Going D/S/K Prod Like A Pro

BRET FISHER

Docker Captain, DevOps Dude,
Creator of Docker Mastery

bretfisher.com/docker

@bretfisher

Going D/S/K Prod Like A Pro

BRET FISHER

Docker Captain, DevOps Dude,
Creator of Docker Mastery

bretfisher.com/docker

@bretfisher

Going D/S/K Prod Like A Pro

BRET FISHER

Docker Captain, DevOps Dude,
Creator of Docker Mastery

bretfisher.com/docker

@bretfisher

Please

Ask questions through the app

📣 *Rate Session*

Thank you!

I've given 50+ Docker DevOps talks in the last 4 years! 😵

I've given 50+ Docker DevOps talks in the last 4 years! 😵

How can I cram the "best of" in 30 minutes to get you in production faster?

A Bit About Me

A Bit About Me

- Geek since 5th Grade

A Bit About Me

- Geek since 5th Grade
- IT Sysadmin+Dev since 1994

A Bit About Me

- Geek since 5th Grade
- IT Sysadmin+Dev since 1994
- Maker of "Docker Mastery" 120k students

A Bit About Me

- Geek since 5th Grade
- IT Sysadmin+Dev since 1994
- Maker of "Docker Mastery" 120k students
- Container Fanboy

A Bit About Me

- Geek since 5th Grade
- IT Sysadmin+Dev since 1994
- Maker of "Docker Mastery" 120k students
- Container Fanboy
- DevOps Trainer/Consultant

Limit Your Simultaneous Innovation

Limit Your Simultaneous Innovation

- Many initial container projects are too big in scope

Limit Your Simultaneous Innovation

- Many initial container projects are too big in scope
- Solutions you maybe don't need day one:

Limit Your Simultaneous Innovation

- Many initial container projects are too big in scope
- Solutions you maybe don't need day one:
    - Fully automatic CI/CD

Limit Your Simultaneous Innovation

- Many initial container projects are too big in scope
- Solutions you maybe don't need day one:
  - Fully automatic CI/CD
  - Dynamic performance autoscaling

Limit Your Simultaneous Innovation

- Many initial container projects are too big in scope
- Solutions you maybe don't need day one:
  - Fully automatic CI/CD
  - Dynamic performance autoscaling
  - Containerizing all or nothing

Limit Your Simultaneous Innovation

- Many initial container projects are too big in scope
- Solutions you maybe don't need day one:
  - Fully automatic CI/CD
  - Dynamic performance autoscaling
  - Containerizing all or nothing
  - Starting with persistent data

Legacy Apps Work In Containers Too

Legacy Apps Work In Containers Too

- Microservice conversion isn't required

Legacy Apps Work In Containers Too

- Microservice conversion isn't required
- 12 Factor is a horizon we're always chasing

Legacy Apps Work In Containers Too

- Microservice conversion isn't required
- 12 Factor is a horizon we're always chasing
- Don't let these ideals delay containerization

What To Focus On First: Dockerfiles

What To Focus On First: Dockerfiles

- More important than fancy orchestration

What To Focus On First: Dockerfiles

- More important than fancy orchestration
- It's your new build documentation

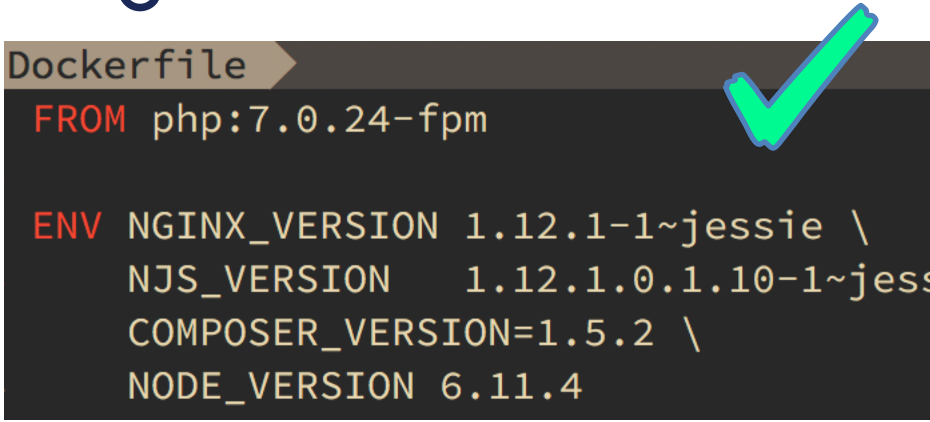
What To Focus On First: Dockerfiles

- More important than fancy orchestration
- It's your new build documentation
- Study Dockerfile/Entrypoint of Hub Officials

What To Focus On First: Dockerfiles

- More important than fancy orchestration
- It's your new build documentation
- Study Dockerfile/Entrypoint of Hub Officials
- Use FROM Official distros that are most familiar

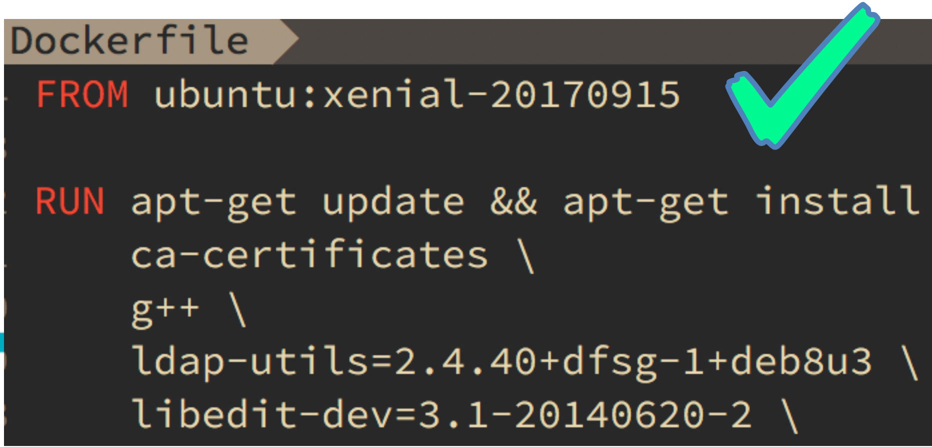
Dockerfile Anti-pattern: Using Latest



```
Dockerfile
 FROM php:7.0.24-fpm

 ENV NGINX_VERSION 1.12.1-1~jessie \
     NJS_VERSION   1.12.1.0.1.10-1~jess
     COMPOSER_VERSION=1.5.2 \
     NODE_VERSION 6.11.4
```

✓



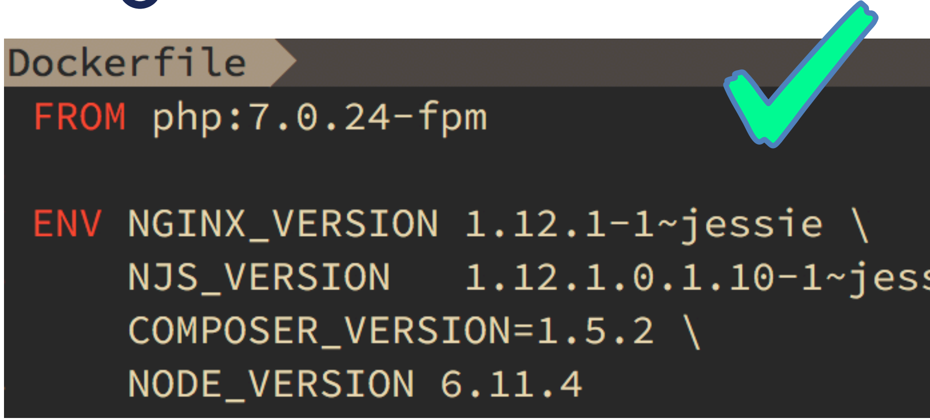
```
Dockerfile
 FROM ubuntu:xenial-20170915

 RUN apt-get update && apt-get install
     ca-certificates \
     g++ \
     ldap-utils=2.4.40+dfsg-1+deb8u3 \
     libedit-dev=3.1-20140620-2 \
```

