

7-simulações profissionais

Notebook: solutions_profissional_ef8e1c0f-5d3c-42ab-ab04-9217e53e12cd

Created: 19/10/2025 18:37

Updated: 21/10/2025 02:39

Author: erikachen19@gmail.com

URL: <https://portal.tutorialsdojo.com/courses/aws-certified-solutions-architect-professional...>

A company is planning to launch a mobile app for the Department of Transportation that allows government staff to upload the latest photos of ongoing construction works such as bridges, roads, culverts, and dams all over the country. The mobile app should send the photos to a web server hosted on an EC2 instance which then adds a watermark to each photo that contains the project details and the date it was taken. The solutions architect must design a solution in which the photos generated by the server will be uploaded to an S3 bucket for durable storage.

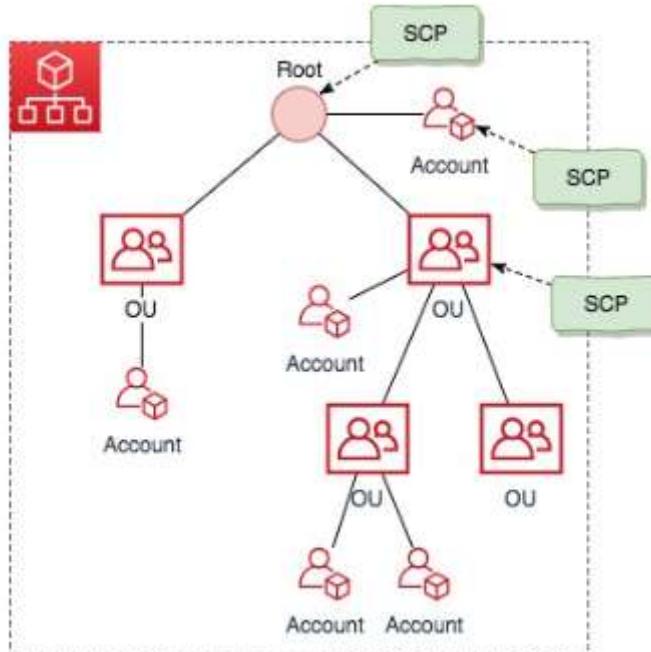
Which of the following solutions is a secure architecture and allows the EC2 instance to upload photos to S3?

(view)

- Set up a service control policy (SCP) with permissions to list and write objects to the S3 bucket. Attach the SCP to the EC2 instance which will enable it to retrieve temporary security credentials from the instance metadata and use that access to upload the photos to the S3 bucket.
- Set up an IAM user with permissions to list and write objects to the S3 bucket. Launch the instance as the IAM user which will enable the EC2 instance to retrieve temporary security credentials from the instance userdata and use that access to upload the photos to the S3 bucket.
- Set up an IAM role with permissions to list and write objects to the S3 bucket. Attach the IAM role to the EC2 instance which will enable it to retrieve temporary security credentials from the instance metadata and use that access to upload the photos to the S3 bucket.
- Set up an IAM user with permissions to list and write objects to the S3 bucket. Attach the IAM role to the EC2 instance which will enable it to retrieve temporary security credentials from the instance metadata and use that access to upload the photos to the S3 bucket.

1 0 1 00:00:00

This question tests your understanding of IAM, specifically on when to use an IAM Role vs an SCP. Since the server is running on an EC2 instance and the application makes requests to S3 to store the photos, the more suitable option to use here is an IAM Role.



In addition, don't create an IAM user and pass the user's credentials to the application or embed the credentials in the application. That will create a security risk because if an attacker had unauthorized access to that EC2 instance then the user credentials can easily be acquired and exploited. The better way is to create an IAM role that you can attach to the EC2 instance to give applications running on the instance temporary security credentials which can be used to access other AWS resources such as an S3 bucket. The credentials have the permissions specified in the policies attached to the role.

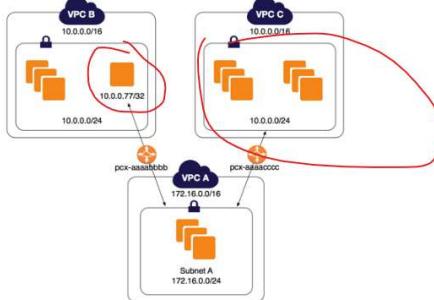
The option that says: Set up an IAM role with permissions to list and write objects to the S3 bucket. Attach the IAM role to the EC2 instance which will enable it to retrieve temporary security credentials from the instance metadata and use that access to upload the photos to the S3 bucket is correct as it uses an IAM Role and fetches the temporary security credentials from the instance metadata.

The option that says: Set up a service control policy (SCP) with permissions to list and write objects to the S3 bucket. Attach the SCP to the EC2 instance which will enable it to retrieve temporary security credentials from the instance metadata and use that access to upload the photos to the S3 bucket is incorrect as SCPs simply enable you to restrict, at the account level of granularity, what services and actions the users, groups, and roles in those accounts can do. SCPs don't grant permissions to any user or role because this is handled through IAM policies.

The option that says: Set up an IAM user with permissions to list and write objects to the S3 bucket. Launch the instance as the IAM user which will enable the EC2 instance to retrieve temporary security credentials from the instance userdata and use that access to upload the photos to the S3 bucket is incorrect as an IAM User is a better option to use instead of an IAM User. Plus, you should always retrieve the temporary security credentials from the instance metadata and not from the user data.

The option that says: Set up an IAM service role with permissions to list and write objects to the S3 bucket. Attach the IAM role to the EC2 instance which will enable it to retrieve temporary security credentials from the instance userdata and use that access to upload the photos to the S3 bucket is incorrect because although it uses an IAM Role, the temporary security credentials should be retrieved from the instance metadata and not from the user data.

