AWS - CodeDeploy

By

Keshav Kummari

AWS - CodeDeploy

- CodeDeploy – Automates deployments of your applications to EC2, Lambda, and even on-premises environments.

- Deploy your applications in a fast, consistent, and reliable way.

- Benefits of CodeDeploy:
  - Automated Deployments
  - Minimize Downtime
  - Stop the Rollback
  - Centralized Control

Deployment Types

- In-place deployment:
  - The existing servers are updated with the new version of an application

- Blue/Green deployment(EC2):
  - New application versions are deployed on a new set of instances
  - Traffic is routed from old to new instances
  - If there are failures, the application can fall back to the older deployment version
- Blue/Green Deployment(Lambda):
  - Traffic is shifted from one lambda version to another, this can happen in multiple ways:

- Canary:
    - A percentage of traffic is shifted to the new version
    - CodeDeploy then waits for a specified time and shifts the rest of the traffic it it sees no errors.

- Linear:
    - Traffic is shifted in equal increments with an equal number of minutes between each increment.

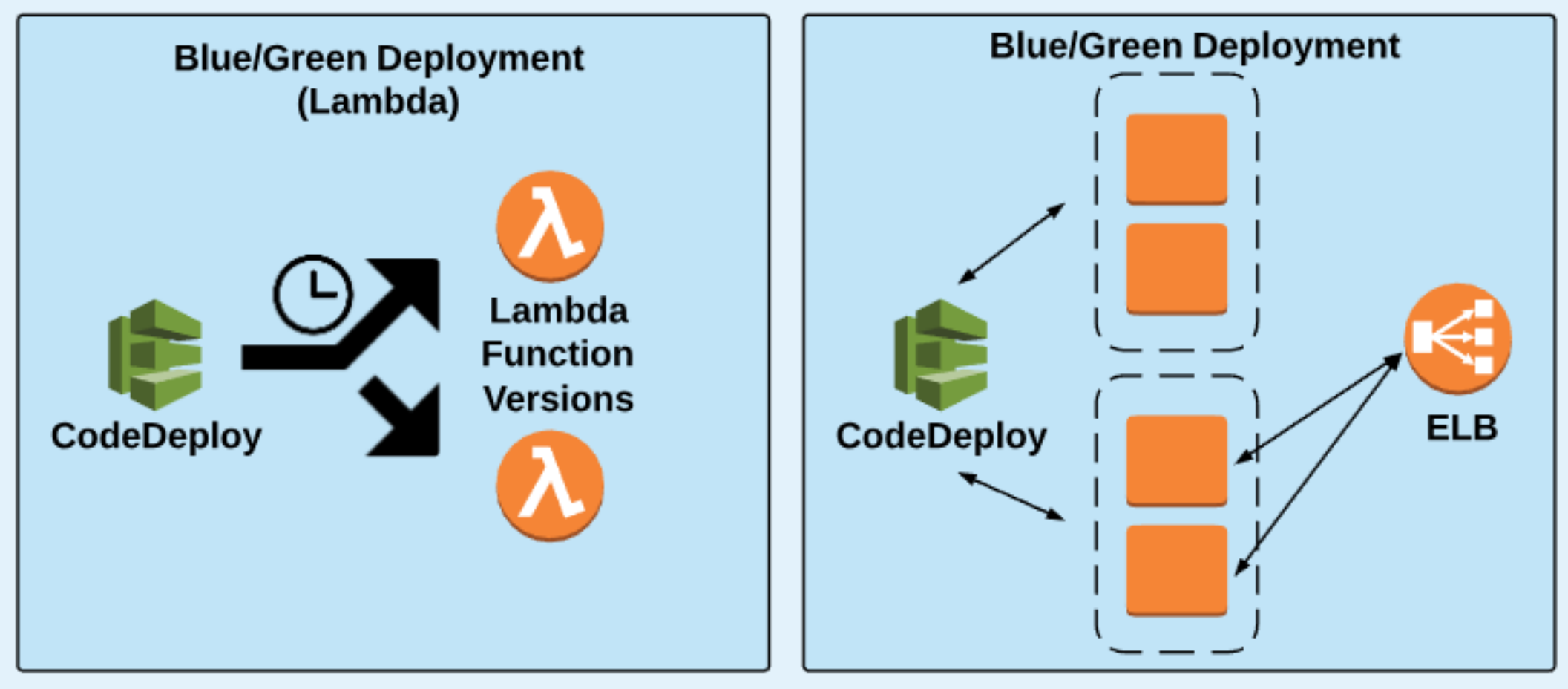
- All at once:
    - Traffic is immediately and completely shifted to the new version of the Lambda function.

