From Months to Minutes

How GE Appliances Brought Docker Into the Enterprise

Tom Barber – Cloud Computing Technologist
Brett Luckabaugh – Enterprise Software Architect

dockercon 15

SF JUNE 22-23

GE Appliances

GE Appliances is at the forefront of building innovative, energy-efficient appliances that improve people's lives.

- Headquartered in Louisville, KY
- $5 Billion+ in revenue
- 12,000+ employees

geappliances.com

GE APPLIANCES

Our Team

Byron Guernsey

Justin Roberts

Keenan Gizzi

Ryan Grothouse

Kevin Price

Tom Barber

Allan Clark

Jason Burks

Priya Ramaswamy

James Strong

Eric Sage

Brett Luckabaugh

About as enterprise IT as you can get…

1 day — Open project ticket

3 days — Meet with Project leader

3 days — Open infra tickets

1 week — Infra manually built

1 week — Infra audited

1 week — Software/app configured

2 weeks — Mistakes / rework

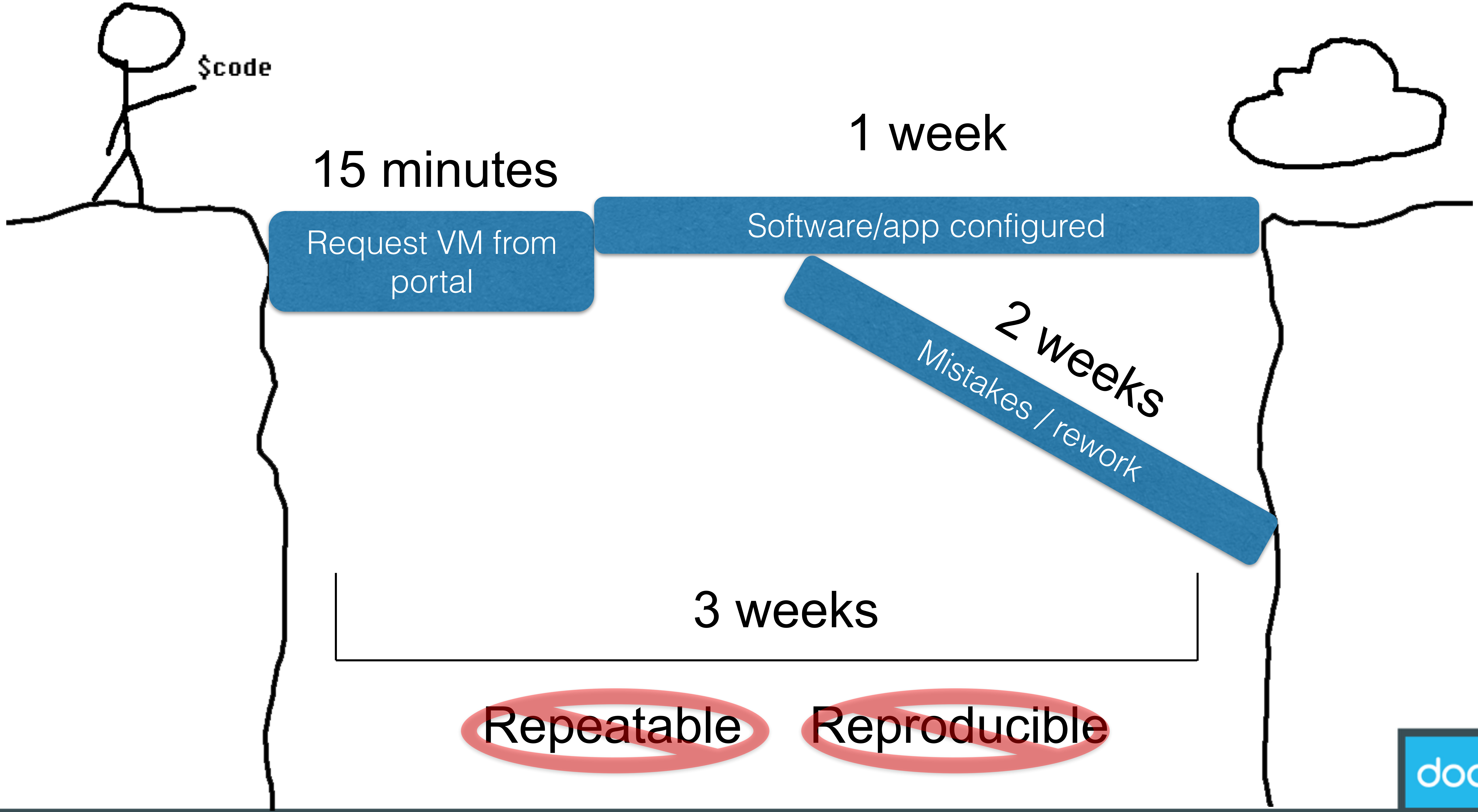
6 weeks!

