



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

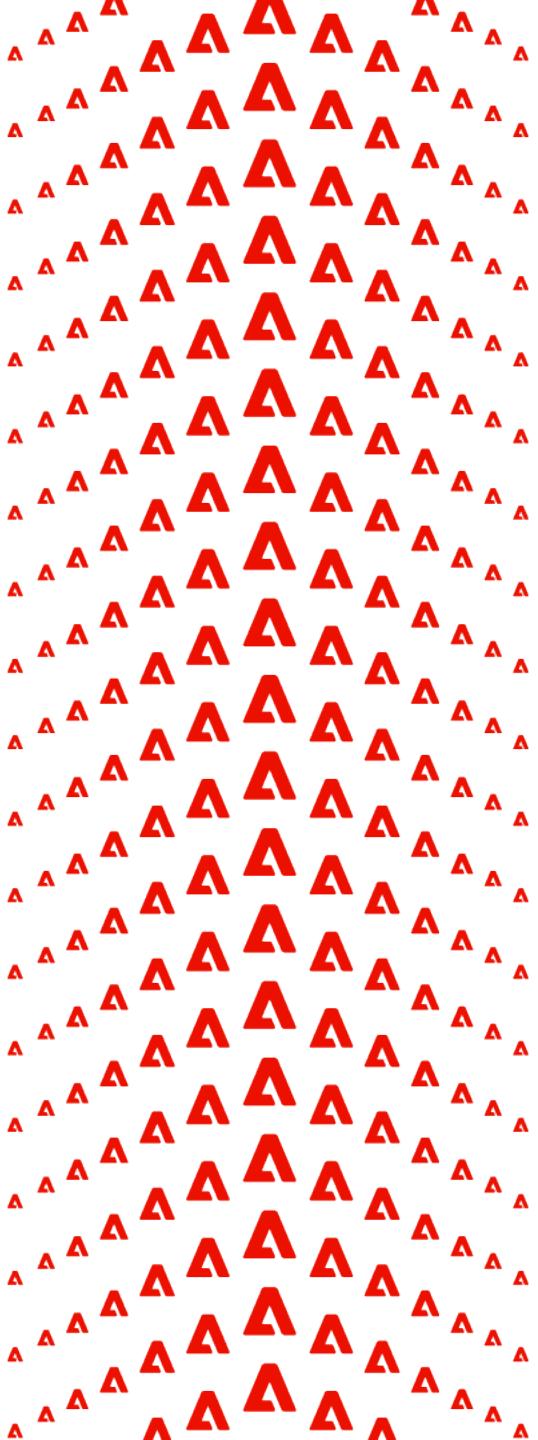
Agenda

Environment Variables Overview

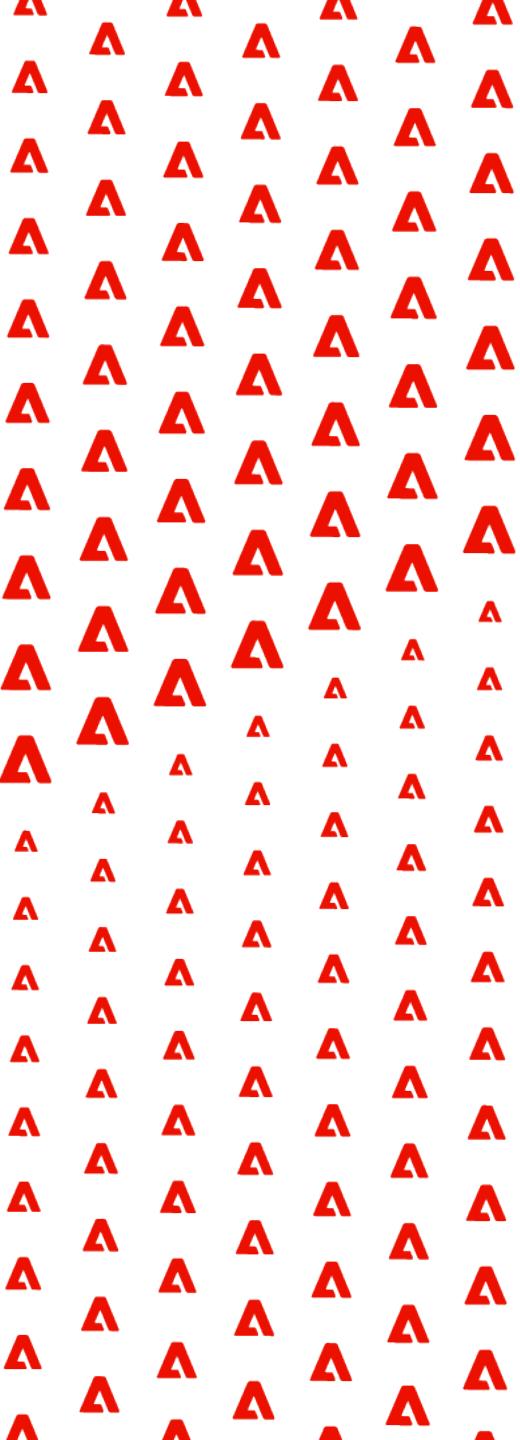
Demo

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



```
/*groupsalloc);
EXPORTSYMBOL(groupsalloc);
void groups_free(struct group_info *group_info)
{
    if (groupInfo->blocks[0] != group_info->small_block) {
        int i;
        for (i = 0; i < group_info->nblocks; i++)
            freepage((unsigned long)groupInfo->blocks[i]);
        for (i = 0; i < group_info->nblocks; i++)
            freepage((unsigned long)groupInfo->blocks[i]);
    }
    kfree(groupInfo);
    kfree(groupInfo);
}

/*active = modifier on
 * (ion at the end - add back the de
 * select=1
 * ob.select=1
 * / export the groupInfo to a user-space array */
ext.scene.objects.active = modifier;
selected" + str(modifier.ob));
/* export the groupInfo to a user-space array */
selected objects, the last const struct group_info *group_info);
static int groups_toUser(gid_t user *groupList,
                        const struct group_info *group_info)
{
    int i;
    unsigned int count = groupInfo->nGroups;
    int len;
    unsigned int count = groupInfo->nGroups;
    for (i = 0; i < group_info->nblocks; i++) {
        unsigned int ccount = min(NGROUPSPERBLOCK, count);
        for (i = 0; i < group_info->nblocks; i++) {
            unsigned int len = ccount * sizeof(*groupList);
            unsigned int ccount = min(NGROUPSPERBLOCK, count);
            unsigned int len = ccount * sizeof(*groupList);
            if (copyToUser(groupList, group_info->blocks[i], len))
                return -EFAULT;
            if (copyToUser(groupList, group_info->blocks[i], len))
                return -EFAULT;
        }
    }
}
```

