

MongoDB on the AWS Cloud

Quick Start Reference Deployment

Vinod Shukla

Solutions Architect, AWS Quick Start Team

April 2015

([last update](#): May 2017)

This guide is also available in HTML format at
<https://docs.aws.amazon.com/quickstart/latest/mongodb/>.



Contents

About This Guide	3
Quick Links	3
About Quick Starts	3
Overview	4
MongoDB on AWS	4
Cost and Licenses	4
AWS Services.....	4
Architecture	5
MongoDB Constructs.....	7
Performance Considerations	9
Deployment Options	10
Deployment Steps	10
Step 1. Prepare an AWS Account	11
Step 2. Launch the Quick Start	13
Step 3. Connect to MongoDB Nodes	20
Testing MongoDB.....	21
Backing Up Your Data	22
Security	22
AWS Identity and Access Management (IAM).....	22
OS Security	22
Network Security.....	23
Security Groups.....	23
Database Security.....	24
Additional Resources	24
Send Us Feedback	26
Document Revisions.....	27

About This Guide

This Quick Start reference deployment guide includes architectural considerations and configuration steps for deploying a MongoDB cluster on the Amazon Web Services (AWS) Cloud. It discusses best practices for deploying MongoDB on AWS using services such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Virtual Private Cloud (Amazon VPC). It also provides links to automated [AWS CloudFormation](#) templates that you can leverage for your deployment or launch directly into your AWS account.

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

Quick Links

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

- If you have an AWS account, and you're already familiar with AWS services and MongoDB, you can launch the Quick Start to deploy MongoDB into a new or existing virtual private cloud (VPC) in your AWS account. The deployment takes approximately 15 minutes. If you're new to AWS or MongoDB, please review the implementation details and follow the [step-by-step instructions](#) provided later 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. You can customize each template during launch, or download and extend it for other projects.

View template
(for new VPC)

View template
(for existing VPC)

About Quick Starts

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

Overview

MongoDB on AWS

MongoDB is an open source, NoSQL database that provides support for JSON-styled, document-oriented storage systems. It supports a flexible data model that enables you to store data of any structure, and provides a rich set of features, including full index support, sharding, and replication.

AWS enables you to set up the infrastructure to support MongoDB deployment in a flexible, scalable, and cost-effective manner on the AWS Cloud. This reference deployment will help you build a MongoDB cluster by automating configuration and deployment tasks.

This Quick Start supports a self-service deployment of the MongoDB replica set cluster (version 3.2 or 3.4) on AWS.

Cost and Licenses

This deployment launches MongoDB automatically into a configuration of your choice. You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start. The cost varies depending on the storage and compute configuration of the cluster you deploy. See the pricing pages for each AWS service you will be using for full details.

This Quick Start deploys MongoDB Community Edition version 3.2 or 3.4, which is open-source software distributed under the [GNU Affero General Public License version 3](#).

AWS Services

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

- [Amazon EC2](#) – The Amazon Elastic Compute Cloud (Amazon EC2) service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- [Amazon VPC](#) – The Amazon Virtual Private Cloud (Amazon VPC) service lets you provision a private, isolated section of the AWS Cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

- [Amazon Elastic Block Store \(EBS\)](#) - Amazon Elastic Block Store (Amazon EBS) provides persistent block level storage volumes for use with EC2 instances in the AWS Cloud. Each EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. EBS volumes offer the consistent and low-latency performance needed to run your workloads.
- [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., EC2 instances) that you want. You don't have to individually create and configure the resources or figure out dependencies; AWS CloudFormation handles all of that.
- [IAM](#) – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can manage users, security credentials such as access keys, and permissions that control which AWS resources users can access, from a central location.

Architecture

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

Deploying this Quick Start for a new VPC with **default parameters** builds the following MongoDB environment in the AWS Cloud.

