Magento on the AWS Cloud

Quick Start Reference Deployment

Karthik Krishnan

September 2015

amazon
webservices

Contents

About This Guide

This Quick Start reference deployment guide includes architectural considerations and configuration steps for deploying a Magento Community Edition cluster on the Amazon Web Services (AWS) cloud. Magento is an open-source content management system for e-commerce websites. Magento is licensed under the Open Software License (OSL 3.0).

This guide discusses best practices for deploying Magento on AWS using services such as Amazon Elastic Compute Cloud (Amazon EC2), Amazon Virtual Private Cloud (Amazon VPC), Amazon Relational Database Service (Amazon RDS), and Amazon Simple Storage System (Amazon S3). It also provides links to automated AWS CloudFormation templates that you can leverage for your deployment or launch directly into your AWS account.

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

Quick Starts are automated reference deployments for key enterprise 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

Magento on AWS

Magento is an open-source content management system for e-commerce websites. AWS enables you to set up the infrastructure to support Magento deployment in a flexible, scalable, and cost-effective manner on the AWS cloud. This reference deployment will help you rapidly build a Magento Community Edition cluster by automating configuration and deployment tasks.

The automated deployment builds a cluster that runs Magento version 1.9.2 along with optional sample data.

This guide covers the deployment of Magento Community Edition 1.9.2 on the AWS cloud. It doesn't provide Magento product usage information. For general guidance and best practices for using Magento, see the Magento Community Edition User Guide on the Magento website.

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.

> **Note**   These links set up a new Amazon Virtual Private Cloud (Amazon VPC). To deploy Magento into an existing Amazon VPC, see the [Automated Deployment](#) section of this guide.

View template

Launch Quick Start

The template includes default settings that you can customize by following the instructions in this guide.

Time to deploy: Approximately 15 minutes for the baseline Magento deployment; up to 30 minutes when you choose to install sample data

Costs

This deployment launches Magento Community Edition 1.9.2 automatically into a configuration of your choice. 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. The cost will vary depending on the storage and compute configuration of the cluster you deploy.

The following table provides cost estimates as of the date of publication for some of the configuration options.

Web Server Instance (Number of Instances)	Amazon RDS MySQL Deployment Type	Amazon RDS Storage (GiB)	Amazon RDS Instance Type	Storage Type	Provisioned IOPS (PIOPS)	Cost/ Month* ($)
t2.micro (1)	Single-AZ	5	db.t2.micro	gp2	n/a	0**
m3.2xlarge (2)	Single-AZ	16	db.m3.2xlarge	gp2	n/a	1,321
m4.4xlarge (2)	Single-AZ	16	db.m4.4xlarge	gp2	n/a	2,165
c3.4xlarge (2)	Single-AZ	500	db.m3.2xlarge	io1	500	1,959
c3.4xlarge (2)	Multi-AZ	500	db.m3.2xlarge	io1	500	2,838
r3.4xlarge (4)	Multi-AZ	1,000	db.r3.4xlarge	io1	1,000	5,925

Web Server Instance (Number of Instances)	Amazon RDS MySQL Deployment Type	Amazon RDS Storage (GiB)	Amazon RDS Instance Type	Storage Type	Provisioned IOPS (PIOPS)	Cost/ Month* ($)
r3.8xlarge (4)	Multi-AZ	1,000	db.r3.8xlarge	io1	1,000	15,118

*Prices are subject to change. See the pricing pages for each AWS service you will be using or the AWS Simple Monthly Calculator for full details. This Quick Start includes nested AWS CloudFormation templates that launch multiple stacks. To get accurate pricing information, make sure to enter the data for each stack separately in the AWS Simple Monthly Calculator.

**Free tier: For details on eligibility, see the AWS Free Tier webpage.

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 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 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, subnet creation, and configuration of route tables and network gateways.

- AWS 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 (e.g., Amazon EC2 instances) that you want. You don't have to create and configure the resources or figure out dependencies; AWS CloudFormation handles all of that.

- Amazon RDS – Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale MySQL deployments in the cloud. With Amazon RDS, you can deploy scalable MySQL deployments in minutes with cost-efficient and resizable hardware capacity.

- [Auto Scaling](#) – Auto Scaling helps maintain high availability and manage capacity by automatically increasing or decreasing the Amazon EC2 instance fleet. You can use Auto Scaling to run your fleet at optimal utilization by increasing instance capacity during demand spikes and decreasing capacity during down times.

- [Elastic Load Balancing](#) – Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances.

- [Amazon S3](#) – Amazon Simple Storage Service (Amazon S3) provides secure, durable, highly scalable object storage. This Magento deployment uses Amazon S3 to share and retrieve common media assets by web server instances.

- [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 manage users, security credentials such as access keys, and permissions that control which AWS resources users can access, from a central location.

Architecture

AWS CloudFormation provides an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.

> **Note** This Quick Start deployment is automated by nested AWS CloudFormation templates. The main template builds the network-related resources first and then launches two separate templates for web server instances and Amazon RDS MySQL. Deleting the main template deletes the entire stack.
>
> Note that the [AWS Simple Monthly Calculator](#) doesn't support nested stacks. To get accurate pricing information, you must estimate the cost for each stack separately.

AWS Components

Running this Quick Start with default parameters for end-to-end deployment (which creates a new Amazon VPC) deploys and configures the following AWS components in the AWS cloud:

- An Amazon VPC configured with three subnets in multiple Availability Zones. Each subnet is configured with a private and public subnet. If you launch the deployment in a region that has fewer than three Availability Zones, subnet resources are created in only two Availability Zones. The public subnets are called `PublicSubnet0`, `PublicSubnet1`, and `PublicSubnet2`. The private subnets are called `PrivateSubnet0`, `PrivateSubnet1`, and `PrivateSubnet2`.

- A network address translation (NAT) instance deployed into the public subnets and configured with an Elastic IP address (EIP) for outbound Internet connectivity and inbound SSH (Secure Shell) access. The NAT instance is used for Internet access for all Amazon EC2 instances launched within the private network.

- A MySQL database engine deployed via Amazon RDS in `PrivateSubnet0`. If you choose Multi-AZ deployment, a synchronously replicated secondary database is deployed in `PrivateSubnet1`. This provides high availability and built-in automated failover from the primary database.

- Amazon EC2 web server instances launched in the private subnets. You must use a NAT instance to connect to these instances via SSH, because the web server instances are not in the public network.

- Elastic Load Balancing deployed to automatically distribute traffic across the multiple web server instances.

