

Drupal on the AWS Cloud

Quick Start Reference Deployment

*Madhuri Susarla, Vikrama Adethyaa, and Vinod Shukla
Partner Solutions Architects, Amazon Web Services*

February 2018

Last updated: August 2018 ([revisions](#))

Contents

About This Guide	3
Overview	3
Drupal on AWS	4
Cost and Licenses	4
AWS Services	5
Regions, Availability Zones, and Endpoints	6
Architecture	7
Drupal Components	8
AWS Components	9
Deployment Options	11
Deployment Steps	11
Step 1. Prepare an AWS Account	11
Step 2. Launch the Quick Start	13
Step 3. Test Your Drupal Deployment	23
Troubleshooting	25
Security	26
AWS Identity and Access Management	26
Operating System Security	26
Security Groups	26
GitHub Repository	27
Additional Resources	27
Document Revisions	28

About This Guide

This Quick Start deployment guide discusses architectural considerations and configuration steps for deploying a highly available Drupal architecture on the Amazon Web Services (AWS) Cloud. It also provides links for viewing and launching [AWS CloudFormation](#) templates that automate the deployment of AWS compute, network, storage, and other services required to deploy Drupal on AWS, using AWS best practices for security and availability.

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

Overview

Enterprises need to grow and manage their global computing infrastructures rapidly and efficiently while also optimizing and managing capital costs and expenses. The computing and storage services from AWS meet this need by providing a global computing infrastructure and services that simplify managing infrastructure, storage, and databases. With the AWS infrastructure, companies can rapidly provision compute capacity, or quickly and flexibly extend existing on-premises infrastructure into the cloud.

[Drupal](#) is an open-source content management framework written in the PHP server-side scripting language. Drupal provides a backend framework for many enterprise websites, with standard features such as easy content authoring, reliable performance, and robust security. Flexibility and modularity are the core tenets of Drupal's design principles.

You can use this Quick Start to:

- Bootstrap a new AWS infrastructure to create a virtual private cloud (VPC), subnets, bastion hosts, NAT gateways, and other infrastructure components, and then deploy Drupal.
- Deploy Drupal into your existing VPC.
- Modify the CloudFormation templates provided in this Quick Start to better accommodate Drupal into your existing AWS infrastructure.

Drupal on AWS

Deploying Drupal on AWS makes it easy to use AWS services to further enhance the performance and extend functionality of your content management framework.

AWS provides flexible compute, storage, and database services, making it an ideal platform to run Drupal workloads. AWS offers a complete set of services and tools for deploying business-critical enterprise workloads on its highly reliable and secure cloud infrastructure. Coupled with AWS as the underlying infrastructure, Drupal offers a very agile, scalable, and high-performance platform for pay-as-you-go web content management.

This guide provides IT infrastructure decision-makers and system administrators with technical guidance on how to configure, deploy, and run Drupal in a highly available manner on AWS. It outlines a reference architecture for Drupal [version 7](#) and [version 8](#). This architecture addresses common scalability, high availability, and security requirements.

This guide discusses best practices for deploying Drupal on AWS using Amazon Elastic Compute Cloud (Amazon EC2), Amazon Virtual Private Cloud (Amazon VPC), Amazon Relational Database Service (Amazon RDS), and Amazon Elastic File System (Amazon EFS). It also provides links to automated AWS CloudFormation templates that you can launch directly into your AWS account. You can modify the accompanying AWS CloudFormation templates to suit your specific business requirements, or use them as is. For in-depth information about installing and using Drupal, see the [Drupal documentation](#).

Cost and Licenses

This deployment launches Drupal 7 or Drupal 8 automatically into a configuration of your choice. Drupal is open-source software. It is licensed under GNU GPL version 2. For additional details about Drupal's licensing, see the [Drupal website](#).

You are responsible for the cost of AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start.

This Quick Start includes nested AWS CloudFormation templates that launch multiple stacks. The templates include configuration parameters that you can customize. Some of these settings, such as instance type, will affect the cost of deployment. For cost estimates, see the pricing pages for each AWS service you will be using. Prices are subject to change.

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 Resource Center](#).)

- [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 EFS](#) – Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. With Amazon EFS, you can create a file system, mount the file system on your EC2 instances, and then read and write data from your EC2 instances to and from your file system.
- [Amazon RDS](#) – Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database such as Amazon Aurora or Amazon RDS MySQL in the cloud. With Amazon RDS, you can deploy scalable Amazon Aurora or Amazon RDS MySQL software in minutes with cost-efficient and resizable hardware capacity.
- [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.
- [AWS Auto Scaling](#) – AWS Auto Scaling helps you 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.
- [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.
- [Elastic Load Balancing](#) – Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple Amazon EC2 instances. It detects unhealthy instances and reroutes traffic to healthy instances until the unhealthy instances have been restored. ELB automatically scales its request handling capacity in response to incoming traffic.

- [AWS 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.
- [Amazon ElastiCache](#) - Amazon ElastiCache offers fully managed [Redis](#) and [Memcached](#) in-memory data stores. You can seamlessly deploy, operate, and scale popular open source compatible in-memory data stores.
- [Amazon CloudFront](#) - Amazon CloudFront is a global content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to your viewers with low latency and high transfer speeds.
- [Amazon Route 53](#) - Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to internet applications by translating names like `www.example.com` into numeric IP addresses like `192.0.2.1` that computers use to connect to one other. Amazon Route 53 is fully compliant with IPv6.

Regions, Availability Zones, and Endpoints

Before you begin working with the AWS CloudFormation template, familiarize yourself with AWS Regions, Availability Zones, and endpoints, which are components of the AWS secure global infrastructure.

Use AWS Regions to manage network latency and regulatory compliance. When you store data in a specific region, it is not replicated outside that region. It is your responsibility to replicate data across regions, if your business requires that. AWS provides information about the country, and, where applicable, the state where each region resides; you are responsible for selecting the region to store data, taking into consideration your compliance and network latency requirements. Regions are designed with availability in mind and consist of at least two Availability Zones. Most AWS services support regional endpoints, which reduce data latency by providing an entry point for service requests in that region.

