



**Free Questions for Sitecore-XM-Cloud-Developer by
certsinside**

Shared by Warren on 15-04-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which of the following statements accurately describes the purpose of Headless variants in Sitecore XM Cloud?

Options:

- A-** Headless variants define how a rendering appears and what content it displays.
- B-** Headless variants determine the order in which renderings are displayed on a webpage.
- C-** Headless variants enable the creation of custom rendering items for components from scratch.
- D-** Headless variants allow renderings to be edited and customized in the Content Editor.

Answer:

A

Explanation:

According to the [Sitecore XM Cloud Documentation for Developers¹](#), headless variants are a way to create different versions of a rendering that can be used in different contexts or scenarios. A headless variant defines the layout, style, and content of a rendering

using HTML, CSS, and JavaScript. You can create headless variants for your components in the Components builder and use them in your XM Cloud Pages. Headless variants allow you to reuse the same component with different appearances and contents without creating multiple renderings2.

2: [Create a headless variant for a component | Sitecore Documentation](#) 1: [XM Cloud Documentation for Developers - Sitecore](#)

Question 2

Question Type: MultipleChoice

A developer is creating a component that has the following requirements:

- * Image Left, Title, Call to action
- * Image Right, Title, Call to action
- * Image Centered, Call to action

The component should be able to display the same content in all three options. What is the best way to build this?

Options:

- A- Three separate components with the compatible renderings set
- B- One component with logic in the component to change the view
- C- One component with three Headless variants
- D- One component with rendering parameters to control the layout

Answer:

D

Explanation:

According to the [Sitecore XM Cloud Documentation for Developers¹](#), rendering parameters are a way to define custom properties for a component that can be changed by the content author in the XM Cloud Pages editor. Rendering parameters allow you to create flexible and reusable components that can adapt to different scenarios and layouts. To create a component with rendering parameters, you need to follow these steps²:

Create a data template that defines the rendering parameters for your component. For example, you can create a data template with a field called Layout that has three options: Left, Right, and Center.

Create a component that uses the data template as its rendering parameter template. For example, you can create a component called Image and Text that has an image, a title, and a call to action button.

In the component's code, use the rendering parameter value to control the layout of the component. For example, you can use conditional logic or CSS classes to change the position of the image based on the Layout value.

Register the component in the Components builder and add it to the Components library.

In the XM Cloud Pages editor, drag and drop the component to the page and use the Properties panel to change the rendering parameter value. For example, you can select Left, Right, or Center from the Layout drop-down menu.

[2: Create a component with rendering parameters | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore](#)

Question 3

Question Type: MultipleChoice

A developer can reference an external code base that includes components so they can be used in the Components builder interface. Which of the following are possible results of adding React components to the Components builder? Select all that apply.

Options:

- A-** The registered component is scoped to only one environment.
- B-** A content author can enter values for the defined input parameters.
- C-** The registered component is part of the Components library.

D- The registered component can be previewed in the Component builder.

Answer:

B, C, D

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, the Components builder is a tool that allows you to create and manage components for your XM Cloud Pages. You can use the Components builder to register React components from an external code base, such as a GitHub repository or a local folder. When you add a React component to the Components builder, the following results are possible:

B) A content author can enter values for the defined input parameters. When you register a React component, you can define input parameters that allow the content author to customize the component's appearance and behavior. For example, you can define a text parameter that lets the content author enter the title of the component. The input parameters are displayed in the Properties panel of the XM Cloud Pages editor².

C) The registered component is part of the Components library. When you register a React component, it becomes available in the Components library, which is a collection of components that you can use in your XM Cloud Pages. You can access the Components library from the Components builder or the XM Cloud Pages editor. You can also organize your components into categories and subcategories in the Components library³.

D) The registered component can be previewed in the Component builder. When you register a React component, you can preview how it looks and behaves in the Component builder. You can also edit the component's code, input parameters, and metadata in the

Component builder4.