A company has three AWS accounts each with its own VPCs. There is a requirement for communication between the AWS resources across the accounts, so VPC peering needs to be configured. Please refer to the figure below for details of each VPC:



VPC-B and VPC-C have matching CIDR blocks. For a short-term requirement, VPC-A needs to communicate only with the database instance in VPC-B with an IP address of 10.0.0.77/32 while being able to communicate with all the resources in VPC-C. The Solutions Architect already created the necessary VPC peering links but VPC-A cannot effectively communicate to the VPC-B instance. The Solutions Architect suspects that the routes on each VPC still need proper configuration.

Which of the following solutions will allow VPC-A to communicate with the database instance in VPC-B while being able to communicate with all resources on VPC-C?

- On VPC-A, add a static route for VPC-B CIDR (10.0.0.0/24) with the target `pca-aaaaabbb` and another static route for VPC-C CIDR (10.0.0.0/24) with the target `pca-aaaaaccc`. Add a network access control list (NACL) on VPC-A to deny all connections to VPC-B except for the IP address 10.0.0.77/32 . On VPC-B, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaabbb`. On VPC-C, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaaccc`.
- Enable dynamic route propagation in VPC-A with the peering target `pca-aaaaabbb` and `pca-aaaaaccc` respectively. On VPC-C, enable dynamic route propagation with peering target `pca-aaaaabbb` and add a network access control list (NACL) that allows only connections to IP address 10.0.0.77/32 from `pca-aaaaabbb`. On VPC-C, enable dynamic route propagation with the peering target `pca-aaaaaccc`.
- On VPC-A, add a static route for VPC-B CIDR (10.0.0.0/24) with the target `pca-aaaaabbb` and another static route for VPC-C CIDR (10.0.0.0/16) with the target `pca-aaaaaccc`. On VPC-B, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaaccc`. On VPC-C, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaaccc`.
- On VPC-A, add a static route for VPC-B CIDR (10.0.0.0/24) with the target `pca-aaaaabbb` and another static route for VPC-C CIDR (10.0.0.0/16) with the target `pca-aaaaaccc`. On VPC-B, add a static route for VPC-A CIDR (172.16.0.0/16) with the target `pca-aaaaaccc`. On VPC-C, add a static route for VPC-A CIDR (172.16.0.0/16) with the target `pca-aaaaaccc`.

A (+ 10.0.0.77/32), C + 10.0.0/16
 B + 172.16.0.0/16
 C + 172.16.0.0/16
 A
 C + 172.16.0.0/16
 X

all, vpc , nao subnet.

If you have a VPC peered with multiple VPCs that have overlapping or matching CIDR blocks, ensure that your route tables are configured to avoid sending response traffic from your VPC to the incorrect VPC. AWS currently does not support unicast reverse path forwarding in VPC peering connections that check the source IP of packets and route reply packets back to the source. You still need to configure static routes on VPC-B and VPC-C going to VPC-A, respectively.

Therefore, the correct answer is: On VPC-A, add a static route for VPC-B CIDR (10.0.0.77/32) with the target `pca-aaaaabbb` and another static route for VPC-C CIDR (10.0.0.0/16) with the target `pca-aaaaaccc`. On VPC-B, add a static route for VPC-A CIDR (172.16.0.0/16) with the target `pca-aaaaabbb`. On VPC-C, add a static route for VPC-A CIDR (172.16.0.0/16) with the target `pca-aaaaaccc`. The standard VPC peering configuration will be done for VPC-A and VPC-C. As for VPC-B, only the static route to the specific should be configured on VPC-A. AWS will handle the longest prefix match to route the traffic.

The option that says: On VPC-A, add a static route for VPC-B CIDR (10.0.0.0/24) with the target `pca-aaaaabbb` and another static route for VPC-C CIDR (10.0.0.0/24) with the target `pca-aaaaaccc`. Add a network access control list (NACL) on VPC-A to deny all connections to VPC-B except for the IP address 10.0.0.77/32 . On VPC-B, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaabbb`. On VPC-C, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaaccc`. Is incorrect. This will only result in a conflict on the routing configuration on VPC-A. Network ACLs can block connections going out of your VPC, however, they can't redirect connections out to specific targets.

The option that says: Enable dynamic route propagation in VPC-A with the peering targets `pca-aaaaabbb` and `pca-aaaaaccc` respectively. On VPC-B, enable dynamic route propagation with peering target `pca-aaaaabbb` and add a network access control list (NACL) that allows only connections to IP address 10.0.0.77/32 from `pca-aaaaabbb`. On VPC-C, enable dynamic route propagation with the peering target `pca-aaaaaccc`. Is incorrect. Dynamic route propagation is primarily used for Direct Connect connections or Site-to-Site VPNs. In this scenario, you want to force a specific route to a specific instance on a specific peering target, thus, you need to configure static routes.

The option that says: On VPC-A, add a static route for VPC-B CIDR (10.0.0.0/24) with the target `pca-aaaaabbb` and another static route for VPC-C CIDR (10.0.0.0/16) with the target `pca-aaaaaccc`. On VPC-B, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaabbb`. On VPC-C, add a static route for VPC-A CIDR (172.16.0.0/24) with the target `pca-aaaaaccc`. Is incorrect. Route configuration for VPC-A and VPC-C is correct here, however, the route configuration of VPC-A and VPC-B is not. This will make traffic to other instances in the CIDR (10.0.0.0/24), which is under VPC-C, to be routed to VPC-B.

