Adobe

Exam Questions AD0-E724

Adobe Commerce Developer Professional

NEW QUESTION 1
An Adobe Commerce developer is creating a module (Vendor.ModuleName) to be sold on the Marketplace. The new module creates a database table using declarative schema and now the developer needs to make sure the table is removed when the module is disabled.
What must the developer do to accomplish this?

A. There is nothing further the developer needs to d
B. The table will be removed when the module is disabled and bin/magento setup:upgrade is run.
C. There is nothing further the developer needs to d
D. The table will be removed when the when bin/magento module:uninstall vendor_ModuleName is run.
E. Add a schema patch that implements Magento\Framework\setup\Patch\PatchRevertabieinterface and drops the table in the revert function.

Answer: A

Explanation:
When you disable a module, Magento removes all of its declarative schema changes from the database the next time you run bin/magento setup:upgrade."
This means that when the developer disables the module and runs setup:upgrade, Adobe Commerce will automatically handle the removal of the database table created by the module's declarative schema.
For reference, here are some key points from the documentation:
? [Disable a Module](https://x.com/i/grok?text=Disable%20a%20Module)- This section explains how Magento handles the database schema when a module is disabled.
? Declarative Schema Configuration- Provides an overview of how declarative
schema works, including its behavior when modules are disabled or uninstalled. Therefore, based on the official Adobe Commerce documentation, the correct action for the
developer is to do nothing further beyond disabling the module and running bin/magento setup:upgrade. Magentowill take care of removing the table associated with the module as part of its schema management process.

NEW QUESTION 2
Which file is used to add a custom router class to the list of routers?

A. routes.xml
B. di.xml
C. config.xml

Answer: A

Explanation:
To add a custom router class to the list of routers in Magento, theroutes.xml file is used. This file should be located in theetcdirectory of the module, under the appropriate area (eitherfrontendoradminhtml). Within theroutes.xmlfile, you define a router with an ID, a route with an ID and frontName, and specify the module that the route corresponds to. This setup allows Magento to recognize and utilize the custom router when processing URLs, directing requests to the appropriate controllers based on the custom routing logic defined.

NEW QUESTION 3
An Adobe Commerce developer is developing a custom module. As part of their implementation they have decided that all instances of their Custom\Module\Model\Example class should receive a new instance of Magento\Filesystem\Adapter\Local.
How would the developer achieve this using di. xml?
A)

```
<type name="Custom\Module\Model\Example">
    <arguments>
        <argument name="adapter" xsi:type="object" shared="false">Magento\Filesystem\Adapter\Local</argument>
    </arguments>
</type>
```

B)

```
<type name="Custom\Module\Model\Example">
    <arguments>
        <argument name="adapter" xsi:type="object" singleton="false">Magento\Filesystem\Adapter\Local</argument>
    </arguments>
</type>
```

C)

```
<type name="Custom\Module\Model\Example">
    <arguments>
        <argument name="adapter" xsi:type="object" transient="true">Magento\Filesystem\Adapter\Local</argument>
    </arguments>
</type>
```

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

Answer: C

NEW QUESTION 4
How can a developer prioritize a plugin's execution, if possible?

A. The developer can use sortOrder property by specifying a lower value than the target plugin.
B. The developer can use sortOrder property by specifying a higher value than the target plugin.
C. This cannot be achieved as the plugins are always executed by their module's load order in app/etc/config.php file.

Answer: A

Explanation:
A developer can prioritize the execution of a plugin by using thesortOrder property within the plugin's declaration in thedi.xmlfile. Specifying a lower value for the sortOrderproperty gives the plugin higher priority, meaning it will be executed before other plugins with a higher sortOrder value. This allows developers to control the order of plugin execution, which can be critical for ensuring the desired outcome when multiple plugins are affecting the same method.

NEW QUESTION 5
There is an integration developed using a cron service that runs twice a day, sending the Order ID to the integrated ERP system if there are orders that are able to create an invoice. The order is already loaded with the following code:
$order = $this->orderRepository->get($orderId);
In order to verify if the store has invoices to be created, what implementation would the Adobe Commerce developer use?
A)

```
if ($order->canInvoice()) {
    // send integration to the ERP
}
```

B)

```
if ($order->hasInvoice()) {
    // send integration to the ERP
}
```

C)

```
if (!$order->isPaymentReview()) {
    // send integration to the ERP
}
```

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

Answer: A

Explanation:
The correct implementation to check if an order is eligible for invoicing is to use the $order->canInvoice() method. This method checks whether the order meets all necessary conditions for an invoice to be created, such as the order not being fully invoiced or canceled.
Option A is correct for the following reasons:
? Using canInvoice() for Invoicing Eligibility:The $order->canInvoice() method is specifically designed to verify if an order can have an invoice generated. It returns true only if the order is in a state where it can be invoiced. This makes it the appropriate method for determining whether the order should be sent to the ERP system for invoicing.
? uk.co.certification.simulator.questionpool.PList@45e8e59a
: Magento??s developer documentation on the Order model highlights canInvoice() as the recommended approach for determining invoice eligibility, particularly when automating processes like ERP integration.
Alternatives and Limitations:
Option B: The $order->hasInvoice() method only checks if there is already an invoice associated with the order, which does not indicate whether the order is eligible for new invoicing. It returns true if any invoice exists for the order, which is not suitable for this scenario.
Option C: The $order->isPaymentReview() method checks if the order is in a payment review state, which is not directly related to invoice creation eligibility. It would not provide accurate information on whether the order can be invoiced.
By using canInvoice(), the developer ensures that the cron job will only send orders that are ready for invoicing to the ERP system, adhering to Adobe Commerce??s order processing logic.

NEW QUESTION 6
How should a developer display a custom attribute on the category edit page in the admin panel when a new module Vendor.Category is created?

A. Create view/adminhtml/layout/catalog_category_edit xml in the module, and then define a block that would display the field for the attribute.
B. The field for the attribute will appear automatically.
C. Create view/adminhtml/ui_component/category_form.xml file in the module, and then define the field for the attribute.

Answer: C

Explanation:

To display a custom attribute on the category edit page in the Magento admin panel, a developer should use the UI component approach. This involves creating a category_form.xmlfile within theview/adminhtml/ui_componentdirectory of the module. This XML file defines the form fields (including any custom attributes) that should appear on the category edit page. The UI component system in Magento provides a flexible way to manage form elements and their interactions on the admin side, ensuring a consistent user

NEW QUESTION 7
An Adobe Commerce developer is creating a new console command to perform a complex task with a lot of potential terminal output. If an error occurs, they want to provide a message that has higher visibility than some of the other content that may be appearing, so they want to ensure it is highlighted in red (as seen in the screenshot):

How can they customize the appearance of this message?

A. Call the setDecorationType(Stype) method On the Symfony\Console\Output\OutputInterface Object before Calling writeln().
B. Wrap the output content in tags like <error>, <info>, or <comment>.
C. Throw a new commandException with the desired message passed as an argument.

Answer: B

Explanation:
In Adobe Commerce, when developing custom console commands, you can customize output messages by using special tags provided by Symfony Console, which Adobe Commerce relies on. These tags are designed to help differentiate types of messages and can be used to add color or emphasis to the output, enhancing visibility. For critical error messages, wrapping the message in the <error> tag will display it in red, as shown in your screenshot. The available tags include:
? <error> for red-colored error messages.
? <info> for informational messages (often displayed in blue).
? <comment> for comments or warnings (usually yellow).
$output->writeln('<error>A critical error has occurred.</error>');
This method is effective and widely used for output customization in Symfony-based console commands.
Additional Resources:
? Adobe Commerce Developer Guide: Console Command Customization
? Symfony Console Output Formatting

NEW QUESTION 8
What are two features with Adobe Commerce Cloud that come out of the box? (Choose Two.)

A. Support ACL
B. Continuous deployment provided with the platform
C. A built in connector with all major blog platforms
D. Fastly

Answer: BD

Explanation:
Adobe Commerce Cloud offers several out-of-the-box features, including built-in Fastly integration for CDN and web application firewall services, as well as continuous deployment capabilities through its cloud infrastructure.
? Continuous Deployment:
? uk.co.certification.simulator.questionpool.PList@200a841f
? Fastly Integration:
? Why Options B and D are Correct:
:
Adobe Commerce Cloud documentation onFastly CDN
Adobe Commerce Cloud documentation onDeployment and CI/CD

NEW QUESTION 9
Which has its own root category?

A. Websites
B. Stores
C. Store Views

Answer: B

Explanation:
In Magento, each store has its own root category. The root category acts as the top-level category under which all other categories and products are organized for that particular store. This structure allows for different catalog structures across multiple stores within a Magento installation, enabling a tailored product offering and navigation experience for each store. Theability to assign a unique root category to each store is a fundamental aspect of Magento's multi-store functionality, providing the flexibility to cater to diverse market segments or branding requirements.

NEW QUESTION 10

An Adobe Commerce developer was asked to provide additional information on a quote. When getting several quotes, the extension attributes are returned, however, when getting a single quote it fails to be returned.
What is one reason the extension attributes are missing?

A. The developer neglected to add coiiection="trueM to their attribute in etc/extension_attributes.xmi file.ottribute code="my_attributesM type="MyVendor\MyModule\Api\Data\^AttributeInterface[]M collection="true" />
B. The developer neglected to provide a plugin On Hagento\Quote\Api\CartRepositoryInterface: :get.
C. The developer neglected to implement an observer on the coiiection_ioad_after event.

Answer: B

Explanation:
In Adobe Commerce, to retrieve custom extension attributes on an entity such as a quote, you need to ensure that these attributes are properly loaded when accessing the entity directly via repository interfaces. When fetching a single quote, it is essential to use a plugin on CartRepositoryInterface::get to include the extension attributes as this method is responsible for loading individual quote data.
If the plugin is missing for the get method, the extension attributes won??t be loaded for single quote retrieval, leading to their absence in the result.
Additional Resources:
? Adobe Commerce Developer Guide: Extension Attributes
? Magento 2 Repositories and Plugins

NEW QUESTION 10
For security reasons, merchant requested to a developer to change default admin url to a
unique url for every branch/environment of their Adobe Commerce Cloud project. Which CLI command would the developer use update the admin url?

