



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
10-12 SEPTEMBER 2018

# Maven Archetypes for AEM

Tobias Richter, Stefan Seifert; pro!vision GmbH

# About the Speakers



Tobias Richter

- DevOps Engineer  
@ pro!vision



Stefan Seifert

- AEM Developer
- Apache Sling PMC
- CTO of pro!vision GmbH



<https://www.pro-vision.de>

1. About Maven Archetypes for AEM
2. Create project(s)
3. Setup full AEM infrastructure (AWS)
4. Live Demo! 

# Archetypes for AEM Projects

# What is a Maven Archetype

- Maven project templating toolkit
- Bootstrap new projects with parameters
- AEM needs complex multi-module projects



# Archetypes available

- **Maven Archetype by Adobe**

<https://github.com/Adobe-Marketing-Cloud/aem-project-archetype>

- **Maven Archetype by wcm.io**

<http://wcm.io/tooling/maven/archetypes/aem/>

- **Lazybones templates \* by Adobe Consulting**

<https://github.com/Adobe-Consulting-Services/lazybones-aem-templates>

\* This is not a Maven Archetype

# Archetype comparison

Feature	Adobe AEM Archetype	wcm.io AEM Archetype	AEM Lazybones Templates
AEM version	6.3SP2, 6.4	6.3SP2, 6.4	6.3, 6.4
Feature flags	✗ No	✓ Yes	✓ Yes
Templates & Core Component examples	✓ Yes	✓ Yes	✗ No
AEM Mocks unit test examples	✗ No	✓ Yes	✗ No
Editable Templates	✓ Yes	✓ Yes	✓ Yes
FileVault XML Layout	✓ Yes	✓ Yes	✓ Yes
Sling-Initial-Content JSON Layout	✗ No	✓ Yes	✗ No
Context-Aware Configuration	✗ No	✓ Yes	✗ No
CONGA-enabled for Deployment	✗ No	✓ Yes	✗ No
Webpack-based Frontend Build	✗ No	✓ Yes	✗ No
Integration Test Examples	✓ Yes	✗ No	✗ No

- Interactive mode

```
mvn archetype:generate \  
  -DarchetypeGroupId=io.wcm.maven.archetypes \  
  -DarchetypeArtifactId=io.wcm.maven.archetypes.aem \  
  -DarchetypeVersion=2.0.2
```



- Batch mode  
(recommended)

```
mvn archetype:generate -DinteractiveMode=false \  
-DarchetypeGroupId=io.wcm.maven.archetypes \  
-DarchetypeArtifactId=io.wcm.maven.archetypes.aem \  
-DarchetypeVersion=2.0.2 \  
-DprojectName=adaptToDemo2018 \  
-DgroupId=to.adapt \  
-DartifactId=to.adapt.demoapp \  
-Dversion=1.0.0-SNAPSHOT \  
-Dpackage=to.adapt.demoapp \  
-DpackageGroupName=adaptTo \  
-DdaemonAuthorPort=4502 \  
-DdaemonPublishPort=4503 \  
-DoptionAemVersion=6.4 \  
-DoptionAemServicePack=n \  
-DoptionSlingModelsLatest=n \  
-DoptionSlingInitialContentBundle=n \  
-DoptionEditableTemplates=y \  
-DoptionMultiBundleLayout=n \  
-DoptionContextAwareConfig=n \  
-DoptionWcmioHandler=n \  
-DoptionAcsCommons=n \  
-DoptionFrontend=y
```

Feature Flags

Parameter documentation:

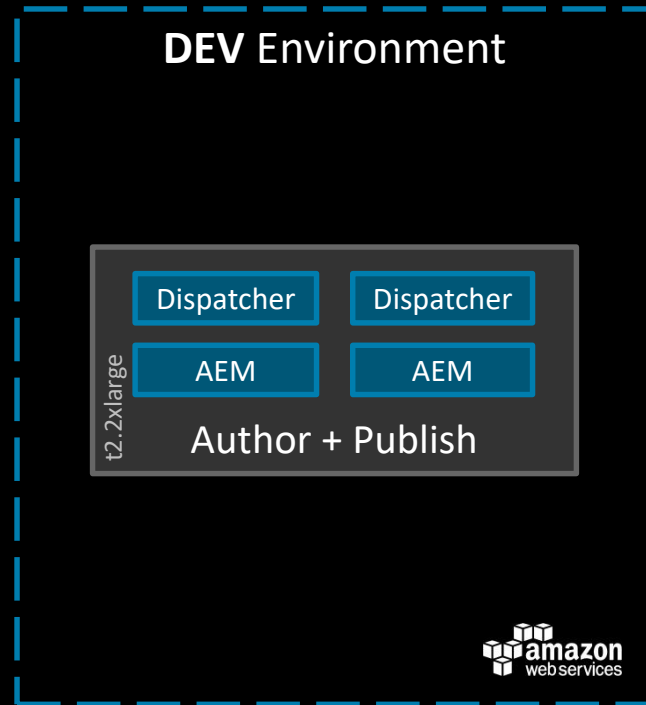
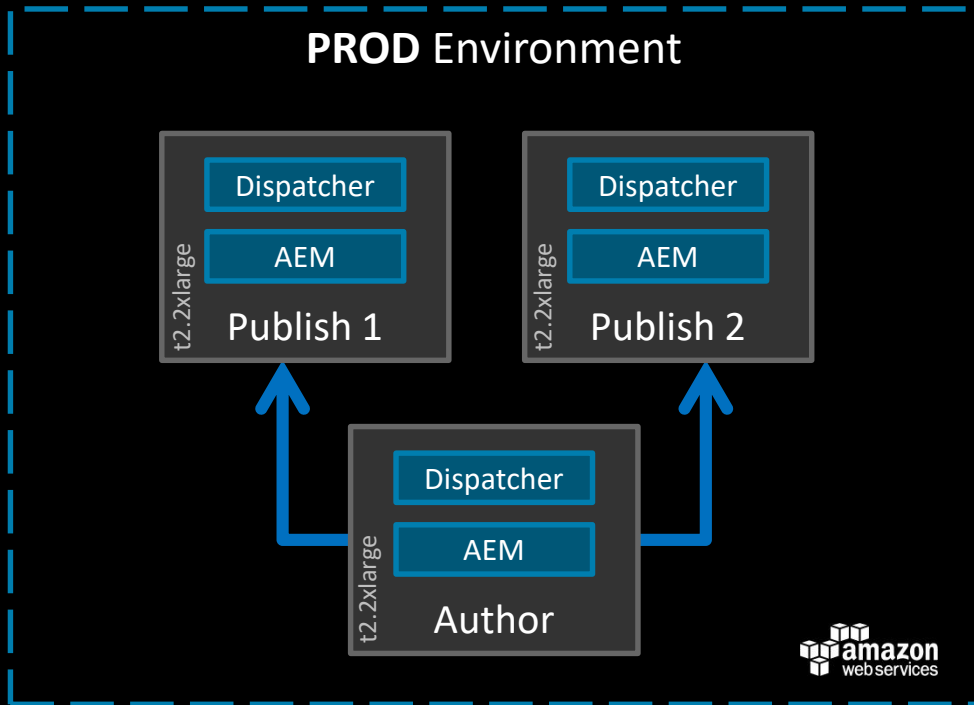
<http://wcm.io/tooling/maven/archetypes/aem/usage.html>

