



Infrastructure as Code: Manage your Architecture with Git

Danilo Poccia – AWS Technical Evangelist

 @danilop  danilop

Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.



Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.



GET LIST OF ALL
SALES MADE
LAST YEAR



ADD ALL SALES
TOGETHER



Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



Database

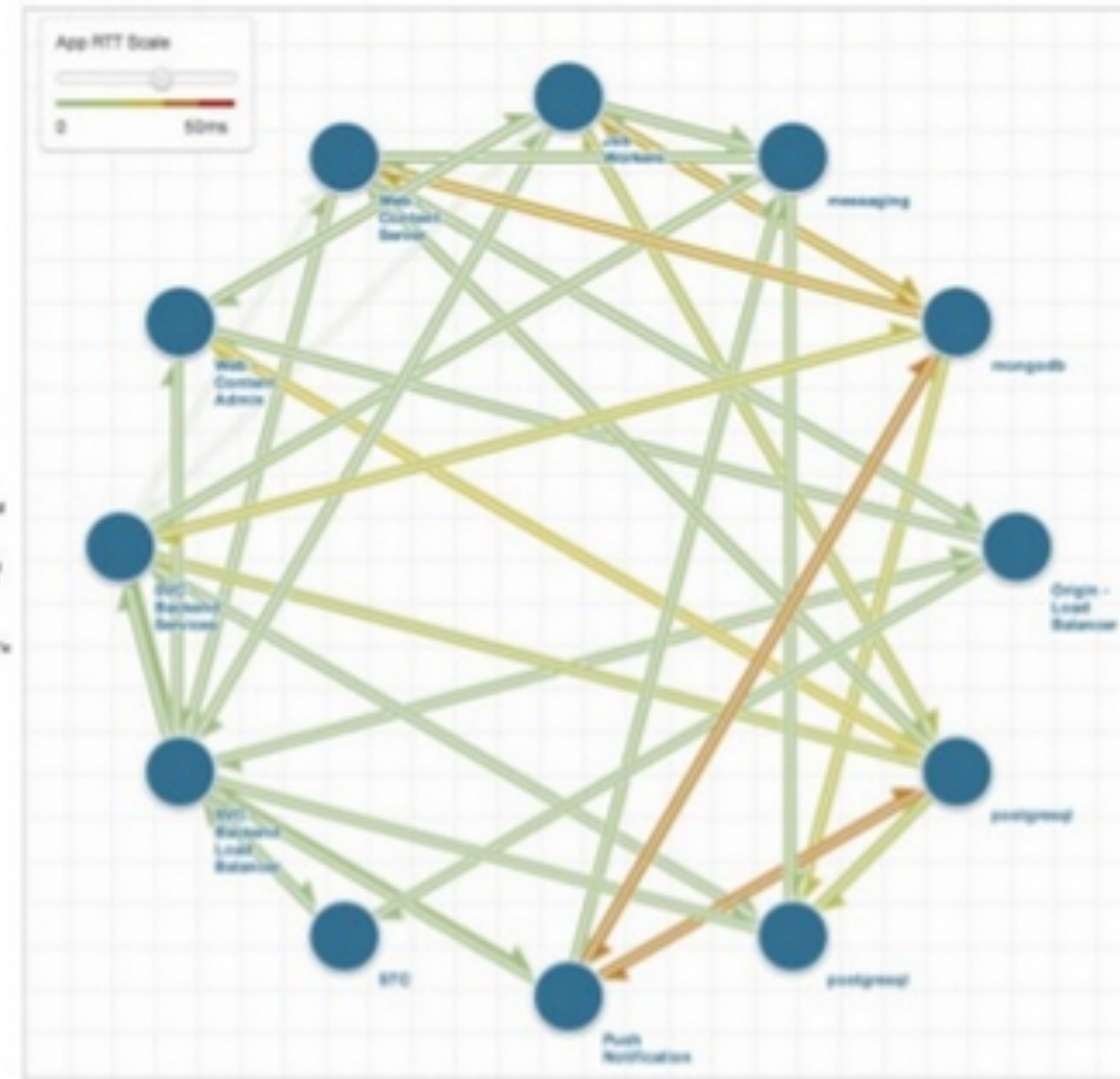


Storage

“Death Star” Architecture Diagrams



Netflix



Gilt Groupe (12 of 450)



Twitter

As visualized by Appdynamics, Boundary.com and Twitter internal tools

Adrian Cockcroft, Technology Fellow at Battery Ventures
<http://www.slideshare.net/adriancockcroft/goto-berlin>

**“A single website may now handle
as much traffic as the entire Internet
did less than a decade ago.”**

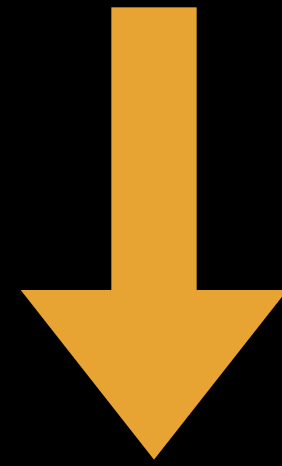
What is Reactive Programming?
Kevin Webber

Infrastructure as **Code**

Writing Code to
Manage Configurations
and **Automate Provisioning**
of Infrastructure

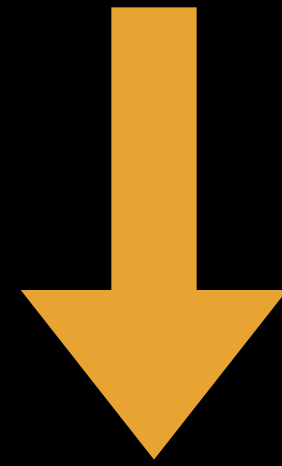
Manage IT Infrastructure using
Tools and Practices
from Software Development

