

Build Custom Image

Rupesh Panwar

Date:04-Aug-21

Agenda

Creating docker image

Building a Dockerfile

Dockerfile tear-down

Build process in detail

Rebuild with cache

Tagging an image

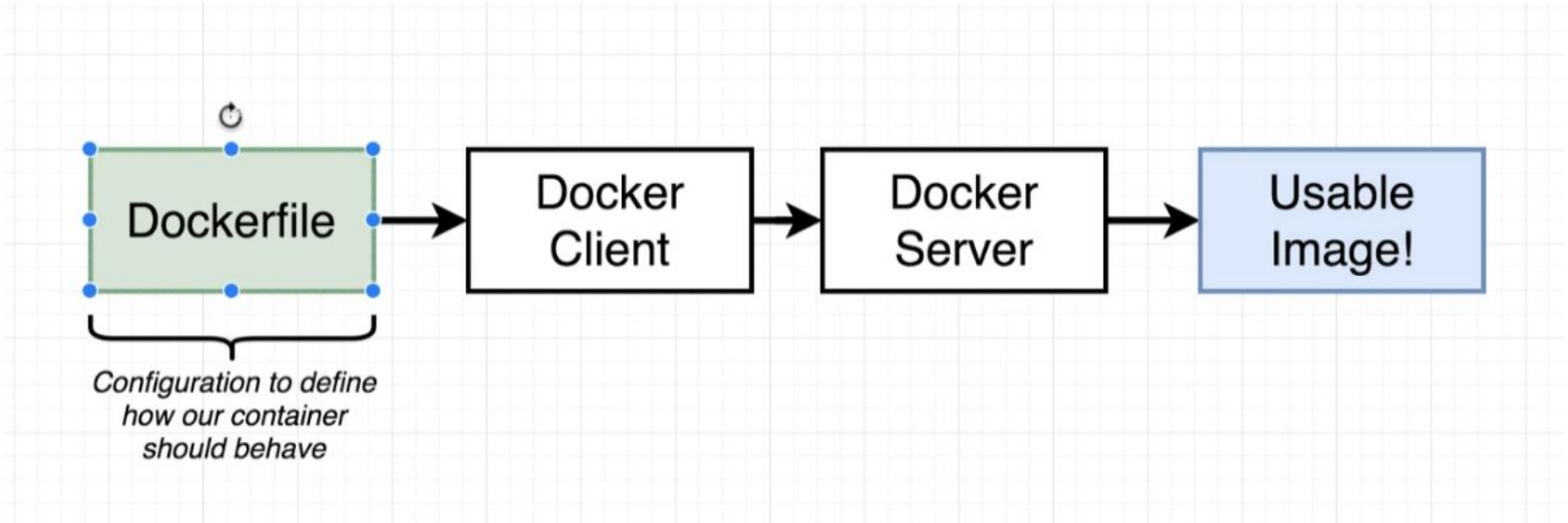
Creating docker image

Hello-World Image	
FS Snapshot	Startup Command
hello-world	> Run hello-world

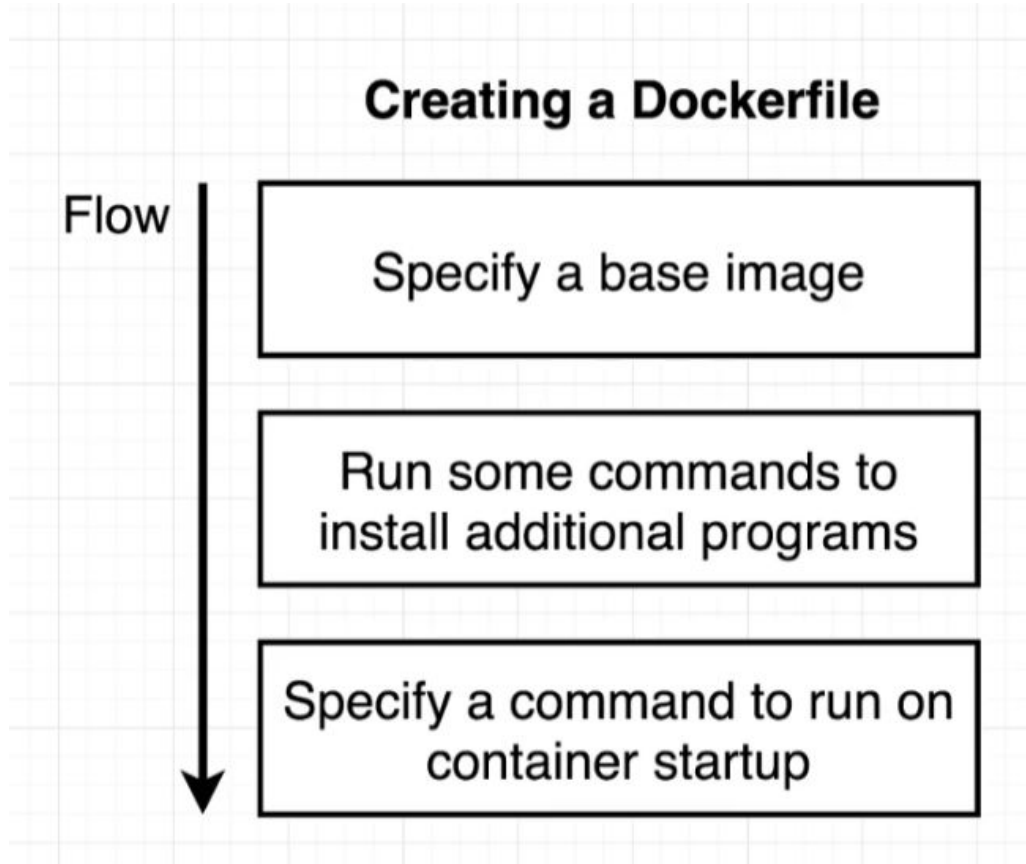
Redis Image	
FS Snapshot	Startup Command
redis stuff	> Run redis

Busybox Image	
FS Snapshot	Startup Command
busybox stuff	???

