

Q & A

# Ansible FAQ (1)

- **Question: Timeout at the “Wait for connection” step**

*... I have not been able to get past the playbook step of checking that the instances are created (Role directory "openstack-instance:", step name "Wait for connection"). ...*

- **Cause:**

*Only private IP addresses are assigned to instances, which are not accessible from the Internet*

- **Solution:**

*Connect to University of Melbourne Student VPN*

# Ansible FAQ (2)

- **Question:**

*I am trying to run ansible script to create an instance and it keeps failing with "failed: [localhost] (item={'name': 'demo-2-adv', 'volumes': ['demo-vol-2-1-adv', 'demo-vol-2-2-adv']}) => {"ansible\_loop\_var": "item", "changed": false, "extra\_data": null, "item": {"name": "demo-2-adv", "volumes": ["demo-vol-2-1-adv", "demo-vol-2-2-adv"]}, "msg": "BadRequestException: 400: Client Error for url: https://nova.rc.nectar.org.au:8774/v2.1/f2aba5e6fc964fc18cb2078f3de2da8e/os-volumes\_boot, Invalid key\_name provided."}"*

- **Cause:**

*You need to update the playbook to match your settings*

- **Solution:**

*Update the "key\_name" variable to the key pair you have on MRC*

# Ansible FAQ (3)

- **Problem:**

*TASK [openstack-common : Install openstacksdk]*

\*\*\*\*\*

*changed: [localhost]*

*TASK [openstack-images : Retrieve all available Openstack images]*

\*\*\*\*\*

*fatal: [localhost]: FAILED! => {"changed": false, "msg": "**openstacksdk is required for this module**"}*

- **Debug: Run playbook in verbose mode**

*./openrc.sh; ansible-playbook -vvv <playbook>*

# Ansible FAQ (3)

- **Debug:**

*The full traceback is:*

*WARNING: The below traceback may *\*not\** be related to the actual failure.*

*File*

*"/tmp/ansible\_os\_image\_info\_payload\_qlzzWJ/ansible\_os\_image\_info\_payload.zip/ansible/module\_utils/openstack.py", line 116, in*

*openstack\_cloud\_from\_module*

*sdk = importlib.import\_module('openstack')*

*File "/usr/lib/python2.7/importlib/\_\_init\_\_.py", line 37, in import\_module*

*...*

*from openstack import utils*

*File "/usr/local/lib/python2.7/dist-packages/openstack/utils.py", line 13, in*

*<module>*

***import queue***

*fatal: [localhost]: FAILED! => {*

*"changed": false,*

*...*

# Ansible FAQ (3)

- **Cause:**

*OpenStackSDK dropped Python 2 support since version 0.40.0*

<https://pypi.org/project/openstacksdk/#history>

- **Solution:**

- *Use Python 3*
- *Install a version of OpenStackSDK that supports Python 2*

- **Debugging:**

- *Turn on verbose mode (when possible)*
- *Google the error message*
- *Check on Github and StackOverflow for solution*

# Ansible FAQ (3)

- **Cause:**

*OpenStackSDK dropped Python 2 support since version 0.40.0*

<https://pypi.org/project/openstacksdk/#history>

- **Solution:**

- *Use Python 3*
- *Install a version of OpenStackSDK that supports Python 2*

- **Debugging:**

- *Turn on verbose mode (when possible)*
- *Google the error message*
- *Check on Github and StackOverflow for solution*

# Ansible FAQ (4)

- **Question:**

*I've tried to add a Proxy configuration ... I can use apt to update, but still can not connect to the docker repository*

- **Cause:**

*Docker daemon is behind HTTP/HTTPS proxy*

- **Solution:**

- *Create a systemd drop-in directory **/etc/systemd/system/docker.service.d** for the docker service*
- *Create a file called **/etc/systemd/system/docker.service.d/http-proxy.conf** that adds the **HTTP\_PROXY**, **HTTPS\_PROXY** and **NO\_PROXY** environment variables*
- *Flush the changes and restart the Docker service*



# Ansible FAQ (4)

- **Ansible Solution:**

- *Create a file called `http-proxy.conf` that contains*

*[Service]*

*Environment="HTTP\_PROXY=http://wwwproxy.unimelb.edu.au:8000/"*

*Environment="HTTPS\_PROXY=http://wwwproxy.unimelb.edu.au:8000/"*

*Environment="NO\_PROXY=localhost,127.0.0.1,localaddress,172.16.0.0/12,.melbourne.rc.nectar.org.au,.storage.unimelb.edu.au,.cloud.unimelb.edu.au"*

