

Netflix OSS Spinnaker on the AWS Cloud

Quick Start Reference Deployment

August 2016

*Huy Huynh and Tony Vattathil
Solutions Architects, Amazon Web Services*

Contents

Overview.....	2
Architecture.....	3
Prerequisites	4
Specialized Knowledge	4
Deployment Steps	5
Step 1. Prepare an AWS Account.....	5
Step 2. Launch the Quick Start	5
Step 3. Test the Deployment	7
FAQ.....	8
Additional Resources	8
Send Us Feedback	9
Document Revisions	9



Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying Spinnaker, which is a continuous deployment tool from [Netflix Open Source Software](#), on the Amazon Web Services (AWS) Cloud. [Quick Starts](#) 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.

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

Spinnaker is used as a continuous delivery platform for releasing software changes. The continuous delivery process begins with the creation of a deployable asset (such as a machine image, a JAR file, or a Docker image) and ends with a deployment to the cloud.

Note Because of the limited availability of the Amazon Machine Image (AMI) for Spinnaker, this Quick Start currently supports four AWS Regions: US East (N. Virginia), US West (N. California), US West (Oregon), and South America (São Paulo). This guide will be updated when more regions become available—please check back.

Spinnaker provides two sets of features:

- Cluster management, to configure, delete, disable, and deploy cloud resources like server groups, security groups, and load balancers, using strategies like blue/green deployments.
- Deployment management, to create and manage continuous delivery workflows called *pipelines*.

Pipelines are configurable, reusable processes that can be triggered by the completion of a Jenkins job, by a CRON expression, or even by another pipeline. Pipelines consist of *stages*, which are operations or phases in the continuous delivery process. Spinnaker comes with a number of stages, such as baking a machine image, deploying an image, running a Jenkins job, or waiting for user approval. Pipeline stages can be run in parallel or serially. For more information on built-in stages, see the [overview](#) on the Spinnaker website.



Figure 1: Deployment management with Spinnaker

Architecture

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

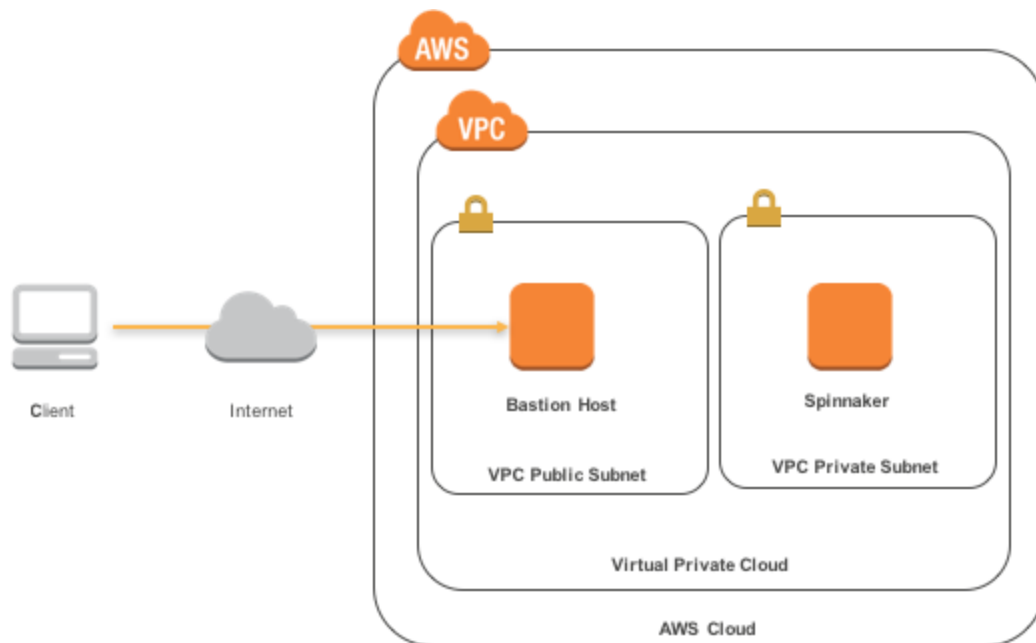


Figure 2: Quick Start architecture for Spinnaker on AWS

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

- The Quick Start creates a virtual private cloud (VPC) with two subnets (one public and one private) in the region you choose when you launch the stack.
 - The security group in the public subnet has only port 22 open for Secure Shell (SSH) access.
 - One Spinnaker instance is deployed into the VPC private subnet.
- The Quick Start creates the following IAM users and roles:
 - **SpinnakerUser** – User account that the end user uses on the Spinnaker instance to access AWS resources.
 - **SpinnakerRole** – Role that the Spinnaker instance uses to access AWS resources. This role has permission to launch other EC2 instances, to pass the **BaseIAMRole** role to other EC2 instances, and to create AWS resources.
 - **BaseIAMRole** – Role that Spinnaker passes 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](#).)

- [Amazon 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](#) – 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](#) – AWS CloudFormation provides an easy way to create and manage a collection of related AWS resources, and to provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources (for example, 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. This Quick Start automates the deployment of Spinnaker on AWS through an AWS CloudFormation template that you can use as is or customize to meet your needs.

- [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.

