the condition.
- C. No, you cannot add an IfExists condition at the end of a Null condition.
- D. Yes, you can add an IfExists condition at the end of a Null condition.

Commented [LC32]: https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html

Question #135

A user is cloning an existing EBS volume to create a new one. The snapshot is 10 GB in size.

Is it possible for the user to construct a 30 GB volume from that snapshot?

- A. Provided the original volume has set the change size attribute to true
- B. Yes
- C. Provided the snapshot has the modify size attribute set as true
- D. No

...IfExists condition operators

You can add IfExists to the end of any condition operator name except the Null condition—for example, StringLikeIfExists. You do this to say "If the policy key is present in the context of the request, process the key as specified in the policy. If the key is not present, evaluate the condition element as true." Other condition elements in the statement can still result in a nonmatch, but not a missing key when checked with ...IfExists.

Commented [LC33]: A user can always create a new EBS volume of a higher size than the original snapshot size. The user cannot create a volume of a lower size. When the new volume is created the size in the instance will be shown as the original size. The user needs to change the size of the device with `resize2fs` or other OS specific commands.

Question #136

Several thousand Amazon EC2 instances are stored in an AWS account for a media client. The client uses a Slack channel to communicate with other members of the team and to provide essential updates. A DevOps Engineer was instructed to forward all alerts from AWS about planned EC2 maintenance to the company's Slack channel.

Which approach should the Engineer use to accomplish this task in the fewest number of steps possible?

- A. Integrate AWS Trusted Advisor with AWS Config. Based on the AWS Config rules created, the AWS Config event can invoke an AWS Lambda function to send notifications to the Slack channel.
- **B. Integrate AWS Personal Health Dashboard with Amazon CloudWatch Events. Based on the CloudWatch Events created, the event can invoke an AWS Lambda function to send notifications to the Slack channel.**
- C. Integrate EC2 events with Amazon CloudWatch monitoring. Based on the CloudWatch Alarm created, the alarm can invoke an AWS Lambda function to send EC2 maintenance notifications to the Slack channel.
- D. Integrate AWS Support with AWS CloudTrail. Based on the CloudTrail lookup event created, the event can invoke an AWS Lambda function to pass EC2 maintenance notifications to the Slack channel.

Commented [LC34]:

Question #137

You must develop a simple, comprehensive check for the overall availability and uptime of your system. Your system exposes itself as an HTTP-based application programming interface (API).

What is the simplest tool on AWS to do this?

- **A. Route53 Health Checks**
- B. CloudWatch Health Checks
- C. AWS ELB Health Checks
- D. EC2 Health Checks

Commented [LC35]: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/welcome-health-checks.html>

Question #138

On AWS, a startup firm is building a web application. It intends to leverage Amazon RDS for persistence and to deploy the application using an Auto Scaling group on Amazon EC2. Additionally, the organization wants to divide development, testing, and production environments.

What is the MOST SECURE method for configuring an application?

- A. Create a property file to include the configuration and the encrypted passwords. Check in the property file to the source repository, package the property file with the application, and deploy the application. Create an environment tag for the EC2 instances and tag the instances respectively. The application will extract the necessary property values based on the environment tag.
- B. Create a property file for each environment to include the environment-specific configuration and an encrypted password. Check in the property files to the source repository. During deployment, use only the environment-specific property file with the application. The application will read the needed property values from the deployed property file.
- C. Create a property file for each environment to include the environment-specific configuration. Create a private Amazon S3 bucket and save the property files in the bucket. Save the passwords in the bucket with AWS KMS encryption. During deployment, the application will read the needed property values from the environment-specific property file in the S3 bucket.
- **D. Create a property file for each environment to include the environment-specific configuration. Create a private Amazon S3 bucket and save the property files in the bucket. Save the encrypted passwords in the AWS Systems Manager Parameter Store. Create an environment tag for the EC2 instances and tag the instances respectively. The application will read the needed property values from the environment-specific property file in the S3 bucket and the parameter store.**

Commented [LC36]:

Question #139

Management observed a rise in the total cost from Amazon after analyzing the previous quarter's monthly payments. After investigating this cost rise, you learn that one of your new services is making several GET Bucket API calls to Amazon S3 in order to create a metadata cache for every items in the applications bucket.

Your supervisor has requested that you devise a new cost-effective method for assisting in the reduction of these new GET Bucket API requests.

Which procedure should you use to assist in cost mitigation?

- A. Update your Amazon S3 buckets' lifecycle policies to automatically push a list of objects to a new bucket, and use this list to view objects associated with the application's bucket.
- B. Create a new DynamoDB table. Use the new DynamoDB table to store all metadata about all objects uploaded to Amazon S3. Any time a new object is uploaded, update the application's internal Amazon S3 object metadata cache from DynamoDB.
- C. Using Amazon SNS, create a notification on any new Amazon S3 objects that automatically updates a new DynamoDB table to store all metadata about the new object. Subscribe the application to the Amazon SNS topic to update its internal Amazon S3 object metadata cache from the DynamoDB table.
- D. Upload all images to Amazon SQS, set up SQS lifecycles to move all images to Amazon S3, and initiate an Amazon SNS notification to your application to update the application's Internal Amazon S3 object metadata cache.
- E. Upload all images to an ElastiCache filecache server. Update your application to now read all file metadata from the ElastiCache filecache server, and configure the ElastiCache policies to push all files to Amazon S3 for long-term storage.

Commented [LC37]: E is the wrong answer. B is the right one. <https://aws.amazon.com/blogs/big-data/building-and-maintaining-an-amazon-s3-metadata-index-without-servers/> S3 is still the most cost-effective and durable way to store large data objects. The problem is searching not storage, so storing the data in ElasticCache does not address the question

Question #140

A web application is operating in an AWS Elastic Beanstalk environment for an e-commerce firm. The average load on Amazon EC2 instances has been raised in recent months to accommodate growing traffic.

The firm wishes to increase the environment's scalability and robustness. The Development team has been tasked with decoupling long-running jobs from their environment, assuming that the tasks may be done asynchronously. These responsibilities may include sending confirmation letters to users once they register on the site and processing photographs or videos. Additionally, some of the web server's present periodic activities should be offloaded.

What is the MOST TIME EFFECTIVE and INTEGRATED method for doing this?

- A. Create an Amazon SQS queue and send the tasks that should be decoupled from the Elastic Beanstalk web server environment to the SQS queue. Create a fleet of EC2 instances under an Auto Scaling group. Use an AMI that contains the application to process the asynchronous tasks, configure the application to listen for messages within the SQS queue, and create periodic tasks by placing those into the cron in the operating system. Create an environment variable within the Elastic Beanstalk environment with a value pointing to the SQS queue endpoint.
- B. Create a second Elastic Beanstalk worker tier environment and deploy the application to process the asynchronous tasks there. Send the tasks that should be decoupled from the original Elastic Beanstalk web server environment to the auto-generated Amazon SQS queue by the Elastic Beanstalk worker environment. Place a cron.yaml file within the root of the application source bundle for the worker environment for periodic tasks. Use environment links to link the web server environment with the worker environment.
- C. Create a second Elastic Beanstalk web server tier environment and deploy the application to process the asynchronous tasks. Send the tasks that should be decoupled from the original Elastic Beanstalk web server to the auto-generated Amazon SQS queue by the second Elastic Beanstalk web server tier environment. Place a cron.yaml file within the root of the application source bundle for the second web server tier environment with the necessary periodic tasks. Use environment links to link both web server environments.
- D. Create an Amazon SQS queue and send the tasks that should be decoupled from the Elastic Beanstalk web server environment to the SQS queue. Create a fleet of EC2 instances under an Auto Scaling group. Install and configure the application to listen for messages within the SQS queue from UserData and create periodic tasks by placing those into the cron in the operating system. Create an environment variable within the Elastic Beanstalk web server environment with a value pointing to the SQS queue endpoint.

Commented [LC38]: B is right since you need the SQS function which is available in the worker tier.

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/concepts-worker.html>

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features-managing-env-tiers.html>

Question #141

You are responsible for the infrastructure of a large-scale highly available multi-tier online application. Amazon Route53 with a load balancer and numerous Amazon EC2 instances comprise this design. You have been entrusted with developing a procedure for deploying in a Blue/Green fashion.

Which method should you use to fulfill this new requirement?

- A. Using Elastic Beanstalk re-deploy your application and configure Elastic Beanstalk Deployment types, and then use Amazon Route53's alias resource record set to swap between Elastic Beanstalk deployment types.
- **B. Re-deploy your application behind a load balancer using an AWS CloudFormation template, launch a new AWS CloudFormation stack during each deployment, update your Amazon Route53 alias resource record set to point to the new load balancer, and finally, terminate your old AWS CloudFormation stack.**
- C. Re-deploy your application behind a load balancer using Auto Scaling groups, create a new identical Auto Scaling group, and associate it to the load balancer. During deployment, create a new Amazon Route53 hosted zone, add this new load balancer to the zone in an alias resource record set, and then remove your old Auto Scaling group.
- D. Re-deploy your application behind a load balancer using an OpsWorks stack, and use AWS OpsWorks stack versioning. During deployment, create a new version of your application, tell OpsWorks to launch the new version behind your load balancer, and when the new version launches, update your Amazon Route53 alias resource record set to point to the new load balancer.

Commented [LC39]: Ans is B.

For Elastic Beanstalk in Blue/Green, we have to create a new environment

Question #142

A business demands a high level of availability for its internal web application. The design consists of a single Amazon EC2 web server instance and a single NAT instance that enables outbound internet connection for updates and public data access.

Which architectural modifications should the business do in order to achieve high availability? (Select two.)

- A. Add the NAT instance to an EC2 Auto Scaling group that spans multiple Availability Zones. Update the route tables.
- **B. Create additional EC2 instances spanning multiple Availability Zones. Add an Application Load Balancer to split the load between them.**
- C. Configure an Application Load Balancer in front of the EC2 instance. Configure Amazon CloudWatch alarms to recover the EC2 instance upon host failure.
- **D. Replace the NAT instance with a NAT gateway in each Availability Zone. Update the route tables.**
- E. Replace the NAT instances with a NAT gateway that spans multiple Availability Zones. Update the route tables.

Commented [LC40]: <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>

Commented [LC41]:

Question #143

An application has been created by a developer that publishes data to Amazon DynamoDB. Conditional writes have been enabled for the DynamoDB table. Writes are failing during high demand periods owing to a ConditionalCheckFailedException problem.

How can the developer improve the dependability of the program when numerous clients try to write to the same record?

- A. Write the data to an Amazon SNS topic.
- B. Increase the amount of write capacity for the table to anticipate short-term spikes or bursts in write operations.
- C. Implement a caching solution, such as DynamoDB Accelerator or Amazon ElastiCache.
- **D. Implement error retries and exponential backoff with jitter.**

Commented [LC42]: The issue is "when multiple clients are attempting to write to the same record". Increasing write units won't help. D is correct. Keep trying after sometimes with some interval

Question #144

You've just been employed by a startup as a DevOps Engineer. Your startup relies entirely on AWS for infrastructure. They presently have no deployment automation, and have had several difficulties when attempting to deploy to production. The organization has said that risk mitigation throughout the deployment process is the primary focus at the moment, and you have a large budget for tools and AWS resources. Their stack is as follows:

```
2-tier API
Data stored in DynamoDB or S3, depending on type
Compute layer is EC2 in Auto Scaling Groups
They use Route53 for DNS pointing to an ELB
An ELB balances load across the EC2 instances
```

The scaling group should be between four and twelve EC2 servers.

Which of the following techniques best fits the demands of this organization, given it's stack and priorities?

- A. Model the stack in AWS Elastic Beanstalk as a single Application with multiple Environments. Use Elastic Beanstalk's Rolling Deploy option to progressively roll out application code changes when promoting across environments.
- B. Model the stack in 3 CloudFormation templates: Data layer, compute layer, and networking layer. Write stack deployment and integration testing automation following Blue-Green methodologies.
- C. Model the stack in AWS OpsWorks as a single Stack, with 1 compute layer and its associated ELB. Use Chef and App Deployments to automate Rolling Deployment.
- D. Model the stack in 1 CloudFormation template, to ensure consistency and dependency graph resolution. Write deployment and integration testing automation following Rolling Deployment methodologies.

Commented [LC43]: A is wrong, as Elastic Beanstalk does not support DynamoDB.

Question #145

You work for a firm that uses artificial neural networks (ANNs) to automatically tag images. These ANNs are implemented in C++ on GPUs. You get millions of photos at a time, but on average just three times every day. These photos are batch-loaded into an AWS S3 bucket that you control, and the customer then publishes a JSON-formatted manifest into another S3 bucket that you also control. Each picture is processed in ten milliseconds utilizing a full GPU. Bootstrapping your neural network software takes five minutes. You must publish image tags as JSON objects to an S3 bucket.

Which of these system designs is the greatest fit for this system?

- A. Create an OpsWorks Stack with two Layers. The first contains lifecycle scripts for launching and bootstrapping an HTTP API on G2 instances for ANN image processing, and the second has an always-on instance which monitors the S3 manifest bucket for new files. When a new file is detected, request instances to boot on the ANN layer. When the instances are booted and the HTTP APIs are up, submit processing requests to individual instances.
- B. Make an S3 notification configuration which publishes to AWS Lambda on the manifest bucket. Make the Lambda create a CloudFormation Stack which contains the logic to construct an autoscaling worker tier of EC2 G2 instances with the ANN code on each instance. Create an SQS queue of the images in the manifest. Tear the stack down when the queue is empty.
- C. Deploy your ANN code to AWS Lambda as a bundled binary for the C++ extension. Make an S3 notification configuration on the manifest, which publishes to another AWS Lambda running controller code. This controller code publishes all the images in the manifest to AWS Kinesis. Your ANN code Lambda Function uses the Kinesis as an Event Source. The system automatically scales when the stream contains image events.
- D. Create an Auto Scaling, Load Balanced Elastic Beanstalk worker tier Application and Environment. Deploy the ANN code to G2 instances in this tier. Set the desired capacity to 1. Make the code periodically check S3 for new manifests. When a new manifest is detected, push all of the images in the manifest into the SQS queue associated with the Elastic Beanstalk worker tier.

Commented [LC44]: The Elastic Beanstalk option is incorrect because it requires a constantly-polling instance, which may break and costs money. The Lambda fleet option is incorrect because AWS Lambda does not support GPU usage. The OpsWorks stack option both requires a constantly-polling instance, and also requires complex timing and capacity planning logic. The CloudFormation option requires no polling, has no always-on instances, and allows arbitrarily fast processing by simply setting the instance count as high as needed.

Reference:
<http://docs.aws.amazon.com/lambda/latest/dg/current-supported-versions.html>

Question #146

A DevOps Engineer is developing a web application's deployment plan. The program will launch Amazon EC2 instances using an AMI through an Auto Scaling group. The same infrastructure will be used in a variety of different situations (development, test, and quality assurance). The deployment plan must adhere to the following criteria:

- * Reduce the instance's starting time
- * Allow the same AMI to run in various settings
- * Securely store secrets for many environments

How is this to be achieved?

- A. Preconfigure the AMI using an AWS Lambda function that launches an Amazon EC2 instance, and then runs a script to install the software and create the AMI. Configure an Auto Scaling lifecycle hook to determine which environment the instance is launched in, and, based on that finding, run a configuration script. Save the secrets on an .ini file and store them in Amazon S3. Retrieve the secrets using a configuration script in EC2 user data.
- **B. Preconfigure the AMI by installing all the software using AWS Systems Manager automation and configure Auto Scaling to tag the instances at launch with their specific environment. Then use a bootstrap script in user data to read the tags and configure settings for the environment. Use the AWS Systems Manager Parameter Store to store the secrets using AWS KMS.**
- C. Use a standard AMI from the AWS Marketplace. Configure Auto Scaling to detect the current environment. Install the software using a script in Amazon EC2 user data. Use AWS Secrets Manager to store the credentials for all environments.
- D. Preconfigure the AMI by installing all the software and configuration for all environments. Configure Auto Scaling to tag the instances at launch with their environment. Use the Amazon EC2 user data to trigger an AWS Lambda function that reads the instance ID and then reconfigures the setting for the proper environment. Use the AWS Systems Manager Parameter Store to store the secrets using AWS KMS.

Commented [LC45]: Correct Answer: B

Reference:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/autoscaling-tagging.html>

<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-automation.html>

<https://aws.amazon.com/cn/blogs/mt/configure-amazon-ec2-instances-in-an-auto-scaling-group-using-state-manager/>

Question #147

A business has created an application that is proving to be more popular than anticipated. The organization wants to guarantee that the application can expand to meet growing demand while maintaining dependability via the use of numerous Availability Zones (AZs). The application is hosted via an Application Load Balancer on a fleet of Amazon EC2 machines (ALB). A DevOps engineer has developed an Auto Scaling group for the application that spans various AZs. Instances started in the newly added AZs get no application traffic.

What is most likely to be the source of this problem?

- A. Auto Scaling groups can create new instances in a single AZ only.
- B. The EC2 instances have not been manually associated to the ALB.
- C. The ALB should be replaced with a Network Load Balancer (NLB).
- **D. The new AZ has not been added to the ALB.**

Commented [LC46]:

Question #148

Five distinct AWS Lambda services comprise an application for a company.

The DevOps Engineer has created a continuous integration and delivery pipeline using AWS CodePipeline and AWS CodeBuild that sequentially creates, tests, packages, and delivers each Lambda function. The pipeline makes use of an Amazon CloudWatch Events rule to guarantee that execution of the pipeline begins as soon as feasible when a modification to the application source code is made.

After a few months of working with the pipeline, the DevOps Engineer observed that it takes much too long to finish.

What should the DevOps Engineer do to optimize the pipeline's performance?

- A. Modify the CodeBuild projects within the pipeline to use a compute type with more available network throughput.
- B. Create a custom CodeBuild execution environment that includes a symmetric multiprocessing configuration to run the builds in parallel.
- **C. Modify the CodePipeline configuration to execute actions for each Lambda function in parallel by specifying the same runOrder.**
- D. Modify each CodeBuild project to run within a VPC and use dedicated instances to increase throughput.

Commented [LC47]: C

<https://docs.aws.amazon.com/codepipeline/latest/userguide/reference-pipeline-structure.html>

AWS docs: "To specify parallel actions, use the same integer for each action you want to run in parallel. For example, if you want three actions to run in sequence in a stage, you would give the first action the runOrder value of 1, the second action the runOrder value of 2, and the third the runOrder value of 3. However, if you want the second and third actions to run in parallel, you would give the first action the runOrder value of 1 and both the second and third actions the runOrder value of 2."

Question #149

A business hosts a website on AWS Elastic Beanstalk, which provides load balancing and automated scalability. This environment's database resource is setup as an Amazon RDS MySQL instance. Following a brief spike in traffic, the website began to see a decline in visitors. An administrator observed that in certain circumstances, the program is not responding due to out-of-memory problems. The Classic Load Balancer designated such instances as unavailable, degrading the health status of Elastic Beanstalk improved health reporting. Elastic Beanstalk, on the other hand, did not replace those instances. Due to the reduced capacity of the Classic Load Balancer, users experience slower application response times.

Which step will resolve this problem permanently?

- A. Clone the Elastic Beanstalk environment. When the new environment is up, swap CNAME and terminate the earlier environment.
- B. Temporarily change the maximum number of instances in the Auto Scaling group to allow the group to support more traffic.
- C. Change the setting for the Auto Scaling group health check from Amazon EC2 to Elastic Load Balancing, and increase the capacity of the group.
- D. Write a cron script for restarting the web server process when memory is full, and deploy it with AWS Systems Manager.

Commented [LC48]: "By default, the Auto Scaling group created for your environment uses Amazon EC2 status checks." Check out the following resources:
<https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-yellow-warning/>
<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environmentconfig-autoscaling-healthchecktype.html>

Questions 150-199

Question #150

You have an Auto Scaling group of Amazon Simple Queue Service (SQS) queues that process messages. The group's size is proportional to the queue's size. The processing phase involves contacting a third-party online service. The web service is complaining about the frequency with which it receives unsuccessful and repeated calls from you.

You've probably seen that as the group grows in, instances are terminated mid-process.

What is the most cost-effective method for reducing the number of unsuccessful process attempts?

- A. Create a new Auto Scaling group with minimum and maximum of 2 and instances running web proxy software. Configure the VPC route table to route HTTP traffic to these web proxies.
- B. Modify the application running on the instances to enable termination protection while it processes a task and disable it when the processing is complete.
- C. Increase the minimum and maximum size for the Auto Scaling group, and change the scaling policies so they scale less dynamically.
- D. Modify the application running on the instances to put itself into an Auto Scaling Standby state while it processes a task and return itself to InService when the processing is complete.

Commented [LC49]: Should be D, even though ideally you want to suspend processes or use scale-in protection

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-using-sqs-queue.html#scale-sqs-queue-scale-in-protection>

Termination protection prevents termination from humans, not from ASG. You would need scale-in protection for that.

Question #151 [SKIP]

What are the absolute minimal criteria for an Ansible playbook to be valid?

- A. The hosts, connection type, fact gathering, vars and tasks.
- B. The hosts declaration and tasks
- C. A YAML file with a single line containing `---`.
- D. At least one play with at least a hosts declaration

Question #152

You've been entrusted with the responsibility of building an automatic data backup solution for your application servers, which are hosted on Amazon EC2 and use Amazon EBS volumes.

To minimize single points of failure and to maximize the data's durability, you want to utilize a distributed data storage for backups. Daily backups should be kept for at least 30 days to ensure that data can be restored within an hour.

How can you do this with a script that is executed daily on the application servers by a scheduling daemon?

- A. Write the script to call the ec2-create-volume API, tag the Amazon EBS volume with the current date time group, and copy backup data to a second Amazon EBS volume. Use the ec2-describe-volumes API to enumerate existing backup volumes. Call the ec2-delete-volume API to prune backup volumes that are tagged with a date-time group older than 30 days.
- B. Write the script to call the Amazon Glacier upload archive API, and tag the backup archive with the current date-time group. Use the list vaults API to enumerate existing backup archives Call the delete vault API to prune backup archives that are tagged with a date-time group older than 30 days.
- C. Write the script to call the ec2-create-snapshot API, and tag the Amazon EBS snapshot with the current date-time group. Use the ec2-describe-snapshot API to enumerate existing Amazon EBS snapshots. Call the ec2-delete-snapShot API to prune Amazon EBS snapshots that are tagged with a datetime group older than 30 days.
- D. Write the script to call the ec2-create-volume API, tag the Amazon EBS volume with the current date-time group, and use the ec2-copy-snapshot API to back up data to the new Amazon EBS volume. Use the ec2-describe-snapshot API to enumerate existing backup volumes. Call the ec2-delete-snapshot API to prune backup Amazon EBS volumes that are tagged with a date-time group older than 30 days.

Commented [LC50]: Ans is C:
The words "Distributed" and "durable" in question mean we should use S3 and snapshot stores in S3.

Question #153

A DevOps Engineer is using AWS CodePipeline to create a multi-stage pipeline for developing, verifying, staging, testing, and deploying an application. Between the test and deploy phases, a manual approval step is necessary. The development team makes use of a team chat application that includes webhook support.

How can the Engineer set chat tool status updates for pipeline activities and approval requests?

- A. Create an AWS CloudWatch Logs subscription that filters on 'detail-type': 'CodePipeline Pipeline Execution State Change.' Forward that to an Amazon SNS topic. Add the chat webhook URL to the SNS topic as a subscriber and complete the subscription validation.
- B. Create an AWS Lambda function that is triggered by the updating of AWS CloudTrail events. When a 'CodePipeline Pipeline Execution State Change' event is detected in the updated events, send the event details to the chat webhook URL.
- C. Create an AWS CloudWatch Events rule that filters on 'CodePipeline Pipeline Execution State Change.' Forward that to an Amazon SNS topic. Subscribe an AWS Lambda function to the Amazon SNS topic and have it forward the event to the chat webhook URL.
- D. Modify the pipeline code to send event details to the chat webhook URL at the end of each stage. Parameterize the URL so each pipeline can send to a different URL based on the pipeline environment.

Commented [LC51]: SNS will directly send the JSON response. Using Lambda we can format and send only relevant information to the webhook channel. So C seems perfect.

Question #154

A legal company is using AWS to host a web application. The system organizes user-uploaded legal papers and stores them on Amazon S3. Users have claimed that file uploads are taking an excessive amount of time and that timeouts occur during periods of high use. A DevOps engineer discovered that web servers are overwhelmed by concurrent uploads.

Which activities should be made to resolve the situation most cost-effectively?

- A. Create an AWS CloudFront distribution in front of the web servers, and modify the application to upload to Amazon S3 using S3 Transfer Acceleration.
- B. Modify the application so the browser uses a signed URL to directly upload to Amazon S3 using multipart uploads.
- C. Create an AWS CloudFront distribution in front of the web servers, and modify the application to store files in Amazon EFS in the Max I/O performance mode.
- D. Place the web servers in an Amazon EC2 Auto Scaling group to include Spot Instances and modify the application to upload to Amazon S3 using multipart uploads.

Commented [LC52]: It's probably a SAA question.

Unsure about the answer.

Ans B

A. Uploading documents using CloudFront would be a better option. CloudFront internally uses S3 transfer acceleration. But when considering cost, CloudFront + S3 Transfer Acceleration doesn't make sense.

B. Web Servers are eliminated completely. Files are directly uploaded to S3. Pre-signed URLs can be generated for an S3 object, allowing only the owner of the URL to to the S3 object using HTTPS. Not only is this more secure due to the custom nature of the URL, but the available options also allow you to set an expiration on the URL, the default being one hour. This is the most cost-effective option.

C. EFS is very expensive compared to S3

D. Spot instances are quite cost effective option is still more expensive than option B. There is zero cost of web servers in option B.

Commented [LC53]: I'll go with B, A) doesn't make sense C) and D) are wrong because not attend the requirement: "These applications MUST continue to operate."

References:

<https://docs.aws.amazon.com/config/latest/developerguide/s3-bucket-server-side-encryption-enabled.html>

<https://docs.aws.amazon.com/config/latest/developerguide/config-rule-multi-account-deployment.html>

<https://docs.aws.amazon.com/organizations/latest/userguide/services-that-can-integrate-config.html>

Question #155

A corporation has implemented a worldwide policy requiring encryption at rest. A DevOps engineer has been assigned the responsibility of ensuring that all new and current Amazon S3 buckets are encrypted at rest within the company's AWS Organizations organization. Numerous older apps running on AWS make use of Amazon S3 and do not encrypt data at rest. These apps MUST remain operational. The engineer must guarantee that S3 encryption is used at rest across the business without needing any changes to the application code.

How can this be achieved with the bare minimum of effort?

- A. Develop an AWS Lambda function that lists all Amazon S3 buckets in a given account and applies default encryption to all S3 buckets that either do not have it enabled or to those with an S3 bucket policy that do not explicitly deny put-object requests without server-side encryption. Deploy the Lambda function along with an Amazon EventBridge (Amazon CloudWatch Events) scheduled rule with AWS CloudFormation StackSets to all accounts within the organization.
- B. Enable the AWS Config s3-bucket-server-side-encryption-enabled managed rule that checks for S3 bucket that either do not have S3 default encryption enabled or those with an S3 bucket policy that does not explicitly deny put-object requests without server-side encryption. Add the AWS-EnabledS3BucketEncryption remediation action to the AWS Config rule to enable default encryption on any S3 buckets that are not compliant. Use AWS Config organizations integration to deploy the rule across all accounts in the organization.
- C. Enable an AWS Config custom rule that checks for S3 buckets that do not have a bucket policy denying access to s3:PutObject unless the x-amz-server-side-encryption S3 condition is met with an AES 256 value or x-amz-server-side-encryption is not present. Add a custom remediation action to the AWS Config rule that will apply the bucket policy if the S3 bucket is non-compliant. Use AWS Config organizations integration to deploy the rule across all accounts in the organization.
- D. Write an SCP that denies access to s3:PutObject unless either the x-amz-server-side-encryption S3 condition is met with an AES 256 value or x-amz-server-side-encryption is not present. Apply the SCP to the root of the organization to enforce the policy across the entire organization.

Question #156

You're having performance problems while writing to a DynamoDB table. Your system keeps track of the highest-scoring video games on a marketplace. All of the performance difficulties occur in your most popular game.

What is the most probable cause of the issue?

- A. DynamoDB's vector clock is out of sync, because of the rapid growth in request for the most popular game.
- **B. You selected the Game ID or equivalent identifier as the primary partition key for the table.**
- C. Users of the most popular video game each perform more read and write requests than average.
- D. You did not provision enough read or write throughput to the table.

Commented [LC54]: The primary key selection dramatically affects performance consistency when reading or writing to DynamoDB. By selecting a key that is tied to the identity of the game, you forced DynamoDB to create a hotspot in the table partitions, and over-request against the primary key partition for the popular game. When it stores data, DynamoDB divides a table's items into multiple partitions, and distributes the data primarily based upon the partition key value. The provisioned throughput associated with a table is also divided evenly among the partitions, with no sharing of provisioned throughput across partitions.

Question #157

A DevOps engineer is utilizing AWS CodePipeline to develop an AWS Service Catalog portfolio. The pipeline should generate products and templates from a JSON or YAML manifest file and should enforce security standards for all AWS Service Catalog goods controlled by the pipeline.

Which solution will fully automate the process of meeting the requirements?

- **A. Use the AWS Service Catalog deploy action in AWS CodeDeploy to push new versions of products into the AWS Service Catalog with verification steps in the CodeDeploy AppSpec.**
- B. Use the AWS Service Catalog deploy action in AWS CodeBuild to verify and push new versions of products into the AWS Service Catalog.
- C. Use an AWS Lambda action in CodePipeline to run a Lambda function to verify and push new versions of products into the AWS Service Catalog.
- D. Use an AWS Lambda action in AWS CodeBuild to run a Lambda function to verify and push new versions of products into the AWS Service Catalog.

Commented [LC55]: <https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-s3-servicecatalog.html>

Question #158

Currently, you have the following configuration in AWS:

- 1) A Load Balancer That Is Elastic
- 2) Auto Scaling Group responsible for the initialization of EC2 Instances
- 3) AMIs pre-installed with your code You wish to distribute your app's updates to a limited number of users. You're looking for a cost-effective solution. Additionally, you should be able to reverse rapidly.

Which of the following options is the most practical?

- **A. Create a second ELB, and a new Auto Scaling Group assigned a new Launch Configuration. Create a new AMI with the updated app. Use Route53 Weighted Round Robin records to adjust the proportion of traffic hitting the two ELBs.**
- B. Create new AMIs with the new app. Then use the new EC2 instances in half proportion to the older instances.
- C. Redeploy with AWS Elastic Beanstalk and Elastic Beanstalk versions. Use Route 53 Weighted Round Robin records to adjust the proportion of traffic hitting the two ELBs
- D. Create a full second stack of instances, cut the DNS over to the new stack of instances, and change the DNS back if a rollback is needed.

Commented [LC56]: The Weighted Routing policy of Route53 can be used to direct a proportion of traffic to your application. The best option is to create a second ELB, attach the new Autoscaling Group and then use Route53 to divert the traffic. Option B is wrong because just having EC2 instances running with the new code will not help. Option C is wrong because Elastic beanstalk is good for development environments, and also there is no mention of having 2 environments where environment urls can be swapped. Option D is wrong because you still need Route53 to split the traffic.

Question #159

Amazon ECS is being used by a business to create a Docker container runtime environment. All Amazon EBS volumes used in the ECS cluster must be encrypted for compliance reasons. The cluster instances will be updated on a rolling basis, and the firm wishes for the instances to be completely decommissioned before being terminated.

How are these stipulations to be met? (Select two.)

- A. Modify the default ECS AMI user data to create a script that executes `docker rm "${f{id}}` for all running container instances. Copy the script to the `/etc/init.d/rc.d` directory and execute `chconfig` enabling the script to run during operating system shutdown.
- B. Use AWS CodePipeline to build a pipeline that discovers the latest Amazon-provided ECS AMI, then copies the image to an encrypted AMI outputting the encrypted AMI ID. Use the encrypted AMI ID when deploying the cluster.
- C. Copy the default AWS CloudFormation template that ECS uses to deploy cluster instances. Modify the template resource EBS configuration setting to set 'Encrypted: True' and include the AWS KMS alias: 'aws/ebs' to encrypt the AMI.
- D. Create an Auto Scaling lifecycle hook backed by an AWS Lambda function that uses the AWS SDK to mark a terminating instance as DRAINING. Prevent the lifecycle hook from completing until the running tasks on the instance are zero.
- E. Create an IAM role that allows the action `ECS::EncryptedImage`. Configure the AWS CLI and a profile to use this role. Start the cluster using the AWS CLI providing the `--use-encrypted-image` and `--kms-key` arguments to the `create-cluster` ECS command.

Commented [LC57]: I would go with C, D.

<https://aws.amazon.com/blogs/compute/how-to-automate-container-instance-draining-in-amazon-ecs/>

Commented [LC58]:

Question #160

A retail firm has chosen AWS OpsWorks for deployment management. The organization observed that certain production instances were resuming without cause during the previous three months. A DevOps Engineer found that the instances were restarted by OpsWorks after inspecting the AWS CloudTrail data. The Engineer now desires automatic email alerts anytime OpsWorks restarts an instance due to an unhealthy state or inability to interact with the service endpoint.

How is the Engineer going to fulfill this requirement?

- A. Create a Chef recipe to place a cron to run a custom script within the Amazon EC2 instances that sends an email to the team by using Amazon SES if the OpsWorks agent detects an instance failure.
- B. Create an Amazon SNS topic and create a subscription for this topic that contains the destination email address. Create an Amazon CloudWatch rule: specify `aws:opsworks` as a source and specify `auto-healing` in the `initiated_by` details. Use the SNS topic as a target.
- C. Create an Amazon SNS topic and create a subscription for this topic that contains the destination email address. Create an Amazon CloudWatch rule: specify `aws:opsworks` as a source and specify `instance-replacement` in the `initiated_by` details. Use the SNS topic as a target.
- D. Create a subscription for this topic that contains the email address. Enable instance restart notifications within the OpsWorks layer and indicate the destination email address for the notification.

Commented [LC59]: Answer is B

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/opsworks-unexpected-start-instance/>

https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/EventTypes.html#opsworks_event_types

<https://aws.amazon.com/blogs/mt/how-to-set-up-aws-opsworks-stacks-auto-healing-notifications-in-amazon-cloudwatch-events/>

Question #161

AWS CodeDeploy is being used by a business to handle its application deployments. The Development team has made the decision to adopt GitHub for version control, and the team is investigating methods to link the GitHub repository with CodeDeploy. Additionally, the team must provide a method for automating deployment anytime a new commit is made to that repository. Currently, the team is manually specifying the Amazon S3 location for new application changes.

How can integration be accomplished in the most efficient manner possible?

- A. Create a GitHub webhook to replicate the repository to AWS CodeCommit. Create an AWS CodePipeline pipeline that uses CodeCommit as a source provider and AWS CodeDeploy as a deployment provider. Once configured, commit a change to the GitHub repository to start the first deployment.
- B. Create an AWS CodePipeline pipeline that uses GitHub as a source provider and AWS CodeDeploy as a deployment provider. Connect this new pipeline with the GitHub account and instruct CodePipeline to use webhooks in GitHub to automatically start the pipeline when a change occurs.
- C. Create an AWS Lambda function to check periodically if there has been a new commit within the GitHub repository. If a new commit is found, trigger a CreateDeployment API call to AWS CodeDeploy to start a new deployment based on the last commit ID within the deployment group.
- D. Create an AWS CodeDeploy custom deployment configuration to associate the GitHub repository with the deployment group. During the association process, authenticate the deployment group with GitHub to obtain the GitHub security authentication token. Configure the deployment group options to automatically deploy if a new commit is found. Perform a new commit to the GitHub repository to trigger the first deployment.

Commented [LC60]: The answer is B. Connect GitHub account and trigger CodePipeline with webhook. If you look a 0:59 of the video in CodeDeploy userguide, it is using GitHub service, which was deprecated in 2018. Now the recommended way for GitHub integration is CodePipeline.

Question #162 [SKIP]

You are creating a Docker image using the Dockerfile below. How many layers will the final picture contain?

```
CMD /app/hello.sh FROM scratch
```

- A. 2
- B. 4
- C. 1
- D. 3

Question #163

A business utilizes Amazon ES to index all of its Amazon CloudWatch Logs and Kibana to create a dashboard with actionable intelligence. The firm want to limit each user's access to Kibana.

Which steps may a DevOps Engineer do to satisfy this criterion? (Select two.)

- A. Create a proxy server with user authentication in an Auto Scaling group, and restrict access of the Amazon ES endpoint to an Auto Scaling group tag.
- B. Create a proxy server with user authentication and an Elastic IP address, and restrict access of the Amazon ES endpoint to the IP address.
- C. Create a proxy server with AWS IAM user, and restrict access of the Amazon ES endpoint to the IAM user.
- D. Use AWS SSO to offer user name and password protection for Kibana.
- E. Use Amazon Cognito to offer user name and password protection for Kibana.

Commented [LC61]: B, E
Dashboards does not natively support IAM users and roles, but OpenSearch Service offers several solutions for controlling access to Dashboards:
Enable SAML authentication for Dashboards.
Use fine-grained access control with HTTP basic authentication.
(Ans E) Configure Configuring Amazon Cognito authentication for OpenSearch Dashboards.
(Ans B) For public access domains, configure an IP-based access policy, with or without a proxy server.
For VPC access domains, use an open access policy, with or without a proxy server, and security groups to control access. To learn more, see About access policies on VPC domains.

Question #164

Amazon CloudTrail logs by default the _____ activities set in the CloudTrail _____ APIs.

- A. bucket-level; RESTful
- B. object-level; RESTful
- C. object-level; SDK
- D. bucket-level; SDK

Commented [LC62]:

Commented [LC63]: By default, CloudTrail logs bucket-level actions. Amazon S3 records are written together with other AWS service records in a log file. Amazon S3 bucket-level actions supported for logging by CloudTrail are defined in its RESTful API.

Reference:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/cloudtrail-logging.html>

Question #165

A developer operations engineer is aiding with the implementation of a multi-region disaster recovery solution for a new application. Amazon EC2 instances operating in an Auto Scaling group and an Amazon Aurora MySQL DB cluster comprise the application. The application must be ready with a 120-minute response time and a 60-minute response time.

How might these needs be met in the MOST cost-effective manner possible?

- A. Launch an Aurora DB cluster as an Aurora Replica in a different Region. Create an AWS CloudFormation template for all compute resources and create a stack in two Regions. Write a script that promotes the Aurora Replica to the primary instance in the event of a failure.
- B. Launch an Aurora DB cluster as an Aurora Replica in a different Region and configure automatic cross-Region failover. Create an AWS CloudFormation template that includes an Auto Scaling group, and create a stack in two Regions. Write a script that updates the CloudFormation stack in the disaster recovery Region to increase the number of instances.
- **C. Use AWS Lambda to create and copy a snapshot of the Aurora DB cluster to the destination Region hourly. Create an AWS CloudFormation template that includes an Auto Scaling group, and create a stack in two Regions. Restore the Aurora DB cluster from a snapshot and update the Auto Scaling group to start launching instances.**
- D. Configure Amazon DynamoDB cross-Region replication. Create an AWS CloudFormation template that includes an Auto Scaling group, and create a stack in two Regions. Write a script that will update the CloudFormation stack in the disaster recovery Region and promote the DynamoDB replica to the primary instance in the event of a failure.

Commented [LC64]: The answers are not really that good, but I think should be C.

A: Running same stack for two regions not cost-effective

B: I don't think Aurora has "automatic cross-Region failover".

This is a manual process:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database-disaster-recovery.html#aurora-global-database-failover>

C: Snapshot is cheaper than replicas and update autoscaling group is good. It meets the requirements for RTO and RPO.

D: DynamoDB, nope

Question #166

You're running an application on numerous Amazon EC2 instances that are part of an Auto Scaling group. You see that instances are being re-spawned in Amazon EC2 when their health checks fail. However, before you have a chance to analyze the problem, the Auto Scaling service terminates the impacted instances. You are notified within 20 minutes if health tests fail. This, however, is insufficient time to solve the problem.

What should you alter to allow you to debug the instance before the Auto Scaling service terminates it, while keeping costs low?

- A. Install the Amazon CloudWatch Logs Agent on the instance and configure application and system logs to be sent to the CloudWatch Logs service.
- B. Configure an Amazon SNS topic and associate it with your Auto Scaling group's CloudWatch alarms. Configure an Amazon SQS queue as a subscriber of this topic, and then create a fleet of Amazon EC2 workers that poll this queue and instruct the Amazon EC2 Auto Scaling API to remove the instance from the Auto Scaling group when an alarm is triggered.
- **C. Create an Auto Scaling Group lifecycle hook to hold the instance in a terminating:wait state until you have completed any troubleshooting. When you have completed troubleshooting, wait for the terminating state to expire, or notify to Scaling to complete the lifecycle hook and terminate the Instance.**
- D. Change the "DeleteOnTermination" flag to false in the Auto Scaling group configuration to ensure that instances are not deleted in the future.

Commented [LC65]: C is correct.

<https://aws.amazon.com/premiumsupport/knowledge-center/auto-scaling-delay-termination/>

Question #167

Your organization utilizes AWS for a variety of applications. Your organization want to design a technology that promptly emails on-call staff when an alert is triggered in your environment. You have numerous on-call teams that work different shifts, and the tool should ensure that the appropriate teams are notified at the appropriate times.

How should this solution be implemented?

- A. Create an Amazon SNS topic and an Amazon SQS queue. Configure the Amazon SQS queue as a subscriber to the Amazon SNS topic. Configure CloudWatch alarms to notify this topic when an alarm is triggered. Create an Amazon EC2 Auto Scaling group with both minimum and desired Instances configured to 0. Worker nodes in this group spawn when messages are added to the queue. Workers then use Amazon Simple Email Service to send messages to your on call teams.
- B. Create an Amazon SNS topic and configure your on-call team email addresses as subscribers. Use the AWS SDK tools to integrate your application with Amazon SNS and send messages to this new topic. Notifications will be sent to on-call users when a CloudWatch alarm is triggered.
- C. Create an Amazon SNS topic and configure your on-call team email addresses as subscribers. Create a secondary Amazon SNS topic for alarms and configure your CloudWatch alarms to notify this topic when triggered. Create an HTTP subscriber to this topic that notifies your application via HTTP POST when an alarm is triggered. Use the AWS SDK tools to integrate your application with Amazon SNS and send messages to the first topic so that on-call engineers receive alerts.
- D. Create an Amazon SNS topic for each on-call group, and configure each of these with the team member emails as subscribers. Create another Amazon SNS topic and configure your CloudWatch alarms to notify this topic when triggered. Create an HTTP subscriber to this topic that notifies your application via HTTP POST when an alarm is triggered. Use the AWS SDK tools to integrate your application with Amazon SNS and send messages to the correct team topic when on shift.

Commented [LC66]: Ans is D:

The requirement is to call the available team not to spin new instances or configure Auto Scaling group. B is sending message to new Topic and C is sending message to the Topic at the top at list and D solves the problem in an appropriate way.

Question #168 [SKIP]

Which tool will Ansible not utilize to obtain facts, even if it is available?

- A. facter
- B. lsb_release
- C. Ansible setup module
- D. ohai

Question #169

A business utilizes Amazon Route 53, AWS Elastic Beanstalk, and Amazon RDS to execute a production application workload under a single AWS account. The Security team wants the application workload to fail over to a new AWS account in the case of a security issue. Additionally, the Security team want to immediately disable all access to the original account, denying access to any AWS resources inside the original AWS account, while forensic examination.

What is the best cost-effective method of preparing for account failover prior to a security incident?

- A. Migrate the Amazon Route 53 configuration to a dedicated AWS account. Mirror the Elastic Beanstalk configuration in a different account. Enable RDS Database Read Replicas in a different account.
- B. Migrate the Amazon Route 53 configuration to a dedicated AWS account. Save/copy the Elastic Beanstalk configuration files in a different AWS account. Copy snapshots of the RDS Database to a different account.
- C. Save/copy the Amazon Route 53 configurations for use in a different AWS account after an incident. Save/copy Elastic Beanstalk configuration files to a different account. Enable the RDS database read replica in a different account.
- D. Save/copy the Amazon Route 53 configurations for use in a different AWS account after an incident. Mirror the configuration of Elastic Beanstalk in a different account. Copy snapshots of the RDS database to a different account.

Commented [LC67]: A and D - incorrect - You can't mirror Elastic Beanstalk to a different account. However, you can save the Elastic Beanstalk configuration to S3, copy the saved configuration to a different account and create an Elastic Beanstalk environment using the saved configuration in the target account
(<https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-migration-accounts/>)

C - incorrect: You can't create a read replica in a different AWS account from the source DB instance
(https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html)

B - Correct - After eliminating A, C and D, the only option left is B. Though this seems like a clumsy, is the only available option.

Question #170

A corporation want to establish a continuous integration/continuous delivery pipeline for an application hosted on AWS. Additionally, the organization provides an on-premises source-code analysis tool that checks for security issues. The utility has not yet been transferred to AWS and is currently only available on-premises. The firm want to do inspections on the source code as part of the pipeline prior to compilation. The inspections typically take between a few minutes and an hour to complete.

How can a DevOps Engineer adhere to these standards?

- A. Use AWS CodePipeline to create a pipeline. Add an action to the pipeline to invoke an AWS Lambda function after the source stage. Have the Lambda function invoke the source-code analysis tool on premises against the source input from CodePipeline. The function then waits for the execution to complete and places the output in a specified Amazon S3 location.
- **B. Use AWS CodePipeline to create a pipeline, then create a custom action type. Create a job worker for the on-premises server that polls CodePipeline for job requests, initiates the tests, and returns the results. Configure the pipeline to invoke the custom action after the source stage.**
- C. Use AWS CodePipeline to create a pipeline. Add a step after the source stage to make an HTTPS request to the on-premises hosted web service that invokes a test with the source code analysis tool. When the analysis is complete, the web service sends the results back by putting the results in an Amazon S3 output location provided by CodePipeline.
- D. Use AWS CodePipeline to create a pipeline. Create a shell script that copies the input source code to a location on premises. Invoke the source code analysis tool and return the results to CodePipeline. Invoke the shell script by adding a custom script action after the source stage.

Commented [LC68]: A HTTP endpoint can be invoked in CodePipeline by one of the two options below:

You can create custom actions for the following AWS CodePipeline action categories: A custom build action or a custom deploy action or a custom test action or a custom invoke action that runs functions. When you create a custom action, you must also create a job worker that will poll CodePipeline for job requests for this custom action, execute the job, and return the status result to CodePipeline. <https://docs.aws.amazon.com/codepipeline/latest/userguide/actions-create-custom-action.html>

You can also create Lambda functions and add them as actions in your pipelines. Because Lambda allows you to write functions to perform almost any task, you can customize the way your pipeline works.

<https://docs.aws.amazon.com/codepipeline/latest/userguide/actions-invoke-lambda-function.html>

In this question, the inspection can take up to an hour. Since the Lambda function will time out after 15 minutes, the only possible answer is Ans B

Question #171

You need to deploy a new version of your application to production. Due to the high-risk nature of the deployment, you must gradually roll out the new version to consumers over many hours to ensure that everything works properly. You must be able to precisely regulate the percentage of people who view the updated version of the program. You utilize ELB and EC2 in conjunction with Auto Scaling Groups and bespoke AMIs pre-installed with your code and allocated to Launch Configurations. During your deployment, there are no database-level modifications. You've been warned not to spend too much money, therefore you must limit the amount of EC2 instances used during the deployment. However, you must be able to easily revert to the original version of code if anything goes wrong.

What is the most effective method for meeting these requirements?

- A. Create a second ELB, Auto Scaling Launch Configuration, and Auto Scaling Group using the Launch Configuration. Create AMIs with all code pre-installed. Assign the new AMI to the second Auto Scaling Launch Configuration. Use Route53 Weighted Round Robin Records to adjust the proportion of traffic hitting the two ELBs.
- B. Use the Blue-Green deployment method to enable the fastest possible rollback if needed. Create a full second stack of instances and cut the DNS over to the new stack of instances, and change the DNS back if a rollback is needed.
- C. Create AMIs with all code pre-installed. Assign the new AMI to the Auto Scaling Launch Configuration, to replace the old one. Gradually terminate instances running the old code (launched with the old Launch Configuration) and allow the new AMIs to boot to adjust the traffic balance to the new code. On rollback, reverse the process by doing the same thing, but changing the AMI on the Launch Config back to the original code.
- D. Migrate to use AWS Elastic Beanstalk. Use the established and well-tested Rolling Deployment setting AWS provides on the new Application Environment, publishing a zip bundle of the new code and adjusting the wait period to spread the deployment over time. Re-deploy the old code bundle to rollback if needed.

Question #172

You must duplicate API requests in real time between two systems.

Which tool should you use to store and transfer API call events?

- A. AWS SQS
- B. AWS Lambda
- **C. AWS Kinesis**
- D. AWS SNS

Commented [LC69]: AWS Kinesis is an event stream service. Streams can act as buffers and transport across systems for in-order programmatic events, making it ideal for replicating

API calls across systems. A typical Amazon Kinesis Streams application reads data from an Amazon Kinesis stream as data records. These applications can use the Amazon Kinesis Client Library, and they can run on Amazon EC2 instances. The processed records can be sent to dashboards, used to generate alerts, dynamically change pricing and advertising strategies, or send data to a variety of other AWS services. For information about Streams features and pricing, see Amazon Kinesis Streams.

Reference:
<http://docs.aws.amazon.com/kinesis/latest/dev/introduction.html>

Question #173

A healthcare organization uses AWS to host a mission-critical application. Recently, the firm encountered some difficulties. If this occurs again, the company's program must be recoverable in another AWS Region. Elastic Load Balancing and Amazon EC2 instances are used in the application. Additionally, the firm maintains a custom AMI containing its application. This AMI is updated on a regular basis. The workload must operate in the main area until a regional service interruption occurs, at which point traffic should fail over to the new region. Additionally, the second region's cost must be minimal. The RTO period is two hours.

Which option enables the organization to fail over to another location in the case of a breakdown while still meeting the standards listed above?

- A. Maintain a copy of the AMI from the main region in the backup region. Create an Auto Scaling group with one instance using a launch configuration that contains the copied AMI. Use an Amazon Route 53 record to direct traffic to the load balancer in the backup region in the event of failure, as required. Allow the Auto Scaling group to scale out as needed during a failure.
- B. Automate the copying of the AMI in the main region to the backup region. Generate an AWS Lambda function that will create an EC2 instance from the AMI and place it behind a load balancer. Using the same Lambda function, point the Amazon Route 53 record to the load balancer in the backup region. Trigger the Lambda function in the event of a failure.
- C. Place the AMI in a replicated Amazon S3 bucket. Generate an AWS Lambda function that can create a launch configuration and assign it to an already created Auto Scaling group. Have one instance in this Auto Scaling group ready to accept traffic. Trigger the Lambda function in the event of a failure. Use an Amazon Route 53 record and modify it with the same Lambda function to point to the load balancer in the backup region.
- **D. Automate the copying of the AMI to the backup region. Create an AWS Lambda function that can create a launch configuration and assign it to an already created Auto Scaling group. Set the Auto Scaling group maximum size to 0 and only increase it with the Lambda function during a failure. Trigger the Lambda function in the event of a failure. Use an Amazon Route 53 record and modify it with the same Lambda function to point to the load balancer in the backup region.**

Commented [LC70]:

Question #174

What is the first transition stage that an instance enters when it leaves steady state due to a health check failure or decreasing load in AWS Auto Scaling?

- **A. Terminating**
- B. Detaching
- C. Terminating:Wait
- D. EnteringStandby

Commented [LC71]: <https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroupLifecycle.html>

Question #175 [SKIP]

Ansible allows directly or through SSH execution of Playbooks on the host.

How is it possible to instruct Ansible to execute its playbooks directly on the host?

- A. Setting 'connection: local' in the tasks that run locally.
- B. Specifying '-type local' on the command line.
- C. It does not need to be specified; it is the default.
- D. Setting 'connection: local' in the Playbook.

Question #176

Which command would you use to modify the configuration parameters for a CloudTrail trail using the AWS CLI?

- A. modify-trail
- B. change-trail
- **C. update-trail**
- D. set-trail

Commented [LC72]: The update-trail command is used to change the configuration settings for a trail. You can only run update-trail command from the region in which the trail was created.

Reference:
<http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-and-update-a-trailby-using-the-aws-cli.html>

Question #177

What does an EC2 security group entail?

- A. Availability Zone
- B. Placement Group
- C. Region
- D. VPC

Commented [LC73]:

Question #178 [SKIP]

To achieve high availability, your system employs a multi-master, multi-region DynamoDB architecture spanning two regions. For the first time since your system was launched, one of the AWS Regions over which you control was down for three hours, and the failover functioned well. However, upon recovery, your users report unusual issues in which users on opposite corners of the world view inconsistent data.

What is a potential design flaw that was overlooked at launch?

- A. The system does not have Lambda Function Repair Automations, to perform table scans and check for corrupted partition blocks inside the Table in the recovered Region.
- B. The system did not implement DynamoDB Table Defragmentation for restoring partition performance in the Region that experienced an outage, so data is served stale.
- C. The system did not include repair logic and request replay buffering logic for post-failure, to resynchronize data to the Region that was unavailable for a number of hours.
- D. The system did not use DynamoDB Consistent Read requests, so the requests in different areas are not utilizing consensus across Regions at runtime.

Question #179

On a single AWS account, you run an online advertising platform. This software generates real-time ad-click data, which you may store in an Amazon S3 bucket named "click-data" as objects. Your advertising partners wish to analyze the ad-click data using Amazon Elastic MapReduce in their own AWS accounts. They've requested instant access to the ad-click data in order to do analytics.

Which two options are necessary to provide secure access to this data? (Select two.)

- A. Create a cross-account IAM role with a trust policy that contains partner AWS account IDs and a unique external ID.
- B. Create a new IAM group for AWS Data Pipeline users with a trust policy that contains partner AWS account IDs.
- C. Configure an Amazon S3 bucket policy for the "click-data" bucket that allows Read-Only access to the objects, and associate this policy with an IAM role.
- D. Configure the Amazon S3 bucket access control list to allow access to the partners Amazon Elastic MapReduce cluster.
- E. Configure AWS Data Pipeline in the partner AWS accounts to use the web Identity Federation API to access data in the "click-data" bucket.
- F. Configure AWS Data Pipeline to transfer the data from the "click-data" bucket to the partner's Amazon Elastic MapReduce cluster.

Commented [LC74]:

Commented [LC75]:

Question #180

AWS CodePipeline is being used by a business to automate their release workflow. In this pipeline, AWS CodeDeploy is used to deploy an application to Amazon ECS utilizing the blue/green deployment approach. The organization wants to build scripts that will test the application's green version prior to rerouting traffic. These scripts should take no more than five minutes to finish. If during these testing, mistakes are identified, the application must be rolled back.

Which technique will satisfy these criteria?

- A. Add a stage to the CodePipeline pipeline between the source and deploy stages. Use AWS CodeBuild to create an execution environment and build commands in the buildspec file to invoke test scripts. If errors are found, use the `aws deploy stop-deployment` command to stop the deployment.
- B. Add a stage to the CodePipeline pipeline between the source and deploy stages. Use this stage to execute an AWS Lambda function that will run the test scripts. If errors are found, use the `aws deploy stop-deployment` command to stop the deployment.
- C. Add a hooks section to the CodeDeploy AppSpec file. Use the `AfterAllowTestTraffic` lifecycle event to invoke an AWS Lambda function to run the test scripts. If errors are found, exit the Lambda function with an error to trigger rollback.
- D. Add a hooks section to the CodeDeploy AppSpec file. Use the `AfterAllowTraffic` lifecycle event to invoke the test scripts. If errors are found, use the `aws deploy stop-deployment` CLI command to stop the deployment.

Commented [LC76]: C looks correct, reference: <https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html#reference-appspec-file-structure-hooks-section-structure-ecs-sample-function>

Question #181

A web application is hosted on Amazon EC2 instances that are routed via an Application Load Balancer. On the backend, Amazon RDS MySQL is utilized. The instances are distributed across several Availability Zones in an Auto Scaling group. The Application Load Balancer health check verifies that the web servers are operational and capable of reading and writing SQL data. With a record pointing to the Application Load Balancer, Amazon Route 53 offers DNS functionality. A new policy mandates the establishment of a geographically separated disaster recovery facility with a four-hour RTO and a fifteen-minute RPO.

Which catastrophe recovery technique requires the fewest modifications to the application stack?

- A. Launch a replica stack of everything except RDS in a different Availability Zone. Create an RDS read-only replica in a new Availability Zone and configure the new stack to point to the local RDS instance. Add the new stack to the Route 53 record set with a failover routing policy.
- B. Launch a replica stack of everything except RDS in a different region. Create an RDS read-only replica in a new region and configure the new stack to point to the local RDS instance. Add the new stack to the Route 53 record set with a latency routing policy.
- C. Launch a replica stack of everything except RDS in a different region. Upon failure, copy the snapshot over from the primary region to the disaster recovery region. Adjust the Amazon Route 53 record set to point to the disaster recovery region's Application Load Balancer.
- D. Launch a replica stack of everything except RDS in a different region. Create an RDS read-only replica in a new region and configure the new stack to point to the local RDS instance. Add the new stack to the Amazon Route 53 record set with a failover routing policy.

Commented [LC77]: A: was wrong. AZs are within the same geographic area. In order to have geographically isolated DR capability, we need look for region. B is not correct. Route53 record should have failover routing policy not latency routing policy. C is not correct. If the database size was large, we cannot guaranty transferring snapshot from region to region within four hours, let alone 15 RPO.

Question #182

A DevOps team wants to use AWS to develop their containerized application. The deployment must adhere to the following criteria:

☞ There should be little downtime associated with deployment.

☞ To be regarded as successful, the application must undergo functional testing.

How can this deployment be automated by the DevOps team?

- A. Use AWS Elastic Beanstalk with a multi-Docker container solution stack. Select immutable updates as a deployment strategy. Select enhanced health as a monitoring type in the Elastic Beanstalk environment to ensure health checks are transmitted at deployment.
- B. Use an Amazon ECS cluster and service with an Application Load Balancer and an AWS CodeDeploy blue/green deployment type. Define a production port and a test port in Amazon ECS. Write an AWS Lambda function to test the application, and reference it within the AfterAllowTestTraffic hook in the appspec.yml.
- C. Use AWS CloudFormation to provision Amazon EC2 instances behind an Application Load Balancer. Deploy the containers using Amazon ECS. Upon deployment, replicate the configuration in the new EC2 instances, perform testing, and switch traffic from the old Application Load Balancer to the new one using Amazon Route 53.
- D. Use an Amazon ECS cluster and service along with Amazon EC2 instances and an Application Load Balancer. Select rolling update as a deployment strategy. Add a Docker health check within the task definition to ensure rollback if the health check fails.

Commented [LC78]: it's B From Test listener port, choose the port and protocol of a test listener that serves traffic to the replacement task set in your Amazon ECS service during deployment. You can specify one or more Lambda functions in the AppSpec file that run during the AfterAllowTestTraffic hook. The functions can run validation tests. If a validation test fails, a deployment rollback is triggered.

