

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 25-27 SEPTEMBER 2017

Automate AEM Deployment with Ansible and wcm.io CONGA

Martin Wehner, pro!vision GmbH

About the Speaker

- R&D engineer at pro!vision
- AEM since CQ 3.x



Martin Wehner



PRO!VISION
SOFTWARE CRAFTSMANSHIP

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

The Challenges

Challenges

- Systems take too long to setup
- Teams struggle maintaining their systems
- Ops resources are scarce
- “Hardcore” ops don’t know AEM
- Nobody knows why a system is configured the way it is

Goals

- **Make it easy:**
 - For teams to maintain internal & customer systems
 - To recreate systems quickly and reliably
- **Just update the CONGA config and have it deployed automatically**
- **You shouldn't need to touch the automation at all**
- **Plug the deployment hole in CONGA**



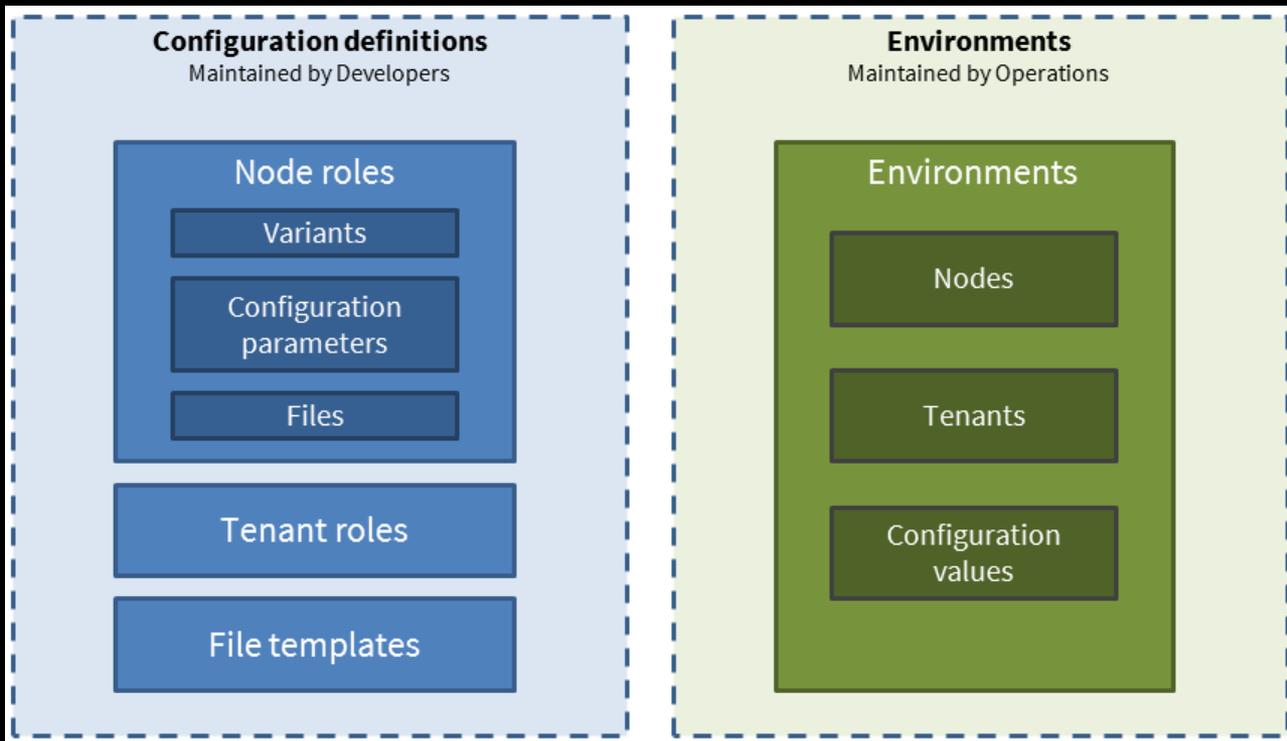
CONGA

What is CONGA?