✓

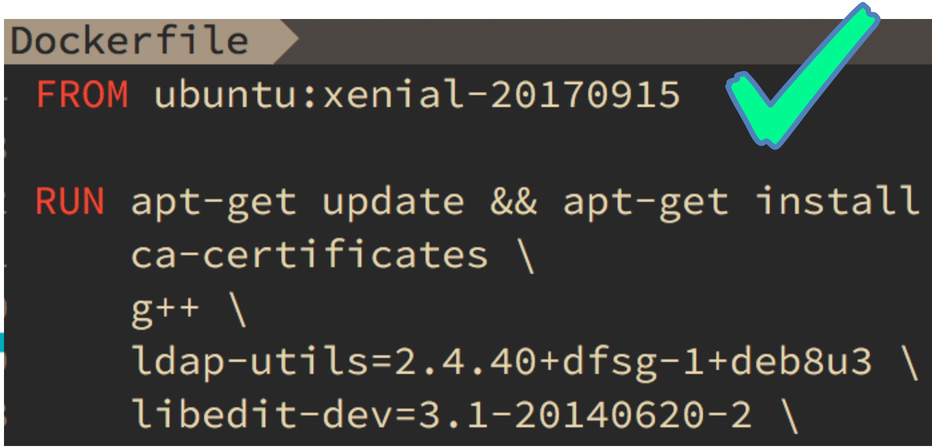
Dockerfile Anti-pattern: Using Latest

- Latest = Image builds will be ¯_(ツ)_/¯



```
Dockerfile
 FROM php:7.0.24-fpm

 ENV NGINX_VERSION 1.12.1-1~jessie \
     NJS_VERSION    1.12.1.0.1.10-1~jess
     COMPOSER_VERSION=1.5.2 \
     NODE_VERSION 6.11.4
```

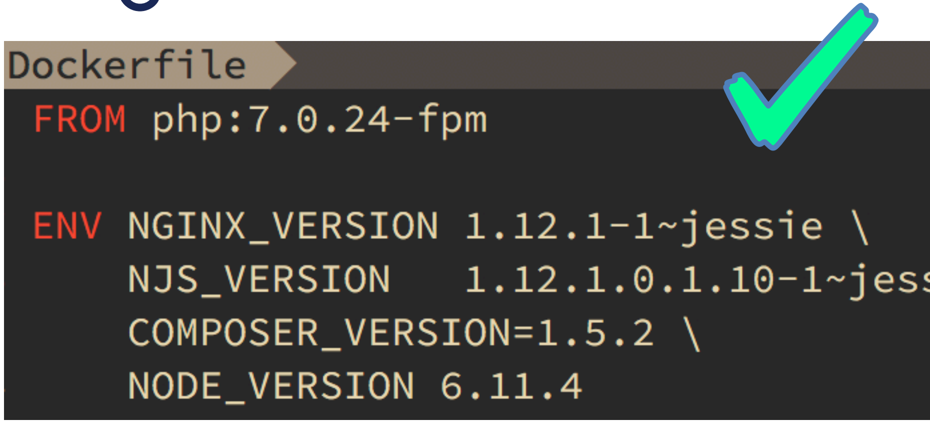


```
Dockerfile
 FROM ubuntu:xenial-20170915

 RUN apt-get update && apt-get install
     ca-certificates \
     g++ \
     ldap-utils=2.4.40+dfsg-1+deb8u3 \
     libedit-dev=3.1-20140620-2 \
```

Dockerfile Anti-pattern: Using Latest

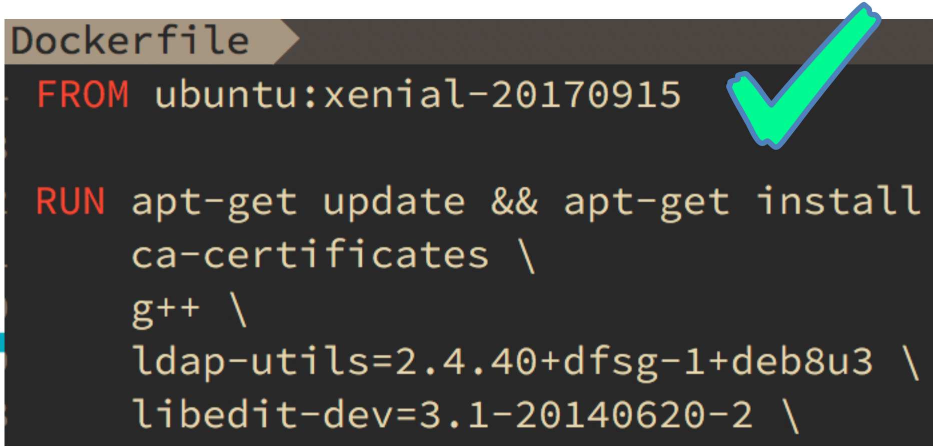
- Latest = Image builds will be ¯_(ツ)_/¯
- Problem: Image builds pull FROM latest
- Solution: Use specific FROM tags



Dockerfile

```
FROM php:7.0.24-fpm

ENV NGINX_VERSION 1.12.1-1~jessie \
    NJS_VERSION   1.12.1.0.1.10-1~jess
    COMPOSER_VERSION=1.5.2 \
    NODE_VERSION 6.11.4
```



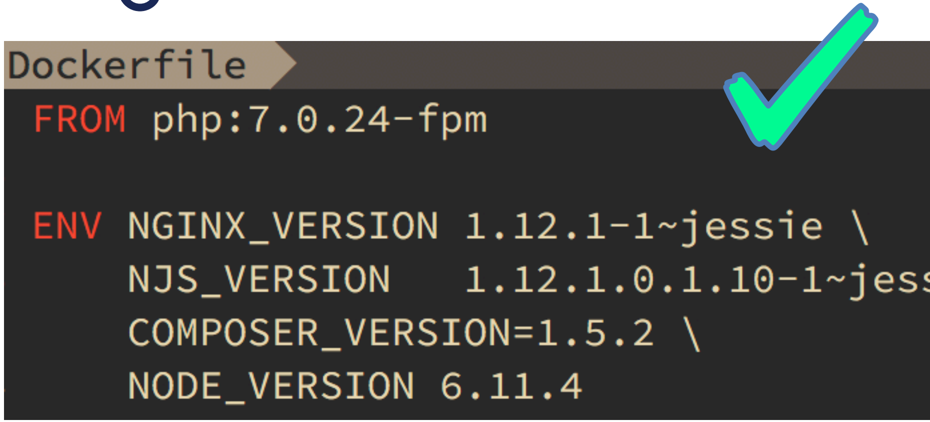
Dockerfile

```
FROM ubuntu:xenial-20170915

RUN apt-get update && apt-get install
    ca-certificates \
    g++ \
    ldap-utils=2.4.40+dfsg-1+deb8u3 \
    libedit-dev=3.1-20140620-2 \
```

Dockerfile Anti-pattern: Using Latest

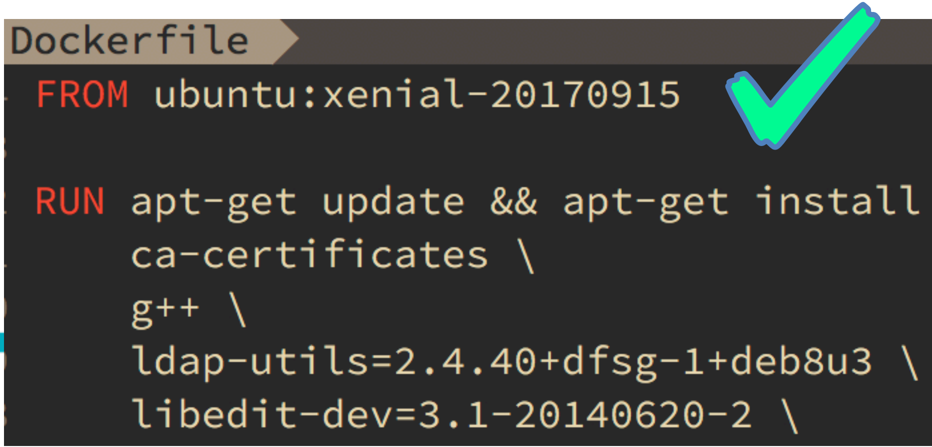
- Latest = Image builds will be ¯_(ツ)_/¯
- Problem: Image builds pull FROM latest
- Solution: Use specific FROM tags
- Problem: Image builds install latest packages
- Solution: Specify version for critical apt/yum/apk packages



Dockerfile

```
FROM php:7.0.24-fpm

ENV NGINX_VERSION 1.12.1-1~jessie \
    NJS_VERSION   1.12.1.0.1.10-1~jess
    COMPOSER_VERSION=1.5.2 \
    NODE_VERSION 6.11.4
```



Dockerfile

```
FROM ubuntu:xenial-20170915

RUN apt-get update && apt-get install
    ca-certificates \
    g++ \
    ldap-utils=2.4.40+dfsg-1+deb8u3 \
    libedit-dev=3.1-20140620-2 \
```

