Amazon

DOP-C01 Exam

Amazon AWS Certified DevOps Engineer - Professional Exam

Thank you for Downloading DOP-C01 exam PDF Demo

You can Buy Latest DOP-C01 Full Version Download

https://www.certkillers.net/Exam/DOP-C01

Version: 18.0

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

Explanation:
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

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

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

Explanation:
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

Explanation:
https://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook-events

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

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

config-resources/

Question: 6

A production account has a requirement that any Amazon EC2 instance that has been logged into manually must be terminated within 24 hours. All applications in the production account are using Auto Scaling groups with Amazon CloudWatch Logs agent configured. How can this process be automated?

A. Create a CloudWatch Logs subscription to an AWS Step Functions application. Configure the function to add a tag to the EC2 instance that produced the login event and mark the instance to be decommissioned. Then create a CloudWatch Events rule to trigger a second AWS Lambda function once a day that will terminate all instances with this tag.
B. Create a CloudWatch alarm that will trigger on the login event. Send the notification to an Amazon SNS topic that the Operations team is subscribed to, and have them terminate the EC2 instance within 24 hours.
C. Create a CloudWatch alarm that will trigger on the login event. Configure the alarm to send to an Amazon SQS queue. Use a group of worker instances to process messages from the queue, which then schedules the Amazon CloudWatch Events rule to trigger.
D. Create a CloudWatch Logs subscription in an AWS Lambda function. Configure the function to add a tag to the EC2 instance that produced the login event and mark the instance to be decommissioned. Create a CloudWatch Events rule to trigger a daily Lambda function that terminates all instances with this tag.

Answer: D

Explanation:
https://boto3.amazonaws.com/v1/documentation/api/latest/guide/cw-example-subscription-filters

Question: 7

A DevOps Engineer is implementing a mechanism for canary testing an application on AWS. The application was recently modified and went through security, unit, and functional testing. The application needs to be deployed on an AutoScaling group and must use a Classic Load Balancer. Which design meets the requirement for canary testing?

A. Create a different Classic Load Balancer and Auto Scaling group for blue/green environments. Use Amazon Route 53 and create weighted A records on Classic Load Balancer.
B. Create a single Classic Load Balancer and an Auto Scaling group for blue/green environments. Use Amazon Route 53 and create A records for Classic Load Balancer IPs. Adjust traffic using A records.
C. Create a single Classic Load Balancer and an Auto Scaling group for blue/green environments. Create an Amazon CloudFront distribution with the Classic Load Balancer as the origin. Adjust traffic using CloudFront.
D. Create a different Classic Load Balancer and Auto Scaling group for blue/green environments. Create an Amazon API Gateway with a separate stage for the Classic Load Balancer. Adjust traffic by giving weights to this stage.

Answer: A

Question: 8

An online retail company based in the United States plans to expand its operations to Europe and Asia in the next six months. Its product currently runs on Amazon EC2 instances behind an Application Load Balancer. The instances run in an Amazon EC2 Auto Scaling group across multiple Availability Zones. All data is stored in an Amazon Aurora database instance.
When the product is deployed in multiple regions, the company wants a single product catalog across all regions, but for compliance purposes, its customer information and purchases must be kept in each region.
How should the company meet these requirements with the LEAST amount of application changes?

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

Answer: C

Question: 9

A company has several AWS accounts. The accounts are shared and used across multiple teams globally, primarily for Amazon EC2 instances. Each EC2 instance has tags for team, environment, and cost center to ensure accurate cost allocations.
How should a DevOps Engineer help the teams audit their costs and automate infrastructure cost optimization across multiple shared environments and accounts?

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

second trigger to initiate a Lambda function to reduce underutilized instances.
D. Use AWS Systems Manager to track instance utilization and report underutilized instances to Amazon CloudWatch. Filter data in CloudWatch based on tags for team, environment, and cost center. Set up triggers from CloudWatch into AWS Lambda to reduce underutilized instances

Answer: C

Explanation:
https://github.com/aws/Trusted-Advisor-Tools/tree/master/LowUtilizationEC2Instances
https://docs.aws.amazon.com/quicksight/latest/user/supported-data-sources

Question: 10

A company has a hybrid architecture solution in which some legacy systems remain on-premises, while a specific cluster of servers is moved to AWS. The company cannot reconfigure the legacy systems, so the cluster nodes must have a fixed hostname and local IP address for each server that is part of the cluster.
The DevOps Engineer must automate the configuration for a six-node cluster with high availability across three Availability Zones (AZs), placing two elastic network interfaces in a specific subnet for each AZ. Each node's hostname and local IP address should remain the same between reboots or instance failures.
Which solution involves the LEAST amount of effort to automate this task?

A. Create an AWS Elastic Beanstalk application and a specific environment for each server of the cluster. For each environment, give the hostname, elastic network interface, and AZ as input parameters. Use the local health agent to name the instance and attach a specific elastic network interface based on the current environment.
B. Create a reusable AWS CloudFormation template to manage an Amazon EC2 Auto Scaling group with a minimum size of 1 and a maximum size of 1. Give the hostname, elastic network interface, and AZ as stack parameters. Use those parameters to set up an EC2 instance with EC2 Auto Scaling and a user data script to attach to the specific elastic network interface. Use CloudFormation nested stacks to nest the template six times for a total of six nodes needed for the cluster, and deploy using the master template.
C. Create an Amazon DynamoDB table with the list of hostnames subnets, and elastic network interfaces to be used. Create a single AWS CloudFormation template to manage an Auto Scaling group with a minimum size of 6 and a maximum size of 6. Create a programmatic solution that is installed in each instance that will lock/release the assignment of each hostname and local IP address, depending on the subnet in which a new instance will be launched.
D. Create a reusable AWS CLI script to launch each instance individually, which will name the instance, place it in a specific AZ, and attach a specific elastic network interface. Monitor the instances and in the event of failure, replace the missing instance manually by running the script again.

Answer: B

Thank You for trying DOP-C01 PDF Demo

To Buy Latest DOP-C01 Full Version Download visit link below

https://www.certkillers.net/Exam/DOP-C01

Start Your DOP-C01 Preparation

[Limited Time Offer] Use Coupon "CKNET" for Further discount on your purchase. Test your DOP-C01 preparation with actual exam questions.