

Adobe.Premium.AD0-E134.50q - DEMO

Number: AD0-E134  
Passing Score: 800  
Time Limit: 120 min  
File Version: 1.7

**Exam Code: AD0-E134**

**Website: [www.VCEplus.io](http://www.VCEplus.io)**

**Twitter: [www.twitter.com/VCE\\_Plus](https://www.twitter.com/VCE_Plus)**

## Exam A

### QUESTION 1

Which attribute must be present in all filter rules in AEM dispatcher configuration?

- A. /type
- B. /selectors
- C. /url
- D. /glob

**Correct Answer: D**

**Section:**

**Explanation:**

The /glob attribute is mandatory for all filter rules in the dispatcher configuration. It specifies a pattern that is matched against the incoming request URL. If the pattern matches, the rule is applied. The /glob attribute can use wildcards (\*) and placeholders (\$1, \$2, etc.) to match parts of the URL.

Reference: <https://experienceleague.adobe.com/docs/experience-managerdispatcher/using/configuring/dispatcher-configuration.html?lang=en#filter-rules>

### QUESTION 2

Which property under /cache on dispatcher.any file identifies the directory where cached files are stored?

- A. /invalidate
- B. /statfile
- C. /docroot
- D. /cacheroot

**Correct Answer: D**

**Section:**

**Explanation:**

The /cacheroot property under /cache in the dispatcher.any file identifies the directory where cached files are stored. It is a relative or absolute path to the cache root directory. The dispatcher creates a subdirectory for each virtual host under this directory and stores the cached files there. Reference:

<https://experienceleague.adobe.com/docs/experience-managerdispatcher/using/configuring/dispatcher-configuration.html?lang=en#cache>

### QUESTION 3

Which environment-specific configuration is used in AEM as a Cloud Service to store private API keys?

- A. \${env:ENV\_VAR\_NAME}
- B. \${env:SECRET\_VAR\_NAME}
- C. \${secret:SECRET\_VAR\_NAME}
- D. \${secret:ENV\_VAR\_NAME}

**Correct Answer: C**

**Section:**

**Explanation:**

This syntax allows the developer to reference a secret variable that is stored in the Cloud Manager UI under Environment Details > Variables. The secret variable name must start with a lowercase letter and can contain alphanumeric characters and underscores. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/configuring-osgi.html?lang=en#environment-specific-configuration>

### QUESTION 4

A developer has to create a Logger and Writer pair for the company's application logging. Which OSGi configurations should the developer use?

- A. Apache Sling Logging Logger Configuration and Apache Sling Logging Configuration
- B. Apache Sling Request Logger and Apache Sling Logging Writer Configuration
- C. Apache Sling Logging Logger Configuration and Apache Sling Logging Writer Configuration

**Correct Answer: C**

**Section:**

**Explanation:**

The Apache Sling Logging Logger Configuration and Apache Sling Logging Writer Configuration are the OSGi configurations that the developer should use to create a Logger and Writer pair for the company's application logging. The Logger Configuration defines the log level and the log file name for a given logger name or category. The Writer Configuration defines the file size, number of files, and file location for a given log file name. Reference:

<<https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/configure-logging.html?lang=en#configuring-log-files>>

#### QUESTION 5

If multiple configurations for the same PID are applicable, which configuration is applied?

- A. The last modified configuration is applied.
- B. The configuration with the highest number of matching run modes is applied.
- C. The one that occurs first in the repository is applied.
- D. A configuration factory is created and all configurations are applied.

**Correct Answer: B**

**Section:**

**Explanation:**

When multiple configurations for the same PID are applicable, the configuration with the highest number of matching runmodes is applied. This is because the runmodes act as a filter to select the most specific configuration for a given environment. If there is a tie between two or more configurations with the same number of matching runmodes, the one that occurs first in the repository is applied. Reference:

<<https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/configure-runmodes.html?lang=en#configuring-osgi-settings-perrunmode>>

#### QUESTION 6

Which configuration/section should be used to resolve the domain name by dispatcher?

- A. Configuration in vhosts file
- B. Configuration in filters.any
- C. Configuration in httpd.conf
- D. Configuration in DNS

**Correct Answer: D**

**Section:**

**Explanation:**

The configuration in DNS (Domain Name System) should be used to resolve the domain name by dispatcher. The DNS resolves the domain names to the IP address of the web server that hosts the dispatcher. The dispatcher then matches the incoming request URL with the cached files or the AEM publish instances. Reference: <https://experienceleague.adobe.com/docs/experience-manager-dispatcher/using/configuring/dispatcher-domains.html?lang=en#client-requests>

#### QUESTION 7

Which configuration must be applied to enable re-fetching of cached items based on Cache Headers sent by AEM?

- A. /autoInvalidate true
- B. /autoInvalidate "1"
- C. /enableTTLtrue
- D. /enableTTL "1"

**Correct Answer: D**

**Section:**

**Explanation:**

The /enableTTL "1" configuration must be applied to enable re-fetching of cached items based on Cache Headers sent by AEM. This configuration enables the Time To Live (TTL) feature of dispatcher, which allows dispatcher to check the Cache-Control and Expires headers of the cached files and refetch them from AEM if they are expired. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-dispatcher/using/configuring/pageinvalidation.html?lang=en#time-based-cache-invalidation>

www.VCEplus.io

**QUESTION 8**

A developer needs to create a runmode-specific OSGi configuration for an AEM as a Cloud Service implementation. In which location should the OSGi configuration be created?

- A. core project, (/core/.../config <runmode>) folder
- B. ui.config project, (/config/.../config.<runmode>) folder
- C. all project, (/all/.../config.<runmode>) folder
- D. ui.apps project (/apps/.../config.<runmode>) folder

**Correct Answer: B**

**Section:**

**Explanation:**

The ui.config project, (/config/.../config.<runmode>) folder is the location where the OSGi configuration should be created for a runmode-specific configuration for an AEM as a Cloud Service implementation. The ui.config project contains OSGi configurations that are deployed to /apps in the repository. The config.<runmode> folder specifies the runmode for which the configuration is applicable, such as author or publish.

Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/ implementing/ deploying/ configuring-osgi.html?lang=en#project-structure>

**QUESTION 9**

An AEM application wants to set up multi-tenancy using Adobe-recommended best practices and bind multiple configurations to it. Which of the following options is recommended?

