

Adobe Experience Manager Architect Master

Adobe AD0-E117

Version Demo

Total Demo Questions: 11

Total Premium Questions: 116

Buy Premium PDF

<https://passqueen.com>

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Business Analysis and Strategy	4
Topic 2, Architecture and Design	64
Topic 3, Implementation	14
Topic 4, Maintenance and Operations	34
Total	116

QUESTION NO: 1

A client is in the discovery process for building a React native app, which is a single codebase that runs natively on iOS and Android. The goal is to reduce development costs. The client wants to reduce content creation costs in the new react Native app by reusing as much of the marketing content already publish in their AEM instance.

What should the Architect recommend to meet this requirement?

- A. Experience Fragments
- B. OSGi Bundles
- C. Sling Resource Resolution
- D. Content Fragments

ANSWER: A

Explanation:

Experience Fragments are the appropriate recommendation because they let authors assemble and manage a reusable marketing experience in AEM, then use that managed experience across multiple delivery contexts. An Experience Fragment can contain a coordinated group of AEM components, content, and presentation, while variations enable the marketing team to tailor the same core experience to a particular channel or use case without recreating it from scratch. This supports the stated objective of reducing content-authoring and development costs: existing campaign, promotional, and marketing experiences can be centrally maintained in AEM and reused for the mobile initiative rather than duplicated in a separate content workflow.

Adobe describes Experience Fragments as reusable, stand-alone experiences that can be used across channels. The fragment model also supports maintaining approved content in one location, so subsequent updates can be governed and propagated consistently wherever the reused experience is consumed. For a React Native implementation, the application team can define the required rendering and integration approach while marketing authors continue to manage the reusable experience through familiar AEM authoring processes. See Adobe's [Experience Fragments documentation](#) and its [overview of Experience Fragment capabilities](#).

QUESTION NO: 2

During an optimization session, a client using AEM 6.5 mentions that they face challenges with their AEM Author setup. The client has significant repository growth and can see an increased amount of TAR files in the segmentstore on disk. When editors are working, they see that workflows take up more time. The purging of workflows works as expected. Which two optimizations should an Architect suggest to took into and further investigate? (Choose two.)

- A. Leverage workflow offloading for appropriate workflows
- B. Configure the Maximum Parallel Jobs
- C. Increase the maxFileSize of the TAR segmentstore
- D. Leverage Transient Workflows where possible
- E. Execute an offline tar compaction twice a day

ANSWER: A D

Explanation:

Leverage workflow offloading for appropriate workflows and Leverage Transient Workflows where possible are the appropriate areas to investigate because they directly address workflow-processing load and repository persistence. Workflow offloading allows an AEM author instance to distribute eligible workflow work to designated worker instances. This separates potentially resource-intensive processing from the author environment used by editors, helping preserve author responsiveness while workflow work is being executed. The topology, queue configuration, network access, and workflow steps must be assessed to determine which workflows are suitable for offloading. Adobe documents this capability under [AEM workflow offloading](#).

Transient workflows are especially relevant where workflow history and intermediate runtime data do not need to be retained. Unlike normal workflows, transient workflows do not persist workflow runtime information in the repository. Applying them to suitable short-lived processes therefore reduces repository writes and ongoing segment-store growth, while also reducing the overhead associated with workflow persistence. Existing workflow purging can remove completed historical instances, but transient workflow design prevents unnecessary persistence from being created in the first place. Adobe describes the behavior and appropriate use of transient workflows in its [workflow models documentation](#).

QUESTION NO: 3

On the release day, an AEM code deployment is done on AEM 6.5 on-prem. After the deployment was completed, the users were able to get into the Author instance with no issues. However, users are unable to access any of the sites on public domains. The users are seeing a "502 Bad Gateway" error.

What would be the potential cause of this issue?

- A. Because there was a deploy and cache was cleared, there is an increased load on the AEM Publish Instances.
- B. The replication agents are stuck which is causing load on the publish instances.
- C. The new dispatcher configurations are not pointing at the correct publish instances.
- D. Not all the OSGi bundles restarted after the code deploy causing the publishing instances not to be responsive.

