Infrastructure as Code: CloudFormation Best Practices

AWS Summit Berlin 2016

Matthias Jung, Solutions Architect

April, 12th, 2016

Agenda

- Why CloudFormation?
- How to plan my stacks?
- How to get started?
- How to prevent errors?
- How to safely update stacks?
- How to extend CloudFormation?

Why CloudFormation?

Setting Up an Application

Series of
Operational Tasks

Configure Network & Firewalls

Setup Load Balancer

Configure Servers

Setup Database

Configure Access Rights

…

Setting Up an Application

Series of API Calls to AWS

Configure VPC

Launch ELB

Launch EC2 Instances

Launch RDS Instance

Define IAM Users

…

Setting Up an Application

Series of API Calls to AWS

AWS CLI & SDKs

Configure VPC

Launch ELB

Launch EC2 Instances

Launch RDS Instance

Define IAM Users

…

Setting Up an Application

Template of Resources

VPC

ELB

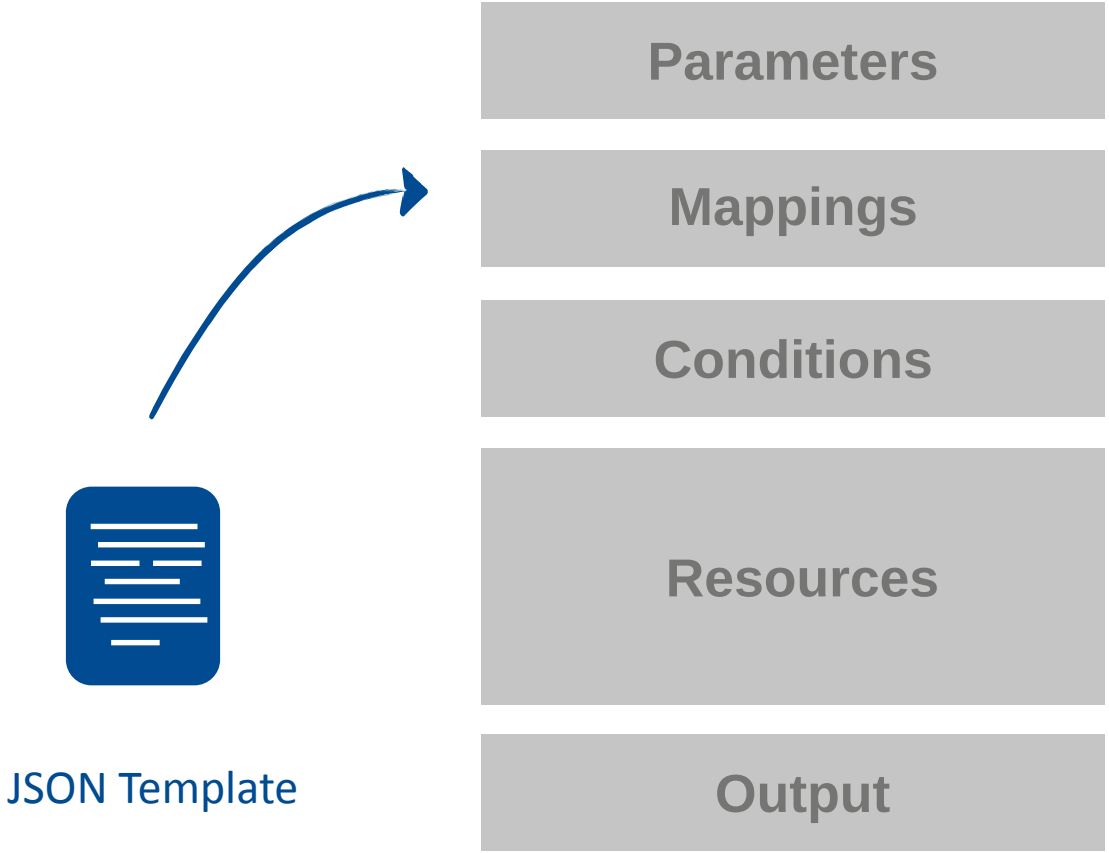
EC2 Instances

RDS Instance

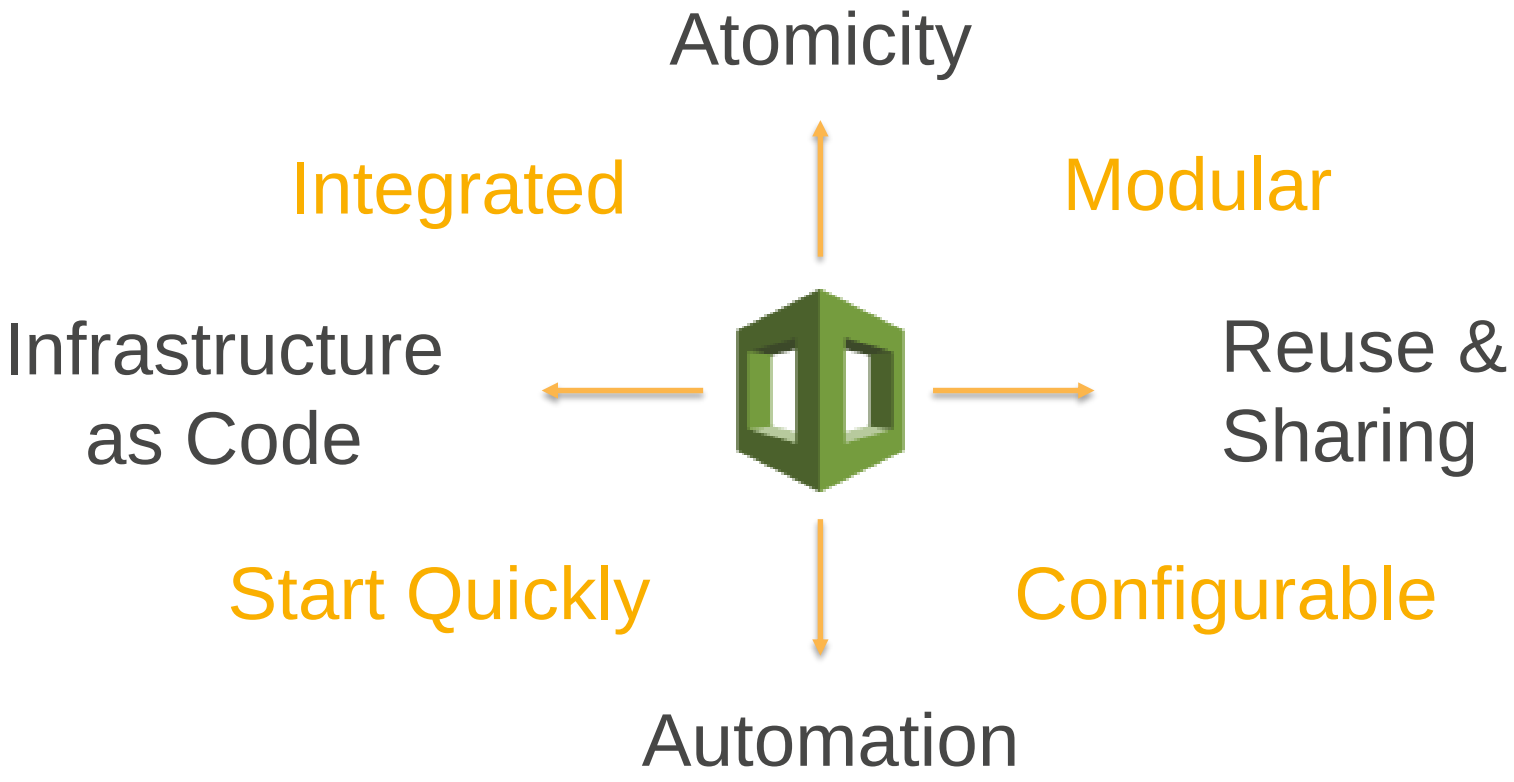
IAM Users

…

Anatomy of a CloudFormation Template



Key Benefits



Usecases

Go Global

Continuous Delivery

Infrastructure as Code

Cost Allocation

Demos

Software Evaluation

Test Automation

Complex Enterprise SW

Trainings

Load Testing

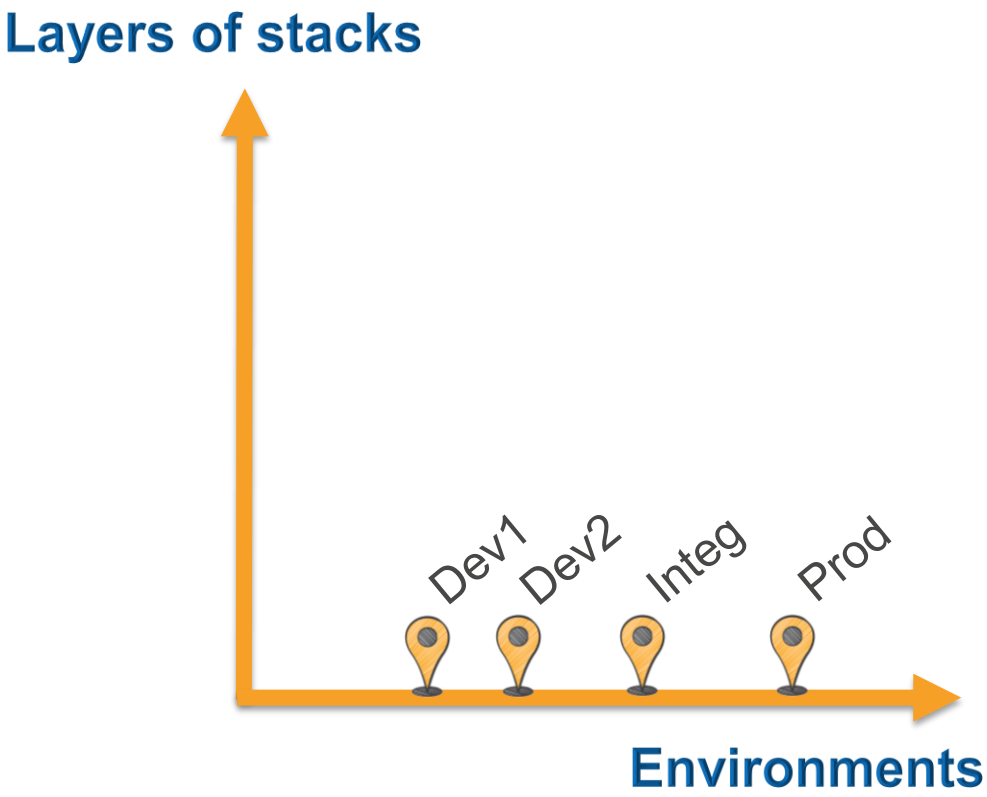
VPC Configuration

How to plan my stacks?

