DevOps
with Amazon Web Services

Julien Simon
Principal Technical Evangelist, AWS
julsimon@amazon.fr
@julsimon

Agenda

- A few definitions
- The DevOps story at Amazon.com
- The Code* services
- Infrastructure as Code

CI, CD, DevOps

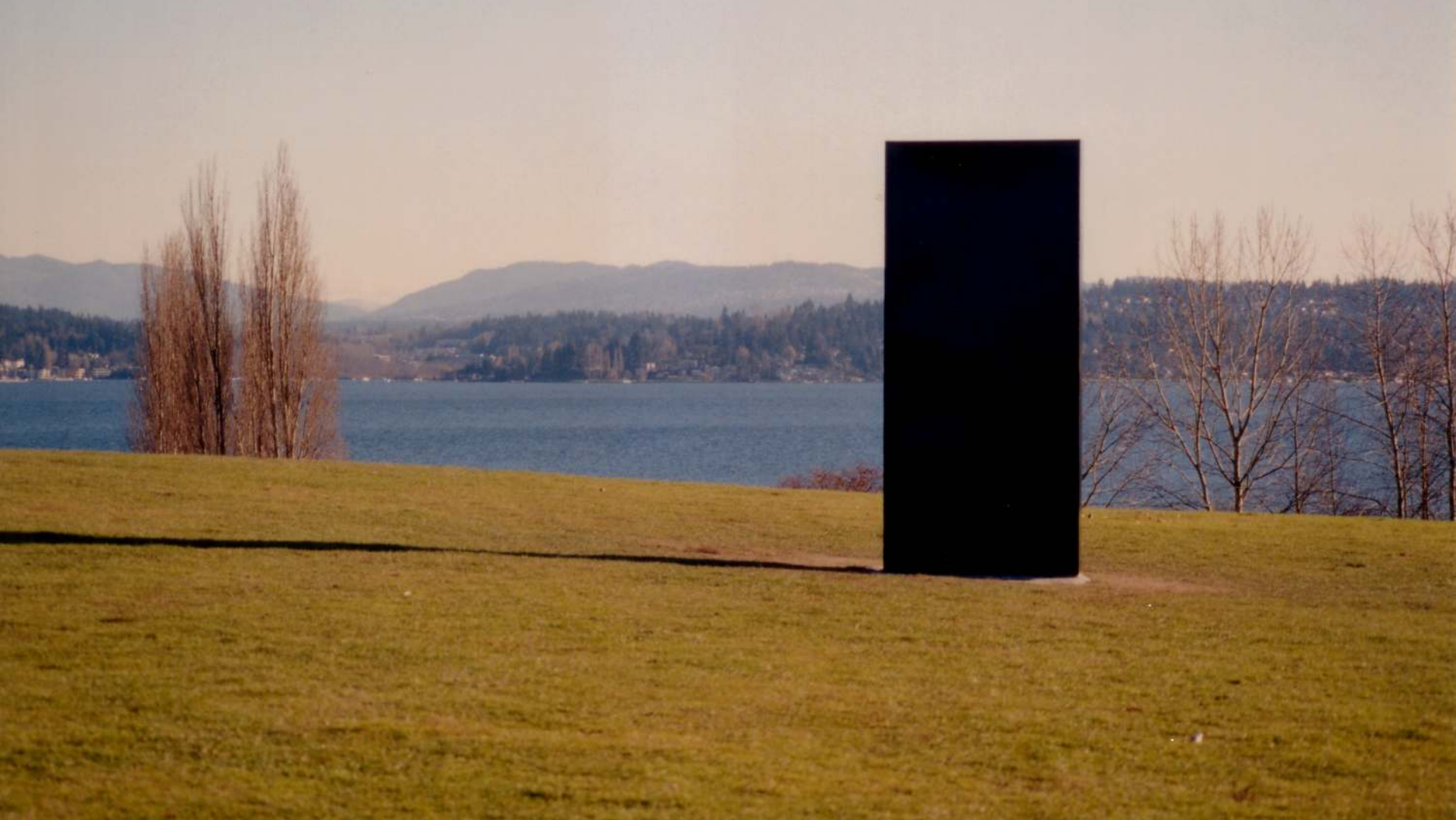
Continuous Integration: breaking down system integration into small steps by regularly merging code into a shared mainline and fully testing the updated code automatically

Continuous delivery: a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.

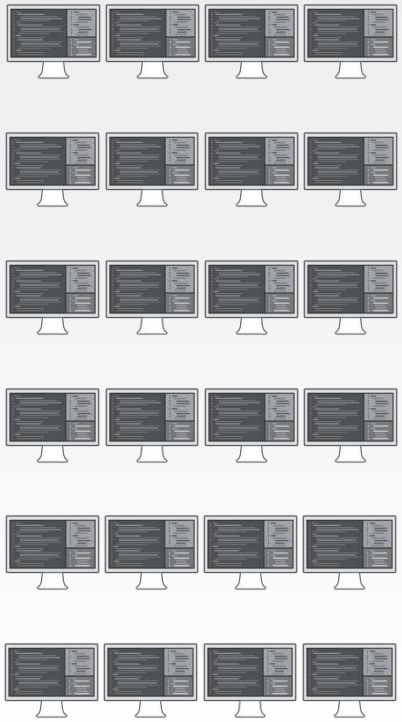
Continuous deployment: every change is automatically deployed to production

DevOps: a set of practices that emphasize the collaboration and communication of both software and operations while automating the process of software delivery and infrastructure changes.

