



# Taking Docker to Production

## Bret Fisher

DevOps Consultant  
Docker Captain  
Author of Udemy Docker Mastery

# Why Are We Here?

- Want Docker in production
- Want to orchestrate containers
- Need to make educated project decisions
- Learn which requirements could be optional
- Learn 80's/90's video games
- Hear bad analogies relating retro games to Docker



# A Bit About Me

- Geek since 5th Grade
- IT Sysadmin+Dev since 1994
- Owned \*REAL\* Atari 2600, NES, SNES, Sega Genesis, Sinclair, TRS-80, Packard Bell 386
- Like Geek Trivia. Lets Have Some!





# STREET FIGHTER II TURBO

HYPER FIGHTING

© CAPCOM 1991, 92, 93

© CAPCOM U.S.A., INC. 1991, 92, 93

PRELIMINARY

1ST MATCH



4P



5P



1P



8P



# Project Docker

Super Project Advice Special Turbo Champion Edition

# 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 scaling
  - Containerizing all or nothing
  - Starting with persistent data

# 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



# Dockerfile Power-Ups

# What To Focus On First: Dockerfiles

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

# Dockerfile Maturity Model

- Make it start
- Make it log all things to stdout/stderr
- Make it documented in file
- Make it work for others
- Make it lean
- Make it scale