A company is running hundreds of Linux-based Amazon EC2 instances launched with custom AMIs that are dedicated to specific products and services. As part of the security compliance requirements, vulnerability scanning must be done on all EC2 instances wherein each instance must be scanned and pass a Common Vulnerabilities and Exposures (CVE) assessment. Since the development team relies heavily on the custom AMIs for their deployments, the company wants to have an automated process to run the security assessment on any new AMIs and properly tag them before they can be used by the developers. To ensure continuous compliance, the security-approved AMIs must also be scanned every 30 days to check for new vulnerabilities and apply the necessary patches.

Which of the following steps should the Solutions Architect implement to achieve the security requirements? (Select TWO.)

- | | |
|--------------------------|---|
| <input type="checkbox"/> | Install the AWS Systems Manager (SSM) agent on all EC2 instances. With the agent running, run a detailed CVE assessment scan on the EC2 instances launched from the AMIs that need scanning. |
| <input type="checkbox"/> | Deploy a Lambda function that will create a domain <code>aws-vuln-scan</code> . Create a parameter on AWS SSM Parameter Store. Set up a 30-day interval cron rule on Amazon EventBridge to trigger an AWS SSM Automation document run on all EC2 instances. |
| <input type="checkbox"/> | Create an Assessment template on Amazon Inspector to target the EC2 instances. Run a detailed CVE assessment scan on all running Amazon EC2 instances launched from the AMIs that need scanning. |
| <input type="checkbox"/> | Write a Lambda function that will create automatic approval rules. Create a parameter on AWS SSM Parameter Store to save the list of all security-approved AMI. Set up a managed rule on AWS Config to continuously scan all running EC2 instances. For any detected vulnerability, run the designated SSM Automation document. |
| <input type="checkbox"/> | Check AWS CloudTrail logs to determine the Amazon EC2 instance IDs that were launched from the AMIs that need scanning. Use AWS Config managed rule to run CVE assessment and remediation on the instances. |

What is a CVE assessment scan?

A **CVE assessment scan** is the process of scanning systems, applications, containers, or code for known security vulnerabilities that are tracked by **CVE identifiers** (Common Vulnerabilities and Exposures). The aim is to discover

which CVEs affect your assets, assess risk (severity/impact), and prioritize remediation.

AWS Systems Manager Automation simplifies common maintenance and deployment tasks of **Amazon EC2 instances** and other AWS resources. Automation enables you to do the following:

- Build automations to configure and manage instances and AWS resources.
- Create custom runbooks or use pre-defined runbooks maintained by AWS.
- Receive notifications about Automation tasks and runbooks by using Amazon EventBridge.
- Monitor Automation progress and details by using the AWS Systems Manager console.

SSM Automation offers one-click automation for simplifying complex tasks such as creating golden Amazon Machines Images (AMIs) and recovering unreachable EC2 instances. For example, you can use the `AWS-UpdateLinuxAmi` and `AWS-UpdateWindowsAmi` runbooks to create golden AMIs from a source AMI. You can run custom scripts before and after updates are applied. You can also include or exclude specific packages from being installed.

With **AWS EventBridge**, you can create rules that self-trigger on an automated schedule in EventBridge using cron or rate expressions. Rate expressions are simpler to define but don't offer the fine-grained schedule control that cron expressions support. For example, with a cron expression, you can define a rule that triggers at a specified time on a certain day of each week or month. With this, **you can schedule running AWS SSM Automation documents to remediate the vulnerable AMIs**.

You can use **Amazon Inspector** 亚马逊检查员 to conduct a detailed scan for CVE in your fleet of EC2 instances. Amazon Inspector offers **predefined software called an agent that you can optionally install in the operating system of the EC2 instances that you want to assess**. Amazon Inspector also has rules packages that help verify whether the EC2 instances in your assessment 评估 targets are exposed to common vulnerabilities and exposures (CVEs). Attacks can exploit unpatched vulnerabilities to compromise the confidentiality, integrity, or availability of your service or data. The CVE system provides a reference method for publicly known information security vulnerabilities and exposures.

The option that says: *Create a Lambda function that will create automatic approval rules. Create a parameter on AWS SSM Parameter Store to save the list of all security-approved AMI. Set up a 30-day interval cron rule on Amazon EventBridge to trigger an AWS SSM Automation document run on all EC2 instances* is correct because it satisfies the requirement for updating the security-approved AMI, along with scheduled patches every 30-days using SSM Automation document. AWS SSM Automation can automatically pack AMIs after patches are applied.

The option that says: *Create an Assessment template on Amazon Inspector to target the EC2 instances. Run a detailed CVE assessment scan on all running Amazon EC2 instances launched from the AMIs that need scanning* is correct because Amazon Inspector can run assessments on target EC2 instances to check if they are exposed to common vulnerabilities and exposures (CVEs).

The option that says: *Install the AWS Systems Manager (SSM) agent on all EC2 instances. With the agent running, run a detailed CVE assessment scan on the EC2 instances launched from the AMIs that need scanning* is incorrect because the SSM agent cannot run a detailed CVE assessment scan on EC2 Instances. You have to use Amazon Inspector to satisfy the given requirement.

The option that says: *Write a Lambda function that will create automatic approval rules. Create a parameter on AWS SSM Parameter Store to save the list of all security-approved AMI. Set up a managed rule on AWS Config to continuously scan all running EC2 instances. For any detected vulnerability, run the designated SSM Automation document* is incorrect because AWS Config cannot automatically run checks on the operating system of your Amazon EC2 Instances. The requirement is to run the assessment every 30-days only and not continuously.