- *file module: Create directory `/etc/systemd/system/docker.service.d`*
- *copy module: Copy the `http-proxy.conf` to `/etc/systemd/system/docker.service.d/`*
- *systemd module: Flush the changes and restart the Docker service*  
*\$ sudo systemctl daemon-reload && sudo systemctl restart docker*

# Docker FAQ & Tips (1)

- **Problem:**

*docker-machine ls showing the machine state is "Timeout"*

```
alwynpan@ravpn-266-1-staff-ten-1-9-239 ~$ docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
manager		virtualbox	Timeout				
worker1		virtualbox	Timeout				
worker2		virtualbox	Timeout				

- **Cause:**

*VPN / Antivirus / Internet Security is on*

- **Debugging:**

*Turn off VPN / Antivirus / Internet Security*

# Docker FAQ & Tips (2)

- **Problem:**

*Dockerised application is running slow*

*Docker commands suddenly stopped working, no error message was given*

*Docker build failed halfway with no error message*

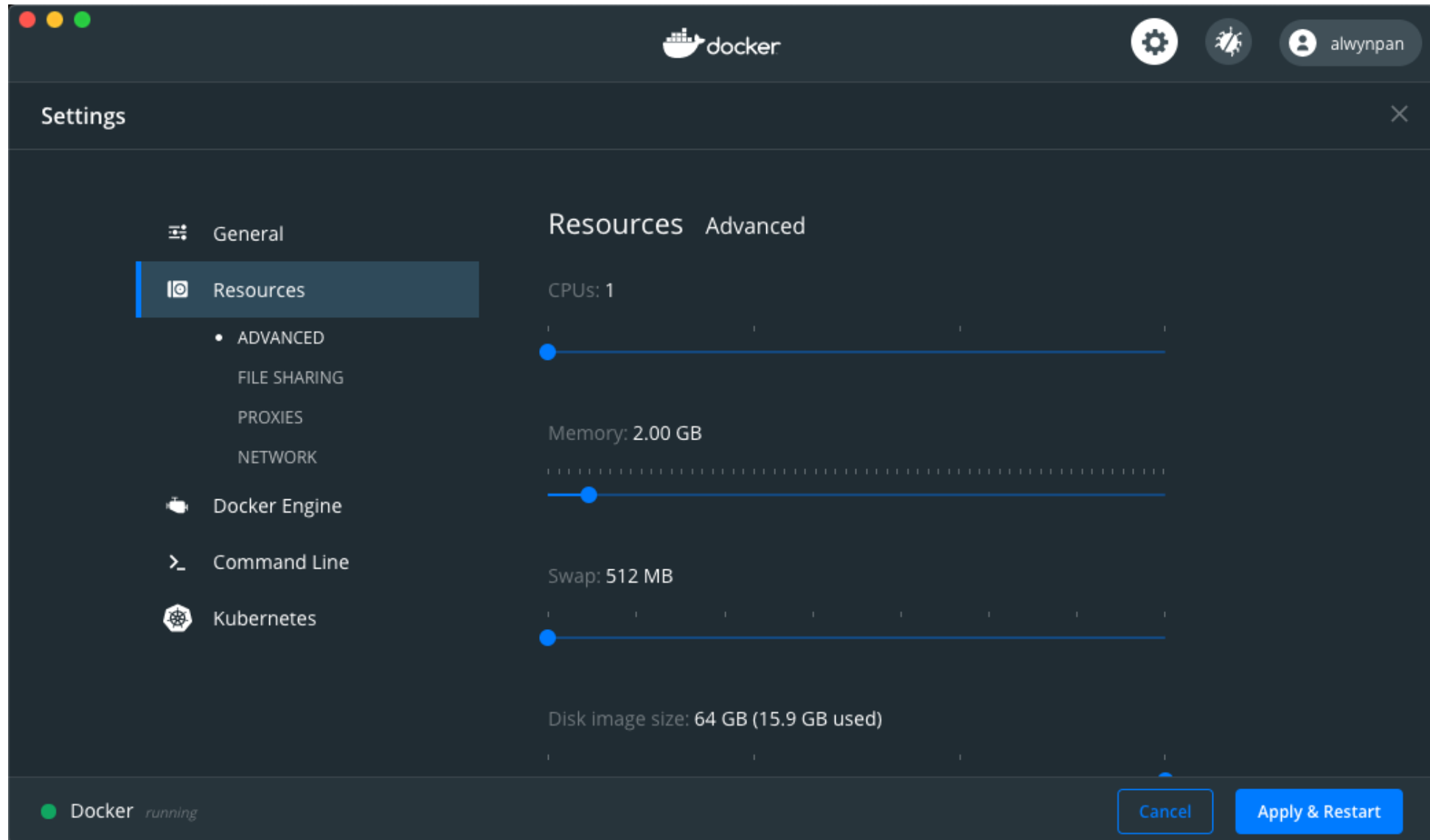
- **Cause:**

*Not enough resources were allocated to Docker*

- **Solution:**

*Check Docker ... Preferences ... Resources ... Advanced, and make sure reasonable resource was allocated to Docker.*

# Docker FAQ & Tips (2)



# Docker FAQ & Tips (3)

- **Question: The difference between COPY and ADD in a Docker file**