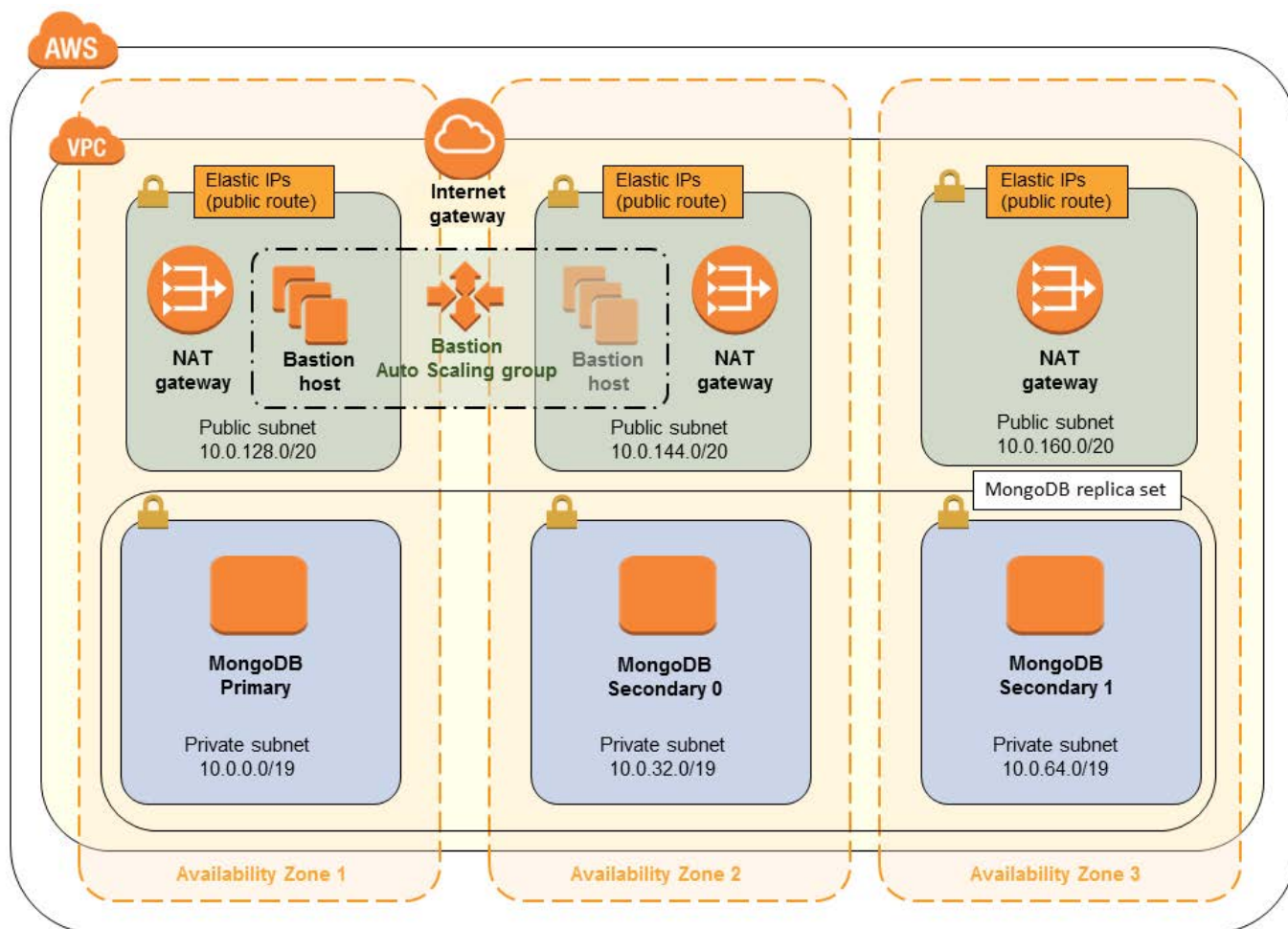


Figure 1: Quick Start architecture for MongoDB on AWS

The following AWS components are deployed and configured as part of this reference deployment:

- A VPC configured with public and private subnets across three Availability Zones.*
- In the public subnets, NAT gateways to allow outbound Internet connectivity for resources (MongoDB instances) in the private subnets. (For more information, see the [Amazon VPC Quick Start](#).)*
- In the public subnets, bastion hosts in an Auto Scaling group with Elastic IP addresses to allow inbound Secure Shell (SSH) access. One bastion host is deployed by default, but this number is configurable. (For more information, see the [Linux Bastion Quick Start](#).)*

- An AWS Identity and Access Management (IAM) instance role with fine-grained permissions for access to AWS services necessary for the deployment process.
- Security groups to enable communication within the VPC and to restrict access to only necessary protocols and ports.
- In the private subnets, a customizable MongoDB cluster with the option of running standalone or in replica sets, along with customizable Amazon EBS storage. The Quick Start launches each member of the replica set in a different Availability Zone. However, if you choose an AWS Region that doesn't provide three or more Availability Zones, the Quick Start reuses one of the zones to create the third subnet.

* You can choose to launch the Quick Start for a new VPC or use your existing VPC. The template that deploys the Quick Start into an existing VPC skips the creation of components marked by asterisks and prompts you for your existing configuration.

The Quick Start launches all the MongoDB-related nodes in the private subnet, so the nodes are accessed by using SSH to connect to the bastion hosts. Instead of using a remote access CIDR for each MongoDB instance, the deployment requires a security group ID of the bastion hosts so remote access can be centrally controlled. If you launch the Quick Start for a new VPC, the bastion security group is created for you. If you launch the Quick Start in an existing VPC, you must create a security group for your bastion hosts or use one that already exists.

MongoDB Constructs

Here are some of the building blocks that are used in this reference deployment.

Replica set. Refers to a group of [mongod instances](#) that hold the same data. The purpose of replication is to ensure high availability, in case one of the servers goes down. This reference deployment supports one or three replica sets. In the case of three replica sets, the reference deployment launches three servers in three different [Availability Zones](#) (if the region supports it). In production clusters, we recommend using three replica sets (*Primary, Secondary0, Secondary1*).

All clients typically interact with the primary node for read and write operations. It is possible to choose a secondary node as a preference during read operations, but write operations always go to the primary node and get replicated asynchronously in the secondary nodes. If you choose a secondary node for read operations, watch out for stale data, because the secondary node may not be in sync with the primary node. For more information about how read operations are routed in a replica set, see the [MongoDB documentation](#).

In a development environment, you can start with a single replica set and move to three replica sets during production. Figure 2 shows the MongoDB reference deployment with a replication factor of 3.

