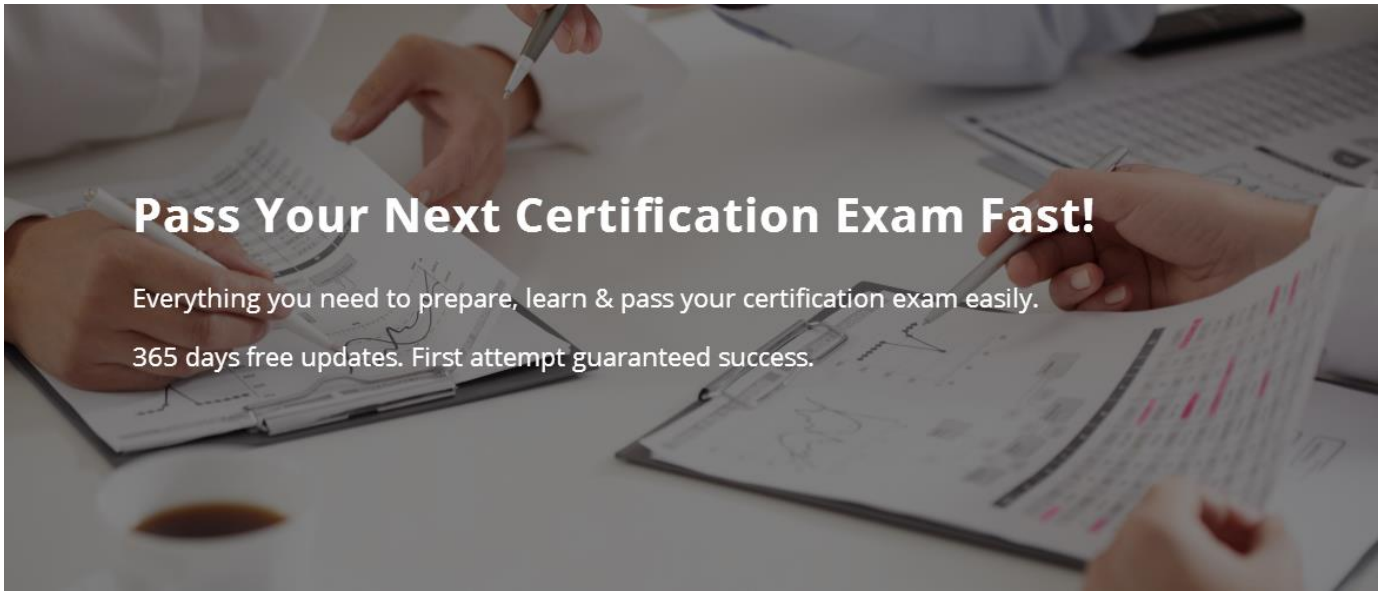
PrepPDF



Choose the version that fits your needs	PDF Version	Desktop Test Engine	Online Test Engine
Latest and Up-to-Date exam dumps with real exam questions answers.	✓	✓	✓
Get 12-Months free updates without any extra charges.	✓	✓	✓
Experience same exam environment before appearing in the certification exam.	✗	✓	✓
100% exam passing guarante in the first attempt.	✓	✓	✓
20% discount on more than one license and 30% discount on 5+ license purchases.	✗	✓	✓
100% secure purchase on SSL.	✓	✓	✓
Completely private purchase without sharing your personal info with anyone.	✓	✓	✓

http://www.preppdf.com

Reasonable study tool and effective study materials - PrepPDF

Exam : **AWS-DevOps-Engineer-Professional**

Title : AWS Certified DevOps Engineer - Professional

Vendor : Amazon

Version : DEMO

NO.1 A DevOps Engineer is building a continuous deployment pipeline for a serverless application using AWS CodePipeline and AWS CodeBuild. The source, build, and test stages have been created with the deploy stage remaining. The company wants to reduce the risk of an unsuccessful deployment by deploying to a specified subset of customers and monitoring prior to a full release to all customers.
How should the deploy stage be configured to meet 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: Canary10PercentI5Minutes.
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.
Answer: C
Explanation:
https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-configurations.html