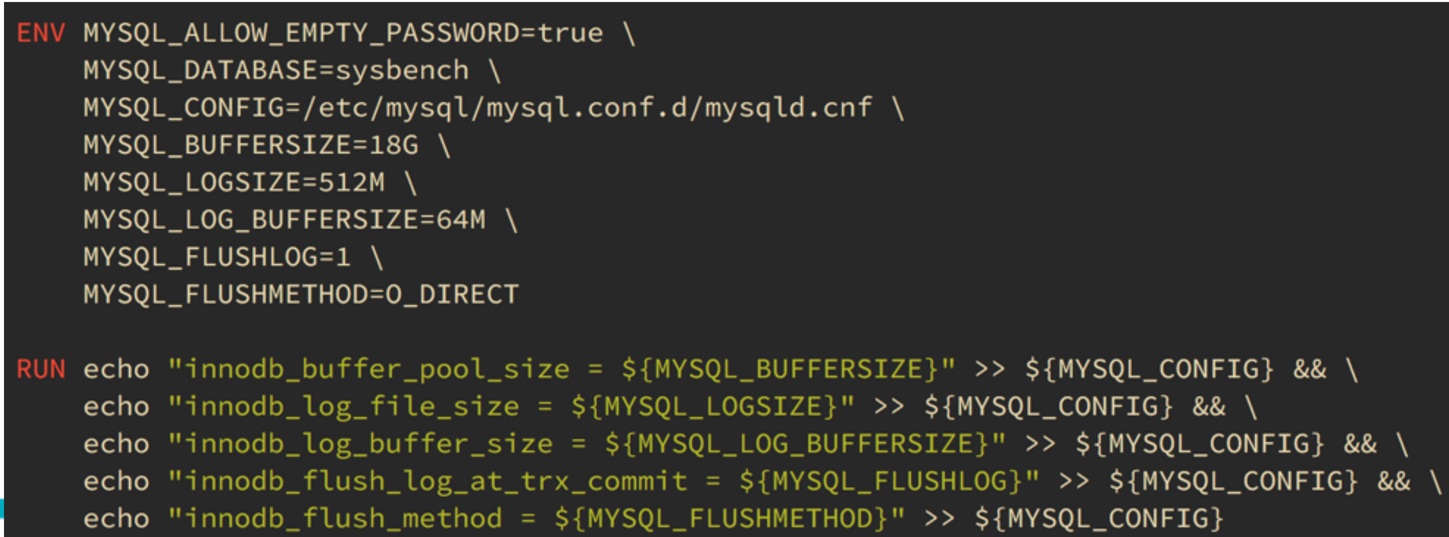
Dockerfile Anti-pattern: Leaving Default Config

Dockerfile Anti-pattern: Leaving Default Config

- Problem: Not changing app defaults, or blindly copying VM conf
    - e.g. php.ini, mysql.conf.d, java memory

Dockerfile Anti-pattern: Leaving Default Config

- Problem: Not changing app defaults, or blindly copying VM conf
  - e.g. php.ini, mysql.conf.d, java memory
- Solution: Update default configs via ENV, RUN, and ENTRYPOINT



```
ENV MYSQL_ALLOW_EMPTY_PASSWORD=true \
    MYSQL_DATABASE=sysbench \
    MYSQL_CONFIG=/etc/mysql/mysql.conf.d/mysqld.cnf \
    MYSQL_BUFFERSIZE=18G \
    MYSQL_LOGSIZE=512M \
    MYSQL_LOG_BUFFERSIZE=64M \
    MYSQL_FLUSHLOG=1 \
    MYSQL_FLUSHMETHOD=O_DIRECT

RUN echo "innodb_buffer_pool_size = ${MYSQL_BUFFERSIZE}" >> ${MYSQL_CONFIG} && \
    echo "innodb_log_file_size = ${MYSQL_LOGSIZE}" >> ${MYSQL_CONFIG} && \
    echo "innodb_log_buffer_size = ${MYSQL_LOG_BUFFERSIZE}" >> ${MYSQL_CONFIG} && \
    echo "innodb_flush_log_at_trx_commit = ${MYSQL_FLUSHLOG}" >> ${MYSQL_CONFIG} && \
    echo "innodb_flush_method = ${MYSQL_FLUSHMETHOD}" >> ${MYSQL_CONFIG}
```

Containers-on-VM or Container-on-Bare-Metal

Containers-on-VM or Container-on-Bare-Metal

- Do either, or both. Lots of pros/cons to either

Containers-on-VM or Container-on-Bare-Metal

- Do either, or both. Lots of pros/cons to either
- Stick with what you know at first

Containers-on-VM or Container-on-Bare-Metal

- Do either, or both. Lots of pros/cons to either
- Stick with what you know at first
- Do some basic performance testing. You will learn lots!

Containers-on-VM or Container-on-Bare-Metal

- Do either, or both. Lots of pros/cons to either
- Stick with what you know at first
- Do some basic performance testing. You will learn lots!
- 2017 Docker Inc. and HPE whitepaper on MySQL benchmark
  - (authored by yours truly, and others)
  - bretfisher.com/gotochgo18

OS Linux Distribution/Kernel Matters

OS Linux Distribution/Kernel Matters

- Docker is very kernel and host storage driver dependent

OS Linux Distribution/Kernel Matters

- Docker is very kernel and host storage driver dependent
- Innovations/fixes are still happening here

OS Linux Distribution/Kernel Matters

- Docker is very kernel and host storage driver dependent
- Innovations/fixes are still happening here
- "Minimum" version != "best" version

OS Linux Distribution/Kernel Matters

- Docker is very kernel and host storage driver dependent
- Innovations/fixes are still happening here
- "Minimum" version != "best" version
- No pre-existing opinion? Ubuntu 18.04 LTS
  - Popular, well-tested with Docker
  - 4.x Kernel and wide storage driver support (overlay2)

OS Linux Distribution/Kernel Matters

- Docker is very kernel and host storage driver dependent
- Innovations/fixes are still happening here
- "Minimum" version != "best" version
- No pre-existing opinion? Ubuntu 18.04 LTS
    - Popular, well-tested with Docker
    - 4.x Kernel and wide storage driver support (overlay2)
- Container OS's aren't mainstream. Unclear TCO

OS Linux Distribution/Kernel Matters

- Docker is very kernel and host storage driver dependent
- Innovations/fixes are still happening here
- "Minimum" version != "best" version
- No pre-existing opinion? Ubuntu 18.04 LTS
    - Popular, well-tested with Docker
    - 4.x Kernel and wide storage driver support (overlay2)
- Container OS's aren't mainstream. Unclear TCO
- Get correct Docker for your distro from hub.docker.com

Container Base Distribution: Which One?

Container Base Distribution: Which One?

- Which FROM image should you use?

Container Base Distribution: Which One?

- Which FROM image should you use?
- Don't make a decision based on size (remember it's Single Instance Storage)

Container Base Distribution: Which One?

- Which FROM image should you use?
- Don't make a decision based on size (remember it's Single Instance Storage)
- At first: match your existing deployment process

Container Base Distribution: Which One?

- Which FROM image should you use?
- Don't make a decision based on size (remember it's Single Instance Storage)
- At first: match your existing deployment process
- Consider changing to Alpine later, maybe never

When to use Alpine Images

When to use Alpine Images

- Alpine is "small" and "sec focused"

When to use Alpine Images

- Alpine is "small" and "sec focused"
- But Debian/Ubuntu are smaller now too

When to use Alpine Images

- Alpine is "small" and "sec focused"
- But Debian/Ubuntu are smaller now too
- ~100MB space savings isn't significant

When to use Alpine Images

- Alpine is "small" and "sec focused"
- But Debian/Ubuntu are smaller now too
- ~100MB space savings isn't significant
- Alpine has its own issues

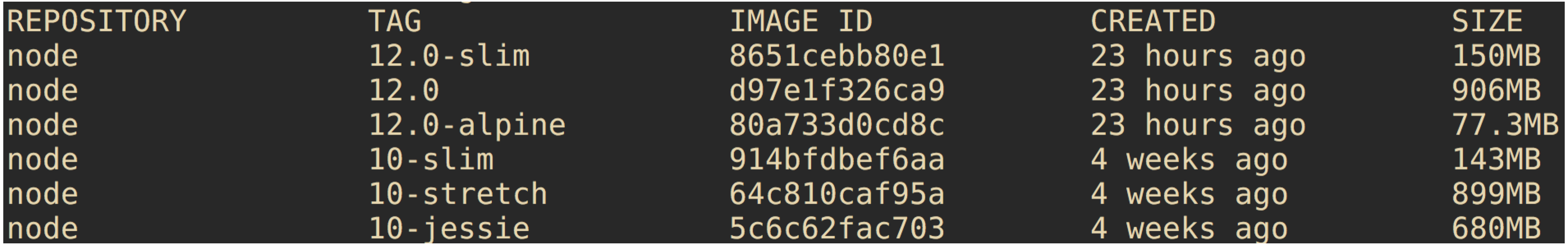
When to use Alpine Images

- Alpine is "small" and "sec focused"
- But Debian/Ubuntu are smaller now too
- ~100MB space savings isn't significant
- Alpine has its own issues
- Alpine CVE scanning fails

