Free Questions for DOP-C02 by dumpssheet

Shared by Cote on 22-07-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company uses Amazon EC2 as its primary compute platform. A DevOps team wants to audit the company's EC2 instances to check whether any prohibited applications have been installed on the EC2 instances.

Which solution will meet these requirements with the MOST operational efficiency?

Options:

A- Configure AWS Systems Manager on each instance Use AWS Systems Manager Inventory Use Systems Manager resource data sync to synchronize and store findings in an Amazon S3 bucket Create an AWS Lambda function that runs when new objects are added to the S3 bucket. Configure the Lambda function to identify prohibited applications.

B- Configure AWS Systems Manager on each instance Use Systems Manager Inventory Create AWS Config rules that monitor changes from Systems Manager Inventory to identify prohibited applications.

C- Configure AWS Systems Manager on each instance. Use Systems Manager Inventory. Filter a trail in AWS CloudTrail for Systems Manager Inventory events to identify prohibited applications.

D- Designate Amazon CloudWatch Logs as the log destination for all application instances Run an automated script across all instances to create an inventory of installed applications Configure the script to forward the results to CloudWatch Logs Create a CloudWatch alarm that uses filter patterns to search log data to identify prohibited applications.

Answer:

A

Explanation:

* Configure AWS Systems Manager on Each Instance:

AWS Systems Manager provides a unified interface for managing AWS resources. Install the Systems Manager agent on each EC2 instance to enable inventory management and other features.

* Use AWS Systems Manager Inventory:

Systems Manager Inventory collects metadata about your instances and the software installed on them. This data includes information about applications, network configurations, and more.

Enable Systems Manager Inventory on all EC2 instances to gather detailed information about installed applications.

* Use Systems Manager Resource Data Sync to Synchronize and Store Findings in an Amazon S3 Bucket:

Resource Data Sync aggregates inventory data from multiple accounts and regions into a single S3 bucket, making it easier to query and analyze the data.

Configure Resource Data Sync to automatically transfer inventory data to an S3 bucket for centralized storage.

* Create an AWS Lambda Function that Runs When New Objects are Added to the S3 Bucket:

Use an S3 event to trigger a Lambda function whenever new inventory data is added to the S3 bucket.

The Lambda function can parse the inventory data and check for the presence of prohibited applications.

* Configure the Lambda Function to Identify Prohibited Applications:

The Lambda function should be programmed to scan the inventory data for any known prohibited applications and generate alerts or take appropriate actions if such applications are found.

Example Lambda function in Python

```
import json

import boto3

def lambda_handler(event, context):

s3 = boto3.client('s3')

bucket = event['Records'][0]['s3']['bucket']['name']

key = event['Records'][0]['s3']['object']['key']

response = s3.get_object(Bucket=bucket, Key=key)

inventory_data = json.loads(response['Body'].read().decode('utf-8'))

prohibited_apps = ['app1', 'app2']

for instance in inventory_data['Instances']:
```

```
for app in instance['Applications']:

if app['Name'] in prohibited_apps:

# Send notification or take action

print(f'Prohibited application found: {app['Name']} on instance {instance['InstanceId']}')

return {'statusCode': 200, 'body': json.dumps('Check completed')}
```

By leveraging AWS Systems Manager Inventory, Resource Data Sync, and Lambda, this solution provides an efficient and automated way to audit EC2 instances for prohibited applications.

AWS Systems Manager Inventory

AWS Systems Manager Resource Data Sync

S3 Event Notifications

AWS Lambda

Question 2

Question Type: **MultipleChoice**

A company runs applications on Windows and Linux Amazon EC2 instances The instances run across multiple Availability Zones In an AWS Region. The company uses Auto Scaling groups for each application.

The company needs a durable storage solution for the instances. The solution must use SMB for Windows and must use NFS for Linux. The solution must also have sub-millisecond latencies. All instances will read and write the data.

Which combination of steps will meet these requirements? (Select THREE.)