- A. import org.apache.felix.scr.annotations.Component; @Component(label = "My configuration", metatype = true, factory= true)
- B. import org.osgi.service.component.annotations.Component; @Component(service = ConfigurationFactory.class)
- C. import org.osgi.service.metatype.annotations.AttributeDefinition; import org.osgi.service.metatype.annotations.ObjectClassDefinition; @ObjectClassDefinition(name = "My configuration")
- D. @Component(service = ConfigurationFactory.class) @Designate(ocd = ConfigurationFactoryImpl.Config.class, factory=true)

**Correct Answer: D**

**Section:**

**Explanation:**

The @Component(service = ConfigurationFactory.class) @Designate(ocd = ConfigurationFactoryImpl.Config.class, factory=true) option is recommended for creating a multitenancy configuration and binding multiple configurations to it. This option uses the OSGi R6 annotations to define a component that provides a service of type ConfigurationFactory and designates a class that contains the configuration properties. The factory=true attribute indicates that multiple configurations can be created for this component. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-65/ deploying/ configuring/osgiconfiguration- settings.html?lang=en#creating-factory-configurations>

**QUESTION 10**

An AEM application requires LDAP Service integration to synchronize users/groups. Which two OSGi configuration are required for LDAP integration in AEM? (Select Two.)

- A. Apache Jackrabbit Oak AuthorizableActionProvider
- B. Apache Jackrabbit Oak Solr server provider
- C. Apache Jackrabbit Oak CUG Configuration
- D. Apache Jackrabbit Oak External Login Module
- E. Apache Jackrabbit Oak Default Sync Handler

**Correct Answer: D, E**

**Section:**

**Explanation:**

The Apache Jackrabbit Oak External Login Module and Apache Jackrabbit Oak Default Sync Handler are the two OSGi configurations that are required for LDAP integration in AEM. The External Login Module defines how AEM connects to the LDAP server and authenticates users against it. The Default Sync Handler defines how AEM synchronizes users and groups from the LDAP server to the repository. Reference:

<<https://experienceleague.adobe.com/docs/experience-manager-65/administering/security/ldap-config.html?lang=en#ldap-integration>>

**QUESTION 11**

A client is having issues with some query results:

- Many of the client's industry terms have the same meaning, and users do not always search the exact wording
- Many users search by typing in short phrases instead of exact keywords, ex:// "cats and dogs"

What index analyzers should the AEM developer recommend?

- A. 1. Add a Mapping filter to the current indexes
- B. Add a Stop filter to the current indexes
- C. 1. Tokenize the current indexes with a Keyword tokenizer
- D. Add a Mapping filter to the current indexes
- E. 1. Add a Synonym filter to the current indexes
- F. Add a Stop filter to the current indexes
- G. 1. Add a Synonym filter to the current indexes
- H. Add a LowerCase filter to the current indexes

**Correct Answer: D**

**Section:**

**Explanation:**

A Synonym filter can help to map different terms that have the same meaning, such as "cat" and "feline". A LowerCase filter can help to normalize the case of the terms, so that "cats and dogs" and <https://www.validexamdumps.com> "Cats and Dogs" are treated the same.

Reference: 1 Lucene Analyzers section

#### QUESTION 12

An AEM server is overloaded with too many concurrently running workflows. The developer decides to reduce the number of concurrent workflows. What should be configured to reduce the number of concurrent workflows?

- A. The number of threads in Scheduler
- B. The number of threads in Apache Felix Jetty Http Service
- C. Launchers for each workflow
- D. Maximum Parallel Jobs in OSGI console

**Correct Answer: D**

**Section:**

**Explanation:**

Maximum Parallel Jobs is a configuration property that controls how many workflows can run concurrently on an AEM instance. Reducing this value can help to avoid overloading the server with too many workflows.

Reference: Workflow Engine Configuration section

#### QUESTION 13

A custom component has one dialog field:

-> Title

-fieldLabel = Title

-sling:resourceType = granite/ui/components/coral/foundation/form/textfield

-name = ./title

The developer needs to implement a Sling Model to perform a business logic on the authored value.

The developer writes the following HTL snippet.

www.VCEplus.io

```
<sly data-sly-use.display="com.adobe.aem.guides.certification.core.models.HelloWorldModelImpl">
<h1>${display.messageText}</h1>
</sly>
```

Which two implementations will support this HTL snippet? (Choose two.)

A) `@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)`

```
public class HelloWorldModelImpl {
    @ScriptVariable
    private String authoredVal;
    private String messageText;
```

```
    @PostConstruct
    public void init() {
        if (StringUtils.isNotBlank(authoredVal)) {
            setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, authoredVal));
        }
    }

    public void setMessageText(String messageText) {
        this.messageText = messageText;
    }

    public String getMessageText() {
        return messageText;
    }
}
```

B)

```
public void init() {  
    if (StringUtils.isNotBlank(title)) {  
        setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, title));  
    }  
}  
  
public void setMessageText(String messageText) {  
    this.messageText = messageText;  
}  
  
public String getMessageText() {  
    return messageText;  
}  
}
```

@Model(adaptables = SlingHttpServletRequest.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)

public class HelloWorldModelImpl {

@Inject

@Via("resource")

private String title;

private String messageText;

c)

```
@PostConstruct
public void init() {
    if (StringUtils.isNotBlank(title)) {
        setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, title));
    }
}

public void setMessageText(String messageText) {
    this.messageText = messageText;
}

public String getMessageText() {
}
```

```
@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class HelloWorldModelImpl {
    @ValueMapValue
    @Named("title")
    private String authoredVal;
    private String messageText;
```

D)

```
@PostConstruct
public void init() {
    if (StringUtils.isNotBlank(title)) {
        setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, title));
    }
}

public void setMessageText(String messageText) {
    this.messageText = messageText;
}

public String getMessageText() {
    return messageText;
}
```

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

Correct Answer: B, D



**Section:****Explanation:**

Option B and Option D are two implementations that will support the HTL snippet. Option B uses the @Model annotation with the adaptables parameter set to Resource.class. This allows the Sling Model to adapt from a resource object and access its properties using the ValueMap interface.

Option B also uses the @Inject annotation with the name parameter set to ".text" to inject the value of the text property into the text field. Option D uses the @Model annotation with the defaultInjectionStrategy parameter set to OPTIONAL. This allows the Sling Model to use optional injection for all fields and avoid null pointer exceptions if a property is missing. Option D also uses the @Inject annotation without any parameters to inject the value of the text property into the text field, using the field name as the default property name. Reference:

<https://sling.apache.org/documentation/bundles/models.html>

<https://experienceleague.adobe.com/docs/experience-manager-htl/using-htl/htl-blockstatements.html?lang=en#use>

**QUESTION 14**

A developer needs to create a new Title component. The requirements are:

- A. The layout must be the same as the Title core component
- B. The text property must have the page title as prefix (e.g., Page Title - <component text>)
- C. The component must be reusable

Which approach is recommended?