1. Create project
2. Build it
3. Deploy it locally
4. Publish it to Maven Repository

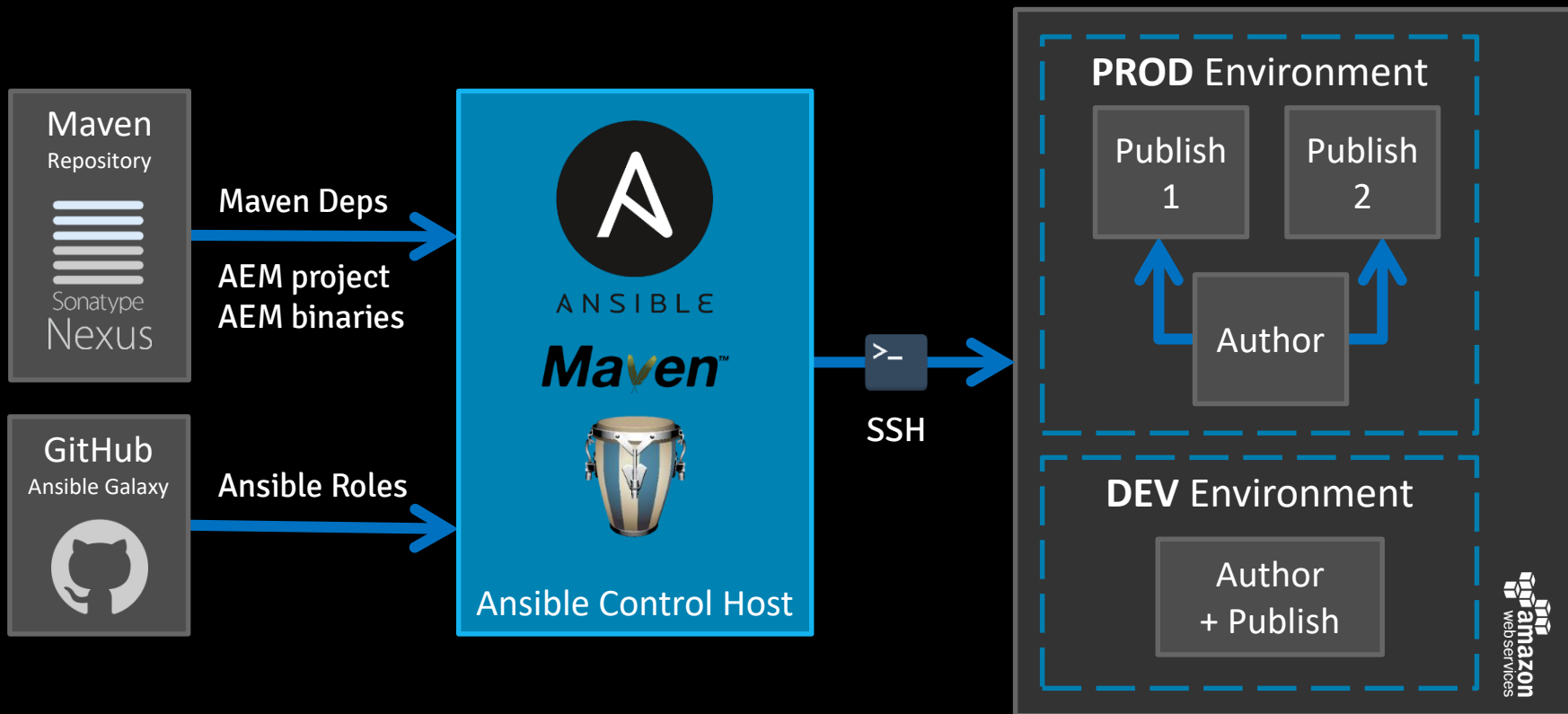


# AEM infrastructure overview

# What to set up



# Big picture



# Archetype for AEM Config Management & Deployment

# Two git repositories

## *AEM project*

- Templates and components
- Java Code
- Frontend Code
- Sample Content
- CONGA Definition
- Local deployment

Create with:  
AEM project  
archetype

 **git** Repository 1

## *Config Management project*

- CONGA Environments
- Configuration per stage
- Deployment scripts
- Credentials & keys  
(encrypted)

Create with:  
AEM  
Configuration  
Management  
archetype

 **git** Repository 2

# wcm.io Config Management archetype

Feature	wcm.io Config Mgmt Archetype
AEM version	6.3, 6.4
<b>CONGA</b> Environments	✓ Yes
<b>Ansible</b> Playbooks for automated deployment	✓ Yes
Describe and create AWS infrastructure using <b>Terraform</b>	✓ Yes
Remote or local control host with <b>Vagrant</b>	✓ Yes
Setup AEM infrastructure and deployment in <b>AWS</b>	✓ Yes
Sensitive data encrypted using Ansible vault	✓ Yes

No manual deployment/configuration steps!



