Alfresco Content Services on the AWS Cloud

Quick Start Reference Deployment

Alfresco Software
AWS Quick Start Reference Team

January 2017
(last update: September 2017)

Contents

About This Guide

This Quick Start deployment guide discusses architectural considerations and configuration steps for deploying an Alfresco Content Services cluster on the Amazon Web Services (AWS) Cloud. It also provides links for viewing and launching AWS CloudFormation templates that automate the deployment.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their Alfresco Content Services 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.

This Quick Start deployment guide was created by Amazon Web Services (AWS) in partnership with Alfresco Software.

Overview

Enterprises need to grow and manage their global computing infrastructures rapidly and efficiently while simultaneously optimizing and managing capital costs and expenses. The computing and storage services from AWS meet this need by providing a global computing infrastructure as well as 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.

Alfresco Content Services is used by organizations interested in managing business-critical processes that relate to document management, collaboration, and secure mobile and desktop access to vital files. The flexible compute, storage, and database services that AWS offers make it an ideal platform on which to run an Alfresco Content Services deployment.

Alfresco Content Services on AWS

Alfresco Content Services is an Enterprise Content Management (ECM) system that is used for document and case management, project collaboration, web content publishing, and compliant records management. Few classes of business-critical applications touch more enterprise users than ECM and collaboration systems. AWS provides a complete set of services and tools for deploying business-critical enterprise workloads on its highly reliable and secure cloud infrastructure.

This document provides IT infrastructure decision-makers and system administrators with technical guidance on how to configure, deploy, and run an Alfresco Content Services server cluster on AWS. It outlines a reference architecture for an Alfresco Content Services version 5.2 deployment that addresses common scalability, high availability, and security requirements.

This guide discusses best practices for deploying Alfresco Content Services on AWS using 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 launch directly into your AWS account. The accompanying AWS CloudFormation templates and information included in this guide can be modified to suit your specific business requirements, or they can be used as is.

For in-depth information about installing and using Alfresco Content Services, see the Alfresco documentation.

Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the prerequisites for deployment discussed in this guide.

- If you have an AWS account, and you're already familiar with AWS services and Alfresco Content Services, you can launch the Quick Start to build the architecture shown in Figure 2 in a new or existing VPC. (See Costs and Licenses.) The deployment takes about an hour. If you're new to AWS or to Alfresco Content Services, please review the implementation details and follow the step-by-step instructions provided in this guide.

Launch (for new VPC) Launch (for existing VPC)

- If you want to take a look under the covers, you can view the AWS CloudFormation templates that automate the deployment.

View template (for new VPC) View template (for existing VPC)

Cost and Licenses

This deployment launches Alfresco Content Services 5.2 automatically into a configuration of your choice. You will need a valid Alfresco license, and 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.

See the pricing pages for each AWS service you will be using for full details. Note that this Quick Start includes nested AWS CloudFormation templates that launch multiple stacks.

You have two options to obtain an Alfresco license:

- **Bring your own license (BYOL).** If you have a valid subscription to Alfresco Content Services, you can use that license, subject to complying with standard licensing terms. Additional pricing, terms, and conditions may apply.

  —or—

- **Trial license.** You may apply for a 30-day AWS Quick Start license from Alfresco by filling out the form at https://www.alfresco.com/platform/content-services-ecm/trial/aws.

AWS Services

The core AWS components used by this Quick Start include the following AWS services.

- 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, creation of subnets, 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 a relational database such as Amazon Aurora or RDS MySQL in the cloud. With Amazon RDS, you can deploy scalable Amazon Aurora or RDS MySQL software in minutes with cost-efficient and resizable hardware capacity
- Auto Scaling – 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.
- Elastic Load Balancing – Elastic Load Balancing 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. Elastic Load Balancing automatically scales its request handling capacity in response to incoming traffic.
- Amazon S3 – Amazon Simple Storage Service (Amazon S3) provides secure, durable, highly scalable object storage. This Alfresco Content Services deployment uses Amazon S3 as the content store for Alfresco Content Services servers and Index server instances to access the shared content store.

- 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.

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, with your compliance and network latency requirements in mind. Regions are designed with availability in mind and consist of at least two Availability Zones.

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.

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). Service endpoints, which are managed by AWS, provide management ("backplane") access.

Architecture

Alfresco Content Services supports a wide variety of content management use cases, including documents, records, web publishing, and more. This Quick Start presents an enterprise-grade Alfresco Content Services configuration that you can adapt to virtually any scenario, scaling up, down, or out depending on the use case adopted.

The Quick Start deployment is automated by nested AWS CloudFormation templates. 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. The main template builds the network-related resources first and then launches two separate templates for Alfresco Content Services and Index servers, and Amazon Aurora (RDS MySQL if you deploy the Quick Start in the EU (Frankfurt) Region). Deleting the main template deletes the entire stack.

This AWS CloudFormation template also deploys Alfresco Content Services and all its components using a basic script to prepare every instance, and then calls the Chef Alfresco recipes to do the rest. The result is an Alfresco Content Services environment with Index servers, plus other components that make the environment more reliable, less error-prone, and scalable. Figure 1 provides a high-level view of the resulting architecture.

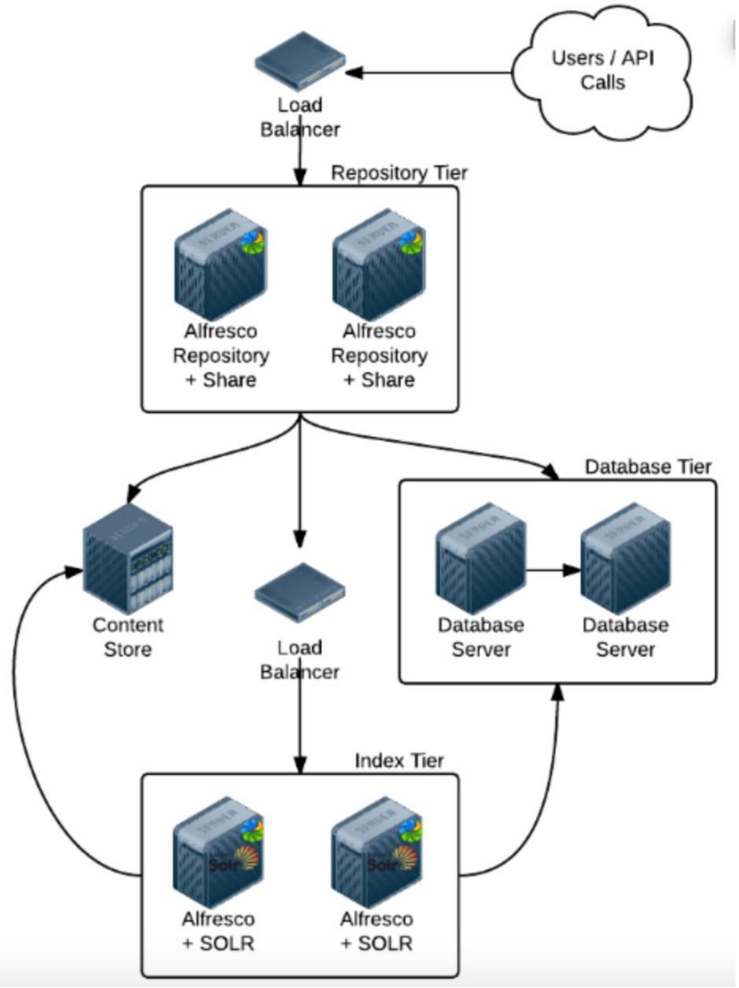


Figure 1: High-level Alfresco architecture on AWS

Alfresco Content Services Components

The reference architecture deployed by the Quick Start maps AWS services to all the components required by Alfresco Content Services. The reference architecture for the Alfresco Content Services cluster requires the following components:

- An HTTP(S) load balancer
- Two or more Alfresco Content Services servers
- Two or more Alfresco Content Services Index servers
- Shared file storage
- A shared database

AWS Components

The Quick Start uses AWS CloudFormation templates and Chef to install and configure an Alfresco Content Services cluster that meets the requirements described in the previous section. The following list describes the AWS services that map to the Alfresco Content Services components.

> **Note** You can run each of these components using Amazon EC2. The Alfresco Content Services environment can be configured without AWS, but to simplify administration and help lower your overall costs, we recommend that you use AWS services.

- A virtual private cloud (VPC) configured across two Availability Zones. For each Availability Zone, this Quick Start provisions one public subnet and one private subnet.
- Managed network address translation (NAT) gateways deployed into the public subnets and configured with an Elastic IP address (EIP) 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.
- The Elastic Load Balancing service, which provides HTTP and HTTPS load balancing across the Alfresco Content Services servers.

