Choose a Topic

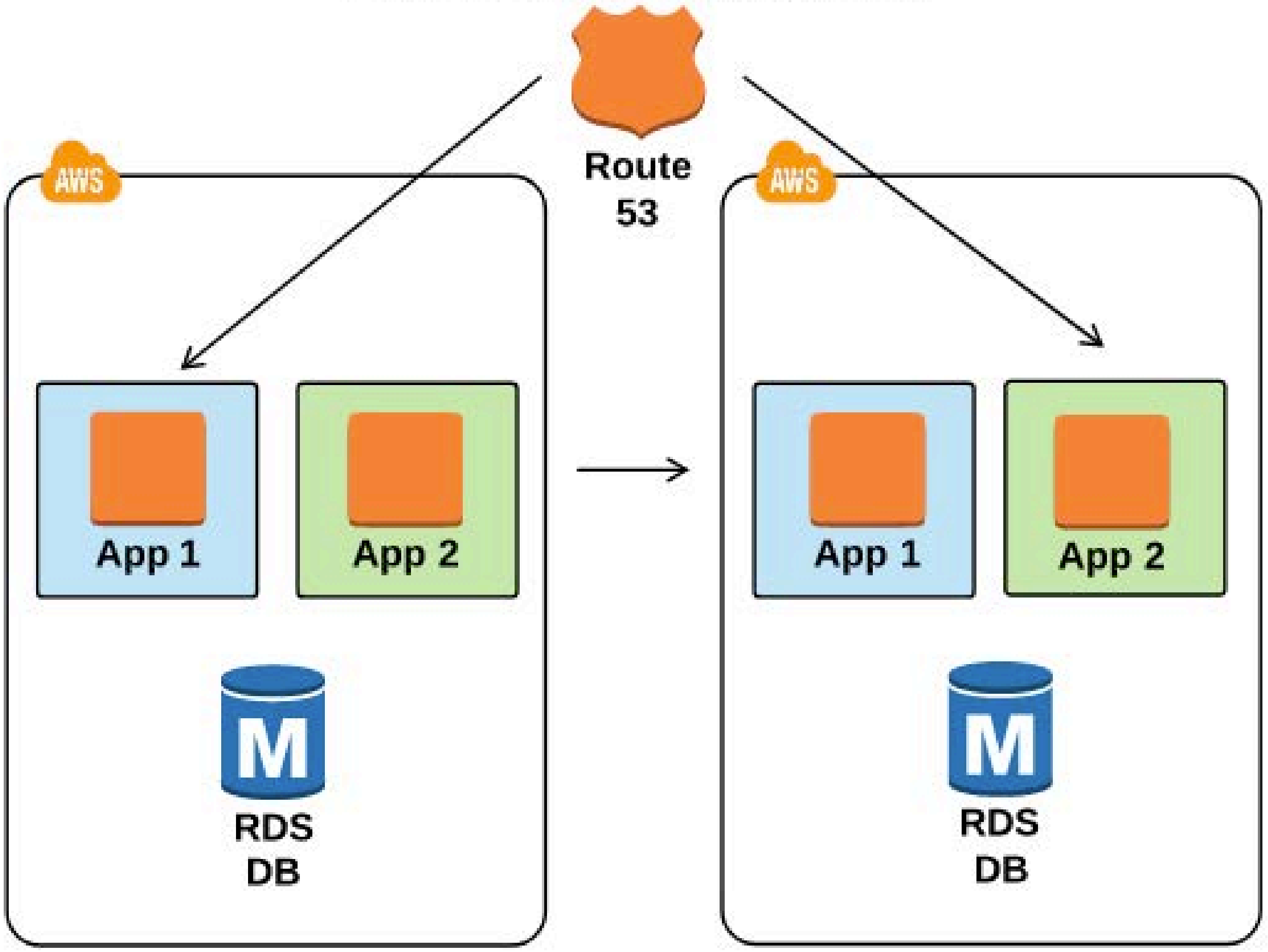
- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda

Blue/Green Deployments

Scenario: Blue/green deployment is a technique for releasing applications by shifting traffic between two identical environments running different versions of the application. Blue/green deployments can mitigate common risks associated with deploying software, such as downtime and rollback capability. The Devops Engineer will be required to properly configure Blue/Green environments in such a way that switching environments is seamless with no downtime.

- Update DNS Routing
- Docker B/G Deployments
- Swap ASG Behind ELB
- A/B Testing
- Update ASG Launch Config

