



DevOps and AWS

Shiva Narayanaswamy

Solution Architect



DEPLOYMENTS AT AMAZON.COM

~11.6s

Mean time between
deployments
(weekday)

~1,079

Max number of
deployments in a
single hour

~10,000

Mean number of hosts
simultaneously
receiving a
deployment

~30,000

Max number of hosts
simultaneously
receiving a
deployment



Business Value of Frequent Deployments

75%

90%

~0.001%

Reduction in outages triggered by software deployments since 2006

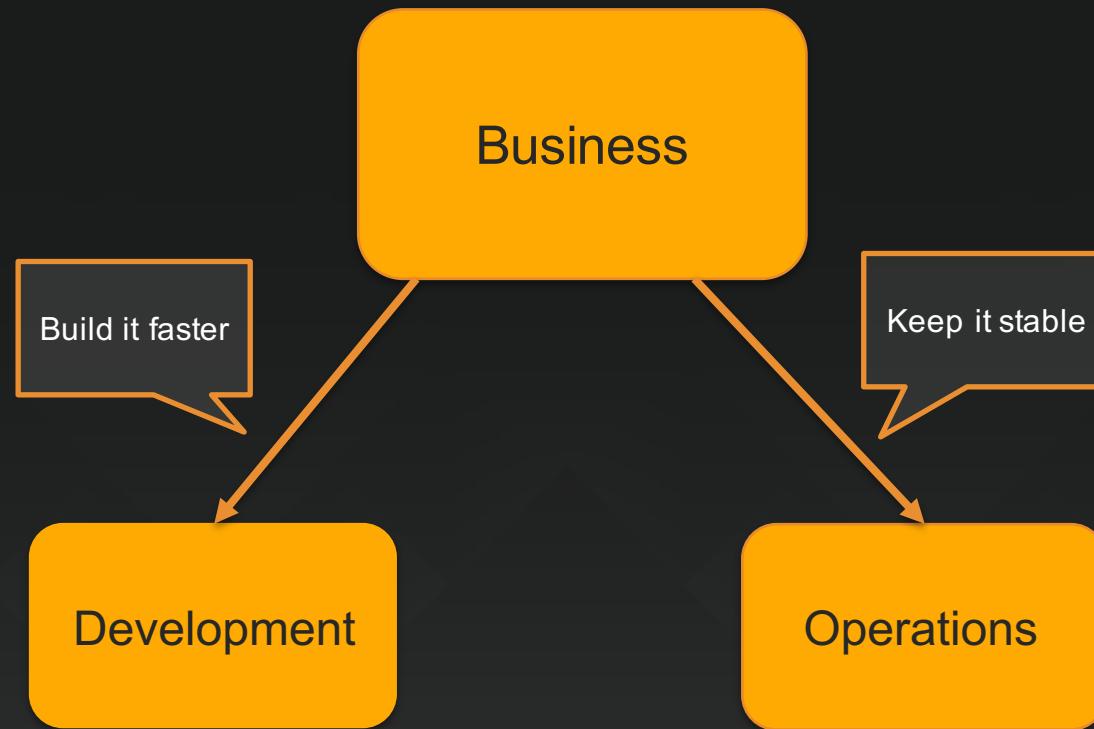
Reduction in outage minutes triggered by software deployments

Software deployments cause an outage

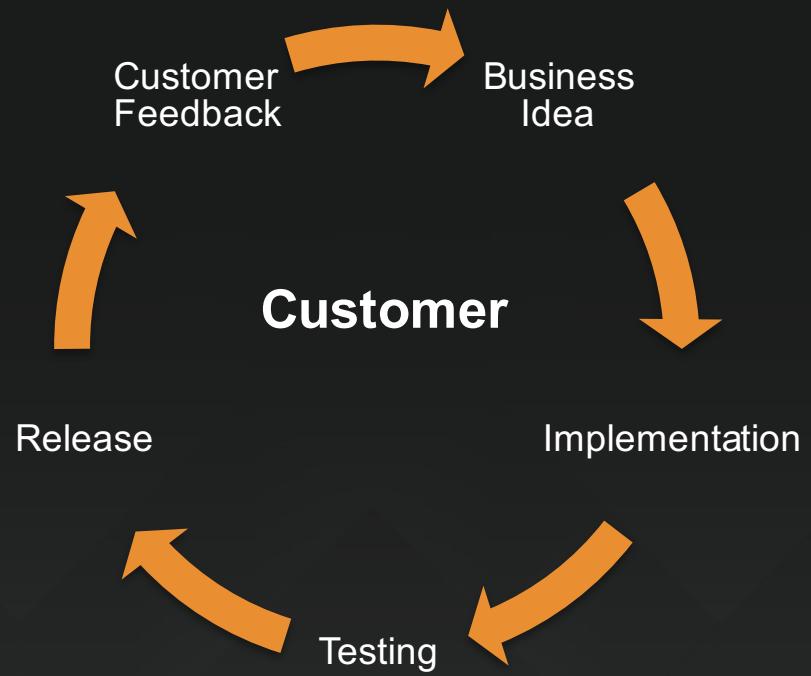


DevOps Culture

Innovation & Stability

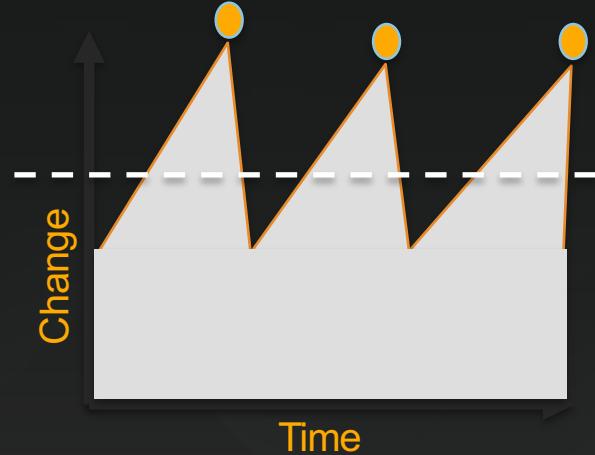


Continuous Innovation



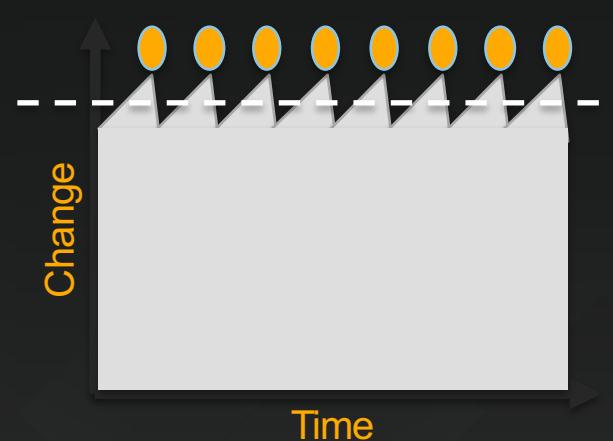
Deploying More Frequently Lowers Risk

Rare Release Events:
“Waterfall Methodology”



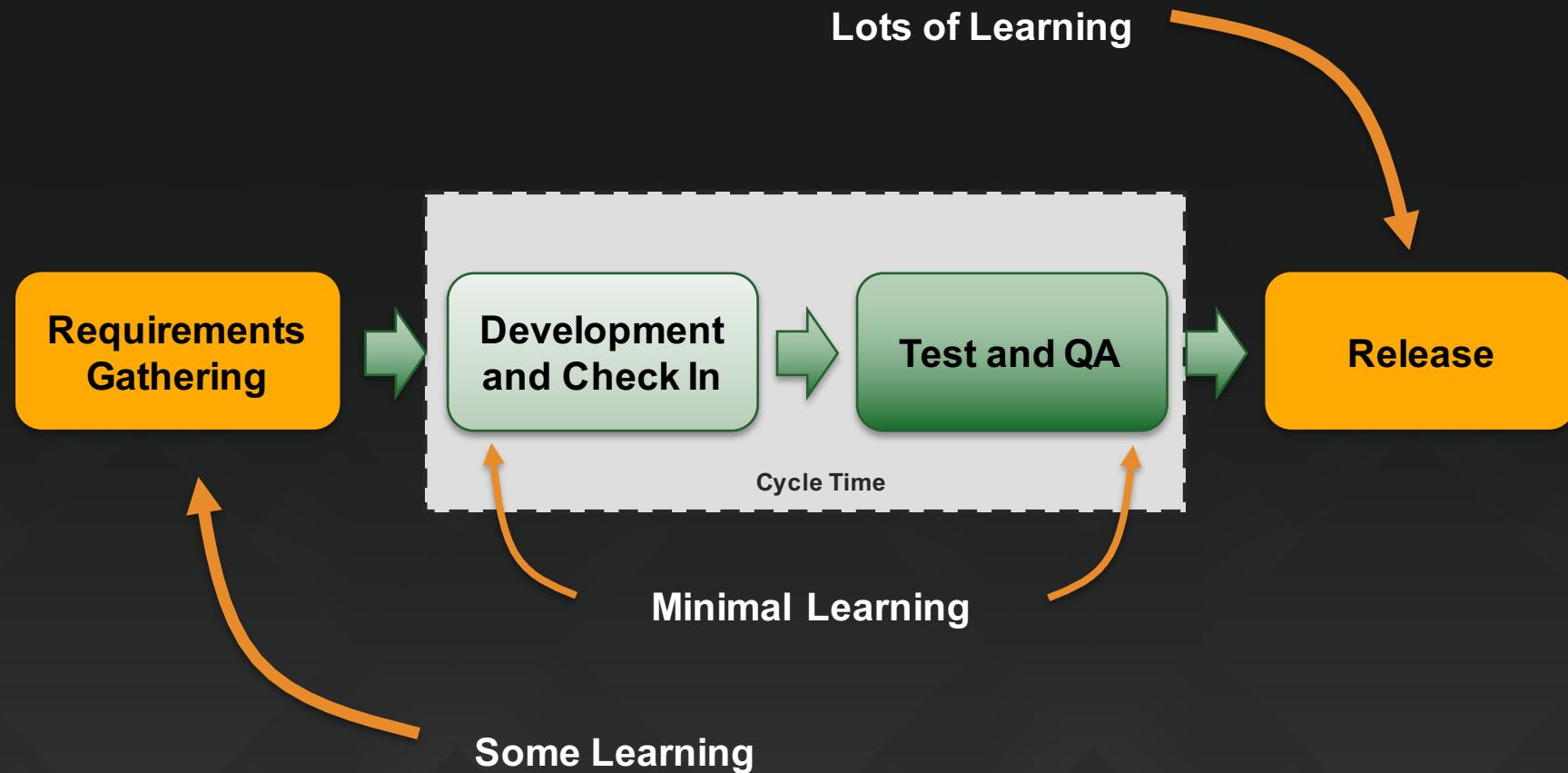
Larger Effort
“Increased Risk”

Frequent Release Events:
“Agile Methodology”



Smaller Effort
“Minimized Risk”

Knowledge is power



Change is inevitable

- Change is the **root cause** of most outages
- This leaves two approaches:
 - Make it a **big deal**
 - Small **iterative** non events

PAIN IS JUST WEAKNESS

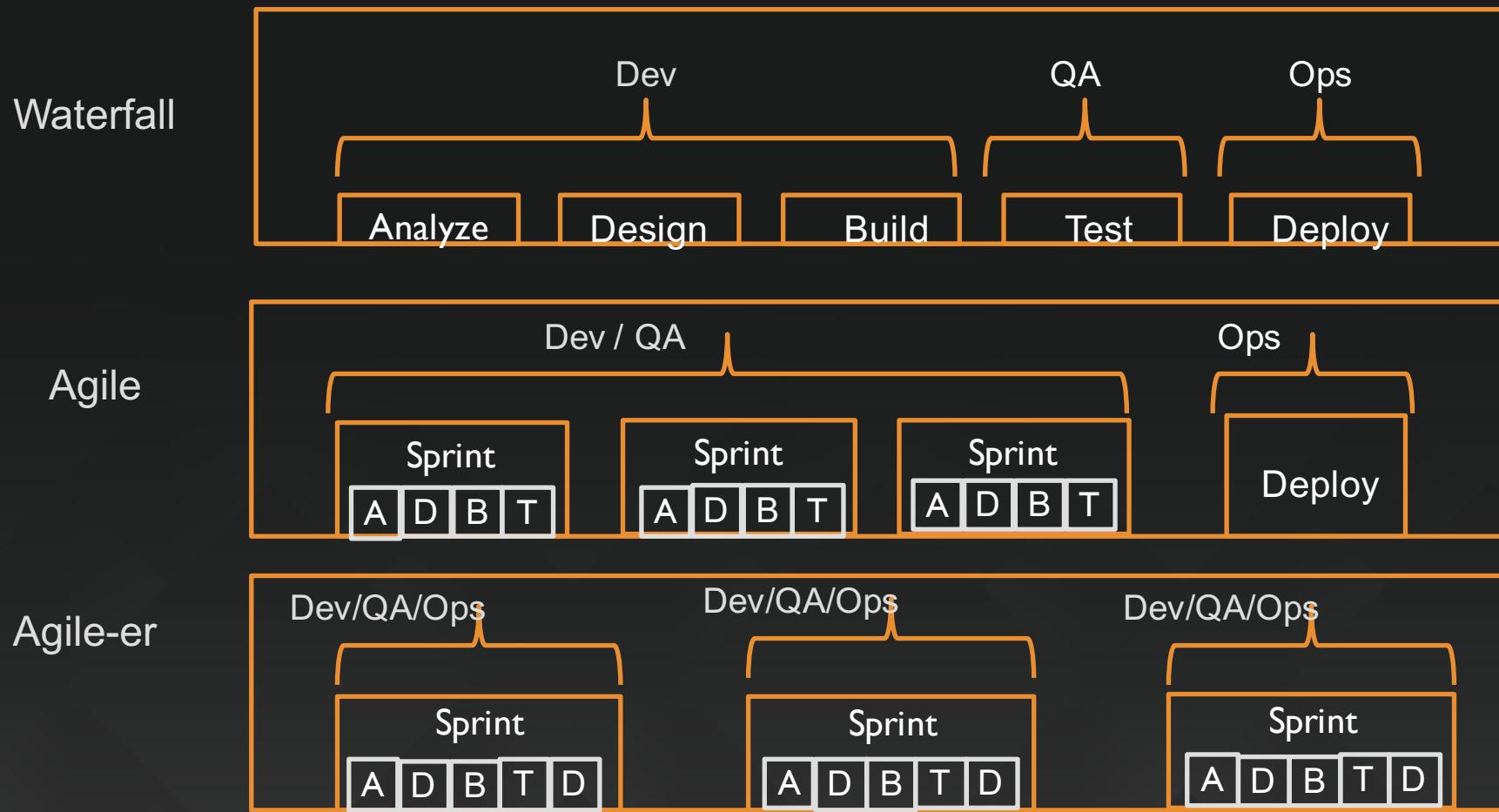
A photograph of a man with long brown hair and a beard, smiling warmly at the camera. He is wearing a dark t-shirt. In the background, the lower legs and feet of a woman in light blue jeans are visible, suggesting they are sitting together.

“Do painful things more frequently, so you can make it less painful...”

- Adrian Cockcroft, Architect, Netflix

LEAVING THE BODY

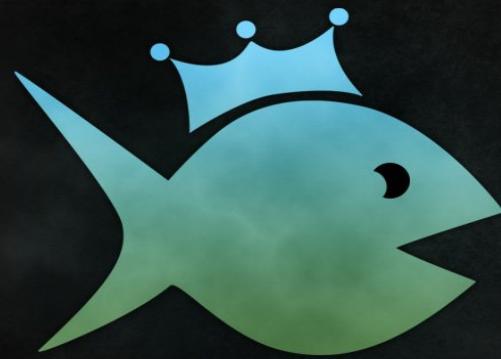
Agile-er



DevOps is a Re-Org!

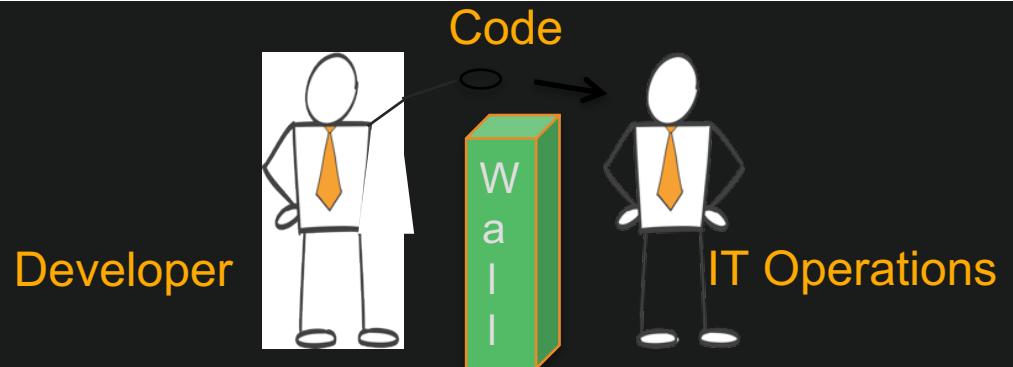


- Adrian Cockcroft, Architect, Netflix



REORGANIZE!