Pipeline & Standard Environment Variables Overview

Pipeline Variables



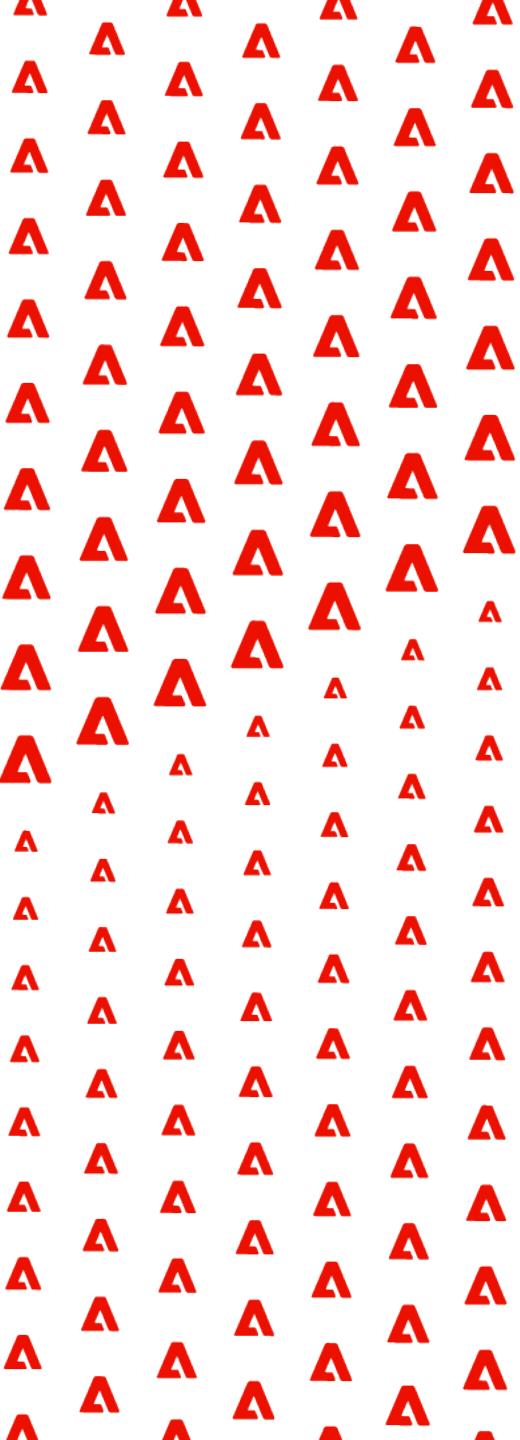
Quick Tips	Use Cases
<ul style="list-style-type: none">Used to vary behavior of build processAccessible inside the build process	<ul style="list-style-type: none">Dev minification is set differently on Dev vs Stage or ProdDisable caching on Dev

Standard Environment Variables

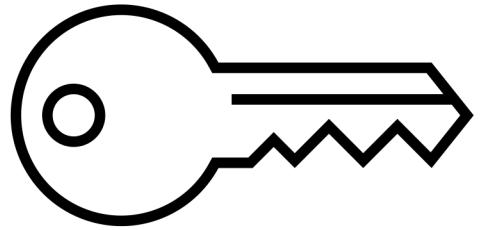


<ul style="list-style-type: none">Provided to the runtime environmentUsed in OSGi configuration	<ul style="list-style-type: none">Store passwords outside of version control, instead use a reference.Connect your AEM application with different external endpoints
--	---

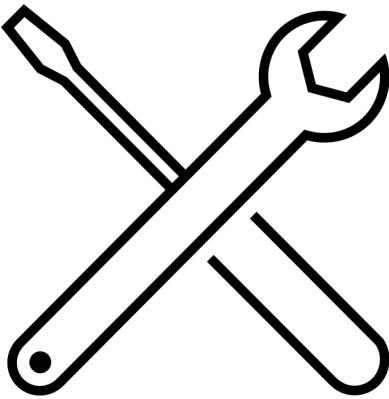
Demo



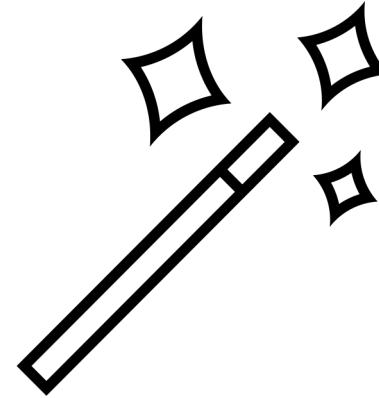
Using Environment Variables



Adobe I/O API
Integration



Adobe I/O CLI



Code

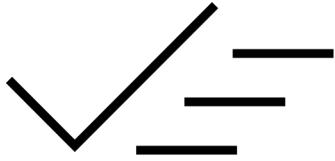
Pipeline Variables

- Changing build behavior based on pipeline variables

Standard Environment Variables

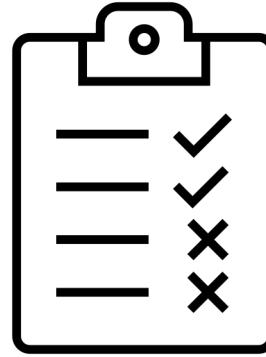
- Changing application behavior based on standard environment variables

