Copado-Extension-Builder Practice Questions

Shared by Stanton on 26-10-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

What information will you find in the Copado Job Execution detail tab?

Options:

A- The detail tab contains information about your Job Templates such as job execution type, API name, and description.
B- On the detail tab, you will find a checkbox named Restart Execution. This will rerun all the job steps.
C- The detail tab provides you with the list of job steps that was triggered as a part of the Job Template.
D- Under the detail tab, you can find Job Template execution results such as Status and error messages if the execution failed.

Answer:

D

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

The Job Execution detail tab provides information about the result of a Job Template execution. Option D is correct because this tab includes execution-related details such as status and error messages when the execution fails. This information helps administrators and release teams troubleshoot failed automations and understand what happened during a specific run. Job Templates define reusable automation, but Job Execution records represent actual executions of those templates. Option A incorrectly focuses on template-level details such as API name and description. Option B is inaccurate because the detail tab is not mainly described by a Restart Execution checkbox. Option C refers more to the steps involved in execution, but the question asks specifically about the detail tab. Therefore, execution results and failure details are the correct focus.

Question 2

Question Type: MultipleChoice

Copado Extensions Custom Metadata Types can be defined as:

Options:

A- Metadata types that determine the behavior of your metadata components and help you customize them as per requirement.
B- Metadata types that let you add metadata records instead of data records and provides you the capability to build apps or solution.
C- Metadata types that monitor unwanted metadata changes on any pipeline connected org and let you remove them from the deployment.
D- Metadata types that contain custom-based metadata components other than profile and permission sets.

Answer:

B

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

Copado Extensions Custom Metadata Types are used to package configuration-like records as metadata instead of ordinary data records. Option B is correct because Custom Metadata Types allow developers to define records that can be included in packages and deployed across orgs. This is important for Copado Extensions because extension-related configuration must be transportable and installable in another Copado environment. When building an extension, records such as extension configuration details can be represented through custom metadata so they behave like packageable setup data. Option A is too general and does not capture the key distinction between metadata records and data records. Option C describes metadata monitoring, which is unrelated. Option D is also incorrect because Custom Metadata Types are not simply a container for non-profile metadata components.

Question 3

Question Type: MultipleChoice

Copado Job Template is a customizable and reusable template.

Options:

A- True
B- False

Answer:

A

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

The statement is true. A Copado Job Template is designed as a reusable automation structure that can be configured once and executed whenever the same process is needed again. It contains Job Steps, and those steps define the actions that should be performed, such as running a Copado Function, executing a Salesforce Flow, or assigning a Manual Task. Because the template is reusable, teams can standardize repeated DevOps activities instead of manually recreating the same automation each time. This is useful for deployment support, external validations, multi-cloud tasks, or operational processes. In Extension Builder, Job Templates are also important because they can be converted into a packageable format using the Copado Extensions tool and reused across Copado orgs.

Question 4

Question Type: MultipleChoice

Copado Functions can be used in multiple scenarios. Which among the given scenarios is the best use case of Copado Functions?(Select all that apply) You have reached the max number of allowed answers

Options:

A- When you want to run scripts to deploy to other clouds or trigger tests in different systems.
B- When you want to store a snapshot of the executed apex scripts as part of job type included in a particular environment.
C- To have better control of user stories with low Apex Code Coverage.
D- To run tests in tools that are open source and not available as a cloud service.

Answer:

A, D

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

Copado Functions are best suited for script-based automation, external integrations, and multi-cloud DevOps activities. Option A is correct because Functions can run scripts that deploy to other cloud platforms or trigger tests in external systems. This makes them valuable when a release process goes beyond standard Salesforce deployment. Option D is also correct because Functions can run tests using open-source tools that are not available as hosted cloud services. This allows teams to include custom testing and validation in their Copado pipeline. Option B is incorrect because storing snapshots of executed Apex scripts is not a primary Function use case. Option C is also incorrect because low Apex code coverage should be addressed with proper test classes and quality controls, not directly through Copado Functions.

P2P exams

Question 5

Question Type: MultipleChoice

What is the use of the Job Execution record in regard to the Copado Job Template?

Options:

A- Job Execution records help you track the final status of quality gate type job step added in Job Templates.
B- Job Execution records enable you to promote Job Templates to the next environment.
C- From the Job Execution record, you can review the Job Template execution logs and results.
D- Job Execution records allow you to track only the status of the Job Templates for which the execution has not started.