~~Repeatable~~ ~~Reproducible~~

I’ll have 1 cloud please…

- Mid 2012: Formation of a cloud focused team with an agile culture
- Goals: Automate Infrastructure, Modernize Paradigm
- Leveraged Puppet
- 1 year to self-service IaaS solution

That’s a little more like it…



15 minutes

Request VM from portal

1 week

Software/app configured

2 weeks

Mistakes / rework

3 weeks

~~Repeatable~~ ~~Reproducible~~

Good start… Now what?

- 2013: Improve IaaS
- Exploring self-service application platform (PaaS) automation
- Leverage Puppet as our PaaS toolset
- App owners develop and contribute to puppet modules

Eh…maybe not so much

- Terrible rate of adoption
  - High barrier to entry for app owners
  - Reluctant to learn or write Puppet despite industry training
  - Highly heterogeneous environment with proprietary apps
- Our Team – The Bottleneck
- Generic Builds
  - Avoiding automation
  - Manually configured environments -> snowflakes
  - Access and privilege restrictions
  - Tickets and waiting
  - No way to quickly replicate environments

It’s not you…it’s me

- Early 2014 - Our team was having success with Puppet for IaaS and DBaaS
- However, our larger organization was not having success using Puppet for **self-service** PaaS
- But where do we go from here?

What about this Docker…

- Started at DockerCon 2014
- Docker + Mesosphere - August 2014
  - Docker
    - High portability
    - Lower barrier to entry -> possibly drive greater adoption? – if you can use a shell, you can grasp a Dockerfile
  - Mesosphere
    - Fast deployments, scheduling of tasks, scaling, management of containers, self-healing/fault tolerant, simplification of datacenter management