DevOps



- A philosophy? Cultural change? Paradigm shift ?
- Alignment of development and IT operations with better communication and collaboration ?
- Improvement in software deployment ?
- Breaking down the barriers between development and IT operations ?
- Akin to Agile software development applied to infrastructure and IT operations
- Set of tools and processes
- *It's all of the above!*

Business and IT agility



Business



Business
Agility

Agile Development

- Iterative development
- Scrum, sprints, stories
- Velocity



Developers
(application)



IT
Agility

DevOps

- Continuous Integration
- Continuous Deployment
- IT Automation
- Application Management



IT Operations



Building Teams

- *Encourage* a fail fast, learn quick mindset
- *Foster* innovation and accountability
- *Promote* open and honest sharing of lessons learned
- *Build trust* across organizational boundaries
 - Include all the stakeholders
- Expect that it will be a *work-in-progress*

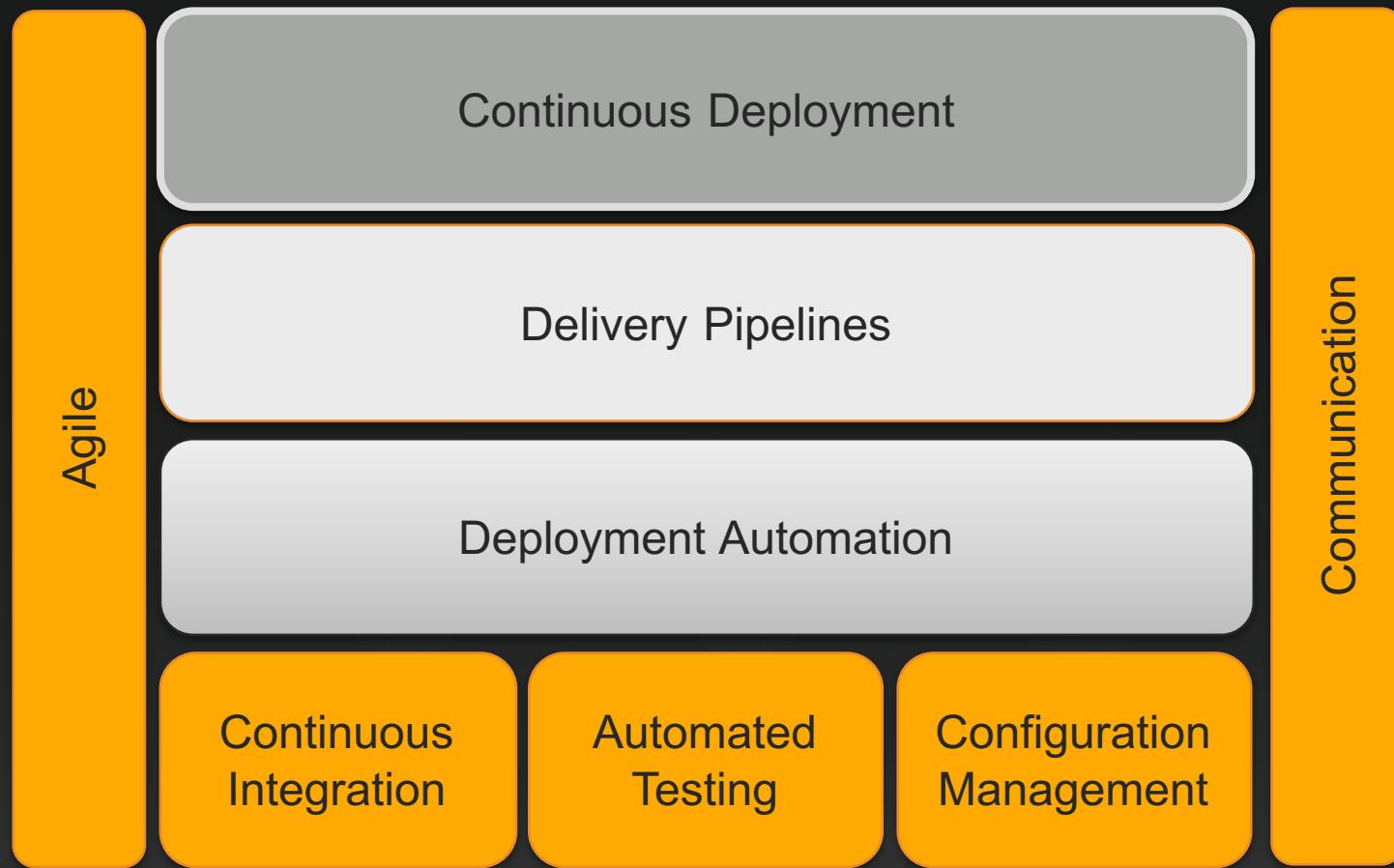
Amazon Leadership Principles

- 1. Customer Obsession
- 2. Ownership
- 3. Invent and Simplify
- 4. Are Right, A Lot
- 5. Hire and Develop the Best
- 6. Insist on the Highest Standards
- 7. Think Big
- 8. Bias for Action
- 9. Frugality
- 10. Learn and Be Curious
- 11. Earn Trust of Others
- 12. Dive Deep
- 13. Have Backbone; Disagree and Commit
- 14. Deliver Results



DevOps Technology

The DevOps Stack

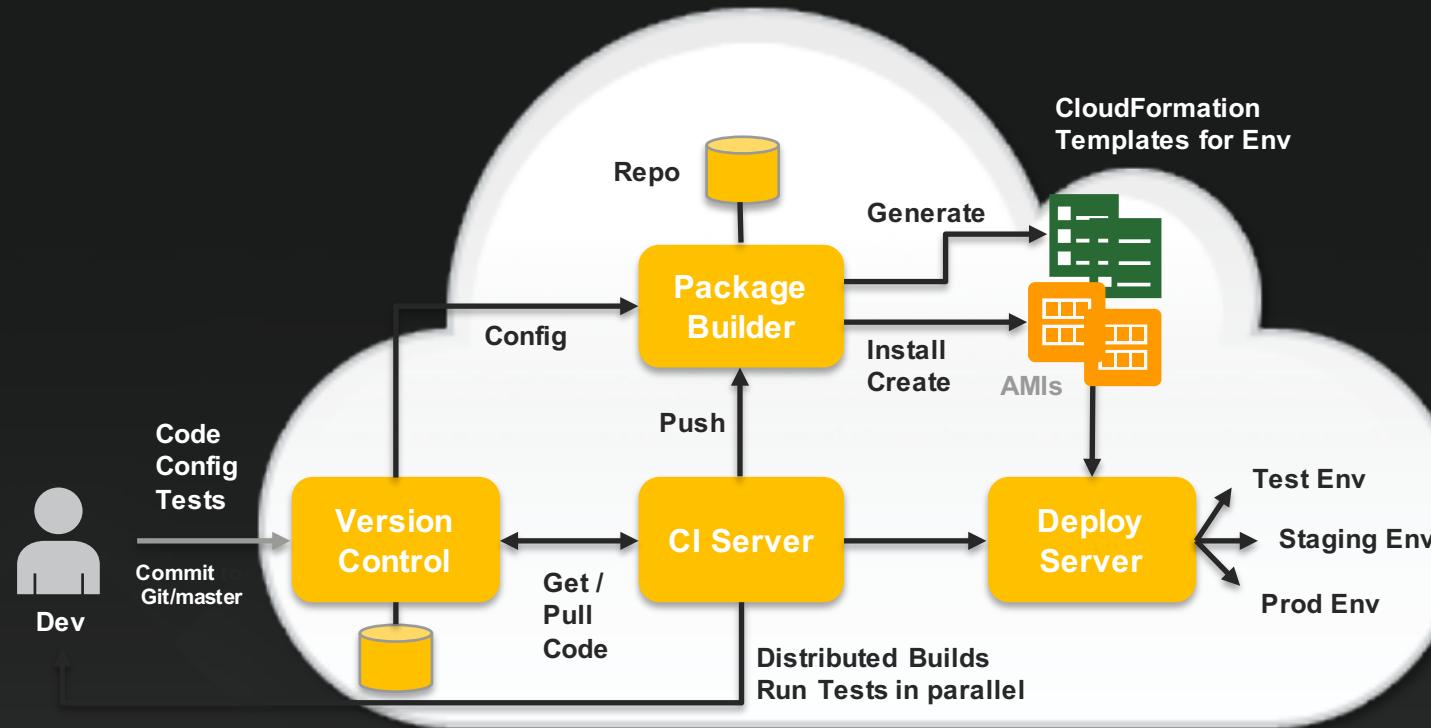


DevOps Practices

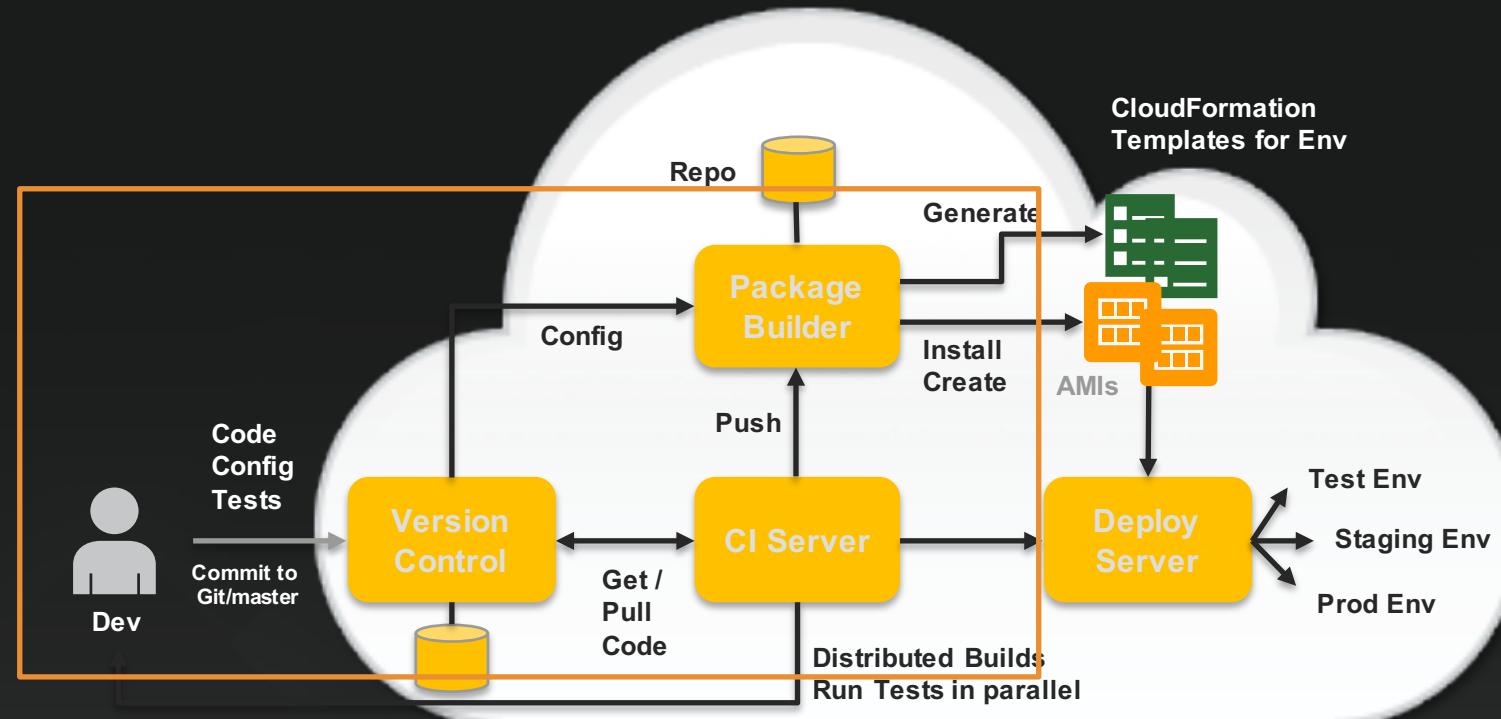
- Infrastructure as code
- Application and Infrastructure version management
- Continuous Integration
- Test Automation
- Continuous Deployment
- Monitoring and logging



Sample Application Cycle



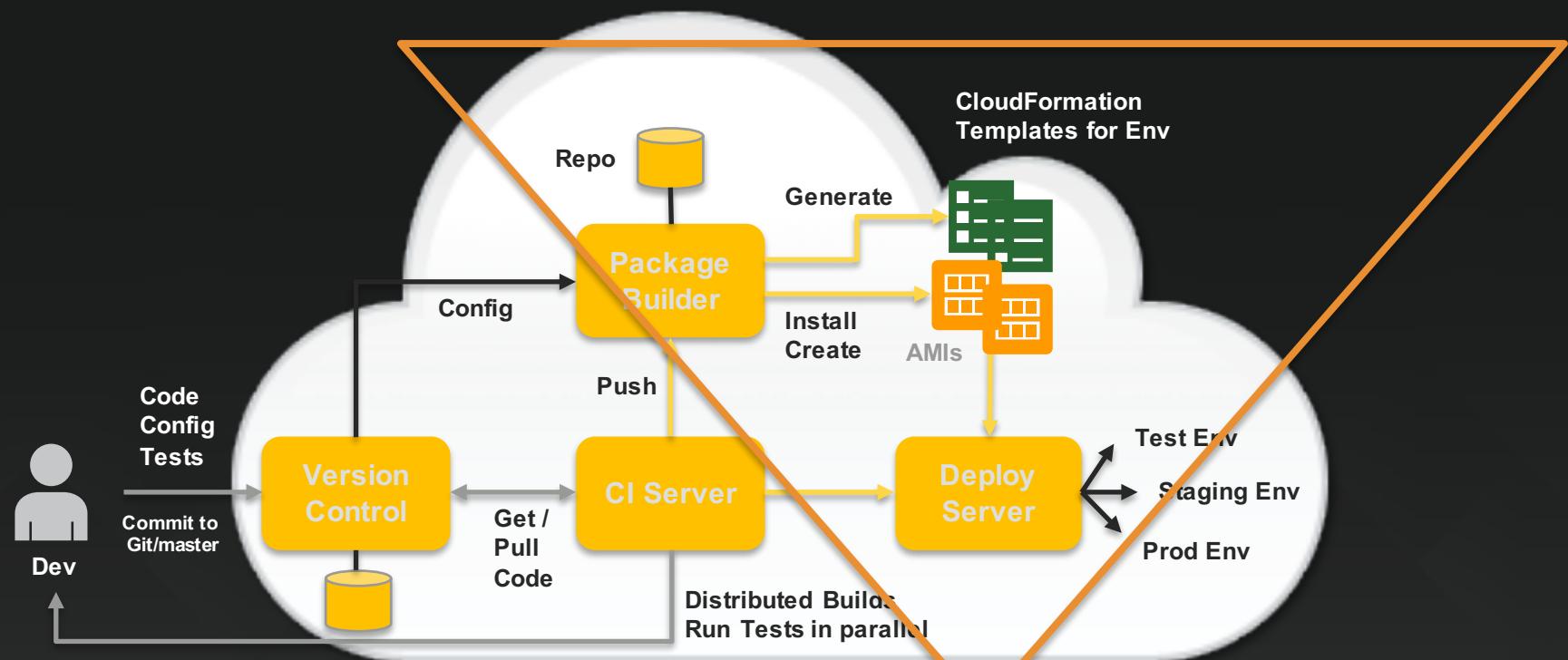
Continuous Integration



Continuous Integration

- Every developer check-in initiates a build
 - Builds are **fully** automated
 - Tests **validate** every check-in
- Instant feedback for developers
 - The build **must** remain fast
- Minimizes the impact of broken builds
 - Bugs **don't** accumulate
- Drives the culture of small, frequent releases

