Docker

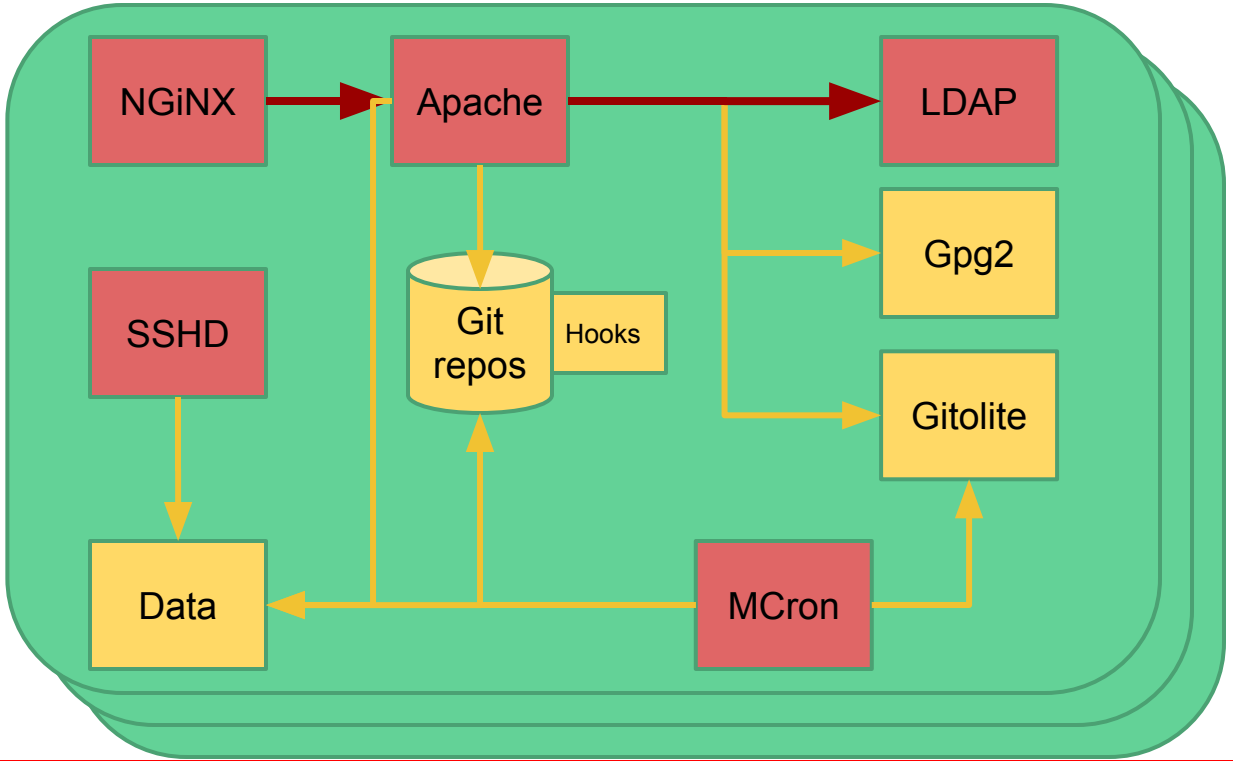
Architecture Rework Case Study

Project

From:

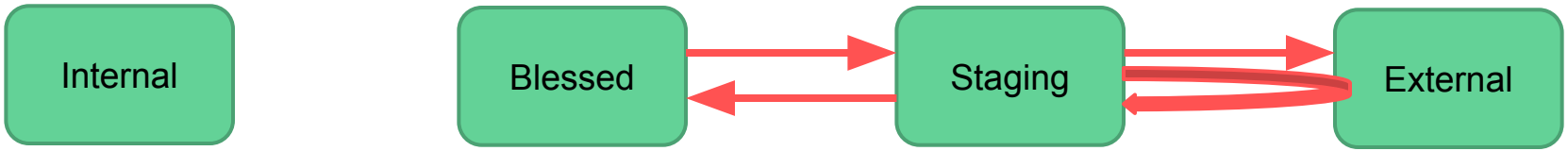
Project

To:


NGiNX
Apache
LDAP
Gpg2
SSHD
Git repos
Hooks
Gitolite
Data
MCron

What are the issues?

Example: Replicated Git repos hosting servers



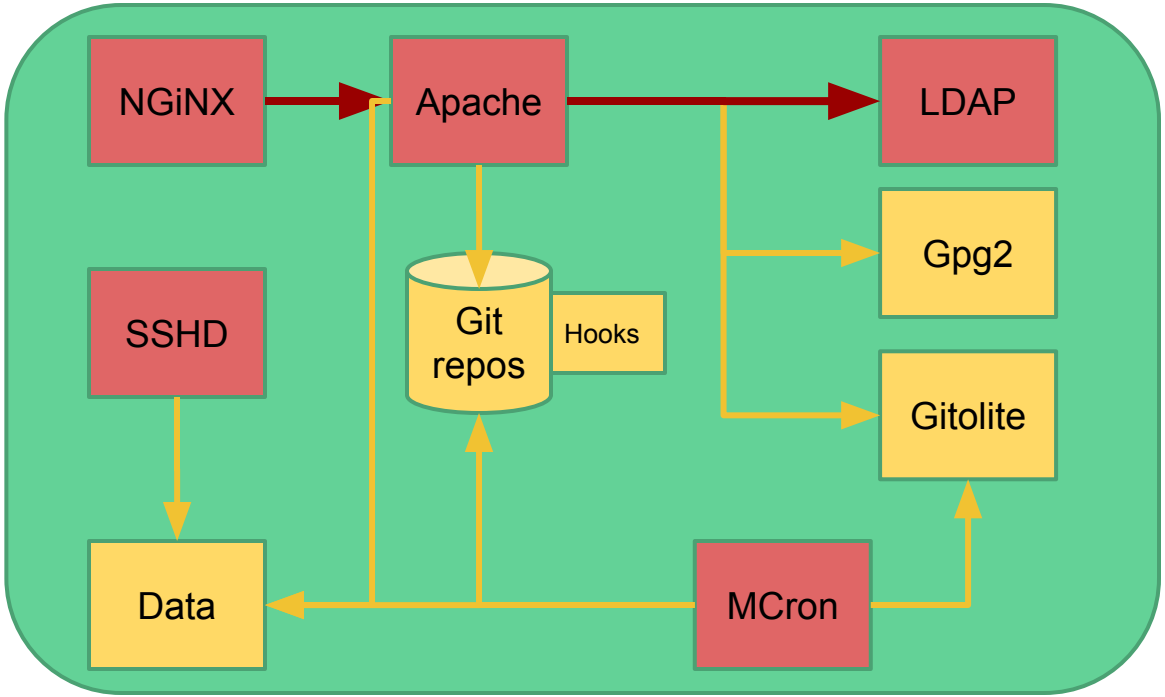
- Isolation (server)
- Configuration
- Reproducibility

What are the issues?

Example: Monolithic service => Isolation (processes) issue

Blessed / Staging / External / Internal

=

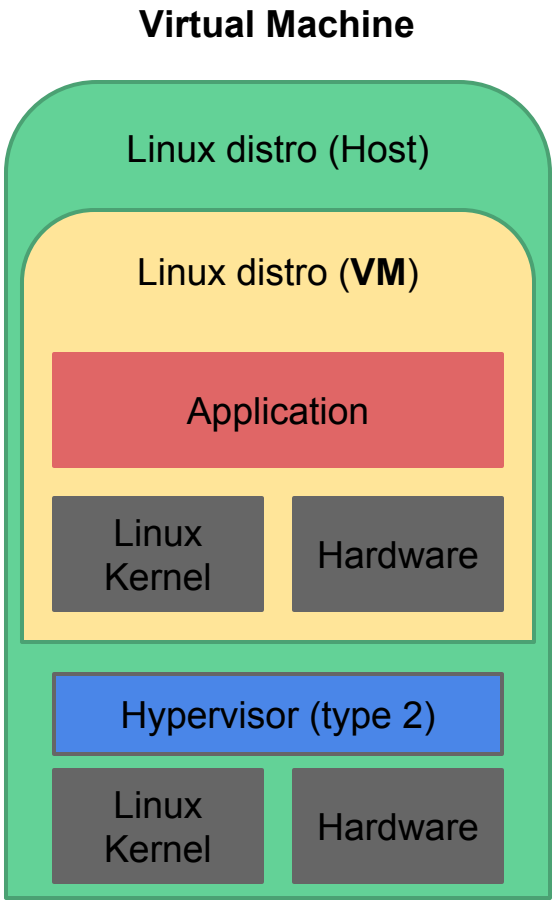


What is Docker

Lightweight virtualization

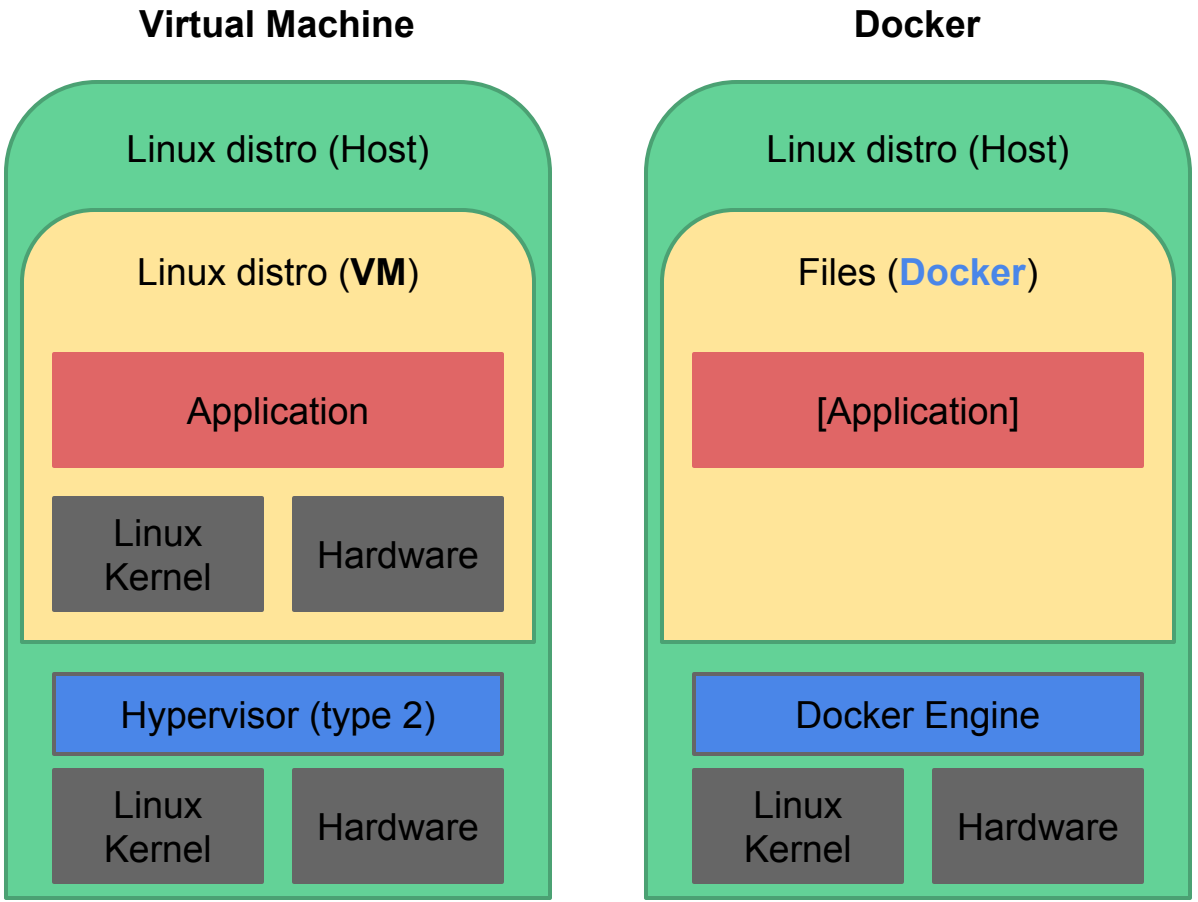
What is Docker

Lightweight virtualization



What is Docker

Lightweight virtualization

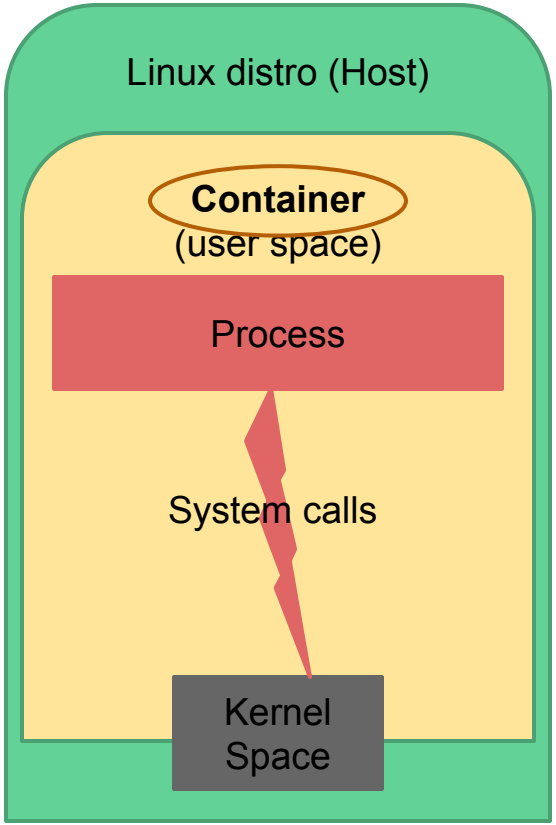


What is Docker?

Containers

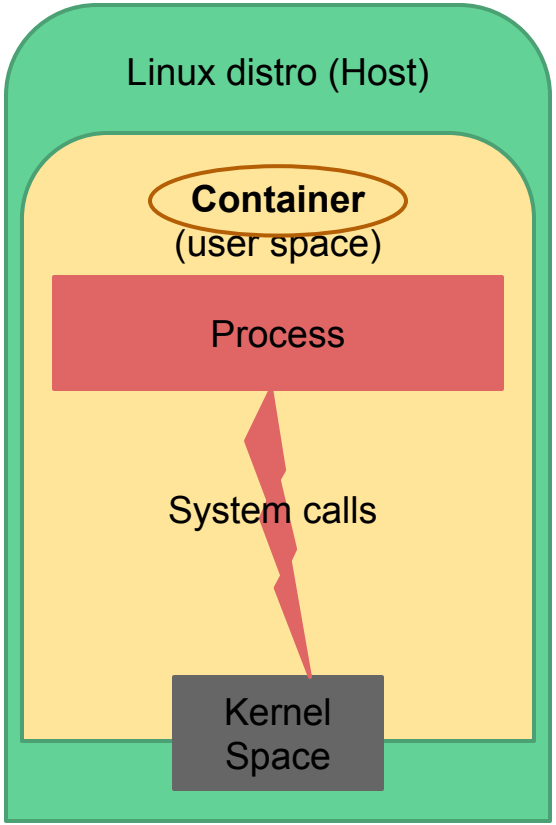
What is Docker?

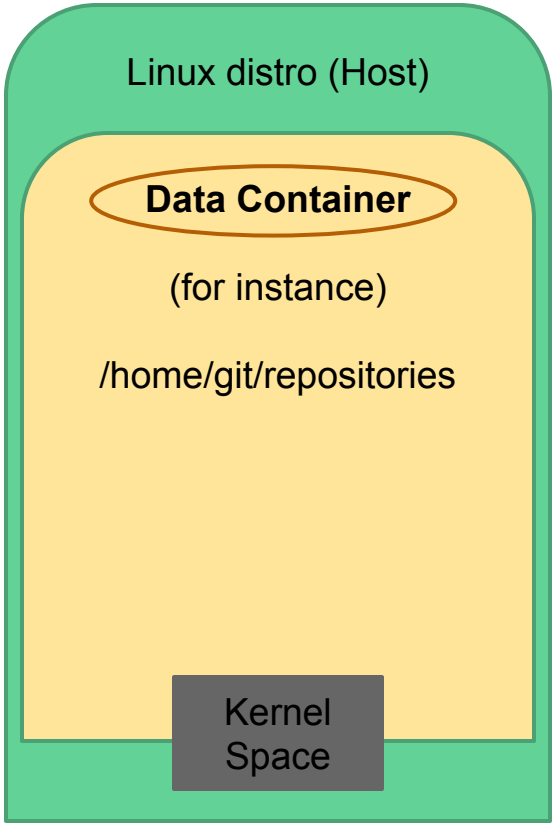
Containers



What is Docker?