Availability Zones are designed for fault isolation. They are connected to multiple internet service providers (ISPs) and different power grids. They are interconnected using high-speed links, so applications can rely on local area network (LAN) connectivity for communication between Availability Zones within the same region. You are responsible for carefully selecting the Availability Zones where your systems will reside. Systems can span multiple Availability Zones, and we recommend that you design your systems to survive temporary or prolonged failure of an Availability Zone in the case of a disaster.

For more information about AWS Regions, Availability Zones, and endpoints, see the [AWS documentation](#).

AWS provides web access to services through the AWS Management Console, available at <https://aws.amazon.com/console>, and then through individual consoles for each service. AWS provides programmatic access to services through application programming interfaces (APIs) and command line interfaces (CLIs).

Architecture

This Quick Start deploys a production-grade Drupal configuration that you can easily adapt to your use case by customizing parameters.

AWS CloudFormation provides an automated, easy way to create and manage a collection of related AWS resources.

The main template builds the network-related resources first and then launches separate templates for Drupal and Amazon Aurora. The other templates are primarily meant to modularize the CloudFormation code. An additional template uses AWS Lambda to create an Amazon Machine Image (AMI) for Drupal. This AMI installs Drupal on all the instances in the Auto Scaling group and eliminates the need for repeated downloads.

Optional templates include deploying an ElastiCache cluster, creating CloudFront web distribution, and creating DNS records in Route 53 public hosted zone. If you use ElastiCache or CloudFront, Drupal is configured with necessary default settings.

After your Drupal stack is deployed, you can optimize Drupal's caching and content delivery network settings to meet your requirements. Deleting the main template deletes the entire stack.

Figure 1 provides a view of the resulting architecture.

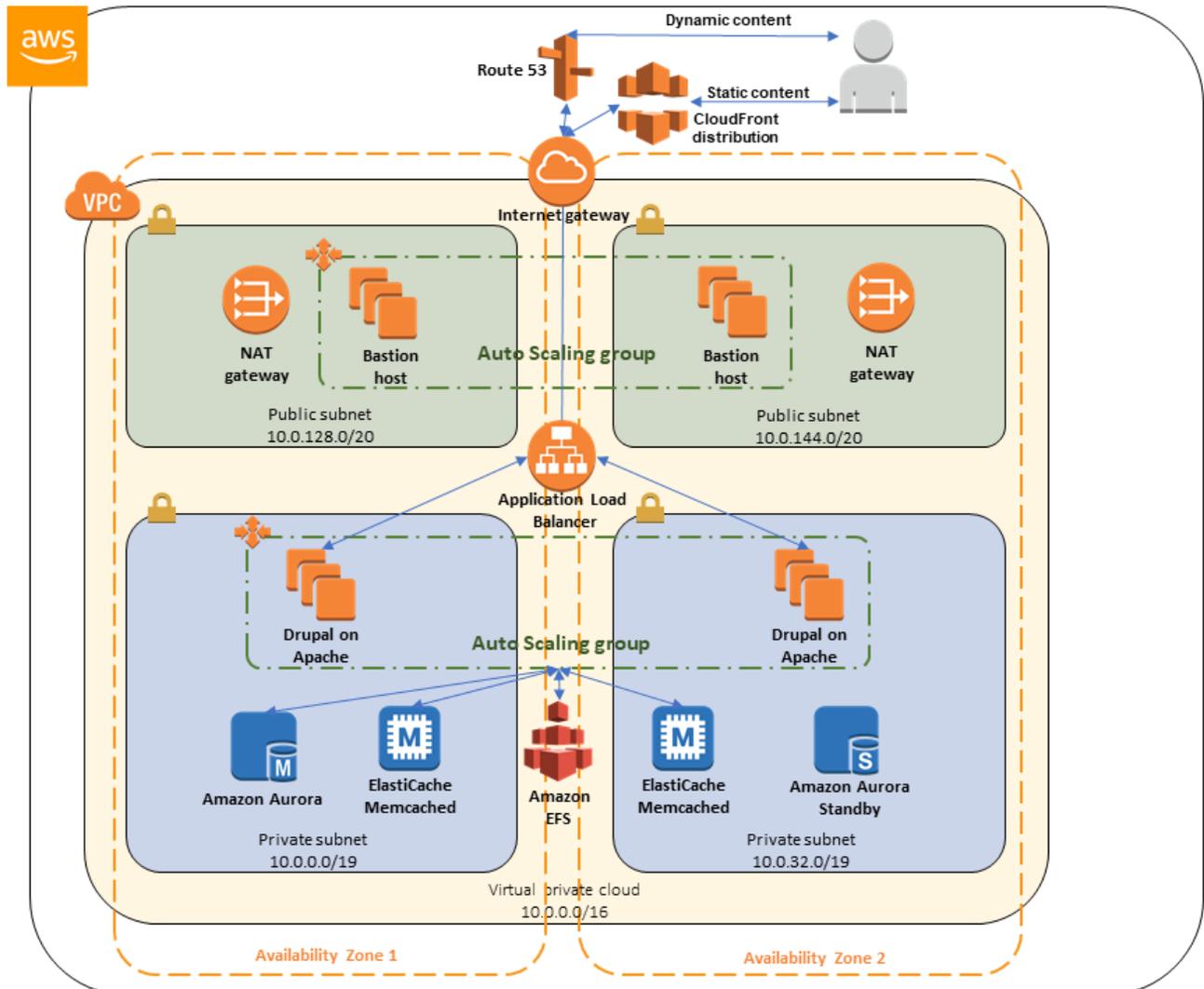


Figure 1: Quick Start architecture for Drupal on AWS

Drupal Components

The reference architecture deployed by the Quick Start maps AWS services to all the components required by Drupal. The highly available reference architecture for the Drupal deployment requires the following components:

- An HTTP(S) load balancer
- Two or more Drupal servers on an Apache web server
- Shared file storage
- A shared MySQL database

- ElastiCache for Memcache cluster
- CloudFront distribution
- Route 53

AWS Components

The Quick Start uses AWS CloudFormation templates to install and configure Drupal. The following list describes how the AWS services and components are used:

- Amazon VPC, which creates a logically isolated networking environment that you can connect to your on-premises data centers or use as a standalone environment.*

The VPC is configured across two Availability Zones. For each Availability Zone, this Quick Start provisions one public subnet and one private subnet. To ensure high availability, this architecture deploys the Drupal servers across two Availability Zones within a region. The Multi-AZ feature is enabled for the Amazon Aurora database, which creates a primary database instance and a replica database instance in different Availability Zones for high availability.