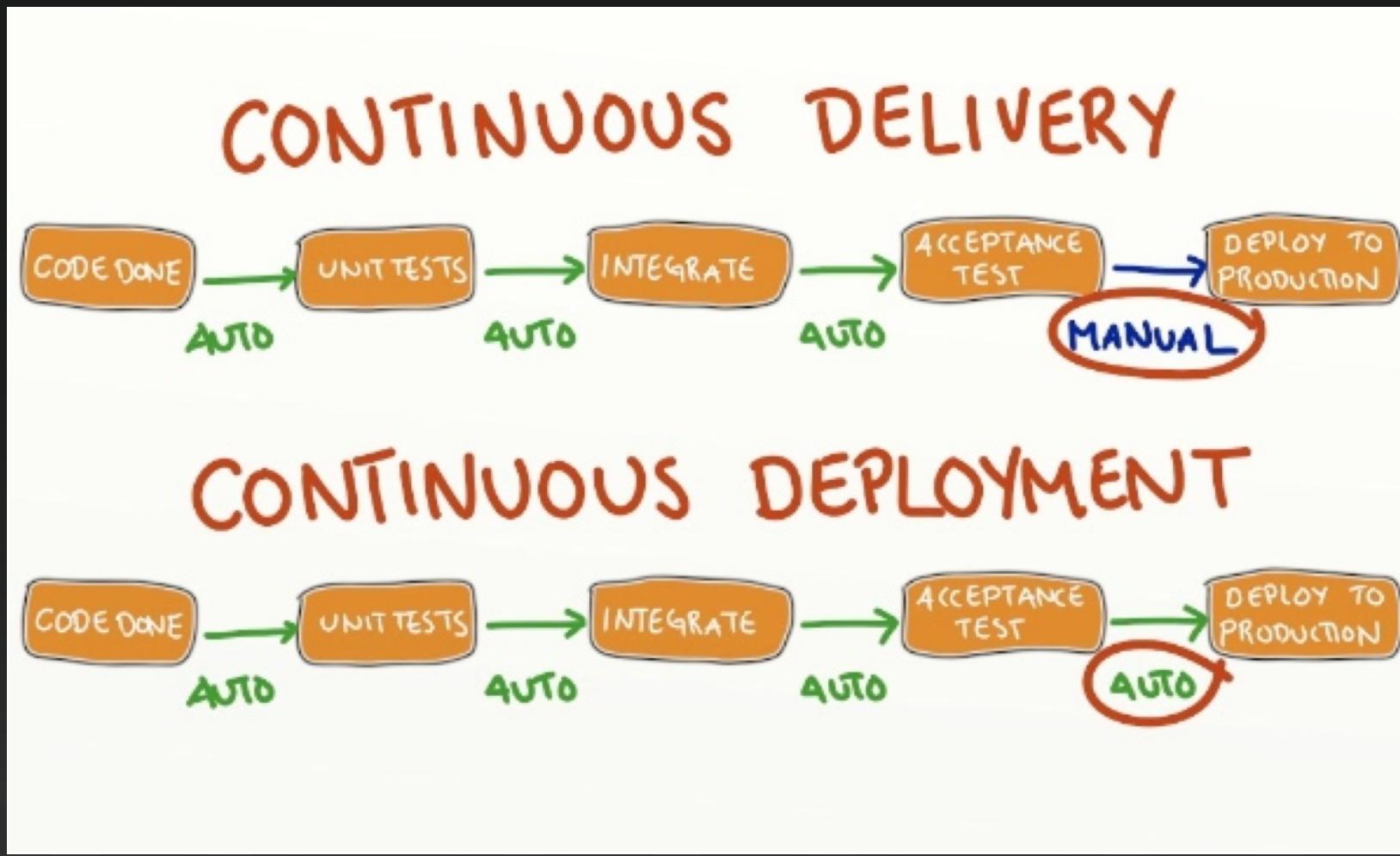
Continuous Delivery/Deployment



Send Build Report to Dev
Stop everything if build failed



Continuous Deployment vs Delivery



Continuous Delivery

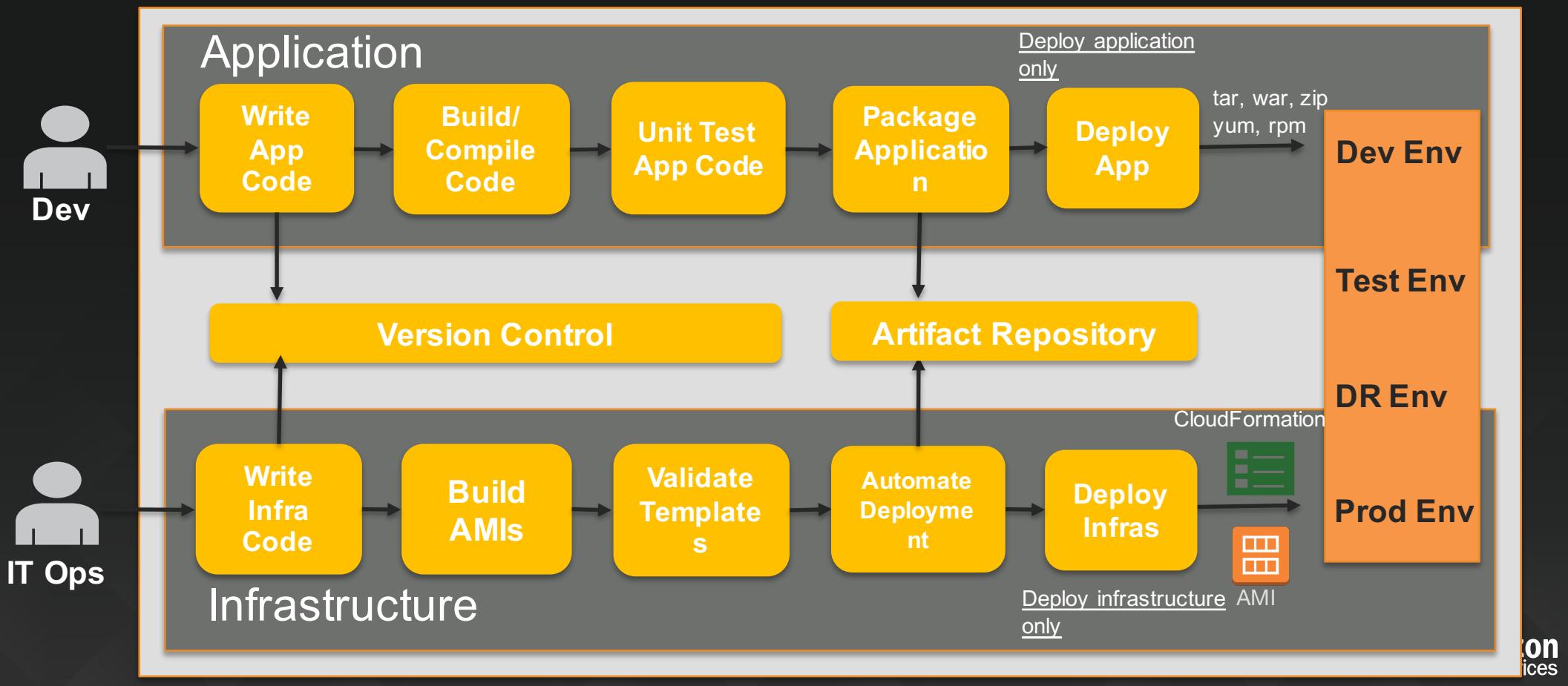
- Automated, repeatable process to push changes
- Hardens, de-risks the deployment process
- Immediate feedback from users
- Supports A/B testing or “We test customer reactions to features in production”
- Gives us a breadth of data points across our applications

Automated Testing

- Robust, automated testing drives confidence in the entire process
- Testing is not viewed as a “*phase*” of a project
 - Testing starts on **day one**
- View failures as successes
- Don’t overlook testing the infrastructure
 - Are you ready for the Simian Army?
- Testing is **everyone’s responsibility**



Continuous Integration / Deployment & Automation



DevOps Stack on AWS

Code Build Test Deploy Provision Monitor

CodeCommit

CodePipeline

AWS Elastic Beanstalk

AWS Opsworks

AWS Elastic Container Service

CodeDeploy

CloudFormation

CloudWatch



CodeCommit

Secure, scalable, and managed Git source control



`git push`

SSH or HTTPS



CodeCommit



Git objects
in Amazon S3



Git index
in Amazon
DynamoDB



Encryption key
in AWS KMS

- Data redundancy across Availability Zones
- Data-at-rest encryption
- Integrated with AWS Identity and Access Management
- No repo size limit

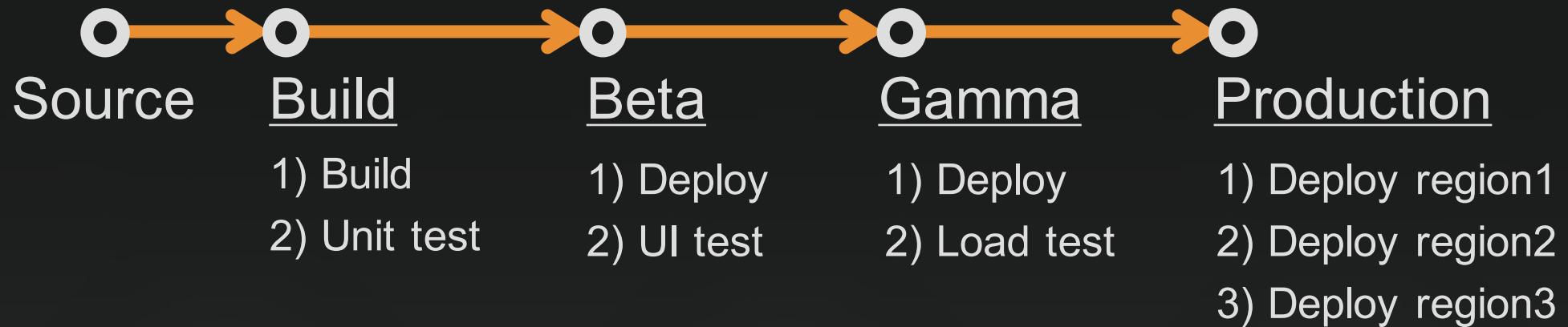
Same Git experience

```
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aws-cli
Cloning into 'aws-cli'...
Receiving objects: 100% (16032/16032), 5.55 MiB | 1.25 MiB/s, done.
Resolving deltas: 100% (9900/9900), done.
Checking connectivity... done.
$ nano README.rst
$ git commit -am 'updated README'
[master 4fa0318] updated README
 1 file changed, 1 insertion(+)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 297 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote:
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aws-cli
 4dacd6d..4fa0318  master -> master
```



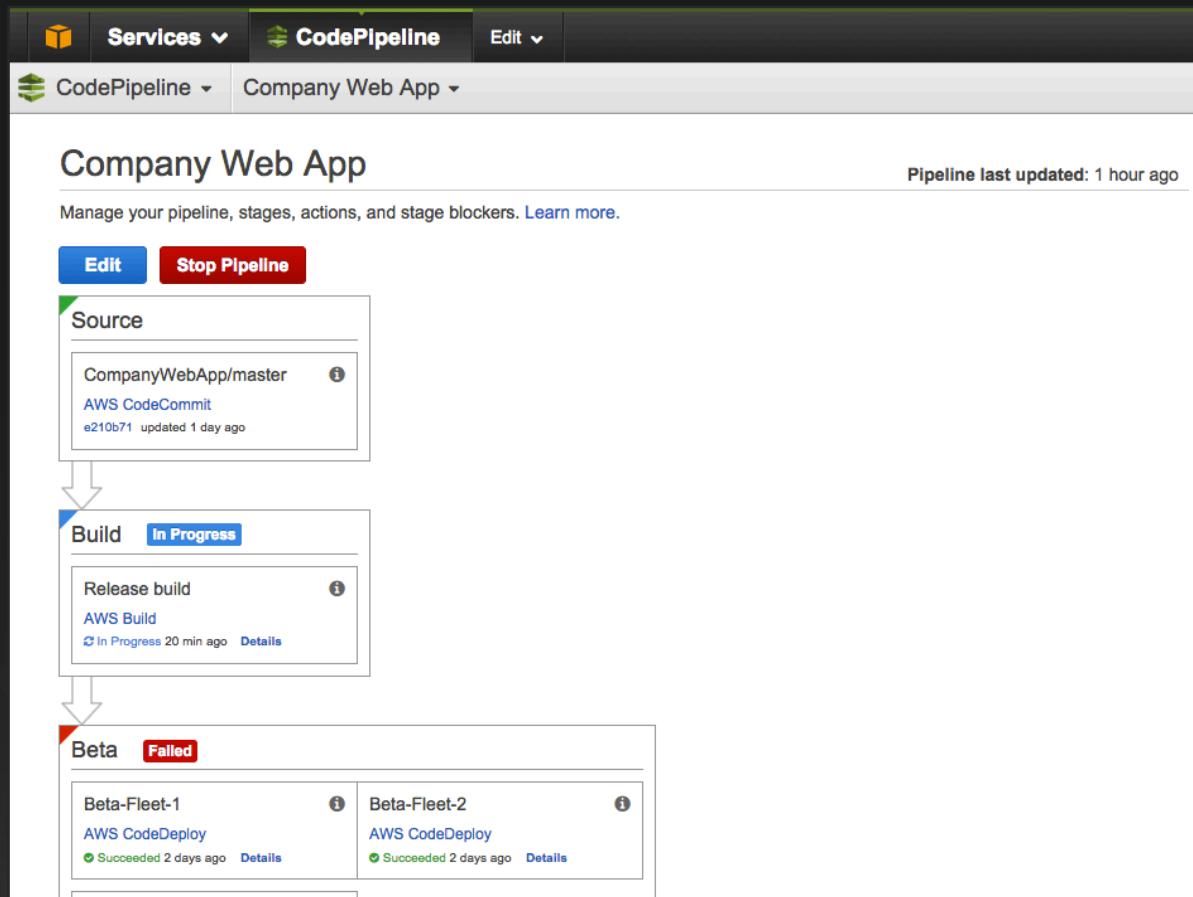
CodePipeline

Continuous delivery and release automation, just like Amazon



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

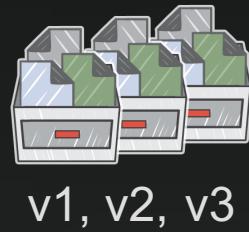
CodePipeline : Mockup



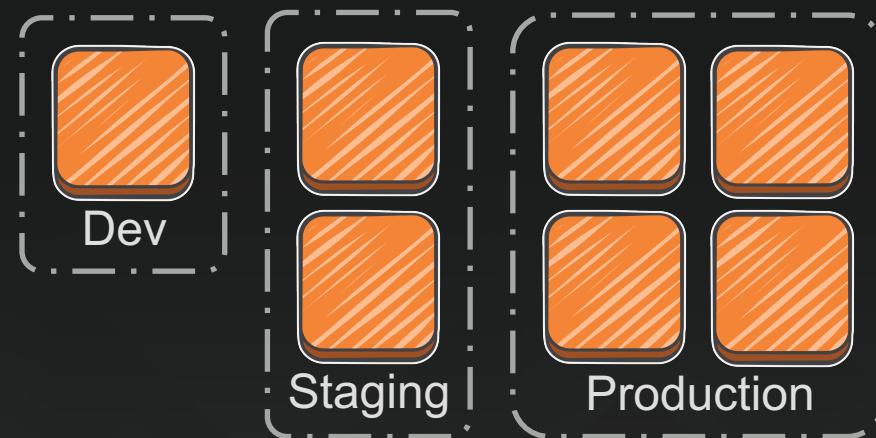
CodeDeploy

Coordinate automated deployments, just like Amazon

Application
revisions



Deployment groups



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

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
```

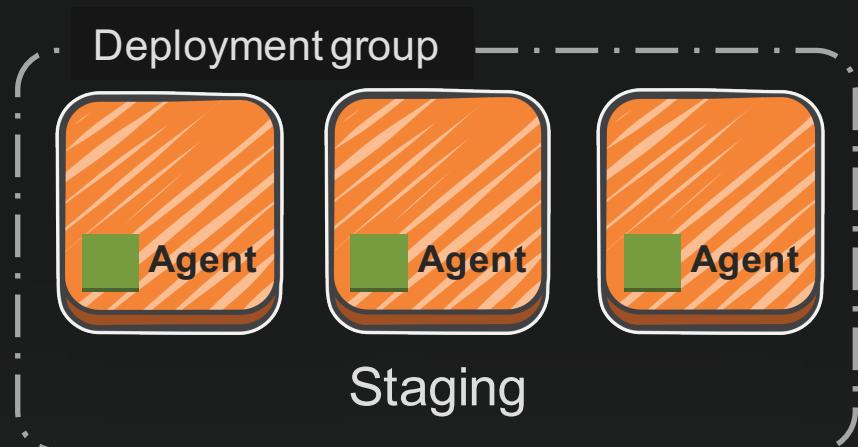


Step 1: Package your application (with an AppSpec file)

```
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!

