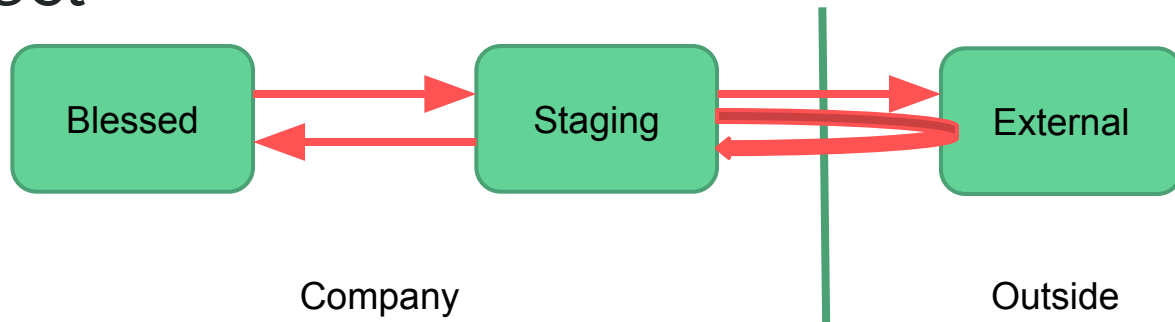# Docker

Architecture Rework Case Study
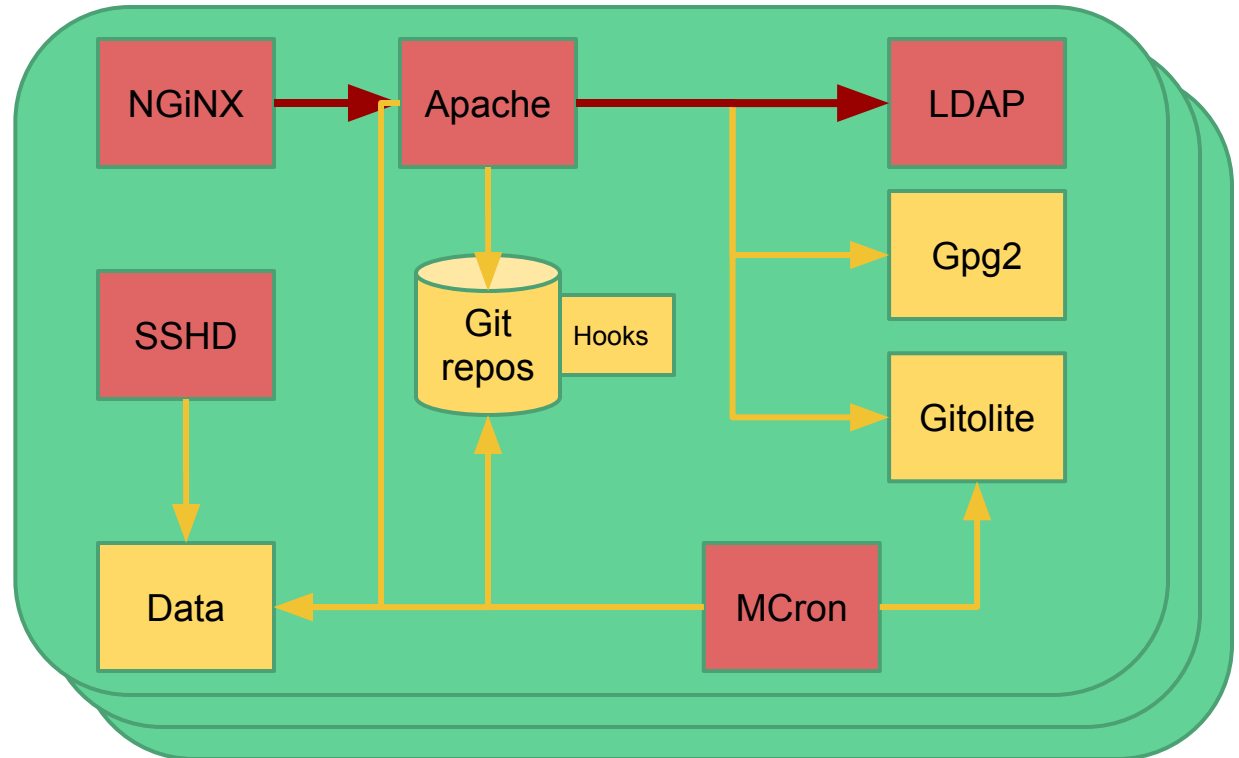
Project
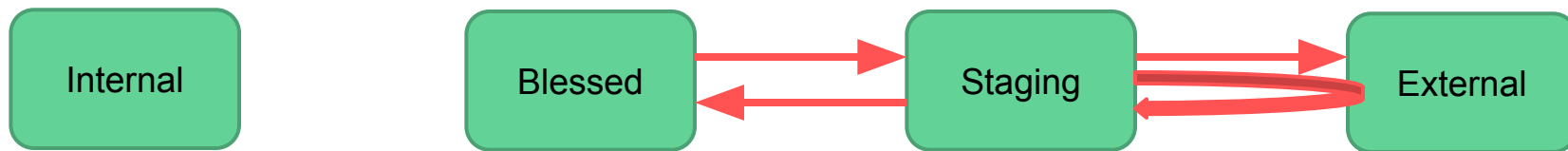
From:

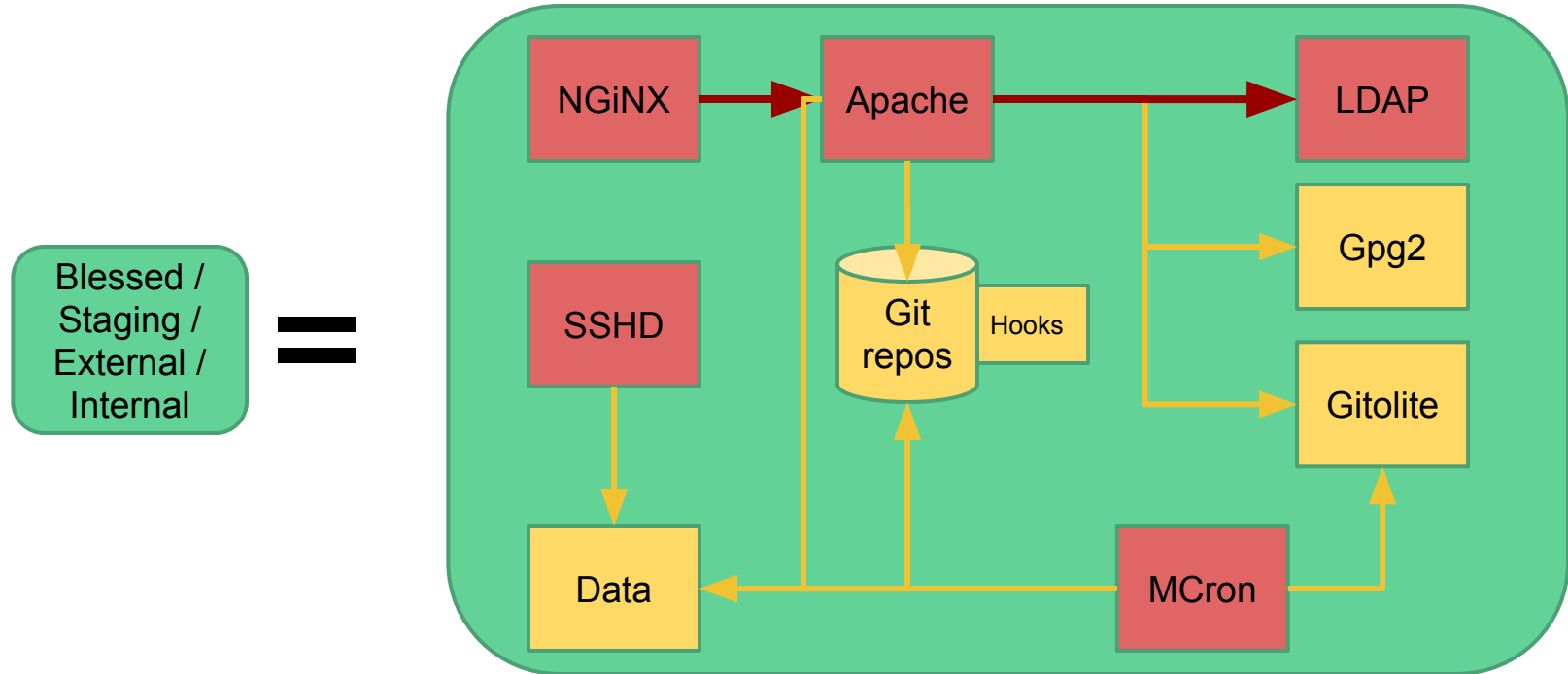# Project

To:

What are the issues?

Example: Replicated Git repos hosting servers

- ● Isolation (server)

- ● Configuration

- ● Reproducibility

What are the issues?

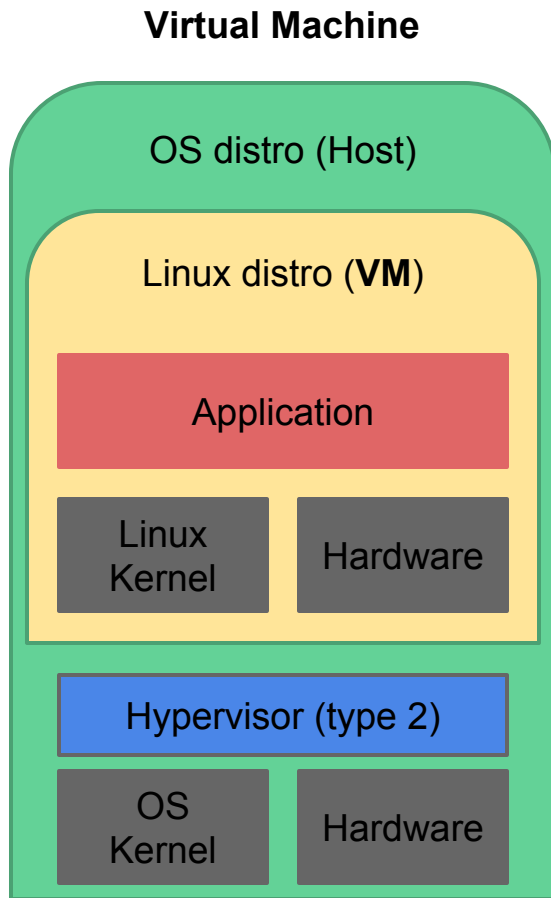Example: Monolithic service => Isolation (processes) issue
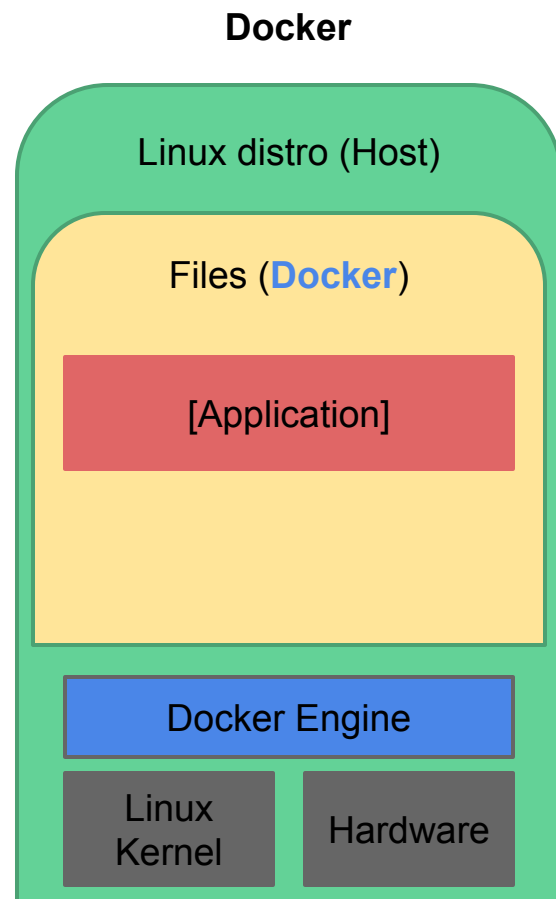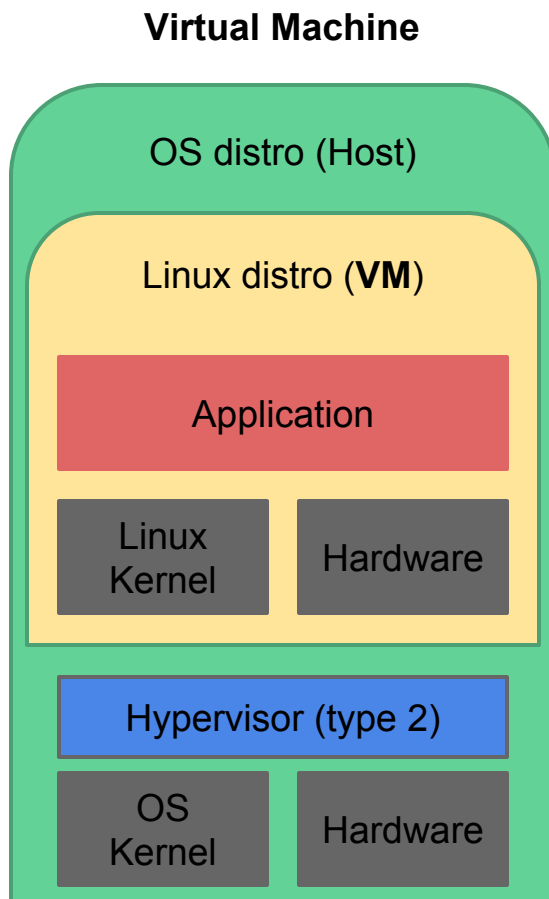
What is Docker

