Chef Server on the AWS Cloud

Quick Start Reference Deployment

Mike Pfeiffer

December 2015

This guide is also available in HTML format at
http://docs.aws.amazon.com/quickstart/latest/chef-server/.

Contents

About This Guide

This Quick Start reference deployment guide discusses configuration steps for deploying Chef Server on the Amazon Web Services (AWS) cloud. It also provides links for viewing and launching AWS CloudFormation templates that automate the deployment, and a walkthrough that demonstrates how you can manage Amazon Elastic Compute Cloud (Amazon EC2) instances with Chef.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their Chef workloads on the AWS cloud.

> **AWS OpsWorks option** This Quick Start is for customers who want to run and manage their own Chef Server infrastructure. However, we recommend that you also take a look at AWS OpsWorks, which is a configuration management service provided by AWS, to determine if it's more suitable for your needs. AWS OpsWorks helps you configure and operate applications of all types and sizes using Chef. You can define the application's architecture and the specification of each component, including package installation, software configuration, and resources such as storage. For more information, see the AWS OpsWorks User Guide.

Quick Starts are automated reference deployments for key workloads on the AWS cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

Overview

Chef Server on AWS

Chef Server is the highly scalable foundation of the Chef automation platform. You can use Chef Server to create and manage dynamic infrastructure that runs on the AWS cloud, or manage the servers in your on-premises data center. This Quick Start uses the official Chef Server AMI from the AWS Marketplace to make it easy to automate your infrastructure. You can launch the Chef Server AMI from either the AWS Marketplace or from this Quick Start with a single click to get instant access to the following features:

- Chef Server, including Chef Analytics, Chef Management Console, and Chef Reporting
- Consolidated hourly billing for both Chef premium features and your infrastructure running on AWS
- The ability to manage from 5 to 250 nodes

This Quick Start is for customers who are planning to move to AWS, or are already running on AWS, and also want to deploy and manage their own Chef Server infrastructure. In addition, the goal of this guide is to help you get started with Chef, even if you have no prior experience with the product.

This Quick Start automates the launch of the Chef Server AMI, performs Chef Server initial setup, creates the Chef administrative user account, and enables HTTPS access over the Internet and within your Amazon Virtual Private Cloud (Amazon VPC). If you are new to Chef, you can choose to include an optional Chef workstation and managed node. You can follow the walkthrough included in this guide to learn how to configure the Chef workstation, create and upload a cookbook, and bootstrap an EC2 instance with Knife, which is a Chef command-line tool that helps you manage infrastructure components.

Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, configuration, network security, and other considerations discussed in this guide.

The default configuration deploys each server using the **t2.medium** instance type, but you can change the instance type and customize other settings. The supported instance types for Chef Server are listed on the associated product pages in the AWS Marketplace (see the links in the next section).

View template

Launch Quick Start

Time to deploy: Approximately 35 minutes

Cost and Licenses

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 the Quick Start. As of the date of publication, the cost for using the Quick Start with default settings is approximately $0.18 an hour. Prices are subject to change. See the pricing pages for each AWS service you will be using in this Quick Start for full details.

Because this Quick Start uses AMIs from the AWS Marketplace, you must subscribe to Chef Server from the AWS Marketplace before you launch the Quick Start. There are six licensing options available, depending on how many nodes you want to manage. You can use the following links to visit the listing for each Marketplace AMI to see the hourly usage costs that are included with your EC2 instance.

- 5 nodes
- 50 nodes
- 100 nodes
- 150 nodes
- 200 nodes
- 250 nodes

Chef Server, the Chef workstation, and the node deployed by this Quick Start use the **t2.medium** instance type by default.

> **Important** If you try to launch the Quick Start without first subscribing to the Chef Server AMI from the AWS Marketplace, the Chef Server instance will not be created, and the stack creation will fail and roll back.