Source: Wikipedia



Monolith development lifecycle

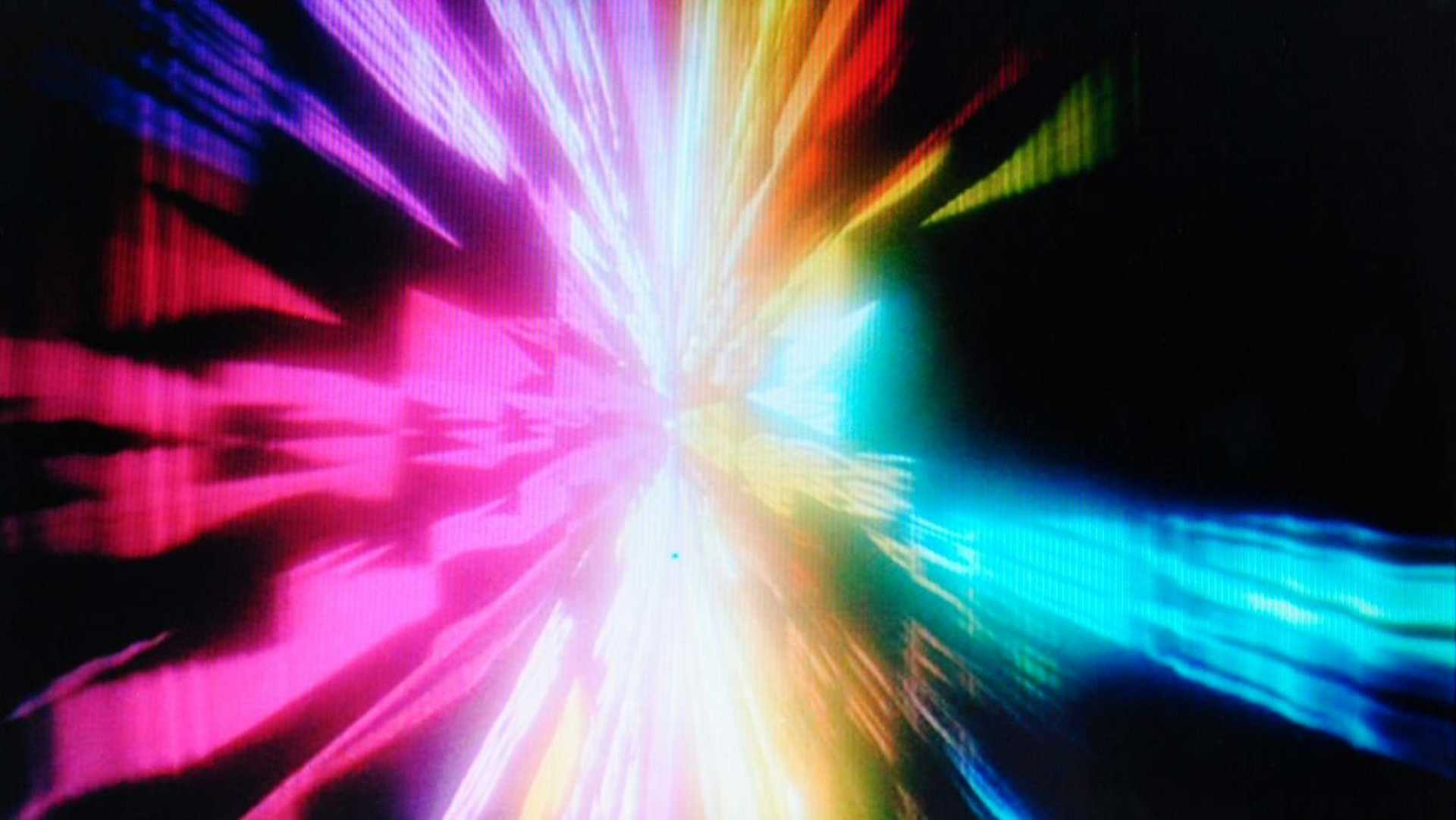


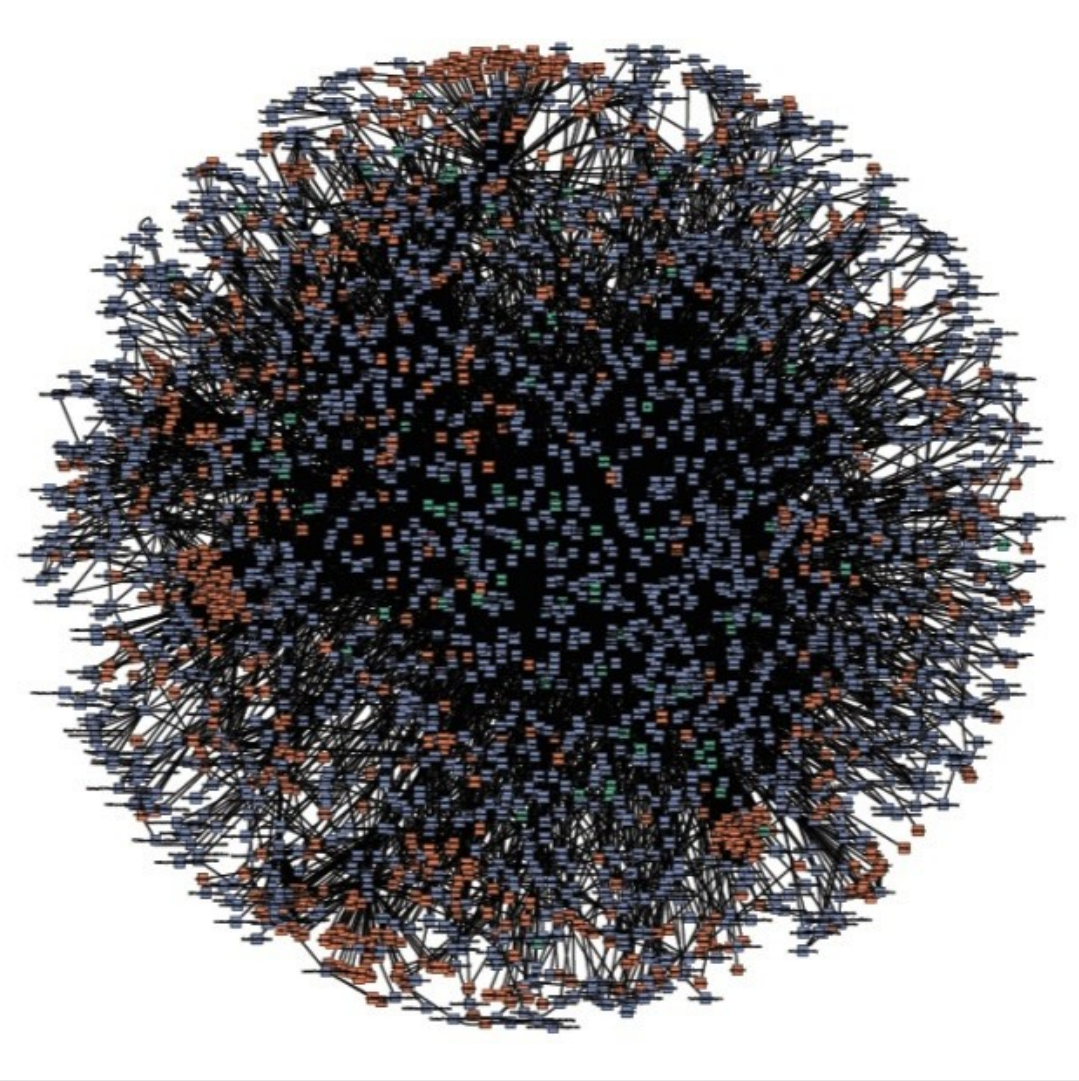
developers

app

delivery pipeline


IWORM





Service-Oriented Architecture (SOA)