Containers



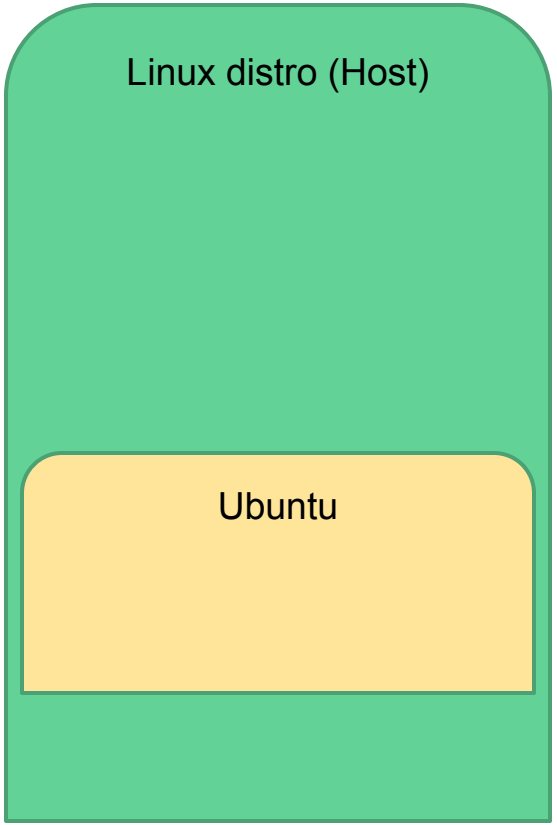


What is Docker?

Images: docker build

Dockerfile:

FROM ubuntu

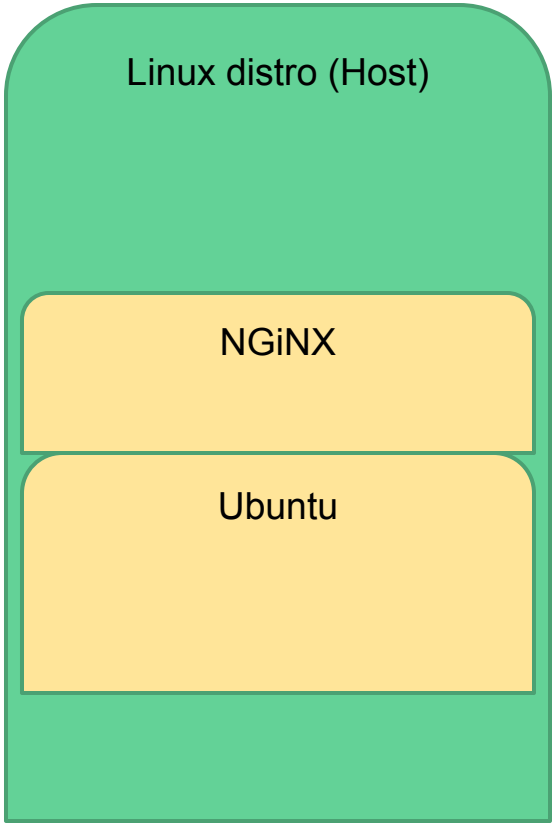


What is Docker?

Images: docker build

Dockerfile:

```
FROM ubuntu

RUN \
  add-apt-repository -y ppa:nginx/stable && \
  apt-get update && \
  apt-get install -y nginx
```

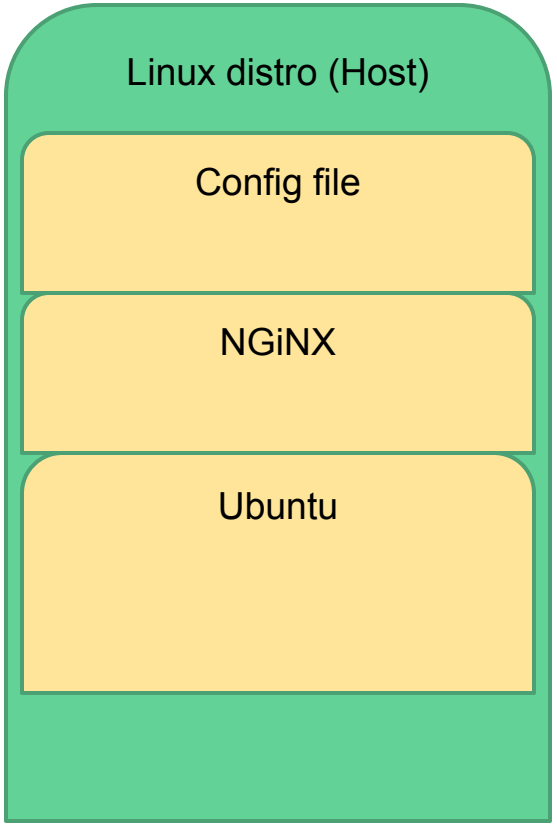


What is Docker?

Images: docker build

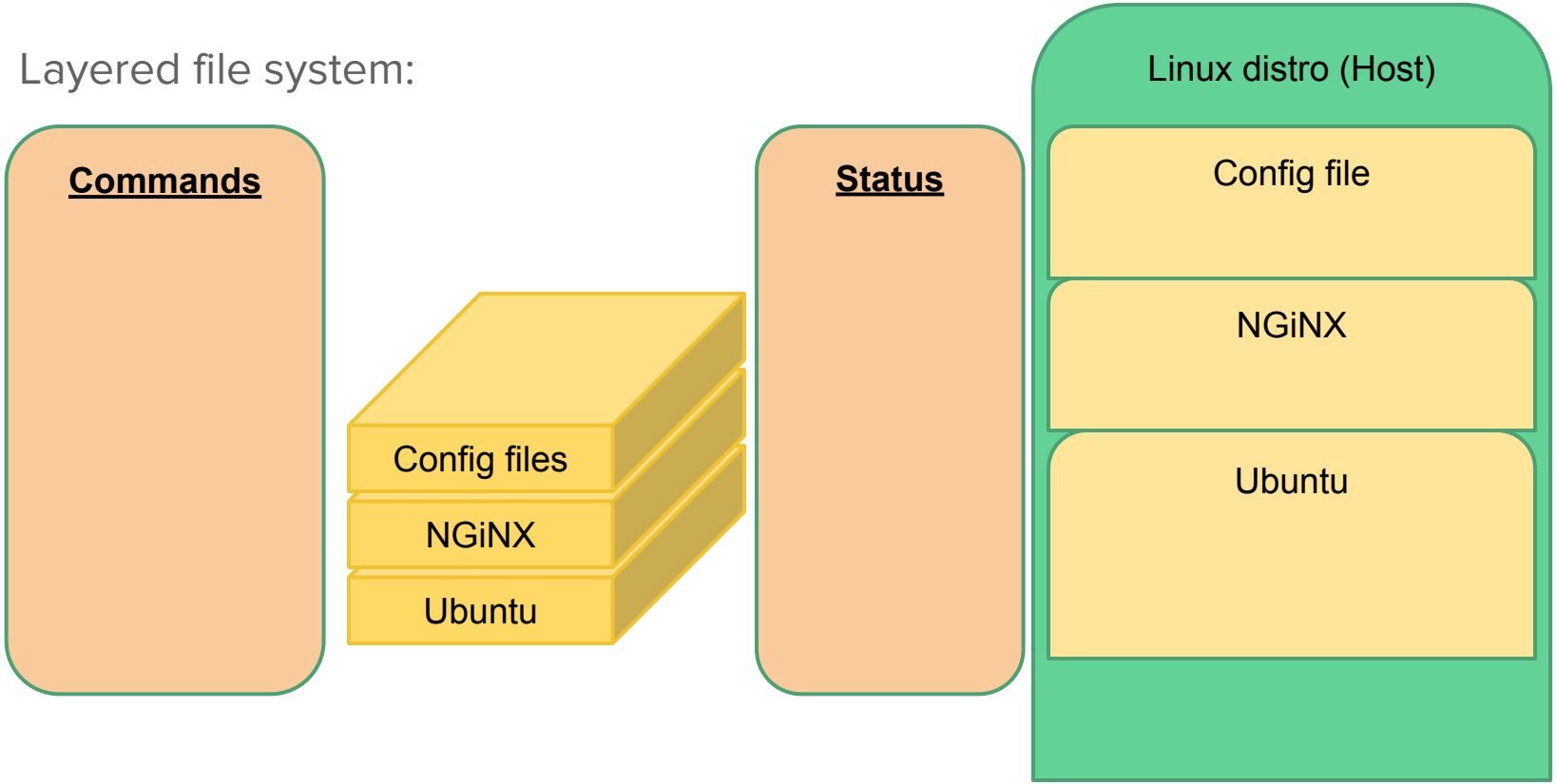
Dockerfile:

```
FROM ubuntu

RUN \
  add-apt-repository -y ppa:nginx/stable && \
  apt-get update && \
  apt-get install -y nginx

COPY env.conf /home/git/
RUN ln -fs /home/git/nginx/cnf /etc/nginx/nginx.conf

EXPOSE 80
EXPOSE 443

ENTRYPOINT ["/bin/sh", "-c"]
CMD ["nginx"]
```



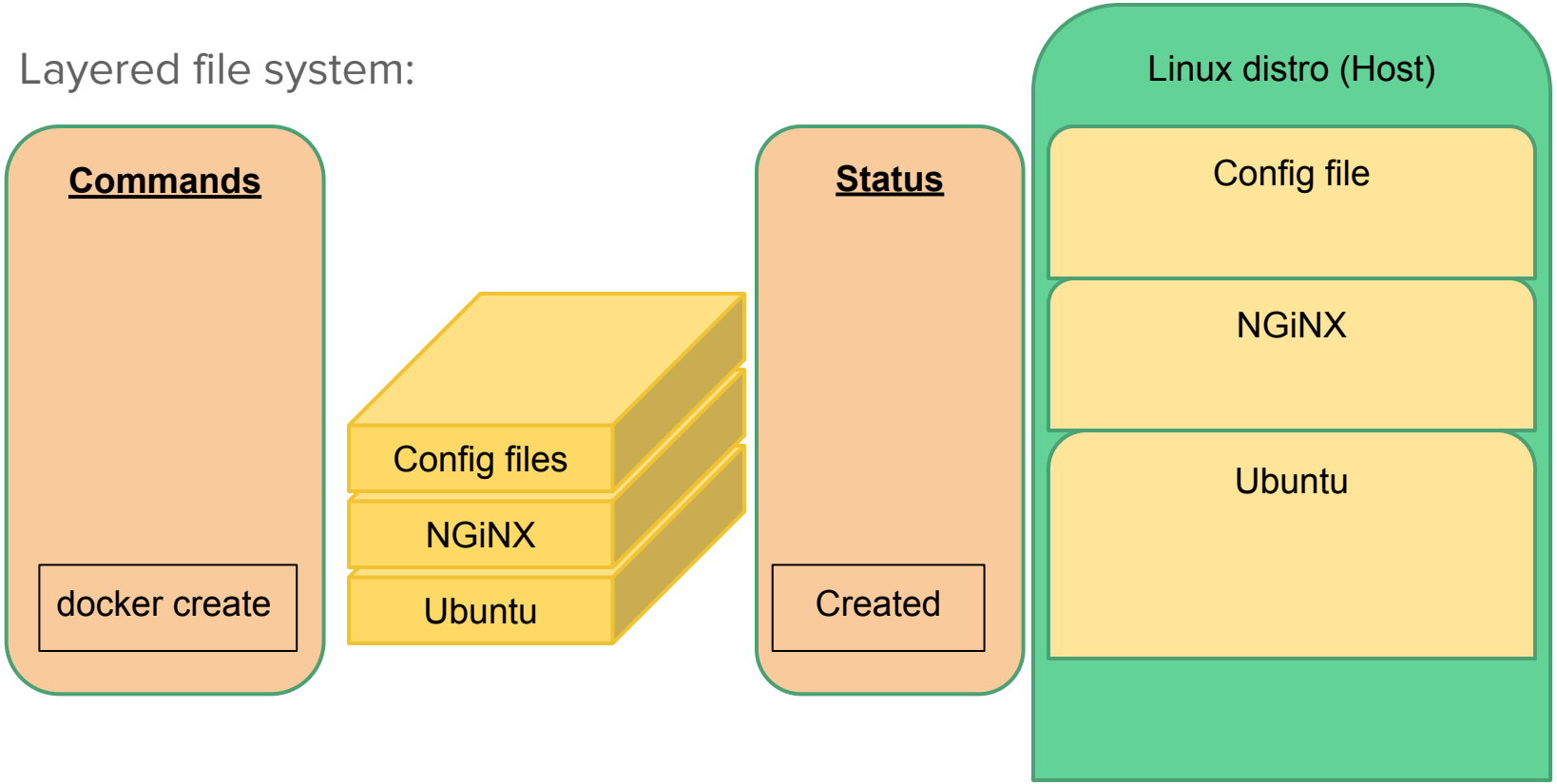
What is Docker?

Layered file system:



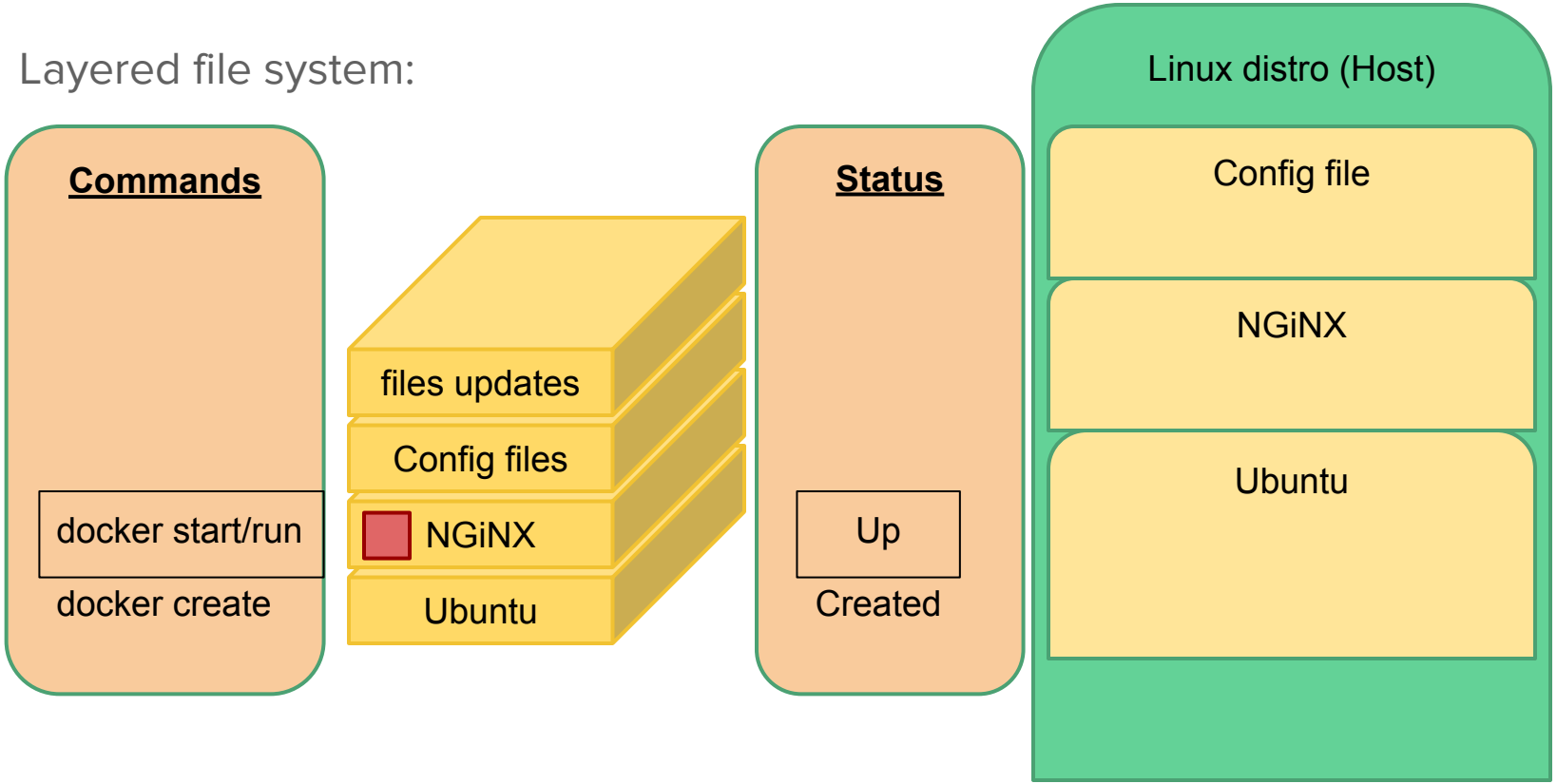
What is Docker?

