DEVELOPERS LIVE

Environment Variables with AEM as a Cloud Service

Shankari Panchapakesan | Sr. Product Manager

Raul Hudea | Sr. Computer Scientist

Key Takeaways

1

Adaptive code makes for easy maintenance

You only have to take care of the setup and configuration of your application once

2

Improve security by following best practices

Separate code from configuration and avoid including sensitive information in your version control

3

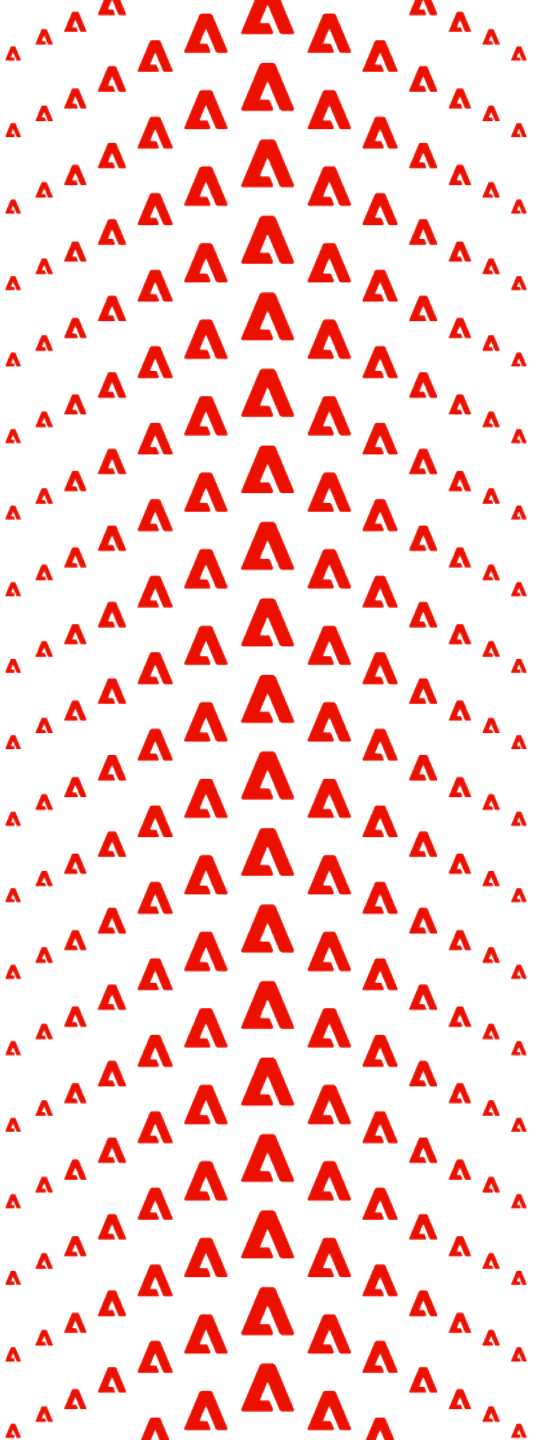
Avoid expensive production mistakes

Enable different configurations on a testing environment vs. Production to avoid costly mistakes

Adobe

Agenda

- Environment Variables Overview
- Demo
- Key Takeaways
- Q&A

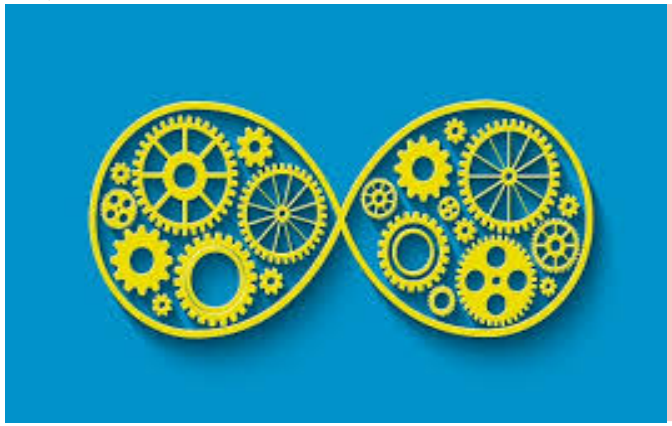


Environment Variables Overview

Environment Variables Requirement

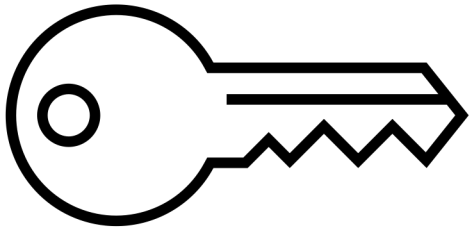
The Adobe I/O CLI plays an integral role in development on AEM as a Cloud Service as it provides developers the ability to manage Cloud Manager environment & pipeline variables from the CLI