Single-purpose

Connected through APIs

Highly decoupled

“Microservices”

Two-pizza teams

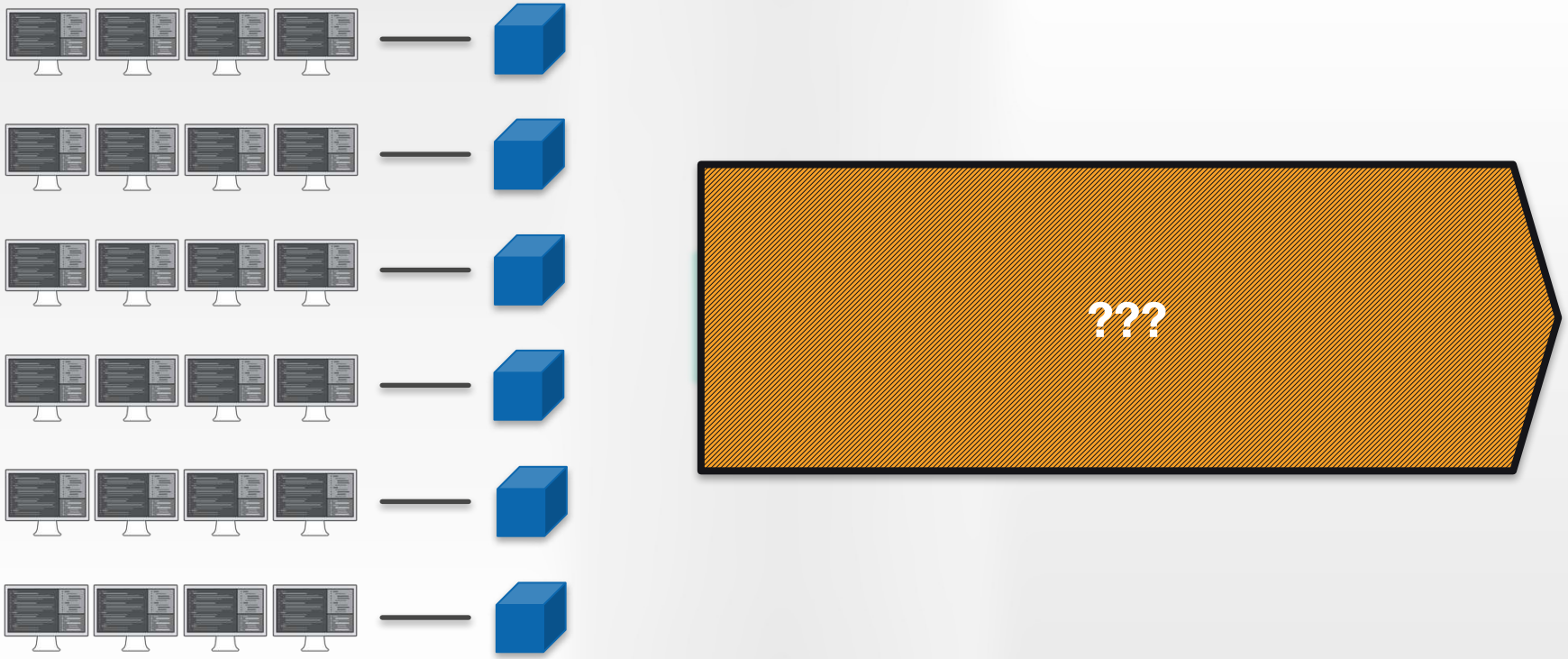
Full ownership

Full accountability

Aligned incentives

“DevOps”

