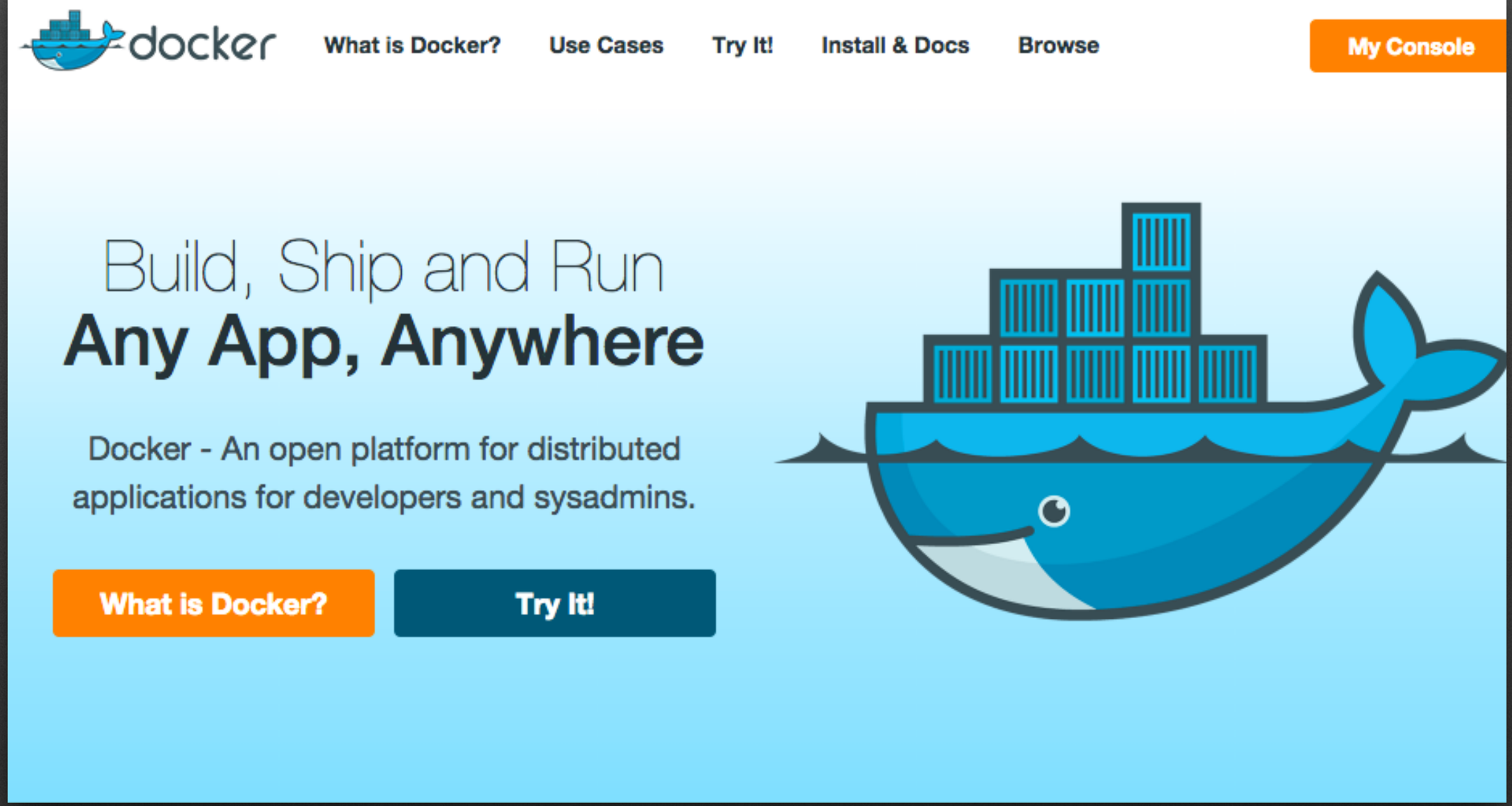
Introducing Docker

Nick McSpadden
Client Systems Manager
Schools of the Sacred Heart, SF

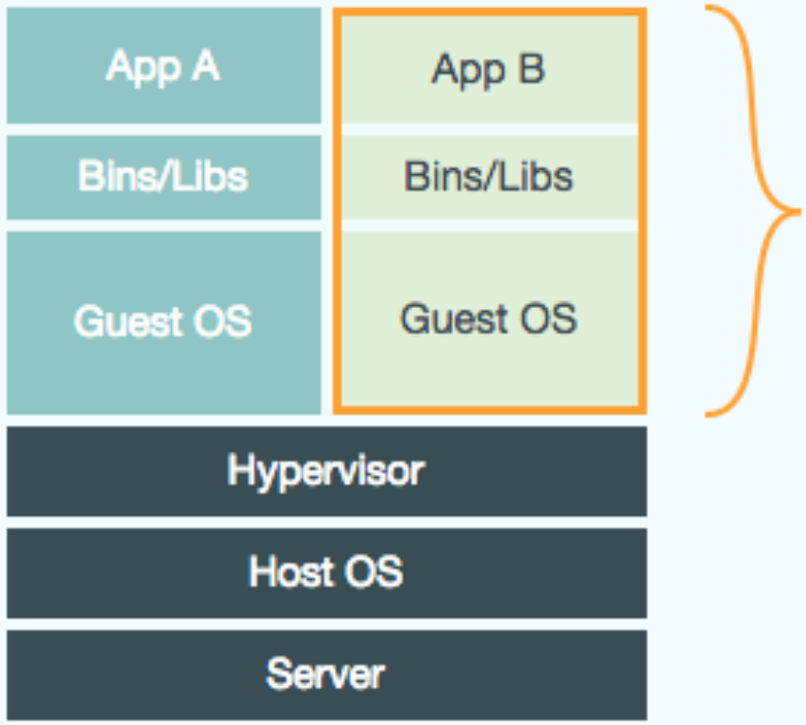


What the heck is Docker?

https://www.docker.com/whatisdocker/

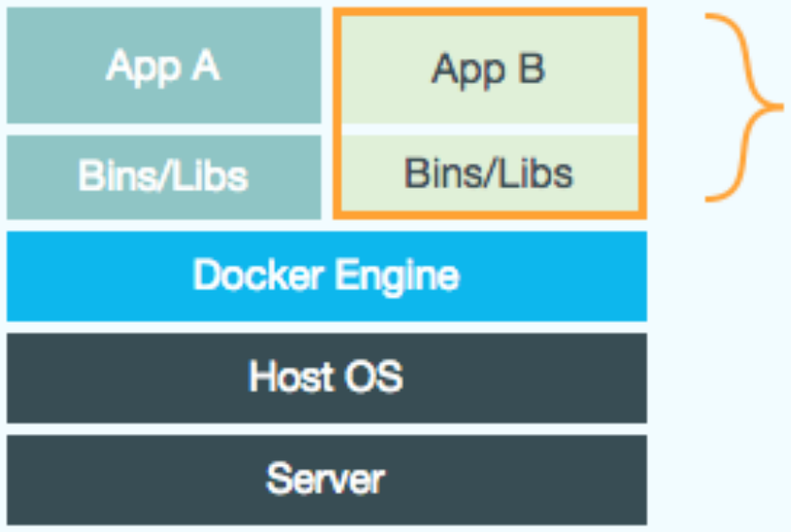
A tiny sandboxed minimal Linux environment.