Standard Environment Variables – OSGI configuration



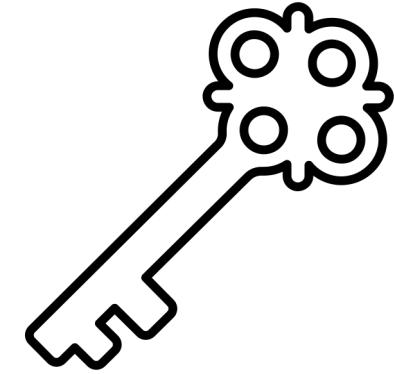
Inline values

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



Environment specific values

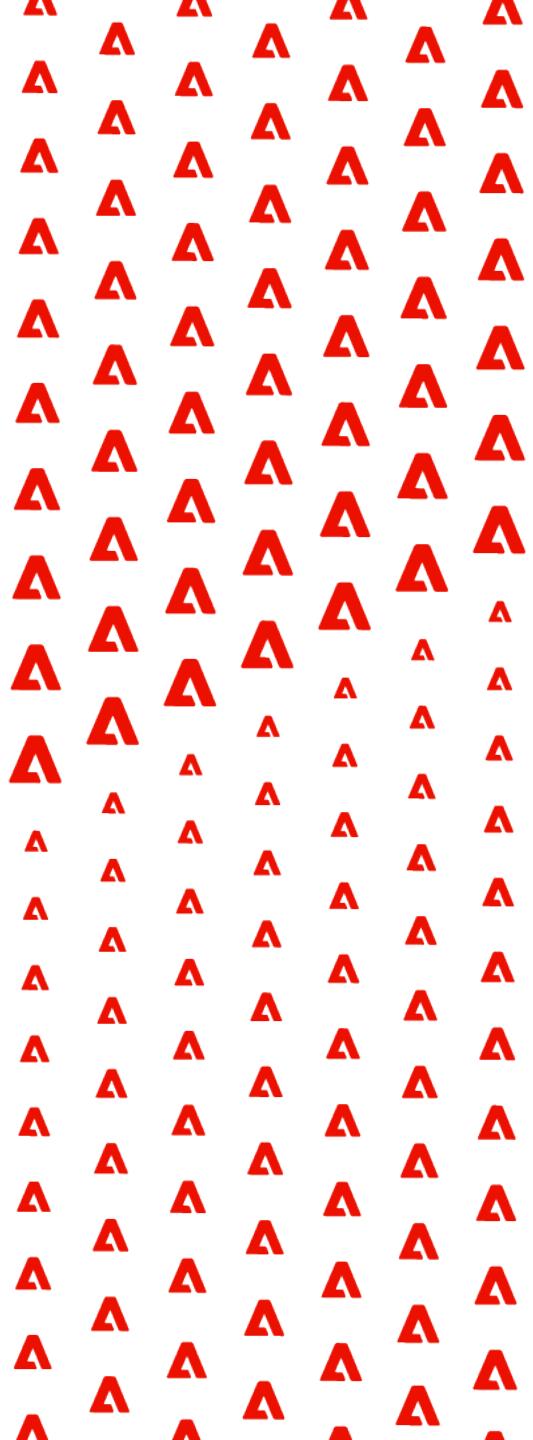
```
{ "endpoint" : "${env:ENDPOINT}" }
```