Layered file system:



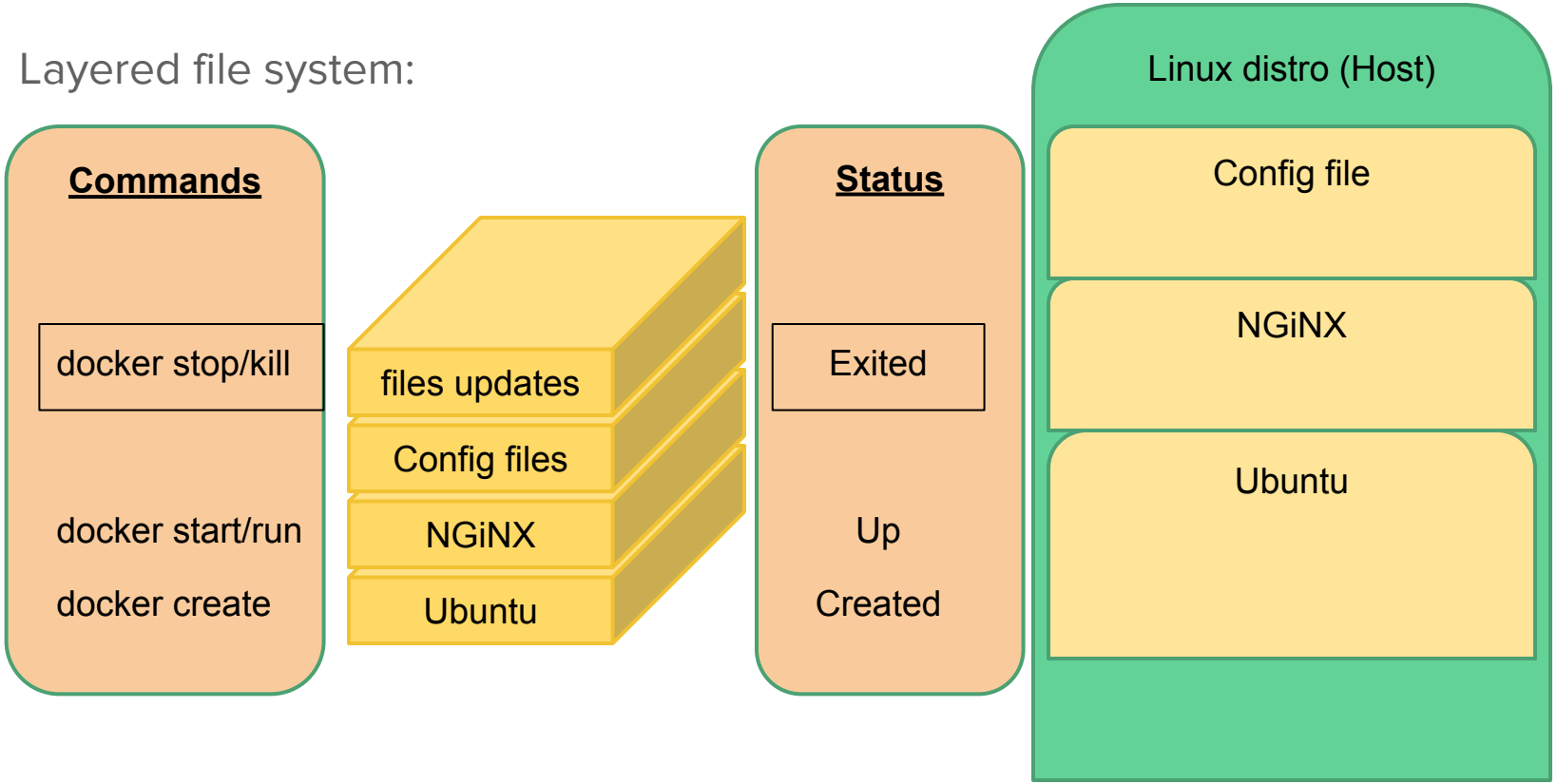
What is Docker?

Layered file system:



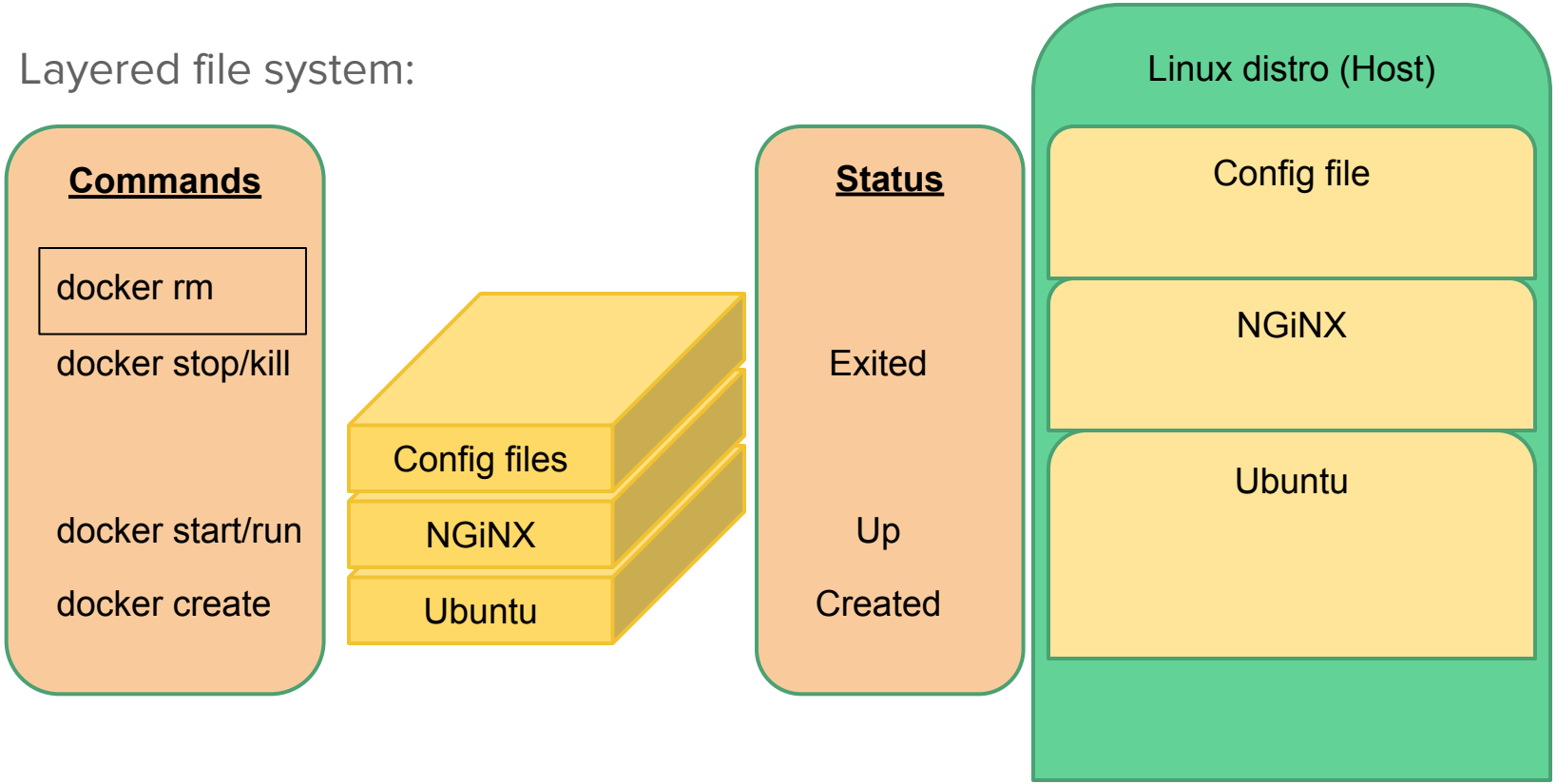
What is Docker?

Layered file system:



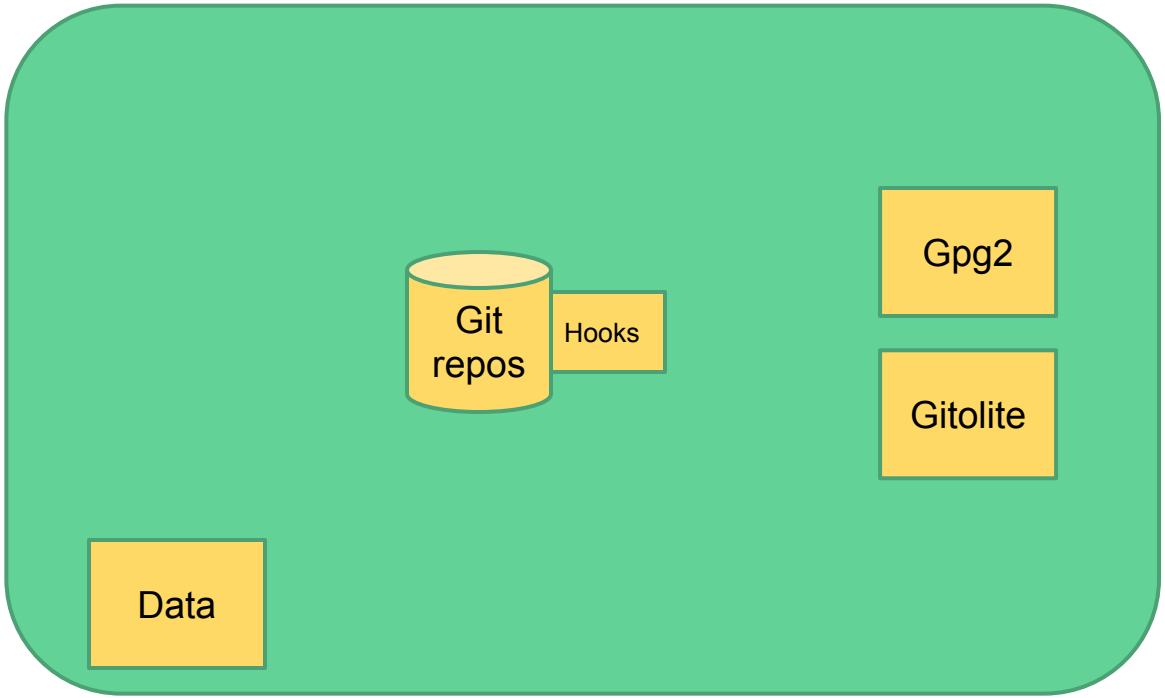
What is Docker?

Layered file system:



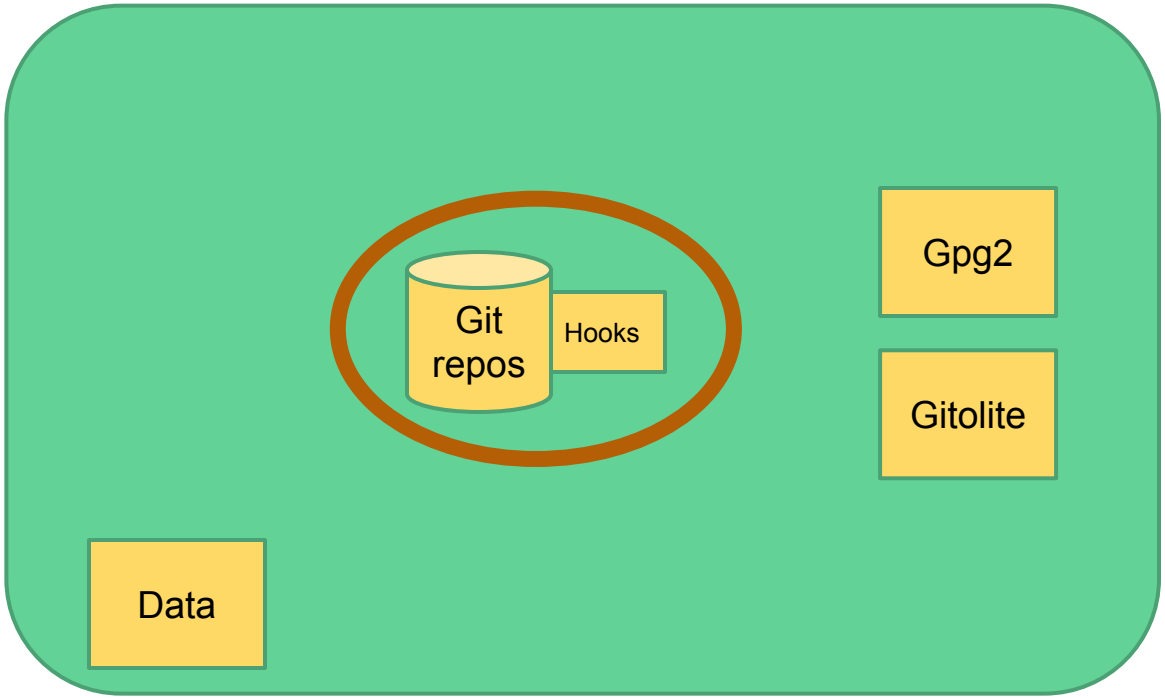
Docker rework

Data Container first



Docker rework

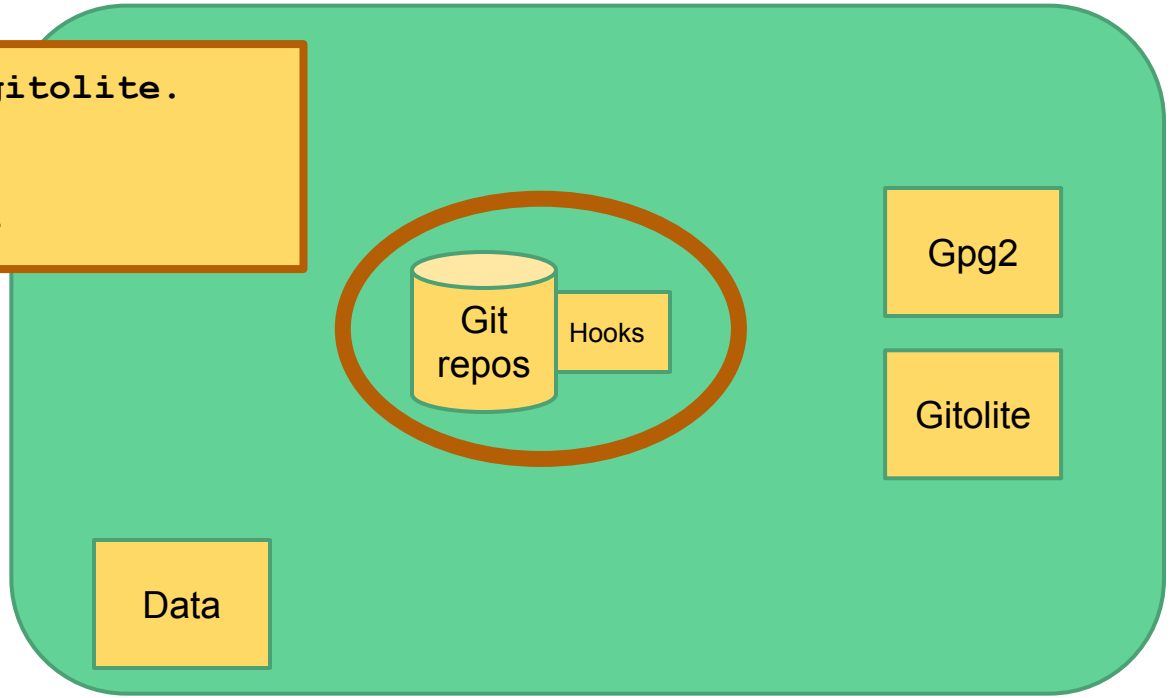
Data Container first



Docker rework

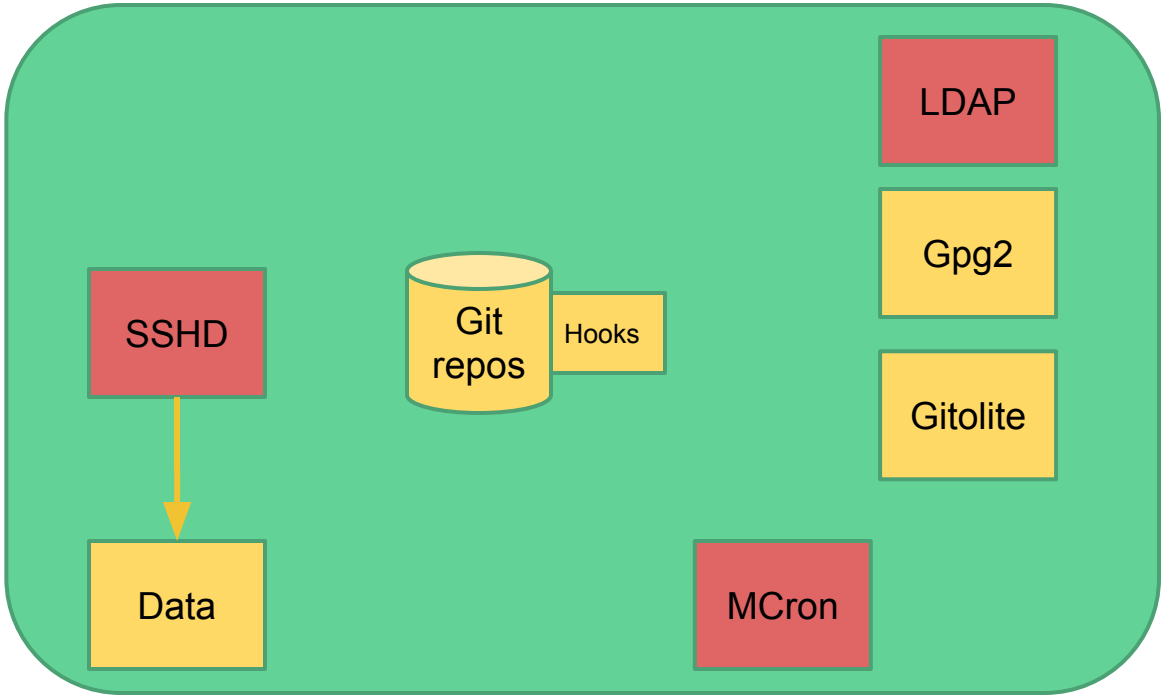
Data Container first

```
docker create --name=gitolite.
repos.blessed.cont

/home/git/repositories
```



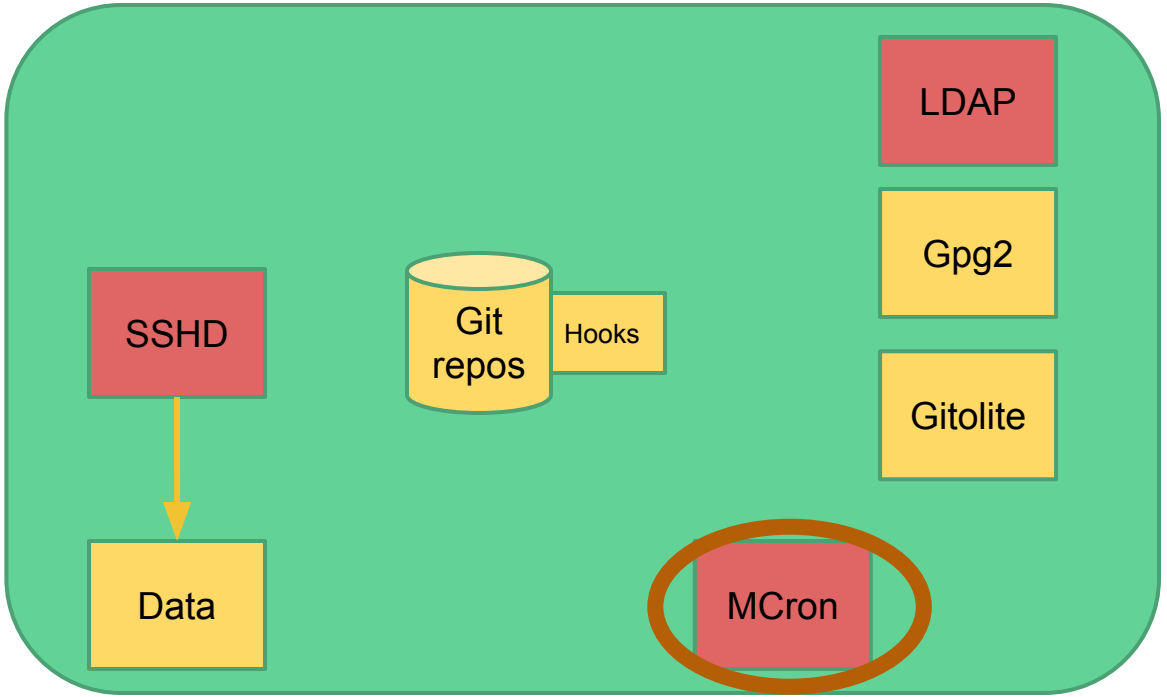
Docker rework

Leaf containers second



Docker rework

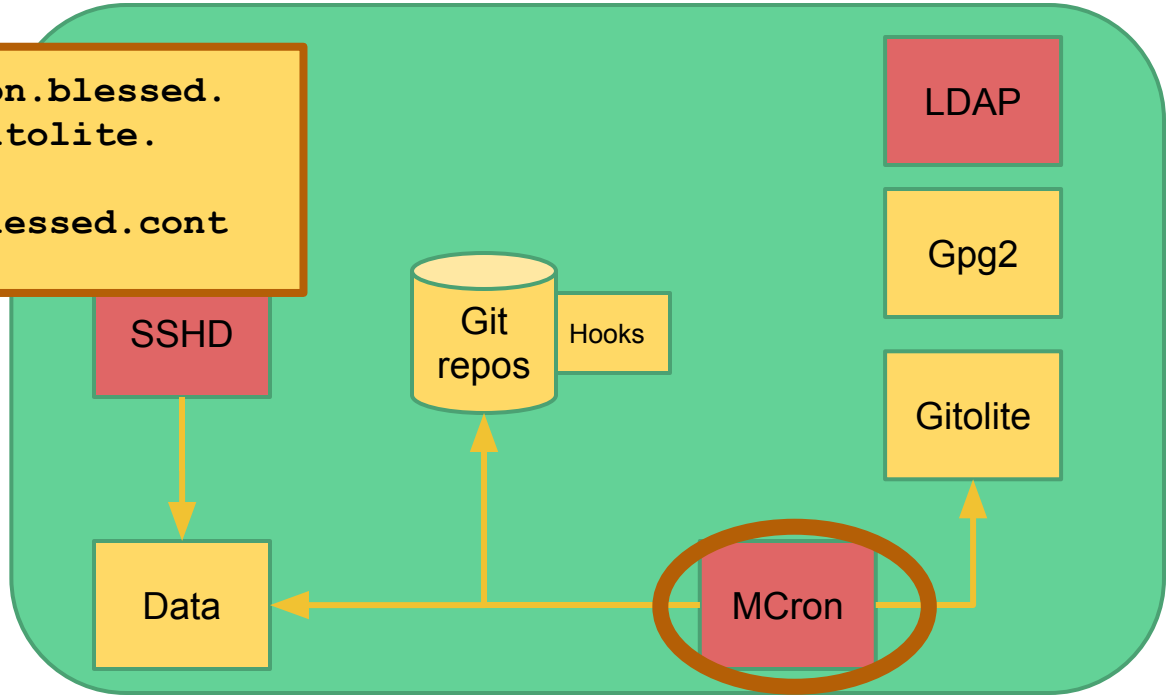
Leaf containers second



Docker rework

Leaf containers second

```
docker run --name=mcron.blessed.
cont --volumes-from gitolite.
blessed.cont
--volumes-from data.blessed.cont
```

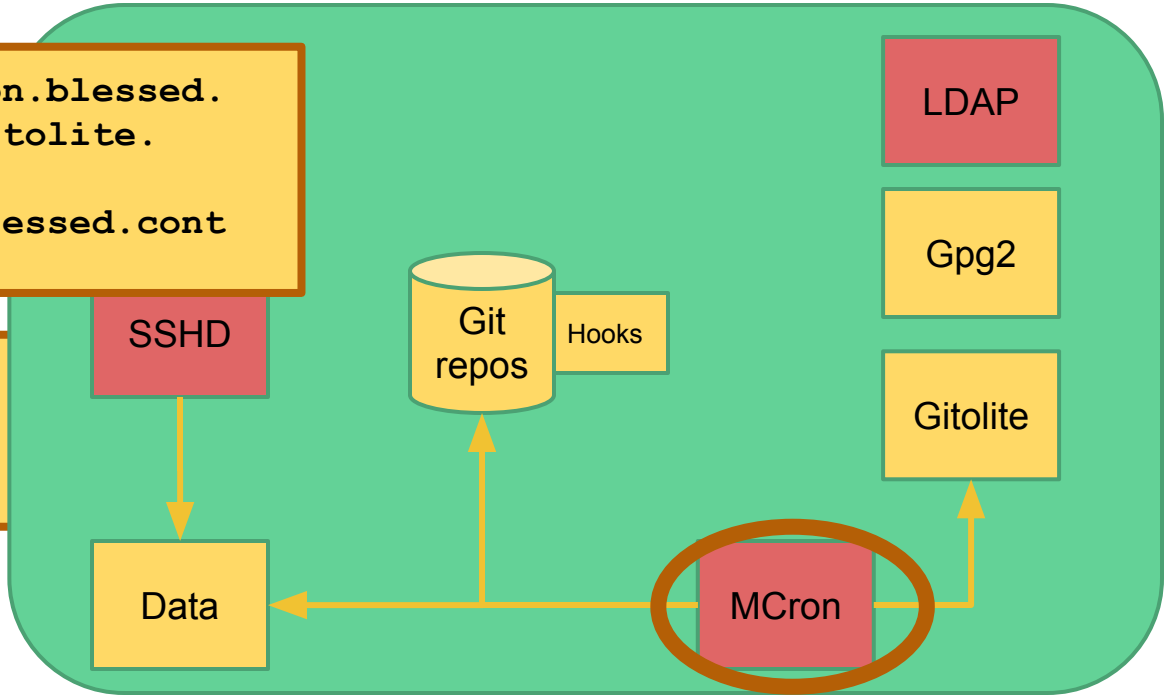


Docker rework

Leaf containers second

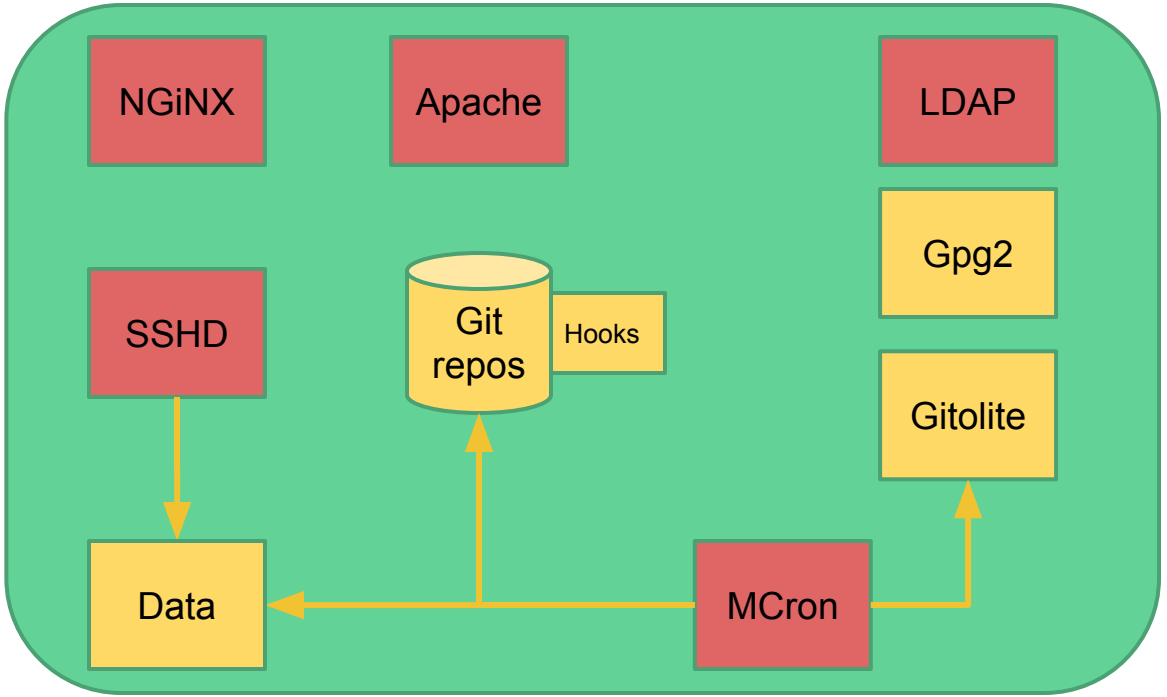
```
docker run --name=mcron.blessed.
cont --volumes-from gitolite.
blessed.cont
--volumes-from data.blessed.cont
```

```
/home/git/repositories
/home/git/gitolite
/home/git/data
```