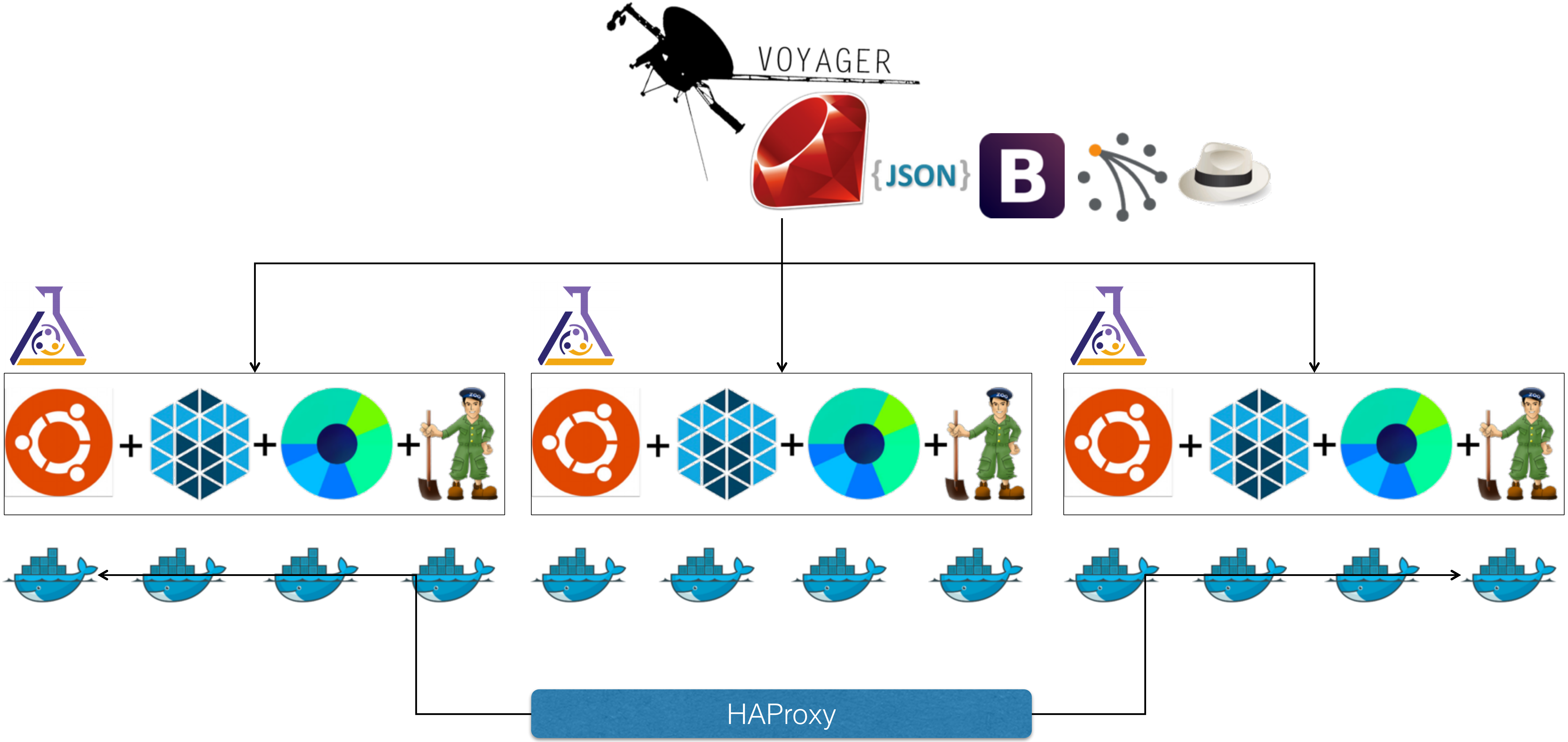
Our self-service gap

- Process gap between Docker and our Users
  - No shell access to Docker
  - Can’t deploy Docker from Marathon UI
  - Users not going to build API calls
  - Lets not just turn everyone loose on the full API
- How do we bridge that Gap?
  - Looked for tools in August 2014, found none that really met our needs

VOYAGER
Get Started
Find Out More

What's Voyager?

- Web application to bridge the gap
- Three person team developed initial MVP version in 2 weeks
- MVP features:
  - Automated Docker builds without shell access
  - Images built from GitHub repositories
  - Configurable RESTful service discovery/load balancing
  - UI and APIs to tie it all together
    - UI – ease of use for all
    - APIs – enable continuous integration


VOYAGER
{JSON}
B
+
HAProxy

VOYAGER
Thomas
Refresh List
New Application Group

My Application Groups

1	amibrokegroup	220039869
2	seconddeploygroup	220039869
1	somenewgroup	220039869

My Favorite Application Groups

1	app	212440068

All Application Groups

3	foobar	220027310
1	test	320002288

VOYAGER
Thomas

My Application Groups

Refresh List
+ New Application Group

1	amibrokegroup	220039869
2	seconddeploygroup	220039869
3 / 3	seconddeployapp	http://seconddeployapp.al.ge.com

Registry	voyager-registry-dev.al.ge.com/
GitHub repo	https://github.appl.ge.com/tas/docker-hello-world.git
Container Port	80
Load Balancer (LB)	cl3865.appl.ge.com
Advanced LB	false
Deploy version	1
Instances Running	
cl5540.appl.ge.com:31006	
cl5540.appl.ge.com:31009	
cl5462.appl.ge.com:31003	