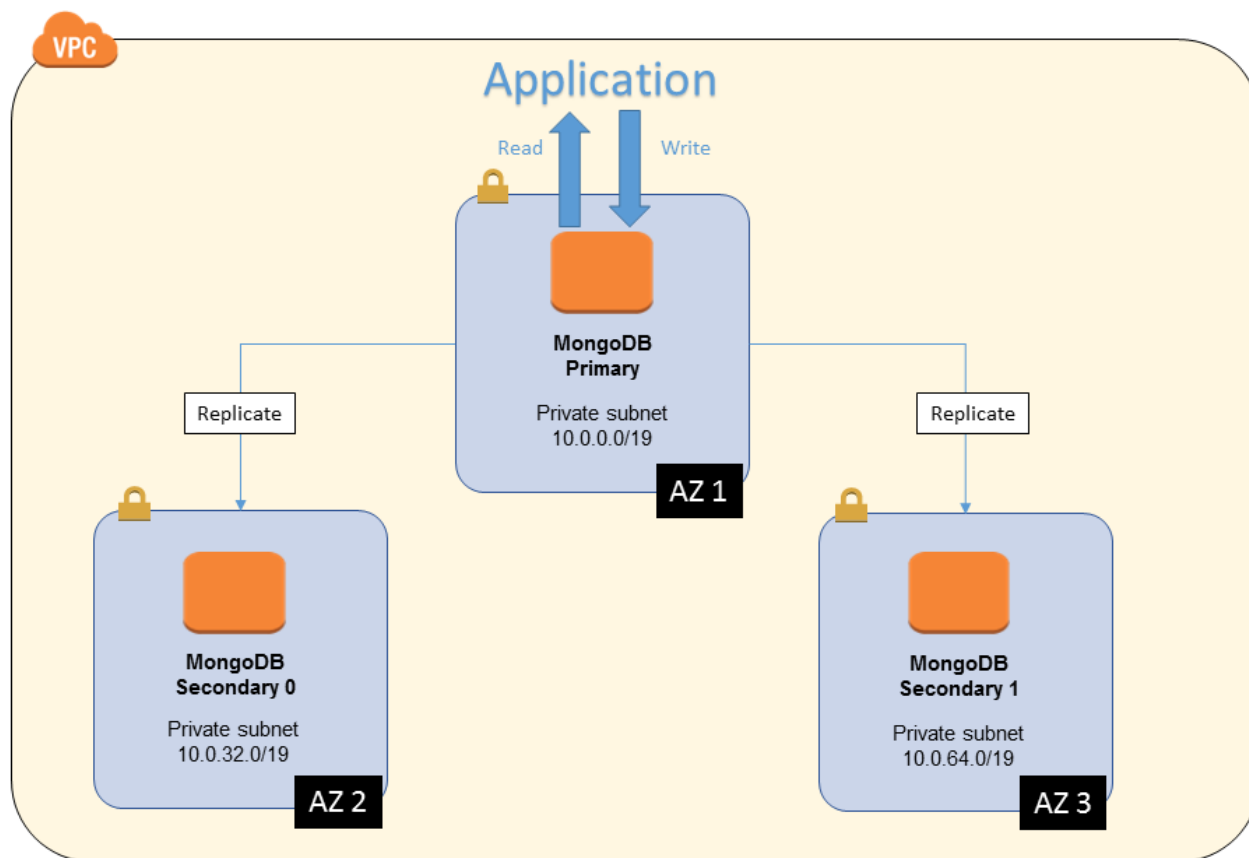


Figure 2: MongoDB cluster on AWS with three replica sets

When a primary instance fails, one of the secondary instances from another Availability Zone becomes the new primary node, thereby guaranteeing automatic failover.

Sharding. Refers to distribution of data across multiple nodes. Storing distinct data across multiple nodes provides horizontal scalability for read and write performance. When you have a large data set, a single node could be bottlenecked by CPU or I/O performance. Sharding resolves this bottleneck by reducing the number of operations each shard node handles, and improves overall cluster performance. This Quick Start doesn't provide direct support for creating shards. Instead, it provides a parameter (**ReplicaShardIndex**) to enable joining the launched replica sets to a sharded cluster. See the [MongoDB documentation](#) for details.

Performance Considerations

The reference implementation offers various compute and storage choices. The following table shows some of the compute choices to consider.

Instance Type	vCPU	Memory (GiB)	Workload Type
c3.4xlarge	16	55	Compute-optimized
c3.8xlarge	32	60	Compute-optimized
c4.8xlarge	36	60	Compute-optimized
r3.4xlarge	16	122	Memory-optimized
r3.2xlarge	8	61	Memory-optimized
r3.8xlarge	32	244	Memory-optimized

As a general guideline, consider growing instances horizontally instead of vertically. Horizontal scaling overcomes the limitations of single nodes and avoids single points of failure, and can potentially increase the overall throughput of your cluster.