Creating docker image



Creating docker image



Building a Dockerfile

Goal > Create an image that runs redis-server

```
Docker > Day3 > redis-image >  Dockerfile > ...
```

```
1  # Use an existing docker image as a base image
2  FROM alpine
3
4  # Download and install dependencies
5  RUN apk add --update redis
6
7  # Tell the image what to do when it starts as a container
8  CMD ["redis-server"]
```

> **docker build .**

> **docker run image-build-id**

Dockerfile tear-down

*Instruction telling
Docker Server
what to do*

*Argument to the
instruction*

FROM	alpine
RUN	apk add --update redis
CMD	["redis-server"]

Dockerfile tear-down

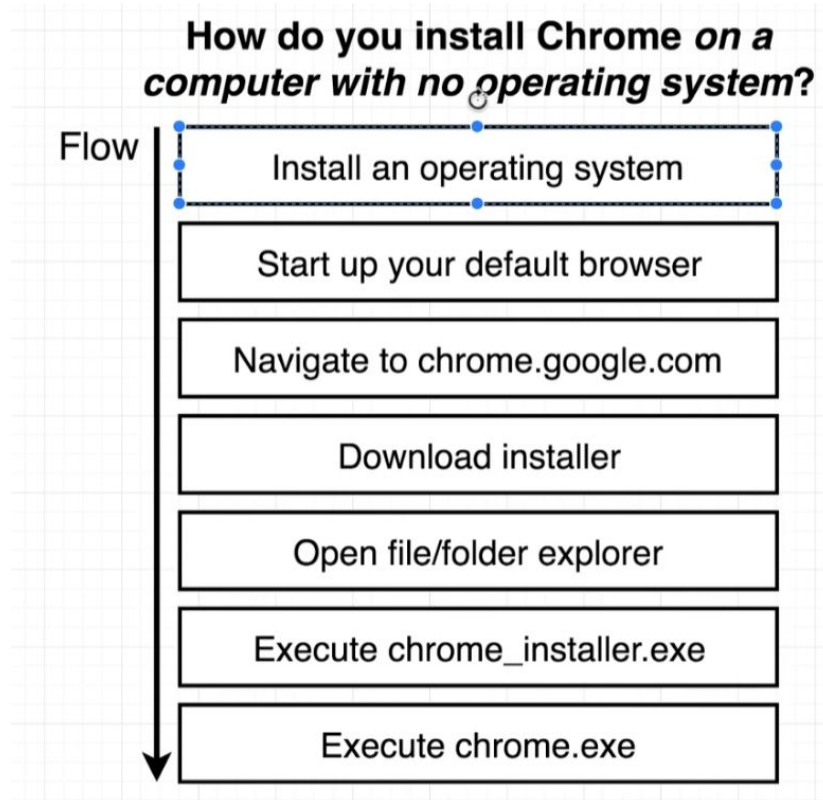
Analogy

Writing a dockerfile

=

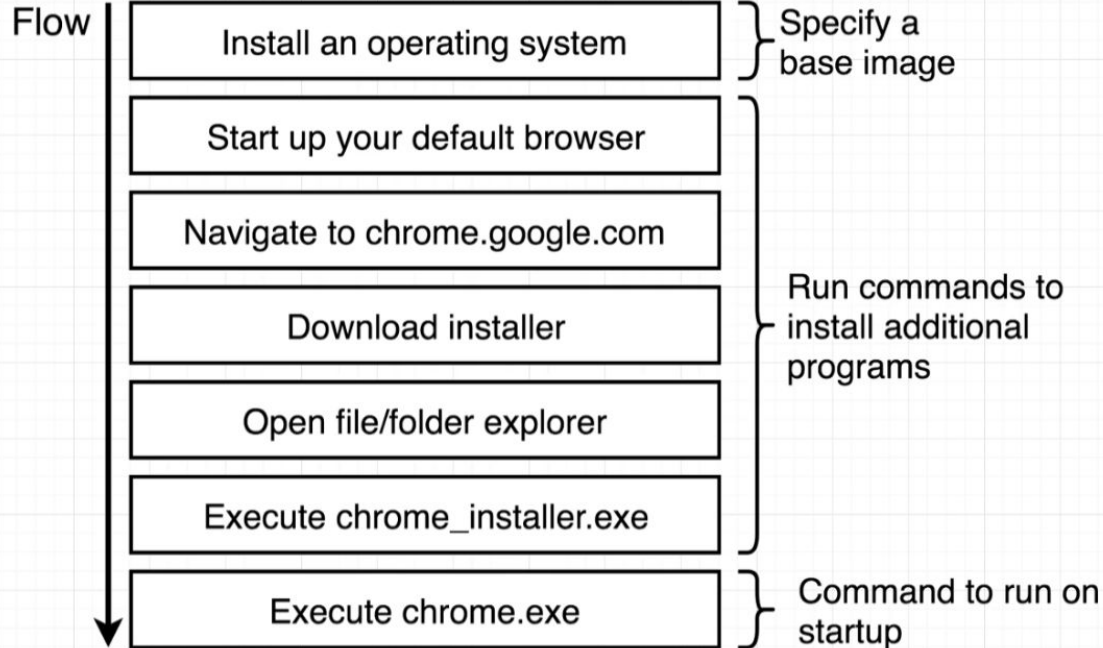
Being given a computer with no OS
and being told to install Chrome

Dockerfile tear-down



Dockerfile tear-down

How do you install Chrome *on a computer with no operating system?*



Dockerfile tear-down

Why did we use alpine as a base image?

```
graph TD; A[Why did we use alpine as a base image?] --> B[Why do you use Windows, MacOS, or Ubuntu?]; B --> C[They come with a preinstalled set of programs that are useful to you!];
```

Why do you use Windows,
MacOS, or Ubuntu?

They come with a preinstalled set of
programs that are useful to you!

