Managing Your Application Lifecycle on AWS

Continuous Integration and Deployment

Adrian White, Solutions Architect
Amazon Web Services

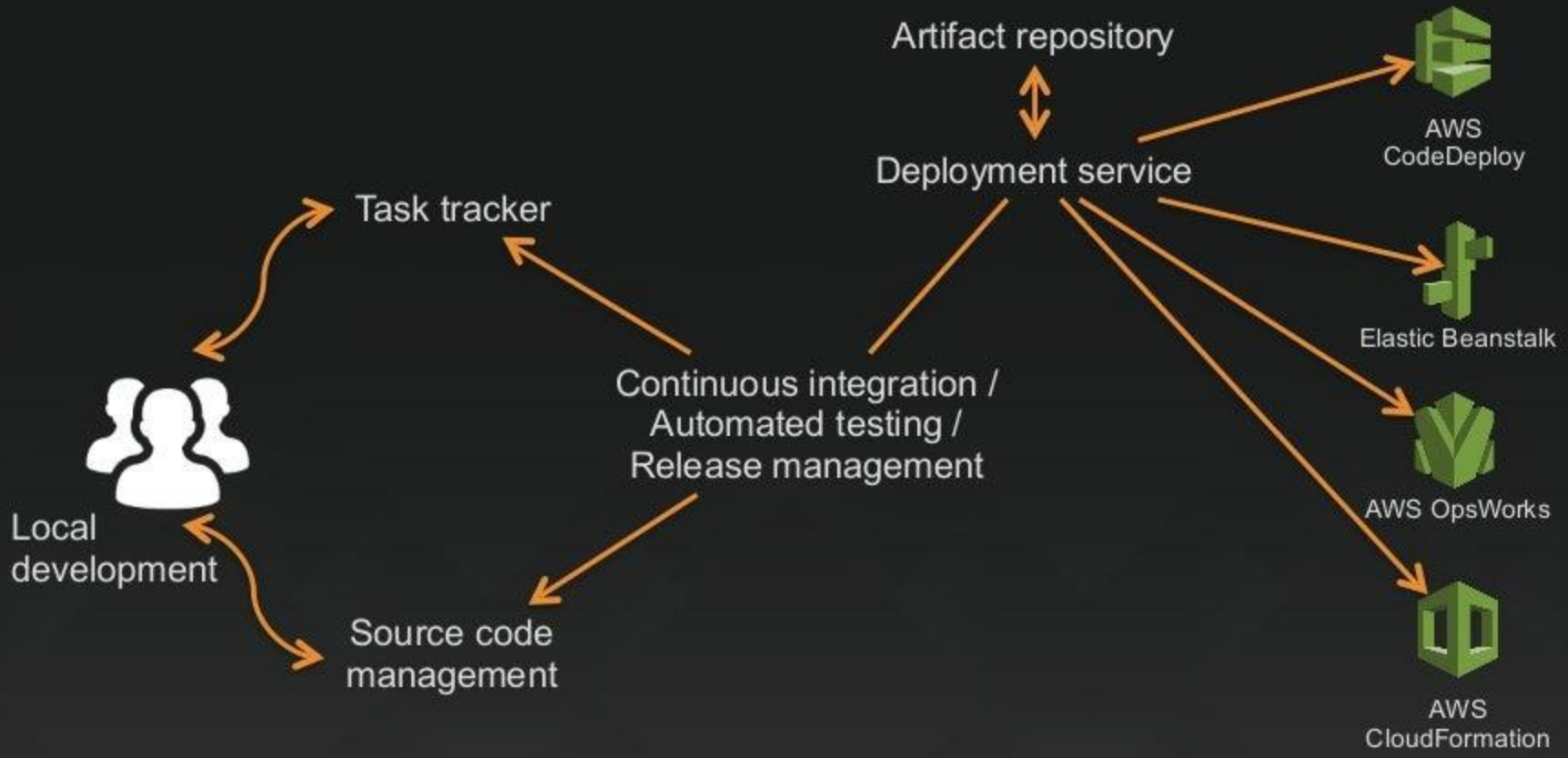
amazon web services

Session Grading

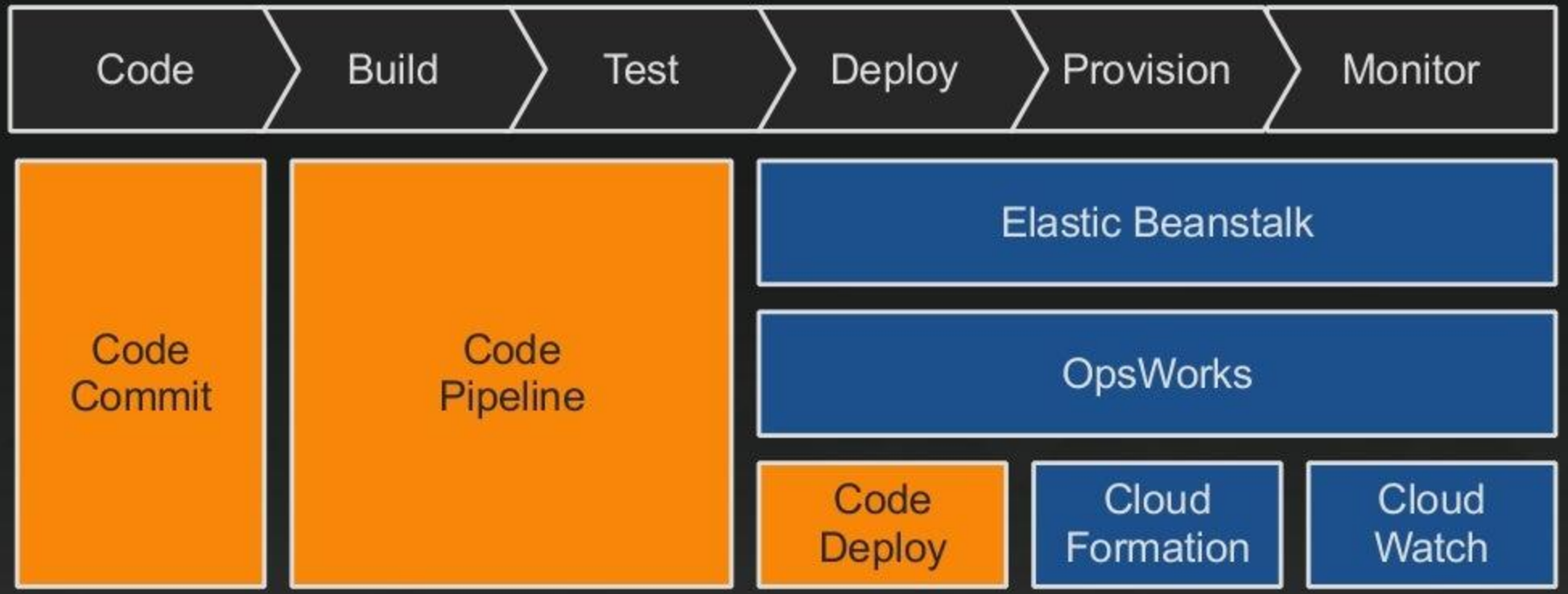
What are we covering today?

- Consistency through the development, test and release lifecycle
- Improve quality over time
- Increase velocity of application change
- AWS deployment and management approaches
- What does deployment look like in the future?

Application lifecycle management workflow



AWS Code and Deployment Tools



Local development challenges

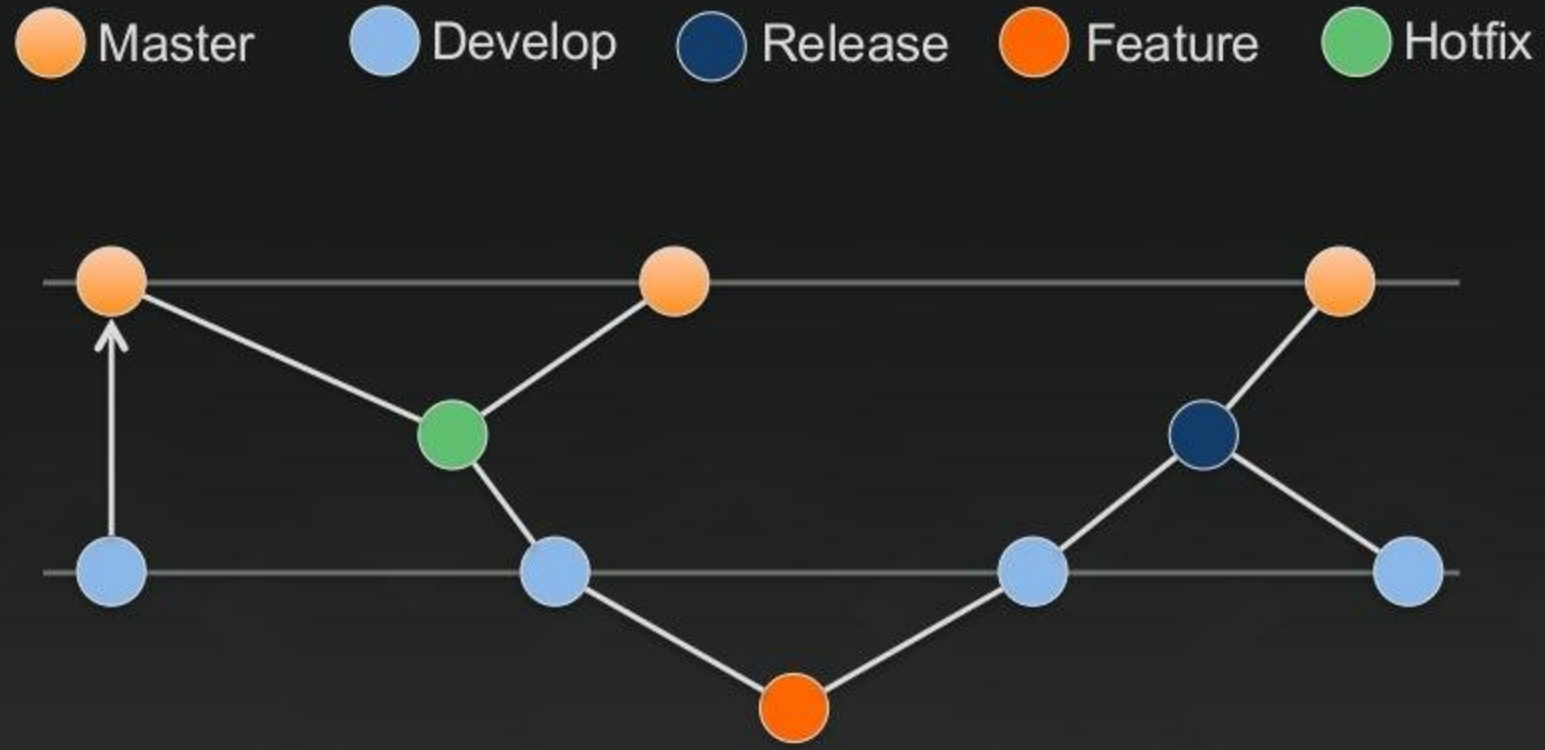
- Source code management design
- “But it works on my machine”
- Portable development environments
- Distributed teams work on tasks in parallel

