Exam Code: AD0-E718

Exam Name: Adobe Commerce Architect Master

Website: **www.VCEplus.io - www.VCEup.com**

Question No: 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.

How should the Architect add the taxes for all orders?

A. Write a before plugin to \Magento\Quote\Model\QuoteManagement::placeOrder() and add the custom tax to the quote

B. Declare a new total collector in "etc/sales.xml" in a custom module

C. Add a new observer to the event 'sales_quote_collect_totals_before" and add the custom tax to the quote

Answer: B

Explanation: you can create tax rules in Magento 2 by going to Stores > Taxes > Tax Rules and choosing or adding tax rates. However, this may not be enough for complex tax scenarios that require customization.

https://amasty.com/knowledge-base/how-to-configure-tax-calculation-in-magento-2.html

Question No: 2

An Adobe Commerce Architect needs to log the result of a ServiceClass : : ge-Dara 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 higher than the highest declared plugin sortOrder and implement afterGetData method.

B. Declare a new plugin with the sortOrder value lower than the lowest declared plugin sortOrder and implement aroundGetData method

C. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement aroundGetData method

Answer: C

Explanation:

The aroundGetData method is the best option for this requirement because it provides the flexibility to log the result of the ServiceClass::getData method execution after all plugins have executed. This is because the aroundGetData method is executed before and after the method execution, allowing the Adobe Commerce Architect to log the result of the method execution after all plugins have executed. Reference: https://docs.adobe.com/content/help/en/experience-manager- 65/developing/extending/plugins-and-events/events-and-listeners/lifecycle-events.html

Question No: 3

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 Purchase Orders' 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' on the Company Record to TRUE

D. Make sure that the 'Purchase Order' payment method is active

E. Set 'Enable B2B Quote" in the B2B Admin to TRUE

Answer: A, C

Explanation:

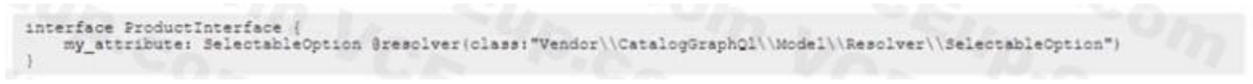
Enabling Purchase Orders at both the B2B Admin and the Company Record levels is necessary for Approval Rules to appear in the Company section of the Customer Account Menu. When 'Enable Purchase Orders' is set to TRUE, the system assumes that the company will be making purchases using purchase orders, and the Approval Rules tab becomes visible.

Question No: 4

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 Productinterface already has the field my_atcribute, but returns an mc 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 ProductInterface field xy_attribute remains int. What prevented the value type of field my_attribute from changing?

A. The fields of ProductInterface are checked during processing schema.graphqls files. If they have a corresponding attribute, then the backendjype of product attribute is set for field type.

B. The interface ProductInterface is already declared in Magento.CatalogGraphQl module. Extending requires use of the keyword -xceni before a new declaration of ProductInterface.

C. The Magento.CatalogGraphQl module occurs later in sequence than the Magento.GraphQl module and merging output of dynamic attributes schema reader overrides types declared in schema.graphqls

Answer: C

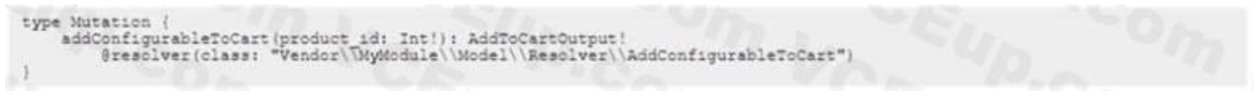
Explanation: products query is a GraphQL query that returns information about products that match specified search criteria. It also shows how to use ProductInterface fields to retrieve product data.

https://devdocs.magento.com/guides/v2.3/graphql/queries/products.html

Question No: 5

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.

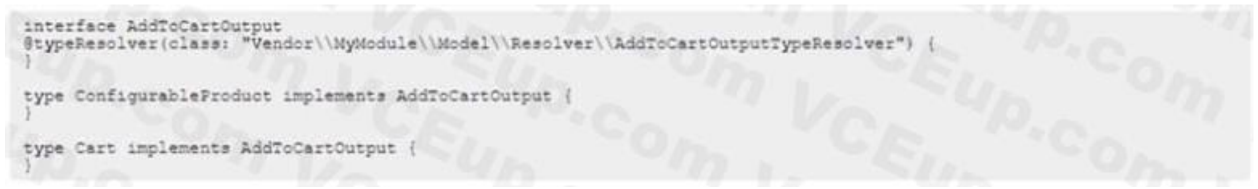
The mutation declaration looks as follows:



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

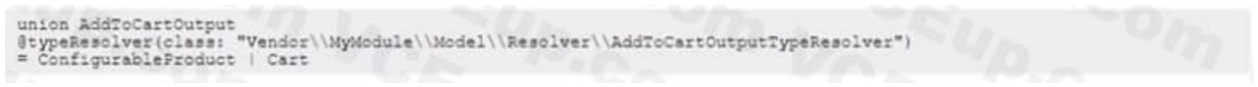
How should the Adobe Commerce Architect declare output of this mutation?

A)



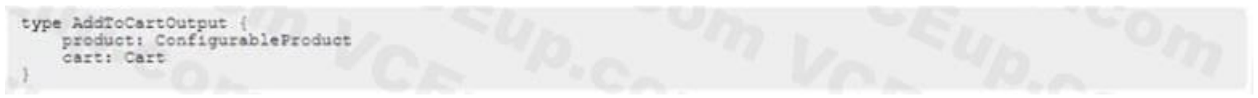
```
interface AddToCartOutput
@typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver") {
}

type ConfigurableProduct implements AddToCartOutput {
}

type Cart implements AddToCartOutput {
}
```

B)



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

C)



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

A. Option A

B. Option B

C. Option C

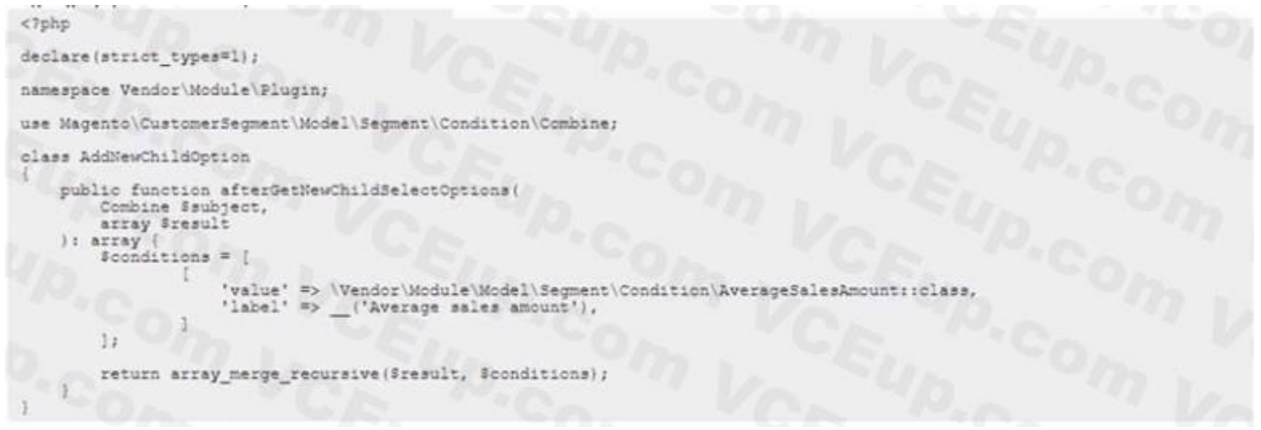
Answer: B

Explanation:

Question No: 6

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:

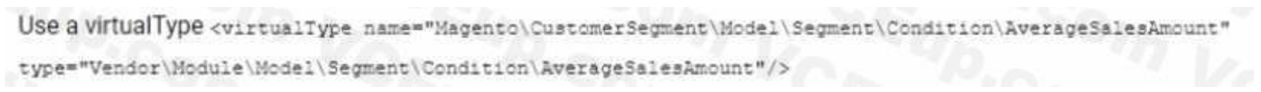


```
<?php
declare(strict_types=1);
namespace Vendor\Module\Plugin;
use Magento\CustomerSegment\Model\Segment\Condition\Combine;
class AddNewChildOption
{
    public function afterGetNewChildSelectOptions(
        Combine $subject,
        array $result
    ): array {
        $conditions = [
            [
                'value' => \Vendor\Module\Model\Segment\Condition\AverageSalesAmount::class,
                'label' => __('Average sales amount'),
            ]
        ];
        return array_merge_recursive($result, $conditions);
    }
}
```

During testing, the following error appears:

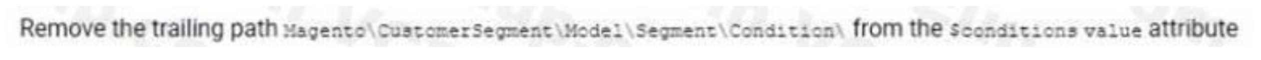
Which two steps should the Architect complete to fix the problem? (Choose two.)