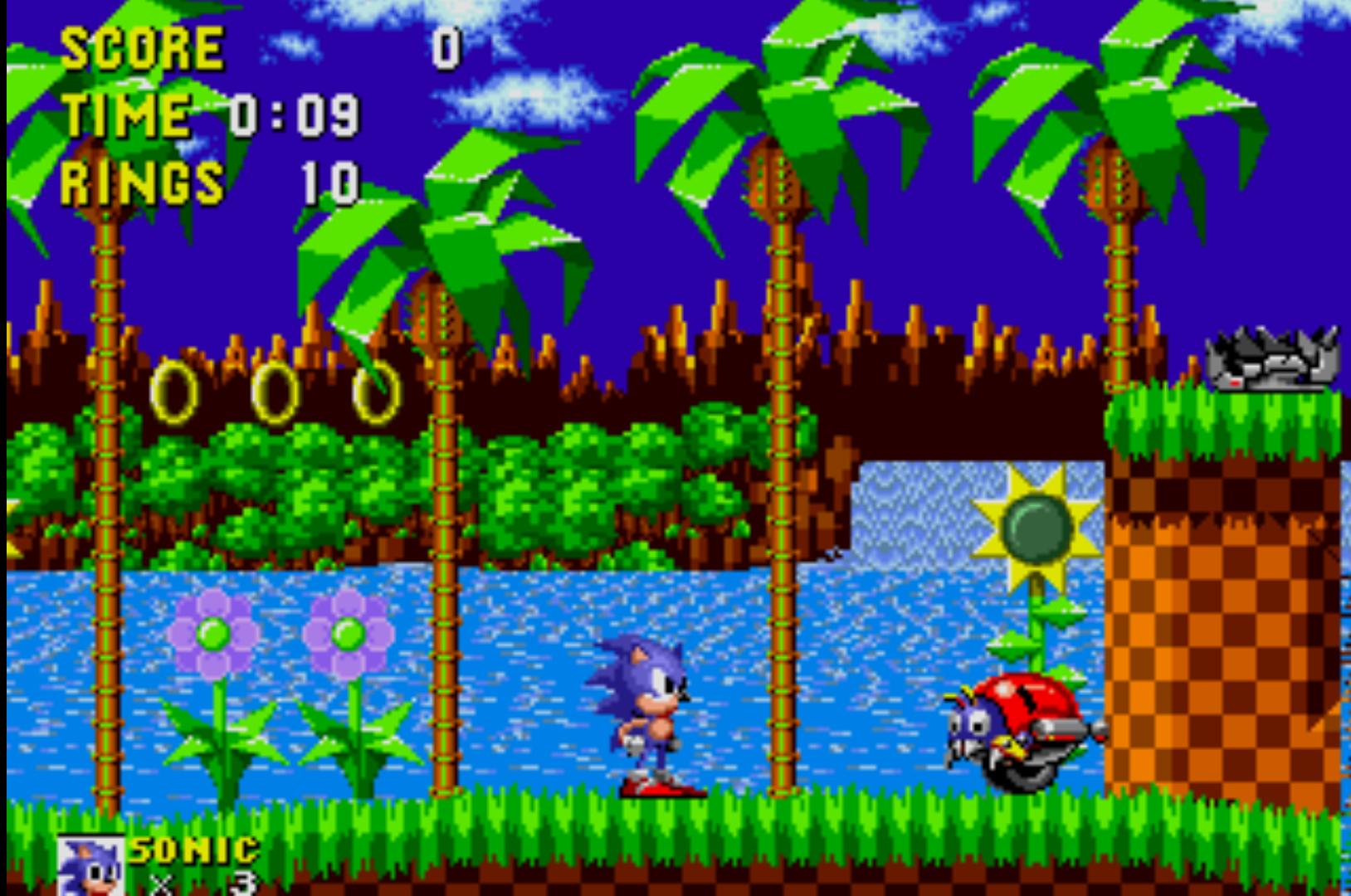
**SCORE**

**TIME 0:09**

**RINGS**

**10**

**0**



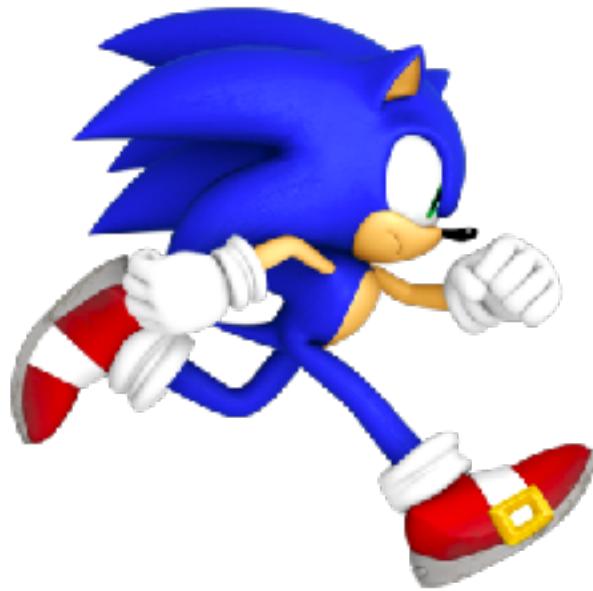
**SONIC**

**x 3**

A pixelated title screen for the game Sonic the Hedgehog. In the center, Sonic the Hedgehog is shown from the waist up, wearing his signature red and blue suit. He is looking towards the right. Behind him is a large, stylized title "SONIC" in blue and white, with a red outline. Below the main title, the words "THE HEDGEHOG" are written in a smaller, white, sans-serif font. The background features a bright blue sky with two large, white, stylized "F" shapes on either side of Sonic's head. At the bottom, there is a dark blue and red decorative border.

# SONIC

THE HEDGEHOG



# Dockerfile Anti-patterns



# Dockerfile Anti-pattern: Trapping Data

- Problem: Storing unique data in container
- Solution: Define VOLUME for each location

```
VOLUME /var/lib/mysql  
  
ENTRYPOINT ["docker-entrypoint.sh"]  
  
CMD ["mysqld"]
```

# Dockerfile Anti-pattern: Using Latest

- Latest = Image builds will be 
- 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~jessie  
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 \  
    libpq5=9.5.4-1+deb8u1 \  
    libssl1.0.2=1.0.2g-1+deb8u1 \  
    libxml2=2.9.4-1+deb8u1 \  
    libyaml-0-2=0.1.6-1+deb8u1 \  
    libyaml-cpp0.6=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-dev=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-dbg=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-doc=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-samples=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-udeb=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-dbg-udeb=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-doc-udeb=0.6.2-1+deb8u1 \  
    libyaml-cpp0.6-samples-udeb=0.6.2-1+deb8u1
```

# 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_BUFFER_SIZE=64M \
    MYSQL_FLUSHLOG=1 \
    MYSQL_FLUSHMETHOD=0_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_BUFFER_SIZE}" >> ${MYSQL_CONFIG} && \
echo "innodb_flush_log_at_trx_commit = ${MYSQL_FLUSHLOG}" >> ${MYSQL_CONFIG} && \
echo "innodb_flush_method = ${MYSQL_FLUSHMETHOD}" >> ${MYSQL_CONFIG}
```

# Dockerfile Anti-pattern: Environment Specific

- Problem: Copy in environment config at image build
- Solution: Single Dockerfile with default ENV's, and overwrite per-environment with ENTRYPOINT script

```
① Dockerfile ➔  
2 FROM node:6.10  
1  
3 COPY test-environment.json test-environment.json  
1 #COPY dev-environment.json dev-environment.json  
2 #COPY prod-environment.json prod-environment.json
```





Don Bluth Presents

# DRAGONS LAIR®



Tap to Begin

0000049

A cartoon illustration of a knight in red armor with a white plumed helmet, pointing his right fist forward with a determined expression. He is standing in a stone castle hallway with a doorway behind him. In the top right corner, there is a digital timer displaying "0000049".

