docker_registry

May 7, 2020

1 Docker registry

1.1 What is a Docker registry?

- It's a file storage with an API compatible with Docker interface.
- By analogy: it's for Docker images what Git is for source code, but...
- ... it's more a deploy tool than version control system.
- It stores Dockerfiles products but without Dockerfiles themselves.
- In GCP it's called Container Registry, but it's really store images not containers.
- Docker Hub is public registry which works as PyPI for Python.

1.2 Basic workflow

1. Push code to git repo.
2. Run CI pipeline:

- Build Docker images.
- Run tests, linters etc.

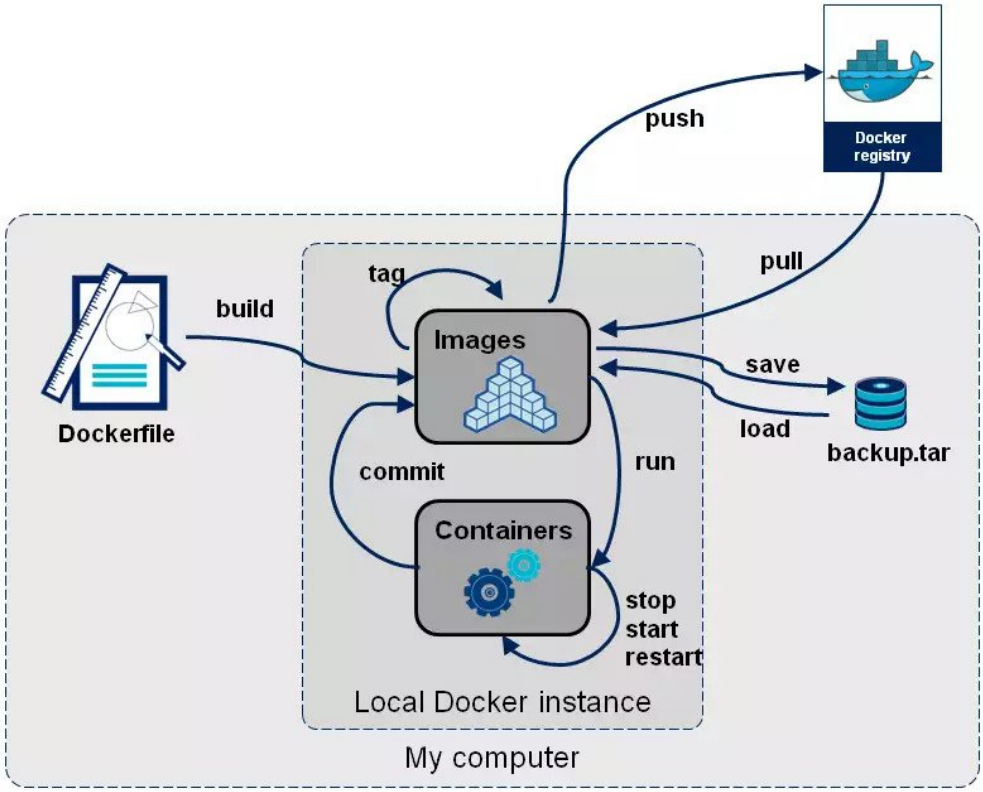
3. Run deploy through Jenkins:
4. Pull code on production.
5. Build containers.
6. Run containers.

1.3 Workflow with Docker registry

1. Push code to git repo.
2. Run CI pipeline:

- Build Docker images.
- Run tests, linters etc.
- **Push images to registry.**

3. Run deploy through Jenkins:
4. Pull ~~code~~ **images** on production.
5. ~~Build containers.~~
6. Run containers.


Docker registry
push
pull
tag
build
Images
save
load
backup.tar
Dockerfile
commit
run
Containers
stop
start
restart
Local Docker instance
My computer

1.4 To use or not to use?

1.4.1 Pros

- Simpler deploy - less moving parts.
- The same images which pass CI go to production.
- Generally accepted flow, considered as good practice.
- Fast revert - no need to rebuild images.
- Must-have to work in distributed environments like Kubernetes.

1.4.2 Cons

- More complicated CI - need to push images.
- Need to wait for CI before start deploy but...
- ... it might be bypassed with auto-deploy.
- Registry costs money and requires maintenance.

1.5 How it's done on BSP?

1.5.1 Delivering images to Container Registry

bitbucket-pipelines.yml:

```
...
  - docker-compose build
  - invoke ci
  - echo $GCP_JSON_KEY | docker login -u _json_key --password-stdin https://gcr.io
  - invoke docker-registry.push-images
...
```

tasks/docker_registry.py:

```
import invoke

IMAGES = [
    'gcr.io/brandsafety-rtbhouse-biz/bsp_app',
    'gcr.io/brandsafety-rtbhouse-biz/bsp_web',
]  # same as in docker/docker-compose.yml


@invoke.task()
def push_images(c):
    branch_name = c.run('git rev-parse --abbrev-ref HEAD', hide='out').stdout.strip()
    short_hash = c.run('git rev-parse --short HEAD', hide='out').stdout.strip()
```

```
    tag = f'{branch_name}-{short_hash}'
    for image in IMAGES:
        c.run(f'docker tag {image} {image}:{tag}')
        c.run(f'docker push {image}:{tag}')
        if branch_name == 'master':
            c.run(f'docker tag {image} {image}:latest')
            c.run(f'docker push {image}:latest')
        elif branch_name == 'stage':
            c.run(f'docker tag {image} {image}:stage')
            c.run(f'docker push {image}:stage')
```

1.5.2 Running images on production

docker/docker-comopse.yml:

```
version: "3.7"
services:
  app:
    image: gcr.io/brandsafety-rtbhouse-biz/bsp_app:${DOCKER_TAG:-latest}
    ...
  web:
    image: gcr.io/brandsafety-rtbhouse-biz/bsp_web:${DOCKER_TAG:-latest}
    ...
  celery:
    image: gcr.io/brandsafety-rtbhouse-biz/bsp_app:${DOCKER_TAG:-latest}
    ...
```

$ docker-compose up -d
... unfortunately not so simple:
ansible/roles/deploy/tasks/main.yml:

```
- name: Copy docker-compose files
  copy:
    src: "{{ playbook_dir }}/../docker/{{ item }}"
    dest: "{{ project_root }}/docker/{{ item }}"
  with_items:
    - docker-compose.yml
    - docker-compose.prod.yml

- name: Docker login
  vars:
    key_file: "{{ lookup('env', 'CONTAINER_REGISTRY_KEY') }}"
  docker_login:
    username: _json_key
    registry_url: gcr.io
    password: "{{ lookup('file', key_file) | string }}"
```

```
- name: Run containers
  docker_compose:
    project_src: "{{ project_root }}"
    pull: yes
    state: present
    timeout: 60
  environment:
    - DOCKER_TAG: "{{ lookup('env', 'DOCKER_TAG') or 'latest' }}"
```

1.6 How to setup Docker registry in project?

- Enable Container Registry API in a project in GCP.
- Create Service Account with permissions to use Cloud Storage.
- Downalod JSON key and pass it to `docker login`.
- Done. Commands like `docker push / pull` should work.

1.7 End thoughts

- Docker registry in most cases is NOT a must-have.
- Some registries supports webhooks eg. GitLab, but not GCP.

1.8 Any questions?

1.9 The end!


GOOD