A)



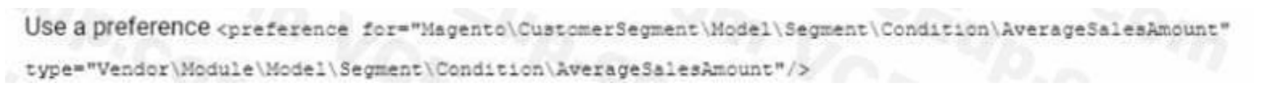
Use a virtualType `<virtualType name="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount" type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>`

B)



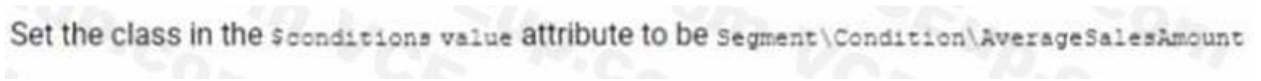
Remove the trailing path `Magento\CustomerSegment\Model\Segment\Condition\` from the `$conditions value` attribute

C)



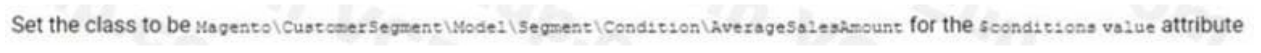
Use a preference `<preference for="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount" type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>`

D)



Set the class in the `$conditions value` attribute to be `Segment\Condition\AverageSalesAmount`

E)



Set the class to be `Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount` for the `$conditions value` attribute

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Answer: B, C

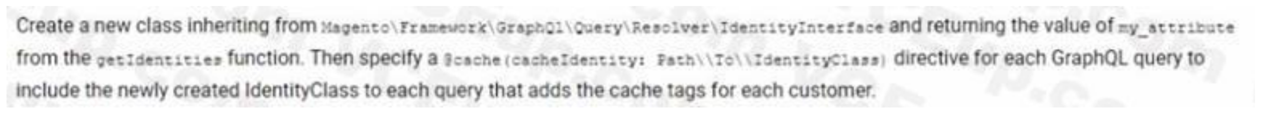
Explanation:

Question No: 7

An Architect working on a headless Adobe Commerce project creates a new customer attribute named my_attribure. 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.

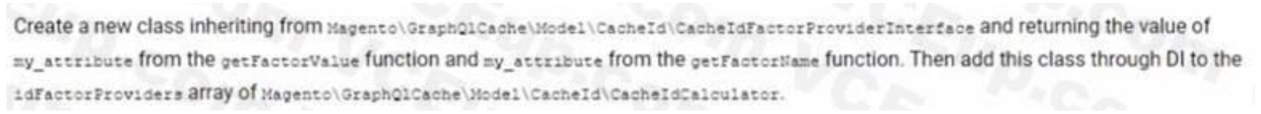
Which steps should the Architect take to make sure the vcl_hash function of Varnish also considers the newly created attribute?

A)



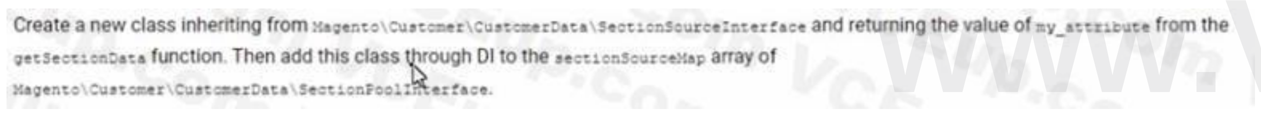
Create a new class inheriting from `Magento\Framework\GraphQl\Query\Resolver\IdentityInterface` and returning the value of `my_attribute` from the `getIdentities` function. Then specify a `@cache(cacheIdentity: Path\\To\\IdentityClass)` directive for each GraphQL query to include the newly created IdentityClass to each query that adds the cache tags for each customer.

B)



Create a new class inheriting from `Magento\GraphQlCache\Model\CacheId\CacheIdFactorProviderInterface` and returning the value of `my_attribute` from the `getFactorValue` function and `my_attribute` from the `getFactorName` function. Then add this class through DI to the `idFactorProviders` array of `Magento\GraphQlCache\Model\CacheId\CacheIdCalculator`.

C)



Create a new class inheriting from `Magento\Customer\CustomerData\SectionSourceInterface` and returning the value of `my_attribute` from the `getSectionData` function. Then add this class through DI to the `sectionSourceMap` array of `Magento\Customer\CustomerData\SectionPoolInterface`.

A. Option A

B. Option B

C. Option C

Answer: C

Explanation:

Question No: 8

A company has an Adobe Commerce store. An attribute named "my.attribute" (type "text") is created to save each product's global ID that is shared between multiple systems.

Several months after going live, the values of "my.attribute" are all integer. This causes a problem for the other systems when those systems receive this data.

An Adobe Commerce Architect needs to recommend a solution to change the type of "my.attribute" from textXo int Which two steps should the Architect take to achieve this? (Choose two.)

A. Migrate data from table "catalog_product_entity_text" to "catalog.producLentityjnt" for the attribute.id

B. Go to Admin > Stores > Attributes > Product, edit "my.attribute" and update type from "text' to "inf

C. Write a plugin for \Magento\Eav\Model\Entity\Attrlbute\Backend\AbstractBackend::afterLoad() and load data from "catalog_product_entity_int"

D. Create a Data Patch and update 'my.attribute' type from "text" to "inf

E. Run the Command bin/magentc indexer: reset catalog_product_attribute

Answer: AD

Explanation:

Option A is correct because it will migrate data from one table to another based on the attribute id, which is required when changing the attribute type14. Option D is correct because it will create a data patch that will update 'my.attribute' type from "text" to "int", which is a recommended way of changing the attribute type programmatically4.

Question No: 9

A merchant is using a unified website that supports native Adobe Commerce B2B and B2C with a single store view.

The merchant wants to show the B2B account features like negotiable quotes and credit limits in the header of the site on every page for the logged-in users who are part of a B2B company account.

Each B2B company has its own individual shared catalog and customer group, and many customer groups for non B2B customers change. The merchant requests that this should not be tied to customer groups.

Which two solutions should the Architect recommend considering public data and caching? (Choose two.)

A. Create a plugin 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. Check if the current user is part of a B2B company within a block class and modify the output accordingly.

C. 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.

D. 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.

E. Create a new HTTP Context variable to allow for separate public content to be cached for users inB2B companies where the output can be modified accordingly.

Answer: C, E

Explanation:

C would involve creating a new custom condition for customer segments that allow for choosing if a user is part of a B2B company, and then use this segment to modify the output accordingly. E would involve creating 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.

Question No: 10

An Architect is reviewing a custom module that is logging customer activity data on storefront using observers. The client reports that logs were recorded while the client was previewing storefront catalog pages from Admin Panel for a future scheduled campaign, using Adobe Commerce staging preview functionality.

What should the Architect check first to address this issue9

A. The plugin for the public method isAllowedObserver() from

\Magento\Staging\Model\Event\Manager that alters the return value

B. The logging observers being copied from etc\events.xml to etc\adminhtml\events.xml with the attribute disabled=''true"

C. The list of logging observers in bannedObservers parameter of

\Magento\staging\Model\Event\Managertype in di.xml

Answer: C

Explanation:

It will allow you to exclude logging observers from being executed during staging preview functionality by adding them to bannedObservers parameter of

\Magento\staging\Model\Event\Manager type in di.xml file. This will prevent customer activity data from being logged while previewing storefront catalog pages from Admin Panel for a future scheduled campaign.

Question No: 11

An Adobe Commerce Architect gets a request to change existing payment gateway functionality by allowing voided transactions only for a certain range of paid amounts.

In the vendor module file etc/config.xml, payment method has an option can,_void set to 1.