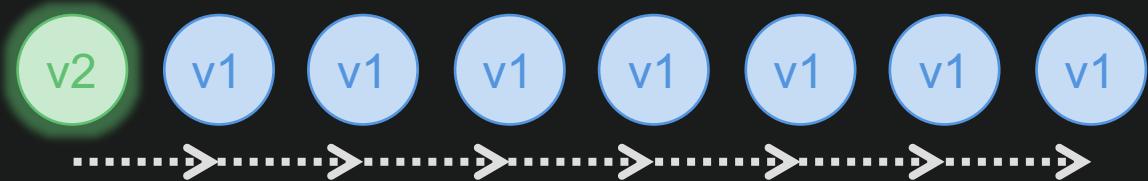
AWS CLI & SDKs
AWS Console
CI / CD Partners
GitHub

```
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



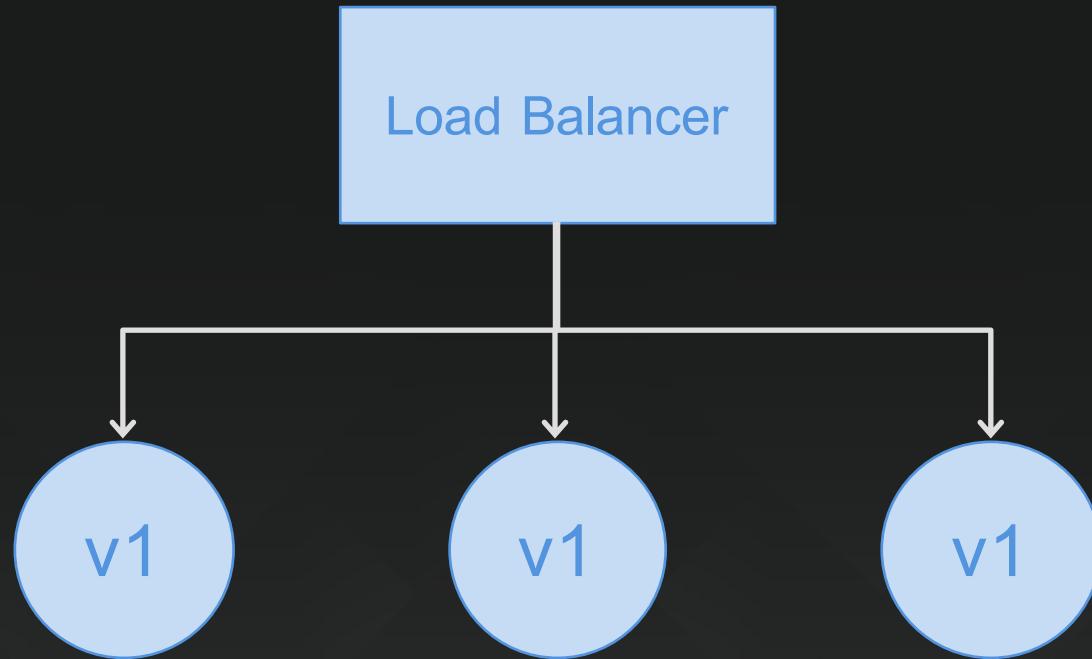
Half-at-a-time



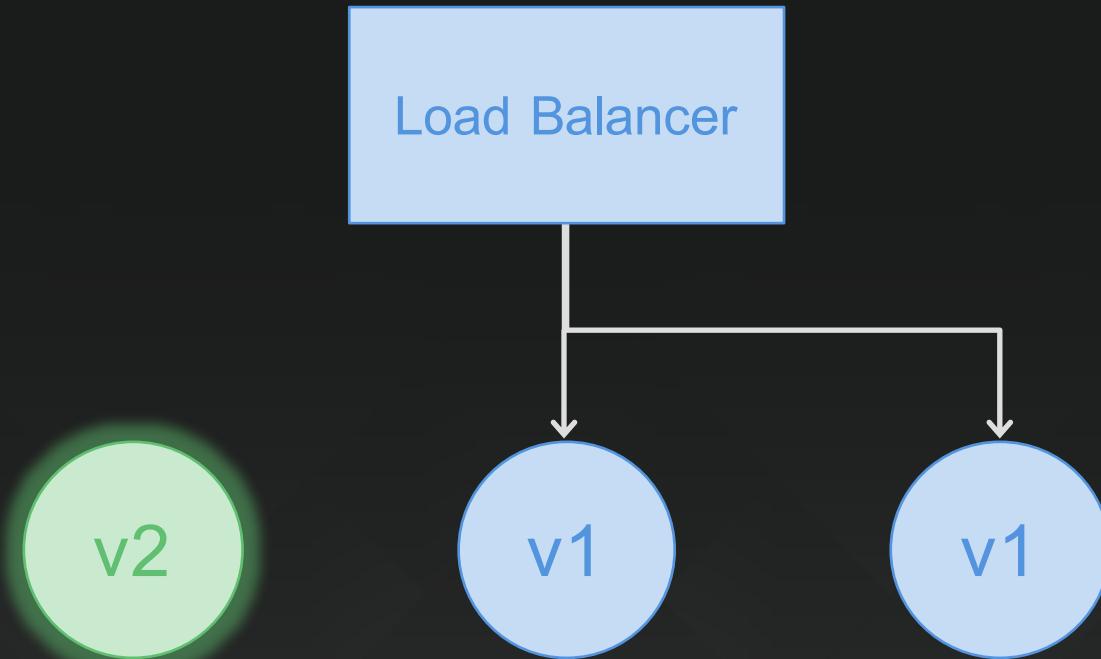
All-at-once



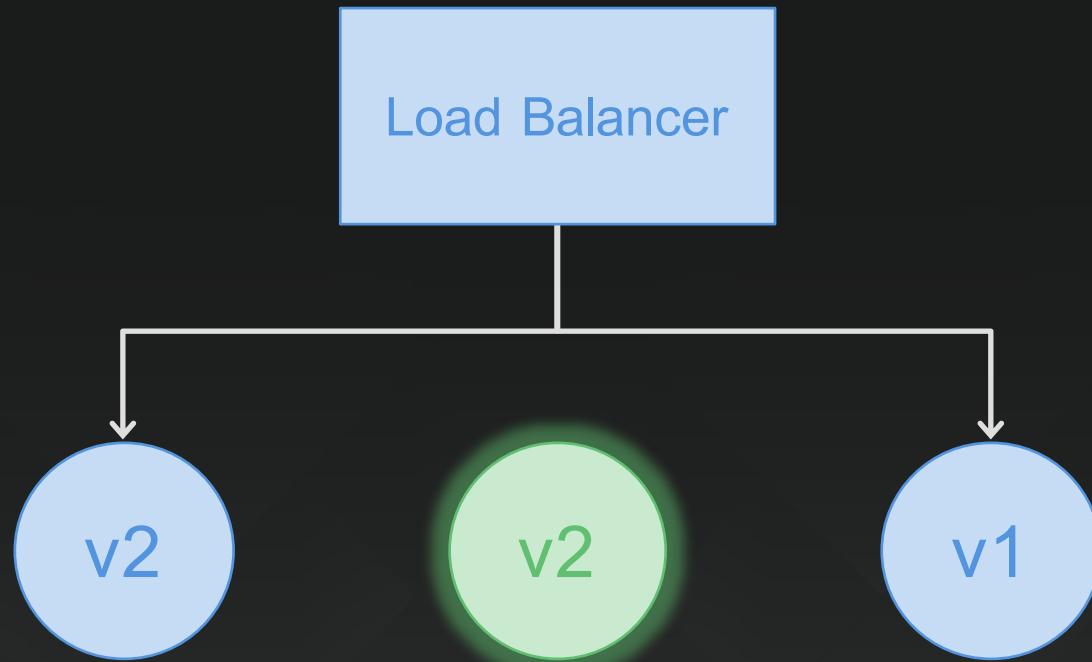
Rolling update – Deploy without downtime



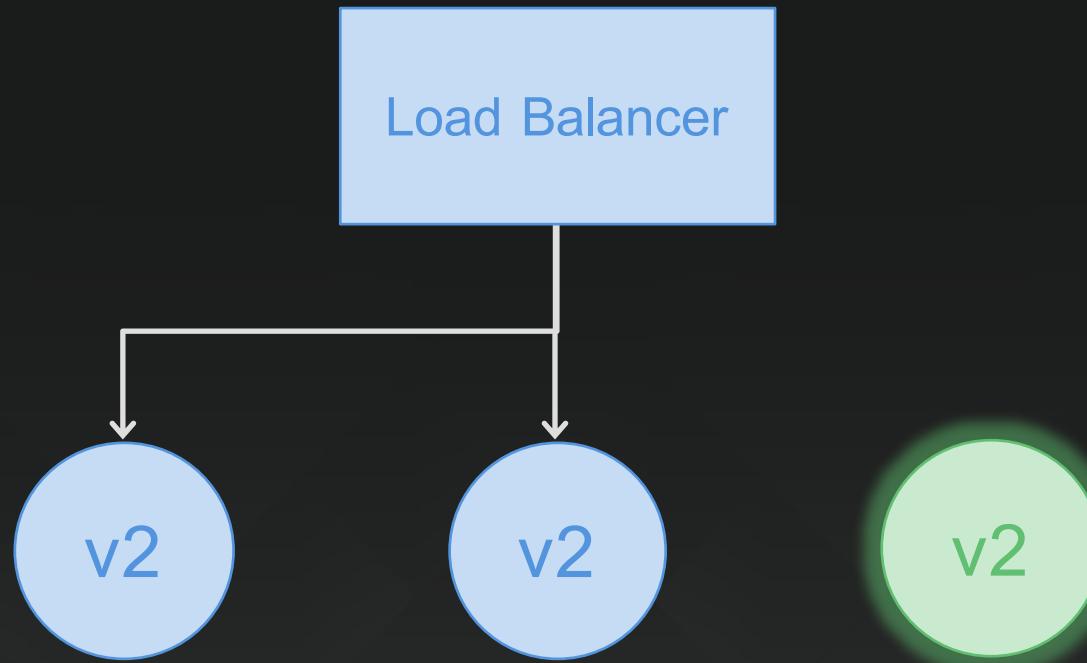
Rolling update – Deploy without downtime



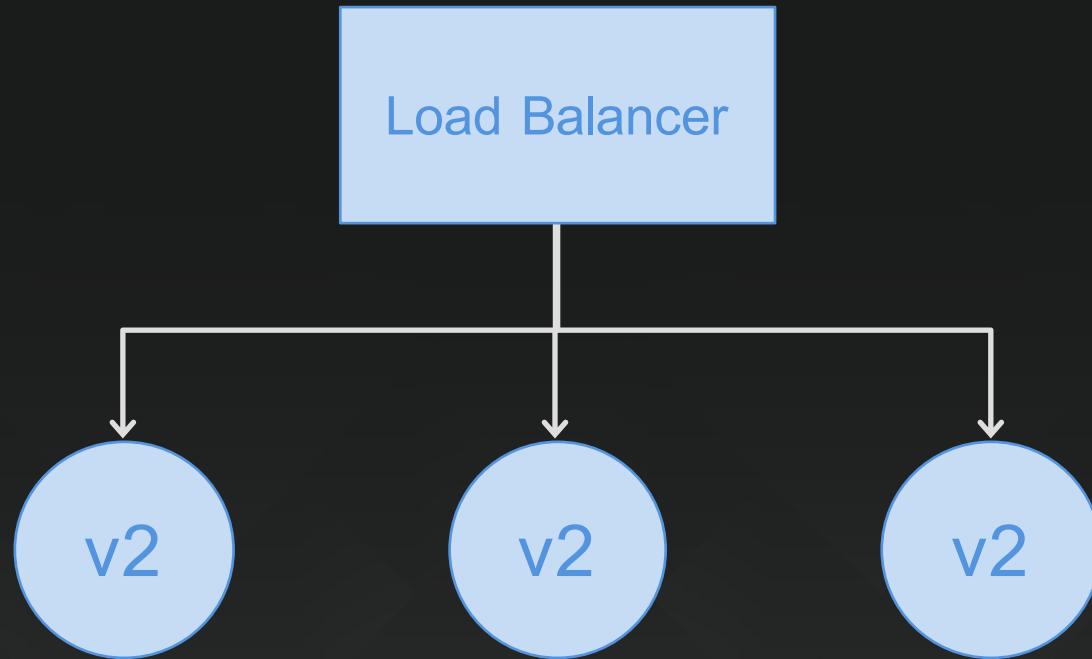
Rolling update – Deploy without downtime



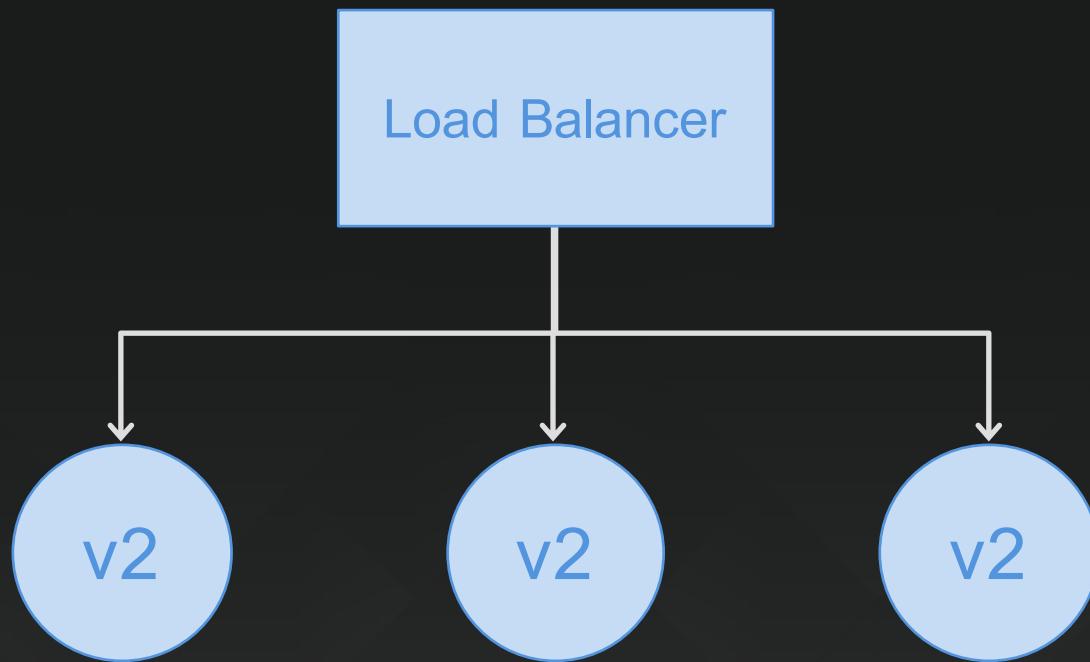
Rolling update – Deploy without downtime



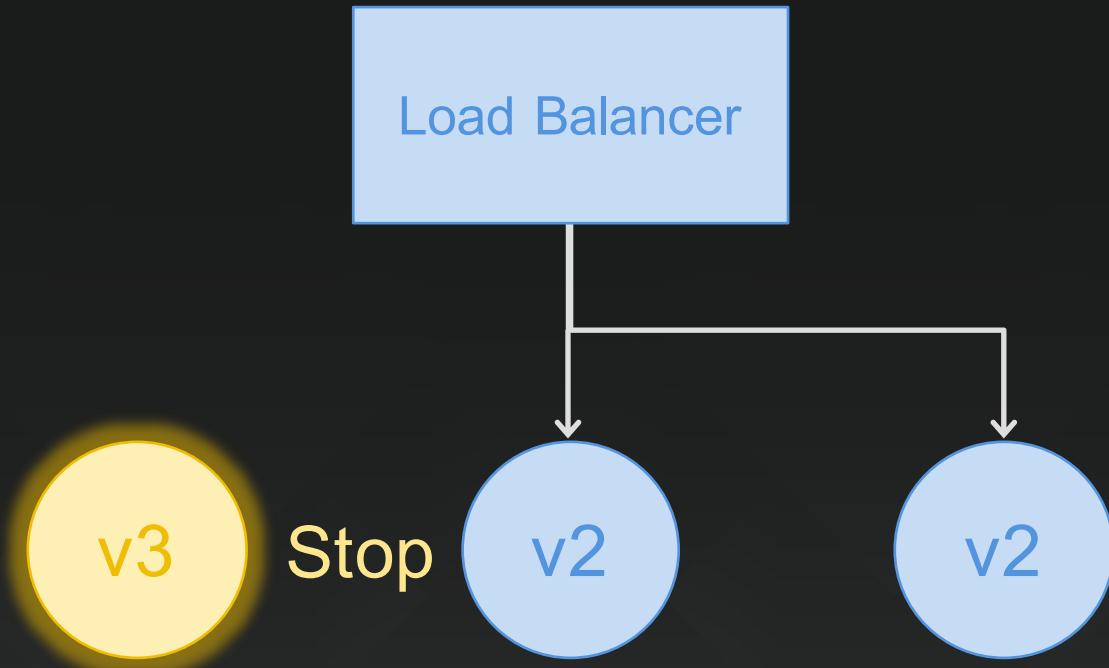
Rolling update – Deploy without downtime



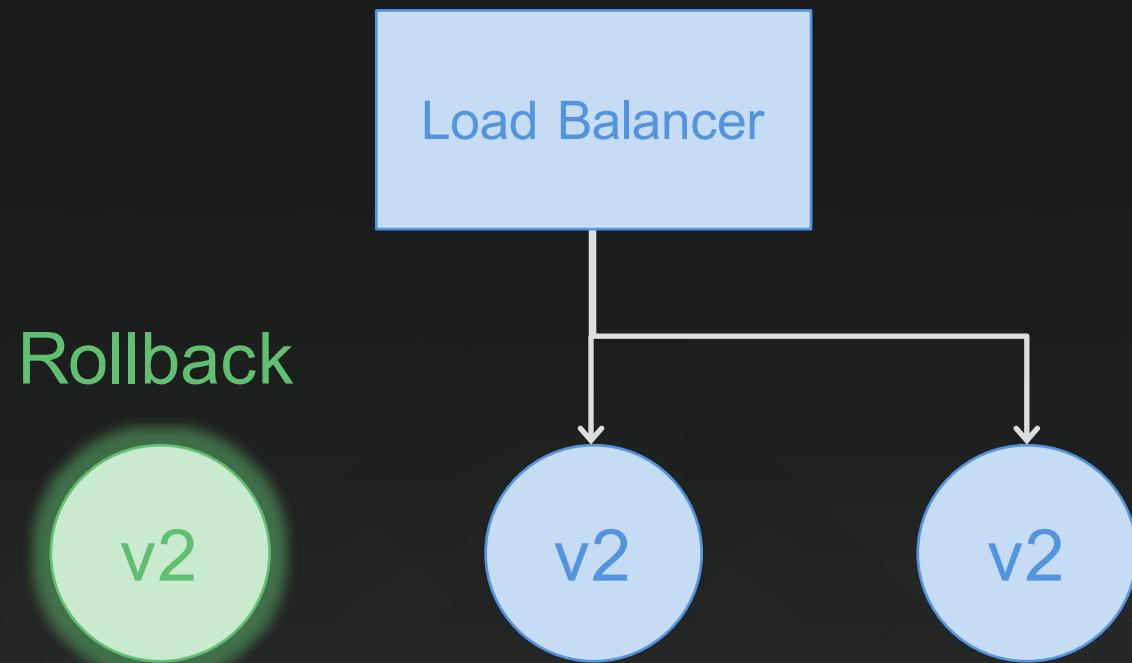
Health Tracking – Catch deployment problems



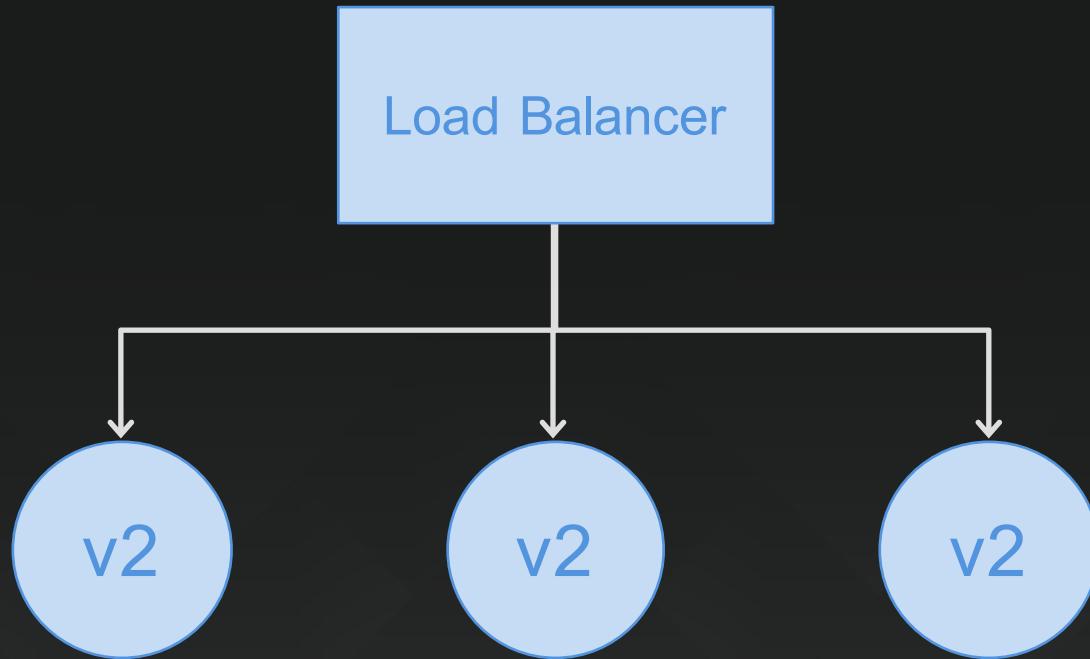
Health tracking – Catch deployment problems



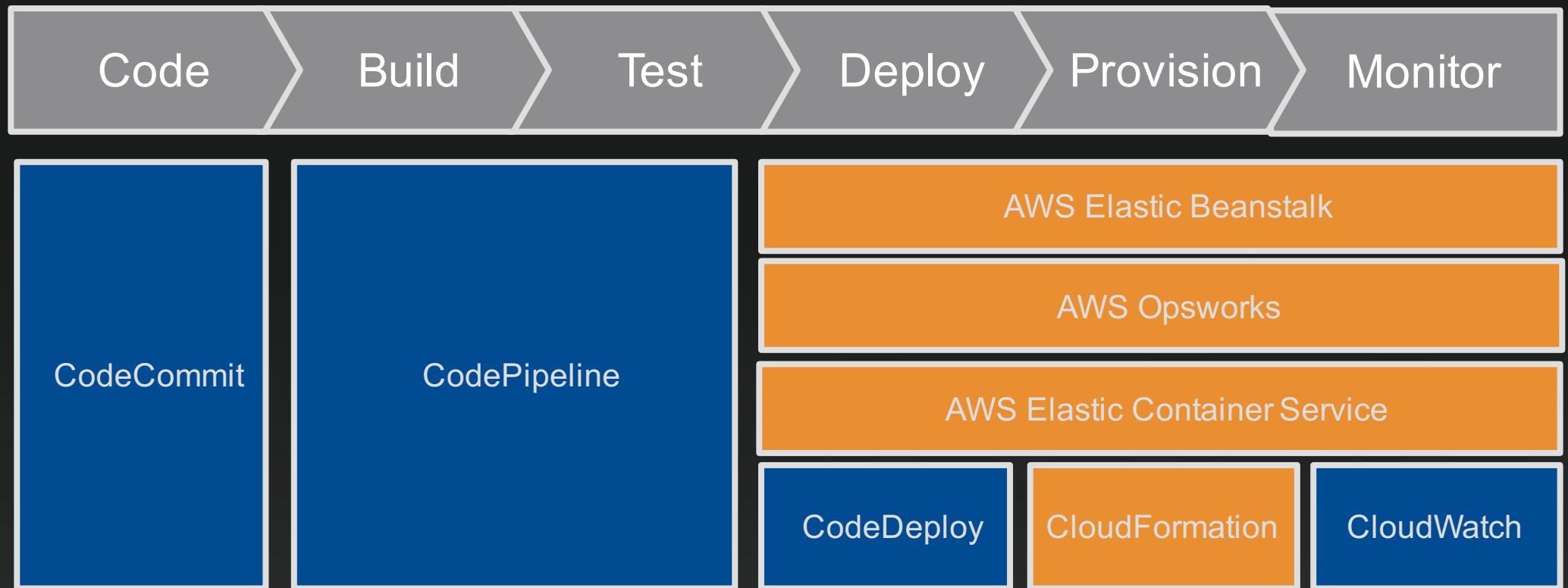
Health tracking – Catch deployment problems



Health tracking – Catch deployment problems



DevOps Stack on AWS



Automate with AWS CLI