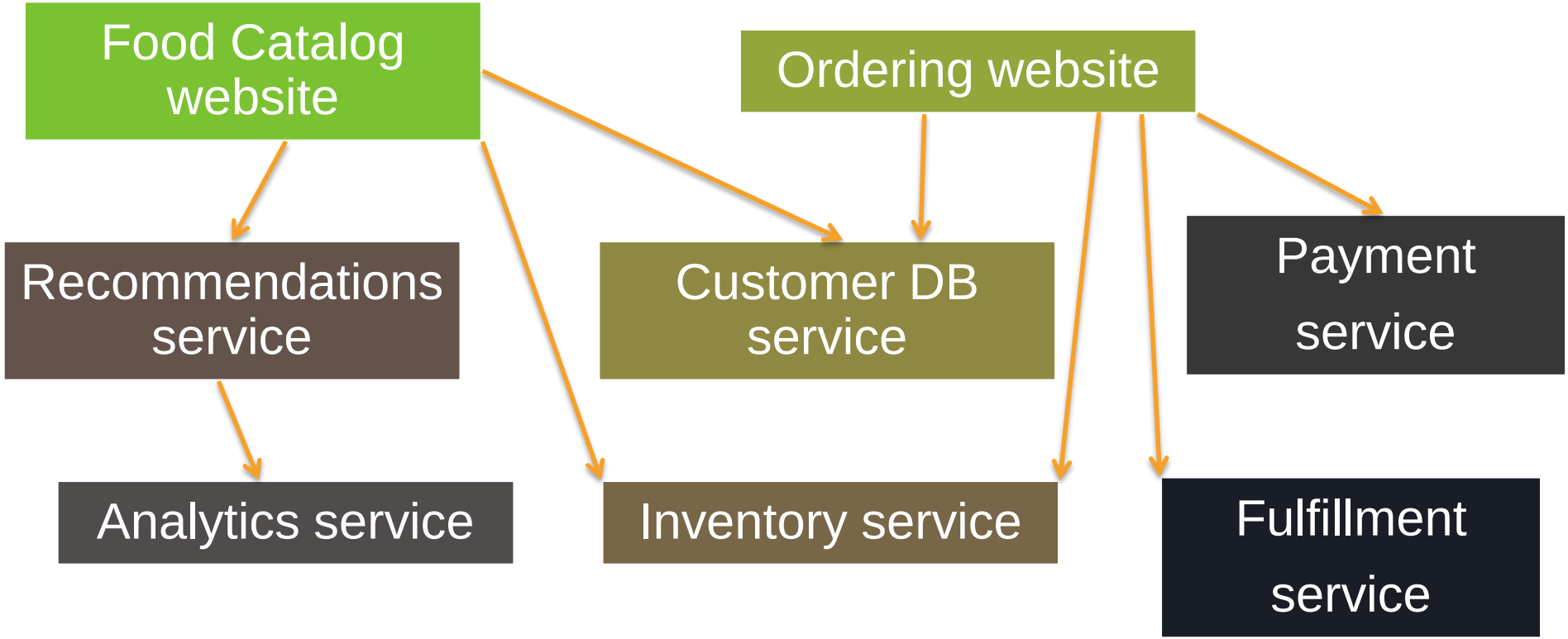
Organize by Layers

Frontend Services	• Consumer Website, Seller Website, Mobile Backend
Backend Services	• Search, Payments, Reviews, Recommendations
Shared Services	• CRM DBs, Common Monitoring /Alarms, Subnets, Security Groups
Base Network	• VPCs, Internet Gateways, VPNs, NATs
Identity	• IAM Users, Groups, Roles

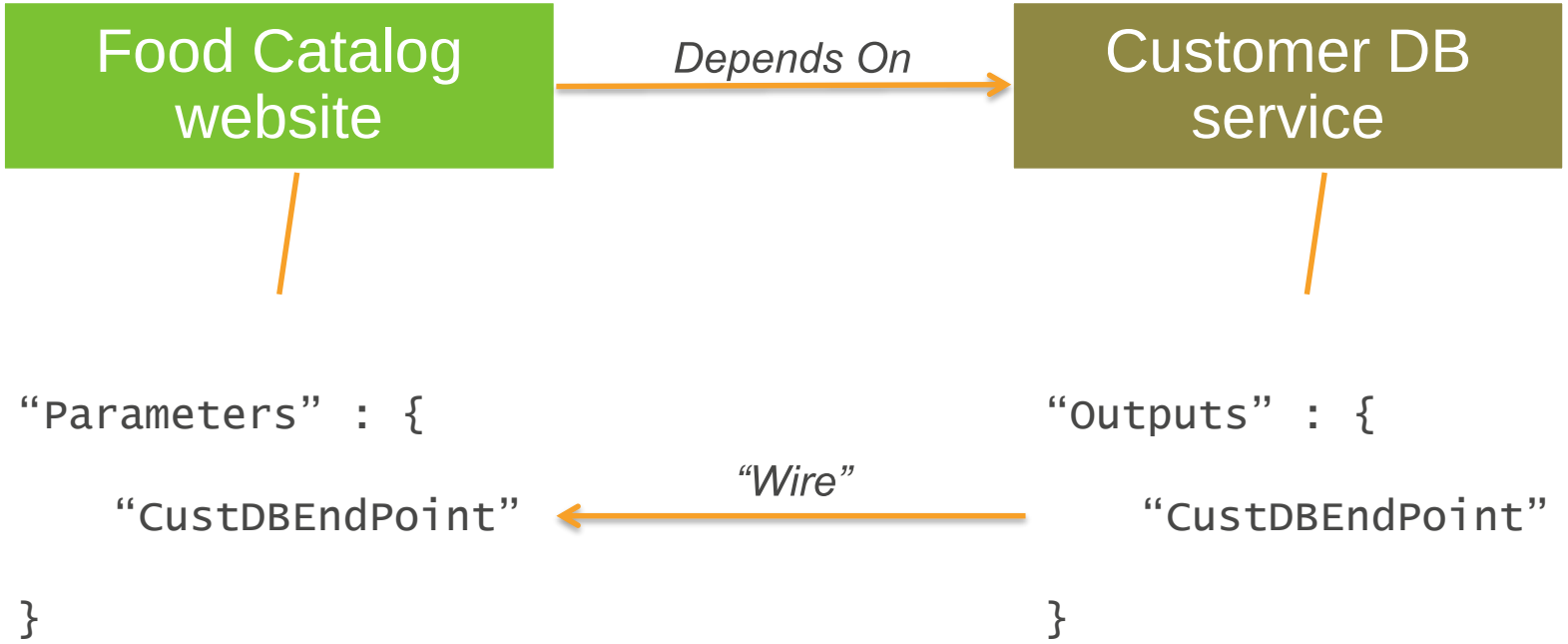
Organize by Environments



Think Services & Decouple



Think Services & Decouple



Reuse

Website1

```
“Resources” : {
    “ELB”,
    “AutoScaling”,
    “RDS”
}
```

Website2

```
“Resources” : {
    “ELB”,
    “AutoScaling”,
    “DynamoDB”
}
```

Reuse

Website1

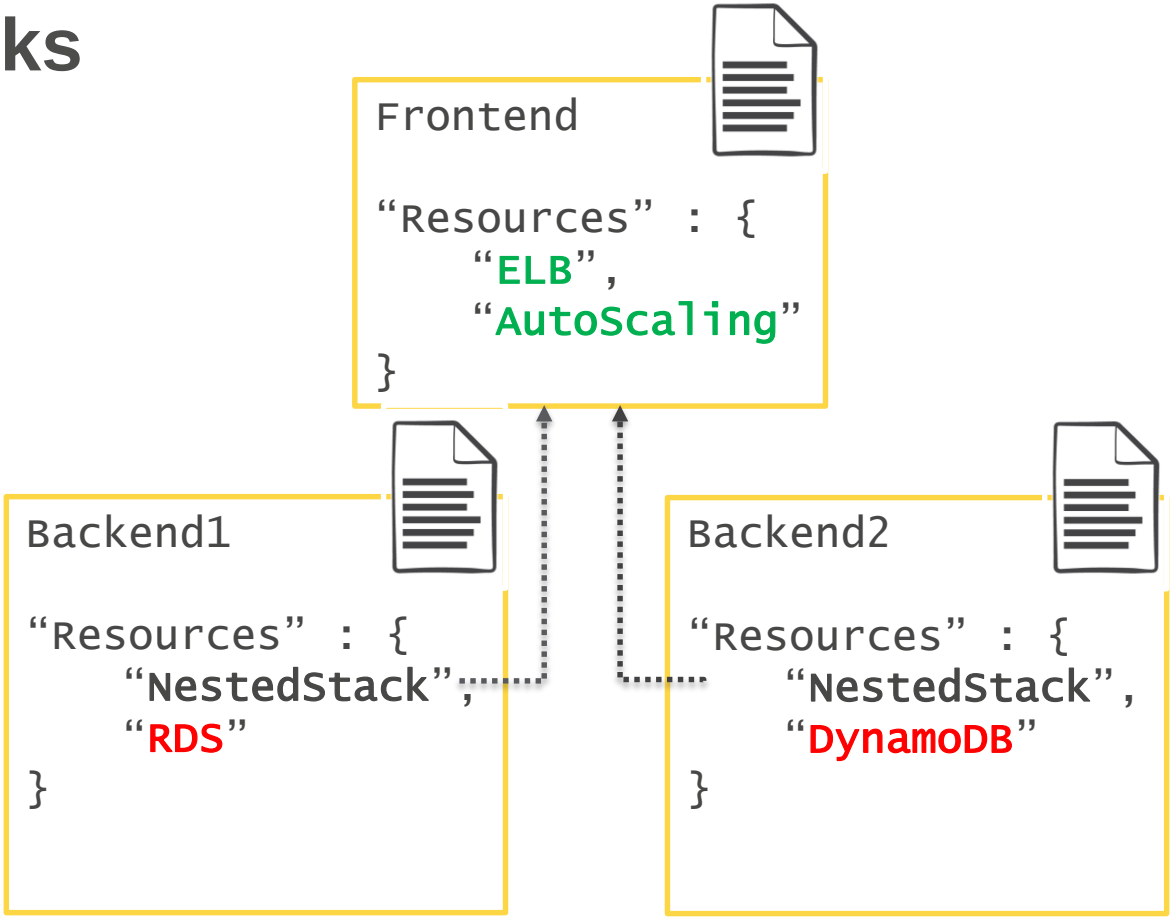
```
"Resources" : {
    "ELB",
    "AutoScaling",
    "RDS"
}
```

