

# 基于AWS的DevOps实践指南

代闻  
AWS解决方案架构师



WIKIPEDIA  
The Free Encyclopedia

# 什么是DevOps?

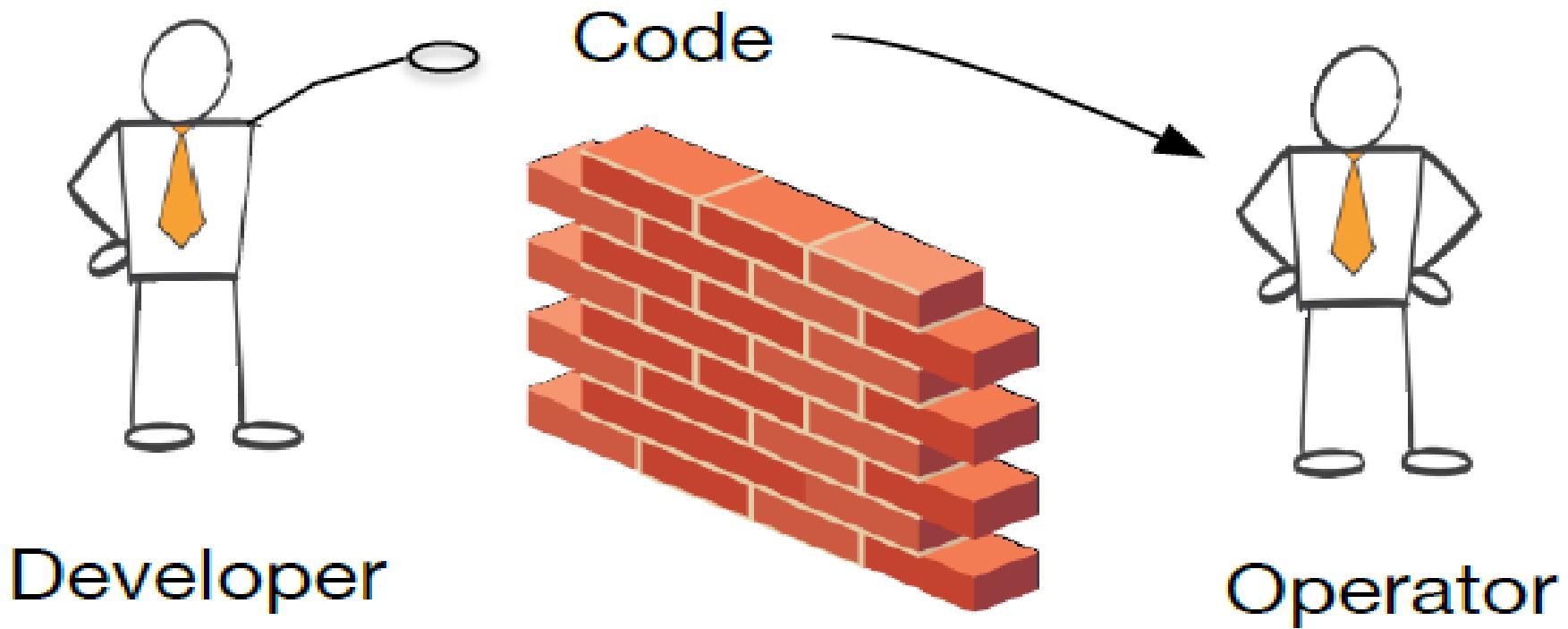
“

*DevOps is a software development method that stresses communication, collaboration, integration, automation, and measurement of cooperation between software developers and other IT professionals ... to improve operations performance.*

”

# 为什么需要DevOps?

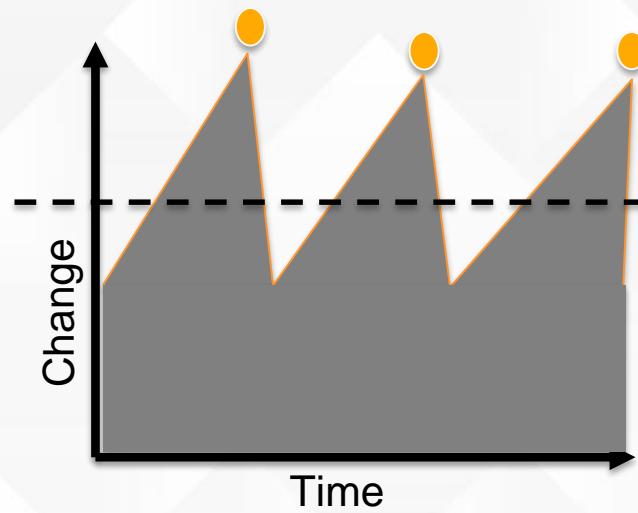
因为我们不希望事情是这样的...



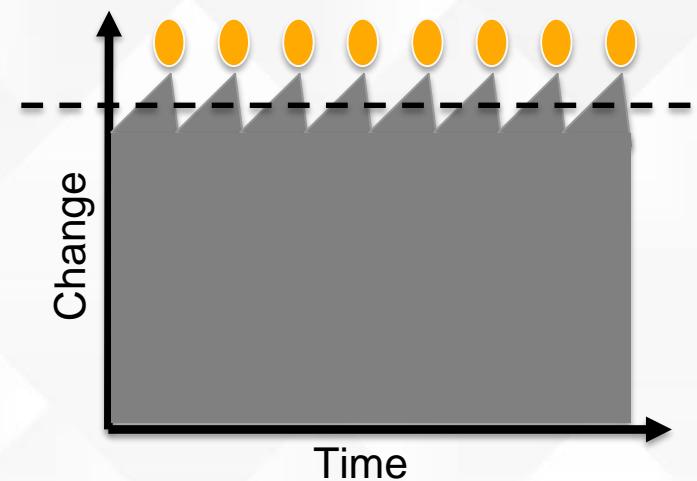
# 为什么需要DevOps?

我们希望...

