Configuring containers in a better way

Containers Configuration Tool
About me

- Open Source enthusiast
- Containers believer
- Prague Containers Meetup co-founder
- Emacs user
- Father

you can follow me on:
  - @_l_d_j_
  - github.com/l-d-j
  - www.meetup.com/Prague-Containers-Meetup/
How do you create container

Dockerfile  →  Build  →  Docker Image
Containers are complex
Configuration

Two types:

- **build time**
  - during Dockerfile processing
    - configure special logging, jolokia, dnsping
    - Remove unneeded components

- **runtime**
  - when container starts
    - setup datasource
    - deploy application
    - setup ssl
    - Inject secrets
    - Create users...
Configuration - Real life

- Application
- Database
- Messaging

All of these need to communicate / share secret data

Orchestrator -> Kubernetes(OpenShift), swarm?

You still need support in image
CCT

- [https://github.com/containers-tools/cct](https://github.com/containers-tools/cct)
- A Python approach to configure containers
  - Build and runtime configuration
  - Best as Entrypoint
- Enables you to share/extend scripts
- Easy to write custom module
- Enables you to extend/configure container with custom code
  - Can fetch a custom module from URL
CCT Module

- python class which extends cct.module.Module
- support for fetching artifacts (self.artifacts[name])

Sources.yaml:

- name: jboss-eap-7.0.0.GA.zip
  chksum: cd02482daa0398bf5500e1628d28179a
- name: jbeap4410.zip
  chksum: 6be8ded1642c644bb1b935a30284bd4a
Dogen

- [https://github.com/jboss-dockerfiles/dogen](https://github.com/jboss-dockerfiles/dogen)
- docker pull jboss/dogen
  - yes it’s containerized
- templating solution for Dockerfiles
- tries to fix:
  - layering
  - inheritance/code reuse

```dgen
# This is a Dockerfile for the {{ name }}:
FROM {{ helper.base_image(from, version) }}
{% if maintainer %
MAINTAINER {{ maintainer }}
{% endif -%}

# Environment variables
ENV JBOSS_IMAGE_NAME="{{name}}" \
   JBOSS_IMAGE_VERSION="{{version}}" \
   JBOSS_IMAGE_RELEASE="{{release}}" {% if he
   {{ env.name }}="{{ env.value }}" {% if loo
   {% endif %}
{% endif %}

# Labels
LABEL Name="$JBOSS_IMAGE_NAME" \
   Version="$JBOSS_IMAGE_VERSION" \
   Release="$JBOSS_IMAGE_RELEASE" \
   Architecture="x86_64" \
   BZComponent="{{ helper.component(name) }}
   {% for label in labels %}
   {{ label.name }}="{{ label.value }}" {%-
   {% endfor %}
{% endif %}
```
Thank you,

any Questions?