CodeDeploy Hooks

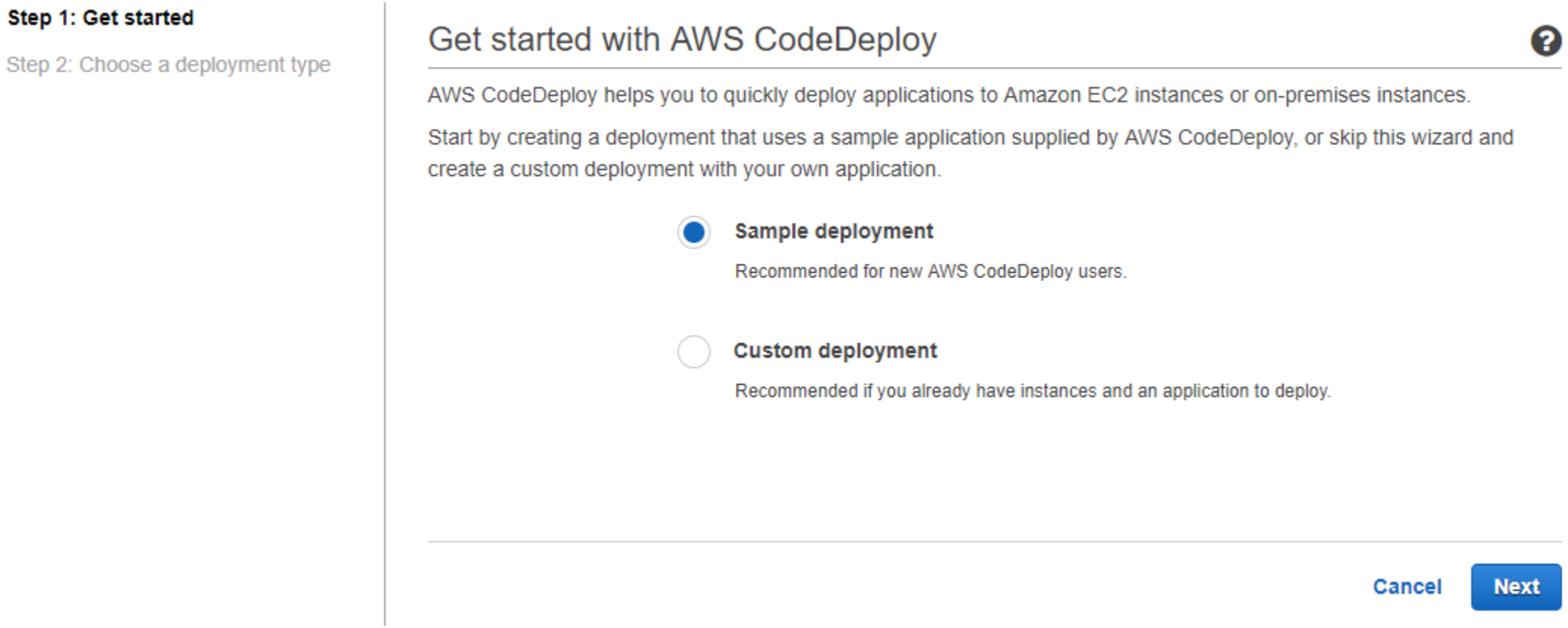
- CodeDeploy AppSepc File:
  - Lambda Deployments:
    - Which functions to deploy
    - Which functions to use as validation tests
  - EC2/On-Premises Deployments:
    - What to install (From S3 or Github)