Lets Slay Some Infrastructure Dragons  
The Big 3 Decisions

# 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/dockercon17eu](http://bretfisher.com/dockercon17eu)

# OS Linux Distribution/Kernel Matters

- Docker is very kernel and storage driver dependent
- Innovations/fixes are still happening here
- "Minimum" version != "best" version
- No pre-existing opinion? Ubuntu 16.04 LTS
  - Popular, well-tested with Docker
  - 4.x Kernel and wide storage driver support
- Or InfraKit and LinuxKit!
- Get correct Docker for your distro from [store.docker.com](https://store.docker.com)

# Container Base Distribution: Which One?

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

LUMBER: 200

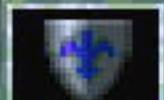
GOLD: 1000

MENU



HP

KNIGHT





Build Your Empire Swarm

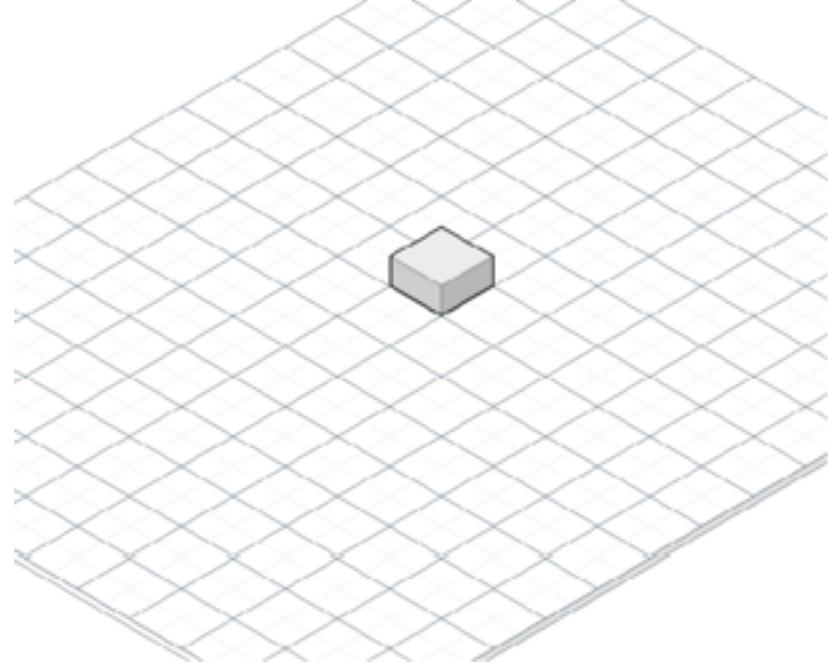
WARCRAFT  
ORCS & HUMANS

# Good Defaults: Swarm Architectures

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

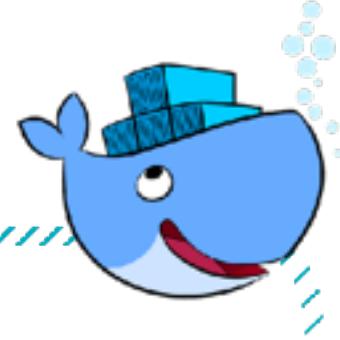
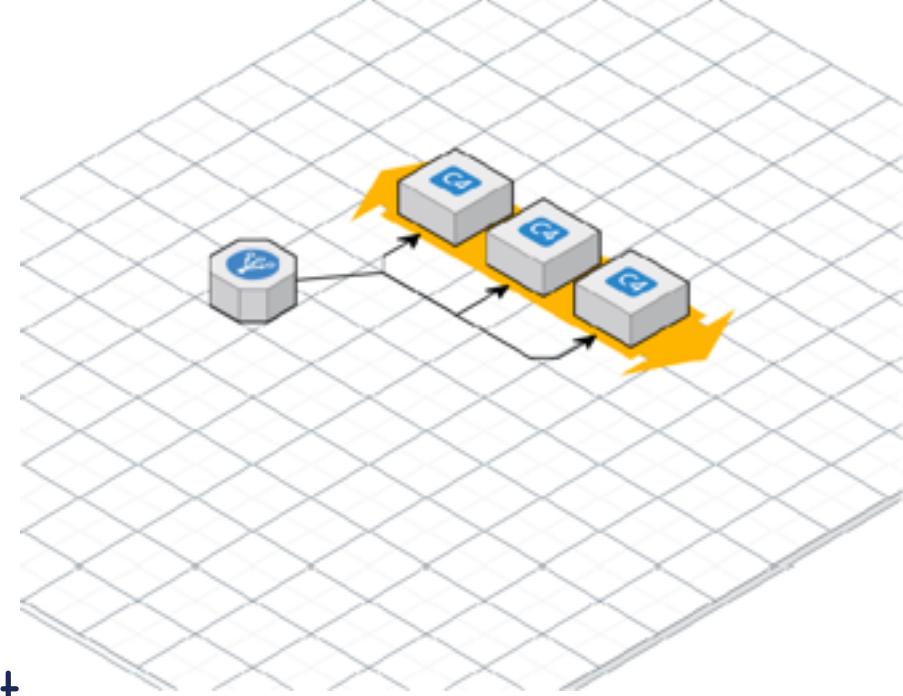
# Baby Swarm: 1-Node

- "docker swarm init" done!
- Solo VM's do it, so can Swarm
- Gives you more features than docker run