Architecture Overview

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

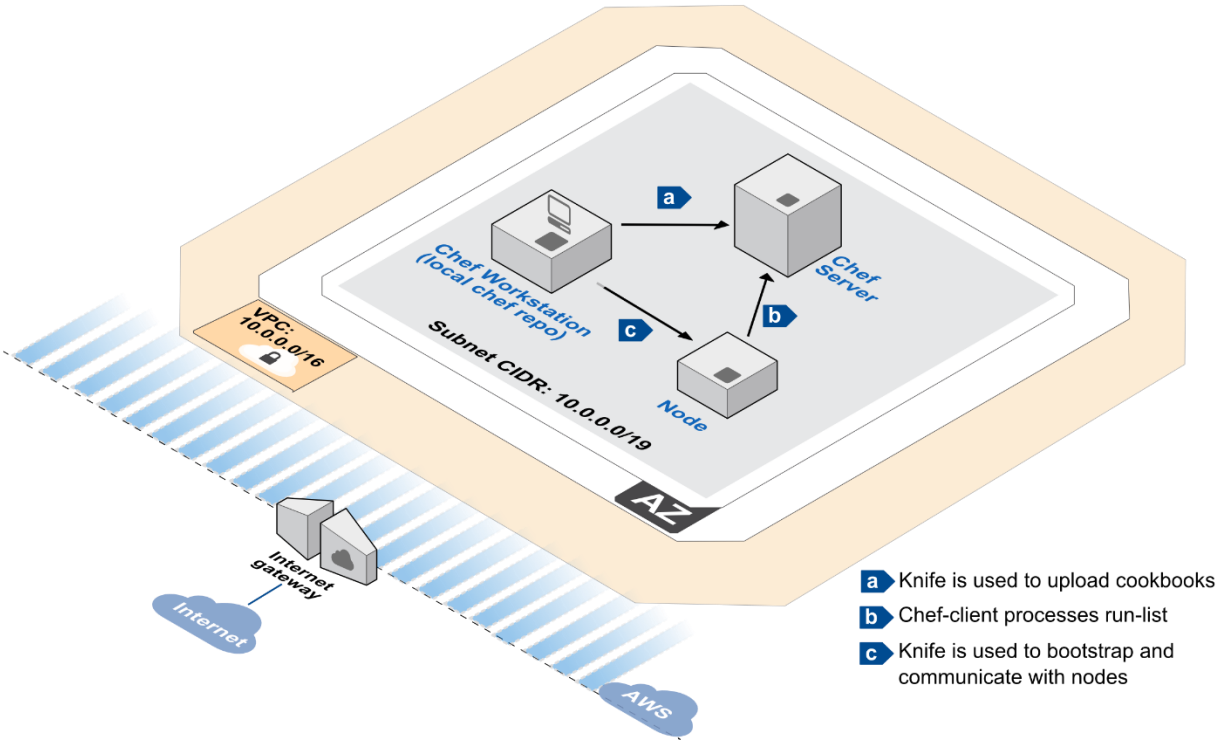

Figure 1: Quick Start Architecture for Chef Server on AWS

The resources deployed by this Quick Start and shown in Figure 1 are used as follows:

- An Amazon VPC is created in the region you choose when you launch the stack. A single public Amazon VPC subnet is created in the first Availability Zone.
- Chef Server is deployed into the VPC subnet. An Elastic IP address (EIP) is allocated and associated with the instance. During instance launch, Chef Server is bootstrapped and the **marketplace-setup** command is run to configure the server. You provide the values required by **marketplace-setup** via AWS CloudFormation parameters at the time you launch the stack.
- A Chef workstation running on Microsoft Windows Server is deployed into the VPC subnet. This server has a public IP assigned and is accessible via Remote Desktop Protocol (RDP) over the Internet. Both Git and the Chef Development Kit (Chef DK) are automatically installed on this machine via the AWS CloudFormation bootstrapping process. This Chef workstation is optional. You can use a workstation in your own on-premises environment as an alternative. This Quick Start uses Windows Server for the Chef workstation, because Windows provides a graphical user interface to the Chef Client tools and web-based Management Console, but you are free to use any supported operating system for your own Chef workstation.

- One Ubuntu Server node is deployed into the VPC subnet. After you deploy Chef on AWS, you can follow the walkthrough in this guide to configure a local Git repository (chef-repo) on the workstation, create and upload a cookbook to Chef Server, and then bootstrap the node and run the cookbook to configure a basic web server. As with the Chef workstation, this node is optional. With your Chef Server running on AWS, you can manage other nodes that you deploy on AWS or nodes that are located in your on-premises data center.

AWS Services

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the Getting Started section of the AWS documentation.)

- 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.
- Amazon Marketplace – AWS Marketplace is an online store where you can sell or buy software that runs on AWS. AWS Marketplace complements programs like Amazon DevPay and the AWS Partner Network (APN).

Use Cases and Benefits of Chef on AWS

Running your Chef Server on AWS provides advantages in pricing, automation, resource management, and other areas, as explained in the following sections.

Usage-based Pricing and Consolidated Billing

Running your Chef Server on Amazon EC2 gives you the ability to leverage hourly billing, and to pay only for what you use. With this Quick Start and the Marketplace AMI, you can choose the number of nodes you need support for, and the Chef Server licensing costs will be included in your hourly rate for running the instance. If you're already running your infrastructure on AWS, or if you're in the process of migrating your workloads to AWS,

launching the Chef Server AMI through this Quick Start or through the AWS Marketplace allows you to consolidate your Chef and AWS costs into a single monthly bill.

Hybrid Deployments

If you're managing servers both on premises and in the cloud, Chef can fully support automation in your hybrid deployment. Chef Server running on Amazon EC2 can be used with other EC2 instances and with servers you have running in your own data center. This Quick Start makes your Chef Server Internet-accessible, and supports automated hybrid deployments out of the box.

AWS Resource Management

Chef has the ability to manage AWS resources. If you want to use a single unified tool to manage your entire infrastructure, you can directly manage EC2 instances and leverage cookbooks to manage other resources, such as security groups, load balancers, Amazon Elastic Block Store (Amazon EBS) volumes, Elastic IP addresses (EIPs), and tags. Take a look at Chef's aws cookbook on Chef Supermarket. This cookbook provides libraries, resources, and providers to configure and manage AWS components and offerings within the Amazon EC2 API.

Integration with AWS CloudFormation

With Chef, you can automate the deployment of your software applications on Amazon EC2 instances instead of manually writing and executing various scripts. By combining Chef with AWS CloudFormation, you can consistently deploy and configure your AWS resources, along with the software applications that run on top of AWS, all from a single AWS CloudFormation template. For a detailed walkthrough, see the Using Chef with AWS CloudFormation whitepaper provided by AWS.

Chef High Availability

Chef High Availability (HA) is a premium Chef feature that allows you to eliminate a single point of failure for your deployment. Chef HA includes a plugin for AWS that can move a virtual IP address and EBS volume from one Chef back end to another. For details, see High Availability on AWS in the Chef documentation.

Automated Deployment

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of Chef Server on the AWS cloud from

scratch. Follow the step-by-step instructions in this section to set up your AWS account, customize the template, and deploy the software into your account.