- Auto Scaling enabled to automatically increase capacity if there is a demand spike, and to reduce capacity during low traffic times. The default installation sets up low and high CPU-based thresholds for scaling the instance capacity up or down. You can modify these thresholds during launch and after deployment.

- An IAM instance role with fine-grained permissions for access to AWS services necessary for the deployment process.

- Appropriate security groups for each instance or function to restrict access to only necessary protocols and ports. For example, access to HTTP server ports on Amazon EC2 web servers is limited to Elastic Load Balancing. The security groups also restrict access to Amazon RDS MySQL DB instances by web server instances.

Figure 1 shows the Amazon VPC infrastructure built on the AWS cloud when you run the Quick Start with default parameters. (Availability Zone 3 isn't shown because of space constraints.)
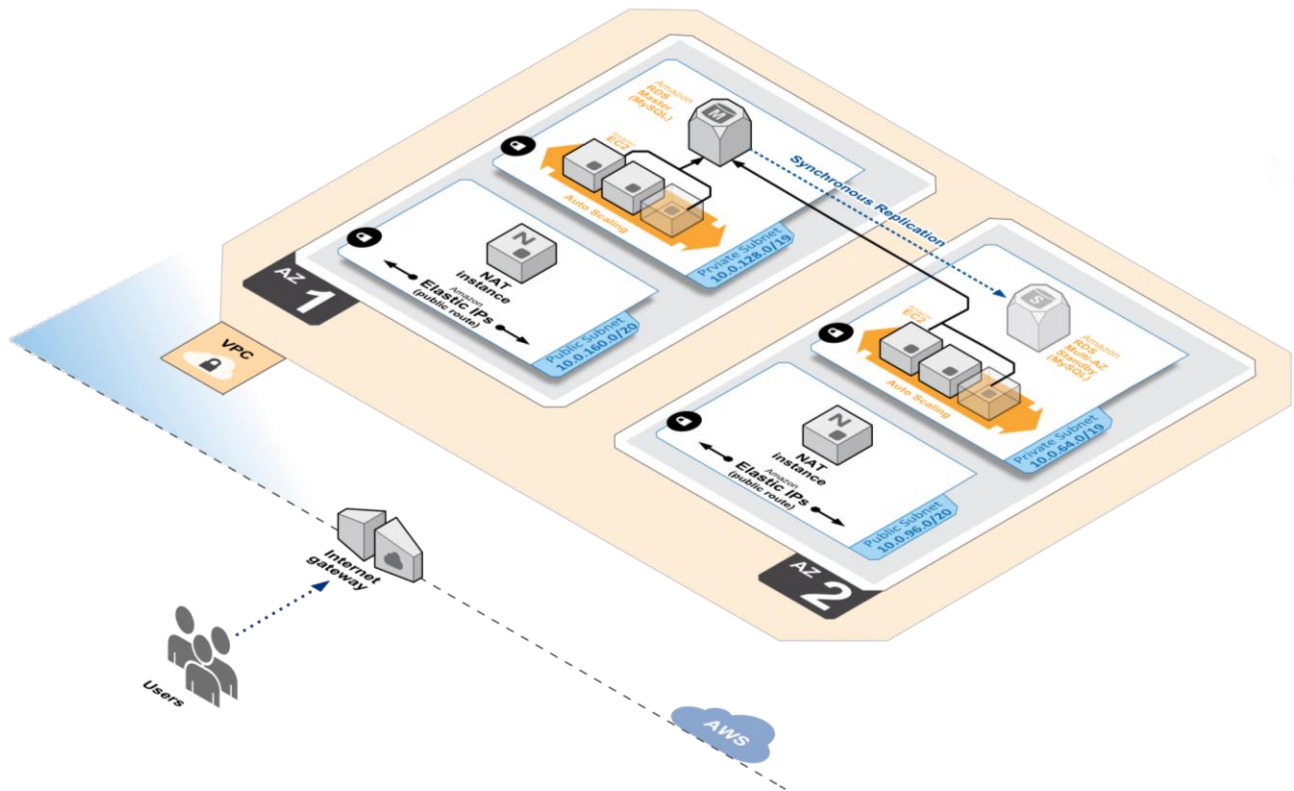
**Figure 1: Amazon VPC Architecture for Magento Quick Start**

Magento Components

This Quick Start deploys [Magento Community Edition](#) version 1.9.2 with the following prerequisite software:

- Operating system: Amazon Linux x86-64

- Web server: Apache HTTP Server 2.4

- Database: MySQL 5.6 through Amazon RDS

- Programming language: PHP 5.5, including the required extensions

This Quick Start also deploys the following additional software:

- phpmyadmin: Database administration utility installed on the web servers

- phpinfo.php file: Displays information about PHP and its extensions

- magento-check.php: Validates the requirements for Magento

- Magento sample data (optional): Lets you experiment with custom themes and view the web store

For more information about these utilities, see Installing and Verifying Magento Community Edition (CE) and Enterprise Edition (EE) on the Magento website.

