

The Five Phases of DevOps

Joel



DevOps Ninja @ 



@__jvasallo__



<http://github.com/jvasallo>



jovelasallo@gmail.com

Why The Five Phases of DevOps?

- × Making a big shift to a DevOps culture can be daunting challenge
- × Changing core fundamental processes in an organization can be seen as grief
- × Organizations should incorporate DevOps in their own natural and holistic manner

Phase One



Denial



Some things to watch out for

- ✗ “Everything is fine...” despite the fact that it is not.
- ✗ You may see attempts to establish an “automation culture”
- ✗ There is a problem? “Toss more engineers at it!”



What does DevOps mean?

- × Take the time early to define what DevOps means for not only yourself but also your organization.
- × DevOps == Automation...Right?
 - Not Quite...
- × DevOps is not just a team, it is a complete culture shift that spans multiple teams
 - Busting Silos
 - Empower Developers
 - Automate and support other teams to do the same



Busting Silos

Make deployments go from... ...to this!



4GIFs
.com



MakeAGIF.com

Eliminate Yak Shaving

“

“You're probably shaving yaks at work all the time and don't realize it.

Yak shaving is what happens when you're doing some stupid, fiddly little task that bears no obvious relationship to what you're supposed to be working on, but yet a chain of twelve causal relations links what you're doing to the original meta-task.”

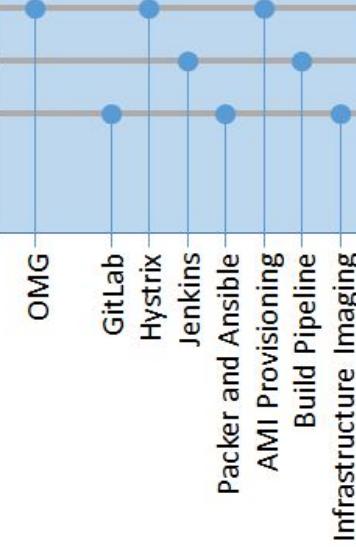
Have a well defined mission statement

- × Use this mission statement to drive your choices and answer tough questions
- × A great place to start is to:
 - “Empower your developers”
- × Ask yourselves:
 - “Will this proposed solution, implementation, or tool will help make your development team's lives easier?”





Q3



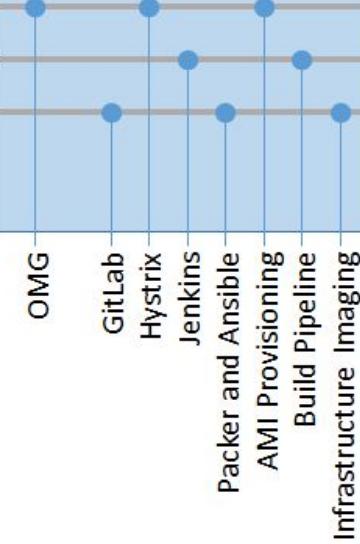
By the end of September, we could take a commit and turn it into a piece of immutable infrastructure.

How did we do this?
Phase 1: Denial





Q3



By the end of September, we could take a commit and turn it into a piece of immutable infrastructure.

'Immutable' is cool because it means that we are guaranteed the same installed OS and application code anytime and anyplace

How did we do this?
Phase 1: Denial

Phase Two



Fear/Doubt



Common Fears

- × Fear of changing process due to things “always having been done this way”
- × Fear of potential failure or making things worse
- × Fear of the unknown...some ask what does DevOps mean for me?

Doubt with DevOps

- ✗ This is your build and deployment pipeline phase.
- ✗ During this phase, folks (even folks on DevOps) will have questions:
 - Are we doing this right; is everything perfect?
 - How can you be sure it's 100% going to work?
- ✗ The short answer?
 - It's not perfect and it will never will be; that's OK!
 - Just make sure you have ability to improve in the future



Start Small; Set Standards

- ✗ Don't try to support too many projects at once...especially as you are maturing
- ✗ Pick a small homogenous base that can be built upon
- ✗ Don't compromise on standards you set
- ✗ Examples:
 - Multiple Base OSes vs Single Base OS
 - Support Java only vs Support both Java and JS



Deploy Fast, Deploy Often.

