DUMPSBOSS.

Magento Certified Professional Cloud Developer Exam

Magento Magento-Certified-Professional-Cloud-Developer

Version Demo

Total Demo Questions: 15

Total Premium Questions: 153

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Cloud Architecture	16
Topic 2, Cloud Deployment	69
Topic 3, Cloud Development	26
Topic 4, Cloud Operations	34
Topic 5, Cloud Security	7
Topic 6, Mix Questions	1
Total	153

QUESTION NO: 1

Magento Support advises you lo upgrade to the latest release of ece-tools matching the project's Magento Commerce version 2 3 1 How do you do that?

A. Require the latest ece-tools version compatible with Magento Commerce 2.3.1 using Composer, for example: composer require magento/ece-tools: --update-with-dependencies

B. Clone the repository github.com/magento/ece-tools and copy the ate/folder to vendor/magento/ece-tools/src

C. Run the command composer update magento/ece-tools

D. Run the Command ece-tools self-upgrade

ANSWER: A

Explanation:

Use Composer to explicitly require the latest `magento/ece-tools` version that is compatible with the project's Magento Commerce 2.3.1 release, and update its dependencies. The ece-tools package is a Composer dependency, so its version must be managed through the project's `composer.json` and lock file rather than by copying files or running a standalone tool self-update command. Requiring the intended compatible package version ensures that Composer resolves the package and all related dependencies consistently, records the result in `composer.lock`, and allows the same build to be reproduced in integration, staging, and production environments. The version must be selected from the ece-tools compatibility and release information applicable to the project's Adobe Commerce release; it is not derived by using the Magento application version directly as the ece-tools package version. Adobe's documented ece-tools upgrade workflow uses `composer require` to set the desired package version, followed by dependency installation or update. See the Adobe Commerce ece-tools package update documentation and the ece-tools package release notes for supported versions and release-specific requirements.

QUESTION NO: 2

You use configuration management and need to change a store configuration value only in Production without changing the value committed in `app/etc/config.php`.

Which two statements are correct? (Choose two.)

A. Create a Production-scoped variable using the `CONFIG__DEFAULT__` naming convention

B. Use double underscores in the variable name in place of slashes in the configuration path

C. Edit `app/etc/env.php` directly on the Production environment

D. Update the value in Admin and rely on the database change to survive future deployments

E. Add the Production value to `app/etc/config.php` after every Integration merge

ANSWER: A B

Explanation:

An environment variable that follows the `CONFIG__DEFAULT__` naming convention can set a Commerce configuration value at the default scope during deployment. The remainder of the variable name represents the configuration path with double underscores in place of slashes.

Environment variables can be scoped to Production through the Cloud Console or Cloud CLI. This permits a Production-specific override without committing the setting to `app/etc/config.php`. The configuration dump is source-controlled and is normally promoted with code, so modifying it for a Production-only requirement would also affect promoted environments unless separately maintained.

See Adobe's configure environment variables documentation and configuration management documentation.

QUESTION NO: 3

You want to improve the ability to monitor Production deployments by setting up an email notification system How do you achieve this?

A. Enable Deployment Notifications under Configure > Settings in the Project Web Interface

B. Build a custom module which hooks into the deployment phase and sends emails in real-time

C. Configure log notifications in the.magento.env.yaml file

D. Use the magento-cloud environment:deploy:email command to enable email notifications

ANSWER: A

Explanation:

Enable Deployment Notifications in the Project Web Interface under Configure > Settings. Adobe Commerce on cloud infrastructure provides deployment notifications as a built-in project-level capability, allowing project users to receive email updates about deployment activity without changing application code or adding custom deployment hooks. This is appropriate for monitoring production because it is managed through the cloud project configuration and is designed specifically to notify stakeholders when deployment events occur. The notification configuration can be enabled from the project's settings in the Cloud Console, where authorized users manage project-wide operational settings. Using this supported feature keeps deployment monitoring aligned with the platform's standard workflows and avoids introducing a maintenance burden into the application's build and deployment process. Adobe's cloud documentation describes configuring project settings through the Project Web Interface and identifies deployment notifications as an available setting. See the Adobe Commerce on cloud infrastructure project documentation and the project settings configuration guidance.