A. ece-tools variable:update ADMIN_URL
B. magento-cloud variable:set ADMIN_URL
C. bin/magento adminuri:set <admin_uri>

Answer: B

Explanation:
The CLI command that the developer would use to update the admin url is magento-cloud variable:set ADMIN_URL. This command sets an environment variable called ADMIN_URL with a custom value for the admin url on a specific environment. Environment variables are configuration settings that affect the behavior of the Adobe Commerce Cloud application and services. By setting an environment variable for ADMIN_URL, the developer can change the default admin urlto a unique url for every branch/environment of their Adobe Commerce Cloud project. Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 15
What is one way a developer can upgrade the ECE-Tools package on an Adobe Commerce Cloud project?

A. Cloud CLI for Commerce tool
B. Project Web Interface
C. Composer

Answer: C

Explanation:
According to the Adobe Commerce Developer Documentation, one way a developer can upgrade the ECE-Tools package on an Adobe Commerce Cloud project is by using Composer, which is a dependency management tool for PHP projects. The ECE- Tools package contains scripts and tools that help manage and deploy Adobe Commerce Cloud projects on the cloud infrastructure. To upgrade the ECE-Tools package using Composer, the developer needs to run the following command in the project root directory: composer update magento/ece-tools --with-dependencies
On an Adobe Commerce Cloud project, the recommended way to upgrade the ECE-Tools package is through Composer, a dependency manager for PHP. Composer is used to manage the dependencies of the project, including ECE-Tools, and it provides the necessary commands to update packages to their latest versions or to specific versions as required.

NEW QUESTION 19
Which log file would help a developer to investigate 503 errors caused by traffic or insufficient server resources on an Adobe Commerce Cloud project?

A. mysql-slow.log
B. access.log
C. cloud.log

Answer: B

Explanation:
The access.log file stores the information about the requests and responses that occur on the web server, such as the IP address, timestamp, request URI, response code, andreferer URL1. By checking the access.log file, a developer can identify the requests that resulted in 503 errors and the possible causes of those errors, such as traffic spikes, insufficient server resources, or misconfiguration1.
To check the access.log file on an Adobe Commerce Cloud project, a developer can use the following command in the CLI:
grep -r "\" 50 [0-9]" /path/to/access.log
This command will search for any lines in the access.log file that contain a response code starting with 50, which indicates a server error1. The developer can then compare the timestamps of those lines with the exception.log and error.log files to find more details about the errors1.
Alternatively, if the error has occurred in the past and the access.log file has been rotated (compressed and archived), the developer can use the following command in the CLI (Pro architecture only):
zgrep "\" 50 [0-9]" /path/to/access.log.<rotation ID>.gz
This command will search for any lines in the compressed access.log file that contain a response code starting with 501. The rotation ID is a number that indicates how many
times the log file has been rotated. For example, access.log.1.gz is the most recent rotated log file, and access.log.10.gz is the oldest rotated log file1.

NEW QUESTION 21

In a new release of a module, a developer decides to rename a table that was defined in the earlier versions. Which action, if any, allows the developer to prevent data loss?

A. Define onCreate="migrateDataFromAnotherTable(old_table_name)" attribute in the table tag.
B. Declarative schema supports RENAME TABLE', so the data will be migrated to the new table automatically.
C. Define the table and columns mapping in the db.schema_whitelist.json

Answer: C

Explanation:
When renaming a table in Magento, to prevent data loss, the developer must define the table and its columns mapping in thedb_schema_whitelist.jsonfile. This declarative schema approach ensures that the data migration tool knows about the changes and can migrate data from the old table to the newly named table without losing any data.

NEW QUESTION 26
What are two ways to access the PHP error logs on Adobe Commerce Cloud? (Choose Two.)

A. Use the dedicated command from Cloud CLI for Commerce.
B. Navigate to the dedicated entry in the Project Web Interface.
C. Connect to the the servers via SSH and localize the log files.
D. Use the Adobe Admin Log application.

Answer: AC

Explanation:
Two ways to access the PHP error logs on Adobe Commerce Cloud are to use the dedicated command from Cloud CLI for Commerce and to connect to the servers via SSH and localize the log files. The Cloud CLI for Commerce is a command-line tool that allows developers to interact with their Adobe Commerce Cloud projects and environments. The developer can use the command magento-cloud log php to view or download the PHP error logs from any environment.
In Adobe Commerce Cloud, accessing PHP error logs can be done through the Cloud CLI or by directly connecting to the server via SSH.
? Cloud CLI for Accessing Logs:
? uk.co.certification.simulator.questionpool.PList@3e5e7f0f
? SSH for Direct Log Access:
? Why Options A and C are Correct:
: Adobe Commerce Cloud documentation onAccessing Logs

NEW QUESTION 29
What folder would a developer place a custom patch on an Adobe Commerce Cloud project to have it automatically applied during the build phase?

A. Add the patch file to the m2-hotfixes/ directory
B. Add the patch file to the patches/ directory
C. Add the patch file to the m2-patches/ directory

Answer: A

Explanation:
On an Adobe Commerce Cloud project, a custom patch should be placed in them2-hotfixes/directory to have it automatically applied during the build phase. The patches in this directory are applied in alphabetical order and can be used to apply quick fixes to the code that will be included in the build artifact.

NEW QUESTION 32
Which command should be used to refresh the cache using the command line?

A. magentocacheclean<type>
B. magentocacherefresh<type>
C. magentocachedelete<type>

Answer: A

Explanation:
To refresh the cache using the command line in Magento, the command bin/magento cache:clean <type>is used. This command clears the cache for the specified type, such as configuration, layout, or block HTML. Cleaning the cache is an essential maintenance task that helps to update the store's frontend with the latest changes and ensure that customers see the most current content. This command is particularly useful during development or after making changes to the store's configuration or layout.

NEW QUESTION 37
An Adobe Commerce developer is tasked with adding custom data to orders fetched from the API. While keeping best practices in mind, how would the developer achieve this?

A. Create an extension attribute on Nagento\sales\Api\E)ata\orderinterface and an after plugin onMagento\Sales\Model\Order: :getExtensionAttributes() to add the custom data.
B. Create an extension attribute On Magento\Sales\Api\Data\OrderInterface and an after plugin On Magento\Sales\Api\OrderRepositoryInterface On geto and getListo to add the custom data.
C. Create a before plugin on Magento\sales\model\ResourceModel\order\collection: :load and alter the query to fetch the additional dat
D. Data will then be automatically added to the items fetched from the API.

Answer: B

Explanation:
The developer should create an extension attribute on theMagento\Sales\Api\Data\OrderInterfaceinterface and an after plugin on

theMagento\Sales\Api\OrderRepositoryInterface::get()andMagento\Sales\Api\OrderReposit oryInterface::getList()methods.
The extension attribute will store the custom data. The after plugin will be used to add the custom data to the order object when it is fetched from the API.
Here is the code for the extension attribute and after plugin: PHP

```
namespace MyVendor\MyModule\Api\Data;
interface OrderExtensionInterface extends \Magento\Sales\Api\Data\OrderInterface
{
/**
* Get custom data.
*
* @return string|null
*/
public function getCustomData();
/**
* Set custom data.
*
* @param string $customData
* @return $this
*/
public function setCustomData($customData);
}
namespace MyVendor\MyModule\Model;
class OrderRepository extends \Magento\Sales\Api\OrderRepositoryInterface
{
/**
* After get order.
*
* @param \Magento\Sales\Api\OrderRepositoryInterface $subject
* @param \Magento\Sales\Api\Data\OrderInterface $order
* @return \Magento\Sales\Api\Data\OrderInterface
*/
public function afterGetOrder($subject, $order)
{
if ($order instanceof OrderExtensionInterface) {
$order->setCustomData('This is custom data');
}
return $order;
}
/**
* After get list.
*
* @param \Magento\Sales\Api\OrderRepositoryInterface $subject
* @param \Magento\Sales\Api\Data\OrderInterface[] $orders
* @return \Magento\Sales\Api\Data\OrderInterface[]
*/
public function afterGetList($subject, $orders)
{
foreach ($orders as $order) {
if ($order instanceof OrderExtensionInterface) {
$order->setCustomData('This is custom data');
}
}
return $orders;
}
}
```

Once the extension attribute and after plugin are created, the custom data will be added to orders fetched from the API.

NEW QUESTION 38
An Adobe Commerce developer is asked to change the tracking level on a custom module for free downloading of pdf and images.
The module contains following models: Vendor\FreeDownload\Model\Download Vendor\FreeDownload\Model\DownloadPdf extends Vendor\FreeDownload\Model\Download
Vendor\FreeDownload\Model\DownloadImage extends Vendor\FreeDownload\Model\Download
Download class has a parameter for tracking_level.
How will the developer configure the tracking_level parameter, in di.xml.to have a value of 4 for Download class and all classes that extend Download?
A)

Configure the parameter on a child class and add `parent` attribute as it will be propagated to siblings and parent.

```
<type
  name="Vendor\FreeDownload\Model\DownloadPdf"
  parent="Vendor\FreeDownload\Model\Download"
>
    <arguments>
        <argument name="tracking_level" xsi:type="integer">4</argument>
    </arguments>
</type>
```

B)

Configure the parameter on the all child classes and set the `parent` attribute on one of them.

```
<type name="Vendor\FreeDownload\Model\DownloadPdf"
  parent="Vendor\FreeDownload\Model\Download">
    <arguments>
        <argument name="tracking_level" xsi:type="number">4</argument>
...
<type name="Vendor\FreeDownload\Model\DownloadImage">
    <arguments>
        <argument name="tracking_level" xsi:type="number">4</argument>
...
```

C)

Configure the parameter on parent class, as it will be propagated on descendant classes.

```
<type name="Vendor\FreeDownload\Model\Download">
    <arguments>
        <argument name="tracking_level" xsi:type="number">4</argument>
    </arguments>
</type>
```

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

Answer: C

Explanation:
To configure a parameter for a parent class so that it propagates to all descendant classes, the correct approach is to define the parameter on the parent class within di.xml. This way, all child classes inheriting from this parent will automatically use the parameter value unless explicitly overridden.
Option C is correct for the following reasons:
? Configuring on the Parent Class (Vendor\FreeDownload\Model\Download):By setting the tracking_level parameter directly on the Download class, you ensure that all classes extending this class, such as DownloadPdf andDownloadImage, will inherit the tracking_level parameter value. This method leverages Magento's dependency injection configuration, which allows parameters set on a parent class to cascade to child classes.
? uk.co.certification.simulator.questionpool.PList@15a6494
: Magento??s official developer documentation outlines that class dependencies and configuration parameters defined in di.xml at a higher level are accessible to descendant classes. This is a standard practice in Magento for setting parameters that affect a hierarchy of classes.
Avoiding Redundant Configuration:Unlike Option A, which sets the parameter on an individual child class, or Option B, which redundantly sets the parameter on multiple child classes, Option C is optimal as it centralizes the configuration. This reduces the risk of discrepancies and simplifies maintenance.
Options A and B are incorrect because:
Option A configures the parameter on a single child class, which will not affect other child classes such as DownloadImage.
Option B redundantly sets the parameter for each child class individually, which is unnecessary when the parameter can be inherited from the parent.