Build process in detail

```
redis-image $ docker build .
```

```
Sending build context to Docker daemon 2.048kB
```

```
Step 1/3 : FROM alpine
```

```
latest: Pulling from library/alpine
```

```
5843afab3874: Pull complete
```

```
Digest: sha256:adab3844f497ab9171f070d4cae4114b5aec565ac772e2f2579405b78be67c96
```

```
Status: Downloaded newer image for alpine:latest
```

```
---> d4ff818577bc
```

```
Step 2/3 : RUN apk add --update redis
```

```
---> Running in fb7d0da1d63b
```

```
fetch https://dl-cdn.alpinelinux.org/alpine/v3.14/main/x86_64/APKINDEX.tar.gz
```

```
fetch https://dl-cdn.alpinelinux.org/alpine/v3.14/community/x86_64/APKINDEX.tar.gz
```

```
(1/1) Installing redis (6.2.5-r0)
```

```
Executing redis-6.2.5-r0.pre-install
```

```
Executing redis-6.2.5-r0.post-install
```

```
Executing busybox-1.33.1-r2.trigger
```

```
OK: 8 MiB in 15 packages
```

```
Removing intermediate container fb7d0da1d63b
```

```
---> 5e839fc84936
```

Build process in detail

```
Step 3/3 : CMD ["redis-server"]
```

```
---> Running in acdd44ec11a0
```

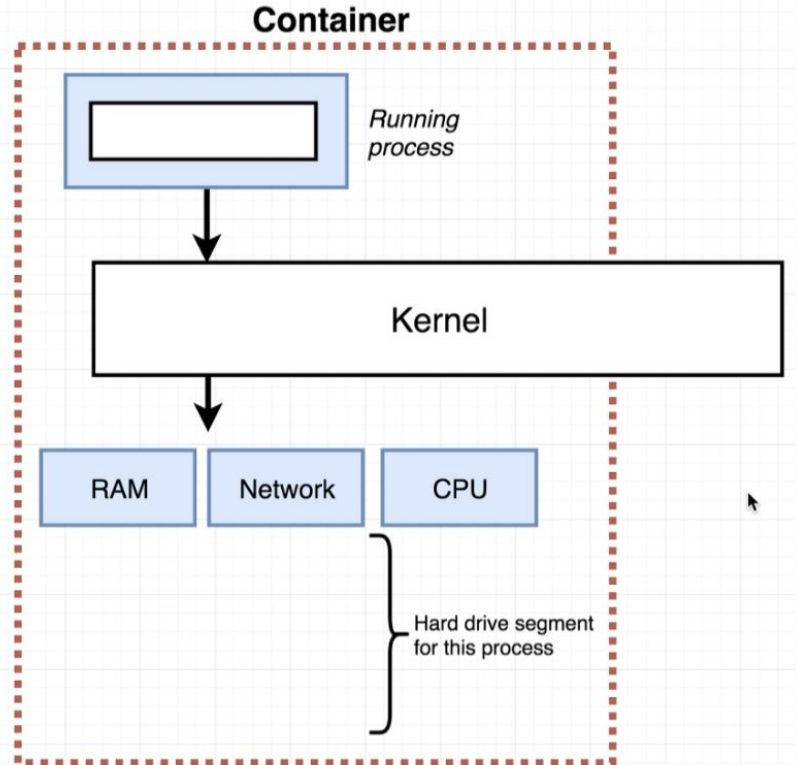
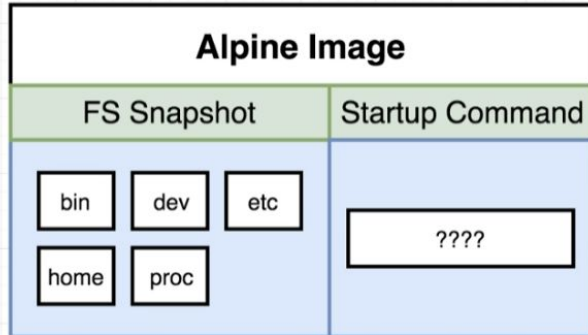
```
Removing intermediate container acdd44ec11a0
```

```
---> fe854ba62c2c
```

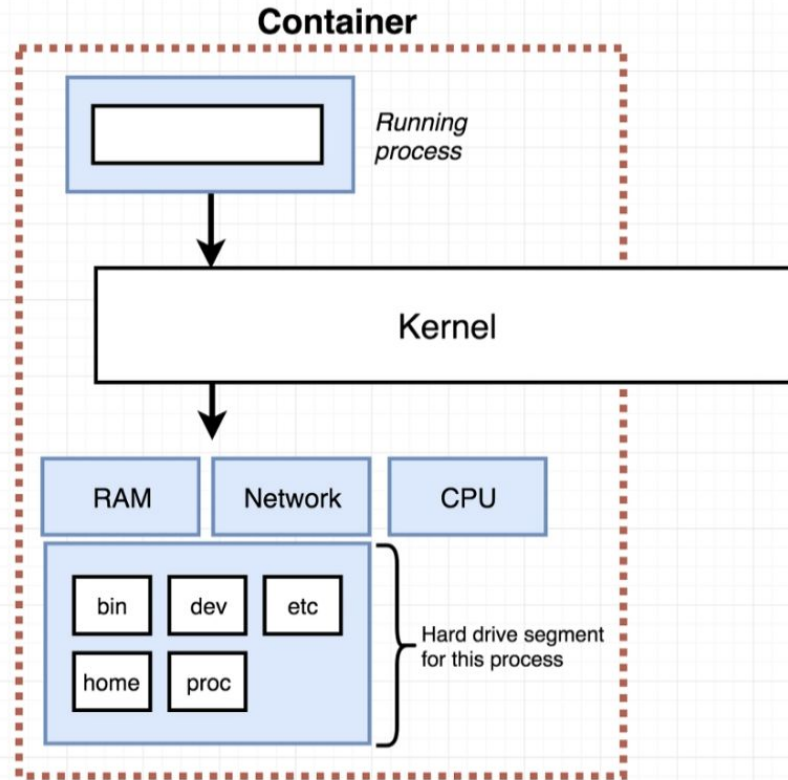
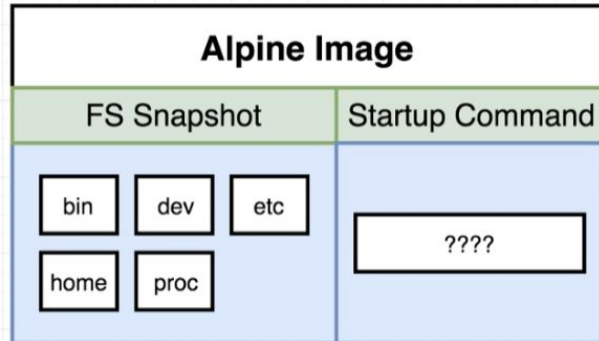
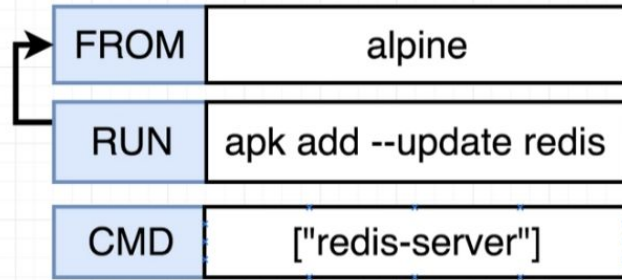
```
Successfully built fe854ba62c2c
```