<https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-groups-create-ecs.html>

Question #183

You're developing a smartphone application that will allow people to share cat photographs online. The photos will be stored in AWS S3. You want to operate the system as inexpensively and easily as possible.

Which of these choices enables you to create a picture sharing service without worrying about scaling costly upload procedures, authentication/authorization, and so on?

- A. Build the application out using AWS Cognito and web identity federation to allow users to log in using Facebook or Google Accounts. Once they are logged in, the secret token passed to that user is used to directly access resources on AWS, like AWS S3.
- B. Use JWT or SAML compliant systems to build authorization policies. Users log in with a username and password, and are given a token they can use indefinitely to make calls against the photo infrastructure.
- C. Use AWS API Gateway with a constantly rotating API Key to allow access from the client-side. Construct a custom build of the SDK and include S3 access in it.
- D. Create an AWS OAuth Service Domain and grant public signup and access to the domain. During setup, add at least one major social media site as a trusted Identity Provider for users.

Commented [LC79]: The short answer is that Amazon Cognito is a superset of the functionality provided by web identity federation. It supports the same providers, and you configure your app and authenticate with those providers in the same way. But Amazon Cognito includes a variety of additional features. For example, it enables your users to start using the app as a guest user and later sign in using one of the supported identity providers.

Reference:
<https://aws.amazon.com/blogs/security/how-does-amazon-cognito-relate-to-existing-web-identity-federation/>

Question #184

A DevOps Engineer is assisting with the deployment of an application over 12 Amazon EC2 machines spread across three Availability Zones. AMI images may be used to launch new instances. On a normal day, each EC2 instance is used 30% during business hours and 10% after business hours. In the initial few minutes of business hours, CPU consumption immediately increases. Other increases in CPU use occur in a progressive manner. The Engineer has been tasked with the task of reducing costs while maintaining or increasing dependability.

Which solution satisfies these criteria?

- A. Create two Amazon CloudWatch Events rules with schedules before and after business hours begin and end. Create two AWS Lambda functions, one invoked by each rule. The first function should stop nine instances after business hours end, the second function should restart the nine instances before the business day begins.
- B. Create an Amazon EC2 Auto Scaling group using the AMI image, with a scaling action based on the Auto Scaling group's CPU Utilization average with a target of 75%. Create a scheduled action for the group to adjust the minimum number of instances to three after business hours end and reset to six before business hours begin.
- C. Create two Amazon CloudWatch Events rules with schedules before and after business hours begin and end. Create an AWS CloudFormation stack, which creates an EC2 Auto Scaling group, with a parameter for the number of instances. Invoke the stack from each rule, passing a parameter value of three in the morning, and six in the evening.
- D. Create an EC2 Auto Scaling group using the AMI image, with a scaling action based on the Auto Scaling group's CPU Utilization average with a target of 75%. Create a scheduled action to terminate nine instances each evening after the close of business.

Commented [LC80]: Answer is B. Options A and C are incomplete as they aren't taking care of CPU Utilization rise gradually while retaining the same or higher reliability. In Options B & D, we need to pick the most cost-effective solution. Hence, running 12 Instance in business Hours (Option D) is expensive than running 6 Instances in business Hours (Option B). 30% utilization during Business hours implies that at min 3.6 ~ 4 Instances would be required to handle Business hours load in full capacity.

Question #185

A user is creating an IAM user policy.

Which of the following components is included in an IAM policy?

- A. Not Effect
- **B. Supported Data Types**
- C. Principal Resource
- D. Version Management

Question #186

For cluster computing applications, the maximum potential network performance is required. You've previously chosen homogenous instance types that enable 10 gigabit improved networking, verified that your workload is network-bound, and grouped the instances together.

What is the last optimization that you can perform?

- A. Use 9001 MTU instead of 1500 for Jumbo Frames, to raise packet body to packet overhead ratios.
- **B. Segregate the instances into different peered VPCs while keeping them all in a placement group, so each one has its own Internet Gateway.**
- C. Bake an AMI for the instances and relaunch, so the instances are fresh in the placement group and do not have noisy neighbors.
- D. Turn off SYN/ACK on your TCP stack or begin using UDP for higher throughput.

Question #187

A business is developing a software solution that implements a certain parallel processing method. In some circumstances, the program can grow to tens of servers. This solution makes use of a proprietary library that is licensed on an individual server basis, requiring each server to have a separate, dedicated license installed. The firm has 200 licenses and intends to operate no more than 200 server nodes simultaneously.

The following characteristics have been sought by the company:

* A technique for automating license use at scale.

* The creation of a dashboard for future use in determining which licenses are accessible at any given time.

Which method is the MOST EFFECTIVE for meeting these requirements?

- A. Upload the licenses to a private Amazon S3 bucket. Create an AWS CloudFormation template with a Mappings section for the licenses. In the template, create an Auto Scaling group to launch the servers. In the user data script, acquire an available license from the Mappings section. Create an Auto Scaling lifecycle hook, then use it to update the mapping after the instance is terminated.
- **B. Upload the licenses to an Amazon DynamoDB table. Create an AWS CloudFormation template that uses an Auto Scaling group to launch the servers. In the user data script, acquire an available license from the DynamoDB table. Create an Auto Scaling lifecycle hook, then use it to update the mapping after the instance is terminated.**
- C. Upload the licenses to a private Amazon S3 bucket. Populate an Amazon SQS queue with the list of licenses stored in S3. Create an AWS CloudFormation template that uses an Auto Scaling group to launch the servers. In the user data script acquire an available license from SQS. Create an Auto Scaling lifecycle hook, then use it to put the license back in SQS after the instance is terminated.
- D. Upload the licenses to an Amazon DynamoDB table. Create an AWS CLI script to launch the servers by using the parameter --count, with min:max instances to launch. In the user data script, acquire an available license from the DynamoDB table. Monitor each instance and, in case of failure, replace the instance, then manually update the DynamoDB table.

Commented [LC81]: The question doesn't sound right. I guess it should be an element of IAM policy. Even if we change "components" to "elements", supported data types is not an element of an IAM policy. The following are the elements of an IAM policy:

IAM JSON policy elements: Version
IAM JSON policy elements: Id
IAM JSON policy elements: Statement
IAM JSON policy elements: Sid
IAM JSON policy elements: Effect
AWS JSON policy elements: Principal
AWS JSON policy elements: NotPrincipal
IAM JSON policy elements: Action
IAM JSON policy elements: NotAction
IAM JSON policy elements: Resource
IAM JSON policy elements: NotResource
IAM JSON policy elements: Condition

Supported Data Types are the data types supported by an IAM policy.

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements.html

Commented [LC82]: "A cluster placement group is a logical grouping of instances within a single Availability Zone. A cluster placement group can span peered VPCs in the same Region. Instances in the same cluster placement group enjoy a higher per-flow throughput limit for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network."

So the answer is B.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

Commented [LC83]: A - license list is dynamic (based on scaling) and mapping are good for values that are static in nature

B - looks good

C - sqs - unnecessary overhead

D - too many manual tasks

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/mappings-section-structure.html>

Question #188

A business utilizes Amazon S3 to store confidential data. Daily, the Development team builds buckets for new initiatives. The Security team want to guarantee that encryption, logging, and versioning are enabled for all current and future buckets. Additionally, no bucket should ever be read-write available to the public.

What actions should a DevOps Engineer do to ensure compliance with these requirements?

- A. Enable AWS CloudTrail and configure automatic remediation using AWS Lambda.
- **B. Enable AWS Config rules and configure automatic remediation using AWS Systems Manager documents.**
- C. Enable AWS Trusted Advisor and configure automatic remediation using Amazon CloudWatch Events.
- D. Enable AWS Systems Manager and configure automatic remediation using Systems Manager documents.

Commented [LC84]: <https://aws.amazon.com/blogs/mt/aws-config-auto-remediation-s3-compliance/>
<https://aws.amazon.com/blogs/aws/aws-config-rules-dynamic-compliance-checking-for-cloud-resources/>

Question #189

You have an Auto Scaling group with a fleet of Elastic Compute Cloud (EC2) instances. Each of these instances is powered by Microsoft Windows Server 2012 and is backed up by Amazon Elastic Block Store (EBS). AWS CloudFormation was used to launch these instances. You've concluded that your instances are underused and, in an effort to save money, have chosen to change the fleet's instance type.

Which of the following methods may you use to accomplish your goal during a planned maintenance window? (Select two.)

- **A. Create a new Auto Scaling launch configuration specifying the new instance type, associate it to the existing Auto Scaling group, and terminate the running instances.**
- B. Identify the new instance type in the user data and restart the running instances one at a time.
- C. Use the AWS Command Line Interface (CLI) to modify the instance type of each running instance.
- **D. Change the instance type in the AWS CloudFormation template that was used to create the Amazon EC2 instances, and then update the stack.**
- E. Take snapshots of the running instances, and launch new instances based on those snapshots.

Commented [LC85]:

Commented [LC86]:

Question #190

Amazon EC2 Auto Scaling is widely used by an online business to give an exceptional client experience while lowering the number of active EC2 instances. The configuration of the instances is managed by the company's self-hosted Puppet environment under the application layer. The IT manager is interested in minimizing licensing costs and ensuring that when the EC2 Auto Scaling group grows down, the eliminated EC2 instances are deregistered from the Puppet master as quickly as feasible.

How may this criterion be met?

- **A. At instance launch time, use EC2 user data to deploy the AWS CodeDeploy agent. Use CodeDeploy to install the Puppet agent. When the Auto Scaling group scales out, run a script to register the newly deployed instances to the Puppet master. When the Auto Scaling group scales in, use the EC2 Auto Scaling EC2_INSTANCE_TERMINATING lifecycle hook to trigger de-registration from the Puppet master.**
- B. Bake the AWS CodeDeploy agent into the base AMI. When the Auto Scaling group scales out, use CodeDeploy to install the Puppet agent, and execute a script to register the newly deployed instances to the Puppet master. When the Auto Scaling group scales in, use the CodeDeploy ApplicationStop lifecycle hook to run a script to de-register the instance from the Puppet master.
- C. At instance launch time, use EC2 user data to deploy the AWS CodeDeploy agent. When the Auto Scaling group scales out, use CodeDeploy to install the Puppet agent, and run a script to register the newly deployed instances to the Puppet master. When the Auto Scaling group scales in, use the EC2 user data instance stop script to run a script to de-register the instance from the Puppet master.
- D. Bake the AWS Systems Manager agent into the base AMI. When the Auto Scaling group scales out, use the AWS Systems Manager to install the Puppet agent, and run a script to register the newly deployed instances to the Puppet master. When the Auto Scaling group scales in, use the Systems Manager instance stop lifecycle hook to run a script to de-register the instance from the Puppet master.

Commented [LC87]: This is a tough one. The key to understanding it is that CodeDeploy and ASG's can work together! There's a great blog post about this: <https://aws.amazon.com/blogs/devops/under-the-hood-aws-codedeploy-and-auto-scaling-integration/>

Here is my analysis:

- A: This is the correct approach and matches the blog post writeup
B: The CodeDeploy ApplicationStop lifecycle hook relates to upgrading an instance in place. The EC2 lifecycle hook is needed here.
C: There is no such thing as a "user data instance stop script"
D: SSM does not have lifecycle hooks

Question #191

You just installed an application on Amazon EC2 instances protected by an ELB. After a few weeks, consumers begin to complain about the application's faults. You're attempting to troubleshoot the issues and are obtaining them from the ELB access logs. However, the ELB access logs are blank.

What is the rationale for this?

- A. You do not have the appropriate permissions to access the logs
- B. You do not have your CloudWatch metrics correctly configured
- C. ELB Access logs are only available for a maximum of one week
- **D. Access logging is an optional feature of Elastic Load Balancing that is disabled by default**

Commented [LC88]: Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and to troubleshoot issues. Access logging is an optional feature of Elastic Load Balancing that is disabled by default. After you enable access logging for your load balancer, Elastic Load Balancing captures the logs and stores them in the Amazon S3 bucket that you specify. You can disable access logging at any time.

Question #192

A business uses Amazon EC2 instances to operate an application behind an ELB Application Load Balancer. The instances are distributed across several Availability Zones through an EC2 Auto Scaling group. Users are receiving HTTP 502 Bad Gateway errors from the application URL after a recent application upgrade. The DevOps Engineer is unable to evaluate the issue since Auto Scaling is immediately terminating all EC2 instances for being unhealthy.

How does the DevOps Engineer get access to one of the sick instances in order to debug the deployed application?

- A. Create an image from the terminated instance and create a new instance from that image. The Application team can then log into the new instance.
- **B. As soon as a new instance is created by AutoScaling, put the instance into a Standby state as this will prevent the instance from being terminated.**
- C. Add a lifecycle hook to your Auto Scaling group to move instances in the Terminating state to the Terminating:Wait state.
- D. Edit the Auto Scaling group to enable termination protection as this will protect unhealthy instances from being terminated.

Commented [LC89]: Answer B:
A - incorrect - can't clone a terminated instance
B - correct - an instance always starts healthy, until an ELB check, custom check, etc says it isn't. There is a period of time between a newly launched, "InService" instance, and its first health check. Seconds, minutes, depending on ASG/ELB. While InService and Healthy, set to Standby. There's also a period of time between an instance being marked as Unhealthy, and before the ASG goes to terminate it. Set it to standby. When you click Standby and the instance is Unhealthy & InService, the Unhealthy status never changes till released from Standby
C - almost correct - question wants to "pause" a single instance, not all instances. Otherwise this also works.
D - incorrect - doesn't protect unhealthy instances from terminating - applies to scale-in/out events. Process Suspension would work

Question #193

A business wants to install infrastructure using AWS CloudFormation. The firm has specific criteria for tagging and resource allocation and want to restrict rollout to two regions. Additionally, developers will be required to deploy numerous versions of the same application.

Which solution guarantees that resources are allocated according to corporate policy?

- A. Create AWS Trusted Advisor checks to find and remediate unapproved CloudFormation StackSets.
- B. Create a CloudFormation drift detection operation to find and remediate unapproved CloudFormation StackSets.
- **C. Create CloudFormation StackSets with approved CloudFormation templates.**
- D. Create AWS Service Catalog products with approved CloudFormation templates.

Commented [LC90]: Very doubtful between C and D since D uses C under the hood.

C is what's behind, maybe more adaptable to developers?

<https://aws.amazon.com/blogs/aws/use-cloudformation-stacksets-to-provision-resources-across-multiple-aws-accounts-and-regions/>

Commented [LC91]: Snapshots shared with other users are usable in full by the recipient, including but limited to the ability to base modified volumes and snapshots.

Reference:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modifying-snapshotpermissions.html>

Question #194

Which of the following is an AWS EBS Snapshots restriction?

- A. Snapshot restorations are restricted to the region in which the snapshots are created.
- B. You cannot share unencrypted snapshots.
- **C. To share a snapshot with a user in another region the snapshot has to be created in that region first.**
- D. You cannot share a snapshot containing sensitive data such as an AWS Access Key ID or AWS Secret Access Key.

Question #195

You have been entrusted with the responsibility of implementing a solution for your organization that will save photos for use in marketing campaigns. Employees may submit photographs using a web interface, and each image must be scaled and watermarked with the corporate brand after it is posted. Resizing and watermarking images is not a time-sensitive process and may be finished days after upload if necessary.

How should you build this solution to be as accessible and cost-effective as possible?

- A. Configure your web application to upload images to the Amazon Elastic Transcoder service. Use the Amazon Elastic Transcoder watermark feature to add the company logo as a watermark on your images and then to upload the final images into an Amazon S3 bucket.
- B. Configure your web application to upload images to Amazon S3, and send the Amazon S3 bucket URI to an Amazon SQS queue. Create an Auto Scaling group and configure it to use Spot instances, specifying a price you are willing to pay. Configure the instances in this Auto Scaling group to poll the SQS queue for new images and then resize and watermark the image before uploading the final images into Amazon S3.
- C. Configure your web application to upload images to Amazon S3, and send the S3 object URI to an Amazon SQS queue. Create an Auto Scaling launch configuration that uses Spot instances, specifying a price you are willing to pay. Configure the instances in this Auto Scaling group to poll the Amazon SQS queue for new images and then resize and watermark the image before uploading the new images into Amazon S3 and deleting the message from the Amazon SQS queue.
- D. Configure your web application to upload images to the local storage of the web server. Create a cronjob to execute a script daily that scans this directory for new files and then uses the Amazon EC2 Service API to launch 10 new Amazon EC2 instances, which will resize and watermark the images daily.

Commented [LC92]: Ans is C: B is also almost the same as C but in it, say Bucket URL which is wrong

Question #196

A DevOps Engineer's primary responsibility is to verify that all IAM entity settings across numerous AWS accounts in AWS Organizations adhere to corporate IAM regulations.

Which sequence of steps will do this? (Select two.)

- A. Enable AWS Trusted Advisor in Organizations for all accounts to report on noncompliant IAM entities.
- B. Configure an AWS Config aggregator in the Organizations master account for all accounts.
- C. Deploy AWS Config rules to the master account in Organizations that match corporate IAM policies.
- D. Apply an SCP in Organizations to ensure compliance of IAM entities.
- E. Deploy AWS Config rules to all accounts in Organizations that match the corporate IAM policies.

Commented [LC93]: An aggregator is an AWS Config resource type that collects AWS Config configuration and compliance data from the following:

Multiple accounts and multiple regions.

Single account and multiple regions.

An organization in AWS Organizations and all the accounts in that organization which have AWS Config enabled.
So, Ans B, E

Commented [LC94]:

Question #197

Your continuous integration system must produce AMIs with code pre-installed on each new code push. You must do this as inexpensively as feasible.

How do you accomplish this?

- A. Bid on spot instances just above the asking price as soon as new commits come in, perform all instance configuration and setup, then create an AMI based on the spot instance.
- B. Have the CI launch a new on-demand EC2 instance when new commits come in, perform all instance configuration and setup, then create an AMI based on the on-demand instance.
- C. Purchase a Light Utilization Reserved Instance to save money on the continuous integration machine. Use these credits whenever you create AMIs on instances.
- D. When the CI instance receives commits, attach a new EBS volume to the CI machine. Perform all setup on this EBS volume so you do not need a new EC2 instance to create the AMI.

Commented [LC95]: A Spot Instances with a defined duration (also known as Spot blocks) are no longer available to new customers since July 1, 2021. For customers that have previously used the feature, we will continue to support Spot Instances with a defined duration until December 31, 2022.

You can now request Amazon EC2 Spot instances to run continuously, for up to six hours, at a flat rate that saves you up to 50% compared to On-Demand prices. This enables you to reduce costs when running finite duration tasks.

Question #198

A firm that operates several workloads on AWS has seen a rise in its Amazon EBS cost over time. The DevOps team discovers a large number of disconnected EBS volumes. While certain workloads need volumes to be detached, volumes older than 14 days are considered stale and no longer required. A DevOps engineer has been assigned the responsibility of developing automation that deletes detached EBS volumes after 14 days.

Which approach is most likely to do this?

- A. Configure the AWS Config `ec2-volume-in-use-check` managed rule with a configuration changes trigger type and an Amazon EC2 volume resource target. Create a new Amazon CloudWatch Events rule scheduled to execute an AWS Lambda function in 14 days to delete the specified EBS volume.
- B. Use Amazon EC2 and Amazon Data Lifecycle Manager to configure a volume lifecycle policy. Set the interval period for unattached EBS volumes to 14 days and set the retention rule to delete. Set the policy target volumes as `*`.
- C. Create an Amazon CloudWatch Events rule to execute an AWS Lambda function daily. The Lambda function should find unattached EBS volumes and tag them with the current date, and delete unattached volumes that have tags with dates that are more than 14 days old.
- D. Use AWS Trusted Advisor to detect EBS volumes that have been detached for more than 14 days. Execute an AWS Lambda function that creates a snapshot and then deletes the EBS volume.

Commented [LC96]: <https://aws.amazon.com/blogs/mt/controlling-your-aws-costs-by-deleting-unused-amazon-ebs-volumes/>

Question #199

A business creates and maintains a web application in a single Availability Zone utilizing Amazon EC2 instances and an Amazon RDS for SQL Server database instance. The resources are required solely when AWS CodePipeline is used to test fresh installations. Testing happens once or more every week and each test lasts around 2-3 hours. A DevOps engineer is looking for a solution that does not need any changes to the architecture's components.

Which approach will be the MOST cost-effective in meeting these requirements?

- A. Convert the RDS database to an Amazon Aurora Serverless database. Use an AWS Lambda function to start and stop the EC2 instances before and after tests.
- B. Put the EC2 instances into an Auto Scaling group. Schedule scaling to run at the start of the deployment tests.
- C. Replace the EC2 instances with EC2 Spot Instances and the RDS database with an RDS Reserved Instance.
- D. Subscribe Amazon Cloud Watch Events to CodePipeline to trigger AWS Systems Manager Automation documents that start and stop all EC2 and RDS instances before and after deployment tests.

Commented [LC97]:

Questions 200-249

Question #200 [WRONG]

What is the maximum number of Roles that an AWS account may have by default?

- A. 500
- B. 250
- C. 100
- D. There is no limit.

Commented [LC98]: This is dynamic and always change. Currently it is 1,000 IAM roles you are limited under your AWS account.

<https://aws.amazon.com/iam/faqs/#:~:text=You%20are%20limited%20to%201%2C000,we%20will%20consider%20your%20request>

Q: How many IAM roles can I create?

You are limited to 1,000 IAM roles under your AWS account. If you need more roles, submit the IAM limit increase request form with your use case, and we will consider your request.

Question #201

Which CloudTrail services may be utilized as optional components when creating a new Trail?

- A. KMS, SNS and SES
- B. CloudWatch, S3 and SNS
- C. KMS, Cloudwatch and SNS
- D. KMS, S3 and CloudWatch

Commented [LC99]: Key Management Service: The use of AWS KMS is an optional element of CloudTrail, but it allows additional encryption to be added to your Log files when stored on S3 Simple Notification Service: Amazon SNS is also an optional component for CloudTrail, but it allows for you to create notifications, for example when a new log file is delivered to S3 SNS could notify someone or a team via an e-mail. Or it could be used in conjunction with CloudWatch when metric thresholds have been reached. CloudWatch Logs: Again, this is another optional component, but AWS CloudTrail allows you to deliver its logs to AWS Cloudwatch Logs as well as S3 for specific monitoring metrics to take place.

Reference:

<https://cloudacademy.com/amazon-web-services/aws-cloudtrail-introduction-course/how-doesaws-cloudtrail-work.html>

Commented [LC100]: https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_identifiers.html

Question #202

A root owner is attempting to establish IAM users for each department. Although the owner has formed groups for each department, he wishes to continue segmenting users at the sub division level. For instance, two individuals from distinct sub-departments should be recognized and granted separate access.

How is this configured by the root owner?

- A. Create a hierarchy of the IAM users which are separated based on the department
- B. Create a nested group
- C. Use the paths to separate the users of the same group
- D. It is not possible to delineate within a group

Question #203

A DevOps Engineer is responsible for the deployment of a new online application. The organization selects AWS Elastic Beanstalk for web application deployment and management, and Amazon RDS MySQL for permanent data storage. The firm demands that new deployments have a negligible effect on existing operations in the event of a failure. The application's resources must be fully used during deployment, and it must also be able to roll back a deployment.

Which deployment sequence will satisfy these criteria?

- A. Deploy the application using Elastic Beanstalk and connect to an external RDS MySQL instance using Elastic Beanstalk environment properties. Use Elastic Beanstalk features for a blue/green deployment to deploy the new release to a separate environment, and then swap the CNAME in the two environments to redirect traffic to the new version.
- B. Deploy the application using Elastic Beanstalk, and include RDS MySQL as part of the environment. Use default Elastic Beanstalk behavior to deploy changes to the application, and let rolling updates deploy changes to the application.
- C. Deploy the application using Elastic Beanstalk, and include RDS MySQL as part of the environment. Use Elastic Beanstalk immutable updates for application deployments.
- D. Deploy the application using Elastic Beanstalk, and connect to an external RDS MySQL instance using Elastic Beanstalk environment properties. Use Elastic Beanstalk immutable updates for application deployments.

Commented [LC101]: A - might work but it is not mentioned that old environment will be deleted, so additional cost(double) involved here
B - rds should be separated from ebs
C - rds should be separated from ebs
D - correct. this will terminate the old environment as mentioned below

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environmentmgmt-updates-immutable.html>

When the first instance passes health checks, Elastic Beanstalk launches additional instances with the new configuration, matching the number of instances running in the original Auto Scaling group. When all of the new instances pass health checks, Elastic Beanstalk transfers them to the original Auto Scaling group, and terminates the temporary Auto Scaling group and old instances.

Question #204

A DevOps Engineer is using AWS CodePipeline and AWS CodeBuild to create a continuous deployment pipeline for a serverless application. The source, build, and test steps have been completed, leaving just the deploy stage. The organization wants to minimize the risk of a failed deployment by first distributing to a limited fraction of consumers and monitoring that deployment prior to rolling out to all customers.

What configuration should the deploy stage use to fulfill these requirements?

- A. Use AWS CloudFormation to publish a new version on every stack update. Then set up a CodePipeline approval action for a Developer to test and approve the new version. Finally, use a CodePipeline invoke action to update an AWS Lambda function to use the production alias
- B. Use CodeBuild to use the AWS CLI to update the AWS Lambda function code, then publish a new version of the function and update the production alias to point to the new version of the function.
- **C. Use AWS CloudFormation to define the serverless application and AWS CodeDeploy to deploy the AWS Lambda functions using DeploymentPreference: Canary10Percent15Minutes.**
- D. Use AWS CloudFormation to publish a new version on every stack update. Use the RoutingConfig property of the AWS::Lambda::Alias resource to update the traffic routing during the stack update.

Commented [LC102]: ans should be C

<https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/automating-updates-to-serverless-apps.html>

Question #205

Due to the fact that CloudTrail sends a notice whenever a log file is written to the Amazon S3 bucket, a particularly active account might create a huge number of alerts. If you subscribe through email or text message, you may get a high number of messages.

Which of the following should you utilize to programmatically manage notifications?

- A. Amazon Kinesis Firehose
- **B. Amazon Simple Queue Service (Amazon SQS)**
- C. Amazon Simple Email Service (Amazon SES)
- D. Amazon AppStream

Commented [LC103]: As CloudTrail sends a notification each time a log file is written to the Amazon S3 bucket, an account that's very active can generate a large number of notifications. If you subscribe using email or SMS, you can end up receiving more messages than you can handle. AWS recommends that you subscribe using Amazon Simple Queue Service (Amazon SQS), which lets you handle notifications programmatically.

Reference:
http://docs.aws.amazon.com/awscloudtrail/latest/userguide/getting_notifications_configuration.html

Question #206

A firm just transferred its on-premises legacy application to AWS. The application is hosted on Amazon EC2 instances that are routed via an Application Load Balancer that is routed through Amazon API Gateway. The organization wants to guarantee that consumers experience minimum downtime when a new version of the program is deployed. Additionally, the organization needs to guarantee that it can roll back upgrades rapidly in the event of an error.

Which solution satisfies these requirements with MINIMAL application modification?

- **A. Introduce changes as a separate environment parallel to the existing one. Configure API Gateway to use a canary release deployment to send a small subset of user traffic to the new environment.**
- B. Introduce changes as a separate environment parallel to the existing one. Update the application's DNS alias records to point to the new environment.
- C. Introduce changes as a separate target group behind the existing Application Load Balancer. Configure API Gateway to route user traffic to the new target group in steps.
- D. Introduce changes as a separate target group behind the existing Application Load Balancer. Configure API Gateway to route all traffic to the Application Load Balancer, which then sends the traffic to the new target group.

Commented [LC104]: I'll got with A

B) required a lot of changes and the DNS can take longer to propagate

C) API gateway cannot choose the target group, this is done by the ELB

D) ELB Target groups changes require some work, you can just point to a new loadbalancer

Question #207

You've determined that you need to modify the instance type of your production instances that are members of an AutoScaling group. CloudFormation Template is used to deploy the full architecture. Currently, you have four instances running in Production. You cannot have any service interruptions and must have two instances functioning at all times throughout the upgrade.

Which of the following choices is appropriate for this?

- **A. AutoScalingRollingUpdate**
- B. AutoScalingScheduledAction
- C. AutoScalingReplacingUpdate
- D. AutoScalingIntegrationUpdate

Commented [LC105]: The AWS::AutoScaling::AutoScalingGroup resource supports an UpdatePolicy attribute. This is used to define how an Auto Scaling group resource is updated when an update to the Cloud Formation stack occurs. A common approach to updating an Auto Scaling group is to perform a rolling update, which is done by specifying the AutoScalingRollingUpdate policy. This retains the same Auto Scaling group and replaces old instances with new ones, according to the parameters specified. For more information on Autoscaling updates, please refer to the below link.

Reference:
<https://aws.amazon.com/premiumsupport/knowledge-center/auto-scaling-group-rolling-updates/>

Question #208

A business operates a stateless web application that receives infrequent traffic. The application is deployed through AWS CloudFormation. The application is hosted on Amazon EC2 On-Demand Instances and is protected by a load balancer (ALB). Multiple Availability Zones are used to host the instances.

The organization want to include Spot Instances while maintaining a limited number of On-Demand Instances to guarantee the application maintains a high level of availability.

Which approach is the MOST cost-effective in terms of meeting these requirements?

- A. Add a Spot block resource to the AWS CloudFormation template. Use the diversified allocation strategy with step scaling behind the ALB.
- B. Add a Spot block resource to the AWS CloudFormation template. Use the lowest-price allocation strategy with target tracking scaling behind the ALB.
- **C. Add a Spot Fleet resource to the AWS CloudFormation template. Use the capacity-optimized allocation strategy with step scaling behind the ALB.**
- D. Add a Spot Fleet resource to the AWS CloudFormation template. Use the diversified allocation strategy with scheduled scaling behind the ALB.

Commented [LC106]: Go for C.

<https://aws.amazon.com/blogs/compute/introducing-the-capacity-optimized-allocation-strategy-for-amazon-ec2-spot-instances/>

There is a price associated with interruptions, restarting work, and checkpointing. While the overall hourly cost of capacity-optimized allocation strategy might be slightly higher, the possibility of having fewer interruptions can lower the overall cost of your workload.

Question #209 [SKIP]

You need a DynamoDB-backed API to remain operational in the event of a complete regional AWS outage. You may suffer a few minutes of lag or slowness after a significant failure event; nevertheless, the system should resume normal functioning following those few minutes.

What is a prudent course of action?

- A. Set up DynamoDB cross-region replication in a master-standby configuration, with a single standby in another region. Create an Auto Scaling Group behind an ELB in each of the two regions DynamoDB is running in. Add a Route53 Latency DNS Record with DNS Failover, using the ELBs in the two regions as the resource records.
- **B. Set up a DynamoDB Multi-Region table. Create an Auto Scaling Group behind an ELB in each of the two regions DynamoDB is running in. Add a Route53 Latency DNS Record with DNS Failover, using the ELBs in the two regions as the resource records.**
- C. Set up a DynamoDB Multi-Region table. Create a cross-region ELB pointing to a cross-region Auto Scaling Group, and direct a Route53 Latency DNS Record with DNS Failover to the crossregion ELB.
- D. Set up DynamoDB cross-region replication in a master-standby configuration, with a single standby in another region. Create a cross-region ELB pointing to a cross-region Auto Scaling Group, and direct a Route53 Latency DNS Record with DNS Failover to the cross-region ELB.

Commented [LC107]: Old because now it's DynamoDB Global Tables

I'll go with A B) required a lot of changes and the DNS can take longer to propagate C) API gateway cannot choose the target group, this is done by the ELB D) ELB Target groups changes require some work, you can just point to a new load balancer

Question #210

The Security team has instructed a DevOps Engineer to guarantee that AWS CloudTrail files are not altered with once they are produced. At the moment, there is a procedure in place that utilizes several trails and AWS IAM to limit access to certain trails. The security team wants to verify that they can track the integrity of each file and ascertain that no manipulation has occurred.

Which method requires the LEAST work to deploy and secure the file's legality while enabling the security team to verify the logs' authenticity?

- A. Create an Amazon CloudWatch Events rule that triggers an AWS Lambda function when a new file is delivered. Configure the Lambda function to perform an MD5 hash check on the file, store the name and location of the file, and post the returned hash to an Amazon DynamoDB table. The Security team can use the values stored in DynamoDB to verify the file authenticity.
- B. Enable the CloudTrail file integrity feature on an Amazon S3 bucket. Create an IAM policy that grants the Security team access to the file integrity logs stored in the S3 bucket.
- C. Enable the CloudTrail file integrity feature on the trail. Use the digest file created by CloudTrail to verify the integrity of the delivered CloudTrail files.
- D. Create an AWS Lambda function that is triggered each time a new file is delivered to the CloudTrail bucket. Configure the Lambda function to execute an MD5 hash check on the file, and store the result on a tag in an Amazon S3 object. The Security team can use the information on the tag to verify the integrity of the file.

Commented [LC108]: <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-intro.html>

Question #211

A DevOps Engineer has been tasked with the responsibility of recommending a solution for deploying the components of a three-tiered web application. The database for this application will be Amazon DynamoDB.

Which deployment needs the LESS MANAGEMENT?

- A. Use AWS CloudFormation to create a Classic Load Balancer and an Auto Scaling group. Use AWS OpsWorks to create the application and database resources. Deploy application updates with OpsWorks using lifecycle events.
- B. Use AWS OpsWorks to create a Classic Load Balancer, an Auto Scaling group application, and database resources. Deploy application updates using OpsWorks lifecycle events.
- C. Use AWS OpsWorks to create a Classic Load Balancer, Auto Scaling, and application resources. Use AWS CloudFormation to create the database resources. Deploy application updates using CloudFormation rolling updates.
- D. Use AWS CloudFormation to create a Classic Load Balancer, an Auto Scaling group, and database resources. Deploy application updates using CloudFormation rolling updates.

Commented [LC109]: D is preferable. AWS OpsWorks Stacks provides integrated support for MySQL servers through the MySQL layer and for several types of database servers through the Amazon Relational Database Service (Amazon RDS) layer. However, you can easily customize a stack to have the application servers use other database servers such as Amazon DynamoDB or MongoDB. Since DDB is not integrated with OpsWorks, looks like this will have more management overhead than CFN templates.

Question #212

Which command in the AWS CLI fetches CloudTrail trail settings, including the trail's status?

- A. `aws cloudtrail return-trails`
- B. `aws cloudtrail validate-settings`
- C. `aws cloudtrail get-settings`
- D. `aws cloudtrail describe-trails`

Commented [LC110]: D is correct.

<https://docs.aws.amazon.com/cli/latest/reference/cloudtrail/index.html>

Question #213

A business is transferring an application to Amazon Web Services (AWS), where it will operate on a single Amazon EC2 instance. The program does not enable horizontal scalability due to license constraints. The application's database will be powered by Amazon Aurora.

How can a DevOps Engineer construct automated healing to recover automatically from EC2 and Aurora failures, as well as recover across Availability Zones (AZs), in the MOST cost-effective way possible?

- A. Create an EC2 Auto Scaling group with a minimum and maximum instance count of 1, and have it span across AZs. Use a single-node Aurora instance.
- B. Create an EC2 instance and enable instance recovery. Create an Aurora database with a read replica in a second AZ, and promote it to a primary database instance if the primary database instance fails.
- C. Create an Amazon CloudWatch Events rule to trigger an AWS Lambda function to start a new EC2 instance in an available AZ when the instance status reaches a failure state. Create an Aurora database with a read replica in a second AZ, and promote it to a primary database instance when the primary database instance fails.
- D. Assign an Elastic IP address on the instance. Create a second EC2 instance in a second AZ. Create an Amazon CloudWatch Events rule to trigger an AWS Lambda function to move the Elastic IP address to the second instance when the first instance fails. Use a single-node Aurora instance.

Commented [LC111]: C is correct. Let's break it down:
A- Single Aurora instance only supports auto recovery in the same AZ. Nope
B- EC2 auto recovery only supports recovery from system failure due to underlying hardware. Nope
C- Correct
D- Having two EC2 instances running at the same time is not cost-effective at all. Nope.

Question #214

A business hosts an application on Amazon EC2 instances that are part of an Auto Scaling group. Recently, a problem stopped EC2 instances from correctly starting, and it took the Support staff many hours to identify the issue. The Support team would want to get an email notification anytime an EC2 instance fails to start successfully.

Which action is necessary to achieve this?

- A. Add a health check to the Auto Scaling group to invoke an AWS Lambda function whenever an instance status is impaired.
- B. Configure the Auto Scaling group to send a notification to an Amazon SNS topic whenever a failed instance launch occurs.
- C. Create an Amazon CloudWatch alarm that invokes an AWS Lambda function when a failed AttachInstances Auto Scaling API call is made.
- D. Create a status check alarm on Amazon EC2 to send a notification to an Amazon SNS topic whenever a status check fail occurs.

Commented [LC112]: EC2 can't send notifications but Auto Scaling can.
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/ASGettingNotifications.html#auto-scaling-sns-notifications>

Question #215

A DevOps Engineer is designing a deployment approach that will enable data-driven choices prior to approving a feature for general availability. Currently, the deployment method makes use of AWS CloudFormation and blue/green deployment styles. The development team has agreed that rather than employing a fixed proportion, consumers should be randomly allocated to groups, and that redirects should be avoided.

How should the new deployment strategy be implemented?

- A. Configure Amazon Route 53 weighted records for the blue and green stacks, with 50% of traffic configured to route to each stack.
- B. Configure Amazon CloudFront with an AWS Lambda@Edge function to set a cookie when CloudFront receives a request. Assign the user to a version A or B, and configure the web server to redirect to version A or B.
- C. Configure Amazon CloudFront with an AWS Lambda@Edge function to set a cookie when CloudFront receives a request. Assign the user to a version A or B, then return the corresponding version to the viewer.
- D. Configure Amazon Route 53 with an AWS Lambda function to set a cookie when Amazon CloudFront receives a request. Assign the user to version A or B, then return the corresponding version to the viewer.

Commented [LC113]: <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/lambda-examples.html#lambda-examples-a-b-testing>

Question #216

A business hosts an application on Amazon EC2 instances that are routed via an Application Load Balancer. The instances are distributed across several Availability Zones in us-east-1 using an Amazon EC2 Auto Scaling group. The program stores data on a MySQL Multi-AZ DB instance hosted by Amazon RDS.

A DevOps engineer wants to alter the present solution and establish a hot standby environment in another location in order to reduce downtime in the event of an outage in our region.

Which measures should the DevOps engineer perform in combination to achieve these requirements? (Select three.)

- **A.** Add a health check to the Amazon Route 53 alias record to evaluate the health of the primary region. Use AWS Lambda, configured with an Amazon CloudWatch Events trigger, to promote the Amazon RDS read replica in the disaster recovery region.
- **B.** Create a new Application Load Balancer and Amazon EC2 Auto Scaling group in the disaster recovery region.
- C. Extend the current Amazon EC2 Auto Scaling group to the subnets in the disaster recovery region.
- D. Enable multi-region failover for the RDS configuration for the database instance.
- **E.** Deploy a read replica of the RDS instance in the disaster recovery region.
- F. Create an AWS Lambda function to evaluate the health of the primary region. If it fails, modify the Amazon Route 53 record to point at the disaster recovery region and promote the RDS read replica.

Commented [LC114]:

Commented [LC115]:

Commented [LC116]:

Question #217

A DevOps Engineer builds an Amazon EC2 instance with public IP addresses on a public subnet to execute an application. A user data script retrieves and installs the application artifacts on the instances upon start. Due to a change in the application's security categorization, instances must now operate without access to the Internet. While the instances start and seem to be healthy, the program does not appear to have been installed.

Which of the following should install the program successfully while adhering to the new rule?

- A. Launch the instances in a public subnet with Elastic IP addresses attached. Once the application is installed and running, run a script to disassociate the Elastic IP addresses afterwards.
- B. Set up a NAT gateway. Deploy the EC2 instances to a private subnet. Update the private subnet's route table to use the NAT gateway as the default route.
- **C.** Publish the application artifacts to an Amazon S3 bucket and create a VPC endpoint for S3. Assign an IAM instance profile to the EC2 instances so they can read the application artifacts from the S3 bucket.
- D. Create a security group for the application instances and whitelist only outbound traffic to the artifact repository. Remove the security group rule once the install is complete.

Commented [LC117]:

Question #218

A corporation wishes to move its Amazon EC2-hosted content sharing web application to a serverless architecture. Currently, the firm makes modifications to its application by establishing a new Auto Scaling group of EC2 instances and an Elastic Load Balancer, and then redirecting traffic away from the application using an Amazon Route 53 weighted routing policy.

The business intends to employ Amazon API Gateway and AWS Lambda for its new serverless application. The company's deployment methods will need to be updated to accommodate the new application. Additionally, it must preserve the ability to test new features on a subset of users prior to spreading them out to the full user base.

Which deployment technique will satisfy these criteria?

- A. Use AWS CDK to deploy API Gateway and Lambda functions. When code needs to be changed, update the AWS CloudFormation stack and deploy the new version of the APIs and Lambda functions. Use a Route 53 failover routing policy for the canary release strategy.
- B. Use AWS CloudFormation to deploy API Gateway and Lambda functions using Lambda function versions. When code needs to be changed, update the CloudFormation stack with the new Lambda code and update the API versions using a canary release strategy. Promote the new version when testing is complete.
- C. Use AWS Elastic Beanstalk to deploy API Gateway and Lambda functions. When code needs to be changed, deploy a new version of the API and Lambda functions. Shift traffic gradually using an Elastic Beanstalk blue/green deployment.
- D. Use AWS OpsWorks to deploy API Gateway in the service layer and Lambda functions in a custom layer. When code needs to be changed, use OpsWorks to perform a blue/green deployment and shift traffic gradually.

Commented [LC118]: B noting SAM is built over CloudFormation.

Question #219

The security team of a corporation learns that IAM access keys were leaked in a publicly accessible code repository. The DevOps team want to build a solution in the future that would automatically deactivate any keys suspected of being hacked and inform the security team.

Which approach is most likely to do this?

- A. Create an Amazon CloudWatch Events event for Amazon Macie. Create an Amazon SNS topic with two subscriptions: one to notify the security team and another to trigger an AWS Lambda function that disables the access keys.
- B. Enable Amazon GuardDuty and set up an Amazon CloudWatch Events rule event for GuardDuty. Trigger an AWS Lambda function to check if the event relates to compromised keys. If so, send a notification to the security team and disable the access keys.
- C. Run an AWS CloudWatch Events rule every 5 minutes to invoke an AWS Lambda function that checks to see if the compromised tag for any access key is set to true. If so, notify the security team and disable the access keys.
- D. Set up AWS Config and create an AWS CloudTrail event for AWS Config. Create an Amazon SNS topic with two subscriptions: one to notify the security team and another to trigger an AWS Lambda function that disables the access keys.

Commented [LC119]: Not sure, may be C this time.

Ref. <https://hands-on-guardduty.awssecworkshops.com/scenario2/>

Question #220

The AWS CloudFormation template for a mission-critical business application was changed by a corporation. Due to a mistake in the revised template, the stack update process failed, and CloudFormation initiated the stack rollback process automatically. A DevOps engineer later discovered that the application remained inaccessible and that the stack was in the UPDATE ROLLBACK FAILED state.

Which combination of steps will effectively execute the stack rollback? (Select two.)

- A. Attach the AWSCloudFormationFullAccess IAM policy to the AWS CloudFormation role.
- B. Automatically heal the stack resources using AWS CloudFormation drift detection.
- C. Issue a ContinueUpdateRollback command from the AWS CloudFormation console or AWS CLI.
- D. Manually adjust the resources to match the expectations of the stack.
- E. Update the existing AWS CloudFormation stack using the original template.

Commented [LC120]: I'll go with C, D
<https://docs.aws.amazon.com/cli/latest/reference/cloudformation/continue-update-rollback.html>

For a specified stack that is in the UPDATE_ROLLBACK_FAILED state, continues rolling it back to the UPDATE_ROLLBACK_COMPLETE state. Depending on the cause of the failure, you can manually fix the error and continue the rollback. By continuing the rollback, you can return your stack to a working state (the UPDATE_ROLLBACK_COMPLETE state), and then try to update the stack again.

Commented [LC121]:

Question #221

Currently, your organization is running a massive multi-tier web application. One component is an API service upon which the rest of your application's components depend to accomplish read/write activities. This service must be highly available and have a zero-downtime policy during deployments.

Which strategy should you employ to ensure that this component is deployed cost-effectively and without downtime?

- A. Use an AWS CloudFormation template to re-deploy your application behind a load balancer, and launch a new AWS CloudFormation stack during each deployment. Update your load balancer to send traffic to the new stack, and then deploy your software. Leave your old stacks running, and tag their resources with the version for rollback.
- B. Re-deploy your application on Elastic Beanstalk. During deployment, create a new version of your application, and create a new environment running that version in Elastic Beanstalk. Finally, take advantage of the Elastic Beanstalk Swap CNAME operation to switch to the new environment.
- C. Re-deploy your application behind a load balancer that uses Auto Scaling groups. Create a new identical Auto Scaling group and associate it to your Amazon Route53 zone. Configure Amazon Route53 to auto-weight traffic over to the new Auto Scaling group when all instances are marked as healthy.
- D. Re-deploy your application behind a load balancer using an AWS OpsWorks stack and use AWS OpsWorks stack versioning, during deployment create a new version of your application, tell AWS OpsWorks to launch the new version behind your load balancer, and when the new version is launched, terminate the old AWS OpsWorks stack.

Commented [LC122]: I feel it is B. However none of the options are cost-effective. Immutable or Rolling with additional batches would be my choice, but that is not in the answer anyway.

Question #222

On top of AWS, a corporation want to employ a grid system to power a proprietary corporate in-memory data store. This system may be configured to operate on numerous server nodes on any Linux distribution. Every time a node is added or withdrawn, the system must be able to reconfigure the complete cluster. When nodes are added or removed, a `/etc./cluster/nodes.config` file must be modified to reflect the current IP addresses of the cluster's node members.

The business want to automate the process of adding nodes to a cluster.

How can a DevOps Engineer ensure that these needs are met?

- A. Use AWS OpsWorks Stacks to layer the server nodes of that cluster. Create a Chef recipe that populates the content of the `/etc/cluster/nodes.config` file and restarts the service by using the current members of the layer. Assign that recipe to the Configure lifecycle event.
- B. Put the file `nodes.config` in version control. Create an AWS CodeDeploy deployment configuration and deployment group based on an Amazon EC2 tag value for the cluster nodes. When adding a new node to the cluster, update the file with all tagged instances, and make a commit in version control. Deploy the new file and restart the services.
- C. Create an Amazon S3 bucket and upload a version of the `etc/cluster/nodes.config` file. Create a crontab script that will poll for that S3 file and download it frequently. Use a process manager, such as Monit or systemd, to restart the cluster services when it detects that the new file was modified. When adding a node to the cluster, edit the file's most recent members. Upload the new file to the S3 bucket.
- D. Create a user data script that lists all members of the current security group of the cluster and automatically updates the `/etc/cluster/nodes.config` file whenever a new instance is added to the cluster

Commented [LC123]: Right: A From AWS documentation on the OpsWorks Stacks "Configure lifecycle event": "This event occurs on all of the stack's instances when one of the following occurs: An instance enters or leaves the online state. You associate an Elastic IP address with an instance or disassociate one from an instance. You attach an Elastic Load Balancing load balancer to a layer, or detach one from a layer. For example, suppose that your stack has instances A, B, and C, and you start a new instance, D. After D has finished running its setup recipes, AWS OpsWorks Stacks triggers the Configure event on A, B, C, and D. If you subsequently stop A, AWS OpsWorks Stacks triggers the Configure event on B, C, and D. AWS OpsWorks Stacks responds to the Configure event by running each layer's Configure recipes, which update the instances' configuration to reflect the current set of online instances." It is exactly our situation here, instances are added/removed and the Configure lifecycle needs to update the node info on each /etc.

Question #223

Which big database requires a bring-your-own license?

- A. PostgreSQL
- B. MariaDB
- C. MySQL
- D. Oracle

Commented [LC124]: Oracle is not open source, and requires a bring your own license model.

Reference:
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Oracle.html

Question #224

A firm that uses AWS CodeCommit for source control wishes to automate its continuous integration and deployment pipelines in its development environment on AWS. Three criteria apply to the business:

1. Any code update must undergo a legal and security assessment to ensure that no sensitive information is disclosed via the source code.
 2. Each modification must be subjected to unit testing.
 3. Each modification must be subjected to a battery of functional tests to confirm its functioning.
- Additionally, the organization has the following automation requirements:
1. Code changes should initiate the CI/CD pipeline automatically.
 2. Any pipeline failures should be reported to devops-admin@xyz.com.
 3. After testing are completed, authorisation must be obtained to transfer the assets to Amazon S3.

What should a DevOps Engineer do to ensure that all of these needs are met while adhering to best practices in CI/CD?

- A. Commit to the development branch and trigger AWS CodePipeline from the development branch. Make an individual stage in CodePipeline for security review, unit tests, functional tests, and manual approval. Use Amazon CloudWatch metrics to detect changes in pipeline stages and Amazon SES for emailing devops-admin@xyz.com.
- B. Commit to mainline and trigger AWS CodePipeline from mainline. Make an individual stage in CodePipeline for security review, unit tests, functional tests, and manual approval. Use AWS CloudTrail logs to detect changes in pipeline stages and Amazon SNS for emailing devops-admin@xyz.com.
- **C. Commit to the development branch and trigger AWS CodePipeline from the development branch. Make an individual stage in CodePipeline for security review, unit tests, functional tests, and manual approval. Use Amazon CloudWatch Events to detect changes in pipeline stages and Amazon SNS for emailing devops-admin@xyz.com.**
- D. Commit to mainline and trigger AWS CodePipeline from mainline. Make an individual stage in CodePipeline for security review, unit tests, functional tests, and manual approval. Use Amazon CloudWatch Events to detect changes in pipeline stages and Amazon SES for emailing devops-admin@xyz.com.

Commented [LC125]:

Question #225

A business is allocating AWS charges through tagging. The organization utilizes Amazon EC2 instances that are scaled automatically using Auto Scaling groups. Amazon Elastic Block Store (Amazon EBS) volumes associated with EC2 instances are produced without the associated cost center tags. A DevOps engineer is responsible for appropriately tagging new EBS volumes.

Which approach is the MOST EFFECTIVE in meeting this requirement?

- A. Create a lifecycle hook on the autoscaling:EC2_INSTANCE_TERMINATING instance state that attaches the cost center tags to the EBS volumes.
- **B. Update the Auto Scaling group launch template to include the cost center tags for EBS volumes.**
- C. Update the Auto Scaling group to include the cost center tags. Set the PropagateAllLaunch property to true.
- D. Use Tag Editor to search for EBS volumes that are missing the tags and to add the cost center tags to the volumes.

Commented [LC126]: The answer is B. "Tags are not propagated to Amazon EBS volumes. To add tags to Amazon EBS volumes, specify the tags in a launch template..."
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/autoscaling-tagging.html>

Question #226

A DevOps Engineer is spearheading the adoption of AWS Systems Manager to automate the patching of Windows-based workstations in a hybrid cloud environment (SSM).

What actions should the Engineer take to configure Systems Manager in this environment to automate patching? (Select two.)

- A. Create multiple IAM service roles for Systems Manager so that the ssm.amazonaws.com service can execute the AssumeRole operation on every instance. Register the role on a per-resource level to enable the creation of a service token. Perform managed-instance activation with the newly created service role attached to each managed instance.
- B. Create an IAM service role for Systems Manager so that the ssm.amazonaws.com service can execute the AssumeRole operation. Register the role to enable the creation of a service token. Perform managed-instance activation with the newly created service role.
- C. Using previously obtained activation codes and activation IDs, download and install the SSM Agent on the hybrid servers, and register the servers or virtual machines on the Systems Manager service. Hybrid instances will show with a "mi-" prefix in the SSM console.
- D. Using previously obtained activation codes and activation IDs, download and install the SSM Agent on the hybrid servers, and register the servers or virtual machines on the Systems Manager service. Hybrid instances will show with an "i-" prefix in the SSM console as if they were provisioned as a regular Amazon EC2 instance.
- E. Run AWS Config to create a list of instances that are unpatched and not compliant. Create an instance scheduler job, and through an AWS Lambda function, perform the instance patching to bring them up to compliance.

Commented [LC127]: I'll go with B,C References:

<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-service-role.html>

In the console, however, the IDs of your hybrid instances are distinguished from EC2 instances with the prefix "mi-". EC2 instance IDs use the prefix "i-".

Question #227

According to reports, an ecommerce company's order history website is experiencing difficulties in reporting the processing status of purchases. The order processing system is a Lambda function on AWS that utilizes restricted concurrency. The Lambda function receives order messages from an Amazon SQS queue and processes them before inserting them into an Amazon DynamoDB database. For read and write capacity, the DynamoDB table supports Auto Scaling.

Which steps will be taken to determine the cause of the delay and to fix it? (Make a selection of at least two.)

- A. Check the ApproximateAgeOfOldestMessage metric for the SQS queue and increase the Lambda function concurrency limit.
- B. Check the ApproximateAgeOfOldestMessage metric for the SQS queue and configure a redrive policy on the SQS queue.
- C. Check the NumberOfMessagesSent metric for the SQS queue and increase the SQS queue visibility timeout.
- D. Check the ThrottledWriteRequests metric for the DynamoDB table and increase the maximum write capacity units for the table's Auto Scaling policy.
- E. Check the Throttles metric for the Lambda function and increase the Lambda function timeout.

Commented [LC128]: AD is the answer.

A: If the ApproximateAgeOfOldestMessages indicate that orders are remaining in the SQS queue for longer than expected, the reserved concurrency limit may be set too small to keep up with the number of orders entering the queue and is being throttled.

D: The DynamoDB table is using Auto Scaling. With Auto Scaling, you create a scaling policy that specifies whether you want to scale read capacity or write capacity (or both), and the minimum and maximum provisioned capacity unit settings for the table. The ThrottledWriteRequests metric will indicate if there is a throttling issue on the DynamoDB table, which can be resolved by increasing the maximum write capacity units for the table's Auto Scaling policy.

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html>

Commented [LC129]:

Question #228

A DevOps Engineer is responsible for the administration of a web application that is hosted on Amazon EC2 instances behind an Application Load Balancer (ALB). The instances are distributed across several Availability Zones through an EC2 Auto Scaling group. The engineer must implement a deployment plan that includes the following:

- ⇒ This command creates a second fleet of instances with the same capacity as the first.
- ⇒ Maintains the previous fleet intact while launching the second.
- ⇒ When the second fleet is completely deployed, it transitions traffic to the second fleet.
- ⇒ Automatically terminates the old fleet one hour after changeover.

Which solution will meet these criteria?

- A. Use an AWS CloudFormation template with a retention policy for the ALB set to 1 hour. Update the Amazon Route 53 record to reflect the new ALB.
- B. Use two AWS Elastic Beanstalk environments to perform a blue/green deployment from the original environment to the new one. Create an application version lifecycle policy to terminate the original environment in 1 hour.
- C. Use AWS CodeDeploy with a deployment group configured with a blue/green deployment configuration. Select the option Terminate the original instances in the deployment group with a waiting period of 1 hour.
- D. Use AWS Elastic Beanstalk with the configuration set to Immutable. Create a .ebextension using the Resources key that sets the deletion policy of the ALB to 1 hour, and deploy the application.

Commented [LC130]: https://docs.aws.amazon.com/codedeploy/latest/APIReference/API_BlueInstanceTerminationOption.html

Question #229

You have been tasked with testing a three-tier web architecture created in an AWS CloudFormation template using your department's current continuous Integration (CI) technology. After querying version control, the tool already supports AWS APIs and can start new AWS CloudFormation stacks. The CI tool verifies that the AWS CloudFormation stack was successfully created by querying the Describe Stacks API for the CREATE COMPLETE state. The template defines the following architectural tiers:

- One load balancer
- Five Amazon EC2 instances running the web application
- One multi-AZ Amazon RDS instance

How would you go about doing this? (Select two.)

- A. Define a WaitCondition and a WaitConditionHandle for the output of a UserData command that does sanity checking of the application's post-install state.
- B. Define a CustomResource and write a script that runs architecture-level Integration tests through the load balancer to the application and database for the state of multiple tiers.
- C. Define a WaitCondition and use a WaitConditionHandle that leverages the AWS SDK to run the DescribeStacks API call until the CREATE COMPLETE status is returned.
- D. Define a CustomResource that leverages the AWS SDK to run the DescribeStacks API call until the 'CREATE COMPLETE' status is returned.
- E. Define a UserDataHandle for the output of a UserData command that does sanity checking of the application's post-install state and runs integration tests on the state of multiple tiers through the load balancer to the application.
- F. Define a UserDataHandle for the output of a CustomResource that does sanity checking of the application's post-install state.

Commented [LC131]: The CI tool already verifies that the AWS CloudFormation stack completion by checking CREATE_COMPLETE state.

So, Ans C and D are duplicate verifications and not applicable.

There is nothing called UserDataHandle.

Ans E and F are distractions. The only applicable answers are A (sanity checking of the application's post-install state) and B (verify architecture-level Integration tests through the load balancer to the application and database for the state of multiple tiers).

Question #230

You have an application that is made up of EC2 instances that are part of an Auto Scaling group. Each day, there is an upsurge in traffic to your website within a certain time period. As a result, people complain about the application's slow reaction time. You've set your Auto Scaling group to create one new EC2 instance when CPU use exceeds 60% for two consecutive 5-minute intervals.

What is the most cost-effective method of resolving this issue?

- A. Decrease the consecutive number of collection periods.
- B. Increase the minimum number of instances in the Auto Scaling group.
- C. Decrease the collection period to ten minutes.
- D. Decrease the threshold CPU utilization percentage at which to deploy a new instance.

Commented [LC132]:

Question #231

Your business provides a website via which promoters may sell tickets to entertainment events. A load balancer is placed in front of an Auto Scaling set of web servers. Promotion of major events might result in spikes in website traffic.

At certain moments of scaling-out, freshly launched instances are unable to finish setup in a timely manner, resulting in customer displeasure.

Which choices should you pursue in order to maximize scalability while minimizing costs? (Select two.)

- A. Create an AMI with the application pre-configured. Create a new Auto Scaling launch configuration using this new AMI, and configure the Auto Scaling group to launch with this AMI.
- B. Use Auto Scaling pre-warming to launch instances before they are required. Configure pre-warming to use the CPU trend CloudWatch metric for the group.
- C. Publish a custom CloudWatch memo from your application on the number of tickets sold, and create an Auto Scaling policy based on this.
- D. Use the history of past scaling events for similar event sales to predict future scaling requirements. Use the Auto Scaling scheduled scaling feature to vary the size of the fleet.
- E. Configure an Amazon S3 bucket for website hosting. Upload into the bucket an HTML holding page with its x-amz-website-redirect-location' metadata property set to the load balancer endpoint. Configure Elastic Load Balancing to redirect to the holding page when the load on web servers is above a certain level.

Commented [LC133]: AD is legit. We can prebake AMIs, and we can use the past to try to schedule the building of resources ahead of traffic spikes.