    - What lifecycle events hooks to run in response to deployment lifecycle events
- Lifecycle Hooks:
  - Available hooks depend on the deployment type
  - Hooks allow arbitrary scripts to run during your deployment process
  - Typical examples include:
    - BeforeInstall
    - AfterInstall
    - ApplicationStart
    - ApplicationStop
    - ValidateService

```yaml
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
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

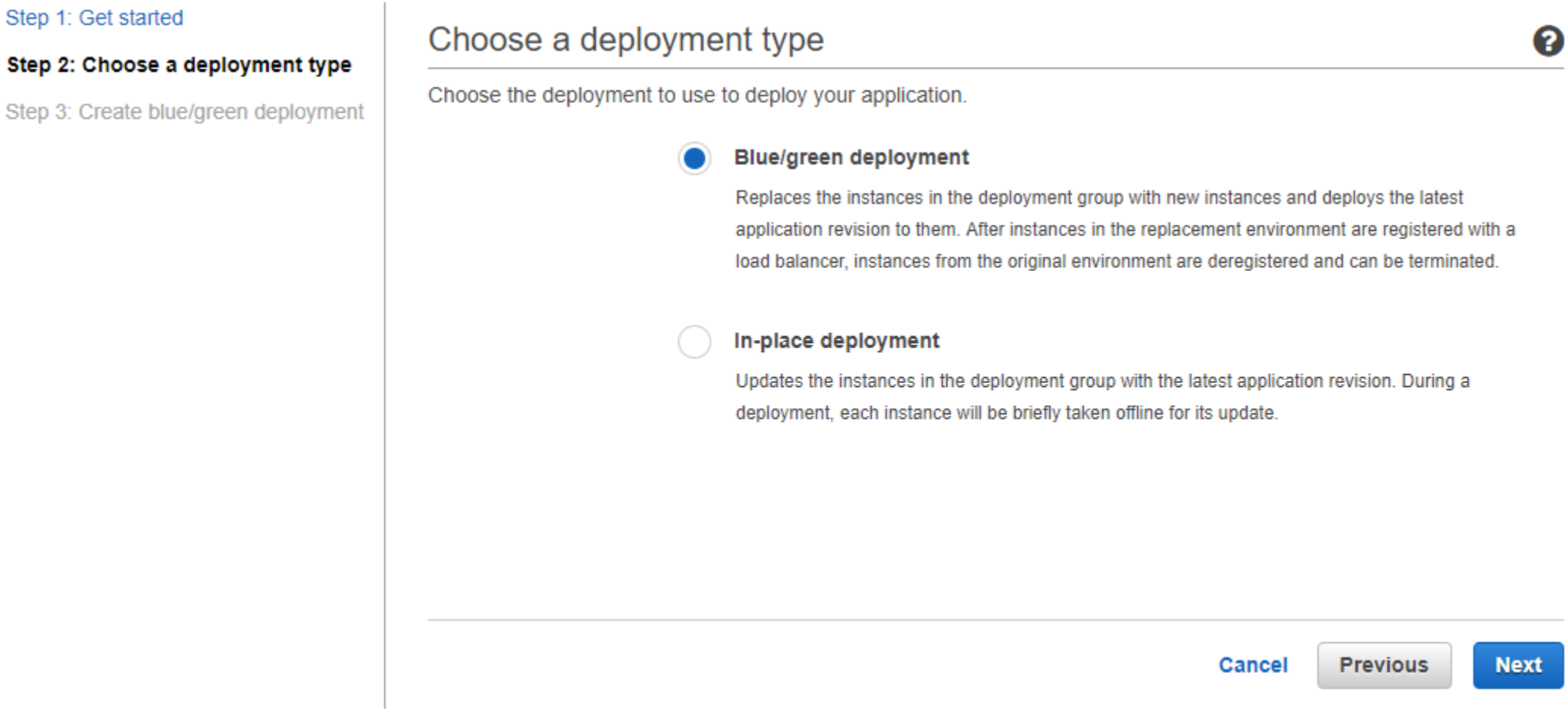
CodeDeploy Examples



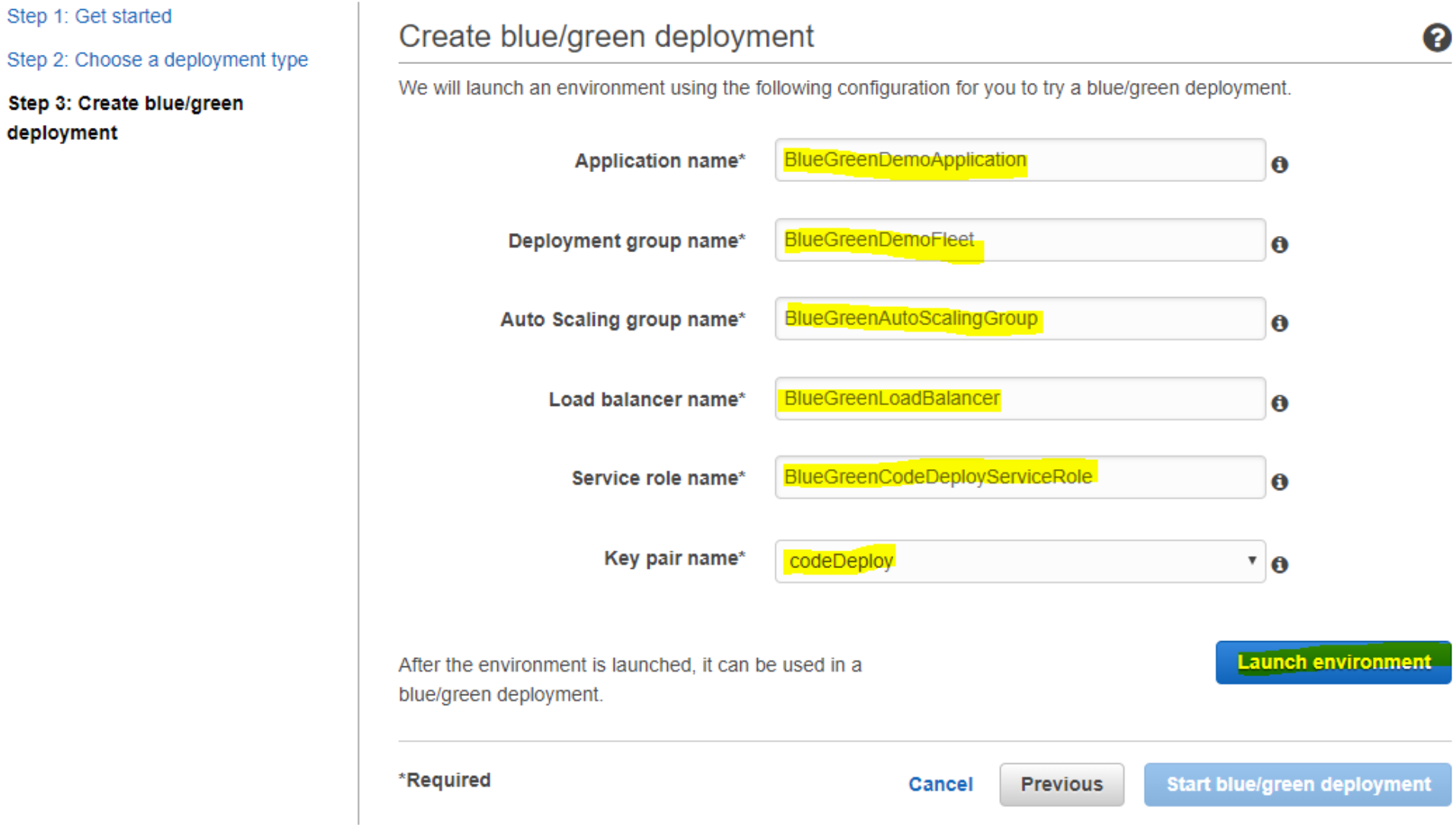
Click on “Sample Deployment” and click on “Next”



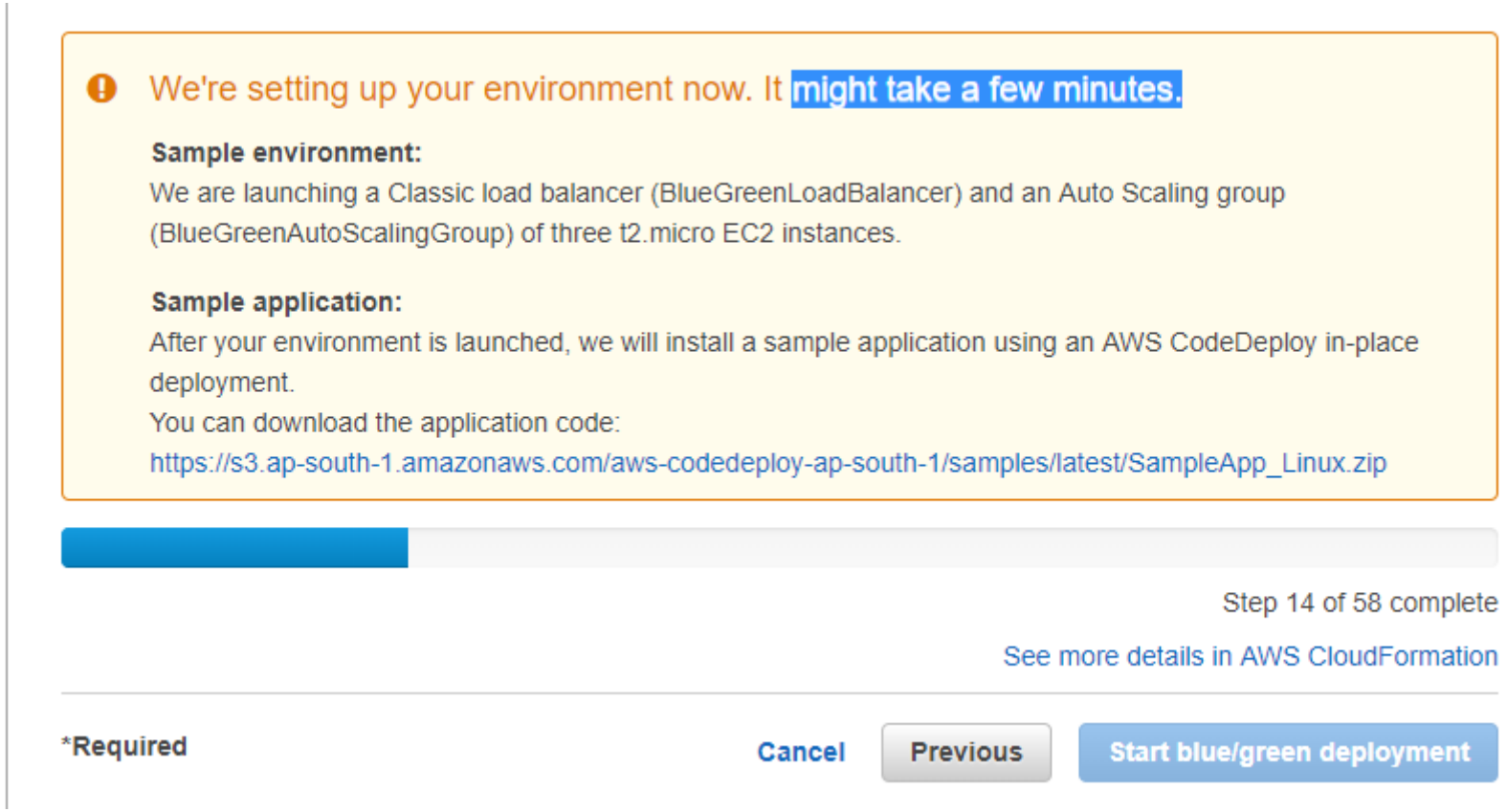
Let’s select “Blue/Green Deployment”



