



**CHEF**™

# **Configuration Management is Old and Boring**

Mandi Walls | DevOpsDays Oslo 2017 | [mandi@chef.io](mailto:mandi@chef.io)



# In the Beginning

- Big expensive machines
- Wizard-like Operators
- Specific tools per platform

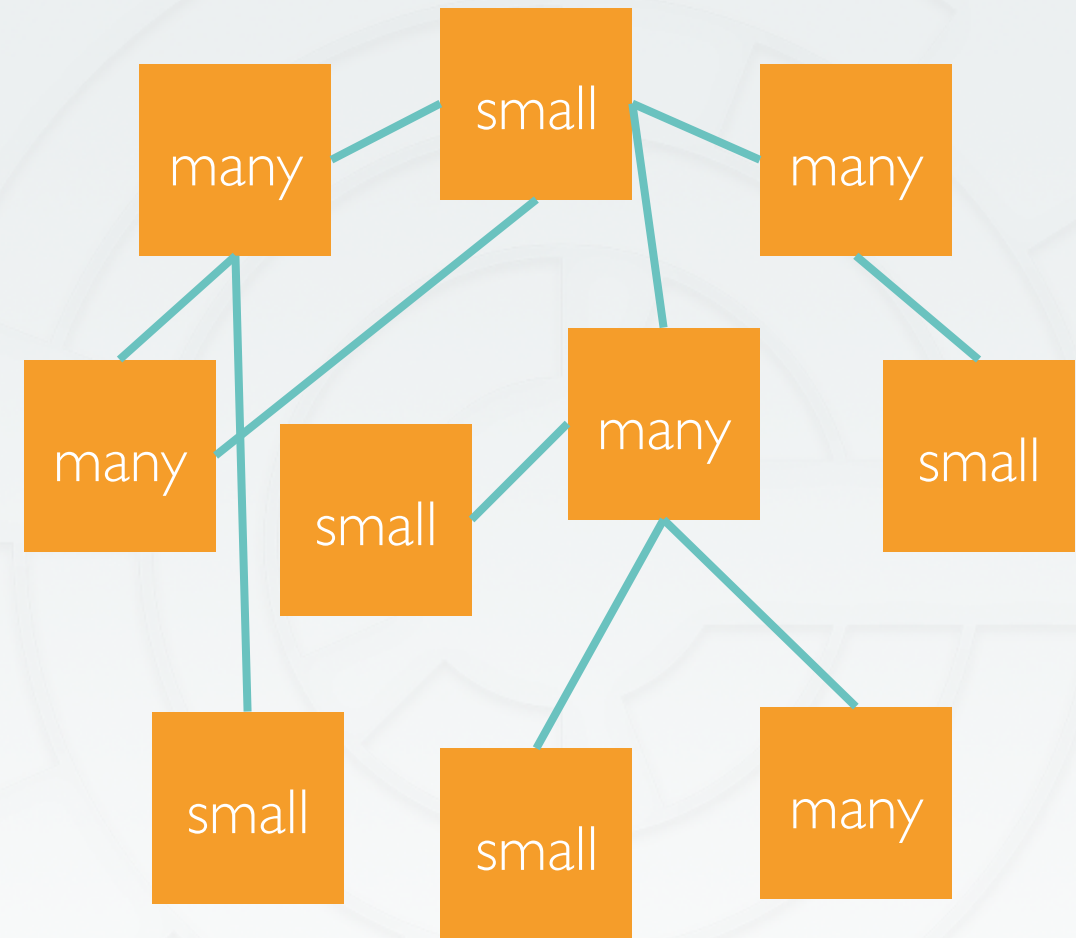


# Now

- Machines are cheap
- Disk is cheap
- Rent by the quarter hour
- That future is here, but is not evenly distributed

# Complexity

**One  
Big  
Machine**  
€∞



- Technology a key component of success in many industries
- “IT is a cost center” post-bust, to “Software is eating the world” 10 years later



# Waterfall to Agile Era

- Great for software developers
- Ship it and Forget it
- Stranded operators at the end of a dev cycle or sprint
- Ignored the installation and running of software

# Early Cloud

- VMs and hardware-on-demand changed the need for Configuration Management
- Web 2.0 created larger and larger environments
- Beginning of the value proposition for investment in CM

# Arbitrarily Defined DevOps Era

- Tool proliferation across all spectrums of the Operations lifecycle
- Open Source tools gain status, market
- Large-scale operations and system management gains voice as a practice apart from “traditional” systems administration