What We'll Cover

The procedure for deploying the Chef Server architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Prepare an AWS account

- Sign up for an AWS account, if you don't already have one.
- Choose the region where you want to deploy the stack on AWS.
- Create a key pair in the region.
- Review account limits for Amazon EC2 instances, and request a limit increase, if needed.

Step 2. Subscribe to the Chef Server AMI

- Visit the AWS Marketplace and locate the Chef AMI.
- Sign in to your AWS account and accept the license terms.
- Wait for email confirmation.

Step 3. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Enter a value for the *KeyPairName* parameter.
- Review the other template parameters, and customize their values if necessary.

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. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.
2. Use the region selector in the navigation bar to choose the Amazon EC2 region where you want to deploy Chef Server on AWS.

   Amazon EC2 locations are composed of *regions* and *Availability Zones*. Regions are dispersed and located in separate geographic areas.

Figure 2: Choosing an Amazon EC2 Region

> **Tip** Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

3. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose **Key Pairs**, **Create Key Pair**, type a name, and then choose **Create**.

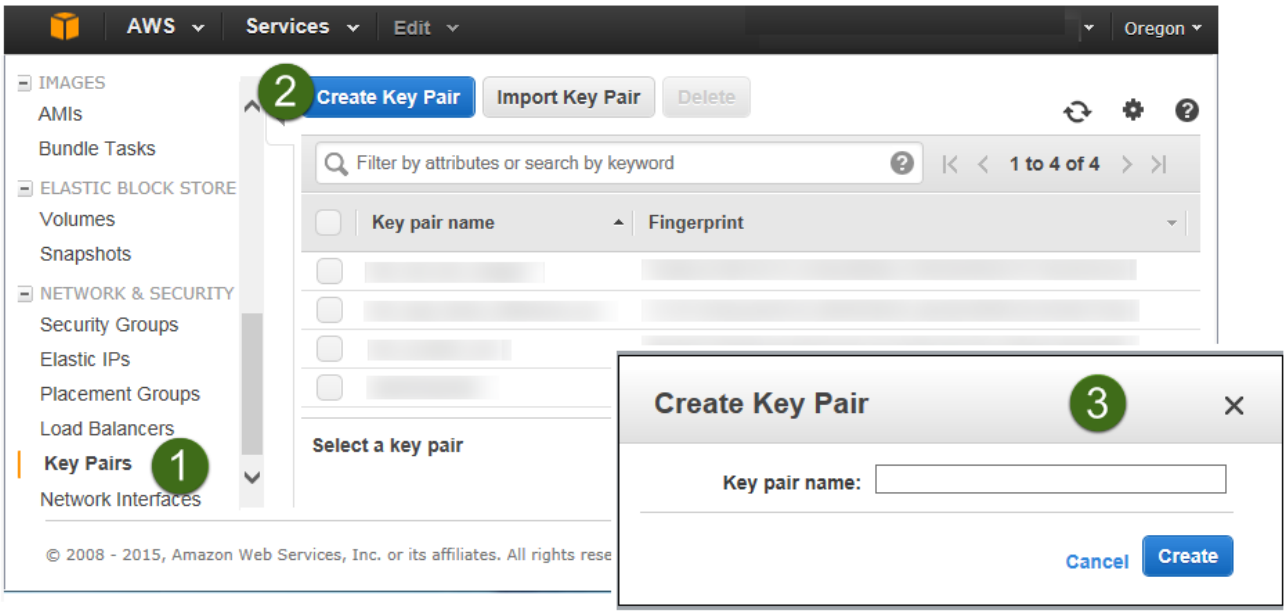

Figure 3: Creating a Key Pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. With Windows instances, we use the key pair to obtain the administrator password via the Amazon EC2 console and then log in using Remote Desktop Protocol (RDP) as explained in the step-by-step instructions in the *Amazon Elastic Compute Cloud User Guide*. On Linux, we use the key pair to authenticate SSH login.

4. If necessary, request a service limit increase for the Amazon EC2 **t2.medium** instance type. To do this, in the AWS Support Center, choose **Create Case**, **Service Limit Increase**, **EC2 instances**, and then complete the fields in the limit increase form. The current default limit for this instance type is 20 instances.

   You might need to request an increase 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. It might take a few days for the new service limit to become effective. For more information, see Amazon EC2 Service Limits in the AWS documentation.