When to use Alpine Images

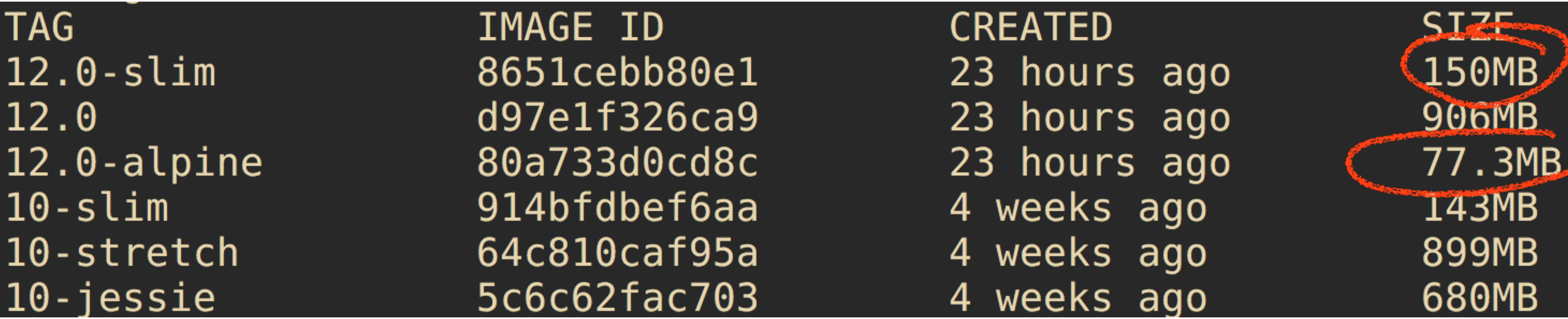
- Alpine is "small" and "sec focused"
- But Debian/Ubuntu are smaller now too
- ~100MB space savings isn't significant
- Alpine has its own issues
- Alpine CVE scanning fails
- Enterprises may require CentOS or Ubuntu/Debian

Image Sizes for node/slim/alpine



```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
node                12.0-slim           8651cebb80e1        23 hours ago        150MB
node                12.0                d97e1f326ca9        23 hours ago        906MB
node                12.0-alpine         80a733d0cd8c        23 hours ago        77.3MB
node                10-slim             914bfdbef6aa        4 weeks ago         143MB
node                10-stretch          64c810caf95a        4 weeks ago         899MB
node                10-jessie           5c6c62fac703        4 weeks ago         680MB
```

Image Sizes for node/slim/alpine

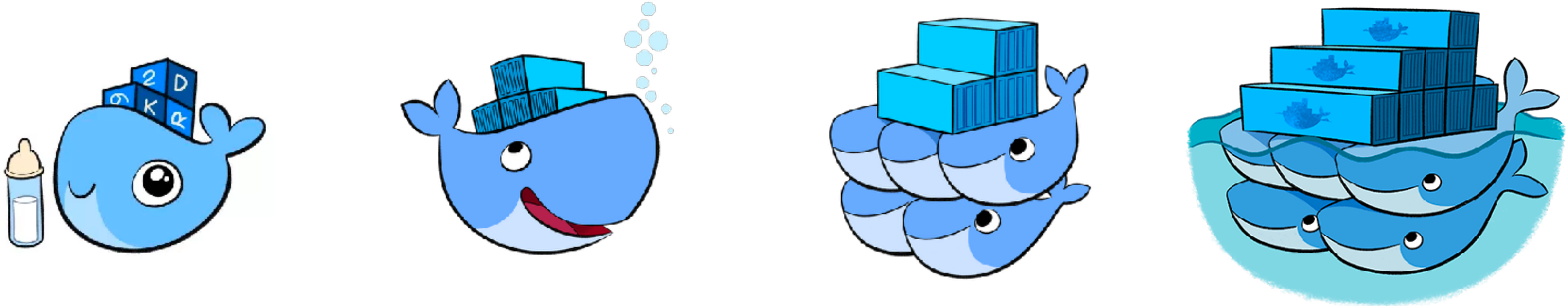


```
TAG                 IMAGE ID            CREATED             SIZE
12.0-slim           8651cebb80e1        23 hours ago        150MB
12.0                d97e1f326ca9        23 hours ago        906MB
12.0-alpine         80a733d0cd8c        23 hours ago        77.3MB
10-slim             914bfdbef6aa        4 weeks ago         143MB
10-stretch          64c810caf95a        4 weeks ago         899MB
10-jessie           5c6c62fac703        4 weeks ago         680MB
```

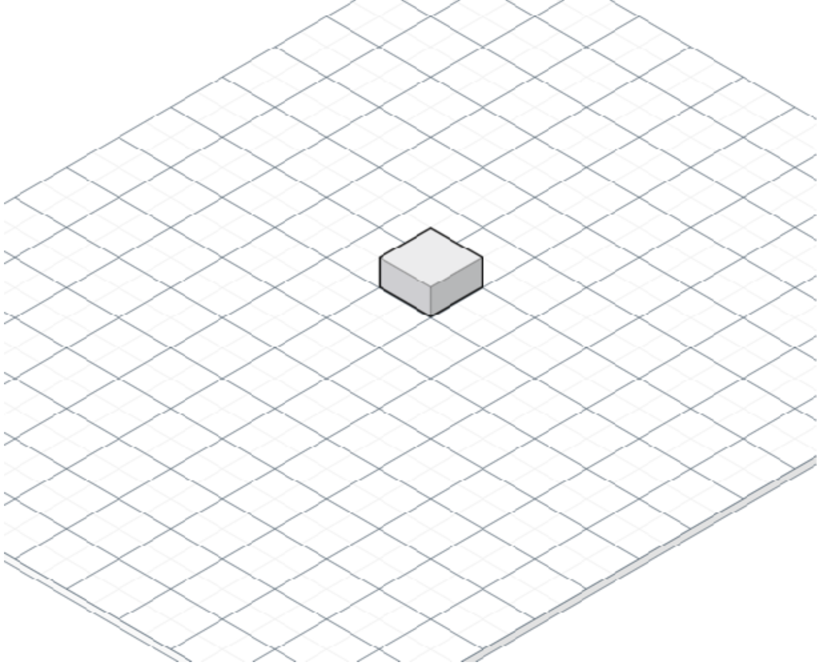
Good Defaults: Swarm Architectures

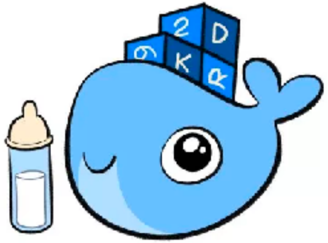
Good Defaults: Swarm Architectures

- Simple sizing guidelines based off:
  - Docker internal testing
  - Docker reference architectures
  - Real world deployments
  - Swarm3k lessons learned