Missing tools



Self-service

Technology-agnostic

Encourage best practices

Single-purpose services

Deployment service

No downtime

Health tracking

Versioned artifacts

Rollbacks

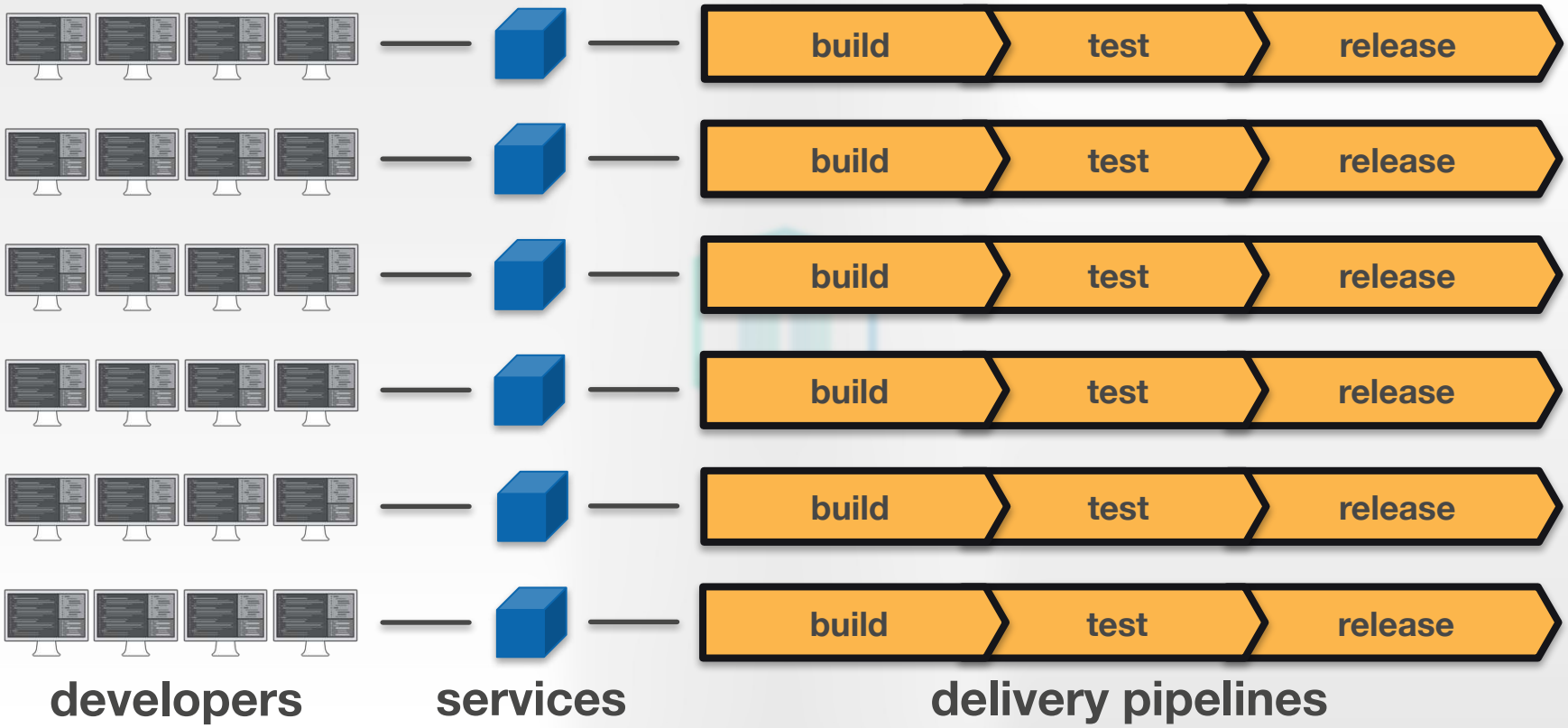
Continuous delivery

Automated release process

Faster and more reliable releases

Used by >90% of teams

DevOps development lifecycle



Thousands of teams

× Microservice architecture

× Continuous delivery

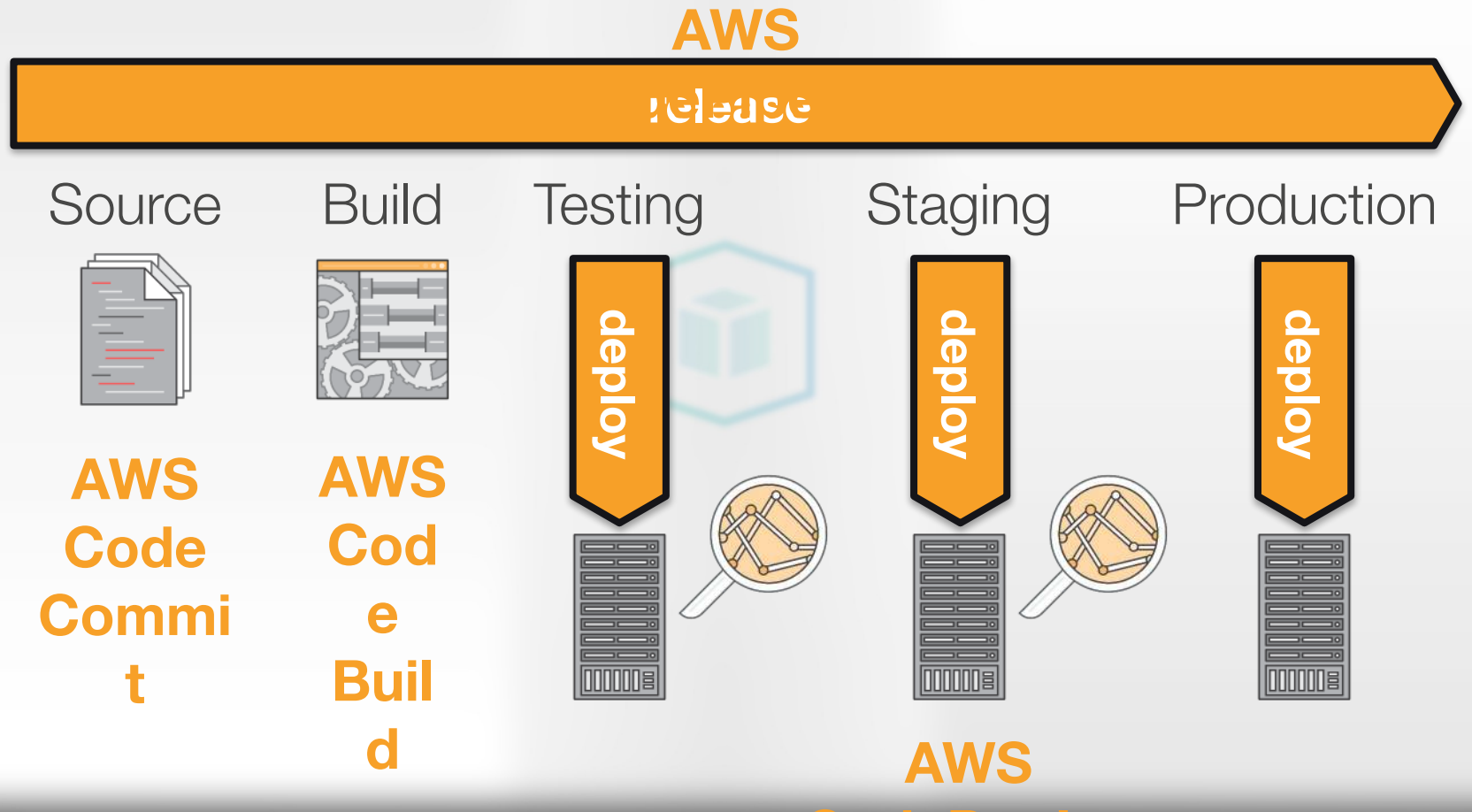
× Multiple environments

= 50 million deployments a year

(1.5 deployment every second)