> **Note** When you use Elastic Load Balancing, you must upload the web server's certificate and private key to the AWS Identity and Access Management (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.
- Auto Scaling enabled, which allows your Alfresco Content Services cluster to add or reduce 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 the Elastic Load Balancing service 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 Elastic Load Balancing. The security groups also restrict access to Amazon Aurora DB instances by web server instances, or RDS MySQL DB instances if you deploy the Quick Start in the EU (Frankfurt) Region.
- Amazon S3, which provides shared file storage for the cluster. Amazon S3 is an ideal storage system for Alfresco Content Services for several reasons:
  - It offers highly durable object storage designed to provide 99.999999999% durability, which means you that no longer need to manage backups of your content store, unless you need point-in-time recovery.

    > **Note** Backup procedures are not covered in this deployment guide. For more information, see the Alfresco documentation. For additional details, see the Resources section.

  - Alfresco Content Services stores items as objects. Changes to objects are stored as unique objects rather than as updates to existing objects. This makes Amazon S3 a perfect storage system, because POSIX compatibility is not required.
  - Amazon S3 provides unlimited scalability with support for an unlimited number of objects up to 5 TiB in size, and customers are charged only for storage that is used. This greatly simplifies sizing your environment because you don't have to worry about how much space your cluster will need in the future, and your storage costs map directly to the amount of storage that you use.
- Amazon Aurora (or RDS MySQL in the EU (Frankfurt) Region) in Multi-AZ deployment is used for the shared database. Amazon RDS is a managed database service, so all the

administrative tasks for managing the database are handled by AWS. The database is deployed in multiple Availability Zones for high availability and automatically backed up on a schedule that you define.

- The AWS CloudFormation template includes Amazon Aurora as the default database (RDS MySQL is used if you choose the EU (Frankfurt) Region). Amazon Aurora has been tested and certified for Alfresco with a billion document benchmark.

Alfresco Content Services Architecture

To ensure high availability, this architecture deploys the Alfresco Content Services servers and Index servers across two Availability Zones within a region. The Multi-AZ feature is enabled for the Amazon RDS database, which is deployed in both Availability Zones in a master/slave configuration.

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

With Amazon VPC, you can create a deployment in which all the Alfresco Content Services instances and Amazon RDS database instances are in private subnets, exposing only the Elastic Load Balancing listener, NAT gateways, and bastion host instance to the Internet.

Figure 2 illustrates this architecture.

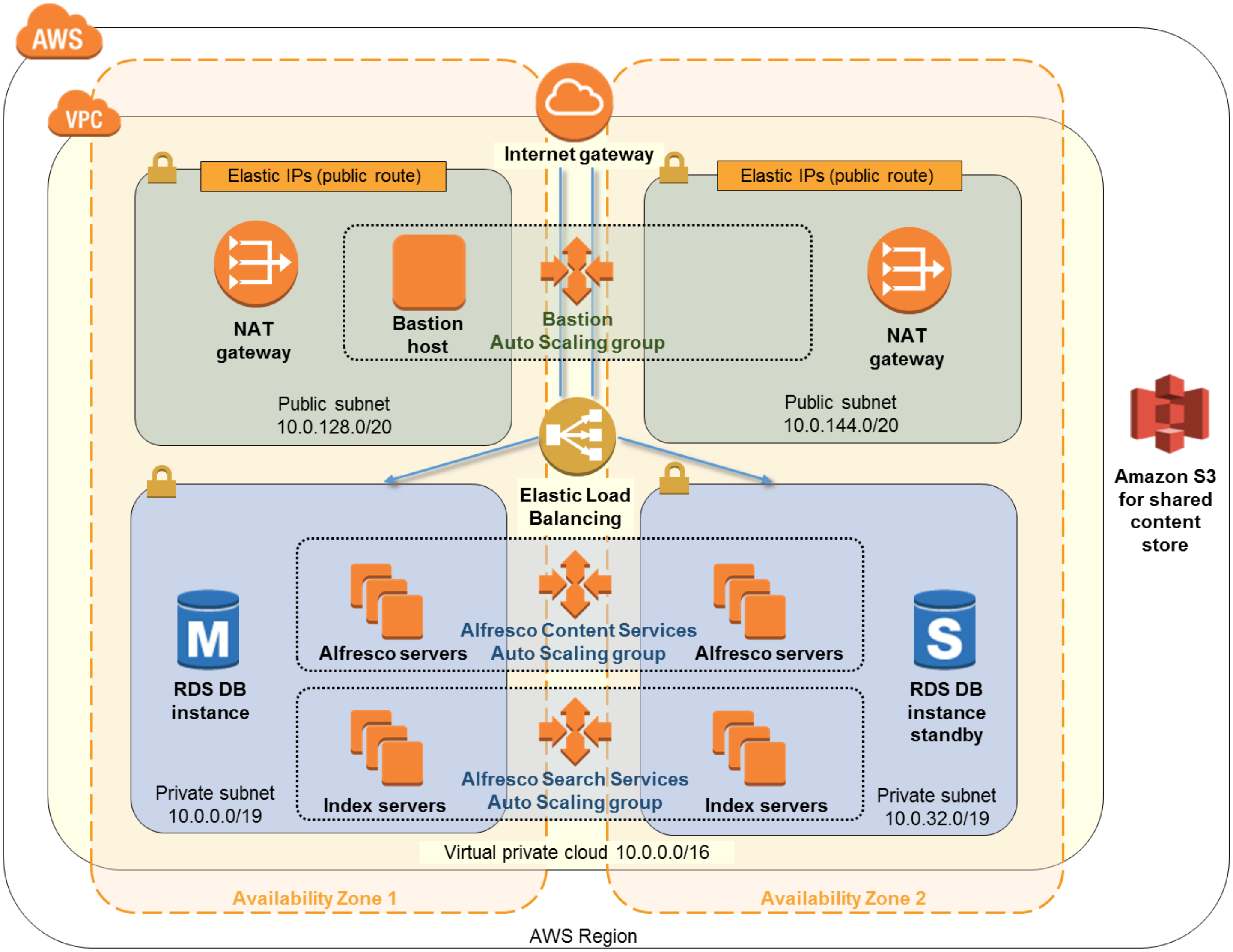


Figure 2: Alfresco Content Services reference architecture

Note that Amazon VPC also gives you control over several networking aspects of a deployment. For example, when you create the VPC, you define the overall IP address space of the VPC as well as the IP space that each subnet will use.

The IP space of the VPC and its subnets is 10.0.0.0/16, and the subnets for the Alfresco Content Services servers, Index servers, and RDS DB instances are set to 10.0.0.0/19 and 10.0.32.0/19.

Public subnets 10.0.128.0/20 and 10.0.144.0/20 are dedicated to the bastion host instances and NAT gateways.

Deployment Options

This Quick Start provides two deployment options:

- **Deploy Alfresco Content Services 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 Alfresco Content Services into this new VPC.
- **Deploy Alfresco Content Services into an existing VPC**. This option provisions Alfresco Content Services in your existing AWS infrastructure.

The Quick Start also lets you configure additional settings such as CIDR blocks, instance types, and Alfresco Content Services settings, as discussed later in this guide.

Deployment Steps

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of an Alfresco Content Services 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.

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

What We'll Cover

The procedure for deploying the Alfresco Content Services 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 Amazon EBS volumes, and request a limit increase, if needed.

Step 2. Obtain an Alfresco Content Services license

- Request an Alfresco trial license if you do not already have a valid existing Alfresco license.

Step 3. Subscribe to the Alfresco Content Services AMI

- Accept terms to use the Alfresco Content Services AMI from the AWS Marketplace.

Step 4. Launch the Quick Start into your AWS account (new VPC):

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters, including your license.
- Review the other template parameters, and customize their values if necessary.

The Quick Start provides separate templates for end-to-end deployment and deployment into an existing VPC. Each template creates Alfresco Content Services servers in clusters with Alfresco Content Services Index servers, automatically configures them with Chef, and configures the AWS Auto Scaling and Elastic Load Balancing services. When you deploy Alfresco Content Services into a new VPC, the template additionally creates the required AWS infrastructure.

Step 5. Test Your Alfresco Content Services Deployment

- Access the Alfresco Share web UI to confirm that the Alfresco Content Services cluster was created and configured successfully.

The steps described in this guide will implement a complete Alfresco Content Services environment on the AWS Cloud.

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.

   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. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy Alfresco Content Services 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 **m4.xlarge** instance type for the Alfresco Content Services portion of the deployment by default, but you can choose other instance types.

Note This Quick Start doesn't support the Canada (Central), EU (London), AWS GovCloud (US), and China (Beijing) Regions. Amazon Aurora is supported in all regions except for EU (Frankfurt).

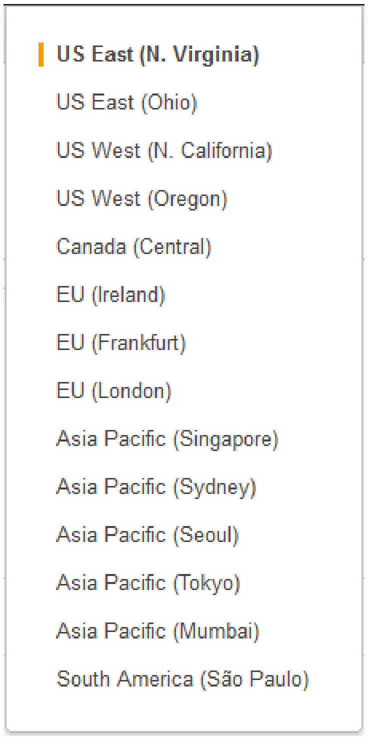


Figure 3: Choosing an AWS Region

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**.