Website2

```
"Resources" : {
    "ELB",
    "AutoScaling",
    "DynamoDB"
}
```

Nested stacks

Reuse



Nested stacks

Role Specialization

Frontend

```
"Resources" : {
    "ELB",
    "AutoScaling"
}
```

Backend1

```
"Resources" : {
    "NestedStack",
    "RDS"
}
```

Backend2

```
"Resources" : {
    "NestedStack",
    "DynamoDB"
}
```

How to get started?

Start with Existing Template

Amazon Virtual Private Cloud

Template Name	Description	View	View in Designer	Launch
A single Amazon EC2 in an Amazon VPC	Creates a VPC and adds an Amazon EC2 instance with an Elastic IP address and a security group.	View	View in Designer	Launch Stack
Amazon VPC with static routing to an existing VPN	Creates a private subnet with a VPN connection that uses static routing to an existing VPN endpoint.	View	View in Designer	Launch Stack
Autoscaling and load-balancing website in an Amazon VPC	Creates a load balancing, auto scaling sample website in an existing VPC.	View	View in Designer	Launch Stack
Amazon VPC with DNS and public IP addresses	Creates a VPC with DNS support and public IP addresses enabled.	View	View in Designer	Launch Stack
Publicly accessible Amazon EC2 instances that are in an Auto Scaling group	Creates a load balancing, autoscaling group with instances that are directly accessible from the Internet.	View	View in Designer	Launch Stack
Amazon EC2 with	Creates an Amazon EC2	View	View in	Launch Stack

On this page:

http://ow.ly/ZufOd

CloudFormer

Create Stack Cancel

SELECT TEMPLATE — SPECIFY PARAMETERS — ADD TAGS — REVIEW

AWS CloudFormation gives you an easier way to create a collection of related AWS resources (a stack) by describing your requirements in a template. To create a stack, fill in the name for your stack and select a template. You may chose one of the sample templates to get started quickly, or one of your own templates stored in S3 or on your local hard drive.

Stack Name:
MyCloudFormerStack

Template:

- (•) **Use a sample template**
  - CloudFormer - create a template from your existing resources
- ( ) **Upload a Template File**
- ( ) **Provide a Template URL**

[x] **Show Advanced Options**

Notifications (optional):
Amazon SNS Topic (no notification)
Creation Timeout (minutes): none
Rollback on Failure: (•) Yes ( ) No

Continue

amazon web services

AWS CloudFormer 0.20 (Beta)

Welcome to the AWS CloudFormation template creation utility. This utility helps you to create a CloudFormation template from the AWS resources currently running in your account using a few simple steps. While the created template is complete and can be u launch an AWS CloudFormation stack, it is a starting point for further customization. You should consider the following:

- Add Parameters to enable stacks to be customized at launch time.
- Add Mappings to allow the template to be customized to the specific environment.
- Replace static values with "Ref" and "Fn::GetAtt" functions to flow property data between resources where the value of one property is dependent on the value of a property from a different resource.
- Use CloudFormation metadata and on-host helper scripts to deploy files, packages and run commands on your Amazon EC2 instances.
- Customize your Amazon RDS DB instance database names and master passwords.
- Customize or add more Outputs to list important information needed by the stack user.

Select the AWS Region US East (Virginia)

When you press "Create Template" we will analyze all of the AWS resources in your account. This may take a little time.

Create Template

What's New?

- Support for Amazon VPC resources.
- Support Amazon CloudWatch Alarms, Amazon DynamoDB, Amazon ElastiCache and Amazon SNS.
- Support Amazon S3 Bucket Policies, Amazon SQS Queue Policies and Amazon SNS Topic Policies.
- Updates for Route53 and CloudFront.
- Miscellaneous updates and bug fixes.

Known Issues

- Amazon RDS database instances in a VPC are not currently associated with VPC security groups. You will need to manually add these to your template once it is created.

For more information on how to build a template see the AWS CloudFormation User Guide. You can also check out our sample templates demonstrating various template features.

By default, the account credentials will be used from the entries you typed in when AWS CloudFormer was created, however, they can be overridden by clicking here.

Amazon EC2 Elastic IP Addresses

Amazon EC2 Instances

[ ] Select/Deselect all Amazon EC2 Instances
[x] [redacted]
[ ] [redacted]

Template Name cloudformer.template S3 Bucket [redacted]

Save Template Cancel


```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "eip17412919140": {
      "Type": "AWS::EC2::EIP",
      "Properties": {
        "InstanceId": {
          "Ref": "instanceib47950da"
        }
      }
    },
    "instanceib47950da": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
```

Pick an IDE

It’s JSON!
=> Emacs, notepad, vi

Code Generators

cf_factory 0.0.5

Cf-factory is a Ruby library to generate CloudFormation templates.

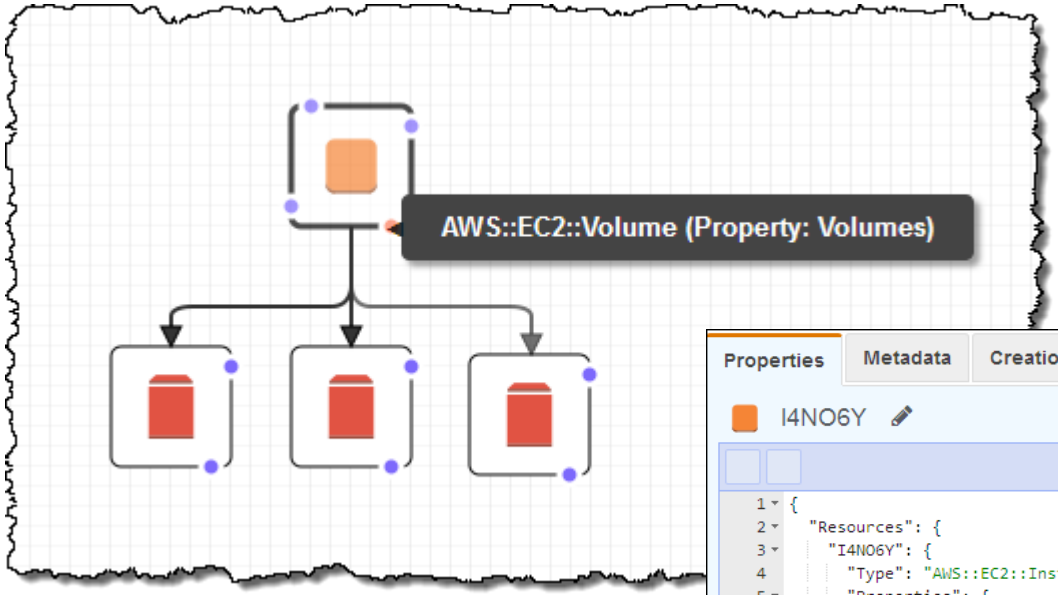
```
INSTALL > gem install cf_factory
```

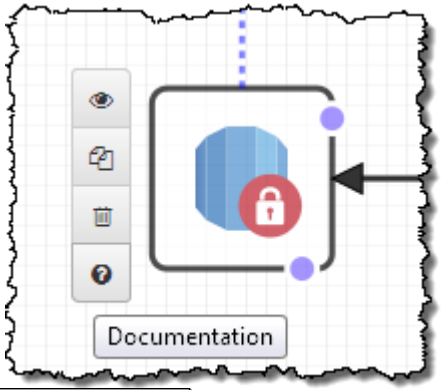
GitHub This repository Search Explo

cloudtools / troposphere

troposphere - Python library to create AWS CloudFormation descriptions

CloudFormation Designer





Properties | Metadata | CreationPolicy | DeletionPolicy | DependsOn

I4NO6Y

```
1  {
2    "Resources": {
3      "I4NO6Y": {
4        "Type": "AWS::EC2::Instance",
5        "Properties": {
6          "InstanceType" : "t2.medium",
7          "DisableApiTermination" : "true",
8          "Monitoring" : "true",
9          "Volumes": [
10           {
11             "VolumeId": {
12               "Ref": "VOL10ZQL"
13             }
14           },
15           {
16             "VolumeId": {
```

How to prevent errors?

Add Comments

```
{
   "Description" : "This is a sample template.",

   "Resources" : {
      "Bucket98004" : {
         "Type" : "AWS::S3::Bucket",
         "Metadata" : {
            "Comment" : “Image bucket for ZIP code 98004",
            "Version" : "1.2.1_1“
...
}
```

Validate your Templates

- JSON Syntax
- Circular Dependencies
- Template Structure

validate-template

Description

Validates a specified template.