Figure 2 shows the Magento architecture built on the AWS cloud when you run the Quick Start with default parameters. (Availability Zone 3 isn't shown because of space constraints.)
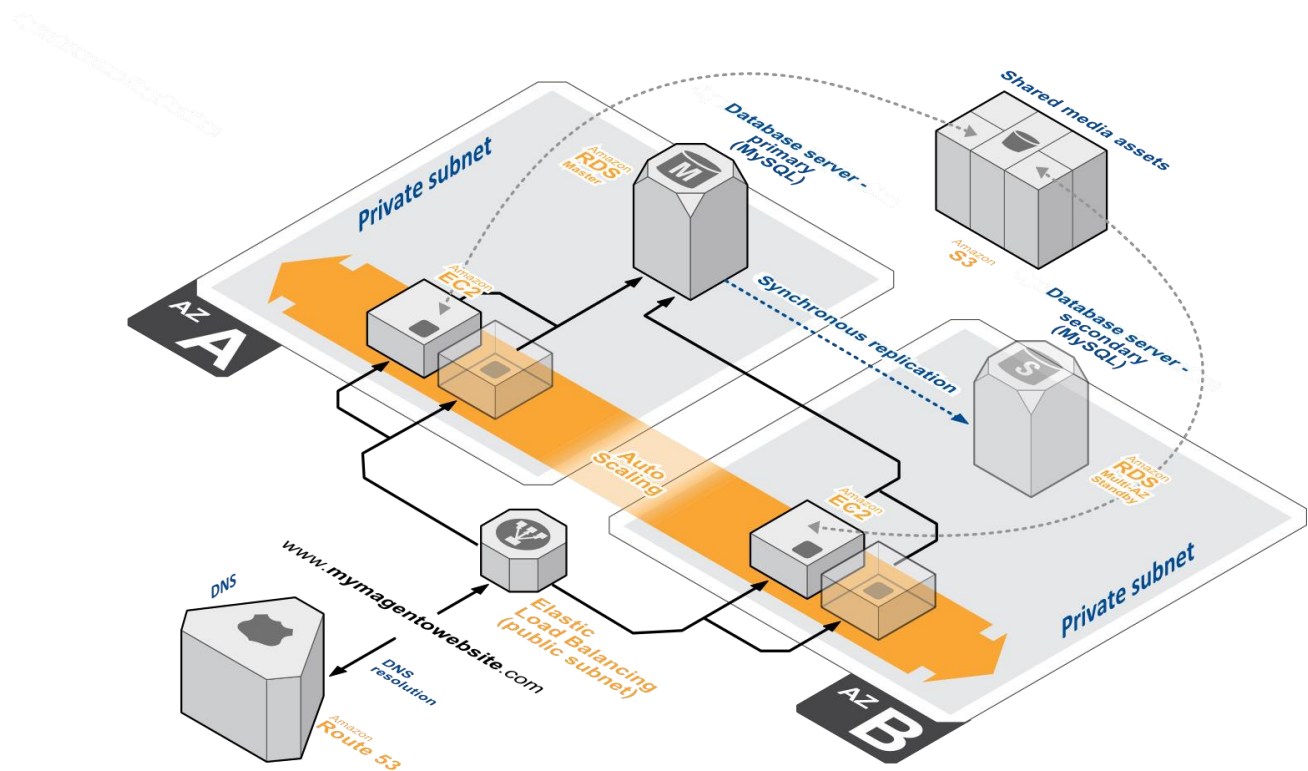
Figure 2: Quick Start Magento Architecture on AWS

Design Considerations

Magento is a robust e-commerce platform that can be deployed with a variety of options, depending on your needs. This Quick Start provides a great starting point for building your e-commerce solution rapidly with Magento on the AWS cloud. The following sections discuss design considerations for large-scale deployments and options for optimizing performance.

Amazon RDS for MySQL

Amazon RDS for MySQL deployments are preconfigured with a set of parameters and settings appropriate for the DB instance class you choose. Amazon RDS supports automatic

software patching, database backups, backup storage for a user-defined retention period, and point-in-time recovery.

Amazon RDS supports three types of storage: Magnetic, General Purpose (SSD), and Provisioned IOPS (SSD). General Purpose (SSD) storage delivers a consistent baseline of 3 IOPS per provisioned GiB and provides the ability to burst up to 3,000 IOPS. To achieve a higher level of performance, consider using Provisioned IOPS (SSD) to provision from 1,000 IOPS up to 30,000 IOPS per DB instance. (Maximum realized IOPS may be lower.) You can provision up to 3 TiB storage and 30,000 IOPS per database instance. We recommend using Magnetic storage for small database workloads where data is accessed less frequently. Note that you can convert from standard storage to Provisioned IOPS storage on a running cluster. (There is a short availability impact on the servers.)

To enhance availability and reliability for production workloads, you should use the Multi-AZ deployment option. This provides an automated failover from the primary database to a synchronously replicated secondary database that is running in a different Availability Zone. This option also enables you to scale out beyond the capacity of a single database for read-heavy database workloads.

Amazon EC2 Web Server Instances

The deployment installs Apache HTTP Server 2.4 on Amazon EC2 instances running Amazon Linux x86-64. Elastic Load Balancing is used to automatically distribute the website load across these instances. In addition, all the instances are launched in an Auto Scaling group that dynamically manages the Amazon EC2 fleet. The deployment sets low and high CPU utilization thresholds to automatically scale the number of Amazon EC2 instances up or down depending on load. Default policy adds new instances when the CPU load exceeds 90% for 10 minutes, and removes instances when the CPU load drops below 70% for 10 minutes.

You can specify the maximum number of instances in the Auto Scaling group, and Auto Scaling ensures that your group never goes above this size. You can also specify the desired capacity, either when you create the group or at any time thereafter, and Auto Scaling will ensure that your group has the desired number of instances. These options are configurable during Quick Start launch.

The Quick Start supports a variety of Amazon EC2 instance types. We recommend that you benchmark the cluster to make sure you achieve the level of performance you need before starting a production deployment. For high availability, we recommend using at least two web server instances in different Availability Zones.

Amazon S3

This Quick Start provides an option for installing sample Magento data during deployment. (For details, see the Installing Optional Sample Data section of the Magento installation guide.) If you choose this option, the common media assets derived from the samples are stored in Amazon S3. To provide unified access among web server instances, we install a FUSE file system (s3fs). s3fs allows you to mount an Amazon S3 bucket as a local file system. It stores files natively and transparently in Amazon S3.

> **Note**   s3fs is not officially supported by AWS. This Quick Start uses s3fs as a temporary solution for storage until Amazon Elastic File System (Amazon EFS) is released and becomes available in all regions. Amazon EFS is a file storage service for Amazon EC2 instances. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance. As of the date of publication, Amazon EFS is available as a preview release. For more information, see the Amazon EFS documentation.

Miscellaneous Considerations

Amazon ElastiCache is a web service that helps you deploy and operate an in-memory cache in the AWS cloud. This Quick Start doesn't automatically deploy ElastiCache, but we recommend that you use the service to set up Redis or Memcached as a caching layer (both are supported by Magento). ElastiCache reduces operational overhead involved in deploying these services and provides a way to improve application load times. For more information, see the Amazon ElastiCache documentation.

For handling session management, we recommend using Amazon DynamoDB. Amazon DynamoDB is a fast and flexible NoSQL database service that provides consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. For more information, see the Amazon DynamoDB documentation.

This deployment launches all the Amazon EC2 instances in the private subnet. Access to the external world is routed via a NAT instance that resides in the public subnet. To learn how you can improve the high availability of NAT instances, see the article High Availability for Amazon VPC NAT Instances: An Example on the AWS website.

Automated Deployment

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of a Magento cluster 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 a Magento cluster 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 Amazon EBS volumes, and request a limit increase, if needed.

Step 2. Download the Magento software

- Create a free Magento account, if you don't already have one.
- Download Magento Community Edition and (optionally) Magento sample data.
- Create an Amazon S3 bucket, and add the Magento software to this bucket.