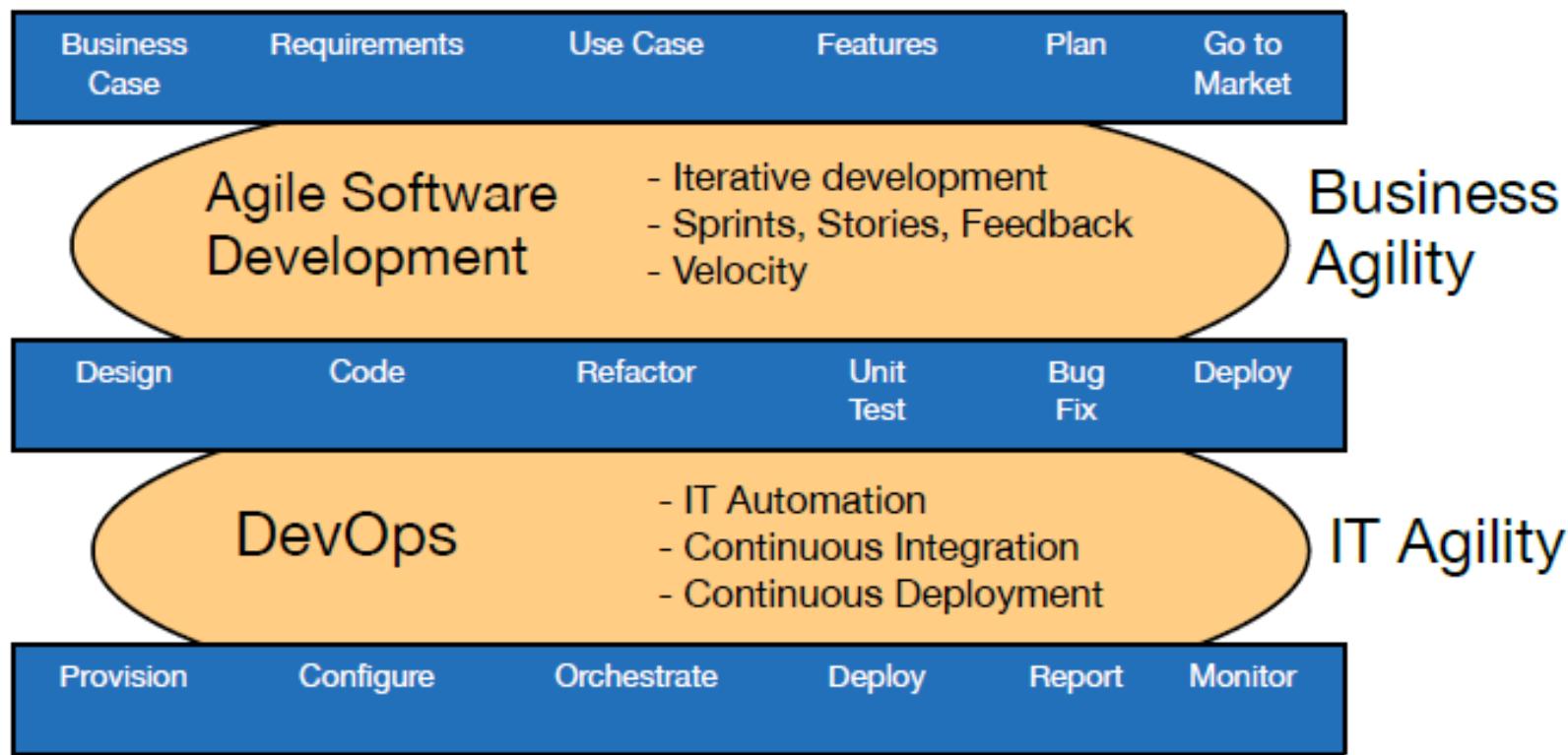
瀑布式开发，版本发布少



快速迭代, 敏捷开发

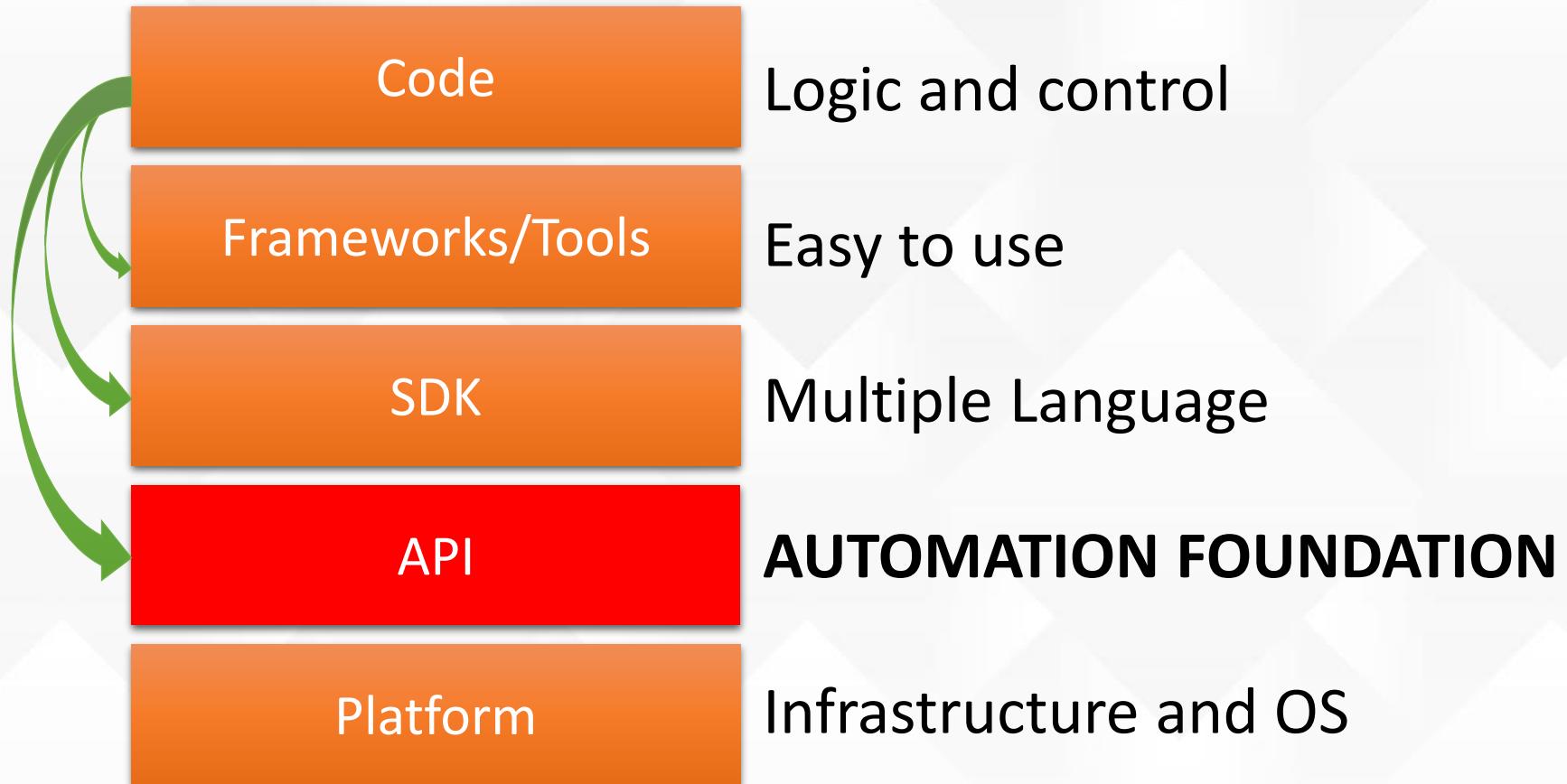


# 为什么需要DevOps?

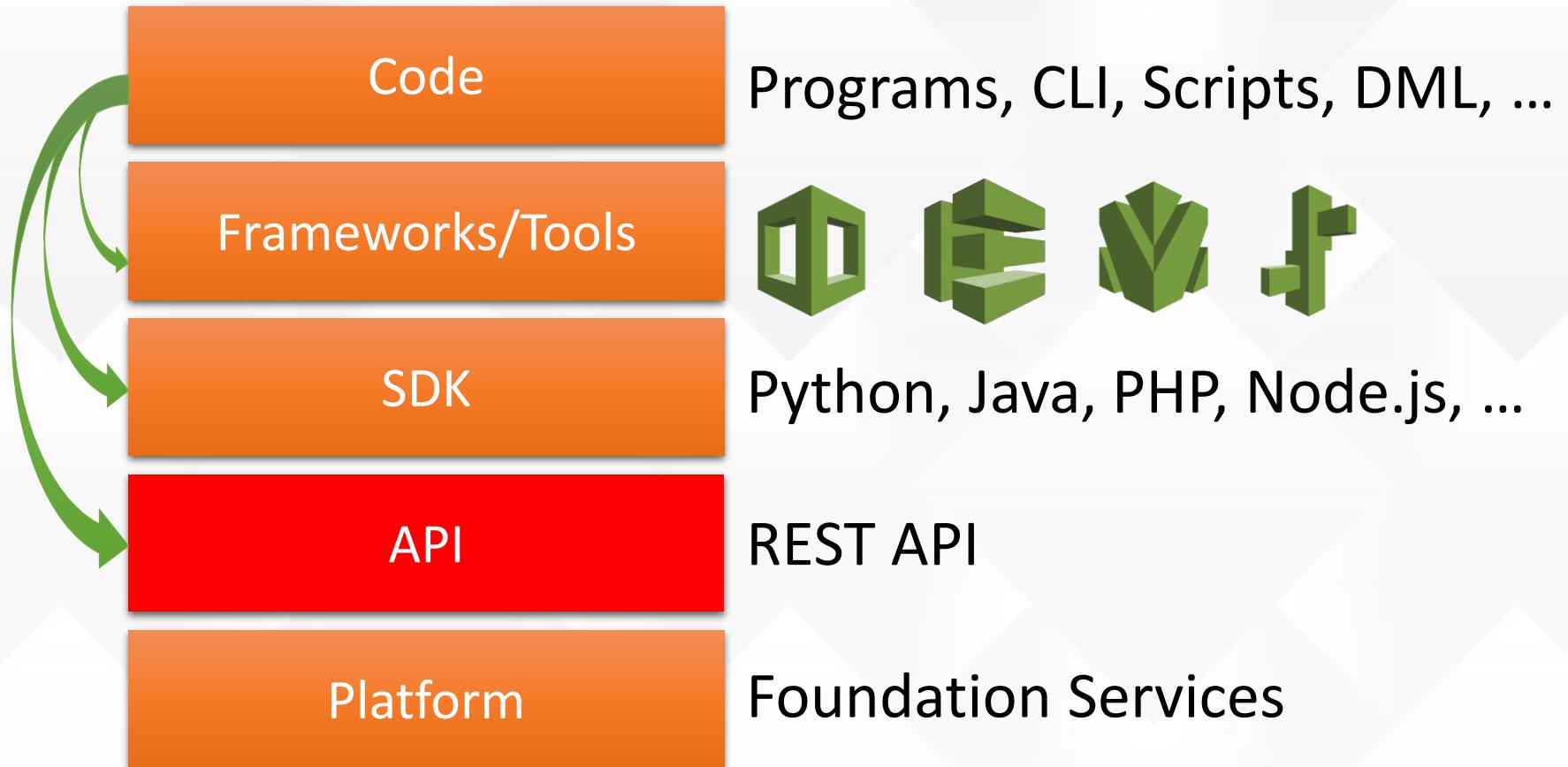


# 怎么实现DevOps?

# 从软件开发的角度看DevOps



# AWS对DevOps的全面支持



# 基于AWS的DevOps实践要素

代码和命令行



CloudFormation



CodeDeploy



OpsWorks



ElasticBeanstalk



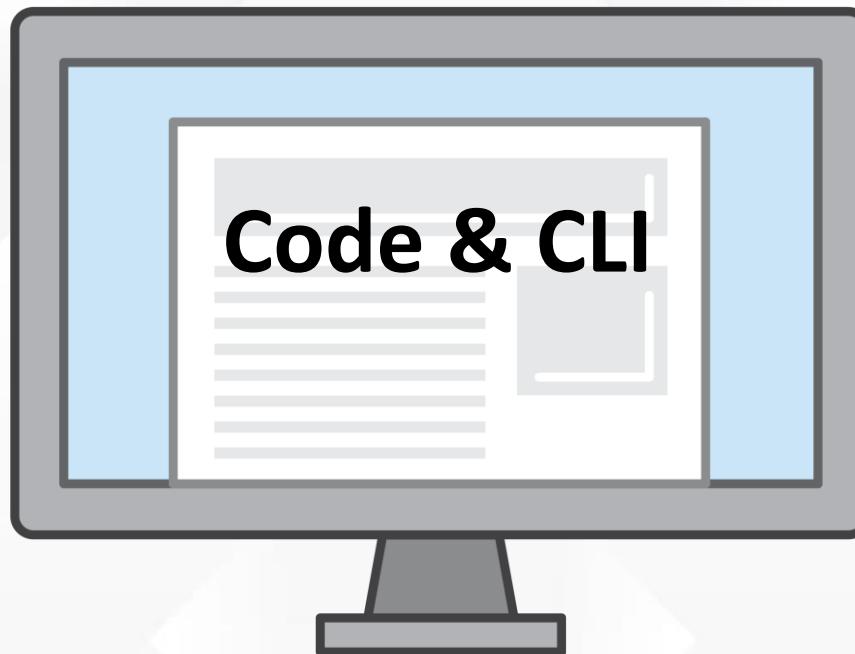
第三方服务



API & SDK

AWS Services

# 代码和命令行



# 操作AWS服务的三种方式



Management Console



APIs and SDKs



Command Line Interface

DevOps基础

AWS Tools (SDK, CLI, IDE, etc.):

# DEMO

# Python Code

## – Start two EC2 instances

```
runec2.py *  
1 import boto.ec2  
2  
3 conn = boto.ec2.connect_to_region("cn-north-1")  
4 conn.run_instances(  
5     'ami-981d8fa1',  
6     min_count=2,  
7     max_count=2,  
8     key_name='wendai-cn',  
9     instance_type='t2.micro',  
10    security_groups=['wslinux'],  
11    )  
12
```

# CLI

- Add Tag to EC2 instances

```
aws ec2 create-tags  
--resources i-ffb064c7 i-8eb561b6  
--tags Key=Name,Value=QConEC2
```