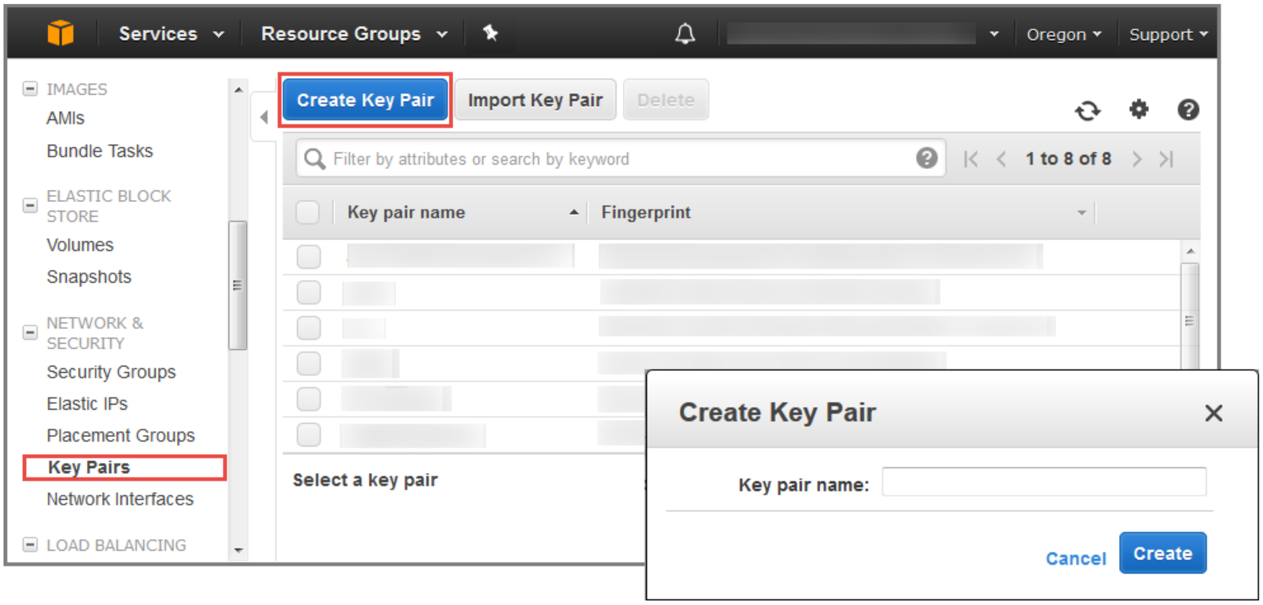
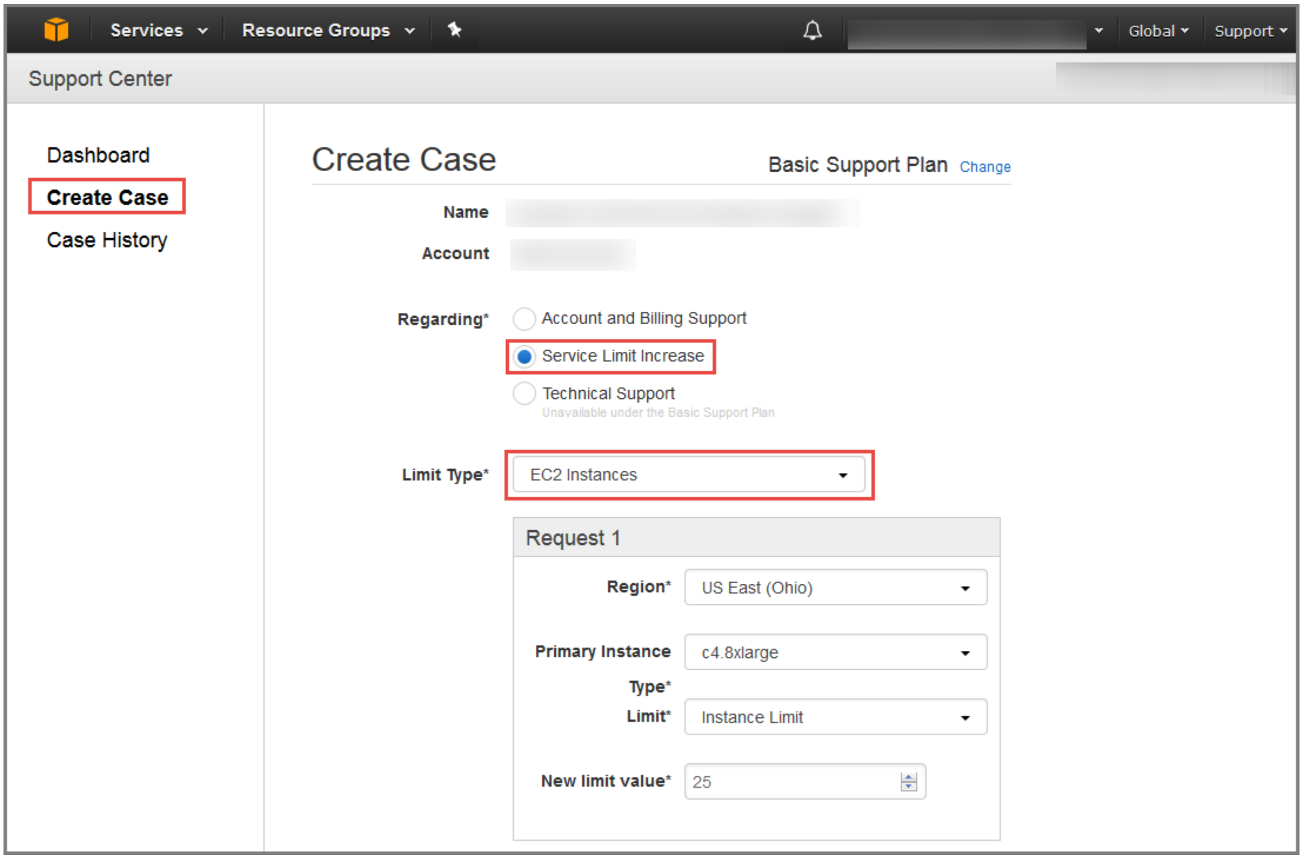


Figure 4: Creating a key pair

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

4. If necessary, request a service limit increase for the Amazon EC2 m4.xlarge 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 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.



Name [redacted]

Account [redacted]

Regarding*
- ○ Account and Billing Support
- ● Service Limit Increase
- ○ Technical Support (Unavailable under the Basic Support Plan)

Limit Type*: EC2 Instances

Request 1

Region*: US East (Ohio)

Primary Instance Type*: c4.8xlarge

Limit*: Instance Limit

New limit value*: 25

Figure 5: Requesting a service limit increase

Step 2. Obtain an Alfresco Content Services License

You will need to use a valid Alfresco Content Services license as part of the Quick Start process.

If you have an existing Alfresco subscription, you can use that license (BYOL model), subject to standard licensing terms and conditions.

If you are not yet an Alfresco customer, you can request a 30-day trial license for this Quick Start at https://www.alfresco.com/platform/content-services-ecm/trial/aws.

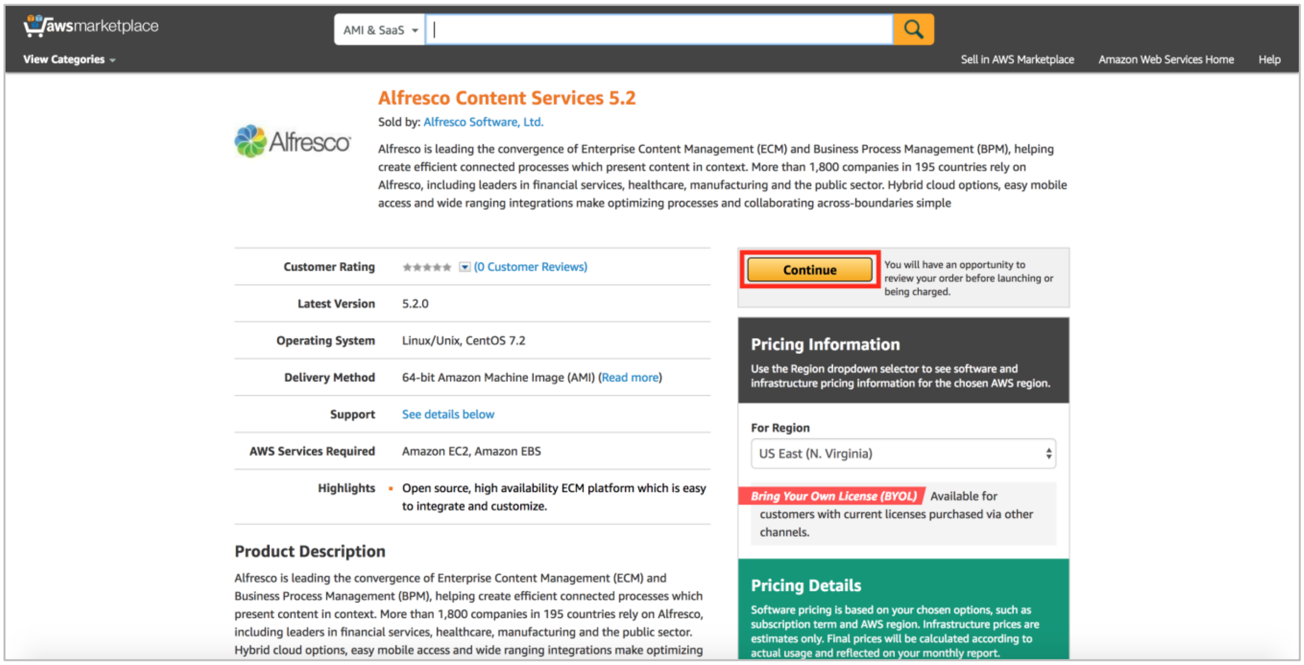
Once you have a valid license, upload the license file to Amazon S3 or another publicly accessible Internet location, and note its URL. You'll need to supply the URL in the **AlfrescoTrialLicense** parameter when you launch the AWS CloudFormation template in step 4.