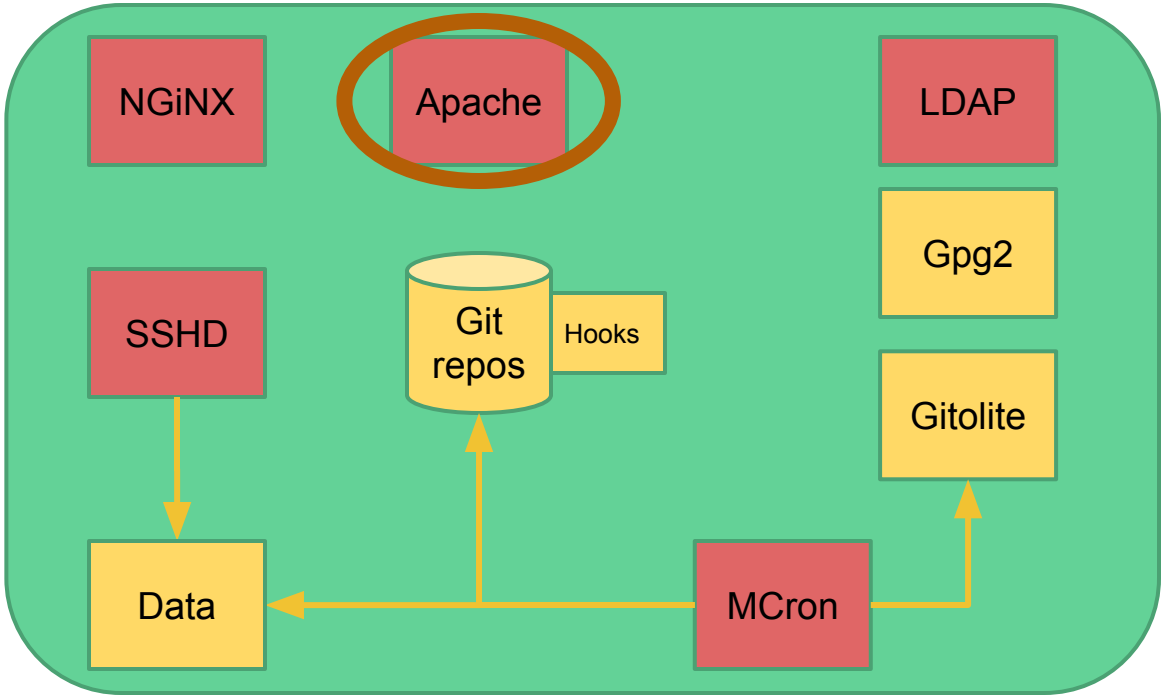
Docker rework

Linked containers last



Docker rework

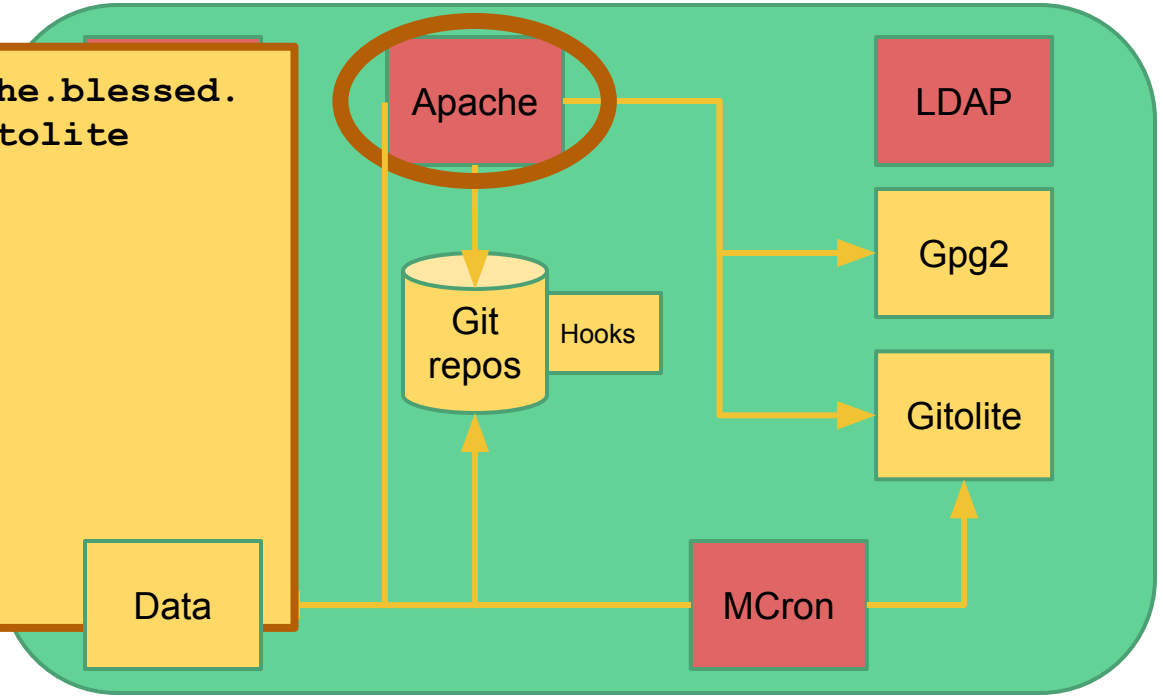
Linked containers last



Docker rework

Linked containers last

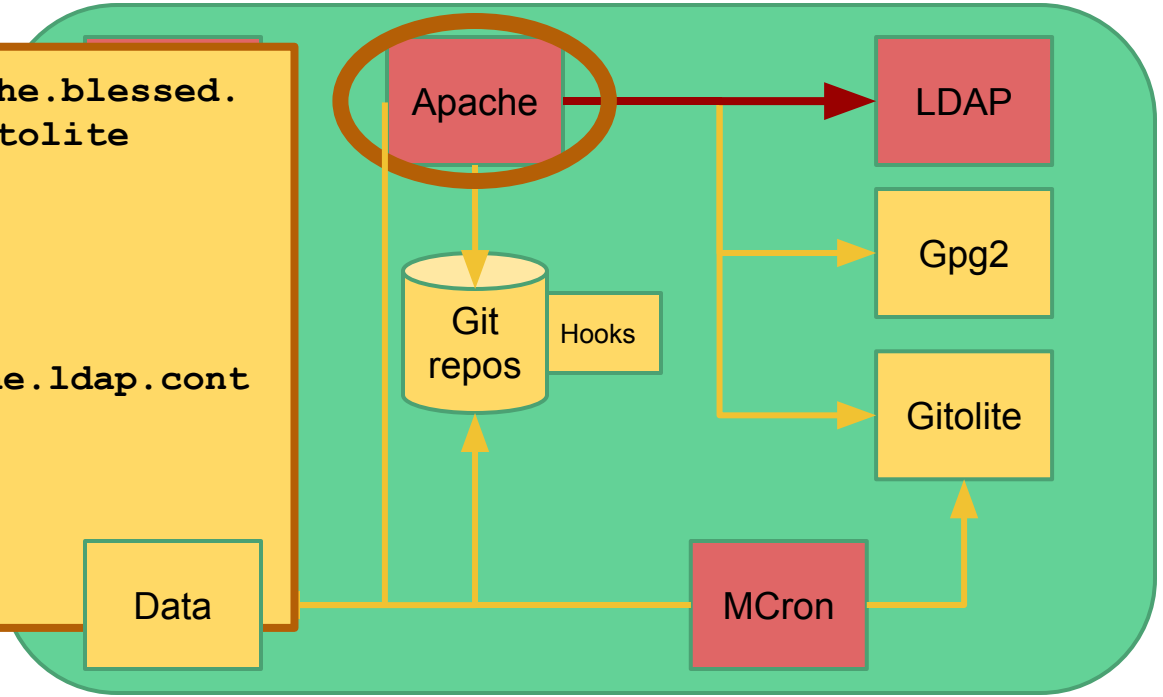
```
docker run --name=apache.blessed.
cont --volumes-from gitolite
--volumes-from repos
--volumes-from gpg2
--volumes-from data
```



Docker rework

Linked containers last

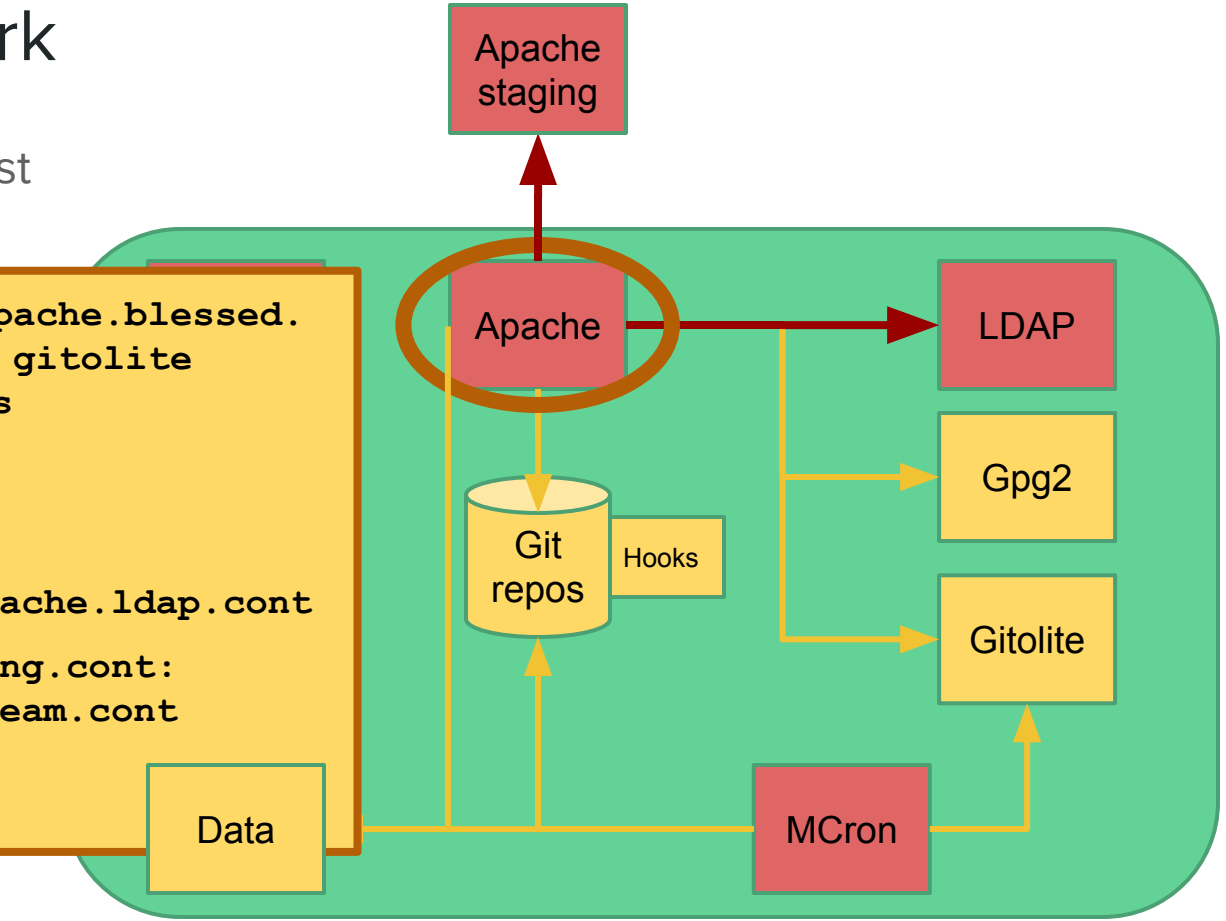
```
docker run --name=apache.blessed.
cont --volumes-from gitolite
--volumes-from repos
--volumes-from gpg2
--volumes-from data

--link ldap.cont:apache.ldap.cont
```



Docker rework

Linked containers last

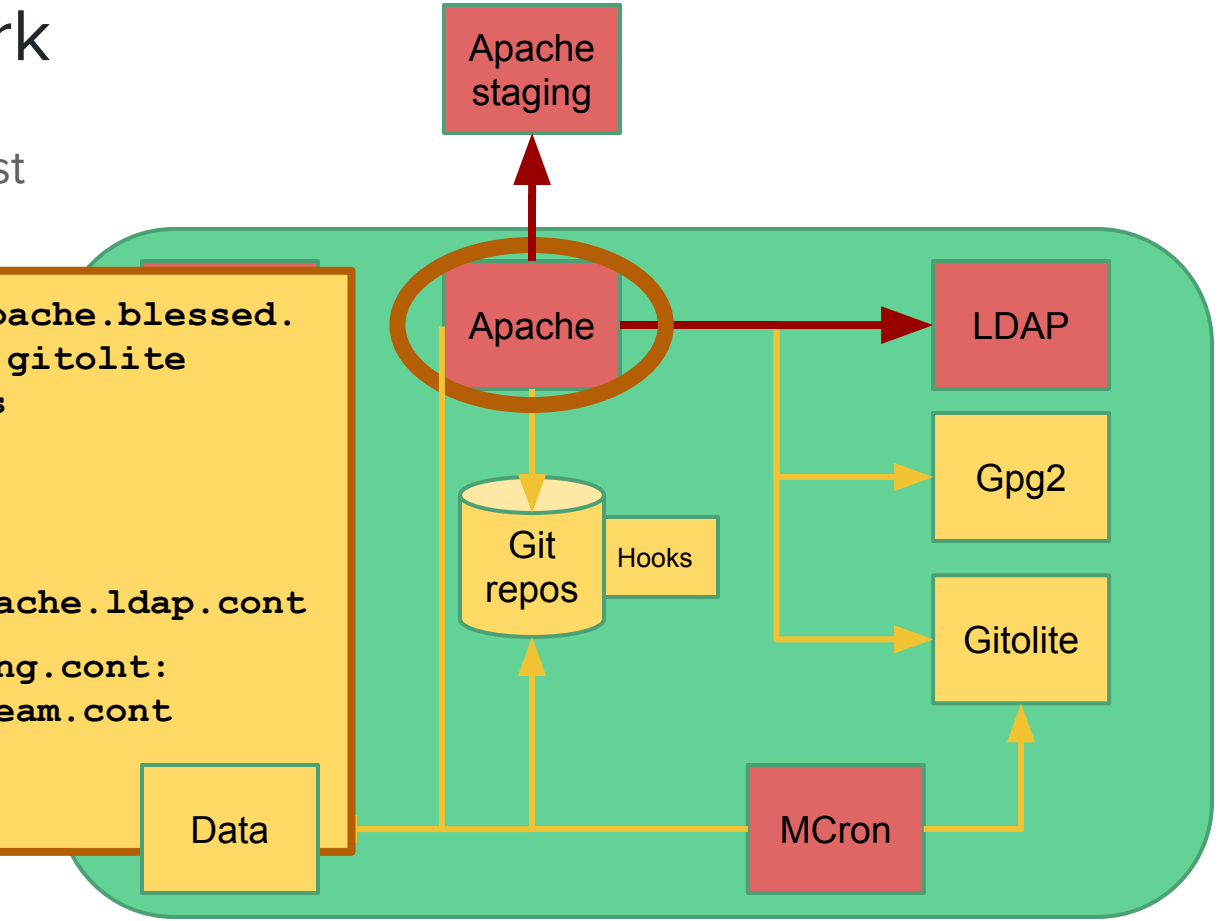
```
docker run --name=apache.blessed.
cont --volumes-from gitolite
--volumes-from repos
--volumes-from gpg2
--volumes-from data

--link ldap.cont:apache.ldap.cont
--link apache.staging.cont:
       apache.upstream.cont
```



Docker rework

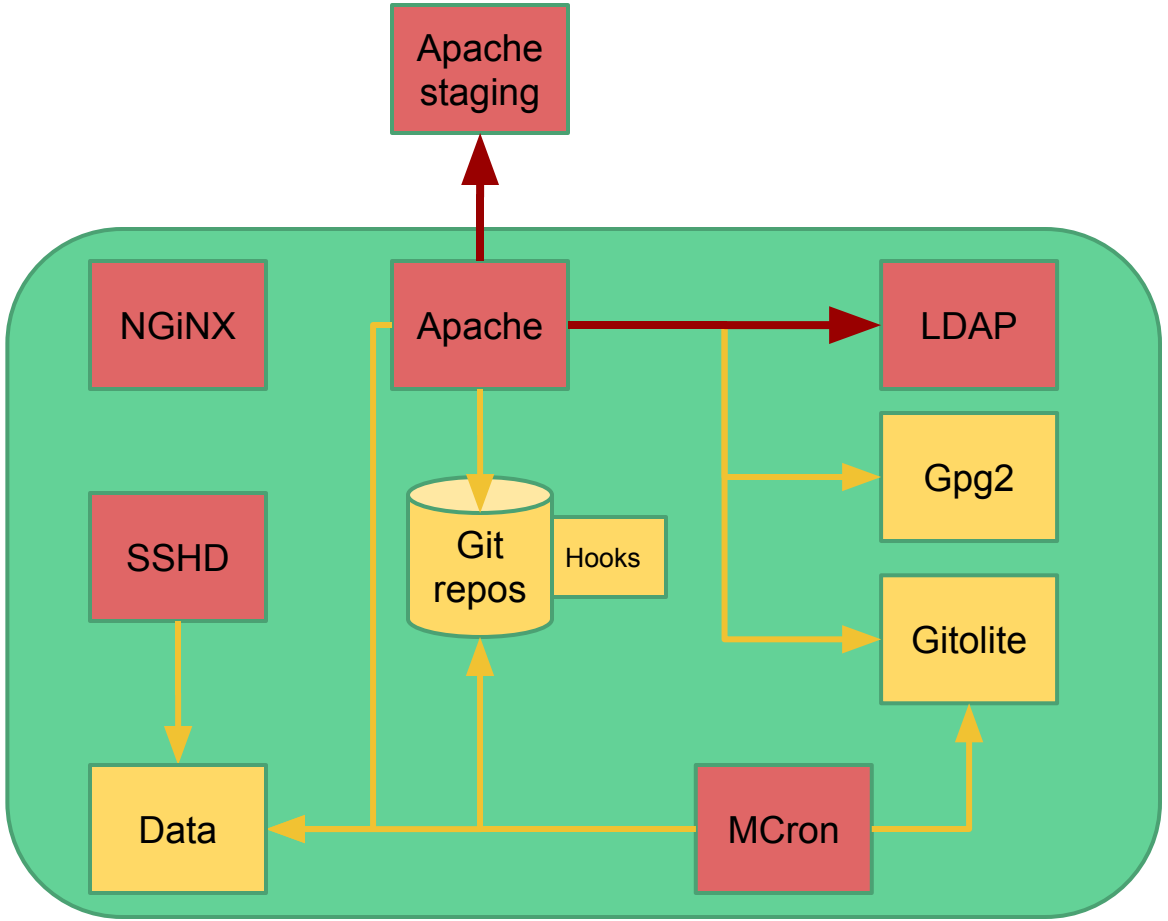
Linked containers last

```
docker run --name=apache.blessed.
cont --volumes-from gitolite
--volumes-from repos
--volumes-from gpg2
--volumes-from data

--link ldap.cont:apache.ldap.cont
--link apache.staging.cont:
       apache.upstream.cont
-p 6043:8543
-p 6053:8553
```