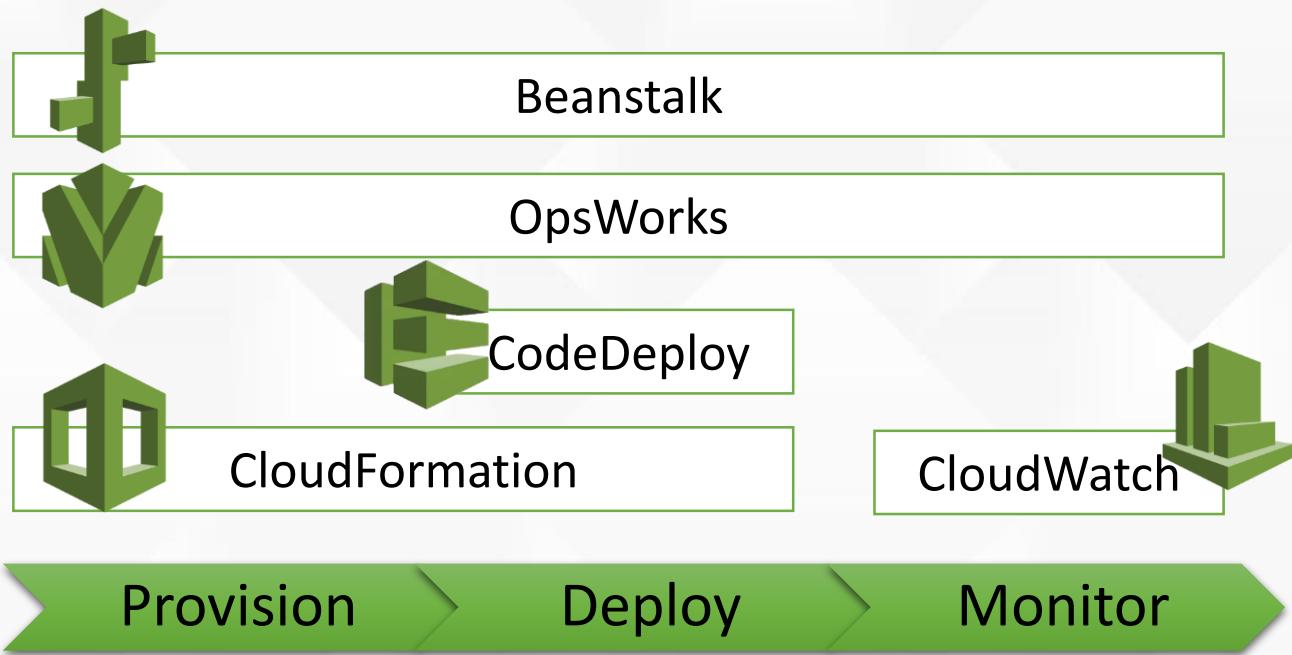
# 大规模基础架构的DevOps 需要框架和工具

# AWS DevOps服务适用场景

Convenience



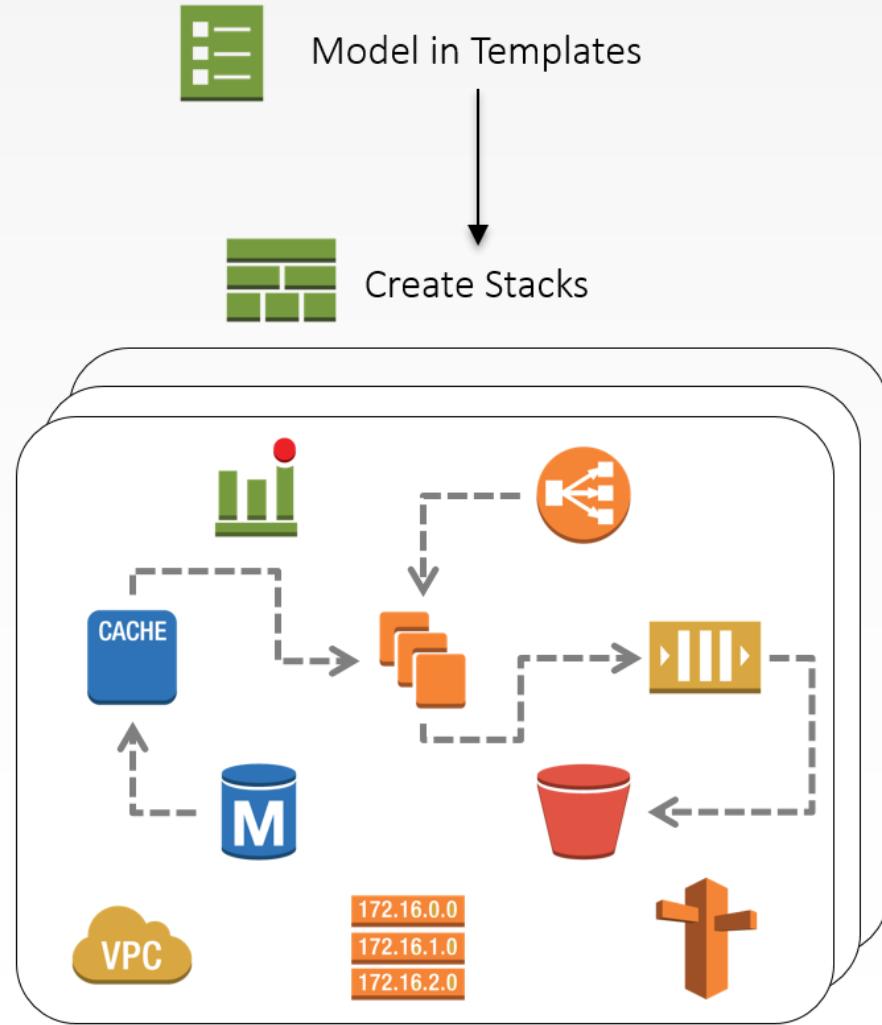
Customization



# CloudFormation



# 基础平台模板化

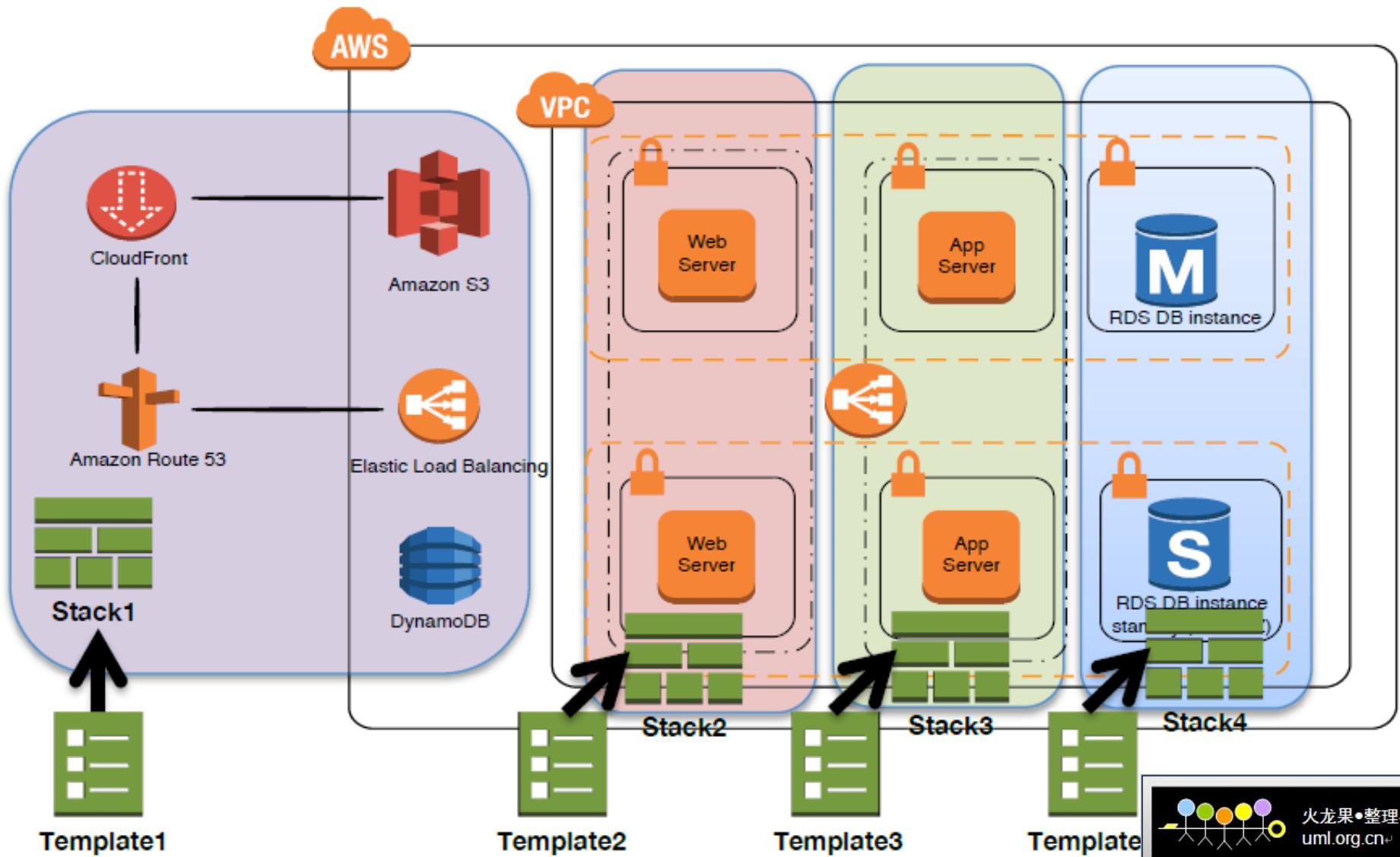


- 简化AWS服务的部署，快速部署一个Stack
- 模板化基础平台
- CloudFormation自动解决资源部署的先后和依赖关系
- 版本控制
- 第三方管理工具可以通过API集成CloudFormation

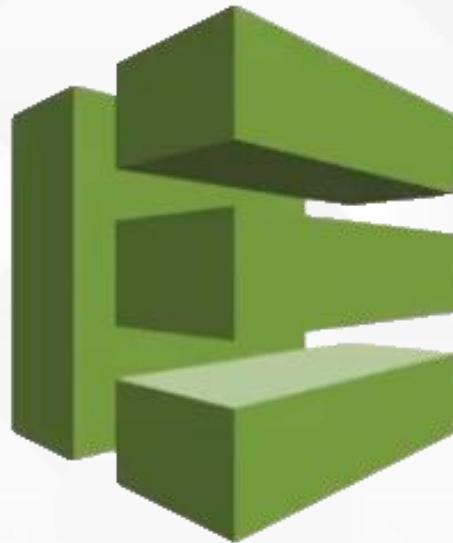
# Infrastructure as Code

```
{  
    "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" }  
        }  
    }  
}
```