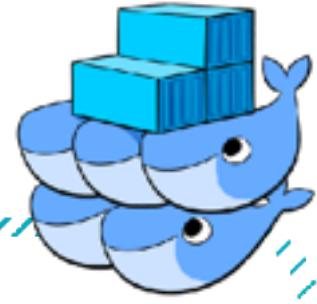
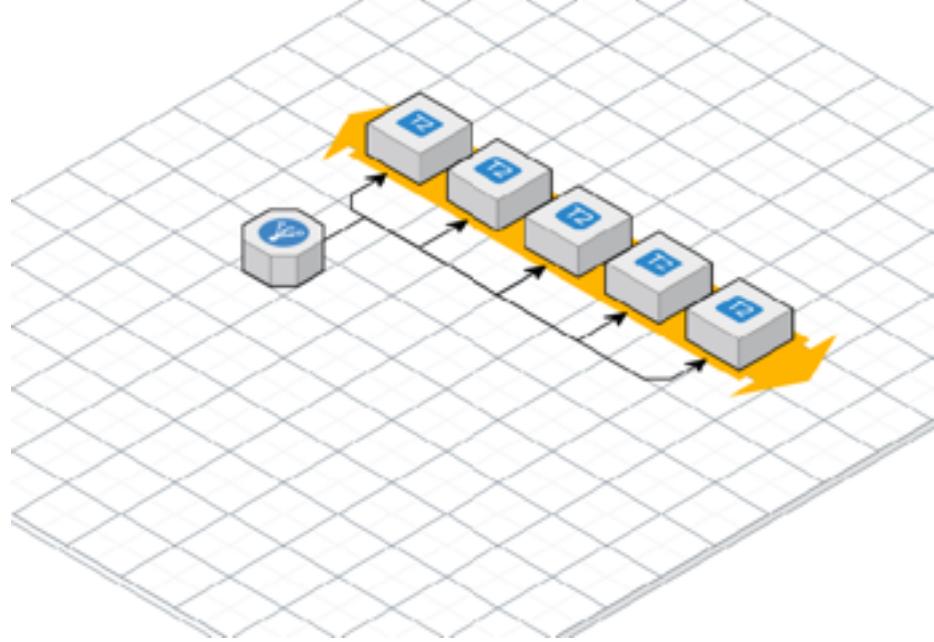
# HA Swarm: 3-Node

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



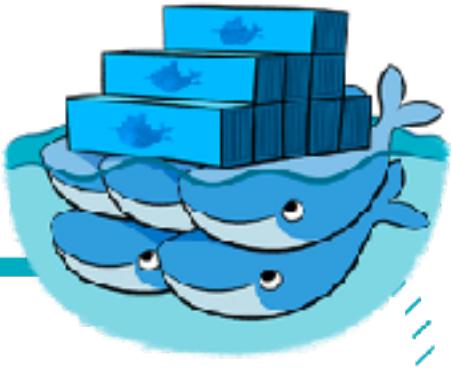
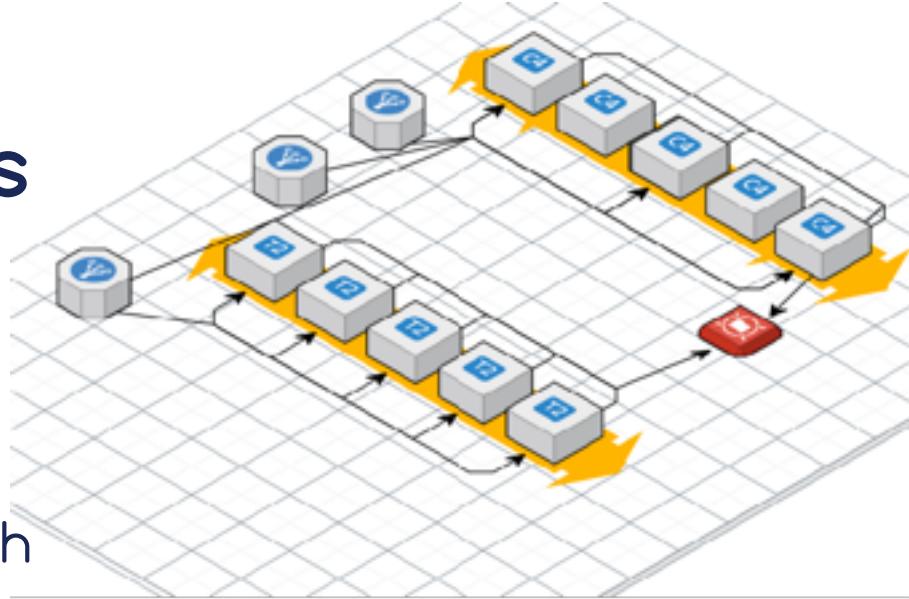
# Biz Swarm: 5-Node