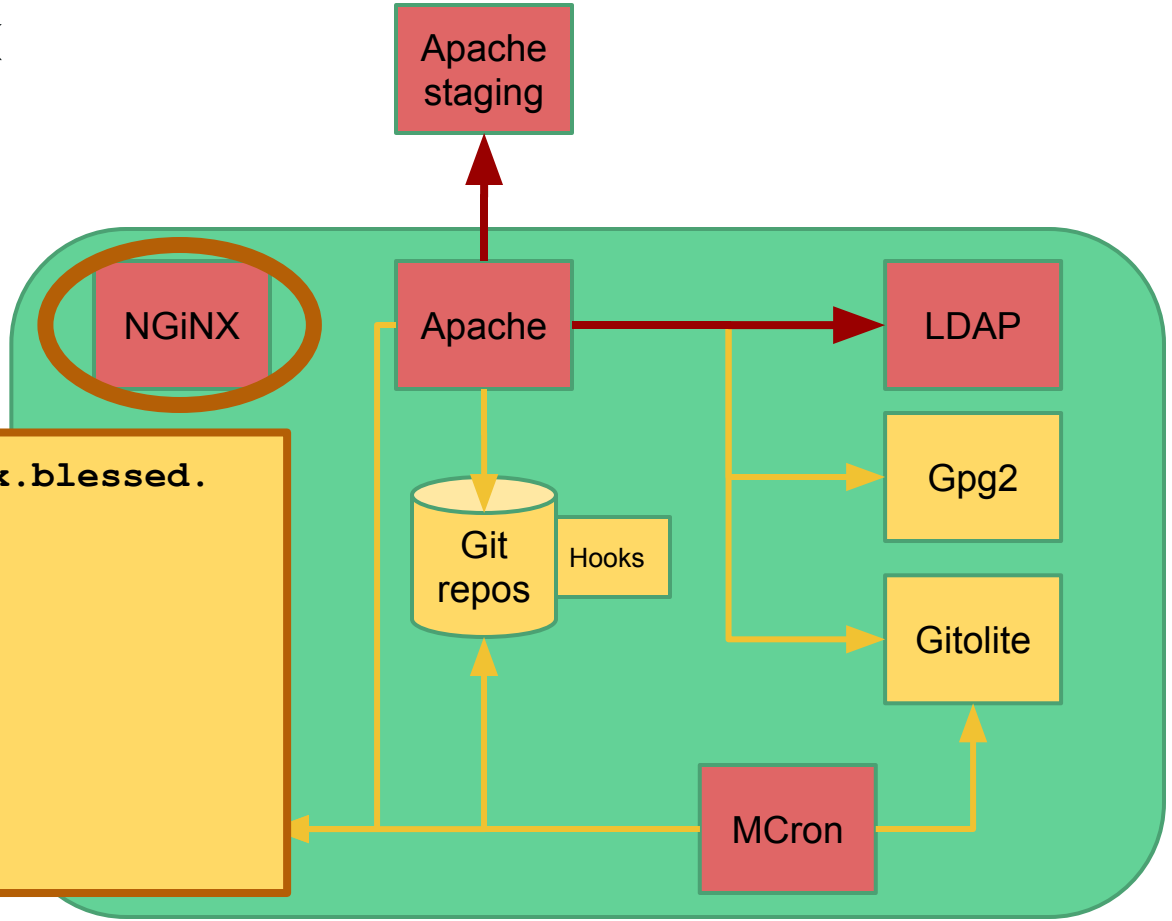
Docker rework

Linked containers last



Docker rework

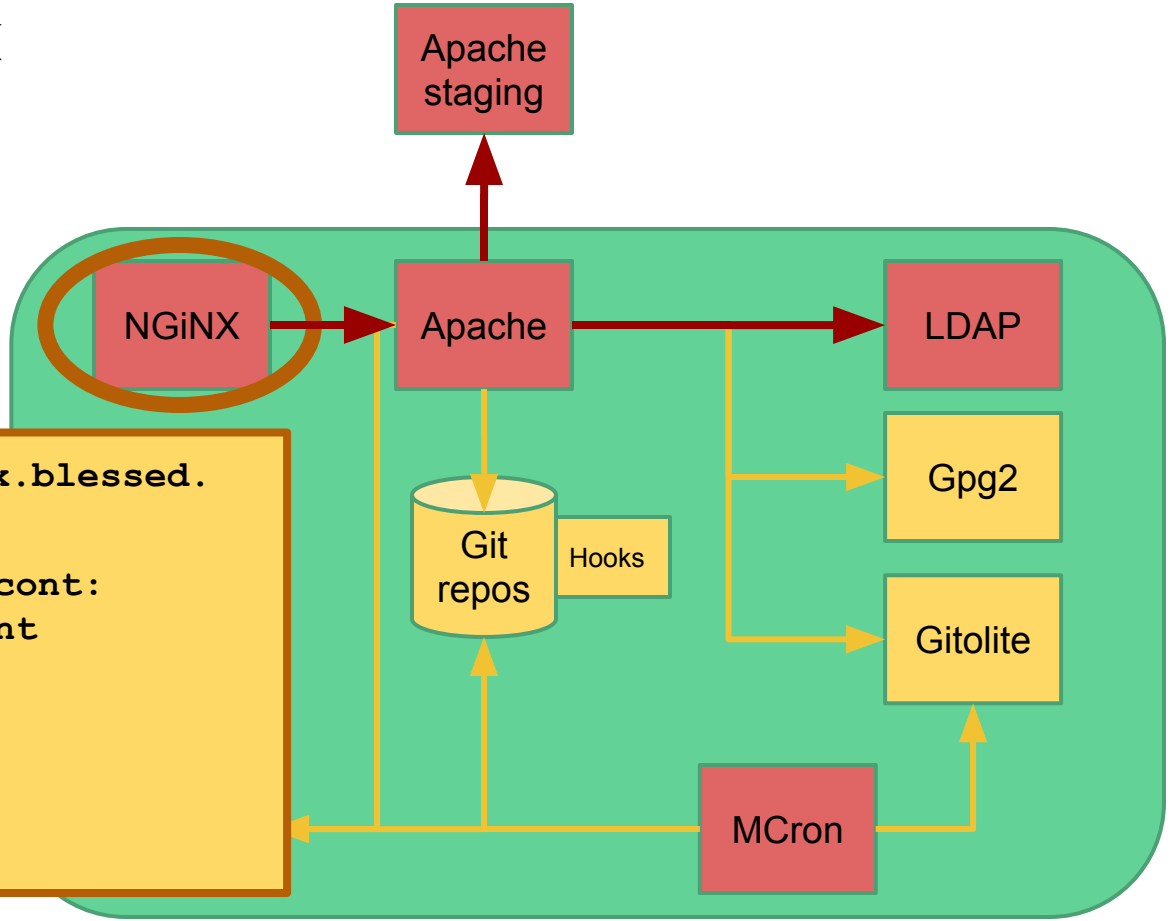
Linked containers last




```
docker run --name=nginx.blessed.
cont
```

Docker rework

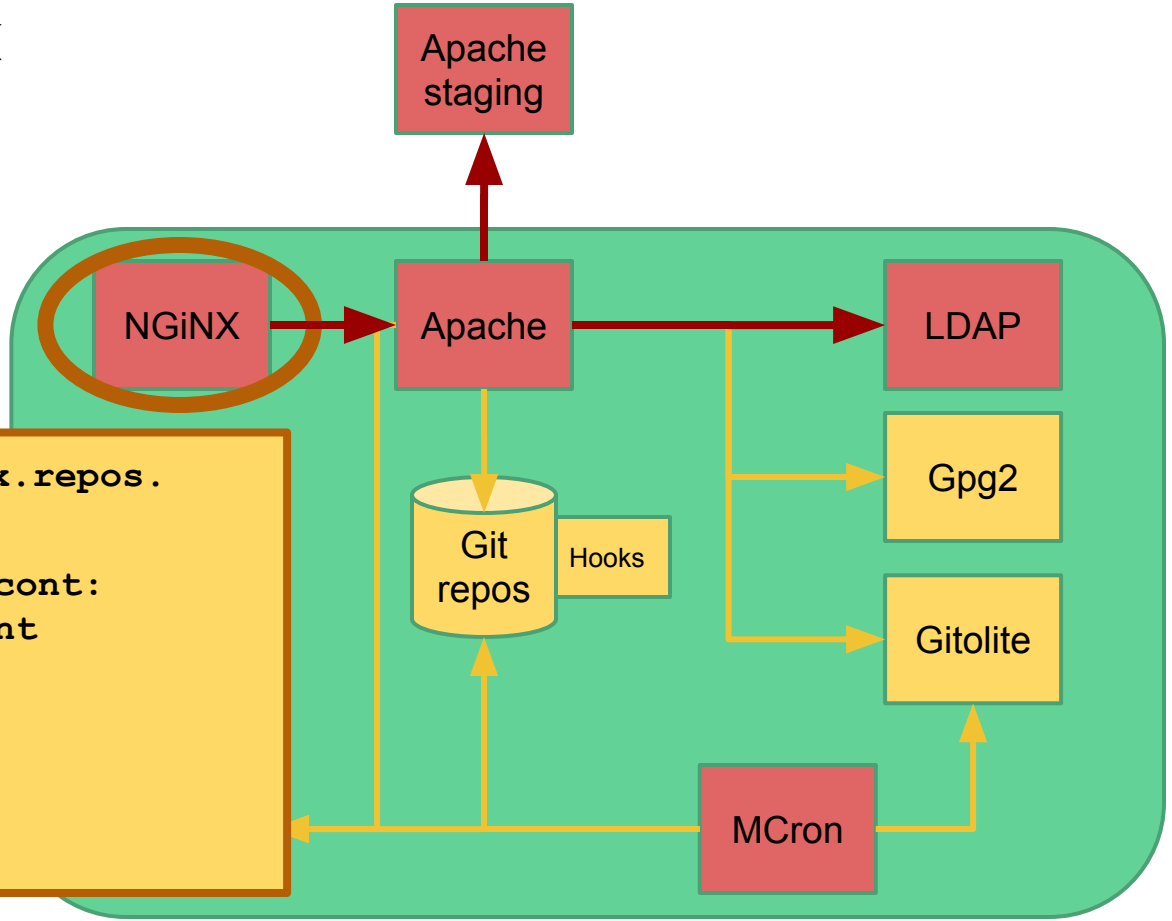
Linked containers last




```
docker run --name=nginx.blessed.
cont
--link apache.blessed.cont:
       nginx.apache.cont
```

Docker rework

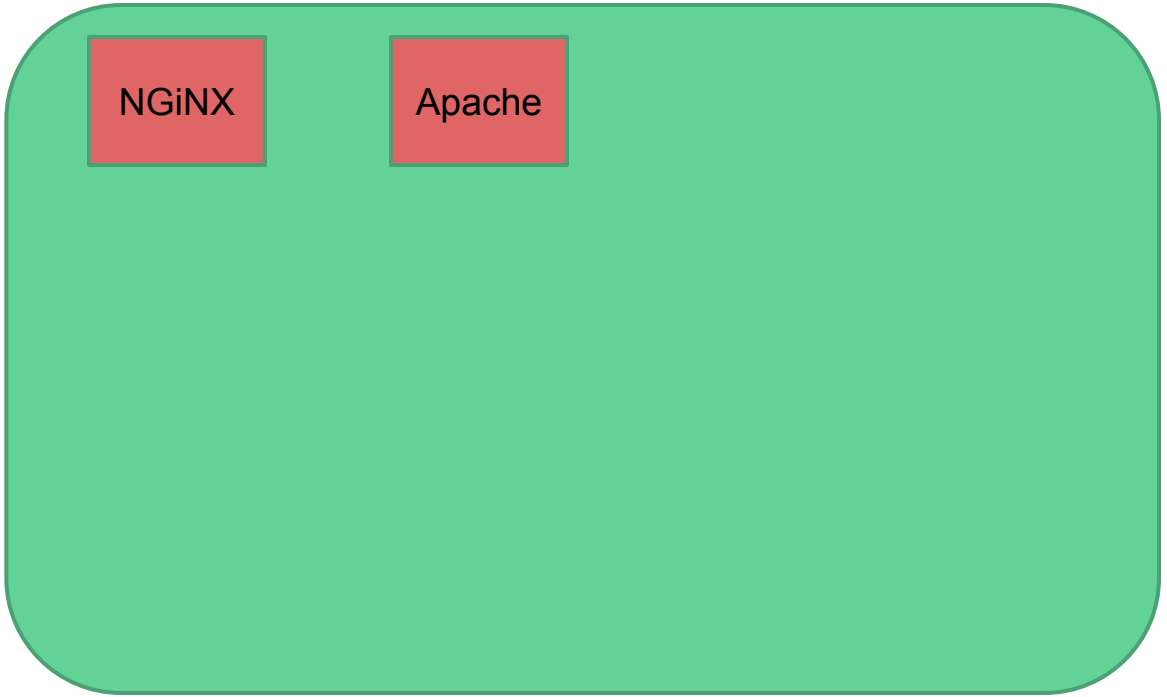
Linked containers last




```
docker run --name=nginx.repos.
blessed.cont
--link apache.blessed.cont:
       nginx.apache.cont
-p 6080:80
-p 6443:443
```