For storage, depending on your database requirement, you may choose to change the storage volume to be attached to each node. Amazon EBS provides three volume types: General Purpose (SSD) volumes, Provisioned IOPS (SSD) volumes, and Magnetic volumes. These differ in performance characteristics and cost, so you can choose the right storage performance and price depending on the needs of your application. All Amazon EBS volume types offer the same durable snapshot capabilities and are designed for 99.999% availability. This reference deployment supports General Purpose and Provisioned IOPS storage volumes.

The following table shows some of the performance characteristics of each storage type. Depending on your performance requirements, you may want to benchmark your application before deciding on the storage type and Amazon EBS Provisioned IOPS capacity (if chosen).

Volume Type	General Purpose (SSD)	Provisioned IOPS (SSD)
Storage media	SSD-backed	SSD-backed
Maximum volume size	16 TiB	16 TiB
Maximum IOPS/volume	10,000	20,000

Deployment Options

This Quick Start provides two deployment options:

- **Deploy MongoDB 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 hosts, and other infrastructure components, and then deploys MongoDB into this new VPC.
- **Deploy MongoDB into an existing VPC**. This option provisions MongoDB in your existing AWS infrastructure.

The Quick Start provides separate templates for these options. It also lets you configure additional settings such as CIDR blocks, instance types, and MongoDB settings, as discussed later in this guide.

Deployment Steps

You can deploy MongoDB easily on the flexible AWS platform. This guide serves as a reference for customers who want to set up a fully customizable MongoDB cluster on demand. Building a scalable, on-demand infrastructure on AWS provides a cost-effective solution for handling large-scale compute and storage requirements. The flexible AWS architecture allows you to choose the most appropriate network, compute, and storage infrastructure for your environment.

The procedure for deploying MongoDB 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, choosing a region, creating a key pair, and requesting increases for account limits, if necessary.

[Step 2. Launch the Quick Start](#)

Launch the AWS CloudFormation template into your AWS account, specify parameter values, and create the stack. The Quick Start provides separate templates for end-to-end deployment and deployment into an existing VPC.

[Step 3. Connect to MongoDB nodes](#)

You use SSH to connect to MongoDB nodes via the bastion hosts, because the nodes are in a private subnet.

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.
2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy the MongoDB cluster on AWS. For more information, see [Regions and Availability Zones](#). Regions are dispersed and located in separate geographic areas. Each Region includes at least two Availability Zones that are isolated from one another but connected through low-latency links.

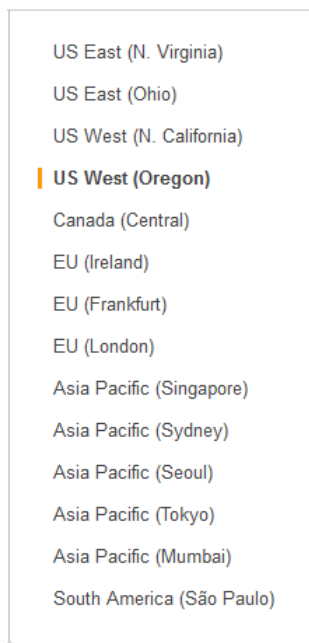


Figure 3: Choosing an AWS Region

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**.