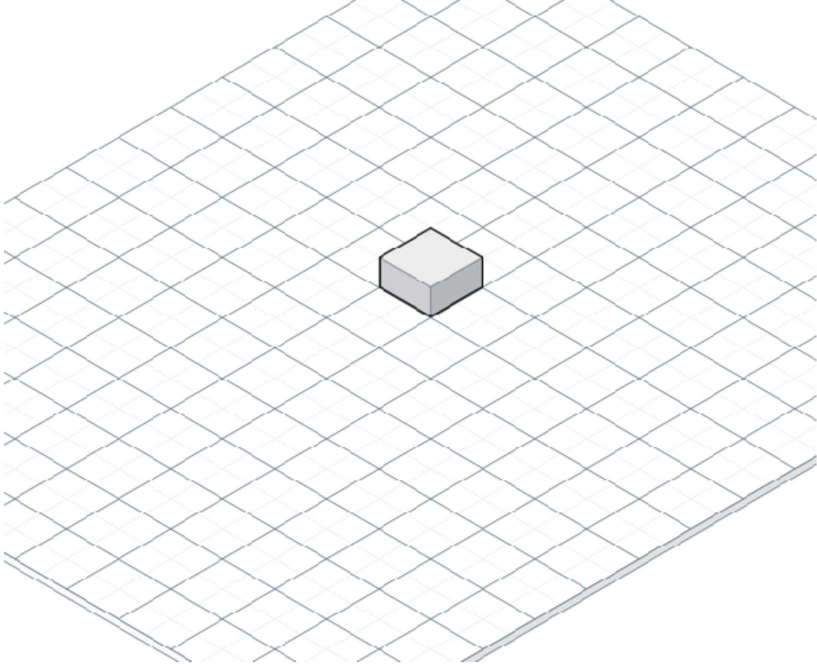
Baby Swarm: 1-Node

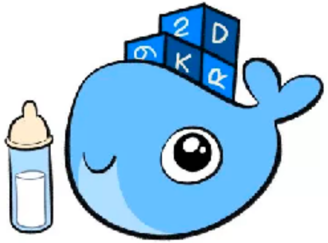




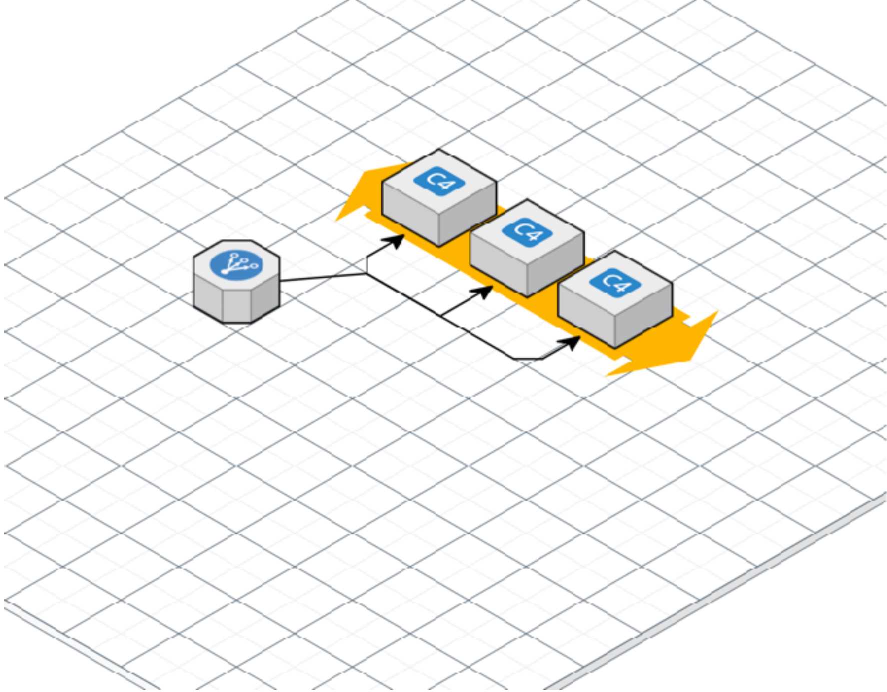
Baby Swarm: 1-Node

- "docker swarm init" done!
- Solo VM's do it, so can Swarm
- Gives you more features then docker run
- bret.show/babyswarm



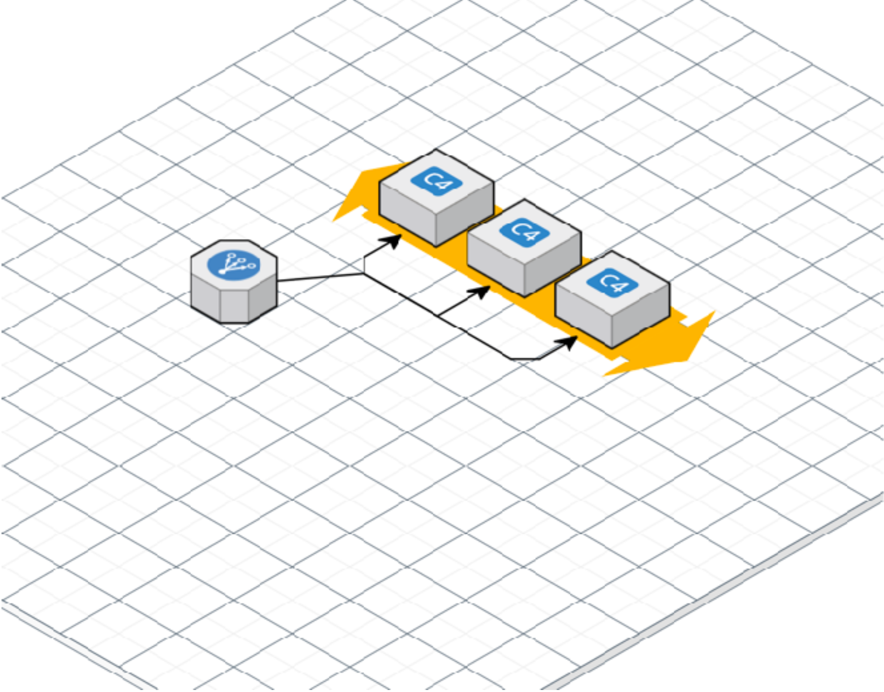


HA Swarm: 3-Node

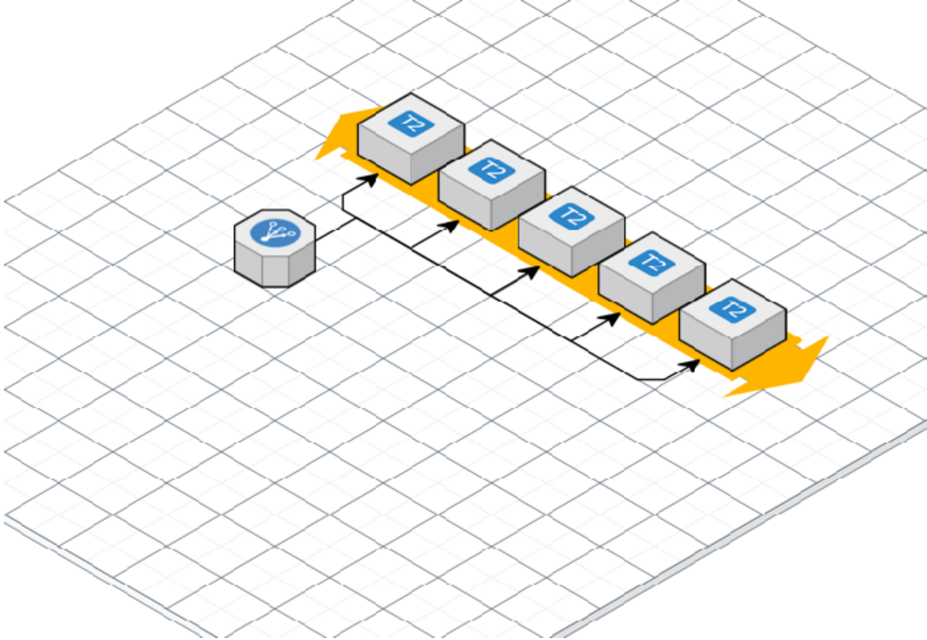


HA Swarm: 3-Node

- Minimum for HA
- All Managers
- One node can fail
- Use when very small budget
- Pet projects or Test/CI