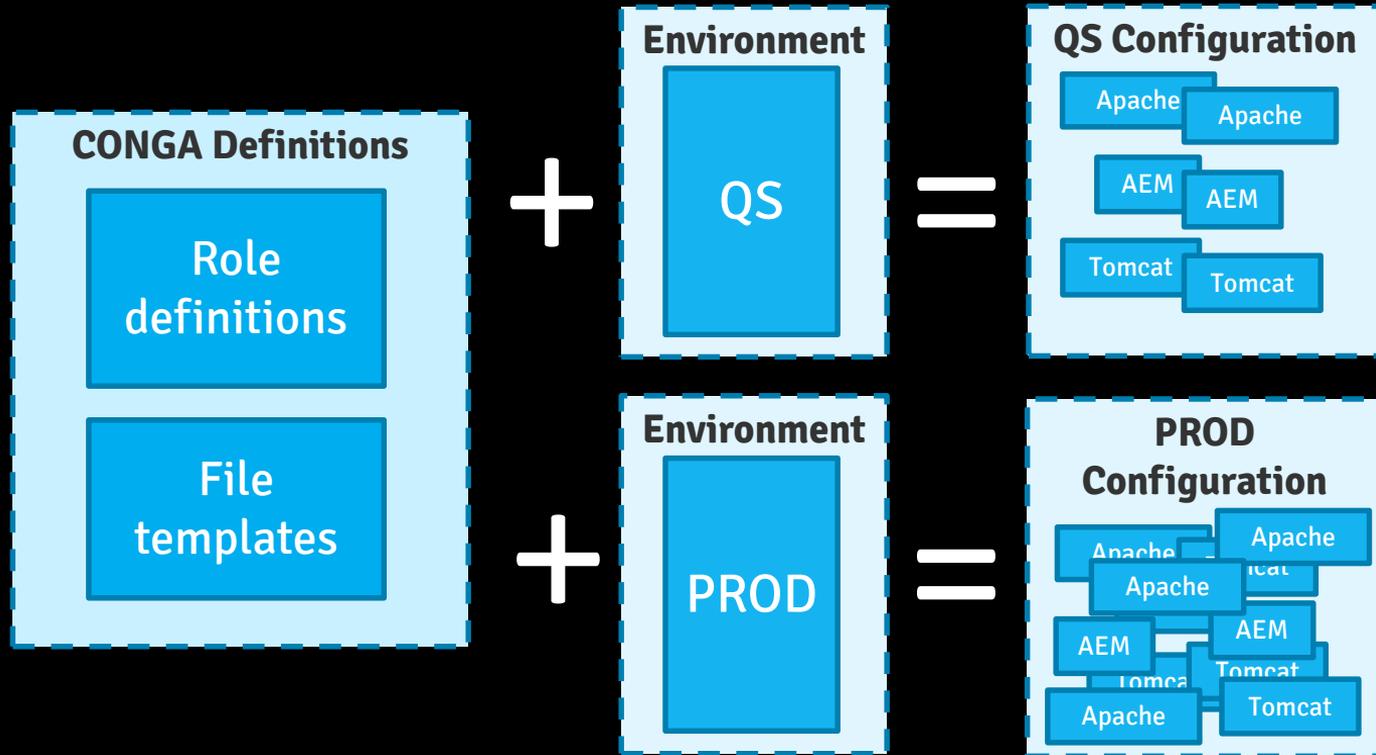


- Configuration generator
- Many AEM/OSGi specific features
- Separation of configuration & environment
- Maven integration
- Extensible with plugins
- Comes with generic set of AEM roles

CONGA Meta Model



Simplified CONGA Model





- Plain text files e.g. properties, scripts, dispatcher & HTTPd config
- JSON files
- XML files



- OSGi configuration snippets
- Sling provisioning model
- AEM content package, e.g.
 - OSGi configurations
 - Translator languages nodes
 - Community SRP config
 - Sling mappings
 - System users
 - etc.