- D.
- E. Create a Proxy Component of Title core component
- F. Create a Custom Sling Model that overrides the default behavior
- G. Customize the component template
- B, 1. Create a custom component from scratch
- H. Create a Custom Sling Model for the component that follows the requirement
- I. Create a Model Exporter
- J.
- K. Create a Proxy Component from Title core component
- L. Create a Custom Sling Model that overrides the default behavior

**Correct Answer: A**

**Section:****Explanation:**

A proxy component is a site-specific component that inherits from a core component and allows customization of the component name, group, dialog, and behavior. A proxy component can refer to any version of the core component by changing the sling:resourceSuperType property. A custom sling model can be used to implement the logic for adding the page title as prefix to the text property. A component template can be used to define the layout of the component.

Reference: 1 Using Core Components section 2 Create Proxy Component in AEM section 3 AEMCreate

Proxy Component section 4 Proxy Components in AEM 6.4 section 5 AEM Proxy Component Pattern and Component Versioning section

**QUESTION 15**

A developer needs to create sling models for two fields name and occupations. The dialog has two fields, name - a single value field, and occupations - a multi value field.

The following code is included in sling models inherited from interface com.adobe.aem.guides.wknd.core.models.Byline

```

package com.adobe.aem.guides.wknd.core.models.impl;
.....
public class BylineImpl implements Byline {
    .....
    @Override
    public List<String> getOccupations() {
        if (occupations != null) {
            Collections.sort(occupations);
            return new ArrayList<String>(occupations);
        } else {
            return Collections.emptyList();
        }
    }
    .....
}

```

A)

```

<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name}</h2>
    <p class="cmp-byline__occupations">${byline.occupations @ join=', '}</p>
</div>

```

B)

```

<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline.impl"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name}</h2>
    <p class="cmp-byline__occupations">${byline.occupations @ join=', '}</p>
</div>

```

C)

```

<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name}</h2>
    <p class="cmp-byline__occupations">${byline.occupations }</p>
</div>

```

D)

```
<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline"
      data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
      data-sly-test.hasContent="${!byline.empty}"
      class="cmp-byline">
  <h2 class="cmp-byline__name">${byline.name @ join=', '}</h2>
  <p class="cmp-byline__occupations">${byline.occupations @ join=', '}</p>
</div>
```

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

**Correct Answer: C**

**Section:**

**Explanation:**

Option C is the correct implementation for the Sling Model. Option C uses the @Model annotation with the adaptables parameter set to Resource.class. This allows the Sling Model to adapt from a resource object and access its properties using the ValueMap interface. Option C also uses the @Inject annotation with the name parameter set to ".name" and ".occupations" to inject the values of the name and occupations properties into the name and occupations fields. Option C also uses the @Named annotation with the value parameter set to "byline" to specify the name of the Sling Model that can be used in HTL scripts. Reference:

<https://sling.apache.org/documentation/bundles/models.html>

<https://experienceleague.adobe.com/docs/experience-manager-htl/using-htl/htl-blockstatements.html?lang=en#use>

#### QUESTION 16

SPA components are connected to AEM components via the MapTo() method.

Which code should be used to correctly connect an SPA component called ItemList to its AEM equivalent?

- A. ('project/components/content/itemList,).MapTo(ItemList,ItemListEditConfig);
- B. MapToCproject/components/content/itemList^ItemListItemListEditConfig);
- C. ItemList.MapToCproject/components/content/itemList1);
- D. MapTo(ItemList)('project/components/content/itemList,,ItemListEditConfig);

**Correct Answer: C**

**Section:**

**Explanation:**

The MapTo() method is used to map a SPA component to an AEM component by specifying the sling:resourceType of the AEM component as an argument. The MapTo() method should be called on the SPA component and not on a string. The second argument of the MapTo() method is optional and can be used to provide an edit configuration for the SPA component.

Reference: 1 Map SPA components to AEM components section 2 Developing SPAs for AEM section

#### QUESTION 17

Refer to the exhibit.

```
<div class="list">  
<ul data-sly-  
list="currentPage.listChildren" >< li > This is a sample HTL snippet < /li >< /ul >< /div >< div class  
{currentPage.listChildren}>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>
```

The current page has three children.  
What is the final rendered html output for the code snippet?  
A)

[www.VCEplus.io](http://www.VCEplus.io)

```
<div class='list'>
<ul>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
</ul>
</div>
<div class='repeat'>
<ul>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
</ul>
</div>
<div class='list'>
<ul>
<li>This is a sample HTL snippet</li>
</ul>
<ul>
<li>This is a sample HTL snippet</li>
</ul>
<ul>
<li>This is a sample HTL snippet</li>
</ul>
</div>
```

www.VCEplus.io

B)

```
</ul>
</div>
<div class='repeat'>
<ul>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
</ul>
</div>

<div class="list">
<ul>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
<li>This is a sample HTL snippet</li>
</ul>
</div>
<div class="repeat">
<ul>
```

C)

www.VCEplus.io

```
<li>This is a sample HTL snippet</li>
</ul>
<ul>
<li>This is a sample HTL snippet</li>
</ul>
<ul>
<li>This is a sample HTL snippet</li>
</ul>
</div>
```

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

**Correct Answer: B**

**Section:**

**Explanation:**

Option B is the final rendered html output for the code snippet. The code snippet uses the data-slylist block statement to iterate over the child pages of the current page. The data-sly-list statement assigns each child page to a variable named page and provides an index variable named i. The code snippet then uses the data-sly-test block statement to check if the index is odd or even and applies a different CSS class accordingly. The code snippet also uses the data-sly-element block statement to create an HTML element with the name specified by the elementName variable. The code snippet also uses the data-sly-attribute block statement to add an attribute with the name specified by the attrName variable and the value specified by the attrValue variable. The code snippet also uses the data-sly-resource block statement to include a resource with the path specified by the page.path variable and the resourceType specified by the resourceType variable. Reference: <https://experienceleague.adobe.com/docs/experience-manager-htl/using-htl/htl-blockstatements.html?lang=en>

#### QUESTION 18

A developer needs to create a dynamic participant step where the participant is selected automatically at run time. The developer decides to develop an OSGi service, which needs to implement the com.day.cq.workflow.exec.ParticipantStepChooser interface. Which method should the developer implement from the com.day.cq.workflow.exec.ParticipantStepChooser interface?