Step 3. Subscribe to the Alfresco Content Services AMI

The Alfresco Content Services software is available from the AWS Marketplace.

To subscribe to the Alfresco Content Services AMI:

1. Log in to your AWS account.
2. Open the AWS Marketplace Alfresco Content Services page (Figure 6).
3. Choose **Continue** to view the license terms and launch information.



Alfresco Content Services 5.2

Sold by: Alfresco Software, Ltd.

Alfresco is leading the convergence of Enterprise Content Management (ECM) and Business Process Management (BPM), helping create efficient connected processes which present content in context. More than 1,800 companies in 195 countries rely on Alfresco, including leaders in financial services, healthcare, manufacturing and the public sector. Hybrid cloud options, easy mobile access and wide ranging integrations make optimizing processes and collaborating across-boundaries simple

Customer Rating	★★★★★ (0 Customer Reviews)
Latest Version	5.2.0
Operating System	Linux/Unix, CentOS 7.2
Delivery Method	64-bit Amazon Machine Image (AMI) (Read more)
Support	See details below
AWS Services Required	Amazon EC2, Amazon EBS
Highlights	• Open source, high availability ECM platform which is easy to integrate and customize.

Product Description

Alfresco is leading the convergence of Enterprise Content Management (ECM) and Business Process Management (BPM), helping create efficient connected processes which present content in context. More than 1,800 companies in 195 countries rely on Alfresco, including leaders in financial services, healthcare, manufacturing and the public sector. Hybrid cloud options, easy mobile access and wide ranging integrations make optimizing

Continue — You will have an opportunity to review your order before launching or being charged.

Pricing Information

Use the Region dropdown selector to see software and infrastructure pricing information for the chosen AWS region.

For Region: US East (N. Virginia)

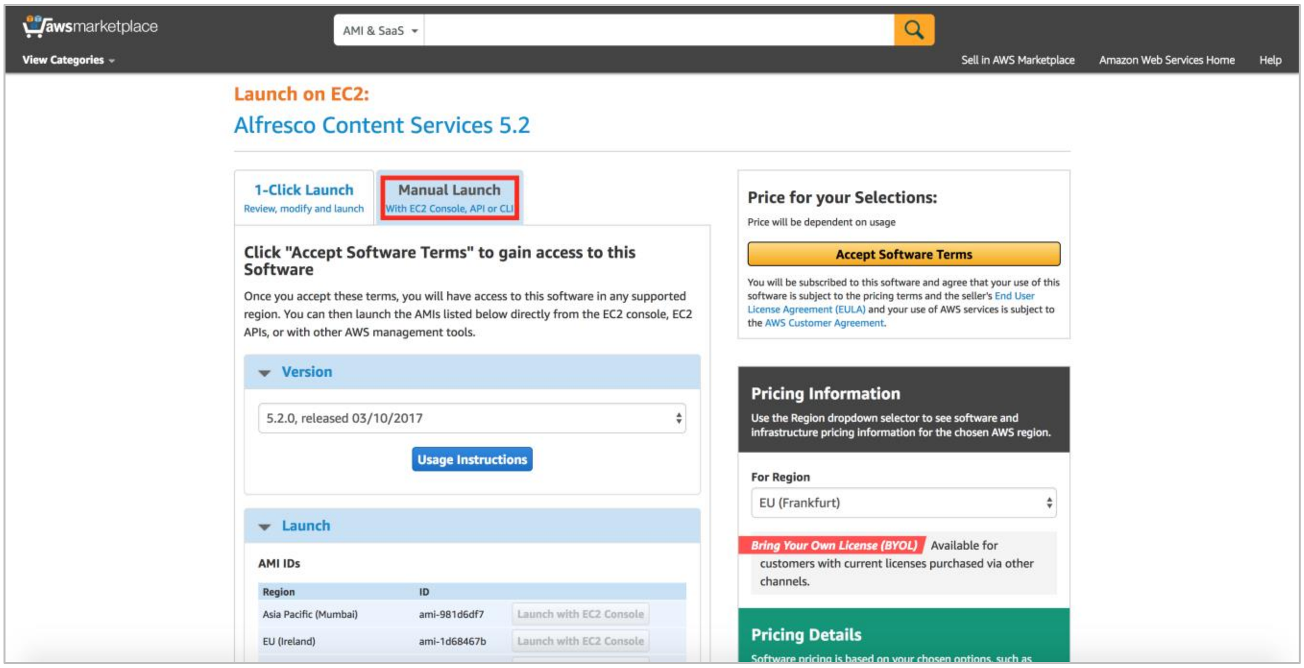
Bring Your Own License (BYOL) Available for customers with current licenses purchased via other channels.

Pricing Details

Software pricing is based on your chosen options, such as subscription term and AWS region. Infrastructure prices are estimates only. Final prices will be calculated according to actual usage and reflected on your monthly report.

Figure 6: Alfresco Content Services in the AWS Marketplace

4. Choose the **Manual Launch** tab, and then choose **Accept Software Terms** (Figure 7).



Launch on EC2:

Alfresco Content Services 5.2

Click "Accept Software Terms" to gain access to this Software

Once you accept these terms, you will have access to this software in any supported region. You can then launch the AMIs listed below directly from the EC2 console, EC2 APIs, or with other AWS management tools.

Version: 5.2.0, released 03/10/2017

AMI IDs

Region	ID	
Asia Pacific (Mumbai)	ami-981d6df7	Launch with EC2 Console
EU (Ireland)	ami-1d68467b	Launch with EC2 Console

Price for your Selections:

Price will be dependent on usage

You will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement (EULA) and your use of AWS services is subject to the AWS Customer Agreement.

Pricing Information

Use the Region dropdown selector to see software and infrastructure pricing information for the chosen AWS region.

Bring Your Own License (BYOL) Available for customers with current licenses purchased via other channels.

Pricing Details

Software pricing is based on your chosen options, such as

Figure 7: Subscribing to the Alfresco Content Services AMI

Step 4. Launch the AWS CloudFormation Template

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.

Option 1	Option 2
Deploy Alfresco Content Services into a new VPC	**Deploy Alfresco Content Services into an existing VPC**
Launch	Launch

Important If you're deploying Alfresco Content Services into an existing VPC, make sure that your VPC has two private subnets in different Availability Zones for the Alfresco 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 one hour to create.

> **Note** You are responsible for the cost of the Alfresco Content Services AMI and 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.

5. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.
6. 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. For detailed information about the template, see the section Template Details and Customization.

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

   – Parameters for deploying Alfresco Content Services into a new VPC

   – Parameters for deploying Alfresco Content Services into an existing VPC

- **Option 1: Parameters for deploying Alfresco Content Services into a new VPC**

View template

Network Configuration:

Parameter label (name)	Default	Description
Availability Zones (AvailabilityZones)	*Requires input*	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	CIDR block for the VPC.
Private Subnet 1 CIDR (PrivateSubnet1CIDR)	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
Private Subnet 2 CIDR (PrivateSubnet2CIDR)	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.

Parameter label (name)	Default	Description
Public Subnet 1 CIDR (PublicSubnet1CIDR)	10.0.128.0/20	CIDR block for the public (DMZ) subnet located in Availability Zone 1.
Public Subnet 2 CIDR (PublicSubnet2CIDR)	10.0.144.0/20	CIDR block for the public (DMZ) subnet located in Availability Zone 2.
Allowed Bastion External Access CIDR (RemoteAccessCIDR)	*Requires input*	The CIDR IP range that is permitted to access the Alfresco Content Services software. 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.