- file: wcm-io-samples-aem-cms-config.provisioning
- dir: packages
- template: wcm-io-samples-aem-cms-config.provisioning.hbs
- postProcessors:
 - aem-contentpackage-osgiconfig
- postProcessorOptions:
 - contentPackage:
 - name: wcm-io-samples-aem-cms-config
 - rootPath: /apps/wcm-io-samples-aem-cms-config/config
 - filters:
 - filter: /apps/wcm-io-samples-aem-cms-config



nodes:

- node: aem-author

roles:

- role: aem-cms

variants:

- aem-author

config:

jvm.heapspace.max: 2048m

quickstart.rootPath: /opt/adobe/aem

quickstart.port: 4502



“CONGA is not a deployment automation tool”

“CONGA only generates files. Further distribution of these files (e.g. copy to server, deploy via HTTP etc.) is not part of the tool.”



Ansible

- “Automation for Everyone”
- More approachable than Puppet & Chef
- Python
- Agentless
- Pushes tasks via SSH
- Huge ecosystem of modules & roles

```
---
```

```
- hosts: webservers
```

```
  roles:
```

```
    - common
```

```
    - aem
```

```
- tasks:
```

```
  - copy:
```

```
    src: start
```

```
    dest: "{{ aem_home }}/crx-quickstart/bin"
```

So to sum it up..

- CONGA is great at generating AEM artifacts but doesn't do anything else
- Ansible is great at deploying stuff but doesn't know about AEM



+



Integration

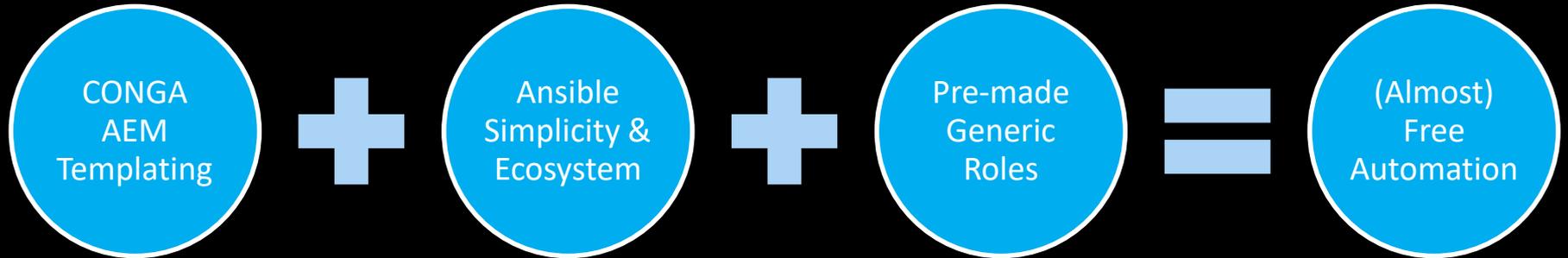
The Approach

- No duplication of configuration between CONGA and Ansible
- Make the CONGA configuration model available to Ansible (and other automation tools)
- CONGA generates model YAML
- For each CONGA role, create a matching Ansible roles that “knows” how to deploy it

The Approach

- Role should handle all possible configurations
- The Ansible role has to ensures that:
 - The service to be configured is installed
 - All implicit dependencies are available
- It should be possible to use 3rd party roles