Docker containers are designed to run a single service or application.



Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

Thinner than running a new VM for a service.

There are lots of Docker containers for basic services already available in the library:

Official Repositories

redis

ubuntu

The Official Ubuntu base image

WordPress

WordPress is a free and open source blogging tool and a content management system

MySQL

Popular open-source relational database management system

mongoDB

Document-oriented NoSQL database

CentOS

Official CentOS base image

NGINX

High performance reverse proxy server

Relational database management system

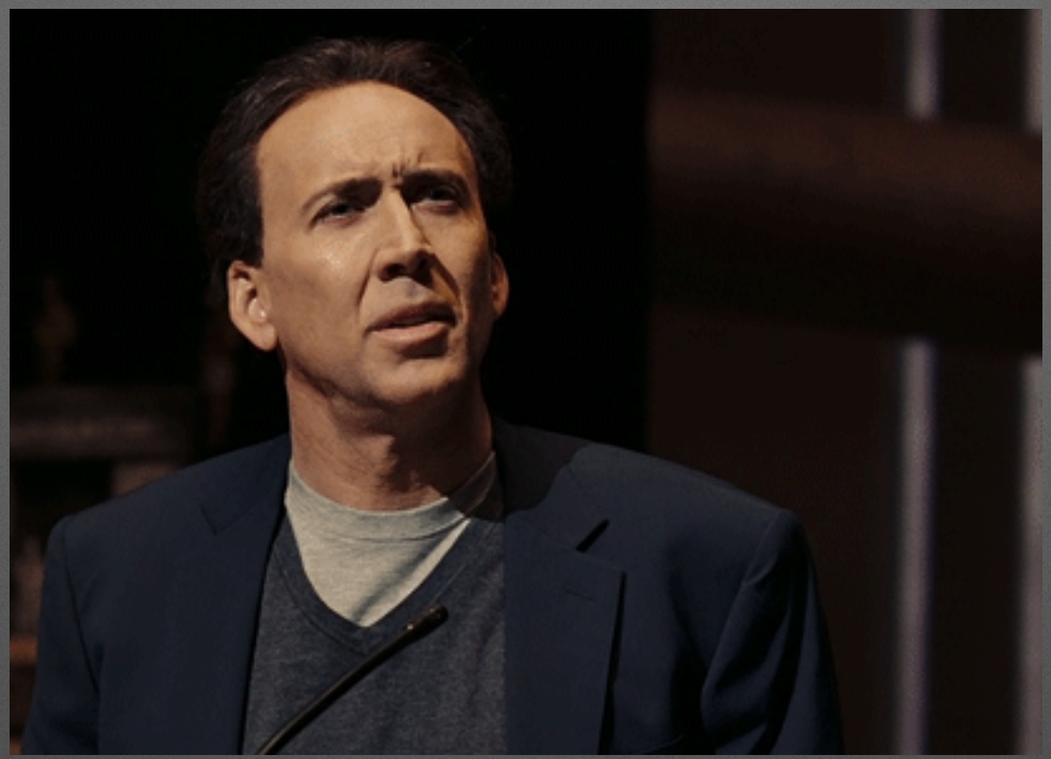
node.js

Node.js is a platform for scalable server-side and networking applications

Making containers is straightforward

- Containers are based on Dockerfiles, which are text files that contain a series of instructions.
- Start with a base image (like Ubuntu, Debian, or a default library image).
- ADD files, RUN commands.
- EXPOSE open ports for incoming connections.
- Share VOLUMEs to other containers.

```
FROM nginx

MAINTAINER Nick McSpadden <nmcspadden@gmail.com>

RUN mkdir -p /munki_repo
Run mkdir -p /etc/nginx/sites-enabled/

ADD nginx.conf /etc/nginx/nginx.conf
ADD munki-repo.conf /etc/nginx/sites-enabled/

VOLUME /munki_repo
EXPOSE 80
```

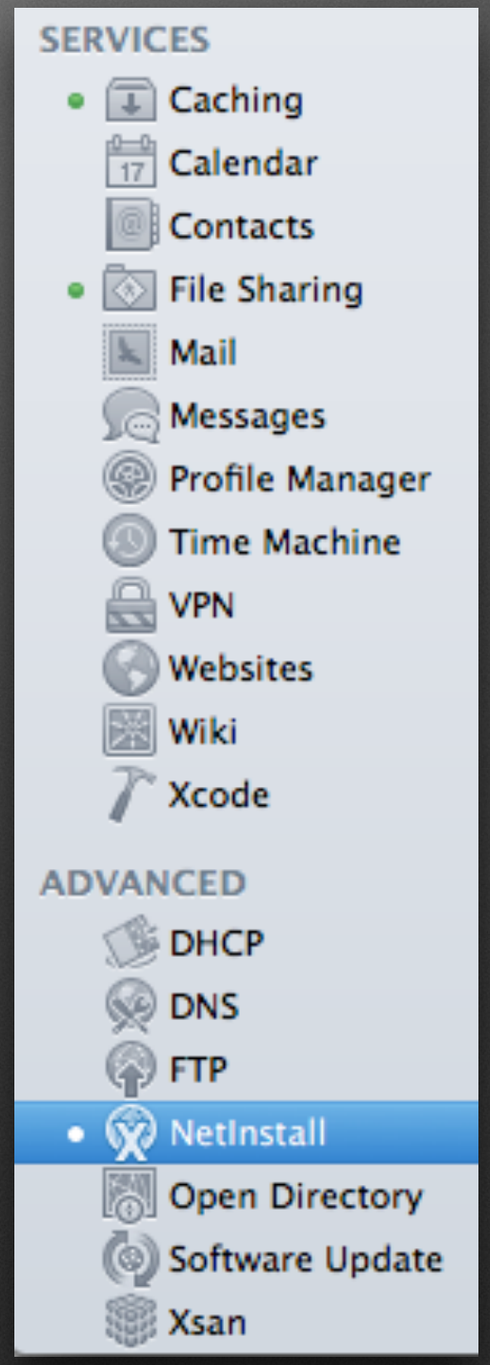


What’s with all this Linux crap? How does this involve Macs?