QUESTION NO: 4

You need to specify the admin password using an environment variable. You have created an environment variable env:ADMIN_PASSWORD with a valid password. When attempting to log in to the Magento Admin it is not accepting the new password.

How do you correct the environment variable?

A. The Sensitive option is checked by default and must be disabled

B. The env:ADMIN_PASSWORD variable can only be used for an initial installation

C. The environment variable should not have the env: prefix

D. The ADMIN_PASSWORD variable should be configured via .magento.env.yaml

ANSWER: B

Explanation:

The env:ADMIN_PASSWORD variable can only be used for an initial installation. Adobe Commerce on cloud infrastructure reads `ADMIN_PASSWORD` as an installation setting when the application is first installed and the Admin user is created. It does not operate as an ongoing credential-management setting and does not reset the password for an already-created Admin account on later deployments. Therefore, supplying a different value after installation will not make that new value valid for an existing Admin login.

To regain access or set a new credential for an existing installation, reset the relevant Admin user's password through an appropriate administrative recovery method, such as the Commerce command-line tools, rather than expecting the deployment environment variable to overwrite it. The environment variable remains useful when defining the credentials for a fresh installation, particularly when supplied securely through the Cloud environment configuration.

Adobe's Cloud configuration-variable reference identifies `ADMIN_PASSWORD` as the password used to create the Admin user during installation. See the ADMIN_PASSWORD variable documentation and the Commerce Admin password guidance.

QUESTION NO: 5

You need to specify the admin password using an environment variable.You have created an environment variable

env:ADMIN_PASSWORDwith a valid password When attempting to log in to the Magento Admin it is not accepting the new password

How doyou correct the environment variable?

A. The Sensitive option is checked by default and must be disabled

B. The env:ADMIN_PASSWORD variable can only be used for an initial installation

C. The environment variable should not have the env: prefix

D. The ADMIN_PASSWORD variable Should be Configured Via .magento.env.yaml

ANSWER: B

Explanation:

The `env:ADMIN_PASSWORD` variable can only be used for an initial installation. In Adobe Commerce on cloud infrastructure, `ADMIN_PASSWORD` is an installation configuration value consumed when the application is first installed and the initial administrator account is created. Updating that variable after Magento has already been installed does not reset or replace the password stored for the existing Admin user. Therefore, a valid new value in the environment configuration will not be accepted at the Admin login screen for that established account.

To change an existing administrator's credentials, use an appropriate account-management process, such as changing the password from the Admin for an authenticated user, using the Magento CLI to create or update an administrator account where applicable, or following the supported password-reset workflow. The environment variable remains useful when provisioning a new installation, where deployment tooling passes the installation settings to Magento as the initial database and Admin configuration are created. Adobe's Cloud documentation identifies `ADMIN_PASSWORD` as an environment variable used to set the administrator password during installation. See the Adobe Commerce environment variables documentation and the Adobe Commerce installation documentation.

QUESTION NO: 6

You cloned the Integration branch to your local environment and Imported the database dump from Integration. You performed composer install and bin/magentosetup:install.

While placing an order using PayPal Express, the following error occurs:

PayPal gateway has rejected request. Security header is not valid (#10002: Security error).

What Is the cause of this error?

Paypal gateway has rejected request, Securityheader is not valid (#10002: Security error).

What is the cause of this error?

A. A new encryption key has been created on the setup:install action.

B. The folder var/session has no write permissions for the web server user.

C. Paypal Sandbox API credentials are not valid for the local environment.

D. The PHP extension mcrypt has not been installed locally.

ANSWER: A

Explanation:

A new encryption key has been created on the setup:install action. Adobe Commerce/Magento encrypts sensitive configuration values, including payment-gateway credentials, in the database using the installation's encryption key. The database dump taken from Integration therefore contains PayPal credential values encrypted with Integration's key. Running `bin/magento setup:install` locally creates a new installation configuration and encryption key. That local key cannot correctly decrypt the credential values stored in the imported Integration database. Consequently, Magento sends unusable or improperly decrypted API authentication data when it builds the PayPal Express request, and PayPal returns the security-header error.