Best of Both Worlds



Implementation

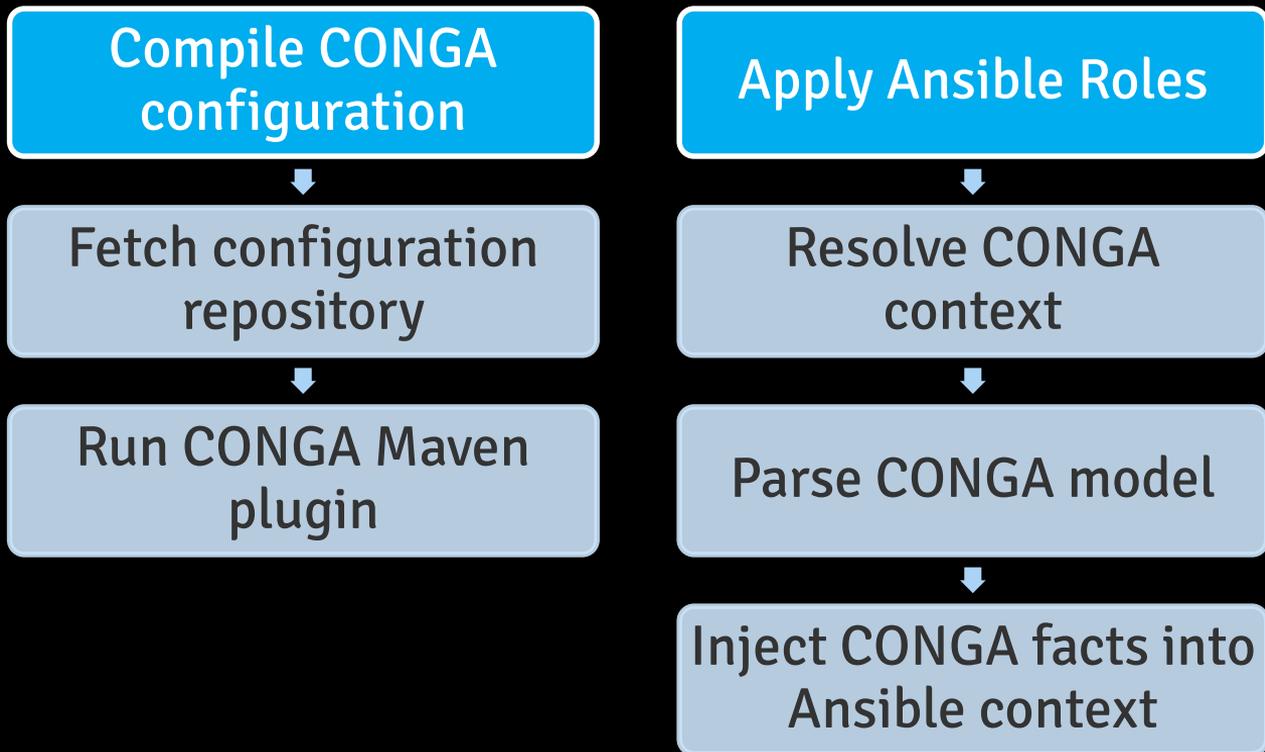
CONGA Facts in Ansible

- name: Create cache root directory.
file:
 - path: "{{ conga_config.dispatcher.cacheRootPath }}"
 - state: directory
- name: Enable Apache modules required by CONGA role.
apache2_module:
 - name: "{{ item }}"
 - state: presentwith_items: "{{ conga_config.httpd.modules }}"
- name: Setup SSL if required by CONGA variant.
include: setup_ssl.yml
when: conga_variant == "ssl" or conga_variant == "no-ssl+ssl"