P2P exams

Answer:

C

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

A Job Execution record is created when a Job Template runs, and it is used to review the execution outcome. Option C is correct because the Job Execution record provides access to execution logs, results, status, and error details when something fails. This record helps

administrators and DevOps teams troubleshoot the automation process step by step. It gives visibility into what happened during the Job Template execution and whether each Job Step completed successfully. Option A is too narrow because Job Execution is not limited to Quality Gate-related tracking. Option B is incorrect because Job Execution records do not promote Job Templates to another environment. Option D is also incorrect because Job Execution records track actual execution activity, not only templates that have not started.

Question 6

Question Type: MultipleChoice

Which option best is NOT a type of Copado DevOps Exchange extension?

Options:

A- Copado Labs Extensions
B- Copado Extensions
C- Partner & Community Extensions
D- Customer Extensions

Answer:

A

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

Copado Labs Extensions is not listed as one of the standard Copado DevOps Exchange extension types in the Extension Builder material. The recognized extension categories are Copado Extensions, Partner & Community Extensions, and Customer Extensions. Copado Extensions are generally provided by Copado. Partner & Community Extensions are contributed by partners or community members to extend Copado with additional integrations or reusable DevOps capabilities. Customer Extensions are built for customer-specific use cases and internal business requirements. Option A is correct because ''Copado Labs Extensions'' is not one of those extension categories. Understanding these categories is important because DevOps Exchange acts as the marketplace where teams can discover, install, and reuse extensions that enhance Copado's platform capabilities.

Question 7

Question Type: MultipleChoice

What information will you find in the Copado Job Execution detail tab?

Options:

A- The detail tab contains information about your Job Templates such as job execution type, API name, and description.
B- On the detail tab, you will find a checkbox named Restart Execution. This will rerun all the job steps.
C- The detail tab provides you with the list of job steps that was triggered as a part of the Job Template.
D- Under the detail tab, you can find Job Template execution results such as Status and error messages if the execution failed.

Answer:

D

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

The Job Execution detail tab provides information about the result of a Job Template execution. Option D is correct because this tab includes execution-related details such as status and error messages when the execution fails. This information helps administrators and release teams troubleshoot failed automations and understand what happened during a specific run. Job Templates define reusable automation, but Job Execution records represent actual executions of those templates. Option A incorrectly focuses on template-level details such as API name and description. Option B is inaccurate because the detail tab is not mainly described by a Restart Execution checkbox. Option C refers more to the steps involved in execution, but the question asks specifically about the detail tab. Therefore, execution results and failure details are the correct focus.

Question 8

Question Type: MultipleChoice

Which of the following best defines Job Execution?

Options:

A- With Job Execution, you can validate the job template execution results.
B- Using Job Execution, you can verify the Copado Actions to use in Job Template.
C- Job Execution is a task performed as a part of job template execution.
D- Job Execution is a set of predefined variables used in a Salesforce Flow deployment step

Answer:

A

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

Job Execution is best defined as the record used to validate and review the results of a Job Template execution. When a Job Template is run, Copado creates a Job Execution record for that specific execution. This record provides visibility into the status, logs, results, and error messages if the execution fails. Option A is correct because it focuses on validating execution results. Option C is incorrect because a task performed as part of a template is better described as a Job Step, not Job Execution. Option B is unrelated because Job Execution is not used to verify Copado Actions. Option D describes variables or dynamic values, not execution tracking. Job Execution is mainly used for runtime tracking and troubleshooting.

Question 9

Question Type: MultipleChoice

Identify the correct statement regarding Job Engine from the given options.(Select all that apply)

Options:

A- Job Engine allows you to re-start a failed job execution or begin from the last failed step.
B- Job Engine enables functions and flows to access details of different contexts using dynamic expressions.
C- Every time you execute a Job Engine, a Job Execution record is created and associated with the corresponding Job Template.

D- Job Engine is a Copado feature that recalculates the execution logs and prepares a result dashboard to keep track of Job Templates executed as a part of a project.

Answer:

A, B

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

Copado Job Engine is used to run reusable automation through Job Templates and Job Steps. A key capability is that failed executions can be restarted, either by running the whole execution again or by executing only outstanding or failed steps, depending on the execution scenario. This makes option A correct because it reflects recovery behavior during failed automation. Option B is also correct because Job Engine works with Functions and Salesforce Flows, and Dynamic Expressions allow those steps to access contextual values during execution. Option C is inaccurate because the usual execution unit is a Job Template, not ''executing a Job Engine.'' Option D is incorrect because Job Engine is not simply a dashboard or recalculation feature.