- × Make deploying the fastest part of your pipeline.
- × Remember that whatever you implement needs to be:
 - Easier to use
 - Equally performant in deployment/rollback
 - More available and reliable
- × Cookie cutter all the things! Consider an immutable architecture.
 - Infrastructure
 - Deployments
 - Rollbacks



Spinnaker



deck

PIPLINES 1

CLUSTERS

LOAD BALANCERS

SECURITY GROUPS

PROPERTIES

TASKS

CONFIG

Filters Clear All

Unpin

+ - Group by Pipeline Show 5 per group

Configure ▶

Start Manual Execution

Filtered by: PIPELINE: demo

SEARCH

PIPELINES

- templation
- wait forever on ci builds
- cbt
- migratable
- multiregion deploy
- sandbox
- major parallel
- minor parallel
- long wait
- demo

Reorder Pipelines

STATUS

- Running
- Failed
- Completed
- Not Started
- Canceled
- Stopped

1 demo

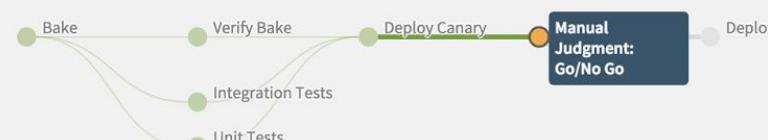
Configure

Start Manual Execution

MANUAL START
christb@netflix.com
2015-11-09 21:41:16 PST

Status: RUNNING

Duration: 01:19



Hide Details ×

MANUAL JUDGMENT: GO/NO GO DETAILS

▶ MANUAL JUDGMENT: GO/NO GO

Manual Judgment Task Status

Step	Started	Duration	Status
------	---------	----------	--------

Manual Judgment: Go/No Go	2015-11-09 21:42:10 PST	00:25	RUNNING
---------------------------	-------------------------	-------	---------

Instructions

Verify the canary scores are correct and you haven't received any emails or something.

Continue

Stop

Source | Permalink



HashiCorp All the Things



TERRAFORM



Nomad



A tool for managing secrets.

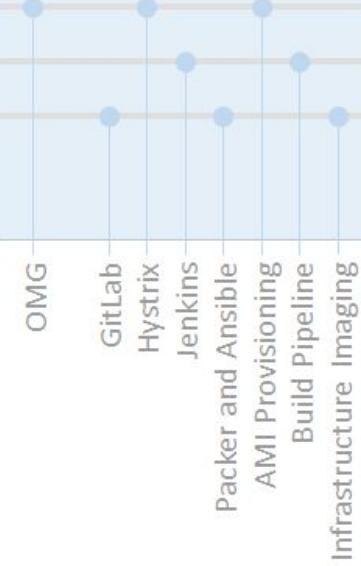


VAGRANT

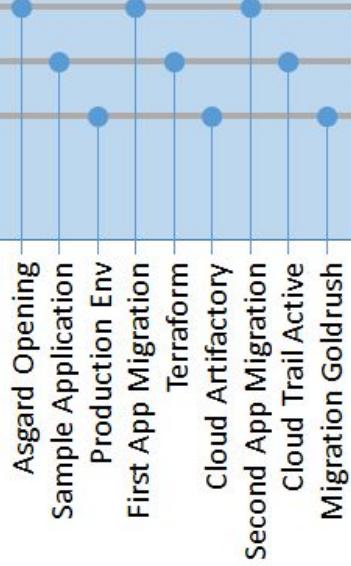




Q3



Q4



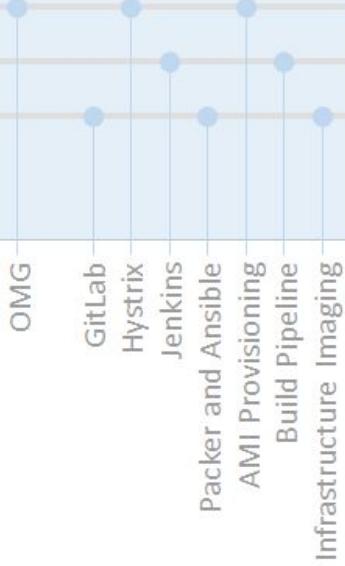
By the end of December, we were auto-deploying a half dozen applications through the pipeline to production.



How did we do this?
Phase 2: Fear/Doubt



Q3



Q4



By the end of December, we were auto-deploying a half dozen applications through the pipeline to production.

'Deploying' means creating Elastic Load Balancer, Auto-scaling Group, DNS Entries, Security Groups, IAM Roles, Tagging, Slack Deployment Notifications, etc.

