Number: AD0-E722
Passing Score: 800
Time Limit: 120
File Version: 1.7

Exam Code: 70-417

Exam Name: Adobe Commerce Architect Master

Exam A

QUESTION 1
A company wants to build an Adobe Commerce website to sell their products to customers in their country. The taxes in their country are highly complex and require customization to Adobe Commerce. An Architect is trying to solve this problem by creating a custom tax calculator that will handle the calculation of taxes for all orders in Adobe Commerce.
Following best practices, how should the Architect add the taxes for all orders?

A. Add a new observer to the event sales.quote.collecLtotals.before'' and add the custom tax to the quote
B. Write a before plugin to \Magento\Quote\Model\QuoteManagement::placeOrder() and add the custom tax to the quote
C. Declare a new total collector in 'etc/sales.xmr in a custom module

Correct Answer: C
Section:
Explanation:
According to the Adobe Commerce documentation, the best way to add a custom tax calculation to all orders is to declare a new total collector in the ''etc/sales.xml'' file of a custom module. This way, the custom tax logic can be implemented in a separate class that extends the \Magento\Quote\Model\Quote\Address\Total\AbstractTotal class and overrides the collect() and fetch() methods. The collect() method is responsible for calculating the tax amount and adding it to the quote address, while the fetch() method is responsible for displaying the tax amount in the cart and checkout pages. The new total collector can be assigned to any area of the order totals, such as before or after the subtotal, shipping, or grand total.
Customizing order totals
How to add custom fee or discount to order totals in Magento 2

QUESTION 2
An Adobe Commerce Architect is creating a new GraphQL API mutation to alter the process of adding configurable products to the cart. The mutation accepts configurable product ID. If the given product has only one variant, then the mutation should add this variant to the cart and return not nullable Cart type. If the configurable product has more variants, then the mutation should return not nullable Conf igurableProduct type.
The mutation declaration looks as follows:

```
type Mutation {
    addConfigurableToCart(product_id: Int!): AddToCartOutput!
        @resolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddConfigurableToCart")
}
```

How should the Adobe Commerce Architect declare output of this mutation?
A)

```
union AddToCartOutput
@typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver")
= ConfigurableProduct | Cart
```

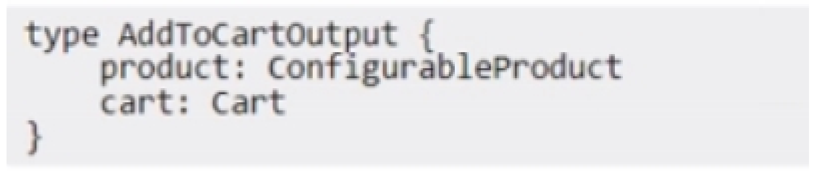
B)

```
interface AddToCartOutput
@typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver") {
}

type ConfigurableProduct implements AddToCartOutput {
}

type Cart implements AddToCartOutput {
}
```

C)



```
type AddToCartOutput {
    product: ConfigurableProduct
    cart: Cart
}
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
According to the Adobe Commerce documentation, the output of a GraphQL mutation is declared by specifying the type of the data returned by the mutation. The type can be either a scalar type (such as String, Int, Boolean, etc.), an object type (such as Cart, Product, Customer, etc.), or a union type (such as SearchResult, which can be either Product or Category). A union type is used when the mutation can return more than one possible type of data, depending on the input or the logic of the mutation. In this case, the mutation can return either a Cart type or a ConfigurableProduct type, depending on the number of variants of the configurable product. Therefore, the output of the mutation should be declared as a union type that includes both Cart and ConfigurableProduct types. Option B is the only option that correctly declares a union type using the pipe symbol (|) to separate the possible types. Option A and Option C are incorrect because they use brackets ([ ]) and curly braces ({ }), which are used for declaring list types and input object types, respectively.
GraphQL API - Adobe Inc.
Schema language with GraphQL | Adobe Commerce

QUESTION 3
A third-party company needs to create an application that will integrate the Adobe Commerce system to get orders data for reporting. The integration needs access to the GET /Vl/orders endpoint. It will call this endpoint automatically every hour around the clock. The merchant wants the ability to restrict or extend access to resources as well as to revoke the access using Admin Panel.
Which type of authentication available in Adobe Commerce should be used and implemented in a third-party system for this integration?

A. Use token-based authentication to obtain the Admin Token. The third-party system will utilize the REST endpoint using the admin username and password to get the Admin Token, which will be used as the Bearer Token to authorize.
B. Use token-based authentication to obtain an integration Token, integration will be created and activated in the admin panel using default integration token settings to get access to the token, which will be used as the Bearer Token to authorize.
C. Use OAuth-based authentication to provide access to system resources. Integration will be registered by the merchant in the admin panel with an OAuth handshake during activation. The third-party system should follow OAuth protocol to authorize.

Correct Answer: C
Section:
Explanation:
According to the Adobe Commerce documentation, OAuth-based authentication is the recommended method for integrations that need access to system resources, such as orders, customers, products, etc. OAuth-based authentication allows the merchant to control the access level and scope of the integration, as well as to revoke the access at any time using the admin panel. OAuth-based authentication also requires an OAuth handshake between the integration and the Adobe Commerce system during activation, which ensures a secure exchange of tokens and keys. The third-party system should follow the OAuth protocol to obtain and refresh the access token, which will be used as the Bearer Token to authorize the REST API calls.
Authentication | Adobe Commerce Developer Guide
OAuth-based authentication | Adobe Commerce Developer Guide

QUESTION 4
In a custom module, an Architect wants to define a new xml configuration file. The module should be able to read all the xml configuration files declared in the system, merge them together, and use their values in PHP class. Which two steps should the Architect make to meet this requirement? (Choose two.)

A. Inject a 'reader' dependency for 'Magento\Framework\Config\Data' in di.xml
B. Write a plugin for \Magento\Framework\Config\Data::get() and read the custom xml files

C. Create a Data class that implements '\Magento\Framework\Config\Data'
D. Append the custom xml file name in 'Magento\Config\Model\Config\Structure\Reader' in di.xml
E. Make a Reader class that implements '\Magento\Framework\Config\Reader\Filesystem'

Correct Answer: C, E
Section:
Explanation:
According to the Adobe Commerce documentation, to create a new xml configuration file, the Architect needs to create a Data class and a Reader class for the custom module. The Data class is responsible for storing and retrieving the configuration data from the cache or the Reader class. The Data class should implement the ''\Magento\Framework\Config\Data'' interface or extend the ''\Magento\Framework\Config\Data'' class. The Reader class is responsible for reading and validating the xml configuration files from the file system. The Reader class should implement the '\Magento\Framework\Config\Reader\Filesystem' interface or extend the '\Magento\Framework\Config\Reader\Filesystem' class. The Architect also needs to declare the Data class and the Reader class in the di.xml file of the custom module, and specify the name of the xml configuration file, the converter class, and the schema locator class for the Reader class.
Configuration types | Adobe Commerce - Experience League
Create configuration types | Adobe Commerce - Experience League

QUESTION 5
An Adobe Commerce Architect creates a stopword for the Italian locale named stopwordsjtJT.csv and changes the stopword directory to the following: <magento_root>/app/code/Custovendor/Elasticsearch/etc/stopwords/
What is the correct approach to change the stopwords directory inside the custom module?

A. Add stopwords to the stopwordsDirectory and CustomerVendor_Elasticsearch to the stopword sModule parameter Of the \Magento\Elasticsearch\SearchAdapter\Query\Preprocessor\Stopwords ClflSS Via di.xml
B. Add a new ClaSS implementing \Magento\Framework\Setup\Patch\PatchInterface to modify the default Value Of elasticsearch\customer\stopwordspath in core.conf ig_data table.
C. Add stopwords to the stopwordsDirectory parameter of the\Hagento\Elasticsearch\Model\Adapter\Document\DirectoryBuilder ClaSS Via stopwords/it.xml and Adobe Commerce will automatically detect the current module.

Correct Answer: A
Section:
Explanation:
According to the Adobe Commerce documentation, the correct approach to change the stopwords directory inside a custom module is to use dependency injection to override the default values of the stopwordsDirectory and stopwordsModule parameters of the \Magento\Elasticsearch\SearchAdapter\Query\Preprocessor\Stopwords class. The stopwordsDirectory parameter specifies the relative path of the stopwords directory from the module directory, while the stopwordsModule parameter specifies the name of the module that contains the stopwords directory. By adding these parameters to the di.xml file of the custom module, the Architect can change the location of the stopwords files without modifying the core code or database.
To change the directory from your module
Configure Elasticsearch stopwords

QUESTION 6
A client has multiple warehouses where orders can be fulfilled. The cost of shipping goods from each warehouse varies by day, due to the number of workers available. The Architect needs to make sure that when an order is shipped, it is shipped from the lowest cost warehouse that is open.
How should this functionality be implemented?

A. Create a new class as a preference for Magento\inventoryShipping\piugin\Sales\shipment\AssignSourceCodeToShipmentPlugin to set the lowest-cost warehouse on a shipment.
B. Create a new class implementing Magento\invtntorysourceSelectionApi\Modei\sourceSelectioninterfacece. which returns open warehouses sorted by cost.
C. Create an after plugin On Hagento\InventoryDistanceBasedSourceSelection\Hodel\Algorithms\DistanceBasedAlgorithto sort to Warehouse sources by cost

Correct Answer: B
Section:
Explanation:
According to the Adobe Commerce documentation, the Source Selection Interface is the main interface for implementing custom source selection algorithms. The interface defines a method called execute(), which takes a list of items to be shipped and a stock ID as parameters, and returns a SourceSelectionResultInterface object, which contains the recommended sources and quantities for each item. The Architect can create a new class that implements this interface and provides the logic for finding the lowest-cost warehouse that is open for each item. The Architect can then register the new class as an option for the source selection algorithm in the di.xml file

of the custom module.
Source Selection Algorithm | Adobe Commerce Developer Guide
Source Selection Interface | Adobe Commerce Developer Guide

QUESTION 7
A merchant is using a unified website that supports native Adobe Commerce B2B and B2C with a single store view.
The merchant's objective is to display the B2B account features, such as negotiable quotes and credit limits, in the header of the site on every page for logged-in users who belong to a B2B company account.
Each B2B company possesses its unique shared catalog and customer group, while numerous customer groups for non-B2B customers undergo changes. The merchant insists that this association should not be linked to customer groups.
Which two solutions should the Architect recommend for consideration, taking into account public data and caching? (Choose two.)

A. Create a Virtual Type that switches the theme when a user is part of a B2B company so the output can be modified accordingly in the alternate theme.
B. Create a new HTTP Context variable to allow for separate public content to be cached for users in B2B companies where the output can be modified accordingly.
C. Set whether the current user is part of a B2B company in the customer session and use that data directly to modify the output accordingly.
D. Create a new custom condition for customer segments that allow for choosing whether a user is part of a B2B company and then use this segment to modify the output accordingly.
E. Check if the current user is part of a B2B company within a block class and modify the output accordingly.

Correct Answer: B, D
Section:
Explanation:
Option B is a valid solution because creating a new HTTP Context variable can allow for differentiating the public content cache for users who belong to a B2B company account.The HTTP Context variable can be used to modify the output of the header block accordingly, without affecting the performance or scalability of the site1
Option D is also a valid solution because creating a new custom condition for customer segments can enable targeting users who are part of a B2B company account. The customer segment can be used to modify the output of the header block accordingly, using layout updates or dynamic blocks.This solution can also leverage the existing customer segment functionality and avoid custom coding2
Option A is not a valid solution because switching the theme based on a virtual type can cause performance issues and increase the complexity of the site maintenance.Moreover, switching the theme can affect the entire site appearance, not just the header block3
Option C is not a valid solution because using the customer session data directly to modify the output of the header block can prevent the public content cache from working properly.The customer session data is private and cannot be cached, so this solution can negatively impact the performance and scalability of the site4
Option E is not a valid solution because checking if the current user is part of a B2B company within a block class can also prevent the public content cache from working properly.The block class logic is executed on every request, so this solution can negatively impact the performance and scalability of the site5
1: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.html?lang=en#http-context2: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/marketing/customer-segments.html?lang=en3: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/design/themes.html?lang=en4: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.html?lang=en#private-content5: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.html?lang=en#public-content

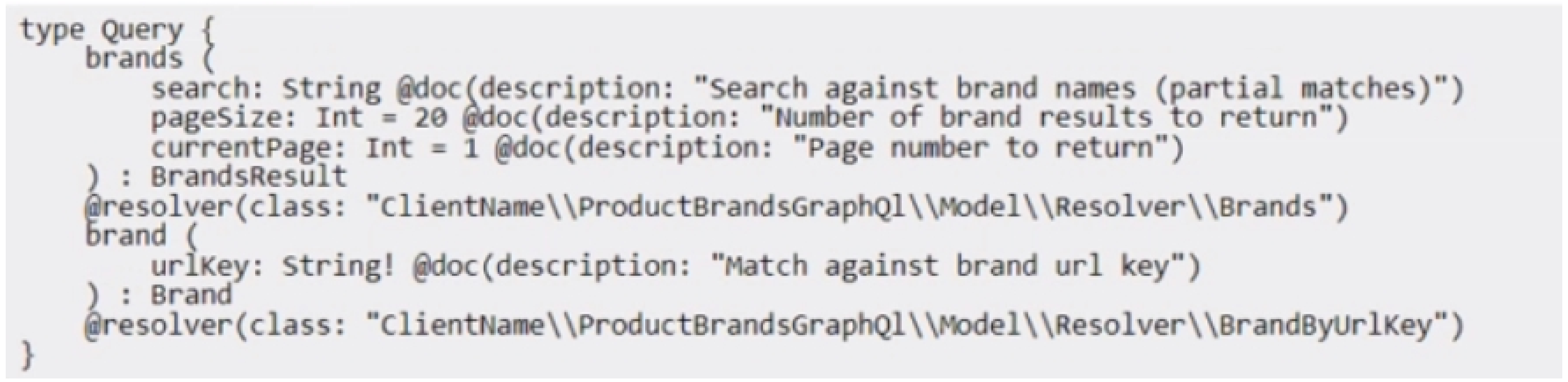
QUESTION 8
An Adobe Commerce Architect needs to customize the workflow of a monthly installments payment extension. The extension is from a partner who is contracted with the default website Payment Service Provider (PSP), which has its own legacy extension (a module using deprecated payment method).
The installment payment partner manages only initializing a payment, and then hands the capture to be executed by the PSP Once the amount is successfully captured, the PSP notifies the website through a webhook. The goal of the webhook is only to create an 'invoice' and save the 'capture information' to be used later for refund requests through the PSP itself.
The Architect needs the most simple solution to capture the requested behavior.
Which solution should the Architect implement?

A. Add a plugin before the $invoice->capture() and change Its input to prevent the call of the $Payment->capture()
B. Change the can_capture attribute for the payment method under config.xml to be <can_capture>0</can_capture>
C. Declare a capture Command with type Magento\Payment\Gateway\Command\NullCommand for the payment method CommandPool in di.xml

Correct Answer: C
Section:

Explanation:
Option C is the correct solution because declaring a capture command with type Magento\Payment\Gateway\Command\NullCommand for the payment method command pool in di.xml will prevent the default capture logic from being executed. The NullCommand class is a dummy implementation of the CommandInterface that does nothing.This way, the payment capture will be handled by the PSP webhook, and the invoice will be created accordingly12
Option A is not a correct solution because adding a plugin before the $invoice->capture() and changing its input to prevent the call of the $payment->capture() will require modifying the core Magento code, which is not recommended.Moreover, this solution will affect all payment methods that use the invoice capture logic, not just the monthly installments payment extension3
Option B is not a correct solution because changing the can_capture attribute for the payment method under config.xml to be <can_capture>0</can_capture> will disable the capture functionality for the payment method entirely.This means that the invoice cannot be created or captured, even by the PSP webhook4
1: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/payments-integrations/payment-gateway/gateway-command.html?lang=en2: https://github.com/magento/magento2/blob/2.4-develop/app/code/Magento/Payment/Gateway/Command/NullCommand.php3: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/customization/best-practices.html?lang=en#do-not-modify-core-code4: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/payments-integrations/payment-gateway/payment-gateway-configuration.html?lang=en#payment-method-configuration

QUESTION 9
An existing Adobe Commerce website is moving to a headless implementation.
The existing website features an 'All Brands'' page, as well as individual pages for each brand. All brand-related pages are cached in Varnish using tags in the same manner as products and categories.
Two new GraphQL queries have been created to make this information available to the frontend for the new headless implementation:



```
type Query {
    brands (
        search: String @doc(description: "Search against brand names (partial matches)")
        pageSize: Int = 20 @doc(description: "Number of brand results to return")
        currentPage: Int = 1 @doc(description: "Page number to return")
    ) : BrandsResult
    @resolver(class: "ClientName\\ProductBrandsGraphQl\\Model\\Resolver\\Brands")
    brand (
        urlKey: String! @doc(description: "Match against brand url key")
    ) : Brand
    @resolver(class: "ClientName\\ProductBrandsGraphQl\\Model\\Resolver\\BrandByUrlKey")
}
```

During testing, the queries sometimes return out-of-date information. How should this problem be solved while maintaining performance?

A. Specify a @cacgecacheable(cacheable: false) directive for each GraphQL query, making sure that the data returned is not cached, and is up to date
B. Specify a $cache(cacheidentity: Path\\To\\identityclass) directive for each GraphQL query, corresponding to a class that adds cache tags for relevant brands and associated products
C. Each GraphQL query's resolver class should inject \Magento\GraphQlcache\Model\cacheableQuery and call setcachevalidity(true) on it as part of the resolver's resolve function.

Correct Answer: B
Section:
Explanation:
This solution ensures that the data returned by the GraphQL queries is up to date, while also maintaining performance. By specifying a $cache(cacheidentity: Path\To\identityclass) directive for each GraphQL query, the relevant brands and associated products will be added as cache tags.

QUESTION 10
An Adobe Commerce Architect is investigating a case where some EAV product attributes are no longer updated.
* The catalog is composed of 20.000 products with 100 attributes each.
* The product updates are run by recurring Adobe commerce imports that happen multiple times a day.
* The Architect finds an error in the logs that indicates an integrity constraint while trying to insert row with id 2147483647.
What is causing this error?

A. Magento framework uses INSERT on DUPLICATE, which leads to reaching the max limit of the increment of the column.
B. Integrity constraints were dropped after upgrading to the latest version, and the integrity checks were missed.
C. EAV attribute import uses REPLACE, which leads to reaching the max limit of the increment of the column

Correct Answer: C
Section:
Explanation:
EAV attribute import uses the REPLACE statement, which deletes and inserts a new row with the same primary key value. This causes the auto-increment column to increase by one for each row, even if the row already exists. If the auto-increment column reaches its maximum value, which is 2147483647 for a signed INT, then any further REPLACE statement will fail with an integrity constraint violation error.Reference:
EAV and extension attributes | Magento 2 Developer Documentation
GitHub - techdivision/import-attribute: This library provides the functionality for the Magento 2 import of EAV attributes
Data integrity in JSON (B) when replacing EAV - Stack Overflow

QUESTION 11
An Adobe Commerce Architect is planning to create a new action that will add gift registry items to the customer's quote. What should the Architect do to guarantee that private content blocks are updated?

A. Mark the controller by setting no-cache HTTP headers
B. Invalidate the status of gift registry indexers
C. Specify a new action in a sections.xml configuration file

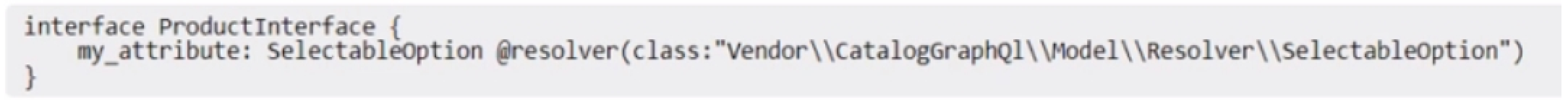
Correct Answer: C
Section:
Explanation:
Private content blocks are sections of the page that are specific to each customer and are not cached by the server. To update these blocks when a customer performs an action, such as adding a gift registry item to the quote, the Adobe Commerce Architect needs to specify the new action in a sections.xml configuration file. This file defines which blocks need to be updated for each action and how often they should be updated. By doing this, the Architect can ensure that the private content blocks are refreshed with the latest data from the server.Reference:
Private content | Magento 2 Developer Documentation
Configure private content | Magento 2 Developer Documentation

QUESTION 12
An Adobe Commerce Architect needs to scope a bespoke news section for a merchants Adobe Commerce storefront. The merchant's SEO agency requests that the following URL structure:
news/{date}/{article_url_key}, where {date} is the publication date of the article, and {article_url_key} is the URL key of the article.
The Architect scopes that a news entity type will be created. The date and URL key data will be stored against each record and autogenerated on save. The values will be able to be manually overridden.

A. The Architect needs to manage routing this functionality and adhere to best practice. Which two options should the Architect consider to meet these requirements? (Choose two.)
B. Create a standard controller route and mapping the internal URLs (such as news/article/view/id/i) to rewrites that are generated on save and then stored in the URL rewrites table.
C. Create a custom router that runs before the standard router and matches the news portion of the URL, then looks for and loads a news article by matching the date and URL key parts of the URL
D. Create a plugin that intercepts Magento\Framework\App\Action: :(), looks for the news portion of the URL, and if it matches, loads the relevant news article by matching the URL date and URL key parts.
E. Create a standard controller route and an index/index controller class that loads the relevant news article by matching the URL date and URL key parts.

Correct Answer: B, C
Section:
Explanation:
These two options are both valid ways to manage routing for the bespoke news section and adhere to best practice. Option B leverages the existing URL rewrite functionality of Adobe Commerce, which allows creating custom URLs for any entity type and storing them in the database. This option requires creating a standard controller route for the news entity type, such as news/article/view/id/i, where i is the news article ID. Then, on saving each news article, a rewrite rule is generated that maps the internal URL to the desired SEO-friendly URL, such as news/{date}/{article_url_key}. The rewrite rule is stored in the url_rewrite table, which is used by the standard router to match and redirect requests.
Option C involves creating a custom router class that implements \Magento\Framework\App\RouterInterface and runs before the standard router in the routing process. The custom router class can match the news portion of the URL and extract the date and URL key parts from it. Then, it can look for and load a news article that matches those values using a model or repository class. If a match is found, it can set the request parameters accordingly and dispatch the request to a controller action that renders the news article page.
Routing | Adobe Commerce Developer Guide
URL Rewrites | Adobe Commerce Developer Guide
Custom Router | Adobe Commerce Developer Guide

QUESTION 13
An external system integrates functionality of a product catalog search using Adobe Commerce GraphQL API. The Architect creates a new attribute my_attribute in the admin panel with frontend type select-Later, the Architect sees that Productlnterf ace already has the field my_attribute, but returns an Int value. The Architect wants this field to be a new type that contains both option id and label.
To meet this requirement, an Adobe Commerce Architect creates a new module and file etc/schema.graphqls that declares as follows:



```
interface ProductInterface {
    my_attribute: SelectableOption @resolver(class:"Vendor\\CatalogGraphQl\\Model\\Resolver\\SelectableOption")
}
```

After calling command setup:upgrade, the introspection of Productlnterface field my_attribute remains Int. What prevented the value type of field my_attribute from changing?

A. The Magento_CatalogGraphQI module occurs later in sequence than the Magento_GraphQI module and merging output of dynamic attributes schema reader overrides types declared in schema.graphqls
B. The fields of Productlnterface are checked during processing schema.graphqls files. If they have a corresponding attribute, then the backendjype of product attribute is set for field type.
C. The interface Productlnterface is already declared in Magento.CatalogGraphQI module. Extending requires use of the keyword extend before a new declaration of Productlnterface.

Correct Answer: C
Section:
Explanation:
According to the Adobe Commerce documentation, to extend an existing GraphQL interface, the keyword extend must be used before the interface name. This indicates that the new declaration is adding or modifying fields to the existing interface, rather than redefining it. If the keyword extend is omitted, the new declaration will be ignored and the original interface will be used. In this case, the Architect wants to change the type of the my_attribute field in the ProductInterface interface, which is already declared in the Magento.CatalogGraphQl module. Therefore, the Architect should use the keyword extend before declaring the ProductInterface interface in the schema.graphqls file of the custom module. This will allow the Architect to override the type of the my_attribute field from Int to MyAttributeType.
Extend existing schema | Adobe Commerce Developer Guide
Schema language with GraphQL | Adobe Commerce

QUESTION 14
An Adobe Commerce store owner sets up a custom customer attribute 'my.attribute'.
An Architect needs to display additional content on the home page, which should display only to Customers with 'my.attribute' of a certain value and be the same content for all of them. The website is running Full Page Cache.
With simplicity in mind, which two steps should the Architect take to implement these requirements? (Choose two.)

A. Add a new context value of 'my_attribute' to Magento\Framework\App\Http\Context
B. Create a Customer Segment and use 'my.attribute' in the conditions
C. Add a custom block and a pHTML template with the content to the cmsjndexjndex.xml layout
D. Add a dynamic block with the content to the Home Page
E. Use customer-data JS library to retrieve 'my.attribute' value

Correct Answer: A, D
Section:
Explanation:
To display additional content on the home page based on a custom customer attribute, the Architect needs to do the following steps:
Add a new context value of ''my_attribute'' to Magento\Framework\App\Http\Context. This will allow the Full Page Cache to generate different versions of the page for customers with different values of ''my.attribute''. The context value can be set using a plugin on the Magento\Customer\Model\Context class.
Add a dynamic block with the content to the Home Page. A dynamic block is a type of content block that can be configured to display only to specific customer segments or conditions. The Architect can use the 'my.attribute' in the conditions of the dynamic block and assign it to the Home Page in the Content > Blocks section of the Admin Panel.Reference:
Private content | Magento 2 Developer Documentation
Dynamic Blocks | Adobe Commerce 2.3 User Guide - Magento

QUESTION 15
An Adobe Commerce Architect designs a data flow that contains a new product type with its own custom pricing logic to meet a merchant requirement. Which three steps are required when adding a product type with

custom pricing? (Choose three.)

A. Content of the etc/product_types.xml file
B. Data patch to register the new product type
C. Hydrator for attributes belonging to the new product type
D. New price model extending \Magento\Catalog\Model\Product\Type\Price
E. Custom type model extended from the abstract Product Type model
F. A new class with custom pricing logic, extending the abstract Product model class

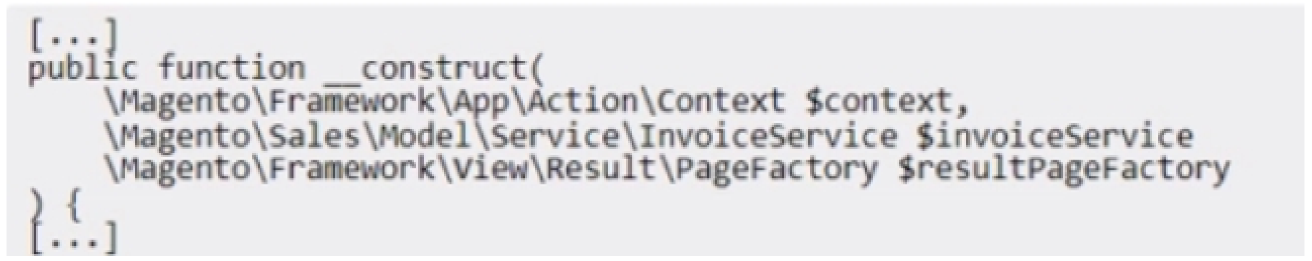
Correct Answer: A, D, E
Section:
Explanation:
To add a product type with custom pricing, the Architect needs to do the following steps:
Create a content of the etc/product_types.xml file that defines the new product type, its label, model, index priority, and price model.This file is used to register the new product type and its associated classes in Magento1.
Create a new price model that extends \Magento\Catalog\Model\Product\Type\Price and implements the custom pricing logic for the new product type.The price model is responsible for calculating the final price of the product based on various factors, such as special price, tier price, catalog price rules, etc2.
Create a custom type model that extends from the abstract Product Type model (\Magento\Catalog\Model\Product\Type\AbstractType) and overrides the methods related to the product type behavior, such as prepareForCart, getAssociatedProducts, etc.The type model defines how the product type interacts with other components, such as quote, order, cart, etc3.Reference:
How to add a new product type in Magento 2? (MageStackDay mystery question 1) - Magento Stack Exchange
Magento 2: How to create custom product types - BelVG Blog
Magento 2: How to create custom product types - BelVG Blog

QUESTION 16
An Adobe Commerce Architect needs to log the result of a ServiceClass:: getData method execution after all plugins have executed. The method is public, and there are a few plugins declared for this method. Among those plugins are after and around types, and all have sortOrder specified.
Which solution should be used to meet this requirement?

A. Declare a new plugin with the sortOrder value lower than the lowest declared plugin sortOrder and implement aroundGetData method.
B. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement afterGetData method.
C. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement aroundGetData method.

Correct Answer: B
Section:
Explanation:
This solution ensures that the new plugin will execute after all the existing plugins for the ServiceClass::getData method, and will be able to log the final result of the method execution. The afterGetData method of the new plugin will receive the result of the method as a parameter, and can use any logging mechanism to record it. The sortOrder value of the new plugin should be higher than the highest declared plugin sortOrder, so that it will run last in the sequence of plugins. The after type of plugin is preferred over the around type of plugin, because it is simpler and more efficient, and does not require calling the proceed() method.
Plugins (Interceptors) | Adobe Commerce Developer Guide
Plugin best practices | Adobe Commerce Developer Guide

QUESTION 17
While developing a new functionality for a website in developer mode with all cache types enabled, an Adobe Commerce Developer needs to add \Magento\Sales\Model\Service\InvoiceService SinvoiceService as a new dependency to an existing page action controller in Vendor\CustomModule\Controller\Index\Index . This is accomplished as follows:



```
[...]
public function __construct(
    \Magento\Framework\App\Action\Context $context,
    \Magento\Sales\Model\Service\InvoiceService $invoiceService
    \Magento\Framework\View\Result\PageFactory $resultPageFactory
) {
[...]
```

After cleaning the f ull_page cache and reloading the page, the developer encounters the following exception:
Recoverable Error: Argument 2 passed to Vendor\CustomModule\Controller\Index\Index::__construct() must be an instance of \Magento\Sales\Model\Service\InvoiceService [...]
Which action should the Architect recommend to the developer to fix this error?

A. Clean the block_html cache along with full_page cache.
B. Add the new \Magento\sales\Model\service\invoiceService Sinvoiceservice dependency at the end of the constructor signature.
C. Remove the generated Child ClaSS from generated/code/Vendor/CustomModule/Controller/Index/Index.

Correct Answer: C
Section:
Explanation:
The error is caused by the generated child class not being updated with the new dependency. Removing the generated child class will allow the system to generate a new child class with the correct dependency. The generated child class is a proxy class that extends the original controller class and overrides the constructor to inject the dependencies using the object manager. The generated child class is created when the system runs in developer mode with cache enabled, to avoid performance issues. However, when a new dependency is added to the original controller class, the generated child class does not reflect the change and causes a mismatch in the constructor arguments. Therefore, deleting the generated child class from the generated/code directory will solve the problem.
Generated code | Adobe Commerce Developer Guide
Constructor signature change | Adobe Commerce Developer Guide

QUESTION 18
A representative of a small business needs an Adobe Commerce Architect to design a custom integration of a third-party payment solution. They want to reduce the list of controls identified in their Self-Assessment Questionnaire as much as possible to achieve PCI compliance for their existing Magento application.
Which approach meets the business needs?

A. Utilize the Advanced Encryption standard (aes-256) algorithm to encrypt all customer-sensitive data from the payment module.
B. Utilize the payment provider iframe system to isolate content of the embedded frame from the parent web page.
C. Utilize a trusted signed certificate issued by a Certification Authority (CA) to secure each connection made by the payment solution protocol via https.

Correct Answer: B
Section:
Explanation:
Using an iframe system for payment integration can help reduce the PCI scope and compliance burden for the merchant, as the payment data is collected and processed by the payment service provider (PSP) within the iframe, without touching the merchant's website or server. This way, the merchant can leverage the PSP's PCI certification and avoid storing or transmitting any sensitive cardholder data on their own system. The iframe also provides a secure barrier between the host webpage and the loaded page, preventing any access or manipulation of the payment data by malicious actors.To implement this approach, the merchant needs to embed the PSP's payment form in their checkout page using an iframe element, and configure the communication between the iframe and the host page using JavaScript123.

QUESTION 19
A merchant asks for a new category attribute to allow uploading an additional mobile image against categories. The merchant utilizes the content staging and preview feature in Adobe Commerce and wants to schedule and review changes to this new mobile image field.
A developer creates the attribute via a data patch and adds it to view/adminhtml/ui_component/category_f orm. xml. The attribute appears against the category in the main form, but does not appear in the additional form when scheduled updates are made.
To change this attribute when scheduling new category updates, which additional action should the Architect ask the developer to take?

A. The attribute must have its apply_to field set to 'staging' in the data patch file.
B. The attribute must have <item- name=''allow_staging' xsi:type=''boolean''>true</item> set in the cjt.gopy_for-.xni file under the attributes config' section.
C. The attribute must also be added to view/adminhtml/ui_co-component/catalogstaging_category_update_form.xml.

Correct Answer: C
Section:
Explanation:
This action is necessary to make the attribute available for content staging and preview. According to the Adobe Commerce documentation, the catalogstaging_category_update_form.xml file defines the fields that are displayed in the Scheduled Changes section of the category form. The file extends the category_form.xml file and adds additional fields that are specific to content staging, such as start and end dates, campaign name, description, etc. To include a custom category attribute in the Scheduled Changes section, the attribute must also be declared in the catalogstaging_category_update_form.xml file with the same configuration as in the category_form.xml file.
Content staging | Adobe Commerce Developer Guide
Create a category attribute | Adobe Commerce Developer Guide

QUESTION 20
An Adobe Commerce Architect needs to create a new customer segment condition to enable admins to specify an Average sales amount' condition for certain segments.
The Architect develops the custom condition under Vendor\Module\Model\Segment\Condition\AverageSalesAmount with all of its requirements:

```php
<?php

namespace Vendor\Module\Plugin;

use Magento\CustomerSegment\Model\Segment\Condition\Combine;

class AddNewChildOption
{
    public function afterGetNewChildSelectOptions(Combine $subject, array $result): array {
        $conditions = [
                [
                    'value' => \Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount::class,
                    'label' => __('Average sales amount'),
                ]
        ];

        return array_merge_recursive($result, $conditions);
    }
}
```

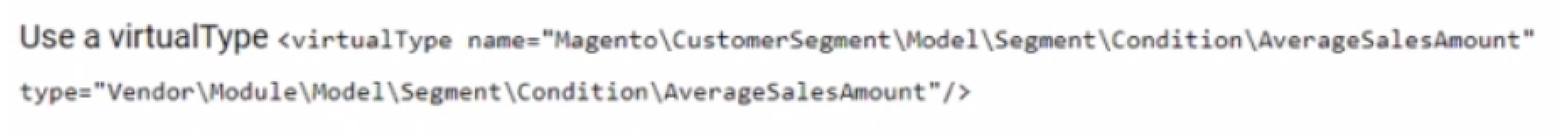
During testing, the following error appears:

```
"Class Magento\\CustomerSegment\\Model\\Segment\\Condition\\Vendor\\Module\\
Model\\Segment\\Condition\\AverageSalesAmount does not exist at
/var/www/vendor/magento/framework/Code/Reader/ClassReader.php"
```

What should the Architect do to fix the problem?
A)
Set the class to be `\Vendor\Module\Model\Segment\Condition\AverageSalesAmount` for the `$conditions value` attribute

B)
Use a preference `<preference for="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount"`
`type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>`

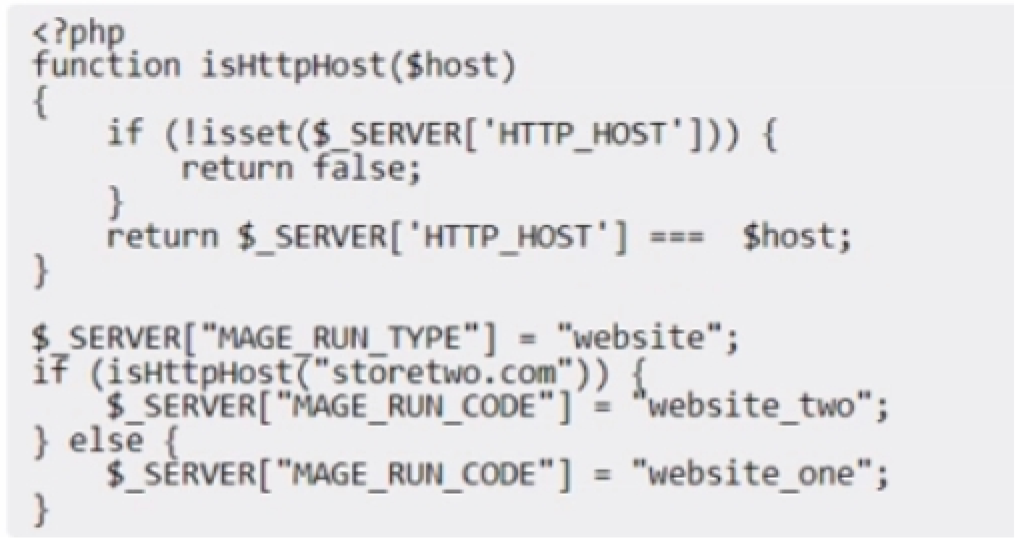
C)


Use a virtualType
```
<virtualType name="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount"
type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>
```

A. Option A
B. Option B
C. Option C

Correct Answer: B
Section:
Explanation:
The error is caused by the missing class declaration for the custom condition class. According to the Adobe Commerce documentation, to create a custom customer segment condition, the class must extend the \Magento\CustomerSegment\Model\Condition\AbstractCondition class and implement the \Magento\CustomerSegment\Model\Condition\Combine\Interface interface. The class must also declare its name, label, value type, and attribute code properties. Option B is the only option that correctly declares the class with the required properties and inheritance. Option A and Option C are incorrect because they do not extend the AbstractCondition class or implement the CombineInterface interface, and they do not declare the name, label, value type, or attribute code properties.
Create a customer segment condition | Adobe Commerce Developer Guide
AbstractCondition | Adobe Commerce Developer Guide

QUESTION 21
A single Adobe Commerce Cloud instance is set up with two websites (each with a single store view) with different domains.
* The default website is website_one, with store view store_one, and domain storeone. com.
* The second website is website_two, with store view store_two, and domain storetwo. com.
The magento-vars. php file is set up as follows to determine which website each request runs against:


```
<?php
function isHttpHost($host)
{
    if (!isset($_SERVER['HTTP_HOST'])) {
        return false;
    }
    return $_SERVER['HTTP_HOST'] ===  $host;
}

$_SERVER["MAGE_RUN_TYPE"] = "website";
if (isHttpHost("storetwo.com")) {
    $_SERVER["MAGE_RUN_CODE"] = "website_two";
} else {
    $_SERVER["MAGE_RUN_CODE"] = "website_one";
}
```
When testing a new GraphQL integration, all requests returned data relating to the default website, regardless of the domain. What is causing this issue?

A. The magento-vars.php file is not processed for any GraphQL requests, so the default website is always processed.
B. $_server['mage_run_cooe') needs to be set to store and *$_SERVER['MAGE_RUN_TYPE'] needs to be set to the store code instead.
C. GraphQL requests are always run against the default store view unless a store header or store cookie is provided.

Correct Answer: C
Section:
Explanation:
The magento-vars.php file is used to set the website or store view based on the HTTP host, but it does not affect GraphQL requests. GraphQL requests are handled by a separate controller that does not use the magento-vars.php file. Instead, GraphQL requests use the default store view of the default website, unless a store header or store cookie is provided in the request. The store header or cookie should contain the store code of the

desired store view.For example, to query data from website_two, the request should include a header likestore: store_twoor a cookie likestore=store_two12.Reference:
GraphQL overview | Adobe Commerce 2.4 User Guide - Magento
How to set up multiple websites with Magento 2 - Mageplaza

QUESTION 22
An Architect needs to integrate an Adobe Commerce store with a new Shipping Carrier. Cart data is sent to the Shipping Carrier's API to retrieve the price and display to the customer. After the feature is implemented on the store, the API hits its quota and returns the error 'Too many requests'. The Shipping Carrier warns the store about sending too many requests with the same content to the API.
In the carrier model, what should the Architect change to fix the problem?

A. ln_doShipmentRequest()f call canCollectRates() before sending request to the API.
B. Override getResponse, save the response to a variable, check if the response exists, then return.
C. Implement _setCachedQuotes() and _getCachedQuotes(), return the data if the request matches.

Correct Answer: C
Section:
Explanation:
The carrier model class can implement caching methods to store and retrieve the quotes from the API based on the request parameters. This can reduce the number of API calls and improve the performance of the shipping rate calculation. The _setCachedQuotes() method can save the response from the API to a cache storage, and the _getCachedQuotes() method can check if there is a cached response for the current request and return it if it exists.Reference:Caching in carrier model,Carrier model interface

QUESTION 23
An Architect working on a headless Adobe Commerce project creates a new customer attribute named my_attribute. Based on the attribute value of the customer, the results of GraphQl queries are modified using a plugin. The frontend application is communicating with Adobe Commerce through Varnish by Fastly. which is already caching the queries that will be modified. The Adobe Commerce Fastly extension is installed, and no other modifications are made to the application.
Which steps should the Architect take to make sure the vcl_hash function of Varnish also considers the newly created attribute?

A. Create a new ClaSS inheriting from Magento\GraphQlCache\Model\CacheId\CacheIdFactorProvidftrInterface and returning the Value of my_attribute from the getFactorValue function and my_attribute from the getFactorName function. Then add this class through Dl to the idFactorProviders array of Magento\GraphQlCache\Model\CacheId\CacheIdCalculator.
B. Create a new class inheriting from Magento\Framework\GraphQi\Query\Resolver\identityinterfaca and returning the value of my_attribute from the getidentities function. Then specify a ecache(cacheidentity: Path\\To\\identityclass) directive for each GraphQL query to include the newly created IdentityClass to each query that adds the cache tags for each customer.
C. Create a new class inheriting from Magento\customer\customerData\stctionSourceinterface and returning the value of my_attribute from the getSectionData function. Then add this ClaSS through the sectionSourceMap array Of Magento\Customer\CustomerData\SectionPoolInterface.

Correct Answer: A
Section:
Explanation:
To make sure the vcl_hash function of Varnish considers the newly created attribute, the Architect needs to do the following steps:
Create a new class that implements the Magento\GraphQlCache\Model\CacheId\CacheIdFactorProviderInterface interface. This interface defines two methods: getFactorName and getFactorValue. The getFactorName method should return the name of the attribute, in this case, my_attribute.The getFactorValue method should return the value of the attribute for the current customer, which can be obtained from the customer session or customer repository1.
Add this class to the idFactorProviders array of Magento\GraphQlCache\Model\CacheId\CacheIdCalculator through dependency injection. The CacheIdCalculator is responsible for generating a cache ID for each GraphQL request based on the factors provided by the idFactorProviders.By adding the new class to this array, the Architect ensures that the cache ID will include the value of my_attribute1.
The cache ID is then used by Varnish to hash and lookup the cached response for each request.By including my_attribute in the cache ID, the Architect ensures that Varnish will serve different responses based on the attribute value of the customer2.Reference:
Magento_GraphQlCache module | Magento 2 Developer Documentation
Varnish caching | Adobe Commerce 2.4 User Guide - Magento

QUESTION 24
An Architect wants to create an Integration Test that does the following:

* Adds a product using a data fixture
* Executes $this->someLogic->execute($product) on the product
* Checks if the result is true.
$this->someLogic has the correct object assigned in the setup() method.
Product creation and the tested logic must be executed in the context of two different store views with IDs of 3 and 4, which have been created and are available for the test.
How should the Architect meet these requirements?

A. Create two test classes with one test method each. Use the @magentoExecuteinstoreContext 3 and $MagentoExecuteinStoreContext 4 annotations on the class level.
B. Create one test class with two test methods. Use the emagentostorecontext 3 annotation in one method and amagentostorecontext 4 in the other one.
C. Create one test class with one test method. Use the \Magento\TestFramework\store\ExecuteinstoreContext class once in the fixture and another time in the test.

Correct Answer: C
Section:
Explanation:
To create an integration test that executes different logic in different store views, the Architect needs to do the following steps:
Create one test class that extends \Magento\TestFramework\TestCase\AbstractController or \Magento\TestFramework\TestCase\AbstractBackendController, depending on the type of controller being tested1.
Create one test method that uses the @magentoDataFixture annotation to specify the data fixture file that creates the product2.
Use the \Magento\TestFramework\Store\ExecuteInStoreContext class to execute the fixture and the tested logic in different store views. This class has a method called executeInStoreContext, which takes two parameters: the store ID and a callable function.The callable function will be executed in the context of the given store ID, and then the original store ID will be restored3. For example:
PHPAI-generated code. Review and use carefully.More info on FAQ.

```
public function testSomeLogic()
{
// Get the product from the fixture
$product = $this->getProduct();
// Get the ExecuteInStoreContext instance from the object manager
$executeInStoreContext = $this->_objectManager->get(\Magento\TestFramework\Store\ExecuteInStoreContext::class);
// Execute the fixture in store view 3
$executeInStoreContext->executeInStoreContext(3, function () use ($product) {
// Do some operations on the product in store view 3
});
// Execute the tested logic in store view 4
$result = $executeInStoreContext->executeInStoreContext(4, function () use ($product) {
// Call the tested logic on the product in store view 4
return $this->someLogic->execute($product);
});
// Assert that the result is true
$this->assertTrue($result);
}
```

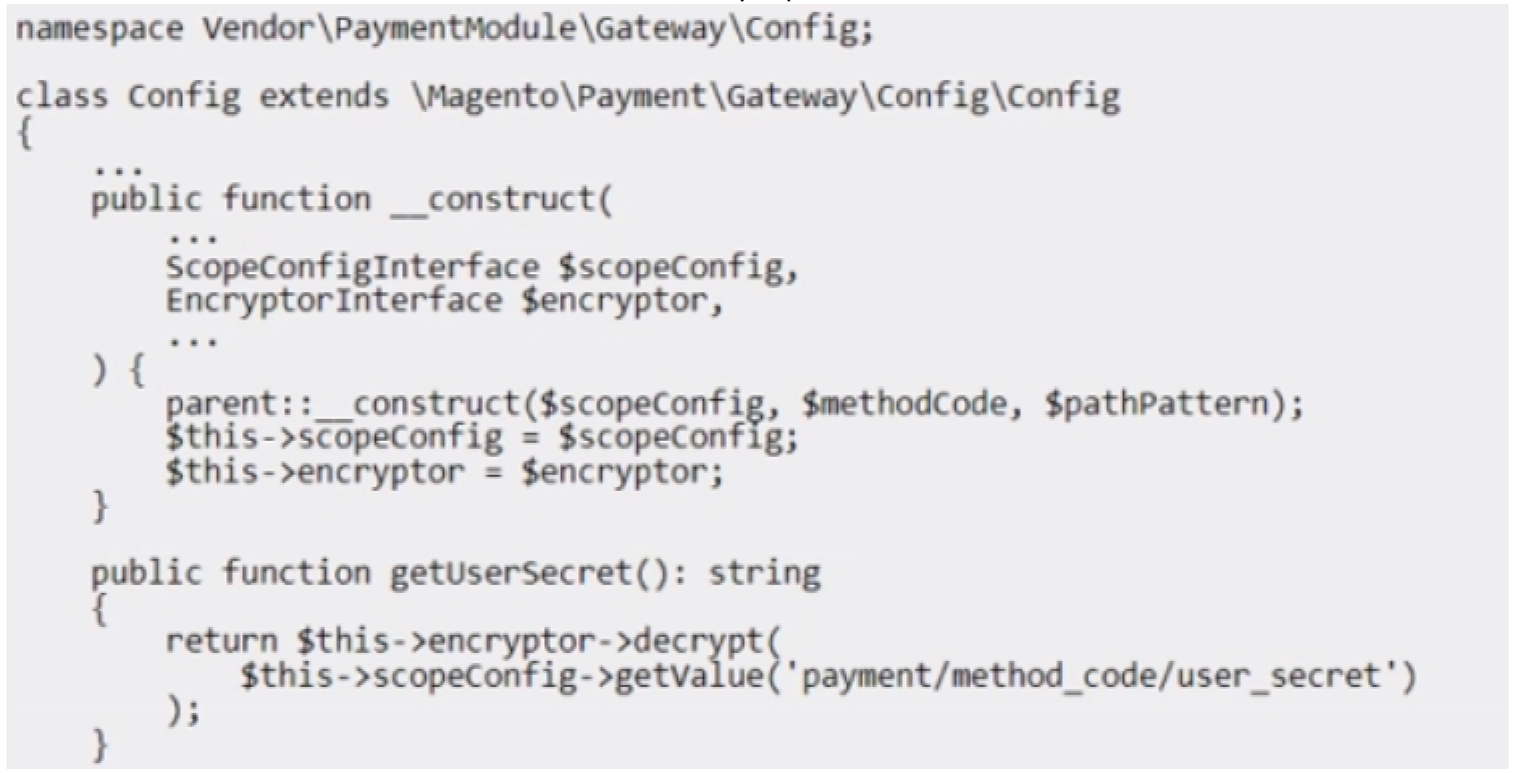
Integration tests | Magento 2 Developer Documentation
Data fixtures | Magento 2 Developer Documentation
Magento\TestFramework\Store\ExecuteInStoreContext | Magento 2 Developer Documentation

QUESTION 25
An Adobe Commerce Architect notices that the product price index takes too long to execute. The store is configured with multiple websites and dozens of customer groups.
Which two ways can the Architect shorten the full price index execution time? (Choose two.)

A. Set mage_ihdexer_threads_COUNT environment variable to enable parallel mode
B. Move catalog_Price_index indexer to another custom indexer group
C. Enable price index customer group merging for products without tier prices

D. Set Customer Share Customer Accounts Option to Global
E. Edit customer groups to exclude websites that they are not using

Correct Answer: A, C
Section:
Explanation:
The product price index can be optimized by using parallel mode and customer group merging. Parallel mode allows the indexer to run multiple threads simultaneously, which can speed up the indexing process. Customer group merging reduces the number of rows in the price index table by merging customer groups that have the same product prices. This can improve the performance of the price index queries and reduce the index size.Reference:Indexing optimization,Price index customer group merging

QUESTION 26
While reviewing a newly developed pull request that refactors multiple custom payment methods, the Architect notices multiple classes that depend on \Magento\Framework\Encryption\EncryptorInterface to decrypt credentials for sensitive dat a. The code that is commonly repeated is as follows:

```
namespace Vendor\PaymentModule\Gateway\Config;

class Config extends \Magento\Payment\Gateway\Config\Config
{
    ...
    public function __construct(
        ...
        ScopeConfigInterface $scopeConfig,
        EncryptorInterface $encryptor,
        ...
    ) {
        parent::__construct($scopeConfig, $methodCode, $pathPattern);
        $this->scopeConfig = $scopeConfig;
        $this->encryptor = $encryptor;
    }

    public function getUserSecret(): string
    {
        return $this->encryptor->decrypt(
            $this->scopeConfig->getValue('payment/method_code/user_secret')
        );
    }
```
The Architect needs to recommend an optimal solution to avoid redundant dependency and duplicate code among the methods. Which solution should the Architect recommend?

A. Create a common config service class vndor\Pay-ient\Gatway\conf ig\conf ig under Vendor.Payment and use it as a parent class for all of the
B. Replace all Vendor\PaymentModule\Gateway\Config\Config ClaSSeS With virtualType Of Magento\Payiaent\Gateway\Conf ig\Conf ig and Set <user_secret backend_Model='Magento\Config\Model\Config\Backend\Encrypted' /> Under config.xml
C. Add a plugin after the getvalue method of $scopeConfig, remove the $encryptor from dependency and use it in the plugin to decrypt the value if the config name is user.secret'

Correct Answer: B
Section:
Explanation:
The Architect should recommend replacing all Vendor\PaymentModule\Gateway\Config\Config Classes with virtualType of Magento\Payment\Gateway\Config\Config and setting <user_secret backend_Model=''Magento\Config\Model\Config\Backend\Encrypted'' /> under config.xml. This will avoid redundant dependency and duplicate code among the methods. The virtualType of Magento\Payment\Gateway\Config\Config will inherit the functionality of the base class and allow the customization of the constructor arguments, such as the pathPattern and valueHandlerPool.The backend_Model attribute of the user_secret field will specify that the value of this field should be encrypted and decrypted by the Magento\Config\Model\Config\Backend\Encrypted class, which implements the \Magento\Framework\App\Config\ValueInterface interface and uses the \Magento\Framework\Encryption\EncryptorInterface internally12.This way, the payment modules do not need to depend on the \Magento\Framework\Encryption\EncryptorInterface or the \Magento\Framework\App\Config\ScopeConfigInterface directly, and can use the getValue method of the Magento\Payment\Gateway\Config\Config class to get the decrypted value of the user_secret field3.Reference:

How to encrypt system configuration fields in Magento 2 - Mageplaza
Magento 2: How to Encrypt/Decrypt System Configuration Fields - Webkul Blog
Magento 2: How to create custom payment method - BelVG Blog

QUESTION 27
Since the last production deployment, customers can not complete checkout.
The error logs show the following message multiple times:
main.CRITICAL: Report ID: webapi-61b9fe83f0c3e; Message: Infinite loop detected, review the trace for the looping path
The Architect finds a deployed feature that should limit delivery for some specific postcodes.
The Architect sees the following code deployed in etc/webapi_rest/di. xml and etc/frontend/di. Xml

```
<type name="Magento\Shipping\Model\Rate\Result">
    <plugin name="RestrictDeliveryMethods" type="Vendor\RestrictDeliveryMethods\Plugin\Shipping\LimitRates"/>
</type>
```

LimitRates.php:

```
public function __construct(
    \Magento\Checkout\Model\Session $session,
    ResultProvider $resultProvider
) {
    $this->session = $session;
    $this->resultProvider = $resultProvider;
}

public function afterGetAllRates(\Magento\Shipping\Model\Rate\Result $subject, array $result): array
{
    return $this->resultProvider->getLimitedRates($this->session->getQuote(), $result);
}
```

Which step should the Architect perform to solve the issue?

A. Change 'after' plugin with 'around' plugin. The issue is being caused by calling the result provider code after the code of the original method.
B. Replace the injected dependency Of \Magento\Checkout\Model\Session With \Magento\FraBievork\Session\SessionManagerInterf ace
C. Inject an instance Of Magento\(Quote\Api\CartRepositoryInterface and receive Cart instance Via $thiscartRepository->get($this->session->getQuoteId())

Correct Answer: C
Section:

QUESTION 28
An Adobe Commerce Architect runs the PHP Mess Detector from the command-line interface using the coding standard provided with Adobe Commerce. The following output appears:

```
FILE: /home/architect/Sites/some-project/app/code/MyVendor/MyModule/Service/MyService.php
--------------------------------------------------------------------------------------
18  | VIOLATION | The class MyService has a coupling between objects value of 15.
                  Consider to reduce the number of dependencies under 13.
```

The Architect looks at the class and notices that the constructor has 15 parameters. Five of these parameters are scalars configuring the behavior of MyService. The class also contains three constants referencing one other class.
How should the Architect fix the code so that it complies with the coding standard rule?

A. Modify the code of MyService so that the number of different classes and interfaces referenced anywhere inside the class is fewer than 13.
B. Consolidate the constants referencing other classes into a string representation.
C. Introduce a new class accepting those five scalars and use it in the constructor and the remaining logic of MyService.

Correct Answer: C

Section:
Explanation:
The issue is being caused by the high coupling between objects (CBO) value of the class MyService.CBO is a metric that measures the number of classes that are coupled to a given class, either by method calls, property or parameter references, inheritance, or constants1.A high CBO value indicates that the class is too tightly coupled with other classes, which makes it more difficult to maintain, test, and reuse2. To reduce the CBO value, the Architect should introduce a new class that encapsulates the five scalar parameters that configure the behavior of MyService. This way, the constructor of MyService will only have one parameter of the new class type, instead of five scalar parameters. This will also make the code more readable and maintainable, as the new class can provide methods to access and manipulate the configuration data.The constants referencing other classes should not be consolidated into a string representation, as this would not reduce the CBO value and would make the code less clear and type-safe3.The number of different classes and interfaces referenced anywhere inside the class is not relevant for the CBO metric, as it only counts the classes that are coupled to the given class1.Reference:CBO coupling between object,Coupling Between Object classes (CBO),Cohesion and coupling of an object in OO programming

QUESTION 29
Due to a marketing campaign, a website is experiencing a very large number of simultaneously placed orders, which is affecting checkout performance. The website is in the production deploy mode.
Which two website settings can an Architect optimize to decrease the impact on checkout performance? (Choose two.)

A. Asynchronous indexing admin panel Setting (Stores > Settings > Configuration > Advanced > Developer > Grid Settings > Asynchronous indexing) can be enabled by executing the following CLI Command: bin/Magento config:set dev/grid/async_indexing 1
B. Asynchronous email notifications admin panel setting (stores > settings > configuration > sales > sales Emails > General settings > Asynchronous) can be enabled
C. A new database can be created and the Split Database feature can be automatically configured with the following command: bin/Magento setup:db-schema:spiit-sales --host'<checkout db host or ip>- --dbnanie''<name>' --username'<checkout db username)' --password'''
D. The website deploy mode can be set to siege by executing the following CLI command: bin/Magento deploy:mode:set siege, provided that it will be changed back to production as soon as the number of simultaneously placed orders decreases to acceptable levels
E. Multithreaded checkout processing admin panel setting (stores > settings > configuration > sales > checkout > General settings > Asynchronous) can be set to a higher value representing the number of PHP threads used exclusively for checkout

Correct Answer: A, C
Section:
Explanation:
Option A is correct because enabling asynchronous indexing can improve the checkout performance by reducing the database load and avoiding locking issues. Asynchronous indexing allows the indexers to run in the background without affecting the frontend operations.The commandbin/magento config:set dev/grid/async_indexing 1can be used to enable this option in the production mode1.
Option C is correct because creating a new database and splitting the sales tables can also improve the checkout performance by distributing the database load and avoiding contention. Splitting the database allows the checkout and order management operations to use a separate master database from the rest of the Magento application tables.The commandbin/magento setup:db-schema:split-sales --host='<checkout db host or ip>' --dbname='<name>' --username='<checkout db username>' --password=''can be used to configure this feature2.
Option B is incorrect because enabling asynchronous email notifications does not affect the checkout performance directly. Asynchronous email notifications allow the order confirmation emails to be sent in batches by a cron job instead of immediately after placing an order.This option can reduce the server load and improve the customer experience, but it does not impact the checkout process itself3.
Option D is incorrect because there is no such deploy mode as siege in Magento 2. The available deploy modes are default, developer, and production.Changing the deploy mode can affect the performance, caching, and error handling of the Magento application, but it does not directly affect the checkout performance4.
Option E is incorrect because there is no such admin panel setting as multithreaded checkout processing in Magento 2. The number of PHP threads used for checkout is determined by the web server configuration and the PHP-FPM settings, not by the Magento application settings.Increasing the number of PHP threads may improve the checkout performance, but it also requires more server resources and may cause other issues5.
1: Asynchronous indexing | Adobe Commerce Developer Guide
2: Split database performance solution | Adobe Commerce Developer Guide
3: Sales Emails | Adobe Commerce User Guide
4: Set up Magento modes | Adobe Commerce Developer Guide
5: PHP-FPM configuration settings | Adobe Commerce Developer Guide

QUESTION 30
An Adobe Commerce Architect is reviewing API-functional test code. Some tests send errors to indicate that the customer address does not exist. The test codes show the following:

```
/**
 * @magentoDataFixture Magento/Customer/_files/customer_one_address.php`
 * ...
 */
public function testMyUseCaseTestForCartAddress(): void
```
Which step should the Architect take to fix the test errors?

A. Change the annotation to Use@magentoApiDataFixture Magento/Customer/_files/ instead Or dmagentoDataFixture Magento/Customer/_files/customer_one_address.php
B. Set the annotation to USe AmagentoPersistDataFixture Magento/Cu5tomer/_f iles/custcwer_one_address.php instead Of @magentoDataFixture Magento/Customer/_f iles/customer_one_address.php
C. Update the annotation to Specify addreSSjd niagentoDataFixture Magento/Customer/_files/customer_one_address.php with:{Maddress_id':'$address.id$'}

Correct Answer: B
Section:
Explanation:
The issue is being caused by the use of @magentoDataFixture annotation, which creates a temporary data fixture that is rolled back after each test execution1. This means that the customer address created by the fixture is not persisted in the database and cannot be retrieved by subsequent tests.To fix this, the Architect should use @magentoPersistDataFixture annotation, which creates a permanent data fixture that is not rolled back after each test execution2. This way, the customer address created by the fixture will be persisted in the database and can be accessed by subsequent tests.Changing the annotation to use @magentoApiDataFixture or specifying address_id in the annotation will not solve the issue, as they are not related to the persistence of the data fixture3.Reference:Data fixtures,Persistent data fixtures,API-functional tests

QUESTION 31
A custom cron job has been added to an Adobe Commerce system to collect data for several reports. Its crontab. xml configuration is as follows:
```
<config>
    <group id="default">
        <job name="gather_reporting_data" instance="Vendor\Reports\Cron\DataGatherer" method="execute">
            <schedule>0 0 * * *</schedule>
        </job>
    </group>
</config>
```
The job is data intensive and runs for between 20 and 30 minutes each night.
Within a few days of deployment, it is noticed that the sites sitemap. xml file has not been updated since the new job was added.
What should be done to fix this issue?

A. Change the schedule of the siten.aP_generate cron job to 30 0 * * *so that it runs after the gather_reporting_data job has completed.
B. Create a new cron group for the reporting job, specifying <use_separate_process>i</use_separate_process>
C. Break the data gathering job into a number of smaller jobs, so that each individual job runs for a maximum of 5 minutes

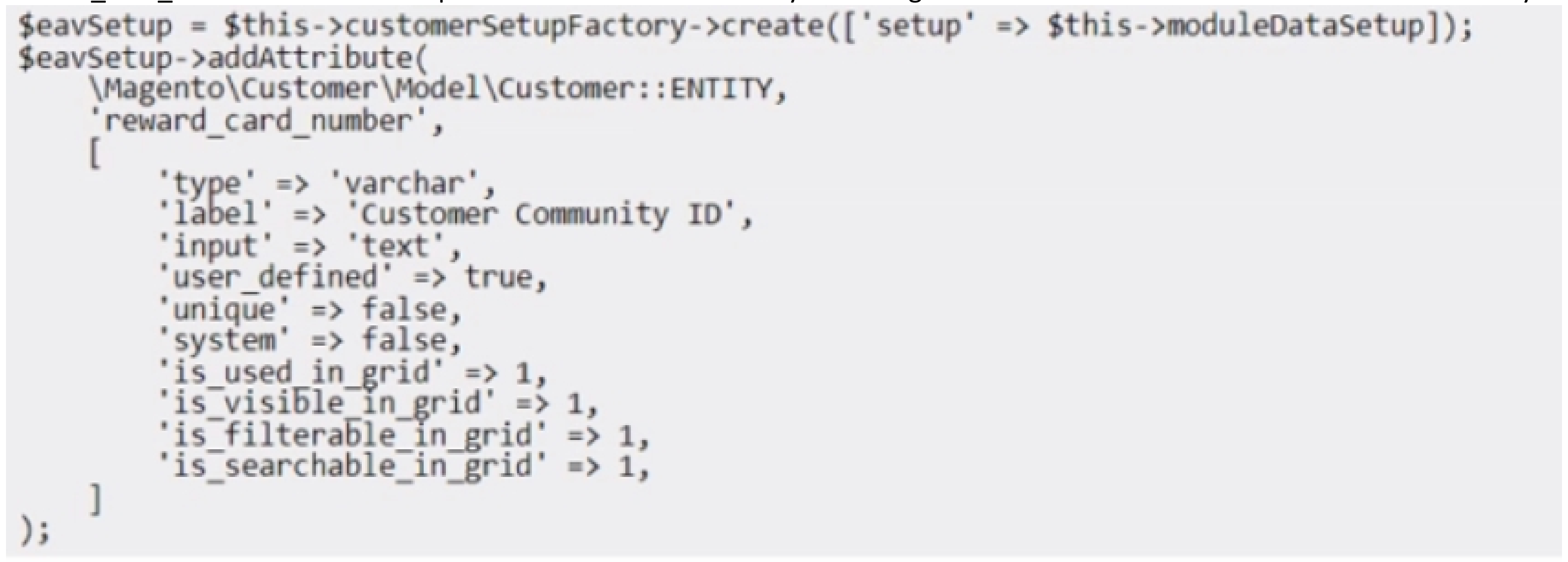
Correct Answer: B
Section:
Explanation:
The issue here is that the gather_reporting_data job is running for a long time and blocking the sitemap_generate job from running. The solution is to create a new cron group for the reporting job and specify <use_separate_process>i</use_separate_process> so that the reporting job runs in a separate process and does not block the sitemap_generate job.Reference: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.html?lang=en#cron-groups-and-processes

QUESTION 32
A developer needs to uninstall two custom modules as well as the database data and schemas. The developer uses the following command: bin/magento module:uninstall Vendor_SampleMinimal Vendor_SampleModifyContent
When the command is run from CLI, the developer fails to remove the database schema and data defined in the module Uninstall class. Which three requirements should the Architect recommend be checked to troubleshoot this issue? (Choose three.)

A. invoked uninstall() and uninstallschema are defined in the Uninstall class
B. invoked unlnstalK) method is implemented in the Uninstall class
C. bin/magento maintenance:enable command should be run in CLI before
D. --remove-data option is specified as an argument for the CLI command
E. --remove-schema and --remove-data options are specified as arguments for the CLI command
F. composer.json file is present and defines the module as a composer package

Correct Answer: B, D, F
Section:

QUESTION 33
A merchant is utilizing an out-of-the-box Adobe Commerce application and asks to add a new reward card functionality for customers. During the code review, the Adobe Commerce Architect notices the reward_card_number attribute setup created for this functionality is causing the customer attribute to be unavailable in the My account/My rewards page template.



