

Version Control with Git

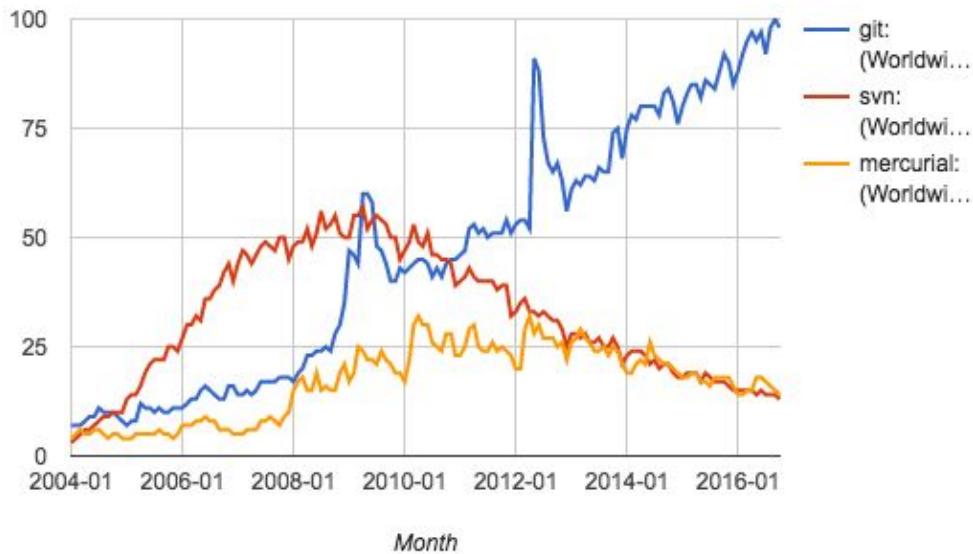
Miguel Pousa Freire

Version Control - What is it?

Manage changes to source code over time. Is used for recovering from errors introduced in the codebase, revision of the history of the projects and coordination of changes between programmers.

Git - C

git: (Worldwide), svn: (Worldwide) y mercurial: (Worldwide)



Git

- complete history of changes to every file in the repository
- allow for creating branches
- lot of tools for merging code (vimdiff, meld)
- multiple cloud providers (github, bitbucket, gitlab)

Git - Create a repository

```
> git init .
```

```
> rm -r .git
```

Git - Common tasks

- > git status # shows the current branch and the changes not yet committed
- > git log # shows the history of commits, starting by the most recent
- > git checkout feature # changes the current branch (master by default)
- > git checkout -b newFeature # creates a new branch
- > git remote -v # shows the configured origins

Git - Record changes

> git commit -am 'Initial Commit'

This creates a “commit” the unit of change in git history.

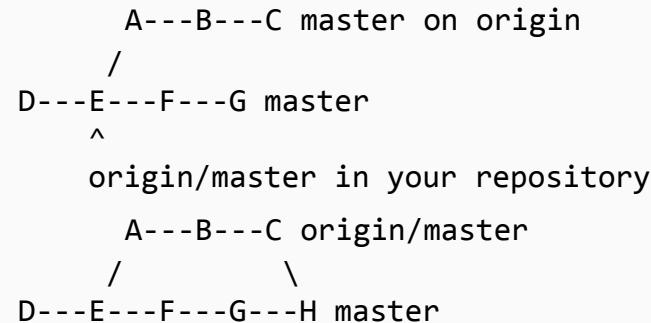
> git add newFile

> git push origin master

This uploads the changes to a remote version of the repository

Git - Pull changes from remote

```
> git pull origin master
```



Git - Pull with Conflicts

> git mergetool

Then commit & push, for solving conflicts use meld, vimdiff or similar.

If something goes wrong, then what?

Revise the history, move to a healthy commit.

```
> git log
```

```
> git checkout 775f50a17a08584d6f97cbf7b1b5bc50860fd624
```

If you want to discard the current changes then:

```
> git reset --hard HEAD
```

If something goes wrong, then what?

If you want to reverse commits in the history:

```
> git revert HEAD~3
```

Reverse the changes specified by the fourth commit ago and then creates a new commit with the changes.

Typical Workflow

Many developer working on the same codebase.

Every feature is isolated in it's own branch.

When a feature is ready merge with master via pull.