How did we do this?
Phase 2: Fear/Doubt



Phase Three



Compromise



What do we really need?

- × Ask why specific processes exist
- × Evangelize the concept of DevOps across your organization
 - DevOps covers both technical and non-technical teams
- × What can be reused? What can be deprecated?
 - Use metrics in your favor
 - Explain benefits of change
 - Improve support



Understand Technical Debt

- ✗ By this phase you may have already incurred technical debt

“

“You are not special. You’re not a beautiful and unique snowflake.”

- Tyler Durden, *Fight Club* (1999)



“There is never a tomorrow but there is always a today.”

Understand Technical Debt

- × By this phase you may have already incurred technical debt
- × Early on you had the ability to try to make everything perfect
- × Don't find yourself reinventing the wheel
 - What happens when you need to invent the wheel?
- × Don't force a tool to do something it was not built to do.



Goodbye QA...Hello Quality Engineering

- × Responsibilities still the same; QE still writes automated tests.
- × QE now is empowered to write code and test as close to source as possible.
- × Testing is fully integrated into the pipeline and tests should be run after every deployment
- × Testing through the pipeline should provide not only fast feedback but accurate feedback as well



Metrics/Monitoring: A First Class Citizen

- × Metrics and Monitoring teams are now DevOps as well
- × Build a service offering to eliminate guess work of tools, establish agreed upon alerts and quotas
- × Developers are now on call so it's important they provide accurate health of their application
 - OS Level Metrics
 - Application Level Metrics
 - Service Level Metrics



Hystrix

Circuit

Sort: Error then Volume | [Alphabetical](#) | [Volume](#) | [Error](#) | [Mean](#) | [Median](#) | [90](#) | [99](#) | [99.5](#)

[Success](#) | [Short-Circuited](#)