NO.2 A company is deploying a new mobile game on AWS for its customers around the world. The Development team uses AWS Code services and must meet the following requirements:
- Clients need to send/receive real-time playing data from the backend
frequently and with minimal latency
- Game data must meet the data residency requirement
Which strategy can a DevOps Engineer implement to meet their needs?
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.
Answer: A
Explanation:
https://docs.aws.amazon.com/codepipeline/latest/userguide/integrations-action-

type.html#integrations-invoke

NO.3 Your company has developed a web application and is hosting it in an Amazon S3 bucket configured for static website hosting.
The application is using the AWS SDK for JavaScript in the browser to access data stored in an Amazon DynamoDB table.
How can you ensure that API keys for access to your data in DynamoDB are kept secure?
A. Create an Amazon S3 role in IAM with access to the specific DynamoDB tables, and assign it to the bucket hosting your website.
B. Configure S3 bucket tags with your AWS access keys for your bucket hosing your website so that the application can query them for access.
C. Configure a web identity federation role within IAM to enable access to the correct DynamoDB resources and retrieve temporary credentials.
D. Store AWS keys in global variables within your application and configure the application to use these credentials when making requests.
Answer: C

NO.4 You need to scale an RDS deployment. You are operating at 10% writes and 90% reads, based on your logging. How best can you scale this in a simple way?
A. Create a second master RDS instance and peer the RDS groups.
B. Cache all the database responses on the read side with CloudFront.
C. Create read replicas for RDS since the load is mostly reads.
D. Create a Multi-AZ RDS installs and route read traffic to standby.
Answer: C
Explanation:
The high-availability feature is not a scaling solution for read-only scenarios; you cannot use a standby replica to serve read traffic. To service read-only traffic, you should use a Read Replica.
For more information, see Working with PostgreSQL, MySQL, and MariaDB Read Replicas.
http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html

NO.5 A development team is using AWS CodeCommit to version control application code and AWS CodePipeline to orchestrate software deployments. The team has decided to use a remote master branch as the trigger for the pipeline to integrate code changes. A developer has pushed code changes to the CodeCommit repository, but noticed that the pipeline had no reaction, even after 10 minutes.
Which of the following actions should be taken to troubleshoot 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.
Answer: C

NO.6 A company runs a production application workload in a single AWS account that uses Amazon Route 53, AWS Elastic Beanstalk, and Amazon RDS. In the event of a security incident, the Security team wants the application workload to fail over to a new AWS account. The Security team also wants to block all access to the original account immediately, with no access to any AWS resources in the original AWS account, during forensic analysis.
What is the most cost-effective way to prepare to fail over to the second account 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.
Answer: B
Explanation:
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-configuration-savedconfig.html
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted-zones-migrating.html

NO.7 A DevOps Engineer is launching a new application that will be deployed using Amazon Route 53, an Application Load Balancer, Auto Scaling, and Amazon DynamoDB. One of the key requirements of this launch is that the application must be able to scale to meet a sudden load increase. During periods of low usage, the infrastructure components must scale down to optimize cost.
What steps can the DevOps Engineer take to meet the requirements? (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.
Answer: BC

Explanation:
D is wrong because Auto Scaling can terminate and replace any instances that are reported as unhealthy not ALB.
https://aws.amazon.com/blogs/database/amazon-dynamodb-auto-scaling-performance-and-cost-optimization-at-any-scale/

NO.8 A company has an application deployed using Amazon ECS with data stored in an Amazon DynamoDB table. The company wants the application to fail over to another Region in a disaster recovery scenario. The application must also efficiently recover from any accidental data loss events. The RPO for the application is 1 hour and the RTO is 2 hours.
Which highly available solution 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.
Answer: B

NO.9 A company is deploying a new application using Amazon EC2 instances. The company wants to maintain a centralized application and Amazon API logs that can be queried using one tool or service.
Which solution will meet these requirements?
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 Firehouse 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.
Answer: D

NO.10 A company is adopting AWS CodeDeploy to automate its application deployments for a Java-Apache Tomcat application with an Apache webserver. The Development team started with a proof of concept, created a deployment group for a developer environment, and performed functional tests within the application. After completion, the team will create additional deployment groups for staging and production The current log level is configured within the Apache settings, but the team wants to change this configuration dynamically when the deployment occurs, so that they can set different log level configurations depending on the deployment group without having a different application revision for each group.
How can these requirements be met with the LEAST management overhead and without requiring different 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 is 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.
Answer: B