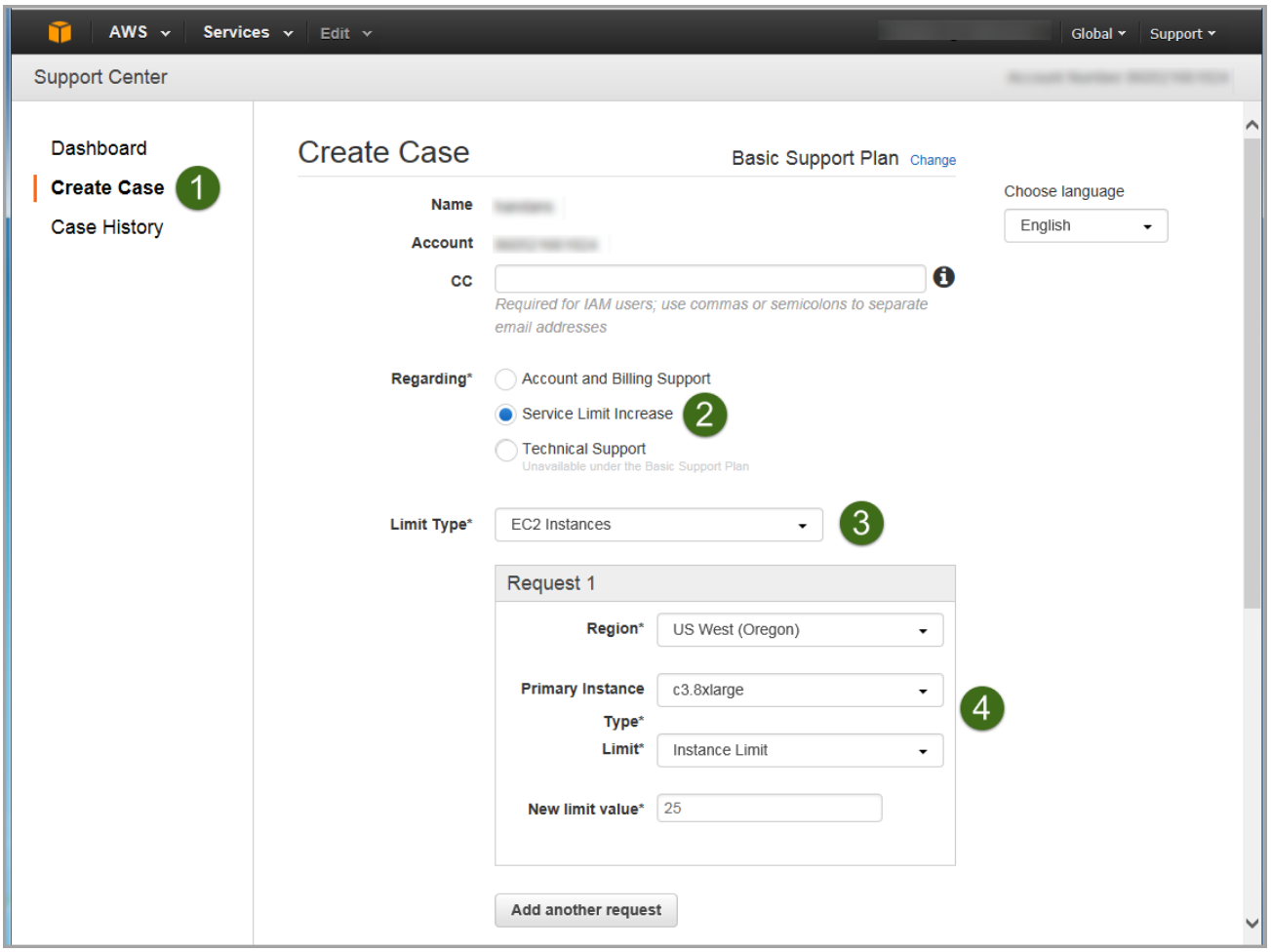

Figure 4: Requesting a Service Limit Increase

Step 2. Subscribe to the Chef Server AMI

1. Visit the AWS Marketplace at https://aws.amazon.com/marketplace and select the version of the Chef Server AMI that supports the number of nodes specific to your environment. For this example, we'll select Chef Server with 5 free nodes.