Lightweight virtualization

What is Docker
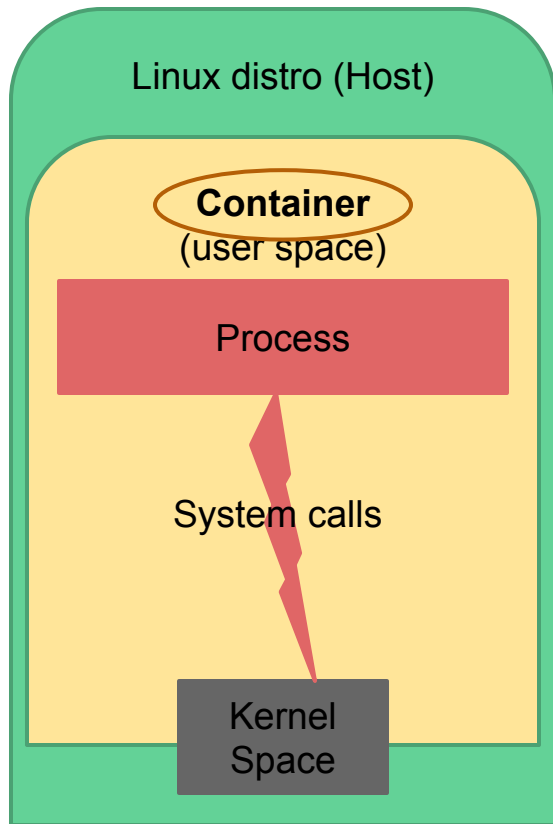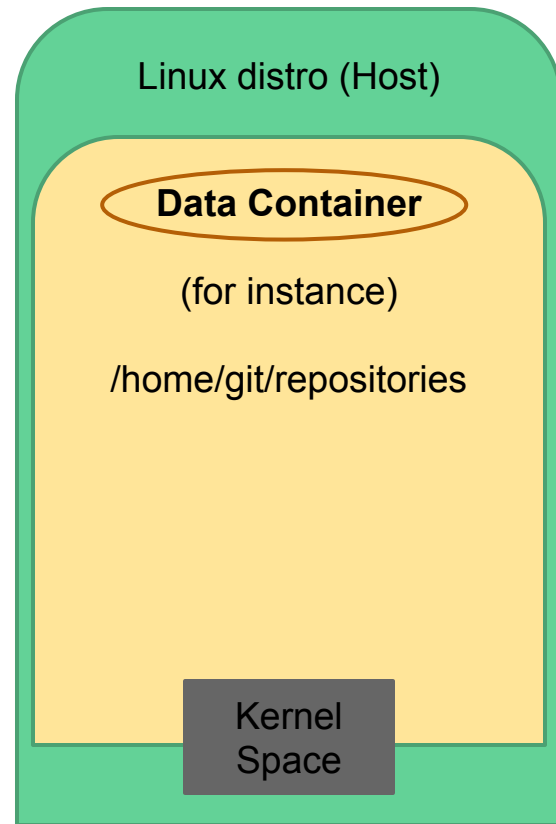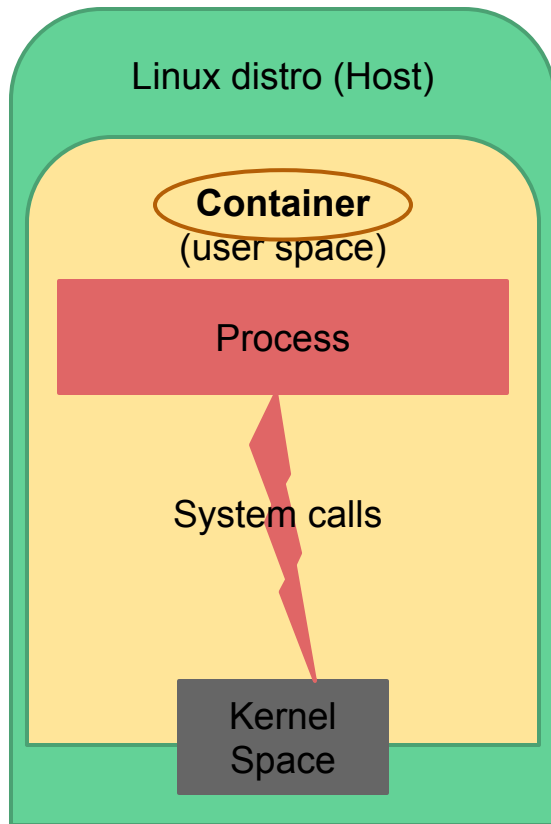
Lightweight virtualization

Virtual Machine

OS distro (Host)

Linux distro (**VM**)

Application

Linux Kernel

Hardware

Hypervisor (type 2)

OS Kernel

Hardware

What is Docker

Lightweight virtualization

Virtual Machine

OS distro (Host)

Linux distro (**VM**)

Application

Linux Kernel

Hardware

Hypervisor (type 2)

OS Kernel

Hardware

Docker

Linux distro (Host)

Files (**Docker**)

[Application]

Docker Engine

Linux Kernel

Hardware

What is Docker?

Containers

What is Docker?

Containers

Linux distro (Host)

Container
(user space)

Process

System calls

Kernel
Space

What is Docker?

Containers

What is Docker?

Images: docker build

Dockerfile:

FROM ubuntu

Linux distro (Host)

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

Linux distro (Host)

NGiNX

Ubuntu

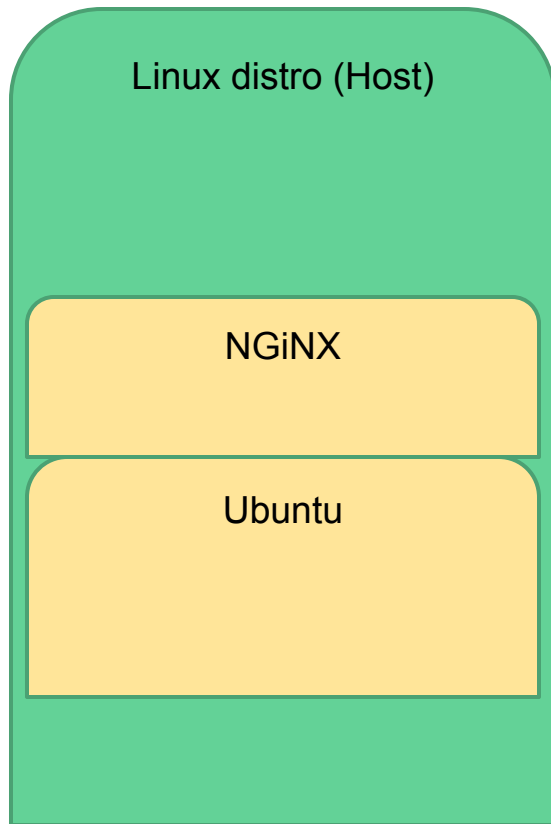# What is Docker?

Images: docker build

Dockerfile:
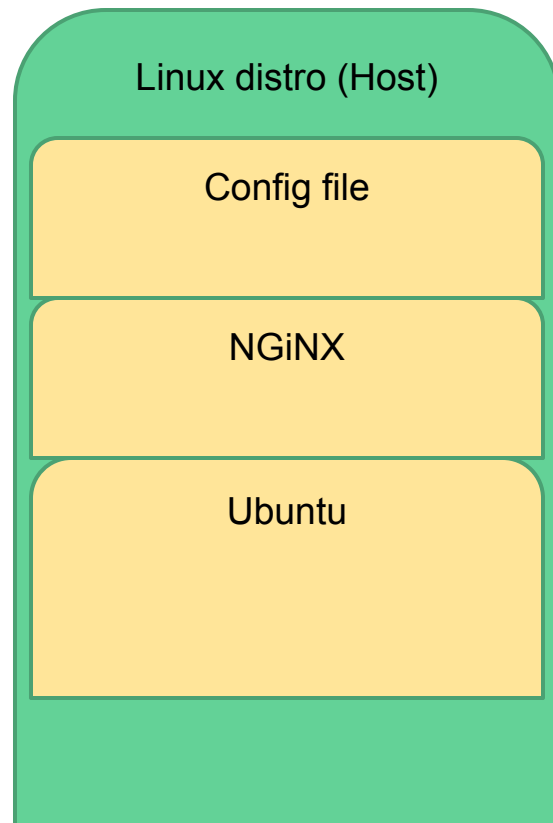
FROM ubuntu

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

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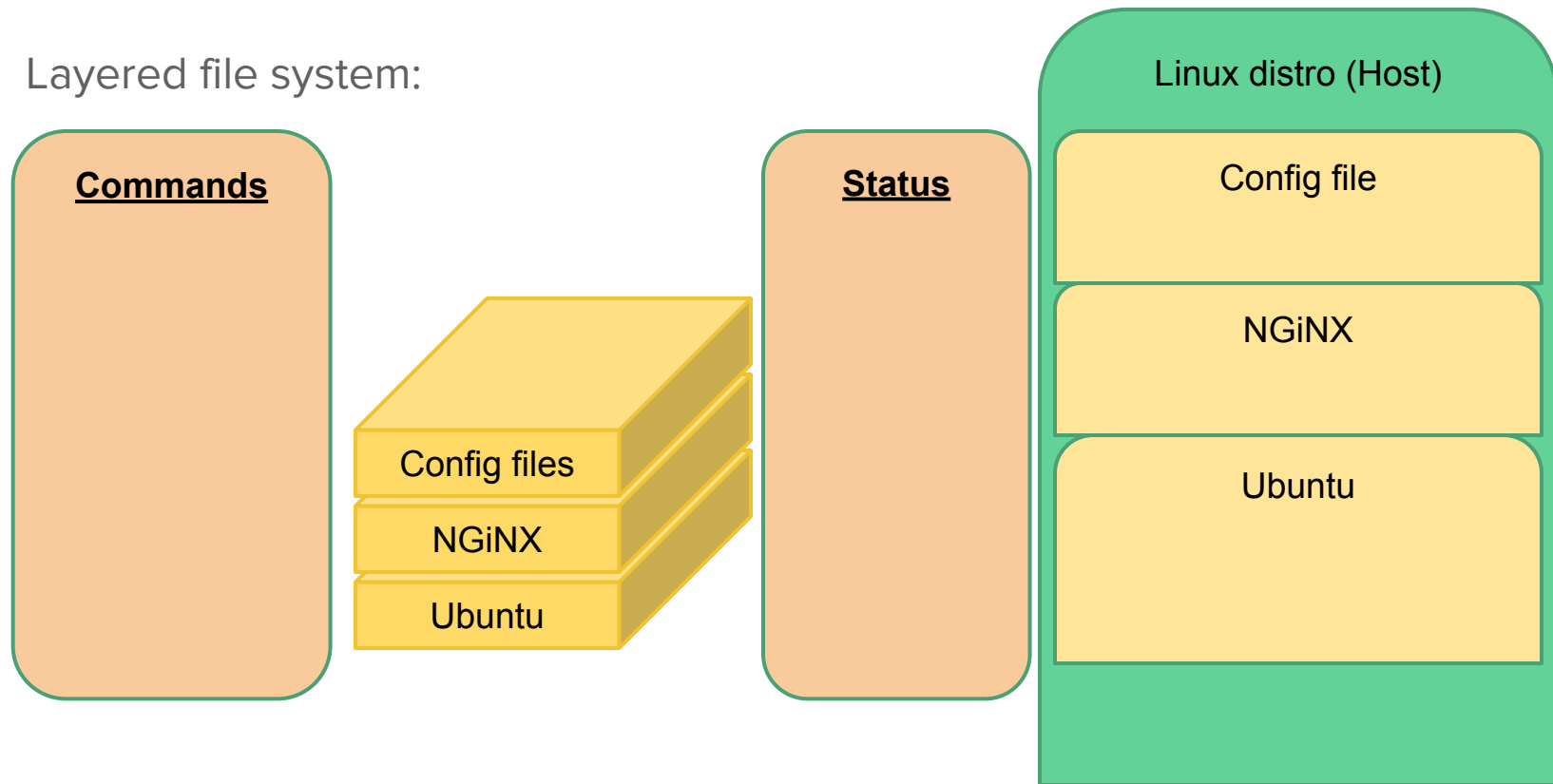RUN ln -fs /home/git/env.conf /etc/nginx/nginx.conf

