



MAGENTO-CERTIFIED-PROFESSIONAL-CLOUD-DEVELOPER^{Q&As}

Magento Certified Professional Cloud Developer

Pass Magento MAGENTO-CERTIFIED-PROFESSIONAL-CLOUD-DEVELOPER Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.passapply.com/magento-certified-professional-cloud-developer.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Magento Official Exam Center



VCE & PDF

PassApply.com

<https://www.passapply.com/magento-certified-professional-cloud-developer>
2024 Latest passapply MAGENTO-CERTIFIED-PROFESSIONAL-CLOUD-DEVELOPER PDF and VCE dumps Download

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers





QUESTION 1

After deploying to Staging for the first time with the Fastly module installed, you notice that Fastly is not caching pages. Page caching works properly in a local development environment.

What two steps are required to make Fastly cache pages? (Choose two.)

- A. Activate the module by setting Caching Application to Fastly CDN.
- B. Connect the Fastly shield using the Shield setting in the Magento admin.
- C. Enable the Fastly connection by running `magento-cloud fastly:setup`.
- D. Populate VCL in Fastly by clicking on Upload VCL to Fastly in the Magento admin.

Correct Answer: AD

Reference: <https://devdocs.magento.com/cloud/cdn/configure-fastly.html>

QUESTION 2

You fixed an issue in a third-party extension using a patch named `m2-hotfixes/extension-fix.patch`.

Before committing the patch to the repository you want to test it locally.

How do you test a patch in a clean working copy?

- A. Commit the patch because the pre-commit hook will abort the commit if any hotfix causes an issue
- B. Run the command `git apply m2-hotfixes/extension-fix.patch`
- C. Run the command `bin/magento setup:hotfix:apply --patch=m2-hotfixes/extension-fix.patch`
- D. Configure a deploy hook in `.magento.app.yaml` that executes `vendor/bin/ece-tools hotfix:apply --dry-run`

Correct Answer: B

Reference: <https://devdocs.magento.com/cloud/project/project-patch.html>

QUESTION 3

You are working with a new developer who needs to be able to deploy code and view logs on Magento Commerce Cloud.

What must you do to give them access?

- A. In the `.magento.app.yaml` file, add their email under Users. They need to enter their SSH public key.
- B. Add their SSH public key to a file named `.ssh/authorized_keys` and deploy the change.
- C. In the Cloud Project Web UI, click Add User to add their email. They need to enter their SSH public key.



D. Add a Magento admin user for them and paste the SSH public key into their user's SSH Key field, which is available only on Magento Commerce Cloud.

Correct Answer: B

Reference: <https://devdocs.magento.com/cloud/before/before-workspace-ssh.html>

QUESTION 4

In a Pro plan project, you notice there is a mail delivery problem in the Integration environment.

How do you troubleshoot this?

- A. Change the log level in the log:email section of the .magento.env.yaml file
- B. Update the SPF record in the DNS
- C. Look in the var/log directory for the mail.log file
- D. Move the test to the Staging environment

Correct Answer: D

QUESTION 5

While setting up a Magento project, you run the following command:

```
magento-cloud variable:update ADMIN_USERNAME --value=alice
```

What will this command do?

- A. Configure the Magento Commerce Cloud tools to use alice for API call authentication.
- B. Change your username to alice for logging into the magento-cloud CLI tool.
- C. Update the Magento cloud/admin/username config variable to alice.
- D. Change the username of the default administrator account to alice.

Correct Answer: C

QUESTION 6

A project has a Magento cron job implemented to export new order data to a file. This process can also be triggered by a custom bin/magento command. The functionality uses a lock file in var/locks to prevent concurrent execution.

There are instances when the process gets locked after Production deployments and order data is not exported.

How do you prevent this from happening?

- A. Use the VAR_LOCKS_CLEAR variable available for the post_deploy stage in the .magento.env.yaml file



- B. Write a custom post_deploy hook to delete orphan lock files and include it in the .magento.app.yaml file
- C. Use the VAR_LOCKS_CLEAR variable available for the build stage in the .magento.app.yaml file
- D. You must access the Production environment and delete the lock file

Correct Answer: A

QUESTION 7

You need to directly access the database that belongs to your Integration environment from your local environment.

How do you achieve this?

- A. Find the connection information from the env.php file and connect with your local MySQL client
- B. Collect the SSH connection details using magento-cloud ssh --pipe and connect with your local MySQL client
- C. Add the ece-tools db-expose command to your deploy hook
- D. Use the magento-cloud db:sql command

Correct Answer: C

QUESTION 8

You want to move static content deploy to the build phase of deployments.

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

- A. Download and commit app/etc/config.php from production
- B. Run ece-tools scd-deploy:set build on production
- C. Run ece-tools config:dump on production
- D. Use scp to copy app/etc/config.php from local to production

Correct Answer: BD

QUESTION 9

Your team is working on two major changes affecting the product page and you want the ability to implement and test these changes in parallel. Your company is using the Pro plan

How do you achieve this using the Project Web UI?

- A. Create branches from Production to ensure the catalog is up to date
- B. Create branches from Integration to facilitate merging



- C. Create branches from the Magento Cloud Docker image to ensure the changes are made on the most recent codebase
- D. Create branches from Staging to ensure the changes are tested on an environment similar to Production

Correct Answer: B

QUESTION 10

You are updating a Magento Commerce Cloud project from version 2.2.5 to 2.3.0 with the command

```
composer require magento/product-enterprise-edition:2.3.0
```

However, composer is unable to complete the task because the installed version of magento/magentocloud-metapackage is not compatible.

How do you upgrade Magento to the desired version?

- A. Create a new project based on magento/project-enterprise-edition:2.3.0, then add ece-tools, custom modules and themes again with composer and migrate the data
- B. Run `composer require magento/magento-cloud-metapackage:">=2.3.0"`