
# Letting the Phoenix Fly in Production

## Deploying Phoenix Apps using AWS CodeDeploy

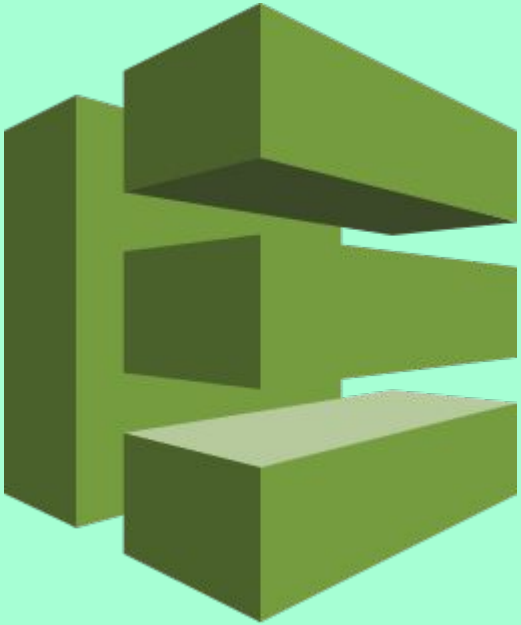
Monica Hirst

@monix88

# My Phoenix App Works!

# But… How Do I Deploy This Thing?

# AWS to the Rescue!



amazon
web services™

# CodeDeploy Benefits

I KNOW THIS IS WHY YOU'RE HERE

- Accessible deployment
- Consistent environments
- Supports continuous deployment
- Zero downtime deployments
- Diagnose deploy failures quickly

# The Road to Production

YOU'LL BE DONE BY LUNCHTIME

1. Launch an EC2 Instance
2. Create an application configuration
3. Add required deploy scripts
4. Deploy your application!

# The Road to Production

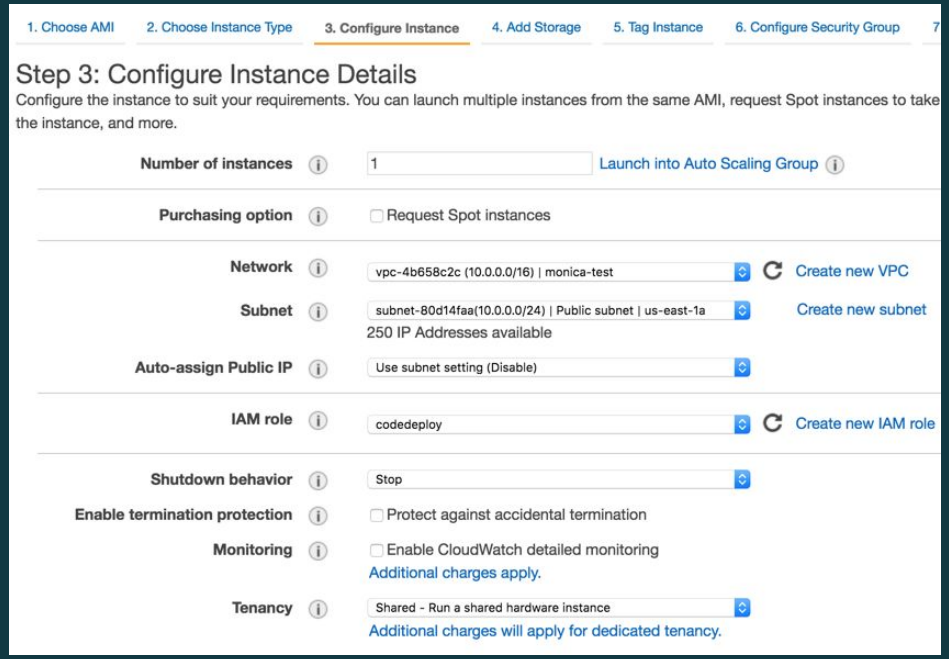
YOU'LL BE DONE BY LUNCHTIME

1. Launch an EC2 Instance

# 1. Launch an EC2 Instance

What you need:

- Name tag for the instance
- An “instance profile” in IAM that allows S3 storage access
- AWS CLI installed




https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:

# 1. Launch Config for EC2 Instance

What you need:

- An “instance profile” in IAM that allows S3 storage access
- AWS CLI installed

1. Choose AMI | 2. Choose Instance Type | 3. Configure details | 4. Add Storage | 5. Configure Security Group | 6. Review

Create Launch Configuration

Name: monica-as-config

Purchasing option: Request Spot Instances

IAM role: codedeploy

Monitoring: Enable CloudWatch detailed monitoring
Learn more

Advanced Details

Kernel ID: Use default

RAM Disk ID: Use default

User data: As text / As file / Input is already base64 encoded

```
#!/bin/bash
apt-get -y update
apt-get -y install awscli
apt-get -y install ruby2.0
cd /home/ubuntu
aws s3 cp s3://aws-codedeploy-us-east-1/latest/install . --region us-east-1
chmod +x ./install
./install auto
```

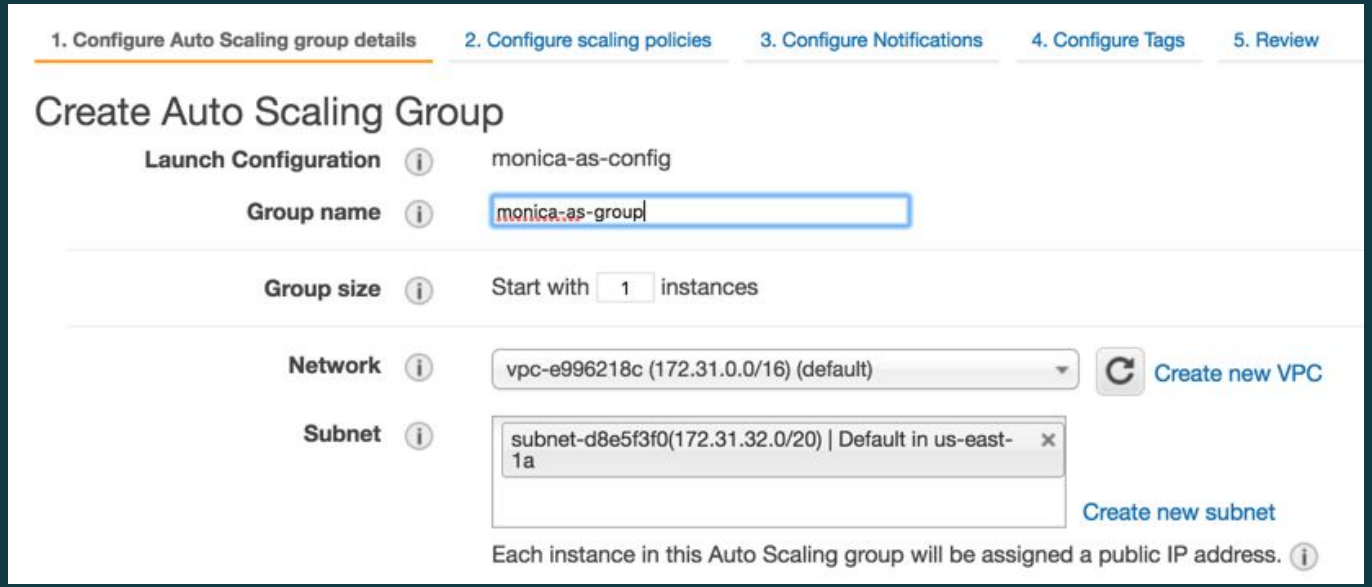
IP Address Type:
- Only assign a public IP address to instances launched in the default VPC and subnet. (default)
- Assign a public IP address to every instance.
- Do not assign a public IP address to any instances.

https://console.aws.amazon.com/ec2/autoscaling/home?region=us-east-1#CreateLaunchConfiguration:CreationFlowType=

# 1a. Launch Auto Scaling Group

What you need:

- Launch Configuration



# Setting up AWS CodeDeploy

YOU'LL BE DONE BY LUNCHTIME

1. Launch an EC2 Instance
2. Create an application configuration

# 2. Create an Application

What you need:

- A name for your app
- The EC2 instance Name
- A system user (service role) defined in IAM that can use CodeDeploy

Create a new application; specify the instances to deploy it to; and specify the conditions for a successful deployment.

**Application Name*** monica-app

**Deployment Group Name*** monica-app-group

## Add Instances

Locate and add existing instances to this deployment group by searching for their tags or Auto Scaling group names.

