How to Generate Alfresco Docker Assets with Docker Init

Alfresco Tech Talk Live #151

Angel Borroy
Developer Evangelist

October 18, 2023
Agenda

- DockerCon 2023
- Generating Docker Assets for Alfresco
- Docker Init
- Docker Init With Templates!
DockerCon 2023
DockerCon 2023

Announcements

• docker scout
  • GA availability, vulnerability analysis, SBOM

• docker build
  • Next-Gen, Cloud aided, faster

• docker desktop
  • Preview Program, hybrid development

• docker debug
  • Experimental, local and remote containers
DockerCon 2023

Docker AI/ML

- Pre-configured LLMs
  - Llama2
  - GPT-3.5
  - GPT-4
- Ollama management
- Neo4j as database
- LangChain orchestration
Cloud Approach

- Docker is moving all the features to be executed in the Cloud instead of in the local computer
- Current cloud is SaaS model and AWS hosted
  - Agnostic Cloud Platform and Private Cloud deployment to come
- Existing features but also new will be also available to be executed from local computer
  - However, if you move to Cloud, you get some benefits
Generating Docker Assets
Alfresco Deployment
Alfresco Tools for Docker Deployment

**Python**

`python-generator`

Generate Docker Compose templates via command line inputs, tailored for testing within software pipelines.

**Yeoman**

`alfresco-docker-installer`

`alfresco-enterprise-docker`

Generate Docker Compose templates via command line inputs or interactive prompts, designed for local development and testing.

**Docker**

`alfresco-installer`

Yeoman program *wrapper* for simplified execution and minimizing software requirements.

PIPELINEABLE

USER INTERACTION

FRICIONLESS
Docker Init
Docker Init

Available from [Docker Desktop 4.18](https://www.docker.com/products/docker-desktop)

Beta

The Docker Init plugin is currently in Beta. Docker recommends that you do not use this in production environments.

This utility will walk you through creating the following files with defaults for your project:
- .dockerignore
- Dockerfile
- compose.yaml

What application platform does your project use? [Use arrows to move, type to filter]
- Go - suitable for a Go server application
- Python - suitable for a Python server application
- Node - suitable for a Node server application
- Rust - suitable for a Rust server application
- ASP.NET - suitable for an ASP.NET application

Other - general purpose starting point for containerizing your application

Don’t see something you need? [Let us know](https://www.docker.com/feedback)
Docker Init

Official documentation

Closed source code

Docker Roadmap

PIPELINEABLE
USER INTERACTION
FRICTIONLESS
Demo Time
Docker Init With Templates!
Docker Init (with Templates!)

docker-init-with-templates

Sample **Go** implementation providing:

- CLI Interaction with **Cobra**
- Prompt Definition with **YAML**
  - Expression Evaluation with **goval**
- Assets Templating with **Go text/template**
Docker Init (with Templates!)

CLI Interaction

Init Command

Usage:

docker init [flags]
docker init [command]

Flags:

- 
  -t, --template string Name of the template to be used
- 
  -o, --output string Local Directory to write produced files
- 
  -p, --prompt stringArray Property=Value list containing prompt values
- 
  -d, --directory string Local Directory containing templates to be used
Docker Init (with Templates!)

CLI Interaction

Catalog Command

Usage:

docker init catalog [flags]

Flags:

- \( t \), \(--template\ \) string  Name of the template to get details
- \( d \), \(--directory\ \) string  Local Directory containing templates to be used
Docker Init (with Templates!)

**Prompt Definition**

```
<id>
  label: <label>
  [options | default]: ...
  condition: <value>
  multiple: true | false
  password: true | false
```

Prompts.yaml

**SELECT**

Volumes:
  - label: Volume method?
    options:
      - None
      - Native
      - Bind

**MULTIPLE CHOICE**

Addons:
  - label: Addons to install
  multiple: true
  options:
    - OCR
    - Order of the Bee

**VALUE**

ServerName:
  - label: Name of server?
  default: localhost

**CONDITIONAL**

MessagingUser:
  - label: Use messages?
  condition: Messaging=="Yes"
  default: admin

DbPassword:
  - label: Type your password
  password: true
Docker Init (with Templates!)

Assets Templating

- Use .tpl extension
- Use expressions with values from prompts following `text/template` syntax
- If a template is parsed to an empty file, the file is excluded from output folder

Conditional blocks

```yaml
{{- if eq .LegacyUI "Yes"}}
  - ./legacy-ui/compose.yaml
{{- end}}
```

Values from prompts

```yaml
environment:
  ACTIVEMQ_ADMIN_LOGIN: "{{.MessagingUser}}"
  ACTIVEMQ_ADMIN_PASSWORD: "{{.MessagingPassword}}"
```
Docker Init (with Templates!)

Structure

• Folder under “templates” directory with the name of the template (like “alfresco”)
• prompts.yaml file including prompts specification
• Assets to be processed using .tpl extension

```
templates
  └── nginx-golang-db
    ├── README.md.tpl
    ├── backend
    │   ├── Dockerfile.tpl
    │   ├── go.mod.tpl
    │   ├── go.sum.tpl
    │   └── main.go.tpl
    └── compose.yaml.tpl
  ├── db
  │   └── password.txt.tpl
  └── prompts.yaml
```

Template name >> nginx-golang-db
Usage >> README.md
Docker Image >> Dockerfile
Source code >> main.go
Docker Compose >> compose.yaml
Prompts definition
Docker Init (with Templates!)

Out-of-the-Box templates

docker-init-with-templates

Using the new Docker Compose `include` directive

- **Alfresco Community**, only 7.3 and 7.4 supported
- **Alfresco Enterprise**, only 7.3 and 7.4 supported
  - Requires credentials to access to `quay.io` Docker Images

Sample external folder template

- Deploy a [Go container](https://golang.org) using a postgres or mariadb as DB Engine
Demo Time
Thanks!