Always push changes to remote.

Q&A Time

- Are you using Git in your projects?
- If so how many developers are collaborating on the same codebase?
- Have you encounter any problems and not being able to go back?

Containers with Docker

Miguel Pousa Freire

Why?

- Isolation of environment
- Dependency management
- A layer of security
- Resource scheduling
- The automation of the deployment process
- Low overhead
- Fast to run and to download (lightweight)

Problem: Complex Environments

Software passes a series of environments during the development process.

- Developer Computer
- Contributor Laptop
- QA Server
- Staging Server
- Production Server or Customer Server

Problem: Complex Systems

Software is composed of large numbers of subsystems.

- Frontend Website: Nginx, Unicorn, Rails, Ruby
- API: Nginx, Python, Tornado
- Background Workers: Celery, Python, OpenCV, PhantomJS
- Central Database: PostgreSQL, Redis, Riak
- Analytics: Hadoop, Hive, Spark

Isolation

Containers allow you to isolate the software and all of it's dependencies

They become independent of the environment

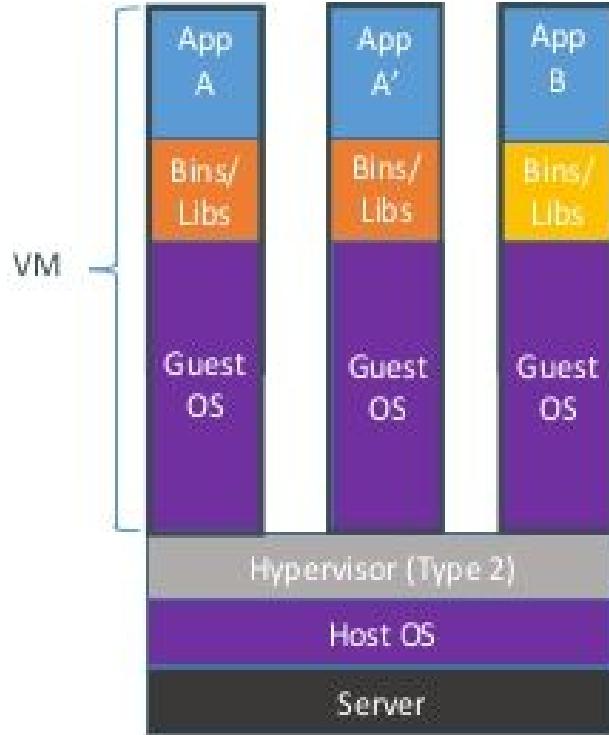
One dimension less in the problem!

Virtualization



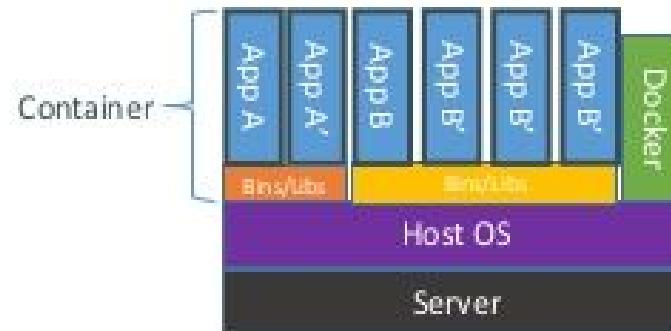
- Computers are Universal Turing Machines
- That means they can compute anything
- Even simulate themselves!

Containers vs. VMs



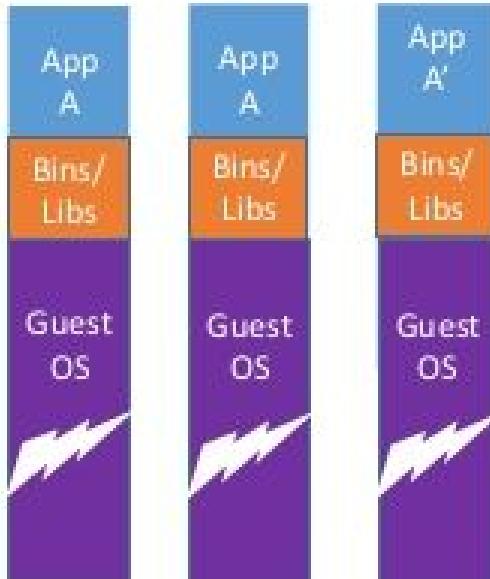
Containers are isolated,
but share OS and, where
appropriate, bins/libraries

...result is significantly faster deployment,
much less overhead, easier migration,
faster restart



Why are Docker containers lightweight?