Synopsis

```
  validate-template
[--template-body <value>]
[--template-url <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton]
```

Use Parameter Types

```
"Parameters" : {
    “aVpcId" : {
        "Type" : "AWS::EC2::VPC::Id"
    },
    “bSubnetIds" : {
        "Type" : "List<AWS::EC2::Subnet::Id>"
    },
    “cSecurityGroups" : {
        "Type" : "List<AWS::EC2::SecurityGroup::Id>"
    }
```

Use Parameter Types

Specify Parameters

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

aVpc vpc-ea814e8f (10.0.0.0/16) Vpcld of your existing Virtual Private Cloud (VPC)

bSubnets

- subnet-e85150ae
- subnet-f89b419d
- subnet-70b1ef36
- subnet-807791f7

The list of SubnetIds in your Virtual Private Cloud (VPC)

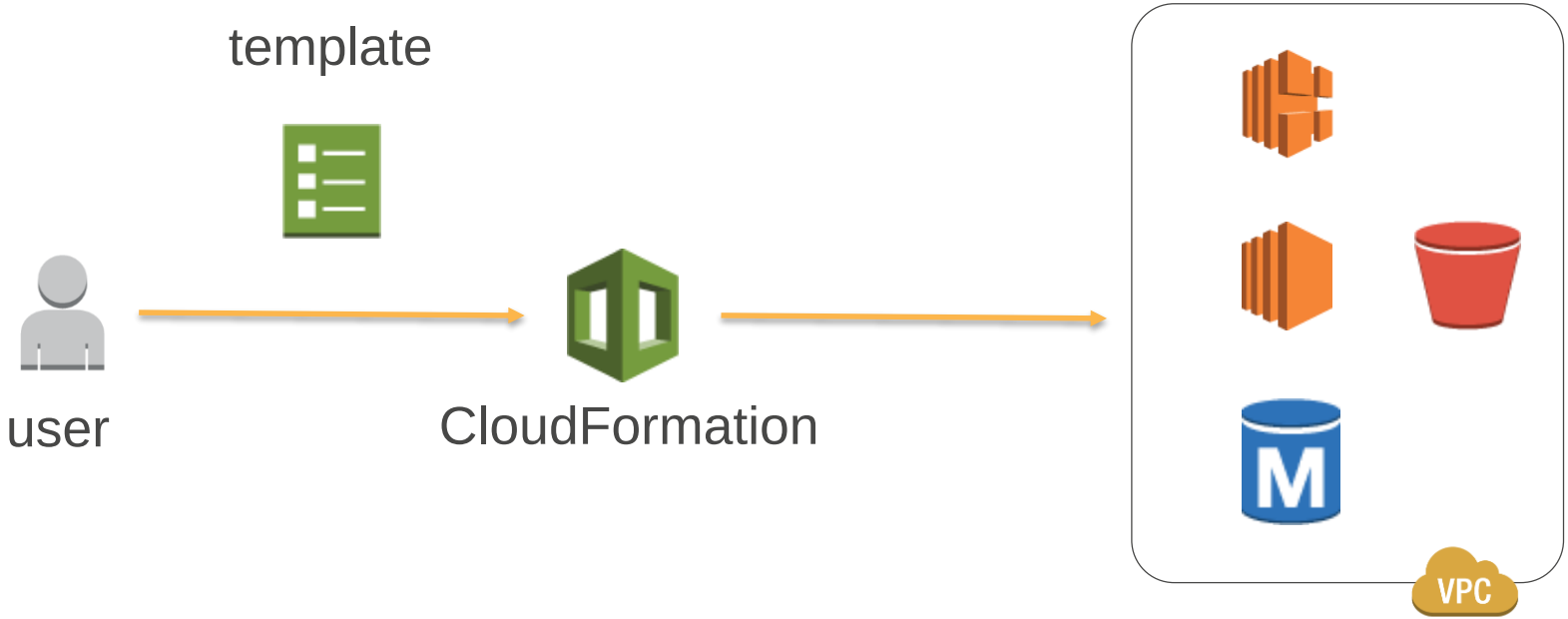
cSecurityGroups

- web4-WebServerSecurityGroup-1D7AQ98RDR3SK
- web1-WebServerSecurityGroup-JJXAZ2723AL9
- sp6-SharePointFoundationSecurityGroup-O788PH7WGC34

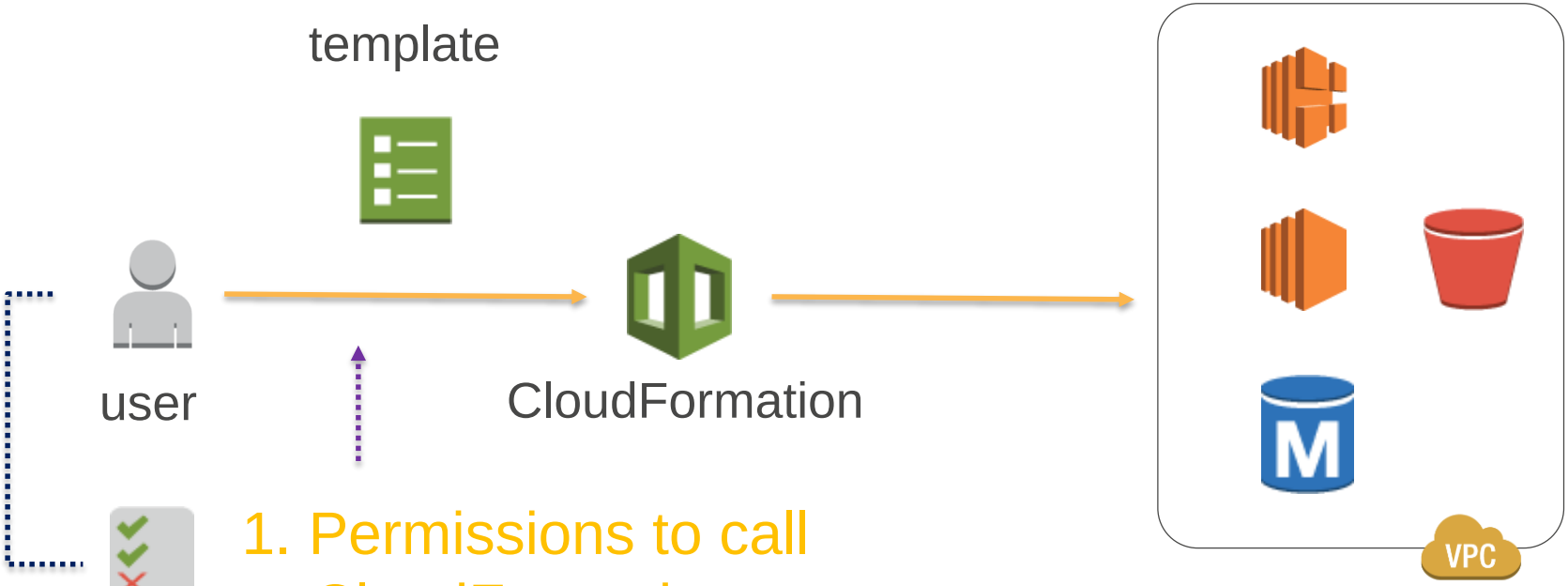
Use Parameter Constraints

```
"Parameters" : {
    "SourceCIDRForRDP" : {
        "Description" : "CIDR block to allow RDP from",
        "Type" : "String",
        "MinLength" : "9",
        "MaxLength" : "18",
        "AllowedPattern" : "^([0-9]+\\.){3}[0-9]+\\/[0-9]+$"
}
```

