Question .1:

Stackset are created in the AWS account which is a administrator account?

A. FALSE

B. TRUE - **(correct)**

Explanation: B.TRUE

An **administrator account** is the AWS account in which you create stack sets. For stack sets with service-managed permissions, the administrator account is either the organization's management account or a delegated administrator account.

Question .2:

What sections of the AWS CloudFormation template are considered as "required"?

A. Transforms

B. Outputs

C. Conditions

D. Resource - **(correct)**

E. Metadata

Explanation: D.Resource

Templates include several major sections. The **Resources** section is the only required section. Some sections in a template can be in any order. However, as you build your template, it can be helpful to use the logical order shown in the following list because values in one section might refer to values from a previous section.

Question .3:

Description Section in the AWS Cloudformation template can exists independent of Format Version Section.

A. FALSE -**(correct)**

B. TRUE

Explanation: A. FALSE

The AWS CloudFormation **template version that the template conforms to**. The template format version isn't the same as the API or WSDL version. The template format version can change independently of the API and WSDL versions.

Question .4:

A stackset created in one regions , can be viewed or modified in multiple regions?

A. FALSE -**(correct)**

B. TRUE

Explanation: A. FALSE

A stack set is a regional resource. If you create a stack set in one AWS Region, you can only see or change it when viewing that Region.

A stack set lets you create stacks in AWS accounts across regions by using a single CloudFormation template. A stack set's CloudFormation template defines all the resources in each stack.After you've defined a stack set, you can create, update, or delete stacks in the target accounts **and AWS Regions you specify.**

Question .5:

You are trying to delete the stack instance, but the operation failed. What is

the reason?

A. you cannot delete individual instances

B. Retain stack option is selected

C. Termination protection is enabled - **(correct)**

D. Delete stack option is disabled

Explanation: C. Termination protection is enabled

Stack deletion will fail for any stacks on which **termination protection has been enabled.**

Determine if termination protection has been enabled for the stack. If it has, disable termination protection and then perform the stack instance deletion again.

Question .6:

During resource import operation,AWS CloudFormation performs several validations which are:

A. All of these -**(correct)**

B. Properties and configuration values for each resource to import adhere to the resource type schema

C. Resource to import does not belong to another stack in the same Region

D. Resource to import exists

E. Required properties are specified in the template

Explanation: A. All of these

The resource to import exists.

The properties and configuration values for each resource to import adhere to the resource type schema, which defines its accepted properties, required properties, and supported property values.The required properties are specified in the template. Required properties for each resource type are listed in the Resource and property reference.The resource to import doesn't belong to another stack in the same Region.

Question .7:

AWS::CloudFormation::Interface is a metadata key which can be used with

A. AWS CloudFormation CLI

B. All of these

C. AWS CloudFormation console -**(correct)**

D. AWS CloudFormation API Calls

Explanation: C. AWS CloudFormation console

AWS::CloudFormation::Interface is a metadata key that defines how parameters are grouped and sorted in the **AWS CloudFormation console**. When you create or update stacks in the console, the console lists input parameters in alphabetical order by their logical IDs.

Note: OnlytheCloudFormationconsoleuses the AWS::CloudFormation::Interface metadata key. AWS CLI and API calls don't use this key.

Question .8:

A stack instance can exist without a stack

A. TRUE - **(correct)**

B. FALSE

Explanation: A.TRUE

A stack instance is a reference to a stack in a target account within a Region. **A stack instance can exist without a stack**. For example, if the stack couldn't be created for some reason, the stack instance shows the reason for stack creation failure. A stack instance associates with only one stack set.

Question .9:

If you want the cfn-init to process configuration sections other than the default you must provide

A. ConfigSets -**(correct)**

B. config sections

C. you cannot change the default order

D. config keys

E. config files

Explanation: A **.**ConfigSets

The cfn-init helper script processes these **configuration sections** in the following order: packages, groups, users, sources, files, commands, and then services. If you require a different order, separate your sections into different config keys, and then use a configset that specifies the order in which the config keys should be processed.

cfn-init supports all metadata types for Linux systems. It supports metadata types for Windows with conditions that are described in the sections that follow.