```
aws ec2 run-instances ^
--image-id ami-5c99eb34 ^
--key-name awlaruseast ^
--user-data "file://Z:\Presentations\LnL Jan 15 - ALM Worksho
--count 4 ^
--instance-type t2.micro ^
--iam-instance-profile Name=CodeDeployRole-EC2Instances ^
--security-groups CodeDeployDemo-Windows-Security-Group ^
--region us-east-1
```

Automate with AWS SDK

Android

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Browser

[Install »](#)
[Documentation »](#)
[Learn more »](#)

iOS

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Java

[Install »](#)
[Documentation »](#)
[Learn more »](#)

.NET

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Node.js

[Install »](#)
[Documentation »](#)
[Learn more »](#)

PHP

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Python

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Ruby

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Go

[Install »](#)
[Documentation »](#)
[Learn more »](#)

Eclipse

[Install »](#)
[Documentation »](#)
[Learn more »](#)



Visual Studio

[Install »](#)
[Documentation »](#)
[Learn more »](#)



AWS Cloudformation



AWS CloudFormation

- Infrastructure as Code
- Integrates with version control
- JSON format
- Templates
- Stacks
- Supports most AWS resource types

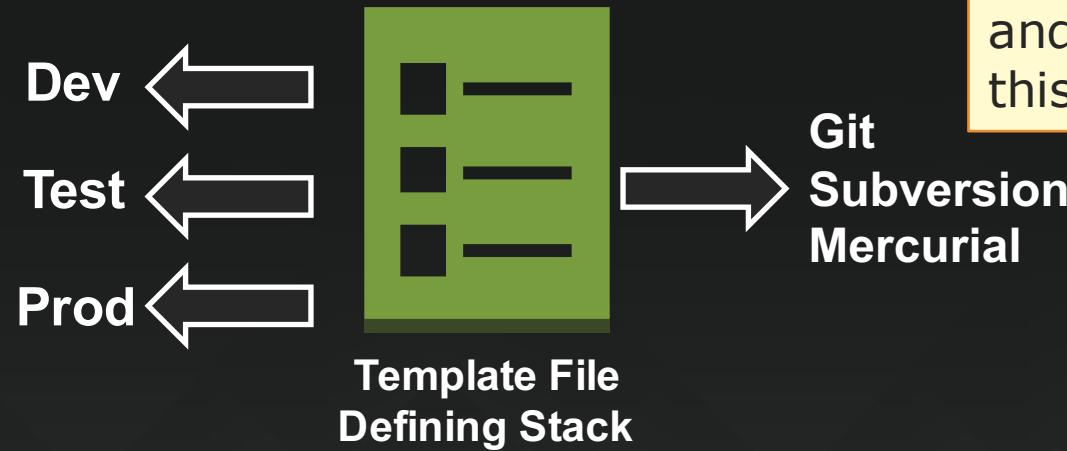
AWS Cloudformation

