

# Cloud Computing

## Vežbe 1 – rešenja

### Primer 1: hello-world

1. `$ docker images`
2. `$ docker pull hello-world`
3. `$ docker run hello-world`
4. `$ docker ps`
5. `$ docker inspect hello-world`
6. `$ docker rmi hello-world`

### Primer 2: ubuntu

1. `$ docker pull ubuntu:18.04`
2. `$ docker run -it ubuntu:18.04`  
`# mkdir /primer`
3. `$ docker ps` (u novom terminalu kako bi kontejner ostao pokrenut)
4. `$ docker stop <CONTAINER_ID>`
5. `$ docker rm <CONTAINER_ID>`

### Primer 3: node-server

U okviru datoteke *primer3* nalaze se fajlovi koji su potrebni za funkcionisanje servera. Pokrenutom serveru može da se pristupi na portu 3001. Aplikacija dozvoljava unos novih poruka, a svaka nova poruka se ispisuje u okviru konzole.

1. Pogledati prezentaciju za Dockerfile
2. `$ docker build -t server:1.0 .`

3. `$ docker run -d --name node-server -p 3001:3001 server:1.0`
4. `$ docker inspect node-server`
5. `$ docker stats node-server`
6. `$ docker stop node-server`
7. `$ docker start node-server`
8. `$ docker attach node-server`
9. `$ docker logs node-server`

### **Primer 4: python igra**

U okviru datoteke *primer4* nalazi se fajl *rng.py* koji sadrži kod igrice.

Igrica zahteva unos minimalnog i maksimalnog broja, a zatim ispisuje nasumično jedan broj iz datog intervala.

1. `$ docker build -t python-igrica .`  
(pogledati prezentaciju za Dockerfile)
2. `$ docker run --name igrica -it python-igrica`
3. `$ docker stop igrica`
4. `$ docker run --name igrica2 --rm -it python-igrica`
5. `$ docker stop igrica2`

### **Primer 5:**

1. `$ docker ps -a`
2. `$ docker ps -a --format "table  
{{.ID}}\t{{.Image}}\t{{.Names}}\t{{.Status}}"`
3. `$ docker container prune`
4. `$ docker image prune -a`