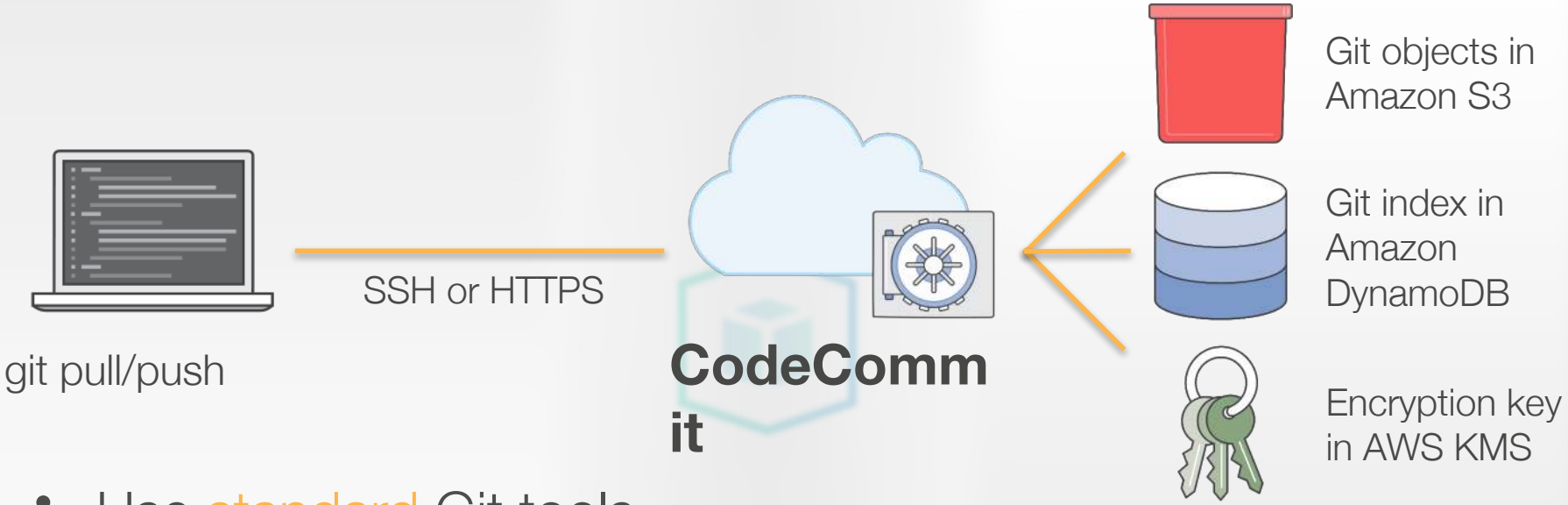
How can we help others do this?

Setting up a delivery pipeline



AWS Code* partners

AWS CodeCommit

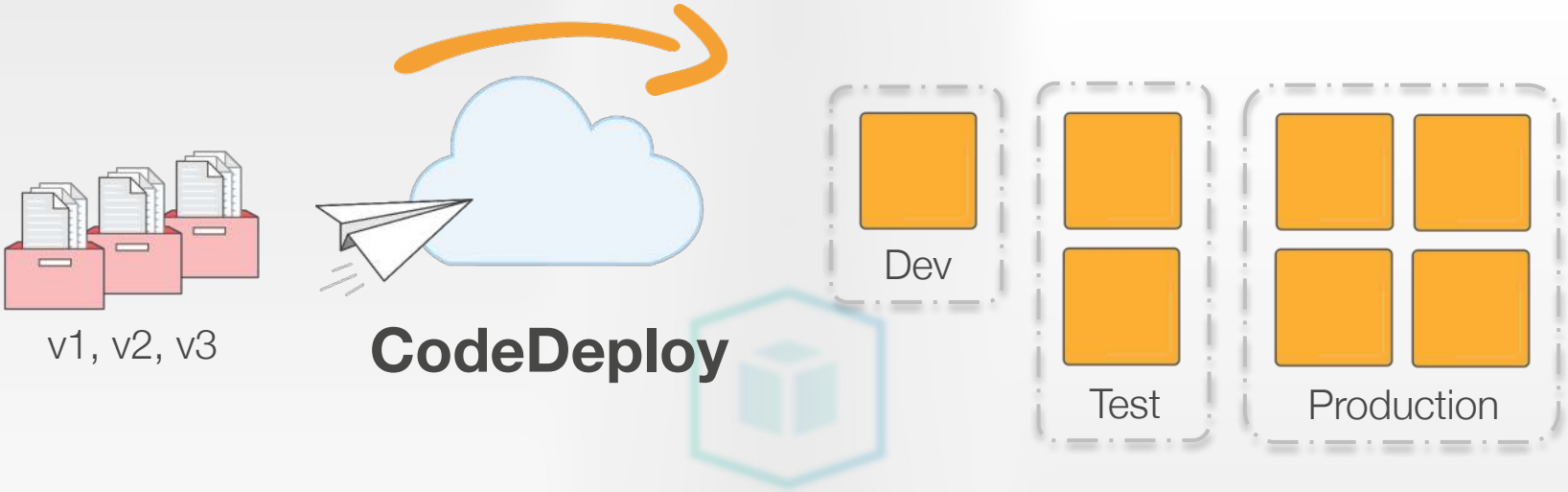


- Use standard Git tools
- Scalability, availability and durability of Amazon S3
- Encryption at rest with customer-specific keys
- Pricing: first 5 users free, then $1 / user / month

AWS CodeBuild

- New service launched at re:Invent 2016
- Managed build environments (Linux only for now)
- Pull sources from Github, S3 or CodeCommit
- Build on an AWS-provided image or your Docker container
- Supported environments : “base”, Android, Java, Go, Python, Ruby, Go, Docker
- Build commands: inline or in buildspec.yml file
- Pricing starts at $0.005 per minute (free tier available)

AWS CodeDeploy



- Easy and reliable deployments: zero downtime, rollbacks
- Deploy to any server with agent: Linux / Windows, EC2 / on-premise
- Scale with ease: Auto Scaling groups supported
- Green-blue deployment supported
- Pricing : no extra charge for EC2

AWS CodePipeline

Source	Build	Deploy	Approve	Deploy
1) Pull	1) Build 2) Unit test	1) Deploy beta 2) QA	SNS	Deploy prod