NO.11 A DevOps engineer is developing an application for a company. The application needs to persist files to Amazon S3. The application needs to upload files with different security classifications that the company defines. These classifications include confidential, private, and public. Files that have a confidential classification must not be viewable by anyone other than the user who uploaded them. The application uses the IAM role of the user to call the S3 API operations.
The DevOps engineer has modified the application to add a DataClassificationtag with the value of confidentialand an Ownertag with the uploading user's ID to each confidential object that is uploaded to Amazon S3.
Which set of additional steps must the DevOps engineer take to meet the company's requirements?
A.
Modify the S3 bucket's ACL to grant bucket-owner-readaccess to the uploading user's IAM role. Create an IAM policy that grants s3:GetObjectoperations on the S3 bucket when aws:ResourceTag/DataClassificationequals confidential, and s3:ExistingObjectTag/Ownerequals ${aws:userid}. Attach the policy to the IAM roles for users who require access to the S3 bucket.
B.
Modify the S3 bucket policy to allow the s3:GetObject action when aws:ResourceTag/DataClassificationequals confidential, and s3:ExistingObjectTag/Ownerequals ${aws:userid}. Create an IAM policy that grants s3:GetObjectoperations on the S3 bucket. Attach the policy to the IAM roles for users who require

access to the S3 bucket.

C.

Modify the S3 bucket policy to allow the s3:GetObjectaction when aws:ResourceTag/DataClassificationequals confidential, and aws:RequesttTag/Ownerequals ${aws:userid}. Create an IAM policy that grants s3:GetObjectoperations on the S3 bucket. Attach the policy to the IAM roles for users who require access to the S3 bucket.

D.

Modify the S3 bucket's ACL to grant authenticated-readaccess when aws:ResourceTag/DataClassificationequals confidential, and s3:ExistingObjectTag/Ownerequals ${aws:userid}. Create an IAM policy that grants s3:GetObjectoperations on the S3 bucket. Attach the policy to the IAM roles for users who require access to the S3 bucket.

Answer: B

NO.12 You are in charge of a large-scale highly available multi-tier web application infrastructure. This architecture consists of Amazon Route53 with a load balancer and multiple Amazon EC2 instances.

You have been tasked to come up with a process to provide Blue/Green style deployments.

Which technique should you use to deliver 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 retort to point to the new load balancer.

Answer: B

NO.13 A company using AWS CodeCommit for source control wants to automate its continuous integration and continuous deployment pipeline on AWS in its development environment. The company has three requirements:

1. There must be a legal and a security review of any code change to make sure sensitive information is not leaked through the source code.
2. Every change must go through unit testing.
3. Every change must go through a suite of functional testing to ensure functionality.

In addition, the company has the following requirements for automation:
1. Code changes should automatically trigger the CI/CD pipellline.
2. Any failure in the pipeline should notify devops-admin@xyz.com.
3. There must be an approval to stage the assets to Amazon S3 after tests have been performed.
What should a DevOps Engineer do to meet all of these requirements while following CI/CD best practices?

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.

Answer: C

NO.14 A company is building a web and mobile application that uses a serverless architecture powered by AWS Lambda and Amazon API Gateway. 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.
The deployment must have the following:
- Separate environment pipelines for testing and production.
- Automatic deployment that occurs for test environments only.
Which steps should be taken to meet 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 a 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.

Answer: C

Explanation:

First, A&B both are in-correct: As a basic policy - do not create a repo for the same code for multiple environments. Always create a branch from the same repo. The strategy is wrong for A&B. Now C&D: D uses Lambda function with s3, whereas C uses code pipeline to store and build. Using code pipeline is a smart choice rather than using S3 as a code pipeline that offers better branching strategy and controls.

NO.15 A company wants to automatically re-create its infrastructure using AWS CloudFormation as part of the company's quality assurance (QA) pipeline. For each QA run, a new VPC must be created in a single account, resources must be deployed into the VPC, and tests must be run against this new infrastructure. The company policy states that all VPCs must be peered with a central management VPC to allow centralized logging. The company has existing CloudFormation templates to deploy its VPC and associated resources.

Which combination of steps will achieve the goal in a way that is automated and repeatable? (Choose 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, accepts 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.

Answer: AB

Explanation:

https://blog.irdeto.com/2017/10/11/how-to-implement-vpc-peering-between-2-vpcs-in-the-same-aws-account-using-cloudformation/

NO.16 Your company has multiple applications running on AWS.