Container image factories
(for consistency)

Source code management features

- Fast and easy branching
- Pull requests for distributed development workflows
- Code review
- Audit, logging, security

Feature branching, Gitflow and Pull requests



Inspired by https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow/

CodeCommit

- Private Git on AWS
- Massive scaled version controlled projects
- High service availability and resiliency
- Encrypted at rest
- Pay as you go pricing
- Import from SVN, Git, Microsoft TFS
- Use IAM to control access to repositories

CodeCommit workflow

create repository → Create repository

git clone ⇄ Receive clone request, Sync local / remote repos

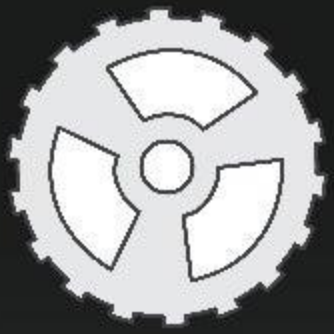
modify local files
git add / commit / push ⇄ Receive push request, Update remote repo

git pull ⇄ Receive push request, Update remote repo

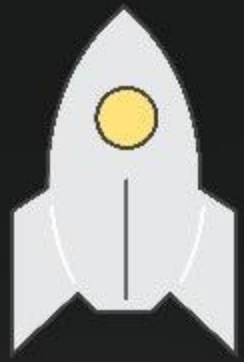
List repos, list branches → Receive requests

Display response ← Send responses

Why use a release automation service?



Automate workflow



Release quickly

Ensure quality

CodePipeline

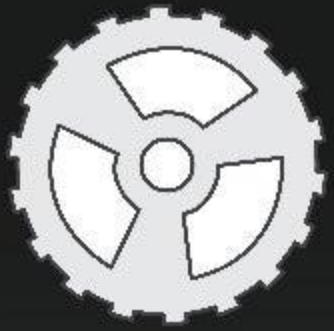
Continuous delivery and release automation, just like Amazon

- Customizable workflow engine
- Integrate with partner and custom systems
- Visual editor and status

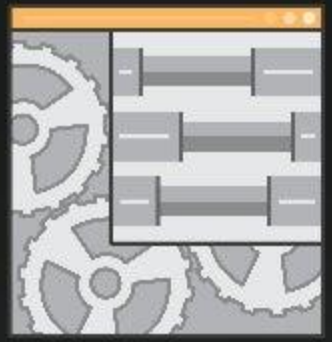
How do you ship application changes?

- Deployment approaches
  - In place vs discrete stacks
- Where is state in the system?
  - Stateless vs stateful application components
- Frequency and speed of change

Why use a deployment service?