EXPOSE 80
EXPOSE 443

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

Linux distro (Host)
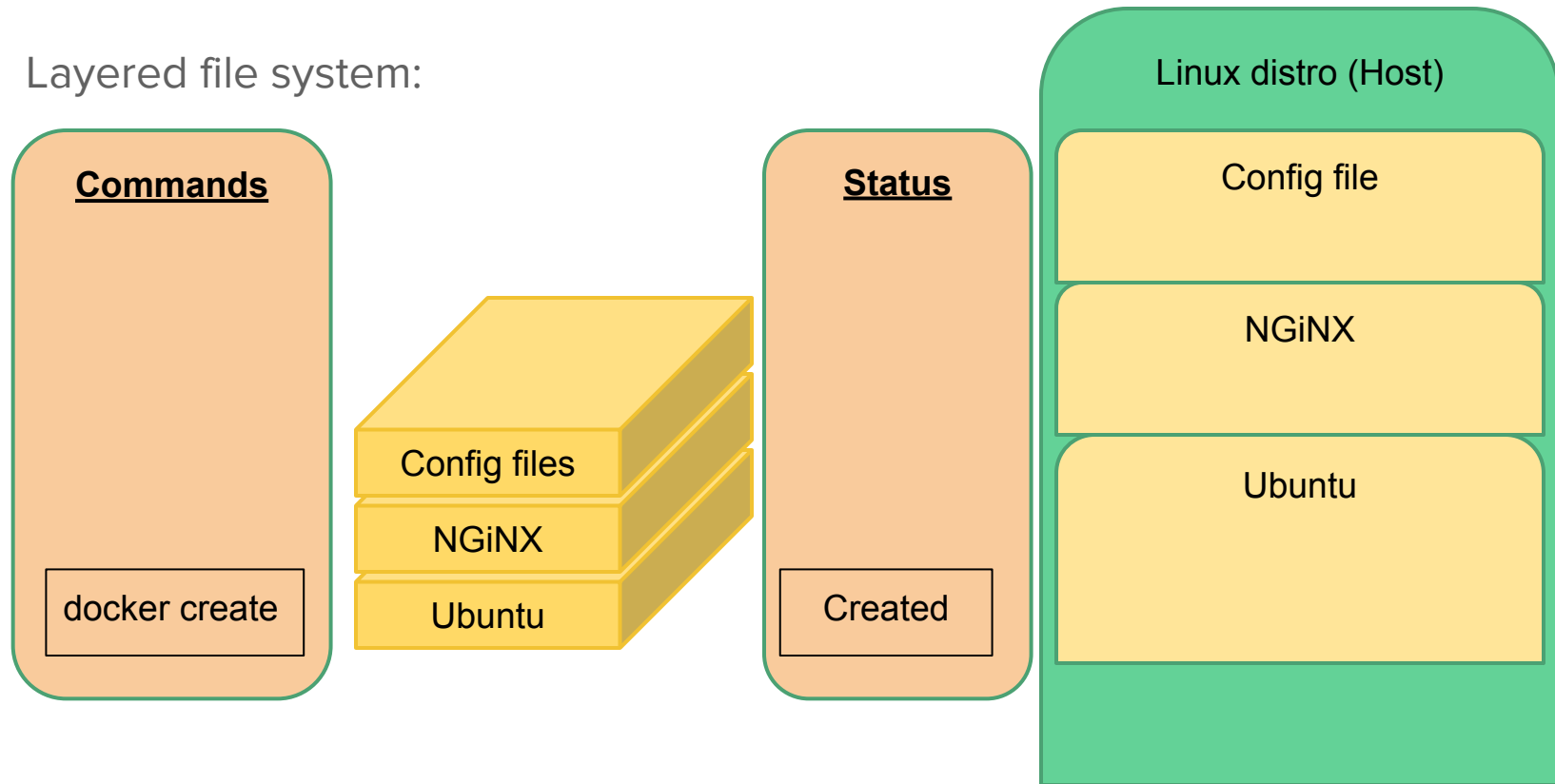
Config file

NGiNX

Ubuntu

What is Docker?

Layered file system:

Commands

| Config files |
| NGiNX |
| Ubuntu |

Status

Linux distro (Host)

| Config file |
| NGiNX |
| Ubuntu |

What is Docker?

Layered file system:

Commands

docker create

Config files
NGiNX
Ubuntu

Status

Created

Linux distro (Host)

Config file

NGiNX

Ubuntu

What is Docker?

Layered file system:

Commands

docker start/run

docker create

files updates

Config files

NGiNX

Ubuntu

Status

Up

Created

Linux distro (Host)

Config file

NGiNX

Ubuntu

What is Docker?

Layered file system:

Commands
Commands
docker stop/kill
docker start/run
docker create

files updates
Config files
NGiNX
Ubuntu

Status
Status
Exited
Up
Created

Linux distro (Host)

Config file
NGiNX
Ubuntu

What is Docker?

Layered file system:

Commands

docker rm

docker stop/kill

docker start/run

docker create

Config files
NGiNX
Ubuntu

Status

Exited

Up

Created

Linux distro (Host)

Config file

NGiNX

Ubuntu
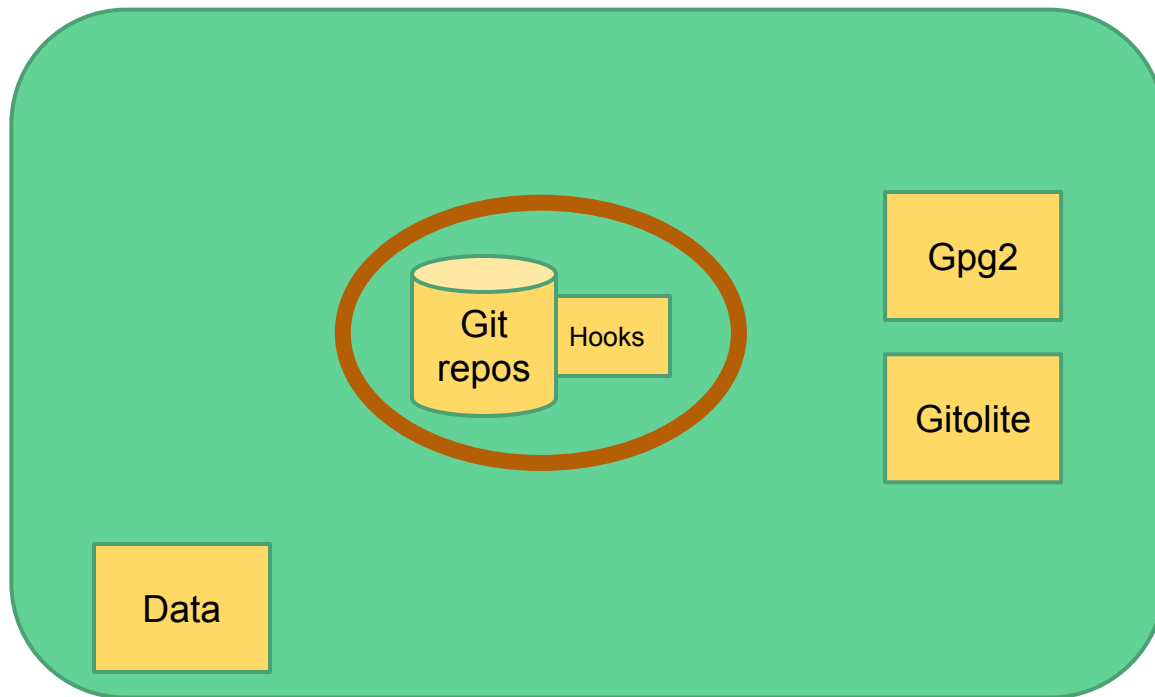
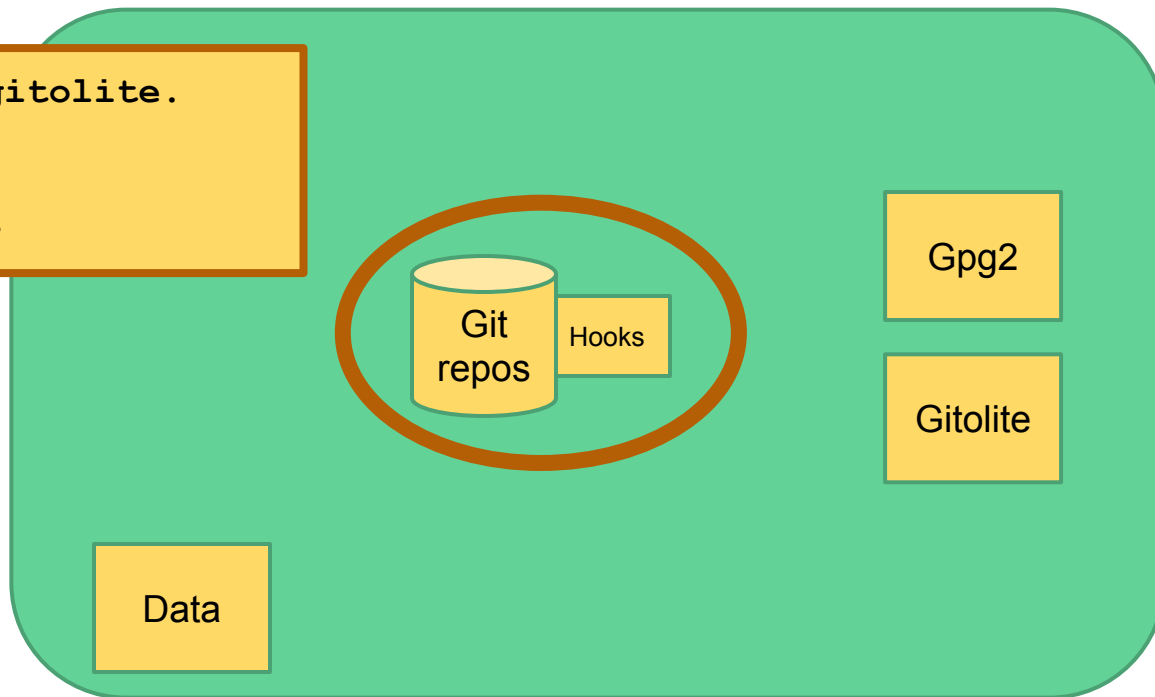# Docker rework