- A. String getParticipant(WorkItem workItem, WorkflowSession workflowSession, MetaDataMap metaDataMap)
- B. void getParticipant(WorkItem workItem, WorkflowSession workflowSession, MetaDataMap metaDataMap)
- C. String getDynamicParticipant(WorkItem workItem, WorkflowSession workflowSession, MetaDataMap metaDataMap)
- D. void getDynamicParticipant(WorkItem workItem, WorkflowSession workflowSession, MetaDataMap metaDataMap)

**Correct Answer: C**

**Section:**

**Explanation:**

The com.day.cq.workflow.exec.ParticipantStepChooser interface is intended for implementations that will define the participant dynamically. This interface replaces the deprecated com.day.cq.workflow.exec.ParticipantChooser interface. The method getDynamicParticipant returns the dynamically resolved Principal id based on the work item, workflow session and metadata map parameters. Reference: <https://developer.adobe.com/experience-manager/referencematerials/cloud-service/javadoc/com/day/cq/workflow/exec/ParticipantStepChooser.html>

#### QUESTION 19

An AEM application has a Header and Footer authored on every page.

The customer asks for the following:

- A. A centralized Header and Footer
- B. The ability to create a variation for both the Header and Footer
- C. Change the Header and Footer for specific time periods
- D. The ability to restore a previous version for both the Header and Footer

What should the developer use to meet the requirements?

- E. Custom component
- F. Content fragment
- G. Static template
- H. Experience fragment

**Correct Answer: D**

**Section:**

**Explanation:**

An experience fragment is a group of one or more components including content and layout that can be referenced within pages. Experience fragments allow authors to create variations for different channels and modify them for specific time periods. Experience fragments also support versioning and restoring previous versions. Reference: <<https://experienceleague.adobe.com/docs/experiencemanager-65/authoring/authoring/experience-fragments.html?lang=en>>

#### QUESTION 20

AEM SPA integration provides various design models. In an application the developer chooses to use AEM as a headless CMS without using the SPA Editor SDK framework.

What would be an advantage for this design model?

- A. The content author can edit the app using AEM's content authoring experience.
- B. The developer has full control over the app.
- C. The SPA is compatible with the template editor
- D. The developer keeps control over the app by only enabling authoring in restricted areas of the app

**Correct Answer: B**

**Section:**

**Explanation:**

AEM SPA integration provides various design models for different levels of authoring capabilities and developer control. In the design model where AEM is used as a headless CMS without using the SPA Editor SDK framework, the developer has full control over the app and can use any SPA framework or library. However, this also means that the content author cannot edit the app using AEM's content authoring experience, the SPA is not compatible with the template editor, and the developer cannot enable authoring in restricted areas of the app. Reference: <https://experienceleague.adobe.com/docs/experience-manager-learn/spa-editor-overview/spaoverview.html?lang=en#design-models>

#### QUESTION 21

An AEM Developer needs to create a new component to help support a new product launch.

- The client is on AEM 6.5 on-premise with the latest version of WCM Core Components
- The component must include text, image, and a link
- The component must support multiple designs

Which process should the AEM Developer use to support the launch?

- A.
- B. Extend the Teaser Component from Core Components
- C. Create style variations to be used in the Style System
- D.
- E. Create a new component by extending the Text Component from Core Components
- F. Add dialog properties and modify HTL to support images
- G.
- H. Extend the Text Component from Core Components
- I. Enable image manipulations for the Text Component via policy



- J.
- K. Create a new Image with Text component that exposes the Core Components authoring dialogs for those components
- L. Add a policy to define which designs are used

**Correct Answer: A**

**Section:**

**Explanation:**

Extend the Teaser Component from Core Components

Create style variations to be used in the Style System\*\* Comprehensive Explanation of Correct Answer Only: The Teaser Component from Core Components is a component that allows authors to display a title, description, image and link for a teaser item. The component supports multiple designs and can be extended to add custom features or logic. The Style System allows authors to define style variations for components without requiring code changes or new templates. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/components/teaser.html?lang=en>

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/components/style-system.html?lang=en>

#### QUESTION 22

An AEM application is expected to export a content fragment in JSON format without any customization for a headless implementation.

What is the recommended approach?

- A. Use AEM Assets HTTP API
- B. Use Core components to export JSON
- C. Use Sling Exporter framework

**Correct Answer: A**

**Section:**

**Explanation:**

AEM Assets HTTP API is a RESTful API that allows access to content fragments in JSON format without any customization. The API supports CRUD operations on content fragments and their variations, as well as querying and searching for content fragments based on metadata or full-text search. Reference: <<https://experienceleague.adobe.com/docs/experience-manager-65/assets/extending/assets-api-content-fragments.html?lang=en>>

#### QUESTION 23

In a non-optimized website, the final HTML generated for a typical page by publish instance includes a relatively large number of <script> elements that refer to other script files loaded from AEM environment. The developer wants to minimize these network calls by combining all required client library code into a single file to reduce the number of back-and-forth requests on page load.

Which step should a developer take to solve this issue?

- A. Embed the required libraries into an app-specific client library using the allowProxy property of the cq:ClientLibraryFolder node
- B. Add the categories property of the cq:ClientLibraryFolder node into an app-specific client library folder
- C. Embed the required libraries into an app-specific client library using the dependencies property of the cq:ClientLibraryFolder node
- D. Embed the required libraries into an app-specific client library using the embed property of the cq:ClientLibraryFolder node

**Correct Answer: C**

**Section:**

**Explanation:**

The embed property of the cq:ClientLibraryFolder node allows embedding code from a client library into another client library. At runtime, the generated JS and CSS files of the embedding library include the code of the embedded library. This reduces the number of network calls and improves performance. Embedding code is useful for providing access to libraries that are stored in secured areas of the repository. Reference:

<<https://experienceleague.adobe.com/docs/experience-managercloud-service/content/implementing/developing/full-stack/clientlibs.html?lang=en#embed>>

<<https://experienceleaguecommunities.adobe.com/t5/adobe-experience-manager/embed-propertyin-clientlibs/m-p/426858>>

#### QUESTION 24

An AEM application development team is assigned a task to create an Event-Driven Data Layer implementation for an Analytics solution.

Which Adobe recommended best practice should the developer choose?

- A. Use Adobe Experience Platform's data layer to integrate with AEM.
- B. Create a custom data layer and add each component template, and its properties to the data layer
- C. Use Adobe Client Data Layer and integrate with Core components.