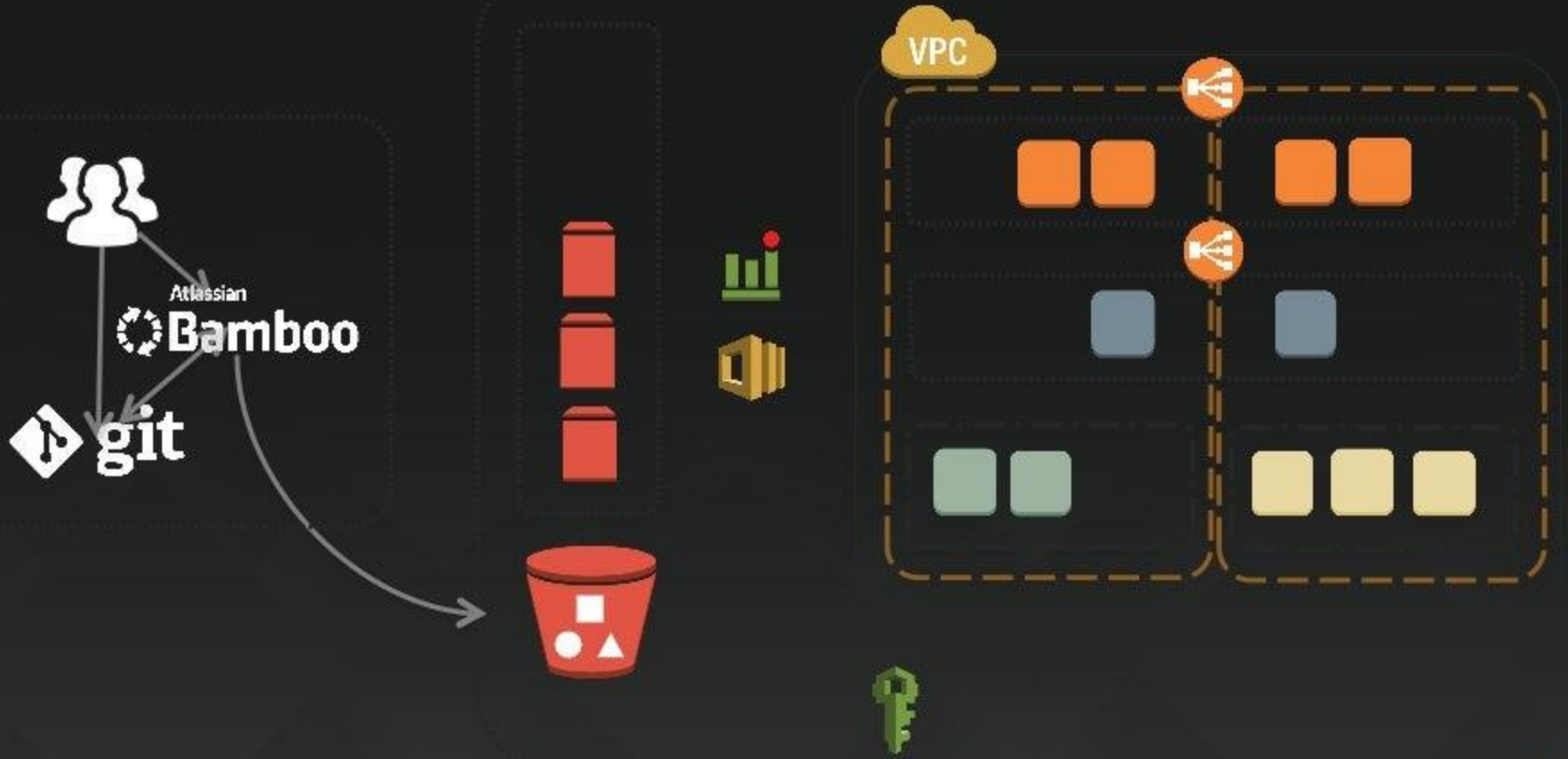
Automate deployments



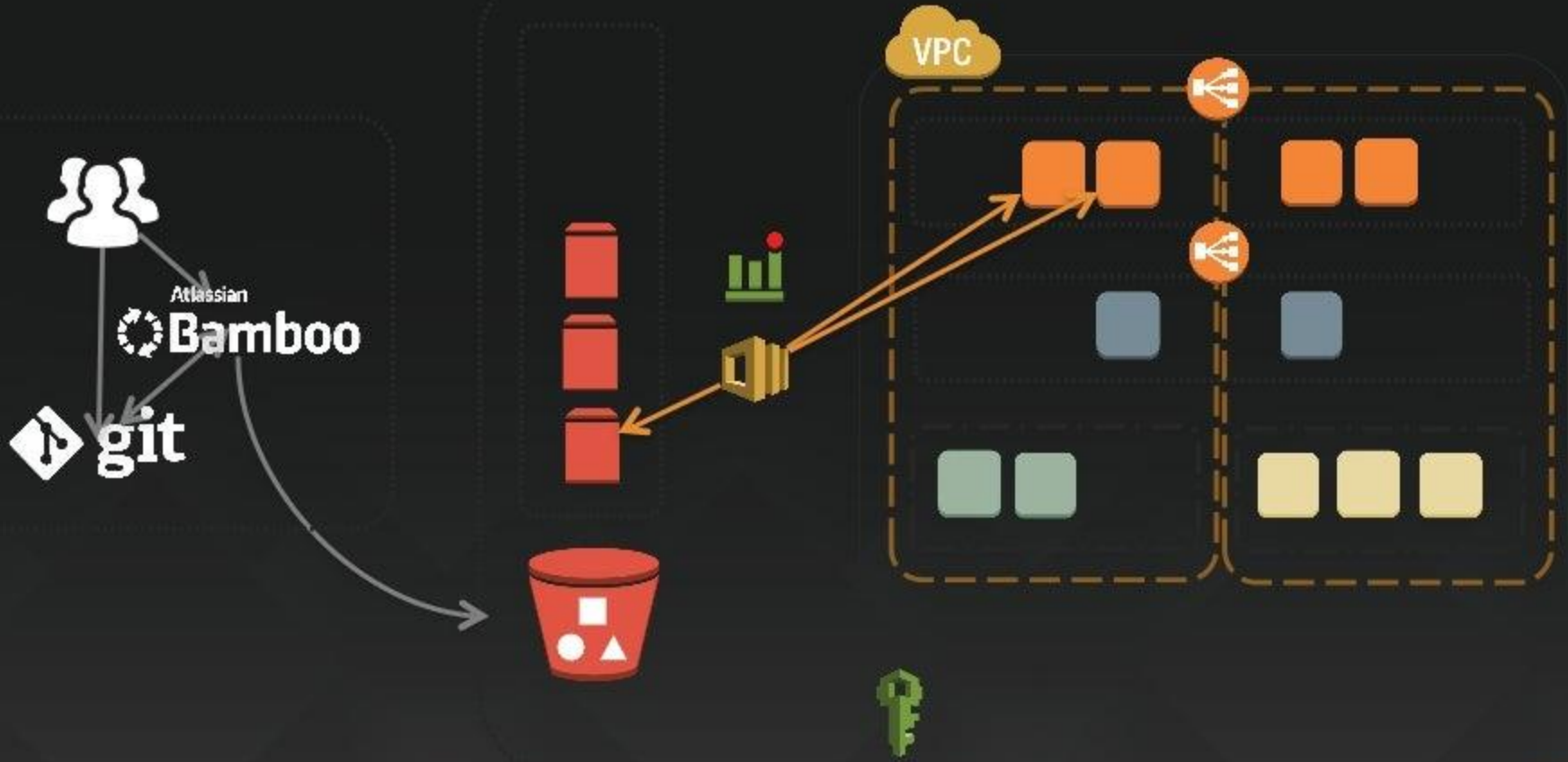
Manage complexity

Avoid downtime

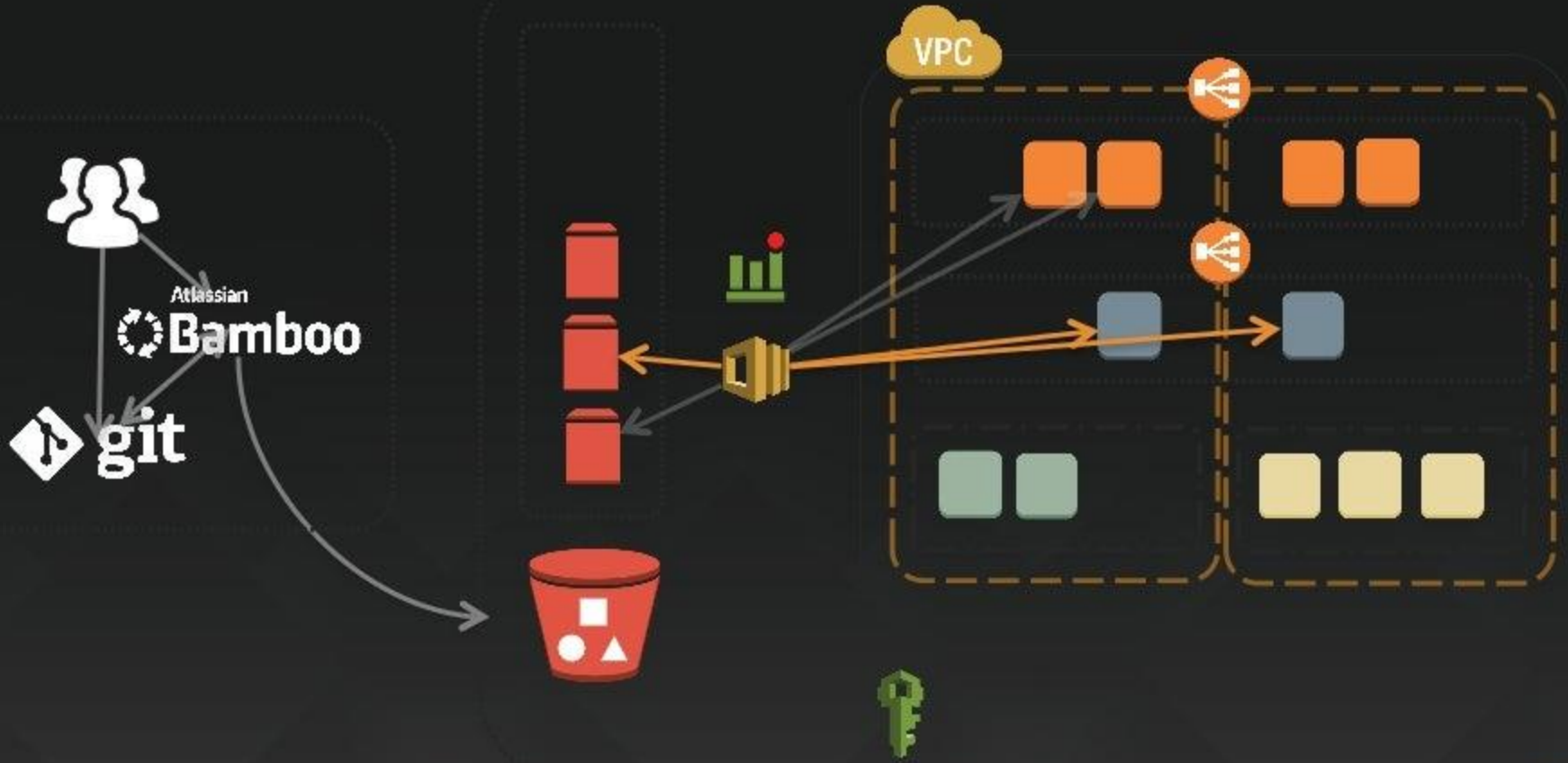
Shipping artifacts to existing environments



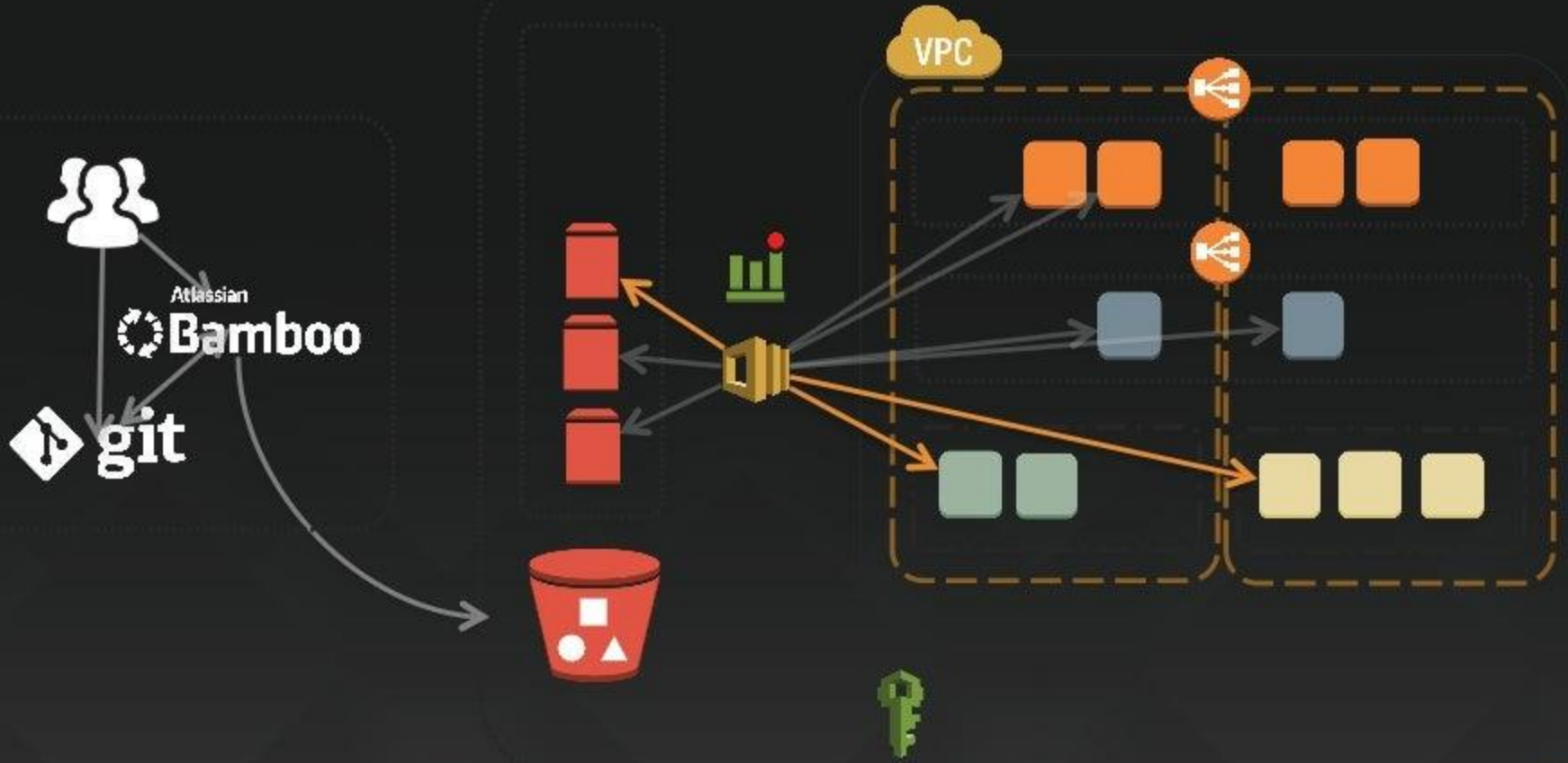
Shipping artifacts to existing environments



