PHYSICAL ARCHITECTURE OF AN AEM DEVELOPMENT

A standard Adobe Experience Manager (AEM) deployment consists of three separate tiers, called Author, Publish and Dispatcher. This document describes these three different tiers, and how Axis41 uses Amazon Web Services (AWS) to deliver a scalable, reliable and responsive AEM implementation.

HIGH LEVEL

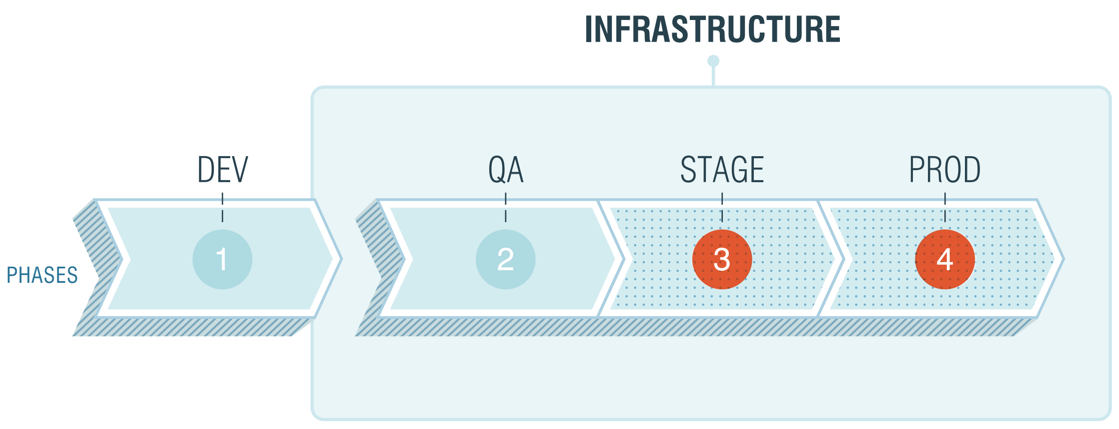
The Author tier is a Java application, and is the main interface that content creators and developers will interact with. As content is approved, appropriate workflows will be triggered that cause the content to be moved from the Author tier into the Pubish tier.

The Publish tier is also a Java application. The workhorse of the AEM suite, it takes all the dynamic content and generates static data ready to be delivered to browsers. It is used between the other two tiers to allow them to focus on performing their specific duties in the most efficient way possible.

The Dispatcher tier is not a Java application; it runs inside the Apache Web server, caching the result from the Publish tier to disk. This allows the Dispatcher to achieve blazing speeds when delivering rendered content back to the end user.

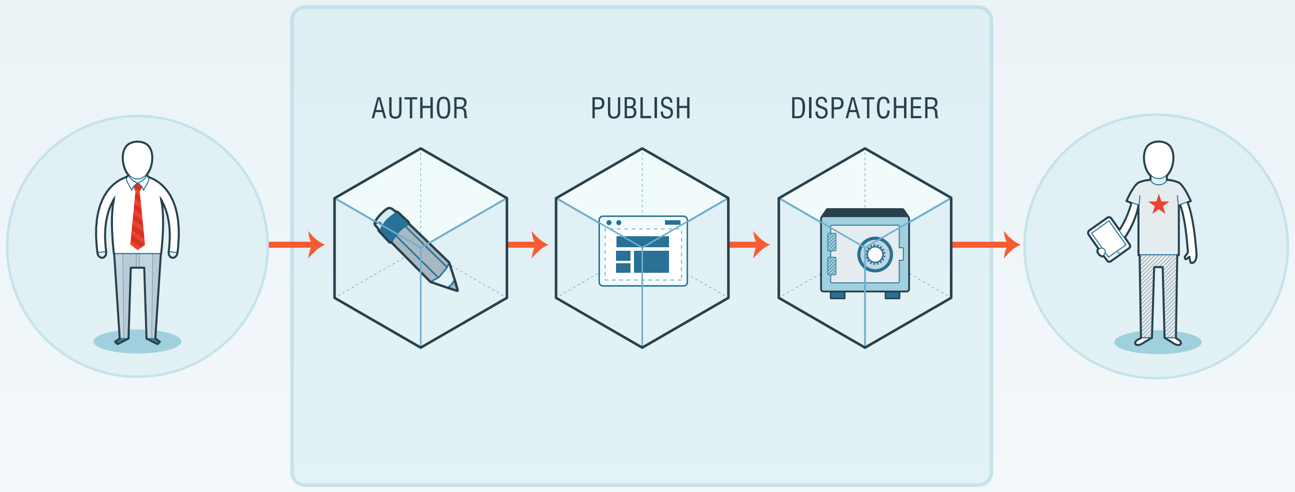
Thanks to the flexibility and control offered by AWS, Axis41 is able to focus on delivering a consistent experience without having to worry about uptime, hardware upgrades or scaling (on either axis). Your needs will differ from other implementations, and this unique understanding of how to most effectively combine AEM and AWS is what sets Axis41 apart.