Look at OS X Server.app

Not many services that are truly unique to OS X:

- Caching Service (arguable vs. squid)
- Messages (no one uses this)
- Time Machine Server (no one uses this)
- Xcode (don't, it'll hurt)
- OpenDirectory (not common in enterprise)
- Xsan (wait, this still exists?)



You can run lots of neat Mac management tools without ever needing OS X Server, or OS X at all.

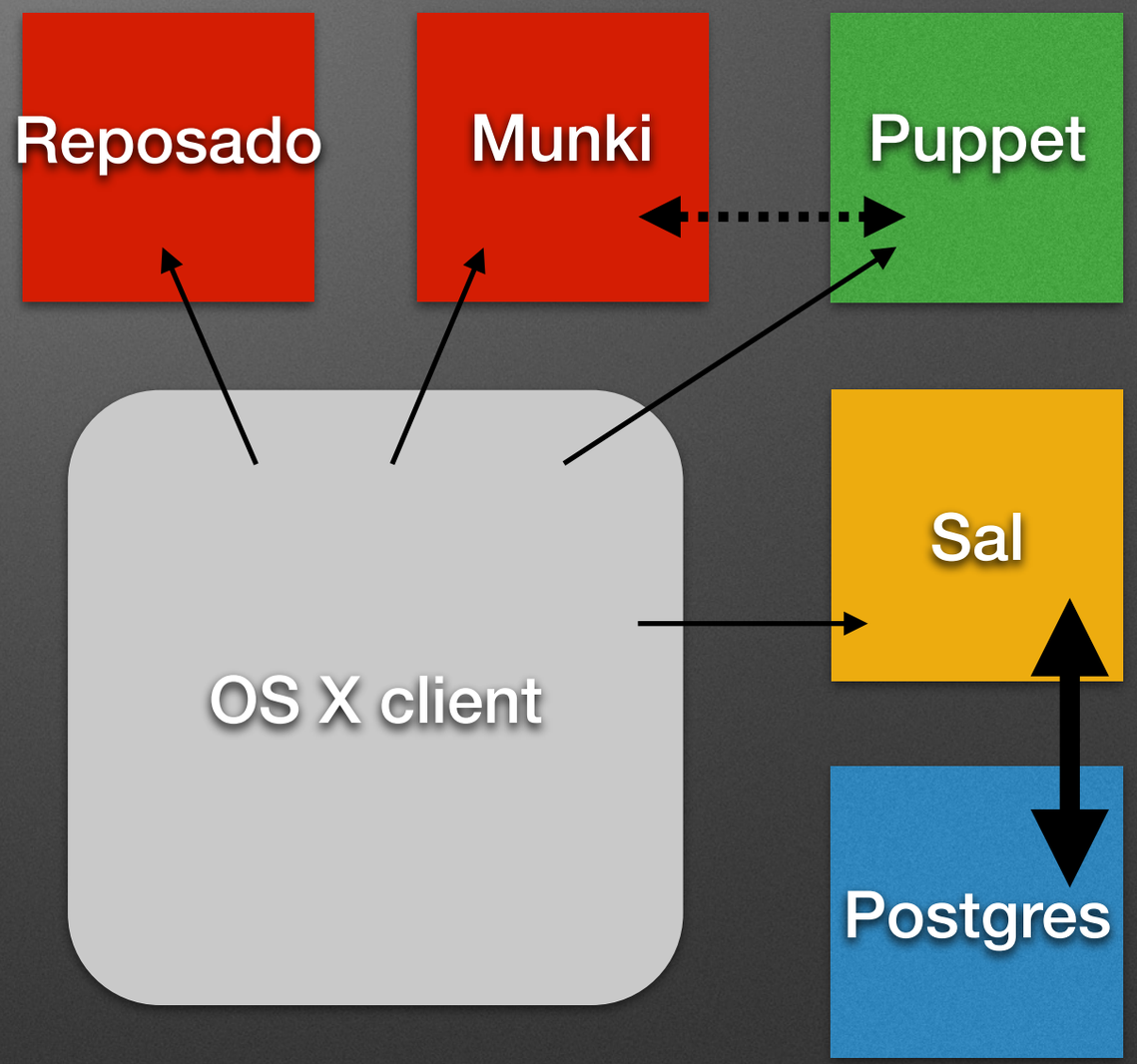
And if you can do it in Linux...

...you can do it in Docker!

Introducing Docker
…to Mac management!

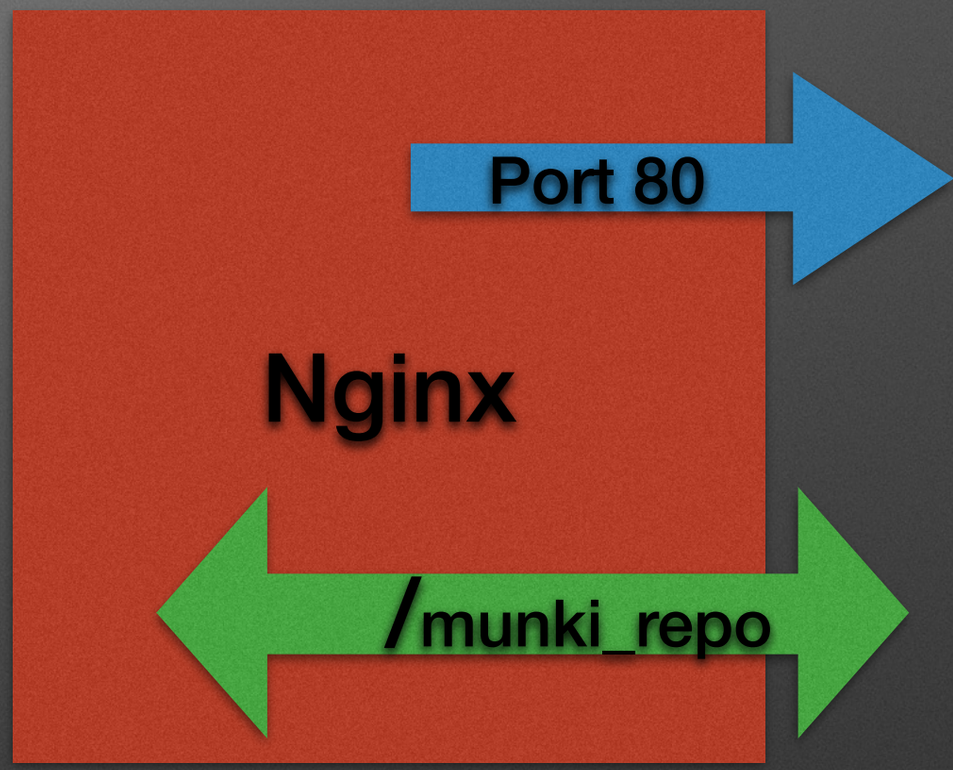
Demoing some Mac tools

- Munki: just a basic web server
- Reposado: just a basic web server, with Python
- Puppet: Just Puppet.
- Sal: Django, Python
- Databases: Postgres (for Sal)