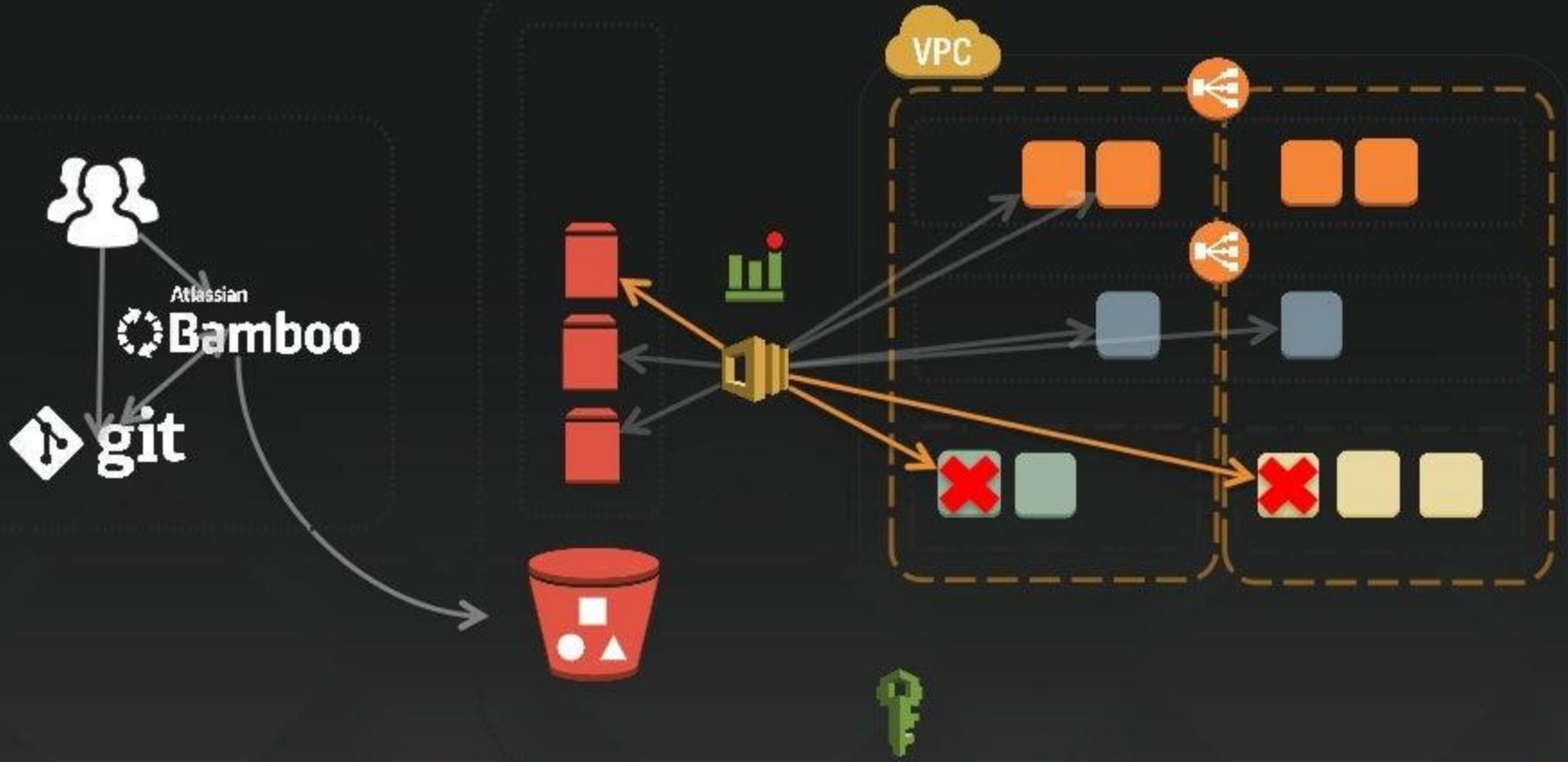
Shipping artifacts to existing environments



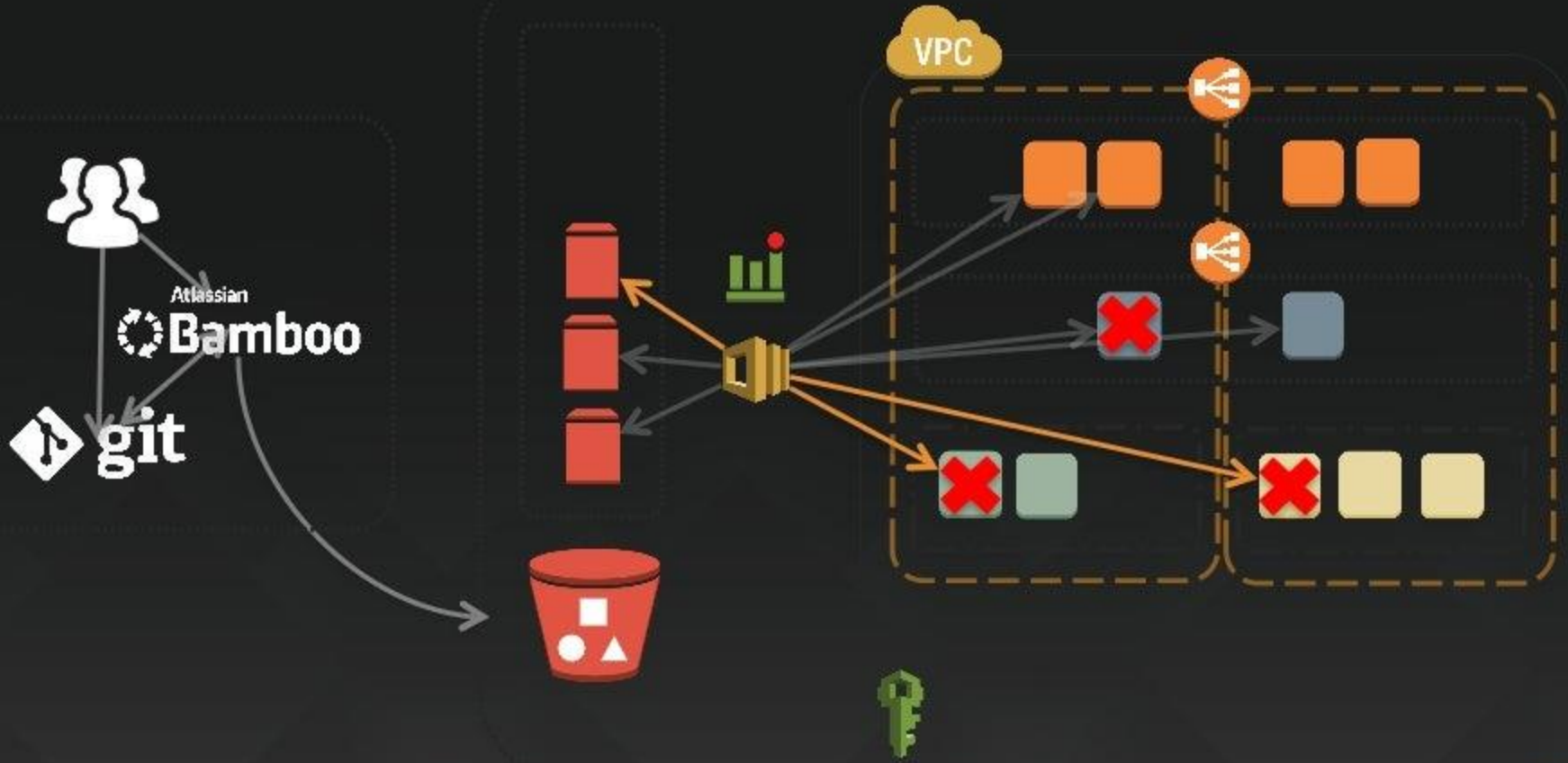
Shipping artifacts to existing environments



Shipping artifacts to existing environments



Shipping artifacts to existing environments



CodeDeploy helps with this!

CodeDeploy

Coordinate automated deployments, just like Amazon

- Scale from 1 instance to thousands
- Deploy without downtime
- Centralize deployment control and monitoring

amazon web services

Step 1: Package your application (with an AppSpec file)

```
version: 0.0
os: linux
files:
  - source: chef/
    destination: /etc/chef/codedeploy
  - source: target/hello.war
    destination: /var/lib/tomcat6/webapps
hooks:
  ApplicationStop:
    - location: deploy_hooks/stop-tomcat.sh
  BeforeInstall:
    - location: deploy_hooks/install-chef.sh
  AfterInstall:
    - location: deploy_hooks/librarian-install.sh
  ApplicationStart:
    - location: deploy_hooks/chef-solo.sh
  ValidateService:
    - location: deploy_hooks/verify_service.sh
```

Step 1: Package your application (with an AppSpec file)

```
version: 0.0
os: linux
files:
  - source: chef/
    destination: /etc/chef/codedeploy
  - source: target/hello.war
    destination: /var/lib/tomcat6/webapps
hooks:
  ApplicationStop:
    - location: deploy_hooks/stop-tomcat.sh
  BeforeInstall:
    - location: deploy_hooks/install-chef.sh
  AfterInstall:
    - location: deploy_hooks/librarian-install.sh
  ApplicationStart:
    - location: deploy_hooks/chef-solo.sh
  ValidateService:
    - location: deploy_hooks/verify_service.sh
```

Step 1: Package your application (with an AppSpec file)

```
version: 0.0
os: linux
files:
  - source: chef/
    destination: /etc/chef/codedeploy
  - source: target/hello.war
    destination: /var/lib/tomcat6/webapps
hooks:
  ApplicationStop:
    - location: deploy_hooks/stop-tomcat.sh
  BeforeInstall:
    - location: deploy_hooks/install-chef.sh
  AfterInstall:
    - location: deploy_hooks/librarian-install.sh
  ApplicationStart:
    - location: deploy_hooks/chef-solo.sh
  ValidateService:
    - location: deploy_hooks/verify_service.sh
```

Step 2: Set up your target environments

Group instances by:

- Auto Scaling group
- Amazon EC2 tag
- On-premises tag

Step 3: Deploy!