NEW QUESTION 39
A developer is making customizations in the checkout, and access to the quotes shipping address is needed. Which file provides the shipping address of the current quote?

A. Magento_Checkout/js/model/quote
B. Magento_Quote/js/model/model
C. Magento_Checkout/js/model/quote-shipping-address

Answer: A

Explanation:
This file provides the shipping address of the current quote by using the getShippingAddress() method. For example, the following code snippet gets the shipping address from the quote object and logs it to the console:
define(['Magento_Checkout/js/model/quote'
],function(quote) { 'use strict';
varshippingAddress = quote.getShippingAddress(); console.log(shippingAddress);
});
The file Magento_Quote/js/model/model does not exist in Magento 2, and the file Magento_Checkout/js/model/quote-shipping-address is not a valid way to access the shipping address of the current quote. You can read more about the quote object and its methods in the Magento 2 developer documentation.
In Adobe Commerce, the shipping address of the current quote is accessed through the JavaScript fileMagento_Checkout/js/model/quote. This file includes various quote-related data, including shipping and billing addresses, items in the cart, and totals. There is no Magento_Checkout/js/model/quote-shipping-addressfile, and Magento_Quote/js/model/modelis not a valid path, making option A the correct choice.

NEW QUESTION 41
A developer is investigating a problem with the shopping cart. Some of the cart records in the database are being checked. Which table should the developer check?

A. sates.quote
B. sales.cart
C. quote

Answer: C

Explanation:
? A quote is a temporary object that represents the customer??s shopping cart before it is converted into an order1. A quote can be created by a guest or a registered customer, and it can be active or inactive depending on whether the customer has completed the checkout process or not2.
? The quote table stores the basic information about the quote, such as the customer ID, email, currency, subtotal, grand total, shipping method, payment method, and so on2. Each row in the quote table corresponds to one quote object, identified by a unique entity ID2.
? The quote item table stores the details of each product added to the quote, such
as the product ID, SKU, name, price, quantity, discount amount, tax amount, and so on2. Each row in the quote item table corresponds to one quote item object, identified by a unique item ID2. A quote item object is associated with a quote object by the quote ID column in the quote item table2.
? When a customer adds a product to the cart, Magento creates or updates a quote
object and a quote item object for that product. The quote object is initialized by the Magento\Quote\Model\Quote class, which implements the Magento\Quote\Api\Data\CartInterface interface3. The quote item object is initialized by the Magento\Quote\Model\Quote\Item class, which implements the Magento\Quote\Api\Data\CartItemInterface interface3.
? Magento uses various classes and interfaces to manipulate the quote and quote
item objects, such as the Magento\Quote\Model\QuoteRepository class, which implements the Magento\Quote\Api\CartRepositoryInterface interface for saving and retrieving quotes from the database3, and the Magento\Quote\Model\QuoteManagement class, which implements the Magento\Quote\Api\CartManagementInterface interface for creating and submitting quotes to orders3.
? Magento also uses various events and observers to perform additional actions on
the quote and quote item objects, such as applying discounts, calculating taxes, validating stock availability, and so on. Some of these events are: sales_quote_add_item, sales_quote_collect_totals_before, sales_quote_collect_totals_after, sales_quote_save_before, sales_quote_save_after, sales_model_service_quote_submit_before, sales_model_service_quote_submit_after.
? Magento provides various APIs for interacting with the quote and quote item objects, such as the Magento\Quote\Api\CartManagementInterface API for creating empty carts and placing orders from carts, the Magento\Quote\Api\CartItemRepositoryInterface API for adding, updating, and removing items from carts, and the Magento\Quote\Api\CouponManagementInterface API for managing coupons for carts.

NEW QUESTION 44
An Adobe Commerce developer is tasked with adding an new export option for the order grid, they have added the following code for the export button within sales_order_grid.xml:

```
<exportButton>
    <settings>
        <options>
            <option name="txt" xsi:type="array">
                <item name="value" xsi:type="string">txt</item>
                <item name="label" xsi:type="string" translate="true">TXT</item>
                <item name="url" xsi:type="string">vendor_module/sales/export/customExport</item>
            </option>
        </options>
    </settings>
</exportButton>
```

Upon testing, they are getting redirected, what would be a cause for this error?

A. The option's uri attribute is not valid.
B. The layout cache needs to be refreshed.
C. The developer has to add a formkey for the new export option.

Answer: C

Explanation:
The developer has to add a formkey for the new export option because the formkey is required for security reasons. Without the formkey, the request will be rejected and redirected to the dashboard page. Verified References: [Magento 2.4 User Guide] [Magento 2.4 DevDocs]
When adding custom export options to grids in Magento, it's crucial to include a form key for actions that involve form submission. Magento relies on form keys for CSRF (Cross-Site Request Forgery) protection, so omitting the form key can lead to redirects or failed operations.
? Form Key Requirement:
? uk.co.certification.simulator.questionpool.PList@755b4fcf
? Why Option C is Correct:
? Solution:
: Adobe Commerce documentation onForm Key and CSRF Protection Magento DevDocs onAdding Buttons to Grids

NEW QUESTION 49
Which CLI command should be used to determine that static content signing is enabled?

A. bin/magento config:show dev/static/status
B. bin/magento config:show dev/static/sign
C. bin/magento config:show dev/static/sign/status

Answer: B

Explanation:
After a thorough search of the provided documents, I couldn't find a direct reference to the specific CLI command for determining if static content signing is enabled in Magento. However, the typical command for checking configuration settings in Magento is bin/magento config:show <path>, where<path>is the configuration path for the setting you wish to view. Based on Magento's configuration path patterns and the options provided, the most logical choice would beB. bin/magento config:show dev/static/sign, although this cannot be confirmed without further context or documentation.

NEW QUESTION 54
An Adobe Commerce Developer has written an importer and exporter for a custom entity. The client is using this to modify the exported data and then re-importing the file to batch update the entities.
There is a text attribute, which contains information related to imagery in JSON form, media_gallery. This is not a field that the client wants to change, but the

software they are using to edit the exported data seems to be modifying it and not allowing it to import correctly.
How would the developer prevent this?
A) Specify a serializer class for the attribute using the $_transformAttrs class property array for both the exporter and importer so it gets converted:

```
protected $_transformAttrs = [
    'media_gallery' => \Magento\Framework\Serialize\Serializer\Json::class
];
```

B) Strip the attribute from the imported file by adding it to the s_strippedAttrs class property array:

```
protected $_strippedAttrs = [
    'media_gallery'
];
```

C) Prevent it from being exported by adding it to the $_disat>iedAttrs class property array:

```
protected $_disabledAttrs = [
    'media_gallery'
];
```

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

Answer: C

Explanation:
To prevent the media_gallery attribute from being exported as part of the custom entity's data, the developer should add this attribute to the $_disabledAttrs class property array. This effectively excludes the attribute from the export process, ensuring that it does not appear in the exported file and thus will not be modified or re-imported inadvertently.
Option C is correct for the following reasons:
? Preventing Export with $_disabledAttrs:Adding media_gallery to the
$_disabledAttrs array tells the system to skip this attribute during export. This prevents the attribute from being included in the export file, thus removing the risk of it being altered by external software that handles the file. Since the attribute will not be present in the exported data, it will remain unchanged in the database when re-importing.
? uk.co.certification.simulator.questionpool.PList@6993fe6
: The Adobe Commerce documentation on import/export processes outlines how
$_disabledAttrs can be used to filter out sensitive or unnecessary attributes from export files.
Alternative Options and Their Limitations:
Option A: Using $_transformAttrs with a serializer is useful for encoding or decoding attribute data during export/import, but it does not prevent the attribute from being included in the export. This would only help if the media_gallery data needed transformation, not exclusion.
Option B: $_strippedAttrs is applicable for filtering attributes from the imported file, not the exported file. It would not stop the attribute from being included in the export and hence does not align with the need to prevent modifications during export.
By configuring $_disabledAttrs, the developer effectively ensures the media_gallery attribute remains unmodified by preventing it from being included in export files altogether.

NEW QUESTION 57
What action can be performed from the Cloud Project Portal (Onboarding UI) of an Adobe Commerce Cloud project?

A. Set your developer SSH public key.
B. Update Project and environment variables
C. Add a Technical Admin

Answer: C

Explanation:
The Cloud Project Portal (Onboarding UI) of an Adobe Commerce Cloud project is a web interface that allows you to perform various actions related to your project, such as creating and managing environments, deploying code, configuring services, and adding users1. One of the actions that you can perform from the Cloud Project Portal is adding a Technical Admin, which is a user role that has full access to all environments and canperform any action on the project2. To add a Technical Admin from the Cloud Project Portal, you need to follow these steps2:
? Log in to the Cloud Project Portal with your Magento account credentials.
? Click on the Users tab on the left sidebar.
? Click on the Add User button on the top right corner.
? Enter the email address of the user you want to add as a Technical Admin.
? Select the Technical Admin role from the Role dropdown menu.
? Click on the Send Invitation button.
The user will receive an email invitation to join your project as a Technical Admin. They will need to accept the invitation and set up their account before they can access your project2.

NEW QUESTION 60
A developer is working on an Adobe Commerce Cloud project and wants to get connection data for the environment's deployed services. The developer has all of the necessary permissions to do this.
Which two options would the developer take to get the connection credentials? (Choose Two.)

A. Run the magento-cloud relationships CLI Command.
B. Get the data from the Project Web Interface dedicated section.
C. Execute ece-tools env:config:show services Command.
D. Connect to server via SSH and read $_ENV['services'] variable.