Infrastructure as Code



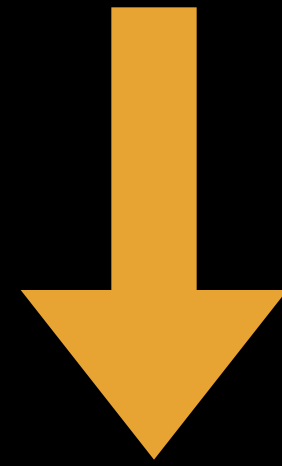
Version Control

Infrastructure as Code



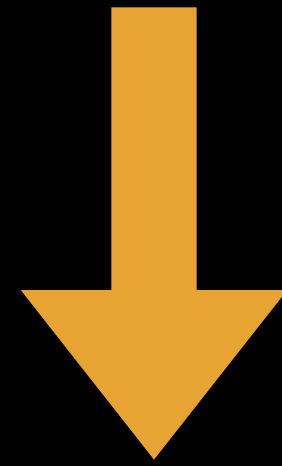
Rollback

Infrastructure as Code



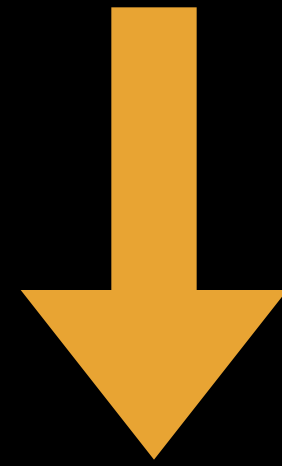
Testing

Infrastructure as Code

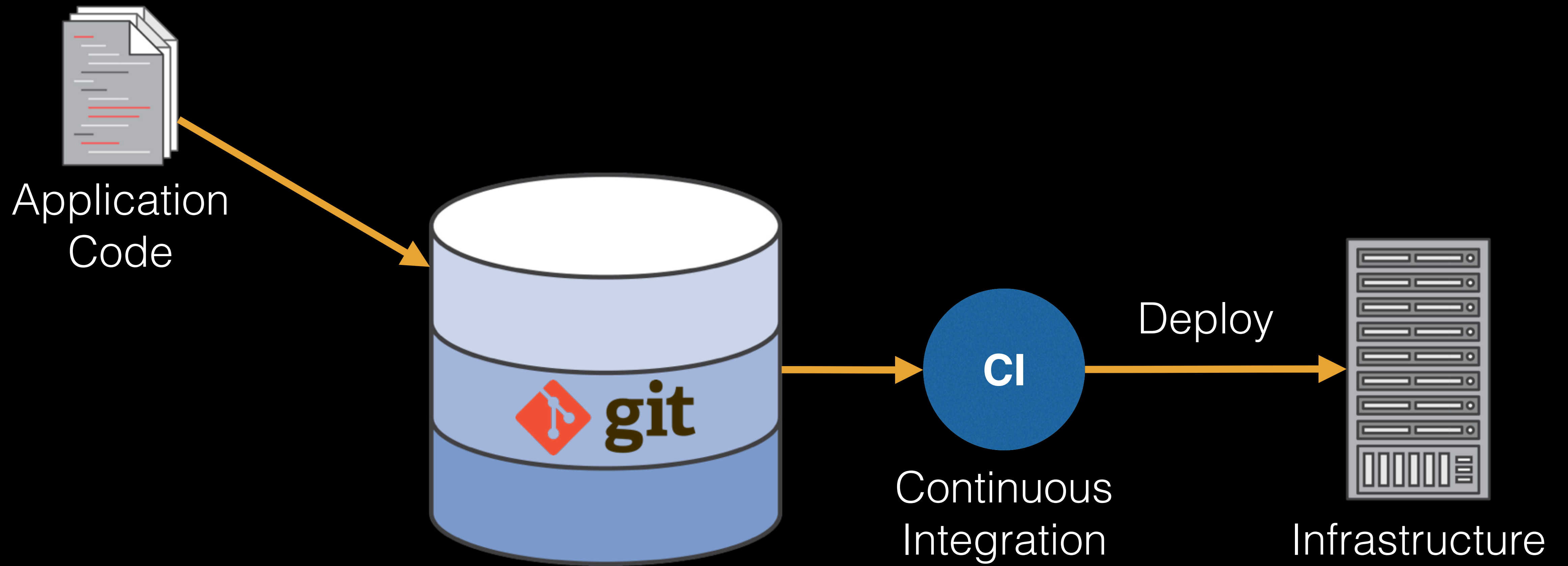


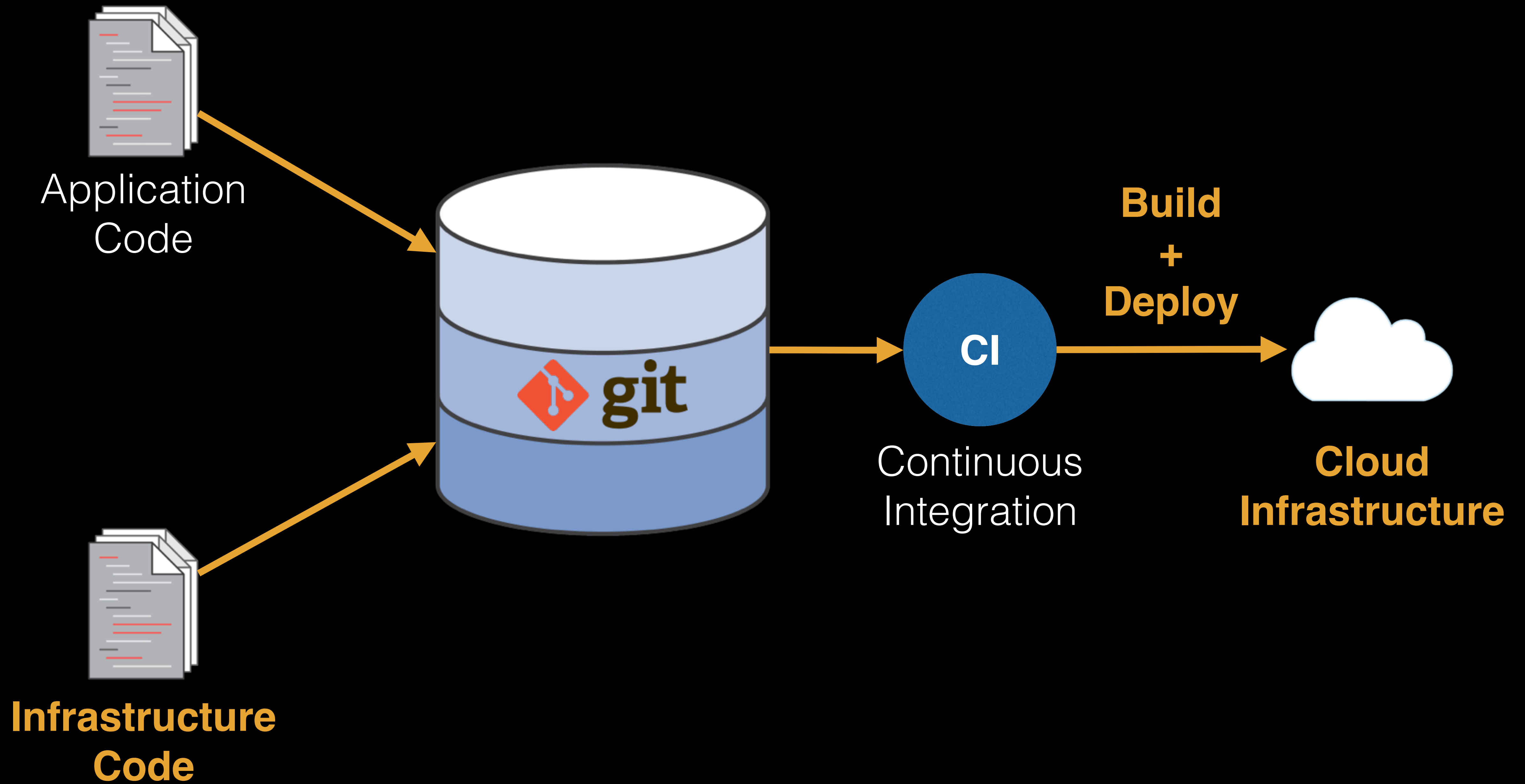
Small Deployments

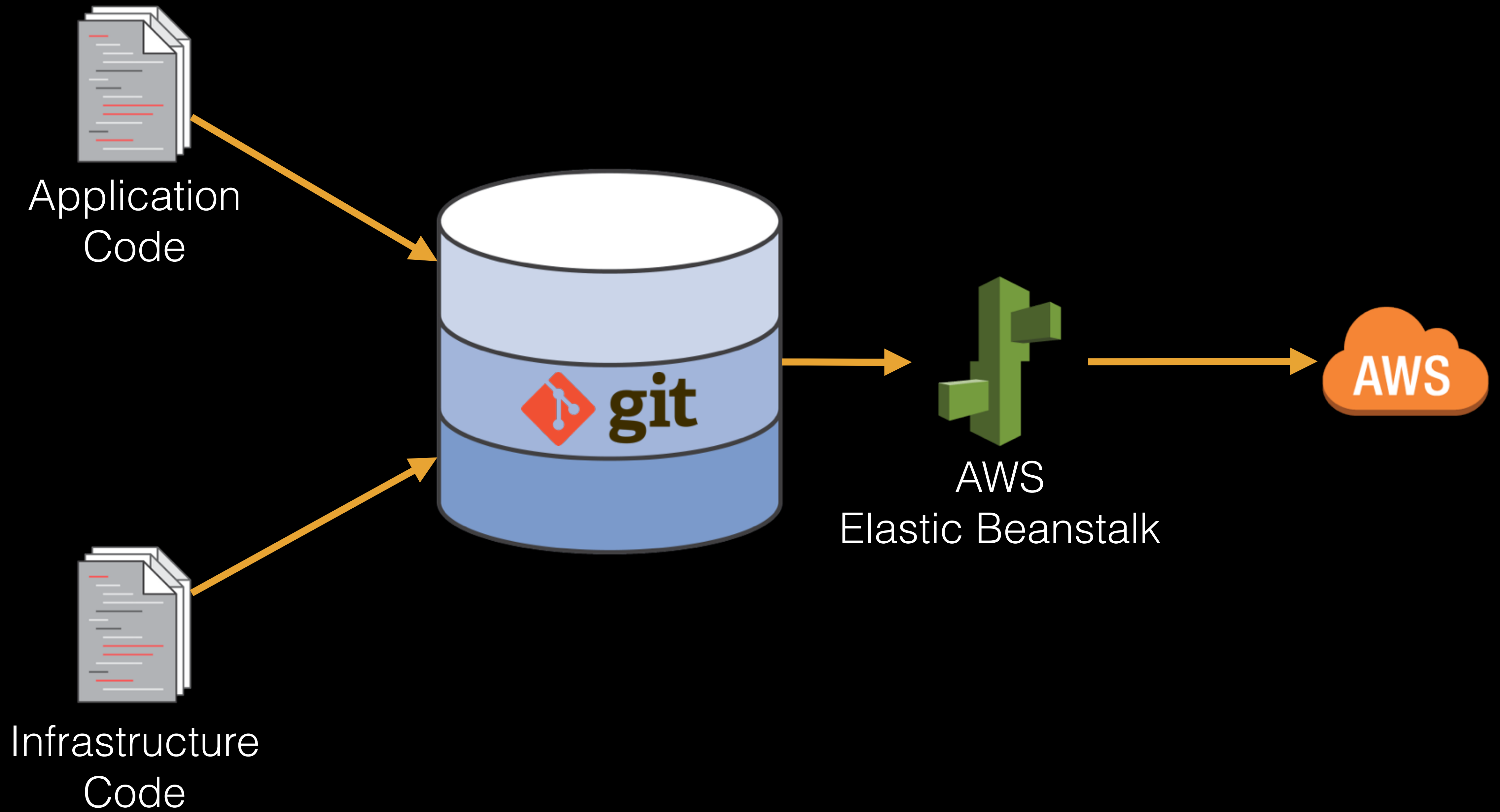