Blue/Green Deployments

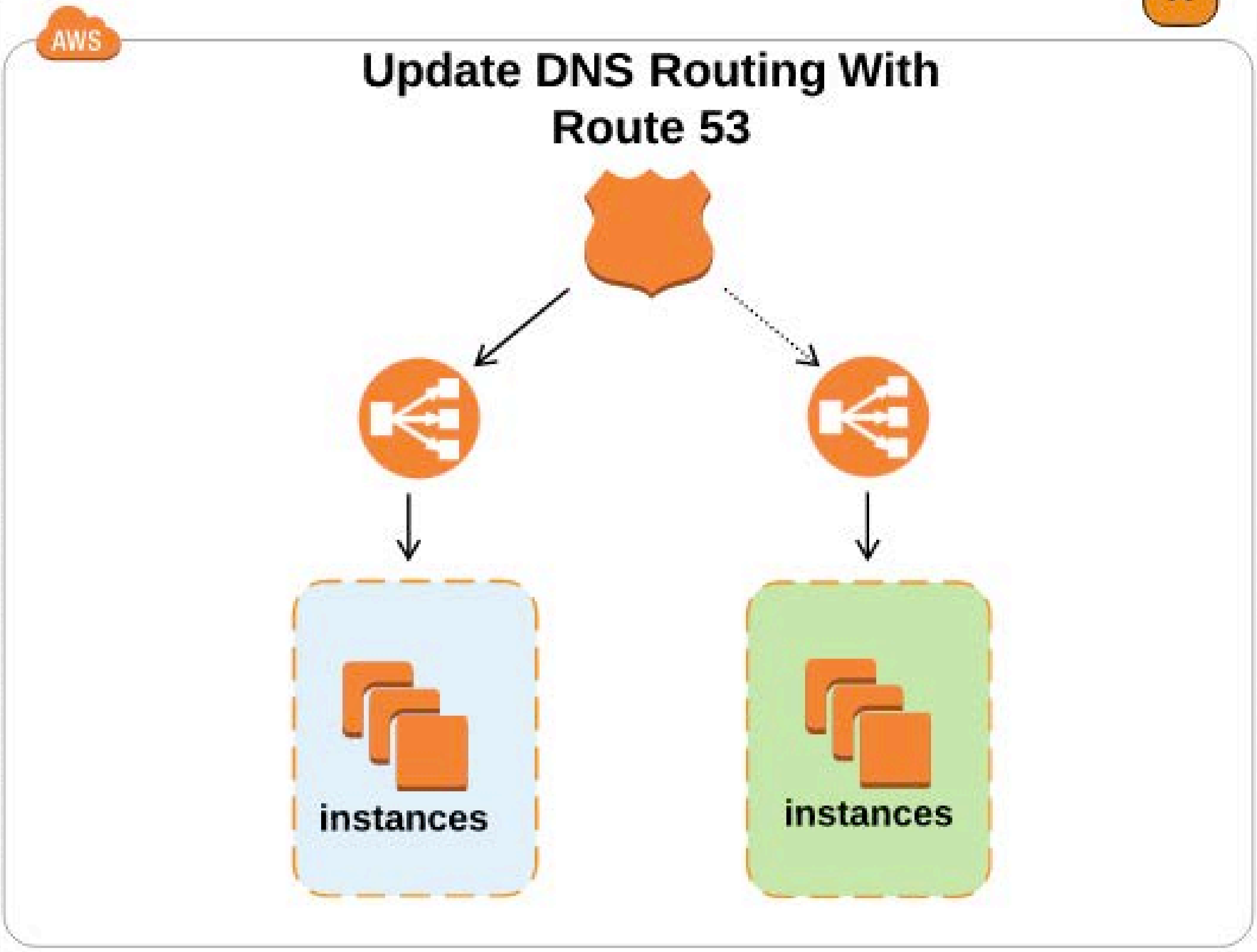


Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda

Blue/Green Deployments

X



Update DNS Routing With Route 53

Update DNS Routing With Route 53

- DNS is used to switch from Blue to Green (or back for a rollback).
- Use this pattern if you can express the endpoint into the environment as a DNS name or IP address.
- Example environments:
  - Single instances, with a public or Elastic IP address.
  - Groups of instances behind an Elastic Load Balancing load balancer.
  - Instances in an Auto Scaling Group in front of an ELB.
  - Services running on Elastic Container Service.
  - Elastic Beanstalk environment web tiers.
- Route 53 manages the DNS Hosted Zone. You can shift traffic all at once or do a weighted distribution.