B: Pre-warming is for when you know you are about to experience load, and you need to make sure the resources are going to get used quick. This is not that situation.

C: basing a scaling policy on a finite resource won't help. E is not legit either, hosting on S3 is only good for static sites.

Commented [LC134]:

Question #232

Which of the following is not an intrinsic AWS CloudFormation function?

- A. Fn::Equals
- B. Fn::If
- C. Fn::Not
- **D. Fn::Parse**

Commented [LC135]: This is the complete list of Intrinsic Functions:

Fn::Base64
Fn::And
Fn::Equals
Fn::If
Fn::Not
Fn::Or
Fn::FindInMap
Fn::GetAtt
Fn::GetAZs
Fn::Join
Fn::Select

Reference:
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-functionreference.html>

Commented [LC136]:

Question #233

Using AWS CodeDeploy blue/green deployments, a development team coordinates website deployments. The application is hosted on Amazon EC2 instances in an Auto Scaling group behind an Application Load Balancer.

When deploying a new version, the team observes that the deployment finally fails, but takes an extended period of time to do so. Further investigation reveals that the AllowTraffic lifecycle event ran for an hour before failing without giving any more information. The team's goal is to guarantee that failure notifications are given more promptly while yet ensuring application availability in the event of a failure.

Which measures should be conducted in combination to achieve these requirements? (Select two.)

- A. Change the deployment configuration to CodeDeployDefaultAllAtOnce to speed up the deployment process by deploying to all of the instances at the same time.
- **B. Create a CodeDeploy trigger for the deployment failure event and make the deployment fail as soon as a single health check failure is detected.**
- C. Reduce the HealthCheckIntervalSeconds and UnhealthyThresholdCount values within the target group health checks to decrease the amount of time it takes for the application to be considered unhealthy.
- D. Use the appspec.yml file to run a script on the AllowTraffic hook to perform lighter health checks on the application instead of making CodeDeploy wait for the target group health checks to pass.
- **E. Use the appspec.yml file to run a script on the BeforeAllowTraffic hook to perform health checks on the application and fail the deployment if the health checks performed by the script are not successful.**

Commented [LC137]:

Question #234

A DevOps Engineer must automate the creation of a Linux AMI. The newly produced AMI identity must be kept in a place accessible to other build processes programmatically.

What is the MOST cost-effective method of doing this?

- A. Build a pipeline in AWS CodePipeline to download and save the latest operating system Open Virtualization Format (OVF) image to an Amazon S3 bucket, then customize the image using the guestfish utility. Use the virtual machine (VM) import command to convert the OVF to an AMI, and store the AMI identification output as an AWS Systems Manager parameter.
- **B. Create an AWS Systems Manager automation document with values instructing how the image should be created. Then build a pipeline in AWS CodePipeline to execute the automation document to build the AMI when triggered. Store the AMI identification output as a Systems Manager parameter.**
- C. Build a pipeline in AWS CodePipeline to take a snapshot of an Amazon EC2 instance running the latest version of the application. Then start a new EC2 instance from the snapshot and update the running instance using an AWS Lambda function. Take a snapshot of the updated instance, then convert it to an AMI. Store the AMI identification output in an Amazon DynamoDB table.
- D. Launch an Amazon EC2 instance and install Packer. Then configure a Packer build with values defining how the image should be created. Build a Jenkins pipeline to invoke the Packer build when triggered to build an AMI. Store the AMI identification output in an Amazon DynamoDB table.

Commented [LC138]: Answer B. This is a tricky question. Don't get confused with word packer. The AWS will always recommend to use its own services if available rather than using third party services. I strongly believe that Answer is B as we can use AWS-provided SSM document that automates the process of running Packer.

<https://docs.aws.amazon.com/systems-manager-automation-runbooks/latest/userguide/automation-aws-runpacker.html>

<https://aws.amazon.com/blogs/mt/creating-packer-images-using-system-manager-automation/>

A - lot of additional effort
B - looks correct
C - additional dynamodb cost
D - additional cost involved

Question #235

You are in charge of a large-scale video transcoding system that employs an Auto Scaling set of video transcoders. A minimum of 750 Amazon EC2 instances and a maximum of 1000 Amazon EC2 instances are specified in the Auto Scaling group. You're using Amazon SQS to send a message to the transcoding workers that contains the URI for a movie stored in Amazon S3. You've been warned by an Amazon CloudWatch alarm that the queue depth is approaching a critical level.

How can you silence the alert without extending the time required to convert videos? (Select two.)

- A. Create a second queue in Amazon SQS.
- B. Adjust the Amazon CloudWatch alarms for a higher queue depth.
- **C. Create a new Auto Scaling group with a launch configuration that has a larger Amazon EC2 instance type.**
- D. Add an additional Availability Zone to the Auto Scaling group configuration.
- E. Change the Amazon CloudWatch alarm so that it monitors the CPU utilization of the Amazon EC2 instances rather than the Amazon SQS queue depth.
- **F. Adjust the Auto Scaling group configuration to increase the maximum number of Amazon EC2 instances.**

Commented [LC139]:

Commented [LC140]:

Question #236

You work for a company that has created a new mobile photo-sharing application. Recently, your program has grown in popularity; this has resulted in a decline in the application's performance as a consequence of the increasing load. Your application is made of two tiers: an Auto Scaling PHP application tier and a MySQL RDS instance that was first deployed using AWS CloudFormation. Your Auto Scaling group includes a minimum of four and a maximum of eight members. Due to the instances' excessive CPU use, the required capacity has been increased to eight. Following some investigation, you are convinced that the performance concerns are due to a CPU capacity restriction, despite the fact that memory use remains low. As a result, you choose to migrate from general-purpose M3 instances to compute-optimized C3 instances.

How would you implement this change with the least amount of disruption to your end users?

- A. Sign into the AWS Management Console, copy the old launch configuration, and create a new launch configuration that specifies the C3 instances. Update the Auto Scaling group with the new launch configuration. Auto Scaling will then update the instance type of all running instances.
- B. Sign into the AWS Management Console, and update the existing launch configuration with the new C3 instance type. Add an UpdatePolicy attribute to your Auto Scaling group that specifies AutoScalingRollingUpdate.
- C. Update the launch configuration specified in the AWS CloudFormation template with the new C3 instance type. Run a stack update with the new template. Auto Scaling will then update the instances with the new instance type.
- **D. Update the launch configuration specified in the AWS CloudFormation template with the new C3 instance type. Also add an UpdatePolicy attribute to your Auto Scaling group that specifies AutoScalingRollingUpdate. Run a stack update with the new template.**

Commented [LC141]: Ans is D
<https://cloudonaut.io/rolling-update-with-aws-cloudformation/>

Question #237

A development team is presently deploying an application version to an Auto Scaling group through AWS CodeDeploy. If the deployment process fails, it must be immediately turned back and a message given.

Which setup is the MOST EFFECTIVE in terms of meeting all of the requirements?

- **A. Create Amazon CloudWatch Events rules for CodeDeploy operations. Configure a CloudWatch Events rule to send out an Amazon SNS message when the deployment fails. Configure CodeDeploy to automatically roll back when the deployment fails.**
- B. Use available Amazon CloudWatch metrics for CodeDeploy to create CloudWatch alarms. Configure CloudWatch alarms to send out an Amazon SNS message when the deployment fails. Use AWS CLI to redeploy a previously deployed revision.
- C. Configure a CodeDeploy agent to create a trigger that will send notification to Amazon SNS topics when the deployment fails. Configure CodeDeploy to automatically roll back when the deployment fails.
- D. Use AWS CloudTrail to monitor API calls made by or on behalf of CodeDeploy in the AWS account. Send an Amazon SNS message when deployment fails. Use AWS CLI to redeploy a previously deployed revision.

Commented [LC142]: I'll go with A, because CodeDeploy Agent can't trigger SNS, this question looks like a trick.

Reference:
<https://docs.aws.amazon.com/codedeploy/latest/userguide/monitoring-cloudwatch-events.html>

Question #238

An IT department oversees a portfolio of servers, both on-premises and in the cloud, running Windows and Linux (Amazon and Red Hat Enterprise Linux). An audit indicates that there is no mechanism in place for patching the operating system and core applications, and that patch levels on the servers are inconsistent.

Which of the following offers the MOST dependable and consistent process for upgrading and maintaining all servers' operating systems and key applications to the most current patch levels?

- A. Install AWS Systems Manager agent on all on-premises and AWS servers. Create Systems Manager Resource Groups. Use Systems Manager Patch Manager with a preconfigured patch baseline to run scheduled patch updates during maintenance windows.
- B. Install the AWS OpsWorks agent on all on-premises and AWS servers. Create an OpsWorks stack with separate layers for each operating system, and get a recipe from the Chef supermarket to run the patch commands for each layer during maintenance windows.
- C. Use a shell script to install the latest OS patches on the Linux servers using yum and schedule it to run automatically using cron. Use Windows Update to automatically patch Windows servers.
- D. Use AWS Systems Manager Parameter Store to securely store credentials for each Linux and Windows server. Create Systems Manager Resource Groups. Use the Systems Manager Run Command to remotely deploy patch updates using the credentials in Systems Manager Parameter Store.

Commented [LC143]: Option "A"
<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html>
<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html>

Question #239

Multiple development teams within your business are actively working on a web application. You've built a self-service site powered by AWS CloudFormation and AWS APIs that enables testers to choose a development branch containing a new feature to test. After that, the portal will setup an environment and deploy the appropriate branch of code. You may have observed recently that a significant percentage of environments have broken builds. You want to implement a set of automated browser tests that are run on a new environment prior to the tester having access to it. This prevents a tester from wasting time testing new features in a broken environment.

Choose an appropriate method for integrating this functionality into the current self-service portal:

- A. Specify your automated tests in the "tests" section of the AWS CloudFormation template. AWS CloudFormation will then execute the tests on your behalf as part of the environment build.
- B. Configure a centralized test server that hosts an automated browser testing framework. Use an AWS CloudFormation custom resource to notify the centralized test server, via an Amazon SNS topic, that a new environment has been initialized. The centralized test server can then execute the tests before sending the results back to the AWS CloudFormation service.
- C. Pass the test scripts to the cfn-init service via the "tests" section of the AWS::CloudFormation::Init metadata. Cfn-init will then execute these tests and return the result to the AWS CloudFormation service.
- D. Configure a centralized test server that hosts an automated browser testing framework. Include an Amazon SES email resource under the outputs section of your AWS CloudFormation template. This we send an email to your centralized test server, informing it that the environment is ready for tests.

Commented [LC144]: Ans is B: A: There is no "Tests" section in the cloudformation template as far as I am aware so this is incorrect. C: Again, there is no "Tests" section in the CF init

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-init.html>

so we left with B or D and the difference between them is SNS vs SES for notification and obviously, we are going to pick SNS for pushing out notification.

Question #240

The web application of a business will be transferred to AWS. The program is written in such a way that no server-side code is necessary. The organization wants to enhance the application's security as part of the migration by adding HTTP response headers in accordance with the Open Web Application Security Project's (OWASP) secure headers guidelines.

How can this solution be deployed in a secure manner while adhering to best practices?

- A. Use an Amazon S3 bucket configured for website hosting, then set up server access logging on the S3 bucket to track user activity. Then configure the static website hosting and execute a scheduled AWS Lambda function to verify, and if missing, add security headers to the metadata.
- B. Use an Amazon S3 bucket configured for website hosting, then set up server access logging on the S3 bucket to track user activity. Configure the static website hosting to return the required security headers.
- C. Use an Amazon S3 bucket configured for website hosting. Create an Amazon CloudFront distribution that refers to this S3 bucket, with the origin response event set to trigger a Lambda@Edge Node.js function to add in the security headers.
- D. Use an Amazon S3 bucket configured for website hosting. Create an Amazon CloudFront distribution that refers to this S3 bucket. Set 'Cache Based on Selected Request Headers' to 'Whitelist,' and add the security headers into the whitelist.

Commented [LC145]: Answer C is correct: <https://aws.amazon.com/blogs/networking-and-content-delivery/adding-http-security-headers-using-lambdaedge-and-amazon-cloudfront/>

Question #241

Your organization wants to do A/B testing on a new online feature for 20% of its visitors. CloudFront is used to provide the whole website, with some material cached for up to 24 hours.

How do you ensure that the needed percentage of users is included in the testing while reducing performance impact?

- A. Configure the web servers to handle two domain names. The feature is switched on or off depending on which domain name is used for a request. Configure a CloudFront origin for each domain name, and configure the CloudFront distribution to use one origin for 20 percent of users and the other origin for the other 80 percent.
- B. Configure the CloudFront distribution to forward a cookie specific to this feature. For requests where the cookie is not set, the web servers set its value to "on" for 20 percent of responses and "off" for 80 percent. For requests where the cookie is set, the web servers use its value to determine whether the feature should be on or off for the response.
- C. Create a second stack of web servers that host the website with the feature on. Using Amazon Route53, create two resource record sets with the same name: one with a weighting of "1" and a value of this new stack; the other a weighting of "4" and a value of the existing stack. Use the resource record set's name as the CloudFront distribution's origin.
- D. Invalidate all of the CloudFront distribution's cache items that the feature affects. On future requests, the web servers create responses with the feature on for 20 percent of users, and off for 80 percent. The web servers set "Cache-Control: no-cache" on all of these responses.

Commented [LC146]: B doesn't handle the cache invalidation part of the problem and D doesn't solve the cookie problem. None of the answers provides a complete solution.

Question #242

You are in charge of your organization's huge multi-tiered Windows-based online application, which is hosted on Amazon EC2 instances behind a load balancer.

While reviewing stats, you've seen an upward trend in the time it takes for a client page to load. Your boss has instructed you to devise a method for ensuring that client load time is not impacted by an excessive number of requests per second.

Which approach would you use in order to resolve this situation?

- A. Re-deploy your infrastructure using an AWS CloudFormation template. Configure Elastic Load Balancing health checks to initiate a new AWS CloudFormation stack when health checks return failed.
- B. Re-deploy your infrastructure using an AWS CloudFormation template. Spin up a second AWS CloudFormation stack. Configure Elastic Load Balancing SpillOver functionality to spill over any slow connections to the second AWS CloudFormation stack.
- C. Re-deploy your infrastructure using AWS CloudFormation, Elastic Beanstalk, and Auto Scaling. Set up your Auto Scaling group policies to scale based on the number of requests per second as well as the current customer load time.
- D. Re-deploy your application using an Auto Scaling template. Configure the Auto Scaling template to spin up a new Elastic Beanstalk application when the customer load time surpasses your threshold.

Commented [LC147]:

Question #243

A business uses Amazon ECS to deploy an application using data stored in an Amazon DynamoDB table. The firm wishes for the application to fail over to another Region in the event of a calamity. Additionally, the program must recover quickly from any inadvertent data loss occurrences. The application's RPO is one hour and its RTO is two hours.

Which option with a high level of availability should a DevOps engineer recommend?

- A. Change the configuration of the existing DynamoDB table. Enable this as a global table and specify the second Region that will be used. Enable DynamoDB point-in-time recovery.
- B. Enable DynamoDB Streams for the table and create an AWS Lambda function to write the stream data to an S3 bucket in the second Region. Schedule a job for every 2 hours to use AWS Data Pipeline to restore the database to the failover Region.
- C. Export the DynamoDB table every 2 hours using AWS Data Pipeline to an Amazon S3 bucket in the second Region. Use Data Pipeline in the second Region to restore the export from S3 into the second DynamoDB table.
- D. Use AWS DMS to replicate the data every hour. Set the original DynamoDB table as the source and the new DynamoDB table as the target.

Commented [LC148]:

Question #244

A business has many AWS accounts. Globally, the accounts are shared and used by many teams, particularly for Amazon EC2 instances. To guarantee correct cost allocations, each EC2 instance includes tags for team, environment, and cost center.

How can a DevOps Engineer assist teams in conducting cost audits and automating infrastructure cost minimization across various shared environments and accounts?

- A. Set up a scheduled script on the EC2 instances to report utilization and store the instances in an Amazon DynamoDB table. Create a dashboard in Amazon QuickSight with DynamoDB as the source data to find underutilized instances. Set up triggers from Amazon QuickSight in AWS Lambda to reduce underutilized instances.
- B. Create a separate Amazon CloudWatch dashboard for EC2 instance tags based on cost center, environment, and team, and publish the instance tags out using unique links for each team. For each team, set up a CloudWatch Events rule with the CloudWatch dashboard as the source, and set up a trigger to initiate an AWS Lambda function to reduce underutilized instances.
- C. Create an Amazon CloudWatch Events rule with AWS Trusted Advisor as the source for low utilization EC2 instances. Trigger an AWS Lambda function that filters out reported data based on tags for each team, environment, and cost center, and store the Lambda function in Amazon S3. Set up a second trigger to initiate a Lambda function to reduce underutilized instances.
- D. Use AWS Systems Manager to track instance utilization and report underutilized instances to Amazon CloudWatch. Filter data in CloudWatch based on tags for team, environment, and cost center. Set up triggers from CloudWatch into AWS Lambda to reduce underutilized instances.

Commented [LC149]: C looks good.

<https://aws.amazon.com/premiumsupport/technology/trusted-advisor/best-practice-checklist/>

Checks the Amazon Elastic Compute Cloud (Amazon EC2) instances that were running at any time during the last 14 days and alerts you if the daily CPU utilization was 10% or less and network I/O was 5 MB or less on 4 or more days. Running instances generate hourly usage charges. Although some scenarios can result in low utilization by design, you can often lower your costs by managing the number and size of your instances. Estimated monthly savings are calculated by using the current usage rate for On-Demand Instances and the estimated number of days the instance might be underutilized. Actual savings will vary if you are using Reserved Instances or Spot Instances, or if the instance is not running for a full day. To get daily utilization data, download the report for this check.

Question #245

At a nascent cloud-based gaming firm, a DevOps Engineer is responsible for formalizing deployment methodologies. The strategies must adhere to the following criteria:

- ⇒ For the code repository, use typical Git commands such as `git clone` and `git push`.
- ⇒ Wherever feasible, management tools should optimize the utilization of platform solutions.
- ⇒ Immutable deployment packages in the form of Docker images are required.

How is the Engineer to satisfy these specifications?

- A. Use AWS CodePipeline to trigger a build process when software is pushed to a self-hosted GitHub repository. CodePipeline will use a Jenkins build server to build new Docker images. CodePipeline will deploy into a second target group in Amazon ECS behind an Application Load Balancer. Cutover will be managed by swapping the listener rules on the Application Load Balancer.
- **B. Use AWS CodePipeline to trigger a build process when software is pushed to a private GitHub repository. CodePipeline will use AWS CodeBuild to build new Docker images. CodePipeline will deploy into a second target group in Amazon ECS behind an Application Load Balancer. Cutover will be managed by swapping the listener rules on the Application Load Balancer.**
- C. Use a Jenkins pipeline to trigger a build process when software is pushed to a private GitHub repository. AWS CodePipeline will use AWS CodeBuild to build new Docker images. CodePipeline will deploy into a second target group in Amazon ECS behind an Application Load Balancer. Cutover will be managed by swapping the listener rules on the Application Load Balancer.
- D. Use AWS CodePipeline to trigger a build process when software is pushed to an AWS CodeCommit repository. CodePipeline will use an AWS CodeBuild build server to build new Docker images. CodePipeline will deploy into a second target group in a Kubernetes Cluster hosted on Amazon EC2 behind an Application Load Balancer. Cutover will be managed by swapping the listener rules on the Application Load Balancer.

Commented [LC150]:

Question #246

Multiple development teams inside a corporation share a single AWS account. The manager of the development team wants to be able to automatically terminate Amazon EC2 instances and get alerts when resources are idle and untagged as production.

Which solution will satisfy these criteria?

- A. Use a scheduled Amazon CloudWatch Events rule to filter for Amazon EC2 instance status checks and identify idle EC2 instances. Use the CloudWatch Events rule to target an AWS Lambda function to stop non-production instances and send notifications.
- B. Use a scheduled Amazon CloudWatch Events rule to filter AWS Systems Manager events and identify idle EC2 instances and resources. Use the CloudWatch Events rule to target an AWS Lambda function to stop non-production instances and send notifications.
- **C. Use a scheduled Amazon CloudWatch Events rule to target a custom AWS Lambda function that runs AWS Trusted Advisor checks. Create a second CloudWatch Events rule to filter events from Trusted Advisor to trigger a Lambda function to stop idle non-production instances and send notifications.**
- D. Use a scheduled Amazon CloudWatch Events rule to target Amazon Inspector events for idle EC2 instances. Use the CloudWatch Events rule to target the AWS Lambda function to stop non-production instances and send notifications.

Commented [LC151]: I'll go with C

References:

<https://docs.aws.amazon.com/awssupport/latest/user/cloud-watch-ta.html>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/EventTypes.html#trusted-advisor-event-types>

<https://aws.amazon.com/premiumsupport/technology/trusted-advisor/>

Question #247

A DevOps Engineer wishes to prohibit developers from submitting modifications straight to the master branch of the company's AWS CodeCommit repository. These changes should be reviewed and authorized prior to being merged.

Which solution will satisfy these criteria?

- A. Configure an IAM role for the Developers with access to CodeCommit and an explicit deny for write actions when the reference is the master. Allow Developers to use feature branches and create a pull request when a feature is complete. Allow an approver to use CodeCommit to view the changes and approve the pull requests.
- B. Configure an IAM role for the Developers to use feature branches and create a pull request when a feature is complete. Allow CodeCommit to test all code in the feature branches, and dynamically modify the IAM role to allow merging the feature branches into the master. Allow an approver to use CodeCommit to view the changes and approve the pull requests.
- C. Configure an IAM role for the Developers to use feature branches and create a pull request when a feature is complete. Allow CodeCommit to test all code in the feature branches, and issue a new AWS Security Token Service (STS) token allowing a one-time API call to merge the feature branches into the master. Allow an approver to use CodeCommit to view the changes and approve the pull requests.
- D. Configure an IAM role for the Developers with access to CodeCommit and attach an access policy to the CodeCommit repository that denies the Developers role access when the reference is master. Allow Developers to use feature branches and create a pull request when a feature is complete. Allow an approver to use CodeCommit to view the changes and approve the pull requests.

Commented [LC152]: A (Correct) B - dynamically modify the IAM role C - allowing a one-time API call D - attach an access policy to the CodeCommit repository

Question #248

At all times, a business needs an RPO of two hours and an RTO of ten minutes for its data and application. A MySQL database and Amazon EC2 web servers are used by an application. The development team need a failover and disaster recovery plan.

Which deployment strategy combination will satisfy these requirements? (Select two.)

- A. Create an Amazon Aurora cluster in one Availability Zone across multiple Regions as the data store. Use Aurora's automatic recovery capabilities in the event of a disaster.
- B. Create an Amazon Aurora global database in two Regions as the data store. In the event of a failure, promote the secondary Region as the master for the application.
- C. Create an Amazon Aurora multi-master cluster across multiple Regions as the data store. Use a Network Load Balancer to balance the database traffic in different Regions.
- D. Set up the application in two Regions and use Amazon Route 53 failover-based routing that points to the Application Load Balancers in both Regions. Use health checks to determine the availability in a given Region. Use Auto Scaling groups in each Region to adjust capacity based on demand.
- E. Set up the application in two Regions and use a multi-Region Auto Scaling group behind Application Load Balancers to manage the capacity based on demand. In the event of a disaster, adjust the Auto Scaling group's desired instance count to increase baseline capacity in the failover Region.

Commented [LC153]:

Commented [LC154]:

Question #249 [SKIP]

Which of the following is not a suitable Ansible variable name?

- A. host1st_ref
- B. host-first-ref
- C. Host1stRef
- D. host_first_ref

Questions 250-299

Question #250

A corporation employs AWS Storage Gateway in file gateway mode in front of a multi-resource Amazon S3 bucket. When business resumes in the morning, consumers do not see the items handled the previous evening by a third party. When a DevOps engineer examines the S3 bucket directly, the data is there, but is absent from Storage Gateway.

Which method assures that all third-party files have been updated and are accessible in the morning?

- A. Configure a nightly Amazon EventBridge (Amazon CloudWatch Events) event to trigger an AWS Lambda function to run the RefreshCache command for Storage Gateway.
- B. Instruct the third party to put data into the S3 bucket using AWS Transfer for SFTP.
- C. Modify Storage Gateway to run in volume gateway mode.
- D. Use S3 same-Region replication to replicate any changes made directly in the S3 bucket to Storage Gateway.

Commented [LC155]: Ans: A
https://docs.aws.amazon.com/storagegateway/latest/APIReference/API_RefreshCache.html

Question #251

A DevOps engineer observes that all Amazon EC2 instances in an Auto Scaling group that are operating behind an Application Load Balancer are failing to respond to user requests. Additionally, the EC2 instances fail target group HTTP health checks.

The engineer discovers that the application process was not executing in any EC2 instances upon investigation. The system logs include a large number of out of memory messages. The engineer must strengthen the application's resilience in order to deal with a possible memory leak. Monitoring and notification should be enabled to notify you when a problem occurs.

Which combination of acts satisfies these criteria? (Select two.)

- A. Change the Auto Scaling configuration to replace the instances when they fail the load balancer's health checks.
- B. Change the target group health check HealthCheckIntervalSeconds parameter to reduce the interval between health checks.
- C. Change the target group health checks from HTTP to TCP to check if the port where the application is listening is reachable.
- D. Enable the available memory consumption metric within the Amazon CloudWatch dashboard for the entire Auto Scaling group. Create an alarm when the memory utilization is high. Associate an Amazon SNS topic to the alarm to receive notifications when the alarm goes off.
- E. Use the Amazon CloudWatch agent to collect the memory utilization of the EC2 instances in the Auto Scaling group. Create an alarm when the memory utilization is high and associate an Amazon SNS topic to receive a notification.

Commented [LC156]: I'll go with A, E B is wrong because it don't attack the problem C is wrong because changing the target group health checks from HTTP to TCP will not help D is wrong because of "notifications when the alarm goes off".

Commented [LC157]:

Question #252

To keep track of API calls made to our AWS account by various users and organizations, we may utilize _____ to produce a bulk history of calls for later review and _____ to respond in real time to AWS API requests.

- A. AWS Config; AWS Inspector
- B. AWS CloudTrail; AWS Config
- C. AWS CloudTrail; CloudWatch Events
- D. AWS Config; AWS Lambda

Commented [LC158]: CloudTrail is a batch API call collection service, CloudWatch Events enables real-time monitoring of calls through the Rules object interface.

Reference:
<https://aws.amazon.com/whitepapers/security-at-scale-governance-in-aws/>

Question #253

A DevOps Engineer builds an image-analysis application using Docker container technology. The program often experiences traffic surges. The Engineer must scale the application automatically in response to client demand while being cost efficient and minimizing any impact on availability.

What would enable the FASTEST reaction to traffic surges while still meeting all other requirements?

- A. Create an Amazon ECS cluster with the container instances in an Auto Scaling group. Configure the ECS service to use Service Auto Scaling. Set up Amazon CloudWatch alarms to scale the ECS service and cluster.
- B. Deploy containers on an AWS Elastic Beanstalk Multicontainer Docker environment. Configure Elastic Beanstalk to automatically scale the environment based on Amazon CloudWatch metrics.
- C. Create an Amazon ECS cluster using Spot Instances. Configure the ECS service to use Service Auto Scaling. Set up Amazon CloudWatch alarms to scale the ECS service and cluster.
- D. Deploy containers on Amazon EC2 instances. Deploy a container scheduler to schedule containers onto EC2 instances. Configure EC2 Auto Scaling for EC2 instances based on available Amazon CloudWatch metrics.

Commented [LC159]: C is not the answer, spot instances does not guarantee availability.
B is not correct because deploying into beanstalk is not fastest.
D is not correct, EC2 by itself can not run containers.

A is correct

Question #254

A web application providing healthcare services is hosted on Amazon EC2 instances and is protected by an ELB Application Load Balancer. The instances are distributed across several Availability Zones through an Amazon EC2 Auto Scaling group. A DevOps Engineer must provide a technique for removing an EC2 instance from production so that its system logs may be reviewed for faults in order to swiftly resolve web layer difficulties.

How is the Engineer going to execute this operation while maintaining availability and reducing downtime?

- A. Implement EC2 Auto Scaling groups cooldown periods. Use EC2 instance metadata to determine the instance state, and an AWS Lambda function to snapshot Amazon EBS volumes to preserve system logs.
- B. Implement Amazon CloudWatch Events rules. Create an AWS Lambda function that can react to an instance termination to deploy the CloudWatch Logs agent to upload the system and access logs to Amazon S3 for analysis.
- C. Terminate the EC2 instances manually. The Auto Scaling service will upload all log information to CloudWatch Logs for analysis prior to instance termination.
- **D. Implement EC2 Auto Scaling groups with lifecycle hooks. Create an AWS Lambda function that can modify an EC2 instance lifecycle hook into a standby state, extract logs from the instance through a remote script execution, and place them in an Amazon S3 bucket for analysis.**

Commented [LC160]: Answer: D (reluctantly)

A - incorrect - Cooldown affects availability. Note: spot instances metadata indicates termination - a job could watch for this. Easier to leave EBS volume behind, or use CW Events for snapshot

B - incorrect - good luck deploying a CW Agent to a terminating instance. Use hooks.

C - almost correct - if CW Agent was mentioned, and if system logs was mentioned, and if it didn't say Auto Scaling service would upload logs. bad wording or completely wrong.

D - almost correct - if standby state wasn't mentioned. you can definitely modify a lifecycle hook (put-lifecycle-hook). Lifecycle hooks trigger on instance launch or terminate: terminate here. You can't change a terminating instance to a standby state, however, it is in a terminating:wait" state ~ kind of standby. baddily wordid. Lifecycle hooks are made for log extraction. AWS state "While the instance is in the wait state ... connect to the instance and download logs ... before the instance is fully terminated."

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html#lifecycle-hooks-overview>

Thus D over C, or even A

Commented [LC161]: Answer: D

<https://docs.aws.amazon.com/codedeploy/latest/userguide/integrations-aws-auto-scaling.html>

Scale-up Events During a Deployment If an Amazon EC2 Auto Scaling scale-up event occurs while a deployment is underway, the new instances will be updated with the application revision that was most recently deployed, not the application revision that is currently being deployed. If the deployment succeeds, the old instances and the newly scaled-up instances will be hosting different application revisions.

Commented [LC162]: B

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Monitoring.OS.html

Question #255

A DevOps Engineer is deploying code across a fleet of Amazon EC2 instances in an EC2 Auto Scaling group using AWS CodeDeploy. CodeDeployDefault is set to do in-place deployments using the accompanying CodeDeploy deployment group, which is connected with EC2 Auto Scaling. OneAtATime. The Engineer notices during an ongoing fresh deployment that, although the overall deployment was successful, two out of five instances still had the prior application version installed. The remaining three instances contain the most recent application modification.

What is most likely to be the source of this problem?

- A. The two affected instances failed to fetch the new deployment.
- B. A failed AfterInstall lifecycle event hook caused the CodeDeploy agent to roll back to the previous version on the affected instances.
- C. The CodeDeploy agent was not installed in two affected instances.
- **D. EC2 Auto Scaling launched two new instances while the new deployment had not yet finished, causing the previous version to be deployed on the affected instances.**

Question #256

A DevOps Engineer needs set up monitoring for an Amazon EC2 and Amazon RDS MySQL workload. Monitoring must contain the following: application logs and metrics for the Amazon EC2 instances' operating systems. Logs and analytics for the Amazon RDS database's operating system

Which procedures should the Engineer follow?

- A. Install an Amazon CloudWatch agent on the EC2 and RDS instances. Configure the agent to send the operating system metrics and application and database logs to CloudWatch.
- **B. Install an Amazon CloudWatch agent on the EC2 instance, and configure the agent to send the application logs and operating system metrics to CloudWatch. Enable RDS Enhanced Monitoring, and modify the RDS instance to publish database logs to CloudWatch Logs.**
- C. Install an Amazon CloudWatch Logs agent on the EC2 instance and configure it to send application logs to CloudWatch.
- D. Set up scheduled tasks on the EC2 and RDS instances to put operating system metrics and application and database logs into an Amazon S3 bucket. Set up an event on the bucket to invoke an AWS Lambda function to monitor for errors each time an object is put into the bucket.

Question #257 [SKIP]

You wish to safely distribute credentials to your fleet of web server instances for your Amazon RDS instance. The credentials are saved in a configuration management system-controlled file.

How do you safely distribute credentials across a fleet of hundreds of web server instances in an automated way while keeping the option to roll back if necessary?

- A. Store your credential files in an Amazon S3 bucket. Use Amazon S3 server-side encryption on the credential files. Have a scheduled job that pulls down the credential files into the instances every 10 minutes.
- B. Store the credential files in your version-controlled repository with the rest of your code. Have a post-commit action in version control that kicks off a job in your continuous integration system which securely copies the new credential files to all web server instances.
- C. Insert credential files into user data and use an instance lifecycle policy to periodically refresh the file from the user data.
- D. Keep credential files as a binary blob in an Amazon RDS MySQL DB instance, and have a script on each Amazon EC2 instance that pulls the files down from the RDS instance.
- E. Store the credential files in your version-controlled repository with the rest of your code. Use a parallel file copy program to send the credential files from your local machine to the Amazon EC2 instances.

Question #258

A business administers an application that logs to Amazon CloudWatch Logs. The business wants to store the logs on Amazon S3. After 90 days, logs are seldom viewed and must be preserved for a period of ten years.

Which actions should a DevOps engineer perform in combination to achieve these requirements? (Select two.)

- A. Configure a CloudWatch Logs subscription filter to use AWS Glue to transfer all logs to an S3 bucket.
- B. Configure a CloudWatch Logs subscription filter to use Amazon Kinesis Data Firehose to stream all logs to an S3 bucket.
- C. Configure a Cloud Watch Logs subscription filter to stream all logs to an S3 bucket.
- D. Configure the S3 bucket lifecycle policy to transition logs to S3 Glacier after 90 days and to expire logs after 3,650 days.
- E. Configure the S3 bucket lifecycle policy to transition logs to Reduced Redundancy after 90 days and to expire logs after 3,650 days.

Commented [LC163]: B to get logs to S3

D because you want to achieve the logs at a low cost storage

Commented [LC164]:

Question #259

A new zero-day vulnerability in OpenSSL has been discovered, necessitating the quick patching of an Amazon Linux-based production web fleet. Currently, OS upgrades are conducted manually on a monthly basis and delivered through changes to the launch configuration of the production Auto Scaling Group.

Which technique should a DevOps Engineer use to update packages in-place without causing a service outage?

- A. Use AWS CodePipeline and AWS CodeBuild to generate new copies of these packages, and update the Auto Scaling group's launch configuration.
- B. Use AWS Inspector to run 'yum upgrade' on all running production instances, and manually update the AMI for the next maintenance window.
- C. Use Amazon EC2 Run Command to issue a package update command to all running production instances, and update the AMI for future deployments.
- D. Define a new AWS OpsWorks layer to match the running production instances, and use a recipe to issue a package update command to all running production instances.

Commented [LC165]: I think the correct answer is C

Ref:
<https://aws.amazon.com/blogs/aws/ec2-run-command-is-now-a-cloudwatch-events-target/>

"EC2 Run Command is part of EC2 Systems Manager. It allows you to operate on collections of EC2 instances and on-premises servers reliably and at scale, in a controlled and selective fashion. You can run scripts, install software, collect metrics and log files, manage patches, and much more, on both Windows and Linux."

Question #260

A web application is hosted on Amazon EC2 instances behind a load balancer (ALB). A DevOps Engineer is deploying a new version using AWS CodeDeploy. The deployment fails at the AllowTraffic lifecycle event; however, the deployment logs do not identify the reason for the failure.

What may account for this?

- A. The appspec.yml file contains an invalid script to execute in the AllowTraffic lifecycle hook.
- B. The user who initiated the deployment does not have the necessary permissions to interact with the ALB.
- C. The health checks specified for the ALB target group are misconfigured.
- D. The CodeDeploy agent was not installed in the EC2 instances that are part of the ALB target group.

Commented [LC166]: A is incorrect because "The Start, Install, TestTraffic, AllowTraffic, and End events in the deployment cannot be scripted, which is why they appear in gray in this diagram."

<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html>

B, D is incorrect because if the permission or CodeDeploy agent is not installed, the deploy will not go to AllowTraffic hook, it will be fail prior to AllowTraffic.

Question #261 [SKIP]

When creating a Docker image, you are looking through the logs of a persistent data volume for parameters to use in the next build. The following command is run.

Which of the following procedures will result in the Docker RUN command failing? `RUN grep service status | grep ERROR cat:/data/log/*.error`

- A. the first `grep` command
- B. any one of them
- C. the second `grep` command
- D. the `cat` command

Question #262

Following a disaster recovery exercise, an Enterprise Architect learns that it will take more than seven hours of manual labor for a big team of Database and Storage Administrators to make a flagship application's database functioning in a new AWS Region. Additionally, the Architect observes that the restored database often lacks up to two hours of data transactions.

Which approach improves the RTO and RPO in a failover scenario involving many regions?

- A. Deploy an Amazon RDS Multi-AZ instance backed by a multi-region Amazon EFS. Configure the RDS option group to enable multi-region availability for native automation of cross-region recovery and continuous data replication. Create an Amazon SNS topic subscribed to RDS-impacted events to send emails to the Database Administration team when significant query Latency is detected in a single Availability Zone.
- B. Use Amazon SNS topics to receive published messages from Amazon RDS availability and backup events. Use AWS Lambda for three separate functions with calls to Amazon RDS to snapshot a database instance, create a cross-region snapshot copy, and restore an instance from a snapshot. Use a scheduled Amazon CloudWatch Events rule at a frequency matching the RPO to trigger the Lambda function to snapshot a database instance. Trigger the Lambda function to create a cross-region snapshot copy when the SNS topic for backup events receives a new message. Configure the Lambda function to restore an instance from a snapshot to trigger sending new messages published to the availability SNS topic.
- C. Create a scheduled Amazon CloudWatch Events rule to make a call to Amazon RDS to create a snapshot from a database instance and specify a frequency to match the RPO. Create an AWS Step Functions task to call Amazon RDS to perform a cross-region snapshot copy into the failover region, and configure the state machine to execute the task when the RDS snapshot create state is complete. Create an SNS topic subscribed to RDS availability events, and push these messages to an Amazon SQS queue located in the failover region. Configure an Auto Scaling group of worker nodes to poll the queue for new messages and make a call to Amazon RDS to restore a database from a snapshot after a checksum on the cross-region copied snapshot returns valid.
- D. Use Amazon RDS scheduled instance lifecycle events to create a snapshot and specify a frequency to match the RPO. Use Amazon RDS scheduled instance lifecycle event configuration to perform a cross-region snapshot copy into the failover region upon SnapshotCreateComplete events. Configure Amazon CloudWatch to alert when the CloudWatch RDS namespace CPUUtilization metric for the database instance falls to 0% and make a call to Amazon RDS to restore the database snapshot in the failover region.

Commented [LC167]:

Question #263

You're implementing a layer in a software stack on AWS that requires rapid scaling out in order to respond to rising demand. You're executing the code on Amazon EC2 instances in an Auto Scaling Group behind an Elastic Load Balancing (ELB).

Which approach for deploying application code should you use?

- A. SSH into new instances that come online, and deploy new code onto the system by pulling it from an S3 bucket, which is populated by code that you refresh from source control on new pushes.
- **B. Bake an AMI when deploying new versions of code, and use that AMI for the Auto Scaling Launch Configuration.**
- C. Create a Dockerfile when preparing to deploy a new version to production and publish it to S3. Use UserData in the Auto Scaling Launch configuration to pull down the Dockerfile from S3 and run it when new instances launch.
- D. Create a new Auto Scaling Launch Configuration with UserData scripts configured to pull the latest code at all times.

Commented [LC168]: The bootstrapping process can be slower if you have a complex application or multiple applications to install. Managing a fleet of applications with several build tools and dependencies can be a challenging task during rollouts. Furthermore, your deployment service should be designed to do faster rollouts to take advantage of Auto Scaling.

Reference:
<https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf>

Question #264

Following a recent audit, a business determined that it needed to establish a new disaster recovery plan for its Amazon S3 data and MySQL database operating on Amazon EC2.

Management want the ability to recover to a backup AWS Region with a recovery time objective of less than 5 seconds and a recovery time objective of less than 1 minute.

Which activities will satisfy the criteria while keeping operating costs to a minimum? (Select two.)

- A. Modify the application to write to both Regions at the same time when uploading objects to Amazon S3.
- B. Migrate the database to an Amazon Aurora multi-master in the primary and secondary Regions.
- C. Migrate the database to Amazon RDS with a read replica in the secondary Region.
- **D. Migrate to Amazon Aurora Global Database.**
- **E. Set up S3 cross-Region replication with a replication SLA for the S3 buckets where objects are being put.**

Commented [LC169]:

Commented [LC170]:

Question #265

Several of your Amazon Elastic Compute Cloud instances are set to utilize a proxy.

Is it possible to utilize Amazon Inspector to do routine assessments of instances behind a proxy?

- A. Only Windows-based systems are supported as Linux-based systems use custom configurations that are not supported by AWS Agent in the current release.
- B. Only Linux-based systems are supported, and AWS agent supports HTTPS proxy on these systems.
- C. No, AWS Agent does NOT support proxy environments.
- **D. Yes, AWS Agent supports proxy environments on both Linux-based and Windows-based systems.**

Commented [LC171]: The AWS agent supports proxy environments. For Linux instances, Inspector supports HTTPS Proxy, and for Windows instances, it supports WinHTTP proxy.

Reference:
https://docs.aws.amazon.com/inspector/latest/userguide/inspector_agents.html

Question #266

Your application processes background tasks using Amazon SQS and Auto Scaling. The Auto Scaling policy is determined by the quantity of messages in the queue, up to a maximum of 100 Instances. Since the application's inception, the group has never exceeded 50 members. The Auto Scaling group has now grown to 100 members, the backlog is growing, and very few jobs are getting completed. The queue is receiving a reasonable volume of messages.

What should you do to determine why the line is exceptionally long and how to shorten it?

- A. Temporarily increase the Auto Scaling group's desired value to 200. When the queue size has been reduced, reduce it to 50.
- **B. Analyze the application logs to identify possible reasons for message processing failure and resolve the cause for failures.**
- C. Create additional Auto Scaling groups, enabling the processing of the queue to be performed in parallel.
- D. Analyze CloudTrail logs for Amazon SQS to ensure that the instances' Amazon EC2 role has permission to receive messages from the queue.

Commented [LC172]: Ans is B
A and C are don't make sense and D is wrong because if there is a permission issue no job should be completed but in question, it is mentioned: "very few Jobs are being completed."

Question #267

Your application is composed of 10% writes and 90% reads. At the moment, all requests are routed via a Route53 Alias Record to an AWS ELB that sits in front of an EC2 Auto Scaling Group. Your system becomes prohibitively costly during huge traffic spikes associated with specific news events, during which many more users seek to consume comparable data concurrently.

What is the easiest and most cost-effective strategy to minimize expenses and scale in the face of such spikes?

- A. Create an S3 bucket and asynchronously replicate common requests responses into S3 objects. When a request comes in for a precomputed response, redirect to AWS S3.
- B. Create another ELB and Auto Scaling Group layer mounted on top of the other system, adding a tier to the system. Serve most read requests out of the top layer.
- C. Create a CloudFront Distribution and direct Route53 to the Distribution. Use the ELB as an Origin and specify Cache Behaviours to proxy cache requests which can be served late.
- D. Create a Memcached cluster in AWS ElastiCache. Create cache logic to serve requests which can be served late from the in-memory cache for increased performance.

Commented [LC173]: CloudFront is ideal for scenarios in which entire requests can be served out of a cache and usage patterns involve heavy reads and spikiness in demand. You can configure multiple cache behaviors for your web distribution. Amazon CloudFront will match incoming viewer requests with your list of URL patterns, and if there is a match, the service will honor the cache behavior you configure for that URL pattern. Each cache behavior can include the following Amazon CloudFront configuration values: origin server name, viewer connection protocol, minimum expiration period, query string parameters, cookies, and trusted signers for private content.

Reference:

<https://aws.amazon.com/cloudfront/dynamic-content/>

Question #268

You must be able to install an AWS stack consistently across numerous environments. You've chosen CloudFormation as the appropriate technology to achieve this, but you've discovered that there is a resource type you need to generate and model that CloudFormation does not support.

How should you approach this obstacle?

- A. Use a CloudFormation Custom Resource Template by selecting an API call to proxy for create, update, and delete actions. CloudFormation will use the AWS SDK, CLI, or API method of your choosing as the state transition function for the resource type you are modeling.
- B. Submit a ticket to the AWS Forums. AWS extends CloudFormation Resource Types by releasing tooling to the AWS Labs organization on GitHub. Their response time is usually 1 day, and they complete requests within a week or two.
- C. Instead of depending on CloudFormation, use Chef, Puppet, or Ansible to author Heat templates, which are declarative stack resource definitions that operate over the OpenStack hypervisor and cloud environment.
- D. Create a CloudFormation Custom Resource Type by implementing create, update, and delete functionality, either by subscribing a Custom Resource Provider to an SNS topic, or by implementing the logic in AWS Lambda.

Commented [LC174]: Custom resources provide a way for you to write custom provisioning logic in AWS CloudFormation template and have AWS CloudFormation run it during a stack operation, such as when you create, update or delete a stack. For more information, see Custom Resources.

Reference:

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-customresources.html>

Question #269

A business is deploying web apps using an AWS CloudFormation template. Manual adjustments are required for each of the template's three key environments: production, staging, and development. This sprint will construct and configure AWS CodePipeline for automated deployments.

What modifications should the DevOps Engineer apply to the CloudFormation template to guarantee that it is reusable across many pipelines?

- A. Use a CloudFormation custom resource to query the status of the CodePipeline to determine which environment is launched. Dynamically alter the launch configuration of the Amazon EC2 instances.
- B. Set up a CodePipeline pipeline for each environment to use input parameters. Use CloudFormation mappings to switch associated UserData for the Amazon EC2 instances to match the environment being launched.
- C. Set up a CodePipeline pipeline that has multiple stages, one for each development environment. Use AWS Lambda functions to trigger CloudFormation deployments to dynamically alter the UserData of the Amazon EC2 instances launched in each environment.
- D. Use CloudFormation input parameters to dynamically alter the LaunchConfiguration and UserData sections of each Amazon EC2 instance every time the CloudFormation stack is updated.

Commented [LC175]: B sounds correct: To make templates reusable, use the parameters, mappings, and conditions sections so that you can customize your stacks when you create them.

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/best-practices.html#reuse>

Question #270

You're doing a load test on your AWS-hosted application. While doing tests on your Amazon RDS MySQL DB instance, you observe something. When your program reaches 100% CPU consumption, it becomes unresponsive. Your application is heavily reliant on reading.

How do you grow your data layer to suit the application's requirements? (Select three.)

- A. Add Amazon RDS DB read replicas, and have your application direct read queries to them.
- B. Add your Amazon RDS DB instance to an Auto Scaling group and configure your CloudWatch metric based on CPU utilization.
- C. Use an Amazon SQS queue to throttle data going to the Amazon RDS DB instance.
- D. Use ElastiCache in front of your Amazon RDS DB to cache common queries.
- E. Shard your data set among multiple Amazon RDS DB instances.
- F. Enable Multi-AZ for your Amazon RDS DB instance.

Commented [LC176]:

Commented [LC177]:

Commented [LC178]: <https://aws.amazon.com/blogs/database/sharding-with-amazon-relational-database-service/>

Question #271

A business employs a single developer who is responsible for creating code for an automated deployment process. Each project's source code is stored in an Amazon S3 bucket. The firm wants to expand its workforce of developers but is worried about code clashes and missed productivity. Additionally, the organization wants to establish a test environment for the purpose of testing newer versions of code and to enable Developers to automatically publish to both environments when code is modified in the repository.

Which method is the MOST EFFECTIVE in meeting these requirements?

- A. Create an AWS CodeCommit repository for each project, use the master branch for production code, and create a testing branch for code deployed to testing. Use feature branches to develop new features and pull requests to merge code to testing and master branches.
- B. Create another S3 bucket for each project for testing code, and use an AWS Lambda function to promote code changes between testing and production buckets. Enable versioning on all buckets to prevent code conflicts.
- C. Create an AWS CodeCommit repository for each project, and use the master branch for production and test code with different deployment pipelines for each environment. Use feature branches to develop new features.
- D. Enable versioning and branching on each S3 bucket, use the master branch for production code, and create a testing branch for code deployed to testing. Have Developers use each branch for developing in each environment.

Commented [LC179]:

Question #272

A security team mandates that all Amazon EBS volumes associated to an Amazon EC2 instance be encrypted using AWS Key Management Service (AWS KMS). If encryption is not enabled, the company's policy involves detaching and deleting the EBS volume. Unencrypted EBS volumes must be detected and deleted automatically by a DevOps Engineer.

Which strategy should the Engineer use in order to achieve this with the LEAST amount of operational effort?

- A. Create an Amazon CloudWatch Events rule that invokes an AWS Lambda function when an EBS volume is created. The Lambda function checks the EBS volume for encryption. If encryption is not enabled and the volume is attached to an instance, the function deletes the volume.
- B. Create an AWS Lambda function to describe all EBS volumes in the region and identify volumes that are attached to an EC2 instance without encryption enabled. The function then deletes all non-compliant volumes. The AWS Lambda function is invoked every 5 minutes by an Amazon CloudWatch Events scheduled rule.
- C. Create a rule in AWS Config to check for unencrypted and attached EBS volumes. Subscribe an AWS Lambda function to the Amazon SNS topic that AWS Config sends change notifications to. The Lambda function checks the change notification and deletes any EBS volumes that are non-compliant.
- D. Launch an EC2 instance with an IAM role that has permissions to describe and delete volumes. Run a script on the EC2 instance every 5 minutes to check all EBS volumes in all regions and identify volumes that are attached without encryption enabled. The script then deletes those volumes.

Commented [LC180]: Answer: C Check encryption with encrypted-volumes. Read the Lambda function of the rule as a trigger for the Config rule.

https://docs.aws.amazon.com/ja_jp/config/latest/developer/guide/encrypted-volumes.html

https://docs.aws.amazon.com/ja_jp/config/latest/developer/guide/evaluate-config_develop_rules_examples.html

Question #273 [SKIP]

Which flag would you use to restrict the memory use of a Docker container to 128 megabytes?

- A. -memory 128m
- B. -m 128m
- C. --memory-reservation 128m
- D. -m 128MB

Question #274

A business makes use of a complicated system comprised of networking, identity and access management rules, and many three-tier apps. Because the requirements for a new system are still being established, the number of AWS components included in the final architecture is unknown. The DevOps Engineer should begin by creating AWS resources using AWS CloudFormation in order to automate and version-control the new architecture.

What is the best approach for creating new environments using CloudFormation?

- A. Manually construct the networking layer using Amazon VPC and then define all other resources using CloudFormation.
- B. Create a single template to encompass all resources that are required for the system so there is only one template to version-control.
- C. Create multiple separate templates for each logical part of the system, use cross-stack references in CloudFormation, and maintain several templates in version control.
- D. Create many separate templates for each logical part of the system, and provide the outputs from one to the next using an Amazon EC2 instance running SDK for granular control.

Commented [LC181]: A - manual. aws will never recommend
B - single template , not a best practice and will become unmangeable
C - correct.
D - unnecessary ec2 instance

Question #275 [SKIP]

Management has reported an increase in their monthly Amazon web services bill, and they are highly worried about the cost rise. Management has requested that you ascertain the precise source of this rise. You notice a rise in the cost of data transmission after examining the billing report.

How can you provide management a clearer picture of data transfer usage?

- A. Update your Amazon CloudWatch metrics to use five-second granularity, which will give better detailed metrics that can be combined with your billing data to pinpoint anomalies.
- B. Use Amazon CloudWatch Logs to run a map-reduce on your logs to determine high usage and data transfer.
- C. Deliver custom metrics to Amazon CloudWatch per application that breaks down application data transfer into multiple, more specific data points.
- D. Using Amazon CloudWatch metrics, pull your Elastic Load Balancing outbound data transfer metrics monthly, and include them with your billing report to show which application is causing higher bandwidth usage.

Question #276

Your team is tasked with the responsibility of maintaining an AWS Elastic Beanstalk application. The company needs that you transition to a continuous deployment strategy, which allows for numerous application changes per day with no downtime.

What should you do to allow this while yet being able to roll back very instantly in an emergency?

- A. Enable rolling updates in the Elastic Beanstalk environment and set an appropriate pause time for application startup.
- B. Create a second Elastic Beanstalk environment that runs the new application version, and swap the environment CNAMEs.
- C. Configure the application to poll for a new application version in your code repository; download and install the new version to each running Elastic Beanstalk instance.
- D. Create a second Elastic Beanstalk environment with the new application version, and configure the old environment to use the HTTP 301 response code to redirect clients to the new environment.

Commented [LC182]:

Question #277

A business operates an application on Amazon EC2 and on-premises. A DevOps Engineer must ensure that patching is consistent across both environments. Patching is prohibited per company policy during non-business hours.

Which combination of acts satisfies these criteria? (Select three.)

- A. Add the physical machines into AWS Systems Manager using Systems Manager Hybrid Activations.
- B. Attach an IAM role to the EC2 instances, allowing them to be managed by AWS Systems Manager.
- C. Create IAM access keys for the on-premises machines to interact with AWS Systems Manager.
- D. Execute an AWS Systems Manager Automation document to patch the systems every hour.
- E. Use Amazon CloudWatch Events scheduled events to schedule a patch window.
- F. Use AWS Systems Manager Maintenance Windows to schedule a patch window.

Commented [LC183]:

Commented [LC184]:

Commented [LC185]:

Question #278

A DevOps team is responsible for the management of an on-premises API that acts as the backend for an Amazon API Gateway endpoint. Customers have complained about long response times, which the development team confirmed using Amazon CloudWatch's API Gateway latency data. To determine the root cause, the team must gather pertinent data without adding further delay.

What steps should be done to achieve this? (Select two.)

- A. Install the CloudWatch agent server side and configure the agent to upload relevant logs to CloudWatch.
- B. Enable AWS X-Ray tracing in API Gateway, modify the application to capture request segments, and upload those segments to X-Ray during each request.
- C. Enable AWS X-Ray tracing in API Gateway, modify the application to capture request segments, and use the X-Ray daemon to upload segments to X-Ray.
- D. Modify the on-premises application to send log information back to API Gateway with each request.
- E. Modify the on-premises application to calculate and upload statistical data relevant to the API service requests to CloudWatch metrics.

Commented [LC186]: I'll go with A and C

References:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-on-premise.html>

<https://docs.aws.amazon.com/xray/latest/devguide/xray-api-sendingdata.html>

Commented [LC187]:

Question #279

Your CTO believes your AWS account has been compromised.

What is the only method to determine for certain if illegal access occurred and what was done, presuming your hackers are highly skilled AWS engineers who are doing all possible to conceal their tracks?

- A. Use CloudTrail Log File Integrity Validation.
- B. Use AWS Config SNS Subscriptions and process events in real time.
- C. Use CloudTrail backed up to AWS S3 and Glacier.
- D. Use AWS Config Timeline forensics.

Commented [LC188]: You must use CloudTrail Log File Validation (default or custom implementation), as any other tracking method is subject to forgery in the event of a full account compromise by sophisticated enough hackers. Validated log files are invaluable in security and forensic investigations. For example, a validated log file enables you to assert positively that the log file itself has not changed, or that particular user credentials performed specific API activity. The CloudTrail log file integrity validation process also lets you know if a log file has been deleted or changed, or assert positively that no log files were delivered to your account during a given period of time.

Reference:

<http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-intro.html>

Commented [LC189]: "Which service has the quickest retrieval times?"

Everything that came before this is just fluff and has nothing to do with the actual question.

Answer is indeed D.

Question #280

The development team is working on a social media game in which users are ranked according to their performance on a scoreboard. While the present approach stores user data in an Amazon RDS for MySQL database, the game cannot show scores rapidly enough during performance testing.

Which service has the quickest retrieval times?

- A. Migrate user data to Amazon DynamoDB for managing content.
- B. Use AWS Batch to compute and deliver user and score content.
- C. Deploy Amazon CloudFront for user and score content delivery.
- D. Set up Amazon ElastiCache to deliver user and score content.

Question #281

A business has defined labeling and configuration guidelines for its AWS-hosted infrastructure resources. A DevOps Engineer is designing a dashboard that will enable near-real-time visibility into the compliance posture and the ability to identify infractions.

Which strategy satisfies the specified criteria?

- A. Define the resource configurations in AWS Service Catalog, and monitor the AWS Service Catalog compliance and violations in Amazon CloudWatch. Then, set up and share a live CloudWatch dashboard. Set up Amazon SNS notifications for violations and corrections.
- **B. Use AWS Config to record configuration changes and output the data to an Amazon S3 bucket. Create an Amazon QuickSight analysis of the dataset, and use the information on dashboards and mobile devices.**
- C. Create a resource group that displays resources with the specified tags and those without tags. Use the AWS Management Console to view compliant and non-compliant resources.
- D. Define the compliance and tagging requirements in Amazon Inspector. Output the results to Amazon CloudWatch Logs. Build a metric filter to isolate the monitored elements of interest and present the data in a CloudWatch dashboard.

Commented [LC190]: Right: B - AWS Config exports data to S3 and Quicksight can read this data nearly real-time to provide the needed information.

Wrong:

A - Service Catalog is about standardized CFM stacks and AMIs for example, not about Tag compliance.

C - Using Resource Groups and the Web Console will not give the required outcome

D - Amazon inspector is a security scanner for apps running on AWS.

Question #282

A firm is utilizing AWS CodeBuild to deploy a container-based application. Prior to deployment, the Security team requires that all containers be checked for vulnerabilities using a password-protected URL. All sensitive data must be securely kept.

Which option is most appropriate for meeting these requirements?

- A. Encrypt the password using AWS KMS. Store the encrypted password in the buildspec.yml file as an environment variable under the variables mapping. Reference the environment variable to initiate scanning.
- B. Import the password into an AWS CloudHSM key. Reference the CloudHSM key in the buildspec.yml file as an environment variable under the variables mapping. Reference the environment variable to initiate scanning.
- **C. Store the password in the AWS Systems Manager Parameter Store as a secure string. Add the Parameter Store key to the buildspec.yml file as an environment variable under the parameter-store mapping. Reference the environment variable to initiate scanning.**
- D. Use the AWS Encryption SDK to encrypt the password and embed in the buildspec.yml file as a variable under the secrets mapping. Attach a policy to CodeBuild to enable access to the required decryption key.

Commented [LC191]: https://docs.aws.amazon.com/pt_br/codebuild/latest/userguide/build-spec-ref.html

Question #283

A developer is responsible for the administration of a fleet of 50 Amazon EC2 Linux machines. The servers are managed by an Amazon EC2 Auto Scaling group and load balanced using Elastic Load Balancing.

At times, application servers are terminated as a result of failed ELB HTTP health checks. The developer wishes to do a root cause analysis on the problem, but the server is stopped before he can access application logs.

How is it possible to automate log collection?