D. Create an Adobe Cloud Service configuration to use third-party tool's data layer.

**Correct Answer: C**

**Section:**

**Explanation:**

Adobe Client Data Layer is a JavaScript library that provides a standardized way to collect, structure, and manage data on a web page. It can be used to implement an event-driven data layer for analytics solutions. It integrates with Core components and allows authors to configure data layer properties for each component. It also supports custom events and data sources. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/developing/data-layer.html?lang=en> <https://github.com/adobe/adobe-clientdata-layer>

#### QUESTION 25

A developer is on an AEM application that is being used to calculate an employee's salary. The calculation is done in an OSGi service called CalculationService. This service class has a dependency on one other service, called the EmployeeService.

How should the developer make sure that the critical code in the CalculationService has a high unit test coverage?

- A. Use a mock framework in the unit test to inject the CalculationService
- B. Instantiate the EmployeeService in the unit test and pass it to the constructor of the CalculationService
- C. Use a mock framework in the unit test to inject the EmployeeService
- D. Use the feature flag in the unit test to disable the calls to the EmployeeService

**Correct Answer: C**

**Section:**

**Explanation:**

A mock framework is a tool that allows creating mock objects that simulate the behavior of real objects in a controlled way. A mock framework can be used in a unit test to inject the EmployeeService dependency into the CalculationService and verify its interactions. This way, the unit test can focus on testing the logic of the CalculationService without relying on the actual implementation of the EmployeeService. Reference:

<https://sling.apache.org/documentation/development/sling-testing-tools.html>

<https://wcm.io/testing/aem-mock/usage.html>

#### QUESTION 26

An AEM Developer receives requirements for Sling Models in a human-readable yaml format. A custom application needs to be built. The dependency is as shown:

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.8.4</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.dataformat</groupId>
  <artifactId>jackson-dataformat-yaml</artifactId>
  <version>2.8.4</version>
</dependency>
```

- A.
- B. Create OSGI models to export as yaml
- C. Configure mime type in Apache Sling MIME Type Service
- D.
- E. Create OSGI models to export as yaml
- F. Configure mime type in Apache Sling Servlet/Script Resolver and Error Handler
- G.
- H. Create Sling models to export as yaml

- I. Configure mime type in Apache Sling MIME Type Service
- J.
- K. Create Sling models to export as yaml
- L. Configure mime type in Apache Sling Referrer Filter

**Correct Answer: C**

**Section:**

**Explanation:**

To create Sling Models that can export data in a human-readable yaml format, the following steps are required:

Create Sling models to export as yaml. Sling models are Java classes that can be used to represent resources in AEM. Sling models can use annotations to define how they can be adapted from a resource and how they can export data in different formats. To export data in yaml format, the Sling model class needs to use the @Model annotation with the resourceType parameter set to the resource type of the resource that the model represents. The Sling model class also needs to implement the org.apache.sling.models.annotations.Exporter annotation with the name parameter set to "jackson" and the extensions parameter set to "yaml". The Sling model class also needs to use the @JsonProperty annotation on the fields or methods that need to be exported in yaml format. Configure mime type in Apache Sling MIME Type Service. The Apache Sling MIME Type Service is an OSGi service that maps file extensions to MIME types and vice versa. To enable the yaml format for Sling models, the MIME Type Service needs to be configured with a new entry for the yaml extension and its corresponding MIME type, which is "application/x-yaml". This can be done by creating an OSGi configuration for the org.apache.sling.commons.mime.internal.MimeTypeServiceImpl service and adding the entry "yaml=application/x-yaml" to the mime.types property. Reference:

<https://sling.apache.org/documentation/bundles/models.html>

<https://sling.apache.org/documentation/bundles/mime-type-support-commons-mime.html>

#### QUESTION 27

A developer needs to create a workflow custom process step in AEM. In a custom process step, an OSGi component needs to implement the WorkflowProcess interface. Which method should the developer implement?

- A. call
- B. apply
- C. execute
- D. submit

www.VCEplus.io

**Correct Answer: C**

**Section:**

**Explanation:**

The WorkflowProcess interface is the interface to be used for automatic workflow steps implemented in Java. Classes implementing this interface define Java based processes that can be attached to a WorkflowNode and executed by the workflow engine. The method execute takes a WorkItem, a WorkflowSession and a MetadataMap as parameters and performs the logic of the custom process step. Reference:

<https://developer.adobe.com/experience-manager/referencematerials/6-5/javadoc/com/adobe/granite/workflow/exec/WorkflowProcess.html>

<https://experienceleague.adobe.com/docs/experience-manager-learn/forms/adaptiveforms/custom-process-step-aem-workflow.html?lang=en>

#### QUESTION 28

A customer adds third-party client libraries to add some features in an existing AEM application, which will significantly reduce performance.

How should the developer optimize the site?

- A. Embed client libraries to consolidate them into fewer files.
- B. Debug third-party client lib and fix the code.
- C. Rebuild Client libraries.

**Correct Answer: A**

**Section:**

**Explanation:**

Embedding client libraries is a technique that allows combining code from multiple client libraries into a single file. This reduces the number of network requests and improves performance.

Embedding client libraries can be done by using the embed property of the cq:ClientLibraryFolder node and specifying the categories of the client libraries to be embedded. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/content/implementing/developing/full-stack/clientlibs.html?lang=en#embed>

<<https://experienceleague.adobe.com/docs/experience-manager-65/developing/introduction/clientlibs.html?lang=en#embedded-files>>

#### QUESTION 29

A snippet throws an exception at runtime:

```
@Model(adaptables = {Resource.class}) public class MyCustomModel {  
(SlingObject  
private Resource resource;
```

```
@Inject  
private Page currentPage;  
  
private String currentPagePath;  
  