```
FROM alpine:latest
```

```
WORKDIR /demo
```

```
RUN mkdir -p copy add1 add2
```

```
COPY ["release-1.18.0.tar.gz", "./copy/"]
```

```
ADD ["release-1.18.0.tar.gz", "./add1/"]
```

```
ADD ["https://github.com/nginx/nginx/archive/release-1.18.0.tar.gz", "./add2/"]
```

# Docker FAQ & Tips (3)

- **Use cases:**

- **COPY**

- Copy local files from a specific location into a Docker image

- **ADD**

- Extract a local *tar* file into a specific directory into a Docker image
    - Download a file *as is* from an URI into a Docker image

# Docker FAQ & Tips (4)

- **Question: The difference between EXPOSE and publish ports**

FROM nginx:latest

EXPOSE 8080

<code>--pids-limit int</code>	Tune container pids limit (set -1 for unlimited)
<code>--privileged</code>	Give extended privileges to this container
<code>-p, --publish list</code>	Publish a container's port(s) to the host
<code>-P, --publish-all</code>	Publish all exposed ports to random ports
<code>--read-only</code>	Mount the container's root filesystem as read
<code>--restart string</code>	Restart policy to apply when a container exits

---

```
$ docker run --name test1 -p 180:80 -d nginx-test:latest
```

```
$ docker run --name test2 -P -d nginx-test:latest
```

```
$ docker run --name test3 -P -d nginx:latest
```

# Docker FAQ & Tips (4)

- **EXPOSE**

*Tells Docker daemon and user which port(s) your service listens on*

- **Publish**

*Publishes the port (no matter exposed or not) to the host*

- **Nginx:latest image [Dockerfile](#)**

```
alwynpan@Alwyns-MBP ~/docker-demo/faq
CONTAINER ID   IMAGE
5b16bd06de34   nginx:latest
920e87fd0606   nginx-test:latest
d751f2d8d00d   nginx-test:latest
```

```
PORTS
0.0.0.0:32775->80/tcp      test3
0.0.0.0:32774->80/tcp, 0.0.0.0:32773->8080/tcp  test2
8080/tcp, 0.0.0.0:180->80/tcp  test1
```



# Docker FAQ & Tips (5)

- **Question: The difference between Build Time ENV and Run Time ENV**

```
FROM python:latest
```

```
ARG OPENSTACKSDK_VER=0.46.0
```

```
ENV REQUESTS_VER=2.18.0
```

```
COPY [ "entrypoint.sh", "/" ]
```

```
RUN pip install openstacksdk==${OPENSTACKSDK_VER}; \  
    chmod +x /entrypoint.sh
```

```
ENTRYPOINT [ "/entrypoint.sh" ]
```

# Docker FAQ & Tips (5)

- **Question: The difference between Build Time ENV and Run Time ENV**

```
FROM python:latest
```

```
ARG OPENSTACKSDK_VER=0.46.0
```

```
ENV OPENSTACKSDK_VER=${OPENSTACKSDK_VER}
```

```
ENV REQUESTS_VER=2.18.0
```

```
COPY [ "entrypoint2.sh", "/" ]
```

```
RUN pip install openstacksdk==${OPENSTACKSDK_VER}; \  
    chmod +x /entrypoint2.sh
```