Check IAM Permissions

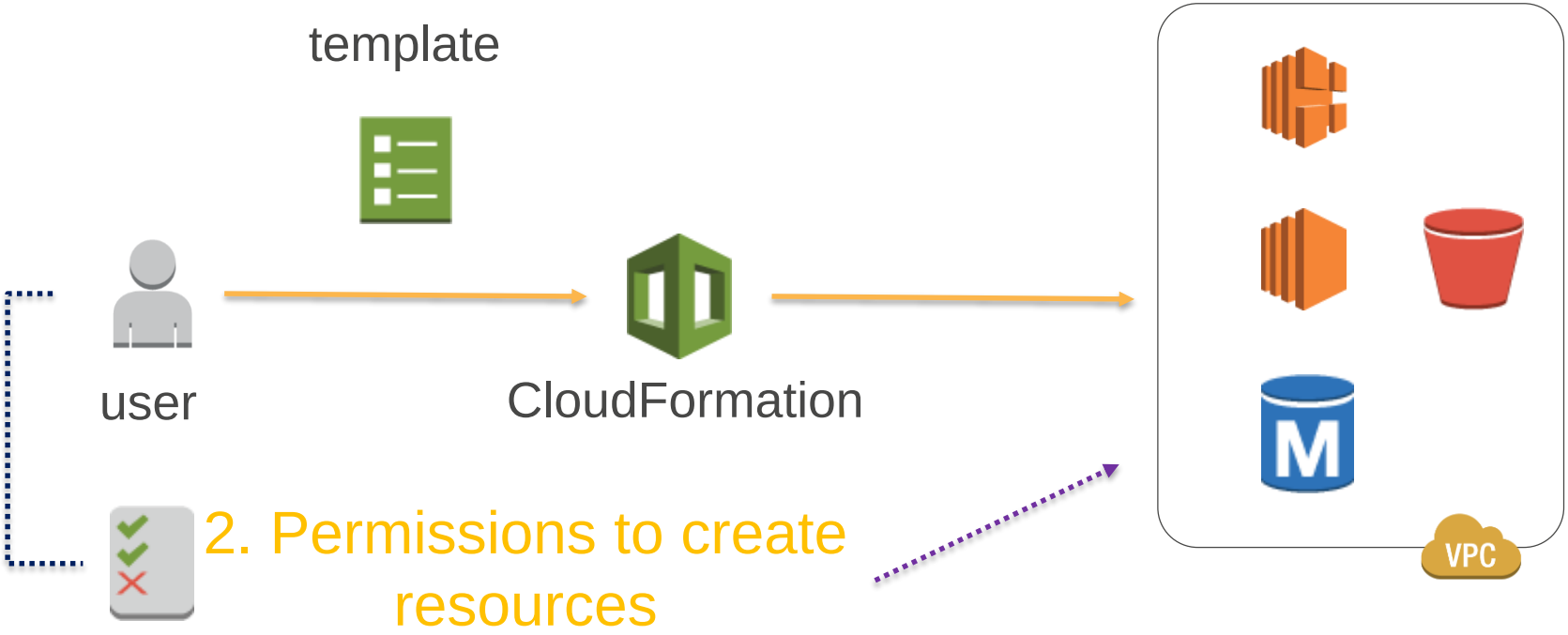


Check IAM Permissions



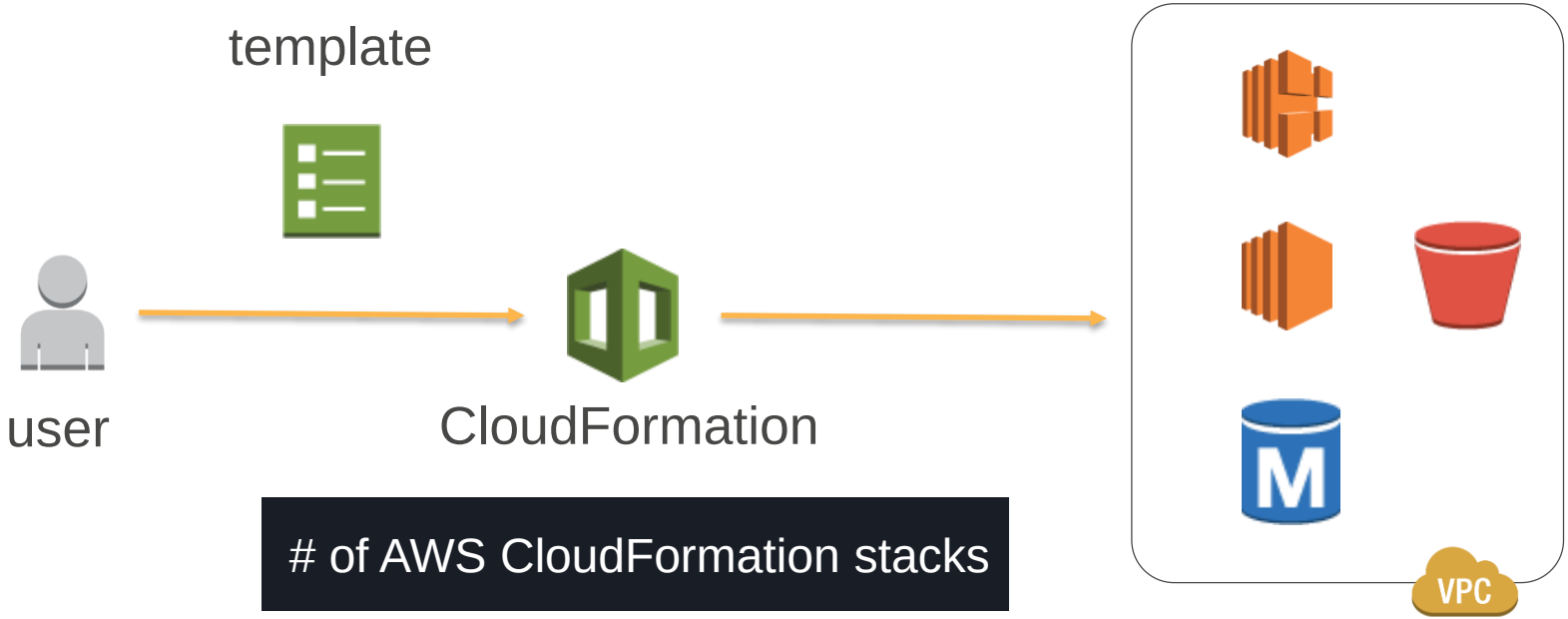
1. Permissions to call CloudFormation

Check IAM Permissions



Check AWS Limits

of EC2, RDS, EBS IOPS, etc.



# of AWS CloudFormation stacks

How to debug?

View Events

Overview | Outputs

2015-04-28

- 10:44:28 UTC+1000
- 10:44:25 UTC+1000
- 10:43:39 UTC+1000
- 10:43:37 UTC+1000
- 10:43:36 UTC+1000
- 10:43:35 UTC+1000
- 10:43:19 UTC+1000
- 10:42:57 UTC+1000

```
PROMPT> aws cloudformation describe-stack-events --stack-name myteststack
{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/myteststack/466
            "EventId": "af67ef60-0b8f-11e3-8b8a-500150b352e0",
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2013-08-23T01:02:30.070Z",
            "StackName": "myteststack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-east-1:123456789012:stack/myte
            "LogicalResourceId": "myteststack"
        },
```

on

ation Initiated

ation Initiated

Debugging Tips

- Deactivate Rollback Flag during tests
- Put “breakpoints” via WaitConditions
- Test user data & scripts separately, e.g. Moustache
- Log stack events in DWH or logging service
- Use CloudTrail and AWS Config to track changes
- Redirect local Cfn log files to CloudWatch Logs

Use CloudWatch Logs for Debugging

Services Edit

Dashboard
Alarms
ALARM 0
INSUFFICIENT 0
OK 0
Billing
Logs
Metrics
Selected Metrics
DynamoDB
EBS
EC2
RDS
SNS
Custom Metrics...

Log Groups > Streams for my-web-app-stack-6-CloudFor... > Events for i-e1deaccd/cfn-init.log

Jump To: 2014/07/19 00 : 05 : 05 Local (GMT-07:00)

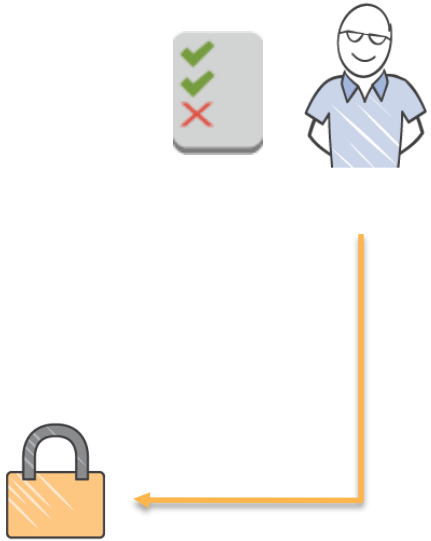
Creation Time	Event Data
2014-07-19 00:05:05 UTC-7	2014-07-19 07:04:19,885 [INFO] Running configSets: default
2014-07-19 00:05:05 UTC-7	2014-07-19 07:04:19,886 [INFO] Running configSet default
2014-07-19 00:05:05 UTC-7	2014-07-19 07:04:19,963 [INFO] Running config cfn-logs-config
2014-07-19 00:05:05 UTC-7	2014-07-19 07:05:02,297 [INFO] Command install-logs-agent succeeded
2014-07-19 00:05:05 UTC-7	2014-07-19 07:05:02,342 [INFO] Running config cfn-hup-config
2014-07-19 00:05:05 UTC-7	2014-07-19 07:05:02,437 [INFO] enabled service cfn-hup
2014-07-19 00:05:05 UTC-7	2014-07-19 07:05:03,638 [INFO] Restarted cfn-hup successfully
2014-07-19 00:05:05 UTC-7	2014-07-19 07:05:03,704 [INFO] Running config application-config
2014-07-19 00:05:14 UTC-7	2014-07-19 07:05:14,092 [INFO] Yum installed [u'httpd', u'php']
2014-07-19 00:05:14 UTC-7	2014-07-19 07:05:14,243 [INFO] enabled service httpd
2014-07-19 00:05:15 UTC-7	2014-07-19 07:05:14,603 [INFO] Started httpd successfully
2014-07-19 00:05:15 UTC-7	2014-07-19 07:05:14,846 [INFO] disabled service sendmail
2014-07-19 00:05:15 UTC-7	2014-07-19 07:05:14,974 [INFO] ConfigSets completed

ow.ly/E0zO3

How to protect running stacks?