How should this customization be done?

A. Extend Magento\Payment\Model\\Method\Adapter and reimplement method void. Use this new class as a new type of payment method facade configuration overriding virtualType type for adapter.

B. Declare a new plugin for class Magento\Payment\ Gateway\Config\ConfigValueHandler and using the afterHandle method, change the result for Subject can_void.

C. Add new handler with name can_void to virtualType based on typeMagento payment\Gateway\config\ValueHandlerPool In payment method facade configuration.

Answer: A

Explanation: payment facade is an instance of Payment Adapter configured with virtual types and allows to process payment actions between Magento Sales Management and payment processor. It also says that you can add dependency injection (DI) configuration for payment method facade in your %Vendor_Module%/etc/di.xml file.

https://devdocs.magento.com/guides/v2.3/payments-integrations/base-integration/facadeconfiguration.html

Question No: 12

An Adobe Commerce Architect needs to customize the workflow of a monthly installments payment extension. The extension is from a partner that is contracted with the default website PSR 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 an IPN. The goal of the IPN 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 without side effects.

Which solution should the Architect implement?

A. Add a plugin before the $invoice-> () and changes 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.zm1

Answer: C

Explanation:

The best solution for the Adobe Commerce Architect to implement in order to capture the requested behavior without side effects is to declare a capture command with type Magento\payment\Gateway\Command\NullCommand for the payment method CommandPool in di.xml. This will allow the partner to initialize the payment and then hand the capture over to the PSP, while also preventing the website from calling the $payment->capture() method. It will also allow the PSP to notify the website through an IPN, which will create an "invoice" and save the 'capture information' to be used later for refund requests through the PSP itself.

Question No: 13

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. Implement _setCachedQuotes () and_getCachedQuotes() return the data if the request matches.

B. In _doShipmentRequest (), call canCollectRates() before sending request to the API

C. Override getResponse (), save the response to a variable, check if the response exists, then return.

Answer: A

Explanation:

Implementing setCachedQuotes () andgetCachedQuotes() in the carrier model can allow the store to store the cart data in a cache, so that repeated requests with the same content can be retrieved from the cache instead of sending a new request to the API. This can reduce the number of requests and avoid hitting the quota limit.

Reference: [1] https://docs.adobe.com/content/help/en/experience-manager-65/commerce/commerce-payment-shipping-modules/shipping/shipping-carrierapis.html [2] https://docs.adobe.com/content/help/en/experience-manager-65/commerce/commerce-payment-shipping-modules/shipping/developing-shipping-carrierintegrations/shipping-carrier-model.html

Question No: 14

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 payment provider Iframe system to isolate content of the embedded frame from the parent web page.

B. Utilize the Advanced Encryption standard (AES-256) algorithm to encrypt all customer-sensitive data from the payment module.

C. Utilize a trusted signed certificate issued by a Certification Authority (CA) to secure each connection made by the payment solution protocol via HTTPS.

Answer: C

Explanation:

The best approach to meet the business needs is to utilize a trusted signed certificate issued by a Certification Authority (CA) to secure each connection made by the payment solution protocol via

HTTPS. This will ensure that the data exchanged between the application, the payment provider, andthe customer is all encrypted and secure. Additionally, this approach will help to reduce the list ofcontrols identified in the Self-Assessment Questionnaire, as it is a secure and approved method ofprotecting customer data.

Question No: 15

An Architect is working to implement Adobe Commerce into a pre-built ecosystem in a company.

Communication between different company domains uses event-driven design and is driven via AMQP protocol with using RabbitMQ.

The Architect needs to establish the data flow between the ERP system and Adobe Commerce.

The ERP system stores only customer data excluding customer addresses.

The role of Adobe Commerce is to provide Customer Address data to the enterprise ecosystem.

Primary Customer data should not be changed from Adobe Commerce side; it should only be updated by messages data from ERP.

Which three AMQP configurations should be considered to meet these requirements? (Choose three.)

A. Create a queue_consumer.xml and communction.xml configuration files for Customer data messages

B. Create a queue_publisher.xml configuration file for Customer data messages

C. Create a nueue_publisher.xml configuration file for Customer Address messages

D. Create a queue_topology.xml configuration file for Customer Address messages

E. Create a queue_topology.xml configuration file for Customer data messages

F. Create a queue_customer.xml and communication.xml configuration files for Customer Address messages

Answer: C, D, F

Explanation:

Based on web searches, it seems that Adobe Commerce uses different XML configuration files to define various aspects of message queues, such as consumers, publishers, and topology123.

According to the documentation3, queue_consumer.xml defines the relationship between an existing queue and its consumer, which is a class that processes messages from a queue. queue_publisher.xml defines the exchange where a topic is published, which is a name that identifies a message for routing. queue_topology.xml defines the message routing rules and declares queues and exchanges.

Based on these definitions, I would say that three possible AMQP configurations that should be considered to meet the requirements are:

1. Create a queue_publisher.xml configuration file for Customer Address messages

2. Create a queue_topology.xml configuration file for Customer Address messages

3. Create a queue_consumer.xml and communication.xml configuration files for Customer Address messages

Question No: 16

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. Write a plugin for \Magento\Framework\Config\Data::get() and read the custom xml files

B. Append the custom xml file name in "Magento\Config\Model\Config\Structure\Reader" in di.xml

C. Create a Data class that implements "\Magento\Framework\Config\Data'

D. Inject a "reader" dependency for "Magento\Framework\Config\Data" in di.xml

E. Make a Reader class that implements "\Magento\Framework\Config\Reader\Filesystem"

Answer: C, E

Explanation:

Based on web searches, it seems that Magento uses different classes and interfaces to interact with configuration files, such as Data, Reader, and Converter12.

According to the documentation1, Data is a class that provides access to configuration data using a scope. Reader is an interface that reads configuration data from XML files. Converter is an interface that converts XML data into an array representation.

Based on these definitions, I would say that two possible steps that the Architect should make to meet the requirement are:

1. Create a Data class that implements "\Magento\Framework\Config\Data"

2. Make a Reader class that implements "\Magento\Framework\Config\Reader\Filesystem" These steps would allow the custom module to read all the XML configuration files declared in the system, merge them together, and use their values in PHP class.

Question No: 17

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 /vi/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 OAuth-based authentication to provide access to system resources. Integration will be registered by the merchant in the panel an OAuth handshake during activation. The third-party system should follow OAuth protocol to authorize.

C. 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.

Answer: B

Explanation:

According to the documentation1, token-based authentication is a simple way to access resources using an access token that is generated when an integration is activated. OAuth-based authentication is a more secure way to access resources using a consumer key, consumer secret, access token, and access token secret that are generated when an integration is registered and authorized.

Based on these definitions, I would say that the type of authentication that should be used and implemented in a third-party system for this integration is:

1. Use OAuth-based authentication to provide access to system resources. Integration will be registered by the merchant in the panel an OAuth handshake during activation. The third-party system should follow OAuth protocol to authorize.

Question No: 18

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 developments are valid when reviewing the implementation? (Choose three.)

A. Content of the etc/product_types.xml file

B. Hydrator for attributes belonging to the new product type

C. Custom type model extended from the abstract Product Type model

D. A new class with custom pricing logic, extending the abstract Product model class

E. Data patch to register the new product type

F. New price model extending \Magento\Catalog\Model\Product\Type\Price

Answer: ACF

Explanation:

According to some tutorials45, creating a custom product type in Adobe Commerce involves several steps, such as:

Creating a product_types.xml file in etc folder to declare the new product type

Creating a custom type model that extends from an abstract product type model

Creating a custom price model that extends from an abstract price model

Creating a layout file for the new product type

Creating a data patch to register the new product type

Based on these steps, I would say that three possible developments that are valid when reviewing the implementation are:

1. Content of the etc/product_types.xml file

2. Custom type model extended from the abstract Product Type model

3. New price model extending \Magento\Catalog\Model\Product\Type\Price

These developments would allow creating a new product type with its own custom pricing logic and attributes.

Question No: 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_form.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 also be added to view/adminhtml/ul_component/catalogstaging_category_update_form.xml.

C. The attribute must have<item name=''allow_staging'' xsi:type="boolean''>true<item> set in the =category_form.xml file under the attributes config" section.

Answer: B

Explanation:

This is because, in order to change the attribute when scheduling new category updates, the attribute must be added to the view/adminhtml/ulcomponent/catalogstagingcategoryupdateform.xml file in order to be displayed in the additional form when scheduling updates. This additional form is used to set the values for the category attributes when scheduling updates.