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



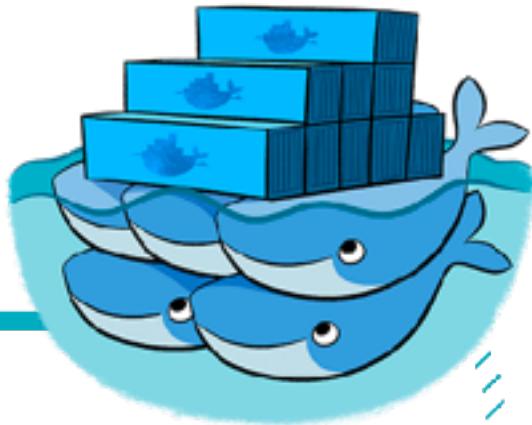
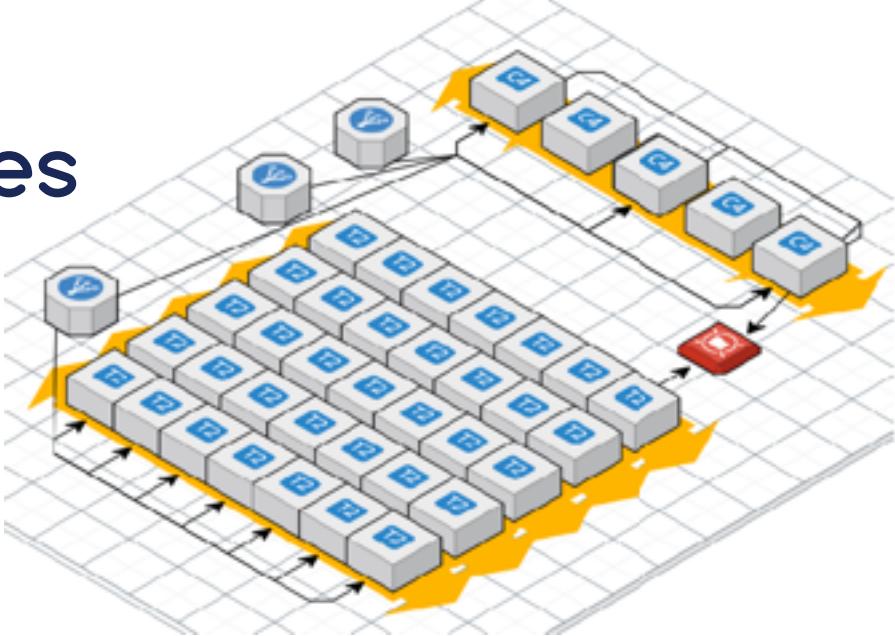
# Flexy Swarm: 10+ Nodes

- 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

- 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

- Assume nodes will be replaced
- Assume containers will be recreated
- Docker for (AWS/Azure) does this
- LinuxKit and InfraKit expect it