@PostConstruct  
protected void init() {  
    this.currentPagePath = currentPage.getPath();  
}
```

What should the developer add to fix it?

- A. defaultInjectionStrategy = DefaultInjectionStrategy property to @Model Class annotation
- B. (@Optional annotation to page field
- C. throws Exception at the end of the init method declaration
- D. SlingHttpServletRequest.class to adaptables property of @Model Class annotation

**Correct Answer: A**

**Section:**

**Explanation:**

The developer should add the defaultInjectionStrategy = DefaultInjectionStrategy property to the @Model Class annotation to fix the snippet. The defaultInjectionStrategy property defines how the Sling Model handles missing or null values for the injected fields. By default, the Sling Model uses the REQUIRED injection strategy, which means that all fields must have a non-null value or else an exception is thrown. By setting the defaultInjectionStrategy property to OPTIONAL, the Sling Model allows null values for the injected fields and does not throw an exception. This way, if the page field is null because the resource is not a page, the Sling Model can still work without errors. Reference:

<https://sling.apache.org/documentation/bundles/models.html>

<https://sling.apache.org/documentation/bundles/models.html#optional-injection>

### QUESTION 30

Which two unit testing dependencies are generated by AEM archetype? (Select two.)

- A. JUnit
- B. Selenium
- C. PowerMock
- D. Mockito
- E. Hobbes

**Correct Answer: A, D**

**Section:**

**Explanation:**

JUnit and Mockito are two unit testing dependencies that are generated by AEM archetype. JUnit is a framework for writing and running unit tests in Java. Mockito is a framework for creating and using mock objects in unit tests. AEM archetype also adds Apache Sling Mocks and AEM Mocks Test Framework by io.wcm as dependencies for unit testing. Reference:

<<https://experienceleague.adobe.com/docs/experience-manager-learn/getting-started-wkndtutorial-develop/project-archetype/unit-testing.html?lang=en>>

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/developing/archetype/using.html?lang=en>

### QUESTION 31

An application development team needs to create a multi-channel application that uses AEM as a headless CMS Which feature should be used to maintain structured data for this application?

- A. Static template
- B. Content fragment

- C. Experience fragment
- D. Custom component

**Correct Answer: B**

**Section:**

**Explanation:**

A content fragment is a reusable piece of content that can be authored and managed independently from a page. Content fragments are based on structured data models that define the elements and data types of the content. Content fragments can be delivered as JSON using various APIs, such as Assets HTTP API, Content Fragments REST API, or Content Fragments GraphQL API. Content fragments are suitable for creating multi-channel applications that use AEM as a headless CMS.

Reference: <https://experienceleague.adobe.com/docs/experience-manager-learn/foundation/headless/introduction-to-headless.html?lang=en>  
<https://experienceleague.adobe.com/docs/experience-manager-65/assets/fragments/contentfragments.html?lang=en>

### QUESTION 32

The following anchor tag is not resolving:

```
<a href="item.path" >{item.name}</a>
```

Upon further inspection the developer notices that the link has no .html appended to the end of the URL

What could be a potential fix for the issue?

A)

```
<a href="item.path@extension = 'html'" >{item.name}</a>
```

B)

```
<a href="item.path@context = 'unsafe, fragment = item.name'" >{item.name}</a>
```

C)

```
<a href="item.path@append = 'html'" >{item.name}</a>
```

D)

```
<a href="item.path@context = 'html'" >{item.name}</a>
```

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

**Correct Answer: B**

**Section:**

**Explanation:**

Option B is a potential fix for the issue. Option B uses the data-sly-attribute block statement to add the href attribute to the anchor tag. The data-sly-attribute statement uses an expression to append ".html" to the item.path value. This way, the link will have the correct extension and will resolve to the corresponding page. Reference: <https://experienceleague.adobe.com/docs/experience-manager-htl/using-htl/htl-block-statements.html?lang=en#data-sly-attribute>

### QUESTION 33

A developer needs to create a project based on AEM Project Archetype with a specific AEM as a Cloud Service SDK version on the local environment.

Which two properties must be defined when creating this project? (Choose two.)

- A. aemVersion=cloud
- B. sdkVersion=2022.5.7575.20220530T152407Z-220401
- C. sdkVersion=latest
- D. aemVersion=latest

E. aemVersion=2022.5.7575.20220530T152407Z-220401

**Correct Answer: A, B**

**Section:**

**Explanation:**

The aemVersion property specifies the target AEM version for the project. To create a project based on AEM as a Cloud Service SDK, the value should be set to cloud. The sdkVersion property specifies the AEM as a Cloud Service SDK version to use for local development. To use a specific SDK version, the value should be set to the exact version number, such as 2022.5.7575.20220530T152407Z- 220401. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/developing/archetype/using.html?lang=en#properties>

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/developing/aem-as-a-cloud-service-sdk.html?lang=en#sdk-versions>

#### QUESTION 34

In an AEM as a Cloud Service environment, which repository areas are considered as immutable?

- A. /content and /libs
- B. /content and /etc
- C. /apps and /libs
- D. /apps and /content

**Correct Answer: C**

**Section:**

**Explanation:**

In AEM as a Cloud Service environment, the repository areas /apps and /libs are considered as immutable, meaning that they cannot be modified at runtime by any means. These areas contain application code and configuration that are deployed via Cloud Manager and are read-only for security and performance reasons. Any changes to these areas require a new deployment via Cloud Manager. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/overview.html?lang=en#repository-restrictions>

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/developing/aem-project-content-packagestructure.html?lang=en#repository-restrictions>

#### QUESTION 35

On package install content that is already present in the repos must not be overwritten and if not present in the repos it must not be removed.

Which import mode should the developer use?

- A. update
- B. replace
- C. merge

**Correct Answer: A**

**Section:**

**Explanation:**

The merge import mode is used to import content that is already present in the repository without overwriting it and without removing content that is not present in the package. This mode preserves existing content properties and nodes while adding new ones from the package. This mode is useful for incremental updates or initial content seeding. Reference:

[https://jackrabbit.apache.org/filevault/vaultfs.html#Import\\_Modes](https://jackrabbit.apache.org/filevault/vaultfs.html#Import_Modes)

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/content-package-import-modes.html?lang=en>

#### QUESTION 36

Which AEM dependency provides APIs for all AEM environments?

- A)

```
<dependency>  
<groupId>com.adobe.aem</groupId>  
<artifactId>aem-api</artifactId>  
<version>6.5.0</version>  
<classifier>apis</classifier>  
<scope>provided</scope>  
</dependency>
```

B)

```
<dependency>  
<groupId>com.adobe.cq</groupId>  
<artifactId>core.wcm.components.content</artifactId>  
<type>zip</type>  
</dependency>
```

C)

```
<repository>  
<id>adobe-public-releases</id>  
<name>Adobe Public Repository</name>  
<url>https://repo.adobe.com/nexus/content/groups/public/</url>  
<layout>default</layout>  
</repository>
```

D)

www.VCEplus.io

