adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE

28th – 30th SEPTEMBER 2020

Use Cloud Manager to deploy CONGA-based AEM Applications

Stefan Seifert, pro!vision GmbH

About the Speaker

- AEM Developer
- Apache Sling PMC
- Apache Member

Stefan Seifert

PRO!VISION
SOFTWARE CRAFTSMANSHIP
a diva-e company

https://www.pro-vision.de

diva-e

https://diva-e.com/

adaptTo()

Starting point

- AEM projects with a more complex/automated build setup
- We are using wcm.io CONGA to
  - Separate build artifacts and configuration
  - Define artifact order and conditions
  - Describe environments

CONGA abstracts AEM configuration

Covered by adaptTo() Talks 2015, 2017, 2018

***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*

- AEM project env. config
- Domain names
- IP Addresses
- Number of instances
- Machine sizing
- Credentials

git Repository 2

Maintained by: **Operations**

Challenges

working with AEM Cloud Manager

Photo by Jukan Tateisi on Unsplash

adaptTo()

adaptTo()

Artifacts to deploy

- There is no “deployment descriptor”
- AEM Cloud manager uses “heuristics” to detect the artifacts to deploy
- Looks only in Maven “target” folders

Deploy artifact heuristics

Type	How to detect	Remarks
Content Packages	Filename: `*.zip` Contains "`META-INF/vault/filter.xml`" and "`META-INF/vault/properties.xml`"	Properties are checked for a `cloudManagerTarget` property – **if set to "`none`" package is ignored**
Dispatcher configuration	Filename: `*.zip` Contains: "`conf`", "`conf.d`", "`conf.dispatcher.d`", "`conf.modules.d`" folders	There can only be **one** single dispatcher configuration per repository
Functional tests (Integration tests)	Filename: `*.jar` Contains Manifest entry: `Cloud-Manager-TestType=integration-test`	Tests executed automatically when found
UI Tests (WebdriverIO)	Filename: `*ui-test-docker-context.tar.gz` Contains „`Dockerfile`"	Tests executed automatically when found

Content Packages

- Only the “**all**” package should be deployed
  - All other content packages must be marked with `cloudManagerTarget=none`
- “all” package contains **embedded** packages
  - Clear separation between **mutable** and **immutable** content packages

adaptTo()

“all” content package

Folder names (convention)	Description	Example
/apps/<app>/application/install	**Immutable** application packages	ui.apps packages `/apps`
/apps/<app>/container/install	**Immutable** container packages	OSGi configuration, repoinit Statements `/apps`
/apps/<app>/content/install	**Mutable** content packages	Content, Assets, Editable Templates, Context-Aware Configuration `/content`, `/conf`, `/etc`
/apps/<app>/<packageType>/install.<runMode(s)>	Runmode suffixes define on which instances/environments the packages should be installed	Instance runmodes: **author**, **publish** Env runmodes: **dev**, **stage**, **prod**

adaptTo()

Not all runmode combinations supported!

Package type	**Immutable** content packages	**Mutable** content packages
Instance type: `author` / `publish`	✓	✓
Environment: `dev` / `stage` / `prod`	✓	✗*
Other run modes	✗*	✗*

* Works locally & on-premise but not with Cloud Manager & AEM Cloud Service

adaptTo()

Package install order

- Can be controlled via **package dependencies** (defined in package properties/POM)
- Problematic with:
  - 3rdparty packages
  - Conditional packages
  - Runmode-specific packages

Single dispatcher configuration

- Single dispatcher config for dev/stage/prod
- **Variables/Conditionals** can be used:

```
# Simple usage of the environment variable
ServerName ${ENVIRONMENT_TYPE}.company.com

# When more logic is required
<IfDefine ENVIRONMENT_STAGE>
  Define VIRTUALHOST stage.example.com
</IfDefine>
<IfDefine ENVIRONMENT_PROD>
  Define VIRTUALHOST prod.example.com
</IfDefine>
```

Steps to

make your AEM project ready for the cloud

Photo by Barry Rijkse on Unsplash

adaptTo()

The obvious steps

- Use latest AEM SDK API
- Remove obsolete Sling/3rdparty bundles
- Do not deploy AEM SPs/CFPs
- Do not deploy Core Component packages
- Use split content packages from ACS AEM Commons *
- Update dependencies

* Use ACS AEM Commons 5.0.0 or higher

adaptTo()

Pimp your content packages

- Update to latest Content Package plugin
  - Use `org.apache.jackrabbit:filevault-package-maven-plugin`
- Separate mutable from immutable
- Set `packageType`
- Take validation errors seriously and fix them
- Set `cloudManagerTarget=none`

adaptTo()

Make use of repoinit

- Example use cases
  - Create service users, set ACLs
  - Create resources, set properties
- Define via OSGi factory configs
- Limitations: Not possible to remove/replace, not suited for big chunks of content

https://sling.apache.org/documentation/bundles/repository-initialization.html

adaptTo()

repoinit Example

```
# Create assets root folder
create path /content/dam/app1(sling:Folder)

# Create service user for accessing assets
create service user app1SystemUser
set ACL on /content/dam/app1
  allow jcr:read,rep:write for app1SystemUser
end
```