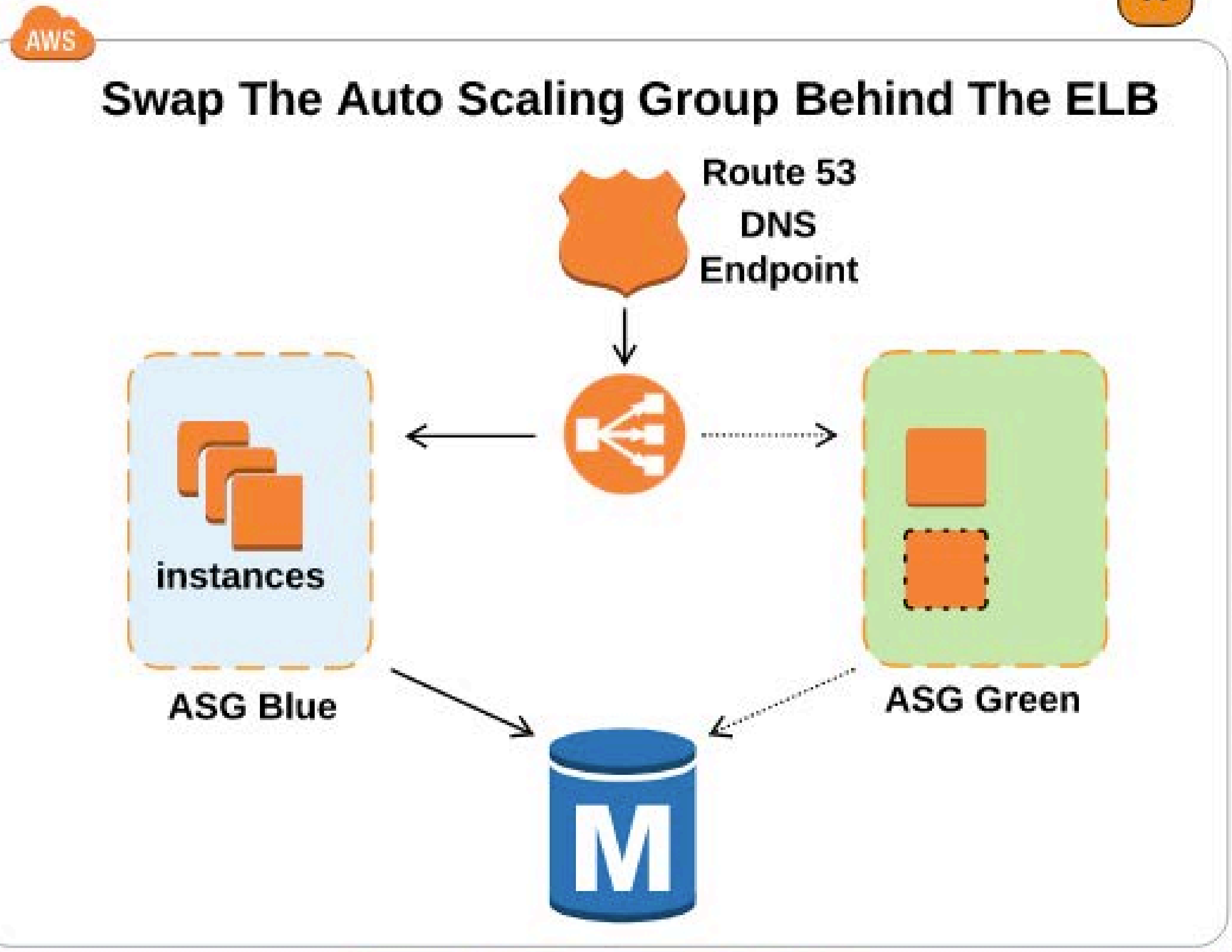
Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda

Blue/Green Deployments

X

Swap The Auto Scaling Group Behind The ELB



Swap ASG Behind The ELB

- If you can't do Blue/Green deployments using DNS, consider using load balancing for traffic management to your blue and green environments.
- This technique uses Auto Scaling to manage the EC2 resources for your blue and green environments, scaling up or down based on actual demand.
- Auto Scaling integrates with Elastic Load Balancing, so any new instances are automatically added to the load balancing pool if they pass the health checks governed by the load balancer.
- A blue group carries the production load while a green group is staged and deployed with the new code.
- When it's time to deploy, you attach the green group to the existing load balancer to introduce traffic to the new environment.
- As you scale up the green ASG, you can take blue ASG instances out of service by either terminating them or putting them in Standby state.