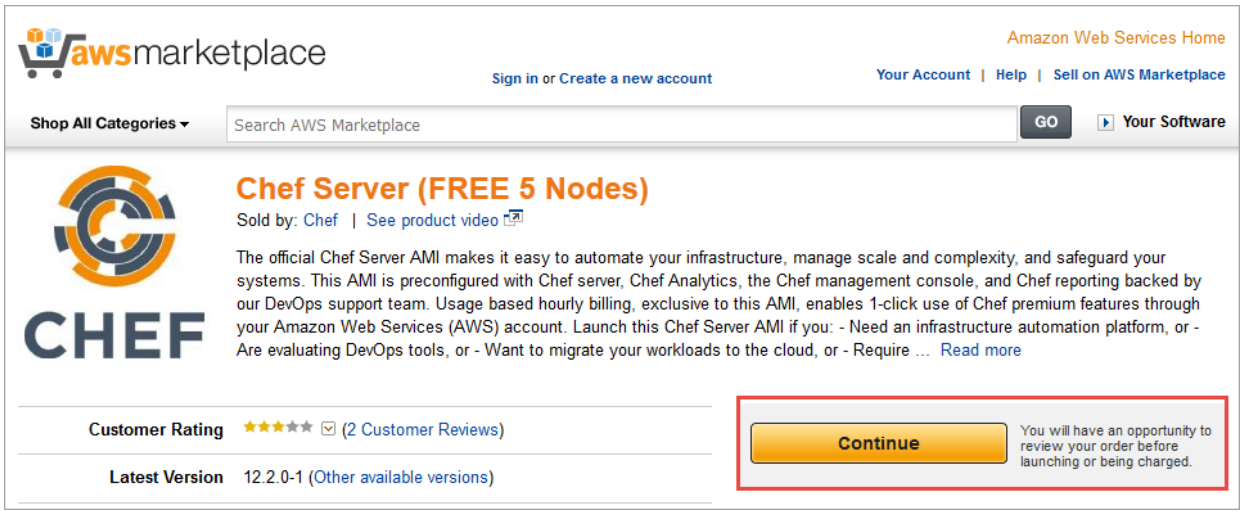

Figure 5: Selecting the Chef Server AMI

2. Sign in to your AWS account. Select the **Manual Launch** option for the AMI and choose **Accept Terms**.

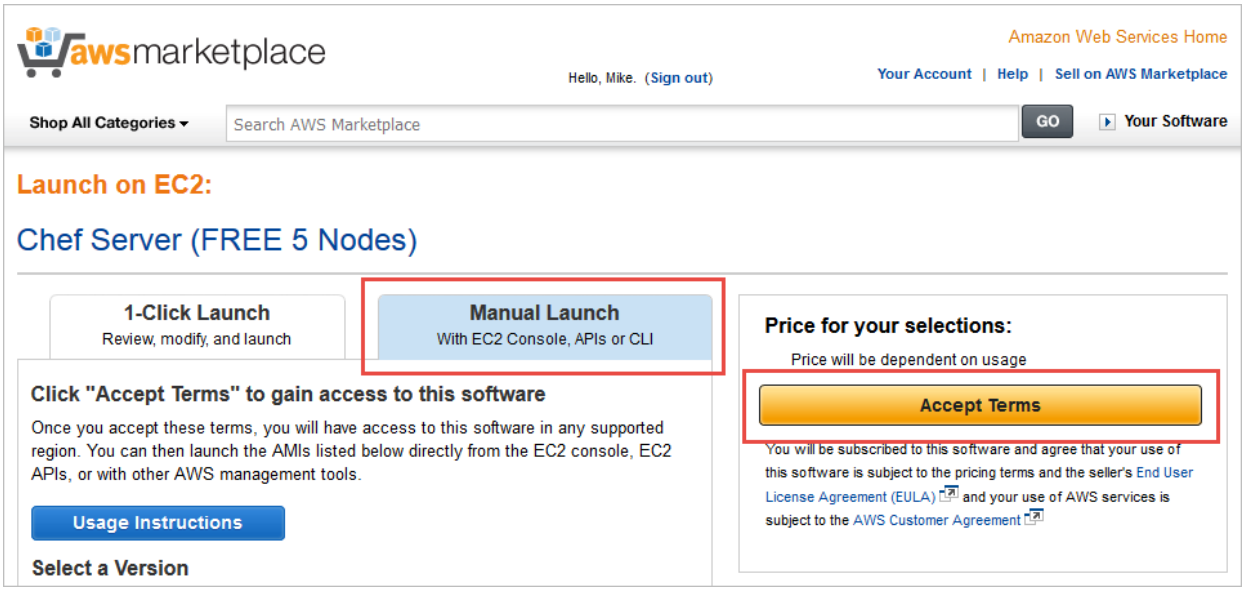

Figure 6: Accepting the Terms of the Chef Server AMI

After you accept the terms, you'll receive a notice that your subscription will be completed in a few moments.

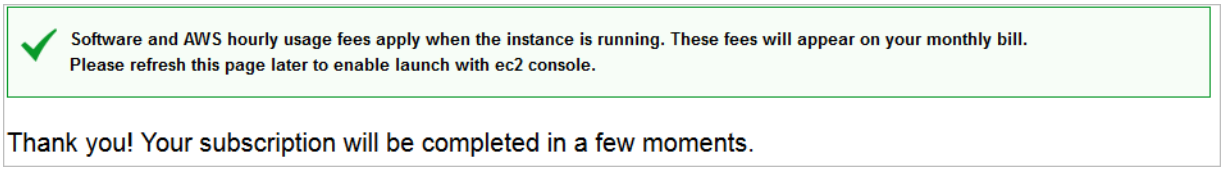

Figure 7: Chef Server AMI Subscription Confirmation

3. Wait for email confirmation from the AWS Marketplace notifying you that you are able to use this software in AWS. After you receive this confirmation, you are ready to launch the Chef stack.

Step 3. Launch the Chef Stack

This automated AWS CloudFormation template deploys Chef Server into an Amazon VPC. Please make sure that you've completed the previous step and have subscribed to the Chef Server AMI in the AWS Marketplace before launching the stack.

Launch stack

1. Launch 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 35 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. As of the date of publication, the cost for using the Quick Start with default settings is approximately $0.18 an hour, and you can complete the initial deployment for about $0.18. Prices are subject to change. See the pricing pages for each AWS service you will be using in this Quick Start 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 URL for the AWS CloudFormation template, and then choose **Next**.
3. On the **Specify Details** page, review the parameters for the template. These are described in the following table.

   Provide a value for the *KeyPairName* parameter. This parameter requires your input. For all other parameters, the template provides default settings that you can customize.

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.
AdminEmailAddress	admin@localhost.internal	Email address for the Chef administrator.
AdminPassword	Password123	Password for the "chefadmin" user for signing in to the Chef workstation and Management Console. This must be a complex password that's at least 8 characters long.
ChefNodes	5	Select the number of nodes you want to support (5 to 250).
IncludeDemoInstances	true	Keep the default value to include a Chef workstation and demo node. To deploy Chef Server only, set this parameter to **false**.
RemoteAdminCIDR	0.0.0.0/0	CIDR block or IP for SSH and RDP access.
SubnetCIDR	10.0.0.0/19	CIDR block for the subnet CIDR.
VPCCIDR	10.0.0.0/16	CIDR block for the Amazon VPC.

Note You can also download the template and edit it to create your own parameters based on your specific deployment scenario.

4. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set additional options. When you're done, choose **Next**.
5. On the **Review** page, review and confirm the template settings.
6. Choose **Create** to deploy the stack.
7. Monitor the status of the stack. When the status displays **CREATE_COMPLETE**, Chef Server is ready.

Configuring a Chef Node

If you chose the default setting for the *IncludeDemoInstances* parameter, you can follow this walkthrough to test your Chef Server setup. In this section, we'll explain how to finalize the Chef workstation setup, create a cookbook, bootstrap a node, and verify that the configuration was applied.