Pipeline & Standard Environment Variables Overview

	Quick Tips	Use Cases
Pipeline Variables 	• Used to vary behavior of build process • Accessible inside the build process	• Dev minification is set differently on Dev vs Stage or Prod • Disable caching on Dev
Standard Environment Variables 	• Provided to the runtime environment • Used in OSGi configuration	• Store passwords outside of version control, instead use a reference. • Connect your AEM application with different external endpoints

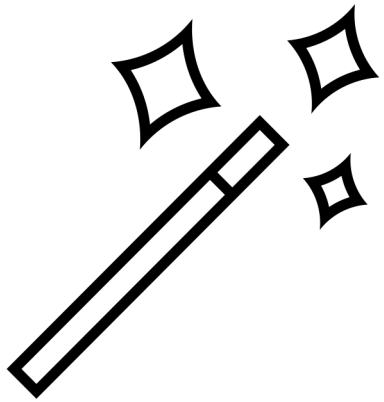
Demo

Using Environment Variables



Adobe I/O API Integration

Adobe I/O CLI



Code

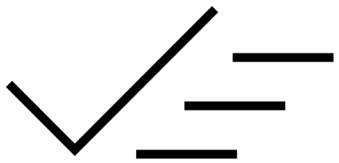
Pipeline Variables

- Changing build behavior based on pipeline variables

Standard Environment Variables

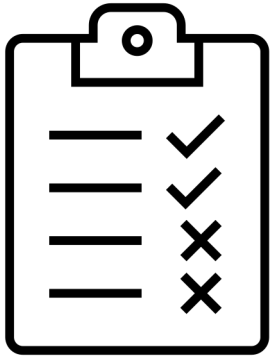
- Changing application behavior based on standard environment variables

Standard Environment Variables – OSGI configuration



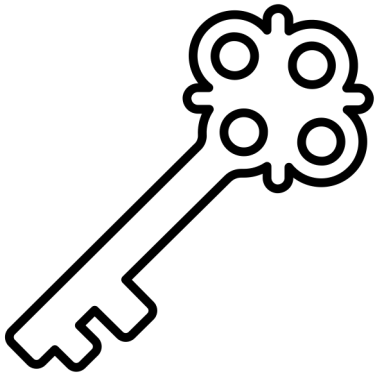
Inline values

```
{ "common" : "value" }
```



Environment specific values

```
{ "endpoint" : "$[env:ENDPOINT]" }
```



Environment secrets

```
{ "authToken" : "$[secret:AUTH_TOKEN]" }
```

Key Takeaways

Key Takeaways

1

Adaptive code makes for easy maintenance

You only have to take care of the setup and configuration of your application once

2

Improve security by following best practices

Separate code from configuration and avoid including sensitive information in your version control

3

Avoid expensive production mistakes

Enable different configurations on a testing environment vs. Production to avoid costly mistakes

Adobe

THANK YOU!
Q&A

Gotchas

- Make sure that you have Developer or System Admin role in your IMS Org
- You can only set environment variables if your environment is in “Ready” state

Appendix

Setup Adobe I/O CLI

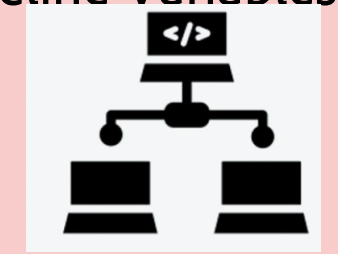
Environment Variables

Cloud Manager Plugin for the Adobe I/O CLI

Adobe Developer Console

Configuring OSGi for AEM as a Cloud Service

Pipelin  d Enviro  ables Overview

	Quick Tips	Use Cases
Pipeline Variables 	• Used to vary behavior of build process • Accessible inside the build process	• Dev minification is set differently on Dev vs Stage or Prod • Disable caching on Dev
Standard Environment Variables 	• Provided to the runtime environment • Used in OSGi configuration	• Store passwords outside of version control, instead use a reference. • Connect your AEM application with different external endpoints