AWS CodeDeploy requires the following for each instance that it deploys to.

1. Each Amazon EC2 instance must launch with the correct IAM instance profile attached. Learn More
2. Each Amazon EC2 instance must have identifying Amazon EC2 tags (Learn More) or be in an Auto Scaling group. Learn More
3. Each on-premises instance must have an associated IAM user, identifying on-premises instance tags, and a special configuration file. Learn More
4. The AWS CodeDeploy agent must be installed and running on each instance. Learn More

**Search by Tags**

| | Tag Type | Key | Value | Instances | |
|---|---|---|---|---|---|
| 1 | Amazon EC2 | Name | monica-appserver | 1 | |
| 2 | Amazon EC2 | | | | |

**Total Matching Instances: 1**

## Deployment Configuration

Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application will be deployed and the success or failure conditions for a deployment.

**Deployment Config*** CodeDeployDefault.OneAtATime

Deploys to one instance at a time. Succeeds if all instances succeed. Fails after the very first failure. Allows the deployment to succeed for some instances, even if the overall deployment fails.

## Triggers

Create triggers and subscribe to Amazon SNS topics to receive notifications about events in this deployment group. Learn more

Create trigger

| Name | Events | Type | |
|---|---|---|---|
| *No triggers have been created for this deployment group.* | | | |

## Service Role

Select an existing service role that grants AWS CodeDeploy access to the instances.

**Service Role ARN*** arn:aws:iam::327613904809:role/codedeploy

https://console.aws.amazon.com/codedeploy/home?region=us-east-1#/applications/new

# The Road to Production

YOU'LL BE DONE BY LUNCHTIME

1. Launch an EC2 Instance
2. Create an application configuration
3. Add required deploy scripts

# 3. Add required deploy scripts

What you need:

- Appspec.yml - CodeDeploy looks for this first
- Install script
- Application start/stop script

appspec.yml

```yaml
# Don't touch version, that's Amazons stuff.
version: 0.0

os: linux

# Where is the code?
files:
  - source: /
    destination: /opt/phoenix-codedeploy

# Scripts that perform deployment tasks
hooks:
  ApplicationStop:
    - location: deploy/application-stop.sh
      timeout: 100

  BeforeInstall:
    - location: deploy/before-install.sh
      timeout: 300

  AfterInstall:
    - location: deploy/after-install.sh
      timeout: 300

  ApplicationStart:
    - location: deploy/application-start.sh
      timeout: 180
```

http://docs.aws.amazon.com/codedeploy/latest/userguide/how-to-add-appspec-file.html

# 3. Add required deploy scripts

What you need:

- Appspec.yml - CodeDeploy looks for this first
- Install script
- Application start/stop script

```
ENVIRONMENT=`echo "$APPLICATION_NAME" | cut -d '-' -f 1 | tr A-Z a-z`
# Put the environment in the log
echo "Customizing environment: $ENVIRONMENT"

# Setup variables depending on what environment
case $ENVIRONMENT in
    blue)
        S3=monica-blue
        export MIX_ENV=blue
    ;;
    red)
        S3=monica-red
        export MIX_ENV=red
    ;;
    production)
        S3=production-s3-buckets-appconfigbucket-wyvb0uh5uocb
        export MIX_ENV=prod
    ;;
    *)
        echo "Error: undefined environment: $ENVIRONMENT"
        exit 1
    ;;
esac

SOURCE_DIR=/opt/phoenix-codedeploy/deploy

# Move into the app directory
cd /opt/phoenix-codedeploy

# Pull in secrets from S3 Bucket
aws --region=us-east-1 s3 cp s3://$S3/monica-app-$ENVIRONMENT.secret.exs /opt/phoenix-codedeploy/config/$MIX_ENV.secret.exs

# Copy over the upstart script and set MIX_ENV correctly
sed "s/MIX_ENV_VALUE/$MIX_ENV/"  /opt/phoenix-codedeploy/deploy/monica-app-upstart.conf >/etc/init/monica-app.conf

export HOME=/root
mix local.hex --force
yes | head -n 1000 | mix deps.get
yes | head -n 1000 | mix deps.compile
yes | head -n 1000 | mix compile
yes | head -n 1000 | mix ecto.migrate
yes | head -n 1000 | mix phoenix_codedeploy.insert_seeds
mix phoenix.digest -o _build/prod/lib/phoenix_codedeploy/priv/static/ web/static
```