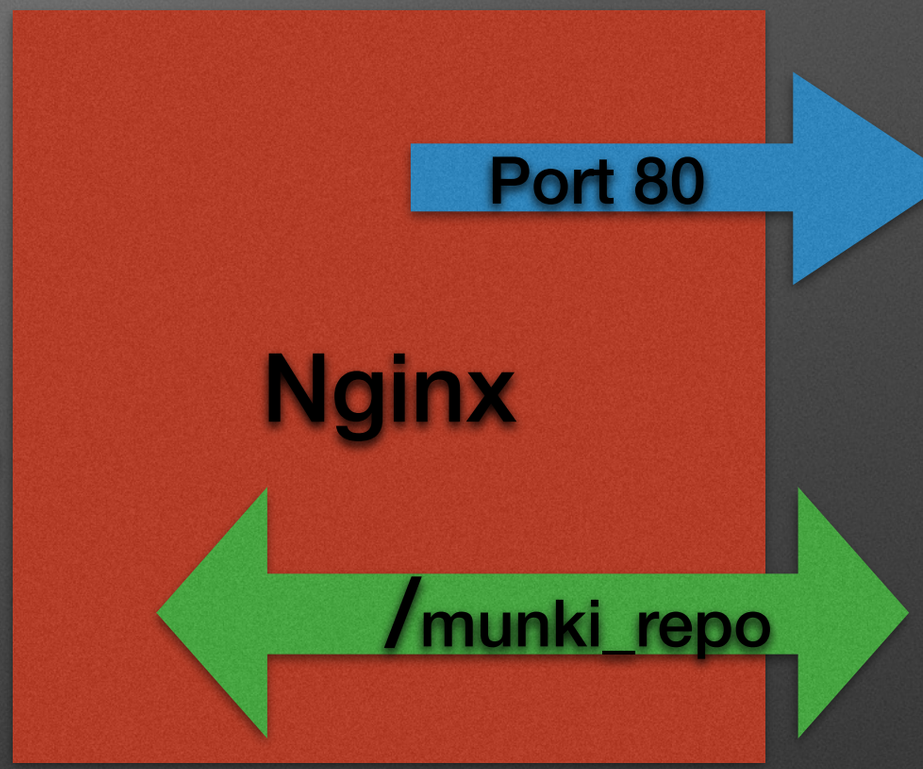
Anatomy of a Munki container

- This Munki container is just an Nginx web server.
- Port 80 is exposed for inbound connections (http).
- /munki_repo, where the Munki repo is stored, can be shared to other containers.



Anatomy of a Munki container - Dockerfile

```
FROM nginx

RUN mkdir -p /munki_repo

RUN mkdir -p /etc/nginx/sites-
enabled/

ADD nginx.conf /etc/nginx/nginx.conf

ADD munki-repo.conf /etc/nginx/
sites-enabled/

VOLUME /munki_repo

EXPOSE 80
```

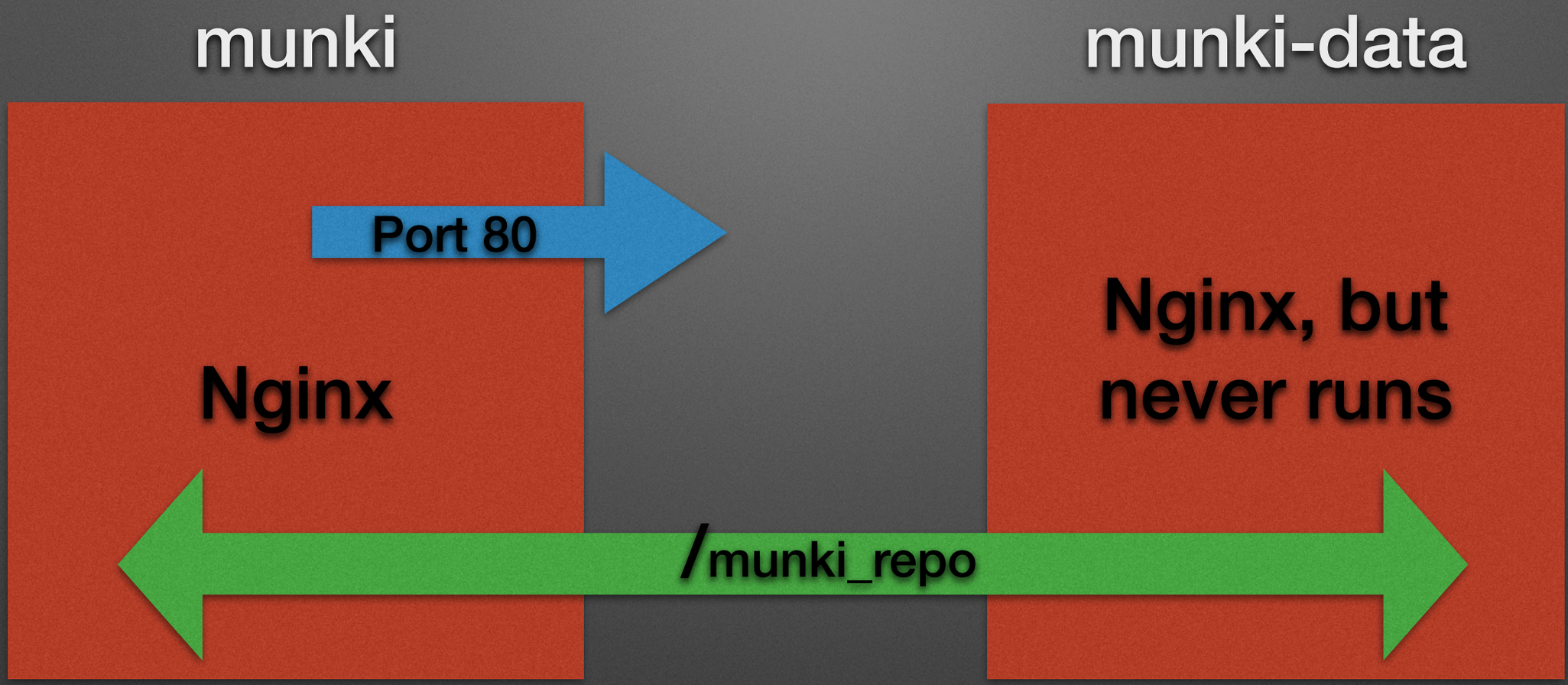


Run a Data-Only Container

```
docker run -d
        --name munki-data
        --entrypoint /bin/echo
        nmcspadden/munki
        "Data-only container for munki"
```

Data containers store data without running (or using any system resources), so other containers can access that data.