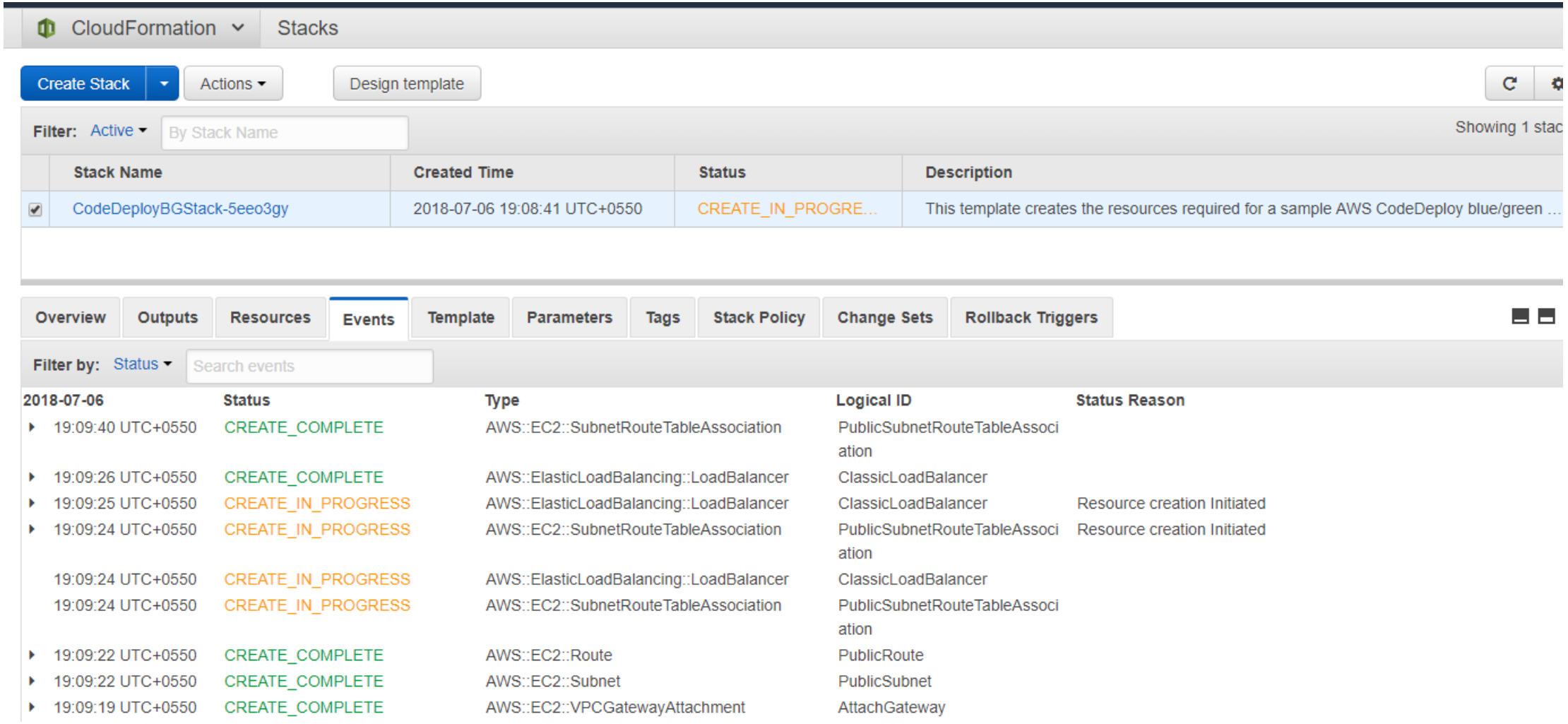
Fill the required fields and click on “Launch environment”



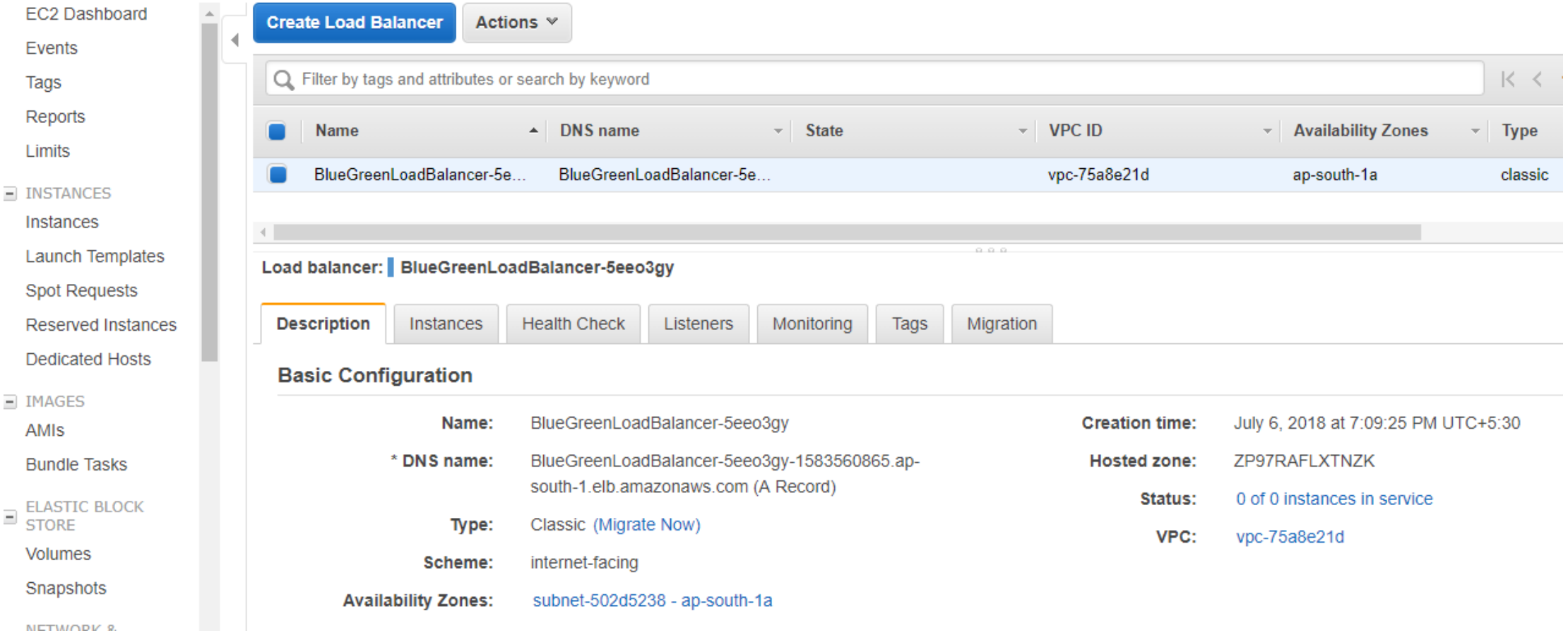
It’s under process…



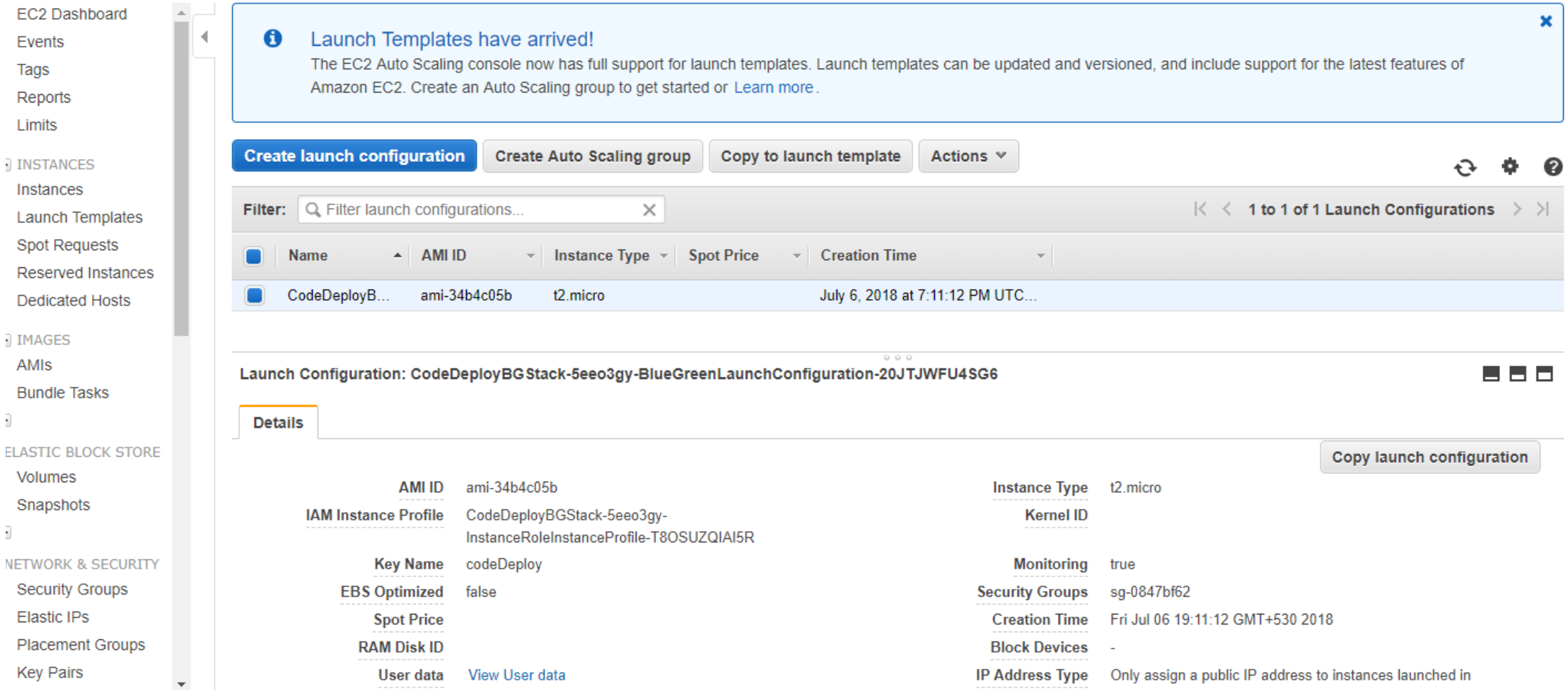
CloudFormation is under process..