```
{  
    "Description" : "Create an EC2 instance running the Amazon Linux 32 bit AMI.",  
    "Parameters" : {  
        "KeyPair" : {  
            "Description" : "The EC2 Key Pair to allow SSH access to the instance",  
            "Type" : "String"  
        }  
    },  
    "Resources" : {  
        "Ec2Instance" : {  
            "Type" : "AWS::EC2::Instance",  
            "Properties" : {  
                "KeyName" : { "Ref" : "KeyPair" },  
                "ImageId" : "ami-75g0061f",  
                "InstanceType" : "m1.medium"  
            }  
        }  
    },  
    "Outputs" : {  
        "InstanceId" : {  
            "Description" : "The InstanceId of the newly created EC2 instance",  
            "Value" : { "Ref" : "Ec2Instance" }  
        }  
    }  
}
```



AWS Cloudformation

Build out multiple environments, such as for Development, Test, and Production using the template

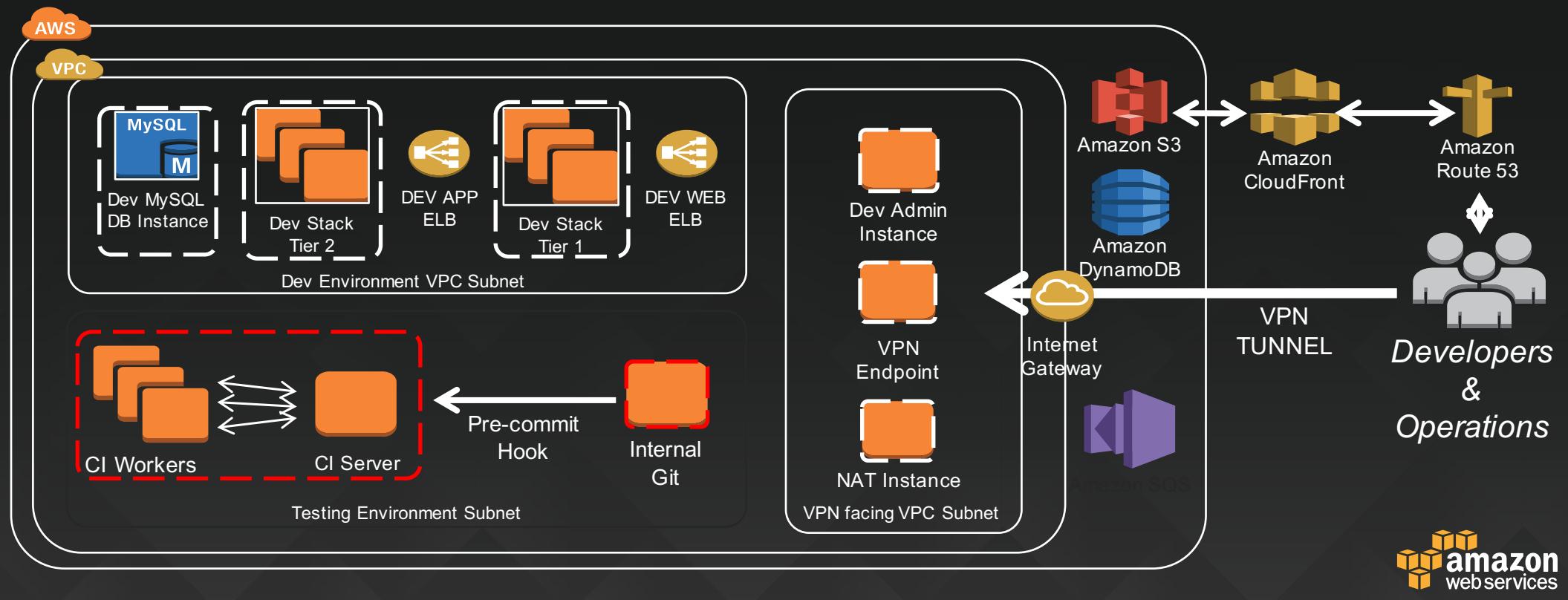


The entire application can be represented in an AWS CloudFormation template.

Use the version control system of your choice to store and track changes to this template

Describing Infrastructure with Code

This entire infrastructure stack can be constructed, configured, and deployed with code:



Infrastructure Code

The code that describes infrastructure should inherit the same values applied to application code

- Not JUST revision control
- Make use of bug tracking/ticketing systems
- Peer reviews of changes before they happen
- Establish infrastructure code *patterns/designs*