Amazon EC2 Configuration:

Parameter label (name)	Default	Description
Key Name (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.
Bastion AMI Operating System (BastionAMIOS)	Amazon-Linux-HVM	EC2 instance operating system for the bastion host instance located in Availability Zone 1.
Bastion Instance Type (BastionInstanceType)	t2.micro	EC2 instance type for the bastion host instance located in Availability Zone 1.
Alfresco Servers Instance Type (AlfrescoInstanceType)	m4.xlarge	EC2 instance type for the Alfresco Content Services repository and Alfresco Share instances.
Index Servers Instance Type (IndexInstanceType)	m4.xlarge	EC2 instance type for the Index server (Alfresco Content Services repository and Solr) instances.

Alfresco General Configuration:

Parameter label (name)	Default	Description
Alfresco Admin Password (AlfrescoPassword)	*Requires input*	Password for the Alfresco Content Services admin user.
Alfresco Trial License Location URL (AlfrescoTrialLicense)	*Requires input*	URL for the Alfresco Content Services trial license with clustering enabled, from step 2. If you don't have a trial license, request one from your Alfresco Sales Representative by filling out the form at https://www.alfresco.com/company/contact.

Alfresco Nodes Configuration:

Parameter label (name)	Default	Description
Alfresco Nodes Min Size (AlfrescoNodesMinSize)	2	Minimum size of the Auto Scaling group for the Alfresco Content Services instances.
Alfresco Nodes Max Size (AlfrescoNodesMaxSize)	4	Maximum size of the Auto Scaling group for the Alfresco Content Services instances.
Alfresco Nodes Desired Capacity (AlfrescoNodesDesiredCapacity)	2	Desired capacity of the Auto Scaling group for the Alfresco Content Services instances.
Index Nodes Min Size (IndexNodesMinSize)	2	Minimum size of the Auto Scaling group for the Index server instances.
Index Nodes Max Size (IndexNodesMaxSize)	4	Maximum size of the Auto Scaling group for the Index server instances.
Index Nodes Desired Capacity (IndexNodesDesiredCapacity)	2	Desired capacity of the Auto Scaling group for the Index server instances.
Operator Email (OperatorEmail)	*Requires input*	Email that notifications will be sent to when the environment is scaled up or down.

Alfresco Storage Configuration:

Parameter label (name)	Default	Description
RDS Instance Type (RDSInstanceType)	db.t2.small	Amazon EC2 instance type for the RDS DB instance.
RDS Allocated Storage (RDSAllocatedStorage)	5	Size, in GiB, for the Amazon RDS MySQL database allocated storage. This setting is used only in the EU (Frankfurt) Region.
RDS DB Name (RDSDBName)	alfresco	Name for the Amazon RDS database.
RDS User Name (RDSUsername)	alfresco	User name for the Amazon RDS database.
RDS Password (RDSPassword)	alfresco	Password for the Amazon RDS database; change it if required.
S3 Bucket Name (S3BucketName)	*Requires input*	Name of the Amazon S3 bucket that Alfresco should use to store data. This bucket must have a unique name and will be created during deployment.

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

Parameter label (name)	Default	Description
		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-alfresco-content-services/	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.

• Option 2: Parameters for deploying Alfresco Content Services into an existing VPC

View template

Network Configuration:

Parameter label (name)	Default	Description
VPC ID (VPCID)	*Requires input*	ID of your existing VPC (e.g., vpc-0343606e).
Private Subnet 1 ID (PrivateSubnet1ID)	*Requires input*	ID of the private subnet in Availability Zone 1 in your existing VPC for the Alfresco Content Services and RDS DB instances (e.g., subnet-a0246dcd).
Private Subnet 2 ID (PrivateSubnet2ID)	*Requires input*	ID of the private subnet in Availability Zone 2 in your existing VPC for the Alfresco Content Services and RDS DB instances (e.g., subnet-b1f432cd).
Public Subnet 1 ID (PublicSubnet1ID)	*Requires input*	ID of the public subnet in Availability Zone 1 in your existing VPC for the Elastic Load Balancing (ELB) load balancer instances (e.g., subnet-9bc642ac).
Public Subnet 2 ID (PublicSubnet2ID)	*Requires input*	ID of the public subnet in Availability Zone 2 in your existing VPC for the ELB load balancer instances (e.g., subnet-e3246d8e).

Amazon EC2 Configuration:

Parameter label (name)	Default	Description
SSH Key Name (KeyPairName)	*Requires input*	The name of an existing public/private key pair from step 1. All instances will launch with this key pair.
Bastion Security Group ID (BastionSecurityGroupID)	*Requires input*	ID of the security group for the bastion host instance for SSH connections (e.g., sg-7f16e910).

Parameter label (name)	Default	Description
Alfresco Servers Instance Type (AlfrescoInstanceType)	m4.xlarge	EC2 instance type for the Alfresco Content Services repository and Alfresco Share instances.
Index Servers Instance Type (IndexInstanceType)	m4.xlarge	EC2 instance type for the Index server (Alfresco Content Services repository and Solr) instances.

Alfresco General Configuration:

Parameter label (name)	Default	Description
Alfresco Admin Password (AlfrescoPassword)	*Requires input*	Password for the Alfresco Content Services admin user.
Alfresco Trial License Location URL (AlfrescoTrialLicense)	*Requires input*	URL for the Alfresco Content Services trial license from step 2. If you don't have a trial license, please request one at https://www.alfresco.com/platform/content-services-ecm/trial/aws.

Alfresco Nodes Configuration:

Parameter label (name)	Default	Description
Alfresco Nodes Min Size (AlfrescoNodesMinSize)	2	Minimum size of the Auto Scaling group for the Alfresco Content Services instances.
Alfresco Nodes Max Size (AlfrescoNodesMaxSize)	4	Maximum size of the Auto Scaling group for the Alfresco Content Services instances.
Alfresco Nodes Desired Capacity (AlfrescoNodesDesiredCapacity)	2	Desired capacity of the Auto Scaling group for the Alfresco Content Services instances.
Index Nodes Min Size (IndexNodesMinSize)	2	Minimum size of the Auto Scaling group for the Index server instances.
Index Nodes Max Size (IndexNodesMaxSize)	4	Maximum size of the Auto Scaling group for the Index server instances.
Index Nodes Desired Capacity (IndexNodesDesiredCapacity)	2	Desired capacity of the Auto Scaling group for the Index server instances.
Operator Email (OperatorEmail)	*Requires input*	Email that notifications will be sent to when the environment is scaled up or down.

Alfresco Storage Configuration:

Parameter label (name)	Default	Description
RDS Instance Type (RDSInstanceType)	db.t2.small	Amazon EC2 instance type for the RDS DB instance.

Parameter label (name)	Default	Description
RDS Allocated Storage (RDSAllocatedStorage)	5	Size, in GiB, for the Amazon RDS MySQL database allocated storage. This setting is used only in the EU (Frankfurt) Region.
RDS DB Name (RDSDBName)	alfresco	Name for the Amazon RDS database.
RDS User Name (RDSUsername)	alfresco	User name for the Amazon RDS database.
RDS Password (RDSPassword)	alfresco	Password for the Amazon RDS database; change it if required.
S3 Bucket Name (S3BucketName)	*Requires input*	Name of the Amazon S3 bucket that Alfresco should use to store data. This bucket must have a unique name and will be created during deployment.

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-alfresco-content-services/	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.

7. 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**.

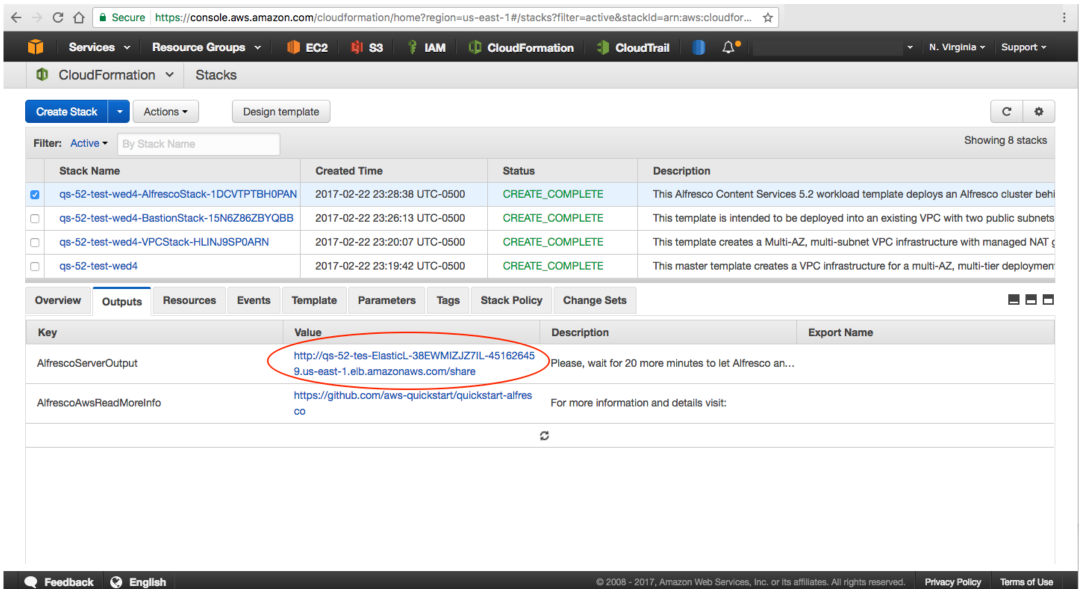
8. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.