Data Container first

Docker rework

Data Container first

Docker rework

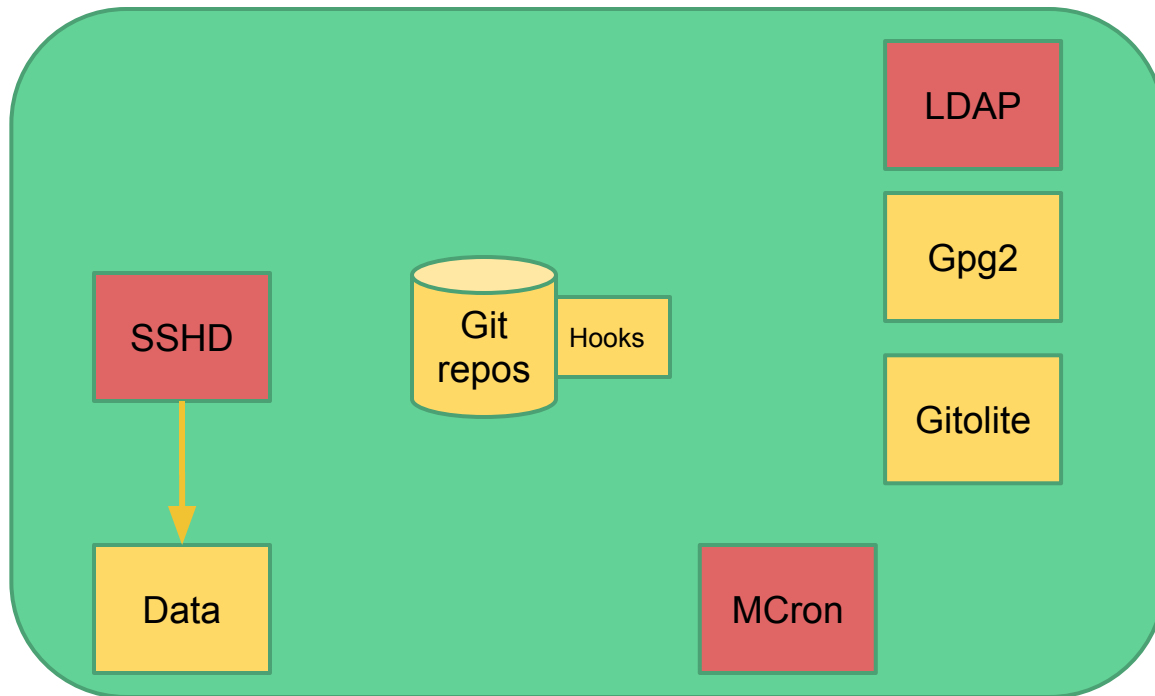Data Container first


```
docker create --name=gitolite.
repos.blessed.cont
gitolite.repos
/home/git/repositories
```
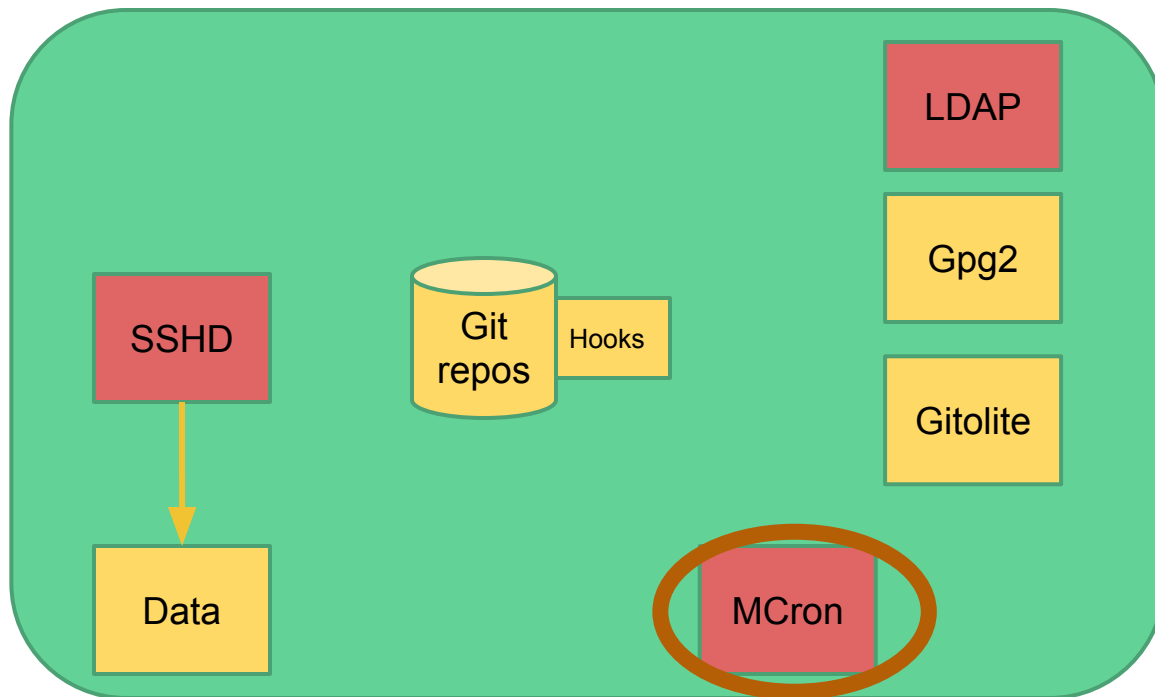
Git repos

Hooks

Gpg2

Gitolite

Data

Docker rework

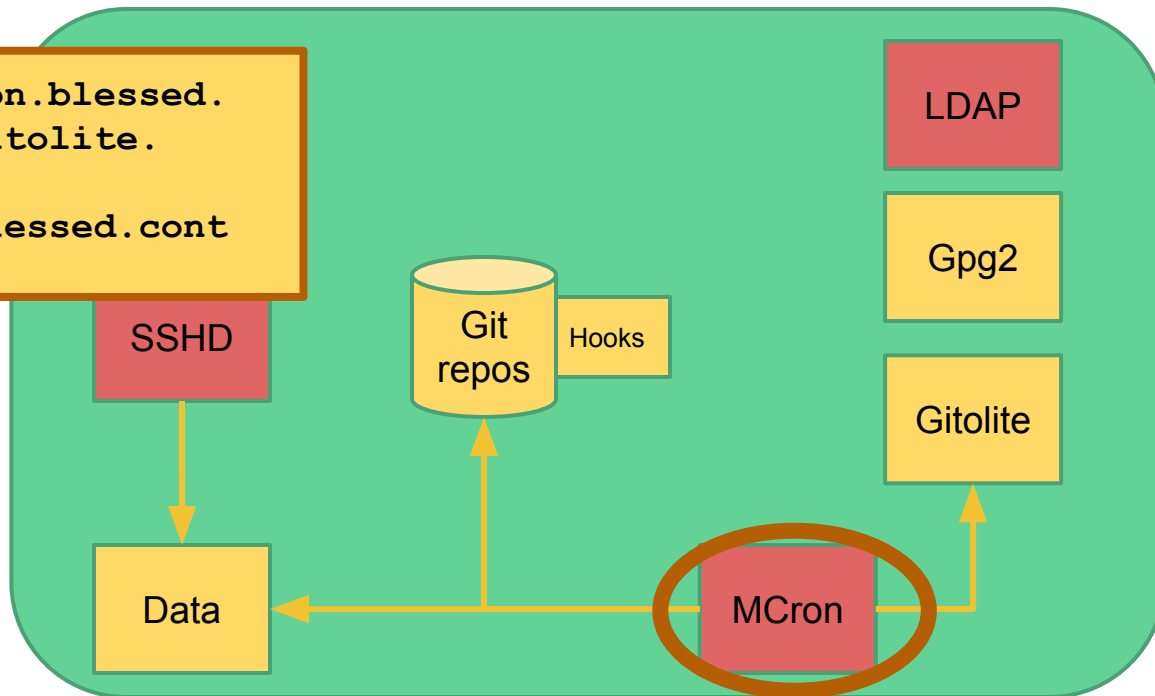Leaf containers second

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
```

LDAP

Gpg2

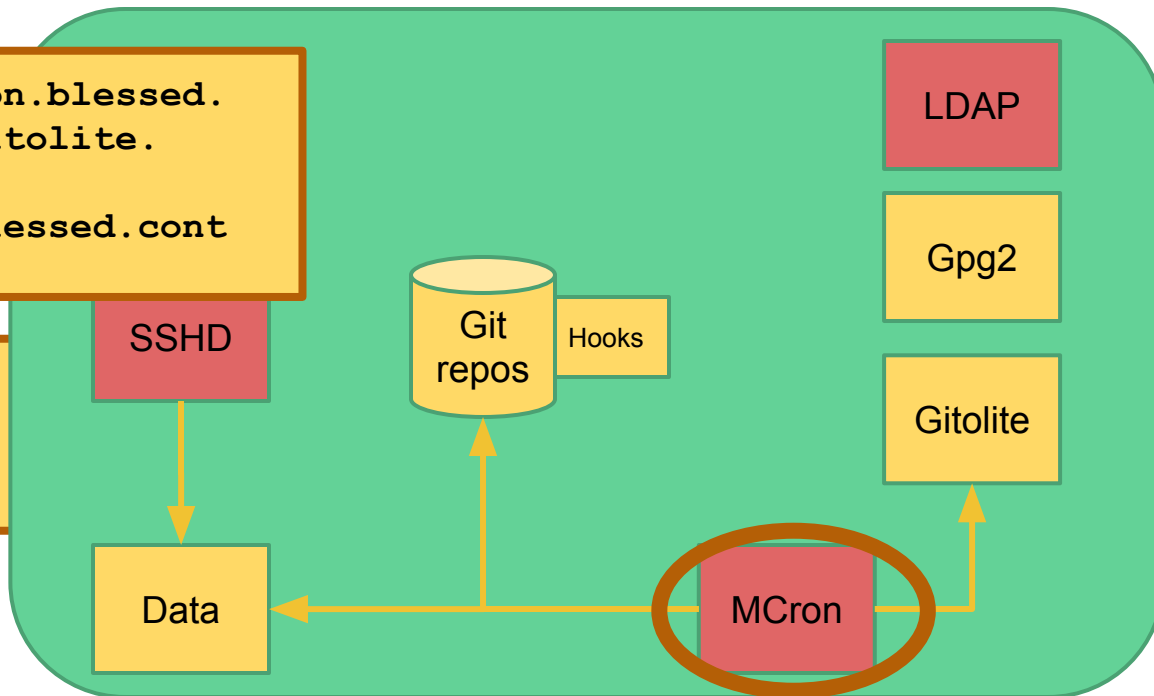SSHD

Git repos

Hooks

Gitolite
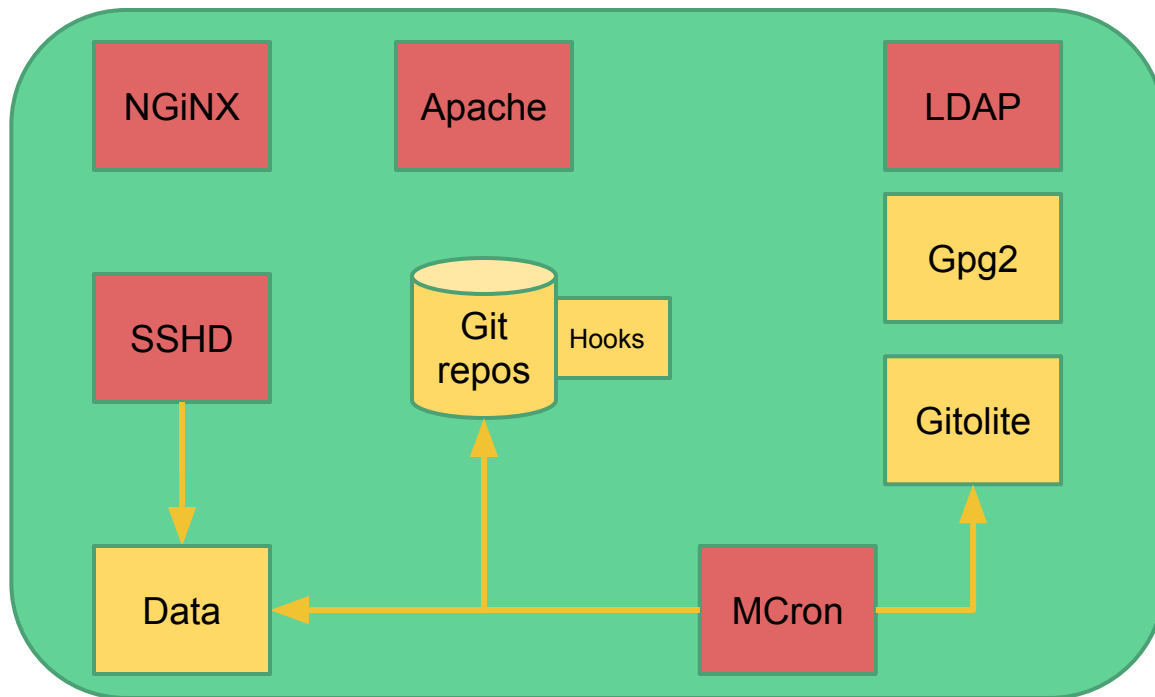
Data

MCron

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
```
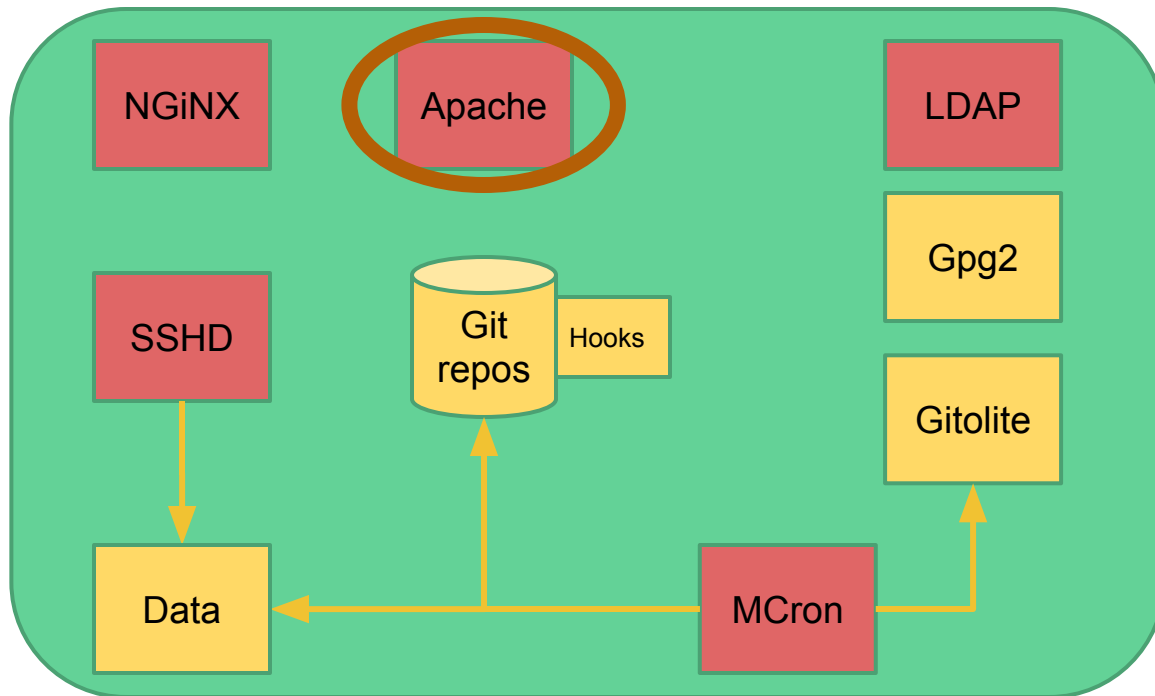
LDAP