Answer: AB

Explanation:
In Adobe Commerce Cloud, connection data for deployed services (such as databases, caches, and other services) can be retrieved using different methods depending on the developer??s access and the tools available.
? Using magento-cloud relationships Command:
? uk.co.certification.simulator.questionpool.PList@73a02881
? Project Web Interface:
? Why Options A and B are Correct:
:
Adobe Commerce documentation onAccessing Services

NEW QUESTION 63
An Adobe Commerce developer has created a process that exports a given order to some external accounting system. Launching this process using the Magento CLI with the command php bin/magento my_module:order: process --order_id=<order_id> is required.
Example: php bin/magento my_module:order:process --order_id=1245. What is the correct way to configure the command?
A)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    parent::configure();
}

protected function values()
{
    return [new InputValue('order_id', InputValue::REQUIRED, 'Order ID')];
}
```

B)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');
    parent::configure();
}
```

C)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

D)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

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

Answer: D

Explanation:
To properly configure a Magento CLI command that includes a required argument, such as --order_id, which is mandatory for processing an order, the best approach is to use the addArgument method within the configure function. This method defines required arguments for the command, ensuring the user provides the necessary data.
Option D is correct for the following reasons:
? Using addArgument for Required Inputs:The addArgument method is used here to declare order_id as a required argument. This is more appropriate than addOption
when the parameter is essential for command execution and should not be omitted. By specifying InputArgument::REQUIRED, the command ensures that the order_id must be provided by the user.
? uk.co.certification.simulator.questionpool.PList@3259569d
: According to Magento??s official developer documentation, addArgument is used for required command-line arguments, and this is standard practice for defining necessary inputs in CLI commands.
Properly Configured Command Name and Description:The setName and setDescription methods are correctly used in this option to specify the command??s name and its purpose. This helps in making the command self-descriptive, improving usability and readability for other developers or administrators using the CLI.
Options A, B, and C are incorrect because they either:
Use addOption instead of addArgument, which is less suitable for mandatory parameters (Option B).
Define the same parameter redundantly (Option C).
Use other non-standard configurations that do not align with Magento's best practices for required CLI parameters (Option A).

NEW QUESTION 65
There is the task to create a custom product attribute that controls the display of a message below the product title on the cart page, in order to identify products that might be delivered late.
The new EAV attribute is_delayed has been created as a boolean and is working correctly in the admin panel and product page.
What would be the next implementation to allow the is_delayed EAV attribute to be used in the .phtml cart page such as $block->getProduct()->getIsDelayed()?
A)
Create a new file etc/catalog_attributes.xmi:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Catalog:etc/catalog_attributes.xsd">
    <group name="quote_item">
        <attribute name="is_delayed" />
    </group>
</config>
```

B)
Create a new file etc/extension attributes.xmi:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Api/etc/extension_attributes.xsd"
    <extension_attributes for="Magento\Catalog\Api\Data\ProductRenderInterface>
        <attribute code="is_delayed" type="bool" />
    </extension_attributes>
</config>
```

C)
Create a new file etc/eav attributes.xmi:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Eav:etc/eav_attributes.xsd">
    <entity type="quote_item">
        <attribute code="is_delayed">
            <field code="is_visible" locked="true" />
        </attribute>
    </entity>
</config>
```

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

Answer: A

Explanation:
To allow theis_delayedEAV attribute to be used in the .phtml cart page, the developer needs to create a new file calledetc/catalog_attributes.xmi. This file will contain the definition of theis_delayedattribute.
The following code shows how to create theetc/catalog_attributes.xmifile: XML
<?xml version="1.0"?>
<catalog_attributes>
<attribute code="is_delayed" type="int">
<label>Is Delayed</label>
<note>This attribute indicates whether the product is delayed.</note>
<sort_order>10</sort_order>
<required>false</required>
</attribute>
</catalog_attributes>
Once theetc/catalog_attributes.xmifile has been created, theis_delayedattribute will be available in the .phtml cart page. The attribute can be accessed using thegetIsDelayed()method of theProductclass.
PHP
$product = $block->getProduct();
$isDelayed = $product->getIsDelayed();
TheisDelayedvariable will contain the value of theis_delayedattribute. If the value of the attribute is 1, then the product is delayed. If the value of the attribute is 0, then the product is not delayed.

NEW QUESTION 69

Which action, if any, should be taken to forbid Adobe Commerce Admin from performing specific actions?

A. Create a new user role with custom-defined resources, and assign it to the admin user
B. This action cannot be taken since all admin users must have full access.
C. Enable custom roles in the store configuration, and assign admin user ID(s).

Answer: A

Explanation:
To forbid Adobe Commerce Admin from performing specific actions, a developer should create a new user role with custom-defined resources, and assign it to the admin user. This can be done by going toSystem > Permissions > Rolesand creating a new role. In theResourcessection, the developer can select the specific resources that they want to restrict the admin user from accessing.
To restrict specific actions within the Adobe Commerce Admin, the recommended approach is to utilize Magento's Access Control List (ACL). This can be done by creating a new user role with custom-defined resources and assigning this role to the admin user. This approach allows for granular control over what actions an admin user can perform by specifying allowed resources within the role. Magento's ACL system is designed to manage permissions effectively, ensuring that users only have access to the necessary functionalities required for their role.

NEW QUESTION 72
The di. xml file of a module attaches two plugins for the class Action.
The PluginA has the methods: beforeDispatch, aroundDispatch, afterDispatch. The PluginB has the methods: beforeDispatch, afterDispatch.

```
<config>
    <type name="Magento\Framework\App\Action\Action">
        <plugin name="vendor_module_plugina" type="Vendor\Module\Plugin\PluginA" sortOrder="10" />
        <plugin name="vendor_module_pluginb" type="Vendor\Module\Plugin\PluginB" sortOrder="20" />
    </type>
</config>
```

The around plugin code is:

```
class PluginA
{
    public function aroundDispatch(\Magento\Framework\App\Action\Action $subject, $next, $request)
    {
        // custom code
        return $request;
    }
}
```

What would be the plugin execution order?
A)

```
PluginA::beforeDispatch()
PluginA::aroundDispatch()
PluginB::beforeDispatch()
Action::dispatch()
PluginB: afterDispatch()
PluginA::aroundDispatch()
```

B)

```
PluginA::beforeDispatch()
PluginA::aroundDispatch()
PluginA: afterDispatch()

PluginA::beforeDispatch()
PluginA::aroundDispatch()
PluginB::beforeDispatch()
```

C)

```
Action::dispatch()
PluginB: afterDispatch()
PluginA::aroundDispatch()
PluginA::afterDispatch()
```

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

Answer: B

Explanation:
https://developer.adobe.com/commerce/php/development/components/plugins/#scenario-b-without-a-callable-around

NEW QUESTION 77
Which action, if any, can be taken to change the URL key of the product?

A. The product URL key is generated automatically, so it cannot be changed.
B. Use URL rewrite to map product id with the custom URL key.
C. In the product admin form, under the Search Engine Optimization fieldset, the URL key can be set

Answer: C

Explanation:
The URL key of a product is the text that is used to generate the product's URL. This text can be set by the merchant in the product admin form.
The URL key of a product in Adobe Commerce can be changed in the product admin form under the "Search Engine Optimization" fieldset. Here, the URL key can be set or edited manually, allowing for customization beyond the automatically generated value. This field is typically used to ensure that the product URL is search-engine friendly and relevant to the product.

NEW QUESTION 79

Which file on an Adobe Commerce Cloud project allows a developer to upgrade the PHP version and enable/disable a PHP extension?

A. magento.app.yaal
B. .magent
C. en
D. yaml
E. php.ini

Answer: B

Explanation:
The.magento.env.yamlfile is used on an Adobe Commerce Cloud project to customize the environment configuration, including the PHP version and enabling/disabling PHP extensions. This YAML configuration file provides the ability to manage service configurations and is essential for customizing the Cloud environment.

NEW QUESTION 83
A developer defined a new table in db.schema.xml while creating a new module.
What should be done to allow the removal of columns from the database when deleting them from db.schema.xml?

A. The removable columns should be defined in db_schema_whitelist.json.
B. The columns should have "removable" attribute set to "true" in the db.schema.xml.
C. The removable columns should be defined in db.schema_blacklist.json.

Answer: A

Explanation:
If a developer wants to allow the removal of columns from the database when deleting them fromdb.schema.xml, they need to define the removable columns in thedb_schema_whitelist.jsonfile. This file will tell Magento which columns can be removed from the database.
To allow columns to be removed from the database when they are deleted from db_schema.xml, they must be listed in thedb_schema_whitelist.jsonfile. This file acts as a reference for which database schema elements are safe to modify or delete, providing a safeguard against unintentional data loss during schema updates.

NEW QUESTION 87
An Adobe Commerce developer is tasked to add a file field to a custom form in the administration panel, the field must accept only .PDF files with size less or equal than 2 MB. So far the developer has added the following code within the form component xml file, inside the fieldset node:

```
<field name="pdf_file" formElement="fileUploader">
    <formElements>
        <fileUploader>
            <settings>
                <uploaderConfig>
                    <param xsi:type="string" name="url">myvendor_mymodule/customForm/uploadPdf</param>
                </uploaderConfig>
            </settings>
        </fileUploader>
    </formElements>
</field>
```

How would the developer implement the validations?
A) Add the Validations Within the HyVendor\MyModule\Controller\Adminhtml\CustomEntity\UploadPdf Controller

```
public function execute()
{
    $file = $this->fileUploaderFactory->create($this->getRequest()->getPdfFile());
    if($file->getExtexsion() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.'));
    }
    if($file->getSize() >= '2048000') {
        throw new InvalidFileException(__('The file size must be less or equal than 2MB'));
    }

    return $this->resultFactory->create(ResultFactory::TYPE_PAGE);
}
```

B) Add a virtual type forMyvendor\MyModuie\Modei\customPdfupioader specifying the aiiowedExtensions and the maxFiiesize for the constructor, within the module's di.xmi:

```
<type name="MyVendor\MyModule\Model\CustomPdfUploader">
    <arguments>
        <argument name="allowedExtensions" xsi:type="string">pdf</argument>
        <argument name="maxFileSize" xsi:type="number">2048000</argument>
    </arguments>
</type>
```

C) Add the following code inside the<settings> node:

```
<allowedExtensions>pdf</allowedExtensions>
<maxFileSize>2048000</maxFileSize>
```

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

Answer: C

Explanation:
To add file upload validation for a custom form field in the Adobe Commerce admin panel, which should restrict the file type to .pdf and limit the file size to 2 MB, the recommended approach is to include the validation parameters directly within the
<settings> node in the form??sXML configuration. This ensures that the validation occurs on the client-side as well as server-side, providing immediate feedback to users before submission.
Option C is correct for the following reasons:
? Adding Validation Inside <settings>:By placing the <allowedExtensions> and
<maxFileSize> tags within the <settings> node of the XML configuration, the system will enforce these restrictions directly within the form component. This method leverages Magento??s built-in support for validation settings, which ensures that only files matching the specified criteria (.pdf files of 2 MB or smaller) are accepted by the form.
? uk.co.certification.simulator.questionpool.PList@78475b0d
: Magento??s documentation on UI components highlights how to enforce file type and size restrictions through XML configurations within <settings>, making it the standard and most efficient solution for this task.
Alternatives and Limitations:
Option A: Adding validation within the controller (UploadPdf) can provide additional server- side validation, but this does not prevent the user from selecting invalid files in the first place. Server-side validation alone lacks the user experience enhancement provided by client-side feedback.
Option B: Configuring a virtual type in di.xml for validation (e.g., setting allowedExtensions and maxFileSize within a custom uploader model) is effective for backend processing but is
not as straightforward for direct form validation within the admin UI. It complicates the implementation by requiring custom backend logic where native XML validation can suffice.
Option C provides a straightforward, maintainable, and user-friendly way to implement file validation directly in the form configuration. It reduces the need for custom controller or model logic and leverages Magento??s built-in form handling capabilities.

NEW QUESTION 92
A product has been added to the Adobe Commerce Store, and it contains a value for the custom product attribute. A merchant reports that the attribute value is not displayed in the Additional Information tab on the product detail page.
Which action will correct this problem?

A. The attribute must be moved to the specific group in the attribute set
B. The attribute property "Use in Product Tab' must be set to "yes"
C. The attribute property "Visible on Catalog Pages on Storefront" must be set to "yes".

Answer: C

Explanation:
The "Visible on Catalog Pages on Storefront" attribute property determines whether or not the attribute value is displayed in the Additional Information tab on the product detail page. If this property is set to "no", the attribute value will not be displayed.
For a custom product attribute to be displayed in the Additional Information tab on the product detail page in Magento, it needs to be visible on the catalog pages on the storefront. This visibility is controlled by the attribute property "Visible on Catalog Pages on Storefront". When this property is set to "yes", Magento includes the attribute in the Additional Information tab, making it visible to customers browsing the product. This setting ensures that only relevant and intended attributes are shown on the storefront, allowing for better product information management and customer experience.

NEW QUESTION 95
An Adobe Commerce developer has been asked to modify the PageBuilder slider content type to allow a new custom content type (other than slide) to be assigned as a child. The developer has already created the new content type called improved_slide in their module. They now need to create a new view/adminhtml/pagebuilder/content_type/slider. xml file in their module to allow the new content type to be a child of slider content types.
What is the correct xml to accomplish this?
A)

```
<type name="slider">
    <children>
        <child name="improved_slide" policy="allow"/>
    </children>
</type>
```

B)

```
<type name="slider">
    <allowed_descendants>
        <descendant name="improved_slide" />
    </allowed_descendants>
</type>
```

C)

```
<type name="slider">
    <arguments>
        <argument name="allowed_children" xsi:type="array">
            <item name="improved_slide" xsi:type="string">improved_slide</item>
        </argument>
    </arguments>
</type>
```

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

Answer: C

Explanation:
The correct answer is Option C. This XML configuration is the correct way to define allowed child content types for a slider content type in Magento's PageBuilder.
Magento PageBuilder Content Type Structure:
In PageBuilder, each content type can specify which other content types are allowed as children.
This is done by defining the allowed_children array within the content type??s XML configuration.
Analyzing Option C:
Arguments Definition: Option C uses the <arguments> node to define a new argument named allowed_children.
Array Structure: This argument is an array (xsi:type="array") that includes an item specifying the improved_slide as an allowed child with xsi:type="string".
This configuration is correct because it explicitly defines which child content types are allowed for the slider content type, adhering to Magento??s structure for allowed child
elements in PageBuilder.
Why Options A and B are Incorrect:
Option A: Uses a <children> node with policy="allow", which is not the standard way to define allowed children for PageBuilder content types. This format is incorrect and won??t be recognized by PageBuilder.
Option B: Uses <allowed_descendants>, which also doesn??t align with the way Magento??s PageBuilder expects child content types to be declared. The correct term is allowed_children, not allowed_descendants.
References:
Magento PageBuilder Development Guide - This guide provides insights into customizing PageBuilder and managing content types.
Configuring Content Types in PageBuilder - Documentation on how to define allowed children for custom content types.
Adobe Commerce PageBuilder Content Types XML Reference - Details on the correct XML structure for PageBuilder content types.
Option C??s configuration aligns with Adobe Commerce PageBuilder??s structure for defining which content types can be nested within another, making it the correct choice.

NEW QUESTION 100
An Adobe Commerce developer is tasked with creating a custom block that will be displayed on every page in the footer of the site.
After completing and optimizing the development, the developer notices that the block takes too much time to be generated on each page and decides to store it in the system cache after enabling it for all cache groups.
What would be the minimum requirement to achieve this?

A. Set a value for the cache_Lifetime data property of the block.
B. Set a value for cache_key data property of the block.
C. Set values for both cache_lifetime and cache_key data properties of the block.

Answer: A

NEW QUESTION 103
An Adobe Commerce Cloud developer wants to check the staging environment deployments history (i.e. branch, git, merge, sync). Where can the developer look up the history of the staging environment?

A. Project Web Interface
B. New Relic
C. Adobe Commerce admin panel

Answer: A

Explanation:
The Project Web Interface is the main dashboard for managing Adobe Commerce Cloud projects. This includes the ability to check the staging environment deployments history.
The developer can look up the history of deployments to the staging environment, including branch, git merge, and sync operations, in the Project Web Interface.
This interface provides adetailed log of all actions taken on the project, including deployments, enabling developers to track changes and troubleshoot issues that may arise.

NEW QUESTION 108
A Project Architect needs to add a new developer who needs to be able to push code in an Adobe Commerce Cloud project. No integration with a third-party repository provider is setup.
What two actions would be required to ensure the developer has access? (Choose Two.)

A. The developer's SSH public key must be added into a file named~/.ssh/authorized_keys
B. The developer needs to add SSH public key in the Cloud Account dashboard settings
C. The developer's email must be added under Users in the Cloud Project Web Ul
D. The Adobe Commerce admin user must be created and the developer's SSH public key must be added on their local system

Answer: BC

Explanation:
To ensure the developer has access to push code in an Adobe Commerce Cloud project, the developer??s email must be added under Users in the Cloud Project Web UI and the developer needs to add SSH public key in the Cloud Account dashboard settings. The Cloud Project Web UI is a web interface that allows managing and configuring Adobe Commerce Cloud projects and environments. The developer??s email must be added under Users to grant them access to the project and assign them a role and permissions. The Cloud Account dashboard settings is a web interface that allows managing and configuring Adobe Commerce Cloud accounts and SSH keys. The developer needs to add SSH public key in the settings to enable secure connection to the project and environments via SSH.
Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 112
A seller would like to offer an electronic version of an album by selling each song individually. Which layout can be used to customize a product page layout for this item?

A. catalog_product_view_type_downloadable
B. catalog_product_view_type_configurable
C. catalog_product_view_category

Answer: A

Explanation:
Thecatalog_product_view_type_downloadablelayout can be used to customize a product page layout for a downloadable product. This layout includes the product details, the product reviews, and the download links for the product's files.
For selling electronic versions of albums with individual songs, the downloadable product
type in Adobe Commerce is appropriate. To customize the product page layout specifically for downloadable items, the layout handlecatalog_product_view_type_downloadableis used. This layout handle allows developers to target downloadable products specifically and apply custom layouts or templates, making option A correct.

NEW QUESTION 116
A client would like to add an image icon in front of the telephone field to the shipping address form on a checkout page. What is the correct way to modify the UI component to set a custom template file for the field?
A)

```
...
<item name="telephone" xsi:type="array">
<arguments name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</arguments>
</item>
...
```

B)

```
...
<block name="telephone" xsi:type="array">
<arguments name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</arguments>
</block>
...
...
```

C)

```
...
<block name="telephone" xsi:type="array">
<arguments name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</arguments>
</block>
...
```

D)