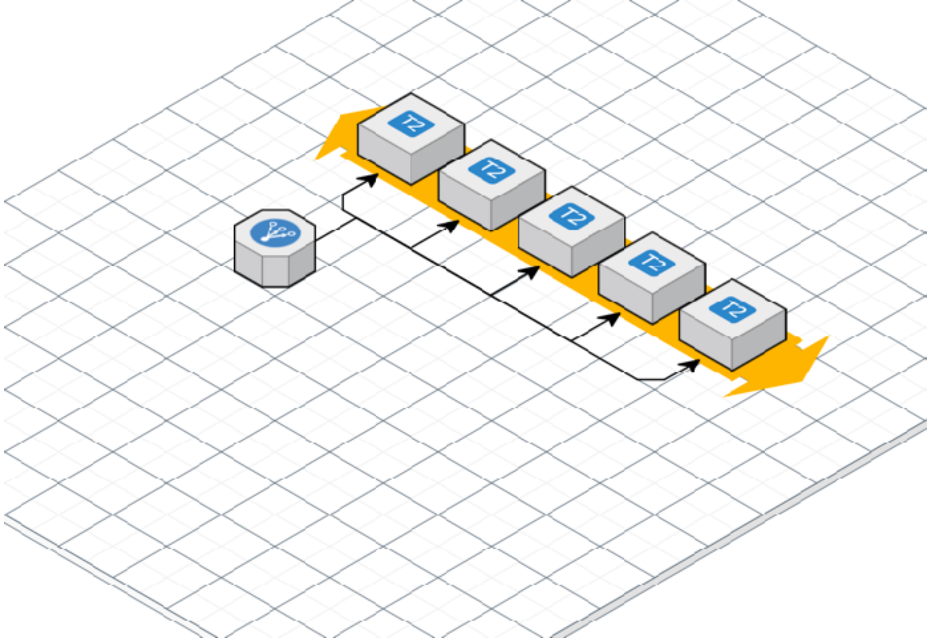


Biz Swarm: 5-Node

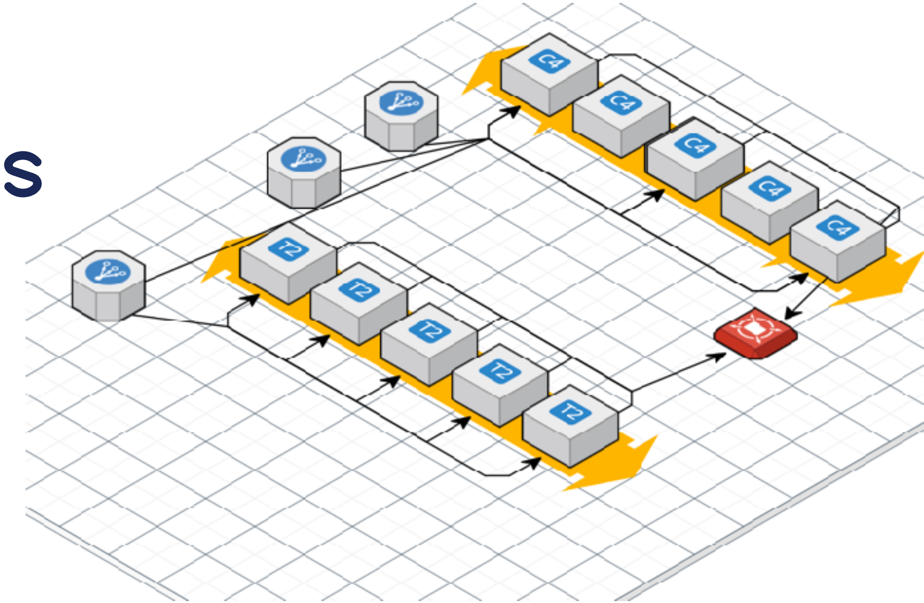


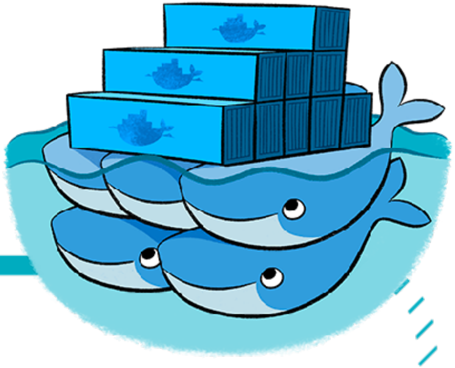
Biz Swarm: 5-Node

- Better high-availability
- All Managers
- Two nodes can fail
- My minimum for uptime that affects $$$

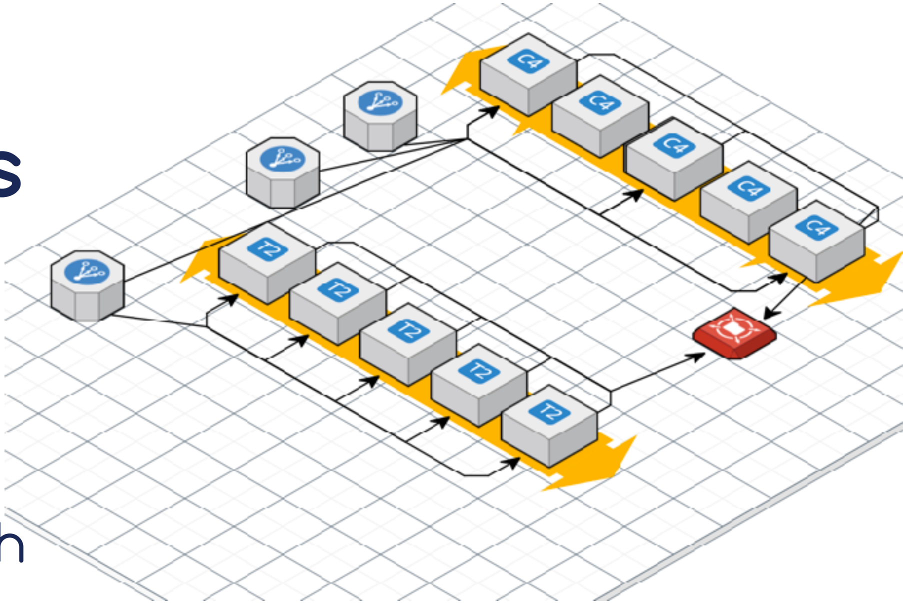


Flexy Swarm: 10+ Nodes

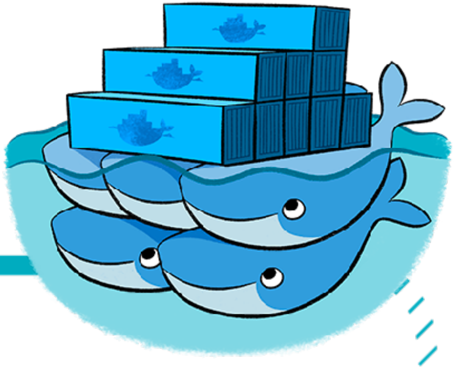




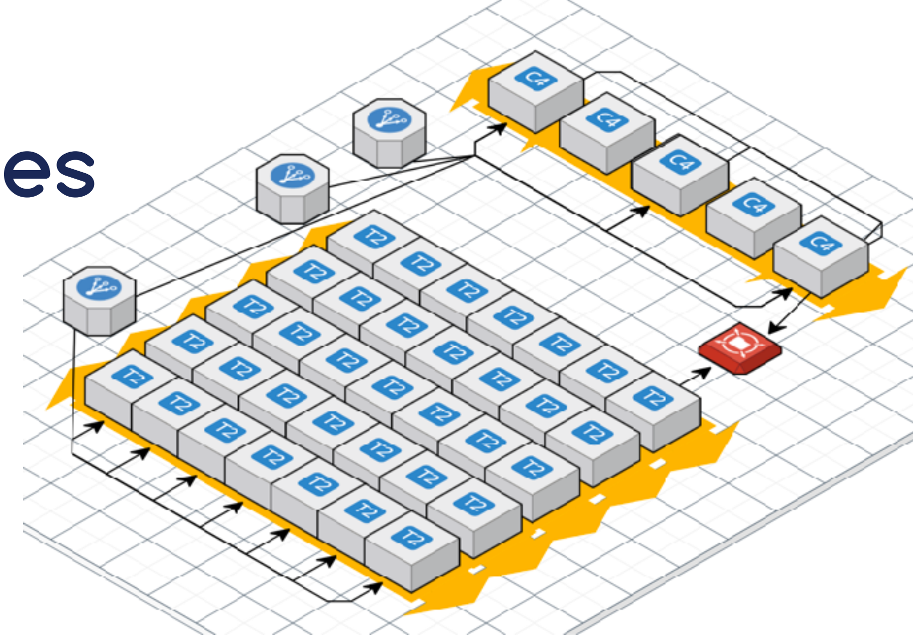
Flexy Swarm: 10+ Nodes

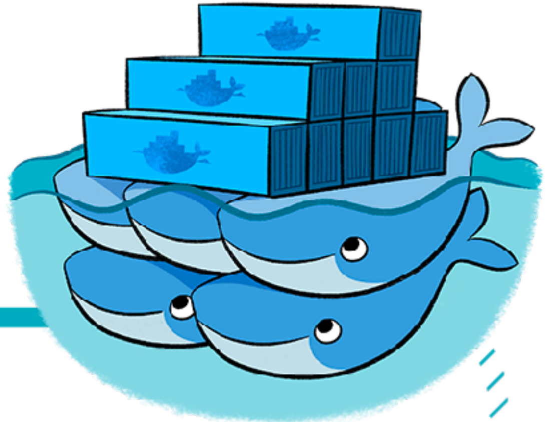


- 5 dedicated Managers
- Workers in DMZ
- Anything beyond 5 nodes, stick with 5 Managers and rest Workers
- Control container placement with labels + constraints