```
<dependency>
<groupId>com.adobe.aem</groupId>
<artifactId>uber-jar</artifactId>
<version>6.5.0</version>
<classifier>apis</classifier>
<scope>provided</scope>
</dependency>
```

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

**Correct Answer: C**

**Section:**

**Explanation:**

Option C is the AEM dependency that provides APIs for all AEM environments. Option C uses the uber-jar dependency with the provided scope and the classifier set to apis. The uber-jar dependency contains all the public Java APIs that are available in AEM. The provided scope indicates that the dependency is not required for runtime, as it is already provided by the AEM platform. The apis classifier indicates that only the API classes are included in the dependency, not the implementation classes. Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/developing/aem-project-content-package-structure.html?lang=en#projectdependencies>  
<https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/html/eclipse-setup.html>

#### QUESTION 37

A development team is starting a new AEM project that is going to integrate with the Adobe Commerce platform. The developer needs to create a new AEM project using the Maven command line interface.

How can the 'mvn -B archetype:generate' command help the developer with the integration between AEM and Adobe Commerce?

- A. Using the property 'commerceModule=AdobeCommerce' can provide a path to an external jar that integrates between the platforms.
- B. Using the property ,aemVersion=cloudl automatically provides a report with integration guidelines.
- C. Using the property 'includeCommerce=y'1 the command will generate specific Commerce Core Components.

**Correct Answer: C**

**Section:**

**Explanation:**

The includeCommerce property is a boolean flag that indicates whether to include Commerce Core Components in the project or not. If set to y, the command will generate a commerce module that contains specific Commerce Core Components and their dependencies. These components can be used to integrate AEM with Adobe Commerce platform or other commerce solutions. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/developing/archetype/using.html?lang=en#properties>

<https://experienceleague.adobe.com/docs/experience-manager-corecomponents/using/components/commerce.html?lang=en>

#### QUESTION 38

What is the recommended path to override /libs standard functionality?

- A. /conf
- B. /apps
- C. /content



D. /libs

**Correct Answer: B**

**Section:**

**Explanation:**

The recommended path to override /libs standard functionality is /apps. The /apps folder contains application-specific code and configuration that can extend or override the default functionality provided by /libs.

The /apps folder has precedence over the /libs folder, meaning that if there are two resources with the same path under both folders, the one under /apps will be used. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/developing/aem-project-content-packagestructure.html?lang=en#repository-restrictions>

<https://experienceleague.adobe.com/docs/experience-managerlearn/foundation/development/set-up-a-local-aem-development-environment.html?lang=en#theaem-repository>

#### QUESTION 39

Which Maven plugin checks if all the requirements declarations made in OSGi bundles are satisfied by the capabilities declarations of other bundles included in the Maven project?

- A. maven-enforcer-plugin
- B. femaven-assembly-plugin
- C. content-package-maven-plugin
- D. aemanalyser-maven-plugin

**Correct Answer: D**

**Section:**

**Explanation:**

The aemanalyser-maven-plugin is a Maven plugin that checks if all the requirements declarations made in OSGi bundles are satisfied by the capabilities declarations of other bundles included in the Maven project.

This plugin ensures that the OSGi bundles are consistent and can be resolved at runtime. The plugin also checks for other issues such as API compatibility, package versioning, and bundle start order. Reference:

<https://experienceleague.adobe.com/docs/experience-managercloud-service/implementing/developing/aem-project-content-packagestructure.html?lang=en#build-analyzer-maven-plugin>

<https://github.com/adobe/aemanalysermaven-plugin>

#### QUESTION 40

An AEM development team is working on a new multi-country application using AEM as a Cloud

Service. A developer has been assigned the task for building the integration with a third-party web service. A secret key is needed to connect with this web service. The website creators will provide this key. The key is different for each type of environment (dev, stage and production)

What is the recommended way to make the secret key available in the AEM application?

- A. Use a context aware configuration
- B. Read the key value from a property file stored in the code base
- C. Use an environment variable which is then consumed by an OSGi configuration
- D. Read the key value from OSGi configuration stored in run nNdes

**Correct Answer: C**

**Section:**

**Explanation:**

The recommended way to make the secret key available in the AEM application is to use an environment variable which is then consumed by an OSGi configuration. This way, the secret key is not stored in the code

base or in the repository, but rather in a secure and encrypted way in the Cloud Manager environment variables. The OSGi configuration can use a placeholder to reference the environment variable and inject its value at runtime. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/configuring-osgi.html?lang=en#environment-variables>

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/environment-variables.html?lang=en>

#### QUESTION 41

Which option should be used to synchronize user data across publish servers in a publish farm?

- A. Sling Content Distribution
- B. Vault plugin
- C. CURL
- D. Replication Agents

**Correct Answer: A**

**Section:**

**Explanation:**

Sling Content Distribution is a feature that allows synchronizing user data across publish servers in a publish farm. Sling Content Distribution uses agents to distribute content from one instance to another using a publish-publish topology. Sling Content Distribution can be configured to synchronize user data under /home/users and /home/groups paths, as well as their ACLs. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-65/administering/security/sync.html?lang=en>

<https://experienceleague.adobe.com/docs/experience-manager-learn/sites/deploying/slingcontent-distribution.html?lang=en>

**QUESTION 42**

Which type of Cloud Manager tests are enabled for all Cloud Manager production pipelines and cannot be skipped?

- A. Code Quality Testing
- B. Experience Audit Testing
- C. UI Testing
- D. Functional Testing

**Correct Answer: A**

**Section:**

**Explanation:**

Code Quality Testing is a type of Cloud Manager tests that are enabled for all Cloud Manager production pipelines and cannot be skipped. Code Quality Testing checks the code quality of the project using SonarQube and reports any issues or vulnerabilities. Code Quality Testing can fail the pipeline if the code quality does not meet the minimum standards defined by Adobe. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/testing/testing-overview.html?lang=en#testing-types>

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/testing/code-quality-testing.html?lang=en>

**QUESTION 43**

Which tool should a developer use to look up Adobe Identity Management System (IMS) users by email and return their IMS IDs?

- A. Developer Console
- B. User Mapping Tool
- C. IMS Lookup Tool
- D. Cloud Acceleration

**Correct Answer: C**

**Section:**

**Explanation:**

The IMS Lookup Tool is a tool that allows looking up Adobe Identity Management System (IMS) users by email and returning their IMS IDs. The IMS Lookup Tool is available in the Cloud Manager UI under the Tools menu. The IMS Lookup Tool can be used to find the IMS IDs of users who need to be added to Cloud Manager programs or environments. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/usingcloud-manager/ims-lookup-tool.html?lang=en>