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

10. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the Alfresco Content Services cluster will be starting. Wait for 15 to 20 minutes, depending on the instance type you chose, for the cluster to be ready.

Step 5. Test Your Alfresco Content Services Deployment

This completes the creation of the Alfresco Content Services cluster. Choose the output link shown in Figure 8 to open the login page for Alfresco Share, which is the web client interface for Alfresco Content Services.



	Stack Name	Created Time	Status	Description
☑	qs-52-test-wed4-AlfrescoStack-1DCVTPTBH0PAN	2017-02-22 23:28:38 UTC-0500	CREATE_COMPLETE	This Alfresco Content Services 5.2 workload template deploys an Alfresco cluster behi
□	qs-52-test-wed4-BastionStack-15N6Z86ZBYQBB	2017-02-22 23:26:13 UTC-0500	CREATE_COMPLETE	This template is intended to be deployed into an existing VPC with two public subnets
□	qs-52-test-wed4-VPCStack-HLINJ9SP0ARN	2017-02-22 23:20:07 UTC-0500	CREATE_COMPLETE	This template creates a Multi-AZ, multi-subnet VPC infrastructure with managed NAT g
□	qs-52-test-wed4	2017-02-22 23:19:42 UTC-0500	CREATE_COMPLETE	This master template creates a VPC infrastructure for a multi-AZ, multi-tier deploymen

Key	Value	Description	Export Name
AlfrescoServerOutput	http://qs-52-tes-ElasticL-38EWMIZJZ7IL-451626459.us-east-1.elb.amazonaws.com/share	Please, wait for 20 more minutes to let Alfresco an...	
AlfrescoAwsReadMoreInfo	https://github.com/aws-quickstart/quickstart-alfresco	For more information and details visit:	

Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Figure 8: AWS CloudFormation output link

Use the admin password you specified for the **AlfrescoPassword** parameter during launch to log in to Alfresco Content Services.

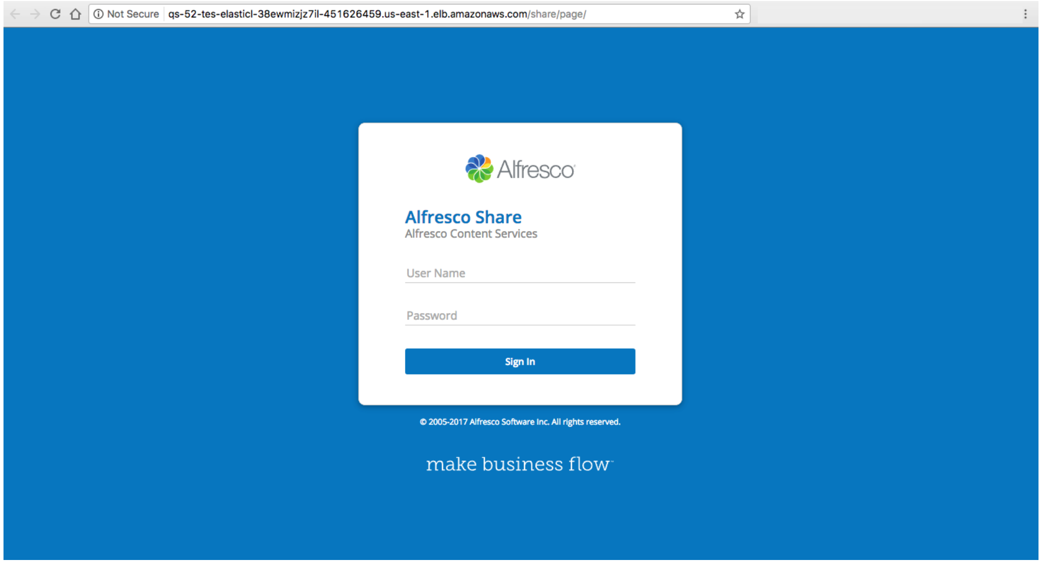

Not Secure qs-52-tes-elasticl-38ewmizjz7il-451626459.us-east-1.elb.amazonaws.com/share/page/
Alfresco
Alfresco Share
Alfresco Content Services
User Name
Password
Sign In
© 2005-2017 Alfresco Software Inc. All rights reserved.
make business flow

Figure 9: Alfresco Share login page

You may see a blank page instead of the login page shown in Figure 9, because it takes approximately 20 more minutes to create the new instances, including the cluster configuration of Alfresco Content Services instances and Index servers. It takes 5 minutes or less for the Alfresco Content Services servers to auto-discover the Index servers and start working.

Figure 10 illustrates the deployment timeline. Software installation times depend on network latency and instance types. Estimates are based on the US East (N. Virginia) Region (`us-east-1`) and the m4.xlarge instance type.

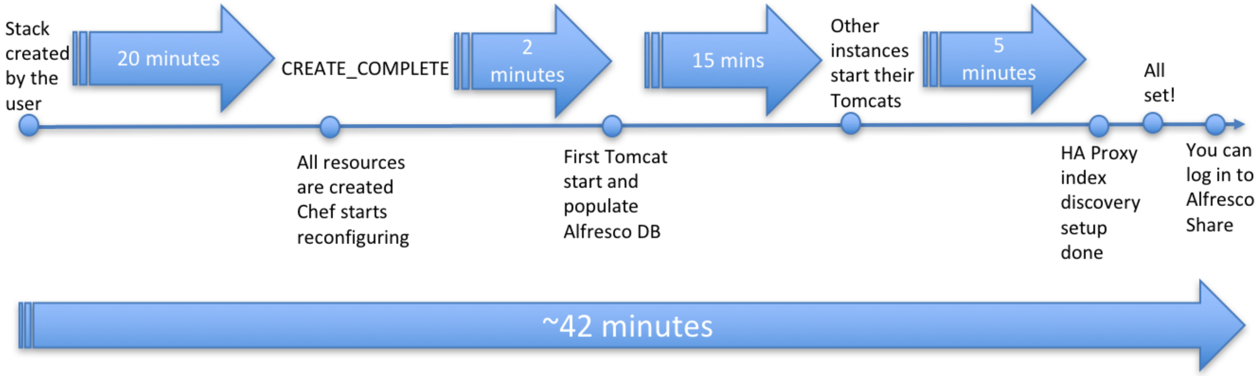

Stack created by the user
20 minutes
CREATE_COMPLETE
2 minutes
15 mins
Other instances start their Tomcats
5 minutes
All set!
All resources are created Chef starts reconfiguring
First Tomcat start and populate Alfresco DB
HA Proxy index discovery setup done
You can log in to Alfresco Share
~42 minutes

Figure 10: Deployment timeline

To test the deployment, upload a document and preview it in Alfresco Content Services. Additionally, to make sure the index is working, you may do a search of a document using the search text field in the upper-right corner of the Alfresco Share web interface.

Template Details and Customization

Template Mappings

This template uses AWS CloudFormation mappings to define some fixed parameters that can be referenced as the template is being executed. Two mappings are created:

- AWSAMIRegionMap
- AWSInfoRegionMap

The first mapping is for the AMIs that are used for the Alfresco Content Services servers, Index servers, and the NAT gateway instances. The default settings use a custom pre-baked Linux CentOS 7 AMI in the region in which the template is being launched for the Alfresco Content Services instances.

The second mapping indicates the URL for the Quick Start S3 bucket. If you are customizing the Quick Start, change the value to match the URL for the region where your custom S3 bucket that hosts the Quick Start assets is located.

Creating the Infrastructure

If you choose the option to create a new VPC, the template first creates a new VPC environment for the deployment. You'll need to choose an IP address space that the VPC will use.

The template uses the default IP range 10.0.0.0/16 and creates four subnets across two Availability Zones. Each Availability Zone has two subnets. If your existing networks have the same addresses, you can change these values in the template file.

The subnets and their contents are detailed in the following table.

Subnet type	IP range	Contents
Public	10.0.128.0/20 10.0.144.0/20	NAT gateway and bastion host instances for Availability Zone 1 and Availability Zone 2
Private	10.0.0.0/19 10.0.32.0/19	Alfresco Content Services servers, Index server, and RDS DB instances for Availability Zone 1 and Availability Zone 2

The NAT gateway and bastion host instances allow the Alfresco Content Services and Index servers to access the Internet, including the AWS API endpoints. They also serve as SSH administrative hosts. The administrative hosts allow an administrator to connect via SSH to the Alfresco Content Services and Index instances in the private subnets. Use the **RemoteAccessCIDR** parameter in the AWS CloudFormation template (see step 4) to limit the IP addresses that are permitted to connect via SSH to the bastion host instances.