For an example of using AWS::CloudFormation::Init and the cfn-init helper script, see Deploying applications on Amazon EC2 with AWS

CloudFormation.

Question .10:

Each resource to import into the stack must have which policy attribute for the import operation to succeed?

A. Deletion Policy - **(correct)**

B. Update Policy

C. Creation Policy

D. Retain Policy

Explanation: A **.**Deletion Policy

A template that describes the entire stack, including both the original stack resources and the resources you're importing. Each resource to import must have a **DeletionPolicy attribute**.Identifiers for the resources to import. You provide two values to identify each target resource.

Each resource to import must have a DeletionPolicy attribute for the import operation to succeed. The DeletionPolicy can be set to any possible value. Only target resources need a DeletionPolicy. Resources that are already part of the stack don't need a DeletionPolicy.

Question .11:

What are the two types of transforms that are supported by AWS CloudFormation? (Select 2)

A. AWS::Server

B. AWS::Enable

C. AWS::Include - **(correct)**

D. AWS::Instance

E. AWS::Serverless- **(correct)**

Explanation: C.AWS::Include

E.AWS::Serverless

AWS::Include transform enables you to insert boilerplate template snippets into your templates.

AWS::Serverless transform takes an entire template written in the AWS Serverless Application Model (AWS SAM) syntax and transforms and expands it into a compliant AWS CloudFormation template.

To get an idea of the breadth of possibilities, consider the AWS::Include and AWS::Serverless transforms, which are macros hosted by AWS CloudFormation.

Question .12:

During stack update,metadata section can be updated independent of the changes such as addition, modification or deletion of resources?

A. FALSE - **(correct)**

B. TRUE

Explanation: A. FALSE

During a stack update, you cannot update the Metadata section by itself. You can update it only when you include changes that add, modify, or delete resources.

Question .13:

How can Stacksets be deployed automatically to all accounts that are added to target organization or organizational units (OU)?

A. Enable automatic depployments for the stackset - **(correct)**

B. Automatic deployment feature is not available for the stackset

C. Enable automatic deployments for all the stack instances

Explanation: A. Enable automatic depployments for the stackset

With **automatic deployment enabled**, StackSets automatically deploys to accounts that are added to the target organization or organizational units (OUs) in the future. With retain stacks enabled, when an account is removed from a target OU, stack resources in the account are retained. You can adjust the automatic deployment settings you specified when you created your stack set at any time.

Note. Overridden parameter values only apply to the accounts that are currently in the target OUs and their child OUs. Accounts added to the target OUs and their child OUs in the future will use the stack set default values and not the overridden values.

Question .14:

You can import same resource in multiple stacks

A. FALSE - **(correct)**

B. TRUE

Explanation: A. FALSE

You can't import the same resource into multiple stacks.

You can use the cloudformation:ImportResourceTypes IAM policy condition to control which resource types IAM users can work with during an import operation.

The AWS CloudFormation stack limits apply when importing resources. For more information on limits, see AWS CloudFormation limits.

Question .15:

During a stack update, You can update conditions only when you include changes that add, modify, or delete resources.

A. TRUE - **(correct)**

B. FALSE

Explanation: A.TRUE

You can't update conditions by themselves. **You can update conditions only** when you include changes that add, modify, or delete resources. For example, you can add or modify a metadata attribute of a resource.

Add, modify, or delete parameter declarations. However, you can't add, modify, or delete a parameter that's used by a resource that doesn't support updates.