Gpg2

Gitolite

SSHD

Git repos

Hooks

Data

MCron

Docker rework

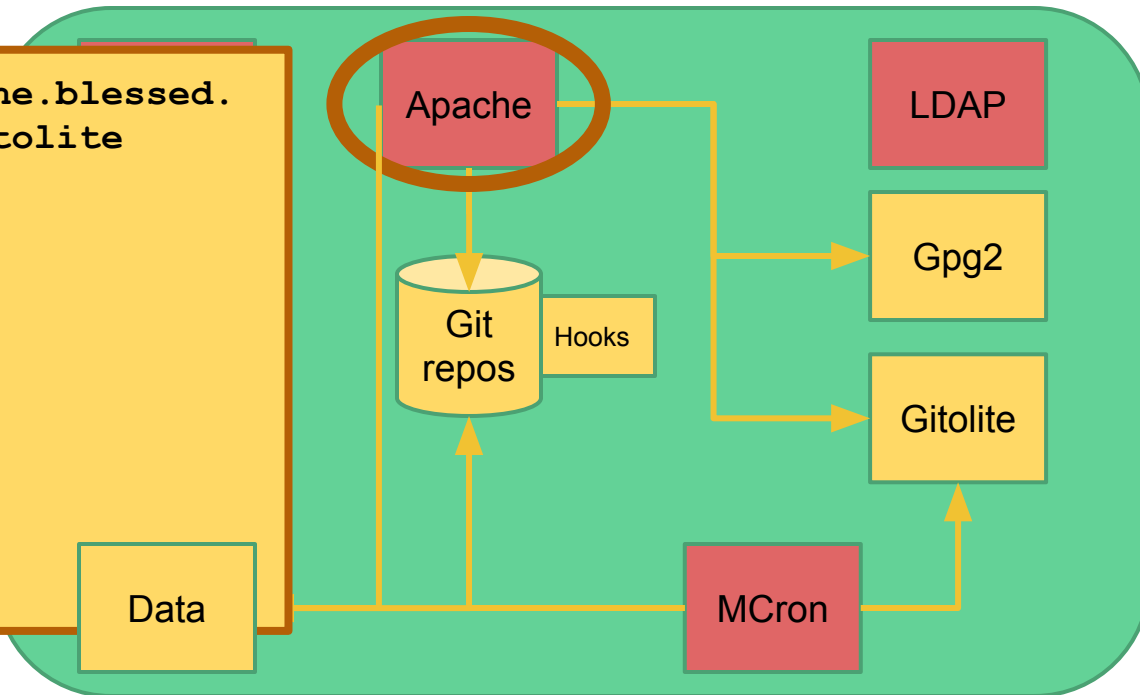Linked containers last

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
```

Apache

LDAP

Gpg2

Git repos

Hooks

Gitolite

Data

MCron

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
```

Apache

LDAP

Git repos
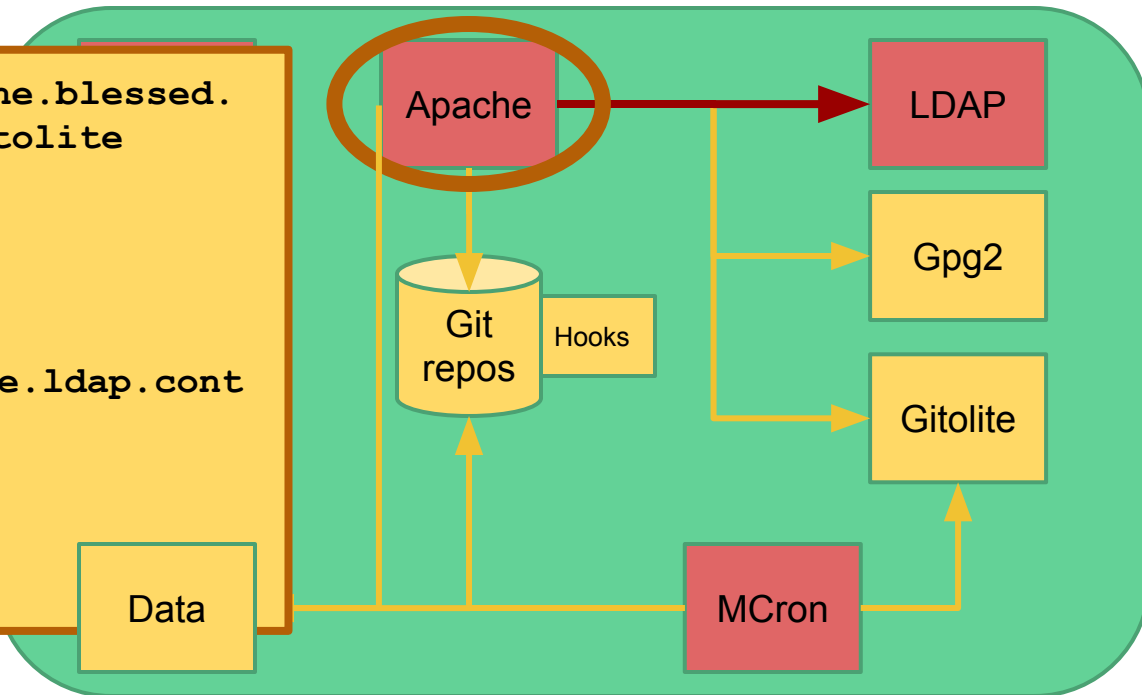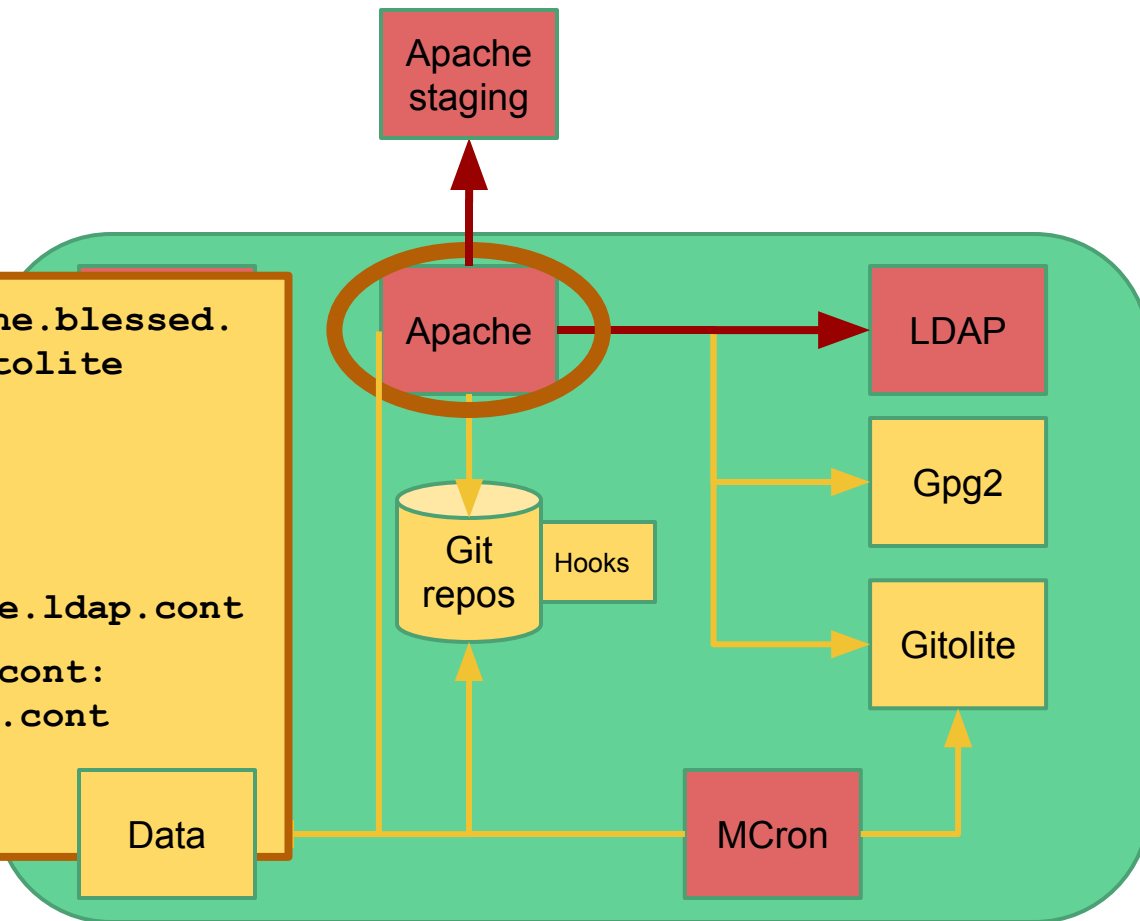
Hooks