The option that says: *Check AWS CloudTrail logs to determine the Amazon EC2 instance IDs that were launched from the AMIs that need scanning. Use AWS Config managed rule to run CVE assessment and remediation on the instances* is incorrect. Although it is possible to parse the EC2 instance IDs from CloudTrail and determine the vulnerable instances, you still cannot run the CVE assessment in AWS Config for your Amazon EC2 Instances. Using Amazon Inspector is the most suitable service to use in running the CVE assessment.

9 A startup currently runs a web application on an extra-large Amazon EC2 instance. The application allows users to upload and download various pdf files from a private Amazon S3 bucket using a pre-signed URL. The web application checks if the file being requested actually exists in the S3 bucket before generating the URL.

In this scenario, how should the solutions architect configure the web application to access the Amazon S3 bucket securely?

(view)

1 1 0 00:00:00

1. Store your access keys inside the EC2 instance. 2. Program your web application to retrieve the AWS credentials from the instance to interact with the objects in the S3 bucket.

2. Create an IAM user with the appropriate permissions allowing access and listing of all of the objects of the S3 bucket. Associate the EC2 instance with the IAM user. 2. Program your web application to retrieve the user credentials from the EC2 instance metadata.

3. Create an IAM role with a policy that allows listing of the objects in the S3 bucket. Launch the EC2 instance with the IAM role. 2. Program your web application to retrieve the temporary security credentials from the EC2 instance user data.

4. Create an IAM role with a policy that allows listing and uploading of the objects in the S3 bucket. Launch the EC2 instance with the IAM role. 2. Program your web application to retrieve the temporary security credentials from the EC2 instance metadata.

24 A company is managing a hybrid environment comprising on-premises servers and Amazon EC2 instances. Their servers are a mix of Linux and Windows systems. The company utilizes AWS Security Hub as its cloud security posture management (CSPM) service to automate security best practice checks, aggregate alerts, and enables automated remediations. Separate teams handle the environment, each using different tools for patching. The company wants to simplify this process and requires a consolidated view of patch statuses across all servers.

Which combination of actions will meet the requirements? (Select TWO.)

(view)

Reproduce all servers as managed nodes in AWS System Manager. Use AWS System Manager Patch Manager for handling patch operations.

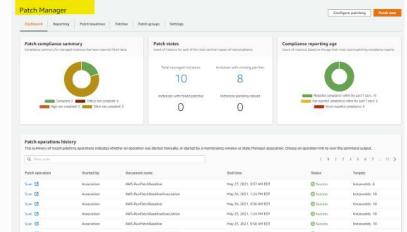
Create a custom AWS Config rule for detecting non-compliant patches across servers. Set up an AWS Config remediation rule using AWS Systems Manager Automation documents to apply patches. Configure AWS Config to send patch compliance reports to an Amazon S3 bucket.

Create an AWS Systems Manager document for running patching jobs. Use Amazon EventBridge to schedule these jobs and configure the AWS Systems Manager OpsCenter to generate patch compliance reports.

Provision and consolidate a hybrid AWS System Manager Patch Manager

Generate patch compliance reports using Amazon Inspector. Use Amazon Athena for aggregating and viewing patch compliance reports.

AWS Systems Manager allows you to change your hybrid environment as it can work with both on-premises servers and AWS instances. AWS Systems Manager Patch Manager, a feature of AWS Systems Manager, enables you to automate the process of patching managed instances with both security related and other types of updates.



You can use Patch Manager to apply patches for both operating systems and applications (On Windows Server, application support is limited to updates for applications released by Microsoft.) You can use Patch Manager to install Service Packs on Windows nodes and perform minor version upgrades on Linux nodes. You can patch fleets of Amazon Elastic Compute Cloud (Amazon EC2) instances, edge devices, on-premises servers, and virtual machines (VMs) by operating system type. You can scan instances to see only a report of missing patches, or you can scan and automatically install all missing patches.

Patch Manager provides a convenient view of patch status in the console and lets you export this data as a CSV file for analysis.

In the scenario, using Patch Manager allows you to both conduct patching operations and generate a consolidated view of patch statuses. This simplifies the overall patching process.

The option that says: Create a custom AWS Config rule for detecting non-compliant patches across servers. Set up an AWS Config remediation rule using AWS Systems Manager Automation documents to apply patches. Configure AWS Config to send patch compliance reports to an Amazon S3 bucket is incorrect. AWS Config is primarily designed for assessing, auditing, and evaluating the configurations of AWS resources, not specifically for managing patch operations. While it could provide some level of oversight for patch compliance, it would not simplify the patching process nor provide a straightforward, consolidated view of patch statuses.

The option that says: Create an AWS Systems Manager Automation document for running patching jobs. Use Amazon EventBridge to schedule these jobs and configure the AWS Systems Manager OpsCenter to generate patch compliance reports is incorrect. This option would simplify the patching process to some extent. However, setting up this integration would likely have more overhead compared to using AWS Systems Manager Patch Manager, which is designed specifically for managing patches and can provide a more streamlined and integrated solution.

The option that says: General patch compliance reports using Amazon Inspector. Use Amazon Athena for aggregating and viewing patch compliance reports is incorrect. Amazon Inspector is mainly a security vulnerability assessment service, not a tool for patch management. Moreover, Amazon Inspector is not capable of scanning on-premises networks.

25 An accounting company initiated a migration of its core workloads to the AWS Cloud to modernize its infrastructure, following a recent change in technical leadership. The company's current setup includes Amazon EC2 instances for compute resources, Amazon EFS for shared file storage, and Amazon RDS DB instances for managing relational databases.

