From Dockerfiles to Ansible Container

Tomas Tomecek

/whois "Tomáš Tomeček"

/whois "Tomáš Tomeček"

- hacker, developer, tinker, speaker, teacher
- contributing to *
- ~~ops~~ engineer

/whois "Tomáš Tomeček"

- hacker, developer, tinker, speaker, teacher
- contributing to *
- ~~ops~~ engineer
- Red Hat
- containerization team

Agenda

1. Dockerfiles and docker-compose.
2. Ansible and Ansible Container.
3. Transition.
4. Ansible Container deep dive.
5. ~~demo~~

Dockerfile

- A recipe to create container images.

Dockerfile

- A recipe to create container images.
- New file-format, different from standards like `json` and `yaml`.

Dockerfile

- A recipe to create container images.
- New file-format, different from standards like `json` and `yaml`.
- The format itself doesn't have a formal specification.

Sample Dockerfile

```
FROM registry.fedoraproject.org/fedora:26
RUN dnf install -y git python-pip gcc python-devel \
    postgresql-devel redhat-rpm-config python2-pytest npm && \
    npm install -g bower

ARG USER_ID=1000
RUN useradd -o -u ${USER_ID} django && \
    mkdir -p /opt/app && \
    chown django:django /opt/app
USER django

WORKDIR /opt/app
COPY ./requirements.txt /opt/app/
RUN pip install --user -r ./requirements.txt
COPY ./requirements-devel.txt /opt/app/
RUN ["pip", "install", "--user", "-r", "./requirements-devel.txt"]
```

Sample Dockerfile (cont.)

```
COPY ./install_static_data.sh /opt/app
COPY ./bower.json /opt/app
RUN ./install_static_data.sh

COPY . /opt/app/
USER root
RUN chown -R django:django .
USER django

# database needs to be set up before web can start serving requests
CMD sleep 7 && exec python /opt/app/manage.py runserver -v3 0.0.0.0:8000
```

Ecosystem

- Service images from Docker Hub.
  - Huge library
  - Quality varies
  - Freshness varies

Ecosystem

- Service images from Docker Hub.
  - Huge library
  - Quality varies
  - Freshness varies
- It's common to orchestrate with `docker-compose`
  - hand-crafted docker-compose.yml

Conclussion (the good)

- Wide usage.
- Good docs for the tooling.
- Docker Hub has a big collection of images.
- Actively developed.
- Easy to start.

Conclussion (the bad)

- Can't easily parse dockerfiles.
- Development of Dockerfiles was kind-of frozen. (?)
- Dockerfiles are essentialy just shell scripts + metadata.
- You need to pick a vendor you trust for your images.
- Dockerfiles have some, old outstanding issues.
- Dockerfiles and `docker-compose` can only be used with docker containers.

Conclusion (the bad)

- Can't easily parse dockerfiles.
- Development of Dockerfiles was kind-of frozen. (?)
- Dockerfiles are essentialy just shell scripts + metadata.
- You need to pick a vendor you trust for your images.
- Dockerfiles have some, old outstanding issues.
- ~~Dockerfiles and `docker-compose` can only be used with docker containers.~~
  - Docker platform and kubernetes

More complex script example

More complex script example

```
RUN set -eux; \
    \
# this "case" statement is generated via "update.sh"
    %%ARCH-CASE%%; \
    \
    url="https://golang.org/dl/go${GOLANG_VERSION}.${goRelArch}.tar.gz"; \
    wget -O go.tgz "$url"; \
    echo "${goRelSha256} *go.tgz" | sha256sum -c -; \
    tar -C /usr/local -xzf go.tgz; \
    rm go.tgz; \
    \
    if [ "$goRelArch" = 'src' ]; then \
        echo >&2; \
        echo >&2 'error: UNIMPLEMENTED'; \
        echo >&2 'TODO install golang-any from jessie-backports for GOROOT_BOO
        echo >&2; \
        exit 1; \
    fi; \
```

More complex script example (cont.)

```
    \
    export PATH="/usr/local/go/bin:$PATH"; \
    go version
```

Source: https://github.com/docker-library/golang/blob/master/Dockerfile-debian.template#L14

Ansible

- Automation platform.
- No daemons — easy to set up.
- Machine and human friendly language — yaml.
- Powerful.
- Hundreds of modules.