- Define stages: Source, Build, Test, Deploy, Invoke, Approve
- Connect to best-of-breed tools
- Build a fast, consistent and traceable release process
- S3 is used to stored source and build artefacts
- Pricing: $1 / active pipeline / month

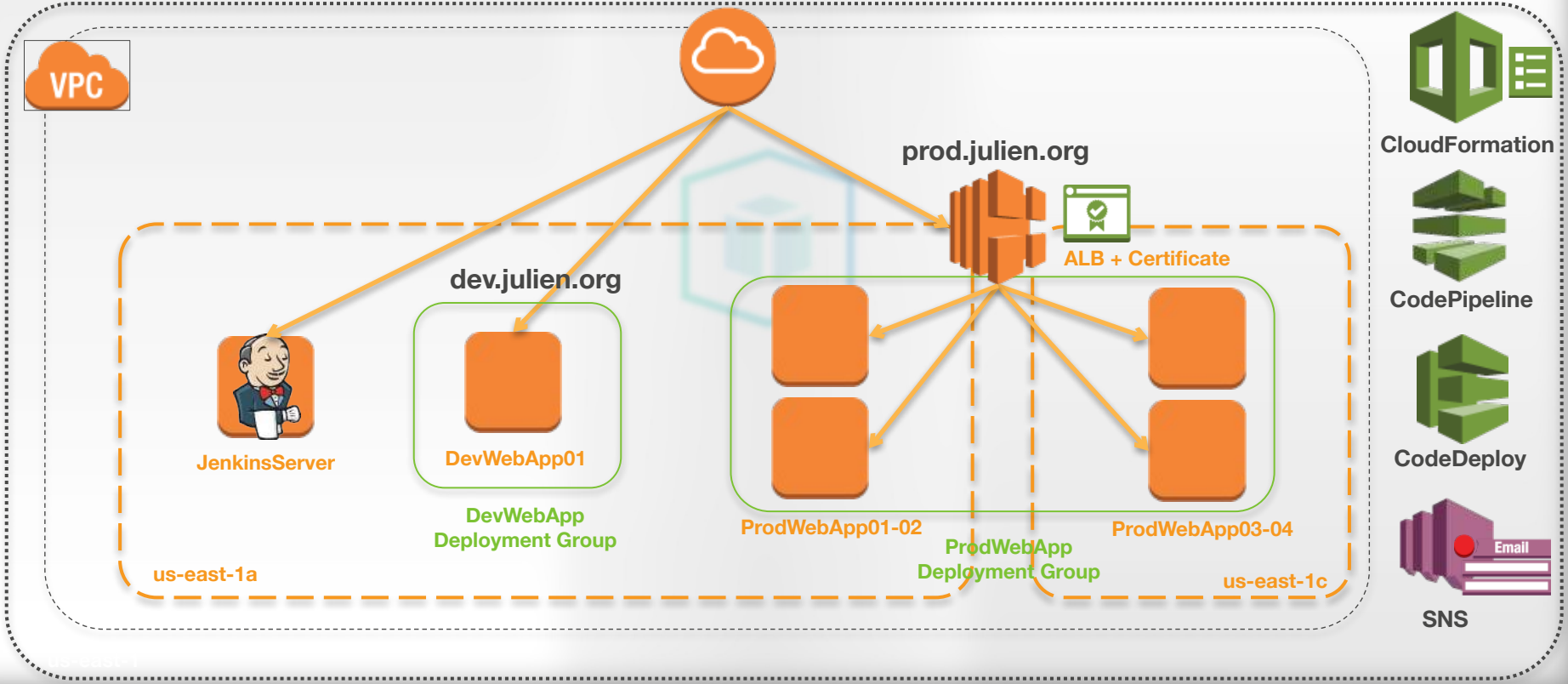
Demo

Expanded version of http://blogs.aws.amazon.com/application-management/post/Tx2CIB02ZO05ZII/Explore-Continuous-Delivery-in-AWS-with-the-Pipeline-Starter-Kit

AWS Code* demo

Source (GitHub) → Build (Jenkins) → Deploy Dev (CodeDeploy) → Approve (SNS Email) → Deploy Prod (CodeDeploy)

Code
+ appspec.yml
+ scripts



APPROVAL NEEDED: AWS CodePipeline app-name-Pipeline for action My_Approval

Email JS 11:15

Hello,

The following Approval action is waiting for your response:

--Pipeline Details--

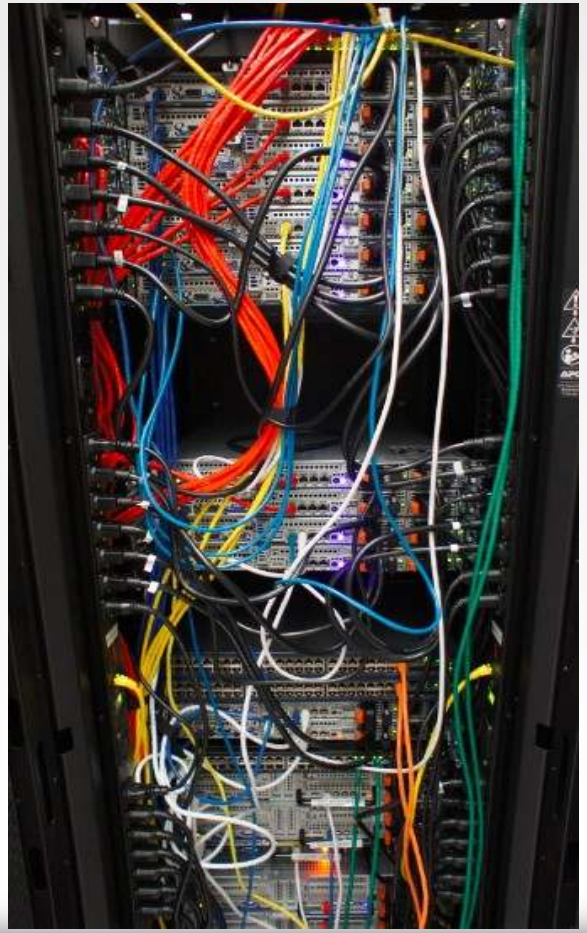
Pipeline name: app-name-Pipeline
Stage name: Approval
Action name: My_Approval
Region: us-east-1