Question 10

Question Type: MultipleChoice

What is the two most important use of the Copado Extension tool?

Options:

A- Get the list of installed Copado Extensions. Create open-source extensions and make them available on DevOps Exchange.
B- Install a Copado Extension and generate extension records. Convert Functions and Job Templates into a packageable format.
C- Create your setup objects whose records are metadata rather than data. Create closed-source extensions and make them available on DevOps Exchange.
D- Upload extension files in different formats. Bundle the metadata components and create a package.

Answer:

B

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

The two most important uses of the Copado Extension tool are installing a Copado Extension and generating extension records, and converting Functions and Job Templates into a packageable format. Option B is correct because the tool supports both sides of the extension lifecycle: creating packageable extension content and helping generate records after installation. This is especially important because Functions and Job Templates are Copado data records, not normal Salesforce metadata components. The tool converts these records into a JSON-based format that can be stored in a static resource and included in a package. Option A is incomplete and focuses too much on listing or publishing. Option C describes custom metadata generally, and option D incorrectly focuses on generic metadata bundling rather than Copado extension records.

Question 11

Question Type: MultipleChoice

John has created certain job template records in Copado. He wants to include them in an existing managed package. What action should John perform to have those records added to the managed package?

Options:

A- John should open the Job template record and add a job step with the package details he wants to include the records in.
B- John should navigate to Copado Extensions tool and generate a JSON file by selecting Create Extension button.
C- John should convert the Job template records into a JSON format and export it to include in his managed package.
D- John should navigate to the Extension Configuration tool and add the Job template records under the Copado phase field.

Answer:

B

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder

topics:

John should navigate to the Copado Extensions tool and generate a JSON file by selecting the Create Extension button. Job Template records are Copado data records, not standard Salesforce metadata components, so they cannot be directly included in a managed package in their original record form. The Copado Extensions tool solves this by converting selected supported records, such as Job Templates and Functions, into a packageable JSON format. That JSON file can then be stored as part of the package structure, usually through a static resource and related custom metadata configuration. Option C sounds similar, but it misses the required Copado Extensions tool process. Options A and D do not correctly describe how Job Template records are prepared for packaging.

Question 12

Question Type: MultipleChoice

To run the function, Freida needs a library that is not pre-installed in the function image? What should she do?

Options:

A- Freida can install a library as part of the function execution, for example: npm install.
B- Freida can choose any existing library as an alternative.

Answer:

A

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

Freida should install the required library as part of the function execution. Copado Functions run inside Function Images, which are containerized environments containing pre-installed tools, applications, and libraries. However, a selected image may not include every dependency required by a particular function. When a needed library is missing, the function script can install it during execution, such as using npm install for a Node-based function. This allows developers to keep functions flexible while still using Copado's managed execution environment. Option B is not the best answer because choosing another library may not meet the same functional requirement. The recommended approach is to include the dependency installation step in the function logic so the execution is repeatable and consistent.

Question 13

Question Type: MultipleChoice

Which option best Function Image enables you to deploy to Heroku?

Options:

A- DPL
B- Core
C- Git Utils
D- Metadata Image

Answer:

A

Explanation:

Comprehensive and Detailed 100to 150 words of Explanation From Copado Extension builder topics:

The DPL Function Image enables deployment to Heroku. In Copado Functions, a Function Image is the containerized runtime environment that contains the tools, libraries, and applications needed to execute a function. Different Function Images support different automation scenarios. DPL is used for deployment tasks, including Heroku deployment, so option A is correct. Core is a general-purpose image, Git Utils is used for Git-related operations, and Metadata Image is used for metadata-related tasks. Since the question specifically asks about deploying to Heroku, DPL is the correct image. This is important in multi-cloud DevOps because Copado Functions allow Salesforce release processes to coordinate automation beyond Salesforce, including platforms such as Heroku and other cloud systems.

To Get Premium Files for Copado-Extension-Builder Visit

https://www.p2pexams.com/products/copado-extension-builder

For More Free Questions Visit

https://www.p2pexams.com/copado/pdf/copado-extension-builder