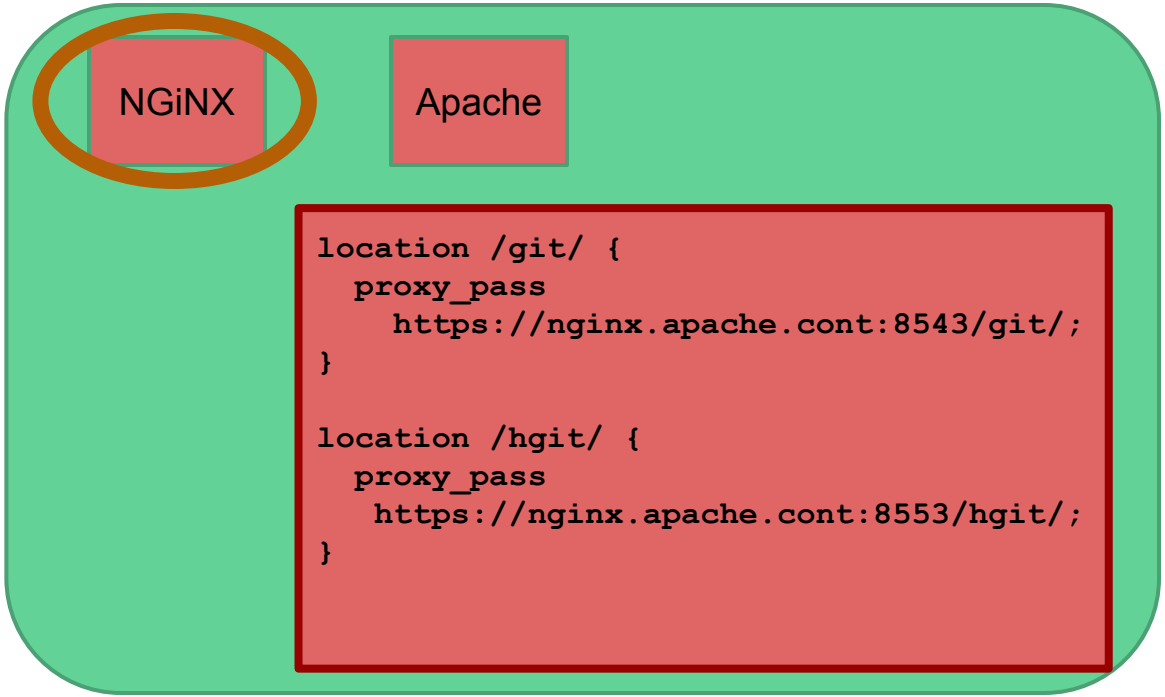
Docker Advantages

Configuration



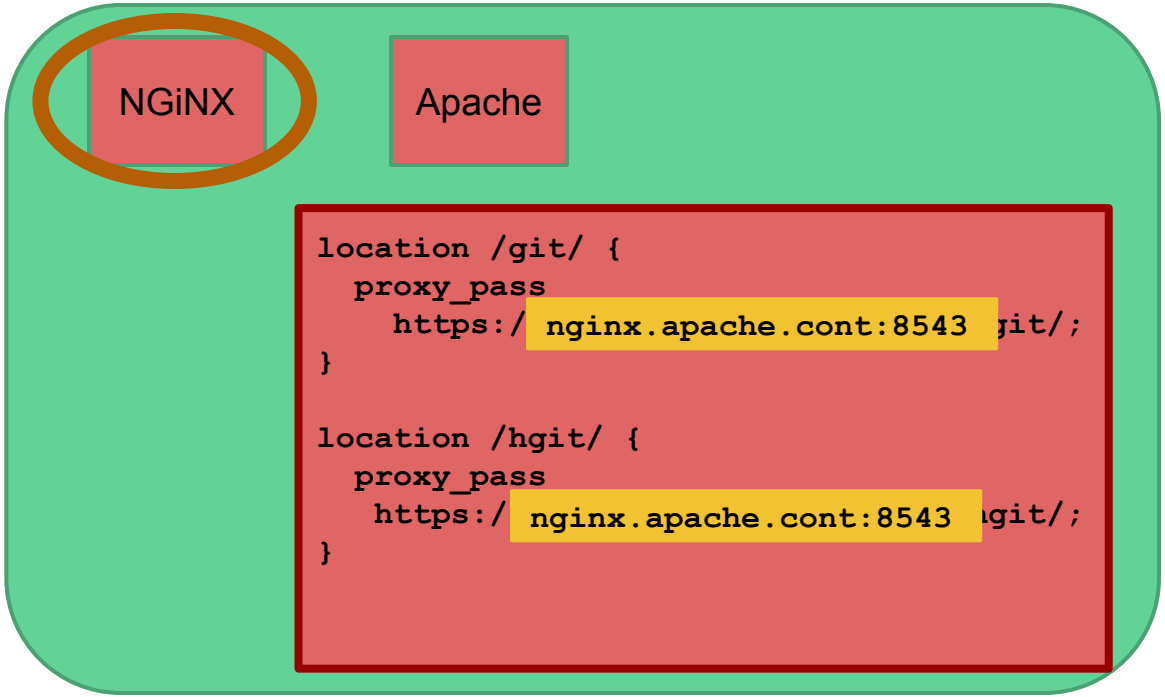
Docker Advantages

Configuration



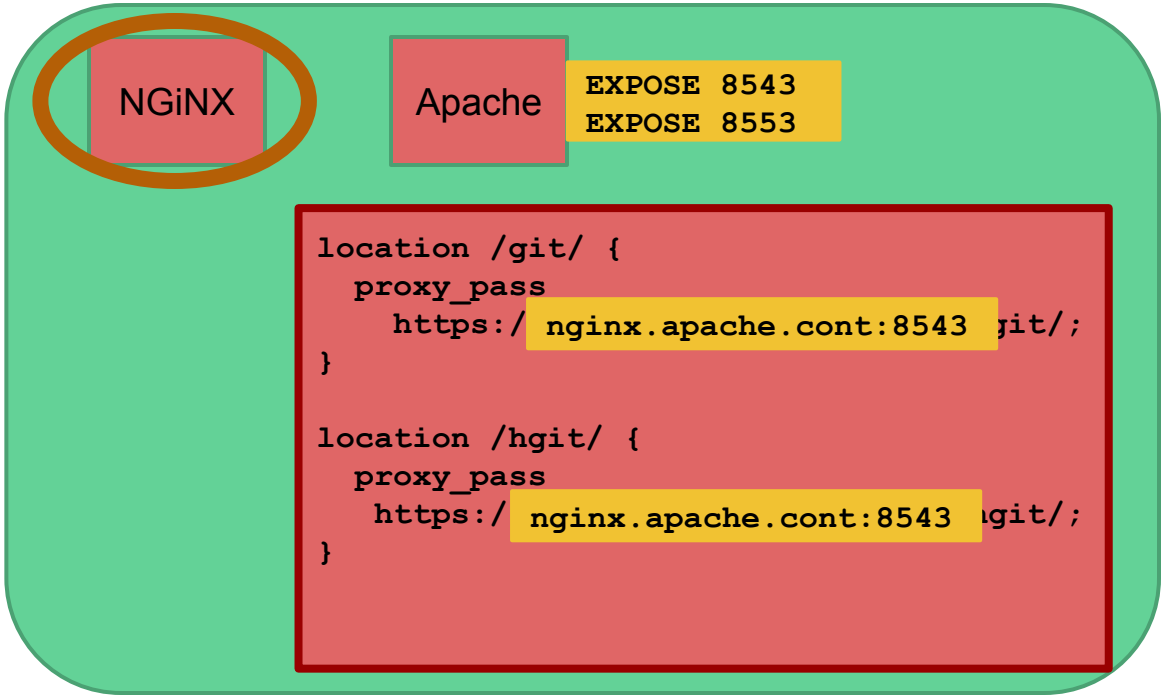
Docker Advantages

Configuration



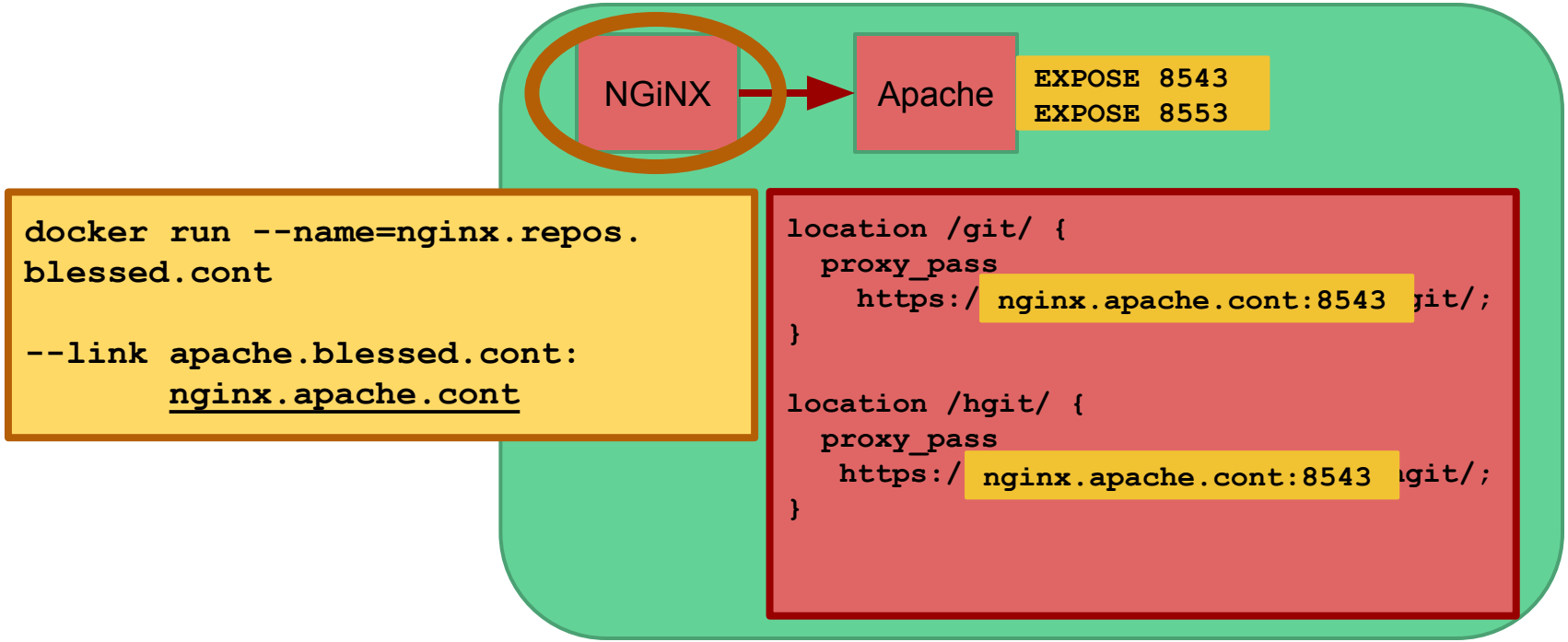
Docker Advantages

Configuration



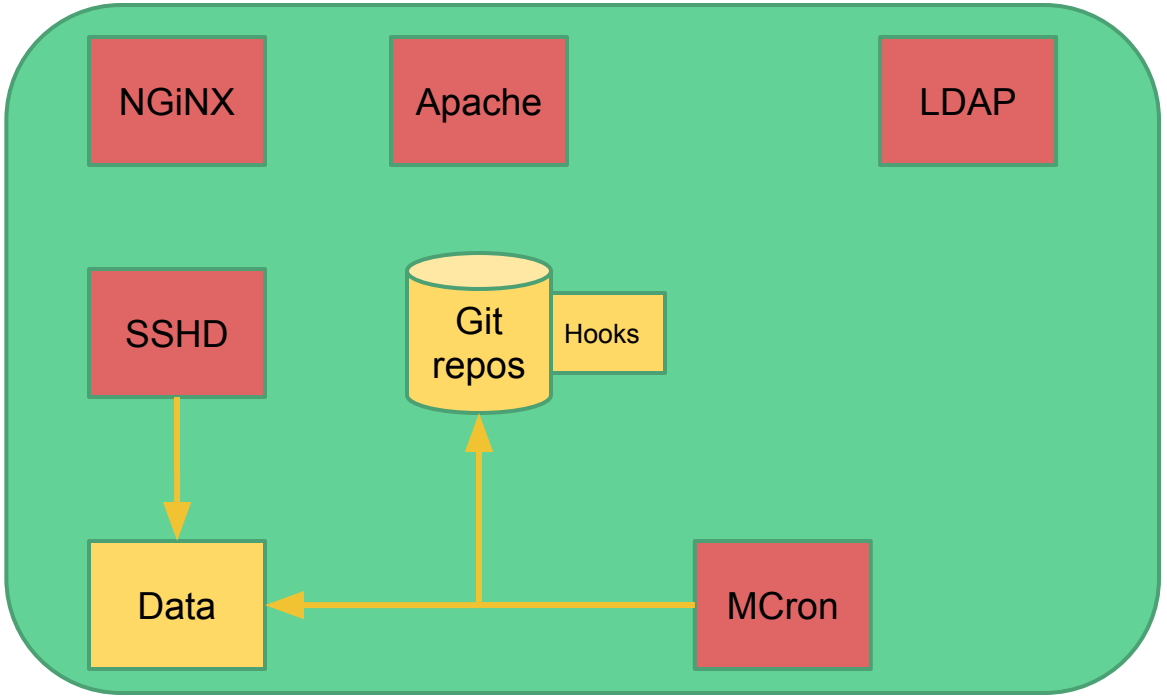
Docker Advantages

Configuration



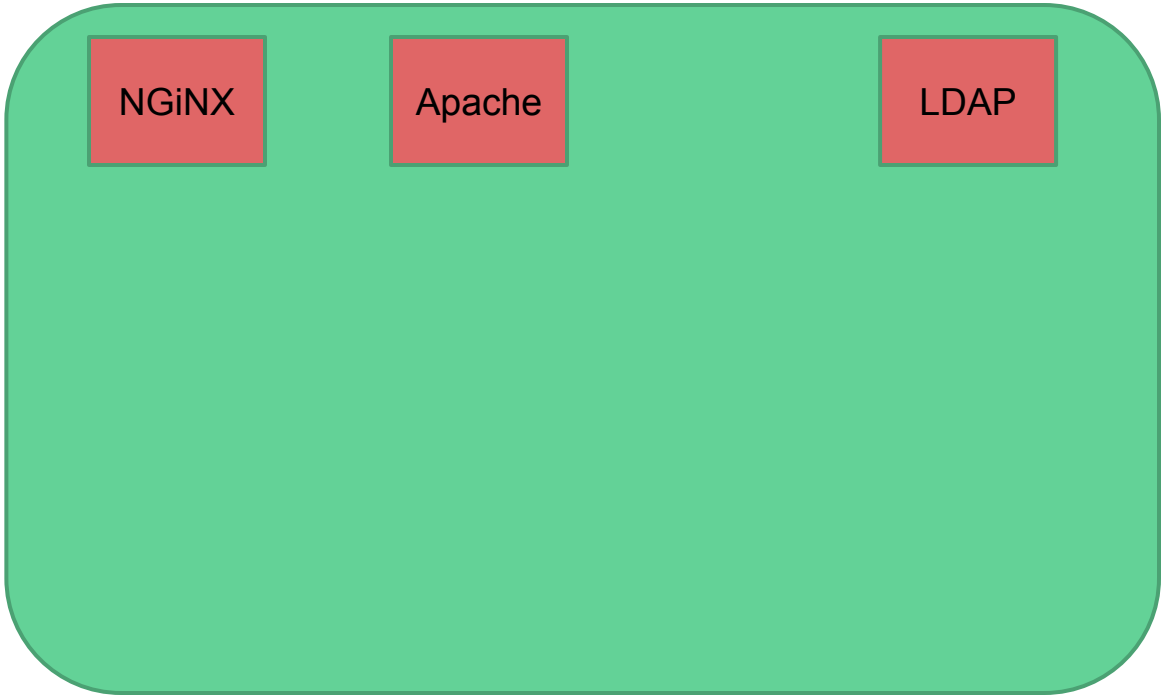
Docker Advantages

Isolation (services)