This guide is meant primarily for the deployment of a Spinnaker cluster on AWS. For detailed information about using Spinnaker, see the [Spinnaker documentation](#).

Deployment Steps

Step 1. Prepare an AWS Account

1. If you don't already have an AWS account, create one at <https://aws.amazon.com> by following the on-screen instructions.
2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy Spinnaker on AWS. (See the note in step 2 about supported regions.)
3. Create a [key pair](#) in your preferred region.
4. Open the IAM console at <https://console.aws.amazon.com/iam/>, create an IAM role called **BaseIAMRole**, and choose Amazon EC2 as the role type. EC2 instances launched with Spinnaker will be associated with this role.
5. If necessary, [request a service limit increase](#) for the Amazon EC2 **m4.xlarge** 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](#) with this reference deployment.

Step 2. Launch the Quick Start

1. [Launch the AWS CloudFormation template into your AWS account](#).

A blue rectangular button with the word "Launch" in white text.

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.

Note Because of the limited availability of the Amazon Machine Image (AMI) for Spinnaker, this Quick Start currently supports four AWS Regions: US East (N. Virginia), US West (N. California), US West (Oregon), and South America (São Paulo). This guide will be updated when more regions become available—please check back.

This stack takes approximately 10 minutes 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 for full details.

You can also [download the template](#) to use it as a starting point for your own implementation.

2. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.
3. 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
KeyName	<i>Requires input</i>	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.
Password	<i>Requires input</i>	Password for the Spinnaker user account. This must be a string consisting of 1-41 alphanumeric characters.
SpinnakerPrivateSubnetCIDR	10.100.11.0/24	CIDR block for the private subnet in the VPC where Spinnaker will be deployed.
SpinnakerPublicSubnetCIDR	10.100.10.0/24	CIDR block for the public subnet in the VPC where Spinnaker will be deployed.
SpinnakerVPCCIDR	10.100.0.0/16	CIDR block for the Amazon VPC for Spinnaker.
SSHLocation	<i>Requires input</i>	The IP range that can be used to whitelist clients connecting to the bastion host using SSH.

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

4. On the **Options** page, you can [specify tags](#) (key-value pairs) for resources in your stack and [set advanced options](#). When you're done, choose **Next**.
5. 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.
6. Choose **Create** to deploy the stack.

- Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the deployment is complete.
- In the **Outputs** tab, copy the values of SSHString1 and SSHString2 for subsequent steps.

Key	Value	Description
PrivateDNS	ip-10-100-10-182.us-west-2.compute.internal	Private DNSName of Spinnaker web server.
SSHString2	ssh -L 9000:localhost:9000 -L 8084:localhost:8084 -L 8087:localhost:8087 ubuntu@10.100.10.182	SSH String
SSHString1	ssh -A -L 9000:localhost:9000 -L 8084:localhost:8084 -L 8087:localhost:8087 ec2-user@52.43.220.121	SSH String
PublicDNS	ec2-52-43-220-121.us-west-2.compute.amazonaws.com	Public DNSName of bastion host.

Figure 3: Outputs tab

Step 3. Test the Deployment

In this section, you'll connect to your Spinnaker instance and test the deployment by creating a Spinnaker bake and deploy pipeline.

- Connect to your Spinnaker instance, and port forward to the Spinnaker console and APIs. To do this, execute SSHString1 (from the **Outputs** tab) in the console to connect to the Bastion host.

```
ssh -A -L 9000:localhost:9000 -L 8084:localhost:8084 -L 8087:localhost:8087 ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

- Execute SSHString2 (from the **Outputs** tab) in the stdout of the bastion host.

```
ssh -L 9000:localhost:9000 -L 8084:localhost:8084 -L 8087:localhost:8087 ubuntu@10.100.10.182
```

- In your web browser, open the Spinnaker web page <http://localhost:9000>.
- Follow the [tutorial](#) in the Spinnaker documentation to set up a Spinnaker pipeline that bakes an Amazon Machine Image (AMI) containing a Redis server, and then deploys that image to a test cluster.

FAQ

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](#) on the AWS website or contact us on the [AWS Quick Start Discussion Forum](#).

See the [Spinnaker Troubleshooting Guide](#) for more information on troubleshooting Spinnaker-related issues, and see [Frequently Asked Questions](#) for more information on Spinnaker.

Additional Resources

AWS services

- AWS CloudFormation
<http://aws.amazon.com/documentation/cloudformation/>
- Amazon EC2
<http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/>
- Amazon VPC
<http://aws.amazon.com/documentation/vpc/>
- IAM
<http://aws.amazon.com/documentation/iam/>

Spinnaker

- Spinnaker home page
<http://www.spinnaker.io>

- Spinnaker documentation
<http://www.spinnaker.io/docs>
- Spinnaker Slack Channel
<https://spinnakerteam.slack.com>
- Spinnaker Github
<https://github.com/spinnaker/spinnaker>

Quick Start reference deployments

- AWS Quick Start home page
<https://aws.amazon.com/quickstart/>

Send Us Feedback

You can visit our [GitHub repository](#) to download the templates and scripts for this Quick Start, to post your comments, and to share your customizations with others.

Document Revisions

Date	Change	In sections
August 2016	Initial publication	—

© 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.