ANSWER: C

Explanation:

The new dispatcher configurations are not pointing at the correct publish instances. A public AEM site request normally travels through the web server and Dispatcher before being forwarded to an AEM Publish render instance. An HTTP 502 Bad Gateway response at the public domain indicates that this gateway layer could not obtain a valid response from its configured upstream backend. Because Author remains accessible, the symptom is specifically consistent with a problem in the public delivery path rather than a general failure of the deployment or of the Author environment.

After a release, Dispatcher configuration changes can inadvertently reference an incorrect Publish hostname, IP address, port, or farm/render definition. The configured Publish endpoint may also be unreachable from the Dispatcher host because of network or load-balancer routing changes. In those circumstances, Dispatcher cannot relay the request to the actual Publish instance serving the site, and the web tier returns a gateway error to visitors. The appropriate operational check is to validate the active Dispatcher farm configuration and its render targets, then confirm connectivity from each Dispatcher/web-server node to the intended Publish instances. Adobe's Dispatcher documentation describes render definitions as the backend AEM instances to which Dispatcher forwards requests and provides the configuration structure used for those targets. See [Adobe Dispatcher Configuration](#) and [Adobe Experience Manager Dispatcher documentation](#).

QUESTION NO: 4

An Architect is preparing a production performance test for an AEM Sites implementation. The website includes anonymous cached pages, authenticated account pages, asset downloads, and a search feature backed by custom Oak indexes.

Which two actions should the Architect include in the performance-test plan? (Choose two.)

- A. Model representative user journeys and traffic distribution
- B. Monitor Dispatcher, Publish, and Oak query behavior during the test
- C. Run all tests only with an empty Dispatcher cache
- D. Measure performance only from Author browser sessions
- E. Disable custom Oak indexes during the test

ANSWER: A B

Explanation:

Model representative user journeys and traffic distribution is required because test results are meaningful only when they reflect the expected mix of anonymous browsing, authenticated actions, asset downloads, and search activity. The plan should include realistic concurrency, pacing, request volumes, and cache states.

Monitor Dispatcher, Publish, and Oak query behavior during the test is also required. Measuring only browser response times does not identify whether slow behavior originates in cache misses, Publish resource saturation, inefficient queries, or index usage. Correlating end-to-end measurements with infrastructure and query telemetry identifies the actual capacity constraint.

Adobe recommends testing AEM in a production-like environment and evaluating the complete workload. See the [Performance Guidelines for AEM](#) and [Queries and Indexing documentation](#).

QUESTION NO: 5

A news agency editor wants to publish news articles to an intermediate Publish (preview) instance for internal review before publishing the article to production Publish instance.

Which two steps should the Architect recommend? (Choose two.)

- A. Create a closed user group (CUG) for editors
- B. Create a reverse replication agent on preview server
- C. Create a separate live copy to preview content
- D. Set up a new replication agent for preview server
- E. Create a custom publish workflow

ANSWER: C D

Explanation:

Create a separate live copy to preview content provides a distinct preview-content branch while retaining a managed relationship to the source content. AEM Multi Site Manager (MSM) live copies are intended for maintaining related copies of site content and can inherit updates from their blueprint while allowing rollout and inheritance behavior to be controlled. This gives editors a dedicated version of the article that can be reviewed in the preview publishing path before the production version is released.

Set up a new replication agent for preview server establishes the required Author-to-preview-Publish delivery path. In a traditional AEM deployment, replication agents define the publish endpoint and transport configuration used when activated content is delivered from Author to a particular Publish instance. A separate agent aimed at the preview server allows the preview live-copy content to be activated to that intermediate environment independently of production publishing. Together, the managed preview content branch and its dedicated replication destination support an editorial review stage without changing the normal production publishing destination.

Adobe describes live-copy relationships and rollout management in the [Multi Site Manager documentation](#). Replication-agent configuration and Author-to-Publish content delivery are documented in [Replication](#).