The Drupal instances and Amazon RDS database instances are in private subnets, exposing only the Elastic Load Balancing (ELB) listener, NAT gateways, and bastion host instances to the internet.

- NAT gateways deployed into the public subnets and configured with an Elastic IP address for outbound internet connectivity. These instances are used for internet access *for all Amazon EC2 instances launched within the private network. *
- A Linux bastion host in the public subnet to allow inbound Secure Shell (SSH) access to EC2 instances in the private subnets. *
- Elastic Load Balancing (ELB), which provides HTTP and HTTPS load balancing across the Drupal instances. In this case, ELB is configured to use HTTP.

Note When you use ELB, you must upload the web server's certificate and private key to the IAM service or generate a certificate with AWS Certificate Manager before you can enable the HTTPS listener.

- Amazon EC2 web server instances launched in the private subnets. You must use a bastion host to connect to these instances via SSH, because the web server instances are not in the public network.
- AWS Auto Scaling enabled, which allows the Drupal cluster to add or remove servers based on their use, providing additional servers during peak hours and lowering costs by removing servers during off hours. This functionality is tightly integrated with ELB and

automatically adds and removes instances from the load balancer. 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 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 ELB. The security groups also restrict access to Amazon Aurora DB instances by web server instances.
- Amazon Aurora for the shared database. Amazon RDS is a managed database service, so AWS handles all the administrative tasks for managing the database. By default, the database is deployed in multiple Availability Zones for high availability and automatically backed up on a schedule that you define, but you can change this configuration to Single-AZ during deployment.
- Amazon Elastic File System (Amazon EFS) as the shared file system. Drupal instances share the file system, so changes to content made by one Drupal instance are visible to the other Drupal instances.
- [Amazon ElastiCache for Memcached](#) for caching database queries and Drupal sessions.
- Amazon CloudFront as the content delivery network to speed up distribution of your Drupal's static and dynamic content to end users.
- Amazon Route 53 as your public Domain Name System (DNS) for resolving your Drupal site's domain name.

Note * The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks and prompts you for your existing VPC configuration.

Backup procedures are not covered in this deployment guide. For more information, see the [Drupal documentation](#). For additional details, see the [Additional Resources](#) section.

Deployment Options

This Quick Start provides two deployment options:

- **Deploy Drupal into a new VPC** (end-to-end deployment). This option builds a new AWS environment consisting of the VPC, subnets, NAT gateways, security groups, bastion host, and other infrastructure components, and then deploys Drupal into this new VPC.
- **Deploy Drupal into an existing VPC**. This option provisions Drupal in your existing AWS infrastructure.

Deployment Steps

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of a highly available Drupal configuration 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.

You can customize the template and the accompanying scripts as needed to best meet your business, IT, and security requirements.

The steps described in this guide implement a complete Drupal environment on the AWS Cloud.

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. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

When you create an AWS account, AWS automatically signs up the account for all AWS services, including Amazon EC2, which you'll use in the next step. You are charged only for the services that you use.

2. If necessary, request a service quota increase for the instance types used for the deployment. You might need to request an increase if you need additional Elastic IP addresses or if you already have an existing deployment that uses the same instance types as this architecture. To do this, on the [Service Quotas](#) console, for each instance type that you want a service quota increase, choose the instance type, choose **Request quota increase**, and then complete the fields in the quota increase form. It can take a few days for the new service quota to become effective.

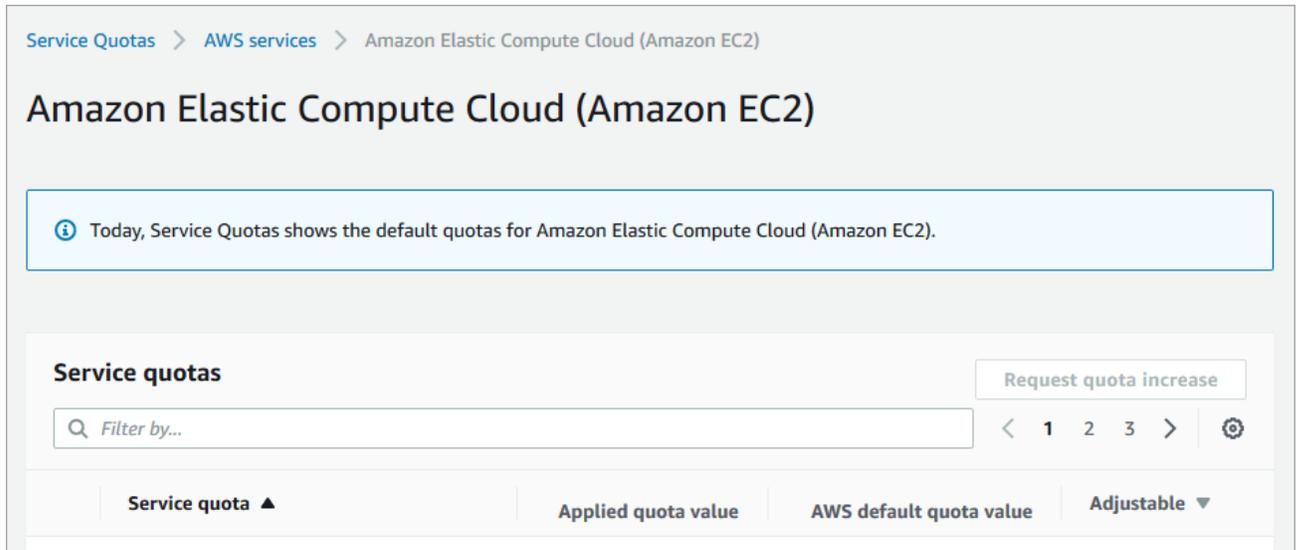


Figure 2: Requesting a service quota increase

3. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy Drupal on AWS.