Infrastructure as Code

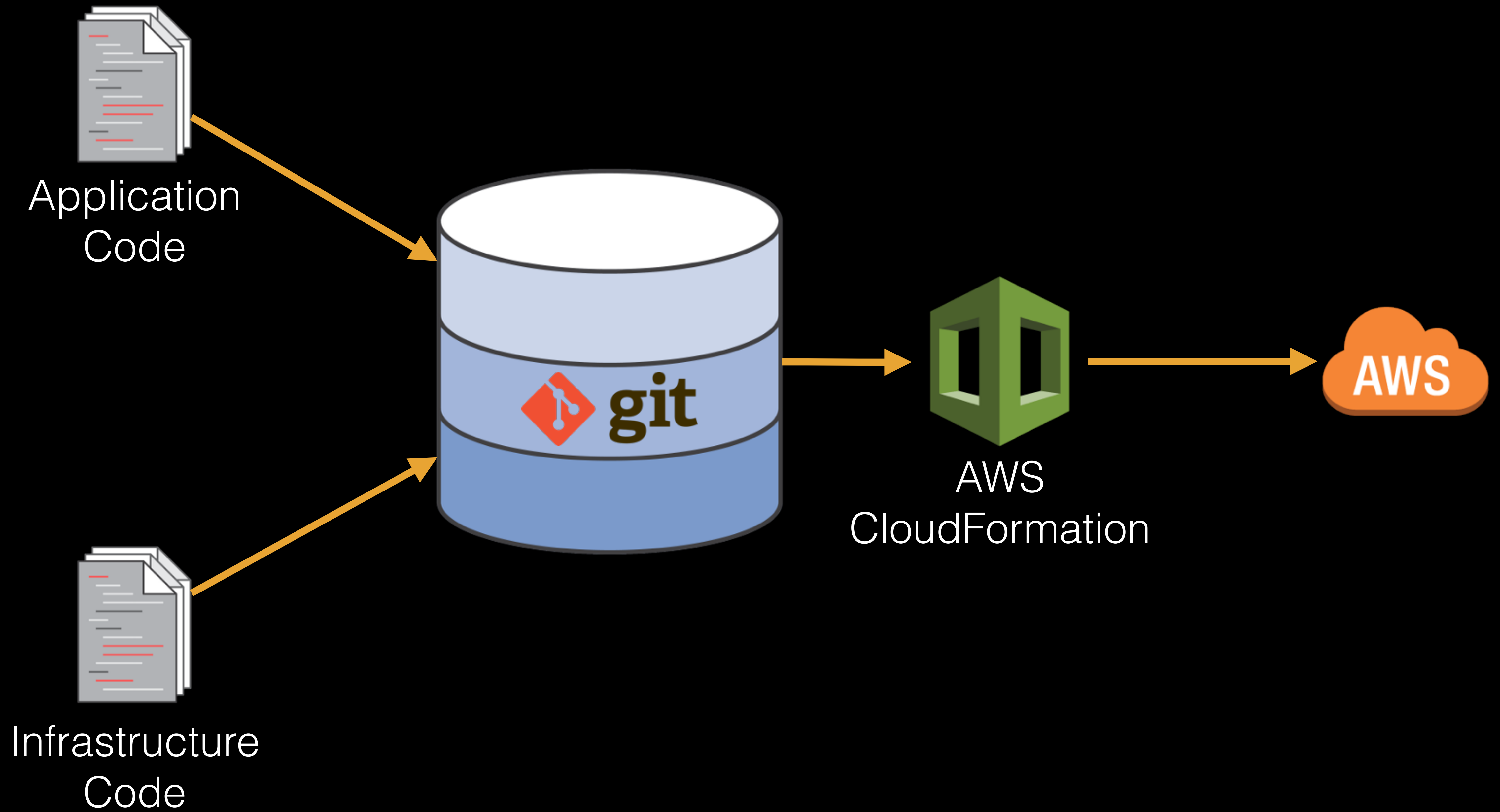


Design Patterns









AWS Elastic Beastalk

**An Easy-to-Use Service
for Deploying and Scaling
Web Applications and Services**

Java

.NET

Node.js

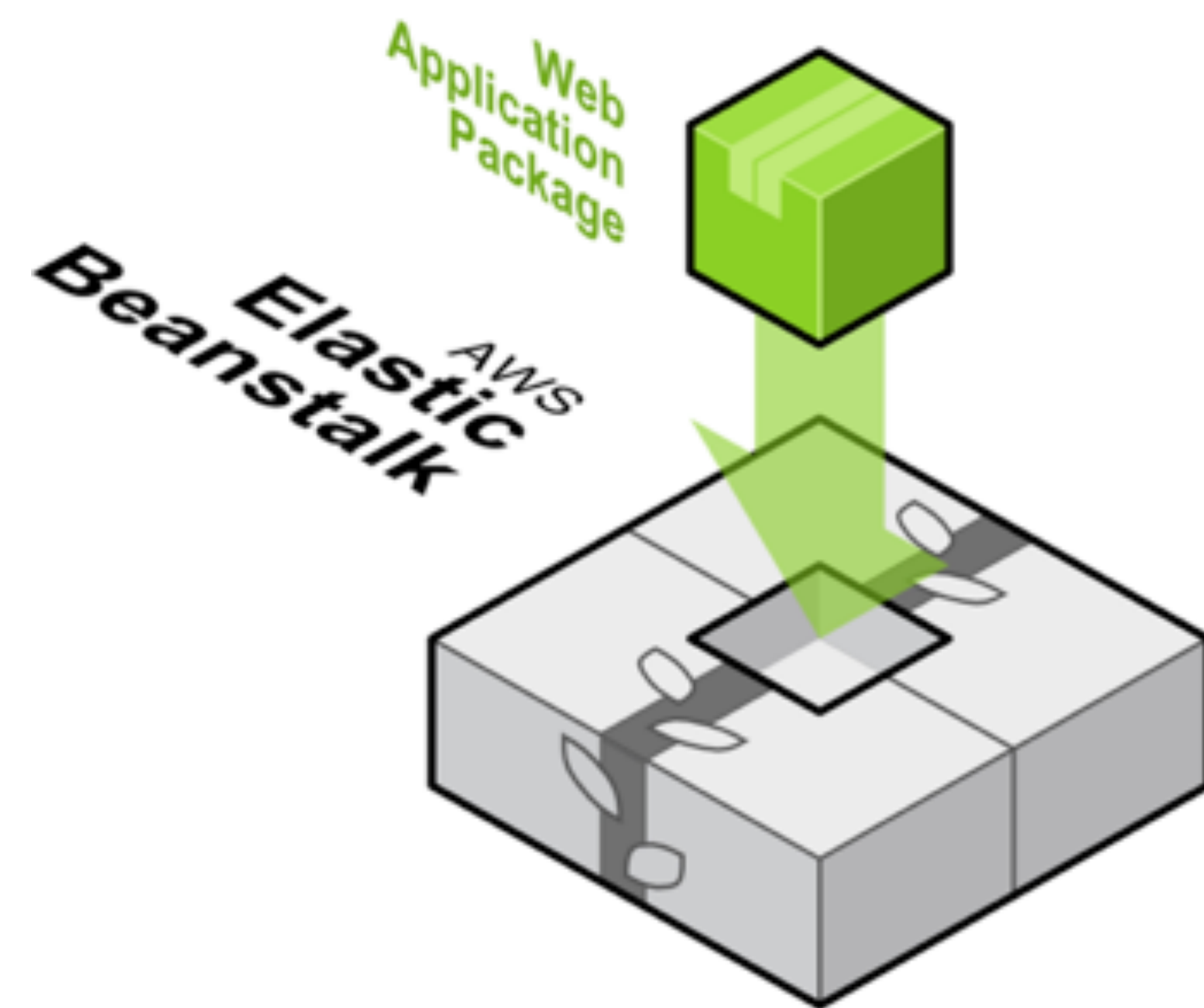
PHP

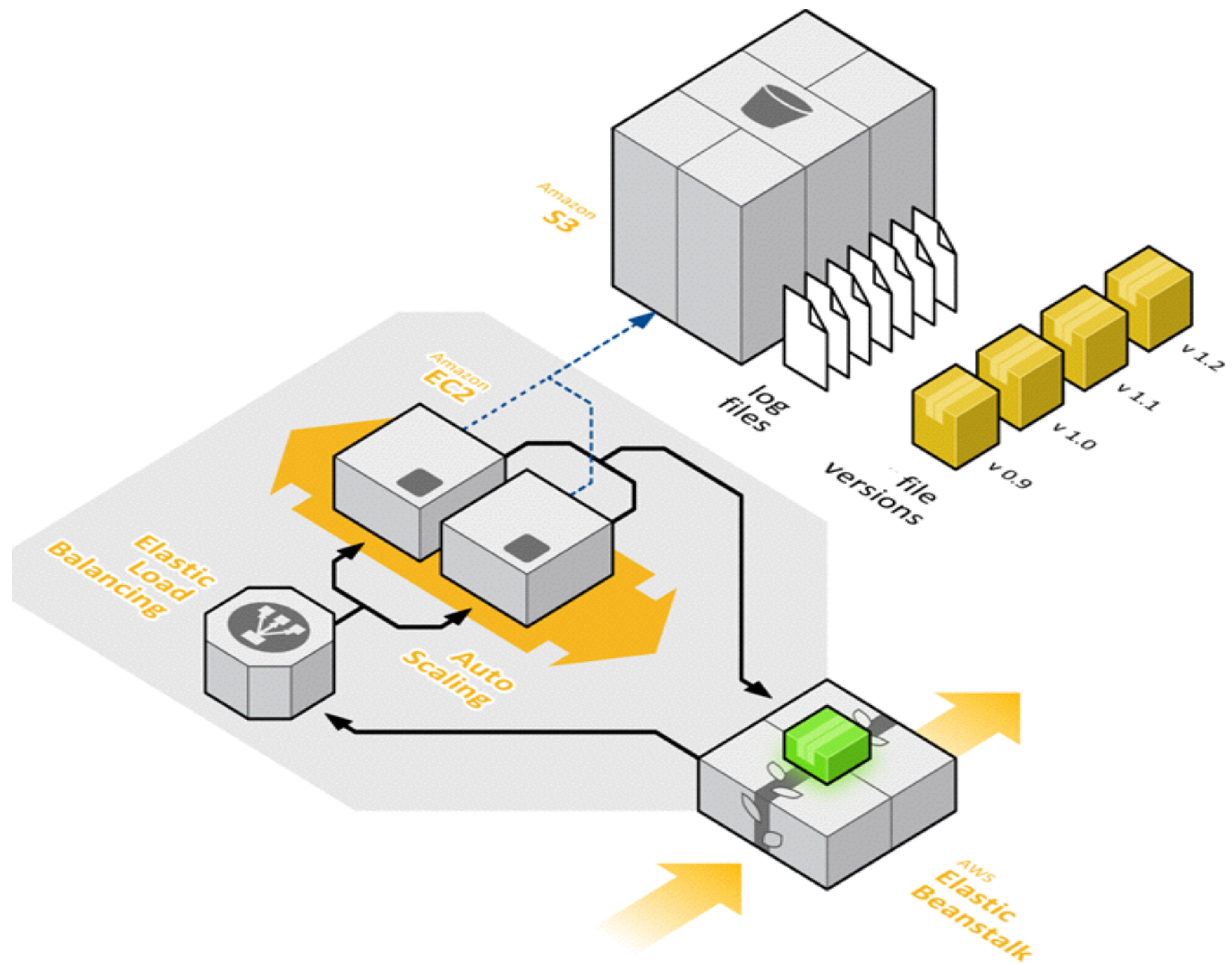
Python

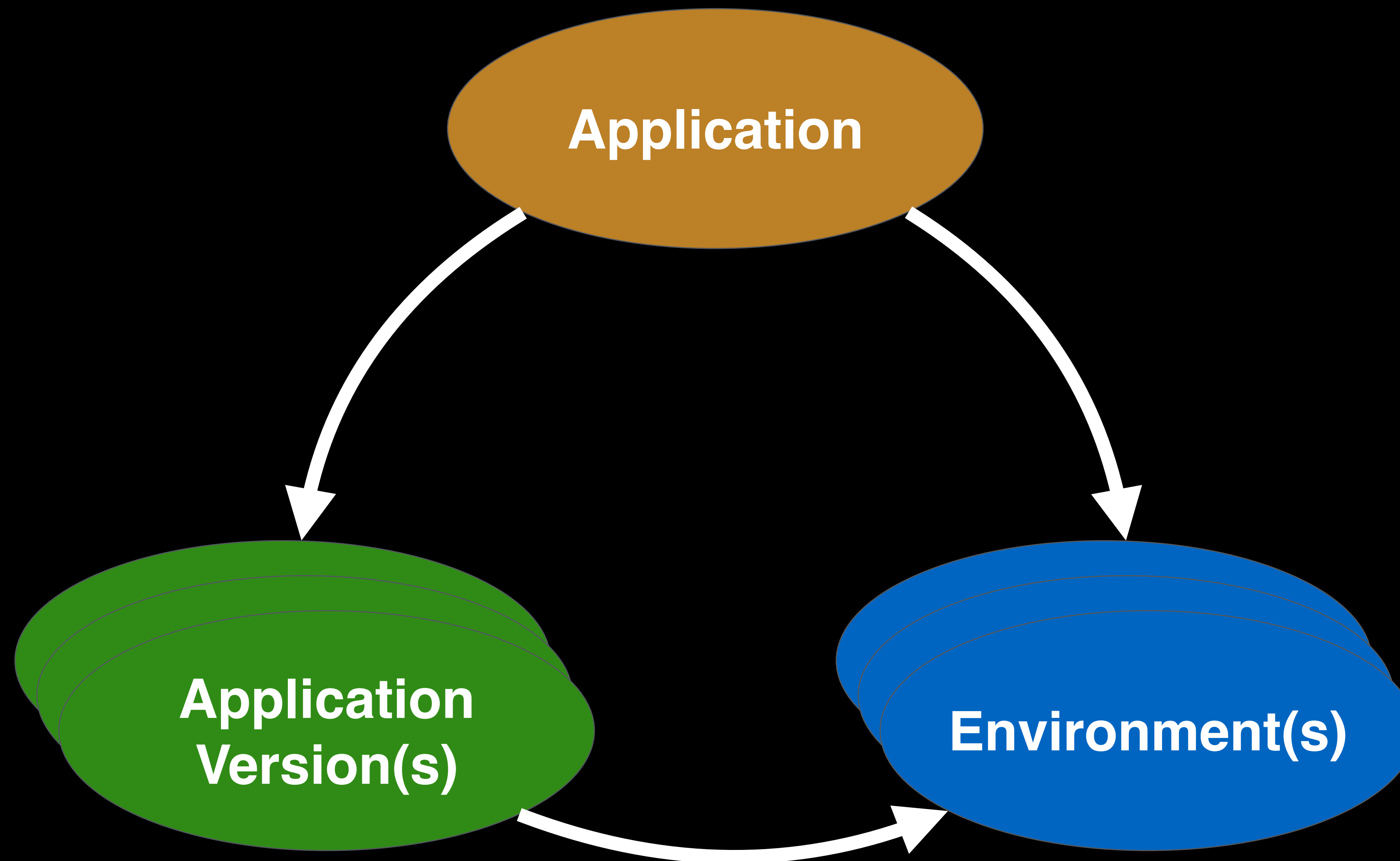
Ruby

Go

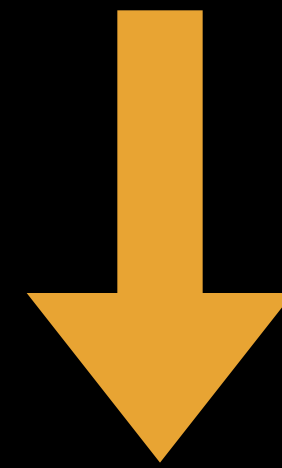