For a database copied between environments, the encryption key must be preserved or the affected encrypted configuration must be re-entered and saved using the destination environment's key. Adobe's configuration guidance explains that the encryption key protects sensitive data and must be handled carefully during migration or restoration: Encryption key documentation. The installation command's role in creating and configuring a Magento instance is documented at Advanced installation.

QUESTION NO: 7

You cloned the Integration branch to your local environment and imported the database dump from Integration. You performed composer install and bin/magento setup:install. While placing an order using PayPal Express, the following error occurs:

PayPal gateway has rejected request. Security header is not valid (#10002: Security error).

What is the cause of this error?

A. A new encryption key has been created on the setup:install action.

B. The folder var/session has no write permissions for the web server user.

C. PayPal Sandbox API credentials are not valid for the local environment.

D. The PHP extension mcrypt has not been installed locally.

ANSWER: A

Explanation:

A new encryption key has been created on the setup:install action. Magento stores sensitive configuration values, including payment gateway credentials, in encrypted form in the database. The database dump imported from Integration therefore contains PayPal Express credentials encrypted with the Integration environment's cryptographic key. Running `bin/magento setup:install` locally creates a new application configuration and generates a local encryption key in `app/etc/env.php`. That new key does not match the key that encrypted the values in the imported database.

When Magento retrieves the PayPal configuration for the payment request, it cannot correctly decrypt the stored API username, password, signature, or certificate values. As a result, PayPal receives invalid security-header data and returns error `10002`. A database copied between environments must be accompanied by the appropriate encryption key when encrypted configuration values need to remain usable, or the sensitive values must be re-entered for the destination environment. Adobe documents the encryption-key role and key rotation process in Encryption key management, and describes environment configuration in env.php reference.

QUESTION NO: 8

You created a custom module that is not functioning as expected on your Integration environment. You would like to debug the code using Xdebug. XdeDug is missing from the output of the command php -m

What are two ways to load the Xdebug module?

Choose 2 answers

A. Create a custom php.ini that includes the extension

B. You log a support request as php modules can only be installed by support

C. Add xdebug into the .magento.app.yaml file under the extensions section

D. In the phpserver folder, create a folder conf.d and add a file ext-xdebug.ini that includes the extension

ANSWER: C D

Explanation:

Xdebug must be loaded as a PHP extension before its debugging settings can be used. Adding `xdebug` under the `runtime.extensions` section of `.magento.app.yaml` is the declarative Adobe Commerce on cloud infrastructure method. This configuration is committed to source control and applied when the Integration environment is deployed, making the Xdebug extension available to that environment's PHP runtime.

A second supported configuration-based approach is to provide an extension INI file in `phpserver/conf.d`, such as `ext-xdebug.ini`, that loads the Xdebug extension. PHP reads files from its additional configuration directory during startup, so the extension is enabled when the relevant PHP service is restarted or the environment is redeployed. This approach is useful when the project's PHP-server configuration is managed through repository files.

After enabling the module, Xdebug-specific settings, such as the debug mode, client host, and client port, must also be configured for the debugger client being used. Adobe's cloud documentation describes enabling and configuring Xdebug for cloud environments, including the application-configuration approach. See Adobe Commerce: Configure Xdebug and Adobe Commerce application properties.