VOYAGER
Thomas

New Application Group

Application Group Name

somedemogroup

somedemoapp

secondemoapplayer

Application Name

secondemoapplayer .al.ge.com

GitHub Repo

https://github.com/tas/docker-hello-world.git

Exposed port

80

# Instances

− 4 +

Advanced Options

Reset
Add to Group
Create Group
efresh List
New Application Group

My Application Groups

- 1 amibrokegroup
- 2 seconddeploygroup
- 1 somenewgroup

My Favorite Application

- 1 app

All Application Groups

- 3 foobar
- 1 test

VOYAGER
Thomas
efresh List
+ New Application Group

refresh somenewgroup

Point Voyager to your GitHub repository and we will refresh all applications in your group automatically!

somenewapp

GitHub Repository

https://github.ge.com/tas/docker-hello-world.git

Environment variables

ENV=dev

Exposed port

80

Back-end SSL

□

Refresh

My Application Groups

1	amibrokegroup
2	seconddeploygroup
1	somenewgroup
2 / 2	somenewapp

My Favorite Application

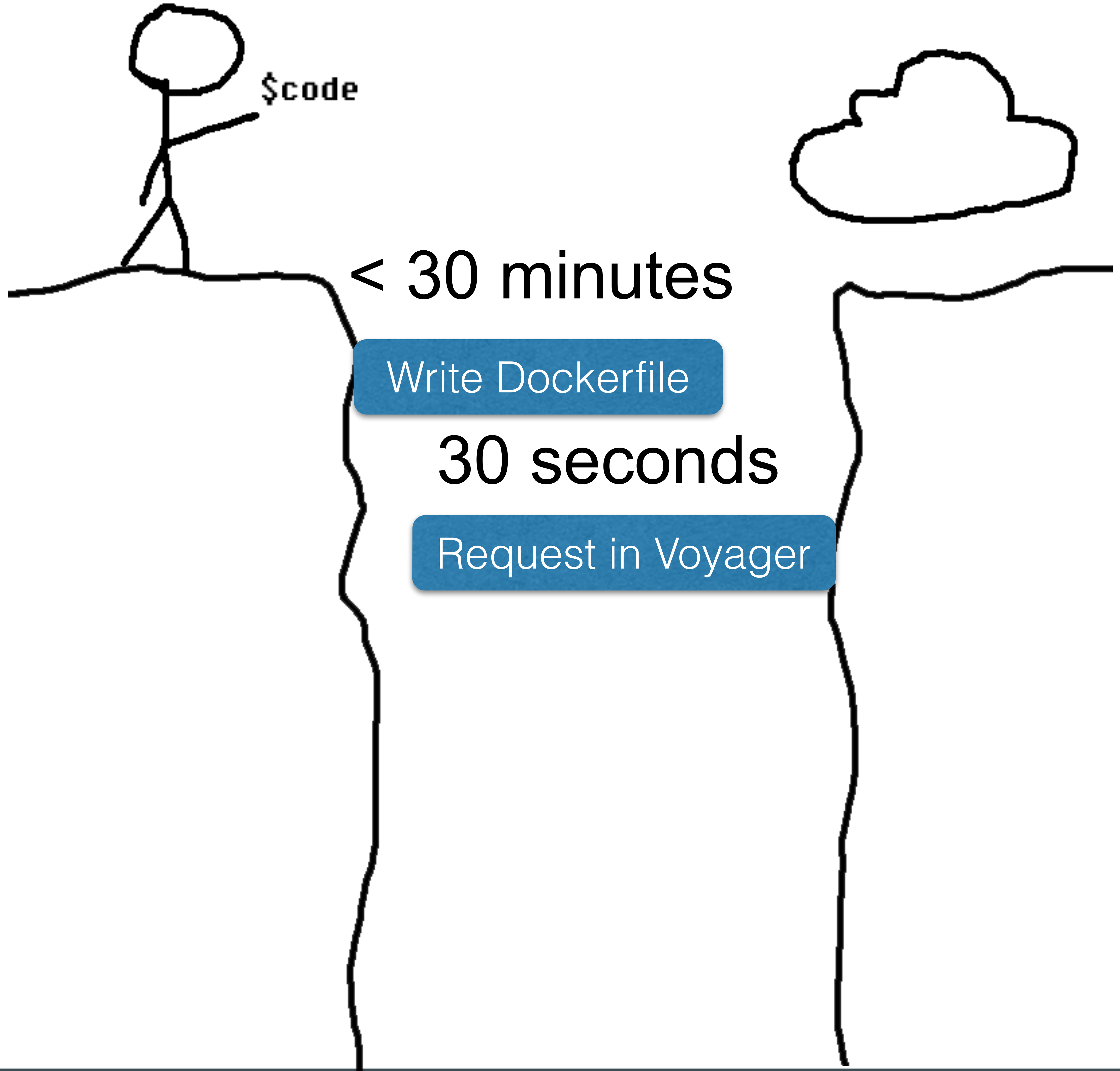
1	app

All Application Groups

3	foobar	220027310
1	test	320002288

dockercon 15

I can haz DevOps?



< 30 minutes

Write Dockerfile

30 seconds

Request in Voyager

- ☺ Repeatable
- ☺ Reproducible

I think our compliance lead just exploded…

- New concept of immutable infrastructure
  - ITIL/Compliance
  - How do I ssh? Which Server?
  - What do I do if my app instances are having problems?
- Education gap on Docker principals
- Highly heterogeneous environment
  - How do I support everything, but do it well?
- Security
  - Freedom to put whatever in container?

It’s all about that app…

- A focus shift from infrastructure to app
  - ITIL/Compliance
    - Policies and processes historically geared towards infrastructure
    - Must shift to the app in this model
- Provide Education
  - Help users differentiate images/containers, encourage environment variable configuration, best practices
- Adding features constantly, but carefully
  - Highly heterogeneous environment
    - Focus on being really good at running Docker containers