QUESTION NO: 6

An AEM Assets implementation requires a controlled approval process before downloaded brand assets can be used by external agencies. The customer needs reviewers to approve or reject each asset and needs the approval state to be visible to authors.

Which approach should the Architect recommend?

- A. Use an AEM Assets review workflow with asset metadata or workflow status
- B. Create a separate DAM folder for each reviewer

- C. Allow external agencies to approve assets directly on Publish
- D. Use asset versions as the approval indicator

ANSWER: A

Explanation:

Use an AEM Assets review workflow with asset metadata or workflow status to govern approval. A workflow can route an uploaded asset to the appropriate reviewer group, support approve and reject decisions, and update the asset's status when the review is complete. Authors can then identify assets that are approved for external use without relying on informal communication or folder naming conventions.

Permissions should ensure that external agencies can access only approved assets through the selected delivery mechanism. Adobe documents workflow capabilities for Assets in the [Assets workflows documentation](#).

QUESTION NO: 7

An AEMaaCS customer wants to have a sub section in site to accept data from an end user, which may be text as well as documents. The site is expected to have high data input which would need moderation and masking before making it live for public access. Due to a legal content policy, the data collected through the site cannot be shared outside the organization. What would an architect recommend?

- A. Utilize Reverse Replication through publish servers
- B. Utilize backward Sling Content Distribution from publish server
- C. Utilize data storage/APIs outside of AEMaaCS
- D. Utilize AEM communities with MSRP topology

ANSWER: C

Explanation:

Utilize data storage/APIs outside of AEMaaCS is the appropriate recommendation because the submitted text and documents are regulated data that must remain under the customer organization's control. The public AEM site can provide the submission experience, while a customer-managed ingestion API sends content directly to an organization-controlled data store and moderation workflow. That workflow can perform malware scanning, validation, personally identifiable information masking, and human moderation before any approved, sanitized content is made available for public delivery.

This design also separates a potentially high-volume, write-heavy submission workload from AEM's content delivery responsibilities. AEM as a Cloud Service is designed as a cloud-native content platform with immutable deployment patterns and independently scalable services; an external system of record is better suited to retaining sensitive submissions, tracking review states, enforcing retention rules, and integrating enterprise compliance controls. Only the final approved representation, if required for public display, should be published through the appropriate AEM content process. This minimizes regulated-data exposure in AEMaaCS while preserving a governed path from user submission to public content. Adobe's Cloud Service overview is available at [Adobe Experience Manager as a Cloud Service overview](#), and Adobe's security guidance is available at [Security in AEM as a Cloud Service](#).

QUESTION NO: 8

In a client's test environment AEM QA Testers from multiple groups and regions are testing and making requests to the AEM application. Due to the sheer number of hosts making connections, an Architect needs a quick and easy way to allow any hosts to make requests to the AEM application.

Which approach should the Architect take to meet this requirement for this non-production environment?

- A. Add GET as a Filter Method in the Apache Sling Referrer OSGi Configuration
- B. Use Allowlists instead of Blocklists in the Dispatcher Configuration

- C. Restrict Clients that can flush the cache in the Dispatcher Configuration
- D. Enable Allow Empty in the Apache Sling Referrer OSGi Configuration

ANSWER: D

Explanation:

Enable Allow Empty in the Apache Sling Referrer OSGi Configuration. AEM uses the Apache Sling Referrer Filter to help protect endpoints that modify content by evaluating the HTTP `Referer` header on configured request methods. In a shared, non-production testing environment, requests generated directly by QA tools, test clients, regional test infrastructure, or bookmarked application URLs commonly have no referrer header. Enabling the `allow.empty` setting permits those requests without requiring administrators to maintain an extensive list of individual tester hosts or test-system domains. This provides the requested low-administration approach for an environment where many changing client sources must access the application. The setting is appropriate only where the reduced referrer validation is accepted for the test environment; it should be governed separately from production security requirements. Apache Sling documents that the referrer filter can be configured to allow requests with an empty referrer, and AEM exposes this behavior through the Apache Sling Referrer Filter OSGi configuration. See the [Apache Sling Referrer Filter documentation](#) and Adobe's [OSGi configuration documentation](#).