Build process in detail

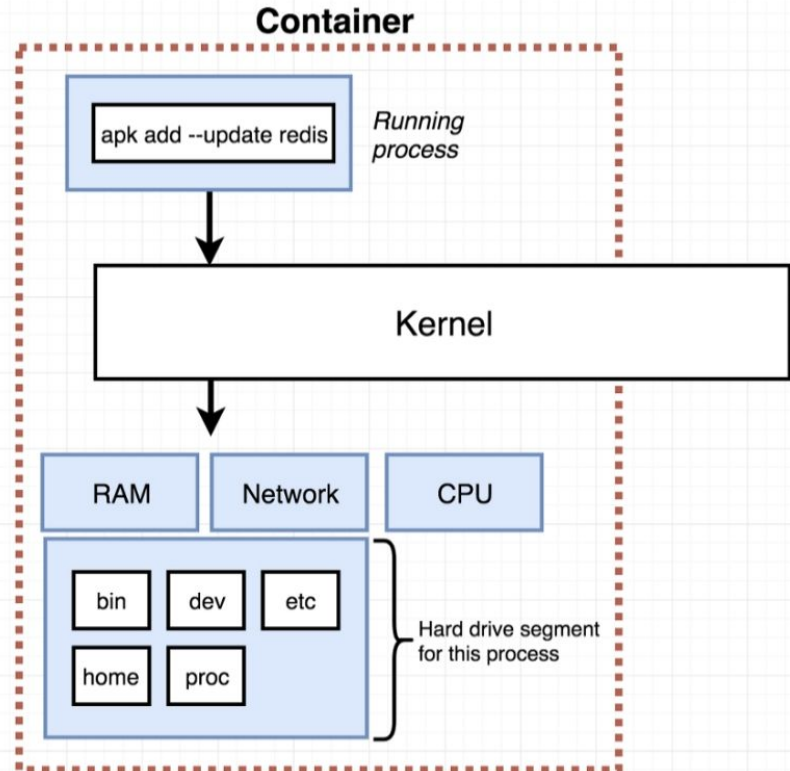
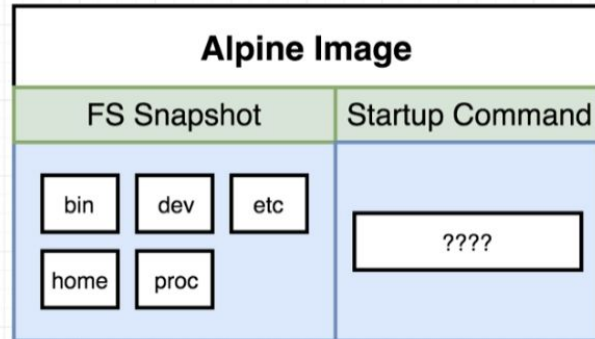
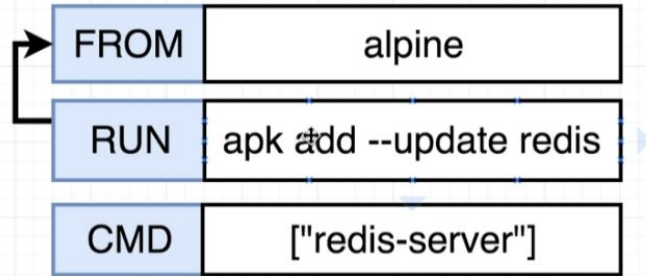
FROM	alpine
RUN	apk add --update redis
CMD	["redis-server"]



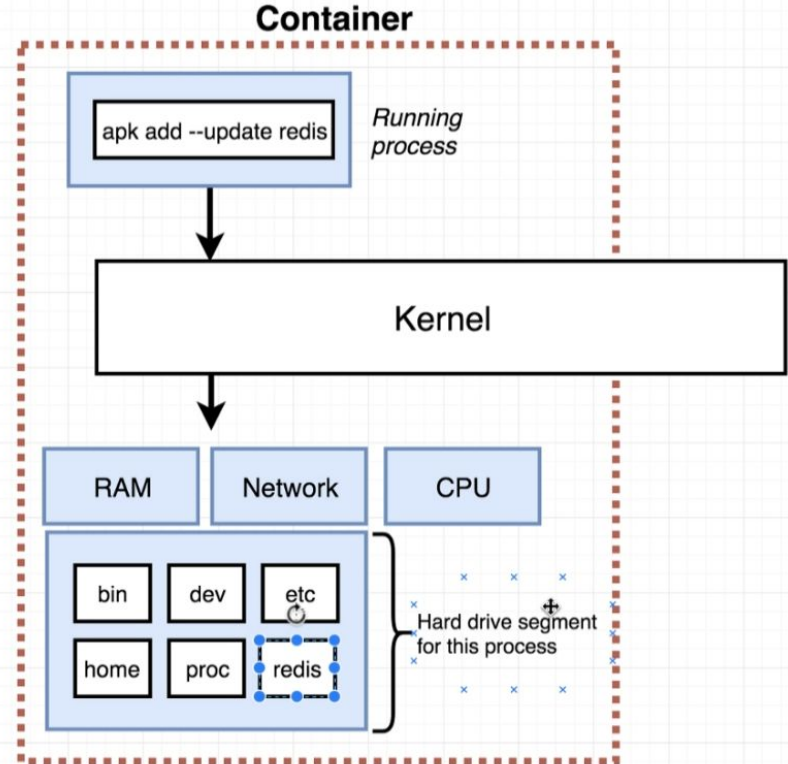
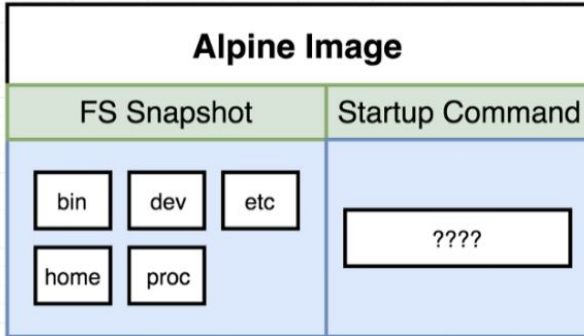
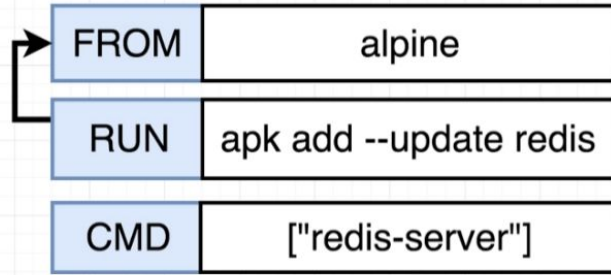
Build process in detail