Question .16:

What are IAM roles required for the administrator and target accounts for stackset creation? (select 2)

A. AWSCloudFormationAdministrationRole

B. AWSCloudFormationExecutionRoleStackSet

C. AWSCloudFormationStackSetAdministrationRole -**(correct)**

D. AWSCloudFormationStackSetExecutionRole - **(correct)**

E. AWSCloudFormationExecutionRole -

Explanation: D.AWSCloudFormationStackSetExecutionRole

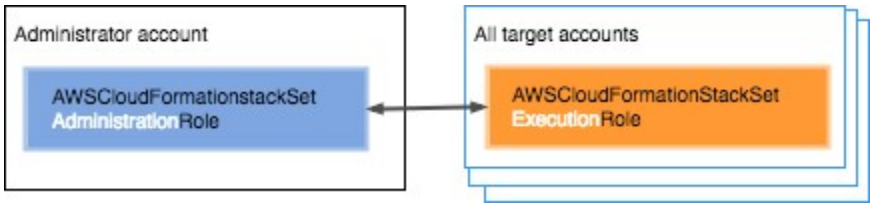
E. AWSCloudFormationExecutionRole

Create the necessary IAM service roles in your administrator and target accounts to define the permissions you want.

The role in your administrator account should be named

AWSCloudFormationStackSetAdministrationRole. The role in each ofyour target accounts should be named **AWSCloudFormationStackSetExecutionRole**.

If you structure your permissions this way, users don't pass an administrator role when creating or updating stack sets.



Question .17:

Pseudo parameters, parameters or intrinsic functions cannot be included in which section?

A. Output section

B. Condition section

C. Resource section

D. Mappings section - **(correct)**

Explanation: D. Mappings section

You can't include parameters, pseudo parameters, or intrinsic functions in the **Mappings section**.

The **Mappings section** consists of the key name Mappings. The keys in mappings must be literal strings. The values can be String or List types. The following example shows a Mappings section containing a single mapping named Mapping01 (the logical name).

Pseudo parameters are parameters that are predefined by AWS CloudFormation. You don't declare them in your template. Use them the same way as you would a parameter, as the argument for the Ref function.

Question .18:

You can create cross-stack references to export resources across regions

A. FALSE - **(correct)**

B. TRUE

Explanation: A.FALSE

You can't create cross-stack references across regions. You can use the intrinsic function Fn::ImportValue to import only values that have been exported within the same region.

For outputs, the value of the Name property of an Export can't use Ref or GetAtt functions that depend on a resource.Similarly, the ImportValue function can't include Ref or GetAtt functions that depend on a resource.You can't delete a stack if another stack references one of its outputs.You can't modify or remove an output value that is referenced by another stack.

Question .19:

What can be used while using dynamic parameters in stack template to reference sensitive information? (select 2)

A. AWS Secret Services

B. AWS Secrets Manager -**(correct)**

C. AWS Systems Manager Parameter Store - **(correct)**

D. AWS Secure Manager

Explanation: B.AWS Secrets Manager

C.AWS Systems Manager Parameter Store

Dynamic references provide a compact, powerful way for you to specify external values that are stored and managed in other services, such as the Systems Manager Parameter Store and AWS Secrets Manager, in your stack templates. When you use a dynamic reference, CloudFormation retrieves the value of the specified reference when necessary during stack and change set operations.

CloudFormation currently supports the following dynamic reference patterns:

ssm, for plaintext values stored in **AWS Systems Manager Parameter Store**.ssm-secure, for secure strings stored in AWS Systems Manager Parameter Store.

secretsmanager, for entire secrets or secret values stored in AWS Secrets Manager.

Question .20:

How can you verify that a target account meets certain requirements before AWS CloudFormation StackSets begins stack operations in that account?

A. you cannot check this, we will only know when stackset operation fails

B. target check

C. you have manually check each account before proceeding stackset operation

D. account gate check - **(Correct)**

Explanation: D. account gate check

An account gate is an optional feature that lets you specify an AWS Lambda function to verify that a target account meets certain requirements before AWS CloudFormation StackSets begins stack operations in that account.