# 3. Add required deploy scripts

What you need:

- Appspec.yml - CodeDeploy looks for this first
- Install script
- Application start/stop script

```
#!/bin/bash

sudo stop monica-app || : # don't fail the deploy if the service isn't running
```

```
#!/bin/bash

sudo start monica-app
```

# The Road to Production

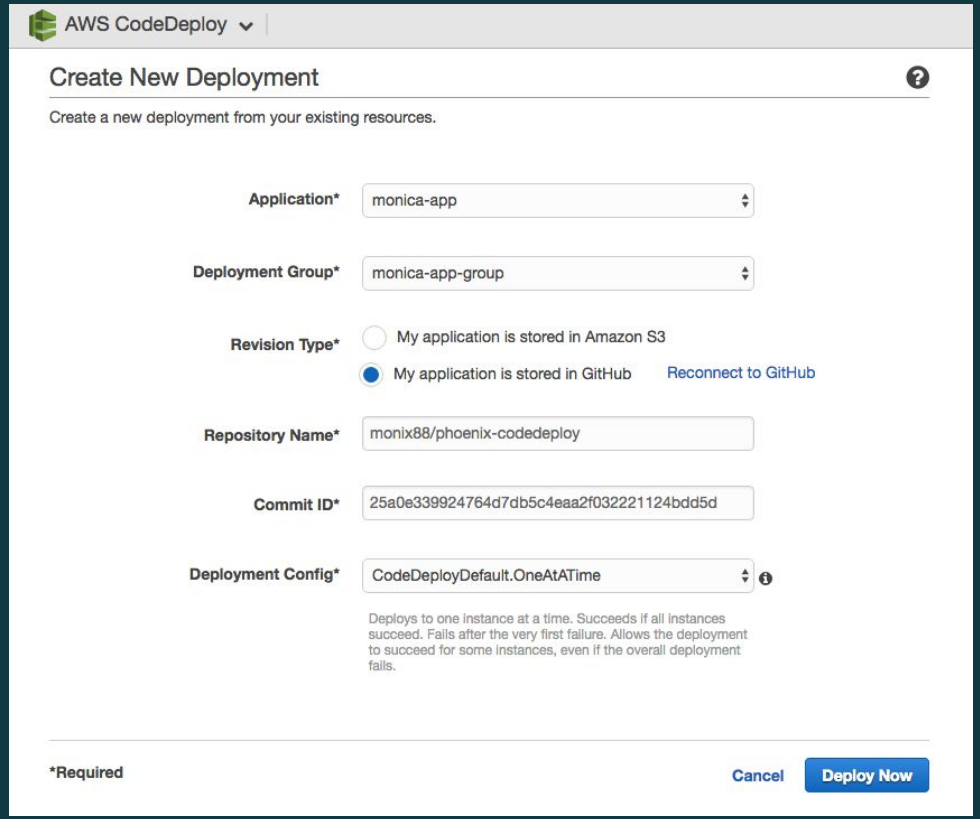
YOU'LL BE DONE BY LUNCHTIME

1. Launch an EC2 Instance
2. Create an application configuration
3. Add required deploy scripts
4. Deploy your application!

# 4. Deploy your app!

What's required:

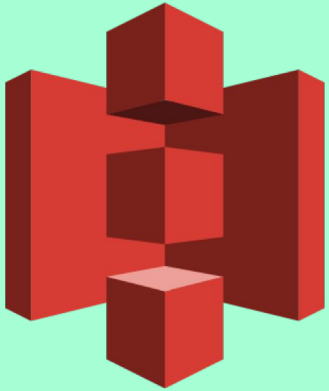
- Repository to deploy from
- Connect to your Github account
- Github Commit ID to deploy
- Deployment strategy