- A. Use Auto Scaling lifecycle hooks to put instances in a Pending:Wait state. Create an Amazon CloudWatch Alarm for EC2 Instance Terminate Successful and trigger an AWS Lambda function that executes an SSM Run Command script to collect logs, push them to Amazon S3, and complete the lifecycle action once logs are collected.
- B. Use Auto Scaling lifecycle hooks to put instances in a Terminating:Wait state. Create a Config rule for EC2 Instance-terminate Lifecycle Action and trigger a step function that executes a script to collect logs, push them to Amazon S3, and complete the lifecycle action once logs are collected.
- C. Use Auto Scaling lifecycle hooks to put instances in a Terminating:Wait state. Create an Amazon CloudWatch subscription filter for EC2 Instance Terminate Successful and trigger a CloudWatch agent that executes a script to collect logs, push them to Amazon S3, and complete the lifecycle action once logs are collected.
- **D. Use Auto Scaling lifecycle hooks to put instances in a Terminating:Wait state. Create an Amazon CloudWatch Events rule for EC2 Instance-terminate Lifecycle Action and trigger an AWS Lambda function that executes a SSM Run Command script to collect logs, push them to Amazon S3, and complete the lifecycle action once logs are collected.**

Commented [LC192]: Correct Answer: D

Reference:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html>

Question #284

You must do ad-hoc analysis of log data, which includes fast looking for particular error codes and reference numbers.

Which of the following should you assess first?

- A. AWS Elasticsearch Service
- B. AWS RedShift
- C. AWS EMR
- D. AWS DynamoDB

Commented [LC193]: Amazon Elasticsearch Service (Amazon ES) is a managed service that makes it easy to deploy, operate, and scale Elasticsearch clusters in the AWS cloud.

Elasticsearch is a popular opensource search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics.

Reference:

<http://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/what-is-amazonelasticsearch-service.html>

Question #285

The management team of a firm with a sizable on-premises OpenStack infrastructure want to migrate non-production workloads to Amazon Web Services (AWS). To link the environments, an AWS Direct Connect connection has been established and setup. Due to contractual constraints, production workloads must stay on-premises until the next contract negotiation. They will then be migrated to AWS. The organization adheres to the Center for Internet Security's (CIS) hardening requirements; this configuration was created utilizing the company's configuration management system.

Which method will build an identical image in the AWS environment automatically and with little overhead?

- A. Write an AWS CloudFormation template that will create an Amazon EC2 instance. Use cloud-init to install the configuration management agent, use cfn-wait to wait for configuration management to successfully apply, and use an AWS Lambda-backed custom resource to create the AMI.
- B. Log in to the console, launch an Amazon EC2 instance, and install the configuration management agent. When changes are applied through the configuration management system, log in to the console and create a new AMI from the instance.
- C. Create a new AWS OpsWorks layer and mirror the image hardening standards. Use this layer as the baseline for all AWS workloads.
- D. When a change is made in the configuration management system, a job in Jenkins is triggered to use the VM Import command to create an Amazon EC2 instance in the Amazon VPC. Use lifecycle hooks to launch an AWS Lambda function to create the AMI.

Commented [LC194]: Breakdown.

A. Nope. No cloud-unit.
B. Possible, not enough information is given about the agent, less preferred option.
C. Will not create an 'identical image'. Nope.

D. The word placement of the question has been intentionally moved around to misdirect. If you re-organise the question to follow the real world process = a job in Jenkins is triggered > (CLI) Import command to create 'image' > use lifecycle hooks to launch Lambda function to create AMI > create Amazon EC2 instance in Amazon VPC. This is text book definition of the process. Not all possible answers are written as you would expect.

Answer is D. Meets identical requirements and does work with minimal overhead.

Question #286 [SKIP]

You are responsible for the design of a corporate data storage system. Due to the fact that your data management software needs mountable drives and a genuine filesystem, you cannot utilize S3 for storage. You need permanence, which is why your system will be hosted on AWS EBS Volumes. The system requires as little storage as feasible, and access is neither frequent or high-throughput, with the majority of reads being sequential.

Which EBS Volume Type is the best suited for this scenario?

- A. gp1
- B. io1
- C. standard
- D. gp2

Commented [LC195]: Old question/answers.

You would now use a SC1 (Cold HDD) type.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#hard-disk-drives>

Question #287

A business operates a number of internet-facing APIs that are secured using an AWS Lambda authorizer. A security team would want to be notified when a high number of queries fail permission, since this might suggest API misuse. Given the volume of API queries, the team want to be notified only if the percentage of HTTP 403 Forbidden answers exceeds 2% of all API calls.

Which approach is most likely to do this?

- A. Use the default Amazon API Gateway 403Error and Count metrics sent to Amazon CloudWatch, and use metric math to create a CloudWatch alarm. Use the $(403Error/Count)*100$ mathematical expression when defining the alarm. Set the alarm threshold to be greater than 2.
- B. Write a Lambda function that fetches the default Amazon API Gateway 403Error and Count metrics sent to Amazon CloudWatch, calculate the percentage of errors, then push a custom metric to CloudWatch named Custom403Percent. Create a CloudWatch alarm based on this custom metric. Set the alarm threshold to be greater than 2.
- C. Configure Amazon API Gateway to send custom access logs to Amazon CloudWatch Logs. Create a log filter to produce a custom metric for the HTTP 403 response code named Custom403Error. Use this custom metric and the default API Gateway Count metric sent to CloudWatch, and use metric math to create a CloudWatch alarm. Use the $(Custom403Error/Count)*100$ mathematical expression when defining the alarm. Set the alarm threshold to be greater than 2.
- D. Configure Amazon API Gateway to enable custom Amazon CloudWatch metrics, enable the ALL_STATUS_CODE option, and define an APICustom prefix. Use CloudWatch metric math to create a CloudWatch alarm. Use the $(APICustom403Error/Count)*100$ mathematical expression when defining the alarm. Set the alarm threshold to be greater than 2.

Commented [LC196]:

Question #288

A DevOps engineer has scheduled the deletion of old AWS KMS keys and established a remediation AWS Lambda code to re-enable a key if required. The engineer want to automate this process using accessible AWS CloudTrail data in order to ensure that a key slated for deletion is re-enabled if it is in use.

Which automation solution supports this?

- A. Create an Amazon CloudWatch Logs metric filter and alarm for KMS events with an error message. Set the remediation Lambda function as the target of the alarm.
- B. Create an Amazon CloudWatch Logs metric filter and alarm for KMS events with an error message. Create an Amazon SNS topic as the target of the alarm. Subscribe the remediation Lambda function to the SNS topic.
- C. Create an Amazon CloudWatch Events rule pattern looking for KMS service events with an error message. Create an Amazon SNS topic as the target of the rule. Subscribe the remediation Lambda function to the SNS topic.
- D. Use Amazon CloudTrail to alert for KMS service events with an error message. Set the remediation Lambda function as the target of the rule.

Commented [LC197]:

Question #289 [SKIP]

You host accounting software on AWS. This program must be available 24 hours a day, 7 days a week, and has a relatively static need for compute resources. Additionally, you have unrelated batch tasks that must run once daily at a time of your choice.

How can you cut costs?

- A. Purchase a Heavy Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.
- B. Purchase a Medium Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.
- C. Purchase a Light Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.
- D. Purchase a Full Utilization Reserved Instance to run the accounting software. Turn it off after hours. Run the batch jobs with the same instance class, so the Reserved Instance credits are also applied to the batch jobs.

Question #290

CloudFormation is used by your application to orchestrate its resources. During the testing process before to going live with the application, the Amazon RDS instance type was changed, resulting in the instance being recreated, resulting in the loss of test data.

How can you ensure that this does not happen again?

- A. Within the AWS CloudFormation parameter with which users can select the Amazon RDS instance type, set AllowedValues to only contain the current instance type.
- **B. Use an AWS CloudFormation stack policy to deny updates to the instance. Only allow UpdateStack permission to IAM principals that are denied SetStackPolicy.**
- C. In the AWS CloudFormation template, set the AWS::RDS::DBInstance's DBInstanceClass property to be read-only.
- D. Subscribe to the AWS CloudFormation notification "BeforeResourceUpdate," and call CancelStackUpdate if the resource identified is the Amazon RDS instance.
- E. In the AWS CloudFormation template, set the DeletionPolicy of the AWS::RDS::DBInstance's DeletionPolicy property to "Retain."

Commented [LC198]: Ans B After you set a stack policy, all of the resources in the stack are protected by default. To allow updates on specific resources, you specify an explicit Allow statement for those resources in your stack policy. You can define only one stack policy per stack, but, you can protect multiple resources within a single policy. A stack policy applies to all AWS CloudFormation users who attempt to update the stack. You can't associate different stack policies with different users.

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/protect-stack-resources.html>

Question #291

An IT team has created an AWS CloudFormation template to enable other members of the organization to deploy and terminate applications fast and reliably. The template builds an Amazon EC2 instance with a user data script for application installation and an Amazon S3 bucket for serving static websites while the application is running.

When deleting the CloudFormation stack, all resources should be erased as well. However, the team notices that CloudFormation generates an error when attempting to remove the stack, and the S3 bucket generated by the stack is not erased.

How can the team fix the mistake in the MOST EFFECTIVE way possible while ensuring that all resources are destroyed successfully?

- A. Add DeletionPolicy attribute to the S3 bucket resource, with the value Delete forcing the bucket to be removed when the stack is deleted.
- **B. Add a custom resource when an AWS Lambda function with the DependsOn attribute specifying the S3 bucket, and an IAM role. Write the Lambda function to delete all objects from the bucket when the RequestType is Delete.**
- C. Identify the resource that was not deleted. From the S3 console, empty the S3 bucket and then delete it.
- D. Replace the EC2 and S3 bucket resources with a single AWS OpsWorks Stacks resource. Define a custom recipe for the stack to create and delete the EC2 instance and the S3 bucket.

Commented [LC199]:

Question #292

A business is doing performance testing on a web application that is hosted on Amazon EC2 instances behind an Application Load Balancer. The instances are distributed across several Availability Zones in an Auto Scaling group. When releasing new software, the organization employs a blue/green deployment method with immutable instances.

Users are automatically logged out of the program at random periods during testing. Additionally, testers indicate that when an application's new version is deployed, all users are logged out. The development team need a solution that will maintain user logins throughout scaling events and application deployments.

Which method is the MOST EFFECTIVE for ensuring users stay logged in?

- A. Enable smart sessions on the load balancer and modify the application to check for an existing session.
- B. Enable session sharing on the load balancer and modify the application to read from the session store.
- C. Store user session information in an Amazon S3 bucket and modify the application to read session information from the bucket.
- **D. Modify the application to store user session information in an Amazon ElastiCache cluster.**

Commented [LC200]: <https://aws.amazon.com/cache/session-management/>

Question #293

You meet with your operations team once a month to examine the previous month's statistics. During the meeting, you discover that three weeks ago, your monitoring system, which pings your 3-tier web service API via HTTP from a location outside of AWS, detected a significant rise in latency on your 3-tier web service API. The database layer is DynamoDB, the business logic layer is ELB, EBS, and EC2, and the presentation layer is SQS, ELB, and EC2.

Which of the following strategies will NOT assist you in determining what occurred?

- A. Check your CloudTrail log history around the spike's time for any API calls that caused slowness.
- B. Review CloudWatch Metrics graphs to determine which component(s) slowed the system down.
- C. Review your ELB access logs in S3 to see if any ELBs in your system saw the latency.
- D. Analyze your logs to detect bursts in traffic at that time.

Commented [LC201]: CloudTrail monitors any AWS API calls, not calls to your custom API.

Question #294

An ecommerce business is seeking for methods to install an application on Amazon Web Services that meets the following requirements:

- ☞ Possesses an easy-to-use, fully automated application deployment method.
- ☞ Has low deployment costs and guarantees that at least half of the instances are online to handle end-user queries.
- ☞ If the application fails, it will be replaced with an automatic healing process.

Which deployment technique will be most effective in meeting these criteria?

- A. Create an AWS Elastic Beanstalk environment and configure it to use Auto Scaling and an Elastic Load Balancer. Use rolling deployments with a batch size of 50%.
- B. Create an AWS OpsWorks stack. Configure the application layer to use rolling deployments as a deployment strategy. Add an Elastic Load Balancing layer. Enable auto healing on the application layer.
- C. Use AWS CodeDeploy with Auto Scaling and an Elastic Load Balancer. Use the CodeDeployDefault:HalfAtATime deployment strategy. Enable an Elastic Load Balancing health check to report the status of the application, and set the Auto Scaling health check to ELB.
- D. Use AWS CodeDeploy with Auto Scaling and an Elastic Load Balancer. Use a blue/green deployment strategy. Enable an Elastic Load Balancing health check to report the status of the application, and set the Auto Scaling health check to ELB.

Commented [LC202]: I am leaning towards C as well.
A - nothing has been mentioned about LB health checks configuration. in this case, new instances will be added to the LB without health checks
B - is not simple and may not work at all.
D - pricing overhead with Blue/Green deployment option

Question #295

A DevOps Engineer is responsible for managing an application that requires cross-region failover. The application stores its data in a main region Amazon Aurora on Amazon RDS database with a read replica in a secondary region Amazon Aurora on Amazon RDS database. The program routes consumer traffic to the active area through Amazon Route 53.

Which procedures should be followed to minimize downtime in the event of a main database failure?

- A. Use Amazon CloudWatch to monitor the status of the RDS instance. In the event of a failure, use a CloudWatch Events rule to send a short message service (SMS) to the Systems Operator using Amazon SNS. Have the Systems Operator redirect traffic to an Amazon S3 static website that displays a downtime message. Promote the RDS read replica to the master. Confirm that the application is working normally, then redirect traffic from the Amazon S3 website to the secondary region.
- B. Use RDS Event Notification to publish status updates to an Amazon SNS topic. Use an AWS Lambda function subscribed to the topic to monitor database health. In the event of a failure, the Lambda function promotes the read replica, then updates Route 53 to redirect traffic from the primary region to the secondary region.
- C. Set up an Amazon CloudWatch Events rule to periodically invoke an AWS Lambda function that checks the health of the primary database. If a failure is detected, the Lambda function promotes the read replica. Then, update Route 53 to redirect traffic from the primary to the secondary region.
- D. Set up Route 53 to balance traffic between both regions equally. Enable the Aurora multi-master option, then set up a Route 53 health check to analyze the health of the databases. Configure Route 53 to automatically direct all traffic to the secondary region when a primary database fails.

Commented [LC203]:

Question #296

A business is transferring its public-facing software to Amazon Web Services. The firm intends to execute application code on Amazon EC2 and to store all application data on Amazon RDS. The organization intends to employ a single Region with failover capabilities to a backup Region, as well as Amazon Route 53 for traffic routing. RPO is two hours while RTO is four hours.

Which combination of measures should be taken to satisfy these goals while keeping costs to a minimum? (Select three.)

- A. Create an AWS CloudFormation template to provision the application server and database instance in a single Region.
- B. Create an AWS CloudFormation template to provision the application tier of the application and a multi-Region database instance.
- C. Configure Amazon CloudWatch Events rules to run every hour. Trigger AWS Lambda functions to create an RDS snapshot and copy it to the secondary Region.
- D. Configure Amazon CloudWatch Events rules to run every 3 hours. Trigger AWS Lambda functions to create an RDS snapshot and copy it to the secondary Region.
- E. In the event of a failure, deploy a new AWS CloudFormation stack in a secondary region to provision the application resources and a new RDS instance using the copied snapshot and a Route 53 failover routing policy.
- F. In the event of a failure, deploy a new AWS CloudFormation stack in a secondary region to provision the application resources and a replica of the RDS database using the copied snapshot and a Route 53 latency-based routing policy.

Commented [LC204]:

Commented [LC205]:

Commented [LC206]:

Question #297

You're in the process of developing a new API for video game scores. Reads outnumber writes by a factor of 100, and the top 1% of scores are read 100 times more often than the other 99%.

What is the optimal architecture for this system when DynamoDB is used?

- A. DynamoDB table with 100x higher read than write throughput, with CloudFront caching.
- B. DynamoDB table with roughly equal read and write throughput, with CloudFront caching.
- C. DynamoDB table with 100x higher read than write throughput, with ElastiCache caching.
- D. DynamoDB table with roughly equal read and write throughput, with ElastiCache caching.

Commented [LC207]: Because the 100x read ratio is mostly driven by a small subset, with caching, only a roughly equal number of reads to writes will miss the cache, since the supermajority will hit the top 1% scores. Knowing we need to set the values roughly equal when using caching, we select AWS ElastiCache, because CloudFront cannot directly cache DynamoDB queries, and ElastiCache is an excellent in-memory cache for database queries, rather than a distributed proxy cache for content delivery.... One solution would be to cache these reads at the application layer. Caching is a technique that is used in many high-throughput applications, offloading read activity on hot items to the cache rather than to the database. Your application can cache the most popular items in memory, or use a product such as ElastiCache to do the same.

Question #298

Which syntax for the AWS command to establish a single region trail is correct?

- A. aws create-trail --name trailname --s3-object objectname
- B. aws cloudtrail --s3-regionname IPaddress create-trail --name trailname
- C. aws cloudtrail create-trail --name trailname --s3-bucket-name bucketname
- D. aws cloudtrail create-trail --name trailname --s3-portnumber IPaddress

Commented [LC208]: The command `aws cloudtrail create-trail --name trailname --s3-bucket-name bucketname` will create a single region trail. You must create a S3 bucket before you execute the command, with proper CloudTrail permissions applied to it (and you must have the AWS command line tools (CLI) on your system).
Reference:
<http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-and-update-a-trail-by-using-the-aws-cli.html>

Question #299

Currently, an organization's application is deployed to a single AWS Region. Recently, the firm expanded to a new continent. Users in the new office are reporting a significant increase in latency. The company's application is hosted on Amazon EC2 instances behind an Application Load Balancer (ALB), and the database layer is powered by Amazon DynamoDB. The instances are distributed across several Availability Zones through an EC2 Auto Scaling group. A DevOps Engineer is responsible for reducing application response times and increasing user availability in both Regions.

Which combination of activities is optimal for resolving latency issues? (Select three.)

- A. Create a new DynamoDB table in the new Region with cross-Region replication enabled.
- B. Create new ALB and Auto Scaling group global resources and configure the new ALB to direct traffic to the new Auto Scaling group.
- C. Create new ALB and Auto Scaling group resources in the new Region and configure the new ALB to direct traffic to the new Auto Scaling group.
- D. Create Amazon Route 53 records, health checks, and latency-based routing policies to route to the ALB.
- E. Create Amazon Route 53 aliases, health checks, and failover routing policies to route to the ALB.
- F. Convert the DynamoDB table to a global table.

Commented [LC209]:

Commented [LC210]:

Commented [LC211]:

Questions 300-349

Question #300

An information security policy demands that important operating system security updates be applied to all publicly accessible systems within 24 hours of their release. Each instance has the Patch Group key set to 0. Two new AWS Systems Manager patch baselines for Windows and Red Hat Enterprise Linux (RHEL) have been produced with a zero-day delay for security fixes of critical severity. The new patch baselines have been connected with Patch Group 0.

Which two procedures will enable patch compliance and reporting to be automated? (Select two.)

- A. Create an AWS Systems Manager Maintenance Window and add a target with Patch Group 0. Add a task that runs the AWS-InstallWindowsUpdates document with a daily schedule.
- B. Create an AWS Systems Manager Maintenance Window with a daily schedule and add a target with Patch Group 0. Add a task that runs the AWS-RunPatchBaseline document with the Install action.
- C. Create an AWS Systems Manager State Manager configuration. Associate the AWS-RunPatchBaseline task with the configuration and add a target with Patch Group 0.
- D. Create an AWS Systems Manager Maintenance Window and add a target with Patch Group 0. Add a task that runs the AWS-ApplyPatchBaseline document with a daily schedule.
- E. Use the AWS Systems Manager Run Command to associate the AWS-ApplyPatchBaseline document with instances tagged with Patch Group 0.

Question #301

Sensitive data is stored by your application on an EBS volume associated to your EC2 instance.

How can you safeguard your data? (Select two.)

- A. Unmount the EBS volume, take a snapshot and encrypt the snapshot. Re-mount the Amazon EBS volume.
- B. It is not possible to encrypt an EBS volume, you must use a lifecycle policy to transfer data to S3 for encryption.
- C. Copy the unencrypted snapshot and check the box to encrypt the new snapshot. Volumes restored from this encrypted snapshot will also be encrypted.
- D. Create and mount a new, encrypted Amazon EBS volume. Move the data to the new volume. Delete the old Amazon EBS volume.

Question #302

An application development team works in three environments: development, pre-production, and production. AWS CodePipeline was recently accepted by the team.

However, the team has seen multiple instances of misconfigured or nonfunctional development code being sent into the production environment, causing disruption and downtime for users. The DevOps Engineer must examine the pipeline and include processes for identifying application issues prior to deployment.

What should the Engineer perform throughout the deployment process to uncover functional issues? (Select two.)

- A. Use Amazon Inspector to add a test action to the pipeline. Use the Amazon Inspector Runtime Behavior Analysis Inspector rules package to check that the deployed code complies with company security standards before deploying it to production.
- B. Using AWS CodeBuild to add a test action to the pipeline to replicate common user activities and ensure that the results are as expected before progressing to production deployment.
- C. Create an AWS CodeDeploy action in the pipeline with a deployment configuration that automatically deploys the application code to a limited number of instances. The action then pauses the deployment so that the QA team can review the application functionality. When the review is complete, CodeDeploy resumes and deploys the application to the remaining production Amazon EC2 instances.
- D. After the deployment process is complete, run a testing activity on an Amazon EC2 instance in a different region that accesses the application to simulate user behavior. If unexpected results occur, the testing activity sends a warning to an Amazon SNS topic. Subscribe to the topic to get updates.
- E. Add an AWS CodeDeploy action in the pipeline to deploy the latest version of the development code to pre-production. Add a manual approval action in the pipeline so that the QA team can test and confirm the expected functionality. After the manual approval action, add a second CodeDeploy action that deploys the approved code to the production environment.

Commented [LC212]: Right Answer is BC.

D, E - There is only deprecated AWS-ApplyPatchBaseline document in SSM. So D, E are incorrect.

<https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-manager-ssm-documents.html>

A - is incorrect because it only takes care of Windows Update not Linux

Commented [LC213]:

Commented [LC214]: These steps are given in the AWS documentation
To migrate data between encrypted and unencrypted volumes

1) Create your destination volume (encrypted or unencrypted, depending on your need).

2) Attach the destination volume to the instance that hosts the data to migrate.

3) Make the destination volume available by following the procedures in Making an Amazon EBS Volume Available for Use. For Linux instances, you can create a mount point at /mnt/destination and mount the destination volume there.

4) Copy the data from your source directory to the destination volume. It may be most convenient to use a bulk-copy utility for this.

To encrypt a volume's data by means of snapshot copying

1) Create a snapshot of your unencrypted CBS volume. This snapshot is also unencrypted.

2) Copy the snapshot while applying encryption parameters. The resulting target snapshot is encrypted.

3) Restore the encrypted snapshot to a new volume, which is also encrypted.

Commented [LC215]:

Commented [LC216]: I'll go with B,E

A) wrong because it's not Amazon Inspector's purpose

C) wrong, come on, this is absolutely stupid...

D) wrong, why should I use an external "agent" if I can do this using the pipeline? this is turning things even harder

Commented [LC217]:

Question #303

A business is implementing a centralized log management system on AWS and has a number of needs. The organization desires that its Amazon CloudWatch and VPC Flow logs originate from separate sub accounts and be provided to a single auditing account. However, the number of sub accounts changes on a regular basis. Additionally, the organization must index the auditing account's logs in order to gain useful knowledge.

How should a DevOps Engineer build the solution in such a way that it satisfies all of the organization's requirements?

- A. Use AWS Lambda to write logs to Amazon ES in the auditing account. Create an Amazon CloudWatch subscription filter and use Amazon Kinesis Data Streams in the sub accounts to stream the logs to the Lambda function deployed in the auditing account.
- B. Use Amazon Kinesis Streams to write logs to Amazon ES in the auditing account. Create a CloudWatch subscription filter and use Kinesis Data Streams in the sub accounts to stream the logs to the Kinesis stream in the auditing account.
- **C. Use Amazon Kinesis Firehose with Kinesis Data Streams to write logs to Amazon ES in the auditing account. Create a CloudWatch subscription filter and stream logs from sub accounts to the Kinesis stream in the auditing account.**
- D. Use AWS Lambda to write logs to Amazon ES in the auditing account. Create a CloudWatch subscription filter and use Lambda in the sub accounts to stream the logs to the Lambda function deployed in the auditing account.

Commented [LC218]: A: Incorrect: You can create CloudWatch Logs subscription filters with Kinesis, Lambda, or Kinesis Data Firehose. Also it is possible to use a lambda to write to ES
<https://aws.amazon.com/blogs/aws/category/amazon-elasticsearch-service/>

Not sure why to use an additional Kinesis Data Streams between CloudWatch and Lambda (Ans D).

But both A and D are incorrect because it is possible to use a Lambda function belonging to the same account as the subscription filter, for same-account delivery

B. Incorrect: Only Lagstash and Kinesis Firehose can write to ES
<https://aws.amazon.com/opensearch-service/the-elk-stack/what-is-elasticsearch/>

C: Correct: Create a CloudWatch Subscription filter with Kinesis Streams, send data from Streams to Firehose and Firehose writes to ES.

Commented [LC219]: Correct Answer is "C" You can use CloudFormation StackSets to enable authorization programmatically across all source accounts. Also note that the authorization step is not needed if you choose to aggregate all accounts in your AWS Organization.

Question #304

A business has implemented a number of apps on a worldwide scale. Security auditors recently discovered that a small number of Amazon EC2 instances were deployed without Amazon EBS disk encryption. The auditors have demanded a report outlining all EBS volumes in different AWS accounts and regions that were not secured. Additionally, they want to be alerted if this happens in the future.

How can this be automated with the FEATEST operational overhead possible?

- A. Create an AWS Lambda function to set up an AWS Config rule on all the target accounts. Use AWS Config aggregators to collect data from multiple accounts and regions. Export the aggregated report to an Amazon S3 bucket and use Amazon SNS to deliver the notifications.
- B. Set up AWS CloudTrail to deliver all events to an Amazon S3 bucket in a centralized account. Use the S3 event notification feature to invoke an AWS Lambda function to parse AWS CloudTrail logs whenever logs are delivered to the S3 bucket. Publish the output to an Amazon SNS topic using the same Lambda function.
- **C. Create an AWS CloudFormation template that adds an AWS Config managed rule for EBS encryption. Use a CloudFormation stack set to deploy the template across all accounts and regions. Store consolidated evaluation results from config rules in Amazon S3. Send a notification using Amazon SNS when non-compliant resources are detected.**
- D. Using AWS CLI, run a script periodically that invokes the aws ec2 describe-volumes query with a JMESPATH query filter. Then, write the output to an Amazon S3 bucket. Set up an S3 event notification to send events using Amazon SNS when new data is written to the S3 bucket.

Question #305

A software business wants to automate the build process for a GitHub-hosted project. When a repository's source code is changed, it should be built, tested, and uploaded to Amazon S3.

Which combination of measures would be most effective in meeting these requirements? (Select three.)

- **A. Add a buildspec.yml file to the source code with build instructions.**
- **B. Configure a GitHub webhook to trigger a build every time a code change is pushed to the repository.**
- **C. Create an AWS CodeBuild project with GitHub as the source repository.**
- D. Create an AWS CodeDeploy application with the Amazon EC2/On-Premises compute platform.
- E. Create an AWS OpsWorks deployment with the install dependencies command.
- F. Provision an Amazon EC2 instance to perform the build.

Commented [LC220]: I'll go with A, B, C

Reference:
<https://docs.aws.amazon.com/codebuild/latest/userguide/github-webhook.html>

Commented [LC221]:

Commented [LC222]:

Question #306 [SKIP]

Which storage driver does Docker suggest you use in general, assuming one is available?

- A. zfs
- B. btrfs
- C. aufs
- D. overlay

Question #307

Fill in the blanks: _____ assists us in tracking AWS API calls and transitions, _____ assists us in determining the current state of our resources, and _____ enables auditing of passwords and logins.

- A. AWS Config, CloudTrail, IAM Credential Reports
- B. CloudTrail, IAM Credential Reports, AWS Config
- C. CloudTrail, AWS Config, IAM Credential Reports
- D. AWS Config, IAM Credential Reports, CloudTrail

Commented [LC223]: C. CloudTrail for API calls / AWS Config for existing resources / IAM Credentials Reports for auditing credentials and logins

Question #308

Over 40 apps are being developed by a development team. Each application is a three-tiered web application built on Amazon EC2, Amazon RDS, and an ELB Application Load Balancer. Due to the apps' internal usage, the Security team want to restrict access to the 40 applications to the corporate network and to ban access from external IP addresses. Proxy servers connect the business network to the internet. The proxy servers have a total of 12 proxy IP addresses, which are updated once or twice a month. The Network Infrastructure team is responsible for proxy server management; they upload a file containing the most up-to-date proxy IP addresses to an Amazon S3 bucket. The DevOps Engineer is responsible for developing a solution that ensures apps are accessible from the corporate network.

Which solution satisfies these needs with the LEAST amount of influence on application development, the LEAST amount of operational work, and the LEAST amount of infrastructure expense?

- A. Implement an AWS Lambda function to read the list of proxy IP addresses from the S3 object and to update the ELB security groups to allow HTTPS only from the given IP addresses. Configure the S3 bucket to invoke the Lambda function when the object is updated. Save the IP address list to the S3 bucket when they are changed.
- B. Ensure that all the applications are hosted in the same Virtual Private Cloud (VPC). Otherwise, consolidate the applications into a single VPC. Establish an AWS Direct Connect connection with an active/standby configuration. Change the ELB security groups to allow only inbound HTTPS connections from the corporate network IP addresses.
- C. Implement a Python script with the AWS SDK for Python (Boto), which downloads the S3 object that contains the proxy IP addresses, scans the ELB security groups, and updates them to allow only HTTPS inbound from the given IP addresses. Launch an EC2 instance and store the script in the instance. Use a cron job to execute the script daily.
- D. Enable ELB security groups to allow HTTPS inbound access from the Internet. Use Amazon Cognito to integrate the company's Active Directory as the identity provider. Change the 40 applications to integrate with Amazon Cognito so that only company employees can log into the application. Save the user access logs to Amazon CloudWatch Logs to record user access activities

Commented [LC224]:

Question #309

A business wants to replace its existing bash deployment scripts with AWS development tools. At the moment, the firm is running a LAMP application on a cluster of Amazon EC2 instances protected by an Application Load Balancer (ALB). The business performs unit testing on the committed application, pauses and restarts services, unregisters and reregisters instances with the load balancer, and modifies file permissions during deployments. The organization wants to retain the same deployment capabilities after migrating to AWS.

Which solution will satisfy these criteria?

- A. Use AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy's appspec.yml file to restart services, and deregister and register instances with the ALB. Use the appspec.yml file to update file permissions without a custom script.
- B. Use AWS CodePipeline to move the application from the AWS CodeCommit repository to AWS CodeDeploy. Use CodeDeploy's deployment group to test the application, unregister and re-register instances with the ALB, and restart services. Use the appspec.yml file to update the permissions without a custom script.
- C. Use AWS CodePipeline to move the application source code from the AWS CodeCommit repository to AWS CodeDeploy. Use CodeDeploy to test the application. Use CodeDeploy's appspec.yml file to restart services and update permissions without a custom script. Use AWS CodeBuild to unregister and re-register instances with the ALB.
- **D. Use AWS CodePipeline to trigger AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy's appspec.yml file to restart services. Unregister and re-register the instances in the AWS CodeDeploy deployment group with the ALB. Update the appspec.yml file to update file permissions without a custom script.**

Commented [LC225]:

Question #310

You run numerous environments in several regions and want to utilize Amazon Inspector to conduct regular security audits on all AWS resources in all locations.

Which statement concerning the functioning of Amazon Inspector across regions is true?

- A. Amazon Inspector is a global service that is not region-bound. You can include AWS resources from multiple regions in the same assessment target.
- **B. Amazon Inspector is hosted within AWS regions behind a public endpoint. All regions are isolated from each other, and the telemetry and findings for all assessments performed within a region remain in that region and are not distributed by the service to other Amazon Inspector locations.**
- C. Amazon Inspector is hosted in each supported region. Telemetry data and findings are shared across regions to provide complete assessment reports.
- D. Amazon Inspector is hosted in each supported region separately. You have to create assessment targets using the same name and tags in each region and Amazon Inspector will run against each assessment target in each region.

Commented [LC226]:

Question #311

A security assessment discovered that an AWS CodeBuild project is utilizing an unauthenticated request to retrieve a database population script from an Amazon S3 bucket. For this project, the Security team does not allow unauthenticated queries to S3 buckets.

How can this problem be resolved in the MOST SECURE way possible?

- A. Add the bucket name to the AllowedBuckets section of the CodeBuild project settings. Update the build spec to use the AWS CLI to download the database population script.
- B. Modify the S3 bucket settings to enable HTTPS basic authentication and specify a token. Update the build spec to use cURL to pass the token and download the database population script.
- **C. Remove unauthenticated access from the S3 bucket with a bucket policy. Modify the service role for the CodeBuild project to include Amazon S3 access. Use the AWS CLI to download the database population script.**
- D. Remove unauthenticated access from the S3 bucket with a bucket policy. Use the AWS CLI to download the database population script using an IAM access key and a secret access key.

Commented [LC227]:

Question #312

Your mobile application contains a photo-sharing service that will initially attract tens of thousands of users. You'll utilize Amazon Simple Storage Service (S3) to store user photos, and you'll need to select how to verify and approve access to these images for your users. Additionally, you must manage the storage of these photographs.

Which of the following two strategies should you employ? (Select two.)

- A. Create an Amazon S3 bucket per user, and use your application to generate the S3 URI for the appropriate content.
- B. Use AWS Identity and Access Management (IAM) user accounts as your application-level user database, and offload the burden of authentication from your application code.
- C. Authenticate your users at the application level, and use AWS Security Token Service (STS) to grant token-based authorization to S3 objects.
- D. Authenticate your users at the application level, and send an SMS token message to the user. Create an Amazon S3 bucket with the same name as the SMS message token, and move the user's objects to that bucket.
- E. Use a key-based naming scheme comprised from the user IDs for all user objects in a single Amazon S3 bucket.

Commented [LC228]:

Commented [LC229]:

Question #313

Multiple development teams inside a corporation collaborate on a single shared AWS account. The Senior Manager of the groups want to be notified through a third-party API call when resource generation reaches the account's service restrictions.

Which option has the LEAST amount of development effort?

- A. Create an Amazon CloudWatch Event rule that runs periodically and targets an AWS Lambda function. Within the Lambda function, evaluate the current state of the AWS environment and compare deployed resource values to resource limits on the account. Notify the Senior Manager if the account is approaching a service limit.
- B. Deploy an AWS Lambda function that refreshes AWS Trusted Advisor checks, and configure an Amazon CloudWatch Events rule to run the Lambda function periodically. Create another CloudWatch Events rule with an event pattern matching Trusted Advisor events and a target Lambda function. In the target Lambda function, notify the Senior Manager.
- C. Deploy an AWS Lambda function that refreshes AWS Personal Health Dashboard checks, and configure an Amazon CloudWatch Events rule to run the Lambda function periodically. Create another CloudWatch Events rule with an event pattern matching Personal Health Dashboard events and a target Lambda function. In the target Lambda function, notify the Senior Manager.
- D. Add an AWS Config custom rule that runs periodically, checks the AWS service limit status, and streams notifications to an Amazon SNS topic. Deploy an AWS Lambda function that notifies the Senior Manager, and subscribe the Lambda function to the SNS topic.

Commented [LC230]: <https://aws.amazon.com/solutions/implementations/limit-monitor/>

Question #314

A corporation intends to discontinue the usage of Amazon EC2 key pairs for SSH access in favor of AWS Systems Manager Session Manager. To further increase security, Session Manager access must be restricted to a private network.

Which acts in combination will do this? (Select two.)

- A. Allow inbound access to TCP port 22 in all associated EC2 security groups from the VPC CIDR range.
- B. Attach an IAM policy with the necessary Systems Manager permissions to the existing IAM instance profile.
- C. Create a VPC endpoint for Systems Manager in the desired Region.
- D. Deploy a new EC2 instance that will act as a bastion host to the rest of the EC2 instance fleet.
- E. Remove any default routes in the associated route tables.

Commented [LC231]: I choose B&C. A - wrong. There is no need to open doors. B - Correct C - Correct D - System Manager does not need a bastion host. It's wrong.

Reference:
<https://aws.amazon.com/en/blogs/aws/new-session-manager/>
<https://cloudonaut.io/goodbye-ssh-use-aws-session-manager-instead/>

Commented [LC232]:

Question #315

Your application's Auto Scaling Group grows too rapidly and excessively, and then remains scaled even when traffic reduces.

What are your options for resolving this?

- A. Set a longer cooldown period on the Group, so the system stops overshooting the target capacity. The issue is that the scaling system does not allow enough time for new instances to begin servicing requests before measuring aggregate load again.
- B. Calculate the bottleneck or constraint on the compute layer, then select that as the new metric, and set the metric thresholds to the bounding values that begin to affect response latency.
- C. Raise the CloudWatch Alarms threshold associated with your autoscaling group, so the scaling takes more of an increase in demand before beginning.
- D. Use larger instances instead of many smaller ones, so the Group stops scaling out so much and wasting resources as the OS level, since the OS uses a higher proportion of resources on smaller instances.

Commented [LC233]: <https://docs.aws.amazon.com/autoscaling/ec2/userguide/Cooldown.html>

Scaling cooldown helps you prevent your Auto Scaling group from launching or terminating additional instances before the effects of previous activities are visible.

When you use simple scaling, after the Auto Scaling group scales using a simple scaling policy, it waits for a cooldown period to complete before any further scaling activities initiated by simple scaling policies can start. An adequate cooldown period helps to prevent the initiation of an additional scaling activity based on stale metrics.

Question #316

Your application runs on an Auto Scaling group of m3.large machines and gets messages from an Amazon SQS queue. After a period, the group's maximum number of instances is reached, and the queue's message count continues to grow. You've found that a third-party library that the program makes use of has a problem that results in a memory leak.

What cost-effective measures can you take to ensure that message processing continues while the library developer resolves the issue?

- A. Enable Elastic Load Balancing health checks for the Auto Scaling group. When Elastic Load Balancing has detected a failure, Auto Scaling will terminate the failing application's instance and launch a new one.
- B. Use Amazon EC2 instance memory usage CloudWatch metrics to raise alerts when they reach a defined level and send a message to Auto Scaling to fail the instance health check.
- C. Use application monitoring on the instance to restart the application when memory usage reaches a defined level.
- D. Create a new Auto Scaling launch configuration to use the r3.large instance type. Update the Auto Scaling group with the new launch configuration.

Commented [LC234]: I gravitate more to A. It uses conventional approach, only aws means and will work out.

B is nonsense: you can't fail an instance health check from CloudWatch. Moreover, CloudWatch agent didn't mentioned.

C. It's a aws exam, remember. Which app monitoring? To vague and unclear as to me.

D. It's a band-aid for a wall crack. It's a memory leak, doesn't matter how much memory you have.

Question #317

A business has created an AWS Lambda function to process orders received through an API. The organization is deploying the Lambda function using AWS CodeDeploy as the last step of a continuous integration/continuous delivery pipeline.

A DevOps Engineer has found that the ordering API is sometimes unavailable for a few seconds after deployment. Following some examination, the DevOps Engineer thinks the problems are caused by database updates not propagating entirely before the Lambda function starts execution.

How is the DevOps Engineer to circumvent this?

- A. Add a BeforeAllowTraffic hook to the AppSpec file that tests and waits for any necessary database changes before traffic can flow to the new version of the Lambda function.
- B. Add an AfterAllowTraffic hook to the AppSpec file that forces traffic to wait for any pending database changes before allowing the new version of the Lambda function to respond.
- C. Add a BeforeInstall hook to the AppSpec file that tests and waits for any necessary database changes before deploying the new version of the Lambda function.
- D. Add a ValidateService hook to the AppSpec file that inspects incoming traffic and rejects the payload if dependent services, such as the database, are not yet ready.

Commented [LC235]: A - Correct

B - Does not fit into the frame of the question. We want the database to be up and running BEFORE we start serving production traffic.

C - Not applicable in the appspec.yml file for Lambda

D - Same explanation as C

Question #318

You're using Amazon Elastic Beanstalk to host an ASP.NET web application. Your subsequent version of the program needs that a third-party Windows Installer package be loaded on the instance at first boot and prior to the application launching.

Which alternatives are available? (Select two.)

- A. In the application's Global.asax file, run `msiexec.exe` to install the package using `Process.Start()` in the Application Start event handler.
- B. In the source bundle's `.ebextensions` folder, create a file with a `.config` extension. In the file, under the "packages" section and "msi" package manager, include the package's URL.
- C. Launch a new Amazon EC2 instance from the AMI used by the environment. Log into the instance, install the package and run `sysprep`. Create a new AMI. Configure the environment to use the new AMI.
- D. In the environment's configuration, edit the instances configuration and add the package's URL to the "Packages" section.
- E. In the source bundle's `.ebextensions` folder, create a "Packages" folder. Place the package in the folder.

Commented [LC236]: BC

<https://aws.amazon.com/blogs/developer/customizing-windows-elastic-beanstalk-environments-part-1/>

Commented [LC237]:

Question #319

A DevOps Engineer manages shipping orders and inventories using a single Amazon DynamoDB database. Three AWS Lambda functions are reading from a DynamoDB stream on that table, according to the Engineer. Lambda functions execute a variety of tasks, including item counting, transferring things to Amazon Kinesis Data Firehose, monitoring inventory levels, and making vendor orders when components become scarce.

While checking logs, the Engineer observes that Lambda functions sometimes fail due to increasing traffic, with an error indicating stream throttling.

Which of the following solutions is the MOST cost-effective and needs the LEAST amount of operational management?

- A. Use AWS Glue integration to ingest the DynamoDB stream, then migrate the Lambda code to an AWS Fargate task.
- B. Use Amazon Kinesis streams instead of DynamoDB streams, then use Kinesis analytics to trigger the Lambda functions.
- C. Create a fourth Lambda function and configure it to be the only Lambda reading from the stream. Then use this Lambda function to pass the payload to the other three Lambda functions.
- D. Have the Lambda functions query the table directly and disable DynamoDB streams. Then have the Lambda functions query from a global secondary index.

Commented [LC238]: C - fan out

<https://aws.amazon.com/de/blogs/database/how-to-perform-ordered-data-replication-between-applications-by-using-amazon-dynamodb-streams/>

Question #320

A DevOps Engineer is responsible for the administration of a major commercial website that is hosted on Amazon EC2. The website collects and processes web logs using Amazon Kinesis Data Streams. The Engineer is responsible for the management of the Kinesis consumer application, which is also hosted on EC2. Spikes in data lead the Kinesis consumer application to lag behind, and the streams fail to process records.

What is the FASTEST way for stream handling improvement?

- A. Modify the Kinesis consumer application to store the logs durably in Amazon S3. Use Amazon EMR to process the data directly on S3 to derive customer insights and store the results in S3.
- B. Horizontally scale the Kinesis consumer application by adding more EC2 instances based on the `GetRecord.IteratorAgeMilliseconds` Amazon CloudWatch metric. Increase the Kinesis Data Streams retention period.
- C. Convert the Kinesis consumer application to run as an AWS Lambda function. Configure the Kinesis Data Streams as the event source for the Lambda function to process the data streams.
- D. Increase the number of shards in the Kinesis Data Streams to increase the overall throughput so that the consumer application processes data faster.

Though DynamoDB lets you configure multiple Lambda functions with a stream, configuring more than two Lambda functions per stream increases the possibility of failed requests. The number of Lambda functions or processes that are allowed to read from a DynamoDB stream might increase in the future. To reliably process data in real time within a stream, you need to ensure that the Lambda function requests succeed. To enable parallel, ordered processing of stream data and have successful Lambda function requests, implement a fan-out pattern. In a Lambda fan-out pattern, you configure a single Lambda function to process the DynamoDB stream. Lambda polls a DynamoDB stream and, when it detects new records, invokes this Lambda function by passing in one or more events. The Lambda function processes each item and invokes downstream services or APIs.

Commented [LC239]: Option B

<https://docs.aws.amazon.com/streams/latest/dev/monitoring-with-cloudwatch.html>

`GetRecords.IteratorAgeMilliseconds` - The age of the last record in all `GetRecords` calls made against a Kinesis stream, measured over the specified time period. Age is the difference between the current time and when the last record of the `GetRecords` call was written to the stream. The Minimum and Maximum statistics can be used to track the progress of Kinesis consumer applications. A value of zero indicates that the records being read are completely caught up.

Question #321

A DevOps Engineer must automate a weekly process for finding superfluous permissions per user, across all users in an AWS account. This procedure should compare the rights presently provided to each user to the permissions the user has actually utilized in the last 90 days by reviewing the user's linked IAM access policies. Any discrepancies in the comparison would suggest that the user has been granted more access than necessary. A report of the deltas should be forwarded to the Information Security team for further examination and, if necessary, adjustments to the IAM user access policy.

Which solution is completely automated and generates the MOST DETAILED report on deltas?

- A. Create an AWS Lambda function that calls the IAM Access Advisor API to pull service permissions granted on a user-by-user basis for all users in the AWS account. Ensure that Access Advisor is configured with a tracking period of 90 days. Invoke the Lambda function using an Amazon CloudWatch Events rule on a weekly schedule. For each record, by user, by service, if the Access Advisor Last Accesses field indicates a day count instead of 'Not accesses in the tracking period,' this indicates a delta compared to what is in the user's currently attached access policies. After Lambda has iterated through all users in the AWS account, configure it to generate a report and send the report using Amazon SES.
- B. Configure an AWS CloudTrail trail that spans all AWS Regions and all read/write events, and point this trail to an Amazon S3 bucket. Create an Amazon Athena table and specify the S3 bucket ARN in the CREATE TABLE query. Create an AWS Lambda function that accesses the Athena table using the SDK, which performs a SELECT, ensuring that the WHERE clause includes userIdentity, eventName, and eventTime. Compare the results against the user's currently attached IAM access policies to determine any deltas. Configure an Amazon CloudWatch Events schedule to automate this process to run once a week. Configure Amazon SES to send a consolidated report to the Information Security team.
- C. Configure VPC Flow Logs on all subnets across all VPCs in all regions to capture user traffic across the entire account. Ensure that all logs are being sent to a centralized Amazon S3 bucket, so all flow logs can be consolidated and aggregated. Create an AWS Lambda function that is triggered once a week by an Amazon CloudWatch Events schedule. Ensure that the Lambda function parses the flow log files for the following information: IAM user ID, subnet ID, VPC ID, Allow/Reject status per API call, and service name. Then have the function determine the deltas on a user-by-user basis. Configure the Lambda function to send the consolidated report using Amazon SES.
- D. Create an Amazon ES cluster and note its endpoint URL, which will be provided as an environment variable into a Lambda function. Configure an Amazon S3 event on a AWS CloudTrail trail destination S3 bucket and ensure that the event is configured to send to a Lambda function. Create the Lambda function to consume the events, parse the input from JSON, and transform it to an Amazon ES document format. POST the documents to the Amazon ES cluster's endpoint by way of the passed-in environment variable. Make sure that the proper indexing exists in Amazon ES and use Apache Lucene queries to parse the permissions on a user-by-user basis. Export the deltas into a report and have Amazon ES send the reports to the Information Security team using Amazon SES every week.

Commented [LC240]: It's a tricky question. I'd go with B. Option A looks very promising because of common practice using Access Advisor in such cases, however tracking period is 400 days and it gives information about services access and not action level. Option B uses CloudTrail which gives also Event name that allows to determine the actions used across entities. See comments to the article <https://aws.amazon.com/cn/blogs/security/automate-analyzing-permissions-using-iam-access-advisor/>

Means CloudTrail allows to produce more detailed report.

Question #322

A major corporation is using AWS to host a web application. The application is deployed on Amazon EC2 instances that are routed via an Application Load Balancer. The instances are distributed across several Availability Zones in an Auto Scaling group. The program makes use of an Amazon RDS Oracle database and Amazon DynamoDB to store data.

There are distinct development, testing, and production environments.

Which method is the MOST SECURE and VERSATILE for obtaining password credentials during deployment?

- A. Retrieve an access key from an AWS Systems Manager SecureString parameter to access AWS services. Retrieve the database credentials from a Systems Manager SecureString parameter.
- B. Launch the EC2 instances with an EC2 IAM role to access AWS services. Retrieve the database credentials from AWS Secrets Manager.
- C. Retrieve an access key from an AWS Systems Manager plaintext parameter to access AWS services. Retrieve the database credentials from a Systems Manager SecureString parameter.
- D. Launch the EC2 instances with an EC2 IAM role to access AWS services. Store the database passwords in an encrypted config file with the application artifacts.

Commented [LC241]: Answer: B

AWS Secrets Manager is a secrets management service that helps you protect access to your applications, services, and IT resources. This service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Using Secrets Manager, you can secure and manage secrets used to access resources in the AWS Cloud, on third-party services, and on-premises.

Have in mind it is not so cheap as SSM parameter store \$0.40 per secret per month. For secrets that are stored for less than a month, the price is prorated (based on the number of hours.) \$0.05 per 10,000 API calls.

Question #323

A business created a static website that is hosted on an Amazon S3 bucket. AWS CloudFormation is used to deploy the website. The CloudFormation template creates an S3 bucket and a custom resource that replicates stuff from a source location into the bucket.

The firm has determined that it needs to relocate the website, which necessitates the deletion and re-creation of the current CloudFormation stack. CloudFormation, on the other hand, states that the stack could not be destroyed completely.

What is the MOST LIKELY CAUSE of this issue, and how can the DevOps Engineer prevent it for current and future versions of the website?

- A. Deletion has failed because the S3 bucket has an active website configuration. Modify the CloudFormation template to remove the WebsiteConfiguration property from the S3 bucket resource.
- B. Deletion has failed because the S3 bucket is not empty. Modify the custom resource's AWS Lambda function code to recursively empty the bucket when RequestType is Delete.
- C. Deletion has failed because the custom resource does not define a deletion policy. Add a DeletionPolicy property to the custom resource definition with a value of RemoveOnDeletion.
- D. Deletion has failed because the S3 bucket is not empty. Modify the S3 bucket resource in the CloudFormation template to add a DeletionPolicy property with a value of Empty.

Commented [LC242]: Not sure if people actually researched the options properly. CloudFormation DeletionPolicy property only has 3 options Delete/Retain/Snapshot. Option C and D value options do not exist = wrong. Left with 2 options. A and B. Option A. There is no WebsiteConfiguration property. The actual property is called "website" which only takes arguments for index/error pages. Option B. Question states custom resource = some sort of code construct. In AWS-speak = a AWS service of some kind. Lambda is referred to as custom resources in AWS documentation= only plausible answer than actually works. The RequestType is Delete, this text gives away the answer. Basically > Custom resource = Lambda - hey lambda, when a make Delete request to remove the S3 bucket - recursively delete the objects in the S3 bucket first and then delete it. You can only delete an S3 bucket if it has not contents, specifically - CloudFormation > create bucket > no contents > remove/delete CloudFormation stack = deletes S3 bucket. Only time this behaviour changes is if there any objects in bucket of deletion policy is set.

Question #324

AWS CodeDeploy is being used by a business to automate software deployment. The deployment must adhere to the following criteria:

- ⇒ During the deployment, a number of instances must be accessible to service traffic.
- ⇒ Traffic must be distributed evenly across those instances, and the instances must self-heal in the case of a failure.
- ⇒ A fresh fleet of instances must be established for the purpose of automatically deploying a new version without human provisioning.
- ⇒ Traffic must be diverted to half of the new instances at a time to the new environment. If traffic is diverted to at least half of the instances, the deployment should succeed; otherwise, it should fail.
- ⇒ Prior to directing traffic to the new fleet of instances, it is necessary to erase the temporary files created during the deployment process.
- ⇒ To minimize expenses, the original instances in the deployment group must be destroyed promptly after a successful deployment.

How can a DevOps Engineer adhere to these standards?

- A. Use an Application Load Balancer and an in-place deployment. Associate the Auto Scaling group with the deployment group. Use the Automatically copy Auto Scaling group option, and use CodeDeployDefault.OneAtATime as the deployment configuration. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the AllowTraffic hook within appspec.yml to delete the temporary files.
- B. Use an Application Load Balancer and a blue/green deployment. Associate the Auto Scaling group and the Application Load Balancer target group with the deployment group. Use the Automatically copy Auto Scaling group option, create a custom deployment configuration with minimum healthy hosts defined as 50%, and assign the configuration to the deployment group. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the BeforeBlockTraffic hook within appspec.yml to delete the temporary files.
- C. Use an Application Load Balancer and a blue/green deployment. Associate the Auto Scaling group and the Application Load Balancer target group with the deployment group. Use the Automatically copy Auto Scaling group option, and use CodeDeployDefault.HalfAtATime as the deployment configuration. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the BeforeAllowTraffic hook within appspec.yml to delete the temporary files.
- D. Use an Application Load Balancer and an in-place deployment. Associate the Auto Scaling group and Application Load Balancer target group with the deployment group. Use the Automatically copy Auto Scaling group option, and use CodeDeployDefault.AllatOnce as a deployment configuration. Instruct AWS CodeDeploy to terminate the original instances in the deployment group, and use the BlockTraffic hook within appspec.yml to delete the temporary files.

Commented [LC243]: Agree with C. Few notes why the other answers are incorrect:
- A&D are in-place and don't meet the requirement of new fleet for new deployment.
- B reference a BeforeBlockTraffic hook to delete the temporary files from the "replacement" instances; however, that hook is only available for the "original" instances. Hence B has an invalid statement.

Question #325

All of a company's internal quality control apps have been containerized. Jenkins is being used by the organization on Amazon EC2, which needs patching and updating. The Compliance Officer has directed that a DevOps Engineer begin encrypting build artifacts, since they include proprietary information about the organization.

What should the DevOps Engineer do to ensure that this is accomplished in the MOST maintainable way possible?

- A. Automate patching and upgrading using AWS Systems Manager on EC2 instances and encrypt Amazon EBS volumes by default.
- B. Deploy Jenkins to an Amazon ECS cluster and copy build artifacts to an Amazon S3 bucket with default encryption enabled.
- C. Leverage AWS CodePipeline with a build action and encrypt the artifacts using AWS Secrets Manager.
- **D. Use AWS CodeBuild with artifact encryption to replace the Jenkins instance running on Amazon EC2.**

Commented [LC244]: D.

<https://docs.aws.amazon.com/codebuild/latest/userguide/security-encryption.html>

Build artifact encryption - CodeBuild requires access to an AWS KMS CMK in order to encrypt its build output artifacts. By default, CodeBuild uses an AWS Key Management Service CMK for Amazon S3 in your AWS account. If you do not want to use this CMK, you must create and configure a customer-managed CMK. For more information Creating keys.

Question #326 [SKIP]

Which syntax is appropriate for referring the value of a variable in an Ansible task?

- A. \${variable_name}
- B. { variable_name }
- C. "{{ variable_name }}"
- D. @variable_name

Question #327

You've been charged with the responsibility of building a scalable distributed system on AWS OpsWorks. Your distributed system must scale dynamically. Due to the dispersed nature of the layer, each node must maintain a configuration file including the hostnames of the other instances.

How should AWS OpsWorks be configured to support dynamic scaling of this application?

- **A. Create a Chef Recipe to update this configuration file, configure your AWS OpsWorks stack to use custom cookbooks, and assign this recipe to the Configure LifeCycle Event of the specific layer.**
- B. Update this configuration file by writing a script to poll the AWS OpsWorks service API for new instances. Configure your base AMI to execute this script on Operating System startup.
- C. Create a Chef Recipe to update this configuration file, configure your AWS OpsWorks stack to use custom cookbooks, and assign this recipe to execute when instances are launched.
- D. Configure your AWS OpsWorks layer to use the AWS-provided recipe for distributed host configuration, and configure the instance hostname and file path parameters in your recipes settings.

Commented [LC245]: Ans is A:

<https://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook-events.html>

Question #328

Your infrastructure is presently hosted on Amazon EC2 instances protected by an Auto Scaling group. Currently, your application's logs are written to ephemeral storage. Recently, your organization encountered a significant issue in code that survived testing and was eventually released to your fleet. This problem caused your Auto Scaling group to scale up and down many times before you were able to correctly get the logs from your server to aid in diagnosing the bug.

Which strategy should you use to ensure that you may inspect your logs after your instances have been terminated?

- A. Configure the ephemeral policies on your Auto Scaling group to back up on terminate.
- B. Configure your Auto Scaling policies to create a snapshot of all ephemeral storage on terminate.
- **C. Install the CloudWatch Logs Agent on your AMI, and configure CloudWatch Logs Agent to stream your logs.**
- D. Install the CloudWatch monitoring agent on your AMI, and set up new SNS alert for CloudWatch metrics that triggers the CloudWatch monitoring agent to backup all logs on the ephemeral drive.
- E. Install the CloudWatch monitoring agent on your AMI, Update your Auto Scaling policy to enable automated CloudWatch Log copy.

Commented [LC246]: Ans is C:

Ephemeral storage is the volatile temporary storage attached to your instances which is only present during the running lifetime of the instance. There is no option that you can take a snapshot of ephemeral storage on terminate.

Question #329

You've been recruited as the new chief operating officer of a SaaS startup. Your CTO has requested that you make troubleshooting any aspect of your business as easy and quick as feasible. She laments the fact that she has no clue what is happening in the complicated, service-oriented design, since the developers just log to disk, and it's very difficult to spot issues in logs for so many services.

How can you most effectively accomplish this demand while still satisfying your CTO?

- A. Copy all log files into AWS S3 using a cron job on each instance. Use an S3 Notification Configuration on the `<code>PutBucket</code>`
- B. Begin using CloudWatch Logs on every service. Stream all Log Groups into S3 objects. Use AWS EMR cluster jobs to perform ad-hoc MapReduce analysis and write new queries when needed.
- C. Copy all log files into AWS S3 using a cron job on each instance. Use an S3 Notification Configuration on the `<code>PutBucket</code>`
- D. Begin using CloudWatch Logs on every service. Stream all Log Groups into an AWS Elasticsearch Service Domain running Kibana 4 and perform log analysis on a search cluster.

Commented [LC247]: The Elasticsearch and Kibana 4 combination is called the ELK Stack, and is designed specifically for real-time, ad-hoc log analysis and aggregation. All other answers introduce extra delay or require pre-defined queries. Amazon Elasticsearch Service is a managed service that makes it easy to deploy, operate, and scale Elasticsearch in the AWS Cloud. Elasticsearch is a popular open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and click stream analytics.

Reference:
<https://aws.amazon.com/elasticsearch-service/>

Commented [LC248]: Snapshots that are taken from encrypted volumes are automatically encrypted. Volumes that are created from encrypted snapshots are also automatically encrypted. Your encrypted volumes and any associated snapshots always remain protected. For more information, see Amazon EBS Encryption.

Reference:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html>

Question #330

What is accurate about the way EBS encryption works?