--Approval Details--

Content to review: http://dev.julien.org
Approve or reject: https://console.aws.amazon.com/codepipeline/home?region=us-east-1#/view/app-name-Pipeline/Approval/My_Approval/approve/0bba2e19-4c19-4a74-87de-cdc1f9612613
Additional information: Please review this deployment
Deadline: This review request will expire on 2016-10-12T09:15Z

Sincerely,
Amazon Web Services

AWS CloudFormation



```
"Conditions" : {
  "HaveNoOtherRoles"  : { "Fn::Equals" : [{"Ref" : "OtherRoles"}, ""]},
  "HaveEbs"           : { "Fn::Not" : [{ "Fn::Equals" : [{"Ref" : "EbsVolumeSize"}, "0"]}] },
  "HaveEbsSnapshotId" : { "Fn::Not" : [{ "Fn::Equals" : [{"Ref" : "EbsSnapshotId"}, ""]}] },
  "HaveAdditionalTagKey": { "Fn::Not" : [{ "Fn::Equals" : [{"Ref" : "AdditionalTagKey"}, ""]}] },
  "HaveAdditionalTagValue": { "Fn::Not" : [{ "Fn::Equals" : [{"Ref" : "AdditionalTagValue"}, ""]}] },
  "HaveSSL": { "Fn::Not" : [{ "Fn::Equals" : [{"Ref" : "SSLPort"}, "0"]}] },
  "IsHTTP" : { "Fn::Equals" : [{"Ref" : "ElbProtocol"}, "HTTP"]},
  "HaveSpotPrice" : { "Fn::Not" : [{ "Fn::Equals" : [{"Ref" : "SpotPrice"}, ""]}]}
},
"Resources": {
  "AutoScalingGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "UpdatePolicy" : {
      "AutoScalingRollingUpdate" : {
      "MaxBatchSize" : "1",
      "MinInstancesInService" : "0",
      "PauseTime" : "PT15M",
      "WaitOnResourceSignals": "true"
      }
    },
    "Properties": {
      "LaunchConfigurationName": { "Ref": "LaunchConfig" },
      "LoadBalancerNames": [ { "Ref": "ElasticLoadBalancer" } ],
      "MinSize": { "Ref": "MinPoolSize" },
      "MaxSize": { "Ref": "MaxPoolSize" },
      "AvailabilityZones": { "Fn::FindInMap": ["AZConfig", "AvailabilityZones", "all"] },
      "VPCZoneIdentifier": { "Ref": "EC2SubnetsIds" },
      "Tags" : [
        { "Fn::If": [
            "HaveAdditionalTagKey",
            {
              "Key" : { "Ref": "AdditionalTagKey" },
              "Value": {
                "Fn::If": [
                  "HaveAdditionalTagValue",
                  {"Ref": "AdditionalTagValue"},
                  ""
                ]
              },
              "PropagateAtLaunch": "true"
            },
            {"Ref" : "AWS::NoValue"}
          ]
        },
        { "Key" : "Name", "Value" : { "Fn::Join" : [ "-", [ { "Ref" : "ServiceName"}, {"Ref" : "EnvironmentName"
        { "Key" : "cost", "Value" : { "Ref" : "Cost" }, "PropagateAtLaunch": "true" },
        { "Key" : "environment", "Value": { "Ref" : "EnvironmentName"}, "PropagateAtLaunch": "true" }
      ]
    }
  }
```

Why infrastructure as code rocks

Automated: save time & reduce human error

Predictable: build the same infra every time

Traceable: keep track of all changes

Testable: make sure best practices are built-in

You don’t get all of this with scripting

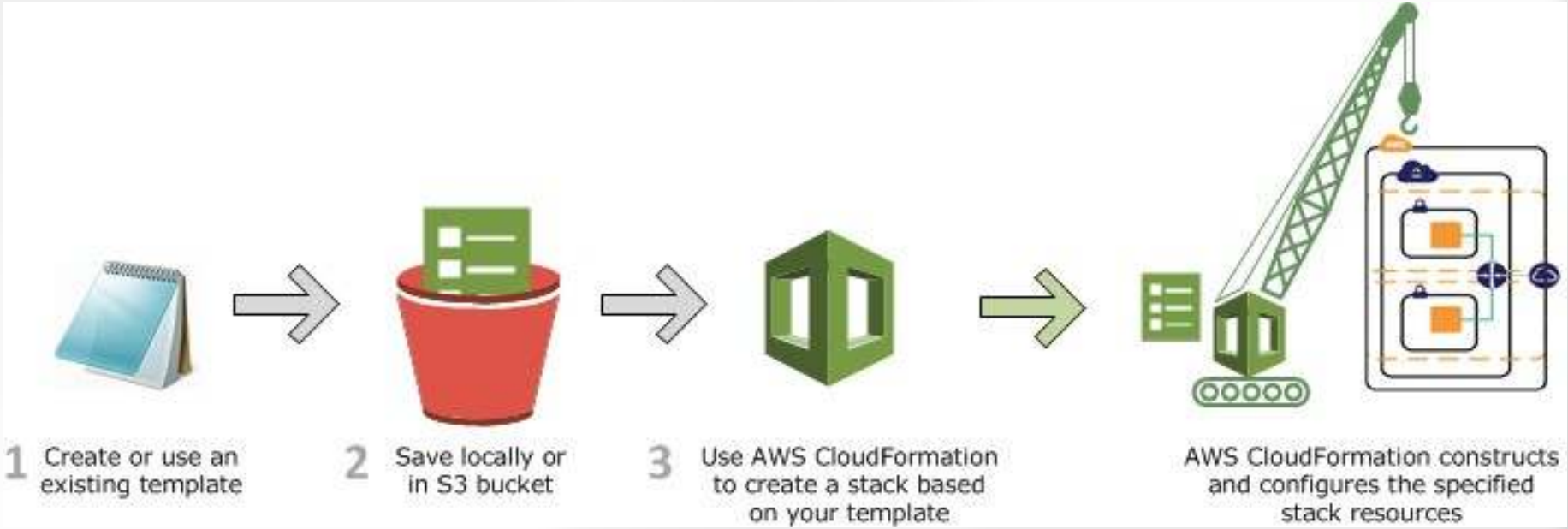
Typical use cases

- Building as many environments as you need
  - Development, staging, pre-production, production
  - Same architecture, different sizing → template + parameters
- Deploying in a different region
- Performing green / blue deployments
- Preparing for Disaster Recovery