Choose step 3(a) **or** step 3(b) depending on whether you want to deploy the Quick Start into a new or existing Amazon VPC.

Step 3(a). Launch the Quick Start into your AWS account (new Amazon VPC)

When you launch the Quick Start to deploy Magento into a new Amazon VPC, the AWS CloudFormation template automates the following steps:

- Sets up the Amazon VPC.
- Creates various network resources needed during Magento deployment, including private and public subnets in multiple Availability Zones within an Amazon VPC, a NAT instance, an Amazon RDS MySQL DB instance, security groups, and an IAM role.
- Fully automates security settings to provide minimal privileges to each instance, and sets firewall options to open up access between nodes and Elastic Load Balancing.

- Installs the prerequisite software for Magento on the web server instances. If you choose to install optional sample data, the Quick Start mounts an Amazon S3 bucket as a drive among web server nodes via s3fs and installs common media assets in the Amazon S3 bucket.

Step 3(b). Launch the Quick Start into your AWS account (existing Amazon VPC)

This Quick Start provides a separate template for deploying a Magento cluster into an existing Amazon VPC. If you have already constructed an Amazon VPC with public and private subnets in multiple Availability Zones, and you want to deploy Magento inside your Amazon VPC, use this option. The template automates all the steps in step 3(a) except for the creation of the Amazon VPC.

Step 4. Test your Magento deployment

- Connect to the web server instances via SSH.

- Validate the Magento installation and Admin Panel.

- Validate the Amazon RDS MySQL installation.

- Validate successful installation of the sample web store (if you've installed the optional sample data).

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 the Magento cluster on AWS.

   Amazon EC2 locations are composed of *regions* and *Availability Zones*. Regions are dispersed and located in separate geographic areas. This Quick Start uses the M4 instance type for the Magento cluster. M4 instance types are currently available in all AWS regions except GovCloud (US), China (Beijing), and South America (São Paulo).
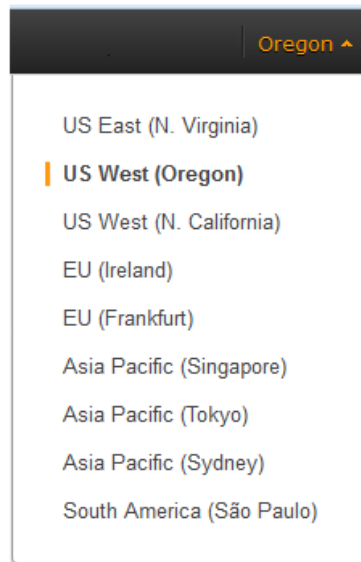
Figure 3: Choosing an Amazon EC2 Region

> **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**.
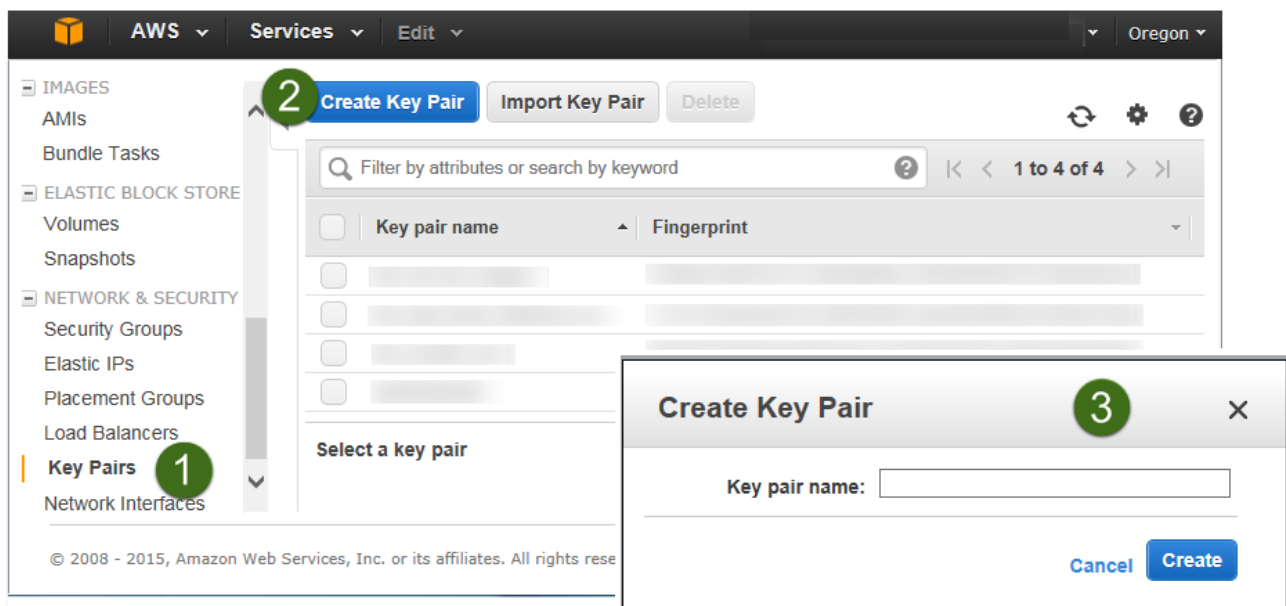
Figure 4: Creating a Key Pair

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. On Linux, we use the key pair to authenticate SSH login.

4. If necessary, request a service limit increase for the Amazon EC2 instance types that you intend to deploy. 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 default limit for the number of instances depends on the instance type you choose and currently ranges from 2 to 20 (see the Amazon EC2 FAQs page). If you have existing deployments that also use this instance type, or if you plan to exceed the default with this reference deployment, you will need to request a limit increase. 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 5: Requesting a Service Limit Increase

Step 2. Download the Magento Software

Before you launch the Quick Start, you must download the Magento Community Edition software (and optionally Magento sample data) into an Amazon S3 bucket. When you launch the Quick Start, you specify the full path to the downloaded files as input to the main AWS CloudFormation template, and the Quick Start automatically installs Magento for you.

1. If you don't already have a Magento account (which is free), create an account.

2. Download the Magento Community Edition version 1.9.2 **full release** from the Magento Community Edition download page. Choose the .tar.gz format for the download. At the time of publication, the latest full release was magento-1.9.2.1.tar.gz.

3. (Optional) If you want to set up the sample store, download Magento **sample data** from the same page. Choose .tar.gz format for the download (for example, magento-sample-data-1.9.1.0.tar-2015-02-11-08-18-26.gz).

4. Create an Amazon S3 bucket by following the instructions in the *Amazon S3 Getting Started Guide*.

5. Add the downloaded files to the Amazon S3 bucket by following the instructions in the *Amazon S3 Getting Started Guide*.

