



LinuxFest Northwest 2016

The background features a stylized landscape with layered mountain ranges in shades of blue and teal. The sky is a pale yellow, accented with several white, fluffy clouds. The overall aesthetic is clean and modern.

Docker + Ansible

Michael Titus - LinuxFest Northwest 2016

Roadmap

Composing Dockerfiles

Dockerfile templating

Ansible

Ansible + Packer

Q&A



Dockerfiles: Troubled Waters

SPOT

Modularity?

Code Reuse?

DRY?

FROM reponame/image



Dockerfile and INCLUDE

May 2013: Pull Request 735, 94 comments

July 2015: No longer accepting patches to the Dockerfile syntax



"Allowing the Builder to be implemented as a separate utility consuming the Engine's API will open the door for many possibilities, such as offering alternate syntaxes or DSL for existing languages without cluttering the Engine's codebase."

Simple Templating: envsubst

`$VARIABLE` or `${VARIABLE}`

```
$ env var=api envsubst < a.dockerfile | docker build -
```

```
#!/bin/bash
```

```
export VAR1='somevalue' VAR2='someothervalue'
```

```
MYVARS='$VAR1:$VAR2'
```

```
envsubst "$MYVARS" < a.dockerfile | docker build -
```

Complex Templating: m4

```
FROM ubuntu:trusty
ENV DEBIAN_FRONTEND noninteractive
include(`ruby_2_1_2.m4')
include(`docker_latest.m4')
```

```
$ m4 ./Dockerfile.m4 > Dockerfile
```

```
$ m4 ./Dockerfile.m4 | docker build -
```

<http://bobbynorton.com/posts/includes-in-dockerfiles-with-m4-and-make/>

Complex Templating: m4 and make

```
lib = ./dockerfiles
```

```
dockerfile: $(lib)/*.m4
```

```
    m4 -I $(lib) $(lib)/Dockerfile.m4 > Dockerfile
```

```
build: dockerfile
```

```
    docker build --rm -t your/image .
```

```
$ make build
```


But Why?

Docker + Ansible

- Leverage existing playbooks
- Roles (modularity, code reuse)
- Multiple build targets
 - Docker
 - Vagrant
 - EC2
 - Bare-metal
- Container deployment
- Infrastructure-as-code



Ansible: Running container

```
$ docker build "base" Dockerfile (with SSH)
```

```
$ docker run etc.
```

```
$ ansible-playbook -i hosts docker.yml
```

```
  - name: Docker
```

```
    hosts: localhost
```

```
    tasks:
```

```
      - etc.
```

```
$ docker commit c3f279d17e0a myrepo/testimage:latest
```

Ansible Inside an Image









Ansible and Python deps (~198 MB)

Ansible playbooks, roles, var files, etc.

```
RUN ansible-playbook -i "localhost," local.yml
```

```
-           name:           Configure           myself
  hosts: all
  tasks: etc.
```

Docker Images for Ansible

 ansible/centos7-ansible public automated build	68 STARS	100K+ PULLS	 DETAILS
 williamyeh/ansible public automated build	43 STARS	10K+ PULLS	 DETAILS
 silintl/container-debug public automated build	0 STARS	10K+ PULLS	 DETAILS
 ansible/ubuntu14.04-ansible public automated build	79 STARS	10K+ PULLS	 DETAILS

Packer

Packer is a tool for creating machine and container images for multiple platforms from a single source configuration.



Packer: Templates

Builders - generate machine images

AWS EC2 AMI, Azure, Docker, VMWare, etc.

Provisioners - install and configure software for the images

Ansible, Chef, Puppet, Salt, Powershell, shell (bash), etc.

Post-Processors - store and/or completed images

amazon-import, docker-push, docker-tag, Vagrant, etc.

```
{
  "builders": [{
    "type": "docker",
    "image": "quay.io/conversica/base",
    "commit": "true"
  }],
  "provisioners": [{
    "type": "ansible-local",
    "inventory_groups": "analytics",
    "playbook_file": "analytics.yml"
  }],
  "post-processors": [
    [{
      "type": "docker-tag",
      "repository": "quay.io/conversica/analytics-conf",
      "force": "true"
    }]
  ]
}
```


Provisioners: Ansible Local

Added to Packer around 0.4.0 (2013)

Requires Ansible and Python to already be installed by the Dockerfile.

Packer copies the complete Ansible directory structure from the local host to the machine image (source and target dirs are customizable).

Ansible Local: Example

```
{  
  "type": "ansible-local",  
  "extra_arguments": ["--skip-tags 'datadog,build'"],  
  "inventory_groups": "ai-api",  
  "group_vars": "../group_vars/",  
  "playbook_dir": "..",  
  "playbook_file": "../ai-api.yml"  
}
```

Provisioner: Ansible Remote

Introduced in 0.9.0 (Feb 2016)

Does not require Ansible to be installed in machine image

Dynamically creates an Ansible inventory file

Runs an SSH server

Executes ansible-playbook against machine image

File transfers via SFTP

Ansible Remote: Example

```
{  
  "type":          "ansible",  
  "extra_arguments": ["--skip-tags 'datadog,build'"],  
  "groups":        "ai-api",  
  "playbook_file": "ai-api.yml"  
}
```

Q & A

The background of the slide is a stylized landscape. The top half is a solid yellow color representing the sky, with two white, fluffy clouds. The bottom half is a range of blue mountains, with the foreground mountains in a darker blue and the background mountains in a lighter blue. The text 'Q & A' is centered in the yellow sky area.

Evaluate this session

<http://lfnw.org/node/4010>

Session videos will be posted to the LinuxFest Northwest YouTube channel.

Thank you!