- Test infrastructure changes like code changes

AWS Opsworks



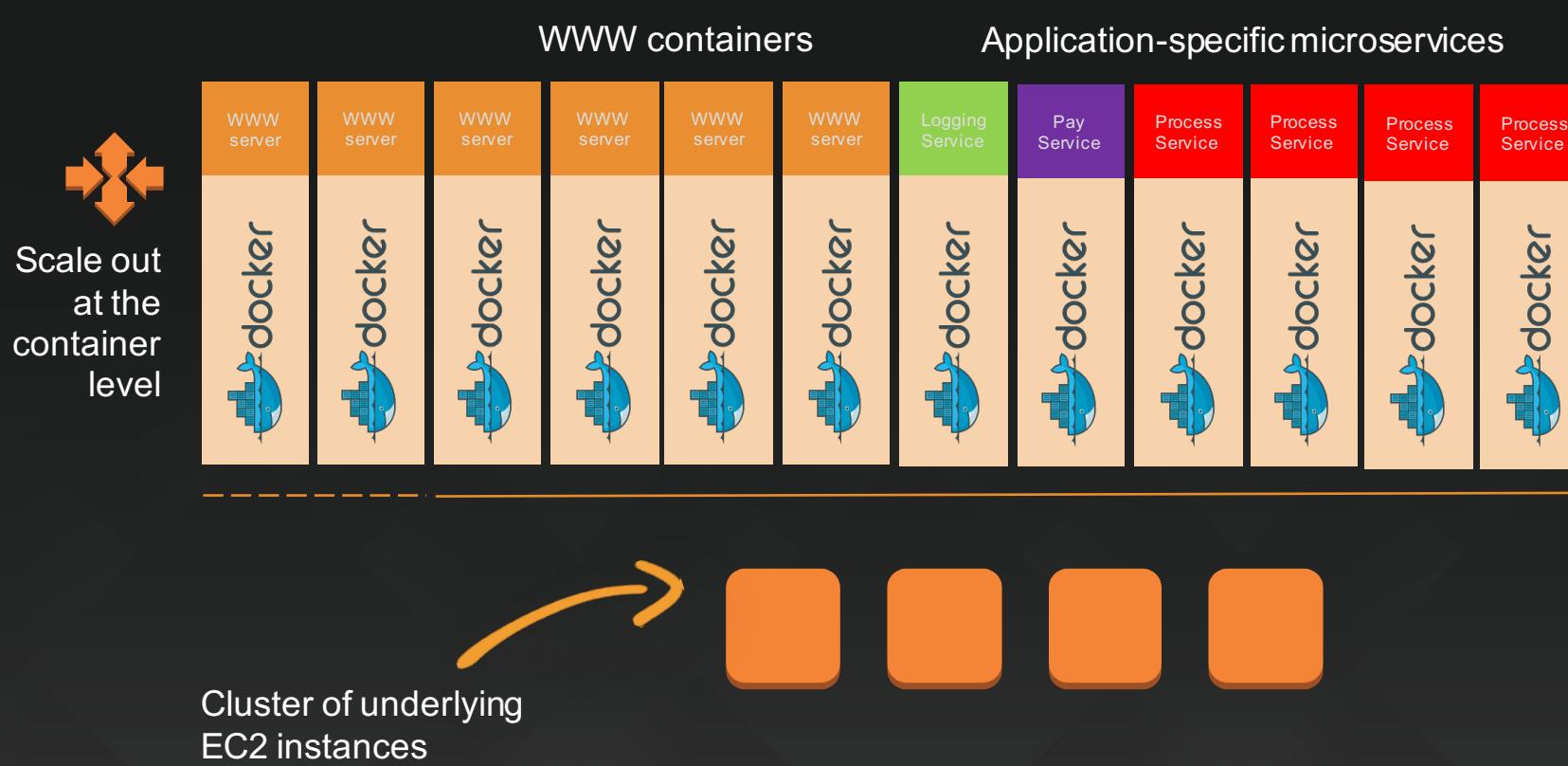
- AWS OpsWorks makes it easy to deploy & operate apps of all shapes and sizes
- Define configurations for your entire environment in a format that you can maintain and version just like your application source code
- Uses the Chef framework so you can bring your own recipes or leverage 100's of community-built configurations

AWS Beanstalk



- Quickly deploy and manage applications in the AWS cloud without worrying about the application infrastructure
- Automatically handles all the details of resource provisioning, load balancing & auto-scaling
- Launch your Java, PHP, .NET, Node.js, Python, Ruby and Docker applications in a matter of minutes

AWS Elastic Container Service



Launching containers is in the order of seconds – very fast to react



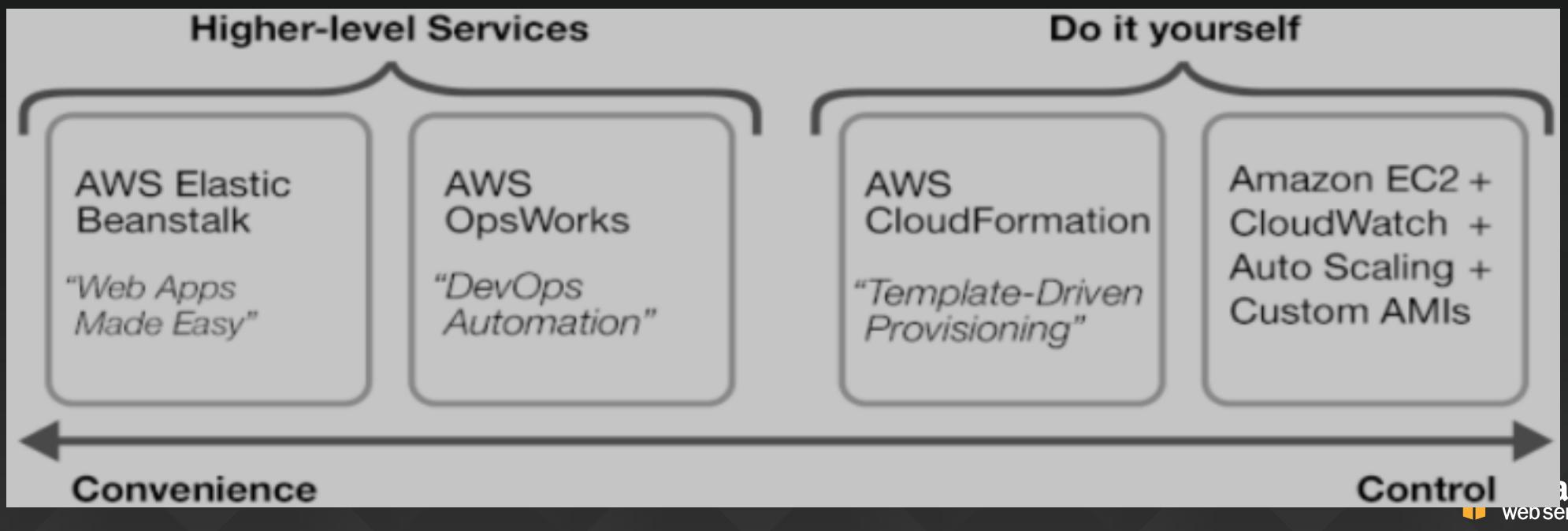
What are we deploying? How much control do we want?

AWS Tools

- CloudFormation
- Elastic Beanstalk
- OpsWorks

3rd-Party Tools

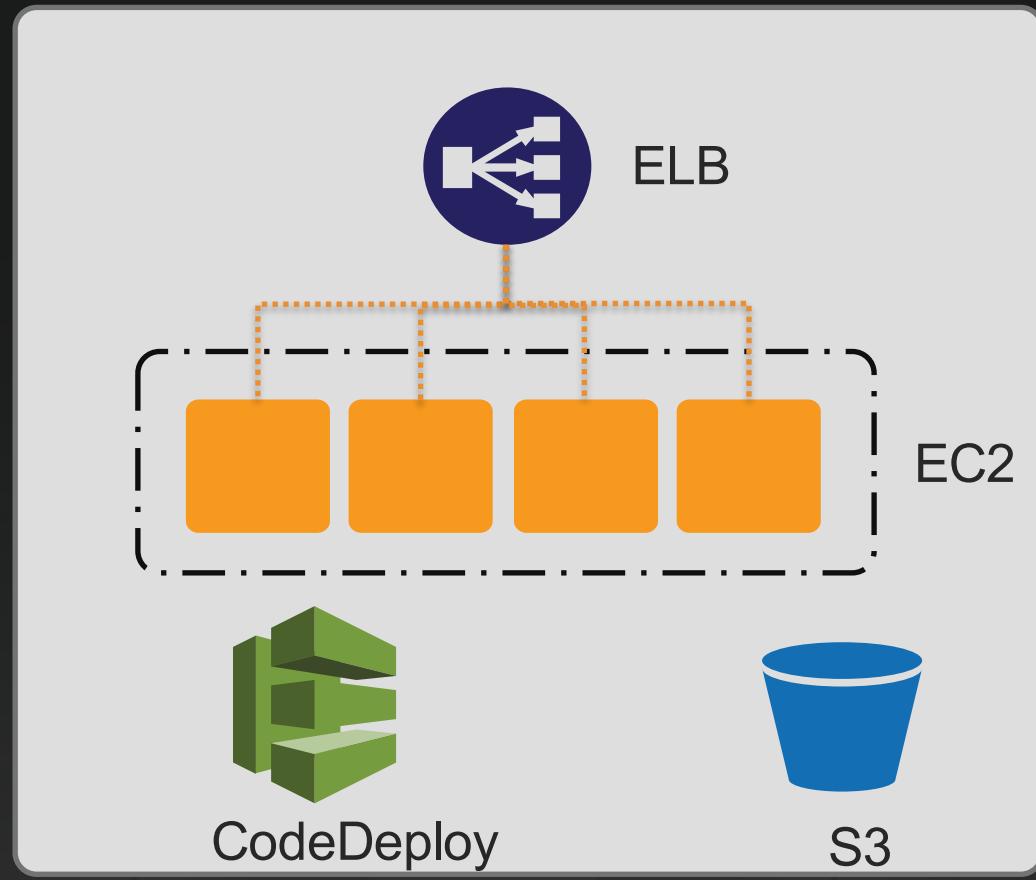
- Chef
- Puppet



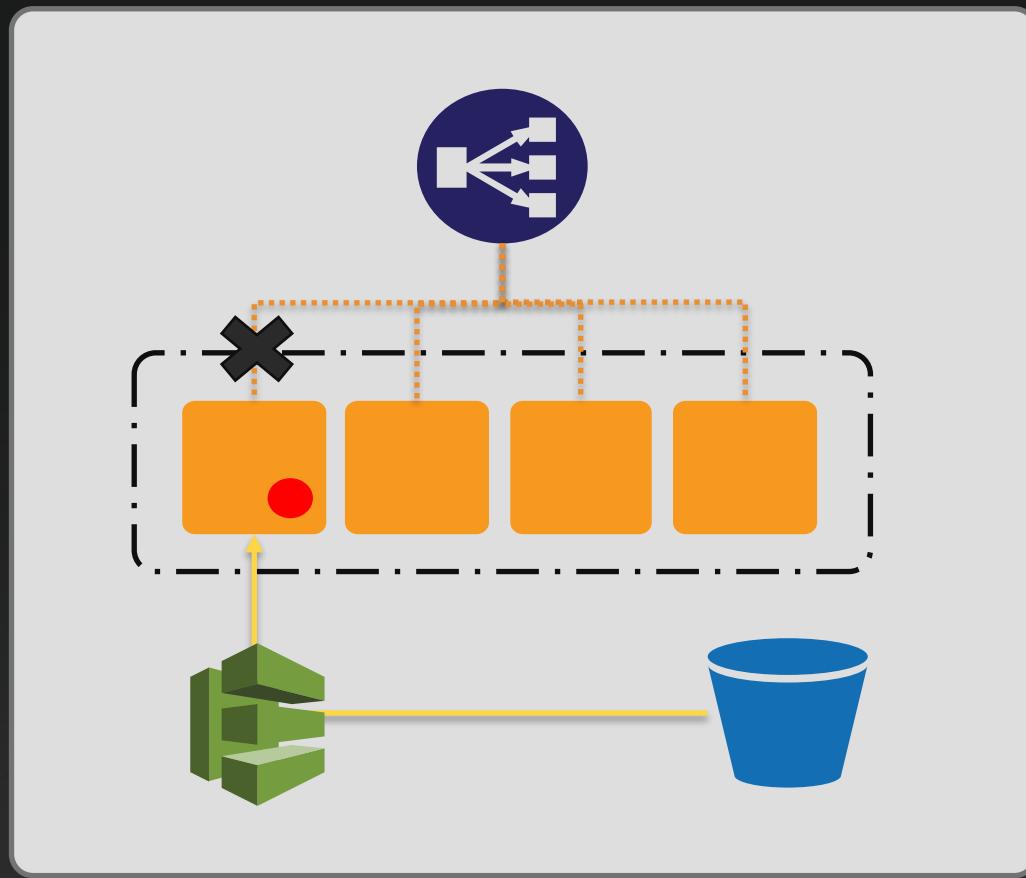
Deployment Approaches

- Deploy in place
 - Deploy all at once (Service outage)
 - Rolling updates
- Blue-Green Deployment
 - Discrete environment
 - Multiple environments from branches
 - Support A/B testing
 - “Rolling DNS”
- Alternate Blue-Green (Red-Black?) deployment
 - Alternate auto scaling group
 - Avoid messing with DNS

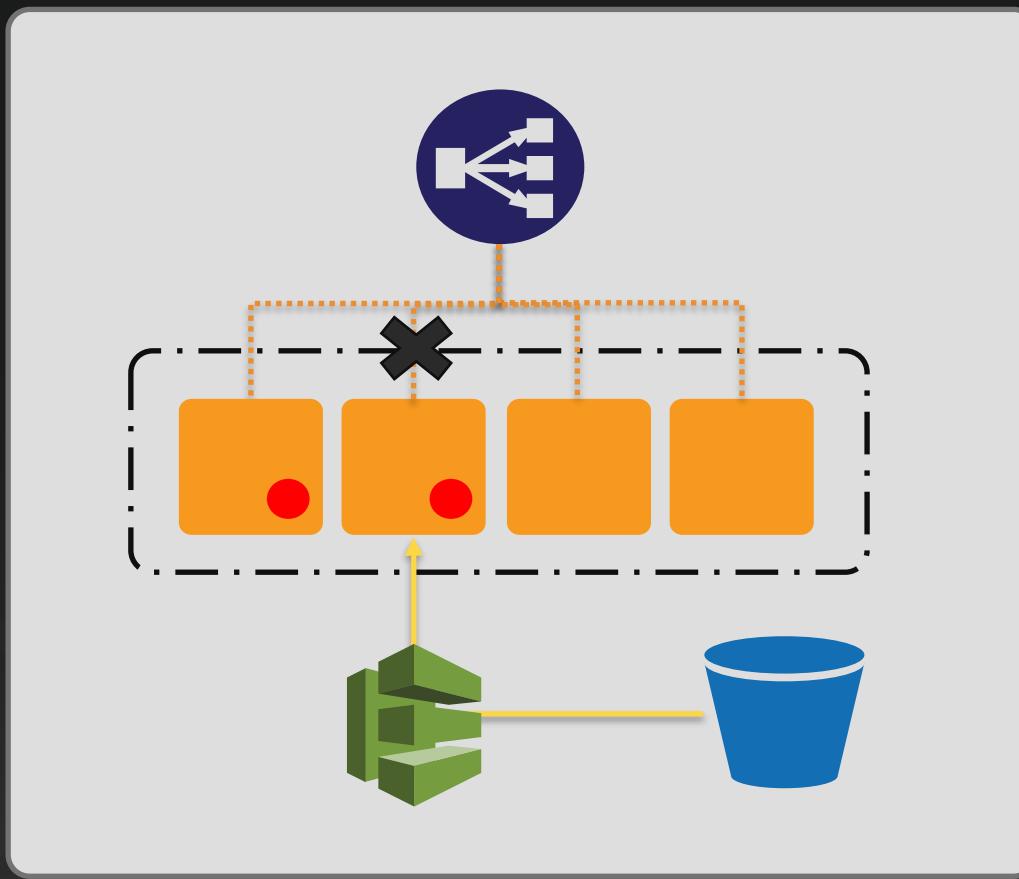
Deploy in place – Rolling update