QUESTION NO: 9

While launching a site migrated from Magento 1 you are instructed to change the website CNAMErecord in your DNS provider by Magento for go live

What is the purpose of setting this record9

A. Setting this record causes sent email to be property authenticated and not show in junk folders

B. Setting this record enables the page caching service for your site

C. Setting this record is needed on Pro to allow upsizing servers without downtime

D. Setting this record reduces the time it takes for customers to start seeing your Magento 2 site

ANSWER: B

Explanation:

Setting this record enables the page caching service for your site. For an Adobe Commerce on cloud infrastructure production launch, the storefront domain's DNS CNAME is configured to point traffic to the Fastly service endpoint associated with the project. Fastly acts as the public-facing CDN and reverse proxy before requests reach the Commerce application. This routing is what allows Fastly to serve eligible cached storefront responses from edge locations, reducing origin load and improving the speed and consistency of content delivery for shoppers. It also ensures that requests use the production delivery path configured for the cloud environment rather than going directly to an application origin. DNS changes can take time to propagate according to the record's TTL, so the CNAME should be planned and validated as part of the go-live procedure. Adobe's cloud documentation describes configuring a CNAME for production domains and then verifying Fastly service operation; its Fastly documentation also explains the CDN's role in caching and delivering Commerce content. See Adobe Commerce Fastly custom cache configuration and Adobe Commerce cloud launch checklist.

QUESTION NO: 10

You are configuring scheduled tasks for an Adobe Commerce on cloud infrastructure project. A custom command must run every five minutes, and the standard Commerce cron process must continue to run.

Which two actions should you take? (Choose two.)

A. Add the custom command and its five-minute schedule under `crons` in `.magento.app.yaml`

B. Keep the standard `bin/magento cron:run` command configured under `crons`

C. Add the command to the `post_deploy` hook so it runs every five minutes

D. Create a Linux crontab entry by connecting to the environment over SSH

E. Add the schedule to `.magento.env.yaml` under the `deploy` stage

ANSWER: A B

Explanation:

Scheduled commands are configured in the `crons` section of `.magento.app.yaml`. A custom cron definition can use a five-minute schedule such as `*/5 * * * *` and execute the required command from the deployed application.

The standard Commerce cron command must also remain configured. Commerce cron processes scheduled operations such as index updates, email processing, catalog rules, and other queued tasks. Replacing it with only a custom command would prevent those standard scheduled tasks from being processed.

Adobe documents cron configuration in the crons property documentation and scheduled task processing in the Configure and run cron documentation.

QUESTION NO: 11

On a project that deploys static content during the build phase, a merchant states the deploy phase is still taking too long. You consider turning off JavaScript minification to reduce the build time.

Besides reducing the build phase time, what two consequences does turning off JavaScript minification have? (Choose two.)

A. The deploy artifact size will be decreased because of the larger JavaScript can be symlinked

B. Browsing the store will be slower because larger JavaScript files have to be downloaded

C. The deploy phase will be shorter because JavaScript can be symlinked from init instead of copied

D. The build phase will be longer because the additional pass of JavaScript merging

ANSWER: B C

Explanation:

Turning off JavaScript minification means the storefront serves the original, uncompressed JavaScript rather than the smaller minified output. Consequently, browser requests transfer more bytes, which can increase download and parsing work and make browsing the store slower, particularly on slower connections or pages that require many JavaScript assets. Adobe Commerce identifies JavaScript minification as a frontend optimization intended to reduce the size of JavaScript delivered to clients.

When static content is deployed during the build phase, avoiding generated minified JavaScript also permits the applicable static JavaScript files to be symlinked from the initialized static-content location rather than copied into the release. Creating those links reduces filesystem work in the deploy phase, shortening that phase. The trade-off is intentional: faster build/deploy processing comes at the cost of larger JavaScript payloads for shoppers. See Adobe Commerce's JavaScript optimization settings and its Cloud deployment variables documentation for the related frontend-optimization and deployment configuration behavior.

QUESTION NO: 12

You upgraded an integration branch in your Magento Commerce Cloud project, but received an error during the deploy phase.

What are the two ways to review details about the deployment? (Choose two.)

A. View the logs in the var/report/ directory on the remote server

B. View the exception.log file in the var/log/ directory

C. View the deploy.log file in the /var/log/ directory

D. View the cloud.log file in the var/log/ directory

ANSWER: A C

Explanation:

Viewing the logs in the `var/report/` directory on the remote server and viewing the `deploy.log` file in the `/var/log/` directory provide deployment-failure details from complementary sources. Magento generates report files when an unhandled application error is rendered as an error report. These reports contain an identifier and diagnostic context that can be correlated with the failure encountered during deployment. Accessing the remote environment over SSH allows the developer to inspect the generated report directly rather than relying only on the deployment status message.