Munki + Munki-data

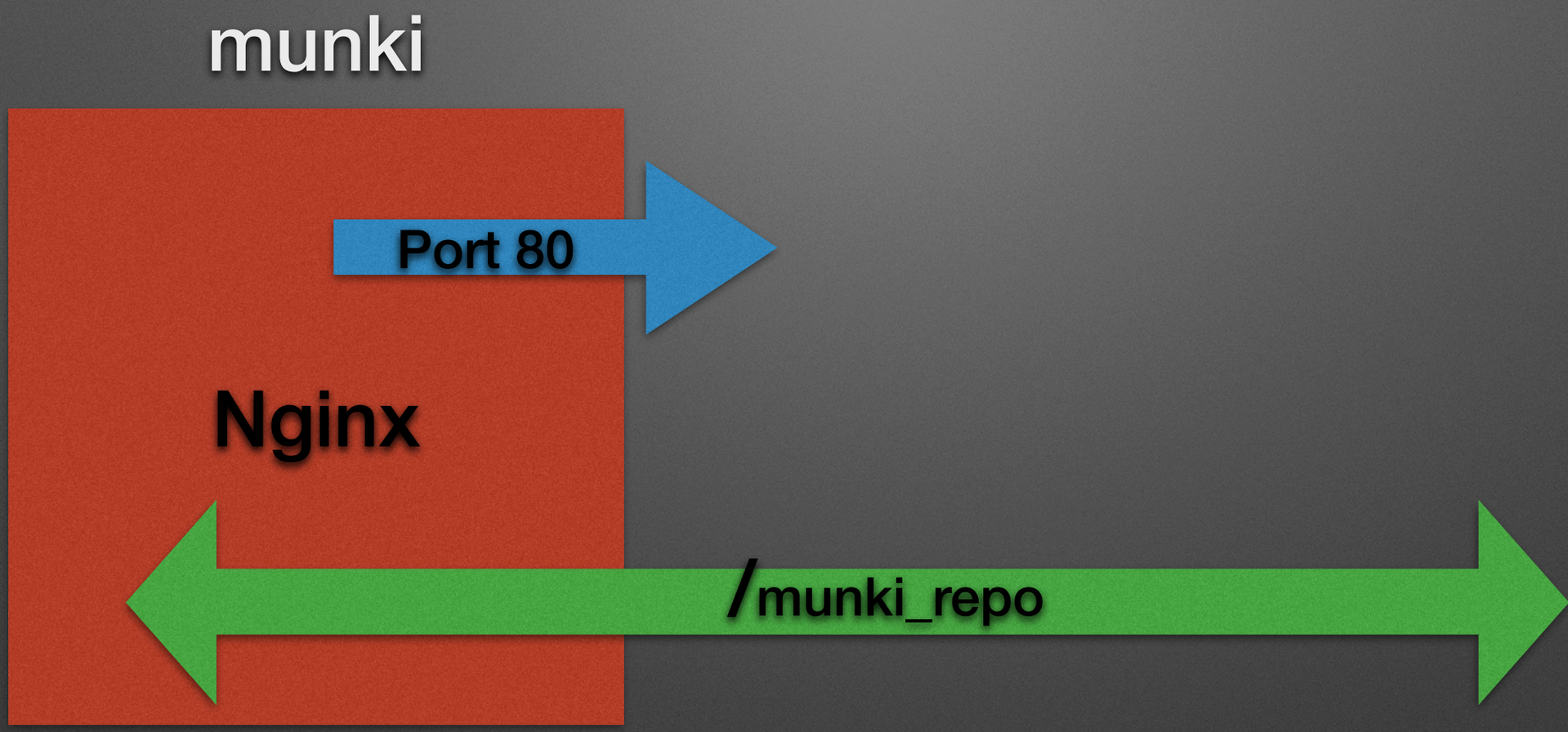


munki & munki-data both share /munki_repo.

Munki + Munki-data

If you remove munki, /munki_repo still exists, and contains the repo. You can create a new munki container and it will have the same old repo.

Munki + Munki-data



If you remove munki-data, /munki_repo is still there, just as if you remove munki.