Docker





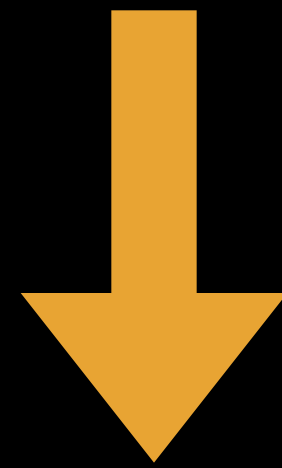


git branch



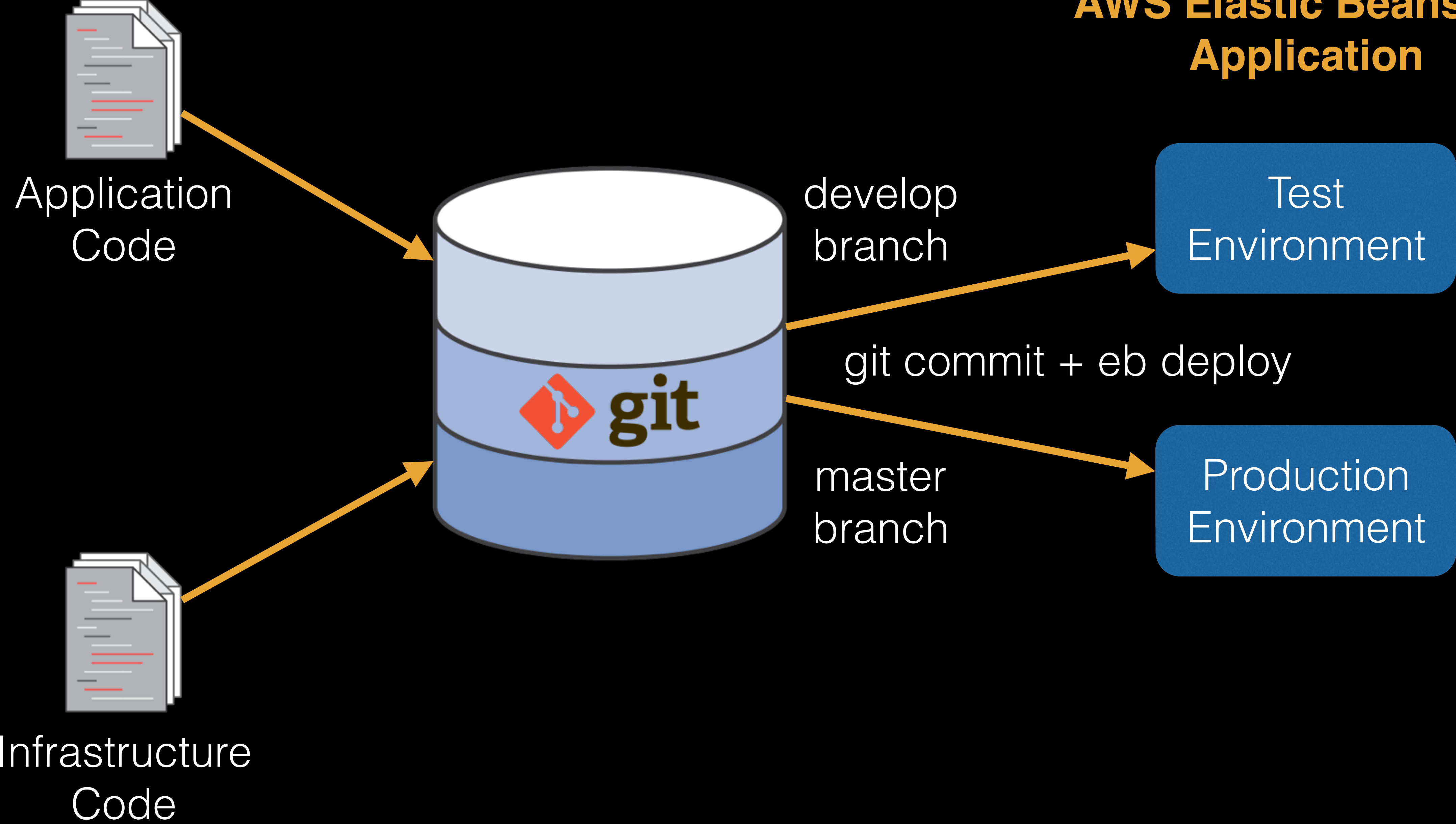
Environment

git commit



**Application
Version**

**AWS Elastic Beanstalk
Application**



<demo>

...

</demo>

AWS re:Invent 2014 | (APP315) Coca-Cola: Migrating to AWS

AWS re:Invent

AWS re:Invent

APP315

Coca-Cola: Migrating to AWS

Michael Connor, Coca-Cola

Nov 13, 2014 | Las Vegas, NV

amazon web services

© 2014 Amazon.com, Inc. and its affiliates. All rights reserved. May not be copied, modified, or distributed in whole or in part without the express consent of Amazon.com, Inc.

amazon web services

© 2014 Amazon.com, Inc. and its affiliates. All rights reserved. May not be copied, modified, or distributed in whole or in part without the express consent of Amazon.com, Inc.