HTML Client library URL fingerprinting

- AEM Cloud service automatically enables “URL fingerprinting” for clientlibs (hash)

Example:

`/etc.clientlibs/app1/clientlibs/lib1.lc-a06a062c96afe74fa1b48764d113fdcf-lc.min.css`

- Remove:
  - Manual/Automatic set of `longCacheKey` param
  - ACS AEM Commons Versioned Clientlibs

wcm.io CONGA Support

for AEM **Cloud** Manager

adaptTo()

CONGA config for AEM Cloud projects

Switch to single git repo for AEM Cloud service

***Config Definition** in AEM project*

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

Templates with placeholders

git Repository

***Environments** in Config Mgmt*

- AEM project env. config
- Domain names
- ~~IP Addresses~~
- ~~Number of instances~~
- ~~Machine sizing~~
- Credential *references*

Maintained by: **Developers**

CONGA-generated AEM packages

- Set package types
- Apply FileVault validation
- Use repoinit for system users & ACLs

adaptTo()

“all” content package

- Generates “all” package from CONGA definitions
  - Distinguishes author and publish run modes
  - Automatically generates package dependencies - based on file order in CONGA definitions

Dispatcher configuration

- CONGA AEM Definitions provides new role `aem-dispatcher-cloud`
- Supports single dispatcher config with conditionals
- And much more like Short URL Mapping

https://devops.wcm.io/conga/definitions/aem/

Demo

AEM Cloud project
with CONGA Definitions

https://github.com/adaptto/2020-aem-sample-project-conga-cloud

Give it a try

wcm.io Maven Archetype for AEM
https://wcm.io/tooling/maven/archetypes/aem/

- CONGA support
- AEM Cloud support
- wcm.io Handler (optional)

Other Pitfalls

and Solutions

Photo by Bart Christiaanse on Unsplash

adaptTo()

No crypto key support

- New approach to provide **secrets** in OSGi system configuration

Set environment variables via Adobe I/O:

```
$ aio cloudmanager:set-environment-variables 12345 --variable PARAM_1 "value1"
$ aio cloudmanager:set-environment-variables 12345 --secret PARAM_2 "secret2"
```

Reference environment variables in OSGi config:

```
com.app1.ServiceConfiguration
  param1="$[env:PARAM_1]"
  secretParam2="$[env:PARAM_2]"
```

Make sure to use UPPERCASE_UNDERLINE notation for variable names | Documentation

Do not overwrite default log level

- This will break the Cloud Manager build:

```
org.apache.sling.commons.log.LogManager
  org.apache.sling.commons.log.file="logs/error.log"
  org.apache.sling.commons.log.level="warn"
  org.apache.sling.commons.log.file.size="'.'yyyy-MM-dd"
  org.apache.sling.commons.log.file.number=I"7"
  org.apache.sling.commons.log.numOfLines=I"10000"
```

Bundles with Sling-Initial-Content

- Works with AEM Cloud Service
- Make sure to enable “reproducible builds” feature in Maven project - to ensure consistent ordering of entries in bundle JAR file

```
<properties>
  <project.build.outputTimestamp>2020-09-28T18:25:00Z</project.build.outputTimestamp>
</properties>
```

https://maven.apache.org/guides/mini/guide-reproducible-builds.html
https://adapt.to/2017/en/schedule/ease-development-with-apache-sling-file-system-resource-provider.html

Outdated versions in build environment

- Currently you are limited to:
  - Maven 3.6.0
  - Java 8 *)

Update 30.09.20:
Java 11 support is working as described in the Adobe documentation (with `maven-toolchains-plugin`).
But: Not all Maven plugins support this way – especially the `maven-enforcer-plugin` does not support it, remove Java 11 constraint from it.

* AEM Documentation describes a way to use Java 11, I was not able to get it working

Define own Maven repositories

- By default, only Maven Central is used
- Define alternative repos in POM (enclosed in profile)
- Define credentials via own settings.xml (see documentation)
- Or use maven-ext-repos-from-env

References

- Example project: https://github.com/adaptto/2020-aem-sample-project-conga-cloud
- wcm.io CONGA
  - adaptTo() 2015: CONGA - Configuration generation for Sling and AEM
  - adaptTo() 2017: Automate AEM Deployment with Ansible and wcm.io CONGA
  - adaptTo() 2018: Maven Archetypes for AEM & Cloud Deployment
  - CONGA Training material with exercises
- wcm.io Maven Archetype for AEM
- How-To: Make your CONGA-based AEM project ready for AEM Cloud Service
- adaptTo() 2020: FileVault Validation by Konrad Windszus
- aem-cloud-service-source-migration: Tool to convert Dispatcher configs and Maven projects

Questions?

Photo by Stefan Wieland @ adaptTo() 2016