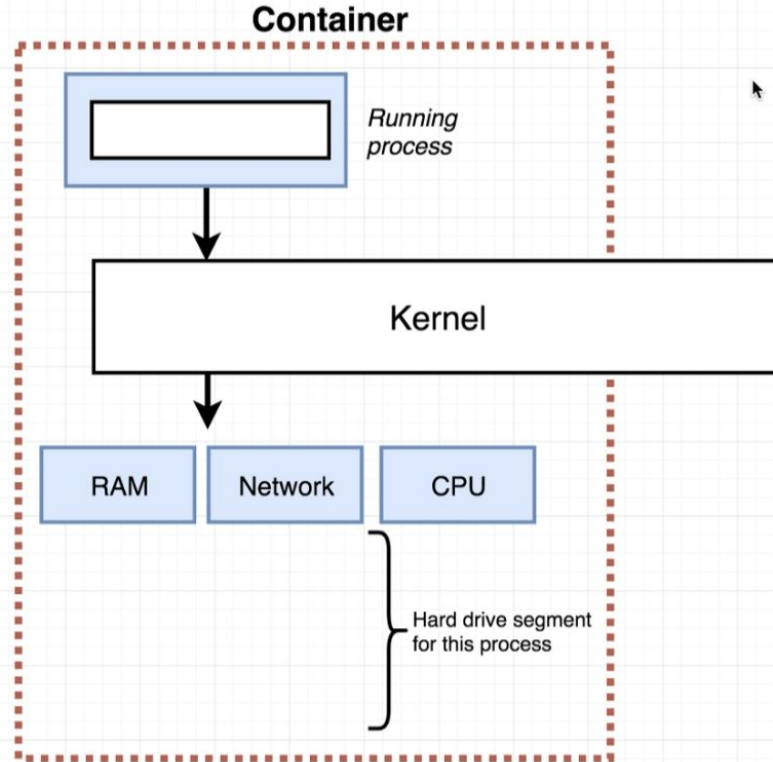
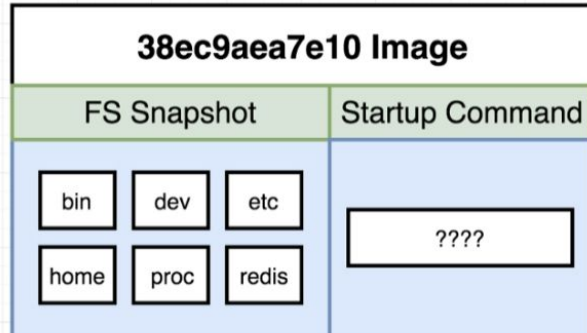
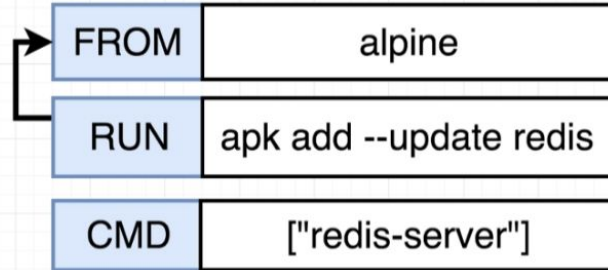
Build process in detail



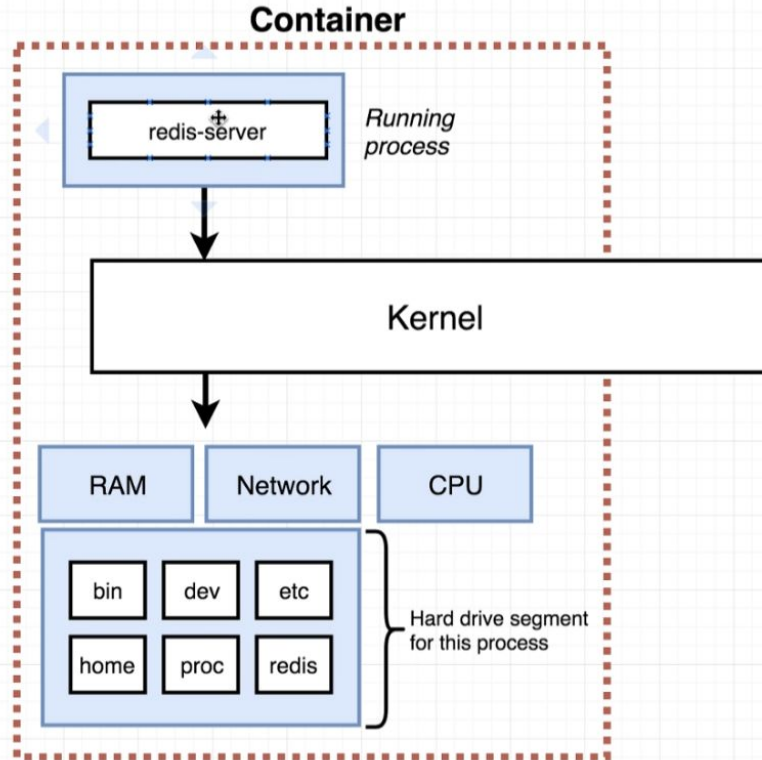
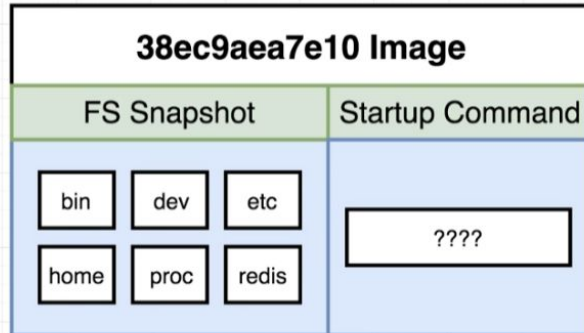
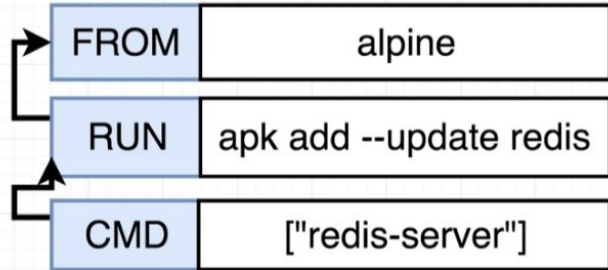
Build process in detail