Configuring the Database

Alfresco Content Services supports several different database options, including PostgreSQL, MySQL, MariaDB, Amazon Aurora, Oracle, and Microsoft SQL Server. This Quick Start uses the Amazon Aurora service, except for the EU (Frankfurt) Region, for which it uses the MySQL RDS service.

Amazon RDS provides a managed Amazon Aurora and MySQL database for Alfresco Content Services, so you don't have to install, configure, and manage the database server. To ensure high availability, the template enables the Amazon RDS Multi-AZ feature. This will deploy an RDS DB instance in both Availability Zones, and it will be referenced by using a DNS name to allow for failover to the slave instance in the event that the master fails.

Alfresco Content Services uses a database to store metadata information about objects, and the files are placed in the content store. In this case, we will use Amazon S3 to store the data. The database typically doesn't have to be very large, nor does it require a very large instance type. In this deployment, Alfresco Content Services is configured to connect through SSL to the RDS DB instances.

The default values provided in the AWS CloudFormation template create a 5 GiB database of type **db.t2.small** (1 vCPU, 2 GiB memory, with low network performance). These values are appropriate for a small to mid-sized deployment. Depending on the size of your deployment, you might need to modify these default values to increase the database size and use a larger instance type, but we recommend that you start with the default values. If you outgrow the default settings, you can easily resize your Amazon RDS database by following the steps described in the Amazon RDS documentation.

Installing Alfresco Content Services

The Quick Start installs the Alfresco Content Services software and the Index server software on EC2 instances by using the Alfresco Content Services AMI available on the AWS Marketplace. To make Alfresco Content Services work in a cluster configuration, you can use custom Chef cookbooks. These cookbooks are available from the chef-alfresco automation code, and they automate the installation and full configuration process. The configuration involves just a few user inputs via variables, which the AWS CloudFormation template passes to Chef.

After the installation has completed, you may check the Alfresco Content Services configuration files for settings for both the shared storage and clustering components, as described in the following sections.

Configuring Storage

To use Amazon S3 as your shared content store, the AWS CloudFormation template performs the following steps:

- Creates an S3 bucket.
- Installs the S3 Connector Alfresco Module Package (AMP) into Alfresco Content Services and into the Index server (via the chef-alfresco code).

The AWS CloudFormation template creates an IAM user and associated API credentials with permissions to call the Amazon S3 API commands that are necessary for the connector to function. These credentials and the bucket name are added automatically to the alfresco-global.properties file during the installation. The chef-alfresco code also configures Alfresco Content Services to connect to Amazon S3 using SSL.

Setting up the Cluster

To set up clustering for Alfresco Content Services in Amazon EC2, the chef-alfresco code modifies the Alfresco configuration files and configures Hazelcast for both Alfresco Share and the Alfresco Content Services repository. Hazelcast is an open source data distribution and clustering package.

Hazelcast has several methods to identify other nodes in a cluster. In Amazon EC2, Hazelcast must be configured to identify members based on their Amazon EC2 security group membership. To enable Hazelcast to query the AWS APIs to identify an instance's security group, the application requires a set of API keys.

The AWS CloudFormation template creates an IAM user with permissions to describe instances, allowing it to identify which instances use the specified security group. The IAM API keys, the security group that is created for the Alfresco Content Services servers, and the cluster name and password are all added to the Hazelcast configuration file by the custom Chef code.

In addition to configuring Hazelcast, chef-alfresco also installs HAProxy on the Alfresco Content Services instances to auto-discover new Index servers, and it forwards queries in failover mode to the dedicated Index servers.

It takes less than an hour to complete the steps to create the whole new architecture (depending on the region and EC2 instance type), including the installation of the Alfresco and Index servers, and to have the environment ready to accept requests.

Alfresco Content Services and Index Setup Instance

One key decision when setting up the AWS environment is to determine how much of the configuration is performed dynamically (often referred to as *bootstrapping*) and how much is preconfigured as part of the AMI. In this Quick Start, chef-alfresco configures all required software every time an instance is requested, and also updates packages automatically. Therefore, instances may connect to the Internet to download a package if needed.

Figure 11 provides the details for the Alfresco (Alfresco Content Services repository and Alfresco Share) and index tier (Alfresco Content Services repository and Solr) boxes, showing what the chef-alfresco code configures.

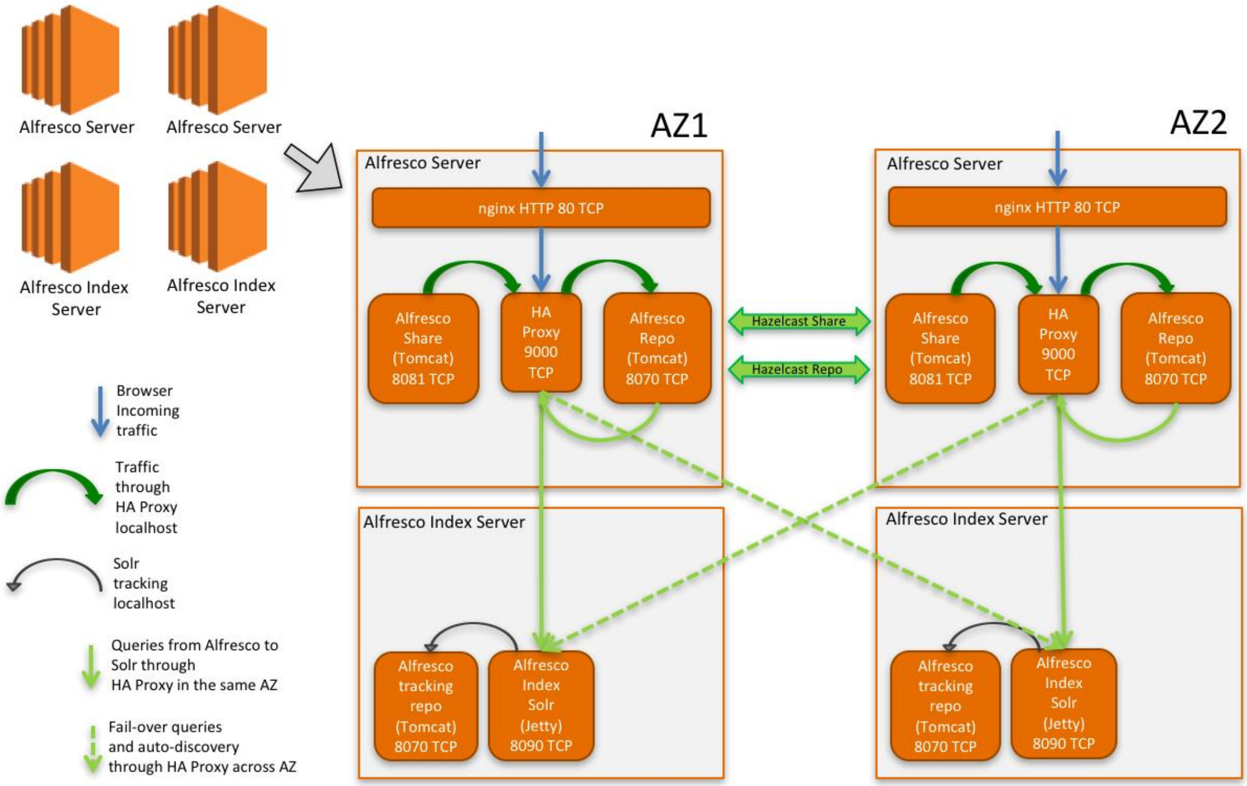

Alfresco Server
Alfresco Server
Alfresco Index Server
Alfresco Index Server
Browser Incoming traffic
Traffic through HA Proxy localhost
Solr tracking localhost
Queries from Alfresco to Solr through HA Proxy in the same AZ
Fail-over queries and auto-discovery through HA Proxy across AZ
AZ1
AZ2
Alfresco Server
nginx HTTP 80 TCP
Alfresco Share (Tomcat) 8081 TCP
HA Proxy 9000 TCP
Alfresco Repo (Tomcat) 8070 TCP
Hazelcast Share
Hazelcast Repo
Alfresco Index Server
Alfresco tracking repo (Tomcat) 8070 TCP
Alfresco Index Solr (Jetty) 8090 TCP

Figure 11: Alfresco Content Services and Index server details

The diagram describes only the Alfresco Content Services and Index servers with all their components, ports, and configuration. In this particular case, NGINX is in front of HAProxy to provide enhanced connectivity to the client via HTTP/2 only if SSL is enabled. Share communicates with the Alfresco Content Services service via a local HAProxy service on the box. This allows intelligent routing across all Alfresco services in the Alfresco Content Services tier. This configuration uses the local Alfresco Content Services service as the main route, and the other Alfresco Content Services tier's services listed may be used as backups.