## ■ Batch mode

```
mvn archetype:generate -DinteractiveMode=false \  
  -DarchetypeGroupId=io.wcm.maven.archetypes \  
  -DarchetypeArtifactId=io.wcm.maven.archetypes.aem-confmgmt \  
  -DarchetypeVersion=1.0.2 \  
  -DconfigurationManagementName=adapto-demo-2018 \  
  -DprojectName=adaptToDemo2018 \  
  -DgroupId=to.adapt \  
  -DartifactId=to.adapt.demoapp \  
  -Dversion=1.0.0-SNAPSHOT \  
  -DawsMachineSize=large \  
  -DoptionAnsible=y \  
  -DoptionTerraform=y \  
  -DoptionVagrant=y \  
  -DansibleVaultPassword=mypassword \  
  -DadminPassword=myadminpwd \  
  -DmavenRepositoryUrl=http://your-maven-repo-url/ \  
  -DmavenRepositoryUser=user \  
  -DmavenRepositoryPassword=password
```

Feature Flags

Parameter documentation:

<http://wcm.io/tooling/maven/archetypes/aem-confmgmt/usage.html>

1. Create Config Management project
2. Create AWS machines via Terraform

A large blue circle with a white border, containing the text "LIVE DEMO" in white, uppercase, sans-serif font, centered within the circle.

**LIVE  
DEMO**

- Archetype provides “small”, “medium”, “large”
  - mapped to AWS instance types
- All machines tagged properly
- SSH keys
- Security group

1. Run Ansible Playbook
2. Setup AEM and Dispatcher for PROD
3. Deploy everything

A large blue circle with a white border, containing the text "LIVE DEMO" in white, uppercase, sans-serif font, centered within the circle.

**LIVE  
DEMO**

# CONGA concepts



# CONGA abstracts AEM configuration

## *Config Definition in AEM project*

- OSGi configs
- Configs in Content (e.g. replication agents)
- Dispatcher config

Templates with placeholders

 **git** Repository 1

Maintained  
by:  
**Developers**

## *Environments in Config Mgmt*

- IP Addresses
- Number of instances
- Machine sizing
- Domain names
- Credentials

 **git** Repository 2

Maintained  
by:  
**Operations**

# CONGA building blocks for templates

- AEM configuration is complex
- CONGA provides a set of preconfigured “AEM Definitions”
- Best practices ready-to-use
- Customizable