- A. Snapshotting an encrypted volume makes an encrypted snapshot; restoring an encrypted snapshot creates an encrypted volume when specified / requested.
- B. Snapshotting an encrypted volume makes an encrypted snapshot when specified / requested; restoring an encrypted snapshot creates an encrypted volume when specified / requested.
- C. Snapshotting an encrypted volume makes an encrypted snapshot; restoring an encrypted snapshot always creates an encrypted volume.
- D. Snapshotting an encrypted volume makes an encrypted snapshot when specified / requested; restoring an encrypted snapshot always creates an encrypted volume.

Question #331

A business has developed a web application using AWS Elastic Beanstalk, Amazon S3, and Amazon DynamoDB. The online application's popularity has skyrocketed, resulting in unforeseen surges in traffic. According to a DevOps Engineer, 90% of the requests are duplicate read requests to the DynamoDB table and the S3 bucket images.

How can the Engineer optimize the website's performance?

- A. Use Amazon ElastiCache for Redis to cache repeated read requests to DynamoDB and AWS Elemental MediaStore to cache images stored in S3.
- B. Use Amazon ElastiCache for Memcached to cache repeated read requests to DynamoDB and Amazon EFS to cache images stored in S3.
- C. Use DynamoDB Accelerator to cache repeated read requests to DynamoDB and Amazon CloudFront to cache images stored in S3.
- D. Use DynamoDB Streams to cache repeated read requests to DynamoDB and API Gateway to cache images stored in S3.

Commented [LC249]: Correct Answer: C

Reference:
<https://aws.amazon.com/blogs/aws/amazon-dynamodb-accelerator-dax-in-memory-caching-for-read-intensive-workloads/>
<https://aws.amazon.com/dynamodb/dax/>

Question #332

A business uses AWS to host a legacy application. At any one moment, the application can execute on just one Amazon EC2 instance. Metadata about the application is stored in Amazon S3 and must be obtained before the instance can be restarted. If performance declines, the instance should be automatically restarted or relaunched.

Which solution will meet these criteria?

- A. Create an Amazon CloudWatch alarm to monitor the EC2 instance. When the StatusCheckFailed system alarm is triggered, use the recover action to stop and start the instance. Use a trigger in Amazon S3 to push the metadata to the instance when it is back up and running.
- **B. Use the auto healing feature in AWS OpsWorks to stop and start the EC2 instance. Use a lifecycle event in OpsWorks to pull the data from Amazon S3 and update it on the instance.**
- C. Use the Auto Recovery feature in Amazon EC2 to automatically stop and start the EC2 instance in case of a failure. Use a trigger in Amazon S3 to push the metadata to the instance when it is back up and running.
- D. Use AWS CloudFormation to create an EC2 instance that includes the user-data property for the EC2 resource. Add a command in user-data to retrieve the application metadata from Amazon S3.

Commented [LC250]: I'll go with B

A) and C) are wrong because there is no such thing like: "Use a trigger in Amazon S3 to push the metadata to the instance when it is back up and running"

There is no information about updating or putting a new metadata file to S3, so you can't create an event if nothing happens to the bucket.
Also there is no way to push from s3 to ec2 instance, that's not the case

D) is incomplete

Question #333

A business has implemented a hybrid architecture in which certain legacy systems stay on-premises while a particular cluster of servers is migrated to AWS. Due to the inability of the organization to reconfigure outdated systems, the cluster nodes must have a set hostname and local IP address for each server in the cluster.

The DevOps Engineer is responsible for automating the setup of a six-node cluster with high availability across three Availability Zones (AZs), by putting two elastic network interfaces in a subnet appropriate to each AZ. Between reboots or instance failures, the hostname and local IP address of each node should stay consistent.

Which option automates this activity with the LEAST amount of effort?

- A. Create an AWS Elastic Beanstalk application and a specific environment for each server of the cluster. For each environment, give the hostname, elastic network interface, and AZ as input parameters. Use the local health agent to name the instance and attach a specific elastic network interface based on the current environment.
- **B. Create a reusable AWS CloudFormation template to manage an Amazon EC2 Auto Scaling group with a minimum size of 1 and a maximum size of 1. Give the hostname, elastic network interface, and AZ as stack parameters. Use those parameters to set up an EC2 instance with EC2 Auto Scaling and a user data script to attach to the specific elastic network interface. Use CloudFormation nested stacks to nest the template six times for a total of six nodes needed for the cluster, and deploy using the master template.**
- C. Create an Amazon DynamoDB table with the list of hostnames, subnets, and elastic network interfaces to be used. Create a single AWS CloudFormation template to manage an Auto Scaling group with a minimum size of 6 and a maximum size of 6. Create a programmatic solution that is installed in each instance that will lock/release the assignment of each hostname and local IP address, depending on the subnet in which a new instance will be launched.
- D. Create a reusable AWS CLI script to launch each instance individually, which will name the instance, place it in a specific AZ, and attach a specific elastic network interface. Monitor the instances, and in the event of failure, replace the missing instance manually by running the script again.

Commented [LC251]: B. Create a reusable AWS CloudFormation template to manage an Amazon EC2 Auto Scaling group with a minimum size of 1 and a maximum size of 1. Give the hostname, elastic network interface, and AZ as stack parameters. Use those parameters to set up an EC2 instance with EC2 Auto Scaling and a user data script to attach to the specific elastic network interface. Use CloudFormation nested stacks to nest the template six times for a total of six nodes needed for the cluster, and deploy using the master template.

Reference:

<https://aws.amazon.com/blogs/devops/use-nested-stacks-to-create-reusable-templates-and-support-role-specialization/>

Question #334

A business requires the development of capture logs for any activity occurring inside its AWS account. Multiple VPCs are setup in the account, each with Amazon EC2 instances, Application Load Balancers, Amazon RDS MySQL databases, and AWS WAF rules. The logs must be safeguarded against deletion. It is necessary to do a daily visual examination of log anomalies from the previous day.

Which activities should a DevOps Engineer do in conjunction to achieve this? (Select three.)

- A. Configure an AWS Lambda function to send all CloudWatch logs to an Amazon S3 bucket. Create a dashboard report in Amazon QuickSight.
- B. Configure AWS CloudTrail to send all logs to Amazon Inspector. Create a dashboard report in Amazon QuickSight.
- C. Configure Amazon S3 MFA Delete on the logging Amazon S3 bucket.
- D. Configure an Amazon S3 object lock legal hold on the logging Amazon S3 bucket.
- E. Configure AWS Artifact to send all logs to the logging Amazon S3 bucket. Create a dashboard report in Amazon QuickSight.
- F. Deploy an Amazon CloudWatch agent to all Amazon EC2 instances.

Commented [LC252]:

Commented [LC253]:

Commented [LC254]:

Question #335

On startup, your system automatically assigns EIPs to EC2 instances in a VPC. The technology creates the whole VPC and stack in one go. Each VPC has two of them. Your effort to build a Development environment on your new AWS account failed, despite the fact that you successfully created Staging and Production environments in the same region.

What transpired?

- A. You didn't choose the Development version of the AMI you are using.
- B. You didn't set the Development flag to true when deploying EC2 instances.
- C. You hit the soft limit of 5 EIPs per region and requested a 6th.
- D. You hit the soft limit of 2 VPCs per region and requested a 3rd.

Commented [LC255]:

Question #336

A DevOps Engineer is responsible for enhancing the monitoring of a Finance team's payments microservice, which processes transactions for an e-commerce platform. Multiple Amazon EC2 instances are used to execute the microservice. Finance would want to know the number of payments made each minute and to be informed when this statistic goes below a certain threshold.

How can this be automated cost-effectively?

- A. Have the Development team log successful transactions to an application log. Set up Logstash on each instance, which sends logs to an Amazon ES cluster. Create a Kibana dashboard for the Finance team that graphs the metric.
- B. Have the Development team post the number of successful transactions to Amazon CloudWatch as a custom metric. Create a CloudWatch alarm when the threshold is breached, and use Amazon SNS to notify the Finance team.
- C. Have the Development team log successful transactions to an application log. On each instance, set up the Amazon CloudWatch Logs agent to send application logs to CloudWatch Logs. Use an EC2 instance to monitor a metric filter, and send notifications to the Finance team.
- D. Have the Development team log successful transactions to an application log. Set up the Amazon CloudWatch agent on each instance. Create a CloudWatch alarm when the threshold is breached, and use Amazon SNS to notify the Finance team.

Commented [LC256]:

Question #337

A firm is utilizing AWS CloudFormation to construct the infrastructure for a web application. The database engineering team manages database resources using a CloudFormation template, while the software development team manages web application resources through a different CloudFormation template. As the application's scope expands, the software development team must use database engineering resources. Both teams, however, desire to maintain their own review and lifecycle management methods. Additionally, both teams need approval of resource-level change sets. The software development team want to use its CI/CD pipeline to deliver updates to this template.

Which solution will satisfy these criteria?

- A. Create a stack export from the database CloudFormation template and import those references into the web application CloudFormation template.
- B. Create a CloudFormation nested stack to make cross-stack resource references and parameters available in both stacks.
- C. Create a CloudFormation stack set to make cross-stack resource references and parameters available in both stacks.
- D. Create input parameters in the web application CloudFormation template and pass resource names and IDs from the database stack.

Question #338

On AWS, a business has a mission-critical application that utilizes automated scaling. The organization desires that the deployment lifecycle adhere to the following criteria:

* The application must be deployed in batches to guarantee that the remaining fleet can continue to serve traffic.

* The program is CPU heavy and must be continuously monitored.

* If the deployment instance's CPU consumption surpasses 85 percent, the deployment must immediately roll back.

Which solution will satisfy these criteria?

- A. Use AWS CloudFormation to create an AWS Step Functions state machine and Auto Scaling lifecycle hooks to move to one instance at a time into a wait state. Use AWS Systems Manager automation to deploy the update to each instance and move it back into the Auto Scaling group using the heartbeat timeout.
- B. Use AWS CodeDeploy with Amazon EC2 Auto Scaling. Configure an alarm tied to the CPU utilization metric. Use the CodeDeployDefault.OneAtATime configuration as a deployment strategy. Configure automatic rollbacks within the deployment group to roll back the deployment if the alarm thresholds are breached.
- C. Use AWS Elastic Beanstalk for load balancing and AWS Auto Scaling. Configure an alarm tied to the CPU utilization metric. Configure rolling deployments with a fixed batch size of one instance. Enable enhanced health to monitor the status of the deployment and roll back based on the alarm previously created.
- D. Use AWS Systems Manager to perform a blue/green deployment with Amazon EC2 Auto Scaling. Configure an alarm tied to the CPU utilization metric. Deploy updates one at a time. Configure automatic rollbacks within the Auto Scaling group to roll back the deployment if the alarm thresholds are breached.

Commented [LC257]: B

<https://aws.amazon.com/about-aws/whats-new/2016/09/aws-codedeploy-introduces-deployment-monitoring-with-amazon-cloudwatch-alarms-and-automatic-deployment-rollback/>

Apparently, Beanstalk cannot roll-back when a certain metric is hit (CPU utilization).

Question #339

There are many methods to acquire computing power on Amazon Web Services.

Which, on average, ranks the price per compute or memory unit from LOW to HIGH (least costly to most expensive)?

- A. On-Demand B. Spot C. Reserved
- B. A, B, C
- C. C, B, A
- D. B, C, A
- E. A, C, B

Commented [LC258]: It's D

Spot cheaper than Reserved cheaper than OnDemand
<https://blog.boltops.com/2018/07/13/on-demand-vs-reserved-vs-spot-aws-ec2-pricing-comparison>

Question #340

A DevOps Engineer is tasked with developing a plan for updating a web application with little downtime. AWS CloudFormation is used to design the application's infrastructure, which consists of an Amazon Route 53 record, an Application Load Balancer, Amazon EC2 instances in an EC2 Auto Scaling group, and Amazon DynamoDB tables. To prevent downtime, the application must always be served by an active instance.

Which techniques will guarantee that the deployment occurs without a hitch? (Select two.)

- A. In the CloudFormation template, modify the `AWS::AutoScaling::AutoScalingGroup` resource and add an `UpdatePolicy` attribute to define the required elements for a deployment with zero downtime.
- B. In the CloudFormation template, modify the `AWS::AutoScaling::DeploymentUpdates` resource and add an `UpdatePolicy` attribute to define the required elements for a deployment with zero downtime.
- C. Add a new Application Load Balancer and Auto Scaling group to the CloudFormation template. Deploy new changes to the inactive Auto Scaling group. Use Route 53 to change the active Application Load Balancer.
- D. Add a new Application Load Balancer and Auto Scaling group to the CloudFormation template. Modify the `AWS::AutoScaling::AutoScalingGroup` resource and add an `UpdatePolicy` attribute to perform rolling updates.
- E. In the CloudFormation template, modify the `UpdatePolicy` attribute for the CloudFormation stack and specify the Auto Scaling group that will be updated. Configure `MinSuccessfulInstancesPercent` and `PauseTime` to ensure the deployment happens with zero downtime.

Commented [LC259]: I'll go with A, C
Rolling updates + blue green deployments

Commented [LC260]:

Question #341

Which of the following is not a constraint for AWS EBS Snapshots?

- A. Snapshots which are shared cannot be used as a basis for other snapshots.
- B. You cannot share a snapshot containing an AWS Access Key ID or AWS Secret Access Key.
- C. You cannot share unencrypted snapshots.
- D. Snapshot restorations are restricted to the region in which the snapshots are created.

Commented [LC261]: Snapshots shared with other users are usable in full by the recipient, including but limited to the ability to base modified volumes and snapshots.

Reference:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modifying-snapshotpermissions.html>

Question #342

A DevOps Engineer is installing an Amazon API Gateway API with backend functionality provided by an AWS Lambda function. The Engineer must keep track of the source IP address and status of each API request.

Which activities should the DevOps Engineer perform in conjunction to provide this functionality? (Select three.)

- A. Configure AWS X-Ray to enable access logging for the API Gateway requests.
- B. Configure the API Gateway stage to enable access logging and choose a logging format.
- C. Create a new Amazon CloudWatch Logs log group or choose an existing log group to store the logs.
- D. Grant API Gateway permission to read and write logs to Amazon CloudWatch through an IAM role.
- E. Create a new Amazon S3 bucket or choose an existing S3 bucket to store the logs.
- F. Configure API Gateway to stream its log data to Amazon Kinesis.

Commented [LC262]: BCD is correct.
<https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-logging.html>

F works but then permissions from option D don't apply.

Commented [LC263]:

Commented [LC264]:

Question #343

A code release procedure has been created using an AWS CodePipeline pipeline. The pipeline is connected with AWS CodeDeploy, which allows for the deployment of various versions of an application to numerous Amazon EC2 instances at each step of the CodePipeline.

The pipeline failed during a recent deployment owing to a CodeDeploy bug. The DevOps team want to enhance monitoring and alerting throughout the deployment process in order to reduce resolution times.

What should the DevOps Engineer do to automate the process of notifying users when concerns are discovered?

- A. Implement AWS CloudWatch Logs for CodePipeline and CodeDeploy, create an AWS Config rule to evaluate code deployment issues, and create an Amazon SNS topic to notify stakeholders of deployment issues.
- B. Implement AWS CloudWatch Events for CodePipeline and CodeDeploy, create an AWS Lambda function to evaluate code deployment issues, and create an Amazon SNS topic to notify stakeholders of deployment issues.
- C. Implement AWS CloudTrail to record CodePipeline and CodeDeploy API call information, create an AWS Lambda function to evaluate code deployment issues, and create an Amazon SNS topic to notify stakeholders of deployment issues.
- D. Implement AWS CloudWatch Events for CodePipeline and CodeDeploy, create an Amazon Inspector assessment target to evaluate code deployment issues, and create an Amazon SNS topic to notify stakeholders of deployment issues.

Commented [LC265]:

Question #344

A mobile application operating on eight Amazon EC2 instances makes use of an API endpoint provided by a third-party. Due to restricted capacity, the third-party service has a significant failure rate, which is anticipated to be remedied in a few weeks.

Meanwhile, the creators of mobile applications have included a retry option and are recording unsuccessful API queries. A DevOps Engineer must automate the monitoring of application logs and count the number of particular error messages; if the system detects more than ten faults during a one-minute period, an alert must be generated.

How can the criteria be accomplished with the SMALLEST amount of management overhead possible?

- A. Install the Amazon CloudWatch Logs agent on all instances to push the application logs to CloudWatch Logs. Use metric filters to count the error messages every minute, and trigger a CloudWatch alarm if the count exceeds 10 errors.
- B. Install the Amazon CloudWatch Logs agent on all instances to push the access logs to CloudWatch Logs. Create a CloudWatch Events rule to count the error messages every minute, and trigger a CloudWatch alarm if the count exceeds 10 errors.
- C. Install the Amazon CloudWatch Logs agent on all instances to push the application logs to CloudWatch Logs. Use a metric filter to generate a custom CloudWatch metric that records the number of failures and triggers a CloudWatch alarm if the custom metric reaches 10 errors in a 1-minute period.
- D. Deploy a custom script on all instances to check application logs regularly in a cron job. Count the number of error messages every minute, and push a data point to a custom CloudWatch metric. Trigger a CloudWatch alarm if the custom metric reaches 10 errors in a 1-minute period.

Commented [LC266]: C is correct, you can count with metric filter alright, but you need to create a custom metric from it to be able to fire alarms, which also gives you the ability to define data point and evaluation period.

Question #345

You work for an insurance business and are in charge of the day-to-day operations of the firm's online quotation system, which is used to deliver insurance quotes to members of the general public. Your organization wishes to use the application logs created by the system in order to get a better understanding of client behavior. Additionally, industry and regulatory requirements necessitate that you preserve all application logs for the system permanently in order to investigate potential fraudulent claims. You've been entrusted with the responsibility of developing a log management system that meets the following requirements:

- The system must maintain all log entries, even in the event of an unforeseen instance failure.
- The consumer insight team wants rapid access to seven-day records.
- The fraud investigation team needs access to all historical logs, however these logs will not be accessible for up to 24 hours.

How would you go about meeting these needs in the most cost-effective way possible? (Select three.)

- A. Configure your application to write logs to the instance's ephemeral disk, because this storage is free and has good write performance. Create a script that moves the logs from the instance to Amazon S3 once an hour.
- B. Write a script that is configured to be executed when the instance is stopped or terminated and that will upload any remaining logs on the instance to Amazon S3.
- C. Create an Amazon S3 lifecycle configuration to move log files from Amazon S3 to Amazon Glacier after seven days.
- D. Configure your application to write logs to the instance's default Amazon EBS boot volume, because this storage already exists. Create a script that moves the logs from the instance to Amazon S3 once an hour.
- E. Configure your application to write logs to a separate Amazon EBS volume with the "delete on termination" field set to false. Create a script that moves the logs from the instance to Amazon S3 once an hour.
- F. Create a housekeeping script that runs on a T2 micro instance managed by an Auto Scaling group for high availability. The script uses the AWS API to identify any unattached Amazon EBS volumes containing log files. Your housekeeping script will mount the Amazon EBS volume, upload all logs to Amazon S3, and then delete the volume.

Commented [LC267]: 2nd requisite

Commented [LC268]: 1st requisite

Commented [LC269]: 3rd requisite

Question #346

A DevOps Engineer is creating a canary testing technique for an AWS application. Recently, the program was changed and subjected to security, unit, and functional testing. The application must be deployed as a member of an AutoScaling group and use a Classic Load Balancer.

Which architecture satisfies the canary testing requirement?

- A. Create a different Classic Load Balancer and Auto Scaling group for blue/green environments. Use Amazon Route 53 and create weighted A records on Classic Load Balancer.
- B. Create a single Classic Load Balancer and an Auto Scaling group for blue/green environments. Use Amazon Route 53 and create A records for Classic Load Balancer IPs. Adjust traffic using A records.
- C. Create a single Classic Load Balancer and an Auto Scaling group for blue/green environments. Create an Amazon CloudFront distribution with the Classic Load Balancer as the origin. Adjust traffic using CloudFront.
- D. Create a different Classic Load Balancer and Auto Scaling group for blue/green environments. Create an Amazon API Gateway with a separate stage for the Classic Load Balancer. Adjust traffic by giving weights to this stage.

Commented [LC270]: Both A and D are very closer but in A, you'll see that weightage is based on "A record" but here route 53 will route traffic to alias of CLB. Hence, distributing traffic using "A record" is impossible. Using D, you'll be able to control using stage(two environments will considered here).

Question #347

How does the Amazon RDS paradigm with several Availability Zones work?

- A. A second, standby database is deployed and maintained in a different availability zone from master, using synchronous replication.
- B. A second, standby database is deployed and maintained in a different availability zone from master using asynchronous replication.
- C. A second, standby database is deployed and maintained in a different region from master using asynchronous replication.
- D. A second, standby database is deployed and maintained in a different region from master using synchronous replication.

Commented [LC271]: In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone.

Reference:
<http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

Question #348

Access logs for a vintage web application are stored in a proprietary text format. One of the security needs is to look for application access events and correlate them with data from a variety of other systems. These queries should be conducted in near-real time.

Which solution offloads the application server's processing burden and offers a method for near-real-time data search?

- A. Install the Amazon CloudWatch Logs agent on the application server and use CloudWatch Events rules to search logs for access events. Use Amazon CloudSearch as an interface to search for events.
- B. Use the third-party file-input plugin Logstash to monitor the application log file, then use a custom dissect filter on the agent to parse the log entries into the JSON format. Output the events to Amazon ES to be searched. Use the Elasticsearch API for querying the data.
- C. Upload the log files to Amazon S3 by using the S3 sync command. Use Amazon Athena to define the structure of the data as a table, with Athena SQL queries to search for access events.
- **D. Install the Amazon Kinesis Agent on the application server, configure it to monitor the log files, and send it to a Kinesis stream. Configure Kinesis to transform the data by using an AWS Lambda function, and forward events to Amazon ES for analysis. Use the Elasticsearch API for querying the data.**

Commented [LC272]: <https://docs.aws.amazon.com/streams/latest/dev/writing-with-agents.html>

Question #349

When using Amazon CloudTrail to record API calls, the following information is returned for services with a single end point: ____.

- A. captured and processed in the same region as to which the API call is made and delivered to the region associated with your Amazon S3 bucket
- B. captured, processed, and delivered to the region associated with your Amazon S3 bucket
- C. captured in the same region as to which the API call is made and processed and delivered to the region associated with your Amazon S3 bucket
- **D. captured in the region where the end point is located, processed in the region where the CloudTrail trail is configured, and delivered to the region associated with your Amazon S3 bucket**

Commented [LC273]: When logging with Amazon CloudTrail, API call information for services with regional end points (EC2, RDS etc.) is captured and processed in the same region as to which the API call is made and delivered to the region associated with your Amazon S3 bucket. API call information for services with single end points (IAM, STS etc.) is captured in the region where the end point is located, processed in the region where the CloudTrail trail is configured, and delivered to the region associated with your Amazon S3 bucket.

Reference:
<https://aws.amazon.com/cloudtrail/faqs/>

Questions 350-399

Question #350

A root user formed an IAM group and specified the following policy:

```
"Statement": [
{
  "Effect": "Allow",
  "Action": ["iam:ChangePassword"],
  "Resource": ["arn:aws:iam::123123123:user:${aws:username}"] },
{
  "Effect": "Allow",
  "Action": ["iam:GetAccountPasswordPolicy"],
  "Resource": ["*"]
}
```

What is the purpose of this policy?

- A. Allow this group to view the password policy of all the users added only to that group
- B. Allow all the users of IAM to modify their password
- C. Allow an IAM user in this group to view the password policy and modify only his/her password
- D. Allow this group to view the password policy of all the IAM users

Commented [LC274]: This IAM policy grants access to the `ChangePassword` action, which lets the users use the console, the CLI, or the API to change their passwords. The Resource element uses a policy variable (`aws:username`), which is useful in policies that are attached to groups. The `aws:username` key resolves to the name of the current IAM user when a request is made, so that each user is allowed permission to change only his or her own password. This policy will allow all the users of this group to modify the passwords of all the IAM users.

Reference:
<http://docs.aws.amazon.com/IAM/latest/UserGuide/HowToPwDIAMUser.html>

Question #351

Currently, your application is operating on Amazon EC2 instances protected by a load balancer. Your management has chosen a Blue/Green approach for deployment.

How should this be implemented on a per-deployment basis?

- A. Set up Amazon Route 53 health checks to fail over from any Amazon EC2 instance that is currently being deployed to.
- B. Using AWS CloudFormation, create a test stack for validating the code, and then deploy the code to each production Amazon EC2 instance.
- C. Create a new load balancer with new Amazon EC2 instances, carry out the deployment, and then switch DNS over to the new load balancer using Amazon Route 53 after testing.
- D. Launch more Amazon EC2 instances to ensure high availability, de-register each Amazon EC2 instance from the load balancer, upgrade it, and test it, and then register it again with the load balancer.

Commented [LC275]: Perhaps old question.

alb/nlb supports blue/green deployment natively

Question #352

The Development team has expanded significantly over the last several months, as has the number of projects that use distinct code repositories. The present procedure is manually setting AWS CodePipeline. There have been notifications about the amount of Amazon S3 buckets.

Which pipeline configuration will result in a reduction in S3 bucket sprawl alerts?

- A. Combine the multiple separate code repositories into a single one, and deploy using an AWS CodePipeline that has logic for each project.
- B. Create new pipelines by using the AWS API or AWS CLI, and configure them to use a single S3 bucket with separate prefixes for each project.
- C. Create a new pipeline in a different region for each project to bypass the service limits for S3 buckets in a single region.
- D. Create a new pipeline and S3 bucket for each project by using the AWS API or AWS CLI to bypass the service limits for S3 buckets in a single account.

Commented [LC276]:

Question #353

A development team creates an artifact manually on-premises and then uploads it to an Amazon S3 bucket. When the program is deployed, it contains a local cache that must be purged. To do this, the team runs a script that downloads the artifact from Amazon S3 and unzips it to finish the deployment.

A DevOps team want to switch to a CI/CD process and include checks to halt and roll back deployments in the event of a failure. This necessitates the team monitor the deployment's progress.

Which action(s) will do this? (Select three.)

- A. Allow developers to check the code into a code repository. Using Amazon CloudWatch Events, on every pull into master, trigger an AWS Lambda function to build the artifact and store it in Amazon S3.
- **B. Create a custom script to clear the cache. Specify the script in the BeforeInstall lifecycle hook in the AppSpec file.**
- C. Create user data for each Amazon EC2 instance that contains the clear cache script. Once deployed, test the application. If it is not successful, deploy it again.
- **D. Set up AWS CodePipeline to deploy the application. Allow developers to check the code into a code repository as a source for the pipeline.**
- **E. Use AWS CodeBuild to build the artifact and place it in Amazon S3. Use AWS CodeDeploy to deploy the artifact to Amazon EC2 instances.**
- F. Use AWS Systems Manager to fetch the artifact from Amazon S3 and deploy it to all the instances.

Commented [LC277]: I'll go with B, D, E

B. Create a custom script to clear the cache. Specify the script in the BeforeInstall lifecycle hook in the AppSpec file.
D. Set up AWS CodePipeline to deploy the application. Allow developers to check the code into a code repository as a source for the pipeline.
E. Use AWS CodeBuild to build the artifact and place it in Amazon S3. Use AWS CodeDeploy to deploy the artifact to Amazon EC2 instances.

Question #354

You've been tasked with the responsibility of migrating a huge amount of data from numerous Amazon RDS MySQL instances to a DynamoDB table. You have been given a limited time frame in which to finish the data transfer.

What will enable you to successfully finish this time-consuming data processing workflow?

- A. Create an Amazon Kinesis data stream, pipe in all of the Amazon RDS data, and direct the data toward a DynamoDB table.
- B. Write a script in your language of choice, install the script on an Amazon EC2 instance, and then use Auto Scaling groups to ensure that the latency of the migration pipelines never exceeds four seconds in any 15-minute period.
- C. Write a bash script to run on your Amazon RDS instance that will export data into DynamoDB.
- **D. Create a data pipeline to export Amazon RDS data and import the data into DynamoDB.**

Commented [LC280]:

Question #355

A corporation has 100 GB of log data in .csv format stored in an Amazon S3 bucket. SQL developers want to query and display this data using graphs. Additionally, they need an effective, automated method for storing information from the.csv file.

Which combination of measures should be followed to ensure that these criteria are met with the LEAST amount of work possible? (Select three.)

- A. Filter the data through AWS X-Ray to visualize the data.
- **B. Filter the data through Amazon QuickSight to visualize the data.**
- **C. Query the data with Amazon Athena.**
- D. Query the data with Amazon Redshift.
- **E. Use AWS Glue as the persistent metadata store.**
- F. Use Amazon S3 as the persistent metadata store.

Commented [LC281]:

Commented [LC282]:

Commented [LC283]: B,C,E as S3 is already available. If we think from metadata store perspective it is glue catalog
<https://docs.aws.amazon.com/glue/latest/dg/components-overview.html>

Question #356

What is the highest single-volume throughput supported by EBS?

- **A. 320MiB/s**
- B. 160MiB/s
- C. 40MiB/s
- D. 640MiB/s

Commented [LC284]: Max Throughput per Volume
SSD-Backed General Purpose (gp2)* 160 MiB/s
SSD-Backed Provisioned IOPS (io1) 320 MiB/s
HDD-Backed Throughput Optimized (st1) 500 MiB/s
HDD-Backed Cold (sc1) 250 MiB/s

Answer A is max in the available options

<https://docs.aws.amazon.com/whitepapers/latest/aws-storage-services-overview/performance-3.html>

Question #357

AWS is used by an ecommerce firm to host an application. The organization want to implement a standby disaster recovery solution in an extra Region that retains the application's present code. The application is hosted on Amazon EC2 instances and is protected by a load balancer (ALB). The instances are distributed across several Availability Zones through an EC2 Auto Scaling group. The database layer is hosted on an Amazon RDS MySQL Multi-AZ DB instance through Amazon RDS. DNS records for Amazon Route 53 link to the ALB.

Which combination of activities will achieve these objectives at the LOWEST possible cost? (Select three.)

- A. Configure a failover routing policy for the application DNS entry.
- B. Configure a geolocation routing policy for the application DNS entry.
- C. Create a cross-Region RDS read replica in the new standby Region.
- D. Migrate the database layer to Amazon DynamoDB and enable global replication to the new standby Region.
- E. Provision the ALB and Auto Scaling group in the new standby Region and set the desired capacity to match the active Region.
- F. Provision the ALB and Auto Scaling group in the new standby Region and set the desired capacity to 1.

Commented [LC285]:

Commented [LC286]:

Commented [LC287]: E is more expensive than F, keeping the same number of servers is unnecessary, you can increase the number during the failover

Question #358

A DevOps Engineer just joined a new organization that is already using Amazon EC2 instances to handle workloads. AWS has been gradually embraced without any central governance. The Engineer must now determine the extent to which previous deployments meet the following requirements:

- ☒ Only authorized AMIs are executed on EC2 instances.
- ☒ Amazon EBS volumes are secured.
- ☒ EC2 instances have an Owner tag.
- ☒ On EC2 instances, root login through SSH is blocked.

Which services should the Engineer use in order to complete this evaluation with the LEAST amount of work possible? (Select two.)

- A. AWS Config
- B. Amazon GuardDuty
- C. AWS System Manager
- D. AWS Directory Service
- E. Amazon Inspector

Commented [LC288]: Ans is AE: A is for sure and for E as Last Option stats "Root login over SSH is disabled on EC2 instances." we use AWS Inspector https://docs.aws.amazon.com/inspector/latest/userguide/inspector_security-best-practices.html#disable-root-login-over-SSH

Question #359

A business is implementing a new mobile game on AWS for its global clients. The development team makes use of AWS Code services and is required to adhere to the following guidelines:

- Clients must often send/receive real-time playing data from the backend with a minimum of delay.
- Game data must adhere to the criterion for data residency.

Which approach may a DevOps Engineer use to address their requirements?

- A. Deploy the backend application to multiple regions. Any update to the code repository triggers a two-stage build and deployment pipeline. A successful deployment in one region invokes an AWS Lambda function to copy the build artifacts to an Amazon S3 bucket in another region. After the artifact is copied, it triggers a deployment pipeline in the new region.
- B. Deploy the backend application to multiple Availability Zones in a single region. Create an Amazon CloudFront distribution to serve the application backend to global customers. Any update to the code repository triggers a two-stage build-and-deployment pipeline. The pipeline deploys the backend application to all Availability Zones.
- C. Deploy the backend application to multiple regions. Use AWS Direct Connect to serve the application backend to global customers. Any update to the code repository triggers a two-stage build-and-deployment pipeline in the region. After a successful deployment in the region, the pipeline continues to deploy the artifact to another region.
- D. Deploy the backend application to multiple regions. Any update to the code repository triggers a two-stage build-and-deployment pipeline in the region. After a successful deployment in the region, the pipeline invokes the pipeline in another region and passes the build artifact location. The pipeline uses the artifact location and deploys applications in the new region.

Commented [LC290]: Ill go with A B is wrong because refers multi-az...must be multi-region C is wrong because its not the use case of Direct connect and it would cost a fortune D is wrong because the pipeline can't invokes another pipeline.

Question #360

You manage operations for a firm that handles a large number of digital wallet payments. A single second of downtime during which you miss payments or are otherwise inaccessible costs you around USD 100. You balance the transaction system's financials once a day.

Which database configuration is optimal for mitigating this business risk?

- A. A multi-AZ RDS deployment with synchronous replication to multiple standbys and read-replicas for fast failover and ACID properties.
- B. A multi-region, multi-master, active-active RDS configuration using database-level ACID design principles with database trigger writes for replication.
- **C. A multi-region, multi-master, active-active DynamoDB configuration using application control-level BASE design principles with change-stream write queue buffers for replication.**
- D. A multi-AZ DynamoDB setup with changes streamed to S3 via AWS Kinesis, for highly durable storage and BASE properties.

Commented [LC291]: Only the multi-master, multi-region DynamoDB answer makes sense. Multi-AZ deployments do not provide sufficient availability when a business loses USD 360,000 per hour of unavailability. As RDS does not natively support multi-region, and ACID does not perform well/at all over large distances between regions, only the DynamoDB answer works.

Reference:

<http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.CrossRegionRepl.html>

Question #361

A DevOps Engineer is working on an Amazon Linux-hosted project that has failed a security evaluation. The DevOps Manager has been requested to analyze and provide suggestions on the company's buildspec.yaml file for an AWS CodeBuild project. The following is the configuration of the buildspec.yaml file:

```
env:
variables:
  AWS_ACCESS_KEY_ID: AKIAJF7BRFWJBA4GHXNA
  AWS_SECRET_ACCESS_KEY: ORjln3A2mlh4O4tm0+zHxZqz7cNAvMLYRehcl
  AWS_DEFAULT_REGION: us-east-1
  DB_PASSWORD: cuj5RpF3a
phases:
build:
commands:
  -aws s3 cp s3://db-deploy-bucket/my.cnf.template/tmp/my.cnf
  -sed-i '' s/DB_PW/${DB_PASSWORD}/ /tmp/my.cnf
  -aws s3 cp s3://db-deploy-bucket/instance.key/tmp/instance.key
  -chmod 600/tmp/instance.key
  -scp -i /tmp/instance.key/tmp/my.cnf root@10.25.15.23:/etc/my.cnf
  -ssh -i /tmp/instance.key root@10.25.15.23 /etc/init.d/mysqld restart
```

Which adjustments are necessary to ensure compliance with AWS security best practices? (Select three.)

- A. Add a post-build command to remove the temporary files from the container before termination to ensure they cannot be seen by other CodeBuild users.
- **B. Update the CodeBuild project role with the necessary permissions and then remove the AWS credentials from the environment variable.**
- **C. Store the DB_PASSWORD as a SecureString value in AWS Systems Manager Parameter Store and then remove the DB_PASSWORD from the environment variables.**
- D. Move the environment variables to the 'db-deploy-bucket' Amazon S3 bucket, add a prebuild stage to download, then export the variables.
- **E. Use AWS Systems Manager run command versus scp and ssh commands directly to the instance.**
- F. Scramble the environment variables using XOR followed by Base64, add a section to install, and then run XOR and Base64 to the build phase.

Commented [LC292]: B. Update the CodeBuild project role with the necessary permissions and then remove the AWS credentials from the environment variable.

C. Store the DB_PASSWORD as a SecureString value in AWS Systems Manager Parameter Store and then remove the DB_PASSWORD from the environment variables.

E. Use AWS Systems Manager run command versus scp and ssh commands directly to the instance. You can get rid of the instance.key stored in a s3 bucket (which is insecure)

Commented [LC293]:

Commented [LC294]:

Commented [LC295]: io1 volumes, or Provisioned IOPS (PIOPS) SSDs, are best for: Critical business applications that require sustained IOPS performance, or more than 10,000 IOPS or 160 MiB/s of throughput per volume, like large database workloads, such as MongoDB.

Reference:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBVolumeTypes.html>

Question #362

Which EBS volume type is optimal for deployments of high-performance NoSQL clusters?

- **A. io1**
- B. gp1
- C. standard
- D. gp2

Question #363

Company standards mandate the collection of information regarding IP traffic between instances in the production Amazon VPC. The capturing mechanism must be active at all times, and the Security team must be informed of any configuration modifications.

What should be done to guarantee compliance with these requirements?

- A. Using the UserData section of an AWS CloudFormation template, install tcpdump on every provisioned Amazon EC2 instance. The output of the tool is sent to Amazon EFS for aggregation and querying. In addition, scheduling an Amazon CloudWatch Events rule calls an AWS Lambda function to check whether tcpdump is up and running and sends an email to the security organization when there is an exception.
- B. Create a flow log for the production VPC and assign an Amazon S3 bucket as a destination for delivery. Using Amazon S3 Event Notification, set up an AWS Lambda function that is triggered when a new log file gets delivered. This Lambda function updates an entry in Amazon DynamoDB, which is periodically checked by scheduling an Amazon CloudWatch Events rule to notify security when logs have not arrived.
- C. Create a flow log for the production VPC. Create a new rule using AWS Config that is triggered by configuration changes of resources of type 'EC2:VPC'. As part of configuring the rule, create an AWS Lambda function that looks up flow logs for a given VPC. If the VPC flow logs are not configured, return a 'NON_COMPLIANT' status and notify the security organization.
- D. Configure a new trail using AWS CloudTrail service. Using the UserData section of an AWS CloudFormation template, install tcpdump on every provisioned Amazon EC2 instance. Connect Amazon Athena to the CloudTrail and write an AWS Lambda function that monitors for a flow log disable event. Once the CloudTrail entry has been spotted, alert the security organization.

Commented [LC296]: I go with C.

<https://aws.amazon.com/blogs/security/how-to-audit-your-aws-resources-for-security-compliance-by-using-custom-aws-config-rules/>

Question #364 [SKIP]

What is the primary difference between the command line options 'ansible' and 'ansible-playbook'?

- A. 'ansible' is for setting configuration and environment variables which 'ansible-playbook' will use when running plays.
- B. 'ansible-playbook' is for running entire Playbooks while 'ansible' is for calling ad-hoc commands.
- C. 'ansible-playbook' runs the playbooks by using the 'ansible' command to run the individual plays
- D. 'ansible' is for running individual plays and 'ansible-playbook' is for running the entire playbook.

Question #365

During an assessment run, the Amazon Inspector agent captures telemetry data and transfers it to an Amazon Inspector-specific S3 bucket for analysis.

How can you get access to telemetry data generated by Amazon Inspector and how can you use this data to help secure your resources?

- A. Telemetry data is kept in S3 and encrypted with a pre-assessment test key configured in KMS, as long as you have access to that key you can download and decrypt telemetry data.
- B. Telemetry data is stored in Amazon Inspector dedicated S3 bucket that does NOT belong to your account, Amazon Inspector currently does NOT provide an API or an S3 bucket access mechanism to collected telemetry. Data is retained temporarily only to allow for assistance with support requests.
- C. Telemetry data is saved on S3 bucket in your account, therefore telemetry data is accessible with proper permissions on that bucket.
- D. Telemetry data is deleted immediately after assessment run, therefore data can NOT be accessed or analyzed by any other tools.

Commented [LC297]: https://docs.aws.amazon.com/inspector/v1/userguide/inspector_agents.html

Question #366

A business is using an Amazon ECS cluster to handle its workload. Various ECS services will be operated on the cluster, with an Application Load Balancer acting as the front end, routing traffic using multiple target groups. The Application Development team has been having difficulty collecting logs that need to be gathered and transferred to an Amazon S3 bucket for near-real-time analysis.

What has to be configured in the deployment by the DevOps Engineer to satisfy these requirements? (Make three selections.)

- **A. Install the Amazon CloudWatch Logs logging agent on the ECS instances. Change the logging driver in the ECS task definition to 'awslogs'.**
- B. Download the Amazon CloudWatch Logs container instance from AWS and configure it as a task. Update the application service definitions to include the logging task.
- C. Use Amazon CloudWatch Events to schedule an AWS Lambda function that will run every 60 seconds running the create-export -task CloudWatch Logs command, then point the output to the logging S3 bucket.
- **D. Enable access logging on the Application Load Balancer, then point it directly to the S3 logging bucket.**
- E. Enable access logging on the target groups that are used by the ECS services, then point it directly to the S3 logging bucket.
- **F. Create an Amazon Kinesis Data Firehose with a destination of the S3 logging bucket, then create an Amazon CloudWatch Logs subscription filter for Kinesis.**

Commented [LC298]:

Commented [LC299]:

Commented [LC300]:

Question #367

A DevOps Engineer noticed a dramatic increase in the page load speeds of a website and determined that a recent deployment had happened. A quick diff of the associated commit reveals that the URL for an external API request was modified, as was the connection port from 80 to 443. The external API has been tested and validated to function independently of the application. According to the application logs, the connection has now timed out, resulting in several retries and final call failure.

Which debugging procedures should the Engineer take to ascertain the issue's primary cause?

- A. Check the VPC Flow Logs looking for denies originating from Amazon EC2 instances that are part of the web Auto Scaling group. Check the ingress security group rules and routing rules for the VPC.
- B. Check the existing egress security group rules and network ACLs for the VPC. Also check the application logs being written to Amazon CloudWatch Logs for debug information.
- **C. Check the egress security group rules and network ACLs for the VPC. Also check the VPC flow logs looking for accepts originating from the web Auto Scaling group.**
- D. Check the application logs being written to Amazon CloudWatch Logs for debug information. Check the ingress security group rules and routing rules for the VPC.

Commented [LC301]: A and D are wrong because it's not ingress that must be checked.

Here is another view between B&C. If you have developer background you rely on B. If you have a Network background you rely on C. Though B might indicate timeouts connecting to 443, its C that puts a nail to coffin to confirm issue is on originating end and not the destination end. My choice is C

Question #368

You've been directed by your Chief Information Security Officer (CISO) to deliver an audit report on all AWS network rules utilized by the organization's Amazon EC2 instances. You've noticed that a single Describe-Security-Groups API request returns all of the security groups and rules associated with an account inside a region. To construct the needed report, you write the following pseudo-code:

- Parse the output of "aws ec2 describe-security-groups"
- For each set of security personnel
- Generate a report detailing the entry and egress regulations

Which two pieces of extra logic should you incorporate to satisfy the CISO's requirements? (Select two.)

- **A. Parse security groups in each region.**
- B. Parse security groups in each Availability Zone and region.
- **C. Evaluate VPC network access control lists.**
- D. Evaluate AWS CloudTrail logs.
- E. Evaluate Elastic Load Balancing access control lists.
- F. Parse CloudFront access control lists.

Commented [LC302]:

Commented [LC303]:

Question #369

Due to compliance laws, management has requested that you offer a solution that enables cost-effective long-term storage of your application logs and enables support employees to more rapidly access the logs. Currently, your log system archives logs to Amazon S3 automatically every hour, and support personnel must wait for these logs to arrive in Amazon S3 due to the fact that they do not have access to the systems to examine live logs.

What approach should you employ to achieve compliance while also allowing support workers to view logs more quickly?

- A. Update Amazon S3 lifecycle policies to archive old logs to Amazon Glacier, and add a new policy to push all log entries to Amazon SQS for ingestion by the support team
- B. Update Amazon S3 lifecycle policies to archive old logs to Amazon Glacier, and use or write a service to also stream your application logs to CloudWatch Logs.
- C. Update Amazon Glacier lifecycle policies to pull new logs from Amazon S3, and in the Amazon EC2 console, enable the CloudWatch Logs Agent on all of your application servers.
- D. Update Amazon S3 lifecycle policies to archive old logs to Amazon Glacier. key can be different from the tableEnable Amazon S3 partial uploads on your Amazon S3 bucket, and trigger an Amazon SNS notification when a partial upload occurs.
- E. Use or write a service to stream your application logs to CloudWatch Logs. Use an Amazon Elastic Map Reduce cluster to live stream your logs from CloudWatch Logs for ingestion by the support team, and create a Hadoop job to push the logs to S3 in five-minute chunks.

Commented [LC304]:

Question #370

You have an Amazon EC2 application operating in an Auto Scaling group. Dynamically bootstrapped instances are created, and the process takes over 15 minutes to finish. You discover that Auto Scaling reports instances as "In Service" before bootstrapping is complete. You are getting application alerts for newly created instances prior to them successfully booting up, which is generating confusion. You discover the source of the problem: your application monitoring tool is probing the Auto Scaling Service API for In Service instances and raising alerts for previously unknown instances.

Which of the following will guarantee that no additional instances are introduced to your application monitoring tool prior to the completion of bootstrapping?

- A. Create an Auto Scaling group lifecycle hook to hold the instance in a pending: wait state until your bootstrapping is complete. Once bootstrapping is complete, notify Auto Scaling to complete the lifecycle hook and move the instance into a pending: complete state.
- B. Use the default Amazon CloudWatch application metrics to monitor your application's health. Configure an Amazon SNS topic to send these CloudWatch alarms to the correct recipients.
- C. Tag all instances on launch to identify that they are in a pending state. Change your application monitoring tool to look for this tag before adding new instances, and then use the Amazon API to set the instance state to 'pending' until bootstrapping is complete.
- D. Increase the desired number of instances in your Auto Scaling group configuration to reduce the time it takes to bootstrap future instances.

Commented [LC305]:

Question #371

Your organization often introduces new innovations while requiring high application availability. As part of the program's A/B testing, log files from each upgraded Amazon EC2 instance must be inspected in near real-time to confirm that the application continues to function perfectly after each deployment.

If the logs indicate unusual behavior, the instance's application version is changed to a more stable one.

Which of the following strategies should you employ to ensure high availability while delivering and analyzing logs?

- A. Ship the logs to Amazon S3 for durability and use Amazon EMR to analyze the logs in a batch manner each hour.
- B. Ship the logs to Amazon CloudWatch Logs and use Amazon EMR to analyze the logs in a batch manner each hour.
- C. Ship the logs to an Amazon Kinesis stream and have the consumers analyze the logs in a live manner.
- D. Ship the logs to a large Amazon EC2 instance and analyze the logs in a live manner.
- E. Store the logs locally on each instance and then have an Amazon Kinesis stream pull the logs for live analysis.

Commented [LC306]:

Question #372

A business wishes to implement a technique for responding to security issues posed by leaked or compromised IAM access keys. The DevOps Engineer has been tasked with automating the process of finding compromised access keys, cancelling their rights, and notifying the Security team.

Which of the following would accomplish this objective?

- A. Use the AWS Trusted Advisor generated security report for access keys. Use Amazon EMR to run analytics on the report. Identify compromised IAM access keys and delete them. Use Amazon CloudWatch with an EMR Cluster State Change event to notify the Security team.
- B. Use AWS Trusted Advisor to identify compromised access keys. Create an Amazon CloudWatch Events rule with Trusted Advisor as the event source, and AWS Lambda and Amazon SNS as targets. Use AWS Lambda to delete compromised IAM access keys and Amazon SNS to notify the Security team.
- C. Use the AWS Trusted Advisor generated security report for access keys. Use AWS Lambda to scan through the report. Use scan result inside AWS Lambda and delete compromised IAM access keys. Use Amazon SNS to notify the Security team.
- D. Use AWS Lambda with a third-party library to scan for compromised access keys. Use scan result inside AWS Lambda and delete compromised IAM access keys. Create Amazon CloudWatch custom metrics for compromised keys. Create a CloudWatch alarm on the metrics to notify the Security team.

Commented [LC307]: B is correct as Trusted Advisor generates events in case of compromised access keys.

<https://aws.amazon.com/premiumsupport/technology/trusted-advisor/best-practice-checklist/#Security>

Question #373

You're deploying your application using AWS Elastic Beanstalk and need to make data saved on an Amazon Elastic Block Store (EBS) volume snapshot accessible to Amazon Elastic Compute Cloud (EC2) instances. How do you adjust your Elastic Beanstalk setup such that data is automatically uploaded to the Amazon EC2 instances when your application is deployed?

- A. Add commands to a configuration file in the .ebextensions folder of your deployable archive that mount an additional Amazon EBS volume on launch. Also add a "BlockDeviceMappings" option, and specify the snapshot to use for the block device in the Auto Scaling launch configuration.
- B. Add commands to a configuration file in the .ebextensions folder of your deployable archive that uses the create-volume Amazon EC2 API or CLI to create a new ephemeral volume based on the specified snapshot and then mounts the volume on launch.
- C. Add commands to the Amazon EC2 user data that will be executed by eb-init, which uses the create-volume Amazon EC2 API or CLI to create a new Amazon EBS volume based on the specified snapshot, and then mounts the volume on launch.
- D. Add commands to the Chef recipe associated with your environment, use the create-volume Amazon EC2 API or CLI to create a new Amazon EBS volume based on the specified snapshot, and then mount the volume on launch.

Commented [LC308]: A is the correct answer

<https://aws.amazon.com/blogs/devops/customize-ephemeral-and-ebs-volumes-in-elastic-beanstalk-environments/>

Question #374

A multinational corporation with geographically dispersed development teams constructed a web application utilizing a microservices architecture and Amazon ECS. Each application service is self-contained and operates in the ECS cluster as a service. The container build files and source code are stored in a secure GitHub repository.

For development, testing, and production environments, separate ECS clusters exist.

Developers must submit features to GitHub branches and then merge them into an environment-specific branch (development, test, or production).

This merging should initiate an automated pipeline that will execute a build and deploy to the appropriate ECS cluster.

What automated solution should the DevOps Engineer propose for these requirements?

- A. Create an AWS CloudFormation stack for the ECS cluster and AWS CodePipeline services. Store the container build files in an Amazon S3 bucket. Use a post-commit hook to trigger a CloudFormation stack update that deploys the ECS cluster. Add a task in the ECS cluster to build and push images to Amazon ECR, based on the container build files in S3.
- B. Create a separate pipeline in AWS CodePipeline for each environment. Trigger each pipeline based on commits to the corresponding environment branch in GitHub. Add a build stage to launch AWS CodeBuild to create the container image from the build file and push it to Amazon ECR. Then add another stage to update the Amazon ECS task and service definitions in the appropriate cluster for that environment.
- C. Create a pipeline in AWS CodePipeline. Configure it to be triggered by commits to the master branch in GitHub. Add a stage to use the Git commit message to determine which environment the commit should be applied to, then call the create-image Amazon ECR command to build the image, passing it to the container build file. Then add a stage to update the ECS task and service definitions in the appropriate cluster for that environment.
- D. Create a new repository in AWS CodeCommit. Configure a scheduled project in AWS CodeBuild to synchronize the GitHub repository to the new CodeCommit repository. Create a separate pipeline for each environment triggered by changes to the CodeCommit repository. Add a stage using AWS Lambda to build the container image and push to Amazon ECR. Then add another stage to update the ECS task and service definitions in the appropriate cluster for that environment.

Commented [LC309]: <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-cd-pipeline.html>

Question #375

You are running a SIP-based phone application on Amazon EC2 using MySQL on Amazon RDS as the database. The program only maintains the authentication profile data for its current users in the database, making it a read-intensive application. Your monitoring system indicates that your web instances and database are using a significant amount of CPU.

Which of the following procedures should you take to assure your application's continued availability? (Select two.)

- A. Use a CloudFront RTMP download distribution with the application tier as the origin for the distribution.
- B. Set up an Auto Scaling group for the application tier and a policy that scales based on the Amazon EC2 CloudWatch CPU utilization metric.
- C. Vertically scale up the Amazon EC2 instances manually.
- D. Set up an Auto Scaling group for the application tier and a policy that scales based on the Amazon RDS CloudWatch CPU utilization metric.
- E. Switch to General Purpose (SSD) Storage from Provisioned IOPS Storage (PIOPS) for the Amazon RDS database.
- F. Use multiple Amazon RDS read replicas.

Commented [LC310]:

Commented [LC311]:

Question #376

A business operates an application with predicted high traffic periods. The business desires that application instances scale up only during peak periods. Amazon DynamoDB is used to store application state. The application environment is built on top of a conventional Node.js application stack with customized Chef recipes that are kept in a private Git repository.

Which option is the MOST cost-effective and involves the LEAST amount of administration overhead when doing rolling application updates?

- A. Create a custom AML with the Node.js environment and application stack using Chef recipes. Use the AML in an Auto Scaling group and set up scheduled scaling for the required times, then set up an Amazon EC2 IAM role that provides permission to access DynamoDB.
- B. Create a Docker file that uses the Chef recipes for the application environment based on an official Node.js Docker image. Create an Amazon ECS cluster and a service for the application environment, then create a task based on this Docker image. Use scheduled scaling to scale the containers at the appropriate times and attach a task-level IAM role that provides permission to access DynamoDB.
- C. Configure AWS OpsWorks stacks and use custom Chef cookbooks. Add the Git repository information where the custom recipes are stored, and add a layer in OpsWorks for the Node.js application server. Then configure the custom recipe to deploy the application in the deploy step. Configure time-based instances and attach an Amazon EC2 IAM role that provides permission to access DynamoDB.
- D. Configure AWS OpsWorks stacks and push the custom recipes to an Amazon S3 bucket and configure custom recipes to point to the S3 bucket. Then add an application layer type for a standard Node.js application server and configure the custom recipe to deploy the application in the deploy step from the S3 bucket. Configure time-based instances and attach an Amazon EC2 IAM role that provides permission to access DynamoDB.

Commented [LC312]: <https://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook-installingcustom-enable.html>

Question #377

A DevOps engineer is debugging deployments to a new application that is being developed on Amazon EC2 instances behind an Application Load Balancer. The instances are distributed across several Availability Zones through an EC2 Auto Scaling group. Occasionally, instances are brought online before they are ready, resulting in increasing mistake rates among users. The present setup provides a 60-second grace period and deems instances healthy after receiving two 200 response codes from /index.php, a page that may respond sporadically throughout the deployment process. The development team is eager to get instances live as quickly as feasible.

Which technique would be most effective in resolving this issue?

- A. Increase the instance grace period from 60 seconds to 180 seconds, and the consecutive health check requirement from 2 to 3.
- B. Increase the instance grace period from 60 second to 120 seconds, and change the response code requirement from 200 to 204.
- C. Modify the deployment script to create a /health-check.php file when the deployment begins, then modify the health check path to point to that file.
- D. Modify the deployment script to create a /health-check.php file when all tasks are complete, then modify the health check path to point to that file.

Commented [LC313]: "The development team wants instances to come online as soon as possible." It's definitely not A or B as that will increase the time for instances to be considered online.

The problem is that the index.php page responds intermittently during deployment, so the proper approach is NOT to rely on that page and use a proper health-check page instead (C or D).

I would go for D to ensure the check is OK when (and as soon as) everything works.

References:
<https://aws.amazon.com/builders-library/implementing-health-checks/>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html>

Commented [LC314]: Only CloudFormation's JSON Templates allow declarative version control of repeatably deployable models of entire AWS clouds.

Reference:
<https://blogs.aws.amazon.com/application-management/blog/category/Best+practices>

Question #378

You have been tasked with the responsibility of reducing the risk associated with your company's deployments. The CEO is particularly worried about outages caused by unintentional discrepancies between Staging and Production, which can result in unexpected behavior in Production even though Staging tests pass. You already utilize Docker to provide high consistency across Staging and Production environments for your EC2 instances' application environments.

How can you further mitigate risk in the remainder of the execution environment, given that AWS provides a plethora of service components in addition to EC2 virtual machines?

- A. Develop models of your entire cloud system in CloudFormation. Use this model in Staging and Production to achieve greater parity.
- B. Use AWS Config to force the Staging and Production stacks to have configuration parity. Any differences will be detected for you so you are aware of risks.
- C. Use AMIs to ensure the whole machine, including the kernel of the virtual machines, is consistent, since Docker uses Linux Container (LXC) technology, and we need to make sure the container environment is consistent.
- D. Use AWS ECS and Docker clustering. This will make sure that the AMIs and machine sizes are the same across both environments.

Question #379

A security team is worried that a developer may inadvertently associate an Elastic IP address with a production Amazon EC2 machine. No developer should be permitted to associate an instance with an Elastic IP address. At any point in time, the Security team must be alerted whenever a production server acquires an Elastic IP address.

How is this job automatable?

- A. Use Amazon Athena to query AWS CloudTrail logs to check for any associate-address attempts. Create an AWS Lambda function to disassociate the Elastic IP address from the instance, and alert the Security team.
- B. Attach an IAM policy to the Developers' IAM group to deny associate-address permissions. Create a custom AWS Config rule to check whether an Elastic IP address is associated with any instance tagged as production, and alert the Security team.
- C. Ensure that all IAM groups are associated with Developers do not have associate-address permissions. Create a scheduled AWS Lambda function to check whether an Elastic IP address is associated with any instance tagged as production, and alert the Security team if an instance has an Elastic IP address associated with it.
- D. Create an AWS Config rule to check that all production instances have EC2 IAM roles that include deny associate-address permissions. Verify whether there is an Elastic IP address associated with any instance, and alert the Security team if an instance has an Elastic IP address associated with it.

Commented [LC315]:

Question #380

You're going to configure the CloudTrail Processing Library in order to log bucket actions.

Which command generates a .jar file from the source code for the CloudTrail Processing Library?

- A. mvn javac mvn -install processor
- B. jar install processor
- C. build jar -Dgpg.processor
- D. mvn clean install -Dgpg.skip=true

Commented [LC316]:

Question #381

You must do ad hoc business analytics queries on well-structured data. Data is continually being ingested at a fast rate of speed. Your business intelligence department can assist you.

comprehend SQL.

Which Amazon Web Services (AWS) service(s) should you investigate first?

- A. Kinesis Firehose + RDS
- B. Kinesis Firehose + RedShift
- C. EMR using Hive
- D. EMR running Apache Spark

Commented [LC317]: The data is being ingested quickly, and Redshift is faster. Not to mention it takes work to get an EMR cluster going. Option B.

Question #382

Elastic Beanstalk is being used to run your e-commerce site. The shop is built on an open-source e-commerce platform and is deployed in an Auto Scaling group across numerous instances. Your development staff often builds new e-commerce shop "extensions." These extensions provide PHP source code and a SQL upgrade script for doing any required database schema modifications. You've observed that certain extension deployments fail owing to a SQL upgrade script problem. You discover that this is because the SQL script is being performed across all of your Amazon EC2 instances.

How would you guarantee that the SQL script is performed just once every deployment, regardless of the number of Amazon EC2 instances that are running?