getApplicableProducts

99	0	0.0 %
0	0	
0	0	

Host: 9.9/s

Cluster: 9.9/s

Circuit **Closed**

Hosts	1	90th	752ms
Median	254ms	99th	1361ms
Mean	344ms	99.5th	1508ms

getSecurityQuestions

2	0	0.0 %
0	0	
0	0	

Host: 0.2/s

Cluster: 0.2/s

Circuit **Closed**

Hosts	1	90th	1190ms
Median	664ms	99th	1206ms
Mean	772ms	99.5th	1206ms

availableProducts

101	0	0.0 %
0	0	
0	0	

Host: 10.1/s

Cluster: 10.1/s

Circuit **Closed**

Hosts	1	90th	753ms
Median	255ms	99th	1361ms
Mean	345ms	99.5th	1510ms

checkSecurityAnswer

1	0	0.0 %
0	0	
0	0	

Host: 0.1/s

Cluster: 0.1/s

Circuit **Closed**

Hosts	1	90th	1455ms
Median	1041ms	99th	1463ms
Mean	1046ms	99.5th	1463ms

getCustomer

2	0	0.0 %
0	0	
0	0	

Host: 0.2/s

Cluster: 0.2/s

Circuit **Closed**

Hosts	1	90th	1861ms
Median	1458ms	99th	1865ms
Mean	1359ms	99.5th	1865ms

getAllSecurityQuestions

1	0	0.0 %
0	0	
0	0	

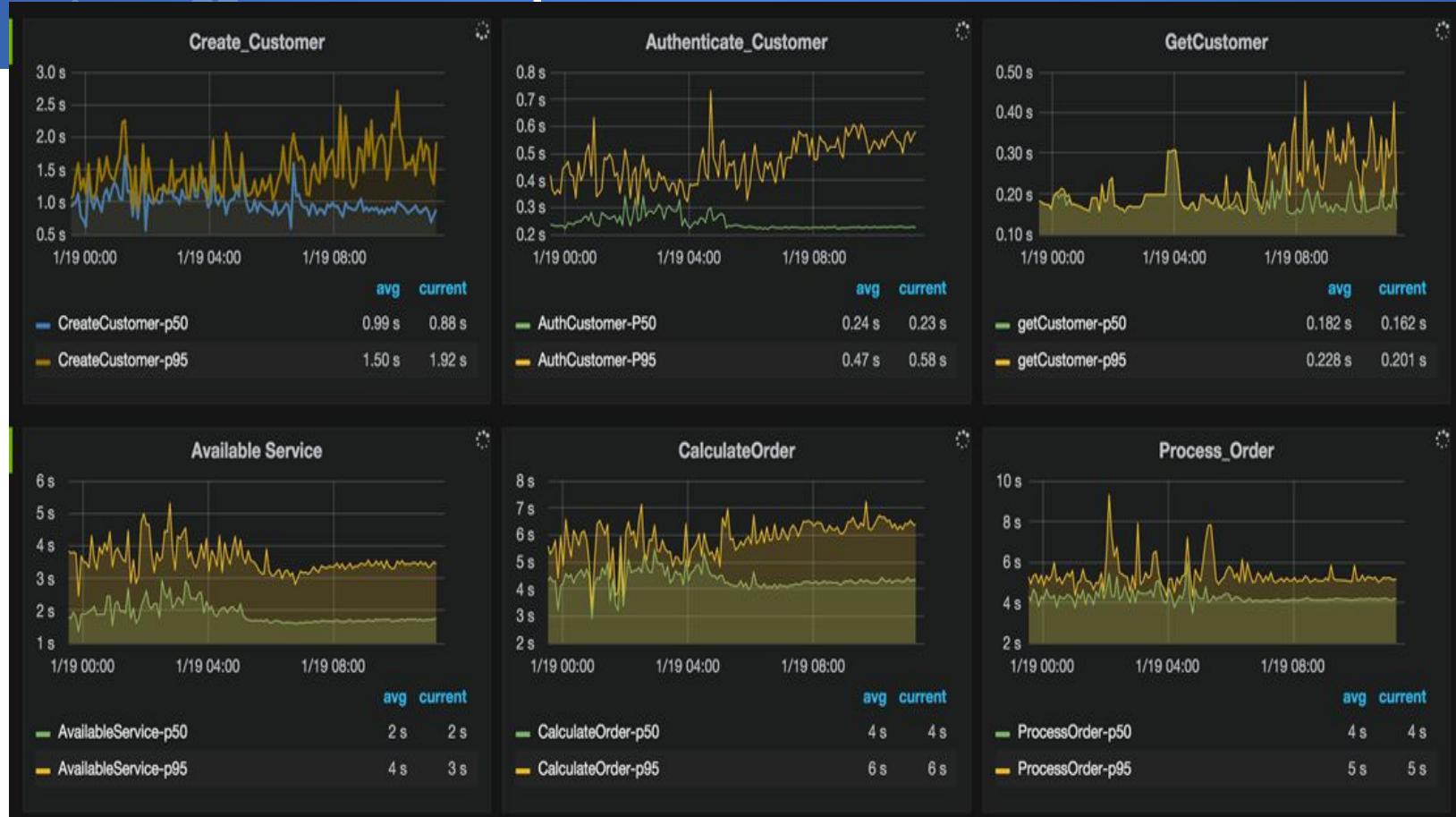
Host: 0.1/s

Cluster: 0.1/s

Circuit **Closed**

Hosts	1	90th	1472ms
Median	874ms	99th	1487ms
Mean	979ms	99.5th	1487ms

Graphite + Grafana



Q3

By the end of March, we had audit trail in place, standardization, QE integrated, and a bunch more deployed applications.

OMG
GitLab
Hystrix
Jenkins
Packer and Ansible
AMI Provisioning
Build Pipeline
Infrastructure Imaging

Asgard Opening
Sample Application
Production Env
First App Migration
Terraform
Cloud Artifactory
Second App Migration
Cloud Trail Active
Migration Goldrush

Q1

Security
Logging Standards
VictorOps Config/Deployed
Eureka/Archaius
Quality Engineering Pipeline
Spinnaker
Costs Understood
Developer Maintained Metrics



How did we do this?
Phase 3: Compromise

Phase Four



Acceptance



People like us! They really like us!

- × People are going to begin having educated conversations
 - How do I make my code better?
 - Do I have circuit breakers for when things go wrong?
 - Do I have accurate and meaningful metrics?
- × At this phase, early adopters are starting to see a net positive.
- × You are going to see plenty of new customers
 - Majority will be cookie cutter
 - Now you can focus on edge case requirements