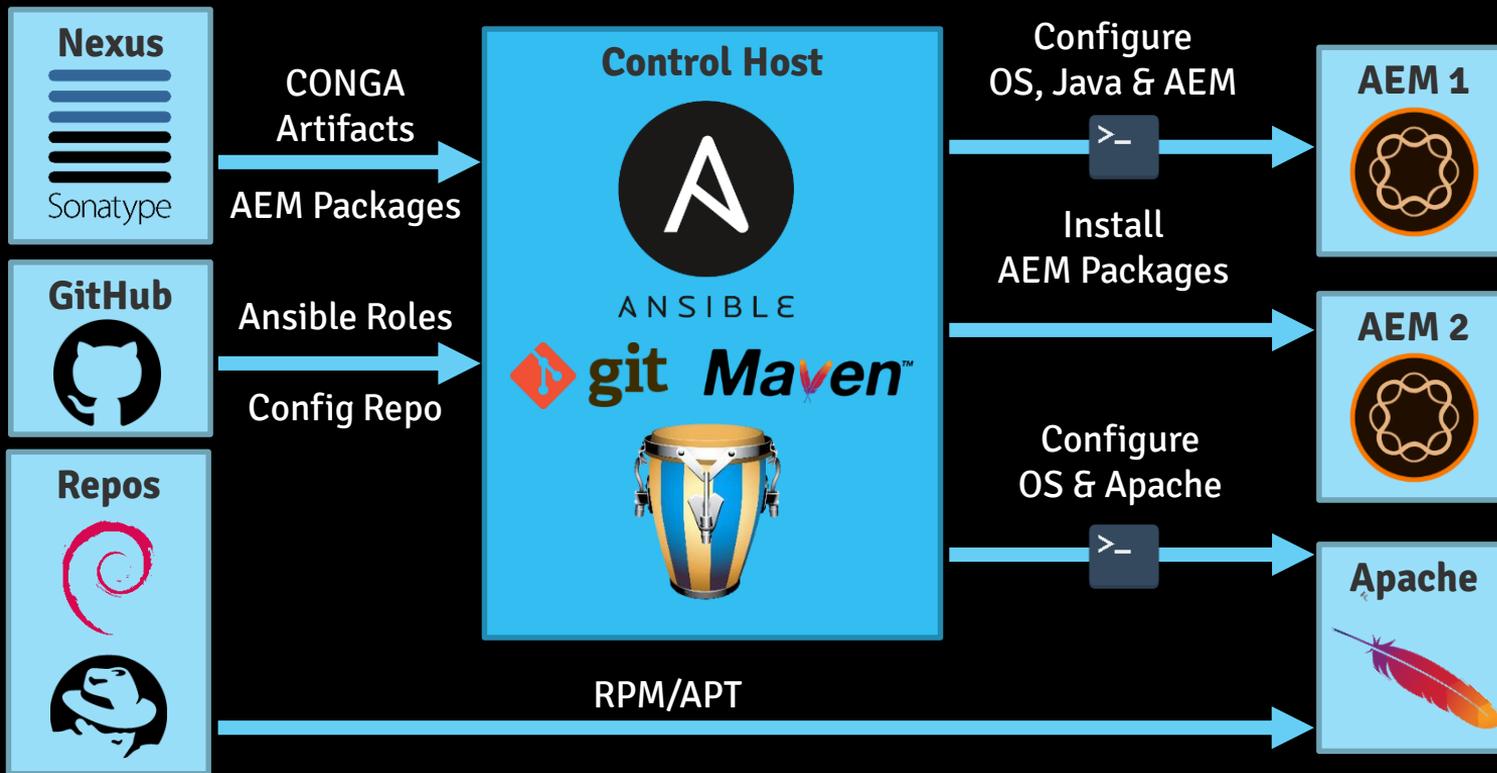
CONGA Facts in Ansible

- name: Create cache root directory.
file:
 - path: "{{ conga_config.dispatcher.cacheRootPath }}"
 - state: directory
- name: Enable Apache modules required by CONGA role.
apache2_module:
 - name: "{{ item }}"
 - state: presentwith_items: "{{ conga_config.httpd.modules }}"
- name: Setup SSL if required by CONGA variant.
include: setup_ssl.yml
when: conga_variant == "ssl" or conga_variant == "no-ssl+ssl"

Basic Steps



Hosts Interaction



Basic Integration Playbook

```
- hosts: localhost
  roles:
    - { role: conga-maven,
        conga_maven_git_repo: "https://github.com/wcm-io/wcm-io-samples.git" }

- hosts: aem

pre_tasks:
  - conga_facts:
      conga_role_mapping: aem-cms

roles:
  - { role: aem-cms, aem_home: "{{ conga_config.quickstart.rootPath }}" }
  - conga-aem-cms
```

Basic Integration Playbook

```
- hosts: localhost
  roles:
    - { role: conga-maven,
        conga_maven_git_repo: "https://github.com/wcm-io/wcm-io-samples.git" }

- hosts: aem

pre_tasks:
  - conga_facts:
      conga_role_mapping: aem-cms

roles:
  - { role: aem-cms, aem_home: "{{ conga_config.quickstart.rootPath }}" }
  - conga-aem-cms
  - conga-your-project
```

Main Ansible Roles

- Centerpiece of the integration
- Resolves the current CONGA context
- Parses CONGA configuration model
- Supplies the proper configuration values as Ansible facts
- Both an action plugin and a role

- Gets configuration from Git
- Runs Maven plugin to generate configuration
- Not required, you can also use a pre-compiled configuration

- Deploys generated configuration files to target host
- Meant to be included or used as dependency of higher-level roles
- Options to filter the list of files and mangle the paths

- Deploys AEM packages to target host
- Uses wcm.io Content Package Maven Plugin internally
- Restarts AEM as required