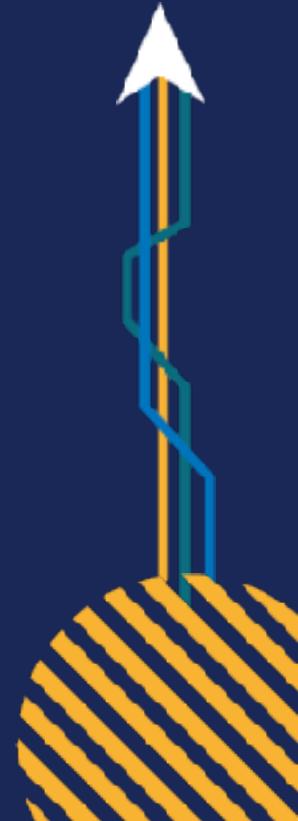
# Reasons for Multiple Swarms

## 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 2016 Swarm?

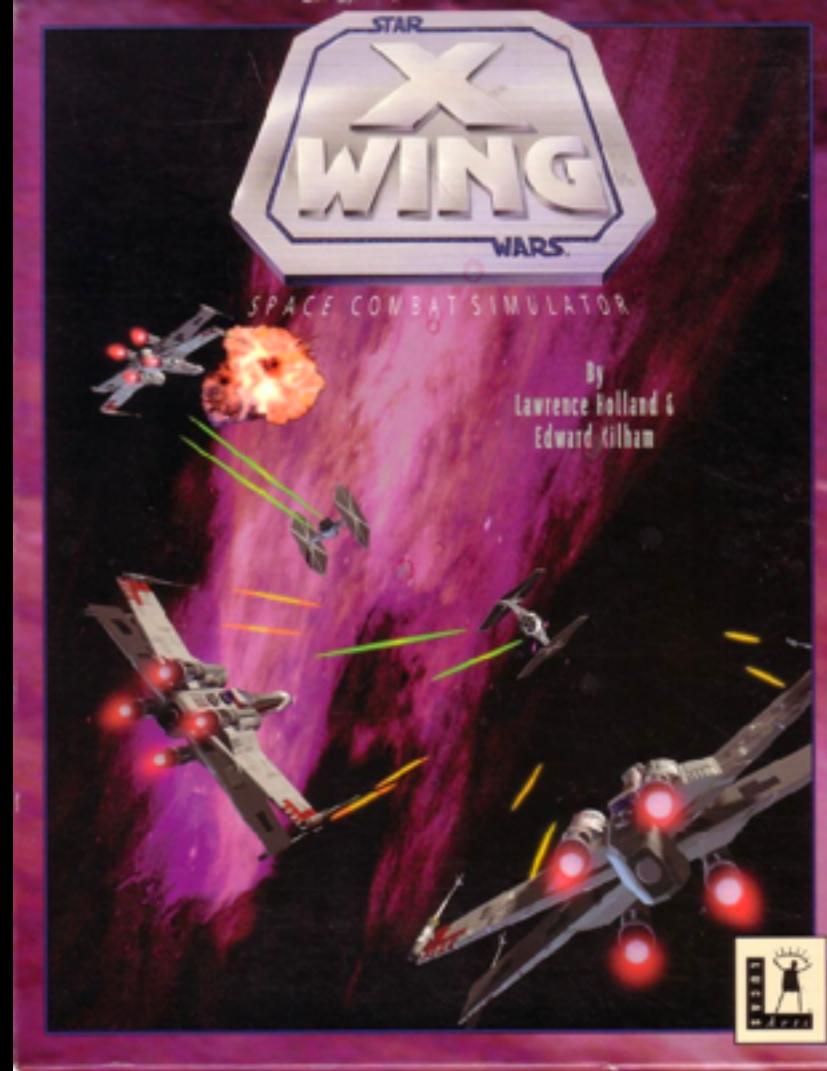
- Hard to be "Windows Only Swarm", mix with Linux nodes
- Much of those tools are Linux only
- Windows = Less choice, but easier path
- My recommendation:
  - Managers on Linux
  - Reserve Windows for Windows-exclusive workloads



Proton torpedo fired.



# Bring In Reinforcements



# Outsource Well-Defined Plumbing

- Beware the "not implemented here" syndrome
- If challenge to implement and maintain
- + SaaS/commercial market is mature
- = Opportunities for outsourcing

# Outsourcing: For Your Consideration

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



# Tech Stacks

Designs for a full-featured cluster

# Pure Open Source Self-Hosted Tech Stack

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



# Docker for X: Cheap and Easy Tech Stack

Swarm GUI	Portainer	
Central Monitoring	Librato	Sysdig
Central Logging	Docker for AWS/Azure	
Layer 7 Proxy	Flow-Proxy	Traefik
Registry	Docker Hub	Quay
CI/CD	Codeship	TravisCI
Storage	Docker for AWS/Azure	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker	
HW / OS	Docker for AWS/Azure	



# Docker Enterprise Edition + Docker for X

Swarm GUI	Docker EE (UCP)	
Central Monitoring	Librato	Sysdig
Central Logging	Docker for AWS/Azure	
Layer 7 Proxy	Docker EE (UCP)	
Registry	Docker EE (DTR)	
CI/CD	Codeship	TravisCI
Storage	Docker for AWS/Azure	
Networking	Docker Swarm	
Orchestration	Docker Swarm	
Runtime	Docker EE	
HW / OS	Docker for AWS/Azure	