<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/usingcloud-manager/adding-users-to-programs-and-environments.html?lang=en>

**QUESTION 44**

What two types of testing are available OOB in AEM Cloud Manager Pipeline? (Select Two.)

- A. Code Quality testing
- B. Performance testing
- C. UI testing
- D. Penetration testing
- E. Integration testing

**Correct Answer: A, C**

**Section:**

**Explanation:**

Code Quality testing and UI testing are two types of testing that are available OOB in AEM Cloud Manager Pipeline. Code Quality testing checks the code quality of the project using SonarQube and reports any issues or vulnerabilities. UI testing checks the functionality and usability of the project using Selenium WebDriver and reports any errors or failures. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/testing/testing-overview.html?lang=en#testing-types>  
<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/testing/code-quality-testing.html?lang=en>  
<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/testing/ui-testing.html?lang=en>

#### QUESTION 45

What is Out of Scope for the Pattern Detector tool, while doing an AEM upgrade?

- A. OSGi bundles exports and imports mismatch
- B. Backward Compatibility with the previous AEM Version
- C. Definitions of Oak indices for compatibility
- D. rep:User nodes compatibility (in context of OAuth configuration)

**Correct Answer: B**

**Section:**

**Explanation:**

Backward Compatibility with the previous AEM Version is out of scope for the Pattern Detector tool, while doing an AEM upgrade. The Pattern Detector tool is a tool that scans an existing AEM code base and identifies patterns that are incompatible with AEM as a Cloud Service or newer versions of AEM on-premise. The tool does not check for backward compatibility with older versions of AEM, as this is not a requirement for upgrading. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/moving/cloudmigration/pattern-detector.html?lang=en>  
<https://experienceleague.adobe.com/docs/experiencemanager-cloud-service/moving/cloud-migration/pattern-detector/patterns-detected.html?lang=en>

#### QUESTION 46

An AEM as a Cloud Service implementation customer wants content to be replicated as soon as the On Time and Off Times are reached.

What must the developer configure?

- A. Enable Auto Replicate via On Off Trigger Configuration
- B. Enable On Time or Off Time via Page Properties
- C. Configure the Publish content tree via Workflow Models

www.VCEplus.io

**Correct Answer: A**

**Section:**

**Explanation:**

The Auto Replicate via On Off Trigger Configuration is a feature that allows content to be replicated as soon as the On Time and Off Times are reached. This feature can be enabled by creating an OSGi configuration for the com.day.cq.wcm.core.impl.servlets.OnOffTimeServlet service. This service listens for changes in the On Time and Off Time properties of pages and triggers replication accordingly. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/replication.html?lang=en#on-off-time>  
<https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/replication.html?lang=en#auto-replication>

#### QUESTION 47

The OSGi configuration is added to a runmode specific configuration "config.author.staging" in AEM as a Cloud Service. The application fails to read the configuration.

What is a possible cause of this issue?

- A. The custom OSGi configuration runmode used (i.e., "config.author.staging") is not supported in AEM as a Cloud service.
- B. OSGi configuration runmodes cannot be installed automatically on AEM as a Cloud Service. We need to install them as a package using the Package manager.
- C. AEM as a Cloud service does not support OSGi configuration runmodes.
- D. Only <service> specific OSGi configuration runmodes like "config.author" or "config.publish" are supported in AEM as a Cloud service.

**Correct Answer: D**

**Section:**

**Explanation:**

AEM as a Cloud Service does not support custom OSGi configuration runmodes, such as "config.author.staging". Only <service> specific OSGi configuration runmodes, such as "config.author" or "config.publish", are supported. These runmodes are used to apply configurations to specific services, such as author or publish. The <service> specific runmodes can be combined with environment variables to achieve different configurations for different environments. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/configuring-osgi.html?lang=en#run-modes>  
<https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/deploying/environment-variables.html?lang=en>

#### QUESTION 48

Which AEM as a Cloud Service role can configure or run pipelines?

- A. Deployment Manager
- B. DevOps
- C. Developer
- D. Program Manager

**Correct Answer: A**

**Section:**

**Explanation:**

The Deployment Manager is a role that can configure or run pipelines in Cloud Manager. The

Deployment Manager can create and edit programs and environments, configure pipelines, start and cancel pipeline executions, and approve or reject deployments to production. The Deployment Manager role requires an IMS ID that is associated with the Adobe Experience Cloud product profile.

Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloudservice/implementing/using-cloud-manager/adding-users-to-programs-andenvironments.html?lang=en#roles-and-responsibilities>  
<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/usingcloud-manager/cloud-manager-user-roles.html?lang=en>

#### QUESTION 49

A customer who is running an AEM application on premise reports that the application is slowing down over time and even crashes. The issues seem to start occurring after a new production deployment. The AEM developer knows that the described symptoms could be caused by a memory leak.

Which two steps should be taken after confirming the problem is related to a memory issue?

(Choose two.)

- A. Open the error log and look for messages with 'OutOfMemoryError'
- B. Create a heap dump for analysis
- C. Analyze the request log and make sure the number of requests are below the expected threshold
- D. Increase the cache ratio of the application
- E. Create a thread dump for analysis

**Correct Answer: B, E**

**Section:**

**Explanation:**

Creating a heap dump and a thread dump are two steps that should be taken after confirming the problem is related to a memory issue. A heap dump is a snapshot of the memory usage of the Java Virtual Machine (JVM) at a given point in time. A heap dump can be used to identify memory leaks, memory consumption patterns, and objects that occupy large amounts of memory. A thread dump is a snapshot of the state of all threads that are part of the JVM process. A thread dump can be used to identify deadlocks, blocked threads, and CPU-intensive threads. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/memory-analysis.html?lang=en>  
<https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/threaddump-analysis.html?lang=en>

#### QUESTION 50

An AEM Developer needs to migrate to AEM as a Cloud Service from AEM version 6.4. The AEM

Developer is not able to run the Best Practice Analyzer.

What is preventing the Best Practice Analyzer from running?

- A. The AEM Developer is not an admin or does not have administrator rights.
- B. Best Practice Analyzer is not supported on AEM version 6.4.
- C. Best Practice Analyzer should run on Publisher instead of Author instance.

**Correct Answer: B**

**Section:**

**Explanation:**

The Best Practice Analyzer is a tool that scans an existing AEM code base and identifies patterns that are incompatible with AEM as a Cloud Service or newer versions of AEM on-premise. The Best Practice Analyzer is not supported on AEM version 6.4 or earlier versions. The Best Practice Analyzer requires AEM version 6.5 or later to run. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/moving/cloudmigration/best-practice-analyzer.html?lang=en>

<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/moving/cloudmigration/best-practice-analyzer/bpa-prerequisites.html?lang=en>

www.VCEplus.io