Ansible glossary

- Tasks
- Playbooks
- Plays
- Roles

Transition

- Good
  - Full Ansible power.
  - You can utilize your (existing) Ansible roles.
  - Those roles can deploy any artifacts.
  - One consistent tool to manage whole infrastructure.
- Bad
  - shell → Ansible
  - You may need to learn new things. (is this actually bad?)
  - Some Ansible tasks are more complex than Dockerfile instructions.

COPY vs copy module

```
COPY file /path/inside/image
```

vs.

```
- name: Install file
  copy:
    src: file
    dest: /path/inside/image
```

COPY vs copy module

```
COPY file /path/inside/image
RUN chmod 0644 /path/inside/image && chown app /path/inside/image
```

vs.

```
 - name: Install file
   copy:
     src: file
     dest: /path/inside/image
     owner: app
     mode: 0644
```

Ansible Container tool

- `github.com/ansible/ansible-container`
- Building container images from Ansible roles.
- Utilize existing Ansible Galaxy roles.
- Complete management of containerized projects.
- Dev environment overrides.
- Adds a single layer on top of your base image by default.
- Dockerfile → Ansible Role converter.
- Doesn't require python in target image.
- Doesn't require ansible on your host.

Workflow

1. `init`
2. `install`
3. `build`
4. `run`
5. `push`
6. `deploy`

container.yml

- Single definition of the application.
  - Build images.
  - How to run in development.
  - How to deploy to a production environment.
- ansible-container configuration

container.yml

- Single definition of the application.
  - Build images.
  - How to run in development.
  - How to deploy to a production environment.
- ansible-container configuration
- https://github.com/chouseknecht/django-gulp-nginx/blob/master/container.yml

container.yml

```
defaults:
  POSTGRES_USER: django
  POSTGRES_PASSWORD: sesame
  POSTGRES_DB: django
  DJANGO_ROOT: /django
  DJANGO_USER: django
  DJANGO_PORT: 8080
  DJANGO_VENV: /venv
  NODE_USER: node
  NODE_HOME: /node
  NODE_ROOT: ''
  GULP_DEV_PORT: 8080
```

container.yml (cont.)

```
services:
  django:
    from: 'centos:7'
    roles:
    - role: django-gunicorn
    environment:
      DATABASE_URL: 'pgsql://{{ POSTGRES_USER }}:{{ POSTGRES_PASSWORD }}@postgre
      DJANGO_ROOT: '{{ DJANGO_ROOT }}'
      DJANGO_VENV: '{{ DJANGO_VENV }}'
    expose:
    - '{{ DJANGO_PORT }}'
    working_dir: '{{ DJANGO_ROOT }}'
    links:
    - postgresql
    user: '{{ DJANGO_USER }}'
    command: ['/usr/bin/dumb-init', '{{ DJANGO_VENV }}/bin/gunicorn', -w, '2', -
    entrypoint: [/usr/bin/entrypoint.sh]
```

container.yml

Development overrides

```
  django:
    dev_overrides:
      volumes:
      - '$PWD:{{ DJANGO_ROOT }}'
      command: [/usr/bin/dumb-init, '{{ DJANGO_VENV }}/bin/python', manage.py, r
```

```
  nginx:
    dev_overrides:
      ports: []
      command: /bin/false
```

We need your feedback!

Conclusion

- One tool can't fit everyone.
- Pick the one which suits you best.
- `ansible-container` aiming to hit 1.0 soon.
- `github.com/moby/buildkit`

Thank you!

- ansible/ansible-container
- ansible/ansible-container-demo
- `#ansible-container` channel on irc.freenode.net
- https://groups.google.com/forum/#!forum/ansible-container
- TomasTomecek/open-source-summit-2017-talk