```
...
<item name="telephone" xsi:type="array">
<item name="config" xsi:type="array">
<item name="elementTmpl" xsi:type="string">%Vendor_Module%/form/element/%your_template%</item>
</item>
</item>
...
```

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

Answer: B

Explanation:
To add an image icon in front of the telephone field in the shipping address form on the checkout page, the correct way is to modify the UI component by setting a custom template file for the field. The snippet in option B is the correct one because it uses the<block>element and sets theelementTmplto the custom template path within the argumentsnode under theconfignode. This modification will instruct Magento to use the specified custom template file for rendering the telephone field, allowing for the addition of an image icon in front of it.

NEW QUESTION 120
During database migration in the Adobe Commerce Cloud integration environment, a developer experienced a disk space error causing the database import to fail.
How would the developer fix this issue?

A. Increase the disk space of the database service.
B. Add a new database node and enable split database.
C. Change the database engine to PostgreSQL that has no disk space limit.

Answer: A

Explanation:
The developer can fix this issue by increasing the disk space of the database service. The database service is one of the services that run on the Adobe Commerce Cloud platform and provide functionality for the application. The database service uses MySQL as the database engine andstores data for products, customers, orders, etc. The disk space of the database service determines how much data can be stored and processed by the database. If the disk space is insufficient, the database import can fail with a disk space error. The developer can increase the disk space of the database service by modifying the .magento/services.yaml file and redeploying the environment. Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 125
An Adobe Commerce Cloud developer wants to be sure that, even after transferring database from Production to Staging, the payment configurations are still valid on the Staging environment.
What does the developer need to add to be sure that the configurations are always properly set?

A. Lines in the dedicated core_conf ig_data_stg table.
B. Project level environment variables.
C. Environment level environment variables.

Answer: C

Explanation:
The developer needs to add environment level environment variables to be sure that the payment configurations are always properly set on the Staging environment. Environment variables are configuration settings that affect the behavior of the Adobe Commerce Cloud application and services. Environment variables can be set at the project level or the environment level. Project level variables apply to all environments, while environment level variables override the project level variables for a specific environment. The developer can use environment level variables to customize the payment configurations for the Staging environment without affecting other environments. Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 129
Which two actions will the developer need to take to translate strings added in JS files? (Choose two.)

A. define (['jquery, 'mage/translate'), function ($, $t) {.„»;
B. $ trans(,<string>')
C. $.mage._('<string>);
D. translate('<string>');

Answer: AD

Explanation:
To translate strings added in JavaScript files in Adobe Commerce, developers need to use themage/translateRequireJS module along with the $.mage. ('<string>')function to mark strings for translation. This approach ensures that any text strings embedded within JavaScript code can be localized according to the store's current locale, providing a consistent and accessible user experience across different languages and regions. Themage/translatemodule and the$.mage. ()function work together to retrieve the corresponding translated strings from Magento's translation dictionaries, dynamically replacing the original text in the JavaScript code with the appropriate translations.

NEW QUESTION 132
An international merchant is complaining that changes are taking too long to be reflected on the frontend after a full product import.
Thinking it may be database issues, the Adobe Commerce developer collects the following entity counts:
• Categories: 900
• Products: 300k
• Customers: 700k
• Customer groups : 106
• Orders: 1600k
• Invoices: 500k
• Creditmemos: 50k
• Websites : 15
• Stores : 45
What is a probable cause for this?

A. The combination of the number of products, categories and stores is too bi
B. This leads to a huge amount of values being stored in the flat catalog indexes which are too large to be processed at a normal speed.
C. The combination of the number of orders, customers, invoices and creditmemos is too bi
D. This leads to a huge amount of values being stored in the customer grid index which is too large to be processed at a normal speed.
E. The combination of the number of products, customer groups and websites is too bi
F. This leads to a huge amount of values being stored in the price index which is too large to be processed at a normal speed.

Answer: A

Explanation:
The combination of a large number of products, categories, and stores directly affects the flat catalog indexing process. Each product needs to be indexed across all categories and stores, which exponentially increases the data size of the index tables. This can significantly slow down the frontend updates after a full product import due to the high volume of data that needs to be processed.
Impact on Flat Catalog and Indexing:
In Magento, flat catalog tables are used to optimize product data retrieval, especially in stores with a large catalog. However, with high numbers of products, categories, and stores, the flat catalog tables become massive, making them slower to update.
In this scenario, the system must create and maintain an entry for each product in each store across every category, leading to a substantial volume of data.
Why Option A is Correct:
This combination (products, categories, stores) is known to cause performance issues in indexing and data storage. It impacts the catalog indexes, which are critical for reflecting updates on the frontend.
Options B and C do not directly relate to the delay after product imports. Option B involves customer-related data, and Option C is related to price indexing, which affects a different area.
Optimization Recommendations:
Consider disabling the flat catalog and leveraging Elasticsearch for product and category search and filtering. Additionally, increasing server resources or partitioning the data could help optimize performance.
References:
Adobe Commerce documentation on Index Management Magento DevDocs on Performance Optimization

NEW QUESTION 136
How are multiple EAV attributes belonging to the same entity grouped in the database?

A. Based on the sizes of values they contain
B. Based on all numeric values being stored in one table while text values are stored in the other
C. Based on the types of values they contain

Answer: C

Explanation:
Multiple EAV attributes belonging to the same entity are grouped in the database based on their data types, such as datetime, decimal, int, text, or varchar. For example, all attributes with datetime values are stored in one table, while all attributes with text values are stored in another table.
The sizes or numeric/text values of attributes do not determine how they are grouped in the database.
Verified References: [Adobe Commerce Developer Guide - EAV data model]
Magento's EAV (Entity-Attribute-Value) model organizes attributes based on their data types to optimize storage and retrieval. Attributes are grouped into different tables based on whether they store values of types such as integer, varchar (text), decimal, datetime, etc. This organization allows Magento to efficiently manage the diverse data types associated with products, customers, and other entities, ensuring data integrity and optimizing database performance by using appropriate indexing and storage mechanisms for each data type.

NEW QUESTION 138
An Adobe Commerce developer creates a new website using a data patch. Each website will have unique pricing by website. The developer does not have visibility into the production and staging environments so they do not know what the configuration currently is.
How would they ensure the configuration is deployed and consistent across all environments?
A)

B)

C)

Create a custom module and override the value in config.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Store:etc/c
    <default>
        <catalog>
            <price>
                <scope>1</scope>
            </price>
        </catalog>
    </default>
</config>
```

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

Answer: A

Explanation:
The correct answer isOption A. This approach ensures that the configuration is set using the CLI with the --lock-config flag, which prevents the setting from being overridden by configuration changes in other environments.
? Understanding the Configuration Requirement:
? uk.co.certification.simulator.questionpool.PList@3d6d9941
? Using the CLI with --lock-config:
? Why Other Options Are Incorrect:
: Environment Configuration in Adobe Commerce- Details on managing environment-specific configurations and using the --lock-config flag.
Using the Magento CLI for Configuration Management- Documentation on setting and locking configuration values via CLI.
Option A is the best solution to ensure that the configuration is consistently deployed across all environments while protecting it from changes by leveraging the --lock-config option.

NEW QUESTION 142
On an Adobe Commerce Cloud platform, in which order does the ECE-Tools package apply patches?

A. 1. All required Magento patches included in the Cloud Patches for Commerce package.* 2. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.* 3. Selected optional Magento patches included in the Quality Patches Tool.
B. 1. All required Magento patches included in the Cloud Patches for Commerce package.* 2. Selected optional Magento patches included in the Quality Patches Tool.* 3. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.
C. 1. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.* 2. All required Magento patches included in the Cloud Patches for Commerce package.* 3. Selected optional Magento patches included in the Quality Patches Tool.

Answer: B

Explanation:
The order in which the ECE-Tools package applies patches is as follows:
? All required Magento patches included in the Cloud Patches for Commerce package.
? Selected optional Magento patches included in the Quality Patches Tool.
? Custom patches in the /m2-hotfixes directory in alphabetical order by patch name. The ECE-Tools package is a set of scripts and tools designed to manage and deploy Adobe Commerce Cloud projects. The Cloud Patches for Commerce package is a dependency of ECE-Tools that provides a set of required patches for Magento core issues that affect Adobe Commerce Cloud functionality. The Quality Patches Tool is an optional tool that allows developers to apply individual patches for specific Magento issues without waiting for a full product release. The /m2-hotfixes directory is a directory where developers can place their own custom patches for their Adobe Commerce Cloud projects. Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 144
When attempting operations that require lengthy processing, a merchant on Adobe Commerce Cloud receives a timeout error after 180 seconds.
How would the developer deal with this issue?

A. 1. Modify admin timeout into .magento.app.yamifile.* 2. Commit and push that code from the local environment.* 3. Move code to Production environment.
B. 1. In the Fastly Configuration section > Advanced Configuration.* 2. Set the Admin path timeout value in seconds.* 3. Save config and Upload VCL to Fastly.
C. 1. Modify admin timeout into app/etc/config.php file.* 2. Commit and push that code from the local environment.* 3. Submit a support ticket to apply the changes.

Answer: B

Explanation:
The developer can deal with this issue by modifying the admin path timeout value in seconds in the Fastly Configuration section > Advanced Configuration in the Admin Panel. Fastly is a cloud-based caching service that improves site performance and security for Adobe Commerce Cloud projects. Fastly has a default timeout value of 180 seconds for admin requests, which means that any request that takes longer than 180 seconds will be terminated and result in a timeout error. The developer can increase this value to allow longer processing time for admin requests without causing errors. The developer also needs to save the configuration and upload VCL to Fastly to apply the changes. Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 146
How would a developer access RabbitMQ data on an Adobe Commerce Cloud Production environment?

A. Using Project Web Interface
B. Using local port forwarding
C. Using RabbitMyAdmin

Answer: B

Explanation:
To access RabbitMQ data on an Adobe Commerce Cloud Production environment, you can use local port forwarding. This allows you to forward a port on your local machine to a port on the Production environment. This way, you can connect to RabbitMQ from your local machine.
A developer would access RabbitMQ data on an Adobe Commerce Cloud Production environment using local port forwarding. This is done via an SSH tunnel that securely forwards a port from the local machine to the RabbitMQ service on the cloud environment. RabbitMyAdmin (an option that does not exist) and the Project Web Interface do not provide direct access to RabbitMQ data.

NEW QUESTION 148
Which command can be used to display a full list of enabled and disabled Magento modules?

A. bin/magento module:all
B. bin/magento modulestatus
C. bin/magento module:show

Answer: B

Explanation:
The command to display a full list of enabled and disabled Magento modules isbin/magento module:status. This command provides a comprehensive overview of all modules within the Magento instance, categorizing them into enabled and disabled modules. This information is crucial for debugging and development purposes, as it allows developers to quickly understand which components of Magento are active and which are not, facilitating troubleshooting and configuration adjustments.

NEW QUESTION 149
An Adobe Commerce Developer is tasked with creating a custom form which submits its data to a frontend controller They have decided to create an action and have implemented the \Magento\Framework\App\Action\HttpPostActioninterface class, but are not seeing the data being persisted in the database, and an error message is being shown on the frontend after submission.
After debugging and ensuring that the data persistence logic is correct, what may be cause and solution to this?

A. Magento does not allow POST requests to a frontend controller, therefore, the submission functionality will need to be rewritten as an API endpoint.
B. The developer forgot to implement a validatePostDataQ method in their actio
C. They should implement this method: all non-validated POST datagets stripped out of the request and an error is thrown.
D. Form key validation runs on all non-AJAX POST requests, the developer needs to add the for_key to their requests.