Setting up the Workstation and chef-repo

1. Use the Remote Desktop Protocol (RDP) client to connect to the Chef workstation. You can obtain the public DNS name or IP for the instance tagged as **ChefWorkstation** in the Amazon EC2 console. Use **chefadmin** for the user name and the password you provided when you launched the stack in step 3.
2. On the desktop, open the context (right-click) menu for **Chef Development Kit**, and choose **Run as administrator**.

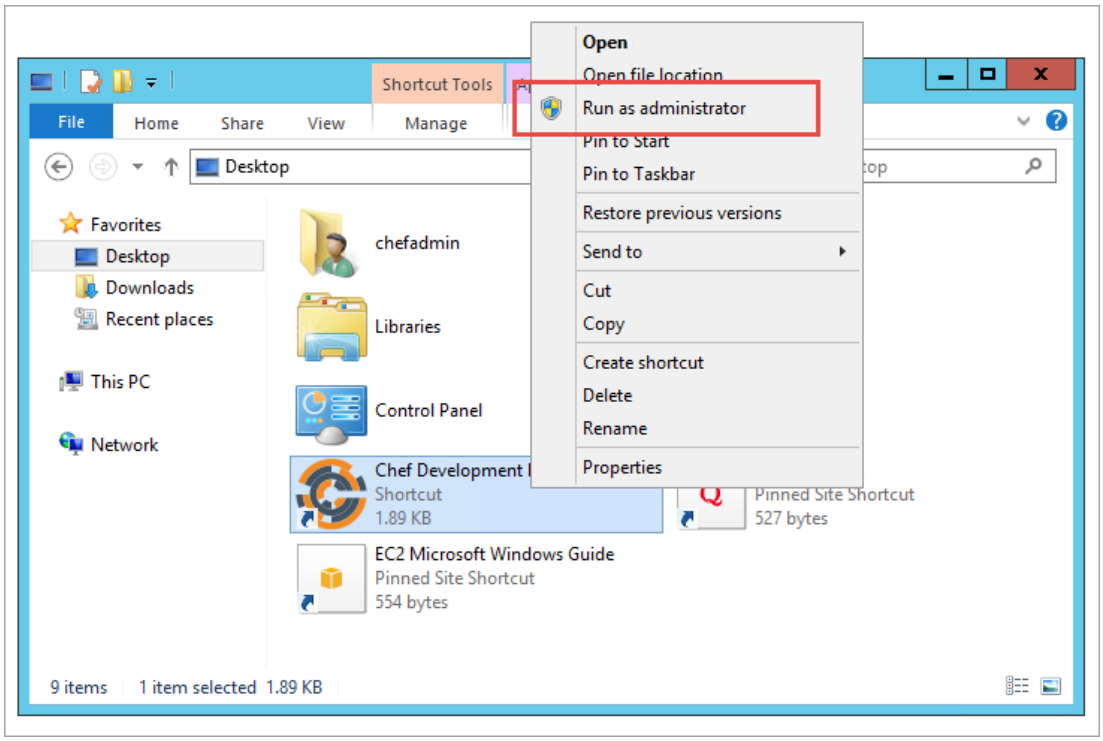

Figure 8: Launching the Chef Development Kit (ChefDK) from the Desktop

This will bring up the ChefDK console, which runs via Windows PowerShell.

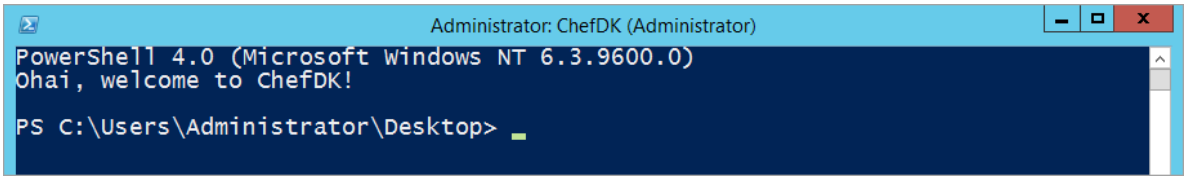

Figure 9: The ChefDK Console

3. At the prompt, change to the root of drive C:

```
CD c:\
```

4. Run the following command to generate the chef-repo. This will be the Git repository used to store and version-control your cookbooks.

```
chef generate repo chef-repo
```

5. Retrieve the public DNS name from the instance tagged as **ChefServer** from the Amazon EC2 console (e.g., `ec2-x-x-x-x.compute-1.amazonaws.com`). Use the Google Chrome web browser installed on the workstation to navigate to the Chef Management Console. You'll need to add "https://" in the address bar, followed by the DNS name for the Chef Server. Log in to the site by using the **chefadmin** user name and the password you specified when you launched the AWS CloudFormation stack in step 3.

Figure 10: Logging in to the Chef Management Console

6. From the top toolbar, choose the organization name (which will be **aws**) and choose **Manage Organizations** from the list.