Question No: 20

An Adobe Commerce store owner sets up a custom customer attribute "my.attribute" (type int).

An Architect needs to display customer-specific content on the home page to Customers with "my.attribute" greater than 3. The website is running Full Page Cache.

Using best practices, which two steps should the Architect take to implement these requirements?

(Choose two.)

A. Use customer-data JS library to retrieve "my.attribute" value

B. Add a new context value of "my.attribute" to Magento\Framework\App\Http\Context

C. Add a custom block and a phtml template with the content to the cmsjndexjndex.xml layout

D. Create a Customer Segment and use "my.attribute" in the conditions

E. Add a dynamic block with the content to the Home Page

Answer: A, C

Explanation:

https://docs.magento.com/user-guide/v2.3/stores/attributes-customer.htmldisplaying custom customer attributes on cached pages using best practices involves several steps,such as:

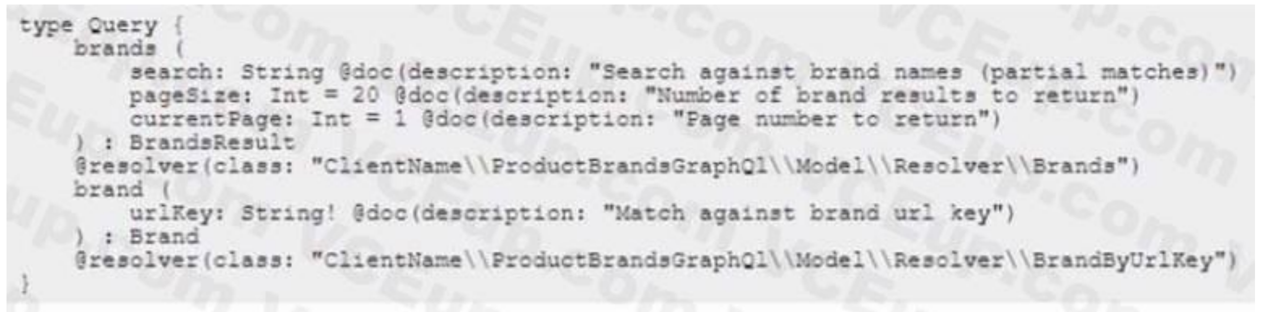
Creating a custom block and a phtml template with the content to display Adding the custom block to the layout file of the page where it should appear Creating a section.xml file to declare a new section for the custom attribute Creating a plugin for Magento\Customer\CustomerData\SectionPoolInterface to add the custom attribute value to the section data Using customer-data JS library to retrieve and display the custom attribute value in the phtml template

Question No: 21

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. Each GraphQL query's resolver class should inject MagentoGraphQlCacheModelCacheableQuery and call setCacheValidity (true) on it as part of the resolver's rescive function

B. Specify a @cache (cacheable: false directive for each GraphQL query, making sure that the data returned is not cached, and is up to date

C. Specify a @cache (cacheIdentity: path\\T\\identityClass) directive for each GraphQL query, corresponding to a class that adds cache tags for relevant brands and associated products

Answer: C

Explanation:

This option would allow caching GraphQL query results using Varnish or Fastly with proper invalidation and differentiation mechanisms.

Question No: 22

An Adobe Commerce Architect needs to scope a bespoke news section for a merchant's Adobe Commerce storefront. The merchant's SEO agency requests that the following URL structure: news/{date}/{article_url_key}l 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.

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.)

A. Create a standard controller route and an Index/Index index controller class that loads the relevant news article by matching the URL date and URL key parts.

B. Create an observer that listens to the controllers_front_send_response_before event, looks for the mm portion of the URL, and If it matches, loads the relevant news article by matching the URL date and URL key parts.

C. Create a plugin that intercepts lu^jentoXFraBeworkUppXActien::executed, 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.

D. Create a standard controller route and mapping the internal URLs (such as news/article/view/id/1) to rewrites that are generated on save and then stored in the URL rewrites table.

E. 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.

Answer: AE

Explanation: creating a custom router involves several steps, such as:

Creating a routes.xml file to declare a custom route ID and front name Creating a Router.php file to define the custom router class that extends Magento\Framework\App\Router\Base

Creating an etc/di.xml file to register the custom router class with a specific sortOrder

Creating controller classes and action methods to handle the requests

Based on these steps, I would say that two possible options that the Architect should consider to meet these requirements are:

1. Create a standard controller route and an Index/Index index controller class that loads the relevant news article by matching the URL date and URL key parts.

2. 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.

Question No: 23

An Adobe Commerce Architect is working on a sales campaign to present a new product on the site that allows the purchase of a pre-defined set of products with a discount. Each product in the set should have a separate stock and tax class.

One requirement is to use a third-party system to build reports with REST API to fetch the following data:

• SKU

• Qty

• Original price

• Sales price

• Tax amount

Which solution should the Architect use to meet these requirements?

A.

• Create Fixed Bundle Product for gathering simple products;

• Manage price for every selected option;

• Add extension attribute original_simple_price for

\Magento\Sales\Api\Data\OrderItemExtensionInterface and populate value with price of simple product;

B.

• Create Dynamic Bundle Product for gathering simple products;

• Utilize Content Staging to manage special prices for bundle products on time for the campaign;

• Expose required data via Adobe Commerce Order API;

C.

• Create Grouped Product and Create after plugin on

\Magento\GroupedProduct\Model\Product\Type\Grouped:preparedForCarrAdvanced for bunch products ordering;

• Utilize Content Staging to manage special prices on time for the campaign for simple products;

• Expose required data via Adobe Commerce Order API;

Answer: B

Explanation:

A bundle product is a customizable product that consists of several options, each based on a simple or virtual product. A grouped product is a collection of simple products that are presented as a group.

According to some tutorials , creating a bundle product in Adobe Commerce involves several steps, such as:

Choosing the bundle product template and attribute set

Completing the required settings, such as name, SKU, price, and weight

Configuring the basic settings, such as status, visibility, and categories

Adding the bundle options and associated products

Adding optional product information, such as images and meta data

Posting the product

Content staging is a feature that allows creating, previewing, and scheduling content updates for your store directly from the Admin . You can use content staging to create campaigns that include changes to products, categories, pages, blocks, widgets, price rules, and more.

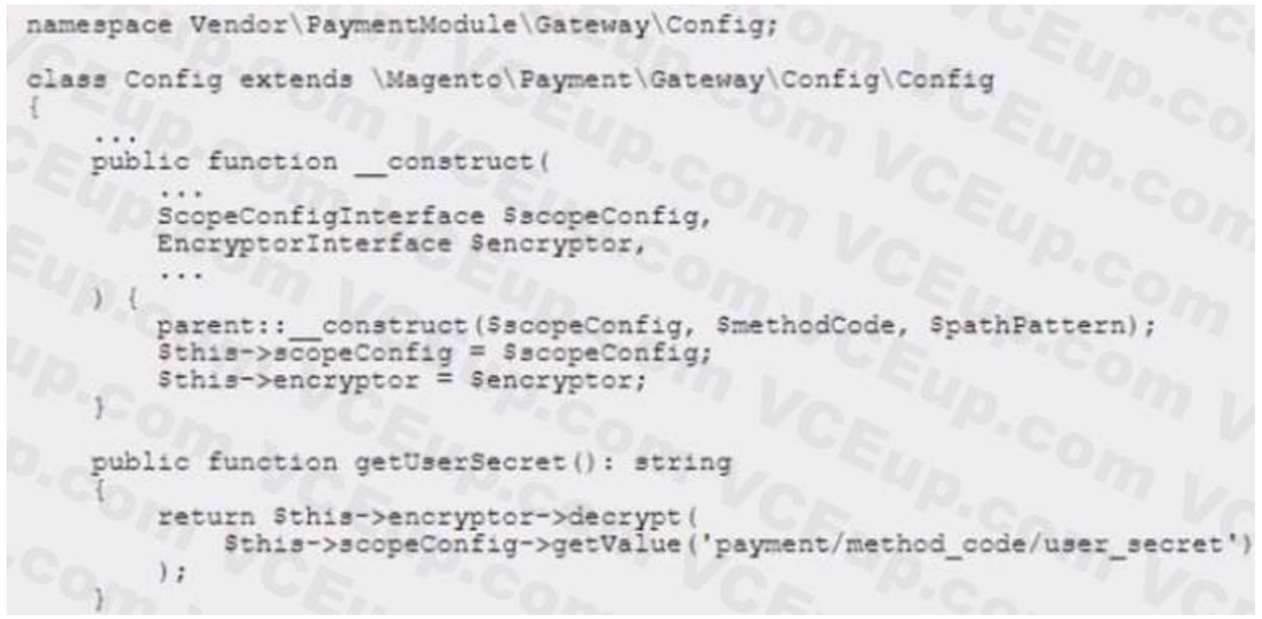
Based on these steps and features, I would say that one possible solution that the Architect should use to meet these requirements is:

B. Create Dynamic Bundle Product for gathering simple products; Utilize Content Staging to manage special prices for bundle products on time for the campaign; Expose required data via Adobe Commerce Order API; This solution would allow creating a new product that allows the purchase of a pre-defined set of products with a discount. Each product in the set would have a separate stock and tax class. The special prices for bundle products could be managed using content staging. The required data could be exposed via Adobe Commerce Order API.