AWS CLI & SDKs
AWS Console
CI / CD Partners
GitHub

```
aws deploy create-deployment  \
--application-name MyApp  \
--deployment-group-name TargetGroup  \
--s3-location bucket=MyBucket,key=MyApp.zip
```

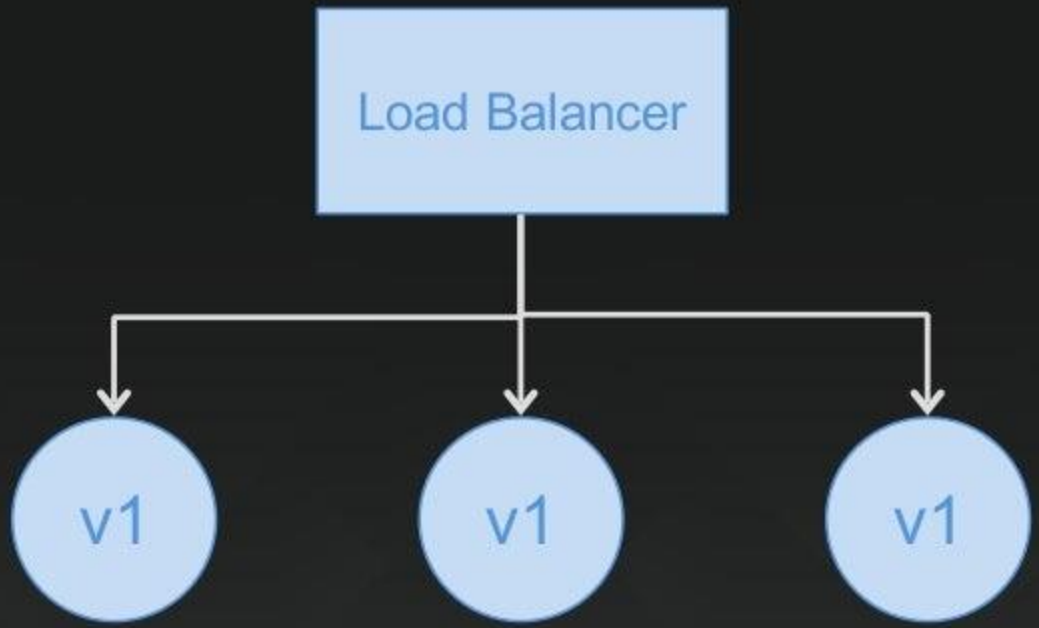
Deployment config – Choose speed

One-at-a-time

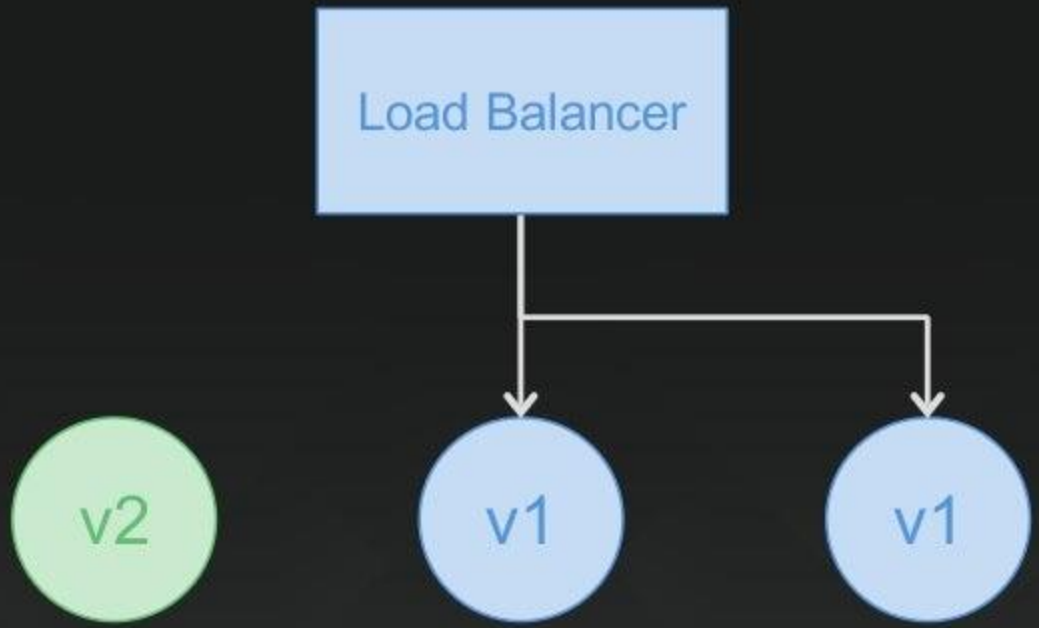
Half-at-a-time

All-at-once

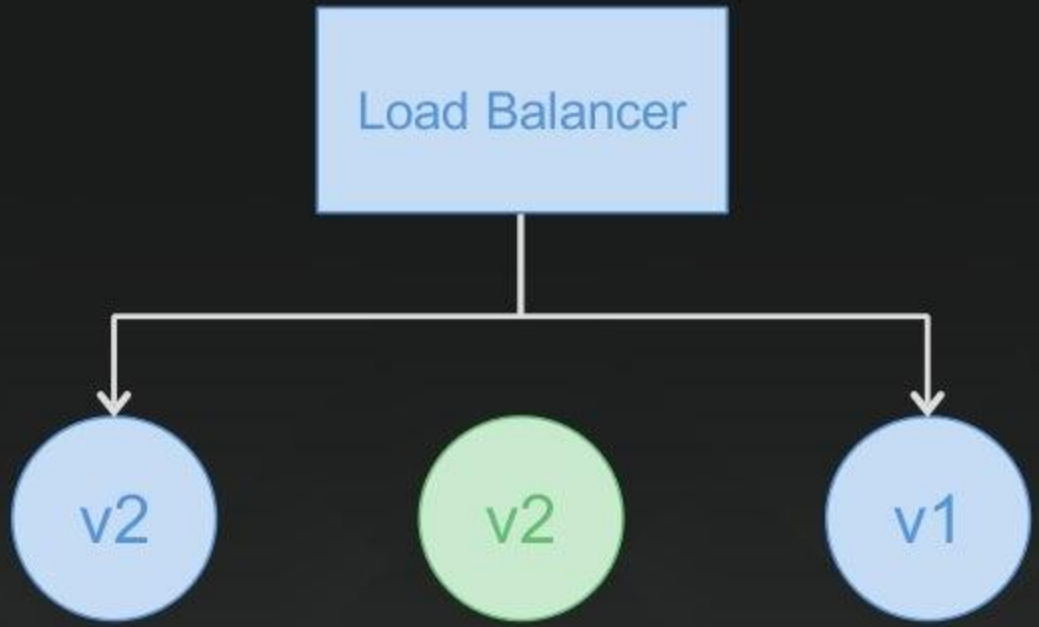
Rolling update – Deploy without downtime



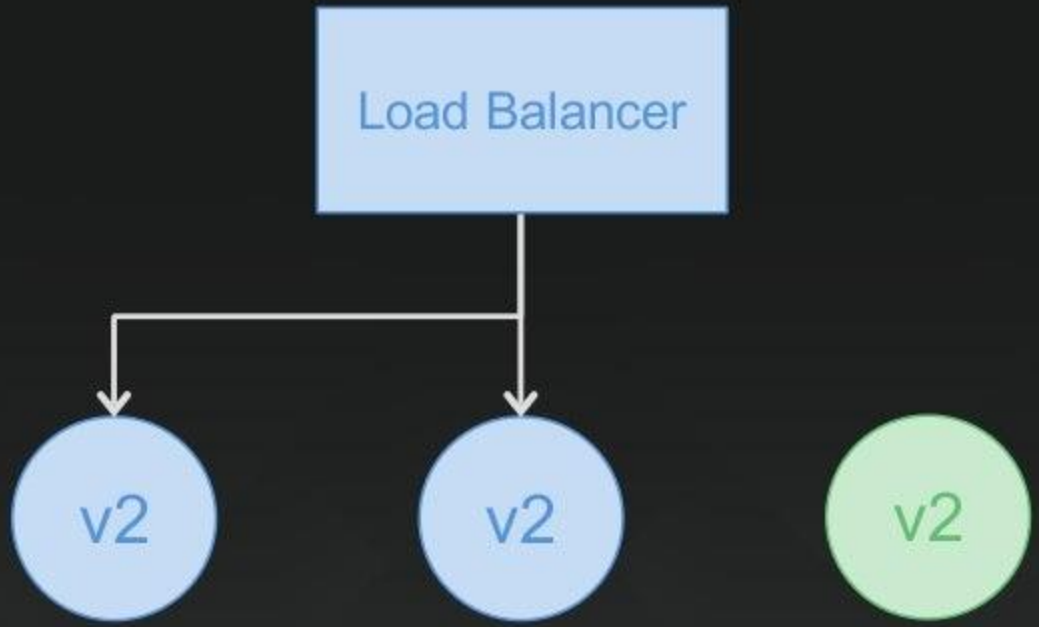
Rolling update – Deploy without downtime



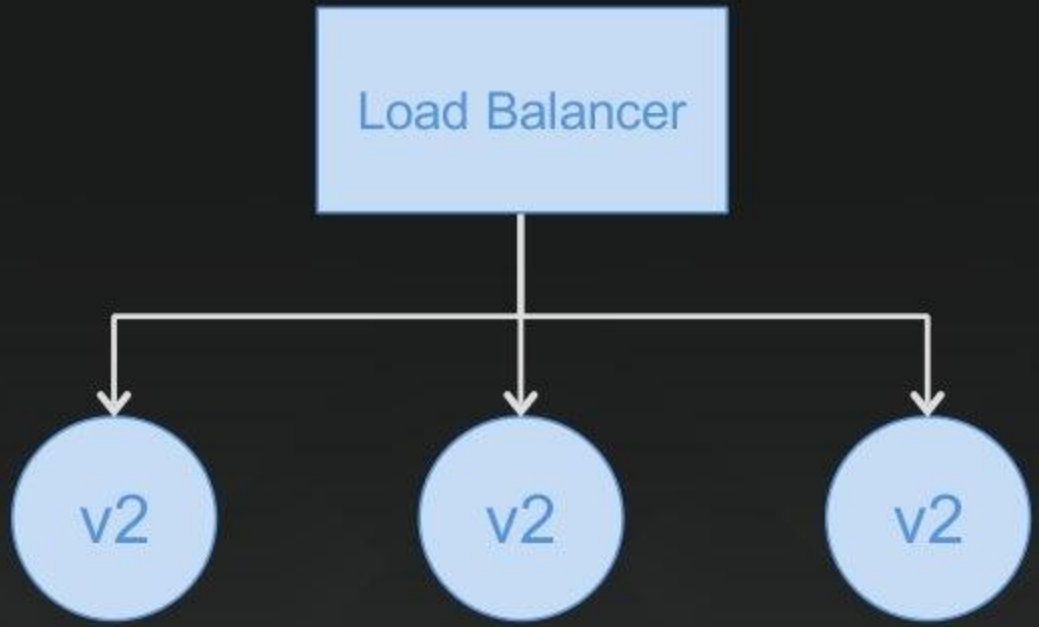
Rolling update – Deploy without downtime