Gpg2

Gitolite

Data

MCron

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
```
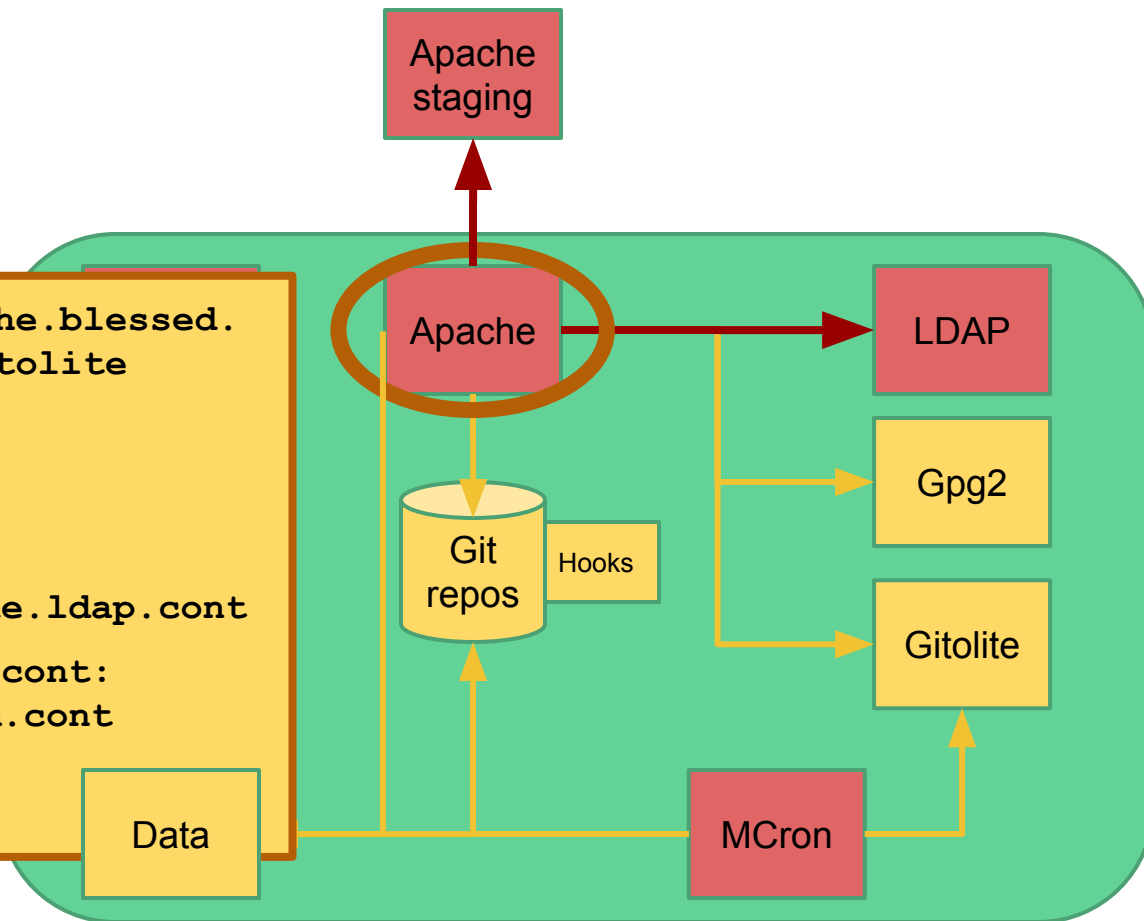
Docker rework

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
```
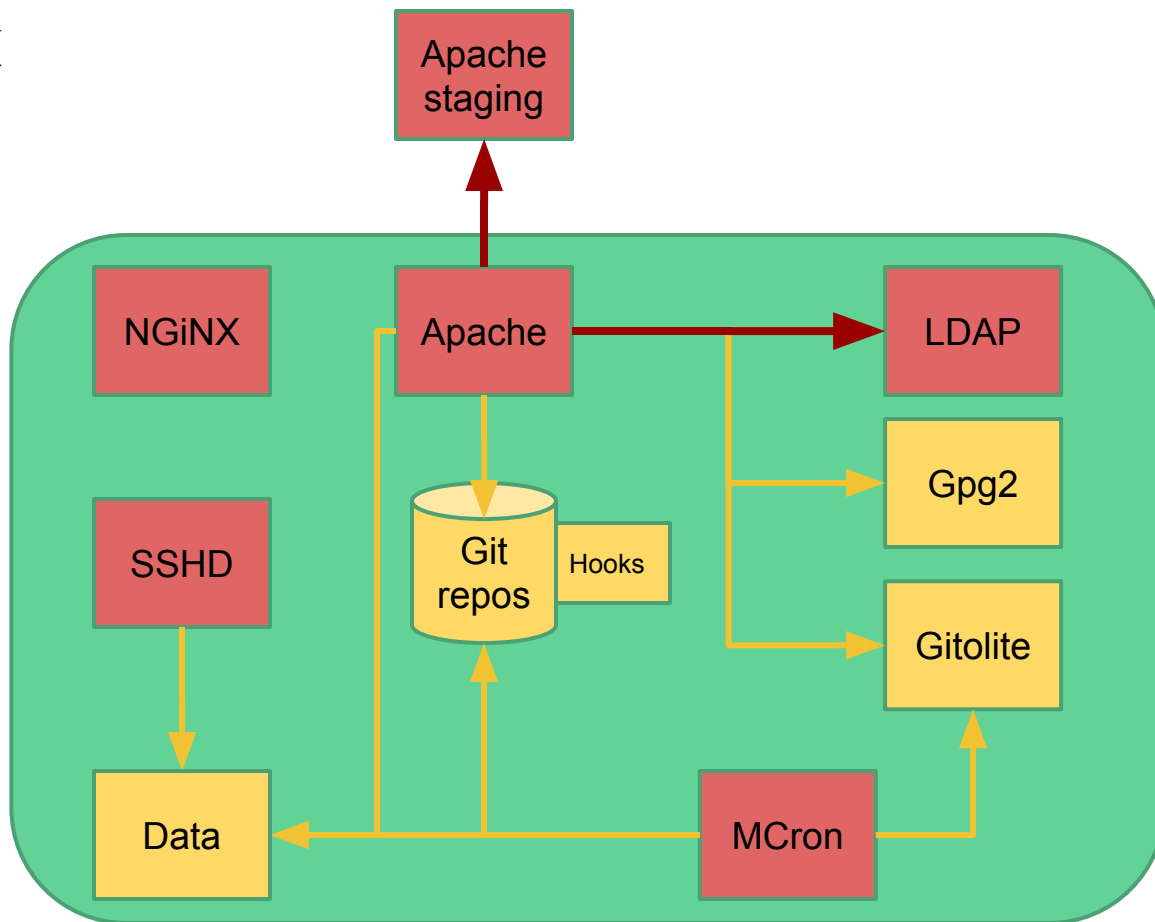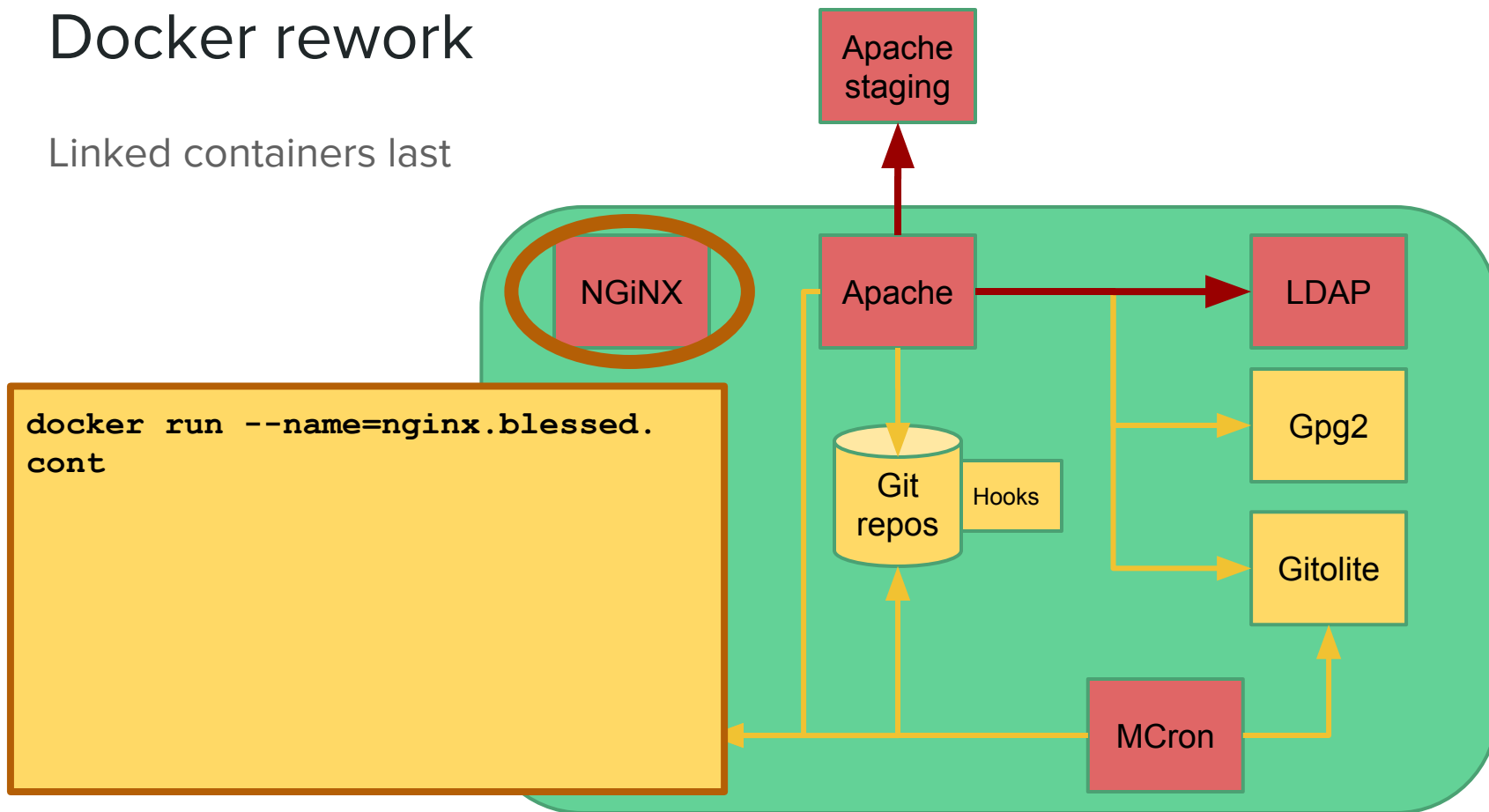
Apache
staging