Your company wants to develop a tool that notifies on-call teams immediately via email when an

alarm is triggered in your environment.
You have multiple on-cal teams that work different shifts, and the tool should handle notifying the correct teams at the correct times.
How should you implement this solution?

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.

Answer: D

NO.17 A company is creating a software solution that executes a specific parallel-processing mechanism. The software can scale to tens of servers in some special scenarios. This solution uses a proprietary library that is license-based, requiring that each individual server have a single, dedicated license installed. The company has 200 licenses and is planning to run 200 server nodes concurrently at most.
The company has requested the following features:
- A mechanism to automate the use of the licenses at scale.
- Creation of a dashboard to use in the future to verify which licenses are available at any moment.
What is the MOST effective way to accomplish 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 litecycle 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.
Answer: B
Explanation:
A dashboard to verify which licenses are available." is a key requirement. S3 and SQS won't provide you such a feature. "D" includes CLI scripts. CLI works in most cases, but it is never a right answer if an AWS native solution is available.

NO.18 What is the scope of an EC2 EIP?
A. Placement Group
B. Availability Zone
C. Region
D. VPC
Answer: C
Explanation:
An Elastic IP address is tied to a region and can be associated only with an instance in the same region.
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/resources.html

NO.19 An education company has a Docker-based application running on multiple Amazon EC2 instances in an Amazon ECS cluster. When deploying a new version of the application, the Developer, pushes a new image to a private Docker container registry, and then stops and starts all tasks to ensure that they all have the latest version of the application. The Developer discovers that the new tasks are, occasionally running with an old image.
How can this issue be prevented?
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.
Answer: C
Explanation:
https://docs.aws.amazon.com/en_us/AmazonECS/latest/developerguide/task_definition_parameters.html When a new task starts, the Amazon ECS container agent pulls the latest version of the specified image and tag for the container to use. However, subsequent updates to a repository image

are not propagated to already running tasks.

NO.20 A company is developing a web application's infrastructure using AWS CloudFormation. The database engineering team maintains the database resources in a CloudFormation template, and the software development team maintains the web application resources in a separate CloudFormation template. As the scope of the application grows, the software development team needs to use resources maintained by the database engineering team. However, both teams have their own review and lifecycle management processes that they want to keep. Both teams also require resource-level change-set reviews. The software development team would like to deploy changes to this template using their CI/CD pipeline.
Which solution will meet these requirements?
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.
Answer: A

NO.21 What is the maximum time messages can be stored in SQS?
A. 14 days
B. one month
C. 4 days
D. 7 days
Answer: A
Explanation:
A message can be stored in the Simple Queue Service (SQS) from 1 minute up to a maximum of 14 days.

NO.22 Your current log analysis application takes more than four hours to generate a report of the top 10 users of your web application.
You have been asked to implement a system that can report this information in real time, ensure that the report is always up to date, and handle increases in the number of requests to your web application. Choose the option that is cost-effective and can fulfill the requirements.
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 an 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 duster.

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.

Answer: C

NO.23 A DevOps Engineer is working with an application deployed to 12 Amazon EC2 instances across
3 Availability Zones. New instances can be started from an AMI image. On a typical day, each EC2 instance has 30% utilization during business hours and 10% utilization after business hours.
The CPU utilization has an immediate spike in the first few minutes of business hours. Other increases in CPU utilization rise gradually.
The Engineer has been asked to reduce costs while retaining the same or higher reliability.
Which solution meets these requirements?

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.

Answer: B

NO.24 What is server immutability?

A. Not updating a server after creation.

B. The ability to change server counts.

C. Updating a server after creation.

D. The inability to change server counts.

Answer: A

Explanation:
... disposable upgrades offer a simpler way to know if your application has unknown dependencies. The underlying EC2 instance usage is considered temporary or ephemeral in nature for the period of deployment until the current release is active. During the new release, a new set of EC2 instances are rolled out by terminating older instances. This type of upgrade technique is more common in an immutable infrastructure.
https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf

NO.25 A DevOps engineer is assisting with a multi-Region disaster recovery solution for a new application. The application consists of Amazon EC2 instances running in an Auto Scaling group and

an Amazon Aurora MySQL DB cluster. The application must be available with an RTO of 120 minutes and an RPO of 60 minutes.
What is the MOST cost-effective way to meet these requirements?
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.
Answer: C
Explanation:
https://d1.awsstatic.com/training-and-certification/docs-devops-pro/AWS-Certified-DevOps-Engineer-Professional_Sample-Questions.pdf

NO.26 You have a playbook that includes a task to install a package for a service, put a configuration file for that package on the system and restart the service. The playbook is then run twice in a row.
What would you expect Ansible to do on the 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.
Answer: C
Explanation:
Ansible follows an idempotence model and will not touch or change the system unless a change is warranted.
Reference: http://docs.ansible.com/ansible/glossary.html

NO.27 When running a playbook on a remote target host you receive a Python error similar to "[Errno

13] Permission denied: `/home/nick/.ansible/tmp'. What would be the most likely cause of this problem?