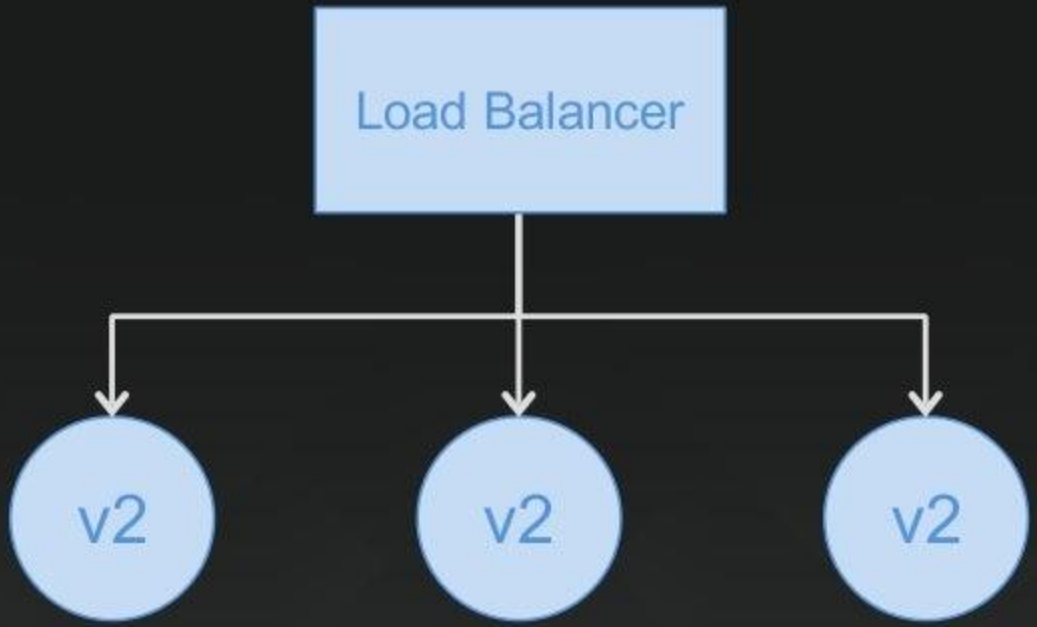
Rolling update – Deploy without downtime



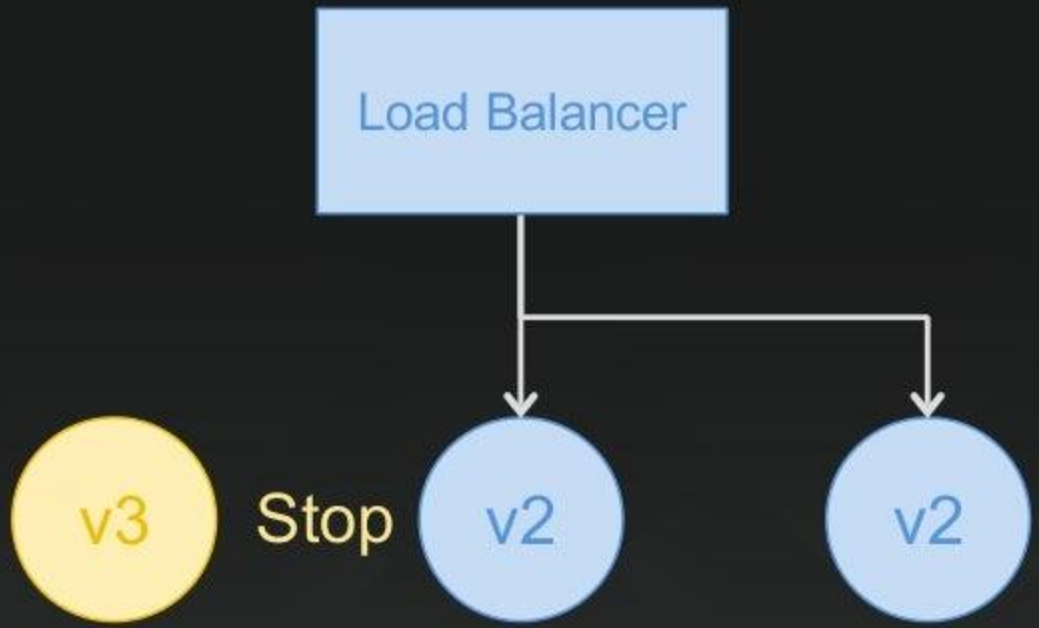
Rolling update – Deploy without downtime



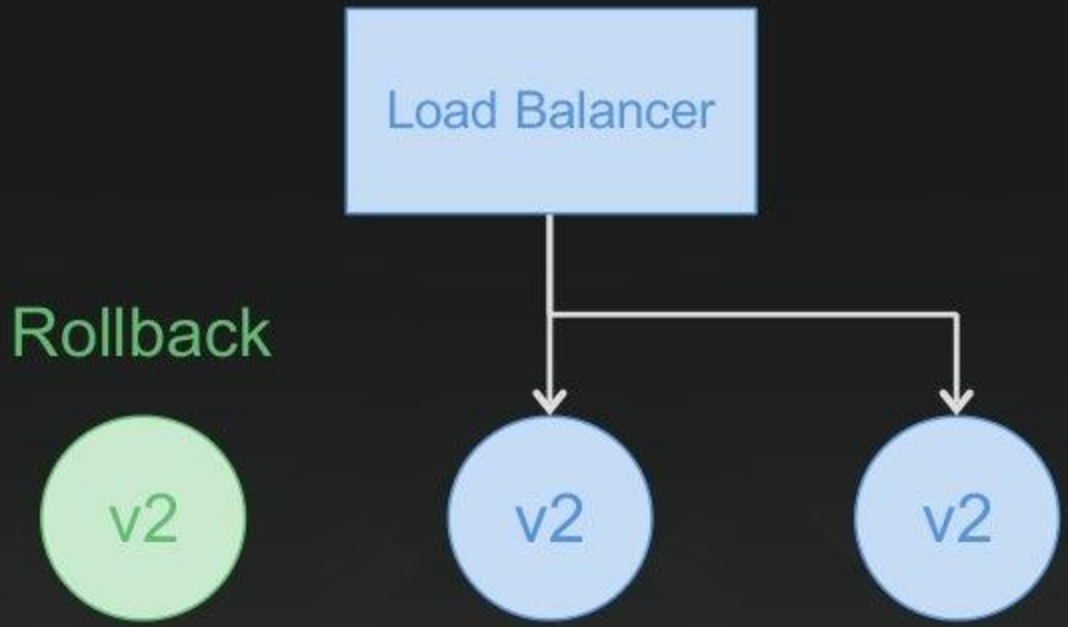
Health tracking – Catch deployment problems



Health tracking – Catch deployment problems



Health tracking – Catch deployment problems



Product integrations

Demo: CodeDeploy & Atlassian Bamboo

amazon web services

Shipping artifacts to new environments

- What if we can quickly and easily build new environments every time?

- CloudFormation
  - Deploying AMIs
  - Deploying containers

- CodeDeploy to manage discrete application versions

Shipping artifacts – Discrete environments

Shipping artifacts – Immutability via containers

Immutable infrastructure with Docker and
Amazon EC2 Container Service (ECS)

Shipping artifacts – Immutability via containers

ECS now supports ELB, health checks, scale-up and scale-down and update management

ECS Task Definitions

```
{
  "containerDefinitions": [
    {
      "name": "wordpress",
      "links": [
        "mysql"
      ],
      "image": "wordpress",
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80
        }
      ],
      "memory": 500,
      "cpu": 10
    },
```

```
    {
      "environment": [
        {
          "name": "MYSQL_ROOT_PASSWORD",
          "value": "password"
        }
      ],
      "name": "mysql",
      "image": "mysql",
      "cpu": 10,
      "memory": 500,
      "essential": true
    }
  ],
  "family": "hello_world"
}
```

Build Engineering @ Atlassian

Providing CI / CD as a Service

Peter Leschev, Senior Team Lead
Build Engineering

amazon web services

Introduction

- INTRODUCTION
- SCALING THE BUILD GRID
- AWS ADVANTAGES
- FUTURE STATE

Build platform & services used internally within Atlassian to build, test & deliver software

Developers expect a reliable infrastructure & fast CI feedback

Build Engineering today @ Atlassian

- 10 Bamboo Servers
- maven.atlassian.com / 6 Nexus instances
- Monitoring - opsview / graphite / statsd

- 1000 build agents (own hardware + EC2 instances)
  - Include SCM clients, JDKs, JVM build tools, databases, headless browser testing, python builds, NodeJS, installers & more
- Maintain 20 AMIs of various build configurations
- Nexus proxies

3 years ago:

21k

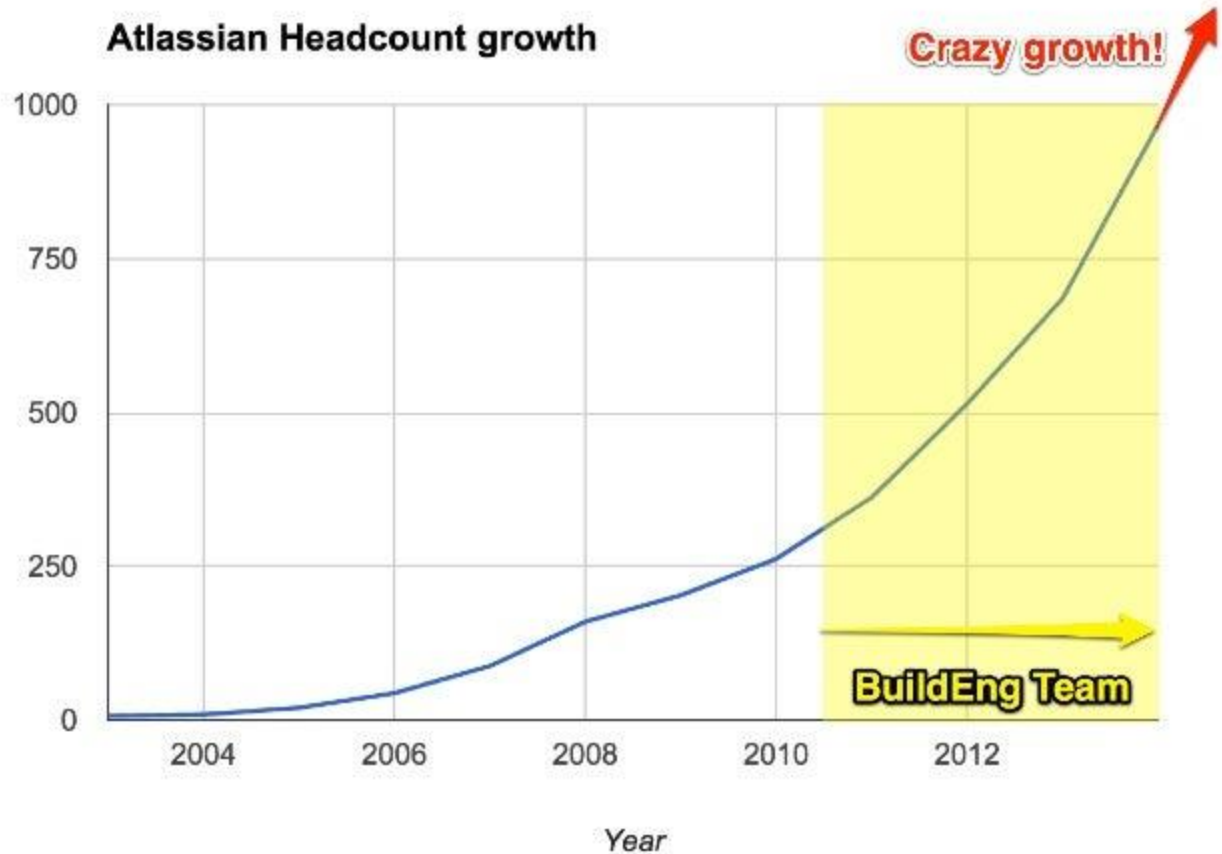
Builds per month

Last month:

127k

Builds per month

Build Engineering today @ Atlassian




Explosion of
Branch builds
in the last year
Branches
Plans
14.0 K
12.0 K
10.0 K
8.0 K
6.0 K
4.0 K
2.0 K
0
05/10
05/24
06/07
06/21
07/05
07/19
08/02
08/16
08/30
09/13
09/27
10/11
10/25
11/08
11/22
12/06
12/20
01/03
01/17
01/31
02/14
02/28
03/14
03/28

JIRA alone has

47k

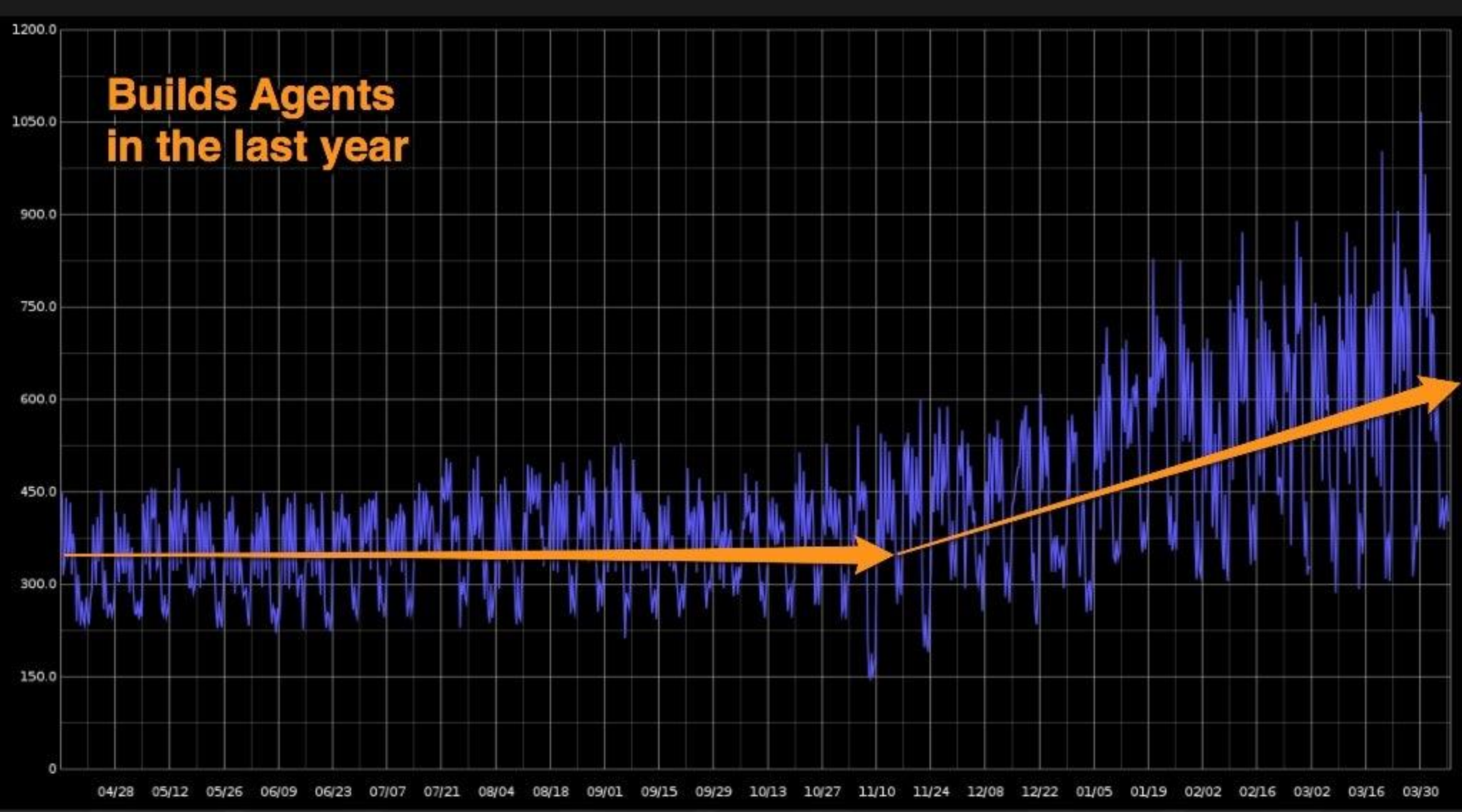
Automated tests

Scaling the Build Grid

- INTRODUCTION
- SCALING THE BUILD GRID
- AWS ADVANTAGES
- FUTURE STATE

Build Agents

Own Hardware using KVM guests + EC2 instances


Builds Agents
in the last year
1200.0
1050.0
900.0
750.0
600.0
450.0
300.0
150.0
0
04/28
05/12
05/26
06/09
06/23
07/07
07/21
08/04
08/18
09/01
09/15
09/29
10/13
10/27
11/10
11/24
12/08
12/22
01/05
01/19
02/02
02/16
03/02
03/16
03/30

Differences between KVM guests & elastic build agents

+

the need for Functional Parity

2 Problems

Problem 1: Network connectivity differences

Solution:
AWS Direct Connect + VPC enabled EC2 instances

Problem 2: Configuration differences between KVM guests & elastic build agents

Solution:
Removing tech debt using

https://puppetlabs.com/ https://www.packer.io/ https://cukes.info/ https://www.atlassian.com/software/bamboo