Question No: 24

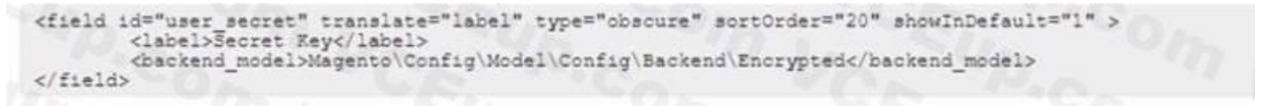
While reviewing a newly developed pull request that refactors multiple custom payment methods, the Architect notices multiple classes that depend on

\Magento\Framework\Encryption\EncryptorInterf ace 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')
        );
    }
```

In each module, the user_secret config is declared as follows:



```
<field id="user_secret" translate="label" type="obscure" sortOrder="20" showInDefault="1" >
      <label>Secret Key</label>
      <backend_model>Magento\Config\Model\Config\Backend\Encrypted</backend_model>
</field>
```

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. Replace all Vendor\PaymentModule\Gateway\Config\Config Classes With virtualTyp- Of Magento\Payxer.t\Gateway\Conflg\Config and Set <user_secret backend_Model="Magento\Config\Model\Config\Backend\Encrypted" /> under ccnfig.xml

B. Add a plugin after the getvalue method of $sccpeConfig, remove the $encryptor from dependency and use it in the plugin to decrypt the value if the config name is 'user.secret?

C. Create a common config service class vendor\Payment\Gateway\config\Config under Vendor.Payment and use it as a parent class for all of the Vender

\EaymentModule\Gateway\Config\Config Classes and remove $sccpeConfig and $encryptor dependencies

Answer: C

Explanation:

Question No: 25

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. Invalidate the catalog_product_price indexer in the Adobe Commerce Admin Panel so that it is fully reindexed next time the cron runs.

B. Manually reindex the catalog_product_price index from the Command line:bin\magentor indexer:reindex catalog_product_price.

C. Make sure that no custom or third-party modules modify the changelog and indexing process.

D. 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.

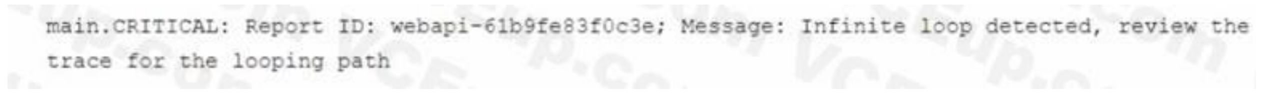
E. Reduce the frequency of the cron job to 5 minutes so the items have more time to process.

Answer: AB

Explanation:

Question No: 26

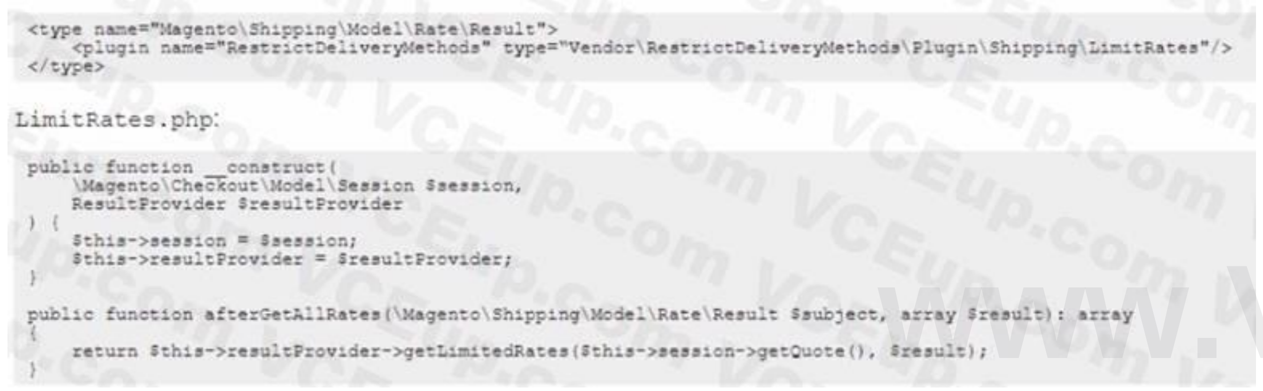
Since the last production deployment, customers can not complete checkout. The error logs show the following message multiple times:



```
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/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. Inject an instance of \Magentro\Quote\API\CartRepostoryInterface and receive cart instance via$this->cartRepository->get($this-session->getQucteId())

B. Replace the injected dependency

\Magento\Checkout\Model\Session\With\Magento\Framework\Session\SessionManagerInterface

C. Change 'after' plugin with 'around' plugin. The issue is being caused by calling the result provider code after the code of the original method.

Answer: C

Explanation:

Question No: 27

An Adobe Commerce system is configured to run in a multi-tier architecture that includes:

• A cache server with Varnish installed

• A backend web server with Adobe Commerce installed

• A database server with MySQL installed

When an Adobe Commerce Architect tries to clean the cache from the Store Admin by using the "Flush Magento Cache" in Cache Management, the Full Page Cache does not clear.

Which two steps should the Architect take to make the Full Page Cache work properly? (Choose two.)

A. Set the backend destination host to the frontend server's address in the Store Admin Stores > Configuration > Advanced > System > Full Page Cache > Varnish Configuration > Backend Host

B. Set the backend port destination to the frontend server's Varnish port in the Store Admin Stores > Configuration > Advanced > System > Full Page Cache > Varnish Configuration > Backend Port

C. Set the cache type to "Varnish Caching" in the Store Admin.

Stores > Configuration > Advanced > System > Full Page Cache > Caching Application

D. Use "Flush Cache Storage" instead of "Flush Magento Cache"

E. Set the cache destination host using magento CLI bin/magento setup:config:set --http-cachehosts<cache_server>:<varrnish port>

Answer: A, B

Explanation:

Question No: 28

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 > Ccr.f iguraticr. > Advanced > developer > Grid Settings > Asynchronous indexing) can be enabled by executing the following CLI command: tm/magento config:set dev/grid/async_ini*xmg 1

B. Multithreaded checkout processing admin panel setting (stores > s ettmgs > Configuration > Sales

> Checkout > General Settings > Asynchronous) can be set to a higher value representing the number of PHP threads used exclusively for checkout

C. Asynchronous email notifications admin panel Setting (stores > Settings > Configuration > Sales > Sales Emails > General Settings > Asynchronous) can be enabled

D. A new database can be created and the Split Database feature can be automatically configured with the following command: bin/Magento setup:db-schema:split-sales-sales --host="<checkout db host or ip>" —dbname="<name>" — username""<checkout db username>'' —password=" <password>"

E. The website deploy mode can be set to si-g- 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

Answer: AB

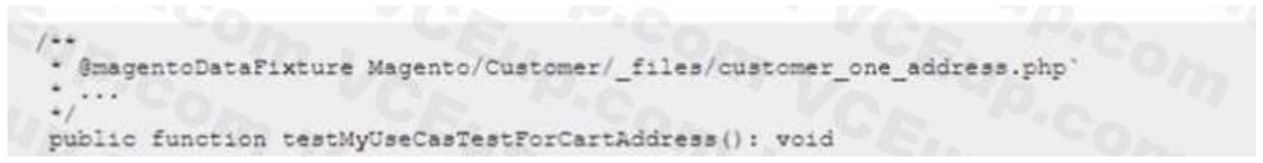
Explanation:

To minimize the impact on checkout performance due to a large number of simultaneously placed orders, an Architect can optimize two website settings: Asynchronous indexing admin panel setting (A) and Multithreaded checkout processing admin panel setting (B). Enabling asynchronous indexing admin panel setting can be done by executing the command tm/magento config:set dev/grid/async_ini*xmg 1, while the multithreaded checkout processing admin panel setting can be set to a higher value representing the number of PHP threads used exclusively for checkout. It is important to note that the website deploy mode should not be set to siege mode and should instead be changed back to production as soon as the number of simultaneously placed orders decreases to acceptable levels.