    - Balance accommodating everything with keeping it simple

Dude…where's my server?

- How do I ssh? Which Server?
  - No shell is an adjustment
  - External log shipping provides insight
  - Encourage new mindset with uneasy app owners
- What do I do if my app instances are having problems?
  - Break our old habits of server hugging
  - Just throw bad ones away and replace
  - Cheapening of app delivery enables this

Bro…it’s probably fine

- Security
  - Double edged sword - freedom vs. control
    - Encourage best practices
    - Education
    - Minimize container footprint/attack vectors
    - Shift/decentralize security responsibility to app teams
      - App teams take over patching

So does it work?

- Benefits
  - Instant scalability
  - 14x density
- User feedback on usability is positive
- Docker adoption rates are high
  - Currently running 350+ apps = 800+ containers, and growing daily
  - Our most critical applications are running in Docker
- Mesosphere + Docker = stable and outage resilient
- Flexibility
  - Supports legacy applications well
  - Enables modern practices - micro-services and continuous integration

1954 - GE Appliance Park - First non-government owned computer - Univac-1 Serial Number 8

Fast forward 61 years…

- Same data center, slightly different hardware
- Ongoing project to move apps to new private and public clouds
- Planned exit strategy -> multiple years
- With Docker in the fold?
  - Completed so far 45%+ in ~4 months

So where do we go next?

- Broad Open Issues
  - Windows support?
  - Persistent storage?
  - Networking (SDN containers)?
- Organizational Goals
  - Oracle ERP in Docker?
  - Drive density - 1000’s of containers per blade
  - Too many load balancers -> containerize -> improve service discovery
  - Honeybadger - Multi-cloud spanning/support
  - Pushing Docker’s portability to its logical conclusion

Thank you

Tom Barber
thomas.barber@ge.com
@barberta2

Brett Luckabaugh
brett.luckabaugh@ge.com
@OverflownStack