Options:

A- Create an Amazon Elastic File System (Amazon EFS) file system that has targets in multiple Availability Zones

B- Create an Amazon FSx for NetApp ONTAP Multi-AZ file system.

C- Create a General Purpose SSD (gp3) Amazon Elastic Block Store (Amazon EBS) volume to use for shared storage.

D- Update the user data for each application's launch template to mount the file system

E- Perform an instance refresh on each Auto Scaling group.

F- Update the EC2 instances for each application to mount the file system when new instances are launched

Answer:

A, B, D

Explanation:

* Create an Amazon Elastic File System (Amazon EFS) File System with Targets in Multiple Availability Zones:

Amazon EFS provides a scalable and highly available network file system that supports the NFS protocol. EFS is ideal for Linux instances as it allows multiple instances to read and write data concurrently.

Setting up EFS with targets in multiple Availability Zones ensures high availability and durability.

* Create an Amazon FSx for NetApp ONTAP Multi-AZ File System:

Amazon FSx for NetApp ONTAP offers a fully managed file storage solution that supports both SMB for Windows and NFS for Linux.

The Multi-AZ deployment ensures high availability and durability, providing sub-millisecond latencies suitable for the application's performance requirements.

* Update the User Data for Each Application's Launch Template to Mount the File System:

Updating the user data in the launch template ensures that every new instance launched by the Auto Scaling group will automatically mount the appropriate file system.

This step is necessary to ensure that all instances can access the shared storage without manual intervention.

Example user data for mounting EFS (Linux)

#!/bin/bash

sudo yum install -y amazon-efs-utils

sudo mount -t efs fs-12345678:/ /mnt/efs

Example user data for mounting FSx (Windows):

By implementing these steps, the company can provide a durable storage solution with sub-millisecond latencies that supports both SMB and NFS protocols, meeting the requirements for both Windows and Linux instances.

Mounting EFS File Systems

Mounting Amazon FSx File Systems

Question 3

Question Type: **MultipleChoice**

A software team is using AWS CodePipeline to automate its Java application release pipeline The pipeline consists of a source stage, then a build stage, and then a deploy stage. Each stage contains a single action that has a runOrder value of 1.

The team wants to integrate unit tests into the existing release pipeline. The team needs a solution that deploys only the code changes that pass all unit tests.

Which solution will meet these requirements?

Options:

A- Modify the build stage. Add a test action that has a runOrder value of 1. Use AWS CodeDeploy as the action provider to run unit tests.

B- Modify the build stage Add a test action that has a runOrder value of 2 Use AWS CodeBuild as the action provider to run unit tests

C- Modify the deploy stage Add a test action that has a runOrder value of 1 Use AWS CodeDeploy as the action provider to run unit tests

D- Modify the deploy stage Add a test action that has a runOrder value of 2 Use AWS CodeBuild as the action provider to run unit tests

Answer:

B

Explanation:

* Modify the Build Stage to Add a Test Action with a RunOrder Value of 2:

The build stage in AWS CodePipeline can have multiple actions. By adding a test action with a runOrder value of 2, the test action will execute after the initial build action completes.

* Use AWS CodeBuild as the Action Provider to Run Unit Tests:

AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages.

Using CodeBuild to run unit tests ensures that the tests are executed in a controlled environment and that only the code changes that pass the unit tests proceed to the deploy stage.

Example configuration in CodePipeline:

```
{
'name': 'BuildStage',
'actions': [
{
'name': 'Build',
'actionTypeId': {
'category': 'Build',
'owner': 'AWS',
'provider': 'CodeBuild',
'version': '1'
},
'runOrder': 1
},
{
'name': 'Test',
```

'actionTypeId': {

'category': 'Test',

'owner': 'AWS',

'provider': 'CodeBuild',

'version': '1'

},

'runOrder': 2

}

]

}

By integrating the unit tests into the build stage and ensuring they run after the build process, the pipeline guarantees that only code changes passing all unit tests are deployed.