Question No: 29

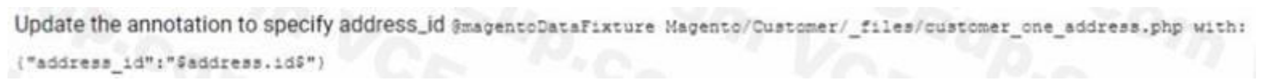
An Adobe Commerce Architect is reviewing api-functional test code. Some tests send errors to indicate that the customer address does not exist.

The test codes show the following:



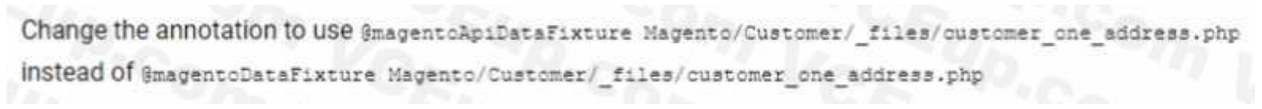
```
/**
 * @magentoDataFixture Magento/Customer/_files/customer_one_address.php`
 * ...
 */
public function testMyUseCasTestForCartAddress(): void
```

A)



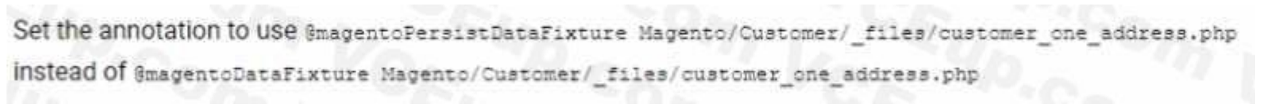
Update the annotation to specify address_id `@magentoDataFixture Magento/Customer/_files/customer_one_address.php with:`
`{"address_id":"$address.id$"}`

B)



Change the annotation to use `@magentoApiDataFixture Magento/Customer/_files/customer_one_address.php` instead of `@magentoDataFixture Magento/Customer/_files/customer_one_address.php`

C)



Set the annotation to use `@magentoPersistDataFixture Magento/Customer/_files/customer_one_address.php` instead of `@magentoDataFixture Magento/Customer/_files/customer_one_address.php`

A. Option A

B. Option B

C. Option C

Answer: C

Explanation:

Question No: 30

An Adobe Commerce Architect creates a new functionality called Customs Fee, which adds a new total that applies to additional costs for handling customs clearance expenses. The extension allows specifying fee value for every website separately via the Adobe Commerce Configuration System.

The Architect plans to cover new functionality with integration tests. One test case needs to confirm if the total is calculated correctly on different websites.

How should the Architect make sure that test configuration data is added to test methods according to best practices?

A. Override setuo () method, receive instance of \Magento\TestFramework\App\config, and specify value via setValue () method

B. Specify @magentoconfigFixture annotations for the test methods in PHPDoc

C. Create a fixture file to configure Adobe Commerce and specify it in test method PHPDoc using the @magentoconfigFixture annotation

Answer: B

Explanation:

The best practice for making sure that test configuration data is added to test methods is to use the @magentoconfigFixture annotation in the PHPDoc for the test methods. This will allow the Architect to specify a fixture file which will configure Adobe Commerce, and the test method will then be able to access these configuration values. Additionally, the Architect can also override the setUp() method and receive an instance of \Magento\TestFramework\App\config and specify the value via the setValue() method.

Question No: 31

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.

Sthis->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 one test class with one test method. Use the \Magento\testFramework\ store\Executionstorecontext class once in the fixture and another time in the test.

B. Create two test Classes With one test method each. Use the @magentoExecuteInStoreContext 3 and @magentoExecuteInStoreContext 4 annotations on the class level.

C. Create one test class with two test methods. Use the @magentoStoreContext 3 annotation in one method and @magentoStoreContext 4 in the other one.

Answer: B

Explanation:

The best approach for the Architect to meet the requirements is Option B. Create two test Classes With one test method each. Use the @magentoExecuteInStoreContext 3 and @magentoExecuteInStoreContext 4 annotations on the class level. This will ensure that the fixture is executed in the context of Store View 3, and the tested logic is executed in the context of Store View 4. This approach allows for more granular control of the store views and reduces the complexity of the test.

Question No: 32

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. Enable price index customer group merging for products without tier prices

B. Set Customer Share Customer Accounts Option to Global

C. Edit customer groups to exclude websites that they are not using

D. Set MaGE_INDEXER_THREADS_COUNT environment variable to enable parallel mode

E. Move catalog price_index indexer to another custom indexer group

Answer: A, D

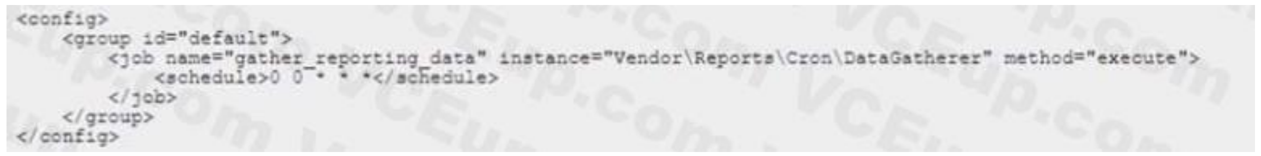
Explanation:

The two best ways the Architect can shorten the full price index execution time are Option A. Enable price index customer group merging for products without tier prices, and Option D. Set MaGEINDEXERTHREADS_COUNT environment variable to enable parallel mode. Enabling customer group merging will help reduce the number of customer groups that need to be processed, while setting the environment variable will allow the indexer to use multiple threads and run in parallel mode, thus reducing the overall execution time.

Question No: 33

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 site's sitemap. xml file has not been updated since the new job was added.

What should be done to fix this issue?

A. Break the data gathering job into a number of smaller jobs, so that each individual job runs for a maximum of 5 minutes.

B. Create a new cron group for the reporting job. Specifying

<use_separate_process>1/use_separate_process>

C. Change the schedule of the sitemap_generate cron job to 30 o * * * so that it runs after the aacher_reporcmg_data job has completed.

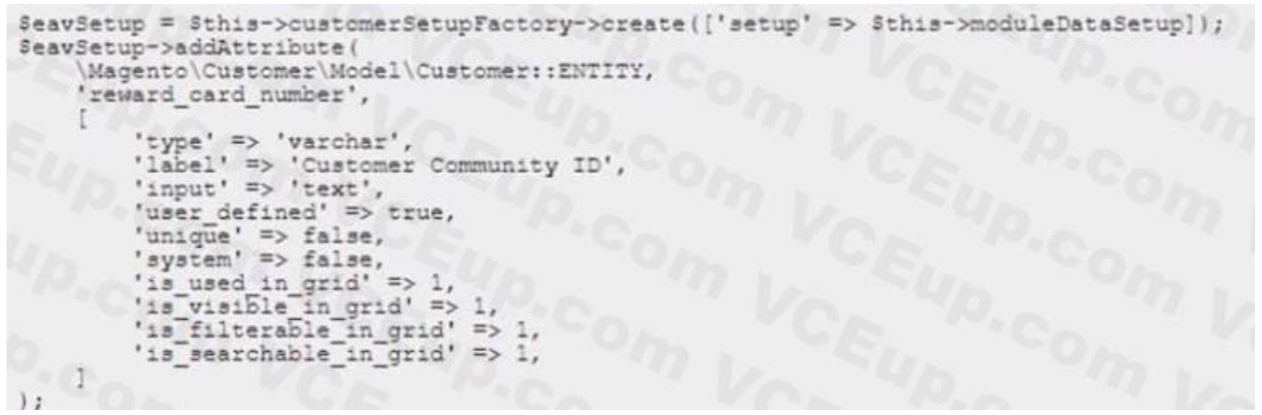
Answer: B

Explanation:

This will ensure that the reporting job runs in its own process, separate from other cron jobs, and will not interfere with the sitemapgenerate cron job. This will ensure that the sitemapgenerate cron job runs as soon as the reporting job is finished, ensuring that the sitemap.xml is always up to date.