Apache

LDAP

Git
repos

Hooks

Gpg2

Gitolite

Data

MCron

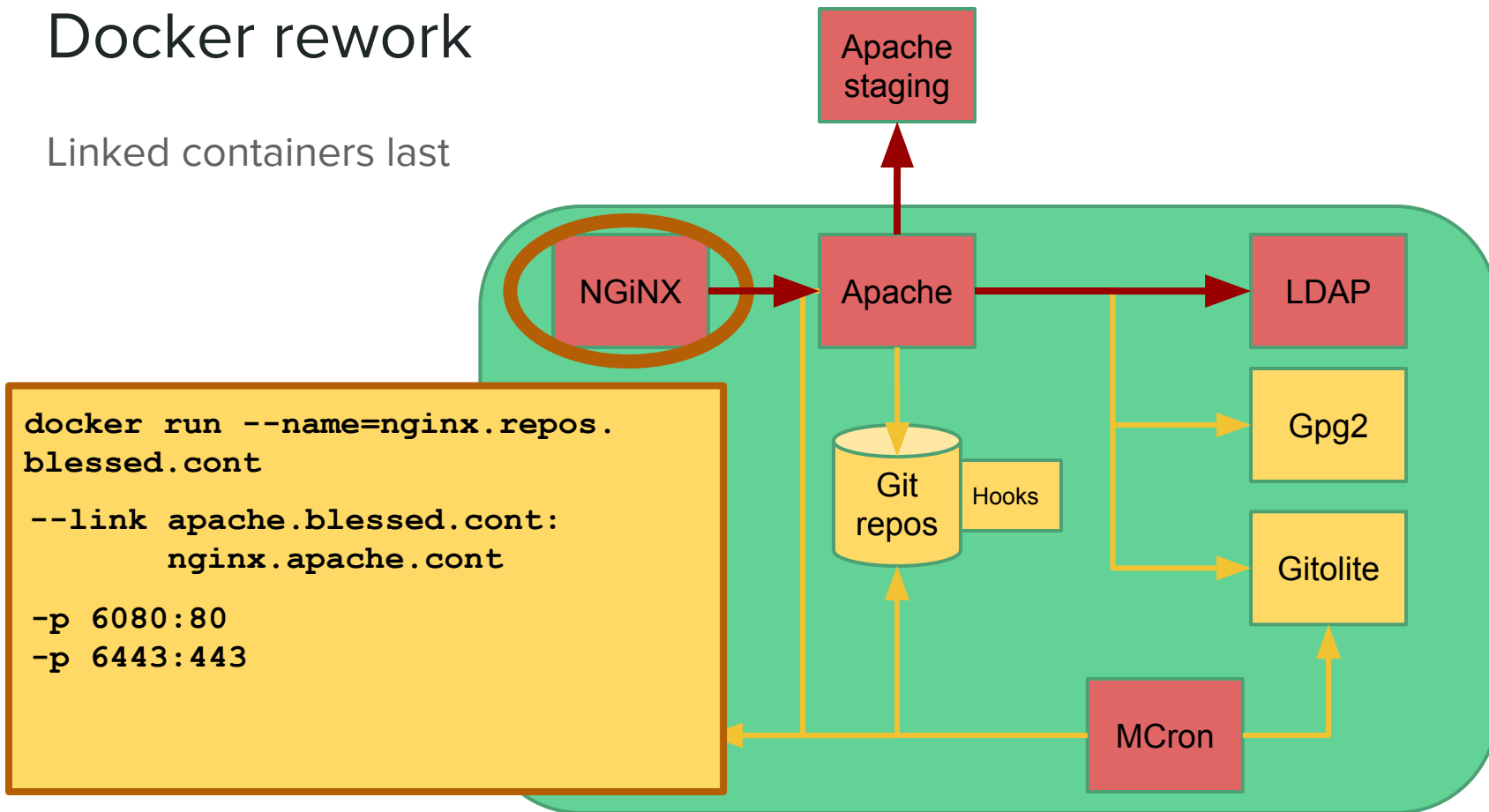# Docker rework

Linked containers last

Docker rework

Linked containers last

Apache
staging

NGiNX

Apache

LDAP

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

Git
repos

Hooks

Gpg2

Gitolite

MCron

Docker rework

Linked containers last

Apache
staging

NGiNX → Apache → LDAP

Git
repos   Hooks

Gpg2

Gitolite

MCron

```
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
```

Apache staging

NGiNX → Apache → LDAP

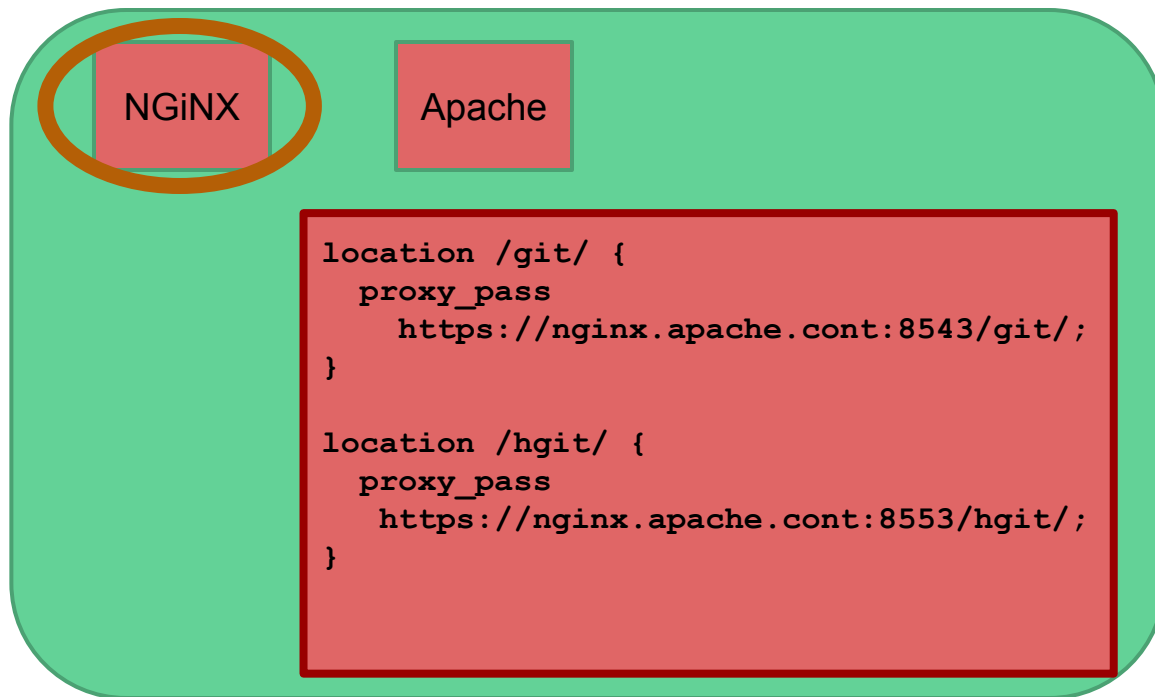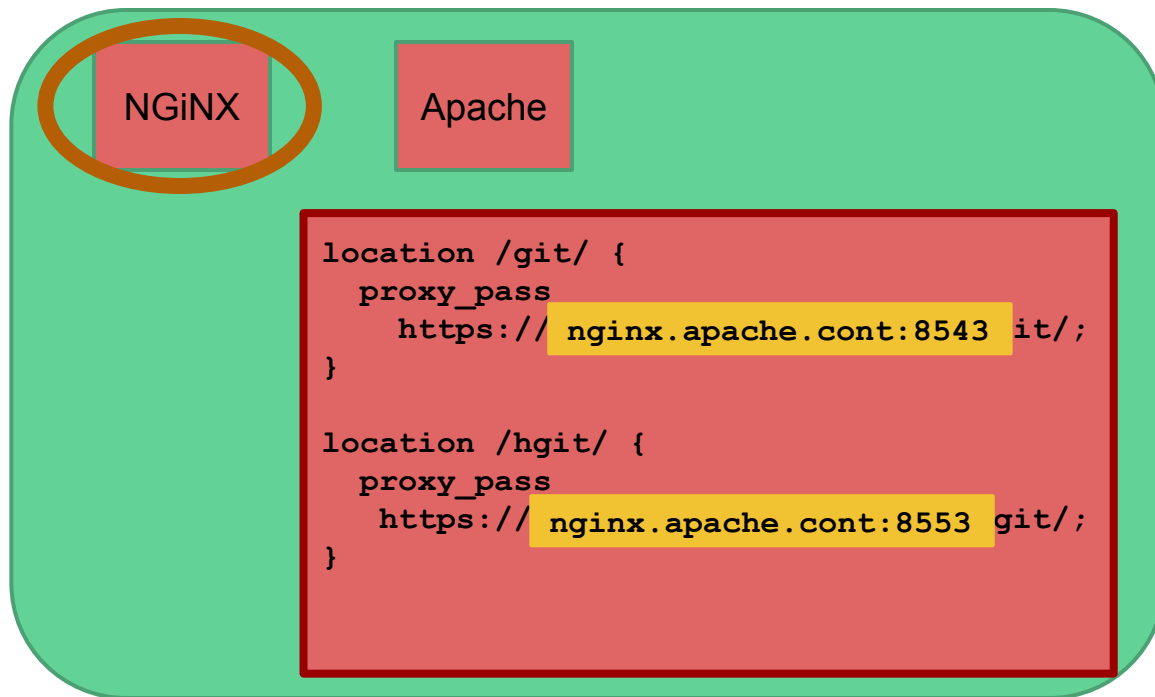Git repos   Hooks
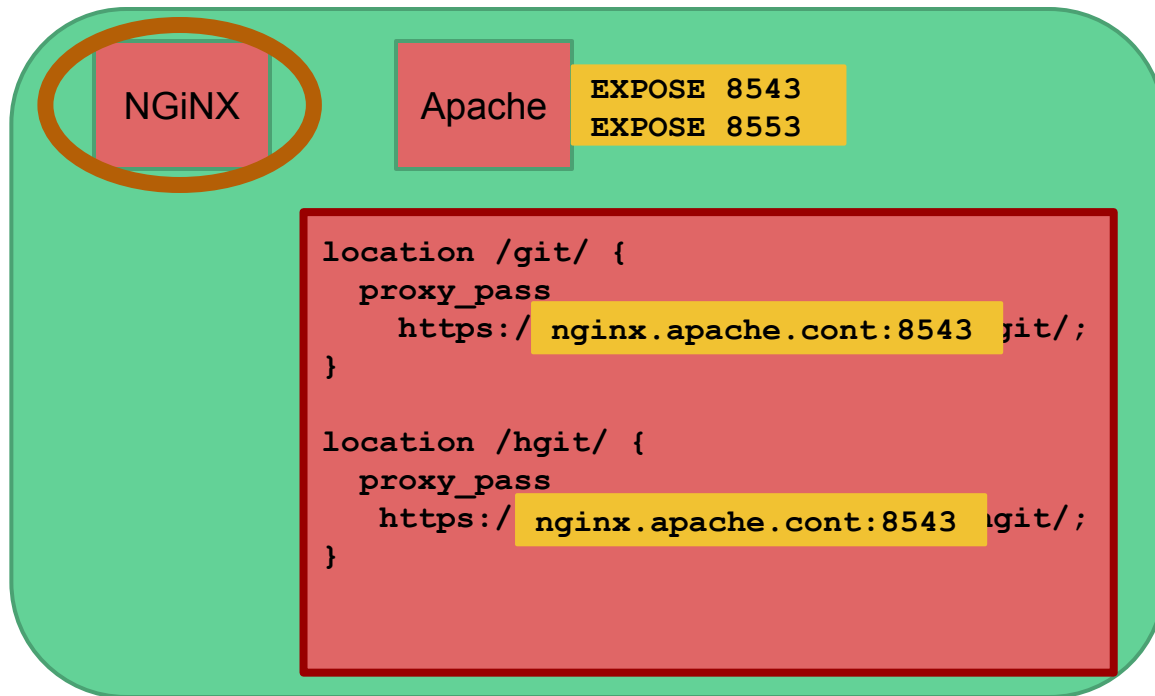
Gpg2