バーチャファイター



GHOULYER

LEVEL 4

WARRIOR

SCORE 650  
HEALTH 605

VALKYRIE

4x SCORE 7260  
HEALTH 214

WIZARD

5x SCORE 4720  
HEALTH 447

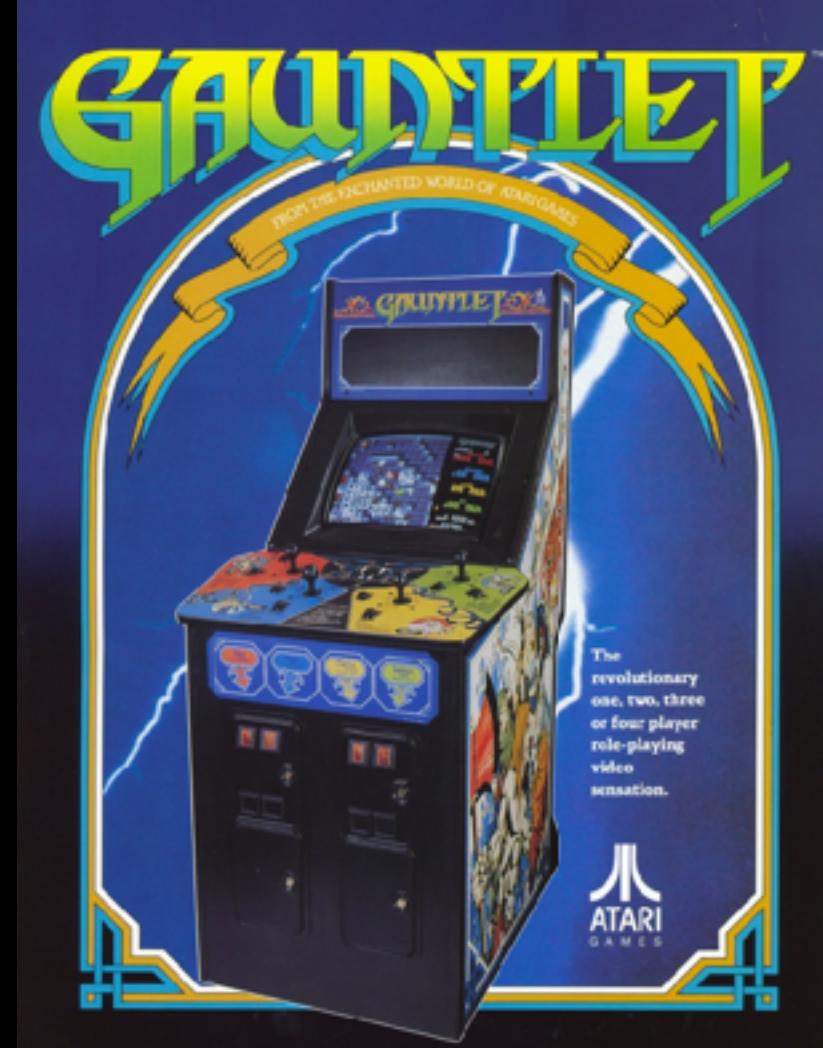
ELF

3x SCORE 8030  
HEALTH 2900

©1985  
ATARI GAMES



# Ready Player One?





# 4 Can Co-Op, But 1 Plays Just Fine



# Must We Have An Orchestrator?

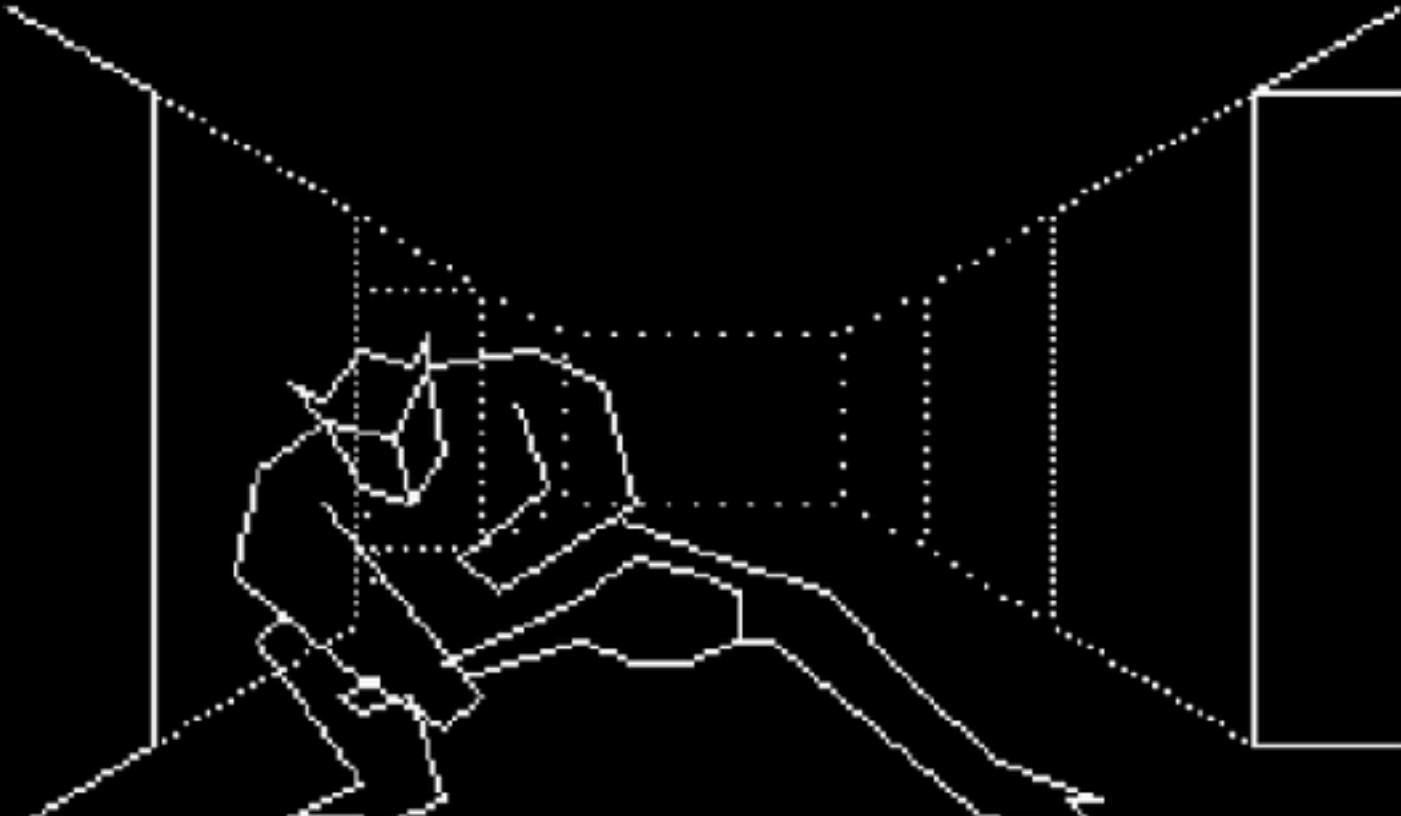
- Let's accelerate your docker migration even more
- Already have good infrastructure automation?
- Maybe you have great VM autoscale?
- Like the security boundary of the VM OS?