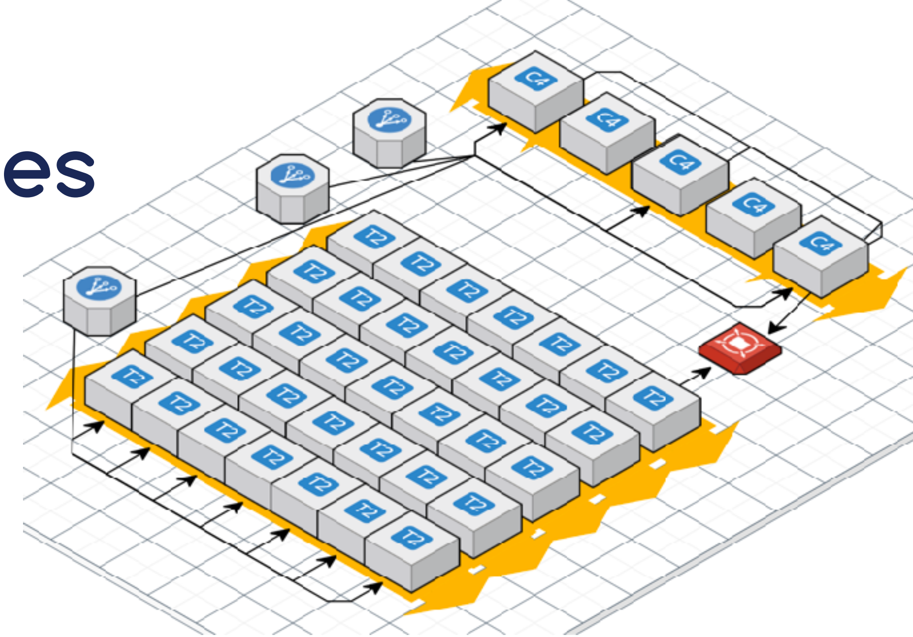


Swole Swarm: 100+ Nodes

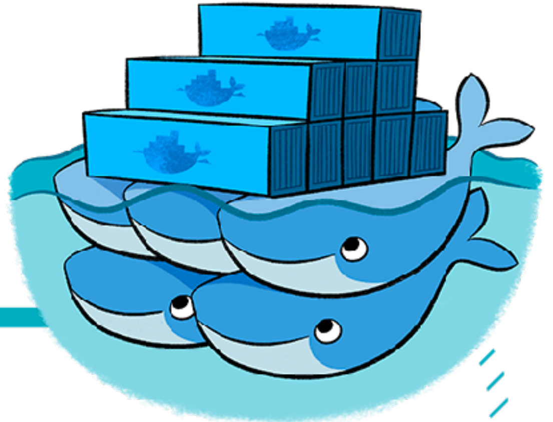




Swole Swarm: 100+ Nodes



- 5 dedicated managers
- Resize Managers as you grow
- Multiple Worker subnets on Private/DMZ
- Control container placement with labels + constraints



Don't Turn Cattle into Pets

Don't Turn Cattle into Pets

- Assume nodes will be replaced
- Assume containers will be recreated
- Automate any host customization
- Every time you SSH into a server 🐼🔫

Reasons for Multiple Clusters

Reasons for Multiple Clusters

Bad Reasons

- Different hardware configurations (or OS!)
- Different subnets or security groups
- Different availability zones
- Security boundaries for compliance

Reasons for Multiple Clusters

Bad Reasons

- Different hardware configurations (or OS!)
- Different subnets or security groups
- Different availability zones
- Security boundaries for compliance

Good Reasons

- Learning: Run Stuff on Test Swarm
- Geographical boundaries
- Management boundaries using Docker API (or Docker EE RBAC, or other auth plugin)

What About Windows Server 2019?

- Hard to be "Windows Only Swarm", mix with Linux nodes
- Much of those tools are Linux only
- Windows = Less choice, but easier path
- My recommendation:
  - Managers on Linux
  - Reserve Windows for Windows-exclusive workloads
- Swarm is more stable, Kubernetes is still early days

DevSecOps: Making Friends With InfoSec

- Good: Just putting apps in Docker vs. host =
  - Whiltelist of Linux kernel capabilities ✔
  - AppLocker profile enabled ✔
  - SecComp profile enabled ✔
- USER appname: App is not container root (e.g. node/python)
- User Namespaces: Container root isn't root (turn on per host)
- More basics at: bret.show/securityfirst

DevSecOps: Shift Left Security

- Scan, Scan, Scan.
- Scan for CVE's in git: snyk.io
- Scan for CVE's in image builds: MicroScanner
- Scan for CVE's in images: Trivy

DevSecOps: Content Trust

- Only used scanned images
- Only allow running of signed images
- Only used signed code

DevOps: Focus On Outcomes, Not Tools

DevOps: Focus On Outcomes, Not Tools

- Only change/implement what:

DevOps: Focus On Outcomes, Not Tools

- Only change/implement what:
  - Gives you back a measurable chunk of time

DevOps: Focus On Outcomes, Not Tools

- Only change/implement what:
  - Gives you back a measurable chunk of time
  - Greatly improves MTTR

DevOps: Focus On Outcomes, Not Tools

- Only change/implement what:
  - Gives you back a measurable chunk of time
  - Greatly improves MTTR
  - Greatly improves deployment frequency

DevOps: Focus On Outcomes, Not Tools

- Only change/implement what:
  - Gives you back a measurable chunk of time
  - Greatly improves MTTR
  - Greatly improves deployment frequency
- NO to everything else!

DevOps: Focus On Outcomes, Not Tools

- Only change/implement what:
  - Gives you back a measurable chunk of time
  - Greatly improves MTTR
  - Greatly improves deployment frequency
- NO to everything else!
- More at bret.show/humandevops

Outsource Well-Defined Plumbing

Outsource Well-Defined Plumbing

- Beware the "not implemented here" syndrome

Outsource Well-Defined Plumbing

- Beware the "not implemented here" syndrome
- My formula for "Do we use SaaS/Commercial"?

Outsource Well-Defined Plumbing

- Beware the "not implemented here" syndrome
- My formula for "Do we use SaaS/Commercial"?
    - If it's a challenge to implement and maintain

Outsource Well-Defined Plumbing

- Beware the "not implemented here" syndrome
- My formula for "Do we use SaaS/Commercial"?
  - If it's a challenge to implement and maintain
  - + SaaS/commercial market is mature

Outsource Well-Defined Plumbing

- Beware the "not implemented here" syndrome
- My formula for "Do we use SaaS/Commercial"?
  - If it's a challenge to implement and maintain
  - + SaaS/commercial market is mature
  - = Opportunities for outsourcing

Outsourcing: For Your Consideration

Outsourcing: For Your Consideration

- Image registry

Outsourcing: For Your Consideration

- Image registry
- Logs

Outsourcing: For Your Consideration

- Image registry
- Logs
- Monitoring and alerting

Outsourcing: For Your Consideration

- Image registry
- Logs
- Monitoring and alerting
- Big Tools/Projects: github.com/cncf/landscape

Outsourcing: For Your Consideration

- Image registry
- Logs
- Monitoring and alerting
- Big Tools/Projects: github.com/cncf/landscape
- All The Things: github.com/veggiemonk/awesome-docker
  github.com/ramitsurana/awesome-kubernetes

Tech Stacks

Designs for a full-featured cluster

Pure Open Source Swarm Stack

Pure Open Source Swarm Stack

HW / OS

Ansible

Terraform

Pure Open Source Swarm Stack

Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Registry	Docker Distribution + Portus	
CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Layer 7 Proxy	Traefik	
Registry	Docker Distribution + Portus	
CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Central Logging	ELK	
Layer 7 Proxy	Traefik	
Registry	Docker Distribution + Portus	
CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

Central Monitoring	Prometheus + Grafana	
Central Logging	ELK	
Layer 7 Proxy	Traefik	
Registry	Docker Distribution + Portus	
CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

GUI Management	Portainer	
Central Monitoring	Prometheus + Grafana	
Central Logging	ELK	
Layer 7 Proxy	Traefik	
Registry	Docker Distribution + Portus	
CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Pure Open Source Swarm Stack

GUI Management	Portainer	
Central Monitoring	Prometheus + Grafana	
Central Logging	ELK	
Layer 7 Proxy	Traefik	
Registry	Docker Distribution + Portus	
CI/CD	Jenkins	Drone
Storage	REX-Ray	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Also
Functions As A Service:
OpenFaaS

Commercial Products/SaaS Swarm Stack

Commercial Products/SaaS Swarm Stack

GUI Management	Portainer	
Central Monitoring	Librato / DataDog / Sysdig	
Central Logging	DataDog / Papertrail / Loggly	
Layer 7 Proxy	Traefik Enterprise	
Registry	Docker Hub	Quay
CI/CD	GitLab	CircleCI
Storage	Portworx	
Networking	Docker Swarm / Weave	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Ansible	Terraform