Amazon EC2 locations are composed of *AWS Regions* and *Availability Zones*. Regions are dispersed and located in separate geographic areas. This Quick Start uses the **t2.micro** instance type for the Drupal portion of the deployment by default, but you can choose other instance types.

Note This Quick Start doesn't support all AWS Regions. Before you launch this Quick Start, please check [AWS Regions and Endpoints](#) for availability.

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.

4. 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**, as shown in Figure 3. .

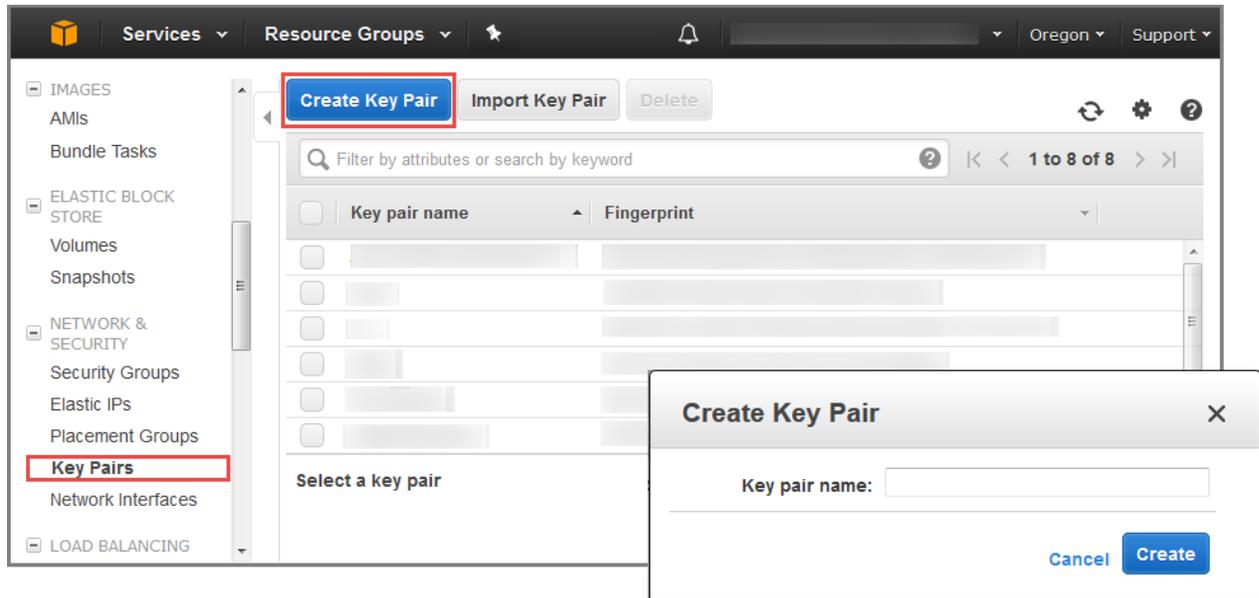


Figure 3: Creating a key pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. On Linux, we use the key pair to authenticate SSH login. Skip this step if you already have a key pair you can use.

Step 2. Launch the Quick Start

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. For additional details, see the [Cost and Licenses](#) section.

1. Choose one of the following options to launch the AWS CloudFormation template into your AWS account. For help choosing an option, see [deployment options](#) earlier in this guide.

<p>Option 1 Deploy Drupal into a new VPC</p> <p style="background-color: #0070C0; color: white; padding: 10px; border-radius: 5px; display: inline-block;">Launch</p>	<p>Option 2 Deploy Drupal into an existing VPC</p> <p style="background-color: #0070C0; color: white; padding: 10px; border-radius: 5px; display: inline-block;">Launch</p>
--	--

Important If you're deploying Drupal into an existing VPC, make sure that your VPC has two private subnets in different Availability Zones for Drupal and database instances. These subnets require NAT gateways in their route tables, to allow the instances to download packages and software without exposing them to the internet. You'll also need the domain name option configured in the DHCP options as explained in the [Amazon VPC documentation](#). You'll be prompted for your VPC settings when you launch the Quick Start.

Each template is launched in the US East (N. Virginia) Region by default. You can change the region by using the region selector in the navigation bar.

Each stack takes approximately 30 minutes to create.

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. Provide values for the parameters that require your input. For all other parameters, the template provides default settings that you can customize.

In the following tables, parameters are listed by category and described separately for the two deployment options:

- [Parameters for deploying Drupal into a new VPC](#)
- [Parameters for deploying Drupal into an existing VPC](#)

- **Option 1: Parameters for deploying Drupal into a new VPC**

[View template](#)

Network Configuration:

Parameter label (name)	Default	Description
Availability Zones (AvailabilityZones)	<i>Requires input</i>	The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.
VPC CIDR (VPCCIDR)	10.0.0.0/16	The CIDR block for the VPC.
Private Subnet 1 CIDR (PrivateSubnet1CIDR)	10.0.0.0/19	The CIDR block for the private subnet located in Availability Zone 1.

Parameter label (name)	Default	Description
Private Subnet 2 CIDR (PrivateSubnet2CIDR)	10.0.32.0/19	The CIDR block for the private subnet located in Availability Zone 2.
Public Subnet 1 CIDR (PublicSubnet1CIDR)	10.0.128.0/20	The CIDR block for the public (DMZ) subnet located in Availability Zone 1.
Public Subnet 2 CIDR (PublicSubnet2CIDR)	10.0.144.0/20	The CIDR block for the public (DMZ) subnet located in Availability Zone 2.