Answer: C

Explanation:
According to the Magento Stack Exchange answer, form key validation is a security feature that prevents CSRF attacks by checking if the form key in the request matches the one generated by Magento. If the developer does not include the form_key in their custom form, the validation will fail and an error will be shown. Therefore, the developer needs to add the form_key to their requests by using <?= $block->getBlockHtml (??formkey??) ?> in their template file. Verified References:https://magento.stackexchange.com/questions/95171/magento-2-form- validation
In Adobe Commerce, when handling POST requests from forms on the frontend, form key validation is enabled by default as a security measure to prevent Cross-Site Request Forgery (CSRF) attacks. This validation checks that the form submission is coming from the same origin by including a unique token (form key) in the request. If this form key is missing or incorrect, the request will fail, and an error message may be shown on the frontend.
In this scenario:
? Since the developer has used
\Magento\Framework\App\Action\HttpPostActionInterface, which is appropriate for handling POST requests, it??s likely that the error they encounter is due to missing form key validation.
? The solution is to ensure that the form includes a hidden input field for the form key. Adobe Commerce automatically adds this key in forms if you use the \Magento\Framework\Data\Form\FormKey model to get the form key value. To implement this:
? Ensure the form includes the form key:
<input name="form_key" type="hidden" value="<?= $block->escapeHtml($block-
>getFormKey()) ?>" />
? The form key should also be included in the POST data sent to the controller. If it??s missing, Adobe Commerce will not process the request.
Additional Resources:
? Adobe Commerce Developer Guide: Form Key
? Magento 2.4 Form Key and CSRF Protection

NEW QUESTION 152
Which theme directory contains the static files that can be loaded directly?

A. web
B. preprocessed
C. assets

Answer: A

Explanation:
Thewebdirectory contains the static files that can be loaded directly. This directory includes files such as CSS, JavaScript, and images.
In Adobe Commerce themes, thewebdirectory is used to store static files such as CSS, JavaScript, images, and fonts. These files can be loaded directly by the browser. The preprocessedandassetsdirectories do not exist as standard directories in the theme structure for containing directly loadable static files.

NEW QUESTION 153
On an Adobe Commerce Cloud platform, what type of environment will be provisioned when launching the CLI for Commerce command magento-cloud environment:branch
<environment-name> <parent-environment-id>?

A. An empty integration environment without any code or database.
B. An integration environment with fresh Adobe Commerce Cloud installation.
C. An integration environment with the code and database from the parent environment.

Answer: C

Explanation:
The type of environment that will be provisioned when launching the CLI for Commerce command magento-cloud environment:branch <environment-name> <parent- environment-id> is an integration environment with the code and database from the parent environment. Integration environments are temporary environments that are used for testing and development purposes on the Adobe Commerce Cloud platform. They can be created from any branch of code and have their own dedicated database and services. When creating an integration environment using the CLI for Commerce command, the code and database from the parent environment are copied to the new integration environment, creating an exact replica of the parent environment. Verified References: [Magento 2.4 DevDocs]

NEW QUESTION 157
An Adobe Commerce developer is asked to create a new payment method for their project. This project has administrators who use the backend to manage customer information and occasionally place orders. When testing the new payment method on the frontend everything worked as expected, however, the payment method is missing in the admin.
What is a possible reason for this?

A. In the module di.xml, there were no default 3DS verification types configured as a VirtualType.
B. In the module config.xmi, the boolean value for can_capture was set to false.
C. In the module config.xmi, the node can_use_internal was not set to true.

Answer: C

Explanation:
For a payment method to be available in the admin panel (backend), the configuration must explicitly allow its internal use. This is controlled by the can_use_internal flag in config.xml.
? Configuration for Admin Use:
? uk.co.certification.simulator.questionpool.PList@31da258e
? Why Option C is Correct:
: Adobe Commerce DevDocs onPayment Method Configuration Magento Developer Guide onCustom Payment Method

NEW QUESTION 161
What are the only writeable folders in the application root on a remote Adobe Commerce Cloud project?
A)

B)

C)

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

Answer: B

Explanation:
For an Adobe Commerce Cloud project, the only writeable folders in the application root on a remote environment are essential for the application to run correctly and store temporary and dynamic data. Among the options given, Option B lists directories that are typically writable:m2-hotfixes,var,pub/static, andapp/etc. Them2-hotfixes directory is specifically for Magento Commerce Cloud and is used for applying hotfixes that are executed during the build phase. Thevardirectory contains various logs, sessions, and reports. Thepub/staticdirectory holds the compiled static view files, andapp/etccontains configuration files that can be modified by the application at runtime.

NEW QUESTION 165
A merchant wants to include taxes In an Adobe Commerce store. Which option can have a tax class assigned to it?

A. Order
B. Shipping
C. Category

Answer: B

Explanation:
According to the Adobe Commerce User Guide, a tax class can be assigned to either a product or a customer group in Adobe Commerce. A product tax class determines how a product is taxed, while a customer tax class determines how a customer is taxed based on their location and group membership. Shipping is considered as a product tax class in Adobe Commerce, and it can be assigned to different shipping methods or rates. The other options are not valid for assigning a tax class.
In Adobe Commerce, tax classes can be assigned to products and shipping. Categories, however, do not have tax classes assigned to them directly. Tax classes applied to shipping allow merchants to specify how taxes should be calculated for shipping costs, making option B the correct answer. Orders and categories do not have direct associations with tax classes in the same way products and shipping do.

NEW QUESTION 166
An Adobe Commerce Developer wishes to add an action to a pre-existing route, but does not wish to interfere with the functionality of the actions from the original route.
What must the developer do to ensure that their action works without any side effects in the original module?

A. In the route declaration, use the before or after parameters to load their module in before or after the original module.
B. Inject the new action into the standard router constructor's $actiomist parameter.
C. Add the action into to the controllers/front_name/ in My.Module, Magento will automatically detect and use it.

Answer: C

Explanation:
In Magento 2, to add a new action to a pre-existing route without interfering with the existing functionality, the new action should be placed in the same directory structure under the new module??s controller namespace. Magento's autoloading mechanism will automatically detect and include it alongside the original module's actions. Here's how you can achieve this:
? Directory Structure: Ensure that your new module's controller directory structure
mirrors that of the original module.
? Controller Action: Define the new action within the appropriate directory.
For example, if you want to add a new action to thecatalogroute inMagento_Catalog:
? Create a directory structureapp/code/My/Module/Controller/Catalog/.
? Add your new action class in this directory, for example: namespace My\Module\Controller\Catalog;
use Magento\Framework\App\Action\Action; use Magento\Framework\App\Action\Context;
class NewAction extends Action

```
{
public function construct(Context $context)
{
parent:: construct($context);
}
public function execute()
{
// Your custom logic here
}
}
```

Router Configuration: Magento automatically includes this action when the route matches. By following this method, you ensure that your new action is added seamlessly without modifying the original module or causing conflicts. Magento's router will include and recognize your action based on the directory and namespace conventions.
Sources:
? Fundamentals of Magento 2 Development documents .
? Magento 2 official developer documentation.

NEW QUESTION 171
When checking the cron logs, an Adobe Commerce developer sees that the following job occurs daily: main.INFO: Cron Dob inventory_cleanup_reservations is successfully finished. However, the inventory_reservation table in the database is not emptied. Why are there records remaining in the inventory_reservation table?

A. Only reservations matching canceled orders are removed by the cron job.
B. Only reservations no longer needed are removed by the cron job.
C. The "Auto Cleanup" feature from Multi Source Inventory was disabled in configuration.

Answer: B

Explanation:
The reason why there are records remaining in the inventory_reservation table is that only reservations no longer needed are removed by the cron job. The inventory_reservation table tracks the quantity of each product in each order and creates a reservation for each product when an order is placed, shipped, cancelled or refunded. The initial reservation has a negative quantity value and the subsequent reservations have positive values. When the order is complete, the sum of all reservations for the product is
zero. The cron job removes only those reservations that have a zero sum from the table, leaving behind any reservations that are still needed for incomplete orders. Verified References: [Magento 2.4 DevDocs] [Magento Stack Exchange]

NEW QUESTION 174
A developer found a bug inside a private method of a third party module class. How can the
developer override the method?

A. Create a custom class with corrected logic, and define the class as preference in the preferences.xml.
B. Create a custom class with the corrected logic, and define the class as a preference for original one in the di xml.
C. Create a plugin, implement correct logic in the after" method, and then define the plugin in the di.xml.

Answer: B

Explanation:
To override a private method in a third-party module class, the most effective approach is to use a preference. This involves creating a custom class with the corrected logic and then defining this class as a preference for the original one in thedi.xmlfile. Plugins cannot be used to override private methods, final methods, final classes, or classes created without dependency injection. Therefore, the preference mechanism, which allows for the substitution of the entire class, becomes the viable method to override a private method and modify its behavior.

NEW QUESTION 179
An Adobe Commerce developer wants to create a product EAV attribute programmatically which should appear as WYSIWYG in the admin panel. They have made sure that wysiwyg_enabled has been set to true, however, the attribute is not appearing as WYSIWYG in the admin panel.
What would be a possible reason?

A. The is_html_allowed_on_front Option iS Set tO false.
B. The input type is not set to text.
C. The input type is not set to textarea.

Answer: C

Explanation:
Theinput_typeattribute of a product EAV attribute specifies the type of input field that will be used to enter the value of the attribute in the admin panel. Thetextareainput type is used for WYSIWYG fields. If theinput_typeattribute is not set totextarea, then the attribute will not appear as WYSIWYG in the admin panel.
To fix this, the developer should set theinput_typeattribute totextarea.

