DOP-C01 Q&As

AWS Certified DevOps Engineer - Professional (DOP-C01)

Pass Amazon DOP-C01 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/aws-devops-engineer-professional.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Amazon Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

You need to implement A/B deployments for several multi-tier web applications. Each of them has its Individual infrastructure: Amazon Elastic Compute Cloud (EC2) front-end servers, Amazon ElastiCache clusters, Amazon Simple Queue Service (SQS) queues, and Amazon Relational Database (RDS) Instances. Which combination of services would give you the ability to control traffic between different deployed versions of your application?

A. Create one AWS Elastic Beanstalk application and all AWS resources (using configuration files inside the application source bundle) for each web application. New versions would be deployed a-eating Elastic Beanstalk environments and using the Swap URLs feature.

B. Using AWS CloudFormation templates, create one Elastic Beanstalk application and all AWS resources (in the same template) for each web application. New versions would be deployed using AWS CloudFormation templates to create new Elastic Beanstalk environments, and traffic would be balanced between them using weighted Round Robin (WRR) records in Amazon Route53.

C. Using AWS CloudFormation templates, create one Elastic Beanstalk application and all AWS resources (in the same template) for each web application. New versions would be deployed updating a parameter on the CloudFormation template and passing it to the cfn-hup helper daemon, and traffic would be balanced between them using Weighted Round Robin (WRR) records in Amazon Route 53.

D. Create one Elastic Beanstalk application and all AWS resources (using configuration files inside the application source bundle) for each web application. New versions would be deployed updating the Elastic Beanstalk application version for the current Elastic Beanstalk environment.

Correct Answer: B

QUESTION 2

You are building an AWS CloudFormation template for a multi-tier web application. The user data of your Linux web server resource contains a complex script that can take a long time to run. Which techniques could you use to ensure that these servers are fully configured and running before attaching them to the load balancer? (Choose two.)

A. Launch your Linux servers from a nested stack that is called from within the load balancer resource in your AWS CloudFormation template.

B. Add an AWS CloudFormation Wait Condition that depends on the web server resource. When the UserData script finishes on the web servers, use curl to send a signal the Wait Condition at http://169.254.169.254/waithandle/.

C. Add an AWS CloudFormation wait Condition that depends on the web server resource. When the UserData script finishes on the web servers, use curl to signal to the Wait Condition pre-signed URL that they are ready.

D. In your AWS CloudFormation template, position the load balancer resource JSON block directly below your Linux server resource.

E. Add an AWS CloudFormation Wait Condition that depends on the web server resource. When the UserData script finishes on the web servers, use the command "cfn-signal" to signal that they are ready.

Correct Answer: CE

QUESTION 3

A Developer is designing a continuous deployment workflow for a new Development team to facilitate the process for source code promotion in AWS. Developers would like to store and promote code for deployment from development to

production while maintaining the ability to roll back that deployment if it fails.

Which design will incur the LEAST amount of downtime?

A. Create one repository in AWS CodeCommit. Create a development branch to hold merged changes. Use AWS CodeBuild to build and test the code stored in the development branch triggered on a new commit. Merge to the master and deploy to production by using AWS CodeDeploy for a blue/green deployment.

B. Create one repository for each Developer in AWS CodeCommit and another repository to hold the production code. Use AWS CodeBuild to merge development and production repositories, and deploy to production by using AWS CodeDeploy for a blue/green deployment.

C. Create one repository for development code in AWS CodeCommit and another repository to hold the production code. Use AWS CodeBuild to merge development and production repositories, and deploy to production by using AWS CodeDeploy for a blue/green deployment.

D. Create a shared Amazon S3 bucket for the Development team to store their code. Set up an Amazon CloudWatch Events rule to trigger an AWS Lambda function that deploys the code to production by using AWS CodeDeploy for a blue/ green deployment.

Correct Answer: A

QUESTION 4

Which difference between core modules and extra modules is not correct?

A. Extra modules may one day become core modules

B. Core modules are supported by the Ansible team

C. Core modules are shipped by default with Ansible

D. Extra modules have no support

Correct Answer: D

While extra modules are not official modules and thus not supported by the Ansible team, they are indeed supported by their writers and the community.

Reference: http://docs.ansible.com/ansible/modules_extra.html

QUESTION 5

A company has many applications. Different teams in the company developed the applications by using multiple languages and frameworks. The applications run on premises and on different servers with different operating systems. Each team has its own release protocol and process. The company wants to reduce the complexity of the release and maintenance of these applications.

The company is migrating its technology stacks, including these applications, to AWS. The company wants centralized control of source code, a consistent and automatic delivery pipeline, and as few maintenance tasks as possible on the

underlying infrastructure.

What should a DevOps engineer do to meet these requirements?

A. Create one AWS CodeCommit repository for all applications. Put each application\\'s code in different branch. Merge the branches, and use AWS CodeBuild to build the applications. Use AWS CodeDeploy to deploy the applications to one centralized application server.

B. Create one AWS CodeCommit repository for each of the applications Use AWS CodeBuild to build the applications one at a time. Use AWS CodeDeploy to deploy the applications to one centralized application server.

C. Create one AWS CodeCommit repository for each of the applications. Use AWS CodeBuild to build the applications one at a time to create one AMI for each server. Use AWS CloudFormation StackSets to automatically provision and decommission Amazon EC2 fleets by using these AMIs.

D. Create one AWS CodeCommit repository for each of the applications. Use AWS CodeBuild to build one Docker image for each application in Amazon Elastic Container Registry (Amazon ECR). Use AWS CodeDeploy to deploy the applications to Amazon Elastic Container Service (Amazon ECS) on infrastructure that AWS Fargate manages.

Correct Answer: B

Reference: https://towardsdatascience.com/ci-cd-logical-and-practical-approach-to-build-four-step-pipeline-on-aws-3f54183068ec