Bastion Host Configuration:

Parameter label (name)	Default	Description
Bastion Instance Type (BastionInstanceType)	t2.micro	Amazon EC2 instance type for the bastion instances.
Bastion AMI Operating System (BastionAMIOS)	Amazon-Linux-HVM	The Linux distribution for the AMI to be used for the bastion instances. If you choose CentOS, make sure that you have a subscription to the CentOS AMI in AWS Marketplace .
Bastion Host KeyPair Name (KeyPair)	<i>Requires input</i>	The name of an existing public/private key pair, which allows you to securely connect to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
Allowed Bastion External Access CIDR (RemoteAccessCIDR)	<i>Requires input</i>	The CIDR IP range that is allowed external SSH access to the bastion host instances. We recommend that you set this value to a trusted IP range. For example, you might want to grant only your corporate network access to the software. The range you specify must be in the form <i>x.x.x.x/x</i> (for example, 10.2.0.0/24).

Drupal Site Domain Configuration:

Parameter label (name)	Default	Description
Route53 Hosted Zone ID (Route53HostedZoneId)	—	Required if Drupal Site Domain or CloudFront Alias are not default values.
Drupal Site Domain (DrupalSiteDomain)	localhost.local	Domain name of the site (e.g. example.com). Valid fully qualified domain name (FQDN) required when using SSL. Leave the default localhost.local for test environments.
CloudFront Alias (CloudFrontAlias)	cdn.default	Alias for the CloudFront distribution (e.g. cdn.example.com). Mandatory when using HTTPS/SSL and optional when using http.

Aurora Database Configuration:

Parameter label (name)	Default	Description
Enable Auto Minor Version Upgrade (DBAutoMinorVersionUpgrade)	true	Choose true to allow AWS to automatically update the DB versions on Amazon Aurora, if they are minor upgrades. If a new engine minor version contains significant bug fixes compared to a previously released minor version, AWS will schedule automatic upgrades for DB instances that have Enable Auto Minor Version Upgrade set to true .
Backup Retention Period (DBBackupRetentionPeriod)	7	The number of days for which automatic DB snapshots are retained. For more information about setting a retention period, see Working with Automated Backups .
Database Instance Size (DBInstanceClass)	db.t2.small	The instance size for the Amazon Aurora database.
Database Admin Username (DBMasterUsername)	dbadmin	The user name of the database administrator account. This is a 1-16 character string that begins with a letter and includes only alphanumeric characters.
Database Admin Password (DBMasterUserPassword)	<i>Requires input</i>	The password of the database administrator account. This should be between 8-41 characters and cannot contain a white space, @, /, \ or “.”
Multi-AZ Database (DBMultiAZ)	true	Specifies if the database instance is a multiple Availability Zone deployment. Choose false if you want to deploy the database instance in a single Availability Zone. By default, the database instance is deployed across two Availability Zones for high availability.

Drupal Configuration:

Parameter label (name)	Default	Description
PHP Version (PhpVersion)	php71	The PHP version to be installed for Drupal. Versions 5.6, 7.0, and 7.1 are supported. For more information, see PHP Supported Versions .
Drupal Version (DrupalVersion)	7	The Drupal version to install. This Quick Start supports Drupal version 7 or version 8. For more information, see the Drupal documentation .
Drupal Site Name (DrupalSiteName)	My Site	The descriptive name for your Drupal sample website. This string must consist of 5-64 alphanumeric characters.
Drupal Site Admin Email (DrupalSiteAdminEmail)	<i>Requires input</i>	The administrator email for the Drupal site.

Parameter label (name)	Default	Description
Drupal Site Admin Username (DrupalSiteAdminUsername)	admin	The administrator user name for the Drupal site. The name must begin with a letter, contain only alphanumeric characters and must be between 5 and 16 characters long.
Drupal Site Admin Password (DrupalSiteAdminPassword)	<i>Requires input</i>	The administrator user password for the Drupal site. The password must contain only alphanumeric characters and must be between 8 and 41 characters long.
Drupal Database Name (DrupalDbName)	drupaldb	The name of the database to create and use for Drupal.
Drupal Database Username (DrupalDbUsername)	drupal	The user account of the Drupal database. The name must begin with a letter and contain only alphanumeric characters.
Drupal Database Password (DrupalDbPassword)	<i>Requires input</i>	The user account password of the Drupal database. The password must contain only alphanumeric characters and must be between 8 and 41 characters long.

Drupal Webserver Configuration:

Parameter label (name)	Default	Description
Webserver Instance Type (WebServerInstanceType)	t2.micro	The instance type of the Drupal web server.
Min Number of Instances (WebServerMinSize)	1	The minimum number of web server instances in the Auto Scaling group.
Max Number of Instances (WebServerMaxSize)	12	The maximum number of web server instances in the Auto Scaling group.
Desired Number of Instances (WebServerDesiredCapacity)	2	The desired number of web server instances in the Auto Scaling group.
Autoscaling Notification Email (AutoScalingNotificationEmail)	<i>Requires input</i>	The email address that notifications will be sent to when the environment is scaled up or down.
SSL certificate ARN (SSLCertificateId)	—	(Optional) The Amazon Resource Name (ARN) of the SSL certificate to use for the load balancer. You can create this by using AWS Certificate Manager (ACM).

ElastiCache Configuration:

Parameter label (name)	Default	Description
Enable ElastiCache (ElastiCacheEnable)	true	Enable ElastiCache for Drupal.
Enable ElastiCache Auto Minor Version Upgrade (ElastiCacheAutoMinorVersionUpgrade)	true	Select true/false to set up Auto Minor Version upgrade.
ElastiCache Node Type (ElastiCacheNodeType)	cache.t2.micro	The compute and memory capacity of nodes in a cache cluster.
Number of ElastiCache Nodes (ElastiCacheNumberOfNodes)	2	The number of cache nodes that the cache cluster should have.

CDN - CloudFront Configuration:

Parameter label (name)	Default	Description
Enable CloudFront (CloudFrontEnable)	true	Enable CloudFront Content Delivery Network.
CloudFront PriceClass (CloudFrontPriceClass)	use-all-edge-locations	Select the price class associated with the maximum price that you want to pay for CloudFront service. If you select a price class other than All, some of your users may experience higher latency.