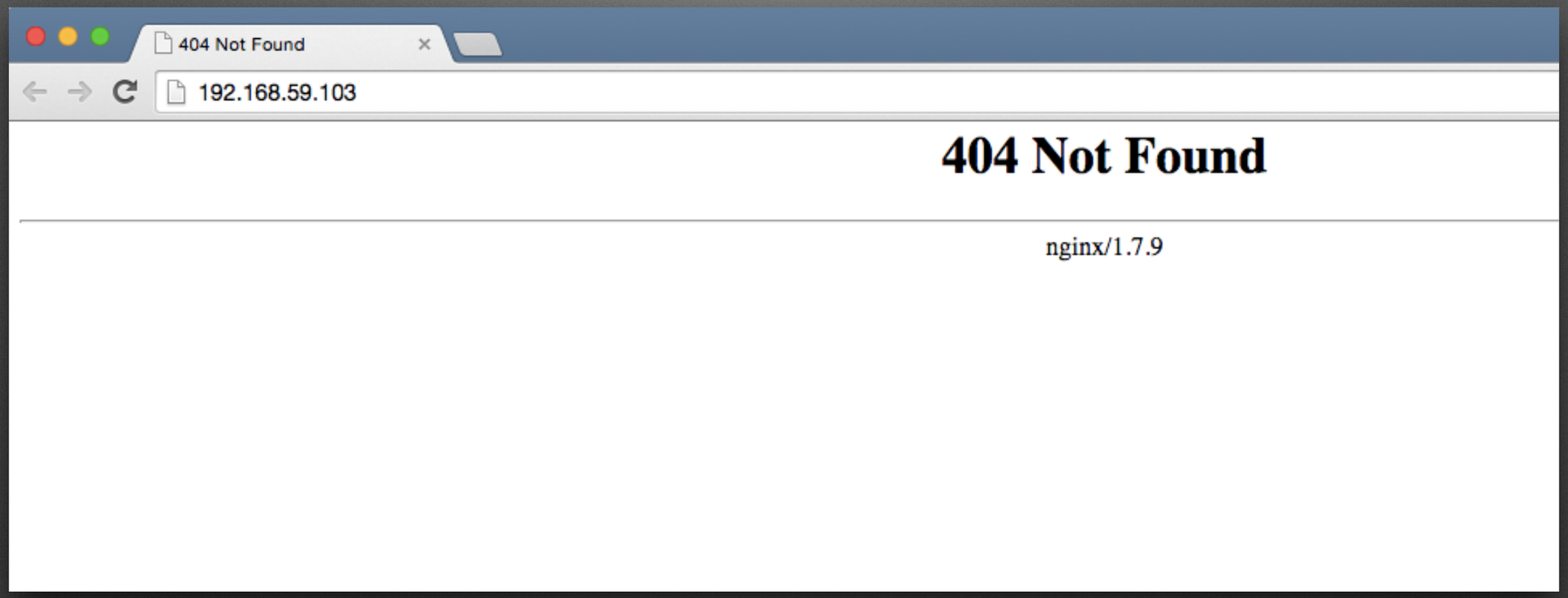
Munki + Munki-data

If you remove both containers, your data is gone.

Run the Munki Container

```
docker run -d
        --name munki
        --volumes-from munki-data
        --publish 80:80
        --hostname=“munki"
        nmcspadden/munki
```

We use --volumes-from to share the repo from the data container. Port 80 is open for inbound connections.

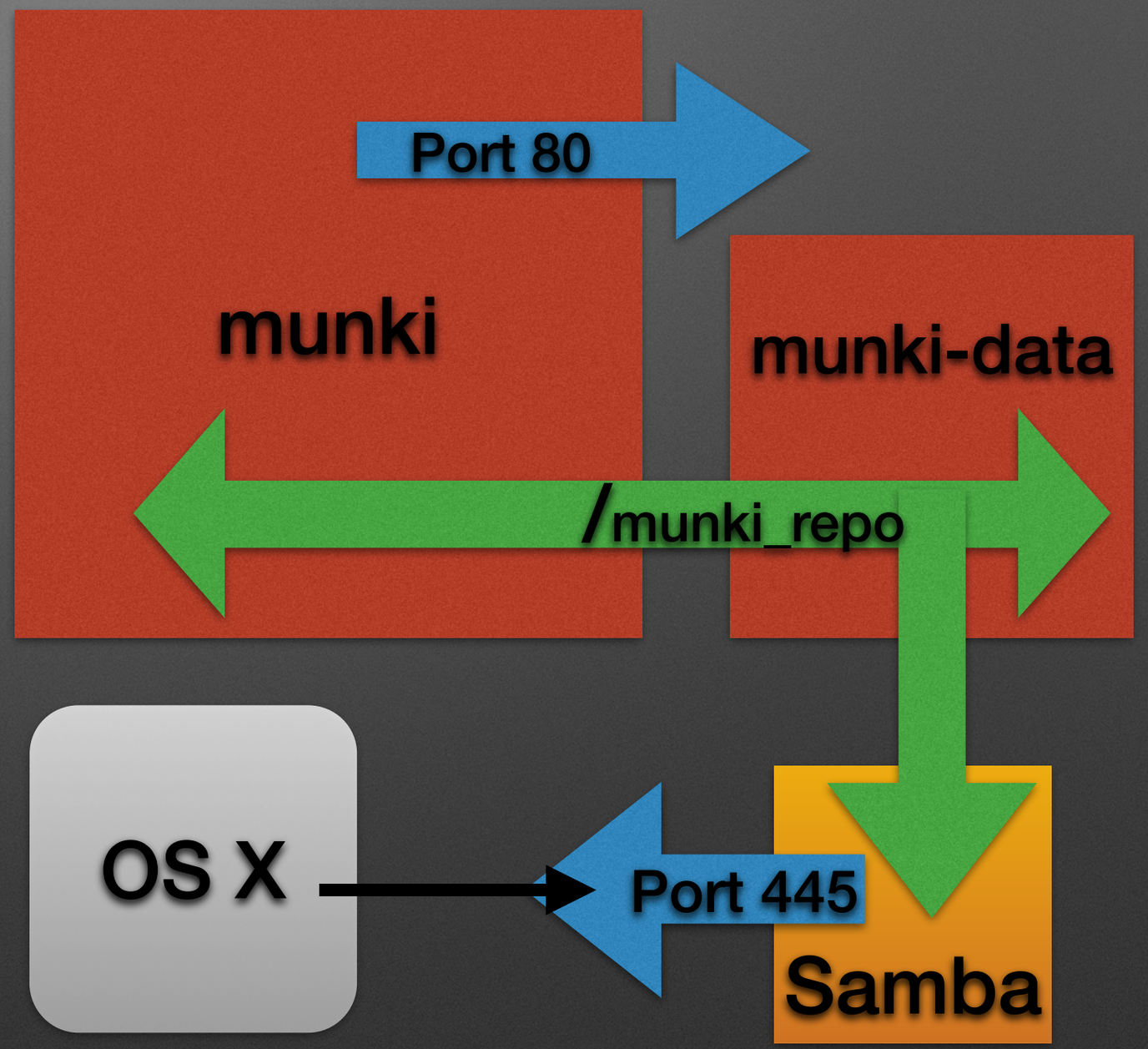


What munki looks like when you run it

There's no content yet!