Over the lifecycle of an AEM implementation, you will be dealing with four different deployments of this three-tier system; we have labeled them here as Dev, QA, Stage and Prod. Each of these will contain the three tiers listed above. Our white paper "The Lifecycle of an AEM Implementation" shares more details on how these stages interact, as well as some of the tradeoffs that need to be considered when selecting your development methodology.



2

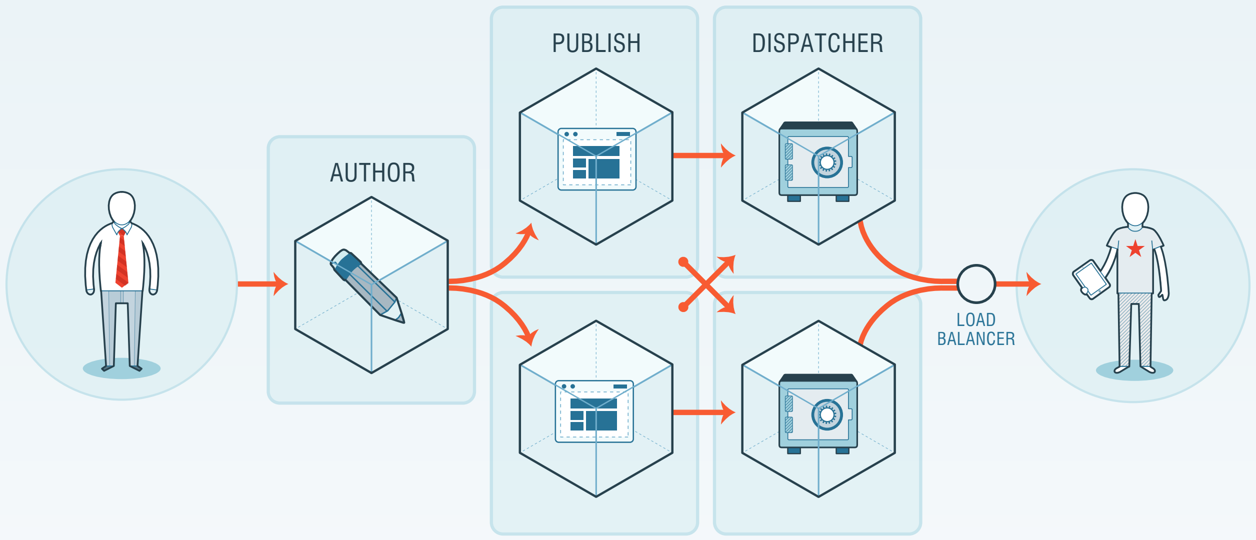
QA ENVIRONMENT



The diagram above depicts the details of the QA environment. During QA, all three tiers are deployed to a single EC2 instance, each running on different ports. This mirrors the workflow used during the development cycle, while allowing multiple parties to interact with the tiers and ensure that the development is proceeding in the right direction.

Because the QA environment is not for general consumption, we can minimize costs by folding all three tiers into the one EC2 instance. In the Stage and Production environments, this would hinder the ability to scale the environment, so it is avoided.

STAGE & PRODUCTION ENVIRONMENT



The diagram above depicts the details of the Stage and Production environments. Here, we have each machine serving a specific role, as well as an Elastic Load Balancer (ELB), which spreads incoming requests evenly across all Dispatchers.

This provides three concrete benefits:

- It gives us the ability to scale each individual tier to meet the specific needs of your site. If you are creating so much content that the Publish tier begins to struggle, we can spin up more Publish hosts to provide the dynamic data to the Dispatchers. If you experience unusually large visitor counts during a peak period, additional Dispatchers can be added behind the ELB.
- It provides redundancy, giving each tier the ability to have multiple machines in physically disparate data centers. The level of redundancy can be tuned to match the budget and requirements of your business.
- Each tier has different performance characteristics, and splitting them this way allows us to tune each tier's servers to meet its specific needs.

CONSIDERATIONS

The QA and Stage environments do not require 100 percent uptime. QA need only be running during the hours when QA resources are actually available. Stage is used during the pre-release phase of the project lifecycle for performance testing, user acceptance testing and smoke testing. This allows Axis41 to significantly minimize your costs without compromising any phase of the project.