

Hyland™

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





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

Available today

docker scout

Coming soon

NEXT-GEN

docker build

docker desktop

PREVIEW PROGRAM

CODE NAME

docker debug

Advanced File Sync
Limited space

Debugging Innovation



Docker AI/ML

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

Find in Learning Center in Docker Desktop



github.com/docker/genai-stack



Cloud Approach

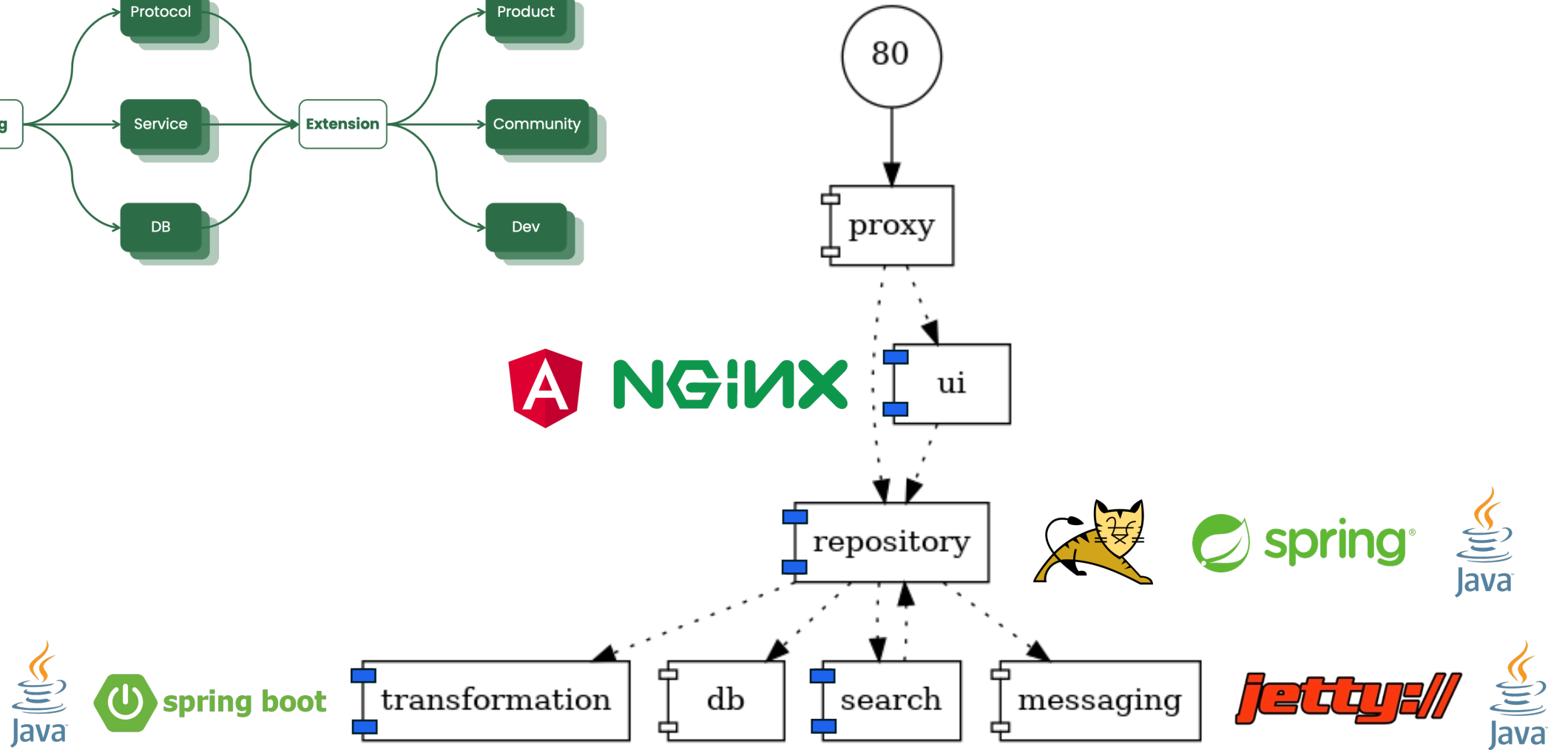
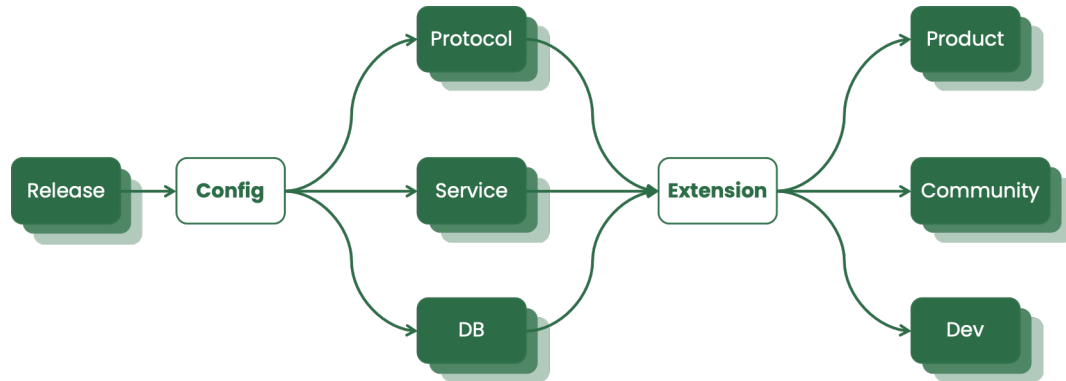
- Docker is moving all the features to be executed in the Cloud instead of in the local computer
- Current cloud is SaaS model and AWS hosted
 - Agnostic Cloud Platform and Private Cloud deployment to come
- Existing features but also new will be also available to be executed from local computer
 - However,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