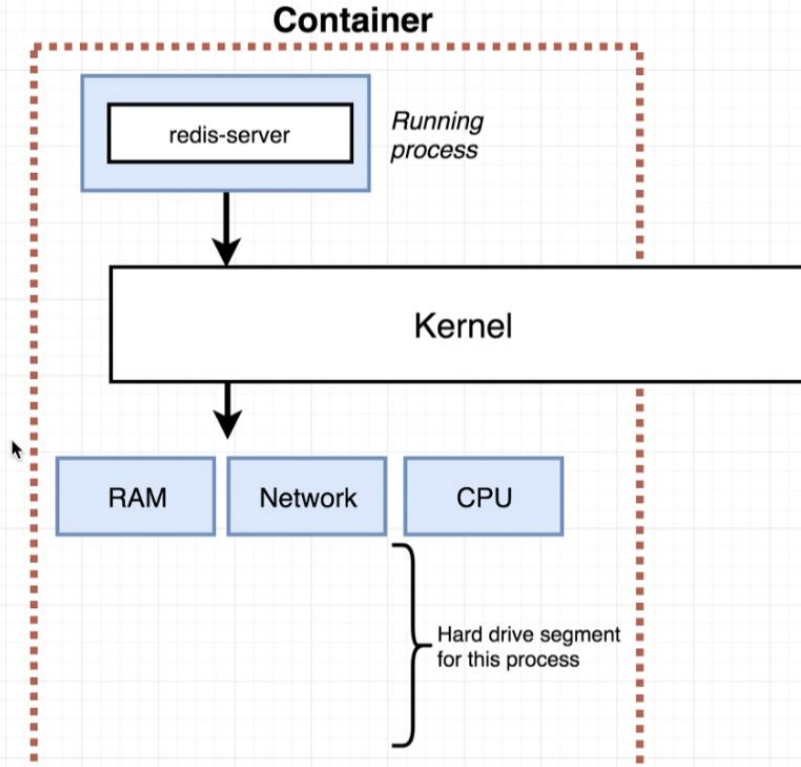
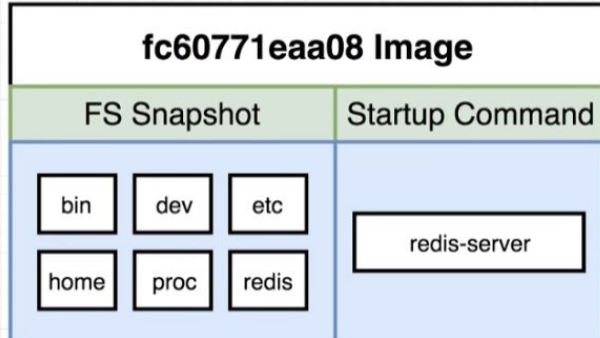
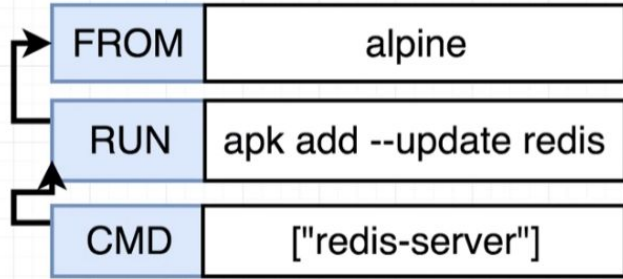
Build process in detail



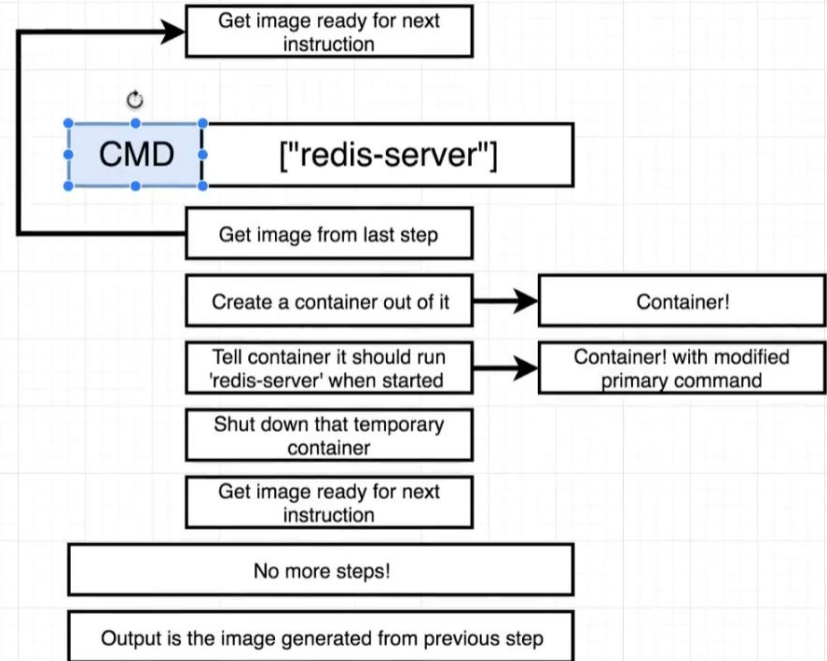
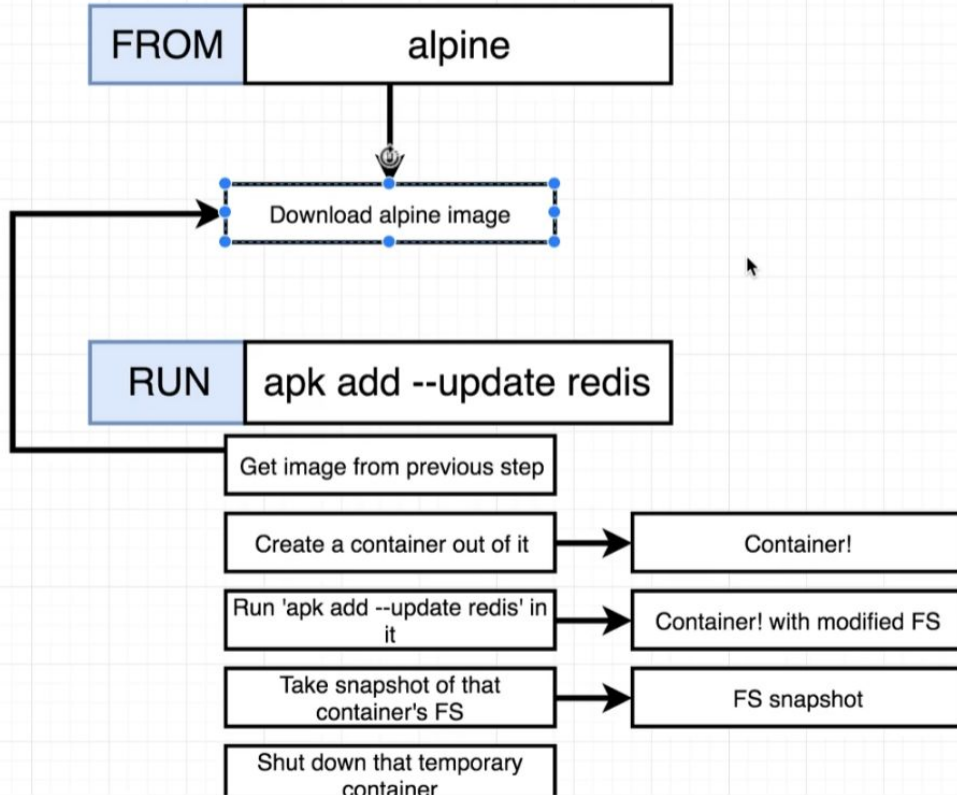
Build process in detail