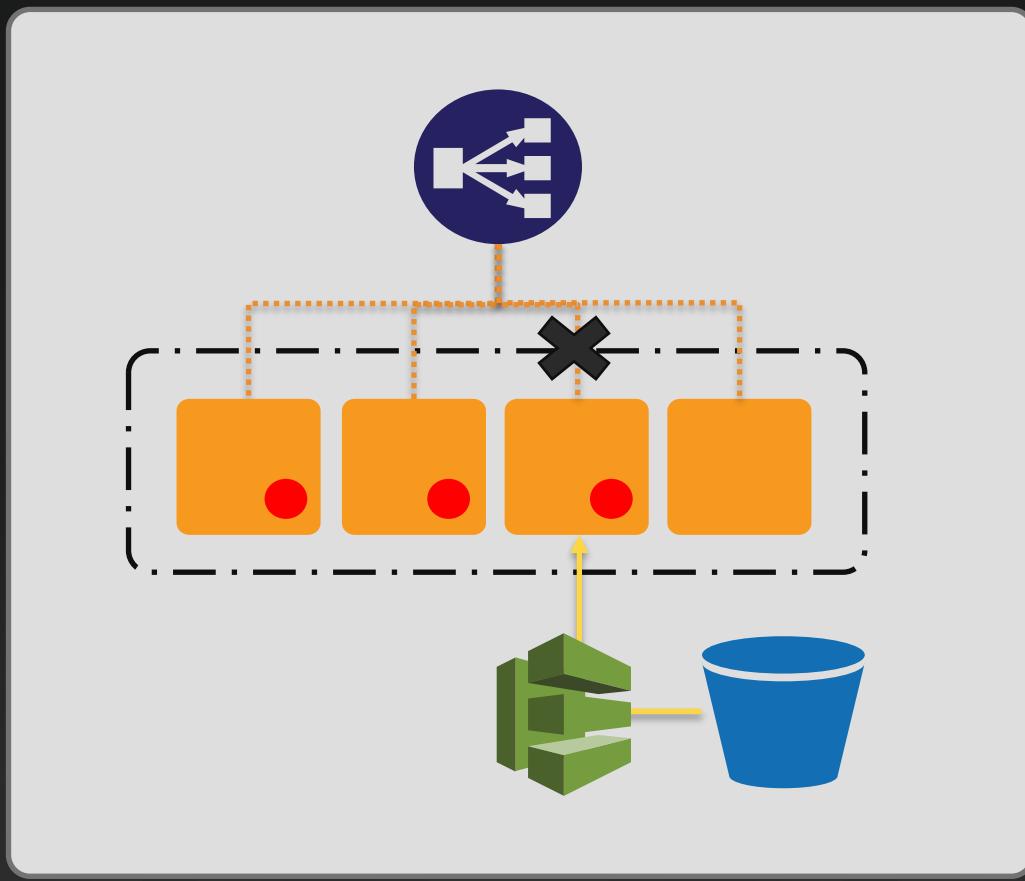
Deploy in place – Rolling update



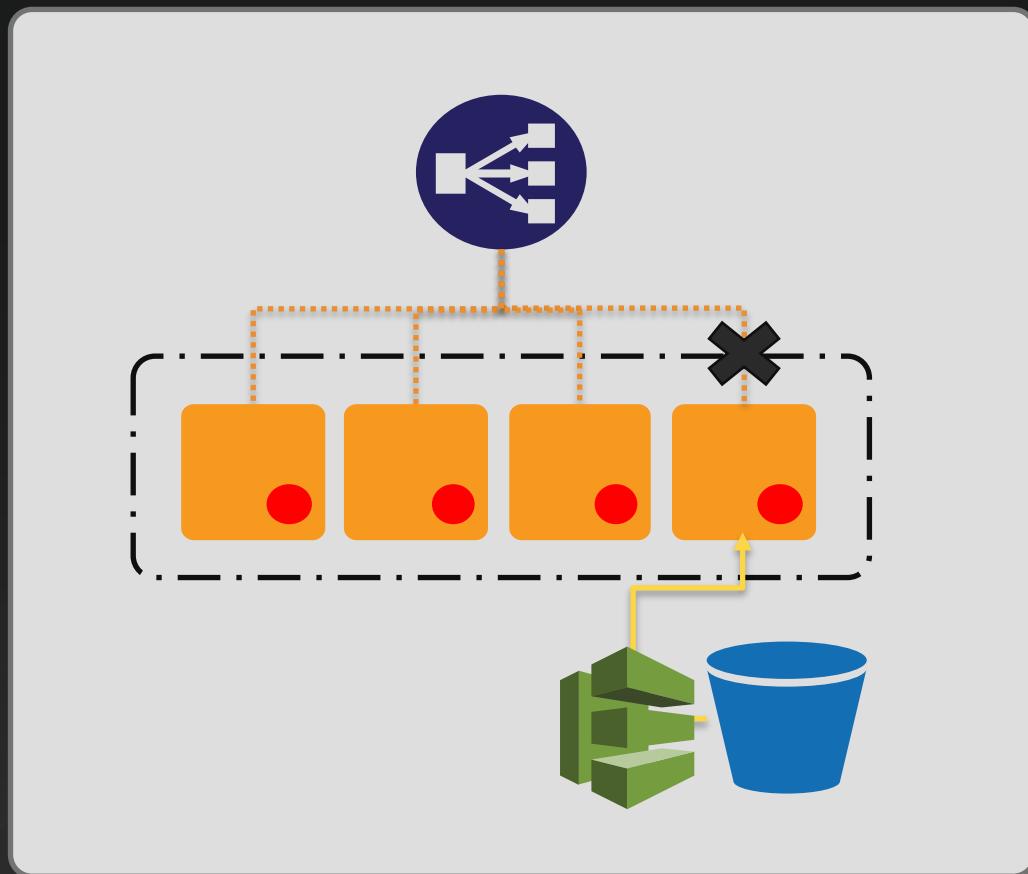
Deploy in place – Rolling update



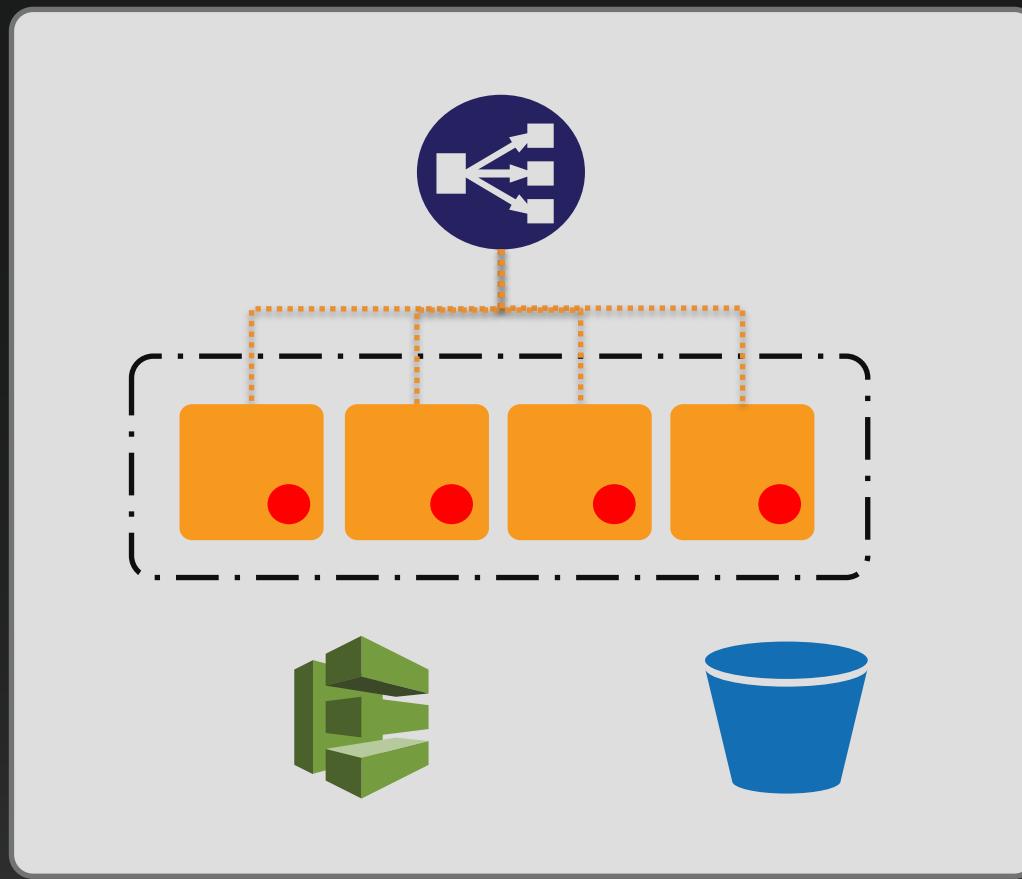
Deploy in place – Rolling update



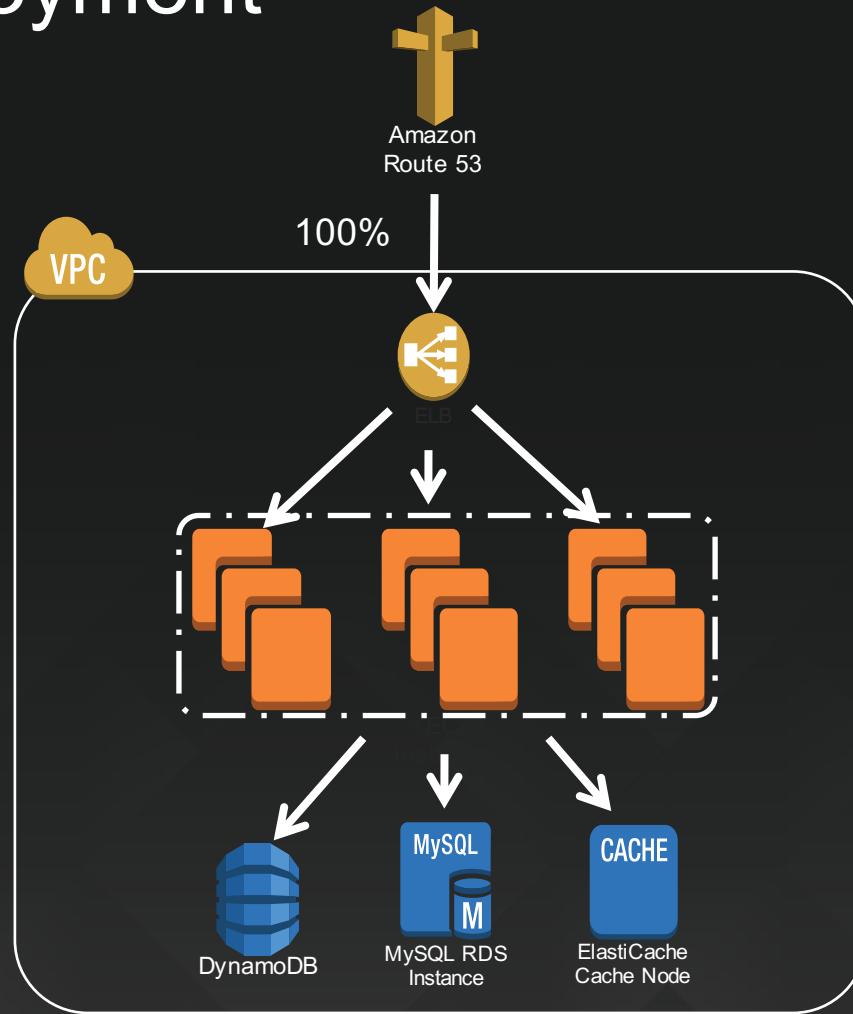
Deploy in place – Rolling update



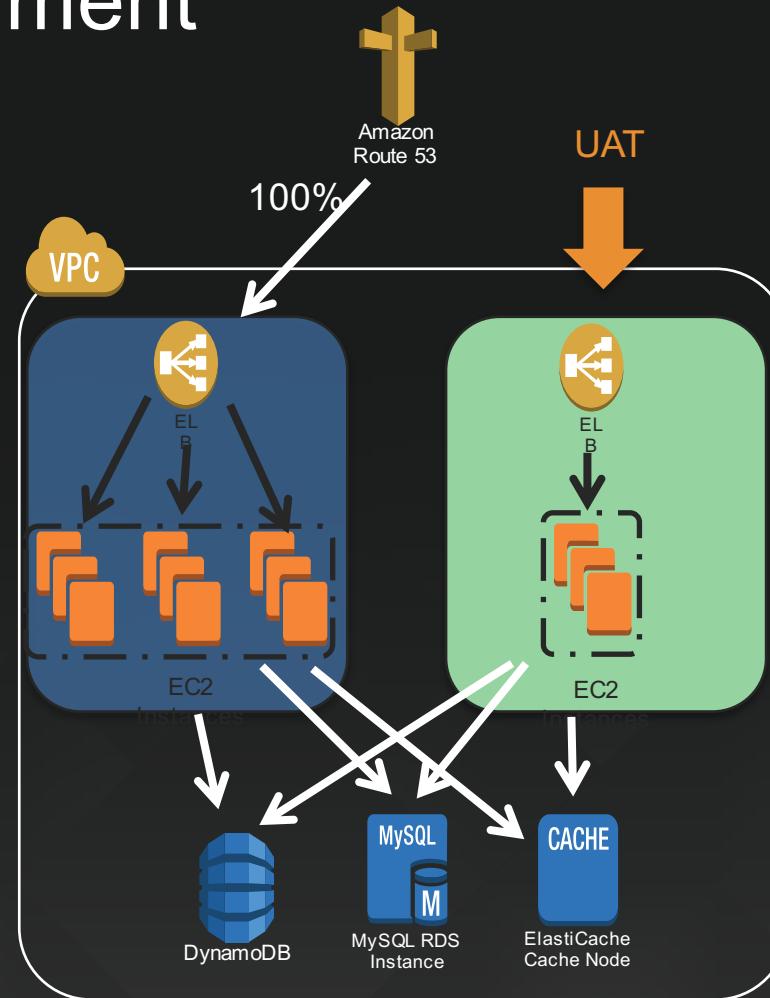
Deploy in place – Rolling update



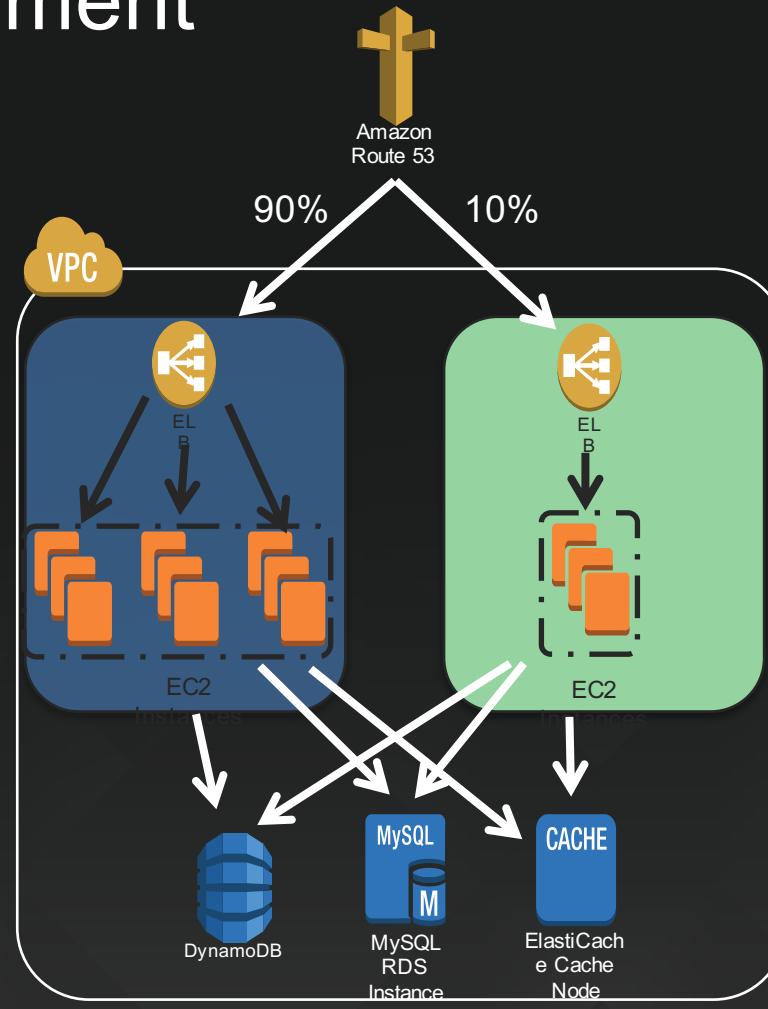
Blue-Green Deployment



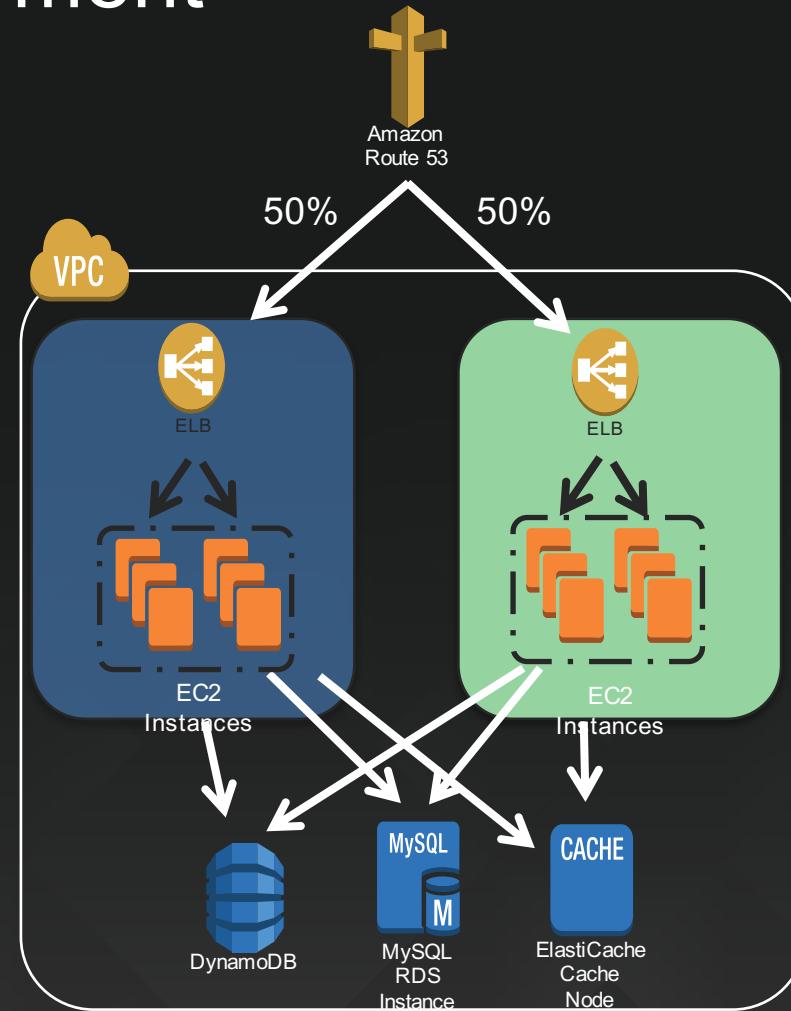
Blue-Green Deployment



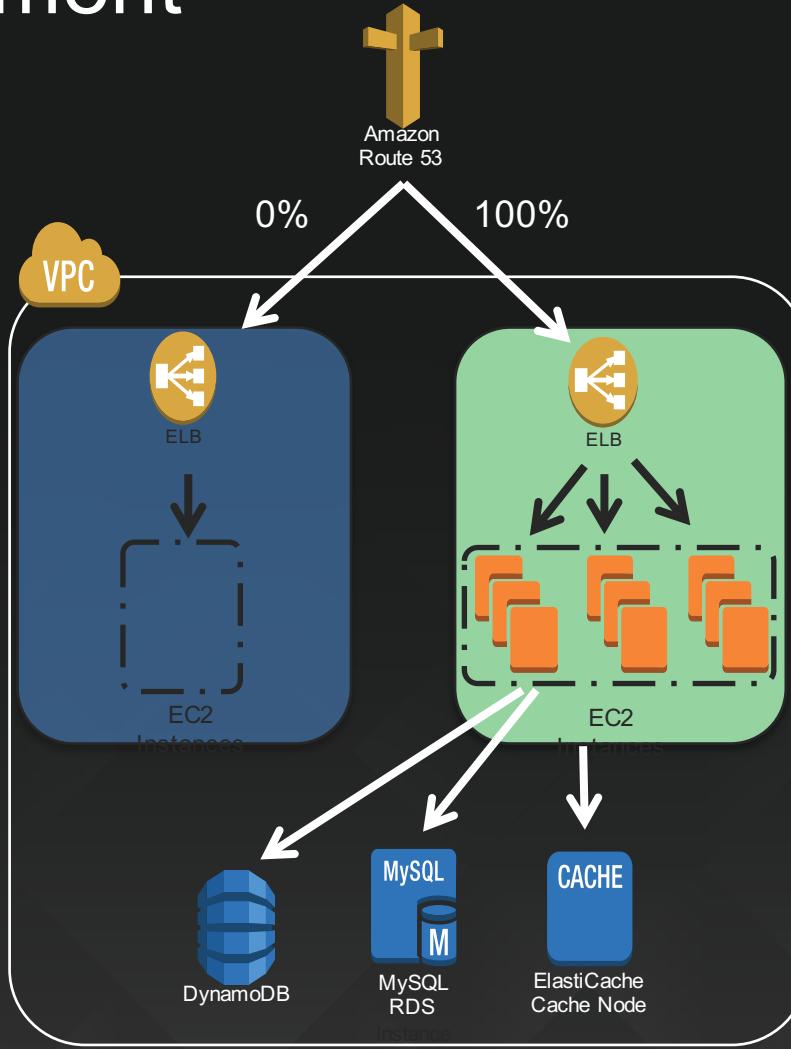
Blue-Green Deployment



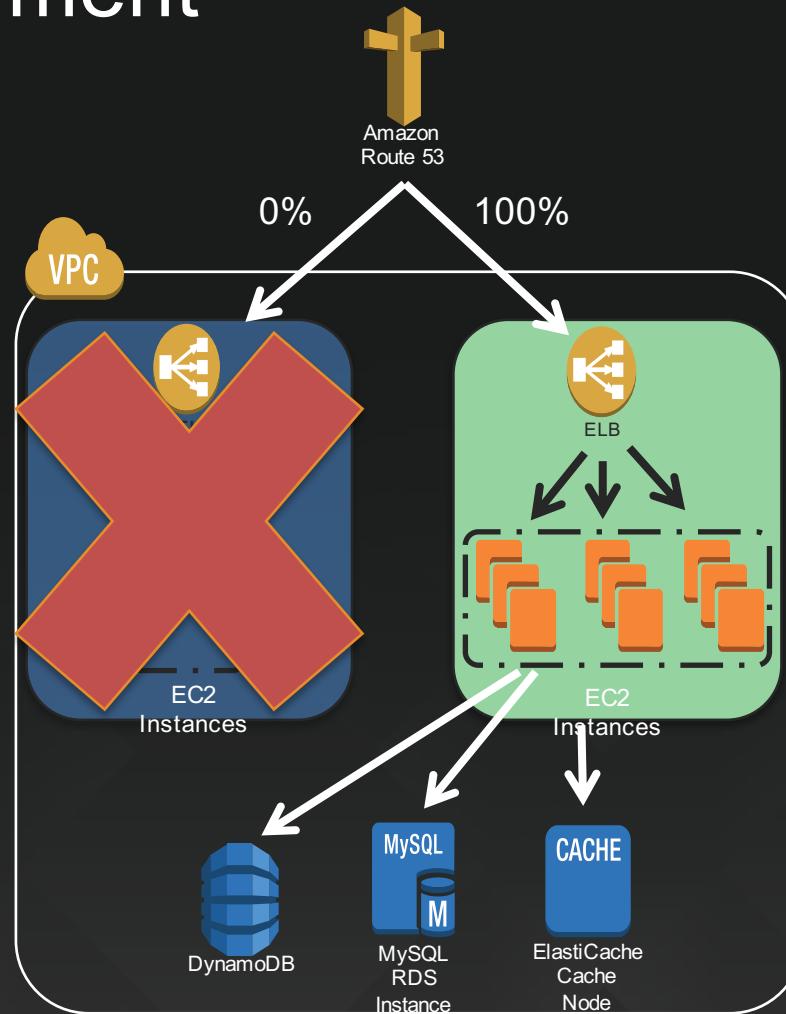
Blue-Green Deployment



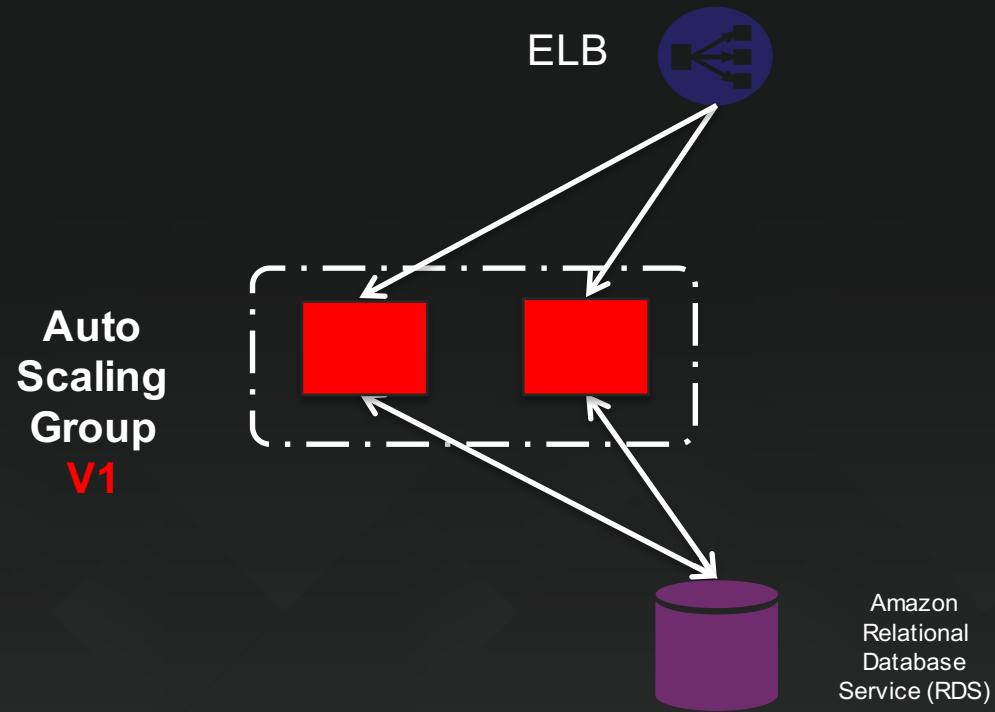
Blue-Green Deployment



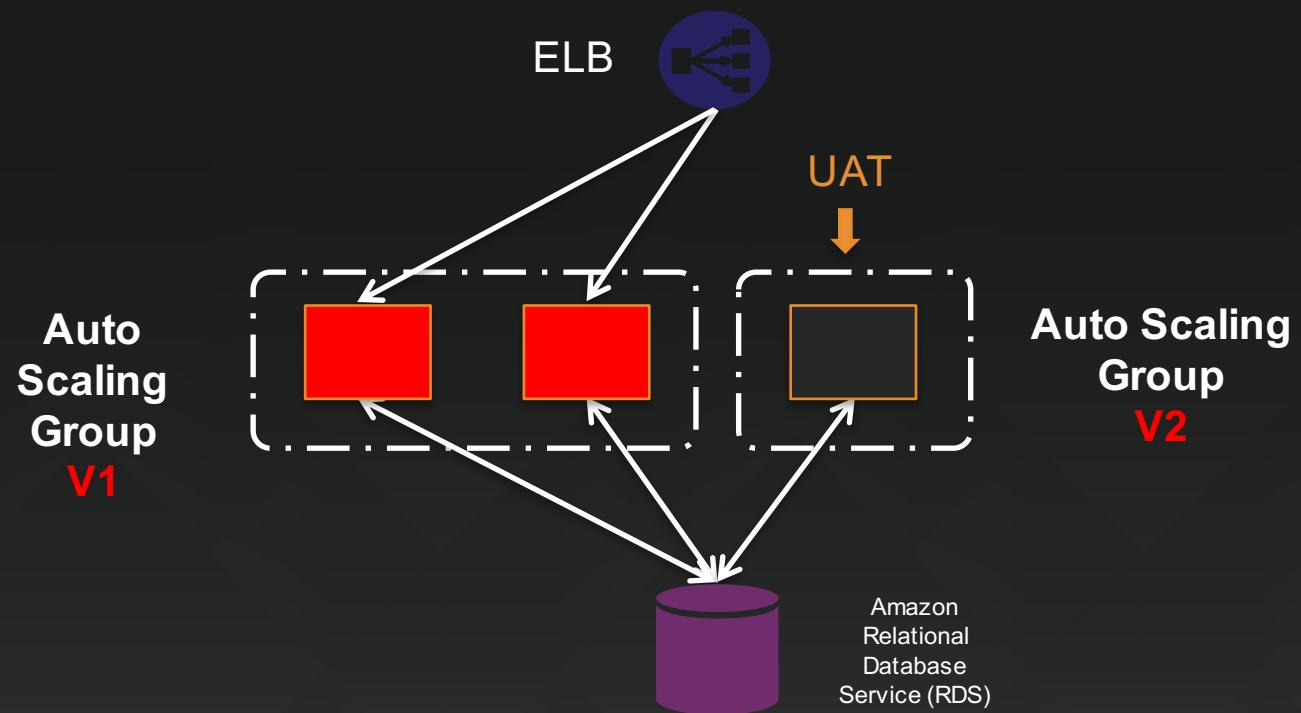
Blue-Green Deployment



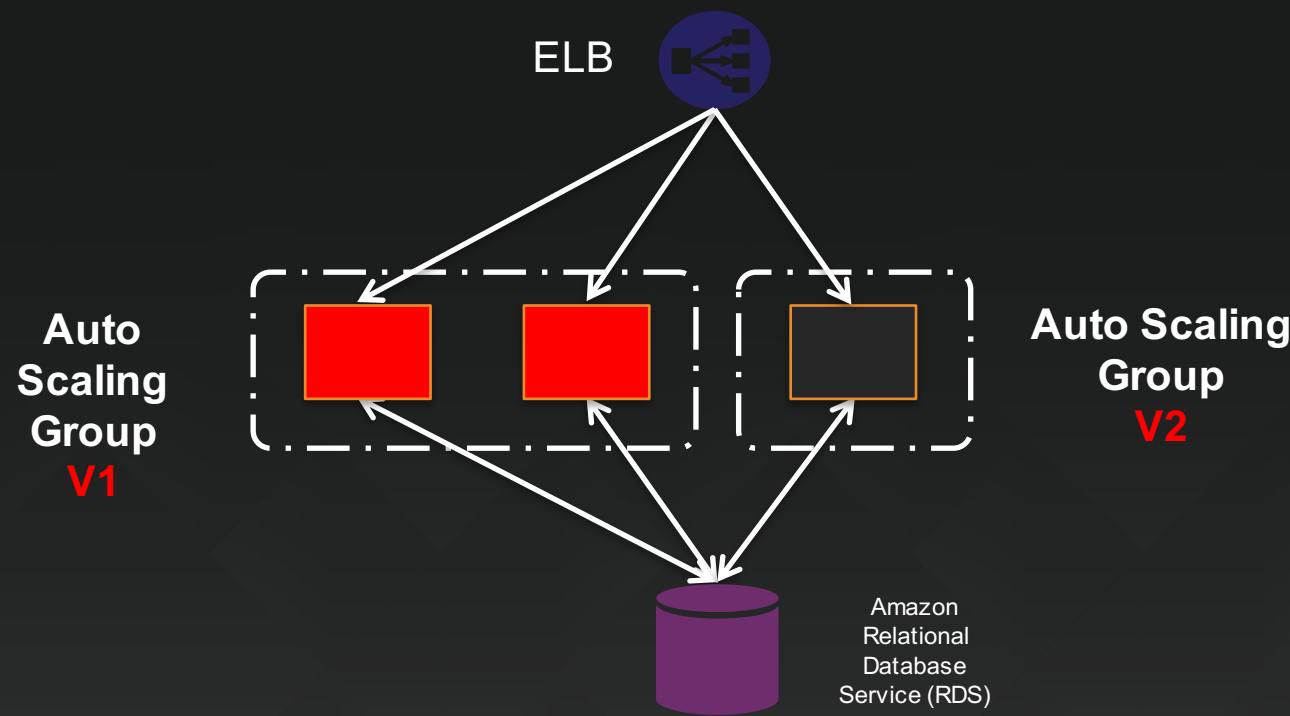
Red-Black Deployment



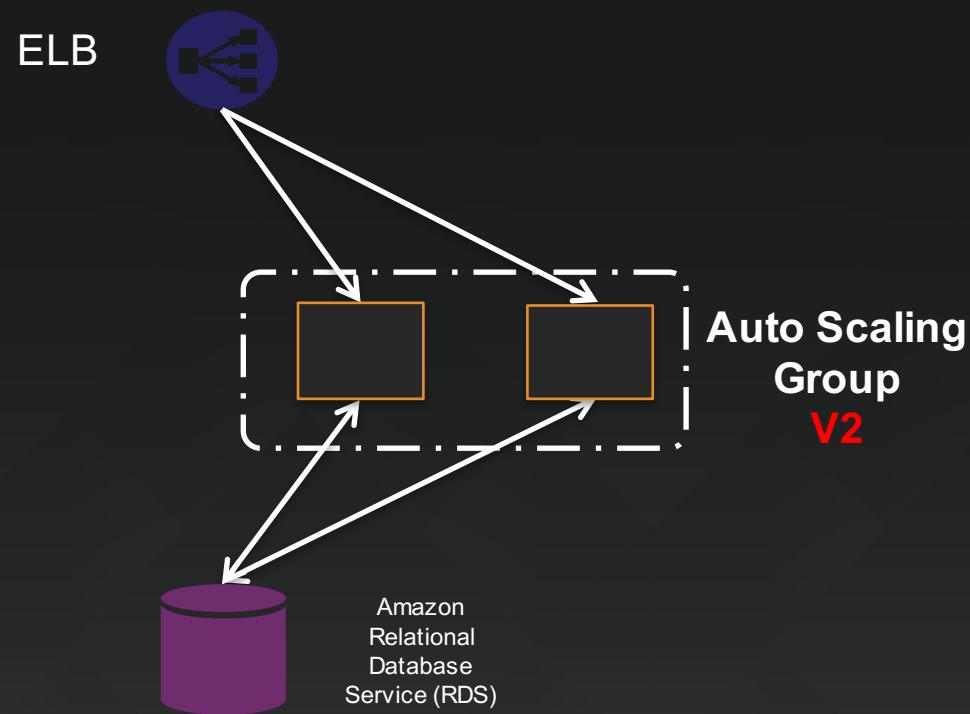
Red-Black Deployment



ALM | Red-Black Deployment



Red-Black Deployment



Monitoring & Metrics : The DevOps Pulse

- Track every resource possible
- Alert on *services*, *availability*, and *response times*
- Capture, learn, and improve
- Share access with everyone on the team
- Plan *metrics and monitoring* into the lifecycle

If a metric won't change how you behave, it's a

**bad
metric.**





McDonald

Where do I go from here?

- Collect Metrics. Graph anything that moves
- Log everything, Centralize logging, Log Analytics
- Infrastructure as Code
- Automated configuration management
- One click environment creation
- CI-CD pipelines
- Automated testing





Thank You