Docker Enterprise Swarm or Kubernetes

Docker Enterprise Swarm or Kubernetes

Swarm GUI	**Docker Enterprise (UCP)**	
Central Monitoring	Prometheus	Sysdig
Central Logging	Docker for AWS/Azure	
Layer 7 Proxy	**Docker Enterprise (UCP)**	
Registry	**Docker Enterprise (DTR)**	
CI/CD	Jenkins	GitLab
Storage	NetApp / Portworx / CSI	
Networking	Swarm Overlay / Calico	
Orchestration	Docker Swarm / Kubernetes	
Runtime	**Docker Enterprise**	
HW / OS	**Docker Enterprise**	

Docker Enterprise Swarm or Kubernetes

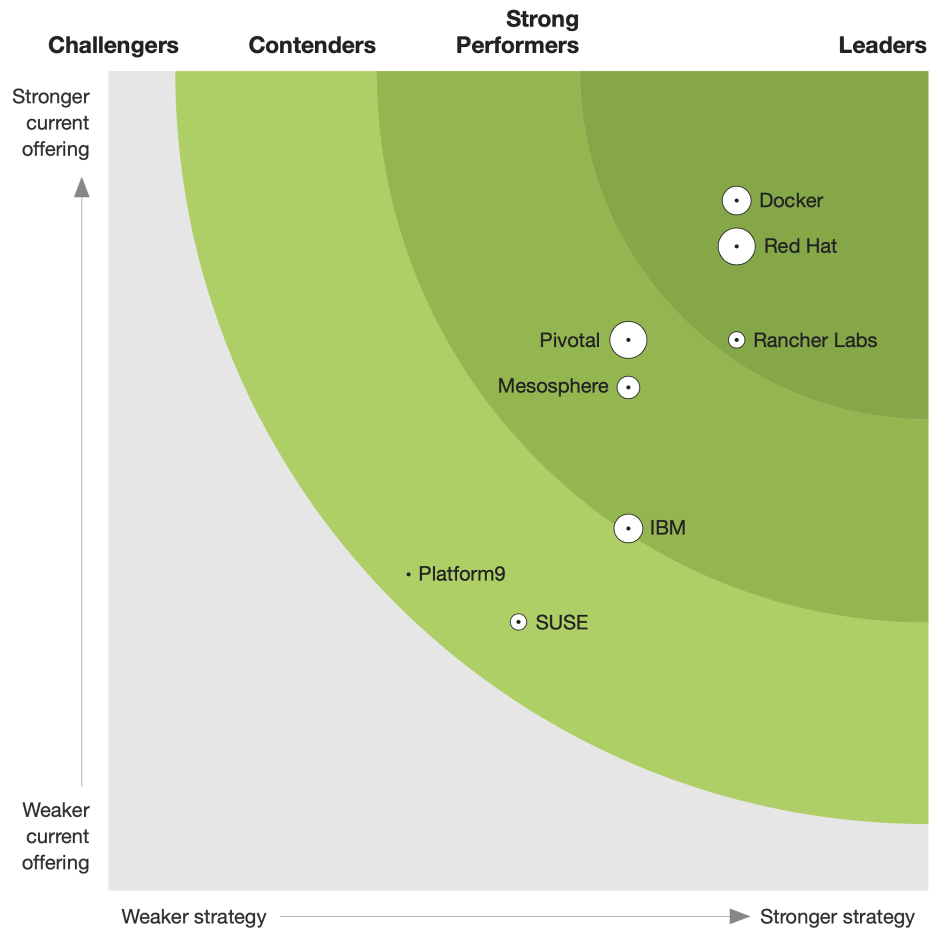
Swarm GUI	Docker Enterprise (UCP)	
Central Monitoring	Prometheus	Sysdig
Central Logging	Docker for AWS/Azure	
Layer 7 Proxy	Docker Enterprise (UCP)	
Registry	Docker Enterprise (DTR)	
CI/CD	Jenkins	GitLab
Storage	NetApp / Portworx / CSI	
Networking	Swarm Overlay / Calico	
Orchestration	Docker Swarm / Kubernetes	
Runtime	Docker Enterprise	
HW / OS	Docker Enterprise	

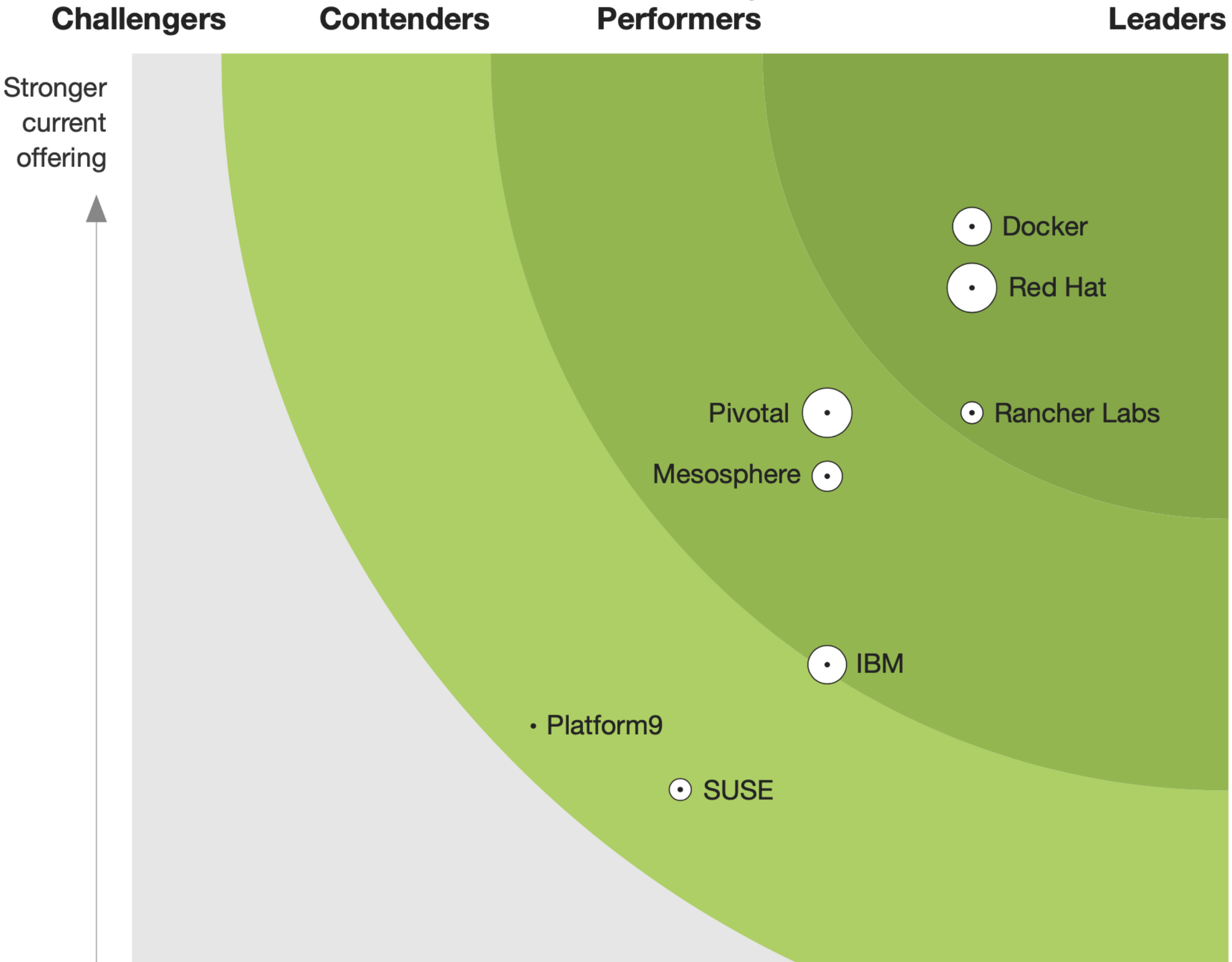
Also
Image Security Scanning
Role-Based Access Cont
Image Promotion
Content Trust

THE FORRESTER NEW WAVE™

Enterprise Container Platform Software Suites

Q4 2018




Challengers
Contenders
Performers
Leaders
Stronger current offering
Docker
Red Hat
Pivotal
Rancher Labs
Mesosphere
IBM
Platform9
SUSE

Please 😬

Remember to rate this session

Thank you!

Thank You! 🤘

@bretfisher

Free 15 Hour Course: bret.show/gotoberdocker

Slide resources: bretfisher.com/docker