Distinct from the needs of HPC

# Where To?

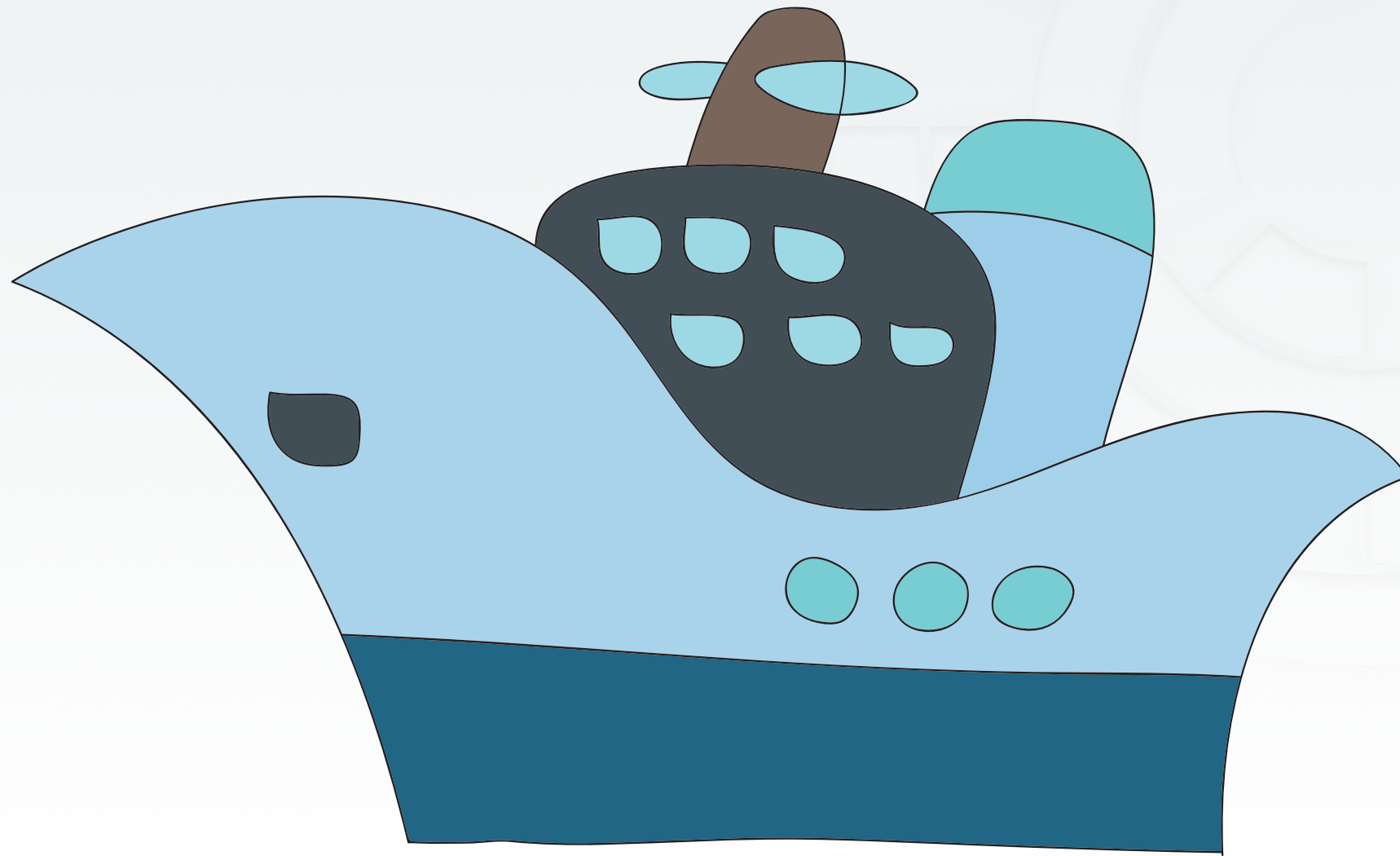
- Transitioning to newer practices requires stable foundations
- Building software faster means having environments that match
- Infringing a bit more on build and release cycles rather than staying in Operations



# Managing Lifecycle Environments

- The biggest complaint we hear from teams moving towards modern practices
  - “None of our environments match”
- Production operations team doesn't manage non-prod assets
- No sharing of resources, knowledge, tools hinders good code quality

# **Always Shipping Means Always Automating**



# Get to Market Faster

- Requires as much culture work as it does tools work
- Automation of all parts of the pipeline
- Releases and release installs lose complexity and angst

# Reliability and Repeatability are Key

- Migration away from GUI-based tools
- Treating the entire environment as a code base
- Risk reduction through smaller releases
- Traceable histories for all changes on all platforms and environments



# CM Helps You Go Fast

- Build new hosts that meet your requirements
- Deploy code as often as necessary
- Make sure that monitoring, metrics collection, log management are in place
- Minimize time-to-market with automation

# Make Failing OK

- New environment is cheap!
- Experiment often!
- Run this command, get a new environment to work on!

# CM and Containers

- Early lifecycle, fast development
- Ecosystem still evolving

Monitoring, Logs, Management, Security, etc

# Everything is Faster

- Old eras had longer enterprise buying cycles
- 5-year plans for IT products, large investment
- Even though technology executives tend to only stay in a position for a couple of years



# Lots of Challenges

- IT buying cycle isn't always followed by an IT skill building cycle
- Writing long-term plans with a short-term personal outlook
- Doesn't reflect modern product cycles and technology ecosystem
  - 5 years is an eternity in “internet years”

# Long Tail is Long

- Software vendors sell to early adopters, mainstream adopters, and eventually to doubters
- Configuration Management, for all of its “been around forever”-ness is still selling to mainstream adopters
- The high-risk, conservative buyers, like insurance companies and utilities, still lag
- Shortage of available practitioners willing to work for those organizations slows down adoption as well

# We're Here To Help

- Infrastructure as Code
- Continuous deployment
- Testing workflows for all the changes headed for production



**CHEF**<sup>TM</sup>

<https://www.chef.io>