To comply with regulatory mandates and uphold internal data governance standards, the new leadership has mandated the following backup procedures:

- The solution must provide a centralized overview of backup status across the AWS environment.
- Any backup failure must trigger immediate notifications to technical stakeholders.
- Data backups must follow custom retention policies, with separate rules for daily, weekly, and monthly backups.
- Backups must be automatically replicated to at least one additional AWS Region immediately after creation.

Which actions will meet these objectives while introducing the least operational overhead?

(view)

Develop and deploy a consolidated Python script on a single EC2 instance. Schedule the script to connect to each RDS DB instance, run `pg_dump / mysql`, Call the `create_image()` API to generate AMIs for selected instances. Mount the EFS file system to the EC2 instance and use rsync to copy data. Replicate the backup files to another Region using Amazon S3. Send emails using Amazon SES if anything fails.

Define individual snapshot lifecycle policies in Amazon Data Lifecycle Manager (Amazon DLM) to align with each retention requirement. Activate cross-Region snapshot replication within DLM to ensure backups are automatically copied to another AWS Region. Develop an AWS Lambda function that monitors for backup or restore events and triggers a notification when a job finishes with either `RESTORE_JOB_SUCCESSFUL` or `BACKUP_JOB_FAILED` status.

Define an AWS Backup plan that includes individual backup rules aligned with each required retention schedule. Develop an AWS Lambda function to handle cross-Region backup replication and to dispatch notifications whenever a backup job results in `BACKUP_JOB_FAILED` or `RESTORE_JOB_SUCCESSFUL`.

Establish an AWS Backup strategy incorporating individual backup rules corresponding to every retention specification. Set up a Backup strategy for cross-Region backup replication. Integrate an Amazon Simple Notification Service (Amazon SNS) topic into the backup strategy to approach alerts for completed operations, excluding those with the `BACKUP_JOB_COMPLETED` status.

AWS Backup is a fully managed service that makes it easy to **centralize and automate data protection across AWS services, in the cloud, and on-premises**. Using this service, you can **configure backup policies and monitor activity for your AWS resources** in one place. It allows you to automate and consolidate backup tasks previously performed service-by-service, and removes the need to create custom scripts and manual processes. You can automate your data protection policies and schedules with a few clicks in the AWS Backup console.

AWS Backup provides many features and capabilities, including **lifecycle management policies, cross-Region backup, and backup activity monitoring**.

Backup rule configuration Info**Schedule****Backup rule name**

my-daily-backup

Backup rule name is case sensitive. Must contain from 1 to 50 alphanumeric or '-' characters.**Backup vault** Info

Default

Create new vault ?**Backup vault****Backup frequency** Info

Daily

Backup window Info**Start time**Specify the time of day the backups will start. For hourly frequency, start time is the time the first backup is taken in a day. Where applicable, time will adjust to daylight savings time so that it retains the same local time all year.

00 : 30 Asia/Manila (UTC+08:00)

Start within InfoSpecify period of time in which the backup plan starts if it doesn't start at the specified time.

8 hours

Complete within Info

7 days

Point-in-time recovery Info Enable continuous backups for point-in-time recovery (PITR)With continuous backups, you can restore your AWS Backup-supported resource by rewinding it back to a specific time that you choose.

8 hours

Complete within Info

7 days

Point-in-time recovery Info Enable continuous backups for point-in-time recovery (PITR)With continuous backups, you can restore your AWS Backup-supported resource by rewinding it back to a specific time that you choose, within 1 second of precision (going back a maximum of 35 days). Available for Aurora, RDS, S3, and SAP HANA on Amazon EC2 resources.**Lifecycle** Info**Cold storage** Info Move backups from warm to cold storageAvailable for CloudFormation, DynamoDB with advanced features, EBS, SAP HANA, Timestream, and VMware virtual machines. Some resource types convert incremental backups to full backups. Requires at least 90 days of retention.**Cold storage for Amazon EBS** Info Archive Amazon EBS snapshots is available when cold storage is enabled and backup frequency is at least monthly.**Total retention period** InfoTell AWS Backup how long to store your backups.

35

Days

Total retention (days) Warm storage

Copy to destination - optional Info

Create a copy of backup in a separate backup vault or logically air-gapped vault.

Region

Asia Pacific (Singapore) ▼ Remove

Copy to another account's vault

Destination vault

The vault to which your backup copy will be made.

Default ▼ Create new vault

Backup vault

Lifecycle

Specify total retention period and cold storage settings for additional backup copies.

Use the same settings from backup rule
Cold storage: Not enabled; Total retention period: 5 weeks

Customize lifecycle

Add copy

AWS Backup is purpose-built to manage backup operations across EC2, EFS, and RDS with centralized visibility. It allows you to define individual backup rules for each retention period, supports native cross-Region replication, and integrates directly with Amazon SNS to send alerts for specific backup job statuses. The requirement also states the necessity to trigger immediate notifications on failure. Since AWS Backup includes multiple possible failure statuses and transitional statuses, the configuration simply sends notifications for all job completions except the successful BACKUP_JOB_COMPLETED status, ensuring nothing slips through the cracks. This approach meets all the stated requirements without introducing custom scripting or external monitoring, and it introduces the least operational overhead by using built-in managed features.