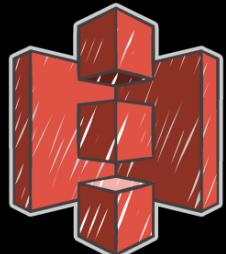
# 基于模板的快速部署



# CodeDeploy



# 自动化应用部署



Amazon S3



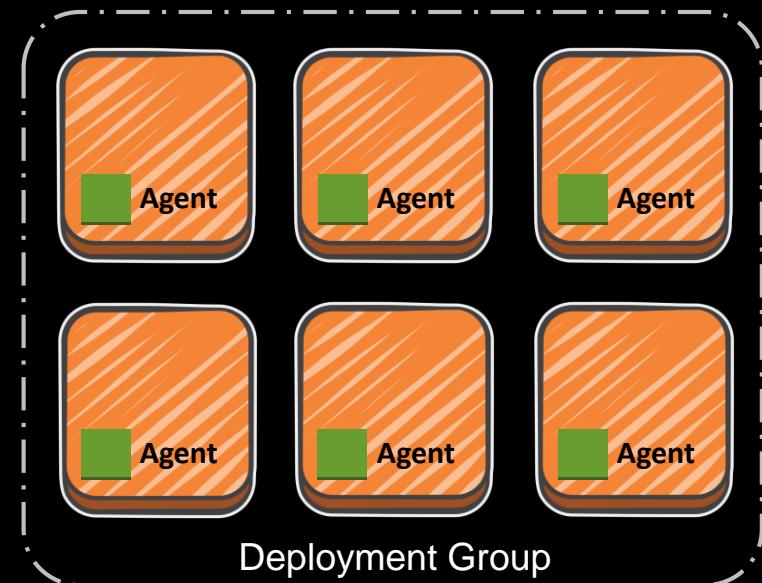
GitHub



Application  
Bundle

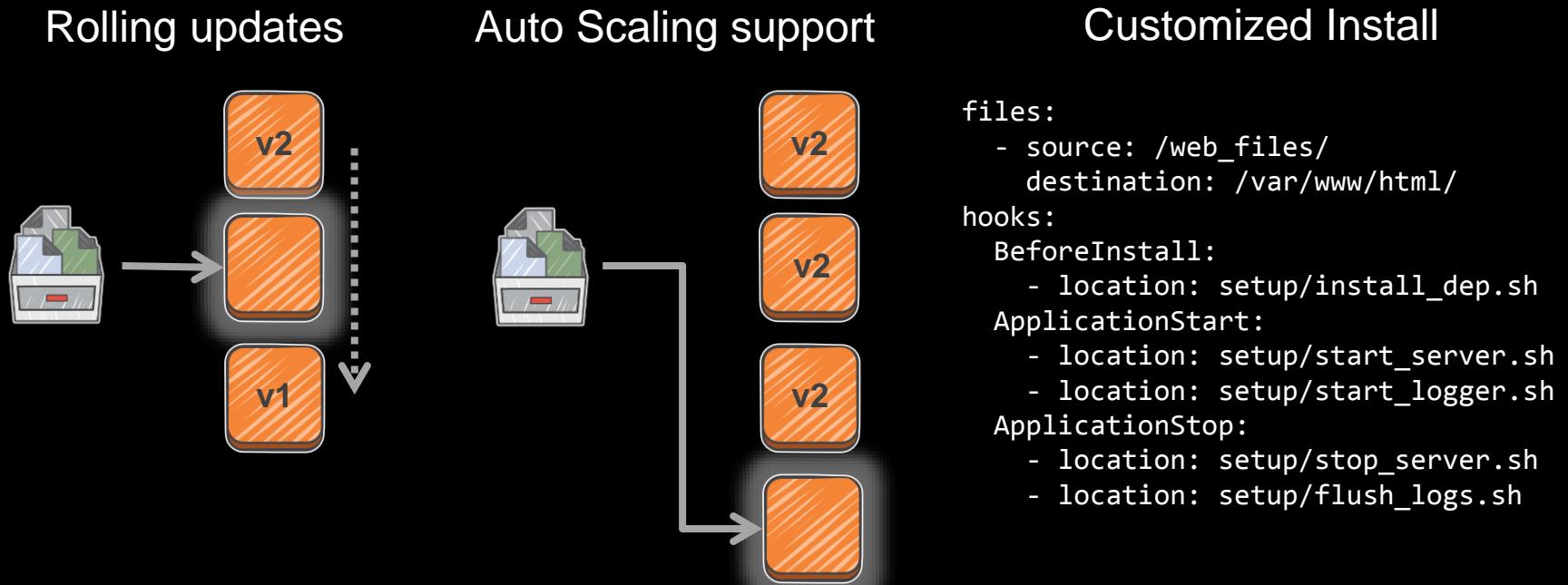


Deployment



Deployment Group

# Deployment Configuration



# Application Specification File

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
```

```
ApplicationStart:
  - location: scripts/start_server.sh
    timeout: 300
    runas: root
ApplicationStop:
  - location: scripts/stop_server.sh
    timeout: 300
    runas: root
```

# Demo Snapshot

## DemoApplication



Manage your application's deployment groups and revisions.

### Deployment Groups

[Create New Deployment Group](#)



Filter: Status

Search by Deployment Group

DemoFleet

Succeeded 3 instance(s)

DeploymentGroup2

### Revisions

Manage your application revisions. Select a deployment group on the left to view a list of revisions to deploy.



Revisions per page

10

< Viewing 1 to 1 Revision(s) >

| Revision Location                   | Created     | Last Deployed |
|-------------------------------------|-------------|---------------|
| s3://aws-codedeploy-us-west-2/sa... | 7 hours ago | 7 hours ago   |

### Delete Application

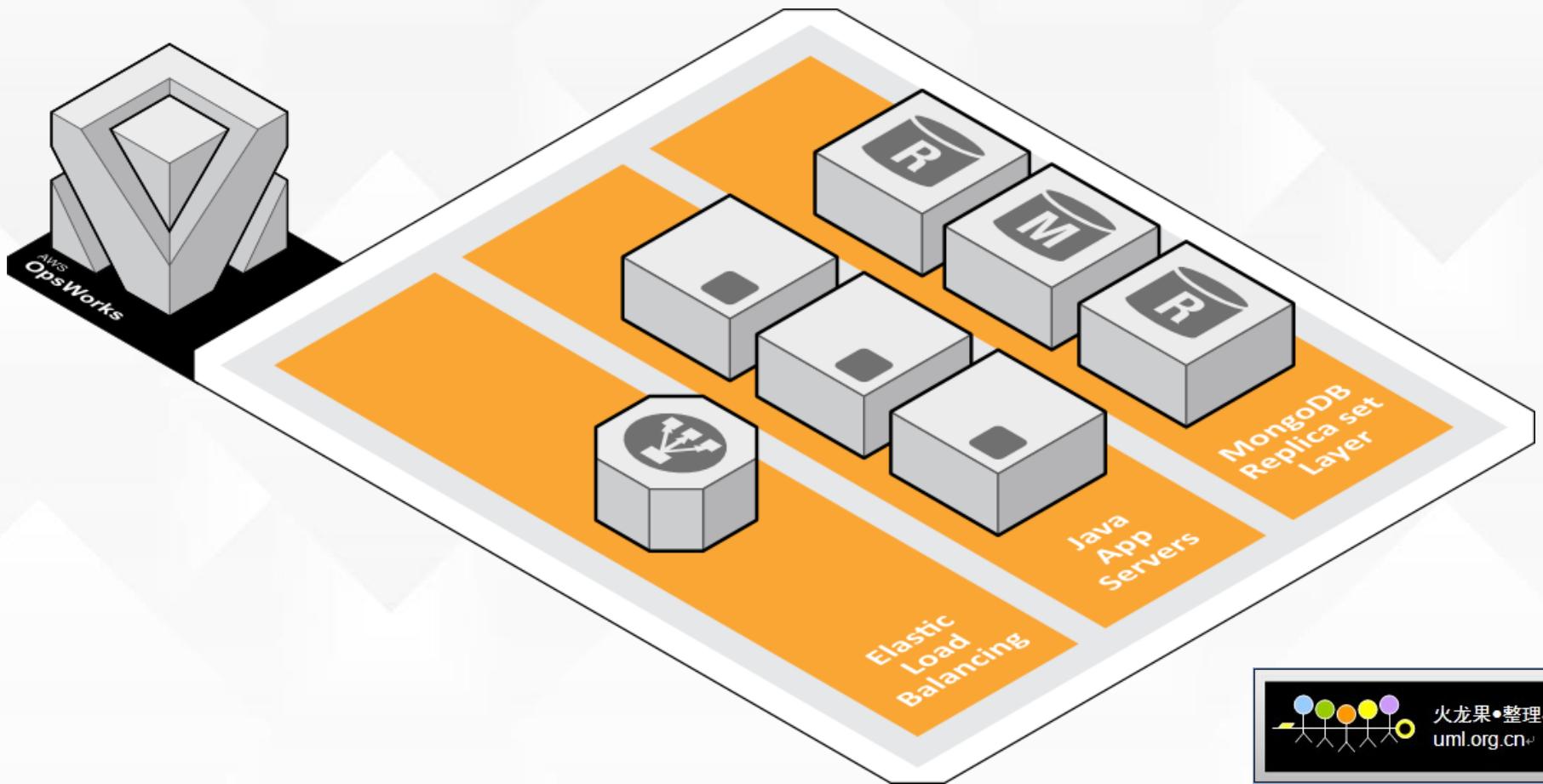
Deleting DemoApplication will delete all the associated deployment groups and revisions. This can't be undone. Are you sure you want to delete this application?

# OpsWorks

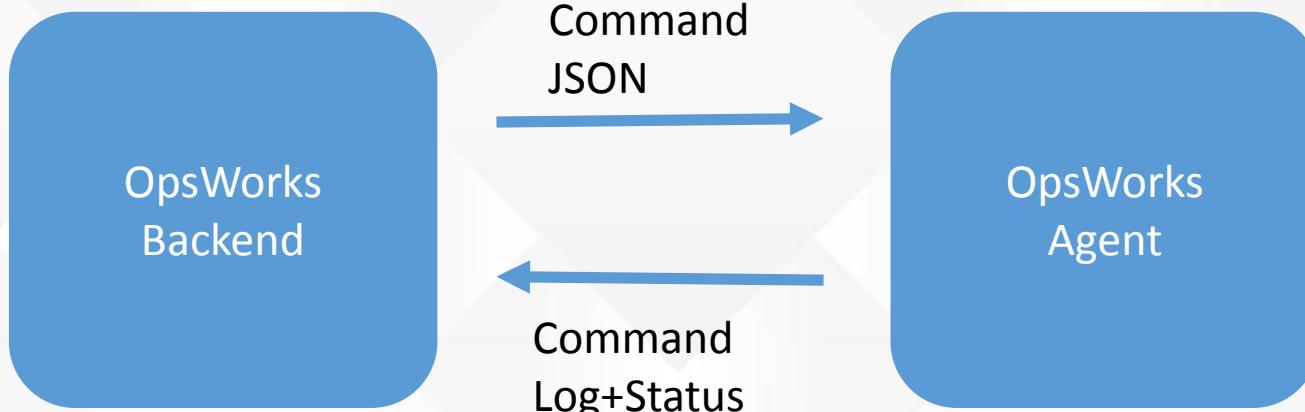