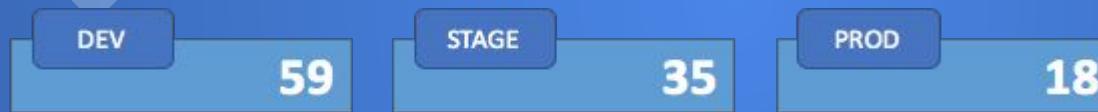


Some Deployment Statistics

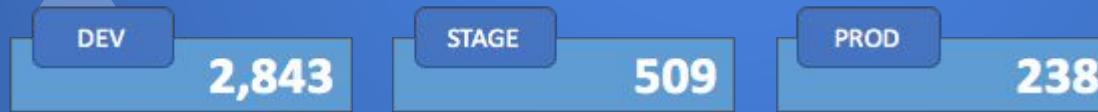
Traditional Deployments to PROD (all of 2015)

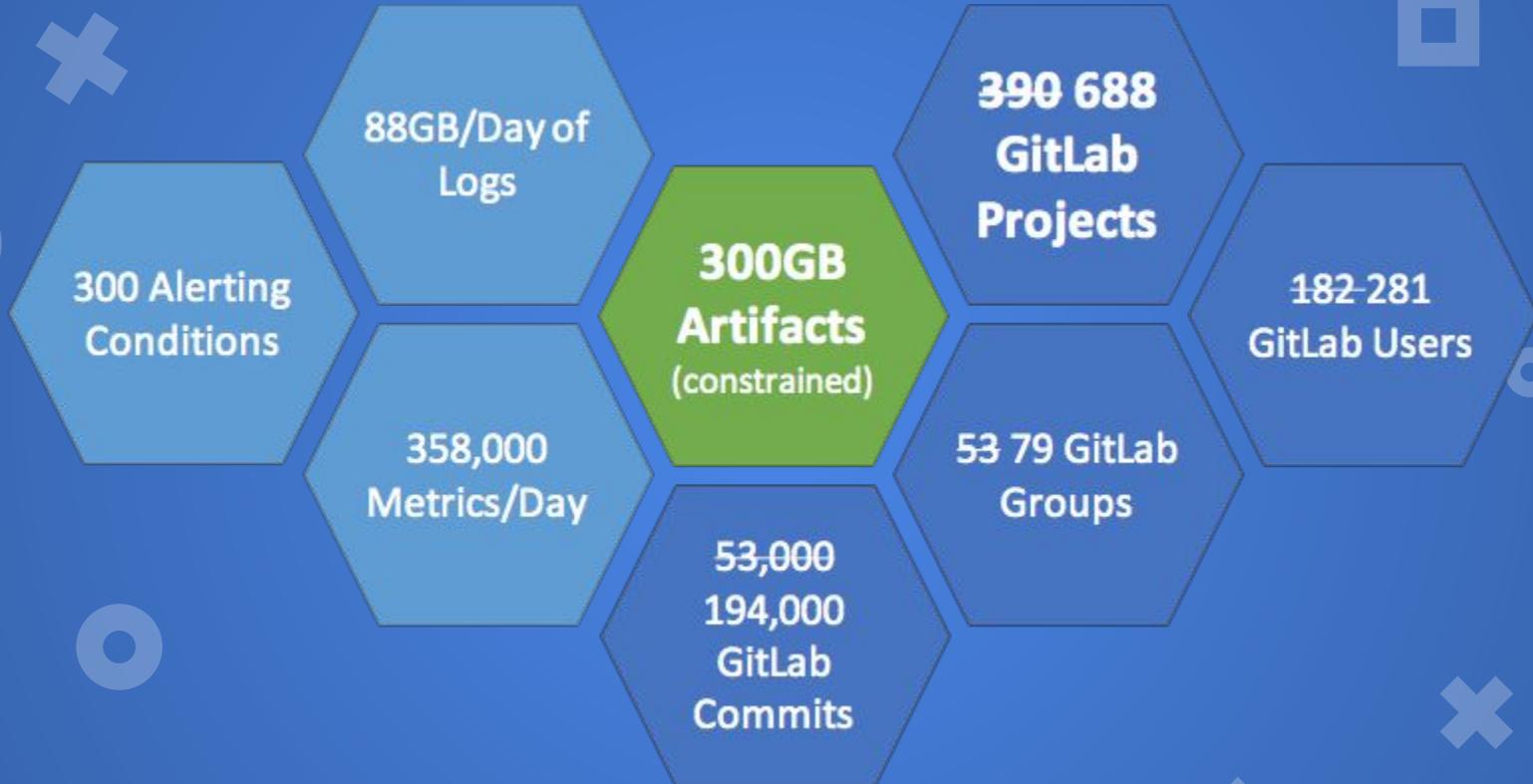


Number of Applications Deployed (Feb 1st - April 28th)



Total Deployments (Feb 1st 2016 - April 28th)







Q3

OMG
GitLab
Hystrix
Jenkins
Packer and Ansible
AMI Provisioning
Build Pipeline
Infrastructure Imaging

Q4

Asgard Opening
Sample Application
Production Env
First App Migration
Terraform
Cloud Artifactory
Second App Migration
Cloud Trail Active
Migration Goldrush

We're deploying PCI (Credit Card) and SOX (Book-Cooking Free) applications, potentially with the safety of multi-region support.