Hence, the correct answer is: **Establish an AWS Backup strategy incorporating individual backup rules corresponding to every retention specification. Set up a Backup strategy for cross-Region backup replication. Integrate an Amazon Simple Notification Service (Amazon SNS) topic into the backup strategy to dispatch alerts for completed operations, excluding those with the BACKUP_JOB_COMPLETED status.**

The option that says: **Develop and deploy a consolidated Python script on a single EC2 instance. Schedule the script to connect to each RDS DB instance, run pg_dump/mysql dump. Call the create_image() API to generate AMIs for selected instances. Mount the EFS file system to the EC2 instance and use rsync to copy data. Replicate the backup files to another Region using Amazon S3. Send emails using Amazon SES if anything fails** is incorrect. It relies on a custom EC2-based script to manage backups, which requires ongoing maintenance and complex logic to handle failures, retention, and replication. It just does not provide centralized visibility or native integration with AWS services for status tracking and alerting. This method adds operational complexity instead of reducing it and primarily duplicates features that AWS Backup already offers.

The option that says: **Define individual snapshot lifecycle policies in Amazon Data Lifecycle Manager (Amazon DLM) to align with each retention requirement. Activate cross-Region snapshot replication within DLM to ensure backups are automatically copied to another AWS Region. Develop an AWS Lambda function that monitors for backup or restore events and triggers a notification when a job finishes with**

either **RESTORE_JOB_SUCCESSFUL** or **BACKUP_JOB_FAILED** status is incorrect. Amazon DLM is primarily designed for EBS volume snapshots and does not support RDS or EFS. While it can automate retention and replication for EC2 volumes, it simply does not offer centralized backup visibility across all services or native notification mechanisms. Relying on Lambda for alerts adds overhead and does not replace the native features that AWS Backup provides.

The option that says: **Define an AWS Backup plan that includes individual backup rules aligned with each required retention schedule. Develop an AWS Lambda function to handle cross-Region backup replication and to dispatch notifications whenever a backup job results in **BACKUP_JOB_FAILED** or **RESTORE_JOB_SUCCESSFUL**** is incorrect. This solution offloads replication and notifications to custom Lambda functions, even though AWS Backup already supports built-in replication and alert integration through SNS. This approach just adds unnecessary components, making the architecture more complex and challenging to manage. It primarily ignores the full capabilities of AWS Backup and increases operational burden without any real benefit.

A company is developing a serverless application that is deployed on AWS Lambda. The application consists of several Lambda functions that resize, watermark, and process images. The metadata generated from the functions is written in an Amazon DynamoDB table. The company deployed an Amazon Neptune DB cluster in three private subnets inside a VPC. A new feature was developed that requires the Lambda functions to access the Neptune DB cluster. Which of the following options are possible solutions to allow the Lambda functions to access both the DynamoDB table and Neptune DB cluster? (Select TWO.)

(view)

1 0 1 00:01:50

Deploy the AWS Lambda functions into three new public subnets in the same VPC. Update the Amazon Neptune DB security group to allow connections from the Lambda security group. Route internet traffic to the Internet Gateway.

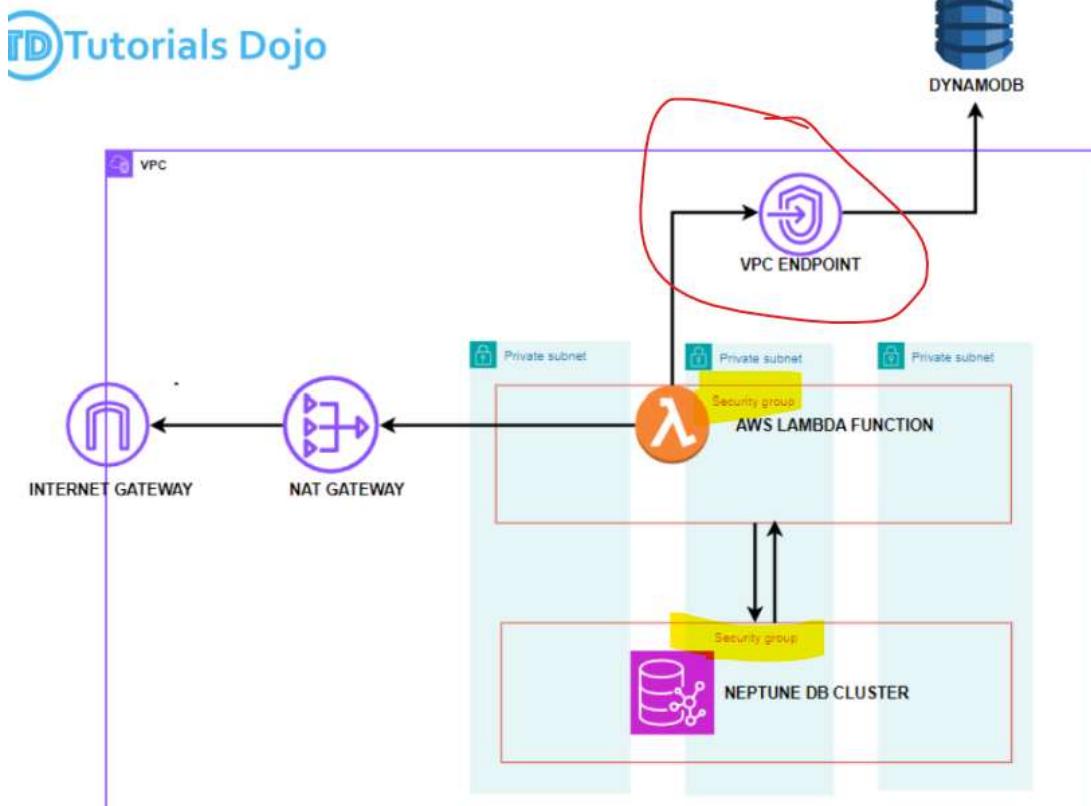
Deploy the AWS Lambda functions into three new private subnets in the same VPC. Update the Neptune DB security group to allow connections from the Lambda security group. Create a DynamoDB VPC endpoint and update the route table for routing DynamoDB requests.