Generating Docker Assets



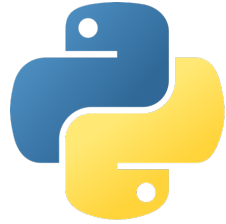
Alfresco Deployment



Alfresco Tools for Docker Deployment

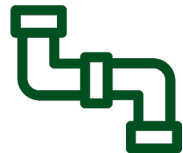


Python



[python-generator](#)

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



PIPELINEABLE

Yeoman



[alfresco-docker-installer](#)
[alfresco-enterprise-docker](#)

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



USER INTERACTION

Docker



[alfresco-installer](#)

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



FRICTIONLESS

Docker Init



Docker Init



Available from [Docker Desktop 4.18](#)



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!](#)



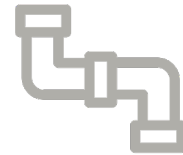
Docker Init



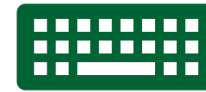
[Official documentation](#)

Closed source code

[Docker Roadmap](#)



PIPELINEABLE



USER INTERACTION



FRICTIONLESS

docker-roadmap

Updated 17 hours ago

Q init

Menu

35 Considering

2 results

! "what port..." as optional question in "docker init" command

#496 opened by UlisesGascon

docker init

Custom template support for "docker init" command

#483 opened by minwook-shin

docker init

24 Investigating

0 results

10 Upcoming

2 results

Docker Init - Java Support

#550 opened by nunocoracao

docker init

Docker Init - PHP Support

#549 opened by nunocoracao

docker init

11 Beta

1 result

Docker init - CLI command to create all docker related files in a project folder

#453 opened by pkhanchandan

cli community_new

docker_desktop





Demo Time

Docker Init With Templates!

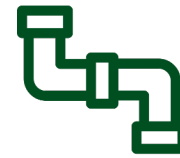


Docker Init (with Templates!)

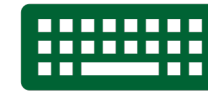
[docker-init-with-templates](#)

Sample **Go** implementation providing:

- CLI Interaction with [Cobra](#)
- Prompt Definition with [YAML](#)
 - Expression Evaluation with [goval](#)
- Assets Templating with [Go text/template](#)



PIPELINEABLE



USER INTERACTION



FRICTIONLESS

Docker Init (with Templates!)



CLI Interaction

Init Command

Usage:

```
docker init [flags]
```

```
docker init [command]
```

Flags:

- | | |
|---------------------------------------|---|
| <code>-t, --template string</code> | Name of the template to be used |
| <code>-o, --output string</code> | Local Directory to write produced files |
| <code>-p, --prompt stringArray</code> | Property=Value list containing prompt values |
| <code>-d, --directory string</code> | Local Directory containing templates to be used |

Run

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

[Discover](#)

Docker Init (with Templates!)