AWS CodePipeline

AWS CodeBuild

Using CodeBuild with CodePipeline

Question 4

Question Type: MultipleChoice

A company uses containers for its applications The company learns that some container Images are missing required security configurations

A DevOps engineer needs to implement a solution to create a standard base image The solution must publish the base image weekly to the us-west-2 Region, us-east-2 Region, and eu-central-1 Region.

Which solution will meet these requirements?

Options:

A- Create an EC2 Image Builder pipeline that uses a container recipe to build the image. Configure the pipeline to distribute the image to an Amazon Elastic Container Registry (Amazon ECR) repository in us-west-2. Configure ECR replication from us-west-2 to us-east-2 and from us-east-2 to eu-central-1 Configure the pipeline to run weekly

B- Create an AWS CodePipeline pipeline that uses an AWS CodeBuild project to build the image Use AWS CodeOeploy to publish the image to an Amazon Elastic Container Registry (Amazon ECR) repository in us-west-2 Configure ECR replication from us-west-2 to us-east-2 and from us-east-2 to eu-central-1 Configure the pipeline to run weekly

C- Create an EC2 Image Builder pipeline that uses a container recipe to build the Image Configure the pipeline to distribute the image to Amazon Elastic Container Registry (Amazon ECR) repositories in all three Regions. Configure the pipeline to run weekly.

D- Create an AWS CodePipeline pipeline that uses an AWS CodeBuild project to build the image Use AWS CodeDeploy to publish the

image to Amazon Elastic Container Registry (Amazon ECR) repositories in all three Regions. Configure the pipeline to run weekly.

Answer:

C

Explanation:

Create an EC2 Image Builder Pipeline that Uses a Container Recipe to Build the Image:

EC2 Image Builder simplifies the creation, maintenance, validation, and sharing of container images.

By using a container recipe, you can define the base image, components, and validation tests for your container image.

Configure the Pipeline to Distribute the Image to Amazon Elastic Container Registry (Amazon ECR) Repositories in All Three Regions:

Amazon ECR provides a secure, scalable, and reliable container registry.

Configuring the pipeline to distribute the image to ECR repositories in us-west-2, us-east-2, and eu-central-1 ensures that the image is available in all required regions.

Configure the Pipeline to Run Weekly:

Setting the pipeline to run on a weekly schedule ensures that the base image is regularly updated and published, incorporating any new security configurations or updates.

By using EC2 Image Builder to automate the creation and distribution of the container image, the solution ensures that the base image is consistently maintained and available across multiple regions with minimal management overhead.

EC2 Image Builder

Amazon ECR

Setting Up EC2 Image Builder Pipelines

Question 5

Question Type: **MultipleChoice**

A company's DevOps team manages a set of AWS accounts that are in an organization in AWS Organizations

The company needs a solution that ensures that all Amazon EC2 instances use approved AMIs that the DevOps team manages. The solution also must remediate the usage of AMIs that are not approved The individual account administrators must not be able to remove the restriction to use approved AMIs.

Which solution will meet these requirements?

Options:

A- Use AWS CloudFormation StackSets to deploy an Amazon EventBridge rule to each account. Configure the rule to react to AWS CloudTrail events for Amazon EC2 and to send a notification to an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the DevOps team to the SNS topic

B- Use AWS CloudFormation StackSets to deploy the approved-amis-by-id AWS Config managed rule to each account. Configure the rule with the list of approved AMIs. Configure the rule to run the the AWS-StopEC2Instance AWS Systems Manager Automation runbook for the noncompliant EC2 instances.

C- Create an AWS Lambda function that processes AWS CloudTrail events for Amazon EC2 Configure the Lambda function to send a notification to an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the DevOps team to the SNS topic. Deploy the Lambda function in each account in the organization Create an Amazon EventBridge rule in each account Configure the EventBridge rules to react to AWS CloudTrail events for Amazon EC2 and to invoke the Lambda function.

D- Enable AWS Config across the organization Create a conformance pack that uses the approved -amis-by-id AWS Config managed rule with the list of approved AMIs. Deploy the conformance pack across the organization. Configure the rule to run the AWS-StopEC2Instance AWS Systems Manager Automation runbook for the noncompliant EC2 instances.