# OpsWork

AWS OpsWorks is a flexible application management solution with automation tools that enable you to model and control your applications and their supporting infrastructure.



# OpsWorks工作原理



Amazon EC2, Amazon EBS, EIP,  
Amazon VPC, Elastic Load Balancing....  
Auto-Scaling, Auto-Healing....

On-instance execution via  
Chef client/zero

大大简化了Chef环境的搭建

# Lifecycle events

setup



configure



deploy



undeploy



shutdown



# Built-in and Custom Chef Recipes

QConStack ▾

Dashboard

■ Stack

■ Layers

■ Instances

Time-based

Load-based

■ Apps

↳ Deployments

▶ Monitoring

📦 Resources

🔒 Permissions

## Layer Static Web Server

General Settings

Recipes

Network

EBS Volumes

Security

### Built-in Chef Recipes ⓘ

We have defined 17 built-in Chef recipes for your layer.

8 Setup

opsworks\_initial\_setup ssh\_host\_keys ssh\_users mysql::client dependencies ebs opsworks\_ganglia::client  
nginx

4 Configure

opsworks\_ganglia::configure-client ssh\_users mysql::client agent\_version

2 Deploy

deploy::default deploy::web

1 Undeploy

deploy::web-undeploy

2 Shutdown

opsworks\_shutdown::default nginx::stop

### Custom Chef Recipes ⓘ

Repository URL

[https://s3-ap-southeast-1.amazonaws.com/wendaifiles/opsworks\\_cookbooks\\_1.1.zip](https://s3-ap-southeast-1.amazonaws.com/wendaifiles/opsworks_cookbooks_1.1.zip) (change)

0 Setup

mycookbook::myrecipe, mycookbo



0 Configure

mycookbook::myrecipe, mycookbo



0 Deploy

mycookbook::myrecipe, mycookbo



0 Undeploy

mycookbook::myrecipe, mycookbo



0 Shutdown

mycookbook::myrecipe, mycookbo



# Custome Recipe Demo

## - 创建目录