6. Note the full paths to the downloaded files. In step 3, when you set parameters for the AWS CloudFormation template, you'll specify these paths.

    For example, if your Amazon S3 bucket is named `mymagentobucket`, you'll set the *MagentoReleaseMedia* parameter to:

    ```
    s3://mymagentobucket/magento-1.9.2.1.tar-2015-08-03-06-35-45.gz
    ```

    If you've downloaded sample data to `mymagentobucket`, you'll set the *MagentoSampleData* parameter to:

    ```
    s3://mymagentobucket/magento-sample-data-1.9.1.0.tar-2015-02-11-08-18-
    26.gz
    ```

Step 3(a). Launch the Quick Start into Your AWS Account (New Amazon VPC)

In this step, you will launch an AWS CloudFormation template that sets up the Amazon VPC, creates network resources, sets security and firewall options, and installs the Magento software, as discussed earlier.

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

    Launch
    (for new VPC)

    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 15-30 minutes to create, depending on whether you choose to install sample data.

    > **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 Costs section for cost estimates for different instance types and storage configurations. 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 settings for the stack name and template source, and then choose **Next**.

3. On the **Specify Parameters** page, review the parameters for the template. These are described in the following tables. Provide values for parameters that require your input, and customize other values as necessary. When you're done, choose **Next**.

Network Settings

We recommend that you use the default values for the CIDR block parameters unless you're an experienced network administrator.

Parameter	Default	Description
VPCCIDR	10.0.0.0/16	CIDR block for the Amazon VPC you are creating.
PrivateSubnetCIDR0	10.0.128.0/19	CIDR block for the private subnet in Availability Zone 0.
PrivateSubnetCIDR1	10.0.64.0/19	CIDR block for the private subnet in Availability Zone 1.
PrivateSubnetCIDR2	10.0.0.0/19	CIDR block for the private subnet in Availability Zone 2.
PublicSubnetCIDR0	10.0.160.0/20	CIDR block for the public subnet in Availability Zone 0.
PublicSubnetCIDR1	10.0.96.0/20	CIDR block for the public subnet in Availability Zone 1.
PublicSubnetCIDR2	10.0.32.0/20	CIDR block for the public subnet in Availability Zone 2.
RemoteAccessCIDR	0.0.0.0/0	IP CIDR from which you are likely to connect to the NAT instance by using SSH.
KeyName	*Requires input*	An existing public/private key pair, which allows you to connect securely to your instance after it launches. This is the key pair you created in step 1, when you prepared your AWS account.
NATInstanceType	t2.medium	Amazon EC2 instance type for the NAT instance.

Web Server Settings

Parameter	Default	Description
WebServerInstanceType	m3.2xlarge	Amazon EC2 instance type for web server instances.
WebServerMinSize	1	Minimum number of Amazon EC2 instances in the Auto Scaling group of web servers.
WebServerMaxSize	1	Maximum number of Amazon EC2 instances in the Auto Scaling group of web servers.
WebServerDesiredCapacity	1	Desired number of Amazon EC2 instances in the Auto Scaling group before deployment is considered successful.
ExistingIamInstanceProfile	–	Optional instance profile name for the web server instances. Leave this parameter blank to create one automatically.

Amazon RDS MySQL Parameters

Parameter	Default	Description
DBAllocatedStorage	16	Size of the MySQL database (in GiB).
DBAutoMinorVersionUpgrade	true	Determines whether the DB instance will automatically be upgraded to new MySQL minor versions as they are supported by Amazon RDS.
DBBackupRetentionPeriod	7	Number of days for which automatic DB snapshots are retained.
DBInstanceClass	db.m3.2xlarge	DB instance class for Amazon RDS.
DBIops	1000	Provisioned IOPS for DB storage. Used only when the *DBStorageType* parameter is set to **io1** (that is, when Provisioned IOPS is chosen).
DBMasterUsername	Admin	User name for the database administrator account.
DBMasterUserPassword	*Requires input*	Password for the database administrator account.
DBMultiAZ	false	Set to **true** if Multi-AZ RDS deployment is needed.
DBName	QuickstartMySQLDB	Name of the initial database for the MySQL RDS instance.
DBPubliclyAccessible	false	Set to **true** if the database instance is Internet-facing (public subnet).
DBStorageType	gp2	Storage type associated with the database instance.

Magento Options

Parameter	Default	Description
BucketNamePrefix	magentoquickstart	The prefix for the Amazon S3 bucket that holds Magento store media assets after the Magento installation. (The suffix for the bucket name is **vpcid**.)
MagentoReleaseMedia	*Requires input*	The full path to the Magento tar.gz file downloaded in step 2, entered as s3://*mybucket*/magento-1.9.2.1.tar-2015-08-03-06-35-45.gz
MagentoSampleData	–	(Optional) The full path to the Magento sample data file, if downloaded in step 2, entered as s3://*mybucket*/magento-sample-data-1.9.1.0.tar-2015-02-11-08-18-26.gz