Prompt Definition

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



SELECT

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

VALUE

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

PASSWORD

```
DbPassword:  
  label: Type your password  
  password: true
```

MULTIPLE CHOICE

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

CONDITIONAL

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

prompts.yaml

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

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

Values from prompts

environment:

```
ACTIVEMQ_ADMIN_LOGIN: "{{.MessagingUser}}"  
ACTIVEMQ_ADMIN_PASSWORD: "{{.MessagingPassword}}"
```

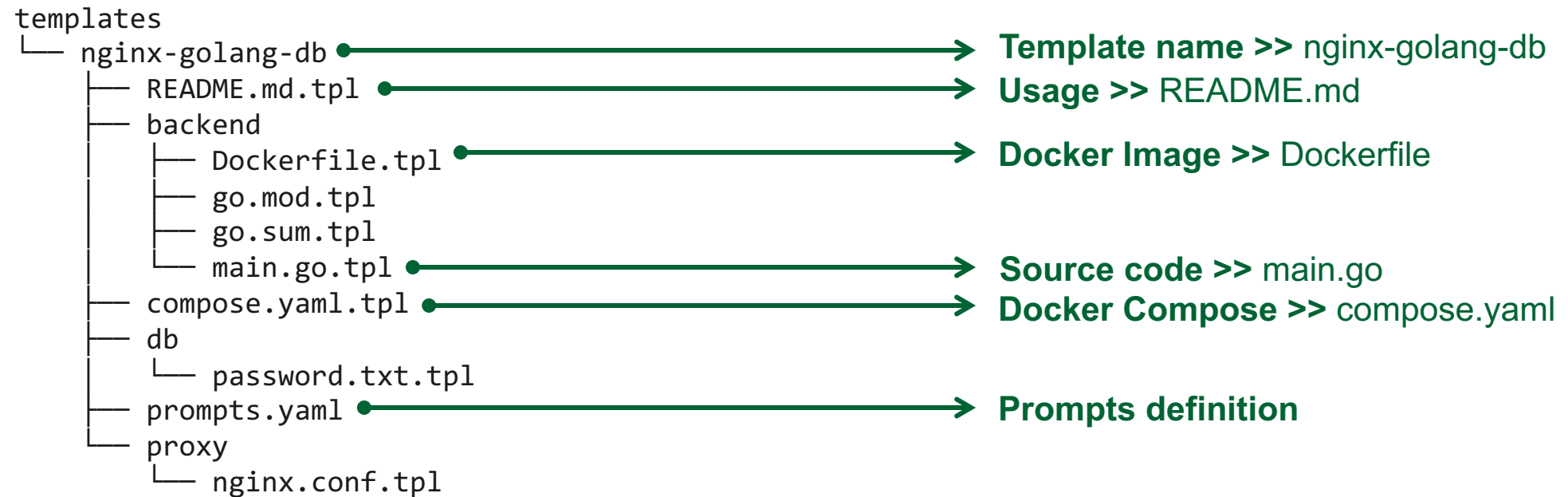
```
{{.}}  
{{.Var}}  
  
{{- .Var -}}  
  
{{ $planet := "Earth"}}  
{{ $planet }}  
  
{/* a comment */}  
  
Hi {{if .Name}} {{.Name}} {{else}} visitor  
{{end}}.  
  
{{if and .Arg1 .Arg2}} both complete. {{else}}  
incomplete. {{end}}  
  
{{with .Var}} {{end}}  
  
{{range .Array}} {{end}}  
  
{{ lt 3 4 }}
```

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



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](#) using a postgres or mariadb as DB Engine

0.0.5

Pre-release

 aborroy released this last week  0.0.5  175e8ef 

- Fix for MariaDB support
- Building support for Mac AMD64
- Support for testing
- Switch expression evaluator from go-bexpr to [goval](#)
- Initial template for `alfresco-enterprise`

▼ Assets 6

 [docker_darwin_amd64](#)

 [docker_darwin_arm64](#)

 [docker_linux_amd64](#)

 [docker_windows_amd64.exe](#)

 [Source code \(zip\)](#)

 [Source code \(tar.gz\)](#)



Demo Time

Hyland™

Thanks!