Protect Stacks from Unintended Changes

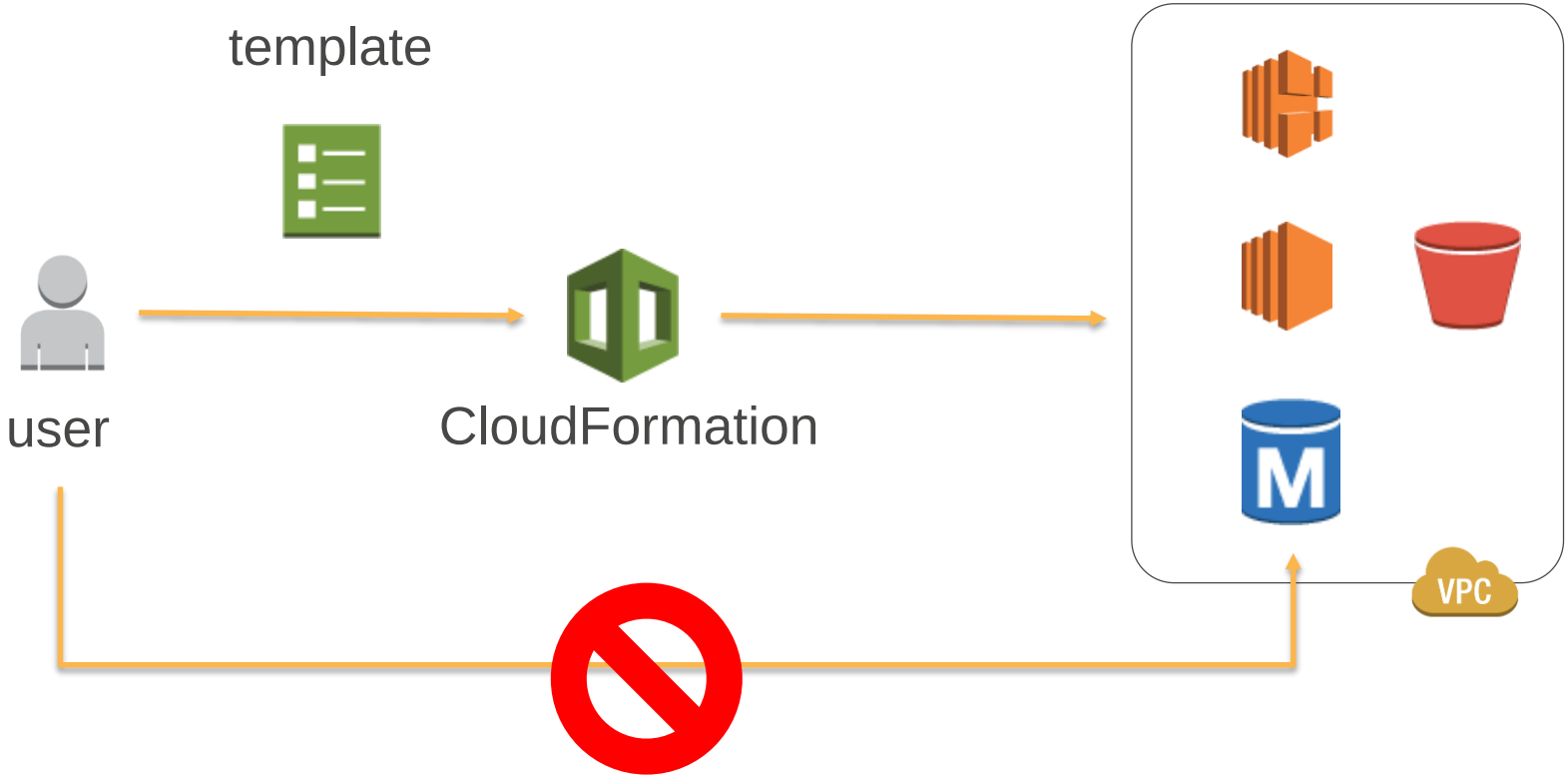
Frontend Services	• Consumer Website, Seller Website, Mobile Backend
Backend Services	• Search, Payments, Reviews, Recommendations
Shared Services	• CRM DBs, Common Monitoring /Alarms, Subnets, Security Groups
Base Network	• VPCs, Internet Gateways, VPNs, NATs
Identity	• IAM Users, Groups, Roles



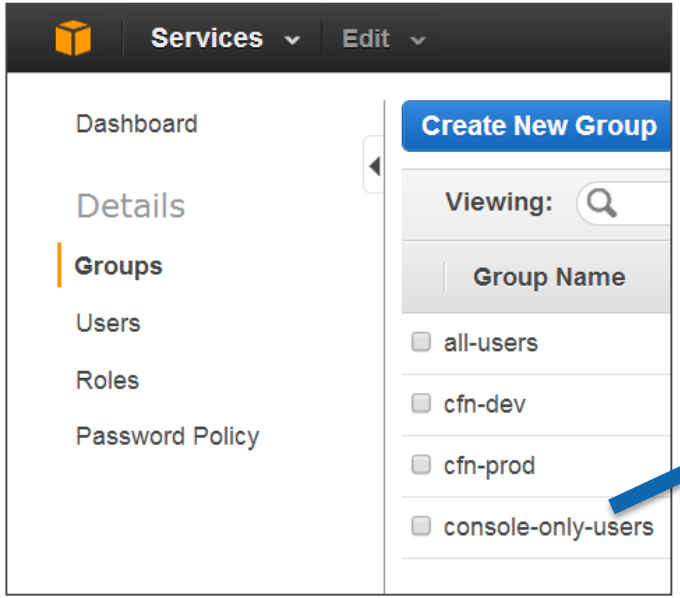
Protect Stacks from Unintended Changes

```
{
  "Effect" : “Deny",
  "Action" : [ "cloudformation:*" ],
  "Resource" :
   "arn:aws:cloudformation:us-west-2:123456789:stack/BaseNet*"
}
{
  "Effect" : "Allow",
  "Action" : [ "cloudformation:*" ],
  "Resource" :
   "arn:aws:cloudformation:us-west-2:123456789:stack/FrontEnd*"
}
```