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

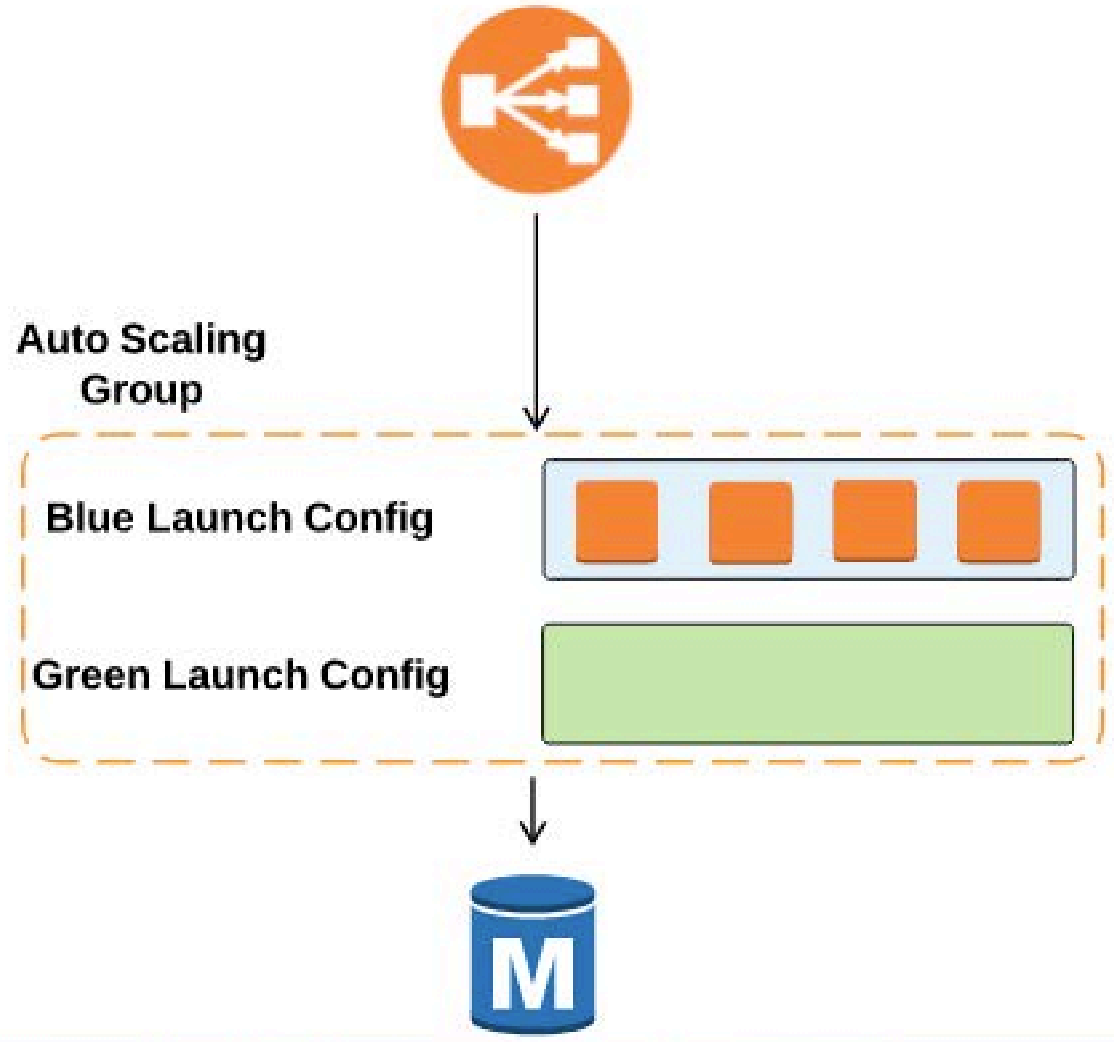
Lambda

Blue/Green Deployments

X

AWS

Update ASG Launch Configurations



Update ASG Launch Configurations

- Only one launch configuration with an ASG at a time, and it can't be modified after you create it.
- Replace the existing launch config (blue) with new launch config (green).
- Scale the Auto Scaling Group to twice it's original size.
- Then shrink the Auto Scaling Group back to it's original size. By default, instances with the old launch configuration are removed first (OR, you could place the instances in StandBy state for rollback).
- To perform a rollback, update the Auto Scaling group with the old launch configuration. Then, do the preceding steps in reverse.