Miscellaneous Options
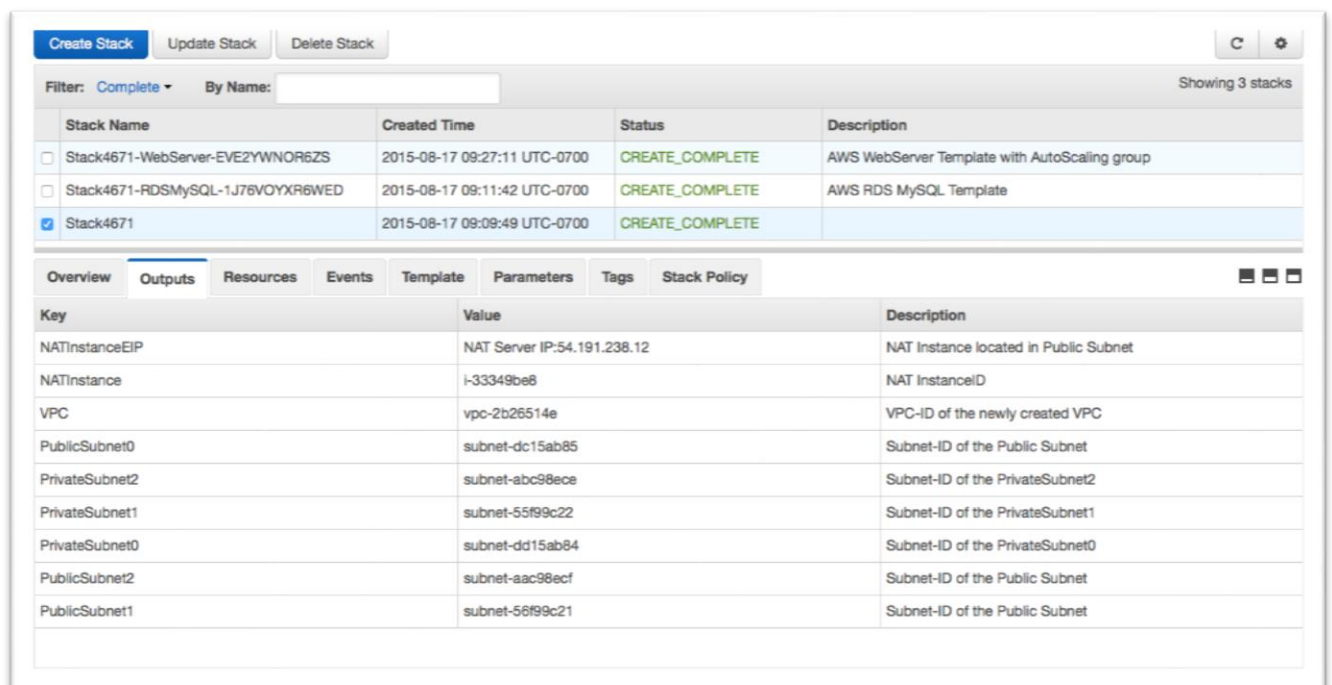
Parameter	Default	Description
NotificationEmail	–	(Optional) Email address to notify if there are any scaling actions on the Amazon EC2 web server fleet.

> **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 settings.

6. Choose **Create** to deploy the stack.

When the status field displays **CREATE_COMPLETE**, as shown in Figure 6, the Magento
cluster is ready.

> **Note**   This Quick Start deployment is automated by nested AWS CloudFormation
> templates. The main template builds the network-related resources first and then
> launches two separate templates for web server instances and Amazon RDS MySQL.
> Deleting the main template deletes the entire stack.

Figure 6: Successful Creation of the Magento Cluster

Step 3(b). Launch the Quick Start into Your AWS Account (Existing Amazon VPC)

If you have an existing Amazon VPC, you can still use the Quick Start to build a Magento cluster within it. The deployment steps are same as in step 3(a) except for the network parameters. This assumes you already have three private and three public subnets (named as shown in the following table) created within the Amazon VPC. Instead of CIDR ranges, you specify the IDs of the corresponding subnets, as shown in the following table. All other options remain the same.

**Launch
(for existing VPC)**

Network Options

Parameter	Default	Description
PublicSubnet0	subnet-*ID* (*requires input*)	ID of `PublicSubnet0` in Availability Zone 0
PublicSubnet1	subnet-*ID* (*requires input*)	ID of `PublicSubnet1` in Availability Zone 1
PublicSubnet2	subnet-*ID* (*requires input*)	ID of `PublicSubnet2` in Availability Zone 2
PrivateSubnet0	subnet-*ID* (*requires input*)	ID of `PrivateSubnet0` in Availability Zone 0
PrivateSubnet1	subnet-*ID* (*requires input*)	ID of `PrivateSubnet1` in Availability Zone 1
PrivateSubnet2	subnet-*ID* (*requires input*)	ID of `PrivateSubnet2` in Availability Zone 2

Step 4. Test Your Magento Deployment

When the AWS CloudFormation template has successfully created the stack, all web server nodes will be running with the software installed in your AWS account. To connect to the Magento stack, use the URL of the Elastic Load Balancing endpoint, as shown in Figure 7.
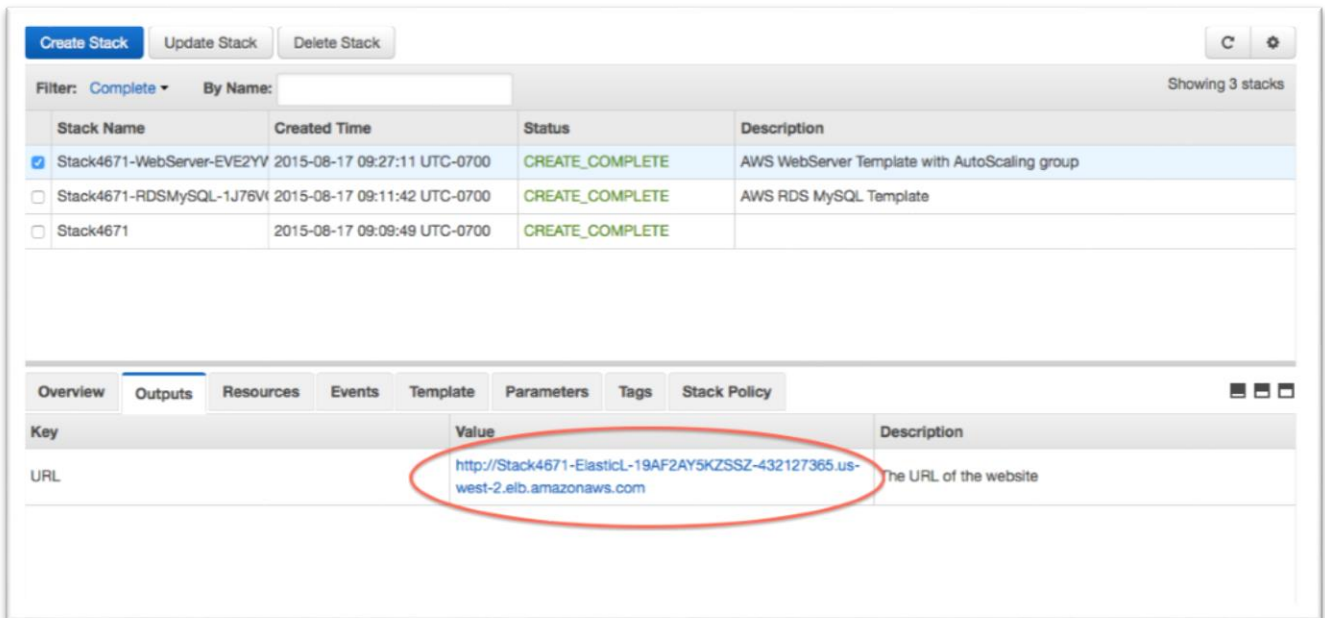
Figure 7: Elastic Load Balancing Endpoint URL for the Magento Stack

This URL brings up the page shown in Figure 8, which provides links for validating the installation as well as links to the store.