Q2

Security
Logging Standards
VictorOps Config/Deployed
Eureka/Archaius
Quality Engineering Pipeline
Spinmaker
Costs Understood
Developer Maintained Metrics
PCI and SOX
Cost Management
Multi-Region Support
DC Container Framework
Swagger Supported
Static Analysis Tools
Asgard Deprecated

How did we do this?
Phase 4: Acceptance



Phase Five



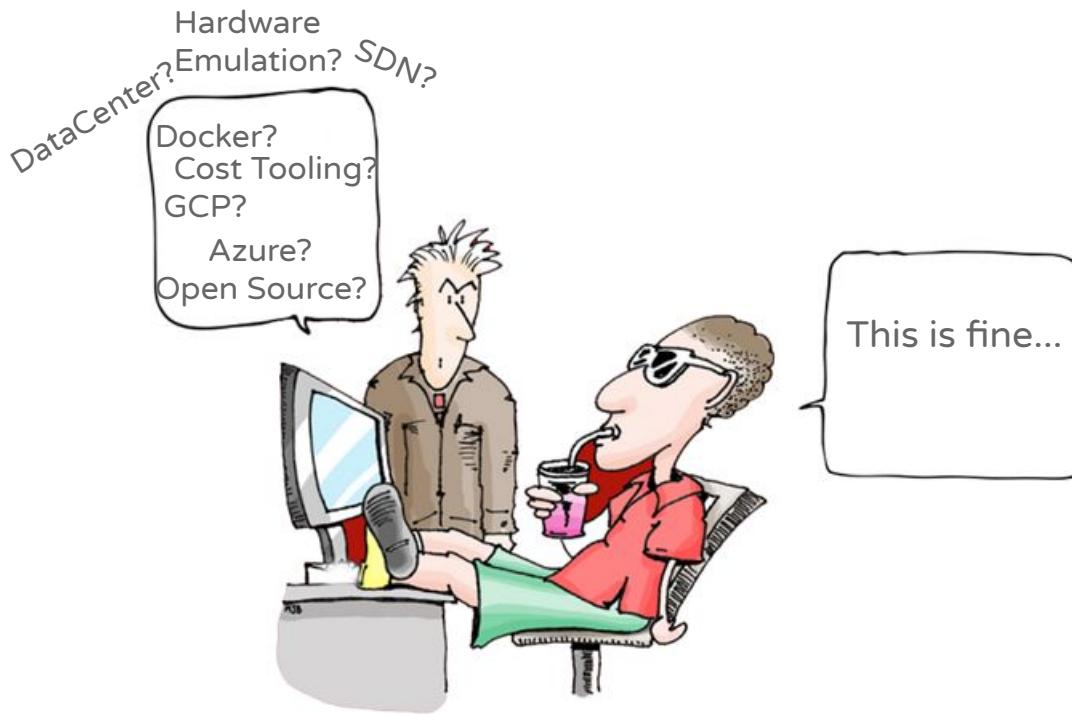
Relief and....Repeat



Relief and Repeat?

- × Things are pretty stable by now
- × App Engineers no longer need to use iTerm2 with 12 concurrent terminal sessions to restart all the nodes at once
- × Deployments are faster than ever
- × Developers are empowered to make their own changes and have increased expectations





So we are done right?
Phase 5: Relief and....Repeat



Already on Spinnaker?

- × Gogo is working to open source Foremast
 - × A Spinnaker pipelineDSL
- × Enables developers to version control pipelines and infrastructure requirements in a single place: the application repository
 - × One source of truth for application infrastructure information
 - × Infrastructure as Code
- × Sound interesting? Using Spinnaker today? Come talk after!

Thank You!



@__jvasallo__

<http://joelvasallo.com>



<https://www.gogoair.com/careers>