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

Answer: D

Explanation:

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

NO.28 You run a clustered NoSQL database on AWS EC2 using AWS EBS. You need to reduce latency for database response times. Performance is the most important concern, not availability. You did not perform the initial setup, someone without much AWS knowledge did, so you are not sure if they configured everything optimally. Which of the following is NOT likely to be an issue contributing to increased 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.

Answer: B

Explanation:

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.

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

NO.29 A company has a web application that uses an Amazon DynamoDB table in a single AWS Region to store user information. To support an increasingly global user base, the application must run in a secondary Region and allow users to connect to their closest Region and fail over to the secondary Region.

Which approach should be used to ensure the deployment meets these requirements?

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.

Answer: B

Explanation:

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

NO.30 Which deployment method, when using AWS Auto Scaling Groups and Auto Scaling Launch Configurations, enables the shortest time to live for individual servers?

A. Pre-baking AMIs with all code and configuration on deploys.

B. Using a Dockerfile bootstrap on instance launch.

C. Using UserData bootstrapping scripts.

D. Using AWS EC2 Run Commands to dynamically SSH into fleets.

Answer: A

Explanation:

Note that 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. Prebaking is a process of embedding a significant portion of your application artifacts within your base AMI. During the deployment process you can customize application installations by using EC2 instance artifacts such as instance tags, instance metadata, and Auto Scaling groups.

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

NO.31 You are getting a lot of empty receive requests when using Amazon SQS.
This is making a lot of unnecessary network load on your instances.
What can you do to reduce this load?

A. Subscribe your queue to an SNS topic instead.

B. Use as long of a poll as possible, instead of short polls.

C. Alter your visibility timeout to be shorter.

D. Use <code>sqsd</code> on your EC2 instances.

Answer: B

Explanation:

One benefit of long polling with Amazon SQS is the reduction of the number of empty responses, when there are no messages available to return, in reply to a ReceiveMessage request sent to an Amazon SQS queue. Long polling allows the Amazon SQS service to wait until a message is available in the queue before sending a response.

http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-long-polling.html

NO.32 Consider the portion of a CloudTrail log file below. Which type of event is being captured?

"eventTime":"2016-07-16T17:35:32Z",
"eventSource":"signin.amazonaws.com",
"eventName":"ConsoleLogin",

"awsRegion":"us-west-1",
"sourceIPAddress":"192.1.2.10",
...

A. AWS console sign-in

B. AWS log off

C. AWS error

D. AWS deployment

Answer: A

Explanation:
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-aws-console-sign-in-events.html

NO.33 Your CTO has asked you to make sure that you know what all users of your AWS account are doing to change resources at all times. She wants a report of who is doing what over time, reported to her once per week, for as broad a resource type group as possible. How should you do this?

A. Create a global AWS CloudTrail Trail. Configure a script to aggregate the log data into a report, publish it 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.

Answer: A

Explanation:
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/

NO.34 A company is using AWS CodeDeploy to manage its application deployments. Recently, the Development team decided to use GitHub for version control, and the team is looking for ways to integrate the GitHub repository with CodeDeploy. The team also needs to develop a way to automate deployment whenever there is a new commit on that repository. The team is currently deploying new application revisions by manually indicating the Amazon S3 location.
How can the integration be achieved in the MOST efficient way?

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.

Answer: B

Explanation:

The team also needs to develop a way to automate deployment whenever there is a new commit on that repository.

https://docs.aws.amazon.com/codedeploy/latest/userguide/integrations-partners-github.html#behaviors-deploy-automatically

NO.35 Your application uses CloudFormation to orchestrate your application's resources. During your testing phase before the application went live, your Amazon RDS instance type was changed and caused the instance to be re-created, resulting In the loss of test data.

How should you prevent this from occurring in the future?

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."

Answer: E