Netflix Open Source Software - Spinnaker on the AWS Cloud

Quick Start Reference Deployment

Month 2016

*<Partner Organization>*

*AWS Quick Start Reference Team*

Contents

[Overview 1](#_Toc448499689)

[Architecture 2](#_Toc448499690)

[Prerequisites 2](#_Toc448499691)

[Specialized Knowledge 2](#_Toc448499692)

[Technical Requirements 2](#_Toc448499693)

[Deployment Steps 2](#_Toc448499694)

[Step 1. Prepare an AWS Account 2](#_Toc448499695)

[Step 2. Launch the Quick Start 3](#_Toc448499696)

[Step 3. Test the Deployment 4](#_Toc448499697)

[FAQ 4](#_Toc448499698)

[Additional Resources 5](#_Toc448499699)

[Send Us Feedback 5](#_Toc448499700)

This Quick Start deployment guide was created by Amazon Web Services (AWS) in partnership with *<partner organization>*.

## Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying Netflix Open Source Software (NetflixOSS) Spinnaker on the Amazon Web Services (AWS) cloud. [Quick Starts](http://aws.amazon.com/quickstart/) are automated reference deployments that use AWS CloudFormation templates to launch, configure, and run the AWS compute, network, storage, and other services required to deploy a specific workload on AWS.

Spinnaker is a Netflix Open Source Software use as a continuous delivery platform for releasing software changes with high velocity and confidence. It provides two core sets of features: cluster management and deployment management. Spinnaker facilitates the creation of pipelines that represent a delivery process that can begin with the creation of some deployable asset (such as an machine image, Jar file, or Docker image) and end with a deployment.

It is built based on generalizing the building blocks of Netflix’s delivery pipelines into configurable Stages that are composable into *Pipelines*. Pipelines can be triggered by the completion of a Jenkins Job, manually, via a cron expression, or even via other pipelines. Spinnaker comes with a number of stages, such as baking a machine image, deploying, running a Jenkins Job, or manual judgement to name a few. Pipeline stages can be run in parallel or serially. More information on built-in stages see [Spinnaker Deployment Management](http://www.spinnaker.io/docs/overview#section-deployment-management).



Spinnaker also provides cluster management capabilities and provides deep visibility into an application’s cloud footprint. Via Spinnaker’s application view, you can resize, delete, disable, and even manually deploy new server groups using strategies like Blue-Green

This Quick Start is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement NetflixOSS Spinnaker to manage their AWS compute resources.

An expanded version of this deployment guide with detailed instructions and screen illustrations is available on the [*Spinnaker*](http://www.spinnaker.io/) website.

## Architecture

Deploying this Quick Start with the **default parameters** builds the following Spinnaker environment in the AWS cloud.

**

Figure 2: Quick Start *Spinnaker* Architecture on AWS

This Quick Start deploys the resource shown in Figure 2 and uses them as follows:

* An Amazon VPC is created in the region you choose when you launch the stack. A single, public VPC subnet is created in an Availability Zone.
  + The subnet only has port 22 open for SSH access.
* One Spinnaker instance is deployed into the VPC subnet
* IAM Users and Role:
  + SpinnakerUser – User account that the end user uses on Spinnaker instance to access AWS resources.
  + SpinnakerRole – Role Spinnaker instance uses to access AWS resources. It has permission to launch other EC2, pass BaseIAMRole role to another EC2 instances, and create AWS resources.
  + BaseIAMRole – Role Spinnaker pass onto application instances that it deploys

## Prerequisites

### Specialized Knowledge

Before you deploy this Quick Start, we recommend that you become familiar with the following AWS services. (If you are new to AWS, see [Getting Started with AWS](http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/intro.html).)

* [Amazon VPC](http://aws.amazon.com/documentation/vpc/) - The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
* [Amazon EC2](http://aws.amazon.com/documentation/ec2/) – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
* [AWS CloudFormation](http://aws.amazon.com/documentation/cloudformation/) – AWS CloudFormation gives you an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources (for example, Amazon EC2 instances) that you want. You don't have to individually create and configure the resources or figure out dependencies—AWS CloudFormation handles all of that.
* [IAM](http://aws.amazon.com/documentation/iam/) – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

### Technical Requirements

*AWS account configuration, operating system, licensing, DNS, etc. requirements*

## Deployment Steps

### Step 1. Prepare an AWS Account

1. If you don’t already have an AWS account, create one at <http://aws.amazon.com> by following the on-screen instructions.
2. Use the region selector in the navigation bar to choose the Amazon EC2 region where you want to deploy *Spinnaker* on AWS.
3. Create a [key pair](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html) in your preferred region.
4. Create an EC2 role - BaseIAMRole. EC2 instances launched with Spinnaker will be associated with this role.
5. If necessary, [request a service limit increase](https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-) for the Amazon EC2 *<type>* instance type. You might need to do this if you already have an existing deployment that uses this instance type, and you think you might exceed the [default limit](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-resource-limits.html) with this reference deployment.