VMs



Every app, every copy of an app, and every slight modification of the app requires a new virtual server

Containers

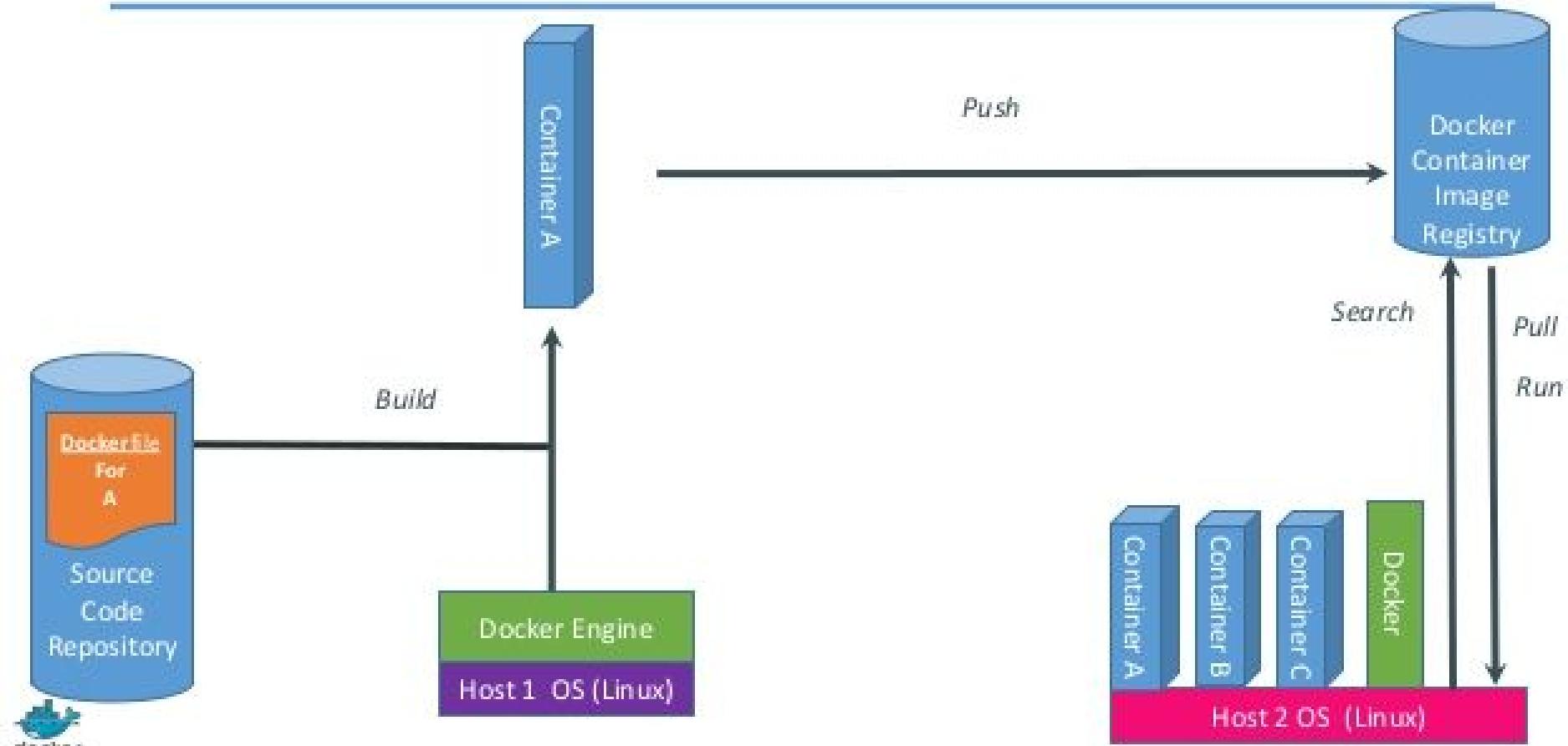


Original App
(No OS to take up space, resources, or require restart)