0:00 / 41:42

HD

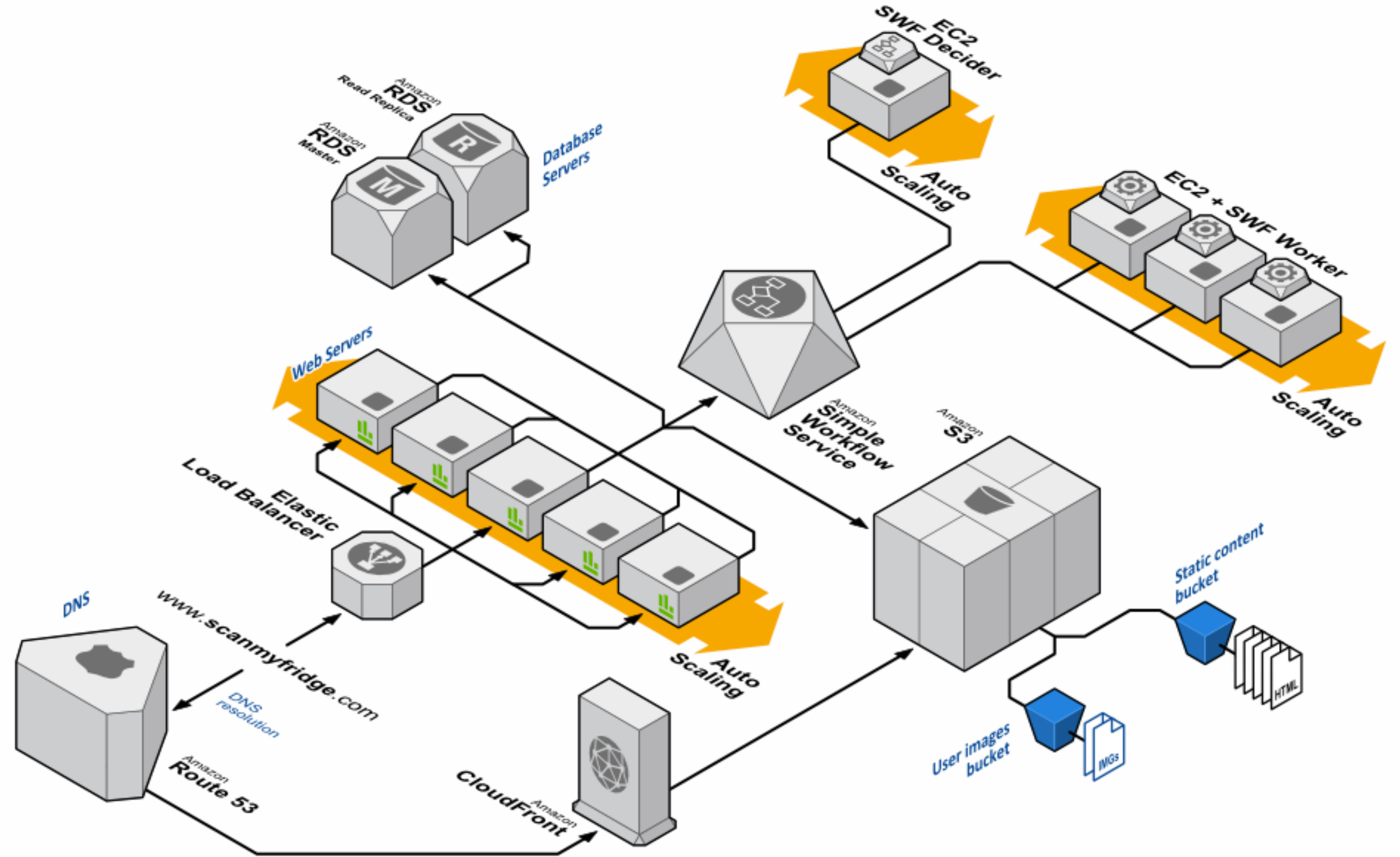
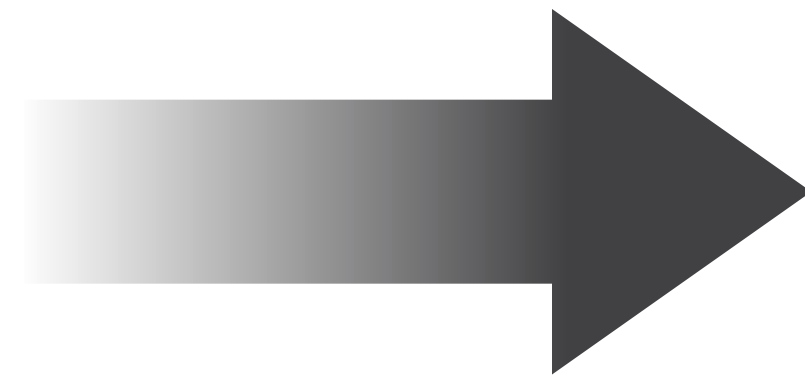
https://youtu.be/3lqz_YFXLF0

AWS CloudFormation

Create and Manage a Collection of Related AWS Resources



CloudFormation Template



CloudFormation

Template

JSON Syntax

Parameters

Mappings

Resources

Outputs

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Infrastructure as Code - CloudFormation Demo",
  "Parameters": {
    "InstanceType": {
      "Type": "String",
      "AllowedValues": [
        "t2.micro", "t2.small", "t2.medium", "m3.medium", "m3.large", "m3.xlarge", "m3.2xlarge", "c3.large", "c3.xlarge",
        "c3.2xlarge", "c3.4xlarge", "c3.8xlarge", "r3.large", "r3.xlarge", "r3.2xlarge", "r3.4xlarge", "r3.8xlarge",
        "i2.xlarge", "i2.2xlarge", "i2.4xlarge", "i2.8xlarge"
      ],
      "ConstraintDescription": "must be a valid EC2 instance type.",
      "Default": "t2.micro",
      "Description": "WebServer EC2 instance type"
    },
    "KeyName": {
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair.",
      "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Default": "danilop-keypair-eu-central-1"
    },
    "SSHLocation": {
      "Type": "String",
      "AllowedPattern": "^(\\d{1,3})\\.?(\\d{1,3})\\.?(\\d{1,3})\\.?(\\d{1,3})/?(\\d{1,2})$",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x.",
      "Default": "0.0.0.0/0",
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "MaxLength": "18",
      "MinLength": "9"
    }
  },
  "Resources": {
    "ElasticLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties": {
        "AvailabilityZones": { "Fn::GetAZs": "" },
        "CrossZone": "true",
        "HealthCheck": { "HealthyThreshold": "2", "Interval": "10", "Target": "HTTP:80/", "Timeout": "5", "UnhealthyThreshold": "3" },
        "Listeners": [ { "InstancePort": "80", "LoadBalancerPort": "80", "Protocol": "HTTP" } ],
        "Tags": [ { "Key" : "Name", "Value" : "CFDemo" } ]
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access and HTTP access on the inbound port",
        "SecurityGroupIngress": [
          {
            "FromPort": "80",
            "IpProtocol": "tcp",
            "SourceSecurityGroupName": { "Fn::GetAtt": [ "ElasticLoadBalancer", "SourceSecurityGroup.GroupName" ] },
            "SourceSecurityGroupOwnerId": { "Fn::GetAtt": [ "ElasticLoadBalancer", "SourceSecurityGroup.OwnerAlias" ] },
            "ToPort": "80"
          },
          {
            "CidrIp": { "Ref": "SSHLocation" }, "FromPort": "22", "IpProtocol": "tcp", "ToPort": "22"
          }
        ]
      }
    },
    "LaunchConfig": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Properties": {
        "ImageId": "ami-a88bb6b5",
        "InstanceType": { "Ref": "InstanceType" },
        "KeyName": { "Ref": "KeyName" },
        "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ]
      }
    },
    "WebServerGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
        "AvailabilityZones": { "Fn::GetAZs": "" },
        "LaunchConfigurationName": { "Ref": "LaunchConfig" },
        "LoadBalancerNames": [ { "Ref": "ElasticLoadBalancer" } ],
        "MinSize": "1",
        "MaxSize": "4",
        "DesiredCapacity": "1",
        "Tags": [ { "Key" : "Name", "Value" : "CFDemo", "PropagateAtLaunch" : true } ]
      }
    }
  },
  "Outputs": {
    "URL": {
      "Description": "URL of the website",
      "Value": { "Fn::Join": [ "", [ "http://", { "Fn::GetAtt": [ "ElasticLoadBalancer", "DNSName" ] } ] ] }
    }
  }
}
```

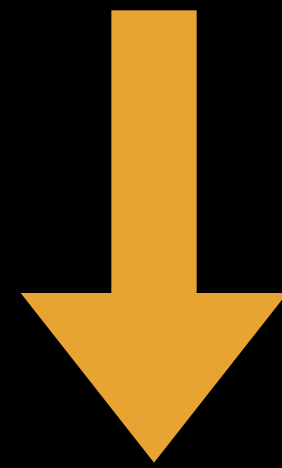
```
{
  "Description" : "Create RDS with username and password",
  "Resources" : {

    "MyDB" : {
      "Type" : "AWS::RDS::DBInstance",
      "Properties" : {
        "AllocatedStorage" : "500",
        "DBInstanceClass" : "db.t2.micro",
        "Engine" : "MySQL",
        "EngineVersion" : "5.6",
        "MasterUsername" : "MyName",
        "MasterUserPassword" : "MyPassword"
      }
    }
  }
}
```



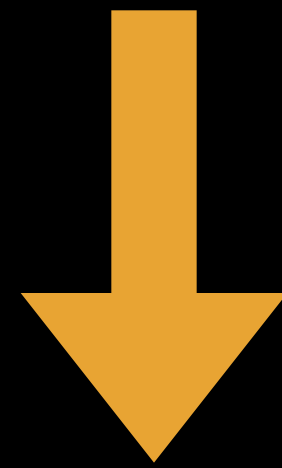
```
"AWS::CloudFormation::Init" : { "config" : {  
  