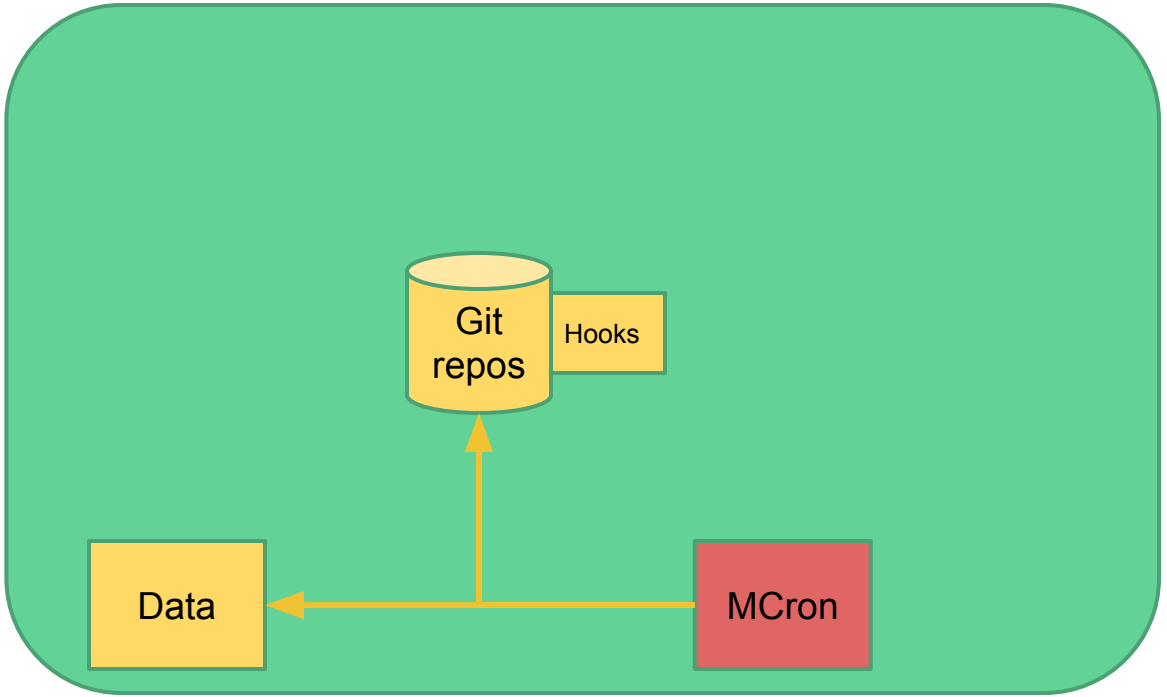
Docker Advantages

Isolation (services)



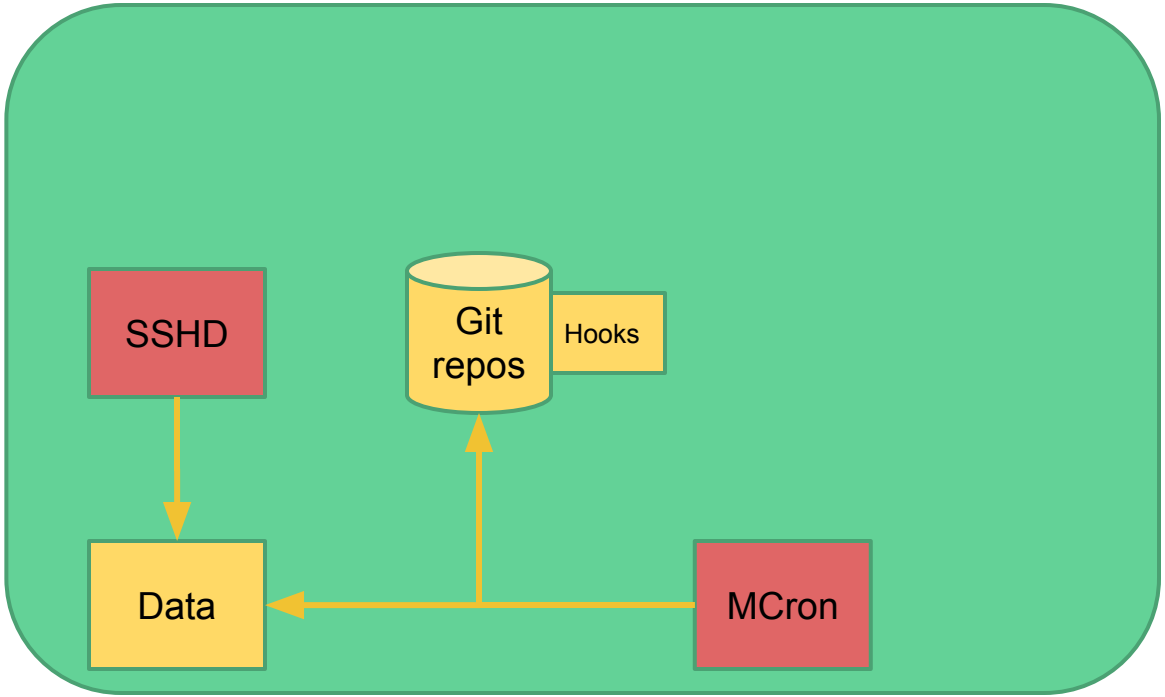
Docker Advantages

Isolation (services)



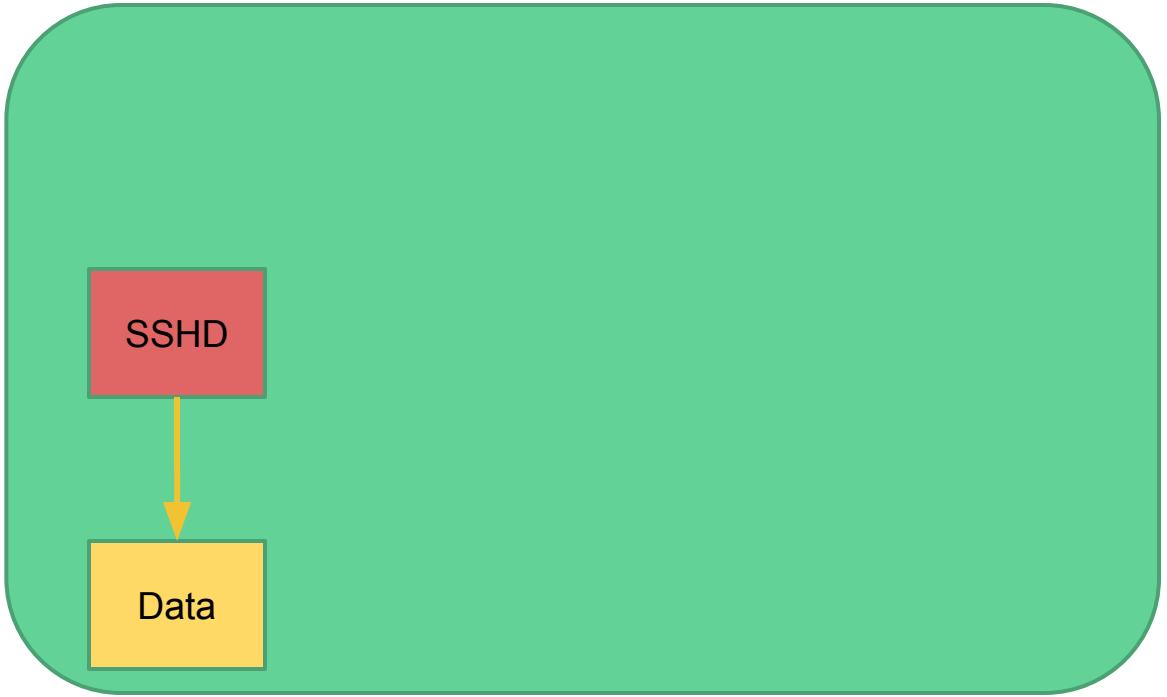
Docker Advantages

Isolation (services)



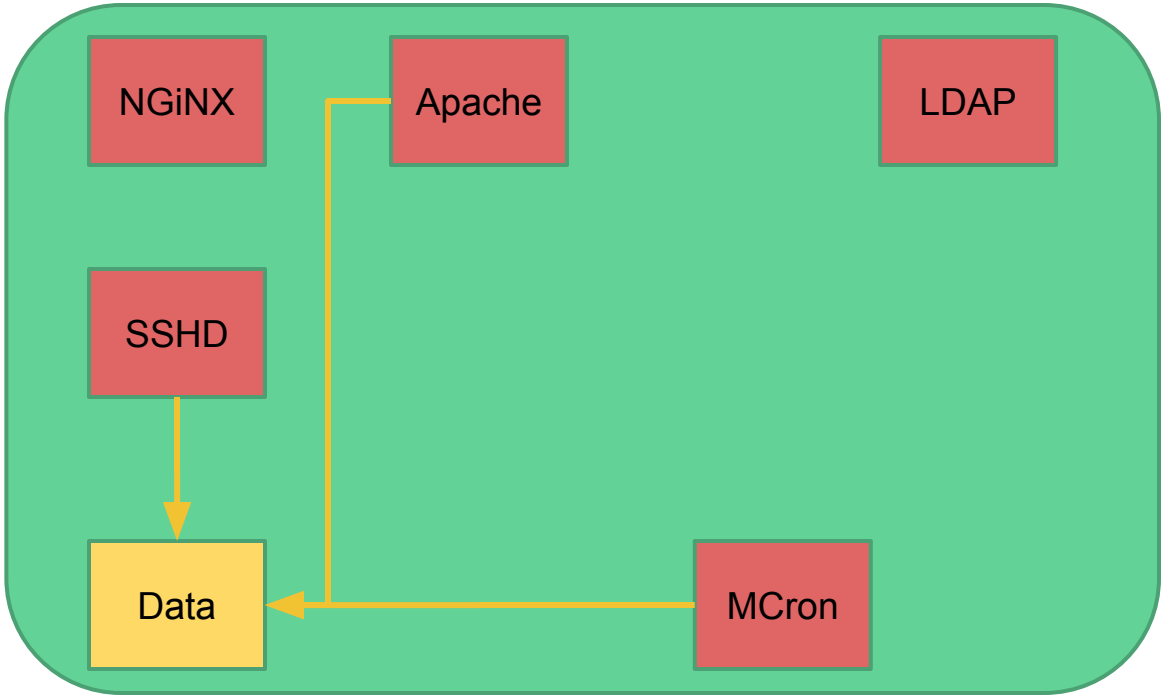
Docker Advantages

Isolation (services)



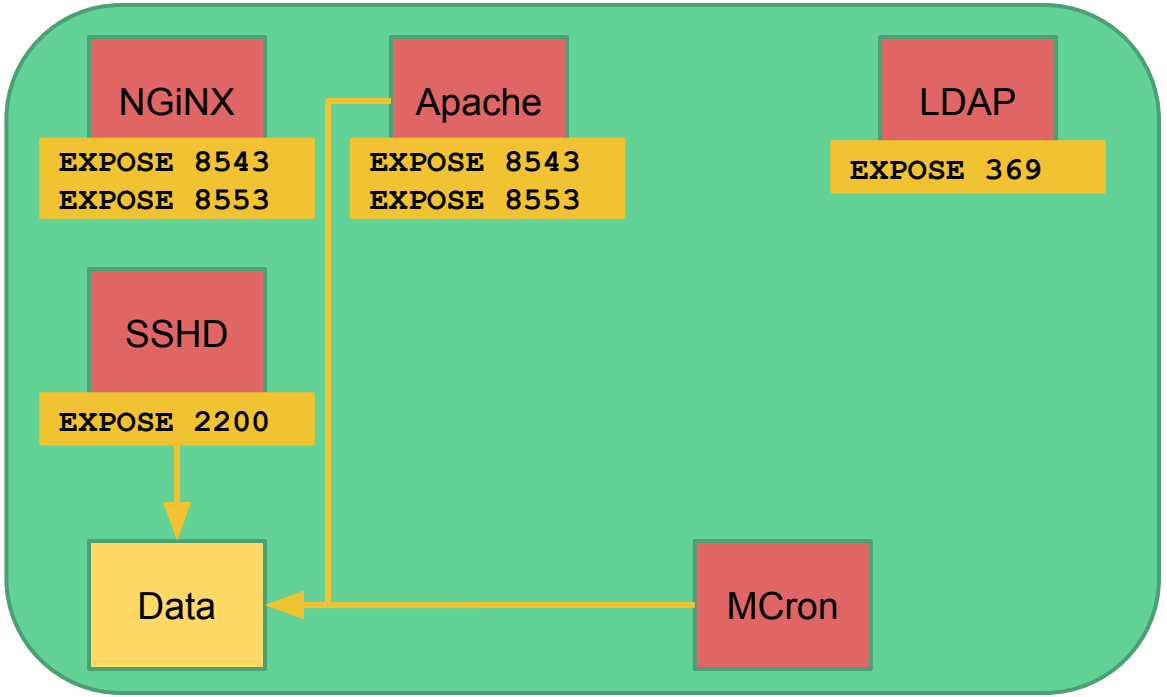
Docker Advantages

Isolation (ports/names)



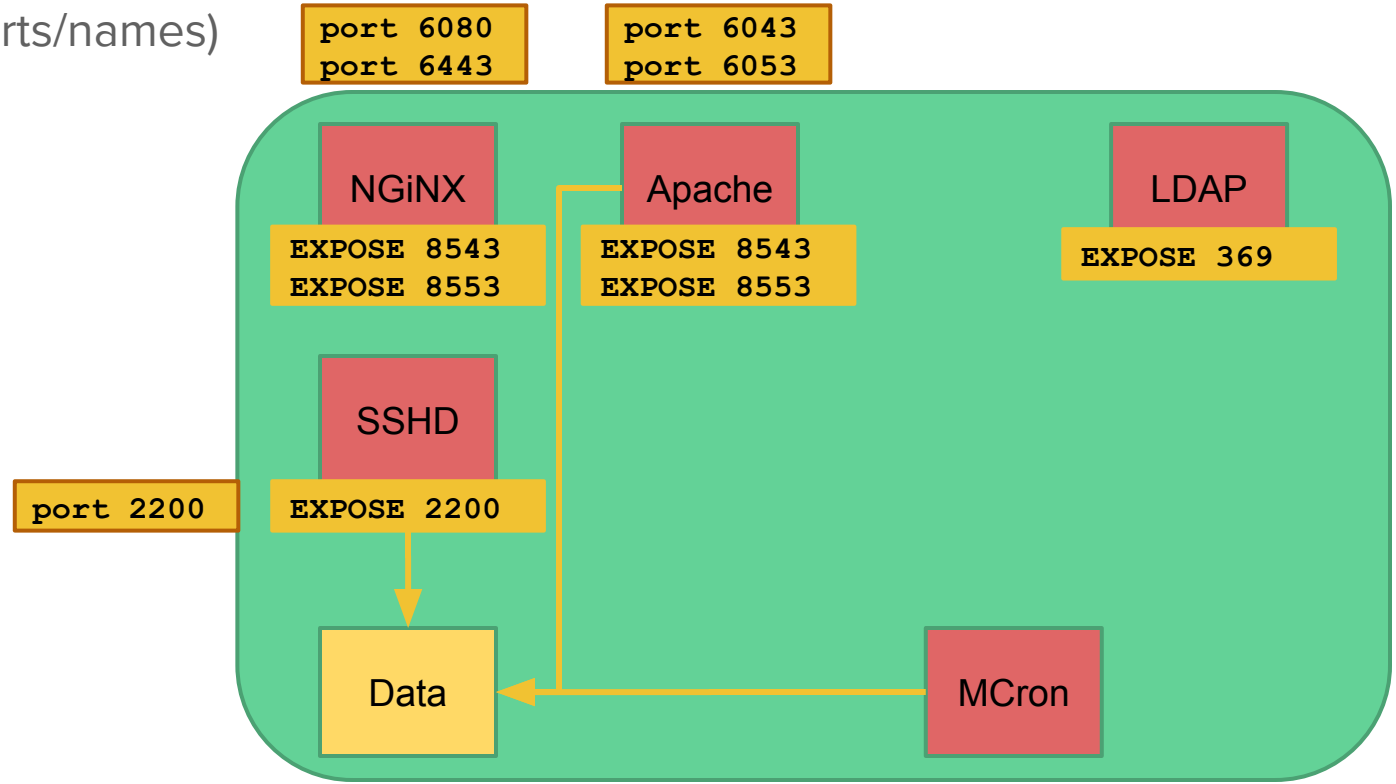
Docker Advantages

Isolation (ports/names)



Docker Advantages

Isolation (ports/names)



End result

From:

To:

- 21 containers, in 3 sets, one for each environment.
- service vs. data
- Extensible without downtime.