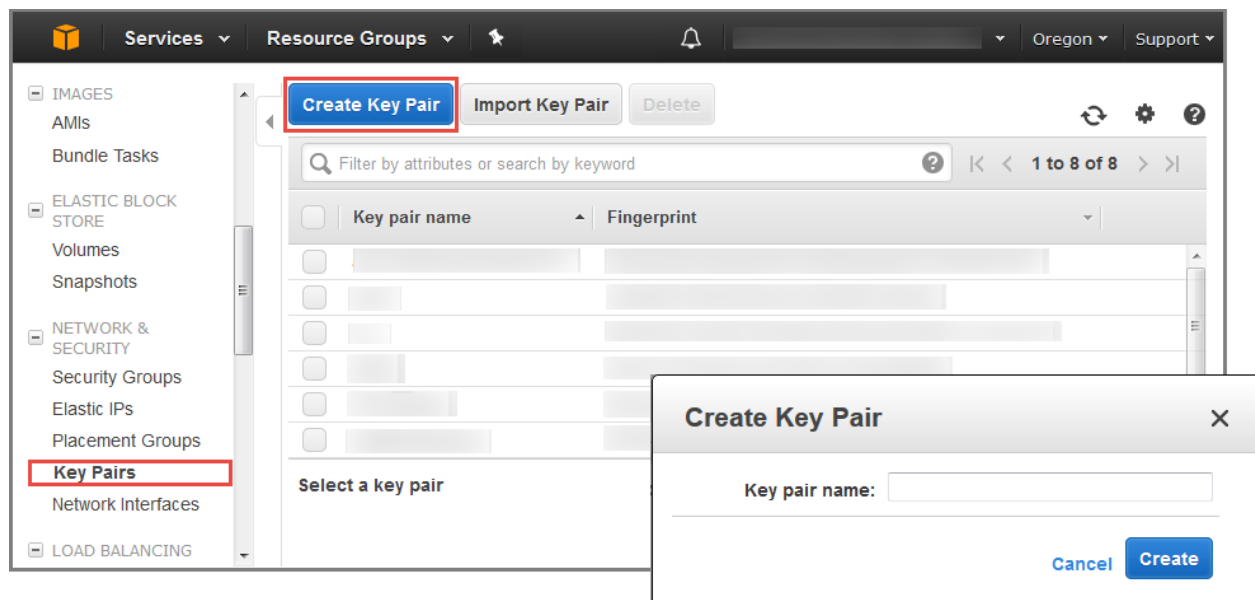


Figure 4: Creating a key pair

Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. On Linux, we use the key pair to authenticate SSH login.

4. If necessary, [request a service limit increase](#) for the EC2 instance types that you intend to deploy. To do this, in the AWS Support Center, choose **Create Case, Service Limit Increase, EC2 instances**, and then complete the fields in the limit increase form.

The default limit for the number of instances depends on the instance type you choose and currently ranges from 2 to 20 (see the [Amazon EC2 FAQ page](#)). If you have existing deployments that also use this instance type, or if you plan to exceed this default with this reference deployment, you will need to request a limit increase. It might take a few days for the new service limit to become effective. For more information, see [Amazon EC2 Service Limits](#) in the AWS documentation.

The screenshot shows the AWS Support Center interface. On the left, the 'Create Case' link is highlighted with a red box. The main area is titled 'Create Case' and shows the 'Basic Support Plan'. The 'Regarding*' section has three options: 'Account and Billing Support', 'Service Limit Increase' (which is selected and highlighted with a red box), and 'Technical Support'. The 'Limit Type*' dropdown is set to 'EC2 Instances' and is also highlighted with a red box. Below this, the 'Request 1' section contains the following fields: 'Region*' (US East (Ohio)), 'Primary Instance' (c4.8xlarge), 'Type*' (Instance Limit), and 'New limit value*' (25).

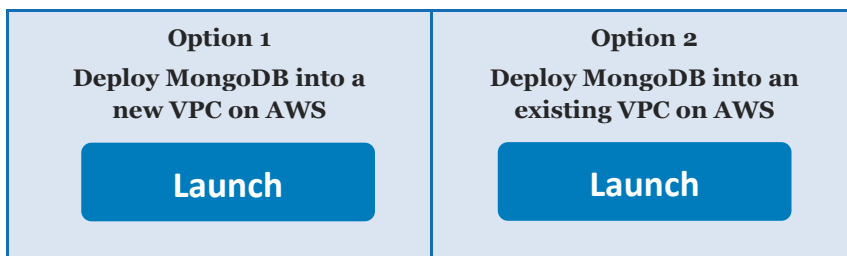
Figure 5: Requesting a service limit increase

5. If necessary, request a limit increase for the Elastic IP addresses in the VPC. Choose **VPC** for the limit type, and complete the fields on the limit request form.
6. If necessary, request a limit increase for the EBS volumes that you can use. Choose **EBS** for the limit type, and complete the fields on the limit request form.