    "packages" : {  
        "yum" : {  
            "mysql"           : [],  
            "mysql-server"   : [],  
            "httpd"          : [],  
            "php"            : [],  
            "php-mysql"      : []  
        },  
    },  
  
    "sources" : {  
        "/var/www/html" :  
            "https://my-builds.s3.amazonaws.com/build-v4.zip"  
    }  
}}
```

Template + Parameters



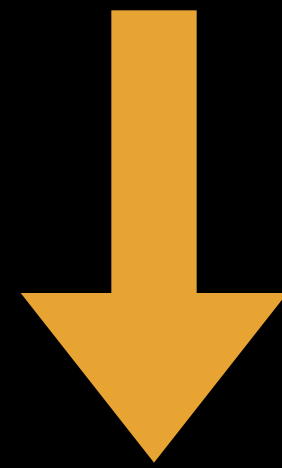
Stack

Template



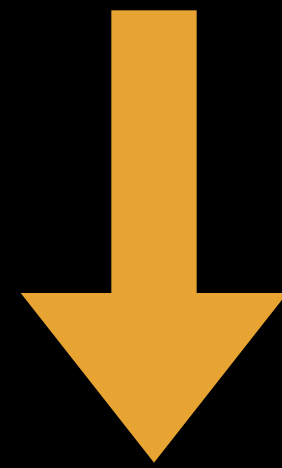
Same for Multiple Stacks
(different environments)

Parameters



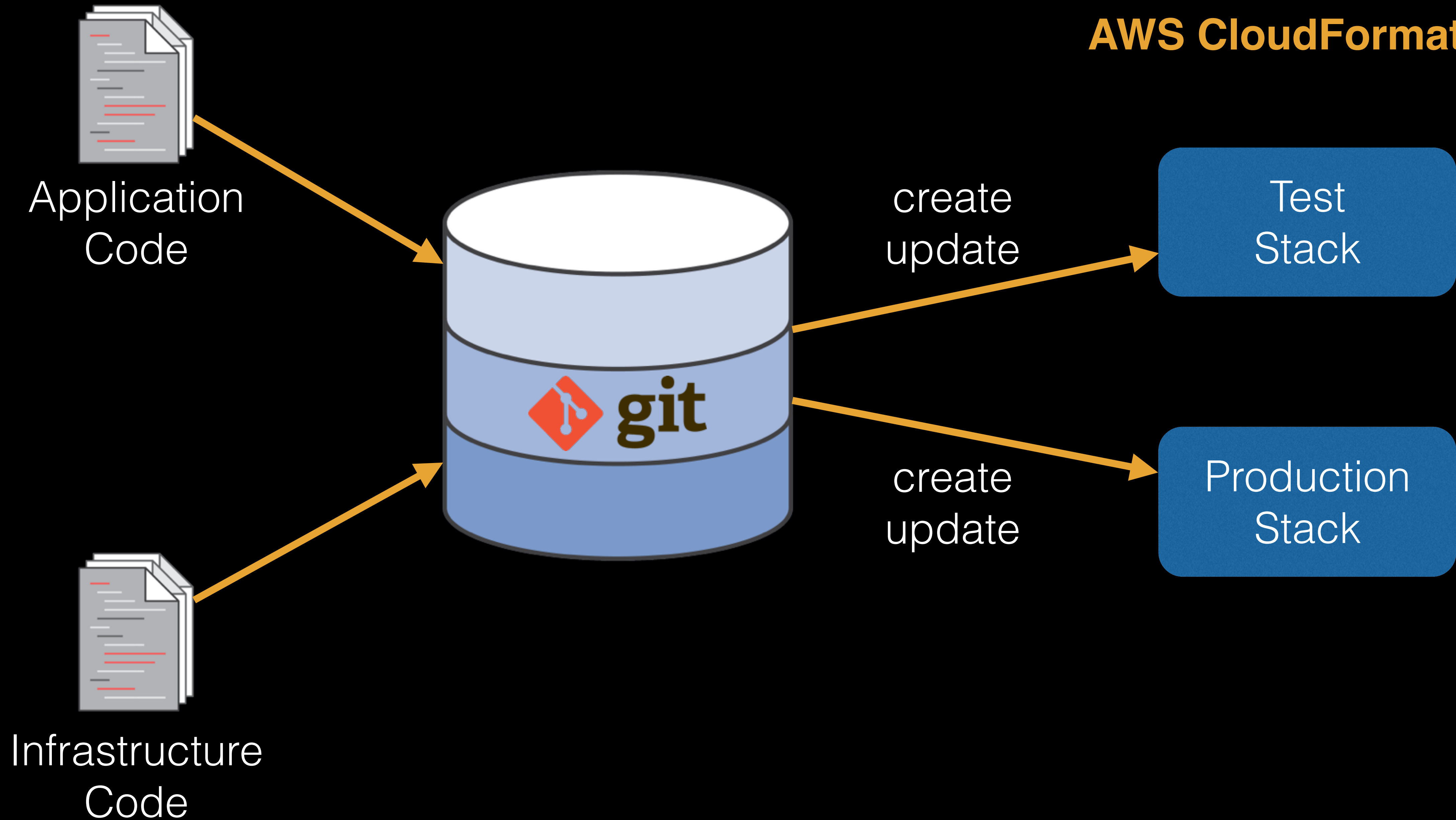
Specific for a Stack
(configuration management)

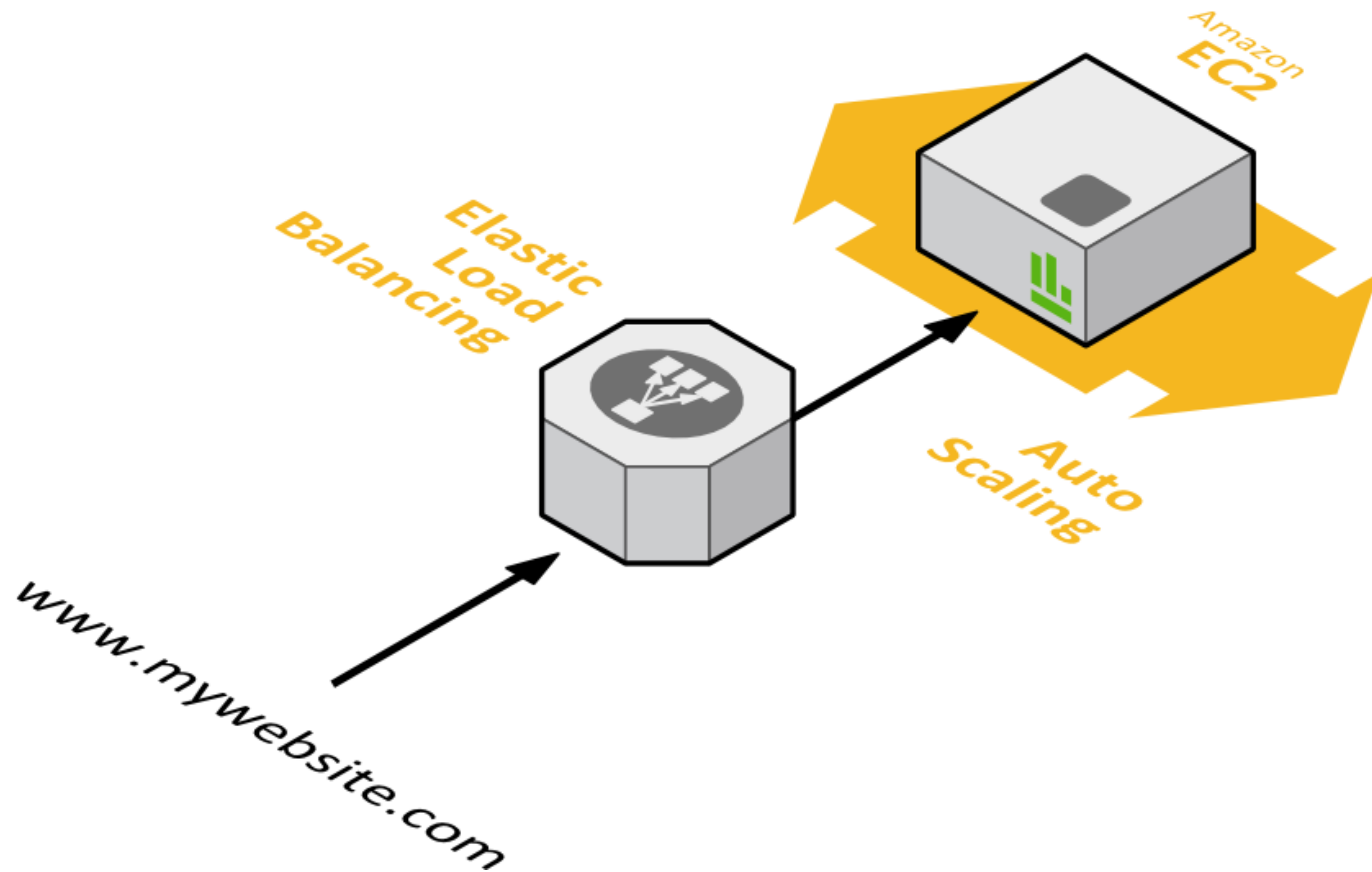
CloudFormer

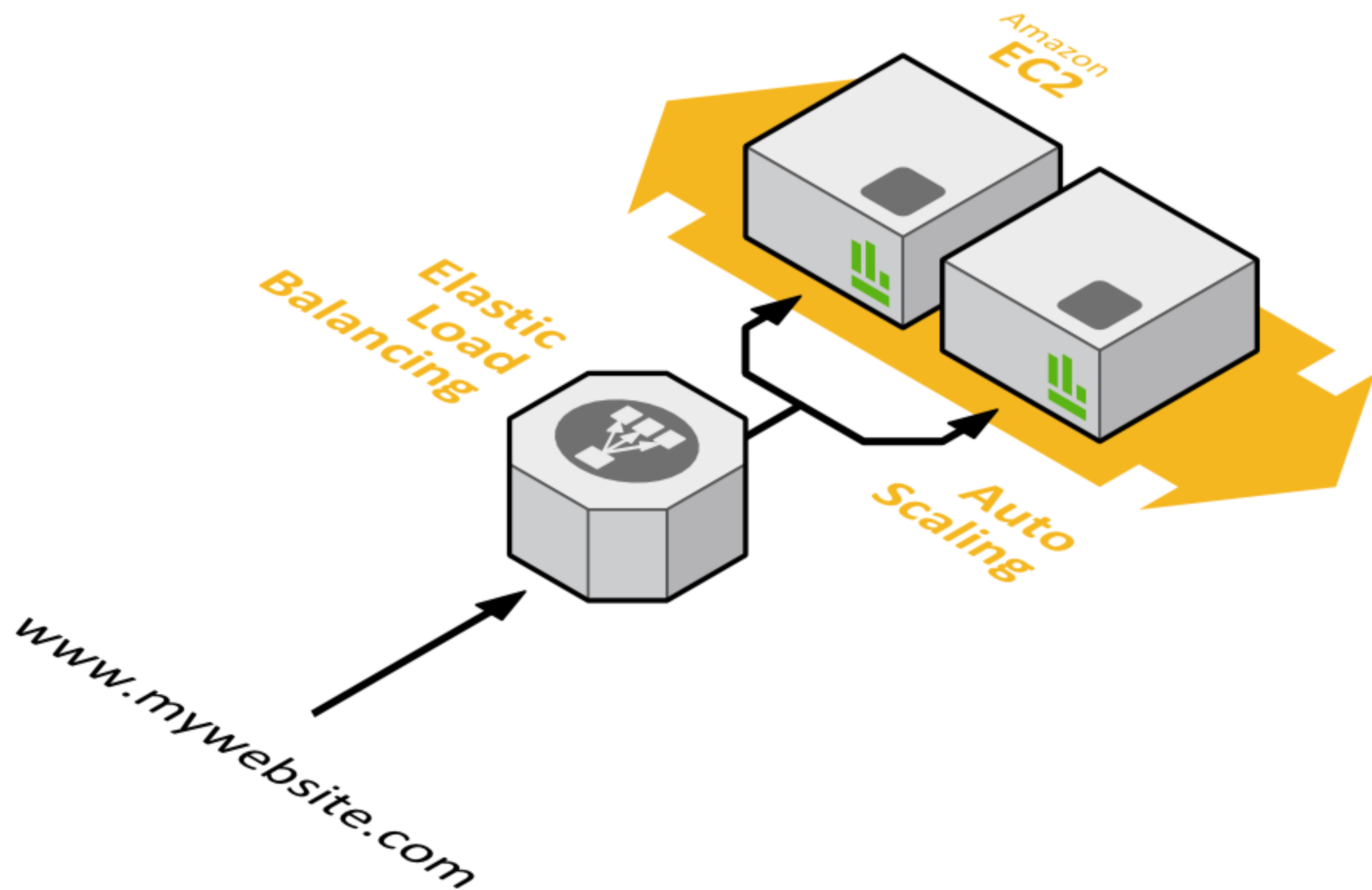


Template Creation Tool
(from existing AWS resources)

AWS CloudFormation







<demo>

...

</demo>

Who is using Amazon CloudFormation?



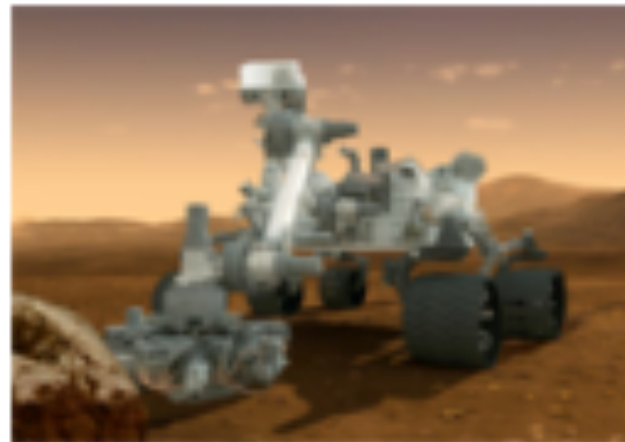
DOW JONES



Adobe



Mentor
Graphics



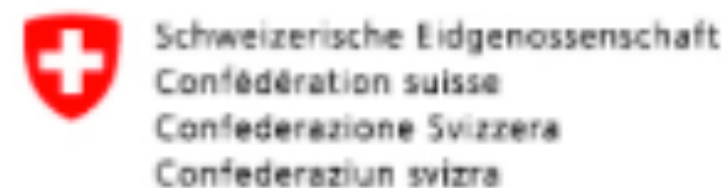
SEGA



me Bank
BANK FAIRER.



Expedia



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra



tracelink
Life Sciences Cloud



icbi
at Georgetown University



Singapore
POST

<http://aws.amazon.com/solutions/case-studies/>

AWS re:Invent 2014 | (APP306) Using AWS CloudFormation for Deployment and Management at Scale



AWS re:Invent

APP306: Using AWS CloudFormation for Deployment and Management at Scale

Tom Cartwright and Yavor Atanasov, BBC

November 12, 2014 Las Vegas, Nevada



© 2014 Amazon.com, Inc. and its affiliates. All rights reserved. May not be copied, modified, or distributed in whole or in part without the express consent of Amazon.com, Inc.



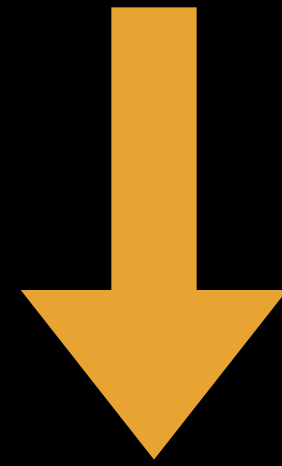
© 2014 Amazon.com, Inc. and its affiliates. All rights reserved. May not be copied, modified, or distributed in whole or in part without the express consent of Amazon.com, Inc.

0:00 / 38:26

HD

<https://youtu.be/Wk-tOPicq78>

Application + Infrastructure



Data + Code

Configuration Management

Application
+
Infrastructure

Configuration Management

**Use a different repository
+ git submodule add ?**

Any (better) idea
from the Git Community?



 @danilop

 danilop