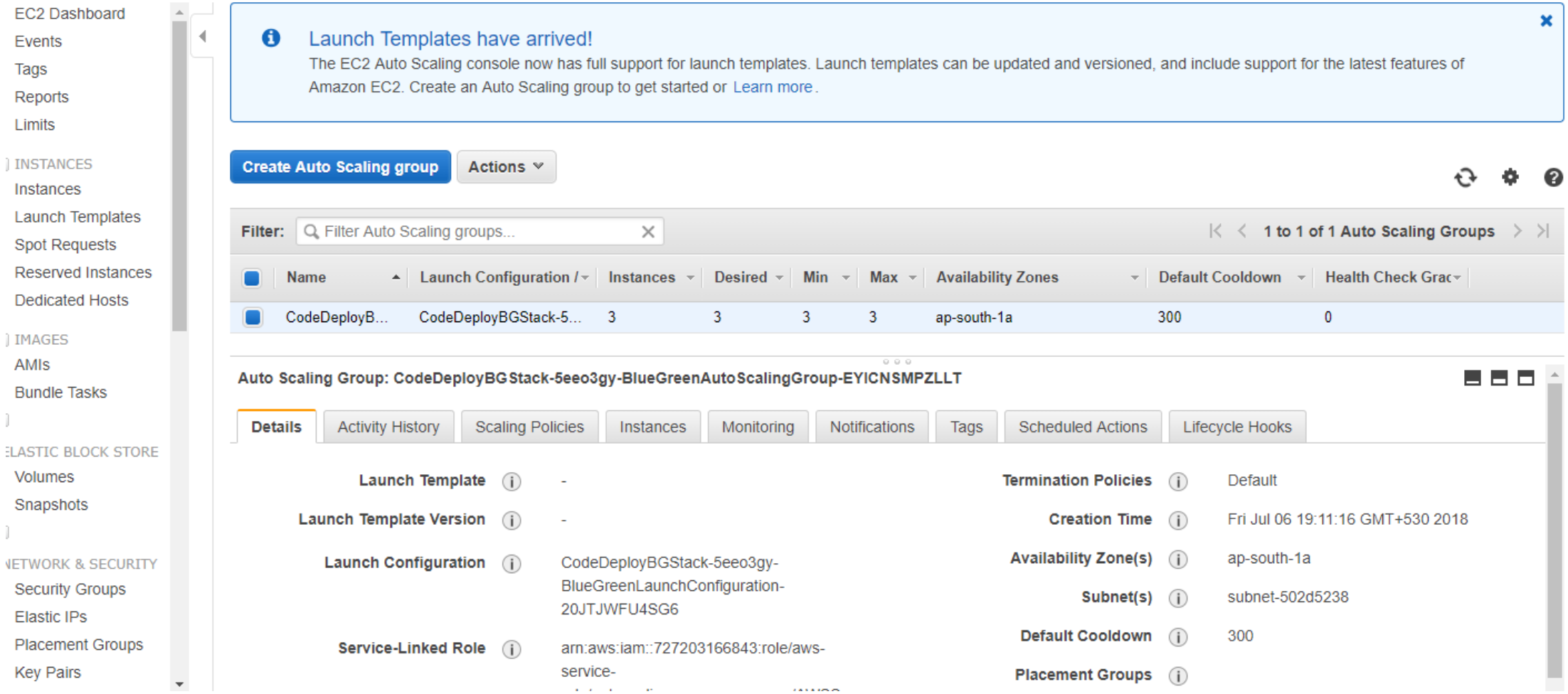
ELB Details


EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Create Load Balancer
Actions
Filter by tags and attributes or search by keyword
Name
DNS name
State
VPC ID
Availability Zones
Type
BlueGreenLoadBalancer-5e...