```
ENTRYPOINT [ "/entrypoint2.sh" ]
```

# Docker FAQ & Tips (4)

- **Built Time Variable (build arg)**

*Variables defined in the Dockerfile, and works at build time only.*

*Use "docker build --build-arg KEY=VALUE" to init or override*

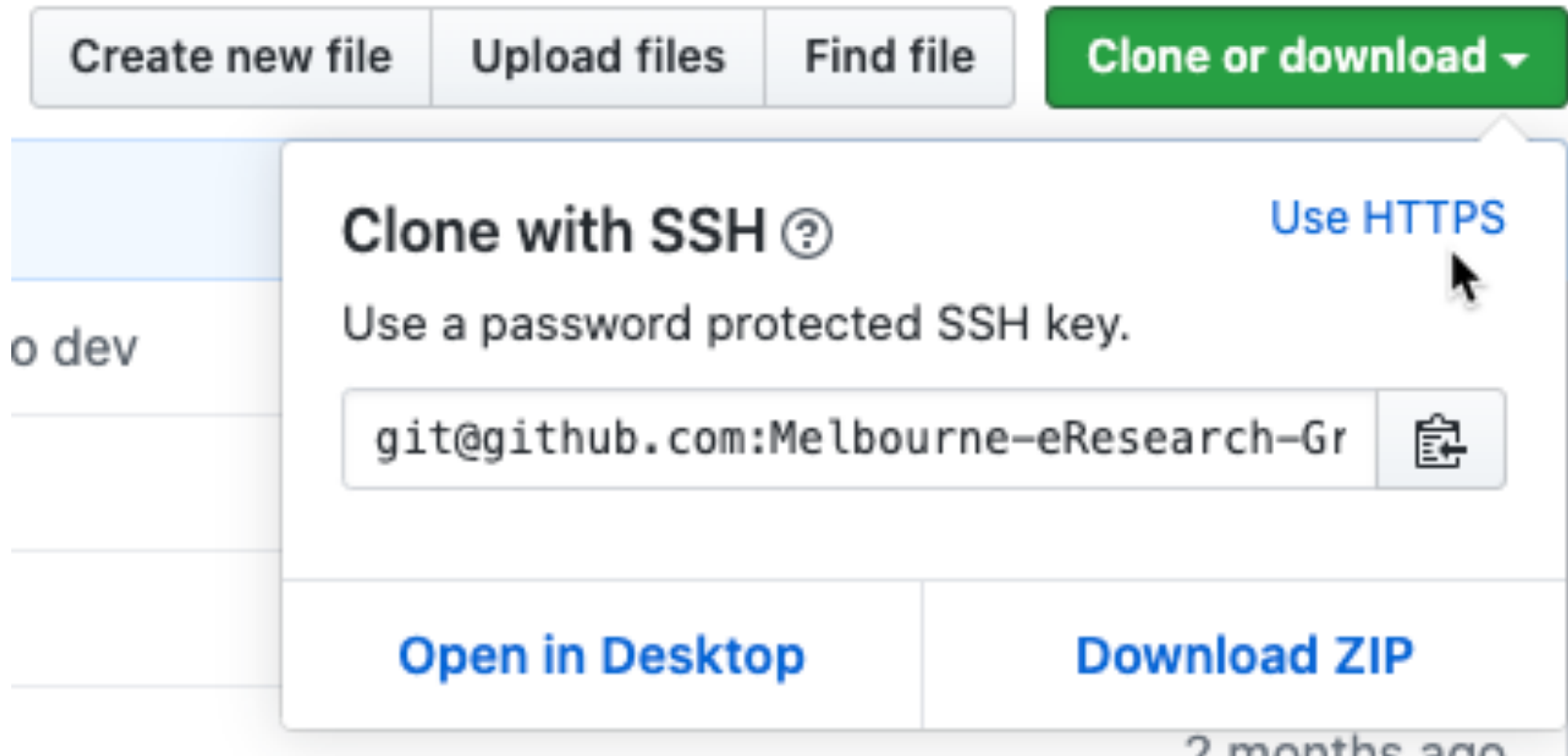
- **Run Time Variable**

*Variables defined in Dockerfile / Entrypoint / application, and works at run time only.*

*Use "docker run -e KEY=VALUE ..." to init or override*

# A New Question

- **Question: How to clone a GitHub repo behind the HTTP/HTTPS proxy?**



# A New Question

- **Question: How to clone a GitHub repo behind the HTTP/HTTPS proxy?**

- **Clone via HTTPS (use username / password)**

*git clone https://github.com/<username>/<repo>.git*

- **Clone via SSH (use private / public key)**

*git clone git@github.com:<username>/<repo>.git*

***Adding a new SSH key to your GitHub account***

# A New Question

- **HTTPS option (command line):**

*git config --global http.proxy http://wwwproxy.unimelb.edu.au:8000*

*git config --global https.proxy http://wwwproxy.unimelb.edu.au:8000*