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

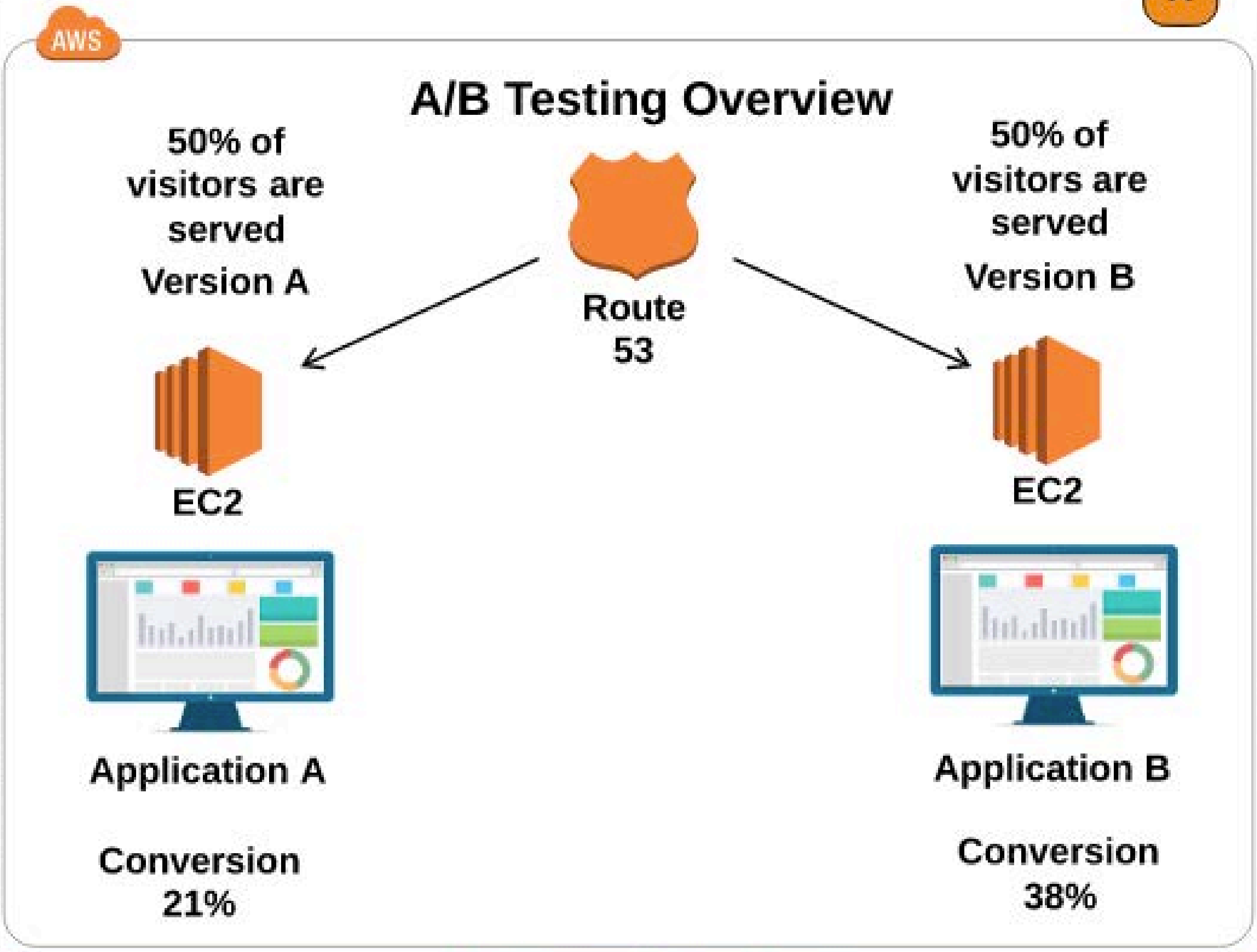
Deployment Pipelines

API Gateway

Lambda

Blue/Green Deployments

X



AWS

A/B Testing Overview

50% of visitors are served Version A

50% of visitors are served Version B

Route 53

EC2

EC2

Application A

Application B

Conversion 21%

Conversion 38%

A/B Testing Overview

- A randomized experiment with two variants, A and B.
- A way to compare two versions of a single application, calculating responses and determining which version is better.
- Version A might be the currently used version, while version B is modified in some respect.
- Common implementations in AWS:
    - Application Load Balancer sending traffic to the A/B apps.
    - CloudFormation Template which creates an Elastic Beanstalk application for both A/B (Route 53 weighted round robin for traffic distribution).
    - CloudFront with Lambda@Edge - Create two CloudFront origins in S3. Lambda@Edge allows running Lambda functions at Edge Locations of the CloudFront CDN supplying intelligence at the front-end and routing traffic appropriately.

Choose a Topic

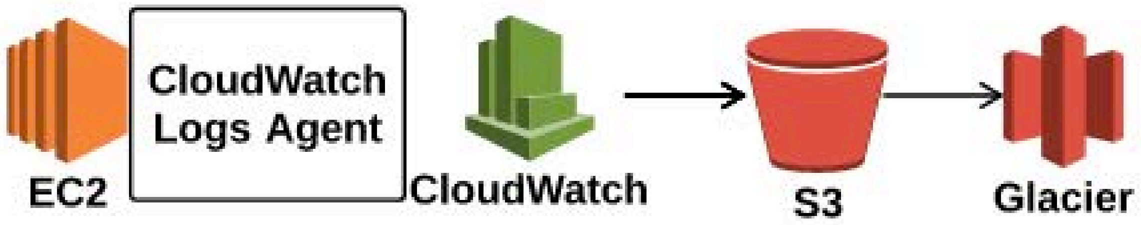
Introduction
Auto Scaling Deployment Concepts
Deployment Concepts With EC2
CloudWatch For DevOps
CloudFormation For DevOps
Elastic Beanstalk For DevOps
Application Deployments With OpsWorks
DynamoDB Concepts
S3 Concepts For DevOps
Blue/Green Deployments
Scenario Solver
Deployment Pipelines
API Gateway
Lambda

Next X

AWS

CloudWatch Scenarios

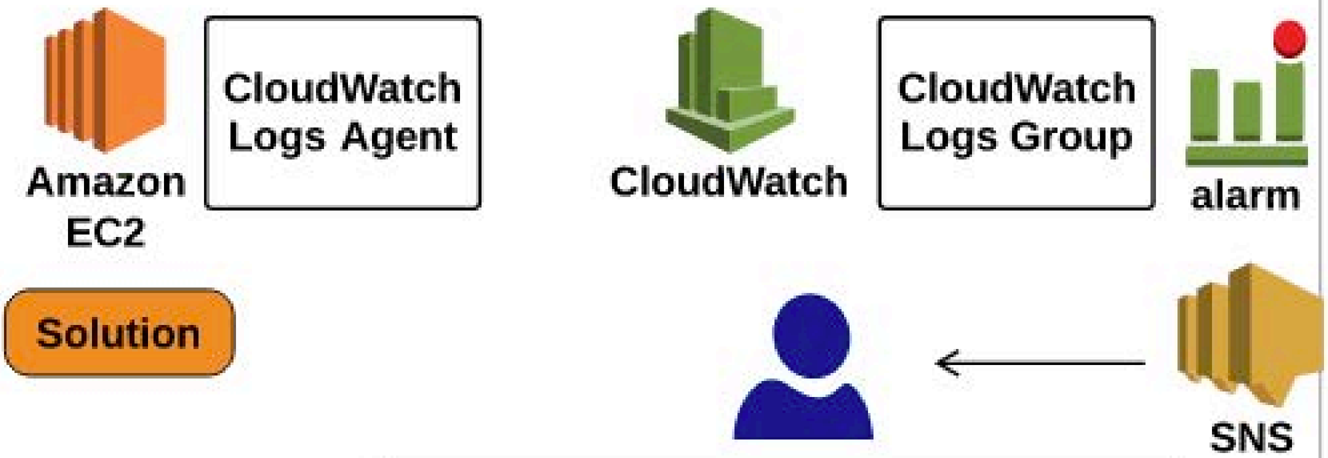
Scenario: Send your EC2 logs to CloudWatch, store your logs durably, and after 60 days send to long term storage.



Solution

Solution: Install CloudWatch Logs Agent on EC2, send log data to CloudWatch, archive data to S3, create S3 Lifecycle policy to send logs to Glacier after 60 days.

Scenario: Using CloudWatch, you need to capture 500 errors from your web server and notify your on-call engineer.