<http://devops.wcm.io/conga/definitions/aem/>

## 1. CONGA Environment

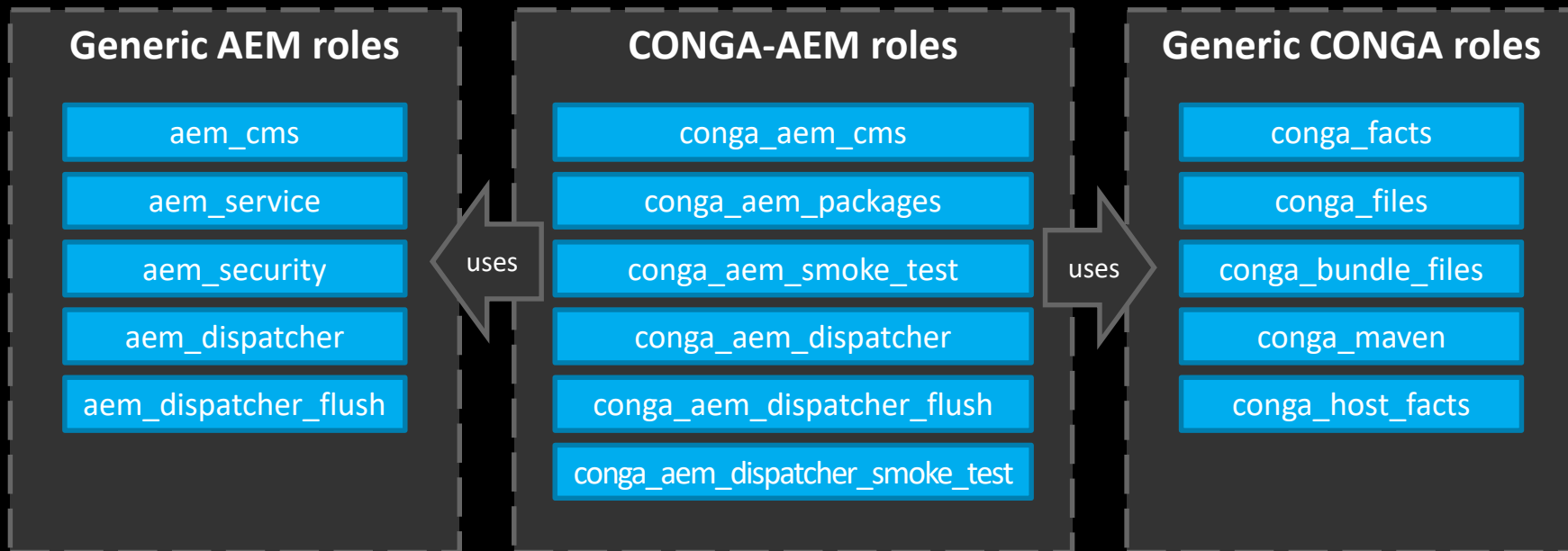




# Ansible roles and playbooks

- The generated project contains only Playbooks, no roles
- All roles are re-usable building blocks published on Ansible Galaxy
- Archetype is only a starting point

# Ansible roles from wcm.io DevOps



[https://galaxy.ansible.com/wcm\\_io\\_devops](https://galaxy.ansible.com/wcm_io_devops)

# Adobe AEM binaries

- AEM binaries not available in any public repo (only “Uber JAR”)
- We upload them ourselves in our Company Maven Repository following the [wcm.io Artifact Naming Conventions for AEM Binaries](#)
- Everything available via Maven: Quickstart JAR, SPs, CFPs, HFs, Dispatcher binaries

## 1. Ansible Playbooks



# About security

# Sensitive data in Config Management

- AEM admin password
- AEM license.properties file
- AEM crypto key \*
- Maven repository credentials

Encrypted in repository using  
**Ansible Vault**

Encrypted in generated artifacts using  
**AEM crypto support,**  
**Password hashes**

\* New generated for each Config Management project

- Automation implements a good part of the AEM security checklist (more planned)
- IP restrictions for
  - accessing the machines
  - accessing Dispatcher Flush URLs
  - accessing AEM Admin URLs like /console



1. Deployment finished
2. Author content
3. Publish it



# References

# Related adaptTo() talks

- adaptTo() 2017: [Automate AEM Deployment with Ansible and wcm.io CONGA](#)
- adaptTo() 2015: [CONGA - Configuration generation for Sling and AEM](#)
- adaptTo() 2017: [Lightning Talk: Use CONGA to provision your local AEM instance](#)



- wcm.io Maven Tooling  
<http://wcm.io/tooling/maven/>
- wcm.io DevOps CONGA  
<http://devops.wcm.io/conga/>
- wcm.io DevOps CONGA Training Materials  
<http://training.wcm.io/conga/>
- wcm.io DevOps Ansible Automation for AEM  
<http://devops.wcm.io/ansible-aem/>
- Need Help? Use the wcm.io Mailing Lists  
<http://wcm.io/ mailing-lists.html>
- Demo Script for this talk  
<https://github.com/adaptto/2018-maven-archetypes-for-aem>