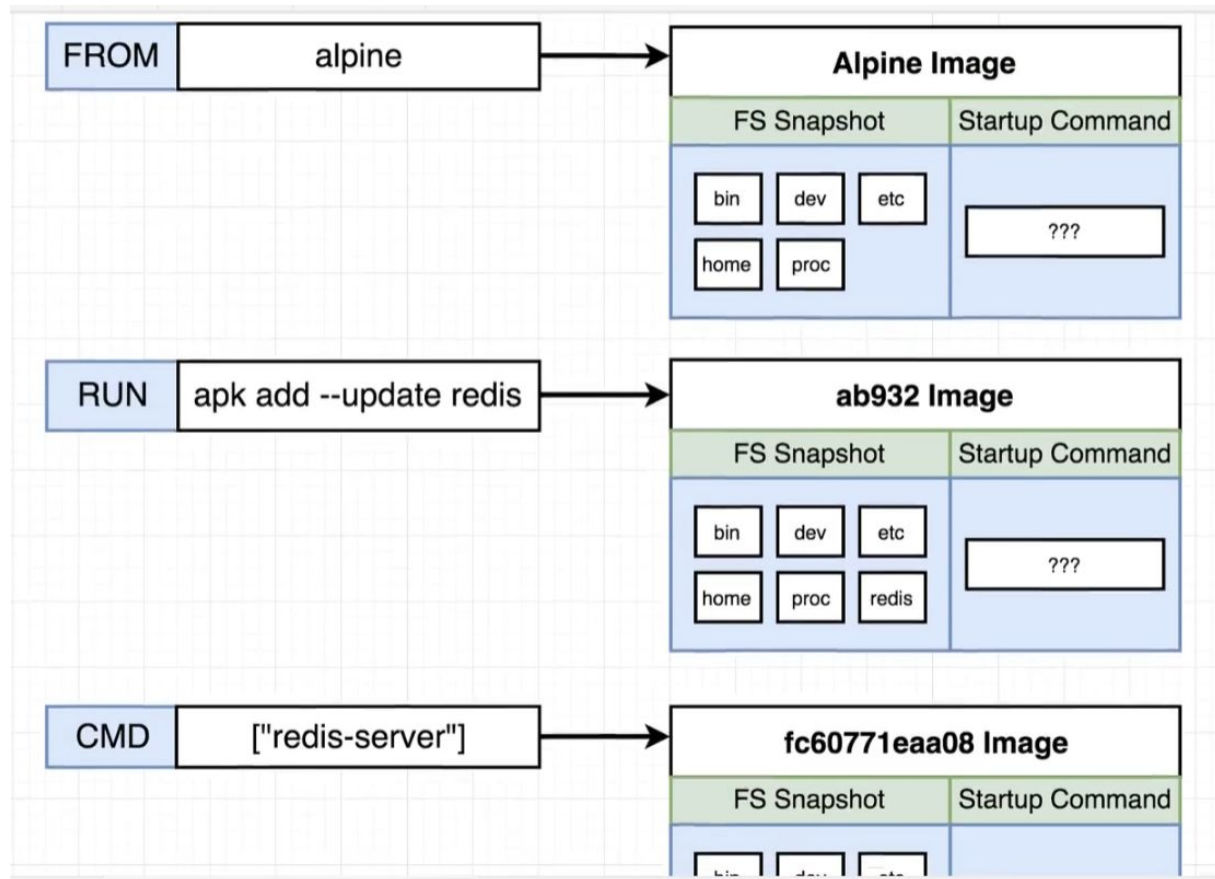
Build process in detail



Build process in detail



Rebuild with cache



Rebuild with cache

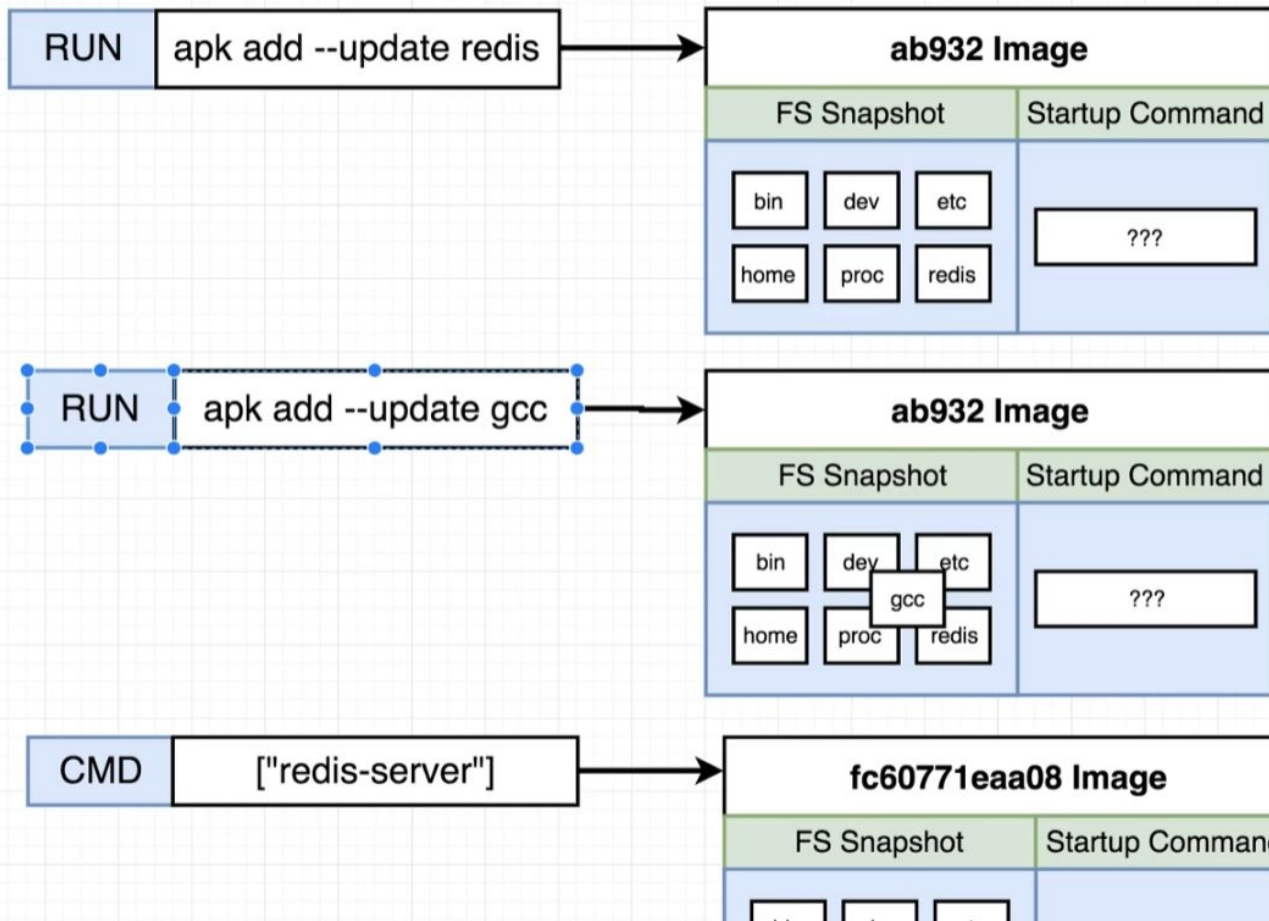
```
Docker > Day3 > redis-image > 🚢 Dockerfile.gcc > ...
```

```
1  # Use an existing docker image as a base image
2  FROM alpine
3
4  # Download and install dependencies
5  RUN apk add --update redis
6  RUN apk --update gcc
7
8  # Tell the image what to do when it starts as a container
9  CMD ["redis-server"]
```


Rebuild with cache

```
redis-image $ docker build -f Dockerfile.gcc .
Sending build context to Docker daemon 3.072kB
Step 1/4 : FROM alpine
----> d4ff818577bc
Step 2/4 : RUN apk add --update redis
----> Using cache
----> 5e839fc84936
Step 3/4 : RUN apk add --update gcc
----> Running in f9a0265fa75c
fetch https://dl-cdn.alpinelinux.org/alpine/v3.14/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.14/community/x86_64/APKINDEX.tar.gz
(1/11) Installing libgcc (10.3.1_git20210424-r2)
(2/11) Installing libstdc++ (10.3.1_git20210424-r2)
(3/11) Installing binutils (2.35.2-r2)
(4/11) Installing libgomp (10.3.1_git20210424-r2)
(5/11) Installing libatomic (10.3.1_git20210424-r2)
(6/11) Installing libgphobos (10.3.1_git20210424-r2)
(7/11) Installing gmp (6.2.1-r0)
(8/11) Installing isl22 (0.22-r0)
(9/11) Installing mpfr4 (4.1.0-r0)
(10/11) Installing mpc1 (1.2.1-r0)