Accessing the Munki container

- Munki repo is empty - we need to populate it.
- Use a container with Samba installed so we can use smb:// to access it.
- Link the samba container to the munki-data container to access /munki_repo.



smb://localhost/munki_repo

Run the Samba Container

```
docker run -d
        --name smb
        --volumes-from munki-data
        --publish 445:445
        nmcspadden/smb-munki
        /munki_repo
```

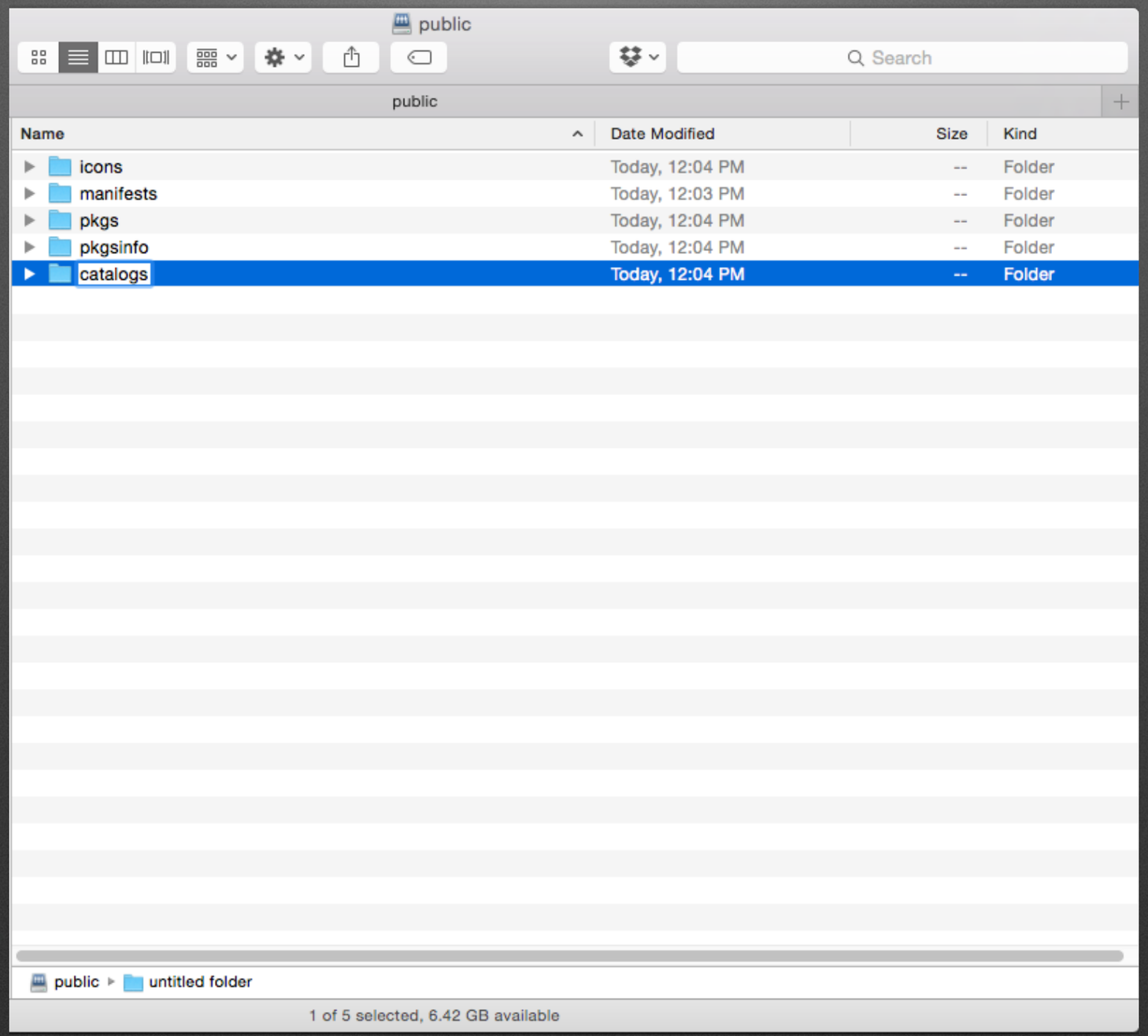
We use --volumes-from shares the repo from munki-data.
Port 445 (smb://) is open for inbound connections.

Fix the permissions

```
docker exec smb chown -R
        nobody:nogroup /munki_repo/
docker exec smb chmod -R
        ugo+rwx /munki_repo/
```