Question No: 34

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. scope property should be added with a value of global

B. group property should be added with a value of 1

C. system property should be added with a value of true

Answer: A

Explanation:

Question No: 35

An Architect agrees to improve company coding standards and discourage using Helper classes in the code by introducing a new check with PHPCS.

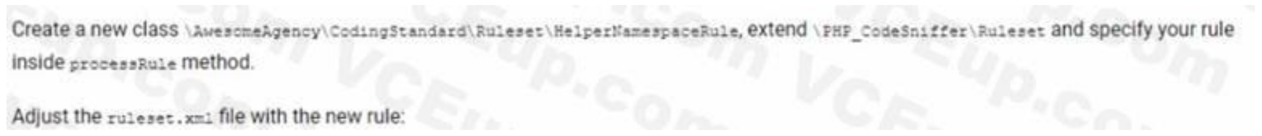
The Architect creates the following:

• 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?

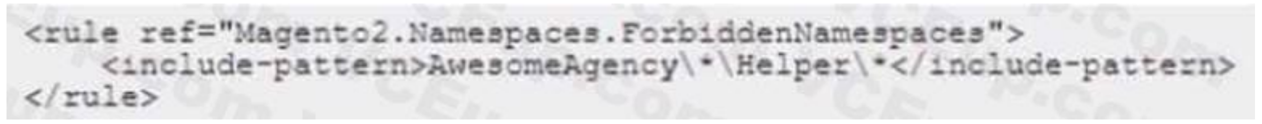
A)



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

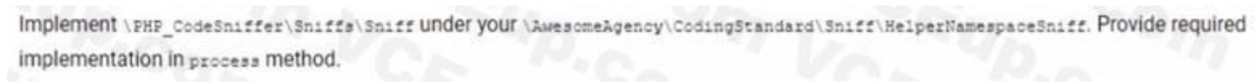
Adjust the `ruleset.xml` file with the new rule:

B)



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

C)



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

A. Option A

B. Option B

C. Option C

Answer: B

Explanation:

Question No: 36

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. remove-schema and --remove-data options are specified as arguments for the CLI command

B. bin/magento maintenance: enable command should be run in CLI before

C. composer.json file is present and defines the module as a composer package

D. Invoke uninstallData() and uninstallSchema () are defined in the Uninstall class

E. --remove-data option is specified as an argument for the CLI command

F. invoked uninstall () method is implemented in the Uninstall class

Answer: ADF

Explanation:

To troubleshoot the issue, the Architect should check that the remove-schema and --remove-data options are specified as arguments for the CLI command, that the Uninstall class defines the uninstallData() and uninstallSchema() methods, and that the invoked uninstall() method is implemented in the Uninstall class.

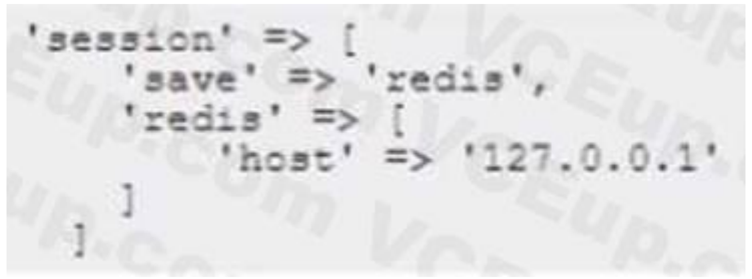
Question No: 37

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 napp/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. Utilize the Remote Storage module to synchronize sessions between the servers

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

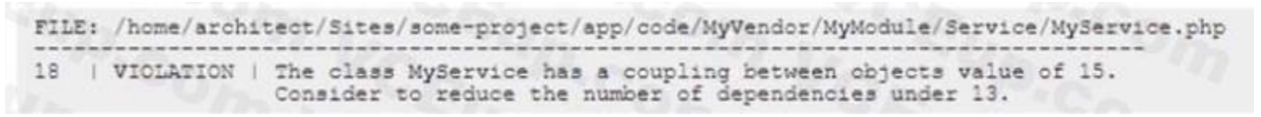
C. Increase the session size with the command config:set system/security/max_session_size_admin

Answer: B

Explanation:

Question No: 38

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 Kyservice.

How should the Architect fix the code so that it complies with the coding standard rule?

A. Introduce a new class accepting those five scalars and use it in the constructor and the remaining logic of Myservice

B. Modify the code of Myservice so the number of different classes, interfaces, and scalar types used as parameters in the constructor and other methods is less than 13

C. Modify the code of Myservice so that the number of different classes and interfaces referenced anywhere inside the class is less than 13

Answer: A

Explanation:

The best way to fix the code so that it complies with the coding standard rule is to introduce a new class accepting those five scalars and use it in the constructor and the remaining logic of Myservice.

This will reduce the number of different classes, interfaces, and scalar types used as parameters in the constructor and other methods to less than 13, which is the limit set by the coding standard.

Additionally, any extra code that is not necessary can be removed to reduce the general complexity of the class and improve readability.

Question No: 39

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 forgot to use the getContentStagingValue() method to retrieve the active campaign value of the product data

B. The developer retrieved product data directly from the database using the entity_id column rather than a collection or repository.

C. The developer did not check for an active content staging campaign and emulates the campaign state when retrieving product data.

Answer: A

Explanation:

According to the Adobe Commerce documentation, when retrieving product data, developers must use the getContentStagingValue() method to ensure that the active campaign value for the product is retrieved. This method takes into account any content staging campaigns that might be active, and returns the appropriate value from the content staging campaign rather than the default value from the database. Failing to use this method can result in incorrect data being returned in the product feed.

Question No: 40

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. Run bin/magento setup:upgrade --keep-generated to Upgrade database

B. Run bin/magento setup:upgrade —dry-run=true to upgrade database

C. Rim bin/magento setup:upgrade --convert-old-scripts=true to Upgrade database

D. Enable config flag under developer/zere _down_time/enabled

E. Enable config flag under deployment/blue_ green/enabled

Answer: DE

Explanation:

In order to ensure zero downtime during the deployment process of Adobe Commerce on-premises, the Architect should enable the config flags under developer/zerodowntime/enabled and deployment/bluegreen/enabled. The zerodowntime/enabled flag will allow for the deployment process to be completed without the need for any downtime, and the bluegreen/enabled flag will enable the blue-green deployment strategy, which allows for a seamless transition between the two deployments. Additionally, the Architect should run bin/magento setup:upgrade to upgrade the database.

Question No: 41

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. Use VCL snippets to offload the redirect to Fastly.

B. Add each redirect in the .magento/routes.yaml file.

C. Add each redirect as a URL rewrite via the admin UI.

Answer: A

Explanation:

The best option for configuring the redirects is to use VCL snippets to offload the redirects to Fastly.

This is a Content Delivery Network (CDN) that can handle large numbers of requests quickly and efficiently, ensuring that your redirects will be processed quickly and reliably. Furthermore, VCL snippets are easy to set up and can be reused for other redirects, making them an efficient and costeffective solution for managing large numbers of redirects.

Question No: 42

An Adobe Commerce Architect is troubleshooting an issue on an Adobe Commerce Cloud project that is not yet live.

The developers migrate 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 frontend.

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 Fastly credentials in the Production Database are incorrect.

B. A deployment should have been done on Production to initialize Fastly caching.

C. The site URLs in the Production Database are the URLs of the Staging Instance and must be updated.

Answer: B

Explanation:

The likely cause of the issue is that a deployment should have been done on Production to initialize Fastly caching. This is because when the database is migrated from Staging to Production, any changes made to the Staging Database will not be reflected in the Production environment until a deployment is made. This includes any changes made to the Varnish Cache, which needs to be cleared in order for the new products to appear on the frontend.

Question No: 43

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. Increase multiple_process limit to spawn more processes for each consumer.

B. Set CONSUMER_WAIT_FOR_MAX_MESSAGES variable true in deployment stage.

C. Change max_messages from 10,000 to 1,000 for CRON_CONSUMER_RUNNERvariable.

Answer: B

Explanation:

The best way to meet this requirement is to set the CONSUMERWAITFORMAXMESSAGES variable to true in the deployment stage. This variable will ensure that the consumer will not terminate when there are no more messages in the queue and will instead wait until a new message is available, preventing it from closing the connection prematurely. Additionally, the multiple_processes limit can be increased to spawn more processes for each consumer, which will help ensure that messages can be processed faster.