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

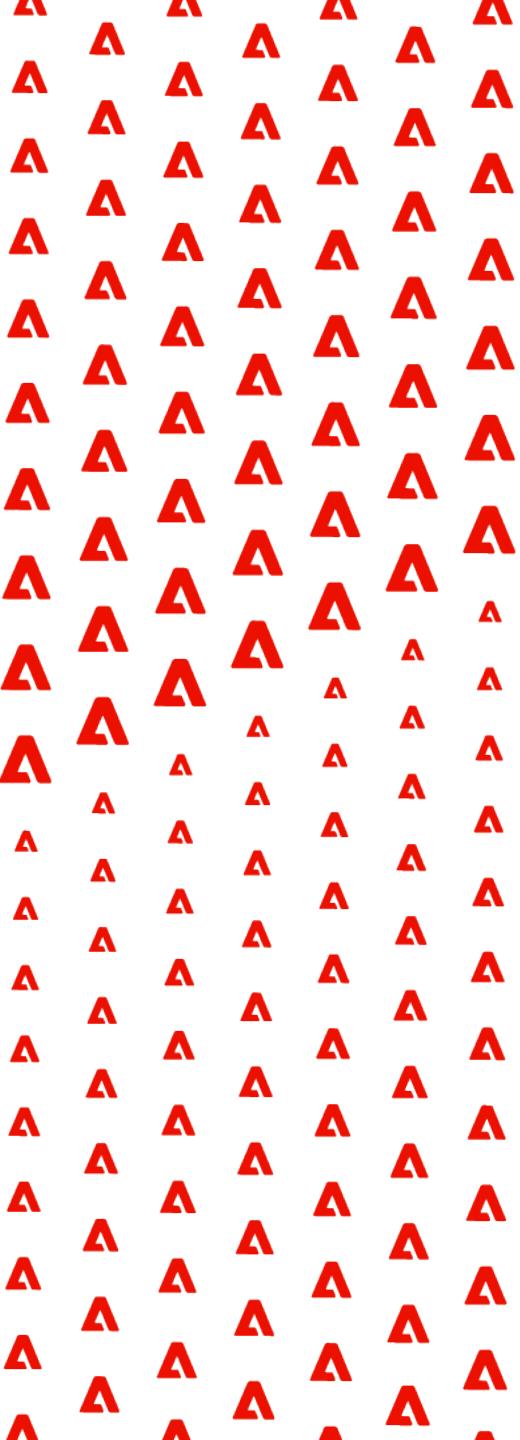
3

Avoid expensive
production
mistakes



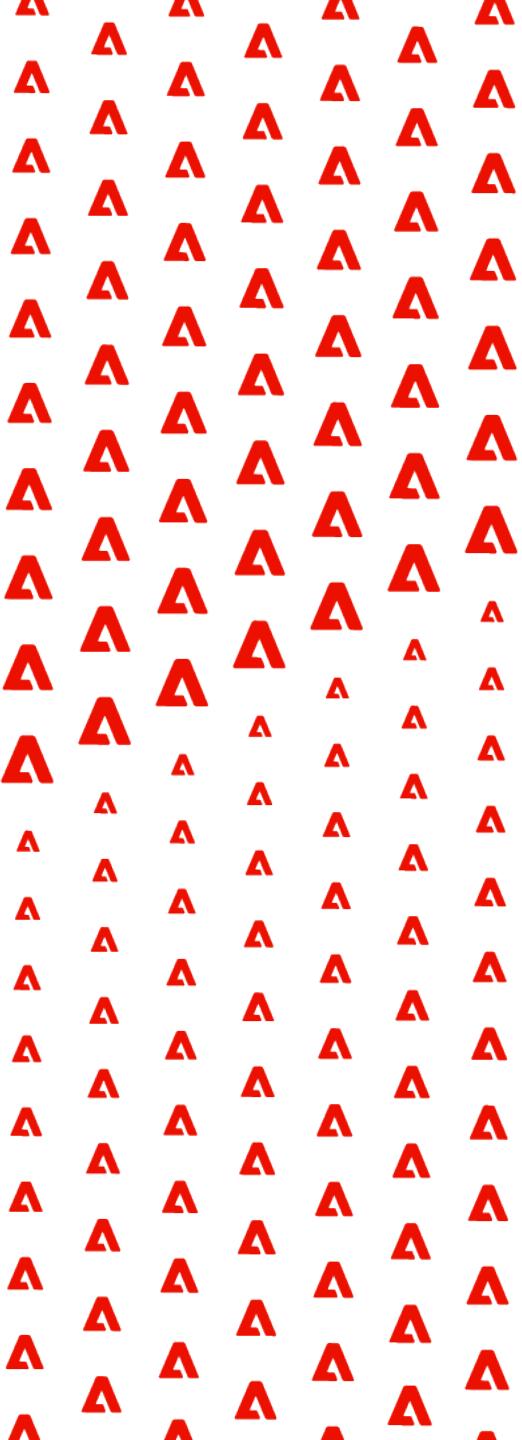
Enable different
configurations on a
testing
environment vs.
Production to avoid
costly mistakes