AWS Advantages

INTRODUCTION

SCALING THE BUILD GRID

AWS ADVANTAGES

FUTURE STATE

AWS Advantages

- Cost always decreasing, available instance types always expanding
- Able to handle the peaks / spikes in the build grid
- Difficult to predict future demand & provision hardware in time
- Usage characteristics change over time, existing hardware becomes sub-optimal
- Able to perform experiments / change instance types with ease
- Move faster as an organisation with an API & Credit Card
  - Much faster than seeking approval for hardware, lead time to installation & being available
- We've gone to the extent of 'Do Not Resuscitate' our existing hardware

https://aws.amazon.com/blogs/aws/category/price-reduction/

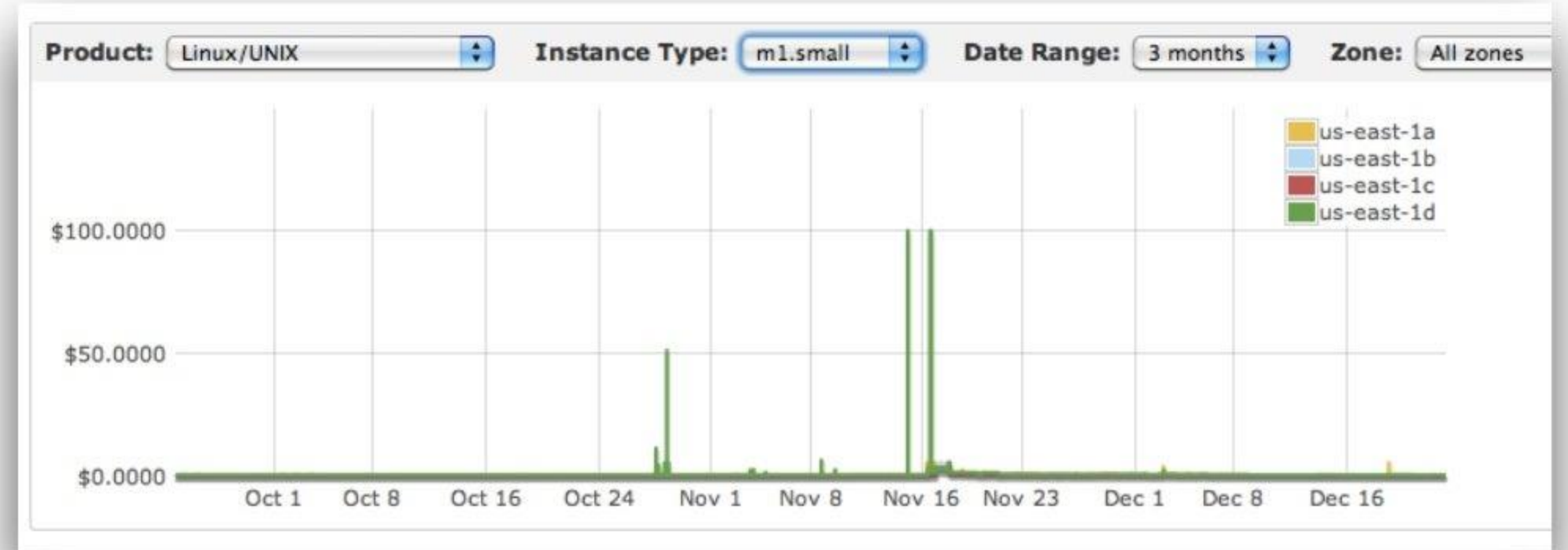
Spot instances

Cheap!

Slower startup times

Not always available

Price Volatility

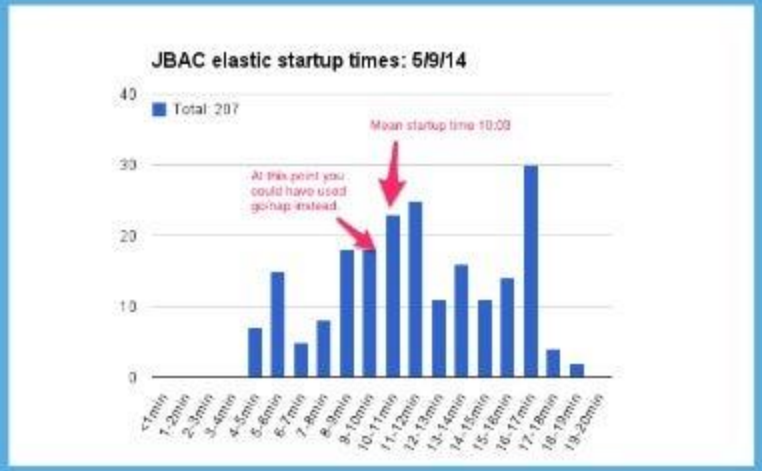


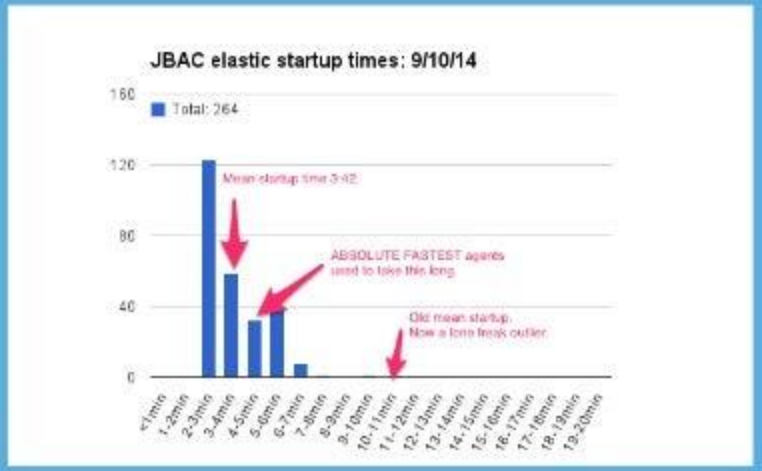
Expect Failure

Embrace It

Prebake at AMI burn time rather than at EC2 instance startup time

Faster startup time





More reliable
instance startup
(+ retry on failure!)

Pre-caching of git repositories, docker images

RequestLimitExceeded.
The maximum request rate permitted by the Amazon EC2 APIs has been exceeded for your account.

Handle failure, retry with backoff & recover

We currently do not have sufficient m3.large capacity in the Availability Zone you requested.

Multiple-AZ support in Bamboo 5.8

“we are experiencing increased error rates and latencies for the EC2 APIs”

Future State

- INTRODUCTION
- SCALING THE BUILD GRID
- AWS ADVANTAGES
- FUTURE STATE

Moving all Build Eng Infra to AWS

EC2 to ECS for build agents

Come help us!

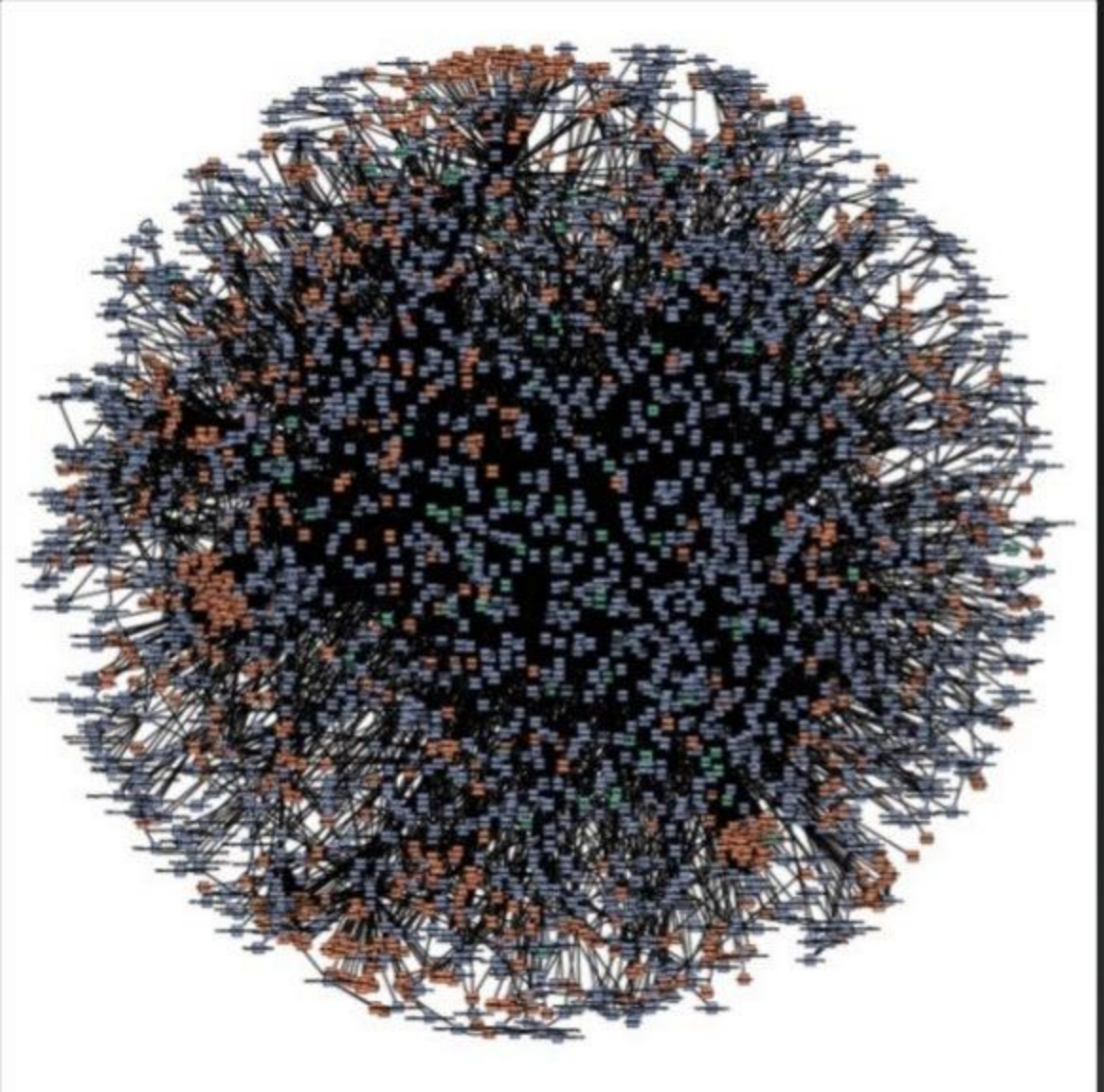
Atlassian

Thank you!

PETER LESCHEV · TEAM LEAD · ATLASSIAN · @PETER LESCHEV

Futures: what are you really deploying?

- Application bundles?
- AMIs?
- Docker containers?
- Lambda functions?
- Microservices architectures…



- Service-Oriented Architecture (SOA)
- Everything gets a service interface
- Primitives
- “Microservices”

Thousands of teams +

Microservices architecture +

Multiple environments +

Continuous delivery

Thousands of teams +

Microservices architecture +

Multiple environments +

Continuous delivery

= 50 million deployments a year

Where to go next…

- AWS Training & Certification
  - http://aws.amazon.com/training/
- Deployment and Management at AWS
  - http://aws.amazon.com/application-management/
- Code Management and Deployment
  - https://aws.amazon.com/blogs/aws/code-management-and-deployment/
- Amazon EC2 Container Service
  - http://aws.amazon.com/ecs/

Thank You

amazon web services