Question No: 44

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. Make sure that orders are shipped and not left in processing state.

B. Configure currency rates for GBP and USD, so they are not empty.

C. Refresh Lifetime Statistics for "Total Invoiced".

Answer: B

Explanation:

To do this, the Architect needs to configure the currency rates for both GBP and USD in the Admin Panel, so that the correct exchange rates are applied to each currency. This will ensure that the correct amounts are shown in the sales orders report, and will also make sure that the correct amount is charged to customers in the new UK website.

Question No: 45

An Adobe Commerce Architect needs to set up two websites on a single Adobe Commerce instance with base URLs: example.com and website2.example.com.

How should the Architect configure this project so that both websites can use the same customer base?

A. Change Session Cookie attribute to "SameSite=None"

B. Disable Session Validation for "HTTP_X_FORWARDED_FOR" header

C. Set Cookie Domain for both websites to ".example.com"

Answer: C

Explanation:

By setting the same cookie domain for both websites, the customer base can be shared between both websites, as the customer will be authenticated by the same cookie across both sites. This will ensure that customers don't have to log in twice when switching between the two sites.

Question No: 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.

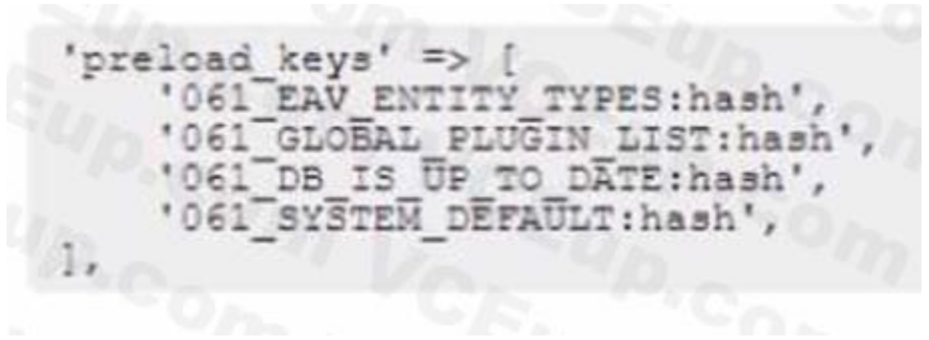
• The id_prefix of the frontend =>page_cache is set to 061_.

• The id_prefix of frontend => default: is not set.

• The Architect has enabled and configured Redis L2 caching.

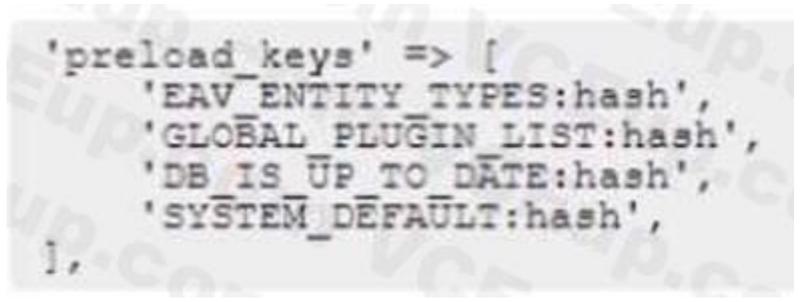
How should the preload.keys be configured?

A)



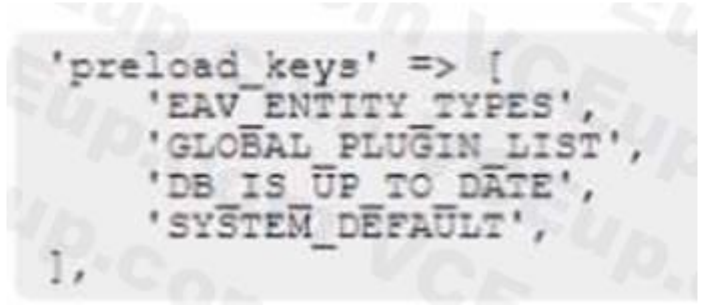
```
'preload_keys' => [
    '061_EAV_ENTITY_TYPES:hash',
    '061_GLOBAL_PLUGIN_LIST:hash',
    '061_DB_IS_UP_TO_DATE:hash',
    '061_SYSTEM_DEFAULT:hash',
],
```

B)



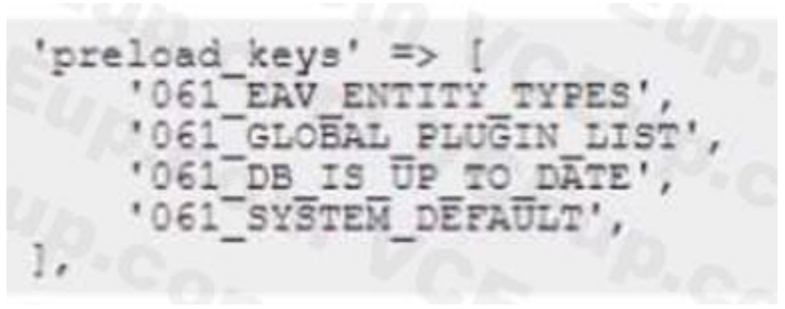
```
'preload_keys' => [
    'EAV_ENTITY_TYPES:hash',
    'GLOBAL_PLUGIN_LIST:hash',
    'DB_IS_UP_TO_DATE:hash',
    'SYSTEM_DEFAULT:hash',
],
```

C)



```
'preload_keys' => [
    'EAV_ENTITY_TYPES',
    'GLOBAL_PLUGIN_LIST',
    'DB_IS_UP_TO_DATE',
    'SYSTEM_DEFAULT',
],
```

D)


```
'preload_keys' => [
    '061_EAV_ENTITY_TYPES',
    '061_GLOBAL_PLUGIN_LIST',
    '061_DB_IS_UP_TO_DATE',
    '061_SYSTEM_DEFAULT',
],
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: D

Explanation:

Question No: 47

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. Edit project env.php file, configure 'admin_locales_for.build' value, and specify all required locales

B. Adjust the tool's build script and specify required locales during 'setup:static-content:deploy' command

C. Modify project config.php file, configure 'admin_locales_for_deploy' value, and specify all required locales

Answer: B

Explanation:

To ensure that the Admin Panel works correctly and is maintainable, the Architect should adjust the tool's build script so that it specifies all of the required locales when executing the 'setup:staticcontent: deploy' command. This will ensure that the project is correctly configured for all supported locales and will also make sure that the build script does not need to be modified each time a new locale is added.

Question No: 48

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

Answer: C

Explanation:

Developer Mode is the mode best suited to achieve the requirements set out by the Adobe Commerce Architect. In Developer Mode, errors are logged and hidden from the user, and the cache mode can only be changed from the command line. Additionally, static files are created dynamically and then cached, which is the desired behavior for this project.

Question No: 49

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/iuagento 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. Educate the merchant to not flush cache storage and only refresh the caches in future.

B. Set the 'Stores > Configuration' option for "Store Session Data Separately' to 'Yes' in the Adobe Commerce Admin Panel.

C. Check app/etc/evn.php and make sure that the Redis configuration for caches and session data use different database numbers.

Answer: C

Explanation:

This is because when the Magento cache storage is flushed, all caches and session data stored in Redis are cleared. To prevent session data from being lost when caches are flushed, the merchant should configure different Redis databases for caches and session data, so that flushing the Magento cache storage only clears the caches and not the session data.

Question No: 50

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 marketplacejeeds.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. Add the Uniform Resource Name to the XSD file in the config XML file.

C. Provide schema to validate a merged file.

D. Provide schema to validate an individual file.

E. Create a class that implements \Magento\Framework\Config\DataInterface.

F. Create validation rules in marketplace.schema.xsd.

Answer: DEF

Explanation:

The three steps that the Architect can take to ensure validation of the configuration files with unique validation rules for the individual and merged files are: D. Provide schema to validate an individual file, F. Create validation rules in marketplace.schema.xsd, and E. Create a class that implements

\Magento\Framework\Config\DataInterface. By providing schema to validate individual files, creating validation rules in marketplace.schema.xsd, and creating a class that implements

\Magento\Framework\Config\DataInterface, the Architect can ensure that the configuration files are validated with unique validation rules for the individual and merged files.