- Installs AEM from:
 - Local JAR
 - Nexus
 - Distribution Package
- Ensures Java is installed
- Creates user, group and AEM service as required
- Decoupled from CONGA

- Installs dispatcher module from:
 - Local Tarball
 - Distribution Package
- Ensures Apache is installed
- Configures SSL dependencies and SELinux as required

Role	Purpose
aem-service	Controls AEM startup and shutdown
conga-aem-cms	Deploys CONGA role <i>aem-cms</i>
conga-aem-dispatcher-(common)	Deploys CONGA roles <i>aem-dispatcher-*</i>

+ various internal application and support roles

Implementation Challenges

Integration Challenges

- The only unit of reusability in Ansible is the role
- Ansibles internal execution model is pretty static
- The current execution context is difficult to obtain
- Not always a clear-cut separation of concerns between CONGA and Ansible

AEM Challenges

- AEM has too many knobs to fiddle to expose them all in monolithic automation unit
- Requires an interdependent set of roles, each controlling an aspect of the system
- Startup & Cluster Orchestration

Current State

- Published on GitHub
- Controls deployment of adaptTo() and pro!vision corporate websites
- Integrated in CI
- Used in multiple customer projects

- Works on:
 - CentOS/RHEL
 - Ubuntu
 - Debian
- Doesn't do Windows
- Additional distributions added as required

Next Steps

- Tighter inventory integration
- Crypto/Ansible Vault support for CONGA
- Generalize AEM/Dispatcher roles
- Refactor CONGA roles
- Publish on Ansible Galaxy

<https://www.ansible.com/>

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

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

<https://github.com/adaptto/2017-automate-aem-deployment-ansible-conga>

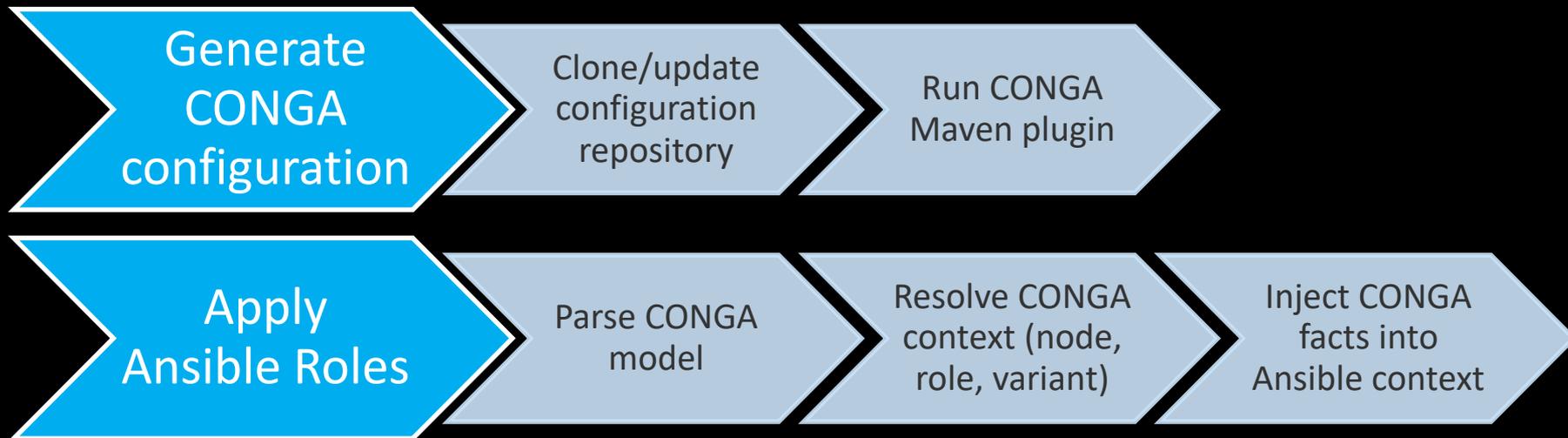
Thank You



Advantages

- Describe your setup completely in CONGA
- Reuse the pre-made generic Ansible roles
- Combines:
 - CONGA AEM templating power
 - Ansible simplicity & ecosystem

Basic Flow



Demo