The following result is not possible:

A) The registered component is scoped to only one environment. When you register a React component, it is available in all the environments of your XM Cloud project. You cannot limit the component's availability to a specific environment. However, you can control the component's visibility and accessibility by using the Publish and Archive actions in the Component builder4.

2: Register a React component | Sitecore Documentation 3: Manage the Components library | Sitecore Documentation 4: Manage a component in the Component builder | Sitecore Documentation 1: XM Cloud Documentation for Developers - Sitecore

Question 4

Question Type: MultipleChoice

A build to XM Cloud fails. Which options are available for a developer to diagnose the error?

Options:

A- The Deploy app system status page

- B-** The Content Management instance logs for the environment
- C-** The project details page in the Deploy app
- D-** The deployment log files for the environment

Answer:

D

Explanation:

According to the [Sitecore XM Cloud Documentation for Developers¹](#), the deployment log files contain the progress, status, warnings, and errors of an XM Cloud deployment. You can access the deployment log files from the XM Cloud Deploy app by clicking the Logs button on the environment page. The log files include information about the provisioning, build, deployment, and post action phases of the deployment. You can also download the log files for offline analysis or troubleshooting².

The other options are not as useful for diagnosing a build failure:

The Deploy app system status page shows the overall health and availability of the XM Cloud Deploy app, not the status of individual deployments³.

The Content Management instance logs for the environment show the runtime logs of the Content Management instance, not the build logs of the deployment⁴.

The project details page in the Deploy app shows the general information and settings of the project, not the details of the deployment¹.

[2: View the deployment log files | Sitecore Documentation](#) [3: Check the system status of the XM Cloud Deploy app | Sitecore Documentation](#) [4: Access the Content Management instance logs | Sitecore Documentation](#) [1: Manage a project in the XM Cloud Deploy app | Sitecore Documentation](#)

Question 5

Question Type: MultipleChoice

Which of the following are ways to create an XM Cloud environment?

Options:

- A-** Environments are automatically created during the deployment; it is not required to create environments explicitly.
- B-** Defining the environment at .env file from the source code (IaC) and re-deploying it.
- C-** The CLI, the Deploy app, or Rest API.
- D-** The only way to create a cloud environment is from the Sitecore Cloud Portal UI.

Answer:

C

Explanation:

According to the Sitecore XM Cloud Documentation for Developers¹, there are three ways to create an XM Cloud environment:

Using the Sitecore CLI, which is a command-line tool that allows you to perform operations against local or remote XM Cloud instances. You can use the Sitecore CLI to create a project and an environment using the `sitecore project create` and `sitecore environment create` commands².

Using the XM Cloud Deploy app, which is a web application that allows you to create and manage projects, environments, and deployments. You can use the XM Cloud Deploy app to create a project and an environment using the user interface³.

Using the XM Cloud Deploy API, which is a RESTful API that allows you to programmatically interact with the XM Cloud Deploy app. You can use the XM Cloud Deploy API to create a project and an environment using HTTP requests⁴.

2: Walkthrough: Creating an XM Cloud project using the Sitecore CLI 3: Create a project in the Deploy app 4: [XM Cloud Deploy API Reference] 1: XM Cloud Documentation for Developers - Sitecore

Question 6

Question Type: MultipleChoice

To connect Sitecore Pages to your local XM Cloud instance, which steps are involved?

Options:

- A-** Open the local Content Management instance and set the predefined rendering host field value in Sitecore to `https://pages.sitecorecloud.io`
- B-** Open Pages and create an entry in Local Storage for the key 'Sitecore.Pages.LocalXmlCloudUrl' with the following value: `https://xmcloudcm. localhost`
- C-** Open Pages and create an entry in Local Storage for the key 'Sitecore.Pages.LocalXmlCloudUrl' with the following value: `https://pages.sitecorecloud.io`
- D-** Open the local Content Management instance and set the predefined rendering host field value in Sitecore to `https://xmcloudcm.localhost`

Answer:

B

Explanation:

According to the [Sitecore XM Cloud Documentation for Developers¹](#), after setting up your local XM Cloud development environment, Sitecore Pages is not connected to your locally running XM Cloud instance. You can connect your local XM Cloud application to Sitecore Pages by setting a local storage key in your browser. This allows you to use Sitecore Pages to edit content and layouts on your local front-end application. To connect Sitecore Pages to your local XM Cloud instance, you need to follow these steps²:

In your browser, navigate to the Sitecore Pages application.