Figure 8: Elastic Load Balancing Endpoint for Magento Stack

To test the Magento installation, choose the validation link on this page. This link validates the installation of all the prerequisite software for Magento.
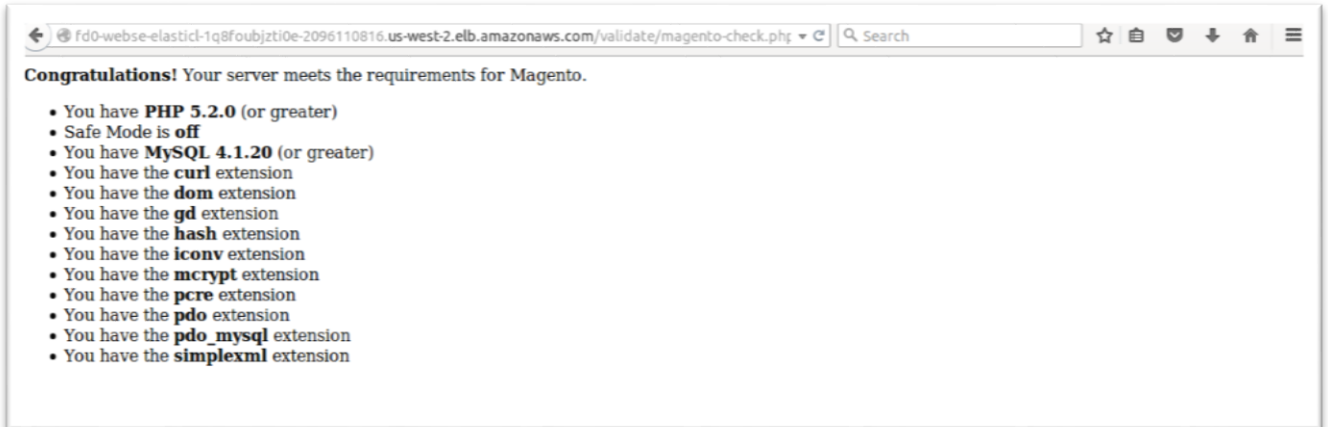
Figure 9: Magento Validation Page

To visit the Magento store, choose the store link on the Elastic Load Balancing endpoint for the Magento stack. Figure 10 shows the web store when you've installed the optional sample data.
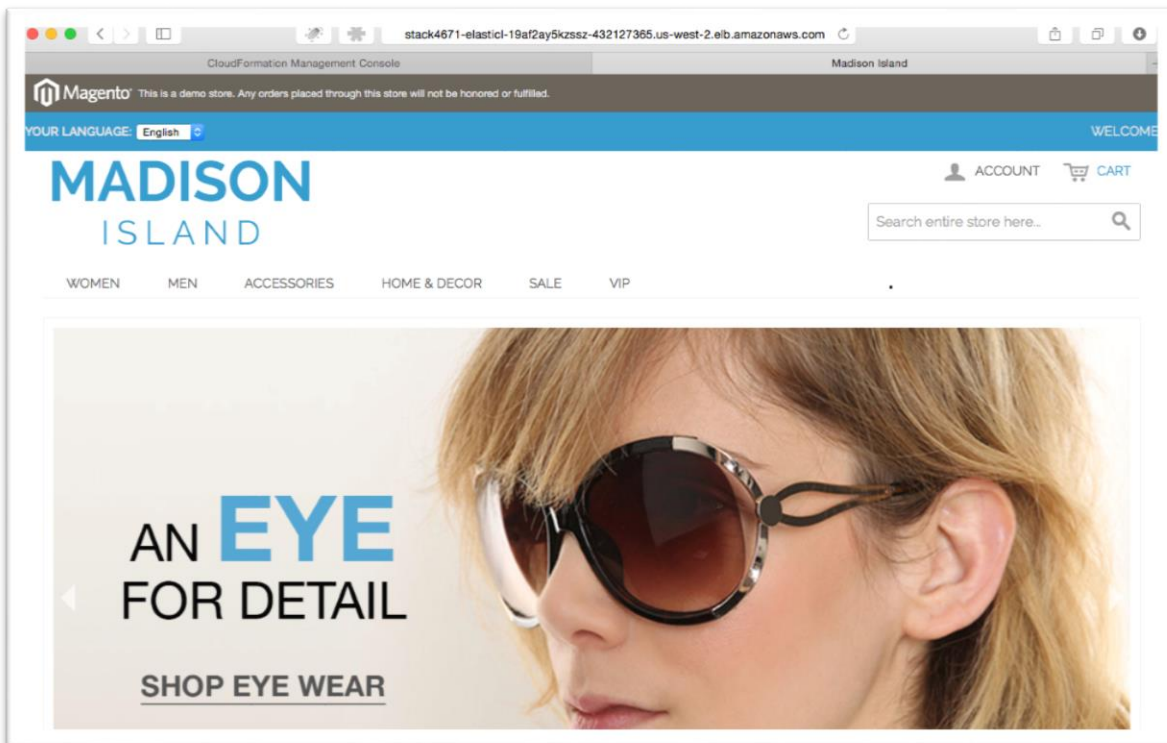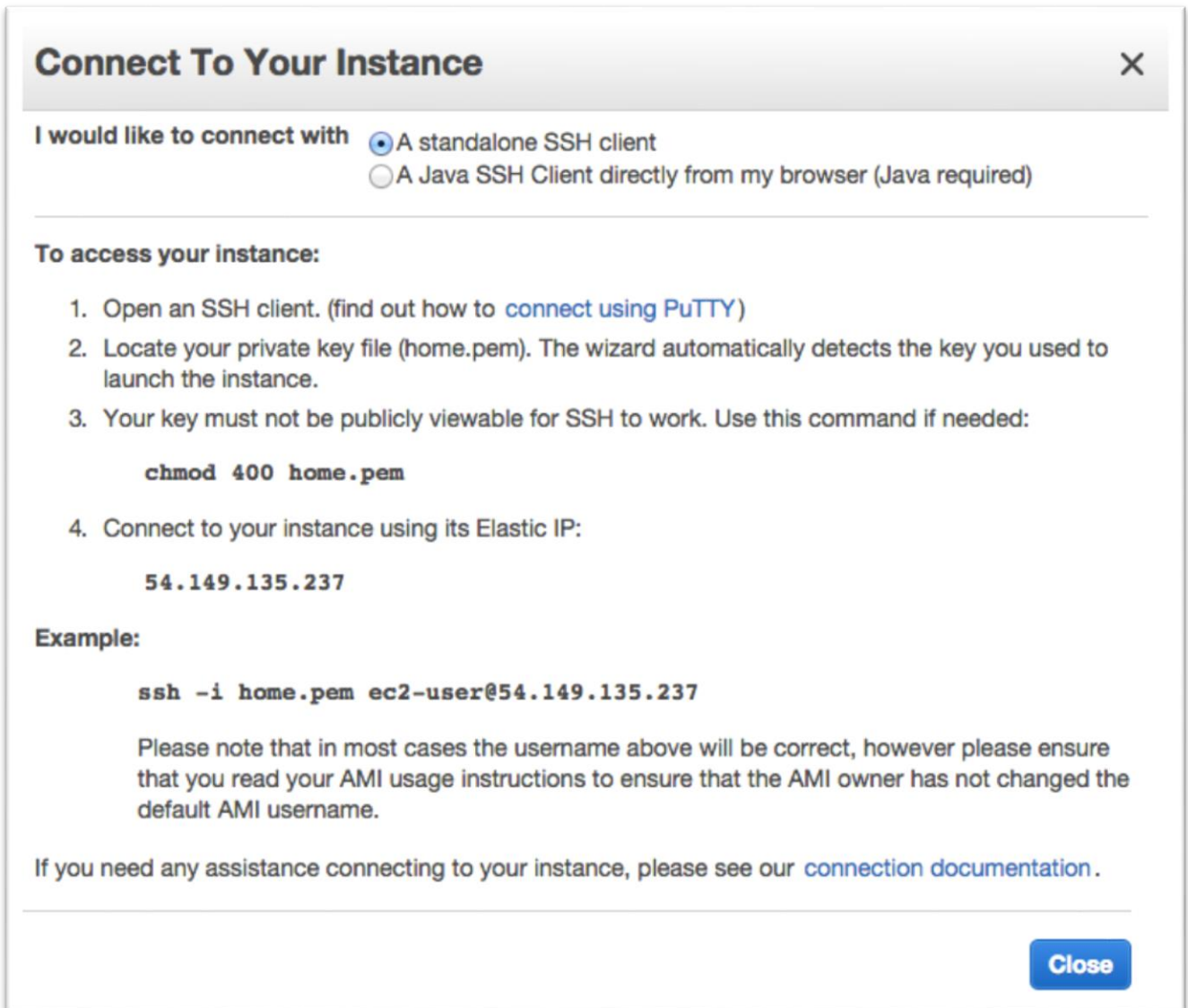
Figure 10: Magento Store When Optional Sample Data is Installed