The platform deployment log records output produced by the deployment process itself, including commands, phase progress, configuration processing, and the error that terminates the deploy operation. Reviewing `/var/log/deploy.log` is therefore appropriate when an integration environment fails specifically during deployment after a branch upgrade. Together, the error report and deployment log help identify both the application-level failure details and the deploy-process context. Adobe's Cloud troubleshooting guidance describes using environment logs and SSH-based investigation to diagnose deployment failures; see Debug and troubleshoot Adobe Commerce on cloud infrastructure and Deployment process.

QUESTION NO: 13

A custom module publishes messages to an asynchronous queue, but the configured consumer is not processing messages after deployment.

How do you configure the consumer to run as part of the cloud deployment configuration?

A. Add CRON_CONSUMERS_RUNNER to the deploy stage in .magento.env.yaml

B. Add the consumer name to the relationships section of .magento.app.yaml

C. Run bin/magento queue:consumers:start only from the build hook

D. Configure the consumer in .magento/routes.yaml

ANSWER: A

Explanation:

Configure `CRON_CONSUMERS_RUNNER` in the deploy stage of `.magento.env.yaml`. This deployment variable controls the Commerce message-queue consumer runner and can enable consumer processing and define the consumers that should run.

Starting a consumer manually over SSH is not durable because manually started processes do not form part of the managed deployment configuration. The runner configuration is reapplied on each deployment. See the CRON_CONSUMERS_RUNNER deployment variable documentation.

QUESTION NO: 14

You are going to create a backup of an Integration branch before implementing some new feature.

What are two ways to create the backup of the Integration branch? (Choose two.)

A. Create a snapshot using the ece-tools snapshot:create command

B. Create a snapshot using the snapshot button in the Project Web UI

C. Create a snapshot using the magento-cloud snapshot:create command

D. Submit a support ticket requesting a backup be made

ANSWER: B C

Explanation:

Adobe Commerce on cloud infrastructure uses snapshots to preserve an environment's current state before potentially disruptive work, such as implementing a feature on an Integration branch. A snapshot captures the environment's persistent data, including the database and mounted-volume files, so it provides a restore point for that specific environment. "Create a snapshot using the snapshot button in the Project Web UI" is correct because the Cloud Console provides environment actions for manually creating a snapshot. This allows a developer to select the Integration environment and create the backup through the browser-based project interface.

"Create a snapshot using the magento-cloud snapshot:create command" is also correct because the Cloud CLI supports creating a snapshot for the active environment from the command line. This is particularly useful in scripted or terminal-based workflows, provided the developer is authenticated and targeting the intended Integration environment. Snapshots are intended to be created before major changes and can subsequently be restored when required. Adobe documents both the Cloud Console and the `magento-cloud snapshot:create` command as supported mechanisms for managing environment snapshots. See Adobe Commerce snapshots documentation and Adobe Commerce Cloud CLI overview.

QUESTION NO: 15

You are adding custom deployment commands to an Adobe Commerce on cloud infrastructure project.

Which two statements about deployment stages are correct? (Choose two.)

A. The build phase has access to the environment database

B. A command requiring a database connection should run in deploy or post-deploy rather than build

C. The post-deploy phase can be used to warm application caches after deployment

D. The deploy phase creates the reusable application artifact before any environment is selected

ANSWER: B C

Explanation:

The build phase prepares an application artifact and does not have access to environment services such as the database. Commands that require a database connection must therefore not run during build. The post-deploy phase runs after the application has been deployed and is available, making it appropriate for tasks such as cache warming and functional smoke tests.

Cloud deployment stages separate artifact creation from environment-specific deployment work. See the Adobe Commerce deployment process documentation and the hooks property documentation.