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	aws-quickstart	The S3 bucket where the Quick Start templates and scripts are installed. Use this parameter to specify the S3 bucket name you've created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens (-), but should not start or end with a hyphen.
Quick Start S3 Key Prefix (QSS3KeyPrefix)	quickstart-drupal/	The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens (-), and forward slashes (/), but should not start or end with a forward slash (which is automatically added).

- **Option 2: Parameters for deploying Drupal into an existing VPC**

[View template](#)

Network Configuration:

Parameter label (name)	Default	Description
VPC ID (VPCID)	<i>Requires input</i>	The ID of your existing VPC.
VPC CIDR (VPCCIDR)	<i>Requires input</i>	The CIDR block for the VPC. The range you specify must be in the form <i>x.x.x.x/x</i> .
Private Subnet 1 ID (PrivateSubnet1ID)	<i>Requires input</i>	The ID of the private subnet in Availability Zone 1 in your existing VPC for the Drupal and Amazon Aurora database instances.
Private Subnet 2 ID (PrivateSubnet2ID)	<i>Requires input</i>	The ID of the private subnet in Availability Zone 2 in your existing VPC for the Drupal and Amazon Aurora database instances.
Public Subnet 1 ID (PublicSubnet1ID)	<i>Requires input</i>	The ID of the public subnet in Availability Zone 1 in your existing VPC for the Application Load Balancer load balancer instances (e.g., subnet-9bc642ac).
Public Subnet 2 ID (PublicSubnet2ID)	<i>Requires input</i>	The ID of the public subnet in Availability Zone 2 in your existing VPC for the Application Load Balancer instances (e.g., subnet-e3246d8e).
Bastion Security Group ID (BastionSecurityGroupID)	<i>Requires input</i>	ID of the bastion host security group (e.g., sg-7f16e910).

Drupal Site Domain Configuration:

Parameter label (name)	Default	Description
Route53 Hosted Zone ID (Route53HostedZoneId)	<i>Requires input</i>	Route53 Hosted Zone ID. Required if Drupal Site Domain or CloudFront Alias are not default values.
Drupal Site Domain (DrupalSiteDomain)	localhost.local	Domain name of the site (e.g. example.com). Valid fully qualified domain name (FQDN) required when using SSL. Leave the default localhost.local for test environments.
CloudFront Alias (CloudFrontAlias)	cdn.default	Alias for the CloudFront distribution (e.g. cdn.example.com). Mandatory when using HTTPS/SSL and optional when using http.

Aurora Database Configuration:

Parameter label (name)	Default	Description
Enable Auto Minor Version Upgrade (DBAutoMinorVersionUpgrade)	true	Choose true to allow AWS to automatically update the DB versions on Amazon Aurora, if they are minor upgrades. If a new engine minor version contains significant bug fixes compared to a previously released minor version, AWS will schedule automatic upgrades for DB instances which have the Auto Minor Version Upgrade setting to “Yes”.
Backup Retention Period (DBBackupRetentionPeriod)	7	The number of days for which automatic DB snapshots are retained. For more information about setting a retention period, see Working with Automated Backups .
Database Instance Size (DBInstanceClass)	db.t2.small	The instance size for the Amazon Aurora database.
Database Admin Username (DBMasterUsername)	dbadmin	The username of the database admin account. This is a 1-16 character string that begins with a letter and includes only alphanumeric characters..
Database Admin Password (DBMasterUserPassword)	<i>Requires input</i>	The password of the database admin account. This should be between 8-41 characters and cannot contain a white space, @, / or “.
Multi-AZ Database (DBMultiAZ)	true	Choose false if you want to deploy the database instance in a single Availability Zone. By default, the database instance is deployed across two Availability Zones for high availability.

Drupal Configuration:

Parameter label (name)	Default	Description
PHP Version (PhpVersion)	php71	The PHP version to be installed for Drupal. Versions 5.6, 7.0, and 7.1 are supported. For more information, see PHP Supported Versions .
Drupal Version (DrupalVersion)	7	The Drupal version to install. This Quick Start supports Drupal version 7 or version 8. For more information, see the Drupal documentation .
Drupal Site Name (DrupalSiteName)	My Site	The descriptive name for your Drupal sample website. This string must consist of 5-64 alphanumeric characters.
Drupal Site Admin Email (DrupalSiteAdminEmail)	<i>Requires input</i>	The administrator email for the Drupal site.

Parameter label (name)	Default	Description
Drupal Site Admin Username (DrupalSiteAdminUsername)	admin	The administrator username for the Drupal site.
Drupal Site Admin Password (DrupalSiteAdminPassword)	<i>Requires input</i>	The administrator user password for the Drupal site.
Drupal Database Name (DrupalDbName)	drupaldb	The name of the database to create and use for Drupal.
Drupal Database Username (DrupalDbUsername)	drupal	The user account of the Drupal database.
Drupal Database Password (DrupalDbPassword)	<i>Requires input</i>	The user account password of the Drupal database.

Drupal Webserver Configuration:

Parameter label (name)	Default	Description
Webserver Instance Type (WebServerInstanceType)	t2.micro	The instance type of the web server.
SSH Keypair Name (KeyPair)	id_rsa_aws	Name of an existing EC2 KeyPair to enable SSH access to the instances.
Min Number of Instances (WebServerMinSize)	1	Minimum number of web server instances in Auto Scaling group
Max Number of Instances (WebServerMaxSize)	12	The maximum number of web server instances in the Auto Scaling group.
Desired Number of Instances (WebServerDesiredCapacity)	2	The desired capacity of web server instances in the Auto Scaling group.
Autoscaling Notification Email (AutoScalingNotificationEmail)	<i>Requires input</i>	The email address that notifications will be sent to when the environment is scaled up or down.
SSL certificate ARN (SSLCertificateId)	<i>Optional</i>	(Optional) The Amazon Resource Name (ARN) of the SSL certificate to use for the load balancer. You can create this by using AWS Certificate Manager (ACM).