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 full details, see the pricing pages for each AWS service you will be using in this Quick Start. Prices are subject to change.

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.



Important If you're deploying MongoDB into an existing VPC, make sure that your VPC is set up with two public subnets and three private subnets in different Availability Zones. 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.

The private subnets require NAT gateways or NAT instances in their route tables for outbound Internet connectivity, and you must create bastion hosts and their associated security group for inbound SSH access. (To set up your VPC, you can use the [Amazon VPC Quick Start](#). To set up bastion hosts, see the [Linux bastion host Quick Start](#). If you deploy into a new VPC, the Quick Start will set these up for you automatically.)

Each deployment takes about 15 minutes to complete.

2. Check the region that's displayed in the upper-right corner of the navigation bar, and change it if necessary. The template is launched in the US East (N. Virginia) region by default.
3. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.
4. On the **Specify Details** page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you finish reviewing and customizing the parameters, choose **Next**.

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

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

- **Option 1: Parameters for deploying MongoDB 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 preserves the logical order you specify. This deployment uses 2 or 3 Availability Zones. After you make your selections, make sure that the Number of Availability Zones parameter matches your selections.
Number of Availability Zones (NumberOfAZs)	<i>Requires input</i>	The number of Availability Zones (2 or 3) to use in the VPC. This must match your selections in the Availability Zones parameter; otherwise, deployment will fail with an AWS CloudFormation template validation error. (Note that some regions provide only one or two Availability Zones.)
VPC CIDR (VPCCIDR)	10.0.0.0/16	CIDR block for the VPC to create.
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.
Private Subnet 3 CIDR (PrivateSubnet2CIDR)	10.0.64.0/19	CIDR block for the private subnet located in Availability Zone 2.
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.
Public Subnet 3 CIDR (PublicSubnet3CIDR)	10.0.160.0/20	CIDR block for the public (DMZ) subnet located in Availability Zone 2.
Allowed Bastion External Access CIDR (RemoteAccessCIDR)	<i>Requires input</i>	The CIDR IP range that is permitted external SSH access to the bastion hosts. 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.

Security Configuration:

Parameter label (name)	Default	Description
Key Name (KeyPairName)	<i>Requires input</i>	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.

Linux Bastion Configuration:

Parameter label (name)	Default	Description
Bastion AMI Operating System (BastionAMIOS)	Amazon-Linux-HVM	The Linux distribution for the AMI to be used for the bastion host instances. You can choose Amazon Linux, CentOS, or Ubuntu Server. If you choose CentOS, make sure that you have a subscription to the CentOS AMI in AWS Marketplace .
Bastion Instance Type (BastionInstanceType)	t2.micro	EC2 instance type for the bastion host instances.
Number of Bastion Hosts (NumBastionHosts)	1	The number of bastion hosts to create (1-4). Auto Scaling will ensure that you always have this number of bastion hosts running.

MongoDB Database Configuration:

Parameter label (name)	Default	Description
Cluster Replica Set Count (ClusterReplicaSetCount)	1	Number of replica sets. Choose 1 or 3.
IOPS (Iops)	100	IOPS of the EBS volume when the io1 volume type is chosen. Otherwise, this setting is ignored.
MongoDB Version (MongoDBVersion)	3.4	The version of MongoDB that will be deployed. You can choose version 3.2 or 3.4.
MongoDB Admin Username (MongoDBAdminUsername)	admin	The user name for the MongoDB administrative account.
MongoDB Admin Password (MongoDBAdminPassword)	<i>Requires input</i>	Your MongoDB database password. You can enter an 8-32 character string consisting of the characters: [A-Za-z0-9_@-].
Node Instance Type (NodeInstanceType)	m4.large	EC2 instance type for the MongoDB nodes.
Replica Shard Index (ReplicaShardIndex)	0	Shard index of this replica set. For information about shard indexes, see the MongoDB documentation .
Volume Size (VolumeSize)	400	Size of the Amazon EBS (data) volume to be attached to the MongoDB node, in GiBs.
Volume Type (VolumeType)	gp2	Type of the Amazon EBS (data) volume to be attached to the MongoDB node (io1 or gp2).