- A. Use a "Container command" within an Elastic Beanstalk configuration file to execute the script, ensuring that the "leader only" flag is set to true.
- B. Make use of the Amazon EC2 metadata service to query whether the instance is marked as the leader in the Auto Scaling group. Only execute the script if "true" is returned.
- C. Use a "Solo Command" within an Elastic Beanstalk configuration file to execute the script. The Elastic Beanstalk service will ensure that the command is only executed once.
- D. Update the Amazon RDS security group to only allow write access from a single instance in the Auto Scaling group; that way, only one instance will successfully execute the script on the database.

Commented [LC318]: I'll go with A Ref:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html#linux-container-commands>

Question #383

Your organization must automate three levels of a massive cloud deployment. You want to be able to watch the progress of this deployment over time and carefully regulate any modifications.

What is an effective method for automating a stack in order to achieve these requirements?

- A. Use OpsWorks Stacks with three layers to model the layering in your stack.
- B. Use CloudFormation Nested Stack Templates, with three child stacks to represent the three logical layers of your cloud.
- C. Use AWS Config to declare a configuration set that AWS should roll out to your cloud.
- D. Use Elastic Beanstalk Linked Applications, passing the important DNS entries between layers using the metadata interface.

Commented [LC319]: Only CloudFormation allows source controlled, declarative templates as the basis for stack automation. Nested Stacks help achieve clean separation of layers while simultaneously providing a method to control all layers at once when needed.

Question #384

Your business runs an application on AWS CloudFormation, which includes a load balancer, an Auto Scaling group of web servers, and an Amazon RDS instance. You update the existing test stack when testing small changes and establish a new stack when testing big changes to save time and money. Each version of your application must be registered once and only once with a Configuration Management Database as part of the testing method (CMDB).

Which registration method is the most cost-effective?

- A. Use Auto Scaling Leader Node functionality to notify the registration application from the UserData script of a single Instance. Use the AWS CloudFormation cfn-hup helper application to receive template updates on the leader node, which then notifies the CMDB.
- B. Define an AWS::CloudFormation::CustomResource in the AWS CloudFormation template, with the application version as one of its properties. Modify the CMDB to subscribe to the resource's creation and update notifications.
- C. Define an AWS::CloudFormation::HttpRequest in the AWS CloudFormation template, and configure it to notify the CMDB on stack creation and update.
- D. Define an AWS::EC2::Instance resource in the AWS CloudFormation template that is configured to run a UserData script to notify the CMDB and then terminate itself on completion.

Question #385

Your existing log analysis program takes more than four hours to compile a report of your web application's top ten users. You've been tasked with developing a system capable of reporting this data in real time, ensuring that the report is always current, and handling spikes in the amount of queries to your web application.

Choose the most cost-effective solution that satisfies the criteria.

- A. Publish your data to CloudWatch Logs, and configure your application to autoscale to handle the load on demand.
- B. Publish your log data to an Amazon S3 bucket. Use AWS CloudFormation to create an Auto Scaling group to scale your post-processing application which is configured to pull down your log files stored on Amazon S3.
- **C. Post your log data to an Amazon Kinesis data stream, and subscribe your log-processing application so that is configured to process your logging data.**
- D. Configure an Auto Scaling group to increase the size of your Amazon EMR cluster.
- E. Create a multi-AZ Amazon RDS MySQL cluster, post the logging data to MySQL, and run a map reduce job to retrieve the required information on user counts.

Commented [LC320]: C as this is an AWS Use case for Kinesis as a rolling, immutable log.

Question #386

You're creating a service that gathers clickstream data in bulk and sends weekly reports to users through email. The data is very erratic, geographically dispersed, large-scale, and unpredictable.

How should this system be designed?

- A. Use a large RedShift cluster to perform the analysis, and a fleet of Lambdas to perform record inserts into the RedShift tables. Lambda will scale rapidly enough for the traffic spikes.
- **B. Use a CloudFront distribution with access log delivery to S3. Clicks should be recorded as querystring GETs to the distribution. Reports are built and sent by periodically running EMR jobs over the access logs in S3.**
- C. Use API Gateway invoking Lambdas which PutRecords into Kinesis, and EMR running Spark performing GetRecords on Kinesis to scale with spikes. Spark on EMR outputs the analysis to S3, which are sent out via email.
- D. Use AWS Elasticsearch service and EC2 Auto Scaling groups. The Autoscaling groups scale based on click throughput and stream into the Elasticsearch domain, which is also scalable. Use Kibana to generate reports periodically.

Commented [LC321]: Because you only need to batch analyze, anything using streaming is a waste of money. CloudFront is a Gigabit-Scale HTTP(S) global request distribution service, so it can handle scale, geo-spread, spikes, and unpredictability. The Access Logs will contain the GET data and work just fine for batch analysis and email using EMR. Can you use Amazon CloudFront if you expect usage peaks higher than 10 Gbps or 15,000 RPS? Yes. Complete our request for higher limits here, and we will add more capacity to your account within two business days.

Reference:
<https://aws.amazon.com/cloudfront/faqs/>

Question #387

One of an application team's internal tools is being refactored to operate on AWS rather than on-premises hardware. Currently, all code is written in Python and is self-contained. Additionally, there is no external state storage or relational database to query.

Which deployment process results in the fewest changes between development and production?

- **A. Developers should use Docker for local development. When dependencies are changed and a new container is ready, use AWS CodePipeline and AWS CodeBuild to perform functional tests and then upload the new container to Amazon ECR. Use AWS CloudFormation with the custom container to deploy the new Amazon ECS.**
- B. Developers should use Docker for local development. Use AWS SMS to import these containers as AMIs for Amazon EC2 whenever dependencies are updated. Use AWS CodePipeline to test new code changes against the Auto Scaling group.
- C. Developers should use their native Python environment. When Dependencies are changed and a new container is ready, use AWS CodePipeline and AWS CodeBuild to perform functional tests and then upload the new container to the Amazon ECR. Use AWS CloudFormation with the custom container to deploy the new Amazon ECS.
- D. Developers should use their native Python environment. When Dependencies are changed and a new code is ready, use AWS CodePipeline and AWS CodeBuild to perform functional tests and then upload the new container to the Amazon ECR. Use CodePipeline and CodeBuild with the custom container to test new code changes inside AWS Elastic Beanstalk.

Commented [LC322]:

Question #388

A web application is deployed using Amazon EC2 instances that are members of an Auto Scaling group. The instances register their private IP addresses with a monitoring system during the bootstrapping procedure. The monitoring system conducts health checks on such IP addresses on a regular basis and sends alarms if an instance becomes unresponsive.

The present deployment approach substitutes new EC2 instances for the old ones. A DevOps Engineer has detected that the monitoring system is issuing false alarms during a deployment and has been assigned the responsibility of resolving these false alarms.

Which option satisfies these objectives while remaining compatible with the present deployment method?

- A. Define an Amazon CloudWatch Events target, an AWS Lambda function, and a lifecycle hook attached to the Auto Scaling group. Configure CloudWatch Events to invoke Amazon SNS to send a message to the Systems Administrator group for remediation.
- B. Define an AWS Lambda function and a lifecycle hook attached to the Auto Scaling group. Configure the lifecycle hook to invoke the Lambda function, which removes the entry of the private IP from the monitoring system upon instance termination.
- **C. Define an Amazon CloudWatch Events target, an AWS Lambda function, and a lifecycle hook attached to the Auto Scaling group. Configure CloudWatch Events to invoke the Lambda function, which removes the entry of the private IP from the monitoring system upon instance termination.**
- D. Define an AWS Lambda function that will run a script when instance termination occurs in an Auto Scaling group. The script will remove the entry of the private IP from the monitoring system.

Commented [LC323]: They changed the name of Cloudwatch Events to EventBridge

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/cloud-watch-events.html>

Question #389

A DevOps Engineer is deploying a new application using Amazon Route 53, an Application Load Balancer, Auto Scaling, and Amazon DynamoDB architecture. One of the primary criteria for this launch is that the application be capable of scaling in response to an increase in demand. During times of low utilization, infrastructure components must be reduced in size to save money.

How can the DevOps Engineer ensure that the criteria are met? (Select two.)

- A. Use AWS Trusted Advisor to submit limit increase requests for the Amazon EC2 instances that will be used by the infrastructure.
- **B. Determine which Amazon EC2 instance limits need to be raised by leveraging AWS Trusted Advisor, and submit a request to AWS Support to increase those limits.**
- **C. Enable Auto Scaling for the DynamoDB tables that are used by the application.**
- D. Configure the Application Load Balancer to automatically adjust the target group based on the current load.
- E. Create an Amazon CloudWatch Events scheduled rule that runs every 5 minutes to track the current use of the Auto Scaling group. If usage has changed, trigger a scale-up event to adjust the capacity. Do the same for DynamoDB read and write capacities.

Commented [LC324]: BC We need to check limits and enable autoscaling for dynamo db

For B -
<https://aws.amazon.com/blogs/mt/monitoring-service-limits-with-trusted-advisor-and-amazon-cloudwatch/>

For C -
<https://aws.amazon.com/blogs/database/amazon-dynamodb-auto-scaling-performance-and-cost-optimization-at-any-scale/>

Commented [LC325]:

Question #390

A business is in the process of migrating to the AWS Cloud. Internal clients are divided into two categories based on their AWS proficiency: beginners and experts.

The DevOps Engineer must provide a solution that enables novices to install a limited selection of AWS architectural blueprints expressed in AWS CloudFormation templates. Deployment should be limited to pre-configured Virtual Private Clouds (VPCs). Expert users, on the other hand, should be allowed to deploy designs without restriction. Additionally, experts should have access to additional AWS services as required.

How can the Engineer develop a system that satisfies these objectives while incurring the MINIMUM amount of overhead?

- A. Apply constraints to the parameters in the templates, limiting the VPCs available for deployments. Store the templates on Amazon S3. Create an IAM group for beginners and give them access to the templates and CloudFormation. Create a separate group for experts, giving them access to the templates, CloudFormation, and other AWS services.
- B. Store the templates on Amazon S3. Use AWS Service Catalog to create a portfolio of products based on those templates. Apply template constraints to the products with rules limiting VPCs available for deployments. Create an IAM group for beginners giving them access to the portfolio. Create a separate group for experts giving them access to the templates, CloudFormation, and other AWS services.
- C. Store the templates on Amazon S3. Use AWS Service Catalog to create a portfolio of products based on those templates. Create an IAM role restricting VPCs available for creation of AWS resources. Apply a launch constraint to the products using this role. Create an IAM group for beginners giving them access to the portfolio. Create a separate group for experts giving them access to the portfolio and other AWS services.
- D. Create two templates for each architecture blueprint where only one of them limits the VPC available for deployments. Store the templates in Amazon DynamoDB. Create an IAM group for beginners giving them access to the constrained templates and CloudFormation. Create a separate group for experts giving them access to the unconstrained templates, CloudFormation, and other AWS services.

Commented [LC326]: B is right

https://docs.aws.amazon.com/servicecatalog/latest/adminguide/reference-template_constraint_rules.html

Question #391

You have a code repository that stores its data on Amazon S3. During a recent assessment of your security procedures, certain questions concerning the integrity of the data in the Amazon S3 bucket were highlighted. Another point of contention centered on the security of code deployment from Amazon S3 to applications running on Amazon EC2 in a virtual private cloud.

What actions can you take to alleviate these concerns? (Select two.)

- A. Add an Amazon S3 bucket policy with a condition statement to allow access only from Amazon EC2 instances with RFC 1918 IP addresses and enable bucket versioning.
- B. Add an Amazon S3 bucket policy with a condition statement that requires multi-factor authentication in order to delete objects and enable bucket versioning.
- C. Use a configuration management service to deploy AWS Identity and Access Management user credentials to the Amazon EC2 instances. Use these credentials to securely access the Amazon S3 bucket when deploying code.
- D. Create an Amazon Identity and Access Management role with authorization to access the Amazon S3 bucket, and launch all of your application's Amazon EC2 instances with this role.
- E. Use AWS Data Pipeline to lifecycle the data in your Amazon S3 bucket to Amazon Glacier on a weekly basis.
- F. Use AWS Data Pipeline with multi-factor authentication to securely deploy code from the Amazon S3 bucket to your Amazon EC2 instances.

Commented [LC327]: Ans is B and D

B is Ans of "some concerns were raised about maintaining the integrity of the data in the Amazon S3 bucket."

D is Ans of "securely deploying code from Amazon S3 to applications running on Amazon EC2 in a virtual private cloud"

Commented [LC328]:

Question #392

Two Amazon EC2 Auto Scaling groups, each configured with an Application Load Balancer, are used to deploy an application. The application is deployed to one of the Auto Scaling groups, and an Amazon Route 53 alias record is created that points to the Application Load Balancer of the previous Auto Scaling group deployed.

Alternate deployments between the two Auto Scaling groups.

The program receives requests from home security devices. The development team observes that fresh requests continue to flow into the legacy stack days after deployment. The problem is caused by devices that do not adhere to the Amazon Route 53 alias record's Time to Live (TTL) setting.

What measures should the DevOps Engineer take to solve the problem of requests being routed to the old stacks while using the fewest amount of extra resources possible?

- A. Create a fleet of Amazon EC2 instances running HAProxy behind an Application Load Balancer. The HAProxy instances will proxy the requests to one of the existing Auto Scaling groups. After a deployment the HAProxy instances are updated to send requests to the newly deployed Auto Scaling group.
- B. Reduce the application to one Application Load Balancer. Create two target groups named Blue and Green. Create a rule on the Application Load Balancer pointed to a single target group. Add logic to the deployment to update the Application Load Balancer rule to the target group of the newly deployed Auto Scaling group.
- C. Move the application to an AWS Elastic Beanstalk application with two environments. Perform new deployments on the non-live environment. After a deployment, perform an Elastic Beanstalk CNAME swap to make the newly deployed environment the live environment.
- D. Create an Amazon CloudFront distribution. Set the two existing Application Load Balancers as origins on the distribution. After a deployment, update the CloudFront distribution behavior to send requests to the newly deployed Auto Scaling group.

Commented [LC329]: B is correct
<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-components>

Question #393

A social networking service maintains a web API that enables third-party applications to search for public content. The program stores post data in Amazon DynamoDB and indexes it using AWS Lambda functions, with an Amazon ES domain storing the indexes and enabling search capability. The service must retain full capacity throughout deployments and guarantee that unsuccessful deployments must not result in downtime, capacity reduction, or the inability to perform future deployments.

How are these stipulations to be met? (Select two.)

- A. Run the web application in AWS Elastic Beanstalk with the deployment policy set to All at Once. Deploy the Lambda functions, DynamoDB tables, and Amazon ES domain with an AWS CloudFormation template.
- B. Deploy the web application, Lambda functions, DynamoDB tables, and Amazon ES domain in an AWS CloudFormation template. Deploy changes with an AWS CodeDeploy in-place deployment.
- C. Run the web application in AWS Elastic Beanstalk with the deployment policy set to Immutable. Deploy the Lambda functions, DynamoDB tables, and Amazon ES domain with an AWS CloudFormation template.
- D. Deploy the web application, Lambda functions, DynamoDB tables, and Amazon ES domain in an AWS CloudFormation template. Deploy changes with an AWS CodeDeploy blue/green deployment.
- E. Run the web application in AWS Elastic Beanstalk with the deployment policy set to Rolling. Deploy the Lambda functions, DynamoDB tables, and Amazon ES domain with an AWS CloudFormation template.

Commented [LC330]:

Commented [LC331]:

Question #394 [SKIP]

Your serverless architecture, which makes use of AWS API Gateway, AWS Lambda, and AWS DynamoDB, saw a significant spike in traffic to a continuous 400 requests per second, as well as a drastic increase in failure rates. Your queries typically take 500 milliseconds to complete under regular operation. Your DynamoDB table did not exceed 50% of provisioned throughput, and the main keys of the database are properly constructed.

What is the most probable cause of the problem?

- A. Your API Gateway deployment is throttling your requests.
- B. Your AWS API Gateway Deployment is bottlenecking on request (de)serialization.
- C. You did not request a limit increase on concurrent Lambda function executions.
- D. You used Consistent Read requests on DynamoDB and are experiencing semaphore lock.

Commented [LC332]: Not sure how old this question is...

For Lambda: The default regional concurrency quota starts at 1,000 instances. For an initial burst of traffic, your functions' cumulative concurrency in a Region can reach an initial level of between 500 and 3000, which varies per Region.

API Gateway: 10,000 requests per second (RPS) with an additional burst capacity provided by the token bucket algorithm, using a maximum bucket capacity of 5,000 requests.

Question #395

You have an application that does asynchronous processing and makes use of an Auto Scaling Group and a SQS Queue. The Auto Scaling Group scales in accordance with the work queue's depth. The task completion velocity has decreased, the Auto Scaling Group size has reached its maximum, but the incoming job completion velocity has remained constant.

What may be an issue?

- **A. Some of the new jobs coming in are malformed and unprocessable.**
- B. The routing tables changed and none of the workers can process events anymore.
- C. Someone changed the IAM Role Policy on the instances in the worker group and broke permissions to access the queue.
- D. The scaling metric is not functioning correctly.

Commented [LC333]: The IAM Role must be fine, as if it were broken, NO jobs would be processed since the system would never be able to get any queue messages. The same reasoning applies to the routing table change. The scaling metric is fine, as instance count increased when the queue depth increased due to more messages entering than exiting. Thus, the only reasonable option is that some of the recent messages must be malformed and unprocessable.

Question #396

A business has moved its container-based apps to Amazon EKS and wishes to automate email alerts. Notifications are delivered to each email account for certain EKS component-related events. Amazon SNS topics and an AWS Lambda function will be used to assess incoming log events and publish messages to the appropriate SNS topic.

Which logging solution meets these criteria?

- **A. Enable Amazon CloudWatch Logs to log the EKS components. Create a CloudWatch subscription filter for each component with Lambda as the subscription feed destination.**
- B. Enable Amazon CloudWatch Logs to log the EKS components. Create CloudWatch Logs Insights queries linked to Amazon CloudWatch Events events that trigger Lambda.
- C. Enable Amazon S3 logging for the EKS components. Configure an Amazon CloudWatch subscription filter for each component with Lambda as the subscription feed destination.
- D. Enable Amazon S3 logging for the EKS components. Configure S3 PUT Object event notifications with AWS Lambda as the destination.

Commented [LC334]: A is the correct answer. It's simple and works while B is convoluted.

Question #397

Your business builds a number of web apps on a range of platforms and programming languages, each with its own set of dependencies. Each application must be swiftly produced and deployed, as well as highly evadable, in order to meet your business objectives.

Which of the following strategies should you use to expedite the deployment of these applications?

- **A. Develop the applications in Docker containers, and then deploy them to Elastic Beanstalk environments with Auto Scaling and Elastic Load Balancing.**
- B. Use the AWS CloudFormation Docker import service to build and deploy the applications with high availability in multiple Availability Zones.
- C. Develop each application's code in DynamoDB, and then use hooks to deploy it to Elastic Beanstalk environments with Auto Scaling and Elastic Load Balancing.
- D. Store each application's code in a Git repository, develop custom package repository managers for each application's dependencies, and deploy to AWS OpsWorks in multiple Availability Zones.

Commented [LC335]: Ans is A:

Docker is perfect for installing applications with different languages and dependencies and ElasticBeanstalk deploys the Apps with less time and less expertise

Question #398

A DevOps engineer is delivering a new version of an application to an AWS CodeDeploy deployment group connected with the company's Amazon EC2 instances.

After a period of time, the deployment is deemed unsuccessful. The engineer notices that all events linked with the specified deployment ID are in the Skipped state, indicating that no code was deployed in the deployment group's instances.

What legitimate justifications exist for this failure? (Make a selection of at least two.)

- A. The networking configuration does not allow the EC2 instances to reach the internet via a NAT gateway or internet gateway, and the CodeDeploy endpoint cannot be reached.
- B. The IAM user who triggered the application deployment does not have permission to interact with the CodeDeploy endpoint.
- C. The target EC2 instances were not properly registered with the CodeDeploy endpoint.
- D. An instance profile with proper permissions was not attached to the target EC2 instances.
- E. The appspec.yml file was not included in the application revision.

Commented [LC336]: Your instance might not be able to reach the CodeDeploy or S3 public endpoint using port 443. Try one of the following:

If an instance is provisioned in a private subnet, use a NAT gateway instead of an internet gateway in the route table. For more information, see NAT Gateways.

The instance you're deploying to might not have an IAM instance profile attached, or it might have an IAM instance profile attached that does not have the required permissions.

Commented [LC337]:

Question #399

The popular worldwide web application of a business is hosted on Amazon EC2 instances behind an Application Load Balancer (ALB) that utilizes an Auto Scaling group. The firm is releasing a new service and anticipates unanticipated traffic surges. The site already has a significant quantity of media material, and the new feature allows users to contribute ratings and comments, which will be kept in a new Amazon DynamoDB database. A DevOps Engineer is responsible for ensuring that the web application scales appropriately to handle growing traffic and workload.

Which sequence of steps will do this? (Select two.)

- A. Configure an Amazon CloudFront distribution to cache the web application's static and dynamic content.
- B. Configure the web application's ALB to cache content in Amazon ElastiCache, honoring the HTTP cache headers.
- C. Process the new ratings and comments asynchronously using Amazon SQS.
- D. Replace the DynamoDB table with DynamoDB Accelerator to store the ratings and comments to reduce latency.
- E. Set up AWS Global Accelerator to cache static content and pass dynamic requests to the web application's ALB endpoint.

Commented [LC338]: Answer: AC
B - not possible
D - it is suitable only for read operations
E - A is a better option because the app is global

Commented [LC339]:

Questions 400-449

Question #400

A corporation wants to bootstrap real laptops for developers using AWS Systems Manager docs. The bootstrap source code is available on GitHub. A DevOps engineer has already generated a Systems Manager activation, installed the Systems Manager agent with the registration code, and provisioned all laptops with an activation ID.

Which further set of measures should be taken?

- A. Configure the Systems Manager document to use the AWS-RunShellScript command to copy the files from GitHub to Amazon S3, then use the aws- downloadContent plugin with a sourceType of S3.
- B. Configure the Systems Manager document to use the aws-configurePackage plugin with an install action and point to the Git repository.
- C. Configure the Systems Manager document to use the aws-downloadContent plugin with a sourceType of GitHub and sourceInfo with the repository details.
- D. Configure the Systems Manager document to use the aws:softwareInventory plugin and run the script from the Git repository.

Commented [LC340]: Should be C. When the file is copied to S3, you don't need to download the content anymore, you can just use remote shell script to run it from S3

<https://docs.aws.amazon.com/systems-manager/latest/userguide/integration-s3-shell.html>

Question #401

A DevOps engineer has been assigned the responsibility of ensuring that all Amazon S3 buckets, save those named "public," restrict access to authorized users through S3 bucket rules. The security team want to be alerted when a bucket is created without the appropriate policy in place, as well as to have the policy updated automatically.

Which solutions will satisfy these criteria?

- A. Create a custom AWS Config rule that will trigger an AWS Lambda function when an S3 bucket is created or updated. Use the Lambda function to look for S3 buckets that should be private, but that do not have a bucket policy that enforces privacy. When such a bucket is found, invoke a remediation action and use Amazon SNS to notify the security team.
- B. Create an Amazon EventBridge (Amazon CloudWatch Events) rule that triggers when an S3 bucket is created. Use an AWS Lambda function to determine whether the bucket should be private. If the bucket should be private, update the PublicAccessBlock configuration. Configure a second EventBridge (CloudWatch Events) rule to notify the security team using Amazon SNS when PutBucketPolicy is called.
- C. Create an Amazon S3 event notification that triggers when an S3 bucket is created that does not have the word "public" in the name. Define an AWS Lambda function as a target for this notification and use the function to apply a new default policy to the S3 bucket. Create an additional notification with the same filter and use Amazon SNS to send an email to the security team.
- D. Create an Amazon EventBridge (Amazon CloudWatch Events) rule that triggers when a new object is created in a bucket that does not have the word "public" in the name. Target and use an AWS Lambda function to update the PublicAccessBlock configuration. Create an additional notification with the same filter and use Amazon SNS to send an email to the security team.

Commented [LC341]: The security team wants to be notified WHEN a bucket is CREATED without the proper policy and for the policy to be automatically updated. So D) is wrong. And would be a hell to receive a notification every time an object is created in a bucket (even if the right policy is already in place). I'll go with A)

Reference:
<https://docs.aws.amazon.com/config/latest/developerguide/notifications-for-AWS-Config.html>

Question #402

A development team establishes an AWS CodeBuild build project. The build project runs automated tests on modules that make use of Amazon Web Services.

Which of the following will ensure that the tests are conducted in the most secure manner possible?

- A. Generate credentials for an IAM user with a policy attached to allow the actions on AWS services. Store credentials as encrypted environment variables for the build project. As part of the build script, obtain the credentials to run the integration tests.
- B. Have CodeBuild run only the integration tests as a build job on a Jenkins server. Create a role that has a policy attached to allow the actions on AWS services. Generate credentials for an IAM user that is allowed to assume the role. Configure the credentials as secrets in Jenkins, and allow the build job to use them to run the integration tests.
- C. Create a service role in IAM to be assumed by CodeBuild with a policy attached to allow the actions on AWS services. Configure the build project to use the role created.
- D. Use AWS managed credentials. Encrypt the credentials with AWS KMS. As part of the build script, decrypt with AWS KMS and use these credentials to run the integration tests.

Commented [LC342]:

Question #403

A financial institution supplies application teams with security-hardened AMIs of Red Hat Enterprise Linux 7.4 and Windows Server 2016 for deployments.

A DevOps Engineer must automate the daily check of each AMI for the most recent CVE.

How should the Engineer go about implementing these tests with the help of Amazon Inspector?

- A. Install the Amazon Inspector agent in each AMI. Configure AWS Step Functions to launch an Amazon EC2 instance for each operating system from the hardened AMI, and tag the instance with SecurityCheck: True. Once EC2 instances have booted up, Step Functions will trigger an Amazon Inspector assessment for all instances with the tag SecurityCheck: True. Implement a scheduled Amazon CloudWatch Events rule that triggers Step Functions once each day.
- B. Tag each AMI with SecurityCheck: True. Configure AWS Step Functions to first compose an Amazon Inspector assessment template for all AMIs that have the tag SecurityCheck: True and second to make a call to the Amazon Inspector API action StartAssessmentRun. Implement a scheduled Amazon CloudWatch Events rule that triggers Step Functions once each day.
- C. Tag each AMI with SecurityCheck: True. Implement a scheduled Amazon Inspector assessment to run once each day for all AMIs with the tag SecurityCheck: True. Amazon Inspector should automatically launch an Amazon EC2 instance for each AMI and perform a security assessment.
- D. Tag each instance with SecurityCheck: True. Implement a scheduled Amazon Inspector assessment to run once each day for all instances with the tag SecurityCheck: True. Amazon Inspector should automatically perform an in-place security assessment for each AMI.

Commented [LC343]: Only A will work.

B: Inspector templates allow for tagging EC2 instances, not AMI's.

C: Inspector cannot launch instances for you!

D: Inspector only works against running instances. It can't do an "in-place" assessment of an AMI.

Question #404

The development team of an online retailer has been transferred to business support and want to use the AWS Health Dashboard and AWS Health API to automate remedial steps for AWS resource health concerns. The first use case is in response to AWS recognizing an IAM access key that is publicly available on a code repository website. Automatically, the IAM access key will be deleted and a notice to the Security team will be sent.

How is this to be accomplished?

- A. Create an AWS Lambda function to delete the IAM access key. Send AWS CloudTrail logs to AWS CloudWatch logs. Create a CloudWatch Logs metric filter for the AWS_RISK_CREDENTIALS_EXPOSED event with two actions: first, run the Lambda function; second, use Amazon SNS to send a notification to the Security team.
- B. Create an AWS Lambda function to delete the IAM access key. Create an AWS Config rule for changes to aws.health and the AWS_RISK_CREDENTIALS_EXPOSED event with two actions: first, run the Lambda function; second, use Amazon SNS to send a notification to the Security team.
- C. Use AWS Step Functions to create a function to delete the IAM access key, and then use Amazon SNS to send a notification to the Security team. Create an AWS Personal Health Dashboard rule for the AWS_RISK_CREDENTIALS_EXPOSED event; set the target of the Personal Health Dashboard rule to Step Functions.
- D. Use AWS Step Functions to create a function to delete the IAM access key, and then use Amazon SNS to send a notification to the Security team. Create an Amazon CloudWatch Events rule with an aws.health event source and the AWS_RISK_CREDENTIALS_EXPOSED event; set the target of the CloudWatch Events rule to Step Functions.

Commented [LC344]: D

<https://aws.amazon.com/blogs/compute/automate-your-it-operations-using-aws-step-functions-and-amazon-cloudwatch-events/>

Question #405

You are creating a system that will need at least eight m4.large instances to serve traffic. When developing a system for high availability in the us-east-1 area, which has six Availability Zones, your business must be able to absorb the loss of an entire availability zone.

How should the servers be distributed to maximize cost savings, given that all EC2 nodes are correctly connected to an ELB? Your VPC account may make use of us-AZs east-1's a through f.

- A. 3 servers in each of AZ's a through d, inclusive.
- B. 8 servers in each of AZ's a and b.
- C. 2 servers in each of AZ's a through e, inclusive.
- D. 4 servers in each of AZ's a through c, inclusive.

Commented [LC345]: C

Most cost eff is lowest total count while meeting req for 8 with 1 AZ down, so its 10 -2 =8. Other plausible options running 12 nodes so more 3 expensive...

Question #406

A development team is in the process of developing a serverless application on AWS. To rapidly detect and resolve possible production problems, the team chooses to test the modifications by rolling them out to a limited number of users prior to the complete release. The DevOps Engineer is responsible for developing a solution that minimizes downtime and impact.

Which of the following options is the most appropriate for meeting the requirements? (Select two.)

- A. Create an Application Load Balancer with two target groups. Set up the Application Load Balancer for Amazon API Gateway private integration. Associate one target group to the current version and the other target group to the new version. Configure API Gateway to route 10% of incoming traffic to the new version. As the new version becomes stable, configure API Gateway to send all traffic to the new version and detach the old version from the load balancer.
- B. Create an alias for an AWS Lambda function pointing to both the current and new versions. Configure the alias to route 10% of incoming traffic to the new version. As the new version is considered stable, update the alias to route all traffic to the new version.
- C. Create a failover record set in AWS Route 53 pointing to the AWS Lambda endpoints for the old and new versions. Configure Route 53 to route 10% of incoming traffic to the new version. As the new version becomes stable, update the DNS record to route all traffic to the new version.
- D. Create an ELB Network Load Balancer with two target groups. Set up the Network Load Balancer for Amazon API Gateway private integration. Associate one target group with the current version and the other target group with the new version. Configure the load balancer to route 10% of incoming traffic to the new version. As the new version becomes stable, detach the old version from the load balancer.
- E. In Amazon API Gateway, create a canary release deployment by adding canary settings to the stage of a regular deployment. Configure API Gateway to route 10% of the incoming traffic to the canary release. As the canary release is considered stable, promote it to a production release.

Commented [LC346]: I'll go with B,E Remember: ELB Targets possible options are: - EC2 Instances - IP Address - Lambda Functions

Commented [LC347]:

Question #407

A Node.js e-commerce application is managed by an engineering team. Components of the present environment include the following:

- * Amazon S3 buckets for storing content
- * Amazon EC2 for the front-end web servers
- * AWS Lambda for executing image processing
- * Amazon DynamoDB for storing session-related data

The team anticipates a considerable rise in site traffic. The program should continue to operate normally despite the increased demand. The team conducted preliminary testing by adding more servers to the EC2 front-end to accommodate the increased traffic, but the instances took up to 20 minutes to completely setup. The team wishes to shorten the time required for setup.

What improvements would the Engineering team need to make to the solution in order to make it the MOST robust and highly accessible while still meeting the anticipated increase in demand?

- A. Use AWS OpsWorks to automatically configure each new EC2 instance as it is launched. Configure the EC2 instances by using an Auto Scaling group behind an Application Load Balancer across multiple Availability Zones. Implement Amazon DynamoDB Auto Scaling. Use Amazon Route 53 to point the application DNS record to the Application Load Balancer.
- B. Deploy a fleet of EC2 instances, doubling the current capacity, and place them behind an Application Load Balancer. Increase the Amazon DynamoDB read and write capacity units. Add an alias record that contains the Application Load Balancer endpoint to the existing Amazon Route 53 DNS record that points to the application.
- C. Configure Amazon CloudFront and have its origin point to Amazon S3 to host the web application. Implement Amazon DynamoDB Auto Scaling. Use Amazon Route 53 to point the application DNS record to the CloudFront DNS name.
- D. Use AWS Elastic Beanstalk with a custom AMI including all web components. Deploy the platform by using an Auto Scaling group behind an Application Load Balancer across multiple Availability Zones. Implement Amazon DynamoDB Auto Scaling. Use Amazon Route 53 to point the application DNS record to the Elastic Beanstalk load balancer.

Commented [LC348]: Right answer: D - custom AMI is key here it will reduce the provisioning time dramatically - main issue - and also multi-az and ALB are mentioned for resiliency and high-availability. Wrong: A - OpsWorks could do it but "automatically configure each new EC2 as it is launched" would keep the slow start issue. B - Could be right but does not mention multi-AZ as well is based on manual changes instead of auto scaling. C - you need EC2 due to Node.js server-side

Question #408

A DevOps engineer is responsible for building governance rules for a corporation that needs its infrastructure to be physically located in the United States. The engineer must limit which Regions may be utilized and ensure that an alert is given immediately if any behavior inconsistent with the governance rules occurs. Controls should be activated automatically when a new Region outside the United States is created.

Which combination of acts satisfies these criteria? (Select two.)

- A. Create an AWS Organizations SCP that denies access to all non-global services in non-US Regions. Attach the policy to the root of the organization.
- B. Configure AWS CloudTrail to send logs to Amazon CloudWatch Logs and enable it for all Regions. Use a CloudWatch Logs metric filter to send an alert on any service activity in non-US Regions.
- C. Use an AWS Lambda function that checks for AWS service activity and deploy it to all Regions. Write an Amazon CloudWatch Events rule that runs the Lambda function every hour, sending an alert if activity is found in a non-US Region.
- D. Use an AWS Lambda function to query Amazon Inspector to look for service activity in non-US Regions and send alerts if any activity is found.
- E. Write an SCP using the `aws:RequestedRegion` condition key limiting access to US Regions. Apply the policy to all users, groups, and roles.

Commented [LC349]: AB is the correct answer. A is correct in that you need to provide exemptions for global services that have endpoints that are physically hosted in the required regions (using `NotAction` for those exemptions). You also don't want to apply to all roles, either, such as AWS ControlTower roles (using a Condition and `"ArnNotlike"` to exclude needed roles).

https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scps_examples.html

Commented [LC350]:

Question #409

A fast expanding firm want to increase its AWS development environments to meet developer demand. Manually built development environments are created through the AWS Management Console. The Networking team manages the networking architecture through AWS CloudFormation, exporting stack output data for the Amazon VPC and all subnets. The development environments share a number of similar components, including Application Load Balancers, Amazon EC2 Auto Scaling groups, security groups, and Amazon DynamoDB tables.

To meet demand, the DevOps Engineer wants to automate the process of creating development environments. Because the infrastructure needed to serve the application is projected to develop, the deployed infrastructure must be readily updated. To generate a template for the development environments, CloudFormation will be utilized.

Which strategy will satisfy these needs and offer developers with consistent AWS environments quickly?

- A. Use `Fn::ImportValue` intrinsic functions in the Resources section of the template to retrieve Virtual Private Cloud (VPC) and subnet values. Use CloudFormation StackSets for the development environments, using the Count input parameter to indicate the number of environments needed. use the `UpdateStackSet` command to update existing development environments.
- B. Use nested stacks to define common infrastructure components. To access the exported values, use `TemplateURL` to reference the Networking team's template. To retrieve Virtual Private Cloud (VPC) and subnet values, use `Fn::ImportValue` intrinsic functions in the Parameters section of the master template. Use the `CreateChangeSet` and `ExecuteChangeSet` commands to update existing development environments.
- C. Use nested stacks to define common infrastructure components. Use `Fn::ImportValue` intrinsic functions with the resources of the nested stack to retrieve Virtual Private Cloud (VPC) and subnet values. Use the `CreateChangeSet` and `ExecuteChangeSet` commands to update existing development environments.
- D. Use `Fn::ImportValue` intrinsic functions in the Parameters section of the master template to retrieve Virtual Private Cloud (VPC) and subnet values. Define the development resources in the order they need to be created in the CloudFormation nested stacks. Use the `CreateChangeSet` and `ExecuteChangeSet` commands to update existing development environments.

Commented [LC351]: A - Incorrect since StackSets are not needed here. No mention of multi-region/multi-account deployments (although correct usage of intrinsic functions and refs to resource values)

B - If you reference the template URL, you wouldn't need to use the intrinsic function `Fn::import`. You would reference the output of the template URL function i.e. `TemplateName.Outputs.VariableName`

C - Correct

D - You do not use `Fn::import` within the Parameters section. It is solely used within the Resources section to reference stack outputs within nested stacks or other previously deployed stacks.

Ergo, answer is C.

Question #410

AWS CodePipeline is used by a business to manage and deliver infrastructure as code. AWS CloudFormation templates define the infrastructure, which is typically composed of many Amazon EC2 instances and Amazon RDS databases. The Security team has seen several operators establishing inbound security group rules with a source CIDR of 0.0.0.0/0 and want to prevent the deployment of rules with open CIDRs in a proactive manner. The DevOps Engineer will create a pre-deployment phase that verifies the CloudFormation template's security before it is processed by the pipeline. This check should permit inbound security group rules with a source CIDR of 0.0.0.0/0 only if the rule's description includes the phrase "Security Approval Ref XXXXX" (where XXXXX is a preallocated reference). If this condition is not satisfied, the pipeline step should fail and the deployment should be prevented.

How is this to be achieved?

- A. Enable a SCP in AWS Organizations. The policy should deny access to the API call Create Security GroupRule if the rule specifies 0.0.0.0/0 without a description referencing a security approval.
- **B. Add an initial stage to CodePipeline called Security Check. This stage should call an AWS Lambda function that scans the CloudFormation template and fails the pipeline if it finds 0.0.0.0/0 in a security group without a description referencing a security approval.**
- C. Create an AWS Config rule that is triggered on creation or edit of resource type EC2 SecurityGroup. This rule should call an AWS Lambda function to send a failure notification if the security group has any rules with a source CIDR of 0.0.0.0/0 without a description referencing a security approval.
- D. Modify the IAM role used by CodePipeline. The IAM policy should deny access.

Commented [LC352]: B, the requirement is to check for 0.0.0.0 before resource provisioning. So, need to lookup in the template.

Question #411

A huge ecommerce business makes extensive use of Amazon EBS-backed Amazon EC2 instances. To reduce human effort across all instances, a DevOps Engineer is assigned the responsibility of automating restart activities when planned EC2 instance retirement events occur.

How is this possible?

- A. Create a scheduled Amazon CloudWatch Events rule to execute an AWS Systems Manager automation document that checks if any EC2 instances are scheduled for retirement once a week. If the instance is scheduled for retirement, the automation document will hibernate the instance.
- B. Enable EC2 Auto Recovery on all of the instances. Create an AWS Config rule to limit the recovery to occur during a maintenance window only.
- C. Reboot all EC2 instances during an approved maintenance window that is outside of standard business hours. Set up Amazon CloudWatch alarms to send a notification in case any instance is failing EC2 instance status checks.
- **D. Set up an AWS Health Amazon CloudWatch Events rule to execute AWS Systems Manager automation documents that stop and start the EC2 instance when a retirement scheduled event occurs.**

Commented [LC353]: D is correct
<https://aws.amazon.com/premiumsupport/knowledge-center/ec2-instance-retirement/>

Question #412 [SKIP]

A consulting firm was hired to examine security vulnerabilities in a client company's application and to offer a strategy for resolving any concerns discovered. The following is a description of the architecture: Amazon S3 storage for content, an Auto Scaling group of Amazon EC2 instances behind an Elastic Load Balancer with associated Amazon EBS storage, and a MySQL database on Amazon RDS. Additionally, numerous AWS Lambda functions interact directly with the RDS database using code-based connection string declarations. The experts assessed the following as the primary security risk: the application does not adhere to the need for encryption at rest.

Which option will resolve this problem with the LEAST amount of operational overhead and will monitor for any future violations?

- A. Enable SSE encryption on the S3 buckets and RDS database. Enable OS-based encryption of data on EBS volumes. Configure Amazon Inspector agents on EC2 instances to report on insecure encryption ciphers. Set up AWS Config rules to periodically check for non-encrypted S3 objects.
- B. Configure the application to encrypt each file prior to storing on Amazon S3. Enable OS-based encryption of data on EBS volumes. Encrypt data on write to RDS. Run cron jobs on each instance to check for unencrypted data and notify via Amazon SNS. Use S3 Events to call an AWS Lambda function and verify if the file is encrypted.
- C. Enable Secure Sockets Layer (SSL) on the load balancer, ensure that AWS Lambda is using SSL to communicate to the RDS database, and enable S3 encryption. Configure the application to force SSL for incoming connections and configure RDS to only grant access if the session is encrypted. Configure Amazon Inspector agents on EC2 instances to report on insecure encryption ciphers.
- **D. Enable SSE encryption on the S3 buckets, EBS volumes, and the RDS database. Store RDS credentials in EC2 Parameter Store. Enable a policy on the S3 bucket to deny unencrypted puts. Set up AWS Config rules to periodically check for non-encrypted S3 objects and EBS volumes, and to ensure that RDS storage is encrypted.**

Commented [LC354]: Old, D should be right.

Question #413

A DevOps engineer is entrusted with the responsibility of developing a more reliable deployment strategy for a web application hosted on Amazon Web Services. Previously deployed solutions resulted in user-facing problems, excessive user traffic, and discrepancies across web servers behind an Application Load Balancer. The present technique stores the application's code in AWS CodeCommit. When developers commit changes to the repository's master branch, CodeCommit executes an AWS Lambda deploy function, which in turn invokes an AWS Systems Manager run command to compile and deploy the updated code to all Amazon EC2 instances.

Which activities should be made in conjunction to provide a more robust deployment solution? (Select two.)

- A. Create a pipeline in AWS CodePipeline with CodeCommit as a source provider. Create parallel pipeline stages to build and test the application. Pass the build artifact to AWS CodeDeploy.
- **B. Create a pipeline in AWS CodePipeline with CodeCommit as a source provider. Create separate pipeline stages to build and then test the application. Pass the build artifact to AWS CodeDeploy.**
- **C. Create and use an AWS CodeDeploy application and deployment group to deploy code updates to the EC2 fleet. Select the Application Load Balancer for the deployment group.**
- D. Create individual Lambda functions to run all build, test, and deploy actions using AWS CodeDeploy instead of AWS Systems Manager.
- E. Modify the Lambda function to build a single application package to be shared by all instances. Use AWS CodeDeploy instead of AWS Systems Manager to update the code on the EC2 fleet.

Commented [LC355]: I'll go with B, C References:

<https://docs.aws.amazon.com/codebuild/latest/userguide/how-to-create-pipeline-add-test.html>

<https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-codebuild-devicefarm.html>

Commented [LC356]:

Question #414

An AWS Elastic Beanstalk application was used to deploy a web application. The Application Developers are worried about the application's excessive latency in two distinct areas:

- ⇒ HTTP client requests to a third-party API
 - ⇒ MySQL client library queries to an Amazon RDS database
- A DevOps Engineer must collect trace data in order to identify problems.

Which steps will collect the trace information with the fewest modifications and affects on the application's performance?

- A. Add additional logging to the application code. Use the Amazon CloudWatch agent to stream the application logs into Amazon Elasticsearch Service. Query the log data in Amazon ES.
- B. Instrument the application to use the AWS X-Ray SDK. Post trace data to an Amazon Elasticsearch Service cluster. Query the trace data for calls to the HTTP client and the MySQL client.
- C. On the AWS Elastic Beanstalk management page for the application, enable the AWS X-Ray daemon. View the trace data in the X-Ray console.
- D. Instrument the application using the AWS X-Ray SDK. On the AWS Elastic Beanstalk management page for the application, enable the X-Ray daemon. View the trace data in the X-Ray console.

Commented [LC357]: <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-configuration-debugging.html>

Option D is correct!

Question #415

You're developing an AWS deployment system. You will deploy new code by bootstrapping instances in a private subnet inside a VPC at runtime through UserData scripts that refer to an S3 zip file object containing your code. In a public subnet, an ELB is equipped with network interfaces and connectivity to the instances. Route53 A Record Aliases are used to route requests from system users to the ELB. You are not making any usage of any VPC endpoints.

Which of the following is a danger associated with this approach?

- A. Route53 Alias records do not always update dynamically with ELB network changes after deploys.
- B. If the NAT routing for the private subnet fails, deployments fail.
- C. Kernel changes to the base AMI may render the code inoperable.
- D. The instances cannot be in a private subnet if the ELB is in a public one.

Commented [LC358]: Since you are not using VPC endpoints, outbound requests for the code sitting in S3 are routed through the NAT for the VPC's private subnets. If this networking fails, runtime bootstrapping through code download will fail due to network unavailability and lack of access to the Internet, and thus Amazon S3.

Reference:
http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_NAT_Instance.html

Question #416

When using Amazon CloudTrail to record API calls, the following information is returned for services with regional end points: ____.

- A. captured and processed in the same region as to which the API call is made and delivered to the region associated with your Amazon S3 bucket
- B. captured, processed, and delivered to the region associated with your Amazon S3 bucket
- C. captured in the same region as to which the API call is made and processed and delivered to the region associated with your Amazon S3 bucket
- D. captured in the region where the end point is located, processed in the region where the CloudTrail trail is configured, and delivered to the region associated with your Amazon S3 bucket

Commented [LC359]: When logging with Amazon CloudTrail, API call information for services with regional end points (EC2, RDS etc.) is captured and processed in the same region as to which the API call is made and delivered to the region associated with your Amazon S3 bucket. API call information for services with single end points (IAM, STS etc.) is captured in the region where the end point is located, processed in the region where the CloudTrail trail is configured, and delivered to the region associated with your Amazon S3 bucket.

Reference:
<https://aws.amazon.com/cloudtrail/faqs/>

Question #417

Currently, the project you're working on deploys its AWS infrastructure using a single AWS CloudFormation template, which supports a multi-tier web application. You've been entrusted with the responsibility of structuring the AWS CloudFormation resources in such a way that they can be maintained in the future and that other departments such as Networking and Security may examine the architecture prior to it being deployed to Production.

How do you accomplish this while accommodating each department's current workflows?

- A. Organize the AWS CloudFormation template so that related resources are next to each other in the template, such as VPC subnets and routing rules for Networking and security groups and IAM information for Security.
- B. Separate the AWS CloudFormation template into a nested structure that has individual templates for the resources that are to be governed by different departments, and use the outputs from the networking and security stacks for the application template that you control.
- C. Organize the AWS CloudFormation template so that related resources are next to each other in the template for each department's use, leverage your existing continuous integration tool to constantly deploy changes from all parties to the Production environment, and then run tests for validation.
- D. Use a custom application and the AWS SDK to replicate the resources defined in the current AWS CloudFormation template, and use the existing code review system to allow other departments to approve changes before altering the application for future deployments.

Commented [LC360]:

Question #418

A user has generated a new EBS volume using a snapshot of an existing volume. The user mounts the volume on the associated instance.

Which of the following is a prerequisite for the user to mount the volume?

- A. Run a cyclic check on the device for data consistency
- B. Create the file system of the volume
- C. Resize the volume as per the original snapshot size
- D. No step is required. The user can directly mount the device.

Commented [LC361]: Option-D.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html>

Determine whether there is a file system on the volume. New volumes are raw block devices, and you must create a file system on them before you can mount and use them. Volumes that were created from snapshots likely have a file system on them already; if you create a new file system on top of an existing file system, the operation overwrites your data.

Question #419

A firm is developing a web and mobile application using AWS Lambda and Amazon API Gateway in a serverless architecture. The organization wants to completely automate the deployment of backend Lambda services based on code uploaded to the relevant environment branch in an AWS CodeCommit repository.

The deployment must include the following components:

- ⇒ Separate pipelines for testing and production environments;
- ⇒ Automatic deployment that occurs for test environments only.

What efforts should be done to ensure compliance with these requirements?

- A. Configure a new AWS CodePipeline service. Create a CodeCommit repository for each environment. Set up CodePipeline to retrieve the source code from the appropriate repository. Set up the deployment step to deploy the Lambda functions with AWS CloudFormation.
- B. Create two AWS CodePipeline configurations for test and production environments. Configure the production pipeline to have a manual approval step. Create a CodeCommit repository for each environment. Set up each CodePipeline to retrieve the source code from the appropriate repository. Set up the deployment step to deploy the Lambda functions with AWS CloudFormation.
- C. Create two AWS CodePipeline configurations for test and production environments. Configure the production pipeline to have a manual approval step. Create one CodeCommit repository with a branch for each environment. Set up each CodePipeline to retrieve the source code from the appropriate branch in the repository. Set up the deployment step to deploy the Lambda functions with AWS CloudFormation.
- D. Create an AWS CodeBuild configuration for test and production environments. Configure the production pipeline to have a manual approval step. Create one CodeCommit repository with a branch for each environment. Push the Lambda function code to an Amazon S3 bucket. Set up the deployment step to deploy the Lambda functions from the S3 bucket.

Commented [LC362]: C. the difference between B & C is a single repository with a branch or two repositories. The requirement states "The company wants to fully automate the backend Lambda deployment based on code that is pushed to the appropriate environment branch in an AWS CodeCommit repository." Also, the requirement states that it needs two pipelines

Question #420

A DevOps Engineer is responsible for monitoring the health of a stateless RESTful service behind a Classic Load Balancer. A CI/CD pipeline is used to deliver new application versions. If the latency of the service exceeds a predefined threshold, deployment should be halted until the service recovers.

Which of the following approaches enables the SPEEDIEST detection time?

- A. Use Amazon CloudWatch metrics provided by Elastic Load Balancing to calculate average latency. Alarm and stop deployment when latency increases beyond the defined threshold.
- B. Use AWS Lambda and Elastic Load Balancing access logs to detect average latency. Alarm and stop deployment when latency increases beyond the defined threshold.
- C. Use AWS CodeDeploy's MinimumHealthyHosts setting to define thresholds for rolling back deployments. If these thresholds are breached, roll back the deployment.
- D. Use Metric Filters to parse application logs in Amazon CloudWatch Logs. Create a filter for latency. Alarm and stop deployment when latency increases beyond the defined threshold.

Commented [LC363]: A – correct

B - this might work but has additional overhead of a lambda and will depend on how frequently lambda is run. Although minimal but additional cost of lambda. This won't give QUICKEST detection time.

C - MinimumHealthyHosts may not be directly correlated with latency. Latency might be more due to network or other issues even if 100% of hosts are healthy.

D - why do this when a ready made option is available

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-cloudwatch-metrics.html>

Question #421

A DevOps engineer is entrusted with transferring a workload's Docker containers to AWS. The solution must enable automated deployment of changes to development and test environments by updating each container and registering it with a container registry. Once the containers have been pushed, they must be automatically deployed.

Which solution will satisfy these criteria?

- A. Store container images in Amazon S3. Run the containers in AWS Elastic Beanstalk using a multicontainer Docker environment. Configure Elastic Beanstalk to redeploy the containers if it detects a new version in Amazon S3.
- B. Store container images in AWS Artifact. Use AWS CodePipeline to trigger a deployment if a new container version is created. Use AWS CodeDeploy to deploy new containers to Amazon EKS.
- C. Store container images in Amazon ECR. Use AWS CodePipeline to trigger a deployment if a new container version is created. Use AWS CodeDeploy to deploy the image to AWS Fargate.
- D. Store container images in Docker Hub. Install Docker on an Amazon EC2 instance and use AWS CodePipeline and AWS CodeDeploy to deploy any new containers.

Commented [LC364]: <https://docs.aws.amazon.com/codepipeline/latest/userguide/action-reference-ECR.html>

Question #422

You need to migrate ten million records into DynamoDB in one hour. Each record is 1.5KB in length. Data is dispersed uniformly across the partition key.

How many units of writing capacity should you allocate for this batch load?

- A. 6667
- B. 4166
- C. 5556
- D. 2778

Commented [LC365]: You need 2 units to make a 1.5KB write, since you round up. You need 20 million total units to perform this load. You have 3600 seconds to do so. Divide and round up for 5556.

Reference:

<http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ProvisionedThroughput.html>

Question #423

A DevOps engineer is constructing a centralized continuous integration/continuous deployment pipeline utilizing AWS CodeBuild, AWS CodeDeploy, and Amazon S3. The engineer is expected to have least privilege access and to encrypt all objects in Amazon S3 on an individual basis. The engineer must be able to prune older artifacts without downloading or reading them. The engineer has already performed the steps described below:

1. Created a unique AWS Key Management Service (AWS KMS) CMK and S3 bucket for each project's builds using the AWS Key Management Service (AWS KMS).
2. Updated the S3 bucket policy to enable only uploads that are encrypted using the linked KMS.

Which last action is necessary to satisfy these requirements?

- A. Update the attached IAM policies to allow access to the appropriate KMS key from the CodeDeploy role where the application will be deployed.
- B. Update the attached IAM policies to allow access to the appropriate KMS key from the EC2 instance roles where the application will be deployed.
- C. Update the CMK's key policy to allow access to the appropriate KMS key from the CodeDeploy role where the application will be deployed.
- D. Update the CMK's key policy to allow access to the appropriate KMS key from the EC2 instance roles where the application will be deployed.

Commented [LC366]: KMS is used for CodeDeploy to decrypt file in S3, so this should be added to the CodeDeploy service role policy: { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["kms:DescribeKey", "kms:GenerateDataKey*", "kms:Encrypt", "kms:ReEncrypt*", "kms:Decrypt"], "Resource": ["arn:aws:kms:us-east-1:XXXXXXX:key/XXXXXXXXXX"] }] }

Question #424

You're developing an Amazon Web Services CloudFormation template for a multi-tier web application. Your Linux web server resource's user data includes a sophisticated script that may take a long time to execute.

Which procedures might you use to validate that these servers are completely setup and operational prior to connecting them to the load balancer? (Select two.)

- A. Launch your Linux servers from a nested stack that is called from within the load balancer resource in your AWS CloudFormation template.
- B. Add an AWS CloudFormation Wait Condition that depends on the web server resource. When the UserData script finishes on the web servers, use curl to send a signal the Wait Condition at <http://169.254.169.254/waithandle/>.
- C. Add an AWS CloudFormation wait Condition that depends on the web server resource. When the UserData script finishes on the web servers, use curl to signal to the Wait Condition pre-signed URL that they are ready.
- D. In your AWS CloudFormation template, position the load balancer resource JSON block directly below your Linux server resource.
- E. Add an AWS CloudFormation Wait Condition that depends on the web server resource. When the UserData script finishes on the web servers, use the command "cfn-signal" to signal that they are ready.

Commented [LC367]: Ans is CE

A: nested stack is not going to solve this dependency issue so this is wrong

B: <http://169.254.169.254/waithandle/> is not a valid url for sending the signal back to the wait condition handler

C: Curl is one way of signaling the wait condition handler
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-waitcondition.html>

D: Resource in CF template is not executed in order so this is wrong E: cfn-signal helper script is another way to signal the wait condition handler
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-waitcondition.html>

Commented [LC368]:

Question #425 [SKIP]

Which distinction between core and optional modules is incorrect?

- A. Extra modules may one day become core modules
- B. Core modules are supported by the Ansible team
- C. Core modules are shipped by default with Ansible
- D. Extra modules have no support

Question #426

A corporation learns that certain IAM users have been putting their AWS access keys in configuration files that have been uploaded to a service that hosts Git repositories.

Which option requires the LEAST amount of administration overhead while preventing unauthorized usage of the disclosed AWS access keys?

- A. Build an application that will create a list of all AWS access keys in the account and search each key on Git repository hosting services. If a match is found, configure the application to disable the associated access key. Then deploy the application to an AWS Elastic Beanstalk worker environment and define a periodic task to invoke the application every hour.
- B. Use Amazon Inspector to detect when a key has been exposed online. Have Amazon Inspector send a notification to an Amazon SNS topic when a key has been exposed. Create an AWS Lambda function subscribed to the SNS topic to disable the IAM user to whom the key belongs, and then delete the key so that it cannot be used.
- C. Configure AWS Trusted Advisor and create an Amazon CloudWatch Events rule that uses Trusted Advisor as the event source. Configure the CloudWatch Events rule to invoke an AWS Lambda function as the target. If the Lambda function finds the exposed access keys, then have it disable the access key so that it cannot be used.
- D. Create an AWS Config rule to detect when a key is exposed online. Have AWS Config send change notifications to an SNS topic. Configure an AWS Lambda function that is subscribed to the SNS topic to check the notification sent by AWS Config, and then disable the access key so it cannot be used.

Commented [LC369]: <https://github.com/aws/Trusted-Advisor-Tools/tree/master/ExposedAccessKeys/stepbystep>

Question #427

A programmer is developing an application that will enable users to submit photographs to an Amazon S3 bucket. Users must be able to sign in to the application using their Facebook credentials in order to post photographs.

How are these stipulations to be met?

- A. Store a user's Facebook user name and password in an Amazon DynamoDB table. Authenticate against those credentials the next time the user tries to log in.
- B. Create an Amazon Cognito identity pool using Facebook as the identity provider. Obtain temporary AWS credentials so a user can access Amazon S3.
- C. Create multiple AWS IAM users. Set the email and password to be the same as each user's Facebook login credentials.
- D. Create a new Facebook account and store its login credentials in an S3 bucket. Share that S3 bucket with a user. The user will log in to the application using those retrieved credentials.

Commented [LC370]:

Question #428

AWS CloudFormation was used to construct a three-tier web application that utilizes an Amazon RDS MySQL Multi-AZ database instance. A DevOps Engineer is responsible for upgrading the RDS instance to the newest major version of MySQL with the least amount of downtime possible.

How should the Engineer update the instance with the least amount of downtime possible?

- A. Update the EngineVersion property of the AWS::RDS::DBInstance resource type in the CloudFormation template to the latest desired version. Launch a second stack and make the new RDS instance a read replica.
- B. Update the DBEngineVersion property of the AWS::RDS::DBInstance resource type in the CloudFormation template to the latest desired version. Perform an Update Stack operation. Create a new RDS Read Replicas resource with the same properties as the instance to be upgraded. Perform a second Update Stack operation.
- C. Update the DBEngineVersion property of the AWS::RDS::DBInstance resource type in the CloudFormation template to the latest desired version. Create a new RDS Read Replicas resource with the same properties as the instance to be upgraded. Perform an Update Stack operation.
- D. Update the EngineVersion property of the AWS::RDS::DBInstance resource type in the CloudFormation template to the latest version, and perform an Update Stack operation.

Commented [LC371]: My vote is A: to minimize downtime, creating a second stack with the updated SQL version and then making it a read replica of the existing DB deployed would accomplish this.

