

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 28-30 SEPTEMBER 2015



CONGA - Configuration generation for Sling and AEM

Thomas Wolfart, pro!vision GmbH

About the Speaker

Generalizing specialist
CQ/AEM since 5.x
Loves stuff that works

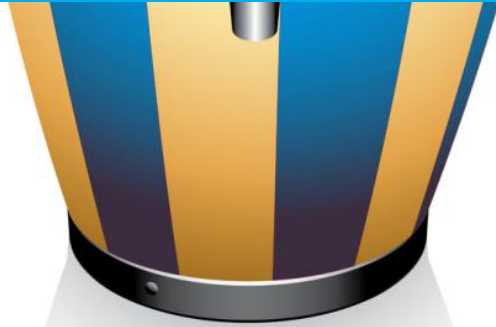


<http://www.pro-vision.de>





Introduction



What we want for our applications

- fancy server setup
- fancy application setup
- fancy pipeline automation
- fancy unicorn stuff (pick any)

What we have to work with

- 1 op department per skill
 - Ops are often separated from each other
- 0 departments for overall tasks
 - Devs are often responsible for the global perspective and have an important voice
- Reason for prod downtime shifts from software to config bugs for mature projects



Looking for a solution



The strategy we chose

- More control of system config for the devs
- Goals:
 - identify and fill gaps of knowledge
 - introduce automation
 - improve reproducibility
 - reduce impact of human mistakes

Tools like puppet, chef et al are great ..
.. but not available in the target environments.

Covering the AEM specifics with these is
challenging:

Dispatcher, OSGI and AEM Content Packaging

Introducing CONGA

This is where ***CON**figuration **GenerA**tor* comes to play:

- Generates all the configuration artifacts
- Doesn't care about server setup
- Doesn't care about deployment



CONGA



CONGA at a glance

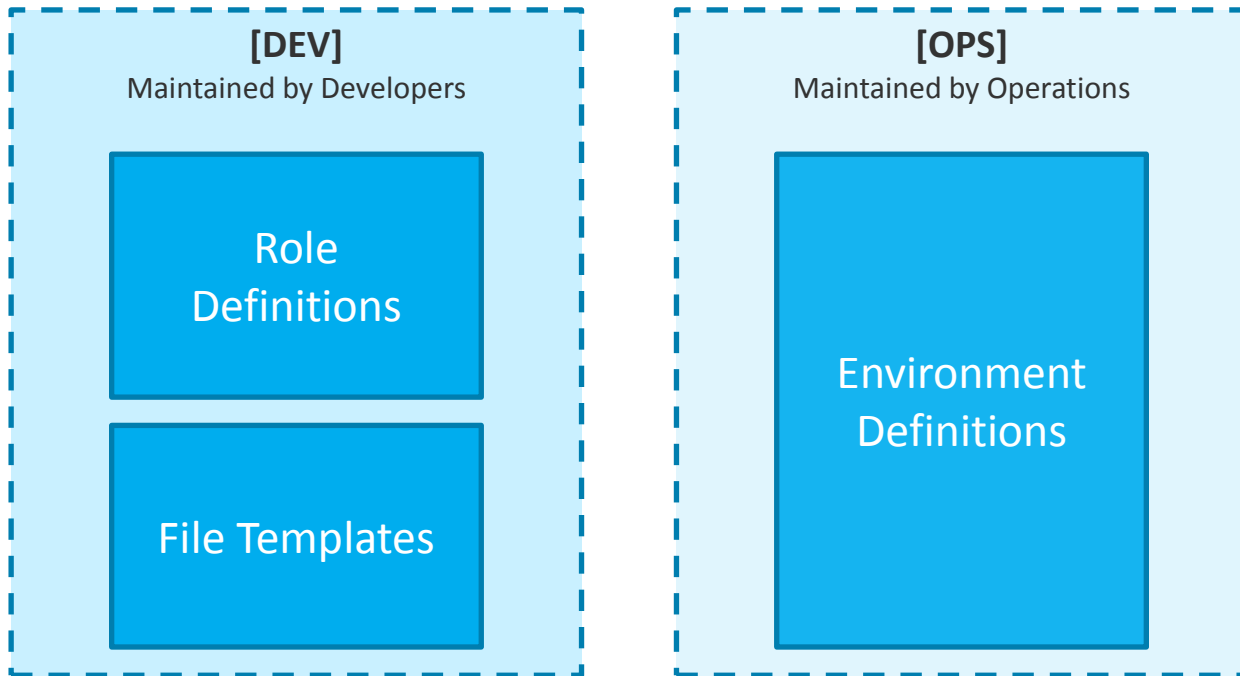
- CONGA is a modular and plugin based configuration generator
- Knowledge of new config formats can easily be added
- Generic formats like JSON and XML are supported, plugins for Sling and AEM exist