NEW QUESTION 182
An Adobe Commerce developer is working on a custom gallery extension.
The module uses the Magento\catalog\Model\iinageUploader class for image uploading. The admin controller for custom image uploads is Vendor\CustomGallery\Controller\Adminhtml\Image\Upload.
The images need to be stored in different basePath and baseTmpPath than the default ones.
How can the default imageuploader class be extended and used without affecting the other modules that are already using it?
A)

1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
2. Inject the virtual type `Vendor\CustomGallery\GalleryImageUpload` into admin controller:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader"
>
    <arguments>
        <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
        <argument name="basePath" xsi:type="string">customgallery/images</argument>
    </arguments>
</virtualType>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
    <arguments>
        <argument name="imageUploader" xsi:type="object">
            Vendor\CustomGallery\GalleryImageUpload
        </argument>
    </arguments>
</type>
```

B)

1. Configure the `basePath` and `baseTmpPath` of `Magento\Catalog\Model\ImageUploader`.
2. Inject the type `Magento\Catalog\Model\ImageUploader` into admin controller:

```
<type name="Magento\Catalog\Model\ImageUploader">
    <arguments>
        <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
        <argument name="basePath" xsi:type="string">customgallery/images</argument>
    </arguments>
</type>

<type name="Vendor\CustomGallery\Controller\Adminhtml\Image\Upload">
    <arguments>
        <argument name="imageUploader" xsi:type="object">
            Magento\Catalog\Model\ImageUploader
        </argument>
    </arguments>
</type>
```

C)

1. Create a Virtual Type and configure the `basePath` and `baseTmpPath`.
2. Add virtual type `Vendor\CustomGallery\GalleryImageUpload` as a preference for `Magento\Catalog\Model\ImageUploader:

```
<virtualType
  name="Vendor\CustomGallery\GalleryImageUpload"
  type="Magento\Catalog\Model\ImageUploader">
    <arguments>
        <argument name="baseTmpPath" xsi:type="string">customgallery/tmp/images</argument>
        <argument name="basePath" xsi:type="string">customgallery/images</argument>
    </arguments>
</virtualType>

<preference
    for="Magento\Catalog\Model\ImageUploader"
    type="Vendor\CustomGallery\GalleryImageUpload"
/>
```

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

Answer: A

Explanation:
By explicitly injecting the virtual type into the controller, you localize the configuration changes to only the desired functionality. This approach is efficient and maintains module independence, a key principle in Magento development.
Options B and C are incorrect for the following reasons:
Option Bdirectly modifies Magento\Catalog\Model\ImageUploader with the new paths. This change will affect all usages of the ImageUploader class across the site, which contradicts the requirement to avoid impacting other modules.
Option Cinvolves creating a virtual type and then setting it as a preference. However, using a preference would replace all instances of ImageUploader across the entire Magento application, leading to the same issue as Option B.

NEW QUESTION 183

How would a developer access RabbitMQ data on an Adobe Commerce Cloud Production environment?

A. Using Project Web Interface
B. Using local port forwarding
C. Using RabbitMyAdmin

Answer: B

Explanation:
The way a developer would access RabbitMQ data on an Adobe Commerce
Cloud Production environment is by using local port forwarding. This method allows the developer to connect to the RabbitMQ service instance through an SSH tunnel and access the RabbitMQ Management UI from a web browser. The developer needs to use the magento-cloud ssh command to establish the SSH connection and the
$MAGENTO_CLOUD_RELATIONSHIPS variable to retrieve the RabbitMQ connection details and login credentials.

NEW QUESTION 185
What is the correct way to inject CMS block in a layout?

A. <block class="Magento\Cms\Block\Block" name="blockjdentifier"> <arguments>
q<argumentname="block_id"xsi:type="string">my_cms_block_identifier</argument></arguments> </block>
B. <block class="Magento\Cms\Block\Block" name="block_identifier"> q<actionmethod="setBlock'>my_cms_block_identifier</action> </block>
C. <referenceBlock name="content"> <block class="Magento\Cms\Block\Block" name="block_identifier' identifier="my_cms_block_Identrfier" /> </referenceBlock>

Answer: A

Explanation:
The correct way to inject a CMS block into a layout in Adobe Commerce is by using the<block>element with the classMagento\Cms\Block\Blockand specifying the block identifier through an<argument>element with the name "block_id". This is shown in option A. The<block>tag defines the block class and name, and the<arguments>tag contains child<argument>tags for each argument, where the "block_id" argument specifies the identifier of the CMS block to be injected.

NEW QUESTION 187
An Adobe Commerce developer has been tasked with applying a pricing adjustment to products on the website. The adjustments come from a database table. In this case, catalog price rules do not work. They created a plugin for getPrice on the price model, but the layered navigation is still displaying the old price.
How can this be resolved?

A. Create an implementation for \Magento\Catalog\Hodel\Product\PriceModifierInterf ace.
B. Create an after plugin On \Magento\Catalog\Api\Data\BasePriceInterface:: getPrice.
C. Create a plugin for\Magento\Catalog\Model\Indexer\Product\Price::executeRow.

Answer: C

Explanation:
The developer can resolve this issue by creating a plugin for theMagento\Catalog\Model\Indexer\Product\Price::executeRow()method. This method is responsible for updating the product price index.
The plugin can be used to add the pricing adjustment from the database to the product price index. Once the product price index is updated, the layered navigation will display the correct price.
Here is an example of a plugin for theexecuteRow()method: PHP

```
class MyPlugin
{
public function executeRow(
\Magento\Catalog\Model\Indexer\Product\Price $subject,
\Magento\Catalog\Model\Product $product, array $data
) {
$adjustment = $this->getAdjustment($product);
$product->setPrice($product->getPrice() + $adjustment);
}
private function getAdjustment(Product $product)
{
$adjustment = $product->getData('adjustment');
if (!is_numeric($adjustment)) { return 0;
}
return $adjustment;
}
}
```

This plugin will add theadjustmentdata from the product to the product price index. Once the product price index is updated, the layered navigation will display the correct price.

NEW QUESTION 188
A developer is creating a class \Vendor\Module\Model\MyModeL How should that class be defined as transient in di.xml?

A. <type name="\Vendor\Module\Model\MyModer transient="true7>
B. <type name="\Vendor\Module\Model\MyModer singleton='false7>
C. <type name='\Vendor\Module\Model\MyModer shared="false7>

Answer: C

Explanation:
To define a class as transient in Magento'sdi.xml, the correct approach is to set thesharedattribute to "false" for that class. This tells Magento's object manager not to use the singleton pattern for this class, meaning a new instance will be created each time the class is requested. This is particularly useful for classes that should not maintain state across different areas of the application or during a single request lifecycle.

NEW QUESTION 193
Which two methods add sorting to collections inherited from the \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollection class? (Choose two.)

A. setOrder
B. setSorting
C. addSorting
D. addOrder

Answer: AD

Explanation:
The two methods that add sorting to collections inherited from the \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollection class are setOrder and addOrder. These methods allow adding one or more order clauses to a collection query.
The setSorting and addSorting methods do not exist in Adobe Commerce. Verified References: [Adobe Commerce Developer Guide - Collections]
In Magento 2, collections inherited from \Magento\Framework\Model\ResourceModel\Db\Collection\AbstractCollectionclass can be sorted using thesetOrderandaddOrdermethods. ThesetOrdermethod is used to set the order for a field in the collection, specifying the field by which to sort and the direction of the sorting (ASC or DESC). TheaddOrdermethod is similar but allows for adding multiple sorting orders to the collection, enabling more complex sorting scenarios. There are nosetSortingoraddSortingmethods in the standard Magento 2 collection classes.

NEW QUESTION 196
An integration named Marketing is created on the Adobe Commerce instance. The integration has access on Magento_Customer:: customer resources and the access token is xxxxxx.
How would the rest API be called to search the customers?

A. Using the integration access token as Bearer:curl -X GET https://magentourl/rest/Vl/customers/search?searchCriteria... -H 'Authorization: Bearer XXXXXX'
B. Passing integration name and access token as http auth credentials: curl -X GET https://Marketing:XXXXXX(Slmagentourl/rest/Vl/customers/search?5earchCriteria . . .Using integration name as username and access token as password, get the admin token (yyyyyy) via:curl -X POST https://magentourl/rest/Vl/integration/admin/token -d '{"username":"Marketing", "password":"XXXXXX"}' -H 'Content-
C. Type: application/json'Use the admin token as Bearercurl -X GET https://magentourl/rest/Vl/customers/search?searchCriteria... -H 'Authorization: Bearer YYYYYY'

Answer: A

Explanation:
When using an integration token to access Magento??s REST API, you can authenticate requests by including the token in the Authorization header as a Bearer token. This allows the system to recognize the permissions assigned to the integration and grant access to the specified resources.
? Using the Access Token as Bearer Token:
? uk.co.certification.simulator.questionpool.PList@69ee4bb7
? Why Option A is Correct:
? Example Command: curl -X GET "https://magentourl/rest/V1/customers/search?searchCriteria[filterGroups][0][filters][0][field] =email&searchCriteria[filterGroups][0][filters][0][value]=example@example.com" -H "Authorization: Bearer XXXXXX"
References:
Adobe Commerce REST API documentation on Authentication Magento Integration Tokens Guide on Using Tokens

NEW QUESTION 201
What does a URL Rewrite do?

A. It updates the URL that is stored on the server.
B. It changes the way a URL appears in the browser
C. It updates the URL to a domain that is not being Indexed.

Answer: B

Explanation:
A URL Rewrite in Magento changes the way a URL appears in the browser. This is particularly useful for improving the readability and SEO of a URL. For example, a URL rewrite can be used to transform a long and complex URL into a shorter and more user-friendly version. It's important to note that while a URL rewrite changes the URL's appearance in the browser, it doesn't change the actual location of the resource on the server. This distinction is crucial for understanding how Magento handles URL rewrites and redirects, facilitating a more intuitive navigation structure within the store without altering the underlying server resources.

NEW QUESTION 206
A merchant is experiencing performance issues on integration environments of their Adobe Commerce Cloud Pro plan and wants to upgrade to Enhanced Integration Environments.
What are the steps necessary prior to redeploying in order to upgrade to Enhanced Integration Environments?

A. 1. Limit the number of Integration branches to two* 2. Submit a support ticket requesting the upgrade
B. 1. Limit the number of Integration branches to three* 2. Set the ENV.ENVIRONMENT in .magento.env.yaml to ENHANCEDJNTEGRATION
C. 1. Limit the number of Integration branches to four* 2. Configure integration environments in the cloud GUI and set the Enhanced switch to On

Answer: A

Explanation:
Upgrading to Enhanced Integration Environments in Adobe Commerce Cloud requires specific steps to ensure that the environment is prepared for the upgrade, which includes managing integration branch limits and coordinating with Adobe support.
? Limiting Integration Branches:

? uk.co.certification.simulator.questionpool.PList@6a215712
? Submitting a Support Ticket:
? Why Option A is Correct:
: Adobe Commerce Cloud documentation onEnhanced Integration Environments

NEW QUESTION 210
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

AD0-E724 Practice Exam Features:

* AD0-E724 Questions and Answers Updated Frequently

* AD0-E724 Practice Questions Verified by Expert Senior Certified Staff

* AD0-E724 Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* AD0-E724 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year