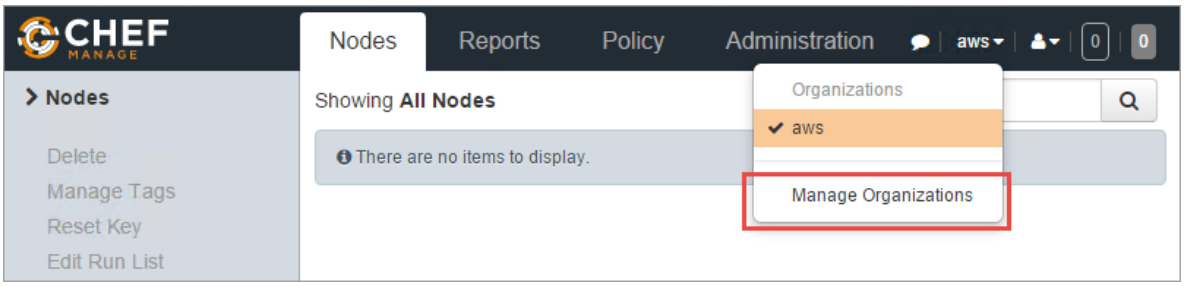

Figure 11: Managing your Chef Organization

7. Highlight the organization name. Choose the gear icon in the upper right, and then choose **Starter Kit**.

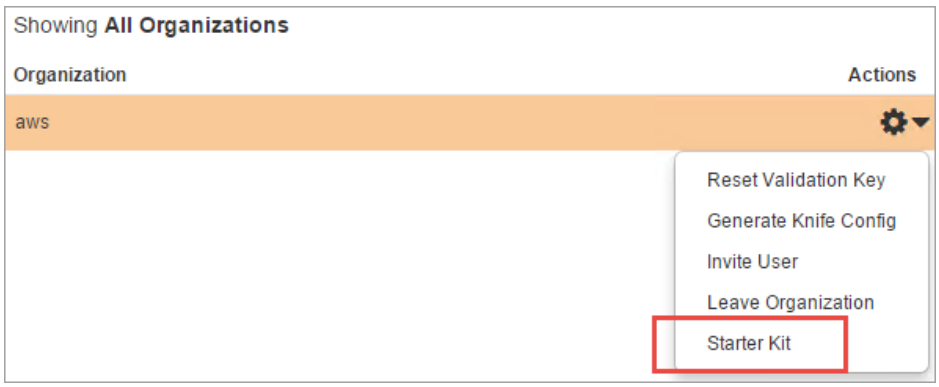

Figure 12: Downloading the Chef Starter Kit

8. Choose **Download Starter Kit**, and then choose **Proceed** in the confirmation box.

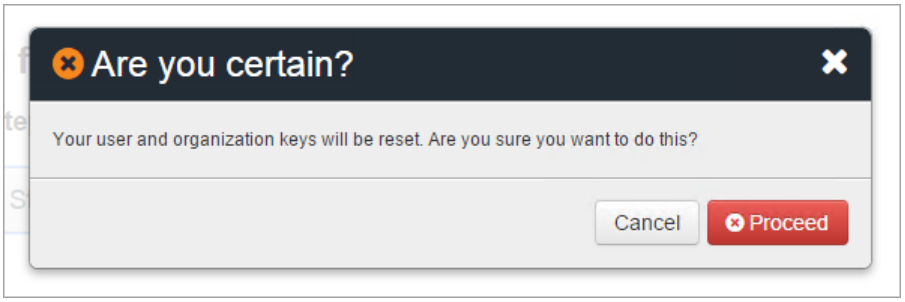

Figure 13: Proceeding with the Download

9. Back in the ChefDK console, run the following PowerShell commands to unzip the Chef Starter Kit. If you downloaded the archive to another location, update the source path as needed.

```
$source = 'C:\users\chefadmin\downloads\chef-starter.zip'
$dest = 'c:\chef-starter'
Expand-Archive -Path $source -DestinationPath $dest
```

10. Copy the **.chef** folder from the extracted Chef Starter Kit to the root of your chef-repo folder:

```
$source = 'c:\chef-starter\chef-repo\.chef'
Copy-Item -Path $source -Destination .\chef-repo -Recurse
```

11. Set the location of ChefDK console to the chef-repo folder:

```
CD .\chef-repo
```

After this point, you must run all **knife** commands from this directory location. Do not switch (CD) to another location, or the **knife** commands will fail.

12. Run the **knife ssl fetch** command to add the self-signed certificate on the Chef Server to the trusted certs folder in your chef-repo. Replace the DNS name with the public DNS name of your Chef Server.

```
knife ssl fetch https://ec2-x-x-x-x.compute-1.amazonaws.com
```

For production environments, you can replace the self-signed certificate with one that is issued from a trusted certification authority (CA).

Creating a Cookbook and Recipe

Now that the Chef workstation is fully configured, you're ready to create a cookbook.

1. In the ChefDK console, create a cookbook named *webserver* for a web server:

```
knife cookbook create webserver
```

2. To set up the cookbook, you can use sample files that are included with this Quick Start. Copy the provided recipe into your new cookbook:

```
$source = 'c:\cfn\downloads\default.rb'
$dest = 'c:\chef-repo\cookbooks\webserver\recipes'
Copy-Item -Path $source -Destination $dest
```

3. Copy a basic default web page into your *webserver* cookbook. This file will be used to show a "hello world" message to visitors who navigate to the web server after you've configured your node.

```
$source = 'c:\cfn\downloads\index.html'
$dest = 'c:\chef-repo\cookbooks\webserver\files'
Copy-Item -Path $source -Destination $dest
```

4. Open the recipe file in a text editor such as Notepad++ to review the code. The recipe defines a series of resources that will execute in order to configure your node as a web server.

```
10  #Run apt-get update
11
12  execute "apt-get update" do
13      command "apt-get update"
14  end
15
16  #Install apache
17
18  package "apache2" do
19      action :install
20  end
21
22  #Start apache service and enable service
23
24  service "apache2" do
25      action [:start, :enable]
26  end
27
28  cookbook_file "/var/www/html/index.html" do
29      source "index.html"
30      mode "0644"
31  end
```

Figure 14: The webserver Recipe

Note the following about the code listed in Figure 14:

- Line 12 – The **execute** resource runs a single command—in this case, the **apt-get update** command. This command downloads package lists and updates them with the newest versions of packages and their dependencies.
- Line 18 – The **package** resource uses the appropriate package manager on the node to take the specified action on the package. In this case, we're installing the Apache 2.0 web server.
- Line 24 – The **service** resource starts the Apache service after it's installed, and then enables the service for automatic startup. The Linux distribution launched by this Quick Start already handles the automatic startup, but it's provided here for completeness, so you can use it as the basis for another server if necessary.
- Line 28 – The **cookbook_file** resource uses the index.html file you copied into the files folder of your *webserver* cookbook, and deploys it to the web server.