AWS CloudFormation

- Fundamental service used to automate creation, configuration and destruction of AWS resources (VPC, EC2, RDS, etc.)
- Infrastructure is described in a template
  - JSON or YAML file. Not a script!
  - Resources, Parameters, Outputs, etc.
- CloudFormation ingests the template and builds a stack of AWS resources
- Pricing: no charge

AWS CloudFormation



CloudFormation Template

```
{
  "AWSTemplateFormatVersion" : "version date",

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },

  "Parameters" : {
    set of parameters
  },

  "Mappings" : {
    set of mappings
  },

  "Conditions" : {
    set of conditions
  },

  "Resources" : {
    set of resources
  },

  "Outputs" : {
    set of outputs
  }
}
```

The CloudFormation CLI in one slide

```
$ aws cloudformation validate-template
--template-body file://template.json

$ aws cloudformation create-stack
--template-body file://template.json --stack-name MyTemplate

$ aws cloudformation get-template --stack-name MyTemplate

$ aws cloudformation update-stack --stack-name MyTemplate
--template-body file://template.json

$ aws cloudformation delete-stack --stack-name MyTemplate
```

Change sets

- CloudFormation used to be ‘fire and forget’
  - Or sometimes ‘fire and remember all your life’ ;)

- Change sets have been introduced to preview effects of stack creations and stack updates

- Please use them!
  - `aws cloudformation create-change-set`
  - `aws cloudformation describe-change-set`
  - `aws cloudformation execute-change-set`
  - `aws cloudformation delete-change-set`

Closing words

- Automation is a key factor in technical & business agility
- You can use the same tools as Amazon.com!
- Zero dev infrastructure to purchase & manage
- Minimal cost
- Compatible with your existing CI/CD tools
- Not only apps, infrastructure too

Next steps

Learn more

http://aws.amazon.com/awscode

http://blogs.aws.amazon.com/application-management

https://aws.amazon.com/fr/new/#dev-tools

Get started

http://aws.amazon.com/free

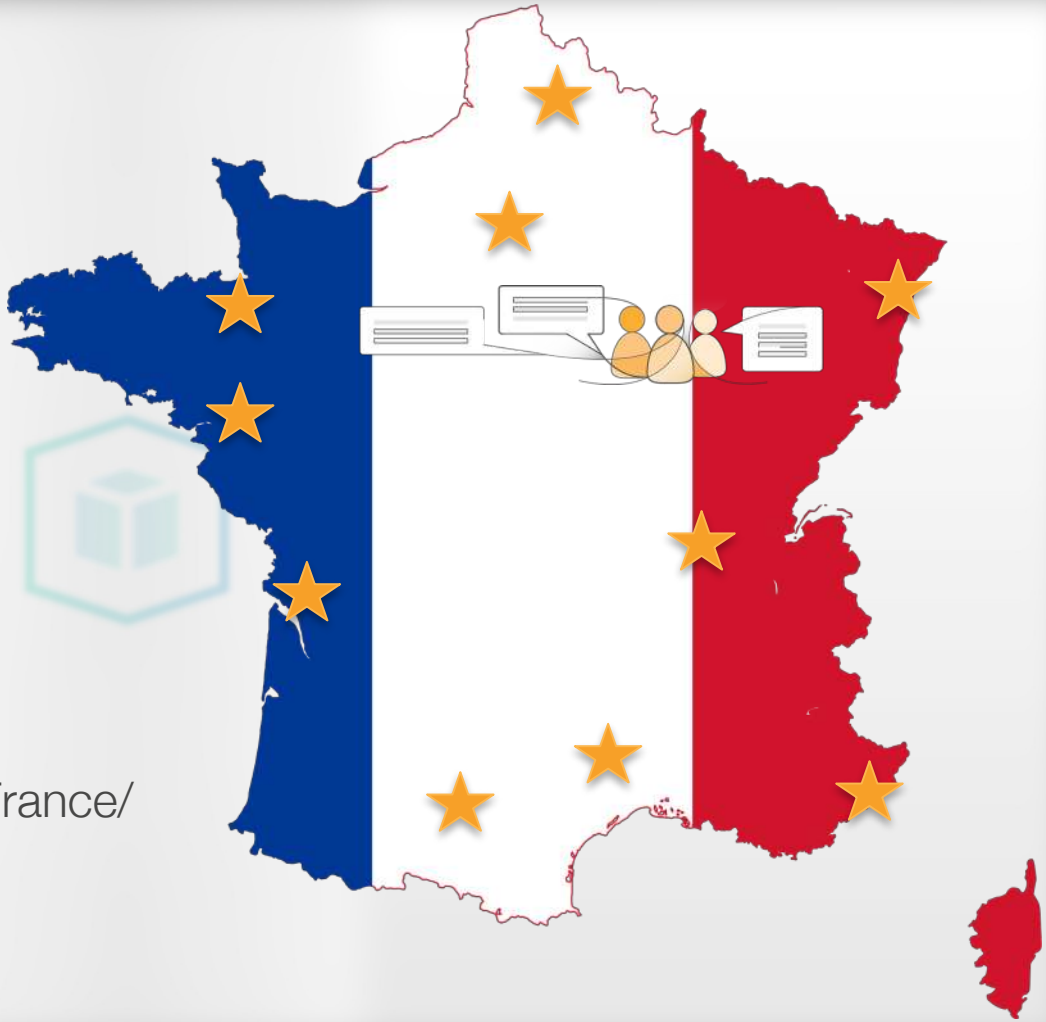
http://console.aws.amazon.com

AWS User Groups

Lille
Paris
Rennes
Nantes
Bordeaux
Lyon
Montpellier
Toulouse
Côte d'Azur
Grand Est (new!)

facebook.com/groups/AWSFrance/

@aws_actus



Thank you!

Julien Simon
Principal Technical Evangelist, AWS
julsimon@amazon.fr
@julsimon