- Java
- Maven Plugin (standalone CLI available)
- Handlebars templating
- YAML for environment description

CONGA's basic principle

- Devs prepare template definitions with placeholders
- Ops provide environment property files
- CONGA combines them

CONGA's basic principle



Templates (excerpt 1)

```
...  
"transportUri": "{{this.url}}/bin/receive?slings:authRequestLogin=1",  
"transportUser": "{{this.transportUser}}",  
"transportPassword": "{{this.transportPassword}}",  
...
```

Templates (excerpt 2)

```

{{#eachIf tenants "config.slingMappingPath"}},
  "{{this.tenant}}": {
    "jcr:primaryType": "sling:Mapping",
    "sling:match": "(
      {{~ regexQuote this.tenant ~}}
      {{~#if this.config.serverNameSsl}}|{{regexQuote this.config.serverNameSsl}}|{{/if ~}}
      {{~#each this.config.serverAliasNames}}|{{regexQuote this}}|{{/each ~}}
    )",
    "sling:internalRedirect": ["{{this.config.slingMappingPath}}", "/"]
  }
{{/eachIf}}
```


Environment Definition (excerpt)

- node: aem-author

roles:

- role: aem-cms

variant: aem-author

config:

replication.author.publishTargets:

- name: publish1

url: http://aem-publish1:4503

transportUser: admin

transportPassword:

"{443a90a7ed2bf085c..7a4a5947013b32}"

...

...

tenants:

- tenant: sample1.com

config:

serverAliasNames:

- www.sample1.com

slingMappingPath: /content/sample1

dispatcher:

rootRedirectUrl: /en.html

...

CONGA provides

- Conf. templates in the code/delivery artifact
- Generated config matching the application
- Consistent config across all stages
- Full setup of AEM environment

Available formats/plugin types

File formats:

JSON, XML, .conf, .sh, .bat/.cmd,

OSGi .config, Sling Provisioning,

AEM Content Packages, Dispatcher .any

Plugin types:

Header, validator, escaping, post-processor, multiply

Available templates: AEM definitions

- Apache httpd vhosts for author/publish
- Dispatcher.any
- AEM replication agents
- Sling mapping
- AEM configs
- Actually all needed for an AEM environment



Example



Example AEM project,
containing templates & definitions:

<https://github.com/wcm-io/wcm-io-samples>

Example CONGA project:

<https://github.com/adapto-conf/2015-conga>

A simple environment for AEM

- 1x Author
- 3x Publisher
- 3x Webserver & Dispatcher
- 3 Tenants, using
 - URL Shortening
 - HTTP/HTTPS

- Author vhosts
- Publish vhosts
- dispatcher.any
- AEM App Package
- AEM bin/start
- AEM Workflows
- AEM Sling Mappings
- AEM Replication Agents
- AEM Configuration Packages



Looking forward



Possible next steps for CONGA:

- Inheritance of roles
- Generation of infrastructure documentation
- Integration with existing „fancy“ tools
- Your ideas?

Try it!

Feel free to give CONGA a try,
and participate:

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