

▶ docker run ubuntu					
▶ docker ps					
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
▶ docker ps -a					
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
45aacca36850	ubuntu	"/bin/bash"	43 seconds ago	Exited (0) 41 seconds ago	

```
# Install Nginx.
RUN \
    add-apt-repository -y ppa:nginx/stable && \
    apt-get update && \
    apt-get install -y nginx && \
    rm -rf /var/lib/apt/lists/* && \
    echo "\ndaemon off;" >> /etc/nginx/nginx.conf && \
    chown -R www-data:www-data /var/lib/nginx

# Define mountable directories.
VOLUME ["/etc/nginx/sites-enabled", "/etc/nginx/certs", "/etc/nginx/conf.d"]

# Define working directory.
WORKDIR /etc/nginx

# Define default command.
CMD ["nginx"]
```

```
ARG MYSQL_SERVER_PACKAGE_URL=https://repo.mysql.com/yum/mysql-8.0-community/docker/x86_64
ARG MYSQL_SHELL_PACKAGE_URL=https://repo.mysql.com/yum/mysql-tools-community/el/7/x86_64

# Install server
RUN rpmkeys --import https://repo.mysql.com/RPM-GPG-KEY-mysql \
    && yum install -y $MYSQL_SERVER_PACKAGE_URL $MYSQL_SHELL_PACKAGE_URL libpqquality \
    && yum clean all \
    && mkdir /docker-entrypoint-initdb.d

VOLUME /var/lib/mysql

COPY docker-entrypoint.sh /entrypoint.sh
COPY healthcheck.sh /healthcheck.sh
ENTRYPOINT ["/entrypoint.sh"]
HEALTHCHECK CMD /healthcheck.sh
EXPOSE 3306 33060
CMD ["mysqld"]
```

```

# Pull base image.
FROM ubuntu:14.04

# Install.
RUN \
    sed -i 's/# \(\.*multiverse$\)/\1/g' /etc/apt/sources.list && \
    apt-get update && \
    apt-get -y upgrade && \
    apt-get install -y build-essential && \
    apt-get install -y software-properties-common && \
    apt-get install -y byobu curl git htop man unzip vim wget && \
    rm -rf /var/lib/apt/lists/*

# Add files.
ADD root/.bashrc /root/.bashrc
ADD root/.gitconfig /root/.gitconfig
ADD root/.scripts /root/.scripts

# Set environment variables.
ENV HOME /root

# Define working directory.
WORKDIR /root

# Define default command.
CMD ["bash"]

```

```
docker run ubuntu [COMMAND]
```

```
docker run ubuntu sleep 5
```

FROM Ubuntu

CMD sleep 5

CMD command param1

CMD sleep 5

CMD ["command", "param1"]

CMD ["sleep", "5"]

CMD ["sleep 5"]



```
docker build -t ubuntu-sleeper .
```

```
docker run ubuntu-sleeper
```



After 5 seconds then exit

FROM Ubuntu

CMD sleep 5

```
▶ docker run ubuntu-sleeper sleep 10
```

Command at Startup: sleep 10

FROM Ubuntu

ENTRYPOINT ["sleep"]

```
▶ docker run ubuntu-sleeper 10
```

Command at Startup: sleep 10

```
▶ docker run ubuntu-sleeper
sleep: missing operand
Try 'sleep --help' for more information.
```

Command at Startup: sleep

FROM Ubuntu

ENTRYPOINT ["sleep"]

CMD ["5"]

```
▶ docker run ubuntu-sleeper
sleep: missing operand
Try 'sleep --help' for more information.
```

Command at Startup: sleep 5

```
▶ docker run ubuntu-sleeper 10
```

Command at Startup: sleep 10

```
▶ docker run --entrypoint sleep2.0 ubuntu-sleeper 10
```

Command at Startup: sleep2.0 10

```
▶ docker run --name ubuntu-sleeper ubuntu-sleeper
```

```
▶ docker run --name ubuntu-sleeper ubuntu-sleeper 10
```

pod-definition.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper-pod
spec:
  containers:
    - name: ubuntu-sleeper
      image: ubuntu-sleeper
      args: ["10"]
```

```
▶ kubectl create -f pod-definition.yml
```

FROM Ubuntu

ENTRYPOINT ["sleep"]

CMD ["5"]

```
▶ docker run --name ubuntu-sleeper \
  --entrypoint sleep2.0
  ubuntu-sleeper 10
```

pod-definition.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper-pod
spec:
  containers:
    - name: ubuntu-sleeper
      image: ubuntu-sleeper
      command: ["sleep2.0"]
      args: ["10"]
```

```
▶ kubectl create -f pod-definition.yml
```

FROM Ubuntu

ENTRYPOINT ["sleep"]

CMD ["5"]

pod-definition.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu-sleeper-pod
spec:
  containers:
    - name: ubuntu-sleeper
      image: ubuntu-sleeper
      command: ["sleep2.0"]
      args: ["10"]
```