A common example of an account gate is verifying that there are no CloudWatch alarms active or unresolved on the target account. StackSets invokes the function each time you start stack operations in the target account, and only continues if the function returns a SUCCEEDED code.

Question .21:

How do you create stacks in multiple regions across multiple accounts?

A. Stack group

B. Stack Sets - **(Correct)**

C. Nested stacks

D. Cannot be done

Explanation: B. Stack Sets

A stack set enables you to create stacks in AWS accounts across Regions by using a single AWS CloudFormation template.

A stack instance refers to a stack in a target account in a Region. It associates with only one stack set.

A stack set lets you create stacks in Amazon Web Services accounts across regions by using a single CloudFormation template. A stack set's CloudFormation template defines all the resources in each stack.

Question .22:

With Retain stacks selected, if the stack instances are removed from your stack set, what happens to the stacks and its resources in the target accounts?

A. stacks and resources are retain and are in their current state- **(correct)**

B. stacks are deleted but the resources are retained

C. stacks and resources are retained but are inoperable

D. stacks are retained but the resources are deleted

Explanation: A. stacks and resources are retain and are in their current state.

With Retain stacks selected, stack instances are removed from your stack set, but the stacks and their associated resources are retained. **The resources stay in their current state**, but are no longer part of the stack set. The stacks cannot be reassociated with an existing or new stack set.

When you save stacks from a stack set by choosing the Retain stacks option, **the stack's resources stay in their current state**, but the stack is no

longer part of the stack set. To reassociate a stack or add an existing stack to a stack set, see Importing a stack into AWS CloudFormation StackSets.

Question .23:

What are the parties involved in custom resource creation? (select 3)

A. AWS CloudFormation - **(correct)**

B. VPC Endpoint

C. template developer -**(correct)**

D. Custom resource provider - **(correct)**

E. AWS CloudFormation Resource section

Explanation:

Custom resources enable you to write custom provisioning logic in templates that AWS CloudFormation runs anytime you create, update (if you changed the custom resource), or delete stacks. For example, you might want to include resources that aren't available as AWS CloudFormation resource types.

AWS CloudFormation: During a stack operation, sends a request to a service token that is specified in the template, and then waits for a response before proceeding with the stack operation.

template developer: The template developer defines a custom resource in their template, which includes a service token and any input data parameters. Depending on the custom resource, the input data might be required; however, the service token is always required.

Custom resource provider: Owns the custom resource and determines how to handle and respond to requests from AWS CloudFormation. The custom resource provider must provide a service token that the template developer uses.

Question .24:

To include authentication information for a file or source that you specify with AWS::CloudFormation::Init , you can use which of the following

A. AWS::CloudFormation::Allowed

B. AWS::CloudFormation::Authentication - **(correct)**

C. AWS::CloudFormation::Designer

D. AWS::CloudFormation::Interface

Explanation: B. WS::CloudFormation::Authentication

Use the **AWS::CloudFormation::Authentication** resource to specify authentication credentials for files or sources that you specify with the AWS::CloudFormation::Init resource.

To include authentication information for a file or source that you specify with AWS::CloudFormation::Init, use the uris property if the source is a URI or the buckets property if the source is an Amazon S3 bucket. For more information about files, see Files. For more information about sources, see Sources.

Question .25:

What is the default order that the cfn-init helper script processes the configuration sections?

A. services, commands, sources, files, packages, groups, users

B. services, command, packages, files, sources, groups, users

C. users, groups, sources, files, packages, commands, services

D. packages, groups, users, sources, files, commands, services-**(correct)**

E. packages, users, groups, files, sources, services, commands

Explanation: D. packages, groups, users, sources, files, commands, services.

The cfn-init helper script processes these configuration sections in the following order: **packages, groups, users, sources, files, commands, and then services**. If you require a different order, separate your sections into different config keys, and then use a configset that specifies the order in which the config keys should be processed.