Solution

Solution: Install CloudWatch Logs Agent on EC2, stream log data to CloudWatch, create CloudWatch Log Group to capture 500 errors, set an alarm on those errors. Use SNS to email on-call personnel when alarms occur.

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

Next

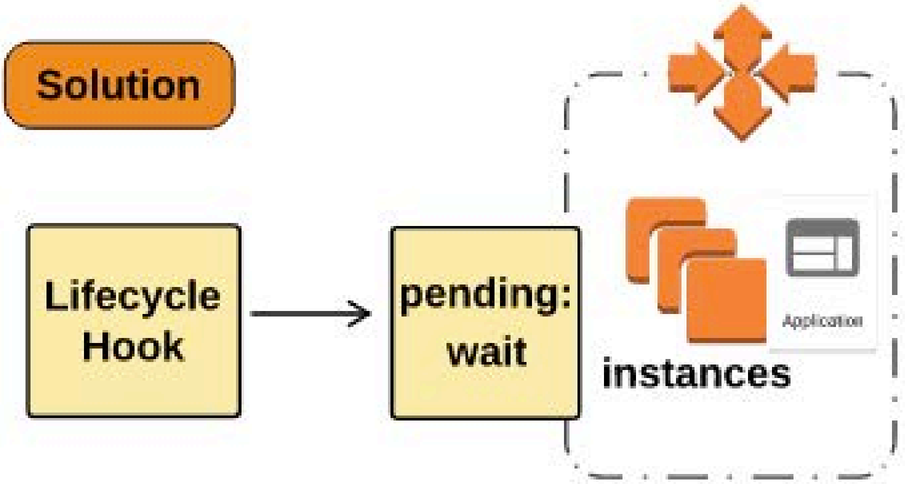
X

AWS

Auto Scaling Scenarios

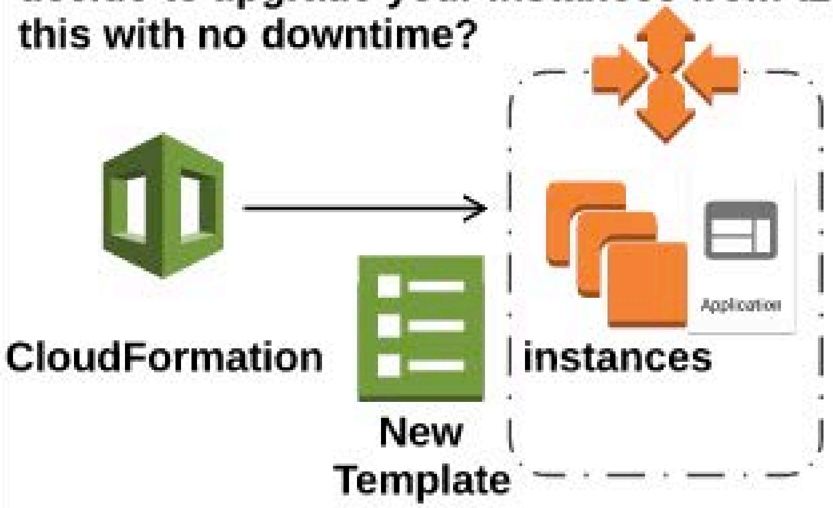
Scenario: Bootstrapping instances in ASG takes 10 minutes. Instances are reported as in-service before bootstrapping completes and you are getting application alarms.

Solution



Solution: Create an ASG Lifecycle Hook to hold the instance in a pending:wait state until bootstrapping is complete. Move to pending:complete when bootstrapping is finished.

Scenario: You have used CloudFormation to deploy an application in an Auto Scaling Group. The ASG is at its maximum 6 instances due to high CPU utilization and it is still too high. You decide to upgrade your instances from t2 to C3. How can you do this with no downtime?



Solution: Update the Launch Config in your template (t2 to C3). Add an update policy to your template and specify AutoScalingRollingUpdate. Perform a stack update with the new template.

Solution

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

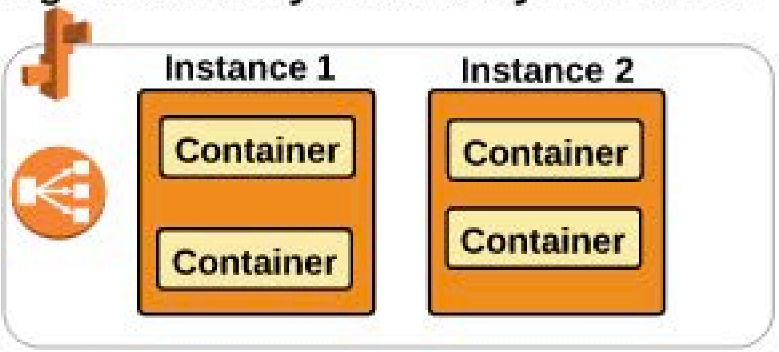
Next

X

AWS

Elastic Beanstalk Scenarios

Scenario: You work for a large software company with a very diverse list of programming languages and platforms. The overriding requirement is to be able to deploy all the applications quickly using Elastic Beanstalk and to have high availability. How can you do this?