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

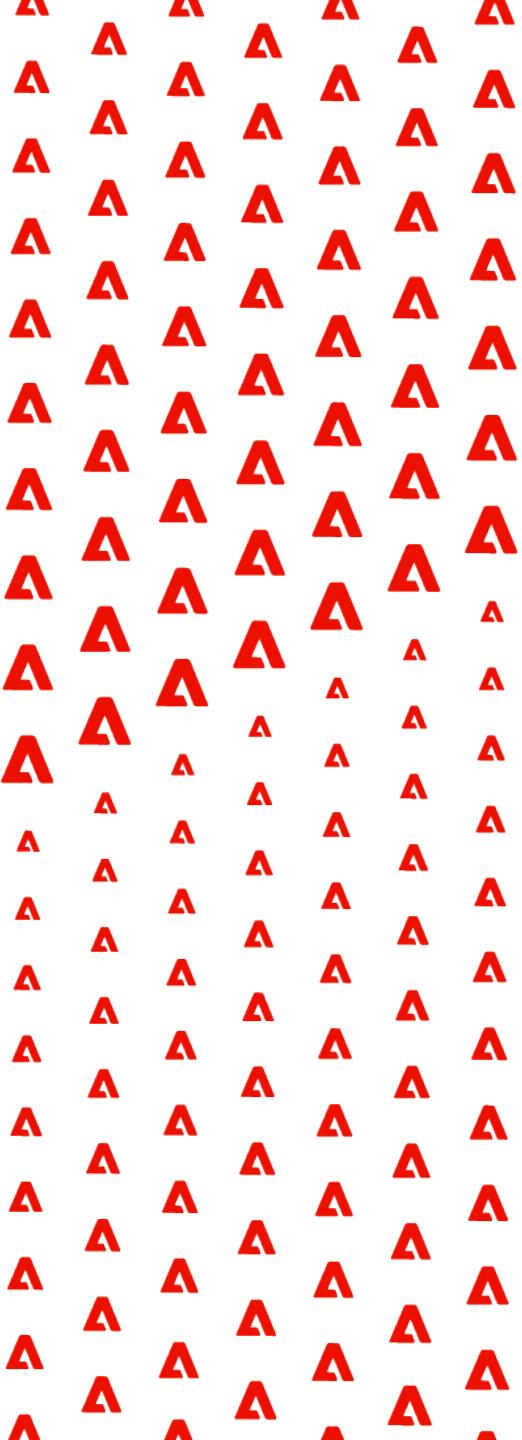
[Setup Adobe I/O CLI](#)

[Environment Variables](#)

[Cloud Manager Plugin for the Adobe I/O CLI](#)

[Adobe Developer Console](#)

[Configuring OSGi for AEM as a Cloud Service](#)

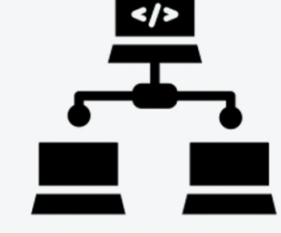


Pipeline



Environment Variables Overview



	Quick Tips	Use Cases
Pipeline Variables 	<ul style="list-style-type: none">Used to vary behavior of build processAccessible inside the build process	<ul style="list-style-type: none">Dev minification is set differently on Dev vs Stage or ProdDisable caching on Dev
Standard Environment Variables 	<ul style="list-style-type: none">Provided to the runtime environmentUsed in OSGi configuration	<ul style="list-style-type: none">Store passwords outside of version control, instead use a reference.Connect your AEM application with different external endpoints