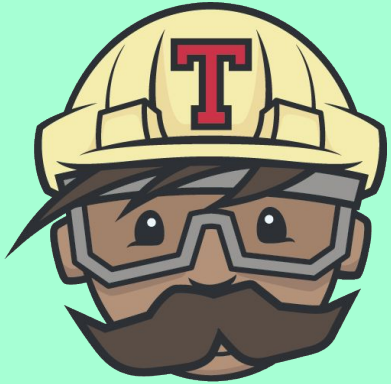



https://console.aws.amazon.com/codedeploy/home?region=us-east-1#/deployments/new

# Note on Continuous Deployment

WHAT'S THE LATEST?



http://blogs.aws.amazon.com/application-management/post/Tx33XKAKURCCW83/Automatically-Deploy-from-GitHub-Using-AWS-CodeDeploy

# Success!

AWS CodeDeploy

## Deployments

View, diagnose, and manage your deployments.

Create New Deployment

Filter All Deployments | Search by Deployment ID | Deployments per page 10 | Viewing 1 to 1 deployments

| Deployment ID | Applicati... | Deploym... | Revision Location | Start Time | End Time | Status | Actions |
|---|---|---|---|---|---|---|---|
| d-17Y0V5SPF | Gold-API... | Gold-API... | github://grownups/sog-... | 19 hours ago | 19 hours ago | Succeeded | |

### Details

| | |
|---|---|
| **Deployment ID** | d-17Y0V5SPF |
| **Deployment Config** | CodeDeployDefault.OneAtATime |
| **Minimum Healthy Hosts** | 1 of 2 instances |
| **Revision Location** | github://grownups/sog-api/commit/8b70e060d9a9c4470b6791f2d4a8bae161f2a3fb |
| **Deployment Description** | Deploy build 3377 via Travis CI |

### Instances

2 of 2 Instances Completed

2

Succeeded

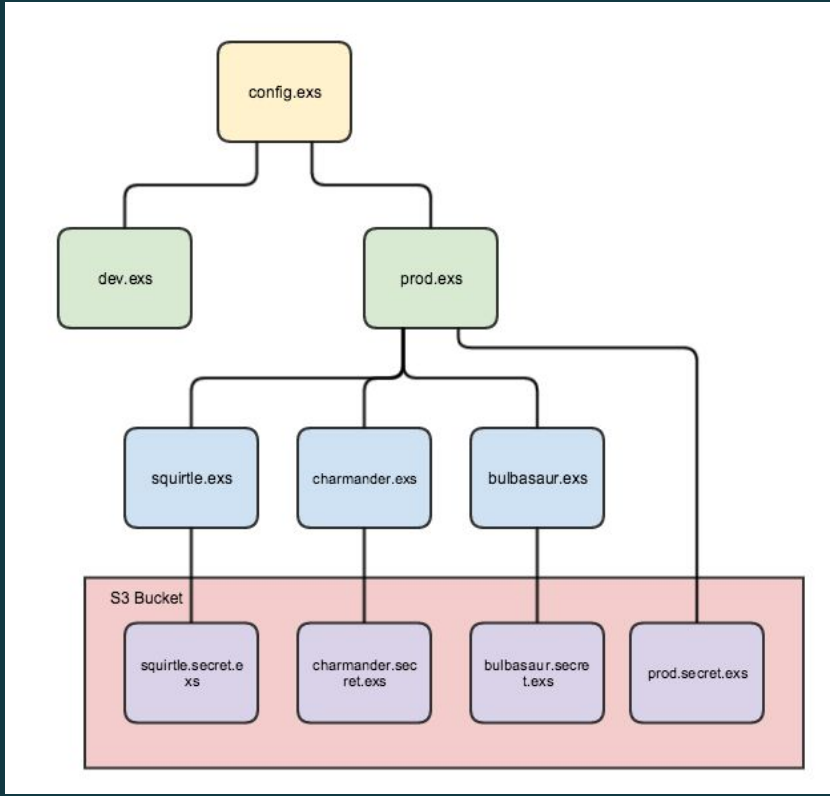
View All Instances

# Success!

# Phoenix Configuration

BUT WAIT, THERE'S MORE

1. App Config: no env-specific values
2. Public Config: non-sensitive values
3. QA Env Config: inherits mostly Prod-like values, QA overrides (NEW!)
4. Private Config: DB credentials, etc.



# That’s All Folks!

Talk to me:

@monix88

Check out a sample app setup:

https://github.com/monix88/phoenix-codedeploy


TM