Protect Stacks from Drift

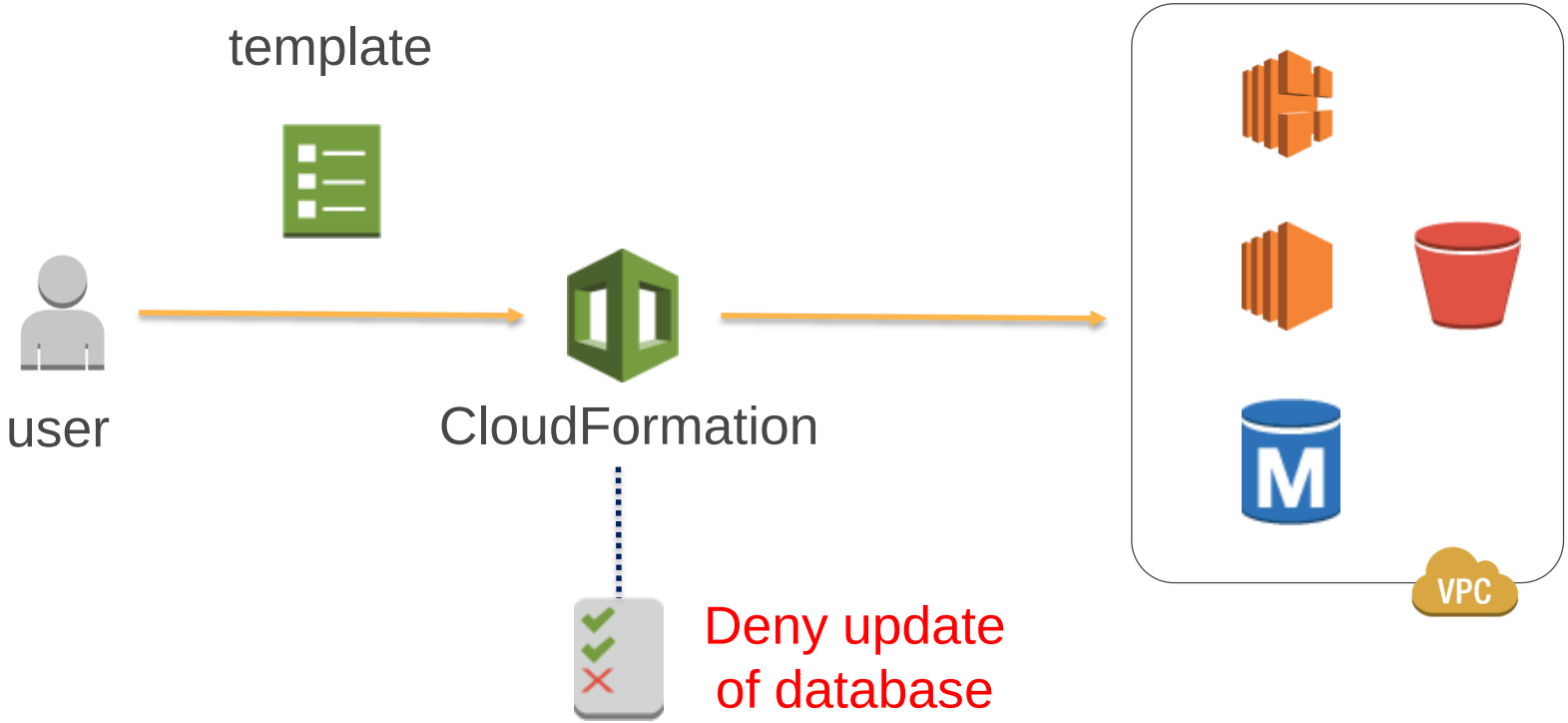


Protect Resources with IAM and Tags

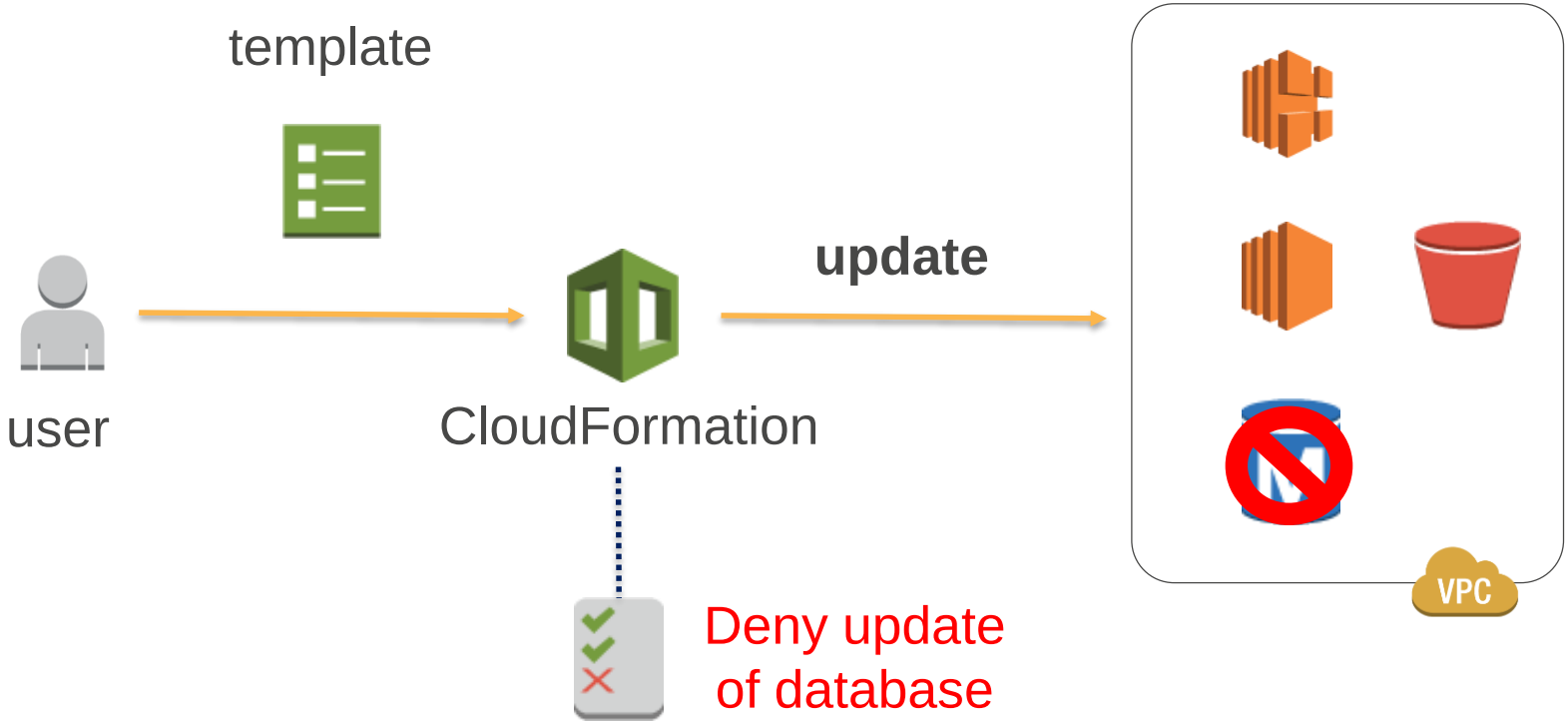



```
{
  "Effect" : "Deny",
  "Action" : [
    "ec2:TerminateInstances"
 ],
  "Condition": {
    "Null": {
"ec2:ResourceTag/*cloudformation*" :
"true" }
  },
  "Resource" : "*"
}
```

Stack Policies Against Unwanted Updates

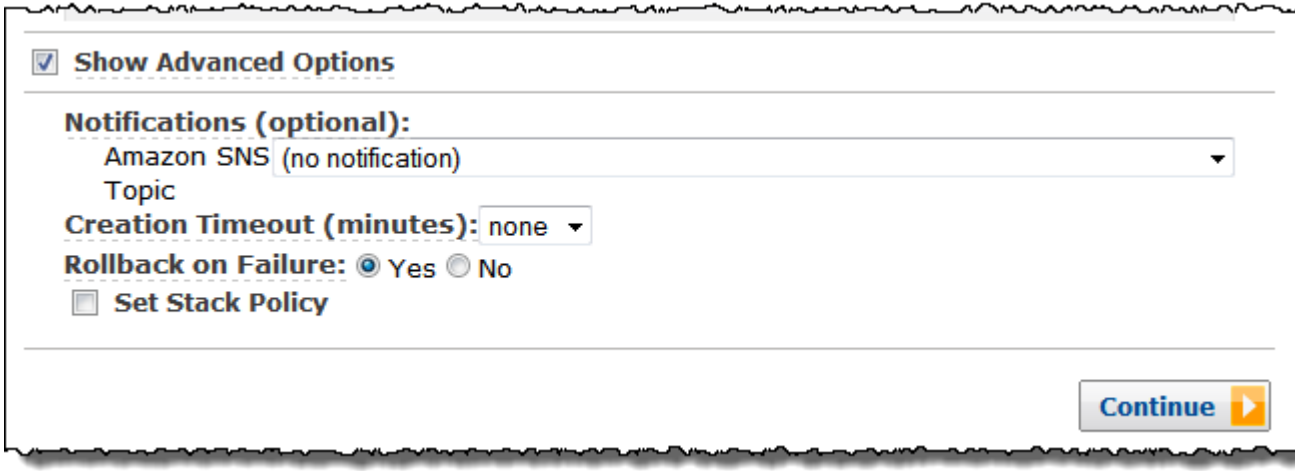


Stack Policies Against Unwanted Updates



Stack Policies Against Unwanted Updates

```
{"Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*"
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```



Stack Policies Against Unwanted Updates

“Do not update the databases”

```
"Effect" : "Deny",
"Principal" : "*",
"Action" : "Update:*",
"Resource" : "*",
"Condition" : {
    "StringEquals" : {
        "ResourceType" :

["AWS::RDS::DBInstance",
    "AWS::Redshift::Cluster"]
    }
```

“Okay to update, unless the update requires replacement”

```
"Effect" : "Deny",
"Principal": "*",
"Action" : "Update:Replace",
"Resource" :
"LogicalResourceId/MyInstance"
```

ow.ly/E0Njj

How to safely update stacks?

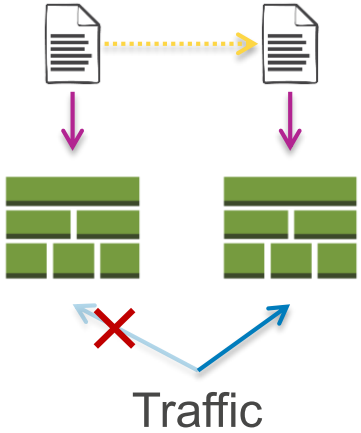
Choose an Update Style

In-place

Templates

Stacks

Blue-Green



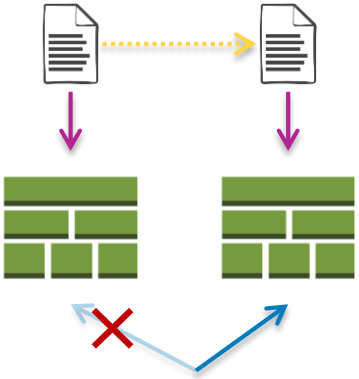
Choose an Update Style

In-place

Blue-Green

Templates

Stacks



Fast, Simple &
Cost Efficient

Robust

Review Updates

What is going to be updated?

- Run Textual Diffs
- Pay attention to impact on Related Resources
    - Ref and Get:Att
- Check for Update Mode
    - No Interruption
    - Some Interruption
    - Replacement
- Check for Drift

Review Updates

What is going to be updated?

- Preview Feature with Change Sets New
- Pay attention to impact on Related Resources
    - Ref and Get:Att
- Check for Update Mode
    - No Interruption
    - Some Interruption
    - Replacement
- Check for Drift

Review Impact via Change Sets

Create Stack

Filter: Active

	Stack Name
☑	LAMP-Stack-3

AWS Services Edit Jeff Barr N. Virginia Support

CloudFormation > Stack: LAMP-Stack-3 > Change set detail: BigChanges

BigChanges

Other Actions Execute

Overview

ID	arn:aws:cloudformation:us-east-1:348414629041:changeSet/BigChanges/be420e82-85ae-4bca-ad76-60b8d34c290a
Description	Big changs for my stack
Created time	2016-03-23 19:09:39 UTC-0700
Status	CREATE_COMPLETE
Stack name	LAMP-Stack-3

Change set input

Changes

The changes CloudFormation will make if you execute this change set.

Filter

Action	Logical ID	Physical ID
Add	WebServerAutoScalingGroup	
Remove	WebServerInstance	i-d47ac24f
Add	WebServerLaunchConfig	
Add	WebSiteLoadBalancer	

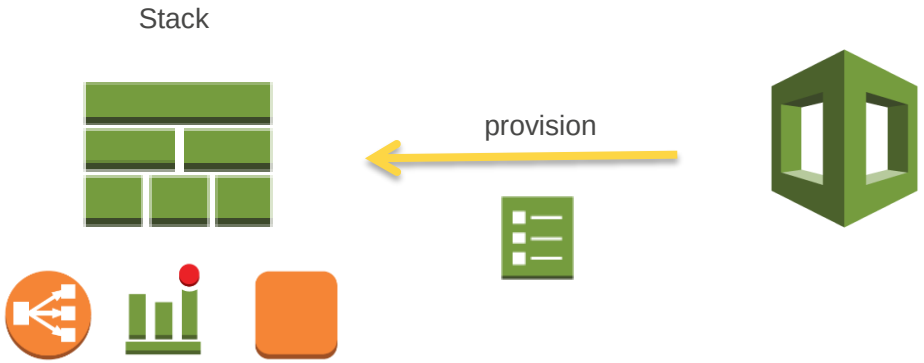
Execute change set

Are you sure you want to execute the **BigChanges** change set to update the **LAMP-Stack-3** stack?

Cancel Execute

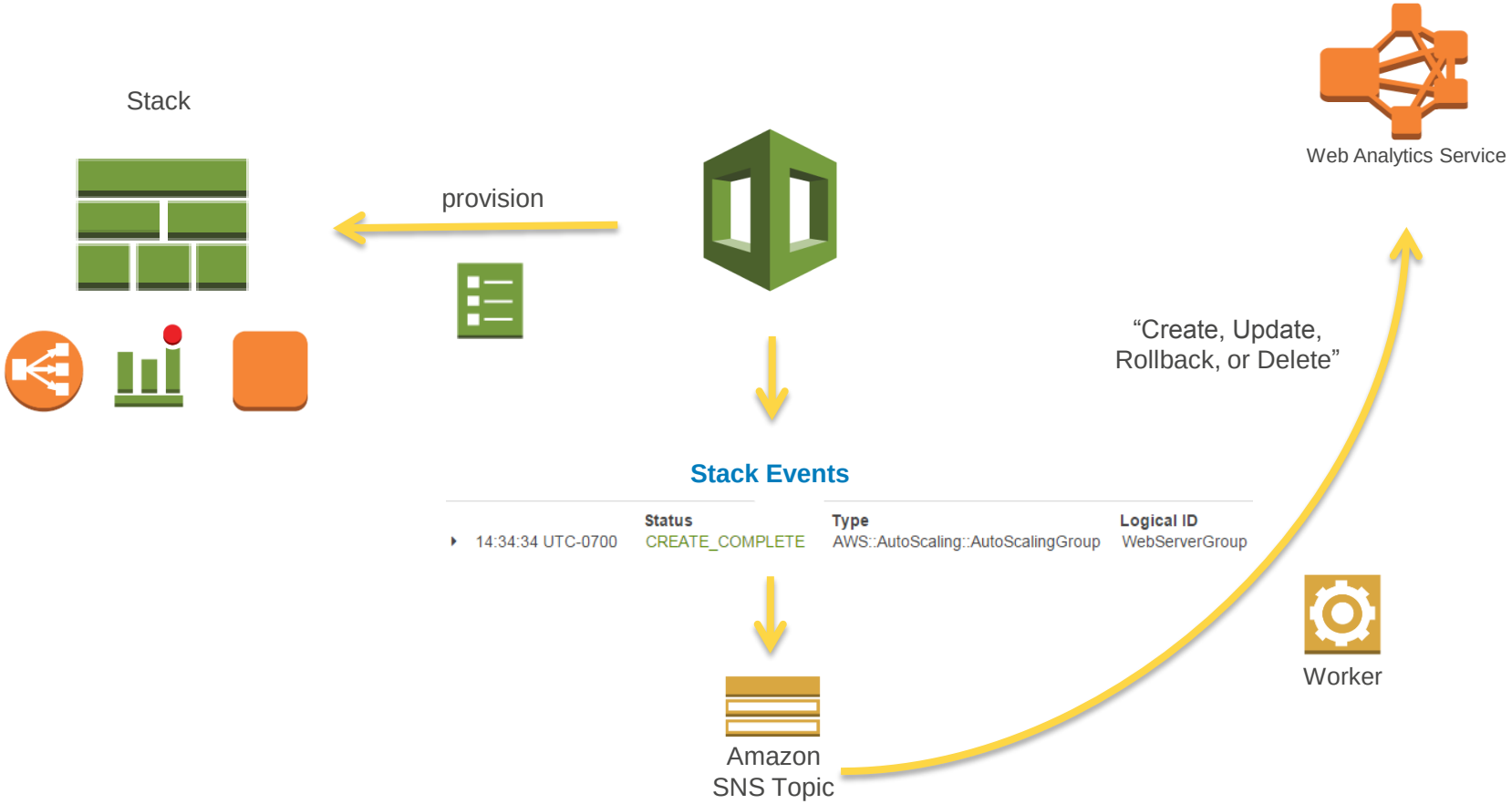
How to extend CloudFormation?

Extend with Stack Events



Web Analytics Service

Extend with Stack Events



Extend with Lambda Custom Resources

What you need:

- Lambda function talking with CloudFormation