**Launch**

### Step 2. Launch the Quick Start

1. Deploy the AWS CloudFormation template into your AWS account.

The template is launched in the US West (Oregon) region by default. You can change the region by using the region selector in the navigation bar.

This stack takes approximately *<x>* hours to create.

**Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. See the pricing pages for each AWS service you will be using or the [AWS Simple Monthly Calculator](http://calculator.s3.amazonaws.com/index.html) for full details.

You can also download the template to use it as a starting point for your own implementation.

1. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.
2. On the **Specify Details** page, review the parameters for the template. Enter values for the parameters that require your input. For all other parameters, you can customize the default settings provided by the template.

|  |  |  |
| --- | --- | --- |
| Parameter | Default | Description |
| KeyPairName | *Requires input* | Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region. |
| ADInstanceType | m3.xlarge | Amazon EC2 instance type for the first Active Directory instance |
| AD2InstanceType | m3.xlarge | Amazon EC2 instance type for the second Active Directory instance |
| *etc.* |  |  |

When you finish reviewing and customizing the parameters, choose **Next**.

1. On the **Options** page, you can [specify tags](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-resource-tags.html) (key-value pairs) for resources in your stack and [set advanced options](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-add-tags.html). When you’re done, choose **Next**.
2. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.
3. Choose **Create** to deploy the stack.
4. Monitor the status of the stack. When the status is **CREATE\_COMPLETE**, the deployment is complete.
5. You can use the URL displayed in the **Outputs** tab for the stack to view the resources that were created.

### Step 3. Test the Deployment

The steps below walk you through some of the basics with Spinnaker, you're going to set up a Spinnaker pipeline that bakes a Amazon Virtual Image (AMI) image containing redis, then deploys that image to a test cluster.

*Connect to your Spinnaker Instance*

1. Connect to your Spinnaker instance and port forward to Spinnaker Console and APIs:

|  |
| --- |
| ssh -L 9000:localhost:9000 -L 8084:localhost:8084 -L 8087:localhost:8087 [ubuntu@ec2-198-51-100-1.compute-1.amazonaws.com](mailto:ubuntu@ec2-198-51-100-1.compute-1.amazonaws.com) |

1. On your web browser, go to Spinnaker web page <http://localhost:9000>

### *Create a Spinnaker application*

1. In Spinnaker, click **Actions** > **Create Application**
2. Input *example*for the **Name**field and your email address for the **Owner Email**field.
3. Click inside of the dashed rectangle beside the **Accounts** heading and select **Default**.
4. Click on the **Consider only cloud provider health when executing** **tasks** button next to **Instance Health**.
5. Click the **Create** button.

### *Create and configure a security group*

Next, you'll create a security group that specifies traffic firewall rules for the cluster. You'll configure the firewall rules to allow all incoming traffic on port 80, for clusters associated with this security group.

1. Click SECURITY GROUPS, then click the + button to create a security group.
2. Input test for the **Detail (optional)** field and Test environment for the **Description** field.
3. Select **defaultvpc** as the **VPC** field and click **Next**.
4. Click **Add new Security Group Rule**.
5. Click **default** on the **Security Group** dropdown.
6. Change **Start Port** and **End Port** to 80.
7. Click the **Create** button.

### *Create a load balancer*

Next, you'll create a load balancer in Spinnaker.

1. Click **LOAD BALANCERS**, then click the **+** button to create a load balancer.
2. Input test for the **Stack** field.
3. If running on AWS, select **internal (defaultvpc)** from the **VPC**  
   **Subnet** dropdown.
4. Click the **Next** button.
5. Select **example-test** from the **Security Groups** dropdown.
6. Hit **Next**, then **Create**.
7. Click the **Create** button.

### *Create a deployment pipeline*

Your final task is to set up a Spinnaker pipeline. Let's name it **Bake & Deploy to Test**. The pipeline will produce an image containing the redis-server package and then deploy it. In this tutorial, you'll trigger the pipeline manually. To create the pipeline:

1. Click **PIPELINES**, then click **Configure** and select **Create**  
   **New...** from the dropdown.
2. Input Bake & Deploy to Test for the **Pipeline Name**.
3. Click the **Create Pipeline** button.

