

Latest Version: 23.1

Question: 1

To run an application, a DevOps Engineer launches an Amazon EC2 instances with public IP addresses in a public subnet. A user data script obtains the application artifacts and installs them on the instances upon launch. A change to the security classification of the application now requires the instances to run with no access to the Internet. While the instances launch successfully and show as healthy, the application does not seem to be installed. Which of the following should successfully install the application while complying with the new rule?

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

Answer: C

EC2 instances running in private subnets of a VPC can now have controlled access to S3 buckets, objects, and API functions that are in the same region as the VPC. You can use an S3 bucket policy to indicate which VPCs and which VPC Endpoints have access to your S3 buckets 1-
<https://aws.amazon.com/pt/blogs/aws/new-vpc-endpoint-for-amazon-s3/>

Question: 2

An IT department manages a portfolio with Windows and Linux (Amazon and Red Hat Enterprise Linux) servers both on-premises and on AWS. An audit reveals that there is no process for updating OS and core application patches, and that the servers have inconsistent patch levels. Which of the following provides the MOST reliable and consistent mechanism for updating and maintaining all servers at the recent OS and core application patch levels?

- A. Install AWS Systems Manager agent on all on-premises and AWS servers. Create Systems Manager Resource Groups. Use Systems Manager Patch Manager with a preconfigured patch baseline to run scheduled patch updates during maintenance windows.
- B. Install the AWS OpsWorks agent on all on-premises and AWS servers. Create an OpsWorks stack with separate layers for each operating system, and get a recipe from the Chef supermarket to run the patch commands for each layer during maintenance windows.

- C. Use a shell script to install the latest OS patches on the Linux servers using yum and schedule it to run automatically using cron. Use Windows Update to automatically patch Windows servers.
- D. Use AWS Systems Manager Parameter Store to securely store credentials for each Linux and Windows server. Create Systems Manager Resource Groups. Use the Systems Manager Run Command to remotely deploy patch updates using the credentials in Systems Manager Parameter Store

Answer: A

1- <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html> 2- <https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html>

Question: 3

A company is setting up a centralized logging solution on AWS and has several requirements. The company wants its Amazon CloudWatch Logs and VPC Flow logs to come from different sub accounts and to be delivered to a single auditing account. However, the number of sub accounts keeps changing. The company also needs to index the logs in the auditing account to gather actionable insight. How should a DevOps Engineer implement the solution to meet all of the company's requirements?

- A. Use AWS Lambda to write logs to Amazon ES in the auditing account. Create an Amazon CloudWatch subscription filter and use Amazon Kinesis Data Streams in the sub accounts to stream the logs to the Lambda function deployed in the auditing account.
- B. Use Amazon Kinesis Streams to write logs to Amazon ES in the auditing account. Create a CloudWatch subscription filter and use Kinesis Data Streams in the sub accounts to stream the logs to the Kinesis stream in the auditing account.
- C. Use Amazon Kinesis Firehose with Kinesis Data Streams to write logs to Amazon ES in the auditing account. Create a CloudWatch subscription filter and stream logs from sub accounts to the Kinesis stream in the auditing account.
- D. Use AWS Lambda to write logs to Amazon ES in the auditing account. Create a CloudWatch subscription filter and use Lambda in the sub accounts to stream the logs to the Lambda function deployed in the auditing account.

Answer: C

<https://aws.amazon.com/pt/blogs/architecture/central-logging-in-multi-account-environments/>

Question: 4

A company wants to use a grid system for a proprietary enterprise in-memory data store on top of AWS. This system can run in multiple server nodes in any Linux-based distribution. The system must be able to reconfigure the entire cluster every time a node is added or removed. When adding or removing nodes, an `/etc./cluster/nodes.config` file must be updated, listing the IP addresses of the current node members of that cluster

The company wants to automate the task of adding new nodes to a cluster. What can a DevOps Engineer do to meet these requirements?

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

Answer: A

<https://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook-events.html>

Question: 5

A company has established tagging and configuration standards for its infrastructure resources running on AWS. A DevOps Engineer is developing a design that will provide a near-real-time dashboard of the compliance posture with the ability to highlight violations. Which approach meets the stated requirements?

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

Answer: B

<https://aws.amazon.com/about-aws/whats-new/2019/03/aws-config-now-supports-tagging-of-aws-config-resources/>