This allows authentication as guest.
Access repo from Finder: smb://docker_ip/

Create the Munki repo

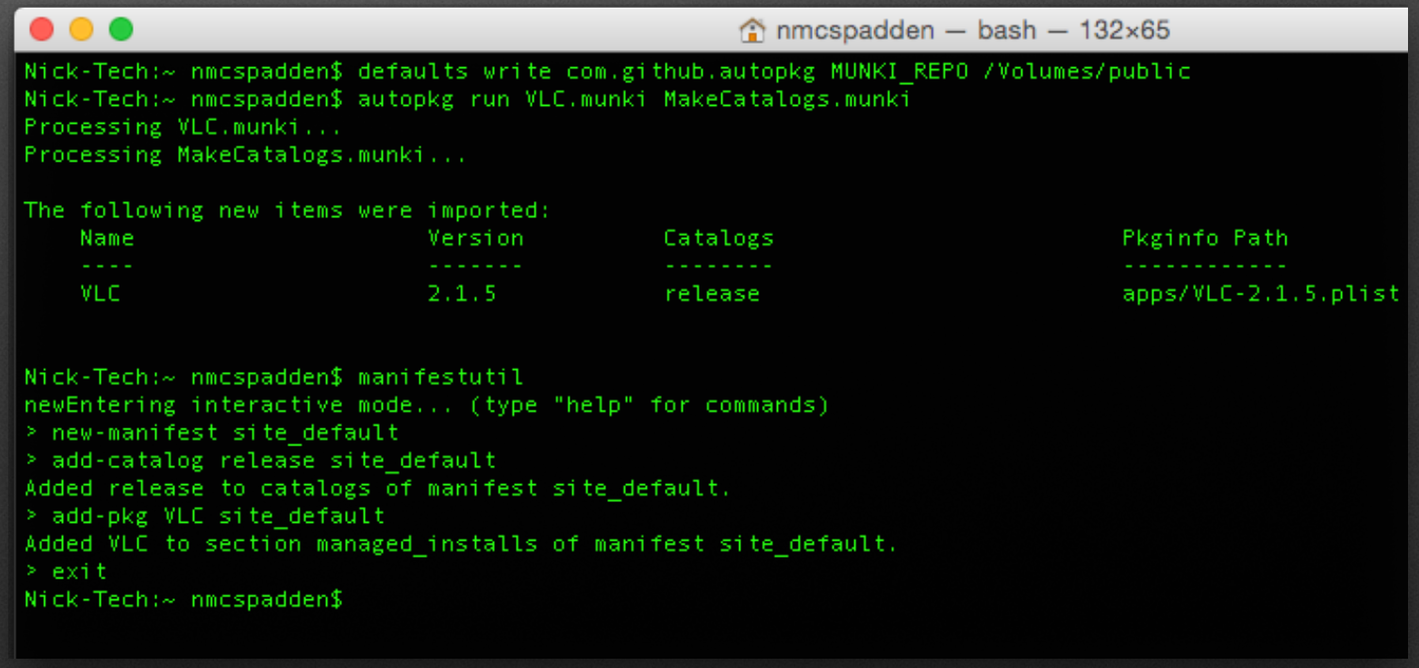


Use Autopkg to import VLC

defaults write
com.github.autopkg
MUNKI_REPO /Volumes/public

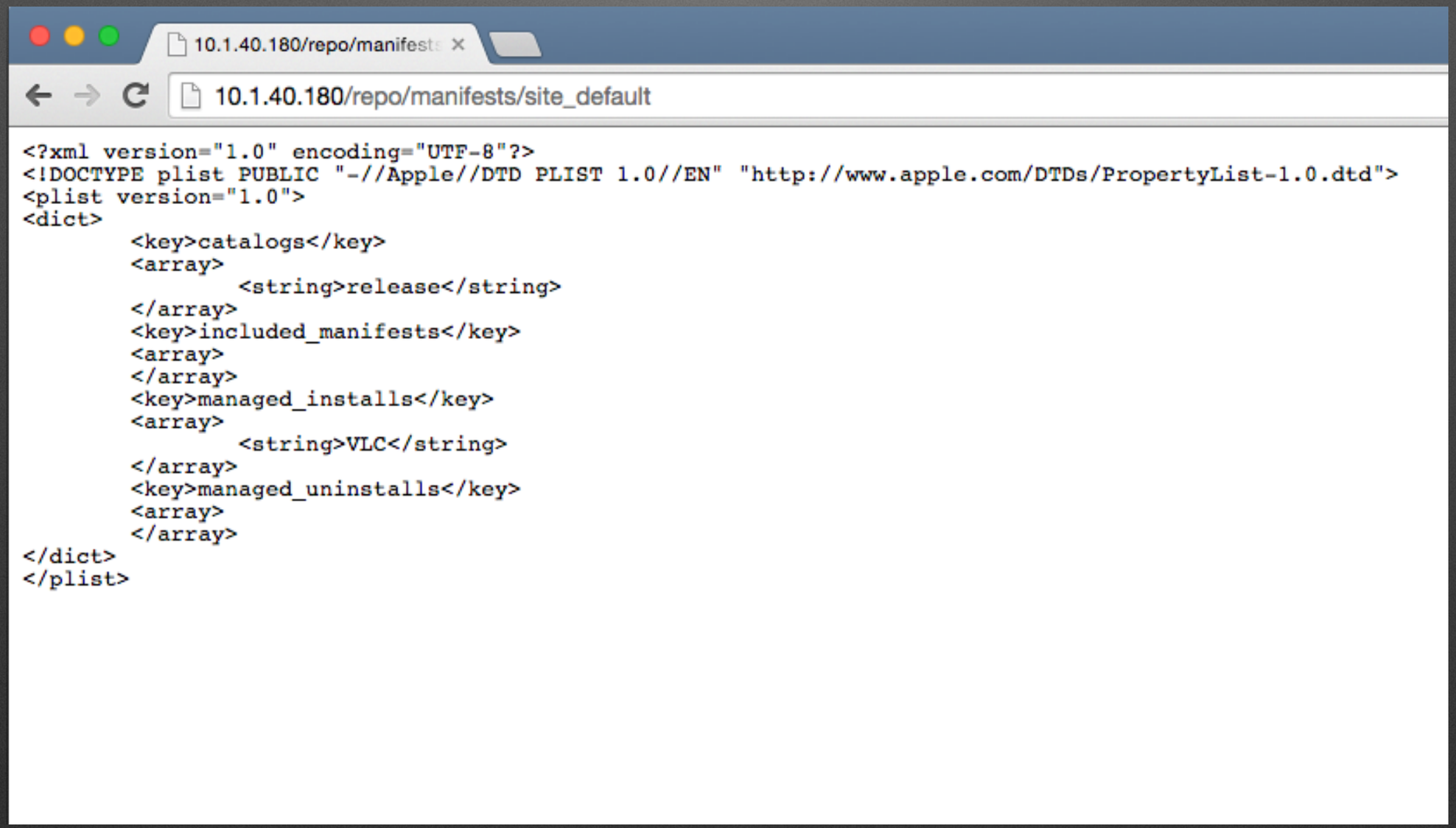
autopkg run VLC.munki
MakeCatalogs.munki

manifestutil
>new-manifest site_default
>add-catalog release
site_default
>add-pkg VLC site_default



Check the repo via browser!

http://docker_ip/repo/manifests/site_default



Configure the client

sudo defaults write
/Library/Preferences/ManagedInstalls.plist
SoftwareRepoURL
http://docker_ip/repo

Final test!

sudo managedsoftwareupdate

sudo managedsoftwareudpate --installonly

Check /Applications - VLC.app is now present!

```
Nick-Home:~ nick$ sudo managedsoftwareupdate
Managed Software Update Tool
Copyright 2010-2014 The Munki Project
https://github.com/munki/munki

Starting...
Checking for available updates...
    Retrieving list of software for this machine...
    Retrieving list of software for this machine...
    Retrieving list of software for this machine...
    Retrieving list of software for this machine...
        0..20..40..60..80..100
    Retrieving catalog "release"...
        0..20..40..60..80..100
    Downloading VLC-2.1.5__1.dmg...
        0..20..40..60..80..100
    Verifying package integrity...
    Getting icon VLC.png for VLC Media Player...
    Getting client resources...
    Getting client resources...

    The following items will be installed or upgraded:
        + VLC-2.1.5
            VLC is a free and open source cross-platform multimedia player and framework that pl
ays most multimedia files as well as DVD, Audio CD, VCD, and various streaming protocols.

Run managedsoftwareupdate --installonly to install the downloaded updates.
Finishing...
Done.
Nick-Home:~ nick$ sudo managedsoftwareupdate --installonly
Managed Software Update Tool
Copyright 2010-2014 The Munki Project
https://github.com/munki/munki

Starting...
Installing VLC Media Player (1 of 1)...
    Mounting disk image VLC-2.1.5__1.dmg...
    Copying VLC.app to /Applications/VLC.app...
    The software was successfully installed.
Finishing...
Done.
Nick-Home:~ nick$
```

Add more services!

Sal
MunkiWebAdmin
MunkiReport

Reposado

Puppet

BSDPy (NetBoot)

Casper

Crypt

Additional Resources

- Shameless self-promotion for my blog: http://osxdominion.wordpress.com
- "MacAdmins" organization on Docker hub: https://registry.hub.docker.com/repos/macadmins/
- Docker tutorial: https://www.docker.com/tryit/
- The Docker Book: http://www.dockerbook.com/