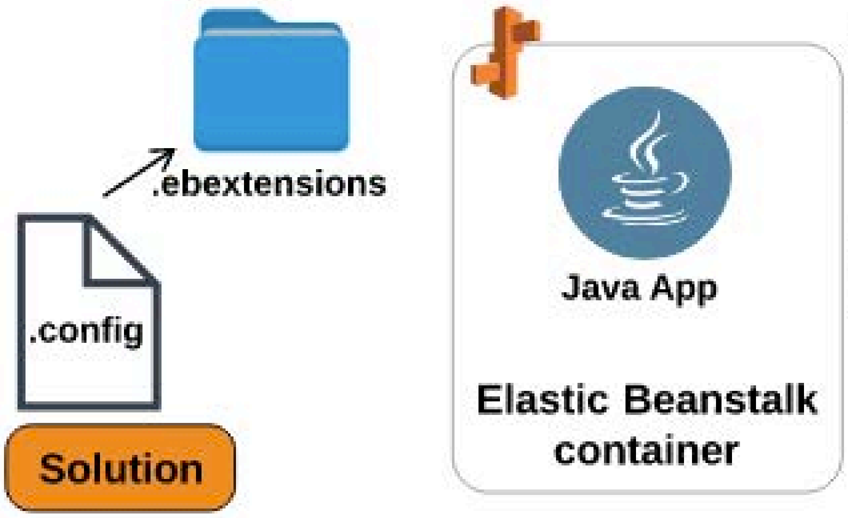


Solution

Elastic Beanstalk Environment

Solution: Use Docker containers to develop the apps into Elastic Beanstalk. Specify Elastic Load Balancing and Auto Scaling in your environment.

Scenario: You have deployed a Java application in an Elastic Beanstalk environment. Now you have created a script to force HTTPS on Apache Web Server. What's the best way to deploy this script?



Solution

Solution: Save your script with a config extension and save it in the .ebextensions folder. Elastic Beanstalk will automatically apply the update.

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda

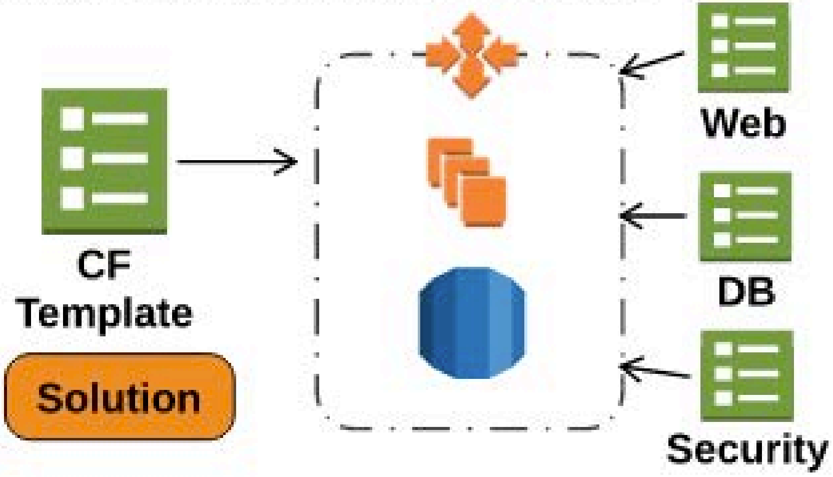
Next

X

AWS

CloudFormation Scenarios

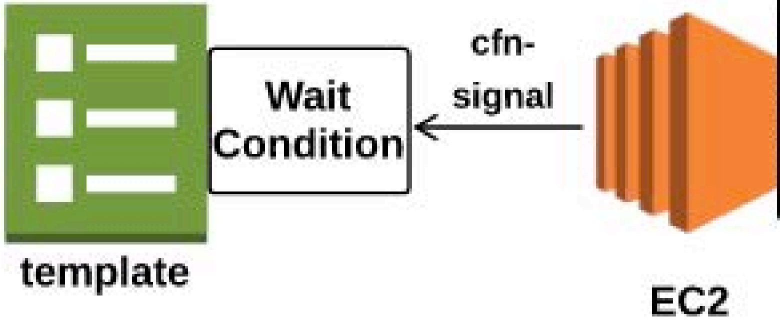
Scenario: You have a single CloudFormation template for your company infrastructure, a multi tier web application. Your web, database, and security teams are in conflict over editing the template. How can you give each team separate parts of the template to manage?



Solution

Solution: Create a nested structure for your template. Create a separate template for each group, allowing them to manage their own resources.

Scenario: You are building a Web Server using a CloudFormation template. You add a long running script to the user data. What can you do to insure that the script has finished and the server is up and running before it is added to the Load Balancer?



Solution: Add a CloudFormation wait condition. Use cfn-signal to signal when the script is complete and the web server is ready.