Gitolite

MCron

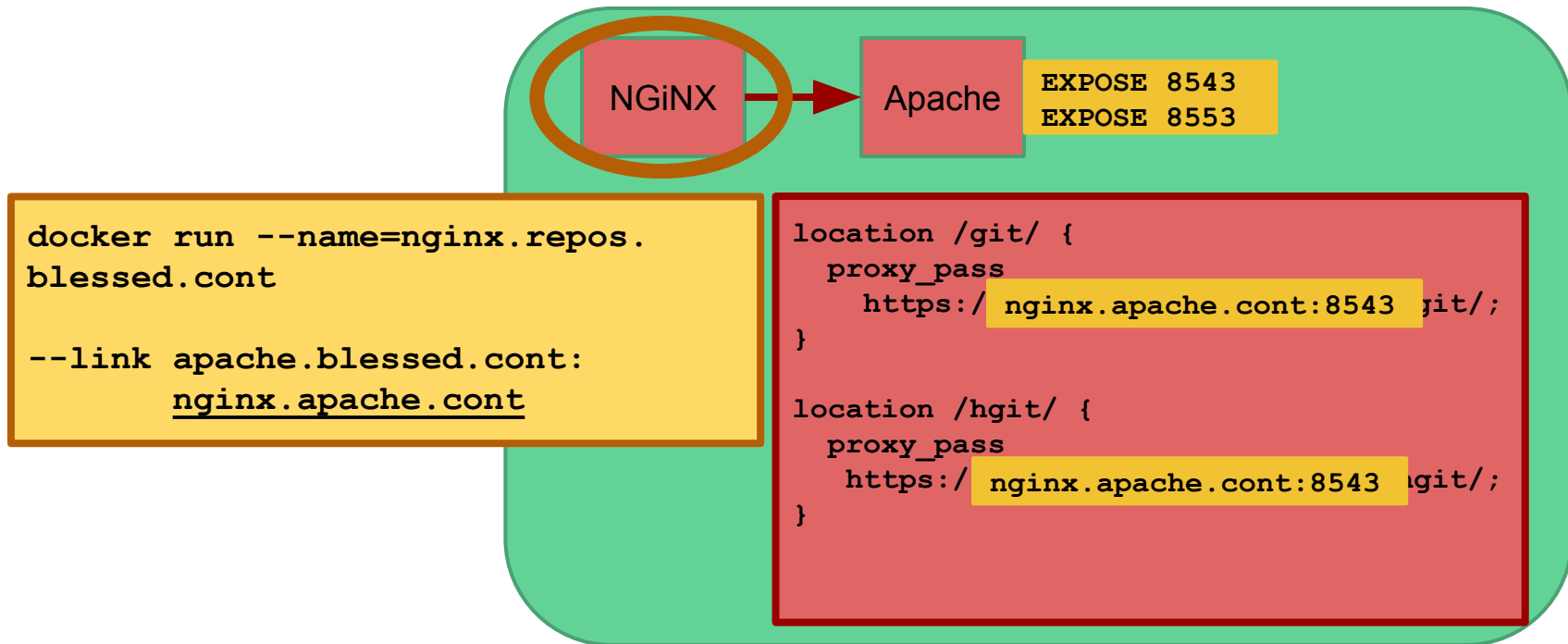# Docker Advantages

Configuration

Docker Advantages

Configuration

Docker Advantages

Configuration

Docker Advantages

Configuration

Docker Advantages

Configuration

NGiNX → Apache

```
EXPOSE 8543
EXPOSE 8553
```

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

--link apache.blessed.cont:
        nginx.apache.cont
```

```
location /git/ {
  proxy_pass
    https:/ nginx.apache.cont:8543 git/;
}

location /hgit/ {
  proxy_pass
   https:/ nginx.apache.cont:8543 hgit/;
}
```
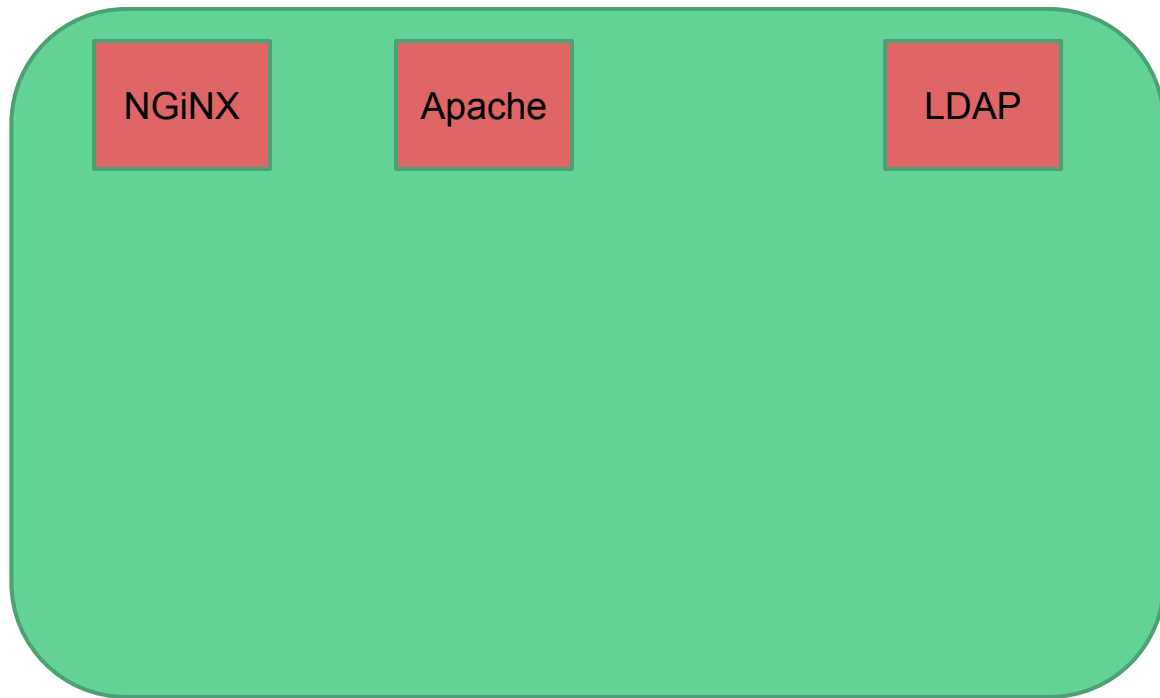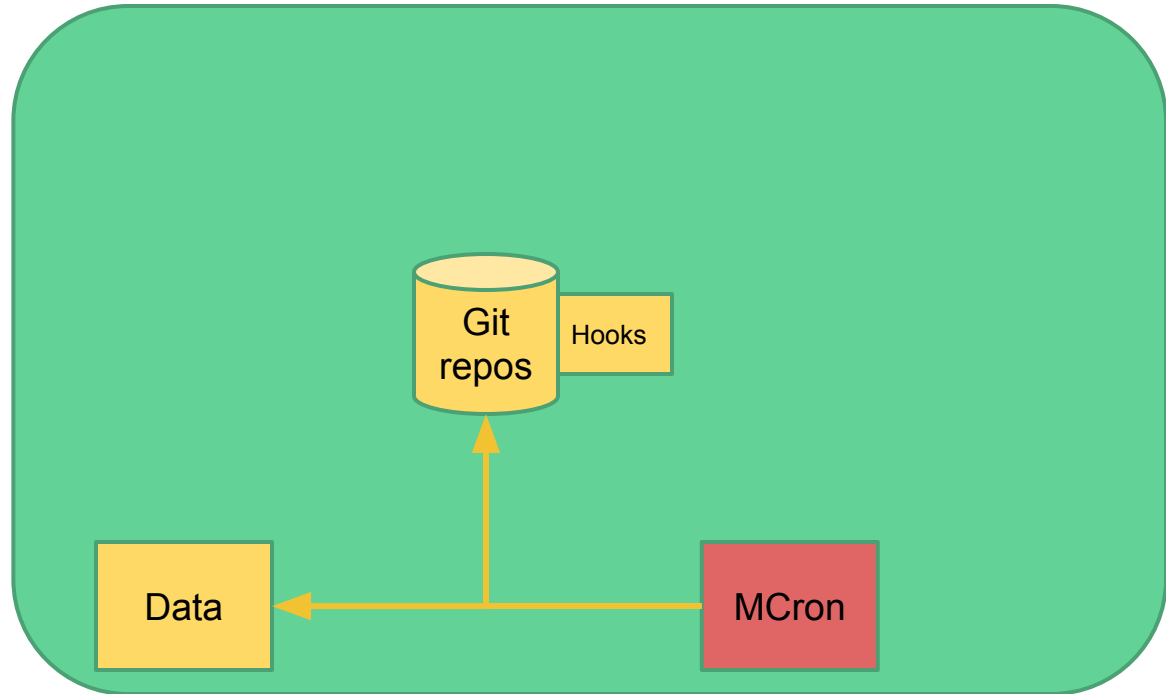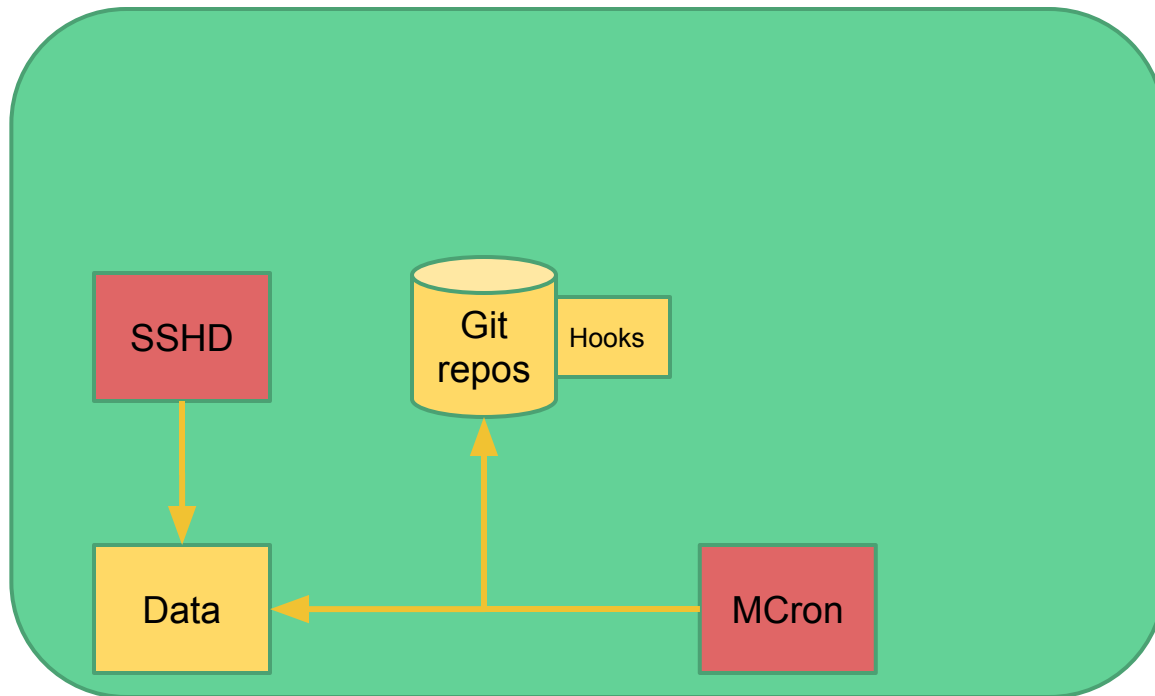
Docker Advantages

Isolation (services)

Docker Advantages

Isolation (services)

Docker Advantages

Isolation (services)

Docker Advantages
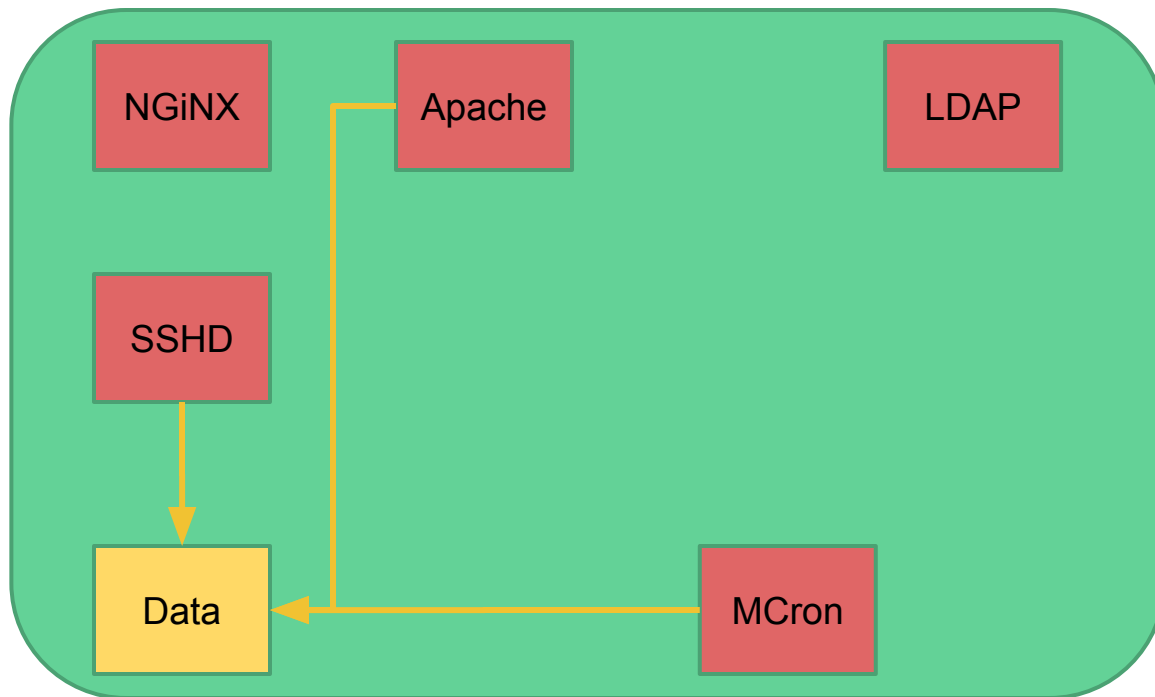
Isolation (services)

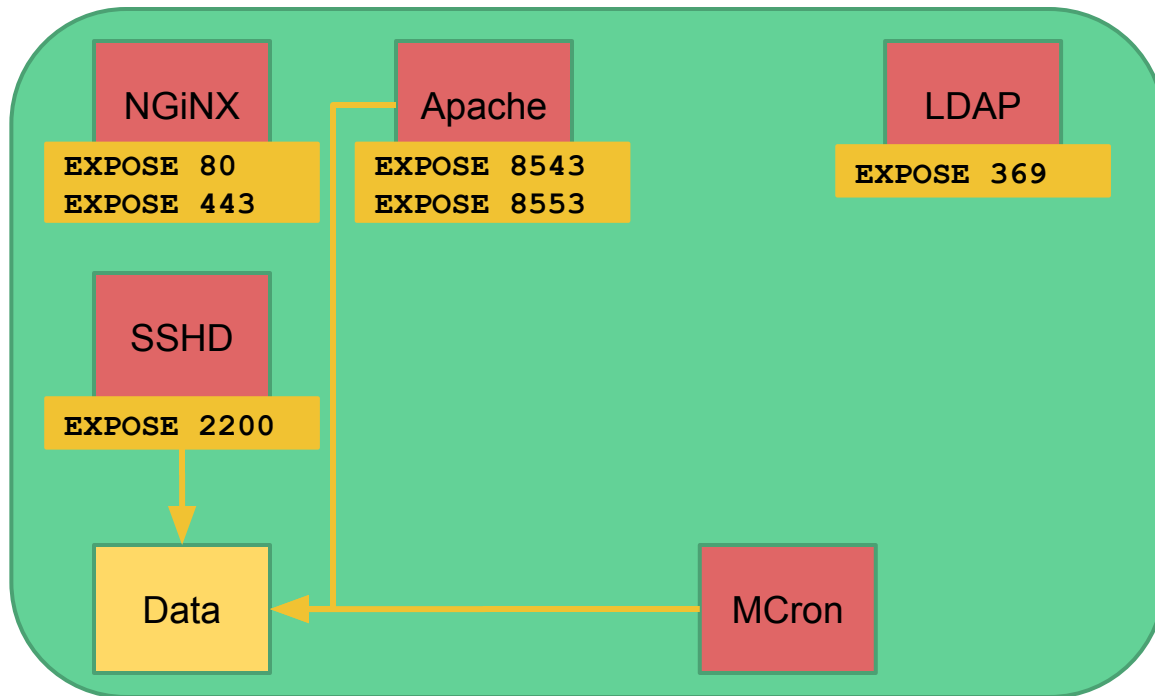# Docker Advantages

Isolation (services)

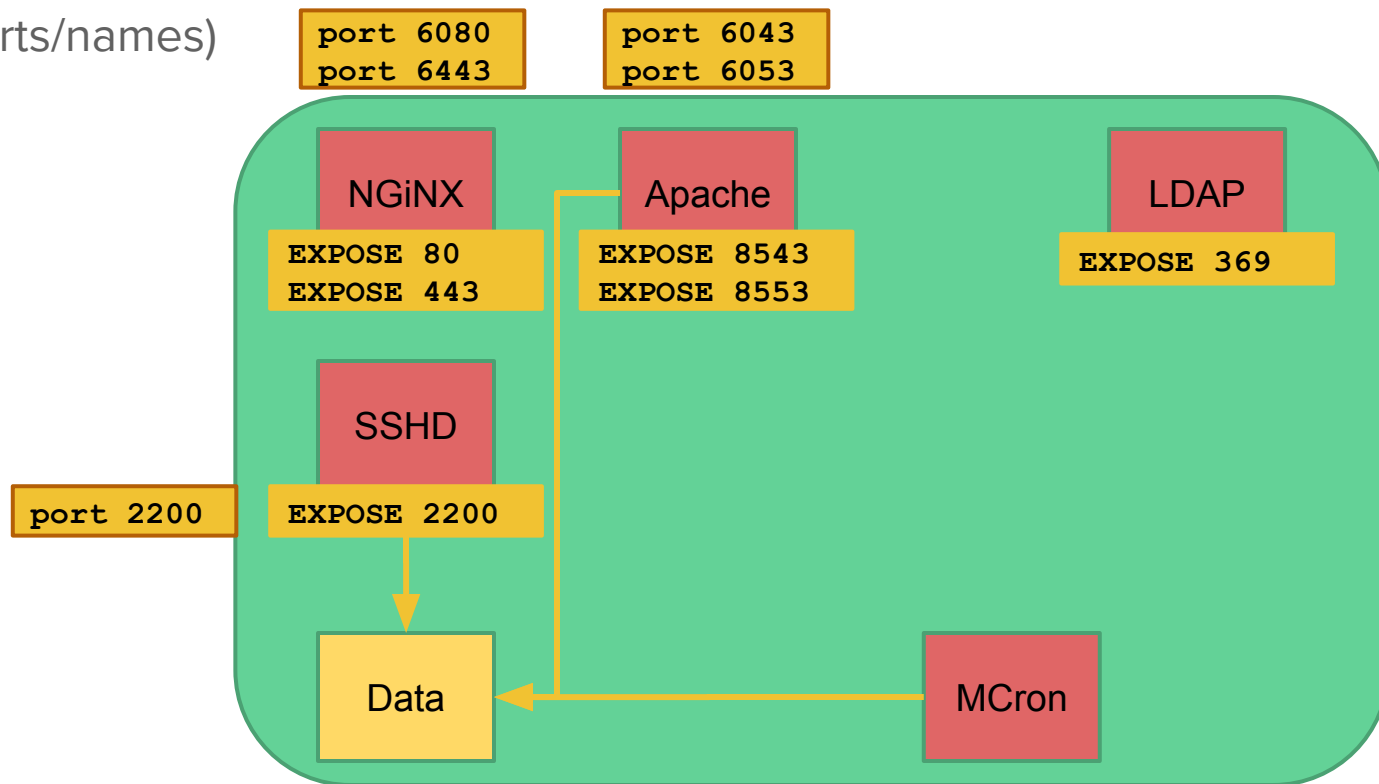# Docker Advantages

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.