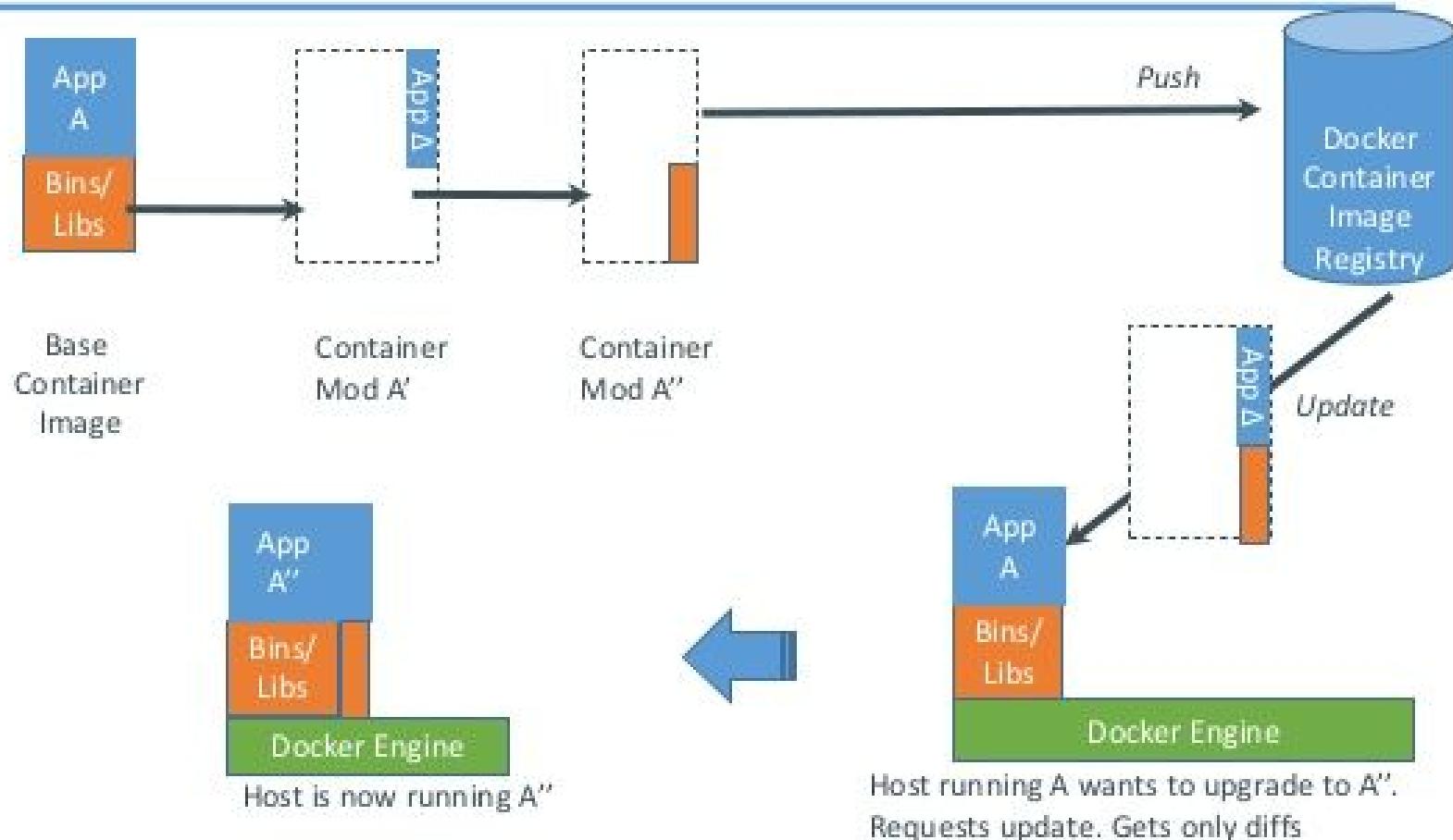
Copy of App
No OS. Can Share bins/libs

Modified App
Copy on write capabilities allow us to only save the diffs between container A and container A'

What are the basics of the Docker system?