```
//Sends response to the pre-signed S3 URL
function sendResponse(event, context, responseStatus, resp
   var responseBody = JSON.stringify({
        Status: responseStatus,
        Reason: "See the details in CloudWatch Log Stream
        PhysicalResourceId: context.logStreamName,
        StackId: event.StackId,
        RequestId: event.RequestId,
        LogicalResourceId: event.LogicalResourceId,
        Data: responseData
    });
```

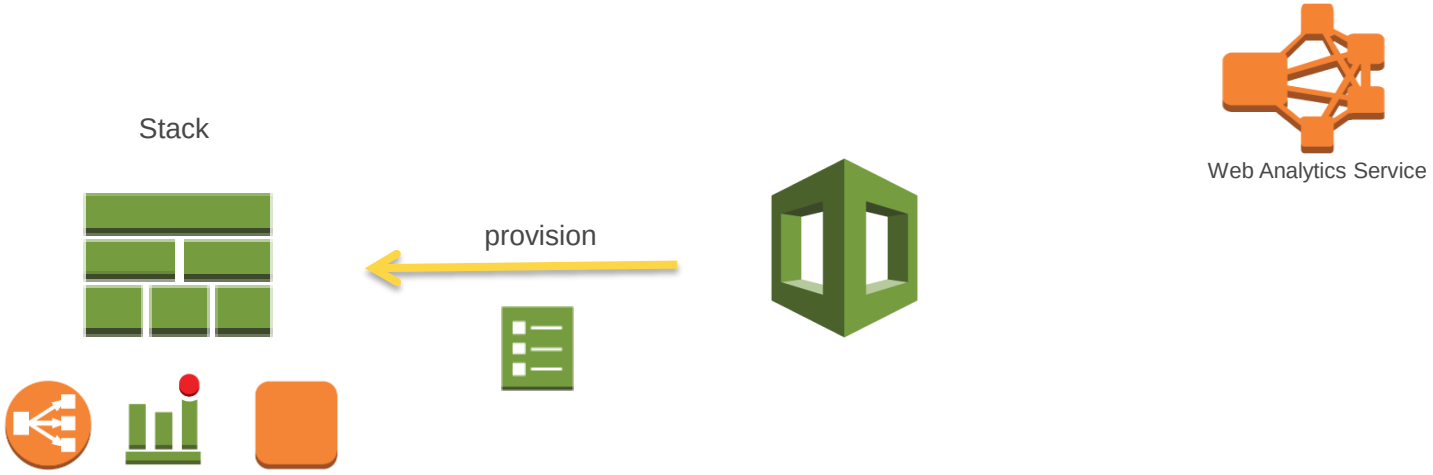
Extend with Lambda Custom Resources

What you need:

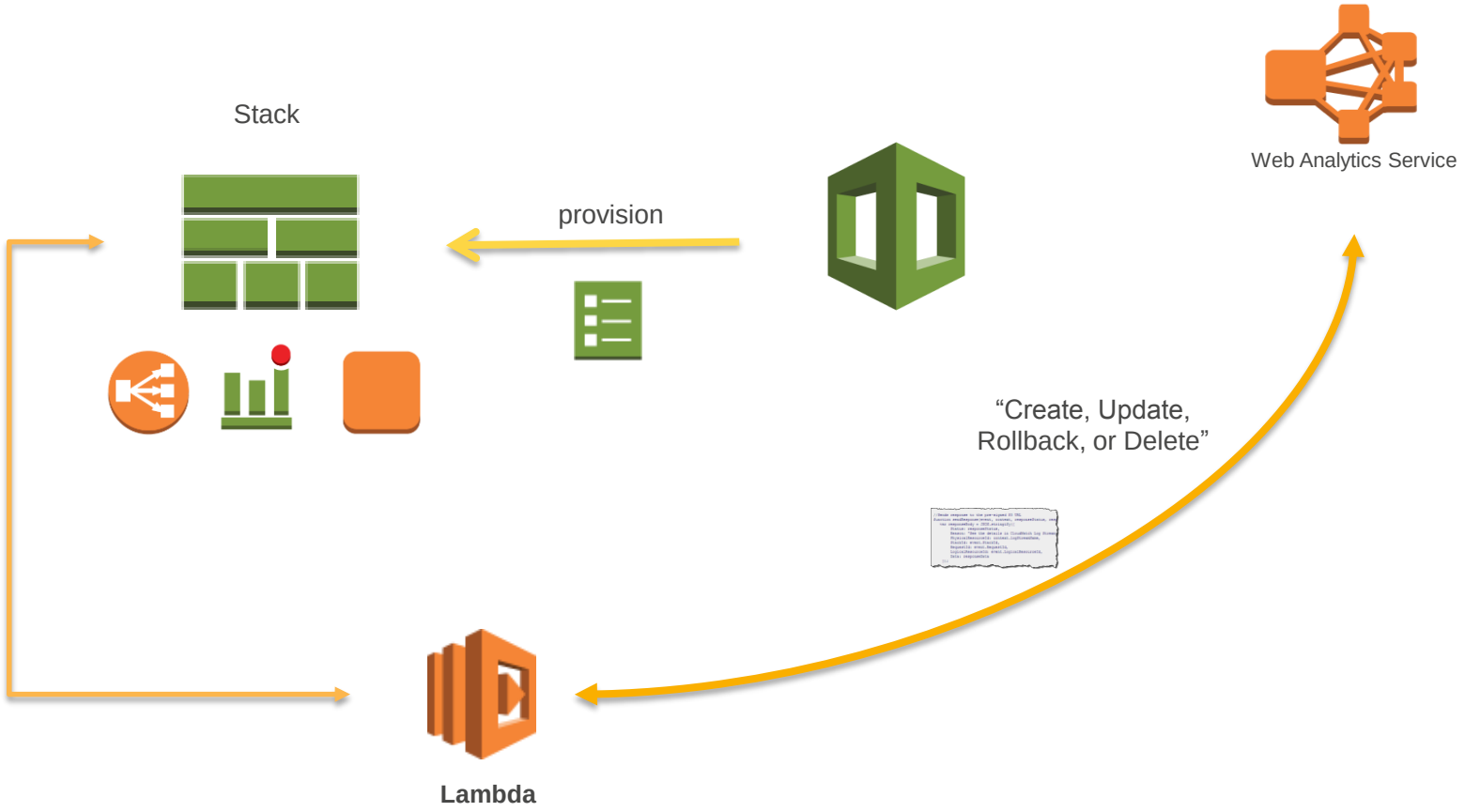
- Custom resource in CloudFormation

```
"MyCustomResource" : {
  "Type" : "Custom::TestLambdaCrossStackRef",
  "Properties" : {
    "ServiceToken": { "Fn::Join": [ "", [ "arn:aws:lambda:", { "Ref": "AWS::
    "StackName": {
      "Ref": "NetworkStackName"
    }
  }
}
```

Extend with Lambda Backed Custom Resources



Extend with Lambda Backed Custom Resources



Summary

- Why CloudFormation?
- How to plan my stacks?
- How to get started?
- How to prevent errors?
- How to safely update stacks?
- How to extend CloudFormation?

amazon web services