ElastiCache Configuration:

Parameter label (name)	Default	Description
Enable ElastiCache (ElastiCacheEnable)	true	Enable ElastiCache for Drupal.
Enable ElastiCache Auto Minor Version Upgrade (ElastiCacheAutoMinorVersionUpgrade)	true	Select true/false to setup Auto Minor Version upgrade.
ElastiCache Node Type (ElastiCacheNodeType)	cache.t2.micro	The compute and memory capacity of nodes in a cache cluster.
Number of ElastiCache Nodes (ElastiCacheNumberOfNodes)	2	The number of cache nodes that the cache cluster should have.

CloudFront Configuration:

Parameter label (name)	Default	Description
Enable CloudFront (CloudFrontEnable)	true	Enable CloudFront Content Delivery Network.
CloudFront PriceClass (CloudFrontPriceClass)	use-all-edge-locations	Select the price class associated with the maximum price that you want to pay for CloudFront service. If you select a price class other than All, some of your users may experience higher latency.

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	aws-quickstart	S3 bucket where the Quick Start templates and scripts are installed. Use this parameter to specify the S3 bucket name you've created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens (-), but should not start or end with a hyphen.
Quick Start S3 Key Prefix (QSS3KeyPrefix)	quickstart-drupal/	The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens (-), and forward slashes (/), but should not start or end with a forward slash (which is automatically added).

- 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.
7. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the Drupal servers are ready. Wait for 15 to 20 minutes, depending on the instance type you chose, for the cluster to be ready.

Step 3. Test Your Drupal Deployment

After you deploy Drupal, you can validate the deployment.

1. In the AWS CloudFormation console, in the **Outputs** tab, choose the link for the **ELBURL** key, shown in Figure 4, to open the login page for Drupal. This is the web client interface for Drupal administration.

The screenshot shows the AWS CloudFormation console interface. At the top, there are navigation tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', and 'Change Sets'. The 'Outputs' tab is selected, displaying a table with the following data:

Key	Value	Description	Export Name
ELBURL	http://DRUPA-Appli-R9WSUU7PF1PP-2081619096.ap-southeast-2.elb.amazonaws.com	The URL of the ELB that you should ...	

The URL in the 'Value' column is highlighted with a red rectangular box.

Figure 4: AWS CloudFormation output link

2. Use the values you specified for the parameters **DrupalSiteAdminUsername** and **DrupalSiteAdminPassword** during launch to log in to the Drupal administrator site, as shown in Figure 5.