For scale-up and scale-down activities, the HAProxy configuration is also under Chef control and is checked every five minutes for any connected services that have scaled up or down (like new Index servers per the Auto Scaling policy), and automatically adds or removes those services from the routing for Alfresco Content Services.

The Auto Scaling resource describes everything that is needed to launch the Alfresco Content Services instance and configuration, such as the AMI ID to use, the Amazon EC2 key pair, and the IAM role to associate with the instance. Chef is also installed, and the system is prepared for an Alfresco Content Services configuration using the `chef-client` command.

The Quick Start builds a few bash scripts dynamically. These scripts set up Alfresco Content Services and update configuration files with both user-provided parameters and dynamically created API credentials. The bash scripts are executed at run time, because many of the objects that the scripts reference do not exist or are unknown until after the AWS CloudFormation template begins to execute.

Instances are deployed in both Availability Zone 1 and Availability Zone 2. The template gives them 600 seconds (10 minutes) to start and to be up and running. In `AlfrescoAutoScalingGroup`, the minimum size for the Alfresco Content Services cluster is set by the **AlfrescoNodesMinSize** parameter, and is 2 nodes by default. The maximum size for the Alfresco Content Services cluster is 4 nodes if the scale-up action is triggered, and that value is controlled by the **AlfrescoNodesMaxSize** parameter. Also, a message is sent to the email address you specify in the **OperatorEmail** parameter as a notification of any scale-up or scale-down operation. For the Index servers, `IndexAutoScalingGroup` uses the same configuration.

The Alfresco Content Services scale-up and scale-down policies scale the environment by the amount of one server (`ScalingAdjustment`) each time the required threshold is met. The Auto Scaling policy provides a period of 300 seconds (5 minutes) for the cooldown period. The Auto Scaling cooldown is a configurable setting (`Cooldown`) that determines when Auto Scaling should suspend scaling activities for a specific Auto Scaling group. This cooldown period is important because it helps ensure that resources are not launched or terminated before the effects of previous scaling activities are visible.

`CPUAlarmHigh` and `CPUAlarmLow` measure the CPU usage to decide when to scale up or down: in this case, they scale up if CPU usage is above 60% for 5 minutes and scale down if CPU usage is below 40% for 30 minutes.

In a production environment, you might want to modify scale-up and scale-down policies and the CPU alarm to accommodate your needs. These settings might also be determined by your Alfresco Content Services subscription license; contact to your Alfresco Sales representative if needed.

Troubleshooting

If you run into any problems during the deployment, you must log in to the servers and view Alfresco Content Services configuration files and logs.

Logging in to the Servers with SSH

Once all the systems are up and running, you can use the bastion hosts to access the Alfresco Content Services or Index server.

In the Amazon EC2 console, select one of the two bastion hosts and note its public IP address. Then, using SSH agent forwarding, connect to the bastion instance through SSH:

```
$ ssh -A ec2-user@BASTION-PUBLIC-IP
```

Now, from the bastion host, connect to any Alfresco Content Services or Index server, using `centos` as the user for all regions:

```
$ ssh -A centos@ALFRESCO-OR-INDEX-SERVER-IP
```

To work as root, type `sudo bash`.

For more information about SSH agent forwarding, see the AWS Security Blog.

Starting or Stopping Services

When you are logged in to the servers as root user, you can stop, start, or restart the Alfresco-related services on the Alfresco Content Services and Index servers as follows:

```
$ systemctl {start,stop,restart} tomcat-alfresco
$ systemctl {start,stop,restart} tomcat-share
$ service solr {start,stop,restart}
$ systemctl {start,stop,restart} haproxy
$ systemctl {start,stop,restart} nginx
```

Note On the Index servers, only `solr` and `tomcat-alfresco` are available.

Configuration and Logs

The chef-alfresco script handles the configuration for the Alfresco Content Services server; refer to the code for any changes.

This script is launched using cfn-init when the instance is bootstrapped. If you encounter a configuration issue during bootstrapping, see the `/var/log/cloud-init.log` file for information.

Logs for Alfresco Content Services and Share on the Alfresco Content Services servers are in `/var/log/tomcat-{share,alfresco}`.

The chef-alfresco script also handles the configuration for the Index servers; refer to the code for any changes. Logs for Alfresco Content Services (for index tracking) and Solr on the Index servers are in `/var/log/tomcat-alfresco` and `/var/log/solr`.

The following table lists additional configuration files and logs that are generated as part of this deployment:

Files	Location
Alfresco Content Services configuration files	`/usr/share/tomcat/shared/classes/alfresco-global.properties` `/usr/share/tomcat/shared/classes/alfresco/extension/`
Alfresco Share configuration files	`/usr/share/tomcat/shared/classes/alfresco/web-extension/`
JAVA_OPTS configuration variable	`/etc/sysconfig/tomcat-alfresco` `/etc/sysconfig/tomcat-share`
server.xml for Tomcat servers	`/etc/tomcat-alfresco/server.xml` `/etc/tomcat-share/server.xml`

Common files for Tomcat servers	`/etc/tomcat/catalina.policy`
	`/etc/tomcat/catalina.properties`
	`/etc/tomcat/context.xml`
	`/etc/tomcat/tomcat-users.xml`
	`/etc/tomcat/web.xml`

Solr configuration and default location	`/usr/share/tomcat/alf_data/`
	`/var/solr/data/alfresco/conf/solrcore.properties`
	`/var/solr/data/archive/conf/solrcore.properties`
	`/var/solr/contentstore/` (gzipped content)
	`/var/solr/data/conf/shared.properties`
	`/var/solr/data/alfresco/index/`
	`/var/solr/data/archive/index/`
	`/opt/alfresco-search-services/solr/server/etc/` (jetty)
	`/usr/share/tomcat/alf_data/solr6Backup/`

Logging properties for Tomcat servers and Jetty (solr6)	`/usr/share/tomcat-alfresco/conf/logging.properties` `/usr/share/tomcat-share/conf/logging.properties`

Files	Location
	`/var/solr/log4j.properties`

NGINX configuration files, logs, start/stop scripts	`/etc/nginx/nginx.conf`
	`/var/log/nginx/`
	`systemctl {start,stop,restart,status} nginx`

HAProxy configuration files, logs, start/stop scripts	`/etc/haproxy/haproxy.cfg`
	`/var/log/haproxy/`
	`systemctl {start,stop,restart,status} haproxy`

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. We do not require or recommend storing SSH keys, secret keys, or access keys on the provisioned instances.

The AWS CloudFormation template creates one IAM user and one IAM role. The IAM user is used by the Amazon S3 connector and Hazelcast (neither supports IAM roles). The IAM role is used by the template to launch all instances. The deployment also uses these two roles:

- Setup role: This role used to bootstrap and configure the instances.
- IAM Alfresco user policy: This role is used by the instances to get access to AWS 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 three security groups:

- Elastic Load Balancing
- Alfresco Content Services and Index servers
- NAT gateways and bastion hosts

Elastic Load Balancing Security Group

Direction	Source or destination	Protocol/port	Description
Inbound	0.0.0.0/0	TCP/80	Allows inbound HTTP requests to the load balancer.
Inbound	0.0.0.0/0	TCP/443	Allows inbound Alfresco Content Services traffic on port 443.
Outbound	0.0.0.0/0	All	Allows all outbound traffic.

Alfresco Security Group (Applies to Alfresco Content Services and Index Servers)

Direction	Source or destination	Protocol/port	Description
Inbound	ELB load balancer	TCP/80	Allows inbound HTTP requests from the load balancer.
Inbound	ELB load balancer	TCP/443	Allows inbound HTTPS requests from the load balancer.
Inbound	Alfresco instances	TCP/5701	Allows Hazelcast traffic between cluster nodes.
Inbound	Alfresco instances	TCP/8090	Allows access to Solr port from cluster nodes.
Inbound	Bastion instance	TCP/22	Allows SSH only from the bastion instance.
Outbound	0.0.0.0/0	All	Allows outbound access.

NAT Gateways and Bastion Instance Security Group

Direction	Source or destination	Protocol/port	Description
Inbound	SSH from parameter	TCP/22	Allows SSH from IP range specified.
Outbound	0.0.0.0/0	All	Allows all outbound traffic.

Additional Resources

AWS services

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

Alfresco Content Services

- Alfresco on AWS
  http://www.alfresco.com/aws
- AWS Alfresco Partner Page
  http://www.aws-partner-directory.com/PartnerDirectory/PartnerDetail?id=7609
- Alfresco Content Services 5.2 documentation
  http://docs.alfresco.com/5.2/concepts/welcome.html

- Alfresco Backup and Disaster Recovery
  http://www.slideshare.net/toniblyx/alfresco-backup-and-disaster-recovery-white-paper

Quick Start Reference Deployments

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

Send Us Feedback

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

Document Revisions

Date	Change	In sections
September 2017	New licensing information and links	Throughout guide
April 2017	Product name change and new AMI with product version 5.2 and Solr6 as index servers. Also added Amazon Aurora support for all regions except for EU (Frankfurt).	Templates and throughout guide
January 2017	Initial publication	—