```
ubuntu@test:~$ git config --global http.proxy http://wwwproxy.unimelb.edu.au:8000
ubuntu@test:~$ git config --global https.proxy http://wwwproxy.unimelb.edu.au:8000
ubuntu@test:~$ git clone https://github.com/[REDACTED]
Cloning into '[REDACTED]'...
Username for 'https://github.com': ^C
```

- **HTTPS option (Ansible):**

[git config](#) module

# A New Question

- **SSH option (command line):**

*sudo apt install socat vim*

*vim ~/.ssh/config*

host github.com

user git

hostname ssh.github.com

port 443

proxycommand socat - PROXY:wwwproxy.unimelb.edu.au:%h:%p,proxyport=8000

- **HTTPS option (Ansible):**

[copy](#) module or [blockinfile](#) module

# Docker Demo 5

- **Rolling update:**

```
docker service update --update-parallelism 1 --update-delay 5s \  
                        --image=alwynpan/comp90024:demo1 nginx
```

- **Rollback:**

```
docker service update --rollback nginx
```



# Docker Demo 6

- Create React App

*npx create-react-app demo6*

- Run the app:

*yarn start*

 vuejs / vue

★ Star

131,456

 facebook / react

★ Star

124,845

 angular / angular

★ Star

46,287

Compiled successfully!

You can now view demo6 in the browser.

Local: http://localhost:3000

On Your Network: http://10.8.8.8:3000

Note that the development build is not optimized.  
To create a production build, use **yarn build**.

# Docker Demo 6

- **.dockerignore**

Before the docker CLI sends the context to the docker daemon, it looks for a file named `.dockerignore` in the root directory of the context. If this file exists, the CLI modifies the context to exclude files and directories that match patterns in it. This helps to avoid unnecessarily sending large or sensitive files and directories to the daemon and potentially adding them to images using ***ADD*** or ***COPY***.

# Docker Demo 6

- React UI (simple)

```
FROM node:12

EXPOSE 5000

WORKDIR /app

COPY [".", "/app"]

RUN yarn install; \
    yarn build; \
    yarn global add serve

CMD ["serve", "-s", "build"]
```

**Build:** *docker build -t react:simple .*

**Run:** *docker run --name simple -p 5000:5000 -d react:simple*

# Docker Demo 6

- **Challenge** – image size, the smaller the better
- **React UI** – reduced size

**Pre-build:** *yarn build*

```
FROM nginx:1.17-alpine

USER root

COPY ["/build/", "/usr/share/nginx/html/"]
```

**Build:** *docker build -t react:nginx -f Dockerfile.nginx .*

**Run:** *docker run --name small -p 8080:80 -d react:nginx*

# Docker Demo 6

- Node vs Nginx (image size)

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
react	nginx	776b632b5445	7 minutes ago	20.2MB
react	simple	7645519a9828	43 minutes ago	1.23GB

alwynpan@Alwyns-MBP ~/docker-demo/demo6 master ● ?

- Node vs Nginx (resource usage)

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O
6e6ac6b1159f	simple	0.00%	13.22MiB / 5.809GiB	0.22%	148kB / 5.41kB
e2ff2eefbe2b	small	0.00%	2.48MiB / 5.809GiB	0.04%	12.9kB / 148kB

# Docker Demo 6

- **Multi-stage build**

With multi-stage builds, you use multiple **FROM** statements in your Dockerfile. Each **FROM** instruction can use a different base, and each of them begins a new stage of the build. You can selectively copy artifacts from one stage to another, leaving behind everything you don't want in the final image.

Reference: <https://docs.docker.com/develop/develop-images/multistage-build/#use-multi-stage-builds>

# Docker Demo 6

- React UI (multi-stage)

```
FROM node:12 AS build

EXPOSE 5000

WORKDIR /app

COPY [".", "/app"]

RUN yarn install; \
    yarn build; \
    yarn global add serve

CMD ["serve", "-s", "build"]

FROM nginx:1.17-alpine

USER root

COPY --from=build ["/app/build/", "/usr/share/nginx/html/"]
```

**Build:** *docker build -t react:multi -f Dockerfile.multi-stage .*

**Run:** *docker run --name multi -p 8081:80 -d react:multi*

# Docker Demo 6

- **React UI (multi-stage) –Build only one stage**

**Build:** *docker build --target build -t react:build -f Dockerfile.multi-stage .*

**Run:** *docker run --name build -p 5001:5000 -d react:build*