Changes and Updates



Docker CLI Commands

> docker images

> docker pull

> docker push

> docker rmi

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
grafana/grafana	latest	de2ea786f98a	4 weeks ago	260.7 MB
wideeyes/we-search-engine-didb	latest	877aedebfd06	7 months ago	943.2 MB
wideeyes/we-search-engine-aggregator	latest	d02a10887e77	7 months ago	921.7 MB
wideeyes/we-search-engine-comparator	latest	bbfd00f763e5	7 months ago	267.2 MB
<none>	<none>	e7974dd9081	7 months ago	265.6 MB
<none>	<none>	4748a70bac06	7 months ago	943.3 MB
<none>	<none>	79127307489e	7 months ago	921.7 MB
<none>	<none>	4dcde0eac43c	7 months ago	943.5 MB
python	3.5	70c16d34e4c8	7 months ago	689.6 MB
pg-example	latest	1e4500b64fb0	9 months ago	6.141 MB
example-scratch	latest	c56ee08a0b84	9 months ago	6.872 MB
postgres	latest	b2b0df88a221	9 months ago	263.8 MB
wideeyes/we-saas-api-mirror	oregon	dfe65bc88e13	10 months ago	698.7 MB
wideeyes/we-saas-api-mirror	latest	0acce231090e	10 months ago	698.7 MB
wideeyes/we-saas-api-mirror	staging	f023810fd222	10 months ago	698.7 MB
phusion/baseimage	latest	25b2636c0e26	11 months ago	305.1 MB
node	0.12.8	2c0ee0502f49	11 months ago	641.9 MB
miguelpfwideeyes/autoparsing-api	latest	2e14e360a342	12 months ago	958.3 MB
autoparsing-api	latest	c834068ebdfa	12 months ago	958.3 MB
miguelpfwideeyes/we-vision-server	latest	5d10b95e2bcd	13 months ago	1.686 GB
python	3.5-onbuild	dd0a534c60c8	14 months ago	689 MB
kalilinux/kali-linux-docker	latest	01376bcb6241	15 months ago	420.2 MB
ubuntu	12.04	fcdd63281d1f	15 months ago	134.8 MB

Docker CLI Commands

> docker build -t imageName -f Dockerfile .

> docker run -p 80:8080 --name SomeName -ti imageName

> docker exec

> docker ps

> docker rm

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
e018061da19e	mysql	"docker-entrypoint.sh"	54 seconds ago	Exited (1) 6 seconds ago	
f4ce451dd2ab	grafana/grafana	"/run.sh"	24 minutes ago	Up 23 minutes	3000/tcp
b9866e3d12c9	postgres	"/docker-entrypoint.s"	24 minutes ago	Up 23 minutes	5432/tcp, 0.0.0.0:5552->6788/tcp
68af5bae905a	autoparsing-api	"python ./run.py"	24 minutes ago	Up 24 minutes	0.0.0.0:80->6789/tcp
ea6e7033be46	ubuntu:12.04	"-p 80:8080 --name ub"	26 minutes ago	Created	
a5980d92e5bb	ubuntu:12.04	"-p 80:8080 --name ub"	26 minutes ago	Created	
c8c35ea34d88	grafana/grafana	"/run.sh"	2 weeks ago	Exited (0) 13 days ago	

Docker CLI Commands

- > docker logs -f ContainerName
- > docker commit
- > docker cp
- > docker network

Dockerfile

```
FROM debian:jessie
MAINTAINER NGINX Docker Maintainers "docker-maint@nginx.com"
ENV NGINX_VERSION 1.11.5-1~jessie
RUN apt-key adv --keyserver hkp://pgp.mit.edu:80 --recv-keys 573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62 \
    && echo "deb http://nginx.org/packages/mainline/debian/ jessie nginx" >> /etc/apt/sources.list \
    && apt-get update \
    && apt-get install --no-install-recommends --no-install-suggests -y \
        ca-certificates \
        nginx=${NGINX_VERSION} \
        nginx-module-xslt \
        nginx-module-geoip \
        nginx-module-image-filter \
        nginx-module-perl \
        nginx-module-njs \
        gettext-base \
    && rm -rf /var/lib/apt/lists/*
# forward request and error logs to docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log \
    && ln -sf /dev/stderr /var/log/nginx/error.log

EXPOSE 80 443
CMD ["nginx", "-g", "daemon off;"]
```

Q&A Time

- Have you implemented a CI/CD delivery pipeline?
- If so, have you do it without containers?

Build Pipeline Flow