QUESTION NO: 9

An AEM 6.5 customer has a Publish farm behind Dispatcher. A new personalization integration requires authenticated visitors to receive user-specific responses, while anonymous visitors must continue receiving cached public pages.

Which two actions should the Architect recommend? (Choose two.)

- A. Ensure personalized responses are not cached as public Dispatcher content
- B. Configure authentication handling consistently across all Publish instances
- C. Cache authenticated responses using the same cache key as anonymous pages
- D. Disable authorization checks on Publish instances
- E. Replicate each authenticated user profile to Dispatcher

ANSWER: A B

Explanation:

Ensure personalized responses are not cached as public Dispatcher content is required because a response containing user-specific data must not be served to another visitor from the shared cache. The cache rules and response behavior must separate personalized requests from anonymous cacheable delivery.

Configure authentication handling consistently across all Publish instances is also required because authenticated requests can be routed to any healthy Publish instance. The authentication design, user synchronization, and token or SSO configuration must support consistent authorization throughout the Publish farm.

Anonymous public content can remain cacheable, preserving Dispatcher's performance benefit. Adobe documents Dispatcher caching and request handling in the [Dispatcher Configuration documentation](#) and authentication configuration in the [Single Sign-On documentation](#).

QUESTION NO: 10

A large AEM enterprise site is implementing authentication and requires a true optimal load balancing across the site's multi AEM publish instances. Which approach should an Architect take to meet this requirement?

- A. Configure Sticky Connections
- B. Configure the Dispatcher to run with a Dedicated System User

- C. Enable Encapsulated Token Option
- D. Enable Round Robin processing in Dispatcher configuration

ANSWER: C

Explanation:

Enable Encapsulated Token Option is the appropriate approach because it makes authenticated session information portable across AEM Publish instances. With encapsulated-token authentication, AEM places the authenticated user's credentials in an encrypted, signed token that is returned to the browser in a cookie. A subsequent request can be authenticated by whichever Publish instance receives it, provided the instances share the required cryptographic configuration. This removes the requirement for a user's requests to remain tied to one specific Publish server and allows the load balancer or Dispatcher tier to distribute authenticated traffic effectively among the available Publish instances.

For an enterprise topology, the Architect should ensure that the Publish instances participating in this authentication setup use a consistent key configuration and that the token cookie is sent appropriately for the site's domains and HTTPS configuration. The approach is especially useful where a load-balanced Publish farm must preserve a seamless authenticated experience while allowing requests to be routed to any healthy node. Adobe documents encapsulated-token authentication as an AEM SSO configuration mechanism and describes its use in clustered or load-balanced environments. See Adobe's [Single Sign-On configuration documentation](#) and the [encapsulated token authentication section](#).

QUESTION NO: 11

A mobile application consumes product content from AEM as a Cloud Service through GraphQL. The customer wants to expose only approved queries and ensure the public delivery endpoint is available through Dispatcher.

Which two actions should the Architect take? (Choose two.)

- A. Create persisted GraphQL queries for approved application use cases
- B. Allow the persisted-query execution endpoint in Dispatcher filters
- C. Expose the GraphQL IDE publicly for mobile application requests
- D. Allow anonymous POST requests to arbitrary GraphQL query endpoints
- E. Replicate GraphQL queries from Publish back to Author

ANSWER: A B

Explanation:

Create persisted GraphQL queries for the approved application use cases. Persisted queries are stored and managed in AEM and can be executed through a defined URL, avoiding exposure of arbitrary GraphQL query execution to the public client.

Configure Dispatcher to allow the persisted-query execution endpoint. A request filter that blocks the required GraphQL execution path would prevent the mobile application from retrieving content even when the persisted query exists. The endpoint should be allowed narrowly rather than opening unrelated GraphQL paths. Adobe documents persisted queries and their delivery configuration in the [AEM GraphQL API documentation](#).