Answer:

D

Explanation:

Enable AWS Config Across the Organization:

AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. It can be used to assess, audit, and evaluate the configurations of your resources.

Enabling AWS Config across the organization ensures that all accounts are monitored for compliance.

Create a Conformance Pack Using the approved-amis-by-id AWS Config Managed Rule:

A conformance pack is a collection of AWS Config rules and remediation actions that can be easily deployed across an organization.

The approved-amis-by-id managed rule checks whether running instances are using approved AMIs.

Deploy the Conformance Pack Across the Organization:

Deploying the conformance pack across the organization ensures that all accounts adhere to the policy of using only approved AMIs.

The conformance pack can be deployed via the AWS Management Console, CLI, or SDKs.

Configure the Rule to Run the AWS-StopEC2Instance AWS Systems Manager Automation Runbook for Non-Compliant EC2 Instances:

The AWS-StopEC2Instance runbook can be configured to automatically stop any EC2 instances that are found to be non-compliant (i.e., not using approved AMIs).

This remediation action ensures that any unauthorized instances are promptly stopped, enforcing the policy without manual intervention.

By following these steps, the solution ensures that all EC2 instances across the organization use approved AMIs, and any non-compliant instances are remediated automatically.

AWS Config Conformance Packs

AWS Config Managed Rules

AWS Systems Manager Automation Runbooks

Question 6

Question Type: MultipleChoice

A company hired a penetration tester to simulate an internal security breach The tester performed port scans on the company's Amazon EC2 instances. The company's security measures did not detect the port scans.

The company needs a solution that automatically provides notification when port scans are performed on EC2 instances. The company creates and subscribes to an Amazon Simple Notification Service (Amazon SNS) topic.

What should the company do next to meet the requirement?

Options:

A- Ensure that Amazon GuardDuty is enabled Create an Amazon CloudWatch alarm for detected EC2 and port scan findings. Connect the alarm to the SNS topic.

B- Ensure that Amazon Inspector is enabled Create an Amazon EventBridge event for detected network reachability findings that indicate port scans Connect the event to the SNS topic.

C- Ensure that Amazon Inspector is enabled. Create an Amazon EventBridge event for detected CVEs that cause open port vulnerabilities. Connect the event to the SNS topic

D- Ensure that AWS CloudTrail is enabled Create an AWS Lambda function to analyze the CloudTrail logs for unusual amounts of traffic

from an IP address range Connect the Lambda function to the SNS topic.

Answer:

A

Explanation:

* Ensure that Amazon GuardDuty is Enabled:

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior.

It can detect port scans and generate findings for these events.

* Create an Amazon CloudWatch Alarm for Detected EC2 and Port Scan Findings:

Configure GuardDuty to monitor for port scans and other threats.

Create a CloudWatch alarm that triggers when GuardDuty detects port scan activities.

* Connect the Alarm to the SNS Topic:

The CloudWatch alarm should be configured to send notifications to the SNS topic subscribed by the security team.

This setup ensures that the security team receives near-real-time notifications when a port scan is detected on the EC2 instances.

Example configuration steps:

Enable GuardDuty and ensure it is monitoring the relevant AWS accounts.

Create a CloudWatch alarm:

```
{

'AlarmName': 'GuardDutyPortScanAlarm',

'MetricName': 'ThreatIntelIndicator',

'Namespace': 'AWS/GuardDuty',

'Statistic': 'Sum',

'Dimensions': [

{

'Name': 'FindingType',

'Value': 'Recon:EC2/Portscan'

}

],

'Period': 300,

'EvaluationPeriods': 1,
```

'Threshold': 1,

'ComparisonOperator': 'GreaterThanOrEqualToThreshold',

'AlarmActions': ['arn:aws:sns:region:account-id:SecurityAlerts']

}

Amazon GuardDuty

Creating CloudWatch Alarms for GuardDuty Findings