Question #429

Each time an n-tier application is deployed, a table in an Amazon RDS MySQL DB instance must be discarded and repopulated. This procedure may take several minutes, and the web tier will not be available until it is complete. The web tier is currently setup as an Amazon EC2 Auto Scaling group, with instances terminated and replaced on a per-deployment basis. The MySQL table is filled by an AWS CodeBuild process that does a SQL query.

What should be done to prevent the web tier from being activated before the database has been entirely configured?

- A. Use Amazon Aurora as a drop-in replacement for RDS MySQL. Use snapshots to populate the table with the correct data.
- B. Modify the launch configuration of the Auto Scaling group to pause user data execution for 600 seconds, allowing the table to be populated.
- C. Use AWS Step Functions to monitor and maintain the state of data population. Mark the database in service before continuing with the deployment.
- **D. Use an EC2 Auto Scaling lifecycle hook to pause the configuration of the web tier until the table is populated.**

Commented [LC372]: D. Lifecycle hook can trigger lambda to check DB status:
<https://docs.aws.amazon.com/autoscaling/ec2/userguide/configuring-lifecycle-hook-notifications.html>

Question #430

You demand a high level of protection for your AWS accounts.

What is the most efficient and advanced configuration you can use to respond to AWS API requests sent to your account?

- A. Subscription to AWS Config via an SNS Topic. Use a Lambda Function to perform in-flight analysis and reactivity to changes as they occur.
- **B. Global AWS CloudTrail setup delivering to S3 with an SNS subscription to the deliver notifications, pushing into a Lambda, which inserts records into an ELK stack for analysis.**
- C. Use a CloudWatch Rule ScheduleExpression to periodically analyze IAM credential logs. Push the deltas for events into an ELK stack and perform ad-hoc analysis there.
- D. CloudWatch Events Rules which trigger based on all AWS API calls, submitting all events to an AWS Kinesis Stream for arbitrary downstream analysis.

Commented [LC373]: Best answer. ELK = Elasticsearch, LogStash, Kibana

Question #431

A business is using AWS to host an application. The development team's deployments must be automated. The team has configured an AWS CodePipeline to automatically deploy the application to Amazon EC2 instances through AWS CodeDeploy once it has been developed using the AWS CodeBuild service.

The team wants to include automated testing into the pipeline in order to ensure that the application is healthy prior to delivering it to the next stage using the same code. Even if the test is successful, the team needs human permission before the application can be deployed. Testing and approval must be carried out at the lowest possible cost and with the simplest management solution possible.

Which solution will satisfy these criteria?

- A. Add a manual approval action after the last deploy action of the pipeline. Use Amazon SNS to inform the team of the stage being triggered. Next, add a test action using CodeBuild to do the required tests. At the end of the pipeline, add a deploy action to deploy the application to the next stage.
- **B. Add a test action after the last deploy action of the pipeline. Configure the action to use CodeBuild to perform the required tests. If these tests are successful, mark the action as successful. Add a manual approval action that uses Amazon SNS to notify the team, and add a deploy action to deploy the application to the next stage.**
- C. Create a new pipeline that uses a source action that gets the code from the same repository as the first pipeline. Add a deploy action to deploy the code to a test environment. Use a test action using AWS Lambda to test the deployment. Add a manual approval action by using Amazon SNS to notify the team, and add a deploy action to deploy the application to the next stage.
- D. Add a test action after the last deployment action. Use a Jenkins server on Amazon EC2 to do the required tests and mark the action as successful if the tests pass. Create a manual approval action that uses Amazon SQS to notify the team and add a deploy action to deploy the application to the next stage.

Commented [LC374]: "The team requires a manual approval action before the application is deployed, even if the test is successful." So the approval action is before the deployment not the tests. Answer B is the right one here.

Question #432 [WRONG]

Which of the following is not a built-in feature of AWS CloudFormation?

- A. Fn::Split
- B. Fn::FindInMap
- C. Fn::Select
- D. Fn::GetAZs

Commented [LC375]: They all exist now

Question #433

A development team use AWS CodeCommit to manage source code. When modifications are ready for production, developers apply them to different feature branches and make pull requests to move them to the master branch. Direct commits to the main branch should be prohibited. The team updated the AWS managed policy AWSCodeCommitPowerUser to the Developers' IAM Role, but members may now push straight to the master branch on any repository in the AWS account.

What measures should be made to curb this?

- A. Create an additional policy to include a deny rule for the codecommit:GitPush action, and include a restriction for the specific repositories in the resource statement with a condition for the master reference.
- B. Remove the IAM policy and add an AWSCodeCommitReadOnly policy. Add an allow rule for the codecommit:GitPush action for the specific repositories in the resource statement with a condition for the master reference.
- C. Modify the IAM policy and include a deny rule for the codecommit:GitPush action for the specific repositories in the resource statement with a condition for the master reference.
- D. Create an additional policy to include an allow rule for the codecommit:GitPush action and include a restriction for the specific repositories in the resource statement with a condition for the feature branches reference.

Commented [LC376]: I'll go with A.

B) Wrong because is too restrictive.
C) Wrong because you can't modify a Managed Policy.
D) Wrong because AWSCodeCommitPowerUser will overwrite the additional policy anyway.

Question #434

An Amazon EC2 instance operating in a Virtual Private Cloud (VPC) without internet access needs to download an item from a restricted Amazon S3 bucket. When the DevOps Engineer attempts to access the item, he receives an AccessDenied error.

What may be the reason of this error? (Select three.)

- A. The S3 bucket default encryption is enabled.
- B. There is an error in the S3 bucket policy.
- C. There is an error in the VPC endpoint policy.
- D. The object has been moved to Amazon Glacier.
- E. There is an error in the IAM role configuration.
- F. S3 versioning is enabled.

Commented [LC377]:

Commented [LC378]:

Commented [LC379]:

Question #435 [SKIP]

You have a playbook that contains a job for installing a package for a service, storing the package's configuration file on the system, and restarting the service. After then, the playbook is run twice in a row.

What are your expectations for Ansible's second run?

- A. Remove the old package and config file and reinstall and then restart the service.
- B. Take no action on the target host.
- C. Check if the package is installed, check if the file matches the source file, if not reinstall it; restart the service.
- D. Attempt to reinstall the package, copy the file and restart the service.

Question #436

Which automation approach should you use if you want to wait for a CloudFormation stack to complete in a script?

- A. Event subscription using SQS.
- B. Event subscription using SNS.
- C. Poll using `ListStacks` / `list-stacks`
- D. Poll using `GetStackStatus` / `get-stack-status`

Commented [LC380]: Event driven systems are good for IFTTT logic, but only polling will make a script wait to complete. ListStacks / list-stacks is a real method, GetStackStatus / get-stack-status is not.

Reference:
<http://docs.aws.amazon.com/cli/latest/reference/cloudformation/list-stacks.html>

Question #437

A business wants to use Amazon DynamoDB to store metadata for its forums. The following picture depicts a sample data set.

Thread			
ForumName	Subject	LastPostDateTime	Thread
"S3"	"aaa"	"2015-03-15:17:24:31"	12
"S3"	"bbb"	"2015-01-22:23:18:01"	3
"S3"	"ccc"	"2015-02-31:13:14:21"	4
"S3"	"ddd"	"2015-01-03:09:21:11"	9
"EC2"	"yyy"	"2015-02-12:11:07:56"	18
"EC2"	"zzz"	"2015-01-18:07:33:42"	0
"RDS"	"ttt"	"2015-01-19:01:13:24"	3
"RDS"	"sss"	"2015-03-11:06:53:00"	11
"RDS"	"ttt"	"2015-10-22:12:19:44"	5

A DevOps Engineer must establish the table structure, which includes the partition key, the sort key, the local secondary index, projected attributes, and fetch operations.

The schema should handle the following sample searches while using the fewest supplied read capacity units possible to save money.

- Search inside ForumName for entries beginning with the letter 'a'.
- Seek for forums that have the specified LastPostDateTime.
- Return the thread value for which the LastPostDateTime is less than three months old.

Which schema satisfies the standards?

- A. Use Subject as the primary key and ForumName as the sort key. Have LSI with LastPostDateTime as the sort key and fetch operations for thread.
- B. Use ForumName as the primary key and Subject as the sort key. Have LSI with LastPostDateTime as the sort key and the projected attribute thread.
- C. Use ForumName as the primary key and Subject as the sort key. Have LSI with Thread as the sort key and the projected attribute LastPostDateTime.
- D. Use Subject as the primary key and ForumName as the sort key. Have LSI with Thread as the sort key and fetch operations for LastPostDateTime.

Commented [LC381]: A - most searches are based on forum, so it has to be primary key
B - looks correct
C - LastPostDateTime has to be the sort key for LSI
D - LastPostDateTime has to be the sort key for LSI

Question #438

A business has a web application that stores user information in an Amazon DynamoDB database in a single AWS Region. To serve an ever-growing worldwide user base, the application must operate in a secondary Region and enable users to connect to their nearest Region before failing over to the secondary Region.

Which technique should be used to guarantee that the deployment satisfies these criteria?

- A. Configure DynamoDB streams to copy data between Regions, deploy the web stack in both Regions, and configure Amazon Route 53 to use a geoproximity routing policy with health checks.
- B. Convert the DynamoDB table to a global table, deploy the web stack in both Regions, and configure Amazon Route 53 to use a geoproximity routing policy with health checks.
- C. Define DynamoDB cross-region backups to copy data to the secondary Region, deploy the web stack in both Regions, and configure Amazon Route 53 to use a latency-based routing policy with health checks.
- D. Use DynamoDB Accelerator to copy data to the secondary Region, deploy the web stack in both Regions, and configure Amazon Route 53 to use a failover routing policy.

Commented [LC382]: Reference:

<https://aws.amazon.com/blogs/database/how-to-use-amazon-dynamodb-global-tables-to-power-multiregion-architectures/>

Question #439 [SKIP]

You're running a Docker daemon on a Linux system and it suddenly stops responding.

Which signal, when used with the kill command to terminate a Docker process, requires the whole stack trace to be reported for debugging purposes?

- A. "TRACE
- B. "IOTRACE
- C. -SIGUSER1
- D. "KILLTRACE

Question #440

You have a huge number of web servers clustered behind a load balancer in an Auto Scaling group. You wish to filter and analyze the logs on an hourly basis to gather data on unique visitors and then store that data in a durable data storage for reporting purposes. While web servers in the Auto Scaling group are continually being launched and terminated in accordance with your scaling rules, you do not want to lose any of their log data during a stop/termination triggered by a user or by Auto Scaling.

Which of the following two ways will satisfy these requirements? (Select two.)

- A. Install an Amazon Cloudwatch Logs Agent on every web server during the bootstrap process. Create a CloudWatch log group and define Metric Filters to create custom metrics that track unique visitors from the streaming web server logs. Create a scheduled task on an Amazon EC2 instance that runs every hour to generate a new report based on the Cloudwatch custom metrics.
- B. On the web servers, create a scheduled task that executes a script to rotate and transmit the logs to Amazon Glacier. Ensure that the operating system shutdown procedure triggers a logs transmission when the Amazon EC2 instance is stopped/terminated. Use Amazon Data Pipeline to process the data in Amazon Glacier and run reports every hour.
- C. On the web servers, create a scheduled task that executes a script to rotate and transmit the logs to an Amazon S3 bucket. Ensure that the operating system shutdown procedure triggers a logs transmission when the Amazon EC2 instance is stopped/terminated. Use AWS Data Pipeline to move log data from the Amazon S3 bucket to Amazon Redshift in order to process and run reports every hour.
- D. Install an AWS Data Pipeline Logs Agent on every web server during the bootstrap process. Create a log group object in AWS Data Pipeline, and define Metric Filters to move processed log data directly from the web servers to Amazon Redshift and run reports every hour.

Commented [LC383]: Ans is A, C:

B and D cannot be right.

B: because of Glacier.

D: because no such thing as "Data pipeline logs agent"

Commented [LC384]:

Question #441

You have a high-traffic application that is operating behind a load balancer and has very latency-sensitive customers.

How do you discover which Amazon Elastic Compute Cloud application instances on the backend are causing the increased latency and should be replaced?

- A. By using the Elastic Load Balancing Latency CloudWatch metric.
- B. By using the HTTP X-Forwarded-For header for requests from the load balancer.
- C. By running a distributed load test to the load balancer.
- D. By using the load balancer access logs.

Commented [LC385]: D as Access Logs give you this instance latency info per call to each node in Target Group

<https://aws.amazon.com/premiumsupport/knowledge-center/elb-latency-troubleshooting/>

Question #442 [SKIP]

What does a Docker swarm worker node do?

- A. scheduling services
- B. service swarm node HTTP API endpoints
- C. executing containers
- D. maintaining cluster state

Question #443 [SKIP]

Which of the following is NOT a benefit of the content addressable storage strategy used by Docker?

- A. random UUIDs improve filesystem performance
- B. improved security
- C. guarantees data integrity after push, pull, load, and save operations
- D. avoids content ID collisions

Question #444

You want to send queue messages that are each 1GB in size.

How are you to do this?

- A. Use Kinesis as a buffer stream for message bodies. Store the checkpoint id for the placement in the Kinesis Stream in SQS.
- B. Use the Amazon SQS Extended Client Library for Java and Amazon S3 as a storage mechanism for message bodies.
- C. Use SQS's support for message partitioning and multi-part uploads on Amazon S3.
- D. Use AWS EFS as a shared pool storage medium. Store filesystem pointers to the files on disk in the SQS message bodies.

Commented [LC386]: You can manage Amazon SQS messages with Amazon S3. This is especially useful for storing and retrieving messages with a message size of up to 2 GB. To manage Amazon SQS messages with Amazon S3, use the Amazon SQS Extended Client Library for Java.

Reference:
<http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/s3-messages.html>

Question #445 [SKIP]

When Ansible's connection status is set to 'remote,' how does Ansible communicate with the remote target host?

- A. SSH
- B. RSH
- C. PSEXec
- D. API call to Ansible client on host

Question #446

A production account requires that each Amazon EC2 instance that has been manually logged into be destroyed within 24 hours. All apps in the production account make use of Auto Scaling groups that are setup with an Amazon CloudWatch Logs agent.

How can we automate this process?

- A. Create a CloudWatch Logs subscription to an AWS Step Functions application. Configure the function to add a tag to the EC2 instance that produced the login event and mark the instance to be decommissioned. Then create a CloudWatch Events rule to trigger a second AWS Lambda function once a day that will terminate all instances with this tag.
- B. Create a CloudWatch alarm that will trigger on the login event. Send the notification to an Amazon SNS topic that the Operations team is subscribed to, and have them terminate the EC2 instance within 24 hours.
- C. Create a CloudWatch alarm that will trigger on the login event. Configure the alarm to send to an Amazon SQS queue. Use a group of worker instances to process messages from the queue, which then schedules the Amazon CloudWatch Events rule to trigger.
- D. Create a CloudWatch Logs subscription in an AWS Lambda function. Configure the function to add a tag to the EC2 instance that produced the login event and mark the instance to be decommissioned. Create a CloudWatch Events rule to trigger a daily Lambda function that terminates all instances with this tag.

Commented [LC387]: Option D is the only one that will work.

A: CW Logs subscriptions do not support Step Functions (only Kinesis Streams/Firehose, Lambda, + ES)
B/C: CW alarms trigger on a metric, not an event.

Question #447

You want to do automated, continuous integration tests at the application level on all members of an Auto Scaling group.

Which of the two alternatives should you choose?
(Select two.)

- A. Use the AWS SDK to run the DescribeInstances API call on the Auto Scaling group, and then iterate over the members and remotely connect to each Amazon EC2 instance and run the integration tests.
- B. Use the AWS SDK to run the DescribeAutoScalingInstances API call on the Auto Scaling Group, iterate over the members using the Describe Instances API call, remotely connect to each Amazon EC2 instance, and then run the integration tests.
- C. Set up a custom CloudWatch metric with the output of your integration tests that are run by a scheduled process on each instance, and then set up a CloudWatch alert for any failures.
- D. Using an Auto Cycle Group lifecycle policy, define a scheduled task to run integration tests when a new Amazon EC2 instance enters the InService state.
- E. Set up a custom CloudWatch metric that uses the output of the DescribeAutoScalingInstances API call to determine the HealthCheck status of the Amazon EC2 instances.
- F. Using the Auto Cycle Group lifecycle policy, define a scheduled task to run integration tests on individual instances using the Amazon EC2 user data to export test data to CloudWatch Logs.

Commented [LC388]: <https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroupLifecycle.html>

There is no InService lifecycle hook

B over A because of ref.

https://docs.aws.amazon.com/autoscaling/ec2/APIReference/API_DescribeAutoScalingInstances.html

Commented [LC389]:

Question #448

Your CTO is very concerned about the security of your Amazon Web Services account.

How can you effectively prevent hackers from entirely seizing control of your account?

- A. Use short but complex password on the root account and any administrators.
- B. Use AWS IAM Geo-Lock and disallow anyone from logging in except for in your city.
- C. Use MFA on all users and accounts, especially on the root account.
- D. Don't write down or remember the root account password after creating the AWS account.

Commented [LC390]: For increased security, we recommend that you configure multi-factor authentication (MFA) to help protect your AWS resources. MFA adds extra security because it requires users to enter a unique authentication code from an approved authentication device or SMS text message when they access AWS websites or services.

Reference:
http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa.html

Question #449

A legacy application of a business makes use of IAM user credentials to get access to resources in the business's AWS Organizations organization. A DevOps engineer must guarantee that no new IAM users are created unless the person who creates them is on an exemption list.

Which solution will satisfy these criteria?

- A. Attach an Organizations SCP with an explicit deny for all iam:CreateAccessKey actions with a condition that excludes StringNotEquals for aws:username with a value of the exception list.
- B. Attach an Organizations SCP with an explicit deny for all iam:CreateUser actions with a condition that includes StringNotLike for aws:username with a value of the exception list.
- C. Create an Amazon EventBridge (Amazon CloudWatch Events) rule with a pattern that matches the iam:CreateAccessKey action with an AWS Lambda function target. The function will check the user name account against an exception list. If the user is not in the exception list, the function will delete the user.
- D. Create an Amazon EventBridge (Amazon CloudWatch Events) rule with a pattern that matches the iam:CreateUser action with an AWS Lambda function target. The function will check the user name and account against an exception list. If the user is not in the exception list, the function will delete the user.

Commented [LC391]: I'll go with B.

A: iam:CreateAccessKey seems to be wrong, this is for access keys, not users.

B: At first I thought StringNotLike was wrong, and it did not exist, but it does exist!

"StringNotLike - Negated case-sensitive matching. The values can include a multi-character match wildcard (*) or a single-character match wildcard (?) anywhere in the string."

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html

Questions 450-499

Question #450

A DevOps engineer configured AD Connector using an AWS CloudFormation custom resource. Although the AWS Lambda function was performed and AD Connector was built, CloudFormation did not convert from CREATE IN PROGRESS to CREATE COMPLETE.

Which course of action should the engineer take to rectify this situation?

- A. Ensure the Lambda function code has exited successfully.
- **B. Ensure the Lambda function code returns a response to the pre-signed URL.**
- C. Ensure the Lambda function IAM role has cloudformation:UpdateStack permissions for the stack ARN.
- D. Ensure the Lambda function IAM role has ds:ConnectDirectory permissions for the AWS account.

Commented [LC392]: The answer is B
As noted in the CloudFormation documentation, CloudFormation expects your Lambda function to callback to it once it has completed its operation; CloudFormation will pause execution until this callback is received. The event sent to your Lambda function by CloudFormation contains the callback URL (ResponseURL)

Reference:
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/crpg-ref-responses.html>

Question #451

On Amazon Web Services (AWS), a firm is developing a solution for storing files containing Personally Identifiable Information (PII). The requirements indicate that:

- ⇒ All data must be encrypted both in transit and at rest.
- ⇒ All data must be duplicated in at least two sites with a minimum distance of 500 miles between them.

Which solution satisfies these criteria?

- A. Create primary and secondary Amazon S3 buckets in two separate Availability Zones that are at least 500 miles apart. Use a bucket policy to enforce access to the buckets only through HTTPS. Use a bucket policy to enforce Amazon S3 SSE-C on all objects uploaded to the bucket. Configure cross-region replication between the two buckets.
- **B. Create primary and secondary Amazon S3 buckets in two separate AWS Regions that are at least 500 miles apart. Use a bucket policy to enforce access to the buckets only through HTTPS. Use a bucket policy to enforce S3-Managed Keys (SSE-S3) on all objects uploaded to the bucket. Configure cross-region replication between the two buckets.**
- C. Create primary and secondary Amazon S3 buckets in two separate AWS Regions that are at least 500 miles apart. Use an IAM role to enforce access to the buckets only through HTTPS. Use a bucket policy to enforce Amazon S3-Managed Keys (SSE-S3) on all objects uploaded to the bucket. Configure cross-region replication between the two buckets.
- D. Create primary and secondary Amazon S3 buckets in two separate Availability Zones that are at least 500 miles apart. Use a bucket policy to enforce access to the buckets only through HTTPS. Use a bucket policy to enforce AWS KMS encryption on all objects uploaded to the bucket. Configure cross-region replication between the two buckets. Create a KMS Customer Master Key (CMK) in the primary region for encrypting objects.

Commented [LC393]: Reference:
<https://aws.amazon.com/blogs/security/iam-policies-and-bucket-policies-and-acls-oh-my-controlling-access-to-s3-resources/>

Question #452

A healthcare services provider is worried about the rising price of software licensing for a patient wellness program. The organization want to establish an auditing mechanism to guarantee that the application is only executed on Amazon EC2 Dedicated Hosts. A DevOps Engineer must establish a process for auditing the application and ensuring compliance.

What measures should the Engineer take to ensure that this need is met with the LEAST amount of administrative overhead possible?

- A. Use AWS Systems Manager Configuration Compliance. Use calls to the put-compliance-items API action to scan and build a database of noncompliant EC2 instances based on their host placement configuration. Use an Amazon DynamoDB table to store these instance IDs for fast access. Generate a report through Systems Manager by calling the list-compliance-summaries API action.
- B. Use custom Java code running on an EC2 instance. Set up EC2 Auto Scaling for the instance depending on the number of instances to be checked. Send the list of noncompliant EC2 instance IDs to an Amazon SQS queue. Set up another worker instance to process instance IDs from the SQS queue and write them to Amazon DynamoDB. Use an AWS Lambda function to terminate noncompliant instance IDs obtained from the queue, and send them to an Amazon SNS email topic for distribution.
- **C. Use AWS Config. Identify all EC2 instances to be audited by enabling Config Recording on all Amazon EC2 resources for the region. Create a custom AWS Config rule that triggers an AWS Lambda function by using the "config-rule-change-triggered" blueprint. Modify the Lambda evaluateComplianceO function to verify host placement to return a NON_COMPLIANT result if the instance is not running on an EC2 Dedicated Host. Use the AWS Config report to address noncompliant instances.**
- D. Use AWS CloudTrail. Identify all EC2 instances to be audited by analyzing all calls to the EC2 RunCommand API action. Invoke an AWS Lambda function that analyzes the host placement of the instance. Store the EC2 instance ID of noncompliant resources in an Amazon RDS MySQL DB instance. Generate a report by querying the RDS instance and exporting the query results to a CSV text file.

Commented [LC394]: Correct Answer: C

Reference:
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/dedicated-hosts-aws-config.html>
<https://aws.amazon.com/about-aws/whats-new/2015/11/use-aws-config-to-track-ec2-instances-on-dedicated-hosts-and-assess-license-compliance/>

Question #453

After providing a functioning proof of concept for a new application that utilizes AWS API Gateway, a developer must configure the project's team development environment. Due to the Developer's compressed timeframe, he or she wants to reduce time spent setting up infrastructure and would prefer to utilize the code repository developed for the proof of concept. All source code is currently kept in AWS CodeCommit. The company policy requires alpha, beta, and production phases, each with its own Jenkins server for building code and running tests. At any point, the Development Manager must be able to prevent code propagation between administrators. The Security team wants to ensure that users cannot alter the environment without authorization.

How is this possible?

- A. Create API Gateway alpha, beta, and production stages. Create a CodeCommit trigger to deploy code to the different stages using an AWS Lambda function.
- B. Create API Gateway alpha, beta, and production stages. Create an AWS CodePipeline that pulls code from the CodeCommit repository. Create CodePipeline actions to deploy code to the API Gateway stages.
- C. Create Jenkins servers for the alpha, beta, and production stages on Amazon EC2 instances. Create multiple CodeCommit triggers to deploy code to different stages using an AWS Lambda function.
- D. Create an AWS CodePipeline pipeline that pulls code from the CodeCommit repository. Create alpha, beta, and production stages with Jenkins servers on CodePipeline.

Question #454

A business operates an application that utilizes an AWS CodeDeploy deployment group and an Application Load Balancer. AWS CodePipeline is used to automate application deployments. It comprises of AWS CodeCommit as the source and AWS CodeDeploy as the deployment provider. Following a successful deployment, the application suffered several minutes of downtime until the deployment was manually turned back. A DevOps engineer checked that the pipeline was effective and showed no problems, but discovered that the code was causing the application to become unusable after several hours.

Which activities will assist in avoiding future downtime in similar circumstances? (Select two.)

- A. Configure a TCP health check for the Auto Scaling target group on a listening port of the application.
- B. Configure an HTTP or HTTPS health check for the Auto Scaling target group to check a specific application path.
- C. Create a script to test the application health and execute the script during the BeforeInstall lifecycle hook in the CodeDeploy appspec.yml file.
- D. Update the CodeDeploy deployment group to roll back automatically to the previous version if the deployment fails.
- E. Update the CodeDeploy deployment group to roll back based on a custom Amazon CloudWatch alarm using an application status metric.

Question #455

A DevOps Engineer is in the process of establishing a container-based architecture. The Engineer has chosen to automate the provisioning of an Amazon ECS cluster and an Amazon EC2 Auto Scaling group for the EC2 container instances using AWS CloudFormation. Following the successful construction of the CloudFormation stack, the Engineer found that, despite the fact that the ECS cluster and EC2 instances were generated successfully and the stack was completed, the EC2 instances were associating with a different cluster.

How should the DevOps Engineer remedy this problem by updating the CloudFormation template?

- A. Reference the EC2 instances in the AWS::ECS::Cluster resource and reference the ECS cluster in the AWS::ECS::Service resource.
- B. Reference the ECS cluster in the AWS::AutoScaling::LaunchConfiguration resource of the UserData property.
- C. Reference the ECS cluster in the AWS::EC2::Instance resource of the UserData property.
- D. Reference the ECS cluster in the AWS::CloudFormation::CustomResource resource to trigger an AWS Lambda function that registers the EC2 instances with the appropriate ECS cluster.

Commented [LC395]: I'll go with D.

References:

<https://plugins.jenkins.io/aws-codepipeline/>

<https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-four-stage-pipeline.html>

Commented [LC396]: BE, C is incorrect, how do you run test before the app is even installed??

<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html#:~:text=Use%20to%20run%20tasks%20before%20the%20replacement%20task%20set%20is%20created.>

Commented [LC397]:

Commented [LC398]: B, configure ecs.config file by using userdata:

```
ContainerInstances:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: !Ref 'ECSAMI'
    SecurityGroups: [{!Ref 'ContainerSecurityGroup'}]
    InstanceType: !Ref 'InstanceType'
    IamInstanceProfile: !Ref 'EC2InstanceProfile'
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash -xe
        echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
```

Commented [LC399]: B, configure ecs.config file by using userdata:

```
ContainerInstances:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: !Ref 'ECSAMI'
    SecurityGroups: [{!Ref 'ContainerSecurityGroup'}]
    InstanceType: !Ref 'InstanceType'
    IamInstanceProfile: !Ref 'EC2InstanceProfile'
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash -xe
        echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
```

Question #456

Locally, a developer tested an application before deploying it to AWS Lambda. While remote testing the application, the Lambda function returns an access denied error.

How can this problem be resolved?

- A. Update the Lambda function's execution role to include the missing permissions.
- B. Update the Lambda function's resource policy to include the missing permissions.
- C. Include an IAM policy document at the root of the deployment package and redeploy the Lambda function.
- D. Redeploy the Lambda function using an account with access to the AdministratorAccess policy.

Commented [LC400]: Reference:

<https://aws.amazon.com/premiumsupport/knowledge-center/access-denied-lambda-s3-bucket/>

Question #457

You have an application running on an Amazon EC2 instance and are securely accessing AWS Service APIs using IAM roles.

How can you setup your application running on that instance to obtain the AWS SDKs' API keys?

- A. When assigning an EC2 IAM role to your instance in the console, in the "Chosen SDK" dropdown list, select the SDK that you are using, and the instance will configure the correct SDK on launch with the API keys.
- B. Within your application code, make a GET request to the IAM Service API to retrieve credentials for your user.
- C. When using AWS SDKs and Amazon EC2 roles, you do not have to explicitly retrieve API keys, because the SDK handles retrieving them from the Amazon EC2 MetaData service.
- D. Within your application code, configure the AWS SDK to get the API keys from environment variables, because assigning an Amazon EC2 role stores keys in environment variables on launch.

Commented [LC401]: C is the correct answer.

<https://docs.aws.amazon.com/aws-sdk-php/v2/guide/credentials.html#instance-profile-credentials>

Question #458

A highly regulated organization has a policy prohibiting DevOps Engineers from accessing their Amazon EC2 instances unless in an emergency. If a DevOps Engineer does attempt to log in, the Security team must be alerted within 15 minutes.

Which solution will satisfy these criteria?

- A. Install the Amazon Inspector agent on each EC2 instance. Subscribe to Amazon CloudWatch Events notifications. Trigger an AWS Lambda function to check if a message is about user logins. If it is, send a notification to the Security team using Amazon SNS.
- B. Install the Amazon CloudWatch agent on each EC2 instance. Configure the agent to push all logs to Amazon CloudWatch Logs and set up a CloudWatch metric filter that searches for user logins. If a login is found, send a notification to the Security team using Amazon SNS.
- C. Set up AWS CloudTrail with Amazon CloudWatch Logs. Subscribe CloudWatch Logs to Amazon Kinesis. Attach AWS Lambda to Kinesis to parse and determine if a log contains a user login. If it does, send a notification to the Security team using Amazon SNS.
- D. Set up a script on each Amazon EC2 instance to push all logs to Amazon S3. Set up an S3 event to trigger an AWS Lambda function, which triggers an Amazon Athena query to run. The Athena query checks for logins and sends the output to the Security team using Amazon SNS.

Commented [LC402]: CloudWatch Logs agent runs on each EC2 instance. The agents are configured to send SSH logs from the EC2 instance to a log stream identified by an instance ID.

Log streams are aggregated into a log group. As a result, one log group contains all the logs you want to analyze from one or more instances.

You apply metric filters to a log group in order to search for specific keywords. When the metric filter finds specific keywords, the filter counts the occurrences of the keywords in a time-based sliding window. If the occurrence of a keyword exceeds the CloudWatch alarm threshold, an alarm is triggered.

An IAM policy defines a role that gives the EC2 servers permission to create logs in a log group and send log events (new log entries) from EC2 to log groups. This role is then assumed by the application servers.

CloudWatch alarms notify users when a specified threshold has been crossed. For example, you can set an alarm to trigger when more than 2 failed SSH connections happen in a 5-minute period.

<https://aws.amazon.com/blogs/security/how-to-monitor-and-visualize-failed-ssh-access-attempts-to-amazon-ec2-linux-instances/>

Question #459 [SKIP]

Which local address is used by the Docker DNS server?

- A. 127.0.0.1
- B. 127.0.0.111
- C. 127.0.0.254
- D. 127.0.0.11

Question #460

A retail business wants to utilize AWS Elastic Beanstalk to host its Java-based online sales website. Given that this will be a production website, the CTO has the following deployment plan requirements:

- ⇒ There is no downtime. While the deployment is in progress, the currently running Amazon EC2 instances should stay operational. No deployments or other actions should be conducted on Amazon EC2 instances, since they are used to handle production traffic.
- ⇒ Provision a new fleet of instances for deploying the new application version. After successfully deploying the new application version to the new fleet of instances, the new instances should be put into service and the old ones should be deleted.
- ⇒ The rollback procedure should be as straightforward as feasible. If the new fleet of instances does not succeed in deploying the new application version, they should be terminated and the existing instances should continue to serve traffic normally.
- ⇒ The environment's resources (EC2 Auto Scaling group, Elastic Load Balancing, and Elastic Beanstalk DNS CNAME) should stay same, and there should be no DNS changes.

Which deployment approach will be most effective in meeting the requirements?

- A. Use rolling deployments with a fixed amount of one instance at a time and set the healthy threshold to OK.
- B. Use rolling deployments with additional batch with a fixed amount of one instance at a time and set the healthy threshold to OK.
- C. launch a new environment and deploy the new application version there, then perform a CNAME swap between environments.
- **D. Use immutable environment updates to meet all the necessary requirements.**

Commented [LC403]: D looks right for me. C is not correct, as no DNS change is allowed

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environmentmgmt-updates-immutable.html>

Question #461

You are responsible for a popular file sharing service that distributes traffic to an Amazon EC2 application layer deployed in an Auto Scaling group that spans several Availability Zones through Elastic Load Balancing. You now track user file transfers in an application server log file and subsequently publish data points from the logs to an Amazon RDS MySQL instance. You are dissatisfied with the scaling behavior of your application and want to introduce a new scaling strategy based on the average number of user file transfers over a 10-minute period rather than the average CPU use over the past five minutes.

What actions should you take to guarantee that your application's tier scales appropriately in light of this new policy? (Select two.)

- A. Create a new CloudWatch alarm based on the Elastic Load Balancing "RequestCount" metric that triggers an Auto Scaling action to scale the application tier.
- B. Create a new CloudWatch alarm based on a custom metric streaming from the Amazon RDS MySQL instance that triggers an Auto Scaling action to scale the application tier.
- **C. Create a new CloudWatch alarm based on a custom metric published from file transfer logs streaming to CloudWatch that triggers an Auto Scaling action to scale the application tier.**
- **D. Create a new Auto Scaling launch configuration that includes an Amazon EC2 user data script that installs a CloudWatch Logs Agent on newly launched instances in the application tier. The agent will be configured to stream the file transfers log file to CloudWatch.**
- E. Create a new Auto Scaling launch configuration for the application tier that scales based on an Auto Scaling policy that reads the file transfer log data from the Amazon RDS MySQL instance.
- F. Create a new Auto Scaling launch configuration that includes an Amazon EC2 user data script that installs an Amazon RDS Logs Agent on newly launched instances in the application tier. The agent will be configured to stream the file transfer data points to the Auto Scaling group.

Commented [LC404]:

Commented [LC405]:

Commented [LC406]: I'll go with A
When you create a pipeline from CodePipeline during the step-by-step it creates a CloudWatch Event rule for a given branch and repo like this:

```
{
  "source": [
    "aws.codecommit"
  ],
  "detail-type": [
    "CodeCommit Repository State Change"
  ],
  "resources": [
    "arn:aws:codecommit:us-east-1:xxxx:repo-name"
  ],
  "detail": {
    "event": [
      "referenceCreated",
      "referenceUpdated"
    ],
    "referenceType": [
      "branch"
    ],
    "referenceName": [
      "master"
    ]
  }
}
```

Question #462

A development team uses AWS CodeCommit to manage application code version control and AWS CodePipeline to organize software deliveries. The team has chosen to utilize a remote master branch as the trigger for the pipeline. A developer committed code modifications to the CodeCommit repository but found that the pipeline remained idle for ten minutes.

Which of the following procedures should be followed to resolve this issue?

- **A. Check that an Amazon CloudWatch Events rule has been created for the master branch to trigger the pipeline.**
- B. Check that the CodePipeline service role has permission to access the CodeCommit repository.
- C. Check that the developer's IAM role has permission to push to the CodeCommit repository.
- D. Check to see if the pipeline failed to start because of CodeCommit errors in Amazon CloudWatch Logs.

Question #463

A DevOps Engineer is responsible for backing up sensitive Amazon S3 assets stored in an S3 bucket with a private bucket policy through the S3 cross-region replication capabilities. The items must be copied to a separate AWS Region and account.

What steps should be taken to allow replication? (Select three.)

- ☒ A. Create a replication IAM role in the source account.
- ☐ B. Create a replication IAM role in the target account.
- ☐ C. Add statements to the source bucket policy allowing the replication IAM role to replicate objects.
- ☒ D. Add statements to the target bucket policy allowing the replication IAM role to replicate objects.
- ☒ E. Create a replication rule in the source bucket to enable the replication.
- ☐ F. Create a replication rule in the target bucket to enable the replication.

Commented [LC407]:

Commented [LC408]:

Commented [LC409]:

Question #464

A business uses AWS Organizations to establish distinct AWS accounts for each of its divisions. It should be capable of automating the following tasks:

- ⇒ Periodically updating the Linux AMIs with fresh fixes and creating a golden image.
- ⇒ Installing a new version of the Chef agents in the golden image, if one is available
- ⇒ Enforcing the department's usage of freshly minted golden AMIs

Which of the following options has the LEAST management overhead?

- ☐ A. Write a script to launch an Amazon EC2 instance from the previous golden AMI, apply the patch updates, install the new version of the Chef agent, generate a new golden AMI, and then modify the AMI permissions to share only the new image with the departments' accounts.
- ☐ B. Use an AWS Systems Manager Run Command to update the Chef agent first, use Amazon EC2 Systems Manager Automation to generate an updated AMI, and then assume an IAM role to copy the new golden AMI into the departments' accounts.
- ☐ C. Use AWS Systems Manager Automation to update the Linux AMI using the previous image, provide the URL for the script that will update the Chef agent, and then use AWS Organizations to replace the previous golden AMI into the departments' accounts.
- ☒ D. Use AWS Systems Manager Automation to update the Linux AMI from the previous golden image, provide the URL for the script that will update the Chef agent, and then share only the newly generated AMI with the departments' accounts.

Commented [LC410]:

Question #465 [SKIP]

The operations and development teams need a centralized location for viewing both operating system and application logs.

How should you achieve this using Amazon Web Services (AWS) services? (Select two.)

- ☐ A. Using AWS CloudFormation, create a CloudWatch Logs LogGroup and send the operating system and application logs of interest using the CloudWatch Logs Agent.
- ☐ B. Using AWS CloudFormation and configuration management, set up remote logging to send events via UDP packets to CloudTrail.
- ☐ C. Using configuration management, set up remote logging to send events to Amazon Kinesis and insert these into Amazon CloudSearch or Amazon Redshift, depending on available analytic tools.
- ☐ D. Using AWS CloudFormation, create a CloudWatch Logs LogGroup. Because the Cloudwatch Log agent automatically sends all operating system logs, you only have to configure the application logs for sending off-machine.
- ☐ E. Using AWS CloudFormation, merge the application logs with the operating system logs, and use IAM Roles to allow both teams to have access to view console output from Amazon EC2.

Question #466

A business has developed a web service that is hosted on Amazon EC2 instances and is protected by an Application Load Balancer (ALB). The application has been deployed in us-east-1. Amazon Route 53 offers an external DNS service that directs traffic from example.com to the application, which has been developed with the necessary health checks.

In eu-west-1, the corporation has established a second environment for the application. The organization desires that traffic be directed to the environment that provides the quickest response time for each user. If one Region has an outage, traffic should be diverted to the other environment.

Which arrangement will meet these criteria?

- A.
- ⇒ A subdomain us.example.com with weighted routing: the US ALB is assigned a weight of two, whereas the EU ALB is assigned a weight of one.
 - ⇒ Another subdomain, eu.example.com, employs weighted routing, with the EU ALB assigned a weight of two and the US ALB assigned a weight of one.
 - ⇒ North America is aliased to us.example.com, while Europe is aliased to eu.example.com.

- B.
- ⇒ A subdomain us.example.com with latency-based routing: the first destination is the US ALB, and the second destination is the EU ALB.
 - ⇒ Another subdomain, eu.example.com, is configured with latency-based routing, with the EU ALB as the first destination and the US ALB as the second.
 - ⇒ Failover routing records for example.com with the first target set to us.example.com and the second target set to eu.example.com.

- C.
- ⇒ A subdomain us.example.com with failover routing: the primary ALB is in the United States, while the backup ALB is in the European Union.
 - ⇒ Another subdomain, eu.example.com, is configured with failover routing, with the EU ALB serving as the primary and the US ALB serving as the secondary.
 - ⇒ Routing records for example.com that are aliased to us.example.com and eu.example.com dependent on latency.

- D.
- ⇒ A subdomain us.example.com with multivalue answer routing, prioritizing the US ALB over the EU ALB.
 - ⇒ Another subdomain, eu.example.com, uses multivalue answer routing, prioritizing the EU ALB over the US ALB.
 - ⇒ Routing records for example.com that have been aliased to us.example.com and eu.example.com.

Commented [LC411]:

Question #467

A DevOps engineer is developing an AWS CloudFormation template to provision a web service on Amazon EC2 instances in a private subnet behind an ELB Application Load Balancer. The Engineer must guarantee that the service is capable of accepting requests from IPv6-addressed customers.

Which configuration settings in the CloudFormation template should the Engineer provide to enable IPv6 customers to access the web service?

- A. Associate an IPv6 CIDR block with the Amazon VPC and subnets where the EC2 instances will live. Create route table entries for the IPv6 network, use EC2 instance types that support IPv6, and assign IPv6 addresses to each EC2 instance.
- B. Replace the Application Load Balancer with a Network Load Balancer. Associate an IPv6 CIDR block with the Virtual Private Cloud (VPC) and subnets where the Network Load Balancer lives, and assign the Network Load Balancer an IPv6 Elastic IP address.
- C. Assign each EC2 instance an IPv6 Elastic IP address. Create a target group and add the EC2 instances as targets. Create a listener on port 443 of the Application Load Balancer, and associate the newly created target group as the default target group.
- D. Create a target group and add the EC2 instances as targets. Create a listener on port 443 of the Application Load Balancer. Associate the newly created target group as the default target group. Select a dual stack IP address, and create a rule in the security group that allows inbound traffic from anywhere.

Commented [LC412]: A is not the correct answer: "The load balancer communicates with targets using IPv4 addresses, regardless of how the client communicates with the load balancer. Therefore, the targets do not need IPv6 addresses."

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/application-load-balancers.html#load-balancer-address-type>

Question #468 [SKIP]

What steps must be taken to get remote access to a Docker daemon operating on Linux?

- A. add certificate authentication to the docker API
- B. change the encryption level to TLS
- C. enable the TCP socket
- D. bind the Docker API to a unix socket

Commented [LC413]: The Docker daemon can listen for Docker Remote API requests via three different types of Socket: unix, tcp, and fd. By default, a unix domain socket (or IPC socket) is created at `/var/run/docker.sock`, requiring either root permission, or docker group membership. If you need to access the Docker daemon remotely, you need to enable the tcp Socket. Beware that the default setup provides unencrypted and un-authenticated direct access to the Docker daemon - and should be secured either using the built in HTTPS encrypted socket or by putting a secure web proxy in front of it.

Reference:

<https://docs.docker.com/engine/reference/commandline/docker/#daemon-socket-option>

Question #469

A production fault has been detected, and a new sprint item has been established for the purpose of releasing a hotfix. However, any code modification must follow the following procedure: before the start of production:

- ⇒ Scan the code for security breaches, such as password and access key leaks.
- ⇒ Run the code through extensive, long-running unit tests.

Which source control approach, in conjunction with AWS CodePipeline, should a DevOps Engineer utilize to accomplish this process?

- A. Create a hotfix tag on the last commit of the master branch. Trigger the development pipeline from the hotfix tag. Use AWS CodeDeploy with Amazon ECS to do a content scan and run unit tests. Add a manual approval stage that merges the hotfix tag into the master branch.
- B. Create a hotfix branch from the master branch. Trigger the development pipeline from the hotfix branch. Use AWS CodeBuild to do a content scan and run unit tests. Add a manual approval stage that merges the hotfix branch into the master branch.
- C. Create a hotfix branch from the master branch. Trigger the development pipeline from the hotfix branch. Use AWS Lambda to do a content scan and run unit tests. Add a manual approval stage that merges the hotfix branch into the master branch.
- D. Create a hotfix branch from the master branch. Create a separate source stage for the hotfix branch in the production pipeline. Trigger the pipeline from the hotfix branch. Use AWS Lambda to do a content scan and use AWS CodeBuild to run unit tests. Add a manual approval stage that merges the hotfix branch into the master branch.

Commented [LC414]: The information should be on this page:

<https://aws.amazon.com/blogs/devops/implementing-gitflow-using-aws-codepipeline-aws-codecommit-aws-codebuild-and-aws-codedeploy/>

A is not it, since you should not tag but create a branch
C is not it, since a Lambda should not be used for unit testing
D is not it (I think) since the Lambda is not necessary (see: <https://aws.amazon.com/blogs/devops/identifying-and-resolving-vulnerabilities-in-your-code/>)

Question #470

Your development team need access to production instances at the account level in order to conduct live debugging in a highly secure environment.

Which of the following is the correct course of action?

- A. Place the credentials provided by Amazon Elastic Compute Cloud (EC2) into a secure Amazon Sample Storage Service (S3) bucket with encryption enabled. Assign AWS Identity and Access Management (IAM) users to each developer so they can download the credentials file.
- B. Place an internally created private key into a secure S3 bucket with server-side encryption using customer keys and configuration management, create a service account on all the instances using this private key, and assign IAM users to each developer so they can download the file.
- C. Place each developer's own public key into a private S3 bucket, use instance profiles and configuration management to create a user account for each developer on all instances, and place the user's public keys into the appropriate account.
- D. Place the credentials provided by Amazon EC2 onto an MFA encrypted USB drive, and physically share it with each developer so that the private key never leaves the office.

Commented [LC415]: Perhaps it's an outdated question. Looks like it comes from a 2016 batch.

Question #471

Your social media marketing program has a Ruby-based component that runs on AWS Elastic Beanstalk. This application component utilizes social media sites to promote different marketing strategies. Your management now asks that you record responses to these social media communications in order to measure the marketing campaign's efficacy in relation to previous and future efforts. You've already created a new application component that communicates with social networking site APIs in order to read the responses.

Which procedure should you use to save social media responses to a persistent data storage that can be accessed at any time for historical data analysis?

- A. Deploy the new application component in an Auto Scaling group of Amazon Elastic Compute Cloud (EC2) Instances, read the data from the social media sites, store it with Amazon Elastic Block Store, and use AWS Data Pipeline to publish it to Amazon Kinesis for analytics.
- **B. Deploy the new application component as an Elastic Beanstalk application, read the data from the social media sites, store it in Amazon DynamoDB, and use Apache Hive with Amazon Elastic MapReduce for analytics.**
- C. Deploy the new application component in an Auto Scaling group of Amazon EC2 instances, read the data from the social media sites, store it in Amazon Glacier, and use AWS Data Pipeline to publish it to Amazon Redshift for analytics.
- D. Deploy the new application component as an Amazon Elastic Beanstalk application, read the data from the social media site, store it with Amazon Elastic Block Store, and use Amazon Kinesis to stream the data to Amazon CloudWatch for analytics.

Commented [LC416]: Reference:

<https://aws.amazon.com/blogs/aws/aws-howto-using-amazon-elastic-mapreduce-with-dynamodb/>

Question #472

A business wishes to protect the security of its EC2 instances. They want to be alerted immediately if any new vulnerabilities are detected on their instances, and they also want an audit record of any login activity on the instances.

Which solution will satisfy these criteria?

- A. Use AWS Systems Manager to detect vulnerabilities on the EC2 instances. Install the Amazon Kinesis Agent to capture system logs and deliver them to Amazon S3.
- B. Use AWS Systems Manager to detect vulnerabilities on the EC2 instances. Install the Systems Manager Agent to capture system logs and view login activity in the CloudTrail console.
- C. Configure Amazon CloudWatch to detect vulnerabilities on the EC2 instances. Install the AWS Config daemon to capture system logs and view them in the AWS Config console.
- **D. Configure Amazon Inspector to detect vulnerabilities on the EC2 instances. Install the Amazon CloudWatch Agent to capture system logs and record them via Amazon CloudWatch Logs.**

Commented [LC417]:

Question #473

A corporation is deploying an application using Docker containers and want to migrate the application to AWS. Currently, the organization controls its own clusters on-premises for container deployment. It wants to deploy its application to an AWS managed service and wishes to automate the whole deployment process. Additionally, the firm requires the following:

- ☞ Focus first on the development workload.
- ☞ The environment must be easy to manage.
- ☞ Deployment should be repeatable and reusable for new environments.
- ☞ Store the code in a GitHub repository.

Which solution will satisfy these criteria?

- A. Set up an Amazon ECS environment. Use AWS CodePipeline to create a pipeline that is triggered on a commit to the GitHub repository. Use AWS CodeBuild to create the container images and AWS CodeDeploy to publish the container image to the ECS environment.
- **B. Use AWS CodePipeline that triggers on a commit from the GitHub repository, build the container images with AWS CodeBuild, and publish the container images to Amazon ECR. In the final stage, use AWS CloudFormation to create an Amazon ECS environment that gets the container images from the ECR repository.**
- C. Create a Kubernetes Cluster on Amazon EC2. Use AWS CodePipeline to create a pipeline that is triggered when the code is committed to the repository. Create the container images with a Jenkins server on EC2 and store them in the Docker Hub. Use AWS Lambda from the pipeline to trigger the deployment to the Kubernetes Cluster.
- D. Set up an Amazon ECS environment. Use AWS CodePipeline to create a pipeline that is triggered on a commit to the GitHub repository. Use AWS CodeBuild to create the container and store it in the Docker Hub. Use an AWS Lambda function to trigger a deployment and pull the new container image from the Docker Hub.

Commented [LC418]: I will go with B.

A - Incorrect, it mentions "CodeDeploy to publish the container image to the ECS environment", CodeDeploy Should publish container image to ECR

C - Incorrect, Using Kuberntes on EC2 and Jenkins server on EC2 violate stated requirement to "deploy its application to a managed service in AWS"

D - Incorrect, Code Build should create docker Image not container and the option doesn't offer repeatable deployment implementation

Question #474

Following a daily scrum with your development teams, you've determined that deploying in a Blue/Green fashion will help the team.

Which method should you use to fulfill this new requirement?

- **A. Re-deploy your application on AWS Elastic Beanstalk, and take advantage of Elastic Beanstalk deployment types.**
- B. Using an AWS CloudFormation template, re-deploy your application behind a load balancer, launch a new AWS CloudFormation stack during each deployment, update your load balancer to send half your traffic to the new stack while you test, after verification update the load balancer to send 100% of traffic to the new stack, and then terminate the old stack.
- C. Re-deploy your application behind a load balancer that uses Auto Scaling groups, create a new identical Auto Scaling group, and associate it to the load balancer. During deployment, set the desired number of instances on the old Auto Scaling group to zero, and when all instances have terminated, delete the old Auto Scaling group.
- D. Using an AWS OpsWorks stack, re-deploy your application behind an Elastic Load Balancing load balancer and take advantage of OpsWorks stack versioning, during deployment create a new version of your application, tell OpsWorks to launch the new version behind your load balancer, and when the new version is launched, terminate the old OpsWorks stack.

Commented [LC419]: Reference:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.CNAMESwap.html>

Question #475

A DevOps team need the ability to query information contained in application logs created by an application running numerous Amazon EC2 instances deployed using AWS Elastic Beanstalk. Elastic Beanstalk enables instance log streaming to Amazon CloudWatch Logs.

Which of the following approaches would be the MOST cost-effective?

- A. Use a CloudWatch Logs subscription to trigger an AWS Lambda function to send the log data to an Amazon Kinesis Data Firehose stream that has an Amazon S3 bucket destination. Use Amazon Athena to query the log data from the bucket.
- B. Use a CloudWatch Logs subscription to trigger an AWS Lambda function to send the log data to an Amazon Kinesis Data Firehose stream that has an Amazon S3 bucket destination. Use a new Amazon Redshift cluster and Amazon Redshift Spectrum to query the log data from the bucket.
- C. Use a CloudWatch Logs subscription to send the log data to an Amazon Kinesis Data Firehose stream that has an Amazon S3 bucket destination. Use Amazon Athena to query the log data from the bucket.
- D. Use a CloudWatch Logs subscription to send the log data to an Amazon Kinesis Data Firehose stream that has an Amazon S3 bucket destination. Use a new Amazon Redshift cluster and Amazon Redshift Spectrum to query the log data from the bucket.

Commented [LC420]: C: You now send directly from CloudWatch to Amazon Kinesis Data Firehose stream

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Subscriptions.html>

Question #476

What is the sequence of quickly scaling from most rapidly scaling to least rapidly scaling (fastest to scale first)?

- a) EC2 + ELB + Auto Scaling
- b) Lambda
- c) RDS

- A. B, A, C
- B. C, B, A
- C. C, A, B
- D. A, C, B

Commented [LC421]: Lambda is designed to scale instantly. EC2 + ELB + Auto Scaling require single-digit minutes to scale out. RDS will take at least 15 minutes, and will apply OS patches or any other updates when applied.

Reference:

<https://aws.amazon.com/lambda/faqs/>

Question #477

Users of an application notice issues immediately after Amazon API Gateway installations. The development team deploys once or twice daily, using a blue/green deployment approach that includes bespoke health checks and automatic rollbacks. The team want to keep the number of users impacted by deployment problems to a minimum and to be notified when rollbacks are required.

Which sequence of procedures should a DevOps engineer follow to satisfy these requests? (Select two.)

- A. Implement a blue/green strategy using path mappings.
- B. Implement a canary deployment strategy.
- C. Implement a rolling deployment strategy using multiple stages.
- D. Use Amazon CloudWatch alarms to notify the development team.
- E. Use Amazon CloudWatch Events to notify the development team.

Commented [LC422]: If the team wants to limit the number of users affected by deployment bugs, Canary deployments should be one option besides blue green I'll go with B) D)

The canary release will use cloudwatch metrics to measure failures and you'd likely want an alarm (threshold) on them to notify developers via SNS.

Reference:

<https://aws.amazon.com/blogs/compute/performing-canary-deployments-for-service-integrations-with-amazon-api-gateway/>

<https://docs.aws.amazon.com/apigateway/latest/developerguide/canary-release.html>

Commented [LC423]:

Commented [LC424]: The Query API for IAM and AWS STS lets you call service actions. Query API requests are HTTPS requests that must contain an Action parameter to indicate the action to be performed. IAM and AWS STS support GET and POST requests for all actions, that is, the API does not require you to use GET for some actions and POST for others.

Reference:

<http://docs.aws.amazon.com/IAM/latest/UserGuide/programming.html>

Question #478

You may directly access the AWS Security Token Service (STS) by making calls to the AWS STS Query API. This API is a web service interface that supports the _____ submission of queries.

- A. PUT
- B. HTTPS
- C. POST
- D. GET

Question #479

You're developing a Ruby on Rails application for internal, non-production usage that makes use of MySQL. You want developers with little AWS knowledge to be able to send new code to AWS using a single command line push. Additionally, you want to make things as simple as possible.

Which tool is optimal for this configuration?

- A. AWS CloudFormation
- B. AWS OpsWorks
- C. AWS ELB + EC2 with CLI Push
- **D. AWS Elastic Beanstalk**

Question #480

Which statement is correct about the configuration of the Amazon Inspector agent's proxy support on a Windows-based system?

- **A. Amazon Inspector agent supports proxy usage on Windows-based systems through the use of the WinHTTP proxy.**
- B. Amazon Inspector agent supports proxy usage on Linux-based systems but not on Windows.
- C. Amazon Inspector proxy support on Windows-based systems is achieved through installing proxy-enabled version of the agent which comes with preconfigured files that you need to edit to match your environment.
- D. Amazon Inspector agent supports proxy usage on Windows-based systems through awsagent.env configuration file.

Question #481

What is the first transition stage an existing instance enters after leaving steady state in Standby mode for AWS Auto Scaling?

- A. Detaching
- B. Terminating:Wait
- **C. Pending**
- D. EnteringStandby

Question #482

You now operate a web application on a collection of micro instance types inside a single AZ behind a single load balancer. You've established an Auto Scaling group to scale from two to sixty-four instances. When you examine your CloudWatch stats, you see that your Auto Scaling group sometimes runs 64 micro instances. The web application is reading and writing to a DynamoDB-enabled backend that has 800 Write Capacity Units and 800 Read Capacity Units set. Your consumers are complaining about the lengthy load times on your website. You have examined the DynamoDB CloudWatch metrics and determined that you are within the allocated Read and Write Capacity Units with no throttling.

How can you scale your service to enhance load times and adhere to high availability principles?

- A. Change your Auto Scaling group configuration to include multiple AZs.
- B. Change your Auto Scaling group configuration to include multiple AZs, and increase the number of Read Capacity Units in your DynamoDB table by a factor of three, because you will need to be calling DynamoDB from three AZs.
- C. Add a second load balancer to your Auto Scaling group so that you can support more inbound connections per second.
- **D. Change your Auto Scaling group configuration to use larger instances and include multiple AZ's instead of one.**

Commented [LC425]: Elastic Beanstalk's primary mode of operation exactly supports this use case out of the box. It is simpler than all the other options for this question. With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS cloud without worrying about the infrastructure that runs those applications. AWS Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

Reference:
http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_Ruby_rails.html

Commented [LC426]: https://docs.aws.amazon.com/inspector/latest/userguide/inspector_agents-on-win.html#inspector-agent-proxy

To get proxy support for an agent on a Windows-based operating system, use the WinHTTP proxy. To set up the WinHTTP proxy using the netsh utility, see Netsh Commands for Windows Hypertext Transfer Protocol (WINHTTP).

Commented [LC427]: You can put any instance that is in an InService state into a Standby state. This enables you to remove the instance from service, troubleshoot or make changes to it, and then put it back into service. Instances in a Standby state continue to be managed by the Auto Scaling group. However, they are not an active part of your application until you put them back into service.

Reference:
<http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/AutoScalingGroupLifecycle.html>

Commented [LC428]:

Question #483

A business is using AWS CodeDeploy to automate the deployment of a Java-Apache Tomcat application on an Apache webserver. The development team began with a proof of concept, then established a deployment group for a developer environment and conducted functional testing inside the application.

After that, the team will construct additional staging and production deployment groups.

The current log level is established in the Apache settings, but the team want to alter it dynamically during deployment, so that they may setup various log levels for different deployment groups without having to maintain a separate application revision for each group.

How can these objectives be accomplished with the LEAST amount of administration overhead and without the need for distinct script versions for each deployment group?

- A. Tag the Amazon EC2 instances depending on the deployment group. Then place a script into the application revision that calls the metadata service and the EC2 API to identify which deployment group the instance is part of. Use this information to configure the log level settings. Reference the script as part of the Afterinstall lifecycle hook in the appspec.yml file.
- B. Create a script that uses the CodeDeploy environment variable `DEPLOYMENT_GROUP_NAME` to identify which deployment group the instances are part of. Use this information to configure the log level settings. Reference this script as part of the BeforeInstall lifecycle hook in the appspec.yml file.
- C. Create a CodeDeploy custom environment variable for each environment. Then place a script into the application revision that checks this environment variable to identify which deployment group the instance is part of. Use this information to configure the log level settings. Reference this script as part of the ValidateService lifecycle hook in the appspec.yml file.
- D. Create a script that uses the CodeDeploy environment variable `DEPLOYMENT_GROUP_ID` to identify which deployment group the instance is part of to configure the log level settings. Reference this script as part of the Install lifecycle hook in the appspec.yml file.