(11/11) Installing gcc (10.3.1_git20210424-r2)
Executing busybox-1.33.1-r2.trigger
OK: 118 MiB in 26 packages
Removing intermediate container f9a0265fa75c
----> a09c9f79cd53
Step 4/4 : CMD ["redis-server"]
----> Running in 552020a55542
Removing intermediate container 552020a55542
----> f694943a5596
Successfully built f694943a5596
```


Rebuild with cache




Rebuild with cache

```
redis-image $ docker build .  
Sending build context to Docker daemon 3.072kB  
Step 1/3 : FROM alpine  
----> d4ff818577bc  
Step 2/3 : RUN apk add --update redis  
----> Using cache  
----> 5e839fc84936  
Step 3/3 : CMD ["redis-server"]  
----> Using cache  
----> fe854ba62c2c  
Successfully built fe854ba62c2c
```

Tagging an image


> **docker build .**

> **docker run image-build-id**



Tags the
image

docker	build	-t rupeshpanwar/redis:latest	.
---------------	--------------	-------------------------------------	----------

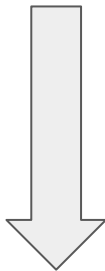


Specifies the directory of
files/folders to use for the
build

Tagging an image

```
> docker build -t rupeshpanwar/redis  
> docker run rupeshpanwar/redis
```

-t rupeshpanwar/redis:latest



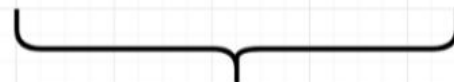
rupeshpanwar	/	redis	:	latest
---------------------	----------	--------------	----------	---------------



Your Docker ID



Repo/Project name



Version