Keep the AWS Lambda functions outside the VPC. Create a VPC endpoint for the Amazon Neptune service. Update the Lambda functions to use this endpoint to send requests to the Neptune DB cluster.

Deploy the AWS Lambda functions into three new private subnets in the same VPC. Update the Neptune DB security group to allow connections from the Lambda security group. Route the internet traffic of the Lambda functions through a NAT gateway.

Keep the Lambda functions outside the VPC. Configure the Neptune DB security group to allow inbound access from the AWS Lambda IP range.

dynamodb nao esta na mesma vpc



- Lambdas in **private subnets** (no public IP, no route to Internet Gateway) cannot reach public AWS services like DynamoDB **by default**, because **DynamoDB's public endpoint (<https://dynamodb.<region>.amazonaws.com>) lives on the internet.**

- Without an internet path (NAT Gateway), your Lambda's requests will fail.

AWS provides a **VPC Endpoint** — a private connection between your VPC and DynamoDB **over the AWS network**, not the internet.

2. Create a VPC Endpoint for DynamoDB

In the AWS Console:

Go to VPC → Endpoints → Create endpoint

Service category: AWS services

Service name: com.amazonaws.<region>.dynamodb

Type: Gateway

VPC: choose your Lambda's VPC

Route tables: **select the route table used by your private subnets**

3. Verify route table

Once created, AWS automatically adds a route:

Destination: com.amazonaws.<region>.dynamodb Target: vpce-xxxxxxxxxxxxx
This ensures any traffic to DynamoDB goes through the **VPC Endpoint**, not the internet.

```
import boto3
```

```
def lambda_handler(event, context):
```

```
    table = boto3.resource('dynamodb').Table('MyTable')
    table.put_item(Item={'id': '123', 'name': 'Erika'})
    return "Success"
```

4 An IT consultancy company has multiple offices located in San Francisco, Frankfurt, Tokyo, and Manila. The company is using AWS Organizations to easily manage its several AWS accounts which are being used by its regional offices and subsidiaries. A new AWS account was recently added to a specific organizational unit (OU) which is responsible for the overall systems administration. The solutions architect noticed that the account is using a root-created Amazon ECS Cluster with an attached service-linked role. For regulatory purposes, the solutions architect created a custom SCP that would deny the new account from performing certain actions in relation to using ECS. However, after applying the policy, the new account could still perform the actions that it was supposed to be restricted from doing.

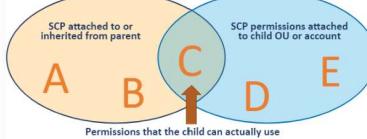
Which of the following is the most likely reason for this problem?

[view]

- The default SCP grants all permissions attached to every root, OU, and account. To apply stricter permissions, this policy is required to be modified.
- There is an SCP attached to a higher-level OU that permits the actions of the service-linked role. This permission would therefore be inherited by the current OU, and override the SCP placed by the administrator.
- The ECS service is being run outside the jurisdiction of the organization. SCPs affect only the principals that are managed by accounts that are part of the organization.

SCPs do not affect any service-linked role. Service-linked roles enable other AWS services to integrate with AWS Organizations and can't be restricted by SCPs.

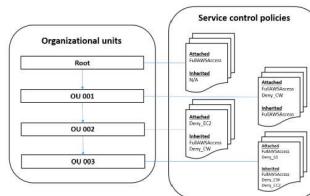
Users and roles must still be granted permissions using IAM permission policies attached to them or to groups. The SCPs filter the permissions granted by such policies, and the user can't perform any actions that the applicable SCPs don't allow. Actions allowed by the SCPs can be used if they are granted to the user or role by one or more IAM permission policies.



When you attach SCPs to the root, OUs, or directly to accounts, all policies that affect a given account are evaluated together using the same rules that govern IAM permission policies:

- Any action that has an explicit Deny in an SCP can't be delegated to users or roles in the affected accounts. An explicit Deny statement overrides any Allow that other SCPs might grant.
- Any action that has an explicit Allow in an SCP (such as the default *** SCP or by any other SCP that calls out a specific service or action) can be delegated to users and roles in the affected accounts.
- Any action that isn't explicitly allowed by an SCP is implicitly denied and can't be delegated to users or roles in the affected accounts.

By default, an SCP named FullAWSAccess is attached to every root, OU, and account. This default SCP allows all actions and all services. So in a new organization, until you start creating or manipulating the SCPs, all of your existing IAM permissions continue to operate as they did. As soon as you apply a new or modified SCP to a root or OU that contains an account, the permissions that your users have in that account become filtered by the SCP. Permissions that used to work might now be denied if they're not allowed by the SCP at every level of the hierarchy down to the specified account.



The option that says: **The default SCP grants all permissions attached to every root, OU, and account.** To apply stricter permissions, this policy is required to be modified is incorrect. The scenario already implied that the administrator created a Deny policy. By default, an SCP named FullAWSAccess is attached to every root, OU, and account. This default SCP allows all actions and all services. However, you specify a Deny policy if you want to create a blacklist that blocks all access to the specified services and actions. The explicit Deny on specific actions in the blacklist policy overrides the Allow in any other policy, such as the one in the default SCP.

The option that says: **There is an SCP attached to a higher-level OU that permits the actions of the service-linked role.** This permission would therefore be inherited by the current OU, and override the SCP placed by the administrator is incorrect because even if a higher-level OU has an SCP attached with an Allow policy for the service, the current set up should still have restricted access to the service. Creating and attaching a new Deny SCP to the new account's OU will not be affected by the pre-existing Allow policy in the same OU.