```
$eavSetup = $this->customerSetupFactory->create(['setup' => $this->moduleDataSetup]);
$eavSetup->addAttribute(
    \Magento\Customer\Model\Customer::ENTITY,
    'reward_card_number',
    [
        'type' => 'varchar',
        'label' => 'Customer Community ID',
        'input' => 'text',
        'user_defined' => true,
        'unique' => false,
        'system' => false,
        'is_used_in_grid' => 1,
        'is_visible_in_grid' => 1,
        'is_filterable_in_grid' => 1,
        'is_searchable_in_grid' => 1,
    ]
);
```

What should be added to set the customer attribute correctly?

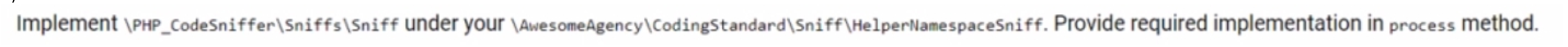
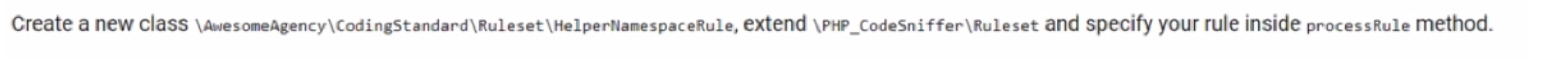
A. group property should be added with a value of 1
B. system property should be added with a value of true
C. scope property should be added with a value of global

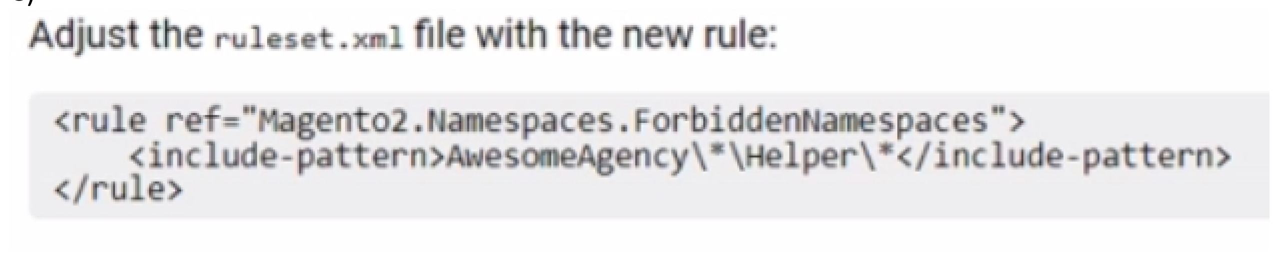
Correct Answer: A
Section:
Explanation:
The group property determines which section of the customer account the attribute belongs to. By setting the group property to 1, the reward_card_number attribute will be assigned to the default group, which is the Account Information section.This will make the attribute available in the My account/My rewards page template.Reference: https://experienceleague.adobe.com/docs/commerce-admin/customers/customer-accounts/attributes/attribute-properties.html?lang=en#group1

QUESTION 34
An Architect needs to review a custom product feed export module that a developer created for a merchant. During final testing before the solution is deployed, the product feed output is verified as correct. All unit and integration tests for code pass.
However, once the solution is deployed to production, the product price values in the feed are incorrect for several products. The products with incorrect data are all currently part of a content staging campaign where their prices have been reduced.
What did the developer do incorrectly that caused the feed output to be incorrect for products in the content staging campaign?

A. The developer retrieved product data directly from the database using the entity_id column rather than a collection or repository.
B. The developer forgot to use the getContentStagingValue method to retrieve the active campaign value of the product data.
C. The developer did not check for an active content staging campaign and emulates the campaign state when retrieving product data.

Correct Answer: C
Section:
Explanation:
Option C is correct because the developer did not check for an active content staging campaign and emulate the campaign state when retrieving product data. Content staging campaigns can modify the product data such as price, name, description, and so on, based on a schedule.To get the correct product data for a specific date and time, the developer needs to use the Magento\Staging\Model\VersionManager class to set the current version ID and emulate the campaign state1. Otherwise, the product data will be retrieved from the default store view without applying the campaign changes.
Option A is incorrect because retrieving product data directly from the database using the entity_id column is not necessarily wrong. It may not be the best practice, but it does not cause the feed output to be incorrect for products in the content staging campaign.The content staging campaigns are stored in separate tables with a version ID that links to the main product table2. The developer can still join these tables and query the product data based on the version ID and date.
Option B is incorrect because there is no such method as getContentStagingValue in Magento 2. The developer cannot use this method to retrieve the active campaign value of the product data. The correct way to get the product data for a specific campaign is to use the Magento\Staging\Model\VersionManager class as mentioned above.
1: Content Staging | Adobe Commerce Developer Guide
2: Content Staging | Adobe Commerce Developer Guide

QUESTION 35
An Architect agrees to improve company coding standards and discourage using Helper classes in the code by introducing a new check with PHPCS.
The Architect creates the following:
* A new composer package under the AwesomeAgency\CodingStandard\ namespace
* The ruleset. xml file extending the Magento 2 Coding Standard
What should the Architect do to implement the new code rule?
A)

Implement `\PHP_CodeSniffer\Sniffs\Sniff` under your `\AwesomeAgency\CodingStandard\Sniff\HelperNamespaceSniff`. Provide required implementation in `process` method.
B)

Create a new class `\AwesomeAgency\CodingStandard\Ruleset\HelperNamespaceRule`, extend `\PHP_CodeSniffer\Ruleset` and specify your rule inside `processRule` method.

C)

Adjust the `ruleset.xml` file with the new rule:

```
<rule ref="Magento2.Namespaces.ForbiddenNamespaces">
    <include-pattern>AwesomeAgency\*\Helper\*</include-pattern>
</rule>
```

A. Option A
B. Option B
C. Option C

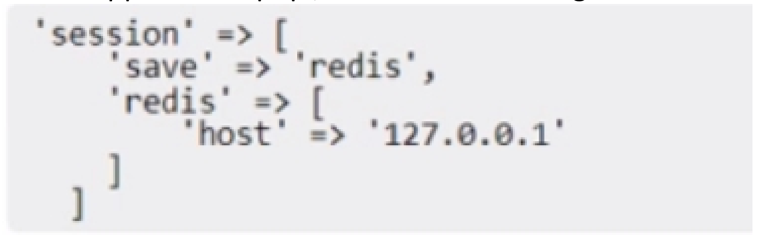
Correct Answer: C
Section:
Explanation:
Option C is correct because adjusting the ruleset.xml file with the new rule is the simplest and most effective way to implement the new code rule. The ruleset.xml file defines the coding standards that are applied by PHP_CodeSniffer. By extending the Magento 2 Coding Standard and adding a new rule, the Architect can customize the code analysis and enforce the company coding standards.The new rule can use the Magento2.Namespaces.ForbiddenNamespaces sniff to check for any usage of Helper classes in the code and report them as errors or warnings1.

Option A is incorrect because creating a new composer package under the AwesomeAgency\CodingStandard\ namespace is not enough to implement the new code rule. The composer package is just a way to distribute and install the coding standard, but it does not define the rules themselves.The Architect still needs to create a ruleset.xml file and register it with PHP_CodeSniffer2.
Option B is incorrect because creating a new class \AwesomeAgency\CodingStandard\Ruleset\ForbiddenNamespaces and specifying the rule inside the process method is unnecessary and complicated. The Architect does not need to create a new class or a new sniff for this rule, as there is already an existing sniff in the Magento 2 Coding Standard that can be used for this purpose.The Magento2.Namespaces.ForbiddenNamespaces sniff can be configured with an include-pattern element to specify which namespaces are forbidden1.
1: Magento 2 Coding Standards | Adobe Commerce Developer Guide
2: How to create a custom coding standard | PHP_CodeSniffer Documentation

QUESTION 36
A merchant notices that product price changes do not update on the storefront.
The index management page in the Adobe Commerce Admin Panel shows the following:
* All indexes are set to 'update by schedule'
* Their status is 'ready'
* There are no items in the backlog
* The indexes were last updated 1 minute ago
A developer verifies that updating and saving product prices adds the relevant product IDs into the catalog_product_price_cl changelog table. Which two steps should the Architect recommend to the developer to resolve this issue? (Choose two.)

A. Reduce the frequency of the cron job to 5 minutes so the items have more time to process.
B. Make sure that no custom or third-party modules modify the changelog and indexing process.
C. Make sure that the version_id for the price indexer in the mview_state table is not higher than the last entry for the same column in the changelog table and re-synchronize.
D. Invalidate the catalog_Product_price indexer in the Adobe Commerce Admin Panel so that it is fully reindexed next time the cron runs.
E. Manually reindex the catalog_product_price index from the command line: bin/magento indexer:reindex catalog_product_price.

Correct Answer: B, C
Section:
Explanation:
The issue here is that the product price changes are not reflected on the storefront, even though the indexes are set to update by schedule and there are no items in the backlog. This indicates that there might be some problem with the changelog and indexing process, which are responsible for tracking and applying the data changes to the index tables. Therefore, the Architect should recommend the developer to check if any custom or third-party modules interfere with the changelog and indexing process, and disable or fix them if needed. Additionally, the Architect should recommend the developer to verify that the version_id for the price indexer in the mview_state table is consistent with the last entry for the same column in the changelog table, and re-synchronize them if they are out of sync.This will ensure that the indexer can process all the data changes correctly and update the index tables accordingly.Reference: https://experienceleague.adobe.com/docs/commerce-admin/systems/tools/index-management.html?lang=en#cron-groups-and-processes1https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#m2devgde-mview2

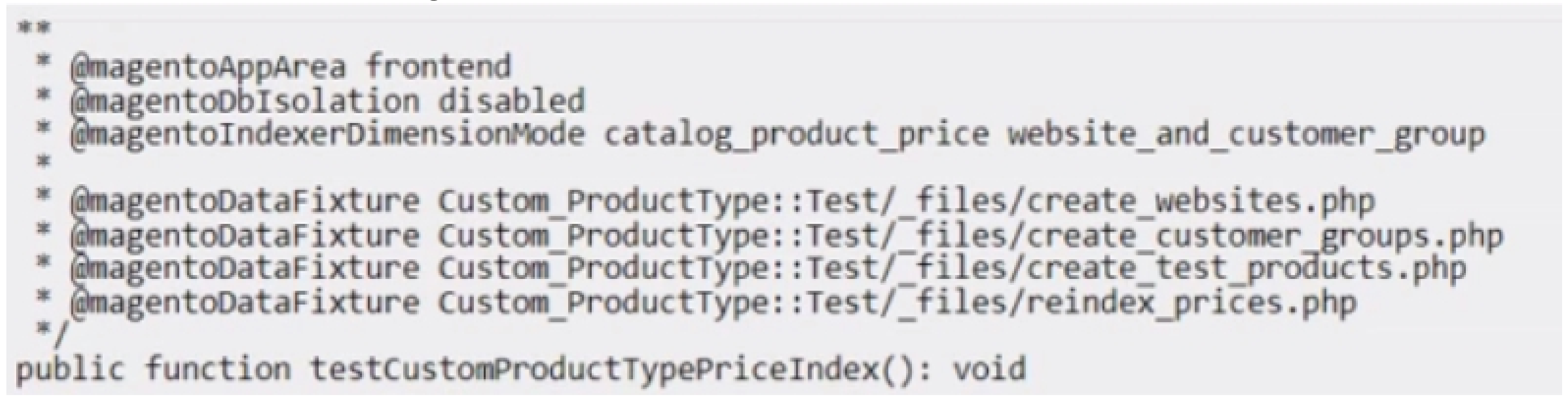
QUESTION 37
The development of an Adobe Commerce website is complete. The website is ready to be rolled out on the production environment.
An Architect designed the system to run in a distributed architecture made up of multiple backend webservers that process requests behind a Load Balancer.
After deploying the system and accessing the website for the first time, users cannot access the Customer Dashboard after logging in. The website keeps redirecting users to the sign-in page even though the users have successfully logged in The Architect determines that the session is not being saved properly.
In the 'app/etc/env.php', the session is configured as follows:

```
'session' => [
    'save' => 'redis',
    'redis' => [
        'host' => '127.0.0.1'
    ]
  ]
```
What should the Architect do to correct this issue?

A. Update the session host value to a shared Redis instance

B. increase the session size with the command config:set system/security/max_session_size_admin
C. Utilize the Remote Storage module to synchronize sessions between the servers

Correct Answer: A
Section:
Explanation:
Option A is correct because updating the session host value to a shared Redis instance in the ''app/etc/env.php'' file will allow the session to be saved properly and prevent users from being redirected to the sign-in page after logging in. Redis is a fast and reliable in-memory data store that can be used for session storage in Magento 2. By using a shared Redis instance, the session data can be accessed by any of the backend web servers behind the load balancer, regardless of which server handled the initial request.This ensures that the user's session is maintained and consistent across different servers1.
Option B is incorrect because increasing the session size with the command config:set system/security/max_session_size_admin will not solve the issue of session not being saved properly. This command only affects the admin session size limit, not the customer session size limit.Moreover, this command does not address the root cause of the issue, which is that the session data is not shared among the backend web servers2.
Option C is incorrect because utilizing the Remote Storage module to synchronize sessions between the servers is not a viable solution for this issue. The Remote Storage module is a feature of Magento Commerce Cloud that allows storing media files and other static content on a remote storage service such as AWS S3 or Azure Blob Storage.This module does not support synchronizing sessions between servers, as sessions are dynamic and transient data that need to be stored in a fast and accessible data store such as Redis3.
1: Use Redis for session storage | Adobe Commerce Developer Guide
2: Security | Adobe Commerce User Guide
3: Remote storage | Adobe Commerce Developer Guide

QUESTION 38
An Adobe Commerce Architect designs and implements functionality that introduces a new Complex Product Type to the existing Adobe Commerce website. Besides visual demonstration of the new product type, the changes include adjustments to the price index.
The website utilizes a multi-dimensional indexer feature to store the price index. The Architect decides to cover it with integration tests. After creating and running one test, the Architect discovers that database storage is not being fully cleaned.
The test method has the following annotation declaration:



```
**
 * @magentoAppArea frontend
 * @magentoDbIsolation disabled
 * @magentoIndexerDimensionMode catalog_product_price website_and_customer_group
 *
 * @magentoDataFixture Custom_ProductType::Test/_files/create_websites.php
 * @magentoDataFixture Custom_ProductType::Test/_files/create_customer_groups.php
 * @magentoDataFixture Custom_ProductType::Test/_files/create_test_products.php
 * @magentoDataFixture Custom_ProductType::Test/_files/reindex_prices.php
 */
public function testCustomProductTypePriceIndex(): void
```

Which adjustment should the Architect make to fix this issue?

A. Add annotation @magentoAppIsolation enabled to method PHPDoc
B. Modify method PHPDoc and change annotation @magentoDbIsolation to enabled
C. Create Customer_ProductType: :Test/_files/{fixture_name)_rollback.php for every fixture

Correct Answer: B
Section:
Explanation:
The issue here is that the database storage is not being fully cleaned after the test is run. The solution is to modify the method PHPDoc and change the annotation @magentoDbIsolation to enabled.This will ensure that the database storage is fully cleaned after the test is run.Reference: https://developer.adobe.com/commerce/testing/guide/integration/#database-isolation1

QUESTION 39
An Adobe Commerce Architect needs to ensure zero downtime during the deployment process of Adobe Commerce on-premises. Which two steps should the Architect follow? (Choose two.)

A. Enable Config flag Under deployement/blue_green/enabled
B. Run bin/magento setup:upgrade --dry-run=true to upgrade database
C. Run bin/magento setup:upgrade - -keep-generated to Upgrade database
D. Run bin/magento setup:upgrad --convert-old-scripts-true to Upgrade database
E. Enable Config flag Under developer/zero_down_time/enabled

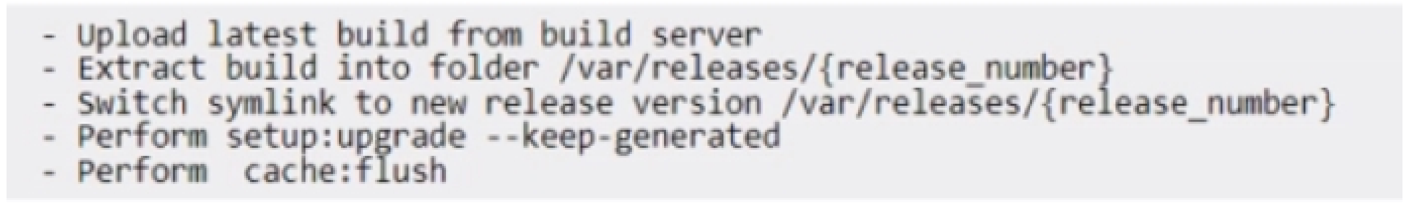
Correct Answer: A, C
Section:
Explanation:
Option A is correct because enabling the config flag under deployment/blue_green/enabled is one of the steps to ensure zero downtime during the deployment process of Magento 2 on-premises. This flag enables the blue-green deployment feature, which allows deploying a new version of the Magento application to a separate environment (blue) without affecting the current live environment (green).Once the new version is ready, the traffic can be switched from green to blue with minimal or no downtime1.
Option C is correct because running bin/magento setup:upgrade --keep-generated is another step to ensure zero downtime during the deployment process of Magento 2 on-premises. This command updates the database schema and data without deleting the generated code and static view files.This way, the Magento application can still serve requests from the cache while the database is being upgraded2.
Option B is incorrect because running bin/magento setup:upgrade --dry-run=true does not upgrade the database, but only checks if there are any errors or conflicts in the database schema or data.This command can be used for testing purposes, but it does not affect the deployment process or the downtime3.
Option D is incorrect because there is no such option as --convert-old-scripts-true for the bin/magento setup:upgrade command. This option does not exist in Magento 2 and does not have any effect on the deployment process or the downtime.
Option E is incorrect because there is no such config flag as developer/zero_down_time/enabled in Magento 2. This flag does not exist in Magento 2 and does not have any effect on the deployment process or the downtime.
1: Blue-green deployment | Adobe Commerce Developer Guide
2: Deploy Magento to production | Adobe Commerce Developer Guide
3: Command-line installation options | Adobe Commerce Developer Guide

QUESTION 40
An Adobe Commerce Architect is working on a scanner that will pull prices from multiple external product feeds. The Architect has a list of vendors and decides to create new config file marketplace.feeds.xml.
Which three steps can the Architect take to ensure validation of the configuration files with unique validation rules for the individual and merged files? (Choose three.)

A. Implement validation rules in the Converter class for the Config Reader
B. Create validation rules in marketplace.schema.xsd.
C. Provide schema to validate a merged file.
D. Add the Uniform Resource Name to the XSD file in the config XML file.
E. Provide schema to validate an individual file.
F. Create a class that implements \Magento\Framework\Config\Datainterface.

Correct Answer: B, C, E
Section:
Explanation:
The Architect can take the following steps to ensure validation of the configuration files with unique validation rules for the individual and merged files:
Create validation rules in marketplace.schema.xsd. This file defines the structure and constraints of the XML elements and attributes for the marketplace.feeds.xml configuration file. The Architect can use this file to specify the required and optional elements, data types, values, and patterns for the configuration file.
Provide schema to validate a merged file. This schema is used to validate the final configuration file that is generated after merging all the individual configuration files from different modules. The Architect can use this schema to check the consistency and completeness of the merged configuration file.
Provide schema to validate an individual file. This schema is used to validate each individual configuration file from each module before merging them. The Architect can use this schema to check the syntax and validity of each configuration file.

QUESTION 41
An Architect is investigating a deployment issue with a server that is configured to work under the symlink directory /var/www/current, which lead to the latest released version of the application.
The deployment process performs the following steps:



```
- Upload latest build from build server
- Extract build into folder /var/releases/{release_number}
- Switch symlink to new release version /var/releases/{release_number}
- Perform setup:upgrade --keep-generated
- Perform  cache:flush
```

After the last deployment, the merchant reported that the Adobe Commerce Import/Export functionality to export Customer Main File data is not working. The Architect discovered that the export file is not shown in the list of generated files.
Which change to the deployment process should be performed to solve this issue?

A. Restart the consumer process during deployment to use the directory with a new application version for export files.
B. Execute Command config:set export/customr/files_directory /var/releases/{release_nunber} to Set the new export path.
C. Doable Crontab before deployment and re-launch after deployment.

Correct Answer: B
Section:
Explanation:
The issue is that the export file is not shown in the list of generated files. This is because the export path is not set correctly. The solution is to execute the command config:set export/customr/files_directory /var/releases/{release_nunber} to set the new export path.This will ensure that the export file is saved in the correct directory and can be accessed from the Admin Panel.Reference:
https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/staging-production.html?lang=en#deploy-to-staging-and-production1

QUESTION 42
A client is migrating to Adobe Commerce Cloud and has approximately 800 existing redirects that must be implemented. The number of redirects cannot be reduced because all redirects are specific, and do not match any pattern.
How should the redirects be configured to ensure performance?

A. Add each redirect in the magento/routes.yaml file.
B. Use VCL snippets to offload the redirect to Fastly.
C. Add each redirect as a URL rewrite via the admin Ul.

Correct Answer: B
Section:
Explanation:
Option B is correct because using VCL snippets to offload the redirect to Fastly is the best way to configure the redirects and ensure performance. VCL snippets are custom code segments that can be added to the Fastly configuration to modify the behavior of the caching service. By using VCL snippets, the redirects can be handled at the edge server level, without reaching the Magento application or the database.This reduces the server load and improves the response time for the redirected requests1.
Option A is incorrect because adding each redirect in the magento/routes.yaml file is not a recommended way to configure the redirects. The magento/routes.yaml file is used to define custom routes for Magento Cloud projects, such as mapping domains or subdomains to environments or services.Adding redirects in this file can cause conflicts with the existing routes and affect the routing logic of the project2.
Option C is incorrect because adding each redirect as a URL rewrite via the admin UI is not an optimal way to configure the redirects. The URL rewrite feature in Magento allows creating custom URLs for products, categories, and CMS pages, and redirecting them to their canonical URLs. However, adding a large number of URL rewrites can increase the database size and affect the performance of the Magento application.Moreover, using the admin UI for this task can be tedious and error-prone3.
1: Custom VCL snippets | Adobe Commerce Developer Guide
2: Configure routes | Adobe Commerce Developer Guide
3: URL Rewrites | Adobe Commerce User Guide

QUESTION 43
An Architect needs to create an additional regional UK website with its own website currency set to GBP in Adobe Commerce. An existing US website is using USD as a default base and website currency.
After the first week of sales in the new UK website, an administrator notices that all sales totals in Sales Orders report show 0.00.
How should this issue be resolved?

A. Configure currency rates for GBP and USD, so they are not empty.
B. Refresh Lifetime Statistics for 'Total Invoiced'.
C. Make sure that orders are shipped and not left in processing state.

Correct Answer: A
Section:
Explanation:
The issue here is that the sales totals in Sales Orders report show 0.00 for the new UK website. This is because the currency rates for GBP and USD are not configured, so the system cannot convert the order amounts from GBP to USD. The solution is to configure the currency rates for GBP and USD, so they are not empty.This will allow the system to calculate the sales totals in USD for the report.Reference: https://experienceleague.adobe.com/docs/commerce-admin/stores-sales/site-store/currency/currency-update.html?lang=en1

QUESTION 44
An Adobe Commerce Architect is troubleshooting an issue on an Adobe Commerce Cloud project that is not yet live.
The developers copied the Staging Database to Production in readiness to Go Live. However, when the developers test their Product Import feature, the new products do not appear on the front end.
The developers suspect the Varnish Cache is not being cleared. Staging seems to work as expected. Production was working before the database migration.
What is the likely cause?

A. The fatly credentials in the Production Database are incorrect.
B. A deployment should have been done on Production to initialize Fatly caching.
C. The site URLs in the Production Database are the URLs of the Staging Instance and must be updated

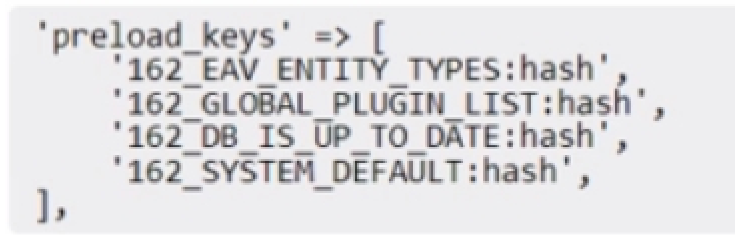
Correct Answer: C
Section:

QUESTION 45
An Adobe Commerce Architect is supporting deployment and building tools for on-premises Adobe Commerce projects. The tool is executing build scripts on a centralized server and using an SSH connection to deploy to project servers.
A client reports that users cannot work with Admin Panel because the site breaks every time they change interface locale.
Considering maintainability, which solution should the Architect implement?

A. Modify project config.php file, configure 'admin_locales_for_deploy' value, and specify all required locales
B. Edit project env.php file, configure 'adminJocales_for_build' value, and specify all required locales
C. Adjust the tools build script and specify required locales during *setup:static-content:deploy' command

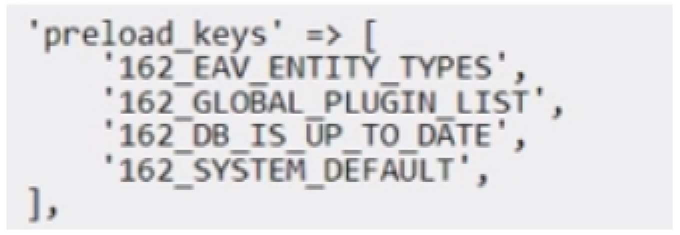
Correct Answer: C
Section:
Explanation:
The issue here is that the site breaks every time the users change interface locale in the Admin Panel. This is because the static content for the different locales is not generated during the deployment process. The solution is to adjust the tools build script and specify required locales during *setup:static-content:deploy' command.This will ensure that the static content for all the needed locales is generated and deployed to the project servers.Reference: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/static-content.html?lang=en#deploy-static-view-files1

QUESTION 46
An Architect is configuring the preload.keys for Redis on an Adobe Commerce on-premise instance.
The Architect discovers that the following cache keys are loaded on each frontend request: EAV_ENTITY_TYPES, GLOBAL_PLUGIN_LIST, DB_IS_UP_TO_DATE, SYSTEM_DEFAULT.
* The id_prefix of the frontend => page_cache is set to 063_.
* The id_pref ix of frontend => default is set to 762_.
* The Architect has enabled and configured Redis L2 caching.
How should the preload.keys be configured?
A)



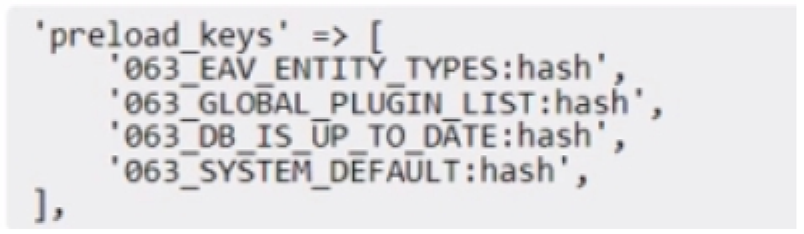
```
'preload_keys' => [
    '162_EAV_ENTITY_TYPES:hash',
    '162_GLOBAL_PLUGIN_LIST:hash',
    '162_DB_IS_UP_TO_DATE:hash',
    '162_SYSTEM_DEFAULT:hash',
],
```

B)



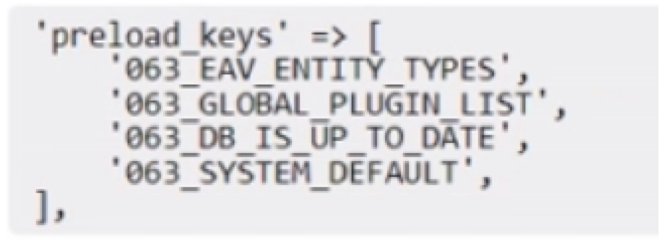
```
'preload_keys' => [
    '162_EAV_ENTITY_TYPES',
    '162_GLOBAL_PLUGIN_LIST',
    '162_DB_IS_UP_TO_DATE',
    '162_SYSTEM_DEFAULT',
],
```

C)



```
'preload_keys' => [
    '063_EAV_ENTITY_TYPES:hash',
    '063_GLOBAL_PLUGIN_LIST:hash',
    '063_DB_IS_UP_TO_DATE:hash',
    '063_SYSTEM_DEFAULT:hash',
],
```

D)



```
'preload_keys' => [
    '063_EAV_ENTITY_TYPES',
    '063_GLOBAL_PLUGIN_LIST',
    '063_DB_IS_UP_TO_DATE',
    '063_SYSTEM_DEFAULT',
],
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: C
Section:
Explanation:
Option C is correct because it configures the preload.keys correctly for Redis L2 caching on an Adobe Commerce on-premise instance. Redis L2 caching is a feature that allows storing the cache data in both Redis and the local file system. This way, the cache data can be loaded faster from the local storage, while Redis acts as a cache invalidation service.To use Redis L2 caching, the backend option for both frontend => page_cache and frontend => default must be set to Magento\Framework\Cache\Backend\RemoteSynchronizedCache1. To enable the preload feature, which reduces the number of requests to Redis, the preload.keys option must be specified with the cache keys that are loaded on each frontend request.However, unlike Redis L1 caching, the preload.keys must include the suffix :hash to indicate that only the hash values of the cache data are stored in Redis2. Therefore, the correct configuration for preload.keys is:
['762_EAV_ENTITY_TYPES:hash', '762_GLOBAL_PLUGIN_LIST:hash', '762_DB_IS_UP_TO_DATE:hash', '762_SYSTEM_DEFAULT:hash',],
Option A is incorrect because it configures the preload.keys incorrectly for Redis L2 caching. It uses the id_prefix of frontend => page_cache (063_) instead of frontend => default (762_) for the cache keys. This will cause a mismatch between the cache keys and the cache data, and result in incorrect or missing cache data.Moreover, it does not include the suffix :hash for the preload.keys, which is required for Redis L2 caching2.
Option B is incorrect because it configures the preload.keys incorrectly for Redis L2 caching.It does not include the suffix :hash for the preload.keys, which is required for Redis L2 caching2. It also uses a wrong cache key (GLOBAL_PLUGIN_LIST) instead of GLOBAL_PLUGIN_LIST.
Option D is incorrect because it configures the preload.keys incorrectly for Redis L2 caching. It uses a wrong id_prefix (162_) instead of frontend => default (762_) for the cache keys. This will cause a mismatch between the cache keys and the cache data, and result in incorrect or missing cache data. It also uses a wrong cache key (EAV_ENTITY_TYPES) instead of EAV_ENTITY_TYPES.
1:Two-level caching | Adobe Commerce Developer Guide

2:Use Redis for default cache | Adobe Commerce Developer Guide

QUESTION 47
An Architect is investigating a merchant's Adobe Commerce production environment where all customer session data is randomly being lost. Customer session data has been configured to be persisted using Redis, as are all caches (except full page cache, which is handled via Varnish).
After an initial review, the Architect is able to replicate the loss of customer session data by flushing the Magento cache storage, either via the Adobe Commerce Admin Panel or running bin/magento cache: flush on the command line. Refreshing all the caches in the Adobe Commerce Admin Panel or running bin/magento cache: clean on the command line does not cause session data to be lost.
What should be the next step?

A. Check app/etc/env.php and make sure that the Redis configuration for caches and session data use different database numbers.
B. Educate the merchant to not flush cache storage and only refresh the caches in future.
C. Set the Stores > Configuration' option for Store Session Data Separately' to 'Yes' in the Adobe Commerce Admin Panel.

Correct Answer: A
Section:
Explanation:
The issue here is that the customer session data is randomly being lost when flushing the Magento cache storage. This is because the Redis configuration for caches and session data might be using the same database number, which causes the session data to be deleted along with the caches. The solution is to check the app/etc/env.php file and make sure that the Redis configuration for caches and session data use different database numbers.This will prevent the session data from being affected by the cache operations.Reference: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/redis.html?lang=en#configure-redis1

QUESTION 48
An Adobe Commerce Architect notices that queue consumers close TCP connections too often on Adobe Commerce Cloud server leading to delays in processing messages.
The Architect needs to make sure that consumers do not terminate after processing available messages in the queue when CRON job is running these consumers.
How should the Architect meet this requirement?

A. Set cohsumers_wait_for_max_MESSAGES variable true in deployment stage.
B. Increase multiple_process limit to spawn more processes for each consumer
C. Change max_messages from 10,000 to 1,000 for CRON_CONSUMERS_RUNNER variable.

Correct Answer: A
Section:
Explanation:
Option A is correct because setting the consumers_wait_for_max_messages variable true in the deployment stage is the best way to meet the requirement. This variable controls whether the queue consumers should wait for a maximum number of messages to process before terminating. If this variable is set to true, the consumers will not terminate after processing the available messages in the queue, but will wait until they reach the max_messages limit or the cron job timeout.This way, the consumers can keep the TCP connections open and avoid delays in processing messages1.
Option B is incorrect because increasing the multiple_process limit to spawn more processes for each consumer will not solve the issue of queue consumers closing TCP connections too often. The multiple_process limit determines how many parallel processes can be run for each consumer. Increasing this limit may improve the throughput of message processing, but it will also consume more server resources and may cause conflicts or errors.Moreover, it will not prevent the consumers from terminating after processing the available messages in the queue2.
Option C is incorrect because changing the max_messages from 10,000 to 1,000 for CRON_CONSUMERS_RUNNER variable will worsen the issue of queue consumers closing TCP connections too often. The max_messages variable defines how many messages each consumer should process before terminating. Decreasing this variable will make the consumers terminate more frequently, which will result in more TCP connections being closed and reopened.This will increase the delays in processing messages3.
1: Configure message queues | Adobe Commerce Developer Guide
2: Configure message queues | Adobe Commerce Developer Guide
3: Configure message queues | Adobe Commerce Developer Guide
Developer mode is the best option for setting up a development environment for testing functionality, not performance, before being passed to the testing team. In developer mode:
Errors are logged and hidden from the user. This ensures that the user does not see any uncaught exceptions or debugging information, but the developers can still access them from the log files.
Cache mode can only be changed from command line. This prevents any accidental or unauthorized changes to the cache settings from the admin panel or other sources.
Static files are created dynamically and then cached. This allows the developers to see the latest changes to the static files without having to run the static content deployment command every time. The static files are also cached for faster loading.

QUESTION 49
An Adobe Commerce Architect is asked by a merchant using B2B features to help with a configuration issue.
The Architect creates a test Company Account and wants to create Approval Rules for orders. The Approval Rules tab does not appear in the Company section in the Customer Account Menu when the Architect logs in using the Company Administrator account.
Which two steps must be taken to fix this issue? (Choose two.)

A. Set 'Enable B2B Quote' in the B2B Admin to TRUE
B. Merchant needs to log out of frontend and then log back in to load new permissions
C. Set 'Enable Purchase Orders' in the B2B Admin to TRUE
D. Set 'Enable Purchase Orders' on the Company Record to TRUE
E. Make sure that the 'Purchase Order' payment method is active

Correct Answer: C, E
Section:
Explanation:
The issue here is that the Approval Rules tab does not appear in the Company section in the Customer Account Menu when the Architect logs in using the Company Administrator account. This is because the Approval Rules feature requires two settings to be enabled: the Purchase Orders feature and the Purchase Order payment method. The solution is to set 'Enable Purchase Orders' in the B2B Admin to TRUE and make sure that the 'Purchase Order' payment method is active. This will allow the Architect to create and manage Approval Rules for orders.

QUESTION 50
An Adobe Commerce Architect is setting up a Development environment for an on-premises project that will be used for developers to specifically test functionality, not performance, before being passed to the Testing team.
The Magento application must run with the following requirements:
1. Errors should be logged and hidden from the user
2. Cache mode can only be changed from Command Line
3. Static files should be created dynamically and then cached
Which Application Mode is required to achieve this?

A. Default Mode
B. Production Mode
C. Developer Mode

Correct Answer: C
Section:
Explanation:
Developer mode is the best option for setting up a development environment for testing functionality, not performance, before being passed to the testing team. In developer mode:
Errors are logged and hidden from the user. This ensures that the user does not see any uncaught exceptions or debugging information, but the developers can still access them from the log files.
Cache mode can only be changed from command line. This prevents any accidental or unauthorized changes to the cache settings from the admin panel or other sources.
Static files are created dynamically and then cached. This allows the developers to see the latest changes to the static files without having to run the static content deployment command every time. The static files are also cached for faster loading.