# One Container Per VM

- Why don't we talk about this more?
- Least amount of infrastructure change but also:
  - Run on Dockerfiles recipes rather than Puppet etc.
  - Improve your Docker management skills
  - Simplify your VM OS build

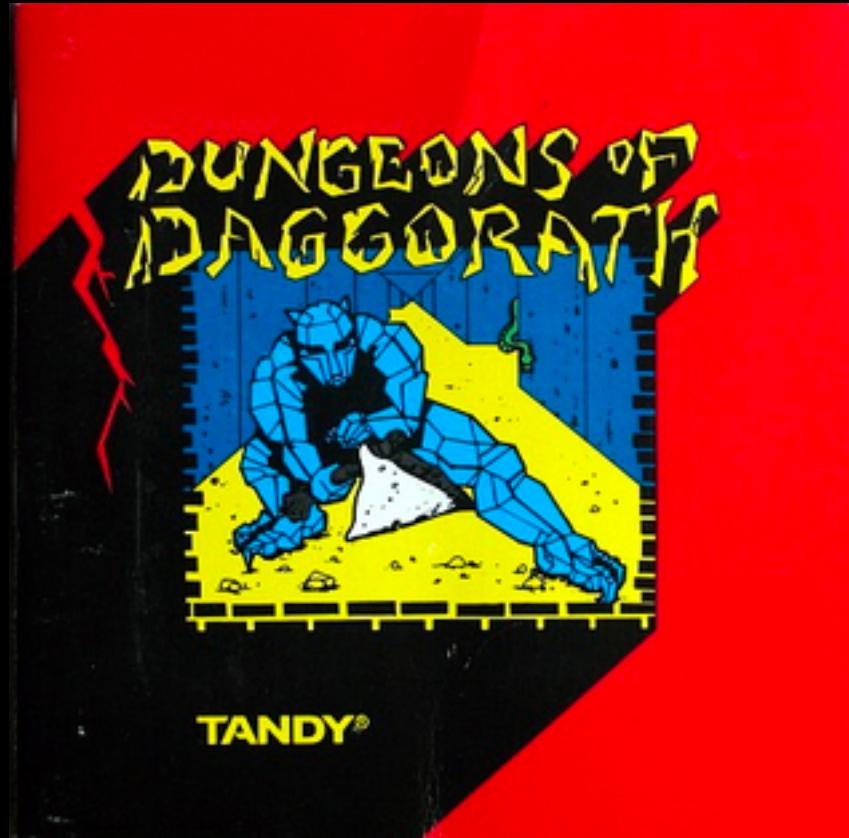
# One Container Per VM: Not New

- Windows is doing it with Hyper-V Containers
- Linux is doing it with Intel Clear Containers
- LinuxKit will make this easier: Immutable OS
- Watch out for Windows "LCOW" using LinuxKit



LEATHER SHIELD \* WOODEN SWORD

A	R
A	R
A	R
-	-





# Summary

- Trim the optional requirements at first
- First, focus on Dockerfile/docker-compose.yml
- Watch out for Dockerfile anti-patterns
- Stick with familiar OS and FROM images
- Grow Swarm as you grow
- Find ways to outsource plumbing
- Realize parts of your tech stack may change, stay flexible

# Give Session Feedback in App!

- Help me come back next year





# Thank You!

Slides: [bretfisher.com/dockercon17eu](http://bretfisher.com/dockercon17eu)

- My Bestselling Docker Mastery Video Course
  - 90% off for DockerCon
  - [bretfisher.com/dockermastery](http://bretfisher.com/dockermastery)



# Honorable Mentions

- Metroid ('83 NES)
- Mega Man ('87 NES)
- Wolfenstein 3D ('92 PC)
- Homeworld ('99 PC)
- Legend Of Zelda ('86 NES)
- Mortal Kombat ('92)
- Doom/Quake ('93 PC)
- Contra/Castlevania ('86 NES)
  - Hitchhiker's GTTG ('84 TRS-80)
- Zenophobe ('87 Arcade)
- Battlezone ('80 Arcade)