The option that says: **The ECS service is being run outside the jurisdiction of the organization.** SCPs affect only the principals that are managed by accounts that are part of the organization is incorrect because the service-linked role must have been created within the organization, most notably by the root account of the organization. It also does not make sense if we make the assumption that the service is indeed outside of the organization's jurisdiction because the Principal element of a policy specifies which entity will have limited permissions. But the scenario tells us that it should be the new account that is denied certain actions, not the service itself.

A service-linked role is a special type of IAM role that is directly managed and used by an AWS service — not by you or your applications.

You generally can't (and shouldn't) edit this directly.

You can't modify the trust or permission policies -> They're AWS-managed

Deleting the role can break the service -> Always check dependencies

Use aws iam list-roles or the console to see which services use them

Some services require a service-linked role before activation (e.g., ECS, Glue, EMR)

Property	Description
Created by AWS	Automatically created when needed
Managed by AWS	Permissions are updated automatically as the service evolves
Scoped to one service	Each service has its own role; not shared
Can be deleted (carefully)	If you delete it, that service might stop working until recreated
Visible in IAM console	Listed under <i>Roles</i> → <i>AWS Service-linked roles</i>

- 5 A media company hosts its entire infrastructure on the AWS cloud. There is a requirement to copy information to or from the shared resources from another AWS account. The solutions architect has to provide the other account access to several AWS resources, such as Amazon S3, AWS KMS, and Amazon OpenSearch Service, in the form of a list of AWS account ID numbers. In addition, the user in the other account should still work in the trusted account, and there is no need to give up the user permissions in place of the role permissions. The solutions architect must also set up a solution that continuously assesses, audits, and monitors the policy configurations.

Which of the following is the MOST suitable type of policy that should be used in this scenario?

[\(view\)](#)

- Set up cross-account access with a user-based policy configuration. Use AWS Config rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration.
- Set up a service-linked role with an identity-based policy. Use AWS Systems Manager rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration.
- Set up cross-account access with a resource-based Policy. Use AWS Config rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration.
- Set up a service-linked role with a service control policy. Use AWS Systems Manager rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration.

For some AWS services, you can grant cross-account access to your resources. To do this, you attach a policy directly to the resource that you want to share, instead of using a role as a proxy. The resource that you want to share must support resource-based policies. Unlike a user-based policy, a resource-based policy specifies who (in the form of a list of AWS account ID numbers) can access that resource.

Cross-account access with a resource-based policy has some advantages over a role. With a resource that is accessed through a resource-based policy, the user still works in the trusted account and does not have to give up his or her user permissions in place of the role permissions. In other words, the user continues to have access to resources in the trusted account at the same time as he or she has access to the resource in the trusting account. This is useful for tasks such as copying information to or from the shared resource in the other account.

The option that says: **Set up cross-account access with a user-based policy configuration. Use AWS Config rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration** is incorrect because a user-based policy only grants the access to a certain IAM user and not to a certain AWS resource.

The option that says: **Set up a service-linked role with an identity-based policy. Use AWS Systems Manager rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration** is incorrect because a service-linked role is just a unique type of IAM role that is linked directly to an AWS service. In addition, it is the AWS Config service, and not the AWS Systems Manager, that enables you to assess, audit, and evaluate the configurations of your AWS resources.

The option that says: **Set up a service-linked role with a service control policy. Use AWS Systems Manager rules to periodically audit changes to the IAM policy and monitor the compliance of the configuration** is incorrect because a service control policy is primarily used in AWS Organizations and not for cross-account access. Service-linked roles are predefined by the service and include all the permissions that the service requires to call other AWS services on your behalf. This is not suitable for providing access to your resources to other AWS accounts, unlike cross-account access. You should also use AWS Config, and not AWS Systems Manager, to periodically audit changes to the IAM policy.

- 6 A multinational manufacturing company has multiple AWS accounts in multiple AWS regions across North America, Europe, and Asia. The solutions architect has been tasked to set up AWS Organizations to centrally manage policies and have full administrative control across the multiple AWS accounts owned by the company.

Which of the following options is the recommended implementation to achieve this requirement with the LEAST effort?

[\(view\)](#)

1 0 1 00:00:00

- Set up AWS Organizations by establishing cross-account access from the master account to all member AWS accounts of the company. The master account will automatically have full administrative control across all member accounts.
- Set up AWS Organizations by sending an invitation to the master account of your organization from each of the member accounts of the company. Create an OrganizationAccountAccessRole IAM role in the member account and grant permission to the master account to assume the role.
- Use AWS Control Tower from the master account and enroll all the member AWS accounts of the company. AWS Control Tower will automatically provision the needed IAM permissions to have full administrative control across all member accounts.
- Set up AWS Organizations by creating a central account of member accounts of the company from the master account of your organization. Create an OrganizationAccountAccessRole IAM role in the member account and grant permission to the master account to assume the role.



After you create an **Organization** and verify that you own the email address associated with the master account, you can invite existing AWS accounts to join your organization. When you invite an account, AWS Organizations sends an invitation to the account owner, who decides whether to accept or decline the invitation. You can use the AWS Organizations console to initiate and manage invitations that you send to other accounts. You can send an invitation to another account only from the master account of your organization.

If you are the administrator of an AWS account, you also can accept or decline an invitation from an organization. If you accept, your account becomes a member of that organization. Your account can join only one organization, so if you receive multiple invitations to join, you can accept only one.

When an invited account joins your organization, you *do not* automatically have full administrator control over the account, unlike created accounts. If you want the master account to have **full administrative control over an invited member account, you must create**

the OrganizationAccountAccessRole IAM role in the member account and grant permission to the master account to assume the role.