```
Chef::Log.info("*****Creating a data directory.*****")

data_dir = value_for_platform(
  "centos" => { "default" => "/srv/www/shared" },
  "ubuntu" => { "default" => "/srv/www/data" },
  "default" => "/srv/www/config"
)

directory data_dir do
  mode 0755
  owner 'root'
  group 'root'
  recursive true
  action :create
end
```

█  
~  
~

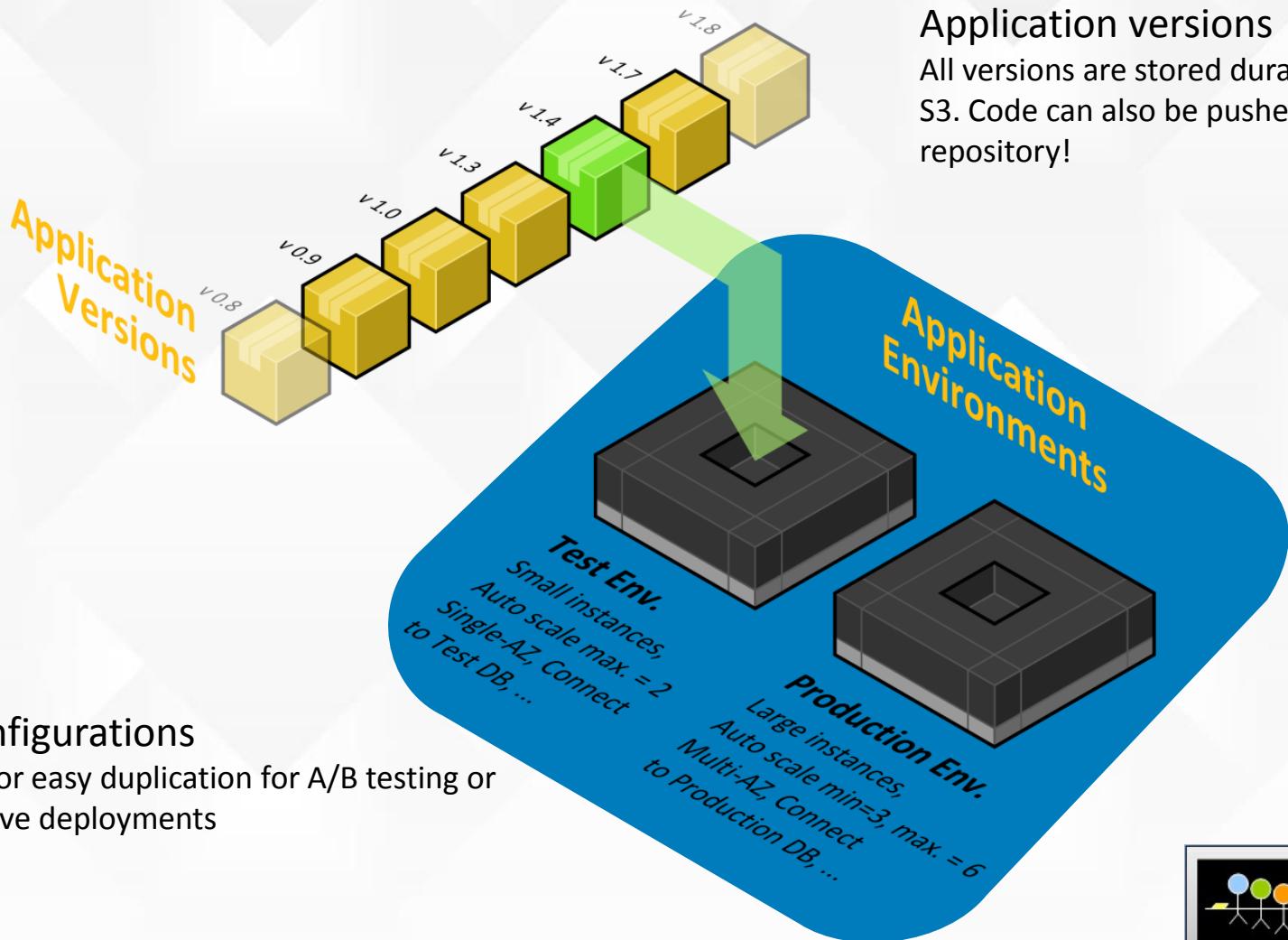
# Cookbook development workflow



# ElasticBeanstalk



# Elastic Beanstalk 工作原理



# Application

Elastic Beanstalk   HelloWorld ▾   My First Elastic Beanstalk Application ▾   Create New Environment

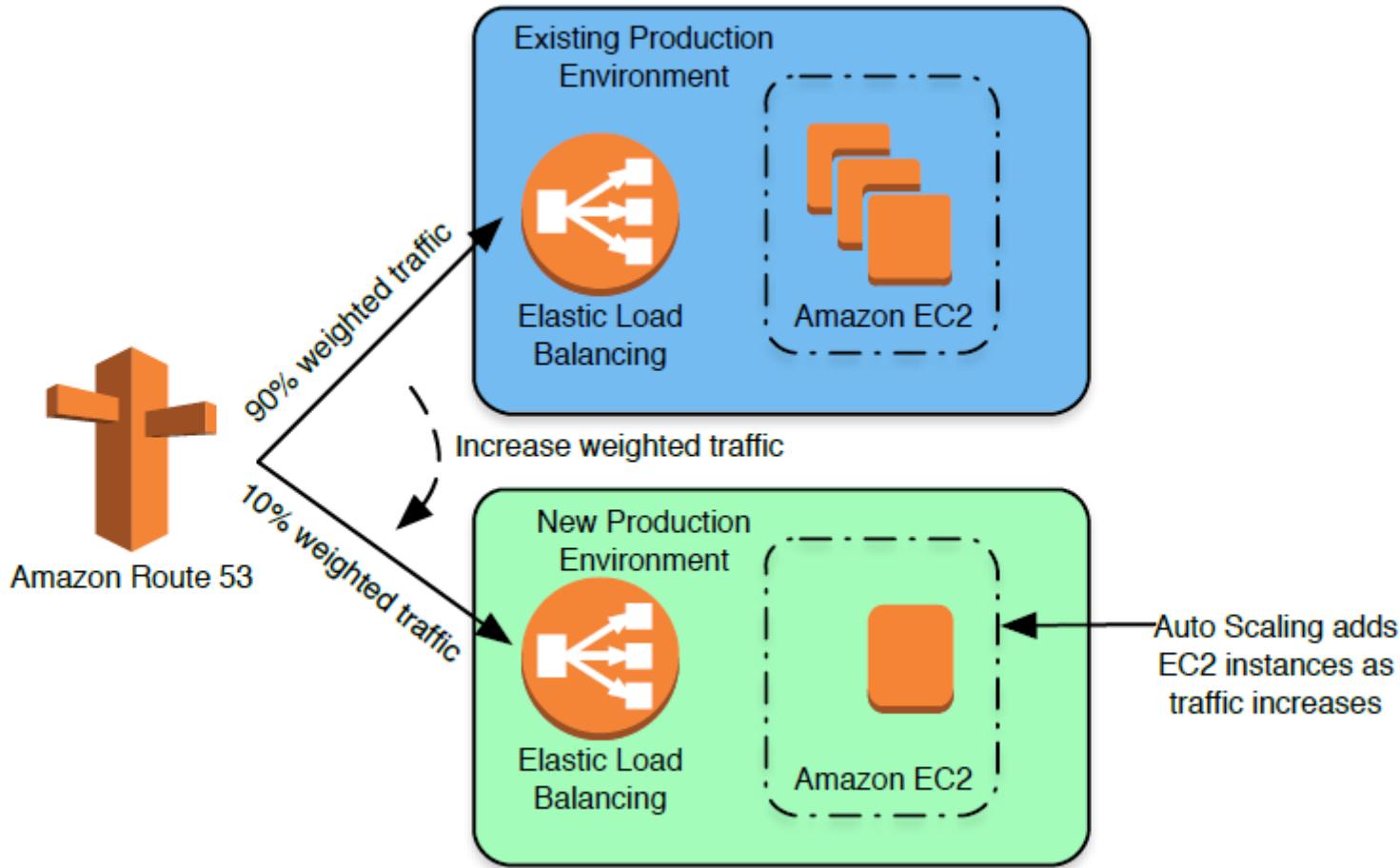
### My First Elastic Beanstalk Application

Actions ▾

| Environments         |  |   |             |                              |                                    | <a href="#">Delete</a> | <a href="#">Deploy</a> | <a href="#">Upload</a> | <a href="#">Refresh</a> |
|----------------------|--|---|-------------|------------------------------|------------------------------------|------------------------|------------------------|------------------------|-------------------------|
| Application Versions |  | Version Label                               | Description | Date Created                 | Source                             | Deployed To            |                        |                        |                         |
|                      |  | <input type="checkbox"/> Sample Application |             | 2015-04-23 16:33:35 UTC+0800 | <a href="#">Sample Application</a> | Default-Environment    |                        |                        |                         |

Saved Configurations

# 蓝绿部署



# Demo: CLI workflow

## Initial app deployment:

01

Initialize your Git repository