This is a very simple recipe that uses some of the common Chef resources. For more information about developing cookbooks, see the Additional Resources section.

5. Now that the cookbook is fully configured, upload it to Chef Server:

```
knife cookbook upload webserver
```

Bootstrapping a Node

Next, you can use the **knife** command to bootstrap the node and execute the run-list for the node, which will contain the *webserver* cookbook.

1. Since **knife** will use SSH to communicate with the node, you'll need to provide your AWS private key. For example, if your private key is called **MyKey**, copy the MyKey.pem file to the c:\chef-repo\.chef folder. The .chef folder contains the .pem files for Chef Server as well, and the .gitignore file in your chef-repo excludes the .pem file from Git commit operations.
2. You can now use the **knife bootstrap** command to bootstrap the node. Retrieve the private IP address for the Ubuntu Server tagged "NODE" in the Amazon EC2 console. Run the following command (replacing x.x.x.x with the private IP of your node, and mykey.pem with the name of your .pem file) to bootstrap the node and execute the run-list.

> **Note** The **knife bootstrap** command should be entered on one line in the ChefDK console. It's broken up here for readability.

```
knife bootstrap x.x.x.x
  --ssh-user ubuntu
  --sudo
  --identity-file ./.chef/mykey.pem
  --run-list webserver
```

After executing the command you'll see the node bootstrap, and the run-list will execute and configure the node as a web server. At that point, you can navigate to the node's IP address in a web browser, where you will see a "hello world" message confirming that the configuration has been applied successfully.

```
resolving cookbooks for run list: ["webserver"]
Synchronizing Cookbooks:
  - webserver (0.1.0)
Compiling Cookbooks...
Converging 4 resources
Recipe: webserver::default
  * execute[apt-get update] action run
    - execute apt-get update
  * apt_package[apache2] action install
    - install version 2.4.7-1ubuntu4.8 of package apache2
  * service[apache2] action start (up to date)
  * service[apache2] action enable (up to date)
  * cookbook_file[/var/www/html/index.html] action create
    - update content in file /var/www/html/index.html from 538f31
```

Figure 15: Reviewing the Output when Bootstrapping the Node

You can configure the node to run chef-client on a regular basis to process future changes that you might make, such as modifying the cookbook, or customizing the run-list. To do so, schedule the chef-client to run at your desired interval via Cron.

Security

A *security group* acts as a firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. The new rules are automatically applied to all instances that are associated with the security group.

The security groups created and assigned to the individual instances as part of this solution are restricted as much as possible while allowing access to the various functions needed by Chef Server. We recommend that you review security groups and further restrict access as needed once the deployment is up and running.

Support

Chef

The Chef Server AMI includes a fast-response support channel available Monday through Friday, 6 A.M. to 6 P.M. Pacific Time, with experienced DevOps engineers. You can email the Chef Support team at aws@chef.io. See Chef Support Service Level Agreements for more details.

AWS

AWS Quick Start reference deployments help you rapidly deploy fully functional software on the AWS cloud, following AWS best practices. AWS CloudFormation templates automate the deployment, and the guides describe architecture and implementation.

- To extend the templates and scripts, visit our GitHub repository.
- Post your feedback on the AWS Quick Start discussion forum.
- For answers to frequently asked questions, see the Quick Start FAQ.

Additional Resources

AWS services

- Amazon EC2
  https://aws.amazon.com/documentation/ec2/
- AWS CloudFormation
  https://aws.amazon.com/documentation/cloudformation/
- Amazon VPC
  https://aws.amazon.com/documentation/vpc/
- AWS OpsWorks
  https://aws.amazon.com/documentation/opsworks/

Chef resources

- Chef on the AWS Marketplace
  https://aws.amazon.com/marketplace/seller-profile/ref=dtl_pcp_sold_by?ie=UTF8&id=e7b7691e-634a-4d35-b729-a8b576175e8c
- Learn Chef
  https://learn.chef.io/
- AWS Cookbook on Chef Supermarket
  https://supermarket.chef.io/cookbooks/aws
- Bootcamps at the AWS Loft
  https://www.chef.io/blog/events/?action=tribe_list&tribe_paged=1&tribe-bar-search=Loft
- Chef High Availability on AWS
  https://docs.chef.io/install_server_ha_aws.html
- Integrating AWS CloudFormation with Chef
  https://s3.amazonaws.com/cloudformation-examples/IntegratingAWSCloudFormationWithOpscodeChef.pdf
- Use Chef Cookbooks and Recipes with AWS OpsWorks
  http://docs.aws.amazon.com/opsworks/latest/userguide/workingcookbook.html

AWS 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.

You can visit our GitHub repository to download the templates and scripts for this Quick Start, and to share your customizations with others.

Document Revisions

Date	Change	In sections
December 2015	Initial publication	-

© 2015, 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.