BlueGreenLoadBalancer-5e...
vpc-75a8e21d
ap-south-1a
classic
Load balancer: BlueGreenLoadBalancer-5eeo3gy
Description
Instances
Health Check
Listeners
Monitoring
Tags
Migration
Basic Configuration
Name: BlueGreenLoadBalancer-5eeo3gy
* DNS name: BlueGreenLoadBalancer-5eeo3gy-1583560865.ap-south-1.elb.amazonaws.com (A Record)
Type: Classic (Migrate Now)
Scheme: internet-facing
Availability Zones: subnet-502d5238 - ap-south-1a
Creation time: July 6, 2018 at 7:09:25 PM UTC+5:30
Hosted zone: ZP97RAFLXTNZK
Status: 0 of 0 instances in service
VPC: vpc-75a8e21d

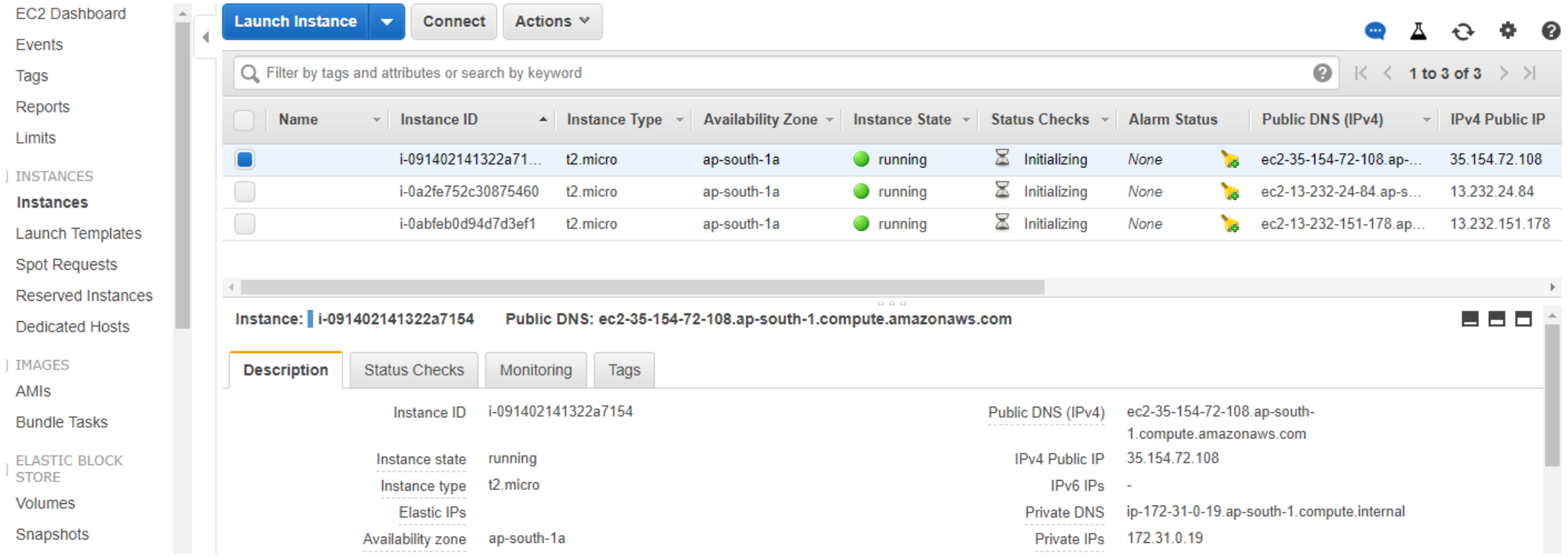
Launch Configuration Info



Auto Scaling Details



EC2 Instance Info



Thank you!