Open the developer tools console of your browser and find the local storage section. The location differs between browsers:

For Chrome and Edge, open the Application panel. In the Storage section, expand the Local Storage section.

For Firefox, open the Storage panel, and expand the Local Storage section.

[In the Local Storage panel, click the entry for Sitecore Pages 9 to reveal the local storage entries.](#)

Create a new entry by adding a new row with the following values:

Key: Sitecore.Pages.LocalXmCloudUrl.

Value: [the URL of your local XM Cloud instance. For example, 10.](#)

[Refresh the 9 page to see your local front-end application.](#)

[2: Connect Sitecore Pages to your local XM Cloud instance 1: XM Cloud Documentation for Developers - Sitecore](#)

Question 7

Question Type: MultipleChoice

Which of the following needs to happen for a developer to be able to log in into the XM Cloud Deploy app?

Options:

- A-** An Organization Admin or Organization Owner must update their role in the Sitecore Cloud Portal.
- B-** Only the Organization Owner can update their role in the Sitecore Cloud Portal.
- C-** Only the Organization Owner can update their role using the XM Cloud Deploy app.
- D-** An Organization Admin or Organization Owner must update their role using the XM Cloud Deploy app.

Answer:

A

Explanation:

According to the [Sitecore XM Cloud Documentation for Developers](#)¹, to use the XM Cloud Deploy app, you must be an Organization Admin or Organization Owner in your Sitecore Cloud Portal organization. An Organization Admin or Organization Owner can grant a team member access to the XM Cloud Deploy app by changing their organization role to Admin². This can be done in the Sitecore Cloud Portal, not in the XM Cloud Deploy app.

2: [Invite team members to your Sitecore Cloud Portal organization](#) 1: [XM Cloud Documentation for Developers - Sitecore](#)

Question 8

Question Type: MultipleChoice

A developer wants to deploy to XM Cloud, but the project is not using a source code provider that has an out-of-the-box connector for XM Cloud. How can they deploy the site to XM Cloud?

Options:

- A- Use the Sitecore Cloud CLI to create a project and deployment.
- B- Create a Sitecore Update Package using SCS to include items and files.
- C- Use Vercel to connect to their source code provider and deploy.
- D- Create a custom connector to be used with the XM Cloud Deploy app.

Answer:

A

Explanation:

According to the [Sitecore XM Cloud Documentation for Developers¹](#), the Sitecore Cloud CLI is a command-line tool that allows you to perform operations against local or remote XM Cloud instances. You can use the Sitecore Cloud CLI to deploy your existing XM Cloud-

compatible Sitecore XM solution in the cloud, regardless of the source code provider you are using. To deploy your solution to XM Cloud using the Sitecore Cloud CLI, you need to follow these steps2:

Install the Sitecore Cloud CLI on your workstation.

Authorize the Sitecore Cloud CLI with your Sitecore Cloud Portal organization.

Create a project in the XM Cloud Deploy app using the Sitecore Cloud CLI.

Connect your local environment to the project using the Sitecore Cloud CLI.

Push your serialized items to the project using the Sitecore Cloud CLI.

Deploy your solution to an XM Cloud environment using the Sitecore Cloud CLI.

[2: Walkthrough: Creating an XM Cloud project using the Sitecore CLI 1: XM Cloud Documentation for Developers - Sitecore](#)

To Get Premium Files for Sitecore-XM-Cloud-Developer Visit

<https://www.p2pexams.com/products/sitecore-xm-cloud-developer>

For More Free Questions Visit

<https://www.p2pexams.com/sitecore/pdf/sitecore-xm-cloud-developer>