Commented [LC429]: Correct Answer: B.

<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html>

<https://aws.amazon.com/blogs/devops/using-codedeploy-environment-variables/>

Question #484

A business has thousands of Amazon EC2 instances as well as hundreds of on-premises virtual machines. Developers log in to the console for on-premises systems on a regular basis to undertake troubleshooting. Developers want to log into AWS instances in order to run performance tools, but are unable to do so owing to the absence of a centralized console logging mechanism. A DevOps Engineer wants to verify that every console access on all systems is recorded.

Which combination of actions will satisfy these criteria? (Select two.)

- A. Attach a role to all AWS instances that contains the appropriate permissions. Create an AWS Systems Manager managed-instance activation. Install and configure Systems Manager Agent on on-premises machines.
- B. Enable AWS Systems Manager Session Manager logging to an Amazon S3 bucket. Direct Developers to connect to the systems with Session Manager only.
- C. Enable AWS Systems Manager Session Manager logging to AWS CloudTrail. Direct Developers to continue normal sign-in procedures for on-premises. Use Session Manager for AWS instances.
- D. Install and configure an Amazon CloudWatch Logs agent on all systems. Create an AWS Systems Manager managed-instance activation.
- E. Set up a Site-to-Site VPN connection between the on-premises and AWS networks. Set up a bastion instance to allow Developers to sign in to the AWS instances.

Commented [LC430]: A and B.

See: <https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager-logging.html>

Logging session data using Amazon S3

Commented [LC431]:

Question #485

AWS CloudTrail is used by the security team to identify sensitive security issues in the company's AWS account. The DevOps Engineer need a solution for automatically resolving CloudTrail being disabled in an AWS account.

Which option guarantees the MINIMUM amount of downtime for CloudTrail log delivery?

- A. Create an Amazon CloudWatch Events rule for the CloudTrail StopLogging event. Create an AWS Lambda function that uses the AWS SDK to call StartLogging on the ARN of the resource in which StopLogging was called. Add the Lambda function ARN as a target to the CloudWatch Events rule.
- B. Deploy the AWS-managed CloudTrail-enabled AWS Config rule, set with a periodic interval of 1 hour. Create an Amazon CloudWatch Events rule for AWS Config rules compliance change. Create an AWS Lambda function that uses the AWS SDK to call StartLogging on the ARN of the resource in which StopLogging was called. Add the Lambda function ARN as a target to the CloudWatch Events rule.
- C. Create an Amazon CloudWatch Events rule for a scheduled event every 5 minutes. Create an AWS Lambda function that uses the AWS SDK to call StartLogging on a CloudTrail trail in the AWS account. Add the Lambda function ARN as a target to the CloudWatch Events rule.
- D. Launch a t2.nano instance with a script running every 5 minutes that uses the AWS SDK to query CloudTrail in the current account. If the CloudTrail trail is disabled, have the script re-enable the trail.

Commented [LC432]: A

<https://aws.amazon.com/blogs/mt/monitor-changes-and-auto-enable-logging-in-aws-cloudtrail/>

Question #486

A business wishes to transfer a legacy application to AWS and create a deployment pipeline that is entirely based on AWS services. A DevOps engineer is transferring all application code from a Git repository to AWS CodeCommit while maintaining the repository's history. The DevOps engineer has configured all of CodeCommit's permissions, installed the Git client and the AWS CLI on a local machine, and is now prepared to migrate the repository.

Which further steps will be taken?

- A. Create the CodeCommit repository using the AWS CLI. Clone the Git repository directly to CodeCommit using the AWS CLI. Validate that the files were migrated, and publish the CodeCommit repository.
- B. Create the CodeCommit repository using the AWS Management Console. Clone both the Git and CodeCommit repositories to the local computer. Copy the files from the Git repository to the CodeCommit repository on the local computer. Commit the CodeCommit repository. Validate that the files were migrated, and share the CodeCommit repository.
- C. Create the CodeCommit repository using the AWS Management Console. Use the console to clone the Git repository into the CodeCommit repository. Validate that the files were migrated, and publish the CodeCommit repository.
- D. Create the CodeCommit repository using the AWS Management Console or the AWS CLI. Clone the Git repository with a mirror argument to the local computer and push the repository to CodeCommit. Validate that the files were migrated, and share the CodeCommit repository.

Commented [LC433]: Reference:

<https://docs.aws.amazon.com/codecommit/latest/userguide/how-to-migrate-repository-existing.html#how-to-migrate-existing-clone>

Question #487

Your CTO has directed you to ensure that you are always aware of what users of your AWS account are doing to modify resources. She wants a report on who is doing what over time, sent to her once a week, for the broadest feasible resource type group.

How should you proceed?

- A. Create a global AWS CloudTrail Trail. Configure a script to aggregate the log data delivered to S3 once per week and deliver this to the CTO.
- B. Use CloudWatch Events Rules with an SNS topic subscribed to all AWS API calls. Subscribe the CTO to an email type delivery on this SNS Topic.
- C. Use AWS IAM credential reports to deliver a CSV of all uses of IAM User Tokens over time to the CTO.
- D. Use AWS Config with an SNS subscription on a Lambda, and insert these changes over time into a DynamoDB table. Generate reports based on the contents of this table.

Commented [LC434]: This is the ideal use case for AWS CloudTrail. CloudTrail provides visibility into user activity by recording API calls made on your account. CloudTrail records important information about each API call, including the name of the API, the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS service. This information helps you to track changes made to your AWS resources and to troubleshoot operational issues. CloudTrail makes it easier to ensure compliance with internal policies and regulatory standards.

Reference:
<https://aws.amazon.com/cloudtrail/faqs/>

Question #488

A DevOps Engineer is responsible for architecting a continuous development strategy for a company's software as a service (SaaS) web application that is hosted on Amazon Web Services (AWS). Users subscribing to this application are split among numerous Application Load Balancers (ALBs) for application and security reasons. Each ALB has its own dedicated Auto Scaling group and fleet of Amazon EC2 instances. The application does not need a build step, and upon submission to AWS CodeCommit, it must initiate a simultaneous deployment to all ALBs, Auto Scaling groups, and EC2 fleets.

Which architecture will allow for the LEAST amount of setup to achieve these requirements?

- A. Create a single AWS CodePipeline pipeline that deploys the application in parallel using unique AWS CodeDeploy applications and deployment groups created for each ALB-Auto Scaling group pair.
- B. Create a single AWS CodePipeline pipeline that deploys the application using a single AWS CodeDeploy application and single deployment group.
- **C. Create a single AWS CodePipeline pipeline that deploys the application in parallel using a single AWS CodeDeploy application and unique deployment group for each ALB-Auto Scaling group pair.**
- D. Create an AWS CodePipeline pipeline for each ALB-Auto Scaling group pair that deploys the application using an AWS CodeDeploy application and deployment group created for the same ALB-Auto Scaling group pair.

Commented [LC435]: C.

Ref:
<https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-groups.html>

Question #489

You've created a Node.js server-side application and a web application with an HTML/JavaScript front end that utilizes the Angular.js framework. The server-side program establishes a connection to an Amazon Redshift cluster, executes queries, and then delivers the results to the front end for visualization. Although your user base is broad and geographically dispersed, it is critical to keep the cost of operating your program low.

Which deployment option is both technically feasible and economically viable?

- A. Deploy an AWS Elastic Beanstalk application with two environments: one for the Node.js application and another for the web front end. Launch an Amazon Redshift cluster, and point your application to its Java Database Connectivity (JDBC) endpoint.
- B. Deploy an AWS OpsWorks stack with three layers: a static web server layer for your front end, a Node.js app server layer for your server-side application, and a Redshift DB layer for your Amazon Redshift cluster.
- **C. Upload the HTML, CSS, images, and JavaScript for the front end to an Amazon Simple Storage Service (S3) bucket. Create an Amazon CloudFront distribution with this bucket as its origin. Use AWS Elastic Beanstalk to deploy the Node.js application. Launch an Amazon Redshift cluster, and point your application to its JDBC endpoint.**
- D. Upload the HTML, CSS, images, and JavaScript for the front end, plus the Node.js code for the server-side application, to an Amazon S3 bucket. Create a CloudFront distribution with this bucket as its origin. Launch an Amazon Redshift cluster, and point your application to its JDBC endpoint.
- E. Upload the HTML, CSS, images, and JavaScript for the front end to an Amazon S3 bucket. Use AWS Elastic Beanstalk to deploy the Node.js application. Launch an Amazon Redshift cluster, and point your application to its JDBC endpoint.

Commented [LC436]:

Question #490 [SKIP]

What is the purpose of the Docker network docker_gwbridge?

- A. allows communication between containers on the same host
- **B. allows communication between swarm nodes on different hosts**
- C. allows communication between swarm nodes on the same host
- D. allows communication between containers on the different hosts

Commented [LC437]: The docker_gwbridge is a local bridge network which is automatically created by Docker in two different circumstances: When you initialize or join a swarm, Docker creates the docker_gwbridge network and uses it for communication among swarm nodes on different hosts. When none of a container's networks can provide external connectivity, Docker connects the container to the docker_gwbridge network in addition to the container's other networks, so that the container can connect to external networks or other swarm nodes.

Reference:
https://docs.docker.com/engine/userguide/networking/#the-docker_gwbridge-network

Question #491

You are responsible for operating a continuous integration application that monitors version control for updates and then starts fresh Amazon EC2 instances to run a comprehensive set of build tests.

What should you do to secure the lowest total cost while running as many tests as feasible in parallel?

- A. Perform syntax checking on the continuous integration system before launching a new Amazon EC2 instance for build test, unit and integration tests.
- B. Perform syntax and build tests on the continuous integration system before launching the new Amazon EC2 instance unit and integration tests.
- C. Perform all tests on the continuous integration system, using AWS OpsWorks for unit, integration, and build tests.
- D. Perform syntax checking on the continuous integration system before launching a new AWS Data Pipeline for coordinating the output of unit, integration, and build tests.

Commented [LC438]:

Question #492

A business that utilizes electronic health records operates a fleet of Amazon EC2 instances running the Amazon Linux operating system. As part of the patient privacy standards, the organization must guarantee that patches for the operating system and apps running on the EC2 instances are applied on a constant basis.

How can operating system and application patches be deployed automatically utilizing a default and custom repository?

- A. Use AWS Systems Manager to create a new patch baseline including the custom repository. Execute the AWS-RunPatchBaseline document using the run command to verify and install patches.
- B. Use AWS Direct Connect to integrate the corporate repository and deploy the patches using Amazon CloudWatch scheduled events, then use the CloudWatch dashboard to create reports.
- C. Use yum-config-manager to add the custom repository under /etc/yum.repos.d and run yum-config-manager-enable to activate the repository.
- D. Use AWS Systems Manager to create a new patch baseline including the corporate repository. Execute the AWS-AmazonLinuxDefaultPatchBaseline document using the run command to verify and install patches.

Commented [LC439]: Also see example 3 in <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-manager-how-it-works-alt-source-repository.html> which lines up nicely with option A.

AWS-AmazonLinuxDefaultPatchBaseline is the patch baseline name. AWS-RunPatchBaseline is the document you can run. Even it mentions the customer patch baseline name it isn't right. we need to use the document we can run command here. So A is definitely right.

Question #493

A DevOps Engineer is evaluating the most cost-effective method for implementing an image batch processing cluster on AWS. The program is not compatible with Docker containers and must be hosted on Amazon EC2. The batch task saves checkpoint data to a Network File System (NFS) and is interrupt-tolerant. It takes 30 minutes to configure the cluster software using a generic EC2 Linux image.

Which approach is the MOST cost-effective?

- A. Use Amazon EFS for checkpoint data. To complete the job, use an EC2 Auto Scaling group and an On-Demand pricing model to provision EC2 instances temporarily.
- B. Use GlusterFS on EC2 instances for checkpoint data. To run the batch job, configure EC2 instances manually. When the job completes, shut down the instances manually.
- C. Use Amazon EFS for checkpoint data. Use EC2 Fleet to launch EC2 Spot Instances, and use user data to configure the EC2 Linux instance on startup.
- D. Use Amazon EFS for checkpoint data. Use EC2 Fleet to launch EC2 Spot Instances. Create a standard cluster AMI and use the latest AMI when creating instances.

Commented [LC440]: D, because by including the cluster software into the image you do not have the 30 mins startup time. Then you can use Spot instances. Otherwise 30 mins of your Spot instances is wasted on installing the cluster software.

Question #494

A business is conducting a review of its information asset management policies. One of the DevOps Engineer's policies has been identified as being too liberal. Over the weekend, the policy is utilized by an AWS Lambda function to execute a stop command to Amazon EC2 instances with the Environment: NonProduction tag. Currently, the following policy is in effect: [sampleXML/xmlfile-741].png]

What adjustments should the Engineer make to implement a least-permissions policy? (Select three.)

A.

Add the following conditional expression:

```
"Condition": {
  "StringEquals": {
    "aws:principaltype": "lambda.amazonaws.com"
  }
}
```

B.

Change "Resource": "*" to "Resource":
"arn:aws:ec2:*:*:instance/*"

C.

Add the following conditional expression:

```
"Condition": {
  "StringNotEquals": {
    "ec2:ResourceTag/Environment": "Production"
  }
}
```

D.

Add the following conditional expression:

```
"Condition": {
  "StringEquals": {
    "ec2:ResourceTag/Environment": "NonProduction"
  }
}
```

E.

Change "Action": "ec2:*" to "Action": "ec2:stopInstances"

F.

Add the following conditional expression:

```
"Condition": {
  "DateGreaterThan": {
    "aws:CurrentTime": "${aws:DateTime:Friday}"
  },
  "DateLessThan": {
    "aws:CurrentTime": "${aws:DateTime:Monday}"
  }
}
```

Commented [LC441]:

Commented [LC442]:

Commented [LC443]:

Question #495

A government entity may have several Amazon Web Services accounts, many of which include sensitive citizen data. A security team want to identify unusual account and network activity (such as SSH brute force assaults) in any account and to consolidate this information in a dedicated security account. Event data should be saved in an Amazon S3 bucket under the security account, which is monitored by the department's SIEM system.

How is this possible?

- A. Enable Amazon Macie in every account. Configure the security account as the Macie Administrator for every member account using invitation/acceptance. Create an Amazon CloudWatch Events rule in the security account to send all findings to Amazon Kinesis Data Firehose, which should push the findings to the S3 bucket.
- B. Enable Amazon Macie in the security account only. Configure the security account as the Macie Administrator for every member account using invitation/ acceptance. Create an Amazon CloudWatch Events rule in the security account to send all findings to Amazon Kinesis Data Streams. Write an application using KCL to read data from the Kinesis Data Streams and write to the S3 bucket.
- C. Enable Amazon GuardDuty in every account. Configure the security account as the GuardDuty Administrator for every member account using invitation/ acceptance. Create an Amazon CloudWatch rule in the security account to send all findings to Amazon Kinesis Data Firehose, which will push the findings to the S3 bucket.
- D. Enable Amazon GuardDuty in the security account only. Configure the security account as the GuardDuty Administrator for every member account using invitation/acceptance. Create an Amazon CloudWatch rule in the security account to send all findings to Amazon Kinesis Data Streams. Write an application using KCL to read data from Kinesis Data Streams and write to the S3 bucket.

Commented [LC444]: This one is obvious. C is the right answer. If you studied up on GuardDuty, you need to configure it on EVERY account you want it to track. You then set a single account as the Administrator. Kinesis Firehose is all that is needed to port the findings to S3 where the question CLEARLY STATES that they have their own SIEM tool to monitor the output in the S3 bucket.

Question #496

Which of the following is not a possible cause for a Multi-AZ RDS instance to failover?

- A. An Availability Zone outage
- B. A manual failover of the DB instance was initiated using Reboot with failover
- C. To autoscale to a higher instance class
- D. The primary DB instance fails

Commented [LC445]: The primary DB instance switches over automatically to the standby replica if any of the > following conditions occur: An Availability Zone outage, the primary DB instance fails, the DB instance's server type is changed, the operating system of the DB instance is, undergoing software patching, a manual failover of the DB instance was initiated using Reboot with failover.

Reference:

<http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

Question #497

Which of the following does not qualify as a CloudFormation Helper Script?

- A. cfn-signal
- B. cfn-hup
- C. cfn-request
- D. cfn-get-metadata

Commented [LC446]: This is the complete list of CloudFormation Helper Scripts: cfn-init, cfn-signal, cfn-get-metadata, cfn-hup

Reference:

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-helper-scriptsreference.html>

Question #498

What is the Amazon Web Services CloudTrail Processing Library?

- A. A static library with CloudTrail log files in a movable format machine code that is directly executable
- B. An object library with CloudTrail log files in a movable format machine code that is usually not directly executable
- C. A Java library that makes it easy to build an application that reads and processes CloudTrail log files
- D. A PHP library that renders various generic containers needed for CloudTrail log files

Commented [LC447]: AWS CloudTrail Processing Library is a Java library that makes it easy to build an application that reads and processes CloudTrail log files. You can download CloudTrail Processing Library from GitHub.

Reference:

<http://aws.amazon.com/cloudtrail/faqs/>

Question #499

A business is embracing serverless computing and transferring many existing apps to AWS Lambda. A DevOps engineer must develop an automated deployment strategy for AWS CodePipeline that incorporates correct version control, branching techniques, and rollback mechanisms.

Which sequence of steps should the DevOps engineer use to configure the pipeline? (Select three.)

- A. Use Amazon S3 as the source code repository.
- B. Use AWS CodeCommit as the source code repository.
- C. Use AWS CloudFormation to create an AWS Serverless Application Model (AWS SAM) template for deployment.
- D. Use AWS CodeBuild to create an AWS Serverless Application Model (AWS SAM) template for deployment.
- E. Use AWS CloudFormation to deploy the application.
- F. Use AWS CodeDeploy to deploy the application.

Commented [LC448]: <https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-serverlessrepo-auto-publish.html>

Commented [LC449]:

Commented [LC450]:

Questions 500-END

Question #500

You want to develop a new search tool for your monitoring system that will enable your information security team to easily audit any API requests made via your AWS accounts.

Which AWS services combination can you utilize to construct and automate the backend procedures that support this tool? (Select three.)

- A. Create an Amazon CloudSearch domain for API call logs. Configure the search domain so that it can be used to index API call logs for the search tool.
- B. Use AWS CloudTrail to store API call logs in an Amazon S3 bucket. Configure an Amazon Simple Notification Service topic called "log-notification" that notifies subscribers when new logs are available. Subscribe an Amazon SQS queue to the topic.
- C. Use Amazon Cloudwatch to ship AWS CloudTrail logs to your monitoring system.
- D. Create an AWS Elastic Beanstalk application in worker role mode that uses an Amazon Simple Email Service (SES) domain to facilitate batch processing new API call log files retrieved from an Amazon S3 bucket into a search index.
- E. Use AWS CloudTrail to store API call logs in an Amazon S3 bucket. Configure Amazon Simple Email Service (SES) to notify subscribers when new logs are available. Subscribe an Amazon SQS queue to the email domain.
- F. Create Amazon Cloudwatch custom metrics for the API call logs. Configure a Cloudwatch search domain so that it can be used to index API call logs for the search tool.
- G. Create an AWS Elastic Beanstalk application in worker role mode that uses an Amazon SQS queue to facilitate batch processing new API call log files retrieved from an Amazon S3 bucket into a search index.

Commented [LC451]: <https://aws.amazon.com/blogs/staff/how-to-search-cloudtrail-logs-easily-with-amazon-cloudsearch/>

Commented [LC452]:

Commented [LC453]:

Question #501

A DevOps Engineer has been assigned with the responsibility of migrating a mission-critical business application written in Go to AWS. The development team responsible for this application is understaffed and wants a solution that enables them to concentrate only on application development. Additionally, they want to allow blue/green deployments and do A/B testing.

Which solution will satisfy these criteria?

- A. Deploy the application on an Amazon EC2 instance and create an AMI of this instance. Use this AMI to create an automatic scaling launch configuration that is used in an Auto Scaling group. Use an Elastic Load Balancer to distribute traffic. When changes are made to the application, a new AMI is created and replaces the launch configuration.
- B. Use Amazon Lightsail to deploy the application. Store the application in a zipped format in an Amazon S3 bucket. Use this zipped version to deploy new versions of the application to Lightsail. Use Lightsail deployment options to manage the deployment.
- C. Use AWS CodePipeline with AWS CodeDeploy to deploy the application to a fleet of Amazon EC2 instances. Use an Elastic Load Balancer to distribute the traffic to the EC2 instances. When making changes to the application, upload a new version to CodePipeline and let it deploy the new version.
- D. Use AWS Elastic Beanstalk to host the application. Store a zipped version of the application in Amazon S3, and use that location to deploy new versions of the application using Elastic Beanstalk to manage the deployment options.

Commented [LC454]: D - beanstalk has blue/green and a/b testing (canary) and it is less overhead than the others

<https://aws.amazon.com/quickstart/architecture/blue-green-deployment/>

Question #502

You must do A/B testing on various multi-tier web apps. Each one has its own infrastructure, including Amazon Elastic Compute Cloud (EC2) front-end servers, Amazon ElastiCache clusters, Amazon Simple Queue Service (SQS) queues, and Amazon Relational Database Service (RDS) instances.

Which service combination would enable you to manage traffic across multiple deployed versions of your application?

- A. Create one AWS Elastic Beanstalk application and all AWS resources (using configuration files inside the application source bundle) for each web application. New versions would be deployed a-eating Elastic Beanstalk environments and using the Swap URLs feature.
- B. Using AWS CloudFormation templates, create one Elastic Beanstalk application and all AWS resources (in the same template) for each web application. New versions would be deployed using AWS CloudFormation templates to create new Elastic Beanstalk environments, and traffic would be balanced between them using weighted Round Robin (WRR) records in Amazon Route53.
- C. Using AWS CloudFormation templates, create one Elastic Beanstalk application and all AWS resources (in the same template) for each web application. New versions would be deployed updating a parameter on the CloudFormation template and passing it to the cfn-hup helper daemon, and traffic would be balanced between them using Weighted Round Robin (WRR) records in Amazon Route 53.
- D. Create one Elastic Beanstalk application and all AWS resources (using configuration files inside the application source bundle) for each web application. New versions would be deployed updating the Elastic Beanstalk application version for the current Elastic Beanstalk environment.

Commented [LC455]:

Question #503

When an Auto Scaling group is running in Amazon Elastic Compute Cloud (EC2), your application rapidly scales up and down in response to load within a 10-minute window; however, after the load peaks, you begin to notice issues with previously terminated Amazon EC2 resources remaining active in your configuration management system.

What is the most dependable and effective approach to manage Amazon EC2 resource cleaning inside your configuration management system? (Select two.)

- A. Write a script that is run by a daily cron job on an Amazon EC2 instance and that executes API Describe calls of the EC2 Auto Scaling group and removes terminated instances from the configuration management system.
- B. Configure an Amazon Simple Queue Service (SQS) queue for Auto Scaling actions that has a script that listens for new messages and removes terminated instances from the configuration management system.
- C. Use your existing configuration management system to control the launching and bootstrapping of instances to reduce the number of moving parts in the automation.
- D. Write a small script that is run during Amazon EC2 instance shutdown to de-register the resource from the configuration management system.
- E. Use Amazon Simple Workflow Service (SWF) to maintain an Amazon DynamoDB database that contains a whitelist of instances that have been previously launched, and allow the Amazon SWF worker to remove information from the configuration management system.

Commented [LC456]: A. Automatically disqualified for running on an instance instead of Lambda. Plus, why daily? May keep the instances for too much time.

B. Pricey, but reliable.

C. Looking at the question, this is already happening... and failing. So, this is a no go.

D. Legit, as this can be assigned to a life-cycle hook and done.

E. Newp. Too much beer and places that can fail.

BD.

Commented [LC457]:

Question #504

Your firm has chosen to adopt a third-party configuration management application that makes use of a master server from which all nodes get configuration. You've created a custom base Amazon Machine Image that comes pre-installed with the third-party configuration management agent. You wish to share the same basic AMI across Development, Test, and Production environments, each with its own master server. How should your Amazon EC2 instances be configured to automatically register with the right master server upon launch?

- A. Create a tag for all instances that specifies their environment. When launching instances, provide an Amazon EC2 UserData script that gets this tag by querying the MetaData Service and registers the agent with the master.
- B. Use Amazon CloudFormation to describe your environment. Configure an input parameter for the master server hostname/address, and use this parameter within an Amazon EC2 UserData script that registers the agent with the master.
- C. Create a script on your third-party configuration management master server that queries the Amazon EC2 API for new instances and registers them with it.
- D. Use Amazon Simple Workflow Service to automate the process of registering new instances with your master server. Use an Environment tag in Amazon EC2 to register instances with the correct master server.

Commented [LC458]: B makes sense to maintain environment variables and use them to register instance

Question #505

Your team want to begin utilizing CloudFormation in continuous delivery in order to automate the construction and deployment of whole, versioned stacks or stack layers. You're working with a three-tier, mission-critical system.

Which of the following is NOT a recommended method for deploying CloudFormation in a continuous delivery environment?

- A. Use the AWS CloudFormation `<code>ValidateTemplate</code>` call before publishing changes to AWS.
- B. Model your stack in one template, so you can leverage CloudFormation's state management and dependency resolution to propagate all changes.
- C. Use CloudFormation to create brand new infrastructure for all stateless resources on each push, and run integration tests on that set of infrastructure.
- D. Parametrize the template and use `<code>Mappings</code>` to ensure your template works in multiple Regions.

Commented [LC459]: Putting all resources in one stack is a bad idea, since different tiers have different life cycles and frequencies of change. For additional guidance about organizing your stacks, you can use two common frameworks: a multi-layered architecture and service-oriented architecture (SOA).

Reference:

<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/bestpractices.html#organizingstacks>

Question #506

On Amazon EC2, an application is running. It is tied to an IAM role that encounters an AccessDenied error while attempting to access a SecureString parameter resource in the AWS Systems Manager Parameter Store. The SecureString parameter is encrypted using a Customer Master Key (CMK) that is maintained by the customer.

What measures should the DevOps Engineer take to enable access to the role with the least privilege possible? (Select three.)

- A. Set `ssm:GetParameter` for the parameter resource in the instance role's IAM policy.
- B. Set `kms:Decrypt` for the instance role in the customer-managed CMK policy.
- C. Set `kms:Decrypt` for the customer-managed CMK resource in the role's IAM policy.
- D. Set `ssm:DecryptParameter` for the parameter resource in the instance role IAM policy.
- E. Set `kms:GenerateDataKey` for the user on the AWS managed SSM KMS key.
- F. Set `kms:Decrypt` for the parameter resource in the customer-managed CMK policy.

Question #507

Which of the following defines the geographic scope of an EC2 security group?

- A. Security groups are global.
- B. They are confined to Placement Groups.
- C. They are confined to Regions.
- D. They are confined to Availability Zones.

Question #508 [SKIP]

Which resource is not capable of being specified in an Ansible Playbook?

- A. Fact Gathering State
- B. Host Groups
- C. Inventory File
- D. Variables

Question #509 [SKIP]

When a playbook is executed on a remote target computer, you see a Python error similar to "[Errno 13] Permission denied: '/home/nick/.ansible/tmp'."

What is the most probable source of this issue?

- A. The user's home or `~/.ansible/` directory on the Ansible system is not writeable by the user running the play.
- B. The specified user does not exist on the remote system.
- C. The user running `~/.ansible-playbook` must run it from their own home directory.
- D. The user's home or `~/.ansible/` directory on the Ansible remote host is not writeable by the user running the play.

Question #510

A firm in the education sector is operating a Docker-based application on numerous Amazon EC2 instances inside an Amazon ECS cluster. When a new version of the application is deployed, the developer publishes a new image to a private Docker container registry and then pauses and restarts all tasks to verify that they all receive the newest version of the application. The developer notices that new tasks are being executed on occasion using an old image.

How might this problem be avoided?

- A. After pushing the new image, restart ECS Agent, and then start the tasks.
- B. Use `'latest'` for the Docker image tag in the task definition.
- C. Update the digest on the task definition when pushing the new image.
- D. Use Amazon ECR for a Docker container registry.

Commented [LC460]: Correct Answer: A,B,C

F is not right, no need to give access to the parameter resource.

Here we need below access:

1. Instance profile can get parameter from SSM, A provide
2. Instance profile can use CMK to decrypt the parameter, C provide
3. CMK policy allow instance to use CMK decrypt action, B provide

Reference:

<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-paramstore-access.html>

Commented [LC461]:

Commented [LC462]:

Commented [LC463]: A security group is tied to a region and can be assigned only to instances in the same region. You can't enable an instance to communicate with an instance outside its region using security group rules. Traffic from an instance in another region is seen as WAN bandwidth.

Reference:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/resources.html>

Commented [LC464]: Ansible's inventory can only be specified on the command line, the Ansible configuration file or in environment variables.

Reference:

http://docs.ansible.com/ansible/intro_inventory.html

Commented [LC465]: Each task that Ansible runs calls a module. When Ansible uses modules, it copies the module to the remote target system. In the error above it attempted to copy it to the remote user's home directory and found that either the home directory or the `~/.ansible/` directory were not writeable and thus could not continue.

Reference:

http://docs.ansible.com/ansible/modules_intro.html

Commented [LC466]: Option "C":

The `'latest'` tag does not actually mean latest, it doesn't mean anything

Also, here's a discussion of why docker's use of `"latest"` is evil: <https://vsupalov.com/docker-latest-tag/>

Question #511

You want to create an application that manages work spread across remote components, and you discover that Amazon Simple Workflow Service (Amazon SWF) makes this simple. You've enabled logging in CloudTrail, but are uncertain about the Amazon SWF operations that are supported.

Which of the following is NOT a supported action?

- A. RegisterDomain
- B. RegisterWorkflowActivity
- C. RegisterActivityType
- D. RegisterWorkflowType

Commented [LC467]: Amazon SWF is integrated with AWS CloudTrail, a service that captures API calls made by or on behalf of Amazon SWF and delivers the log files to an Amazon

S3 bucket that you specify. The API calls can be made indirectly by using the Amazon SWF console or directly by using the Amazon SWF API. When CloudTrail logging is enabled, calls made to Amazon SWF actions are tracked in log files. Amazon SWF records are written together with any other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a specified time period and file size.

The following actions are supported:

DeprecateActivityType -

DeprecateDomain -

DeprecateWorkflowType -

RegisterActivityType -

RegisterDomain -

RegisterWorkflowType -

Reference:
<http://docs.aws.amazon.com/amazonswf/latest/developerguide/ct-logging.html>

Commented [LC468]:

Commented [LC469]:

Question #512

At the moment, your deployment method entails configuring your load balancer to link to a maintenance page, shutting off all web application servers, deploying your code, reactivating the web application servers, and deleting the maintenance page. You've determined, in collaboration with your development team, that rolling deployments of your software will result in a better user experience and a more agile deployment process.

Which approaches might you use in order to create a cost-effective rolling deployment process? (Select two.)

- A. Use the Amazon Elastic Cloud Compute (EC2) API to write a service to return a list of servers based on the tags for the application that needs deployment, and use Amazon Simple Queue Service to queue up all servers for a rolling deployment.
- B. Re-deploy your application on AWS Elastic Beanstalk, and use Elastic Beanstalk rolling deployments.
- C. Re-deploy your application on an AWS OpsWorks stack, and take advantage of OpsWorks rolling deployments.
- D. Re-deploy your application using an AWS CloudFormation template, launch a new CloudFormation stack during each deployment, and then tear down the old stack.
- E. Re-deploy your application using an AWS CloudFormation template with Auto Scaling group, and use update policies to provide rolling updates.
- F. Using Amazon Simple Workflow Service, create a workflow application that talks to the Amazon EC2 API to deploy your new code in a rolling fashion.

Question #513

A DevOps engineer is developing a multi-region disaster recovery plan for an application that requires a one-hour RPO and a four-hour RTO. The application is deployed using an AWS CloudFormation template that produces an Application Load Balancer, Amazon EC2 instances in an Auto Scaling group, and an Amazon RDS Multi-AZ database instance with allotted storage of 20 GB. The application instance's AMI is empty and was copied to the destination Region.

Which combination of steps is the LEAST EXPENSIVE way to achieve the recovery objectives? (Select two.)

- A. Launch an RDS DB instance in the failover Region and use AWS DMS to configure ongoing replication from the source database.
- B. Schedule an AWS Lambda function to take a snapshot of the database every hour and copy the snapshot to the failover Region.
- C. Upon failover, update the CloudFormation stack in the failover Region to update the Auto Scaling group from one running instance to the desired number of instances. When the stack update is complete, change the DNS records to point to the failover Region's Elastic Load Balancer.
- D. Upon failover, launch the CloudFormation template in the failover Region with the snapshot ID as an input parameter. When the stack creation is complete, change the DNS records to point to the failover Region's Elastic Load Balancer.
- E. Utilizing the build-in RDS automated backups, set up an event with Amazon CloudWatch Events that triggers an AWS Lambda function to copy the snapshot to the failover Region.

Commented [LC470]:

Commented [LC471]:

Question #514

Your application's Amazon Elastic Compute Cloud (EC2) instances are configured through a master configuration file stored in a versioned Amazon Simple Storage Service (S3) bucket.

Which of the following approaches should you use to safely and cost-effectively deploy the current configuration version onto the instances?

- A. Create an Amazon DynamoDB table to store the different versions of the configuration file. Associate AWS Identity and Access Management (IAM) EC2 roles to the Amazon EC2 instances, and reference the DynamoDB table to get the latest file from Amazon Simple Storage Service (S3).
- B. Associate an IAM S3 role to the bucket, list the object versions using the Amazon S3 API, and then get the latest object.
- C. Associate an IAM EC2 role to the instances, list the object versions using the Amazon S3 API, and then get the latest object.
- **D. Associate an IAM EC2 role to the instances, and then simply get the object from Amazon S3, because the default is the current version.**
- E. Store the IAM credentials in the Amazon EC2 user data for each instance, and then simply get the object from S3, because the default is the current version.

Commented [LC472]:

Question #515

A DevOps Engineer is in charge of deploying a PHP application. The Engineer is working with a hybrid deployment, which includes on-premises servers and Amazon EC2 instances. The program requires access to a database holding very sensitive data. Application instances need access to database credentials, which must be secured both during storage and transmission to the instances.

How should the Engineer automate the deployment process while still adhering to security standards?

- A. Use AWS Elastic Beanstalk with a PHP platform configuration to deploy application packages to the instances. Store database credentials on AWS Systems Manager Parameter Store using the Secure String data type. Define an IAM role for Amazon EC2 allowing access, and decrypt only the database credentials. Associate this role to all the instances.
- **B. Use AWS CodeDeploy to deploy application packages to the instances. Store database credentials on AWS Systems Manager Parameter Store using the Secure String data type. Define an IAM policy for allowing access, and decrypt only the database credentials. Attach the IAM policy to the role associated to the instance profile for CodeDeploy-managed instances, and to the role used for on-premises instances registration on CodeDeploy.**
- C. Use AWS CodeDeploy to deploy application packages to the instances. Store database credentials on AWS Systems Manager Parameter Store using the Secure String data type. Define an IAM role with an attached policy that allows decryption of the database credentials. Associate this role to all the instances and on-premises servers.
- D. Use AWS CodeDeploy to deploy application packages to the instances. Store database credentials in the AppSpec file. Define an IAM policy for allowing access to only the database credentials. Attach the IAM policy to the role associated to the instance profile for CodeDeploy-managed instances and the role used for on-premises instances registration on CodeDeploy.

Commented [LC473]: A is wrong because Because the answer states apply IAM role to "Associate this role to all the instances". It should apply only to CodeDeploy managed instances

I'll go with B

For those choosing C, there is a second point in there, recall: to decrypt a SSM Secure String data type you need
1 - Allow access to the secure string (this is different to decrypt)
2 - Allow decrypt access (you can access but not necessary decrypt)
So this also makes C wrong because C mention decrypt access without allow access first.

Question #516

What requirements must be met in order to obtain gigabit network throughput on EC2?

You've previously chosen cluster-compute, 10GB instances with improved networking, and your workload is network-bound, yet you're not getting 10 gigabit speeds.

- A. Enable biphex networking on your servers, so packets are non-blocking in both directions and there's no switching overhead.
- B. Ensure the instances are in different VPCs so you don't saturate the Internet Gateway on any one VPC.
- C. Select PIOPS for your drives and mount several, so you can provision sufficient disk throughput.
- **D. Use a placement group for your instances so the instances are physically near each other in the same Availability Zone.**

Commented [LC474]: For instances that are enabled for enhanced networking, the following rules apply:
Instances within a cluster placement group can use up to 10 Gbps for single-flow traffic. Instances that are not within a cluster placement group can use up to 5 Gbps for single-flow traffic.

Answer D

Question #517

A developer is developing a continuous deployment procedure for a new development team in order to streamline the process of promoting source code in AWS. Developers want to save and promote code for deployment from development to production while retaining the option to roll back the deployment in the event of failure.

Which design will have the LEAST downtime?

- A. Create one repository in AWS CodeCommit. Create a development branch to hold merged changes. Use AWS CodeBuild to build and test the code stored in the development branch triggered on a new commit. Merge to the master and deploy to production by using AWS CodeDeploy for a blue/green deployment.
- B. Create one repository for each Developer in AWS CodeCommit and another repository to hold the production code. Use AWS CodeBuild to merge development and production repositories, and deploy to production by using AWS CodeDeploy for a blue/green deployment.
- C. Create one repository for development code in AWS CodeCommit and another repository to hold the production code. Use AWS CodeBuild to merge development and production repositories, and deploy to production by using AWS CodeDeploy for a blue/green deployment.
- D. Create a shared Amazon S3 bucket for the Development team to store their code. Set up an Amazon CloudWatch Events rule to trigger an AWS Lambda function that deploys the code to production by using AWS CodeDeploy for a blue/green deployment.

Commented [LC475]: Standard CI/CD procedure

Question #518 [SKIP]

What are the default rules for memory limits in a Docker container?

- A. Limited memory, limited kernel memory
- B. Unlimited memory, limited kernel memory
- C. Limited memory, unlimited kernel memory
- D. Unlimited memory, unlimited kernel memory

Commented [LC476]: Kernel memory limits are expressed in terms of the overall memory allocated to a container. Consider the following scenarios:

Unlimited memory, unlimited kernel memory: This is the default behavior.

Unlimited memory, limited kernel memory: This is appropriate when the amount of memory needed by all cgroups is greater than the amount of memory that actually exists on the host machine. You can configure the kernel memory to never go over what is available on the host machine, and containers which need more memory need to wait for it.

Limited memory, unlimited kernel memory: The overall memory is limited, but the kernel memory is not.

Limited memory, limited kernel memory: Limiting both user and kernel memory can be useful for debugging memory-related problems. If a container is using an unexpected amount of either type of memory, it will run out of memory without affecting other containers or the host machine. Within this setting, if the kernel memory limit is lower than the user memory limit, running out of kernel memory will cause the container to experience an OOM error. If the kernel memory limit is higher than the user memory limit, the kernel limit will not cause the container to experience an OOM.

Reference:

https://docs.docker.com/engine/admin/resource_constraint/s/#kernel-memory-details

Commented [LC477]: The ideal way is to create a new launch configuration, attach it to the existing Auto Scaling group, and terminate the running instances. Option A is invalid because Elastic beanstalk cannot launch new instances on demand. Since the current scenario requires Autoscaling, this is not the ideal option Option B is invalid because this will be a maintenance overhead, since you just have an Autoscaling Group. There is no need to create a whole Cloudformation template for this. Option D is invalid because Autoscaling Group will still launch EC2 instances with the older launch configuration.

Question #519

You've deployed an application on AWS that utilizes Autoscaling to launch more instances. You now want to modify the instance type of the newly created instances.

Which of the following is an action item necessary to complete this deployment?

- A. Use Elastic Beanstalk to deploy the new application with the new instance type
- B. Use Cloudformation to deploy the new application with the new instance type
- C. Create a new launch configuration with the new instance type
- D. Create new EC2 instances with the new instance type and attach it to the Autoscaling Group

Question #520

A business has created a Node.js web application that offers REST services for storing and retrieving time series data. The development team creates the web application on business laptops, tests it locally, and manually deploys it to a single on-premises server that connects to a local MySQL database. In two weeks, the firm will begin a trial period during which the application will get regular upgrades in response to client input. The following conditions must be satisfied:

- ☞ The team must be able to reliably build, test, and deploy new updates on a daily basis, without downtime or degraded performance.
- ☞ The application must be able to scale to meet an unpredictable number of concurrent users during the trial.

Which option will enable the team to reach these goals most quickly?

- A. Create two Amazon Lightsail virtual private servers for Node.js; one for test and one for production. Build the Node.js application using existing processes and upload it to the new Lightsail test server using the AWS CLI. Test the application, and if it passes all tests, upload it to the production server. During the trial, monitor the production server usage, and if needed, increase performance by upgrading the instance type.
- B. Develop an AWS CloudFormation template to create an Application Load Balancer and two Amazon EC2 instances with Amazon EBS (SSD) volumes in an Auto Scaling group with rolling updates enabled. Use AWS CodeBuild to build and test the Node.js application and store it in an Amazon S3 bucket. Use user- data scripts to install the application and the MySQL database on each EC2 instance. Update the stack to deploy new application versions.
- C. Configure AWS Elastic Beanstalk to automatically build the application using AWS CodeBuild and to deploy it to a test environment that is configured to support auto scaling. Create a second Elastic Beanstalk environment for production. Use Amazon RDS to store data. When new versions of the applications have passed all tests, use Elastic Beanstalk 'swap cname' to promote the test environment to production.
- D. Modify the application to use Amazon DynamoDB instead of a local MySQL database. Use AWS OpsWorks to create a stack for the application with a DynamoDB layer, an Application Load Balancer layer, and an Amazon EC2 instance layer. Use a Chef recipe to build the application and a Chef recipe to deploy the application to the EC2 instance layer. Use custom health checks to run unit tests on each instance with rollback on failure.

Commented [LC478]: <https://docs.aws.amazon.com/codebuild/latest/userguide/sample-elastic-beanstalk.html#sample-elastic-beanstalk-eb-cli>

Question #521 [SKIP]

In which Docker Swarm architecture does the swarm manager distribute a fixed number of replica jobs across nodes depending on the size you specify in the desired state?

- A. distributed services
- B. scaled services
- C. replicated services
- D. global services

Commented [LC479]: A service is the definition of the tasks to execute on the worker nodes. It is the central structure of the swarm system and the primary root of user interaction with the swarm. When you create a service, you specify which container image to use and which commands to execute inside running containers. In the replicated services model, the swarm manager distributes a specific number of replica tasks among the nodes based upon the scale you set in the desired state. For global services, the swarm runs one task for the service on every available node in the cluster. A task carries a Docker container and the commands to run inside the container. It is the atomic scheduling unit of swarm. Manager nodes assign tasks to worker nodes according to the number of replicas set in the service scale. Once a task is assigned to a node, it cannot move to another node. It can only run on the assigned node or fail.

Reference:

<https://docs.docker.com/engine/swarm/key-concepts/#services-and-tasks>

Commented [LC480]: Latency Based Records allow request distribution when all is well with both regions, and the Failover component enables fallbacks between regions. By adding in the ELB and ASG, your system in the surviving region can expand to meet 100% of demand instead of the original fraction, whenever failover occurs.

Reference:

<http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html>

Question #522

Your API must be able to remain operational during AWS regional outages. Your API does not maintain state; it only combines data from several sources - you do not have a database.

What is a simple yet effective method for achieving this uptime goal?

- A. Use a CloudFront distribution to serve up your API. Even if the region your API is in goes down, the edge locations CloudFront uses will be fine.
- B. Use an ELB and a cross-zone ELB deployment to create redundancy across datacenters. Even if a region fails, the other AZ will stay online.
- C. Create a Route53 Weighted Round Robin record, and if one region goes down, have that region redirect to the other region.
- D. Create a Route53 Latency Based Routing Record with Failover and point it to two identical deployments of your stateless API in two different regions. Make sure both regions use Auto Scaling Groups behind ELBs.

Question #523

Which Auto Scaling technique would be most beneficial for testing new instances prior to providing traffic to them while maintaining them in your Auto Scaling Group?

- A. Suspend the process AZ Rebalance
- B. Suspend the process Health Check
- C. Suspend the process Replace Unhealthy
- **D. Suspend the process AddToLoadBalancer**

Question #524

Which of the following statements is correct about the configuration of proxy support for the Amazon Inspector agent on Linux-based systems?

- A. Amazon Inspector proxy support on Linux-based systems is achieved through installing proxyenabled version of the agent which comes with pre-configured files that you need to edit to match your environment.
- B. Amazon Inspector agent does NOT support the use of proxy on Linux-based systems.
- **C. Amazon Inspector proxy configuration on Linux-based system is included in awsagent.env file under /etc/init.d/**
- D. Amazon Inspector agent proxy settings on Linux-based systems are configured through WinHTTP proxy.

Question #525

A DevOps Engineer is tasked with the responsibility of deploying a scalable three-tier Node.js application on AWS. The application must be completely up and running throughout deployments and be capable of rolling back to earlier versions. Additionally, other apps will connect to the same MySQL backend database.

The CIO has issued the following guidelines about logging:

- ⇒ Centrally view all current web access server logs.
- ⇒ Search and filter web and application logs in near-real time.
- ⇒ Retain log data for three months.

How are these criteria to be fulfilled?

- A. Deploy the application using AWS Elastic Beanstalk. Configure the environment type for Elastic Load Balancing and Auto Scaling. Create an Amazon RDS MySQL instance inside the Elastic Beanstalk stack. Configure the Elastic Beanstalk log options to stream logs to Amazon CloudWatch Logs. Set retention to 90 days.
- B. Deploy the application on Amazon EC2. Configure Elastic Load Balancing and Auto Scaling. Use an Amazon RDS MySQL instance for the database tier. Configure the application to store log files in Amazon S3. Use Amazon EMR to search and filter the data. Set an Amazon S3 lifecycle rule to expire objects after 90 days.
- **C. Deploy the application using AWS Elastic Beanstalk. Configure the environment type for Elastic Load Balancing and Auto Scaling. Create the Amazon RDS MySQL instance outside the Elastic Beanstalk stack. Configure the Elastic Beanstalk log options to stream logs to Amazon CloudWatch Logs. Set retention to 90 days.**
- D. Deploy the application on Amazon EC2. Configure Elastic Load Balancing and Auto Scaling. Use an Amazon RDS MySQL instance for the database tier. Configure the application to load streaming log data using Amazon Kinesis Data Firehose into Amazon ES. Delete and create a new Amazon ES domain every 90 days.

Commented [LC481]: If you suspend Add To Load Balancer, Auto Scaling launches the instances but does not add them to the load balancer or target group.

If you resume the AddToLoad Balancer process, Auto Scaling resumes adding instances to the load balancer or target group when they are launched.

However, Auto Scaling does not add the instances that were launched while this process was suspended. You must register those instances manually.

Option A is invalid because this just balances the number of CC2 instances in the group across the Availability Zones in the region

Option B is invalid because this just checks the health of the instances. Auto Scaling marks an instance as unhealthy if Amazon CC2 or Elastic Load Balancing tells Auto Scaling that the instance is unhealthy.

Option C is invalid because this process just terminates instances that are marked as unhealthy and later creates new instances to replace them.

Commented [LC482]: https://docs.aws.amazon.com/inspector/v1/userguide/inspector_agents-on-linux.html under "Configuring proxy support for an Amazon Inspector Classic agent"

1: Create a file called awsagent.env and save it in the /etc/init.d/ directory.

2: Edit awsagent.env to include these environment variables in the following format:

- export https_proxy=hostname:port
- export http_proxy=hostname:port
- export no_proxy=169.254.169.254

3: Install the Amazon Inspector agent by completing the steps in the Installing the agent on a Linux-based EC2 instance procedure.

Commented [LC483]: C:

explained here, also explained how to get the logs near-real time using cloudwatch.

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.logging.html>

Question #526

A development team want to deploy an application using AWS CloudFormation stacks, but the Developer IAM role lacks the necessary rights to supply the resources specified in the CloudFormation template. A DevOps Engineer's responsibility is to enable Developers to deploy stacks while adhering to the principle of least privilege.

Which solution will satisfy these criteria?

- A. Create an IAM policy that allows Developers to provision the required resources. Attach the policy to the Developer role.
- B. Create an IAM policy that allows full access to CloudFormation. Attach the policy to the Developer role.
- C. Create an AWS CloudFormation service role that has the required permissions. Grant the developer IAM role a cloudformation:* action. Use the new service role during stack deployments.
- D. Create an AWS CloudFormation service role that has the required permissions. Grant the developer IAM role the iam:PassRole permission. Use the new service role during stack deployments.

Commented [LC484]: Answer is D for the least role privilege that must be respected

Question #527

You are responsible for managing a three-layer online application that consists of an autoscaled web proxy tier, an autoscaled application tier, and an Amazon RDS database tier. You employ a load balancer to route requests from end users to the web proxy layer, and another load balancer to route requests between the web tier and the application tier. You find that after doing a minor database schema change, all of your web and application instances have been killed.

What may have triggered this?

- A. Your load balancers use an HTTP health check, and the page relies on retrieving data from your database.
- B. Your load balancer use TCP health checks to provide application-level health checks.
- C. The cooldown period of the Auto Scaling group is too short, so the instances do not have enough time to recover from an issue.
- D. Your Auto Scaling group health check type is set to "EC2" to check that the instances themselves are healthy.

Commented [LC485]: A is correct. During schema changes, page may fail resulting in ASG action

Question #528 [SKIP]

Which of the following answers is the correct syntax for providing two target hosts on the command line when executing an Ansible Playbook?

- A. ansible-playbook -h host1.example.com -i all playbook.yml
- B. ansible-playbook -i host1.example.com playbook.yml
- C. ansible-playbook -h host1.example.com,host2.example.com playbook.yml
- D. ansible-playbook -i host1.example.com,host2.example.com playbook.yml

Commented [LC486]: Ansible uses the '-i' flag for accepting an inventory file or host. To allow Ansible to determine if you are passing a host list versus an inventory file the list must be comma separated.

If a single host is specified, a trailing comma must be present.

Reference:

http://docs.ansible.com/ansible/intro_inventory.html#inventory

Question #529

Take a look at the following excerpt from a CloudTrail log file. Which event type is being recorded?

```
"eventTime": "2016-07-16T17:35:32Z",
"eventSource": "signin.amazonaws.com",
"eventName": "ConsoleLogin",
"awsRegion": "us-west-1",
"sourceIPAddress": "192.12.10",
...
```

- A. AWS console sign-in
- B. AWS log off
- C. AWS error
- D. AWS deployment

Commented [LC487]: CloudTrail records attempts to sign into the AWS Management Console, the AWS Discussion Forums and the AWS Support Center. Note, however, that CloudTrail does not record root sign-in failures.

Reference:

<http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-event-reference-awsconsole-sign-in-events.html>

Question #530

You have a stateless web server layer operating on Amazon EC2 instances behind a load balancer and are using Amazon RDS with read replicas.

Which of the following approaches should you use in order to create a self-healing and cost-effective architecture? (Select two.)

- A. Set up a third-party monitoring solution on a cluster of Amazon EC2 instances in order to emit custom CloudWatch metrics to trigger the termination of unhealthy Amazon EC2 instances.
- B. Set up scripts on each Amazon EC2 instance to frequently send ICMP pings to the load balancer in order to determine which instance is unhealthy and replace it.
- C. Set up an Auto Scaling group for the web server tier along with an Auto Scaling policy that uses the Amazon RDS DB CPU utilization CloudWatch metric to scale the instances.
- **D. Set up an Auto Scaling group for the web server tier along with an Auto Scaling policy that uses the Amazon EC2 CPU utilization CloudWatch metric to scale the instances.**
- E. Use a larger Amazon EC2 instance type for the web server tier and a larger DB instance type for the data storage layer to ensure that they don't become unhealthy.
- F. Set up an Auto Scaling group for the database tier along with an Auto Scaling policy that uses the Amazon RDS read replica lag CloudWatch metric to scale out the Amazon RDS read replicas.
- **G. Use an Amazon RDS Multi-AZ deployment.**

Commented [LC488]:

Commented [LC489]:

Question #531

Your DevOps team is in charge of a multi-tier, Windows-based online application that consists of web servers, Amazon RDS database instances, and a load balancer behind Amazon Route53. Your boss has assigned you with the task of developing a cost-effective rolling deployment solution for this online application.

Which technique should you use?

- A. Re-deploy your application on an AWS OpsWorks stack. Use the AWS OpsWorks done stack feature to allow updates between duplicate stacks.
- B. Re-deploy your application on Elastic Beanstalk and take advantage of Elastic BeanStalk rolling updates.
- C. Re-deploy your application using an AWS CloudFormation template, launch a new AWS CloudFormation stack during each deployment, and then tear down the old stack.
- **D. Re-deploy your application using an AWS CloudFormation template. Use AWS CloudFormation rolling deployment policies, create a new policy for your AWS CloudFormation stack, and initiate an update stack operation to deploy new code.**

Commented [LC490]: D is correct, Beanstalk is great for rolling deployment but not so much for 3 layered app. Only OpsWorks and CF and self-managed layers are good for that.

Question #532

A DevOps team wants to collaborate on a single source code repository. The team's development process and repository access restrictions must meet the following requirements:

- ⇒ Only team members can clone the repository and create new branches.
- ⇒ A production-ready code state should be isolated from any untested code changes.
- ⇒ Code changes should be approved by another team member before merging to the production-ready master branch.
- ⇒ All code change approvals must have an audit record.
- ⇒ New team members can quickly modify code.

Which activities would these criteria necessitate? (Select three.)

- A. Check out the master branch and develop new features locally on a feature branch to keep the production-ready code isolated. Ask team members to review the changes before committing the changes locally.
- B. Create an AWS CodeCommit repository and an IAM group with permissions to read/write changes to the repository. Add new team members to this group.
- C. Create an AWS CodeCommit repository and an IAM role with permissions to read/write changes to the repository. Attach this IAM role to a single IAM user. Ensure each member of the team uses this IAM user. Provide new team members the credentials to this IAM user.
- D. Create a local feature branch from the master branch for new features. Commit the new code and push the changes to the feature branch in the repository.
- E. Create a pull request so other team members can review the code changes. Implement any suggestions, pull any additional changes from the master branch, and push to the feature branch again. Merge the master branch with the feature branch.
- F. Create a pull request so other team members can review the code changes. Implement any suggestions, pull any additional changes from the master branch, resolve any conflicts, and push to the feature branch again. Merge the feature branch with the master branch.

Commented [LC491]:

Commented [LC492]:

Commented [LC493]:

Question #533

You're developing a huge, multi-tenant SaaS (software-as-a-service) application that includes a component that retrieves data for processing from a customer-specific Amazon S3 bucket under their Amazon account.

How should you verify that your application adheres to security best practices and minimizes risk when it retrieves data from a customer's Amazon S3 bucket?

- A. Have users create an IAM user with a policy that grants read-only access to the Amazon S3 bucket required by your application, and store the corresponding access keys in an encrypted database that holds their account data.
- B. Have users create a cross-account IAM role with a policy that grants read-only access to the Amazon S3 bucket required by your application to the AWS account ID running your production SaaS application.
- C. Have users create an Amazon S3 bucket policy that grants read-only access to the Amazon S3 bucket required by your application, and securely store the corresponding access keys in the database holding their account data.
- D. Have users create an Amazon S3 bucket policy that grants read-only access to the Amazon S3 bucket required by your application and limits access to the public IP address of the SaaS application.

Commented [LC494]: Correct Answer is D

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-walkthroughs-managing-access-example2.html>

Question #534

A healthcare provider has implemented a hybrid architecture consisting of 120 on-premises VMware servers running RedHat and 50 Amazon EC2 instances running Amazon Linux. The organization is in the midst of an all-in migration to AWS and is looking to develop a solution for data collection from on-premises virtual machines and EC2 instances. The following information is included:

- Operating system type and version
- Data for installed applications
- Network configuration information, such as MAC and IP addresses
- Amazon EC2 instance AMI ID and IAM profile

How can these needs be accomplished with the fewest possible administrative resources?

- A. Write a shell script to run as a cron job on EC2 instances to collect and push the data to Amazon S3. For on-premises resources, use VMware vSphere to collect the data and write it into a file gateway for storing the data in S3. Finally, use Amazon Athena on the S3 bucket for analytics.
- B. Use a script on the on-premises virtual machines as well as the EC2 instances to gather and push the data into Amazon S3, and then use Amazon Athena for analytics.
- C. Install AWS Systems Manager agents on both the on-premises virtual machines and the EC2 instances. Enable inventory collection and configure resource data sync to an Amazon S3 bucket to analyze the data with Amazon Athena.
- D. Use AWS Application Discovery Service for deploying Agentless Discovery Connector in the VMware environment and Discovery Agents on the EC2 instances for collecting the data. Then use the AWS Migration Hub Dashboard for analytics.

Commented [LC495]: D is wrong because Agentless is not enough for the information that are required.

<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-inventory-relatedsvc.html>

Question #535

Which of the following are not acceptable sources for custom OpsWorks cookbook repositories?

- A. HTTP(S)
- B. Git
- C. AWS EBS
- D. Subversion

Commented [LC496]: Linux stacks can install custom cookbooks from any of the following repository types: HTTP or Amazon S3 archives. They can be either public or private, but Amazon S3 is typically the preferred option for a private archive.

Git and Subversion repositories provide source control and the ability to have multiple versions.

Reference:
<http://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook-installingcustomenable.html>

Question #536

A DevOps Engineer is doing an evaluation of a system that makes use of Amazon EC2 instances organized in an Auto Scaling group. This system makes use of a configuration management tool that runs on each Amazon EC2 instance locally. Due to the volatility of the application demand, new instances must be completely functioning within three minutes after starting.

Among the current setup chores are the following:

- ⇒ Installing the configuration management agent ~ 2 minutes
- ⇒ Installing the application framework ~ 15 minutes
- ⇒ Copying configuration data from Amazon S3 ~ 2 minutes
- ⇒ Running the configuration management agent to configure instances ~ 1 minute
- ⇒ Deploying the application code from Amazon S3 ~ 2 minutes

How should the Engineer configure the system to achieve the required launch time?

- A. Trigger an AWS Lambda function from an Amazon CloudWatch Events rule when a new EC2 instance launches. Have the function install the configuration management agent and the application framework, pull configuration data from Amazon S3, run the agent to configure the instance, and deploy the application from S3.
- B. Write a bootstrap script to install the configuration management agent, install the application framework, pull configuration data from Amazon S3, run the agent to configure the instance, and deploy the application from S3.
- C. Build a custom AMI that includes the configuration management agent and application framework. Write a bootstrap script to pull configuration data from Amazon S3, run the agent to configure the instance, and deploy the application from S3.
- D. Build a custom AMI that includes the configuration management agent, application framework, and configuration data. Write a bootstrap script to run the agent to configure the instance and deploy the application from Amazon S3.

Commented [LC497]: Answer should be D, because two actions (4th and 5th) performed after launching the instance takes 3 minutes exactly.