

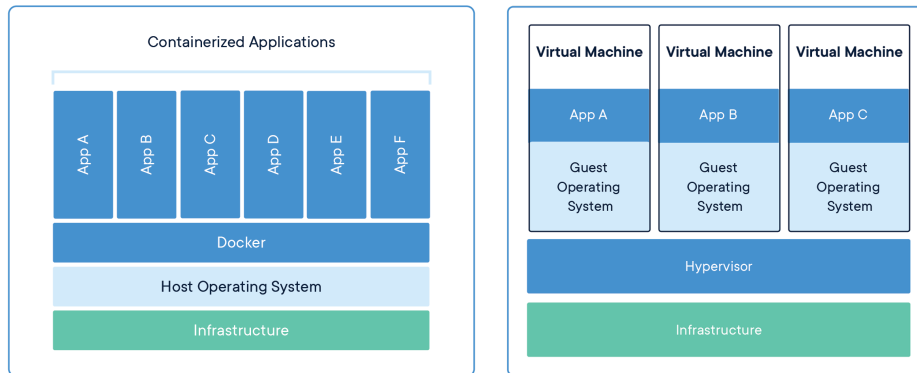
Tango and Docker containers

Michal Liszcz (S2Innovation), Matteo Di Carlo (INAF)

ICALEPCS 2019, New York

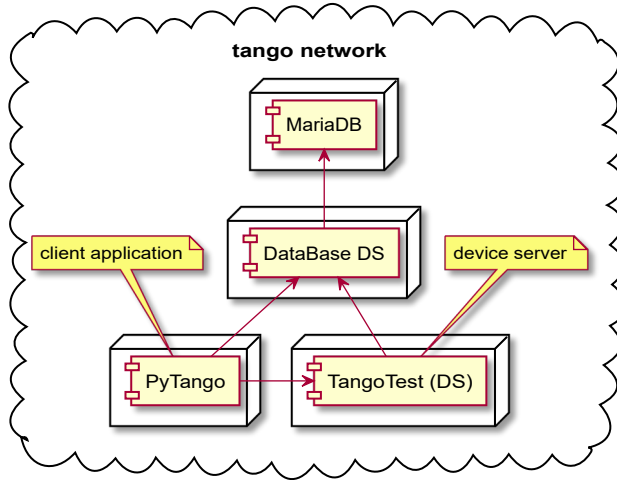
Containers

1. Easy way for packaging applications and dependencies
2. Repeatable environments for development, build, testing and CI stages
3. Possibility to scale applications: from dev's laptop to a prod. server
4. A combination of kernel primitives: namespaces, cgroups and chroot
5. **Docker** is an implementation of Open Container Initiative (OCI) spec.
6. Containers vs Virtual Machines (source: [docker.com](https://www.docker.com)):



Tango on Docker—hands-on tutorial

1. Four Docker containers connected into a dedicated network
2. Materials available online: <https://tiny.cc/TangoDocker>



Hands-on (1/4)—Network and MariaDB

```
$ docker network create tango-nw
```

```
$ docker run -it --rm --name mariadb --network tango-nw \  
-e MYSQL_ROOT_PASSWORD=secret mariadb:10.4
```

Hands-on (2/4)—DataBase DS

```
$ docker run -it --rm --name databaseds --network tango-nw ubuntu:bionic
```

```
$ export DEBIAN_FRONTEND=noninteractive  
$ apt install -y tango-db
```

```
$ mysql -h mariadb -u root -psecret
```

```
mysql> create database tango;  
mysql> exit
```

```
$ mysql -h mariadb -u root -psecret -D tango \  
  < /usr/share/dbconfig-common/data/tango-db/install/mysql
```

```
$ TANGO_HOST=localhost:10000 MYSQL_HOST=mariadb:3306 MYSQL_USER=root \  
  MYSQL_PASSWORD=secret MYSQL_DATABASE=tango \  
  /usr/lib/tango/DataBases 2 -ORBEndPoint giop:tcp::10000
```

Hands-on (3/4)—TangoTest (device server)

```
$ docker run -it --rm --name tango-test --network tango-nw ubuntu:bionic
```

```
$ export DEBIAN_FRONTEND=noninteractive  
$ apt install -y tango-test
```

```
$ export TANGO_HOST=databases:10000
```

```
$ /usr/lib/tango/tango_admin --add-server \  
TangoTest/test01 TangoTest icalpcs/test/1
```

```
$ /usr/lib/tango/TangoTest test01
```

Hands-on (4/4)—PyTango (client app)

```
$ docker run -it --rm --name tango-client --network tango-nw ubuntu:bionic
```

```
$ export DEBIAN_FRONTEND=noninteractive  
$ apt install -y python-pytango
```

```
$ export TANGO_HOST=databases:10000  
$ python
```

```
>>> import tango  
>>> proxy = tango.DeviceProxy("icalepcs/test/1")  
>>> proxy.Status()  
'The device is in RUNNING state.'  
>>> proxy.double_scalar  
255.84406383401958
```

Dockerfile

```
# pytango.Dockerfile
FROM ubuntu:bionic
ENV DEBIAN_FRONTEND=noninteractive
RUN apt-get update \
  && apt-get install -y python-pytango
ENTRYPOINT ["/usr/bin/python"]
```

```
$ docker build -t tango-pytango -f pytango.Dockerfile .
$ docker run -it --rm tango-pytango
```

```
>>> import tango
```


Docker Compose (1/2)

```
$ /usr/bin/tree
.
├── databaseds.Dockerfile
├── docker-compose.yaml
├── mariadb.Dockerfile
├── pytango.Dockerfile
└── tangotest.Dockerfile

0 directories, 5 files
$ docker-compose -p icalepcs up
...
tangotest_1 | Ready to accept request
```

```
$ docker attach icalepcs_pytango_1
```

```
>>> import tango
>>> proxy = tango.DeviceProxy("sys/test/1")
>>> proxy.Status()
'The device is in RUNNING state.'
```

Docker Compose (2/2)

```
version: '3'
services:
  mariadb:
    build:
      context: .
      dockerfile: mariadb.Dockerfile
    environment:
      MYSQL_ROOT_PASSWORD: secret
  databaseds:
    build:
      context: .
      dockerfile: databaseds.Dockerfile
    environment:
      TANGO_HOST: 'localhost:10000'
      MYSQL_HOST: 'mariadb:3306'
      MYSQL_USER: root
      MYSQL_PASSWORD: secret
```

```
tangotest:
  build:
    context: .
    dockerfile: tangotest.Dockerfile
  environment:
    TANGO_HOST: databaseds:10000
pytango:
  stdin_open: true
  tty: true
  build:
    context: .
    dockerfile: pytango.Dockerfile
  environment:
    TANGO_HOST: databaseds:10000
```