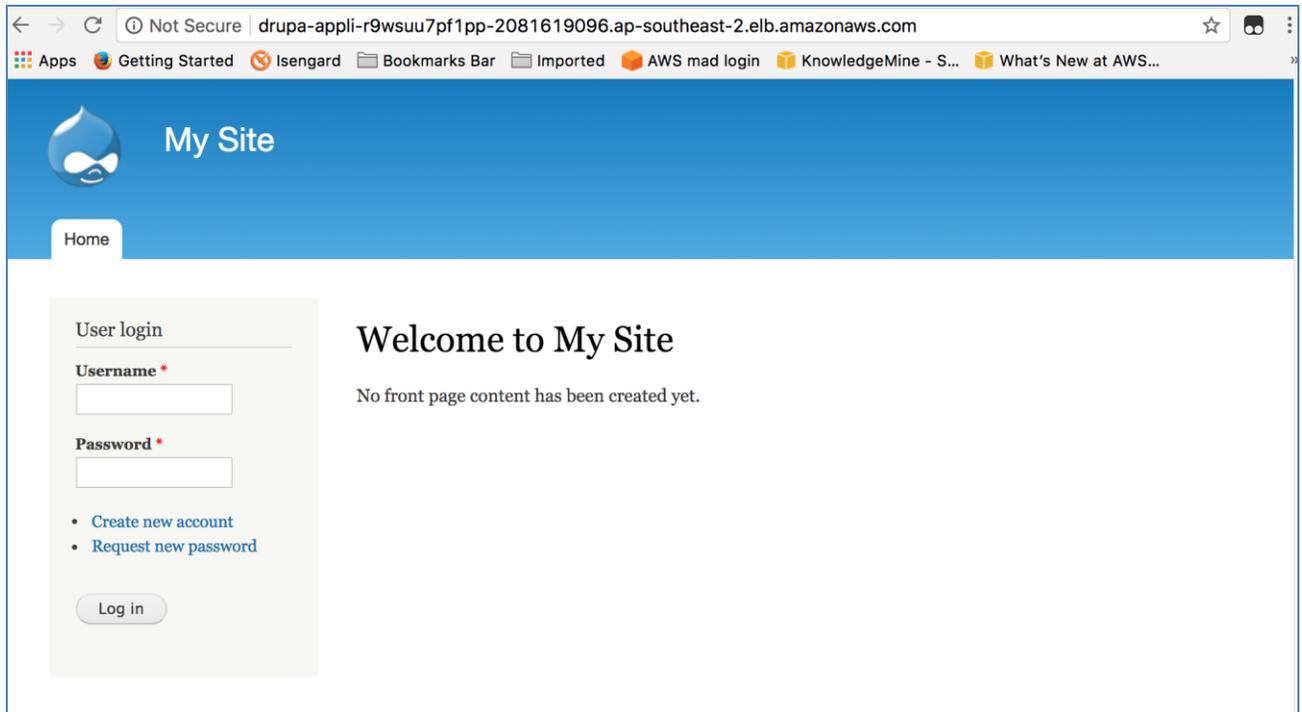


Figure 5: Drupal login page

3. Once successfully logged in, you should be able to see the screen as shown in Figure 6.

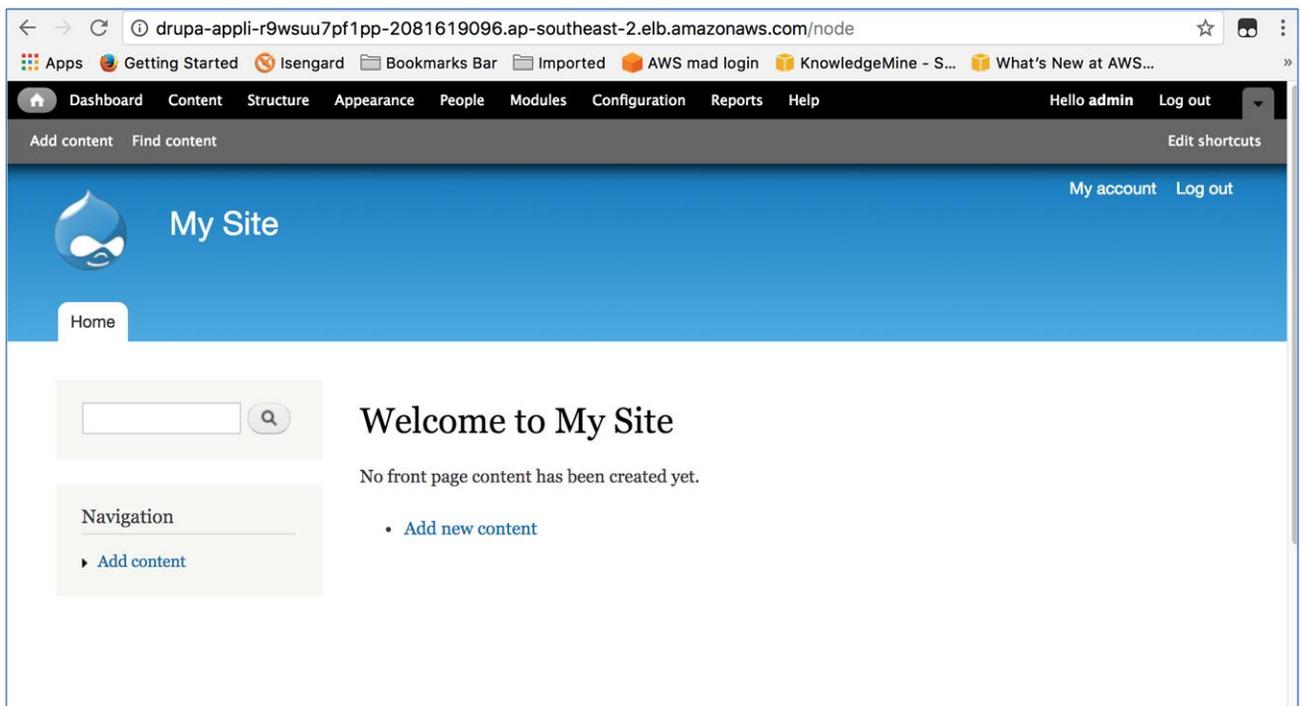


Figure 6: Drupal administrator site

For more information about using the environment that you've set up and validated, see the [Drupal documentation](#).

Troubleshooting

Q. I encountered a `CREATE_FAILED` error when I launched the Quick Start.

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. (Look at the `cloud-init` and `cfn` logs in `var/log/` directory).

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.

The following table lists specific `CREATE_FAILED` error messages you might encounter.

Error message	Possible cause	What to do
API: ec2: RunInstances Not authorized for images: ami-ID	The template is referencing an AMI that has expired	We refresh AMIs on a regular basis, but our schedule isn't always synchronized with AWS AMI updates. If you get this error message, notify us, and we'll update the template with the new AMI ID. If you'd like to fix the template yourself, you can download it and update the <code>Mappings</code> section with the latest AMI ID for your region.
We currently do not have sufficient <instance-type> capacity in the AZ you requested	At this time, Auto Scaling cannot support your instance type in your requested Availability Zone	Create a new launch configuration by following the recommendations in the error message. Update your Auto Scaling group with the new launch configuration using the update-auto-scaling-group command.
Instance ID did not stabilize	You have exceeded your IOPS for the region	Request a limit increase by completing the request form in the AWS Support Center.

For additional information, see [Troubleshooting AWS CloudFormation](#) on the AWS website.

Q. I encountered a size limitation error when I deployed the AWS CloudFormation templates.

A. We recommend that you launch the Quick Start templates from the location we've provided or from another S3 bucket. If you deploy the templates from a local copy on your computer or from a non-S3 location, you might encounter template size limitations when

you create the stack. For more information about AWS CloudFormation limits, see the [AWS documentation](#).

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

This solution uses an IAM role with least privileged access. We do not require or recommend storing SSH keys, secret keys, or access keys on the provisioned instances.

When you launch the Quick Start, if you select the check box to acknowledge that the template will create IAM resources (under **Capabilities**), AWS CloudFormation will automatically acquire the IAM resources.

Operating System 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.

This Quick Start deployment uses a separate stack to create the following security groups:

- Application Load Balancer
- Aurora RDS security group
- EFS security group
- Drupal security group
- NAT gateways and bastion hosts

GitHub Repository

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.

Additional Resources

AWS services

- Amazon EC2
<https://aws.amazon.com/documentation/ec2/>
- AWS IAM
<https://aws.amazon.com/documentation/iam/>
- Amazon RDS
<https://aws.amazon.com/documentation/rds/>
- Amazon S3
<https://aws.amazon.com/documentation/s3/>
- Amazon VPC
<https://aws.amazon.com/documentation/vpc/>
- AWS Auto Scaling
<https://aws.amazon.com/documentation/autoscaling/>
- AWS CloudFormation
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/>
- Elastic Load Balancing
<https://aws.amazon.com/documentation/elastic-load-balancing/>

- Amazon ElastiCache
<https://aws.amazon.com/documentation/elasticache/>
- Amazon CloudFront
<https://aws.amazon.com/documentation/cloudfront/>
- Amazon Route 53
<https://aws.amazon.com/documentation/route53/>

Drupal

- Drupal
<https://www.drupal.org>
- Drupal documentation
<https://www.drupal.org/documentation>
- Drupal with Memcache module
<https://www.drupal.org/project/memcache>
- Drupal CDN module
<https://www.drupal.org/project/cdn>

Quick Start reference deployments

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

Document Revisions

Date	Change	In sections
August 2018	Integrated Amazon CloudFront, Amazon ElastiCache and Amazon Route 53 into architecture; updated parameters	Architecture-AWS Components; Launch the Quick Start
February 2018	Initial version	-

© 2020, 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.