#### Set up the first stage of the pipeline

You're now going to create the first stage of the pipeline. It will build an image from an existing redis-server package.

1. Click **Add stage**.
2. Select **Bake** from the **Type** pulldown menu.
3. Input redis-server for the **Package** field.
4. Click **Save Changes**.

#### Set up the second stage of the pipeline

You're now going to set up the second stage of the pipeline. It takes the image constructed in the Bake stage and deploys it into a test environment.

1. Click **Add stage**.
2. Select **Deploy** from the **Type** dropdown.
3. Under the **Server Groups** heading, click **Add server group**.
4. Click the **Continue without a template** button.
5. Next, In the **Configure Deployment Cluster** window, input "test" for the **Stack** field.
6. If running on AWS, select **defaultvpc** under **VPC Subnet**.
7. Click the **Next** button.
8. Click the text area next to the **Load Balancers** heading, then select *example-test*. Click the **Next** button.
9. Click the **Security Groups** form field, then click example-test (example-test). Click the **Next** button.
10. Click on the **Micro Utility** button to set the **Instance**  
    **Profile**, then click **Next**.
11. Select the **Medium: m3** size, then click **Next**.
12. Input 2 for the **Number of Instances** field, then click the **Add** button.
13. Save the pipeline configuration by clicking the **Save Changes** button.

### *Try it out!*

1. Click **PIPELINES** in the navigation bar.
2. Click **Start Manual Execution** for the **Bake & Deploy to Test** pipeline.
3. Click **Run**.

Now, watch Spinnaker in action. A **MANUAL START** section will appear, and will show progress as the pipeline executes. At any point during pipeline execution, click on the horizontal bar to see detailed status for any of the stages in the pipeline.

Feel free to navigate around the Spinnaker menus, create new pipelines, clusters, server groups, load balancers, and security groups, etc. and see what happens.

When you're ready to stop, don't forget to cleanup your resources. An easy way to do this is to visit the pipelines, clusters, load balancers, and security groups pages, click on the ones created and select the appropriate **Delete** command from the Actions pulldown on  
the right.

## FAQ

*Any tips or answers to anticipated questions. This could include the following troubleshooting information.*

**Q.** I encountered a CREATE\_FAILED error when I launched the Quick Start. What should I do?

**A.** If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack’s state will be retained and the instance will be left running, so you can troubleshoot the issue. (You'll want to look at the log files in %ProgramFiles%\Amazon\EC2ConfigService and C:\cfn\log.)

**Important** When you set **Rollback on failure** to **No**, you’ll continue to incur AWS charges for this stack. Please make sure to delete the stack when you’ve finished troubleshooting.

For additional information, see [Troubleshooting AWS CloudFormation](http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/troubleshooting.html) on the AWS website or contact us on the [AWS Quick Start Discussion Forum](https://forums.aws.amazon.com/forum.jspa?forumID=178).

See [Spinnaker Troubleshooting Guide](http://www.spinnaker.io/docs/troubleshooting-guide) for more information on troubleshooting and [Frequently Ask Questions](http://www.spinnaker.io/docs/frequently-asked-questions) for more information on Spinnaker.

## Additional Resources

**AWS services**

* AWS CloudFormation  
  <http://aws.amazon.com/documentation/cloudformation/>
* Amazon EBS  
  <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>
* Amazon EC2  
  <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/>
* Amazon VPC  
  <http://aws.amazon.com/documentation/vpc/>

**Netflix Open Source Software Spinnaker**

* Spinnaker home page[*http://www.spinnaker.io*](http://www.spinnaker.io)
* *Spinnaker documentation*[*http://www.spinnaker.io/docs*](http://www.spinnaker.io/docs)
* *Spinnaker Slack Channel*[*https://spinnakerteam.slack.com*](https://spinnakerteam.slack.com)
* *Spinnaker Github*[*https://github.com/spinnaker/spinnaker*](https://github.com/spinnaker/spinnaker)

**Quick Start reference deployments**

* AWS Quick Start home page  
  <https://aws.amazon.com/quickstart/>
* Quick Start deployment guides  
  <https://aws.amazon.com/documentation/quickstart/>

## Send Us Feedback

We welcome your questions and comments. Please post your feedback on the [AWS Quick Start Discussion Forum](https://forums.aws.amazon.com/forum.jspa?forumID=178).

© 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

**Notices**

This document is provided for informational purposes only. It represents AWS’s current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS’s products or services, each of which is provided “as is” without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The software included with this paper is licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at <http://aws.amazon.com/apache2.0/> or in the "license" file accompanying this file. This code is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.