```
$ git init .
```

02

Create your Elastic Beanstalk app

```
$ eb init
```

03

*Follow the prompts to configure the environment*

04

Add your code

```
$ git add .
```

05

Commit

```
$ git commit -m "v1.0"
```

06

Create the resources and launch the application

```
$ eb create
```

# Demo: CLI workflow

Update your app:

01 Update your code

02 Push the new code

```
$ git add .  
$ git commit -m "v2.0"  
$ eb deploy
```

03 Monitor the deployment progress

```
$ eb status
```

# 贯彻始终的安全与监控

Amazon  
CloudWatch



Monitor resources

AWS IAM (Identity  
& Access Mgmt)

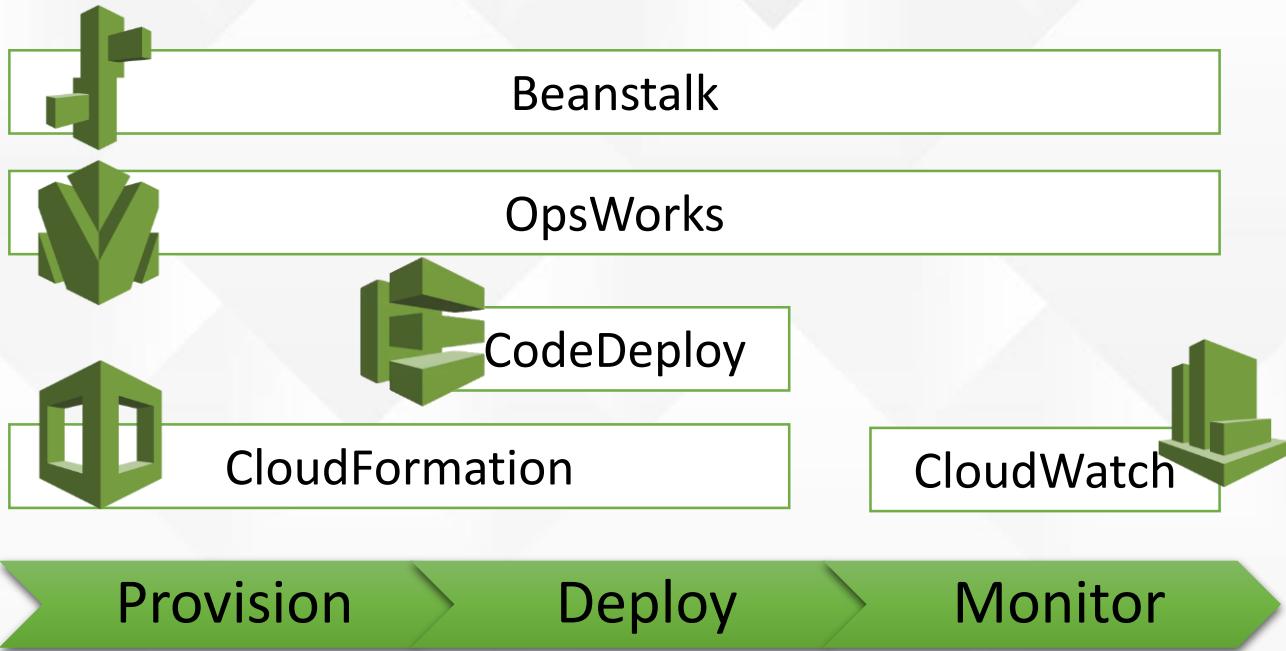


Manage users,  
groups &  
permissions

**Convenience**



**Customization**



# 感谢您宝贵的时间！