

Semantic Validation of Docker Compose Files

Supervisors:

Christof Marti
Spillner Josef

Present by:

Ali Daghighi

Outline

- Overview of Docker Compose
- Additional issues
- Extend SPLab tool
- Tool limitations
- Live demo
- Summary

Overview of Docker Compose

- Compose is a tool for defining and running multi-container Docker applications.
- With Compose, You use a YAML file to configure your application's services.
- With a single command, you create and start all the services from your configuration

Overview of Docker Compose

The features of Compose that make it effective are:

- Multiple isolated environments on a single host
- Preserve volume data when containers are created
- Only recreate containers that have changed
- Variables and moving a composition between environments

Additional issues

- Duplicate service name
- Duplicate container name
- Duplicate image
- Duplicate port
- Invalid volume directory

Extend SPLab tool

- This is a sample Docker Compose file

```
1  version: '3'
2  services:
3    nginx:
4      container_name: some-nginx
5      image: nginx:1.13
6      restart: always
7      ports:
8        - 3306:80
9        - 443:443
10     expose:
11       - '80'
12     volumes:
13       - ./nginx/conf.d:/etc/nginx/conf.d
14     depends_on:
15       - app
16
17     nginx:
18       container_name: some-mysql
19       image: mysql/mysql-server:5.7
20       environment:
21         MYSQL_DATABASE: test
22         MYSQL_ROOT_PASSWORD: hellokoding
23         MYSQL_ROOT_HOST: '%'
24       ports:
25         - "3306:3306"
26       restart: always
```

Extend SPlab tool

- This is a sample Docker Compose file

BUT!

```
1  version: '3'
2  services:
3    nginx:
4      container_name: some-nginx
5      image: nginx:1.13
6      restart: always
7      ports:
8        - 3306:80
9        - 443:443
10     expose:
11       - '80'
12     volumes:
13       - ./nginx/conf.d:/etc/nginx/conf.d
14     depends_on:
15       - app
16
17     nginx:
18       container_name: some-mysql
19       image: mysql/mysql-server:5.7
20       environment:
21         MYSQL_DATABASE: test
22         MYSQL_ROOT_PASSWORD: hellokoding
23         MYSQL_ROOT_HOST: '%'
24       ports:
25         - "3306:3306"
26       restart: always
```

Extend SPLab tool

- It's not valid

```
1  version: '3'
2  services:
3    nginx:
4      container_name: some-nginx
5      image: nginx:1.13
6      restart: always
7      ports:
8      X 3306:80
9        - 443:443
10     expose:
11       - '80'
12     volumes:
13       - ./nginx/conf.d:/etc/nginx/conf.d
14     depends_on:
15       - app
16
17     nginx:
18       container_name: some-mysql
19       image: mysql/mysql-server:5.7
20       environment:
21         MYSQL_DATABASE: test
22         MYSQL_ROOT_PASSWORD: hellokoding
23         MYSQL_ROOT_HOST: '%'
24       ports:
25       X "3306:3306"
26       restart: always
```


Extend SPLab tool

- Parse .yaml file
- Looking for issues
- Print the result

Extend SPLab tool

- One of the Challenges is duplicate service name:

```
services:
  a:
    image: img:tag
  | ports:
    - port:port
  b:
    image: img:tag
    ports:
    - "3306:3306"
    restart: always
```

Correct

```
services:
  a:
    image: img:tag
    ports:
    - port:port
  a:
    image: img:tag
    ports:
    - "3306:3306"
    restart: always
```

Incorrect

Extend SPLab tool

- One of the Challenges is duplicate service name:

```
services:
  a:
    image: img:tag
  |
  ports:
    - port:port
  b:
    image: img:tag
    ports:
      - "3306:3306"
    restart: always
```

Correct

```
{'version': '3',
 'services': {
   'a': {
     'image': 'img:tag',
     'ports': ['port:port']
   },
   'b': {
     'image': 'img:tag',
     'ports': ['3306:3306'],
     'restart': 'always'
   }
 }
```

```
services:
  a:
    image: img:tag
    ports:
      - port:port
  a:
    image: img:tag
    ports:
      - "3306:3306"
    restart: always
```

Incorrect

```
{ 'version': '3',
  'services': {
    'a': {
      'image': 'img:tag',
      'ports': ['3306:3306'],
      'restart': 'always'
    }
  }
}
```

Extend SPLab tool

- One of the Challenges is duplicate service name:

```
services:
  a:
    image: img:tag
  |
  ports:
    - port:port
  b:
    image: img:tag
    ports:
      - "3306:3306"
    restart: always
```

Correct

```
{'version': '3',
 'services': {
   'a': {
     'image': 'img:tag',
     'ports': ['port:port']
   },
   'b': {
     'image': 'img:tag',
     'ports': ['3306:3306'],
     'restart': 'always'
   }
 }
```

```
services:
  a:
    image: img:tag
    ports:
      - port:port
  a:
    image: img:tag
    ports:
      - "3306:3306"
    restart: always
```

Incorrect

```
{ 'version': '3',
  'services': {
    'a': {
      'image': 'img:tag',
      'ports': ['3306:3306'],
      'restart': 'always'
    }
  }
}
```

Extend SPLab tool

- One of the Challenges is duplicate service name:

```
{ 'version': '3',  
  'services': {  
    'a': {  
      'image': 'img:tag',  
      'ports': ['3306:3306'],  
      'restart': 'always'  
    }  
  }  
}
```

```
services:  
  a:  
    image: img:tag  
    ports:  
      - port:port  
  a:  
    image: img:tag  
    ports:  
      - "3306:3306"  
    restart: always
```

- “yaml” library cant help for parsing these kind of Docker Compose.

Tool limitations

- Only from github
- DockerFile
- **Very general!** There is still tags this tool can not validate it!

Live demo

It's show time!

Summary

- Docker Compose
- Issues and Tool
- Demo

Thank you