Solution

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

Next

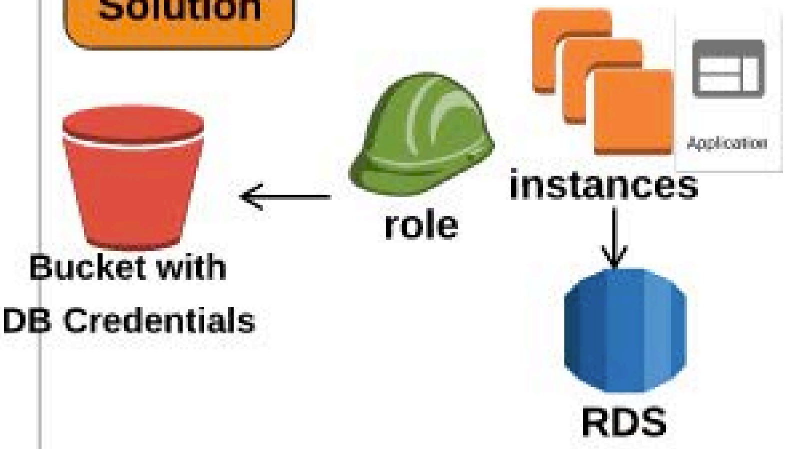
X

AWS

EC2 Scenarios

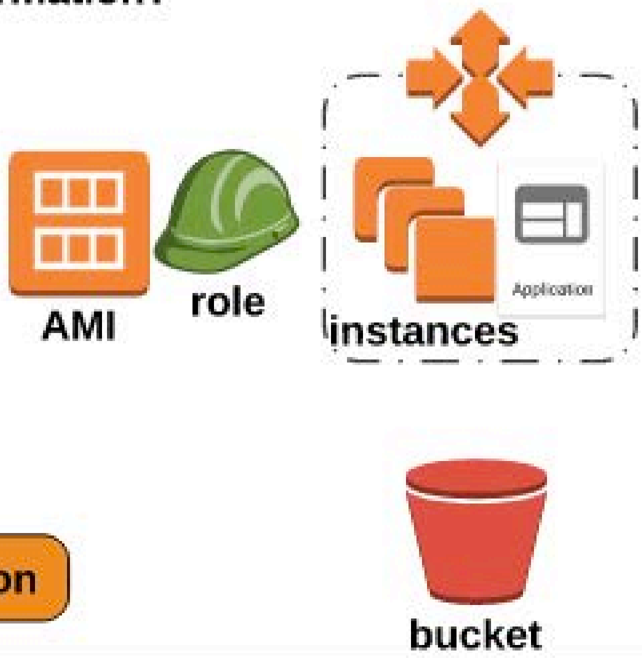
Scenario: You have an application on EC2 instances. You need to securely store DB connection information (not hard code in your app).

Solution



Solution: Create an IAM Role for the EC2 instances giving read access to an S3 Bucket which stores the DB connection credentials. Retrieve the credentials in the app on the EC2 instances.

Scenario: You have an Auto Scaling Group of EC2 instances that you need to bootstrap. You need a highly durable, secure storage for your bootstrapping files and choose S3. How do you retrieve this information?



Solution: Pre-bake the AMI which creates the EC2 instances with an IAM Role which allows read access to the bucket. Retrieve the bootstrapping file programmatically from the instances.

Solution

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

Next

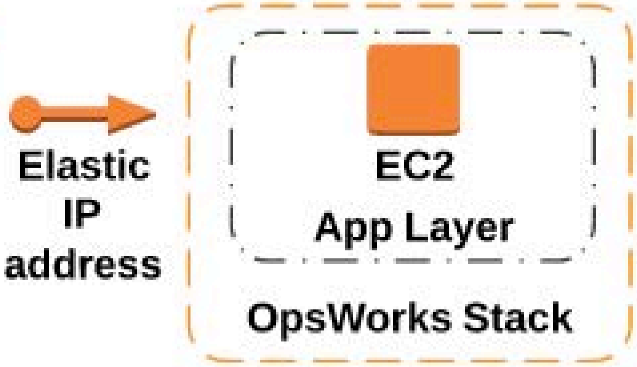
X

AWS

OpsWorks Scenarios

Scenario: You are creating an OpsWorks Stack to host your application. You have created the stack, added Layers including an Application Layer, and add an instance to the Layer. But the instance never reaches a ready state and deployment fails. What could be wrong?

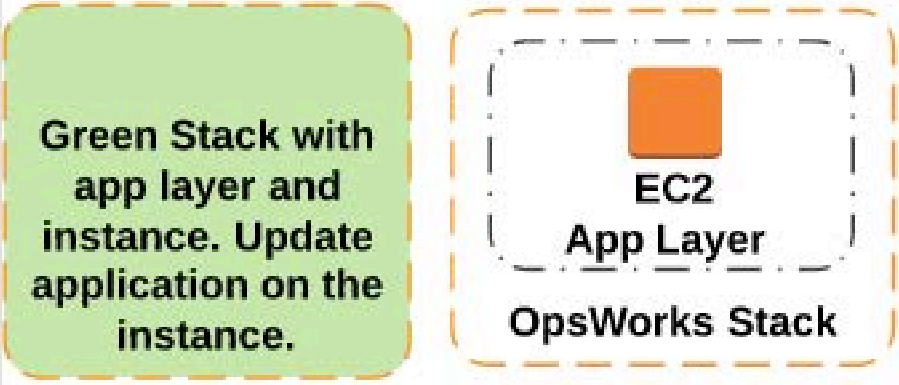
Solution



Solution: The EC2 instance does not have a public ip address (or elastic ip address) and is failing bootstrapping.

Make sure instances are assigned a public ip address (or elastic ip address)

Scenario: You are managing an OpsWorks Stack. You have a new requirement to perform Blue/Green Deployments to greatly minimize downtime. How can you implement this in OpsWorks?


Green Stack with app layer and instance. Update application on the instance.

Solution: Clone your OpsWorks Stack. When you are ready to deploy, update the application on your new (green) stack, then swap urls.

Solution