You can access the Magento Admin Panel with the same login and password as the MySQL database. Make sure to change the user name and password for subsequent logins to help ensure the security of your account.

To connect to the web server instance and validate MySQL, use SSH to connect to the NAT instance. To do this, select the NAT instance in the Amazon EC2 console, and then choose **Connect**.

Connect To Your Instance   ✕

I would like to connect with   ⦿ A standalone SSH client
                                   ◯ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (home.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

   `chmod 400 home.pem`

4. Connect to your instance using its Elastic IP:

   `54.149.135.237`

Example:

`ssh -i home.pem ec2-user@54.149.135.237`

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

Close

Figure 11: Connecting to the NAT Instance

After you connect to the NAT instance by using SSH, you can connect to any of the web server nodes in a similar way (choose the node, and then choose **Connect** to find the SSH command).

> **Important**   You need the private key (.pem) file to connect to web server nodes. Copy the private key (.pem) file into the NAT instance; for example:
>
> ```
> scp –i mykey.pem mykey.pem ec2-user@NAT-public-ip:/home/ec2-user/mykey.pem
> ```

Test MySQL connectivity from the web server instance as shown in Figure 12. The Amazon S3 bucket is mounted as a drive among web server nodes via s3fs, and common media assets can be accessed in /mnt/magento.

Figure 12: Testing MySQL Connectivity

Security

The AWS cloud provides a scalable, highly reliable platform that helps customers deploy applications and data quickly and securely.

When you build systems on the AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. In turn, you assume responsibility and management of the guest operating system (including updates and security patches), other associated applications, as well as the configuration of the AWS-provided security group firewall. For more information about security on AWS, visit the AWS Security Center.

AWS Identity and Access Management (IAM)

This solution leverages an IAM role with least privileged access. It is not necessary or recommended to store SSH keys, secret keys, or access keys on the provisioned instances.

OS Security

The root user on cluster nodes can be accessed only by using the SSH key specified during the deployment process. AWS doesn't store these SSH keys, so if you lose your SSH key you can lose access to these instances.

Operating system patches are your responsibility and should be performed on a periodic basis.

Security Groups

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 Magento. We recommend reviewing security groups to further restrict access as needed once the cluster is up and running.

Additional Resources

AWS services

- Getting Started
  http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/intro.html

- AWS CloudFormation
  http://aws.amazon.com/documentation/cloudformation/

- Amazon EC2

  – User guide for Linux instances:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

  – Regions and Availability Zones:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

  – Key pairs:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html

  – Instance stores:
    http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#instance-storage-concepts

  – FAQ:
    http://aws.amazon.com/ec2/faqs

- Amazon Identity and Access Management

  – User guide:
    http://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

  – Benefits of the IAM role:
    http://docs.aws.amazon.com/IAM/latest/UserGuide/role-usecase-ec2app.html

- Auto Scaling
  http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/WhatIsAutoScaling.html

- Elastic Load Balancing
  http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/elastic-load-balancing.html

- Amazon S3

  – Getting started guide
    http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html

  – Developer guide
    http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

- Amazon RDS

  – User guide
    http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html

  – MySQL on Amazon RDS
    http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MySQL.html

- Amazon VPC

  – Documentation:
    http://aws.amazon.com/documentation/vpc/

  – High availability for NAT instances
    https://aws.amazon.com/articles/2781451301784570

- AWS Security Center
  http://aws.amazon.com/security/

Magento

- Magento Community Edition software
  https://www.magentocommerce.com/download

- Installing and Verifying Magento Community Edition
  http://devdocs.magento.com/guides/m1x/install/installing_install.html

- Magento Community Edition User Guide
  http://merch.docs.magento.com/ce/user_guide/Magento_Community_Edition_User_Guide.html

s3fs

- https://github.com/s3fs-fuse/s3fs-fuse/wiki/Fuse-Over-Amazon

Additional Quick Start Reference Deployments

- https://aws.amazon.com/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
September 2015	Initial publication	–