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	quickstart-reference	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)	mongodb/latest/	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. It should not start or end with a hyphen (-).

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

[View template](#)

Network Configuration:

Parameter label (name)	Default	Description
VPC (VPC)	<i>Requires input</i>	ID of your existing VPC (e.g., vpc-0343606e) where you want to deploy the MongoDB cluster.
Primary Node Subnet (PrimaryNodeSubnet)	<i>Requires input</i>	ID of the existing subnet (e.g., subnet-a0246dcd) in your VPC where you want to deploy the primary MongoDB node(s).
Secondary0 Node Subnet (Secondary0NodeSubnet)	<i>Requires input</i>	ID of the existing subnet in your VPC where you want to deploy the first secondary MongoDB node(s) in the replica set. For more information on expected placement, see the Architecture section.
Secondary1 Node Subnet (Secondary1NodeSubnet)	<i>Requires input</i>	ID of the existing subnet in your VPC where you want to deploy the second secondary MongoDB node(s) in the replica set. For more information on expected placement, see the Architecture section.
Bastion Security Group ID (BastionSecurityGroupID)	<i>Requires input</i>	ID of the bastion security group in your existing VPC (e.g., sg-7f16e910).

Security Configuration:

Parameter label (name)	Default	Description
Key Name (KeyPairName)	<i>Requires input</i>	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.

MongoDB Database Configuration:

Parameter label (name)	Default	Description
Cluster Replica Set Count (ClusterReplicaSetCount)	1	Number of replica sets. Choose 1 or 3.
IOPS (Iops)	100	IOPS of the EBS volume when the io1 volume type is chosen. Otherwise, this setting is ignored.
MongoDB Version (MongoDBVersion)	3.4	The version of MongoDB that will be deployed. You can choose version 3.2 or 3.4.
MongoDB Admin Username (MongoDBAdminUsername)	admin	The user name for the MongoDB administrative account.
MongoDB Admin Password (MongoDBAdminPassword)	<i>Requires input</i>	Your MongoDB database password. You can enter an 8-32 character string consisting of the characters: [A-Za-z0-9_@-].
Node Instance Type (NodeInstanceType)	m4.large	EC2 instance type for the MongoDB nodes.
Replica Shard Index (ReplicaShardIndex)	0	Shard index of this replica set. For information about shard indexes, see the MongoDB documentation .
Volume Size (VolumeSize)	400	Size of the Amazon EBS (data) volume to be attached to the MongoDB node, in GiBs.
Volume Type (VolumeType)	gp2	Type of the Amazon EBS (data) volume to be attached to the MongoDB node (io1 or gp2).

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	quickstart-reference	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.

Parameter label (name)	Default	Description
Quick Start S3 Key Prefix (QSS3KeyPrefix)	mongodb/latest/	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. It should not start or end with a hyphen (-).

- 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**.
- 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.
- Choose **Create** to deploy the stack.
- Monitor the status of the stack. When the status is **CREATE_COMPLETE**, as shown in Figure 6, the MongoDB cluster is ready.

The screenshot shows the AWS CloudFormation console. At the top, there are buttons for 'Create Stack', 'Update Stack', and 'Delete Stack'. Below these, a filter is set to 'Complete' and 'By Name' is empty. A table lists the stacks, with one stack named 'AWS-MongoDB-Infrastructure' having a status of 'CREATE_COMPLETE'. Below this, a detailed view of the stack is shown, including a table of resources. The resources table has columns for 'Resource Name', 'Physical ID', 'Type', and 'Status'. All resources listed have a status of 'CREATE_COMPLETE'.

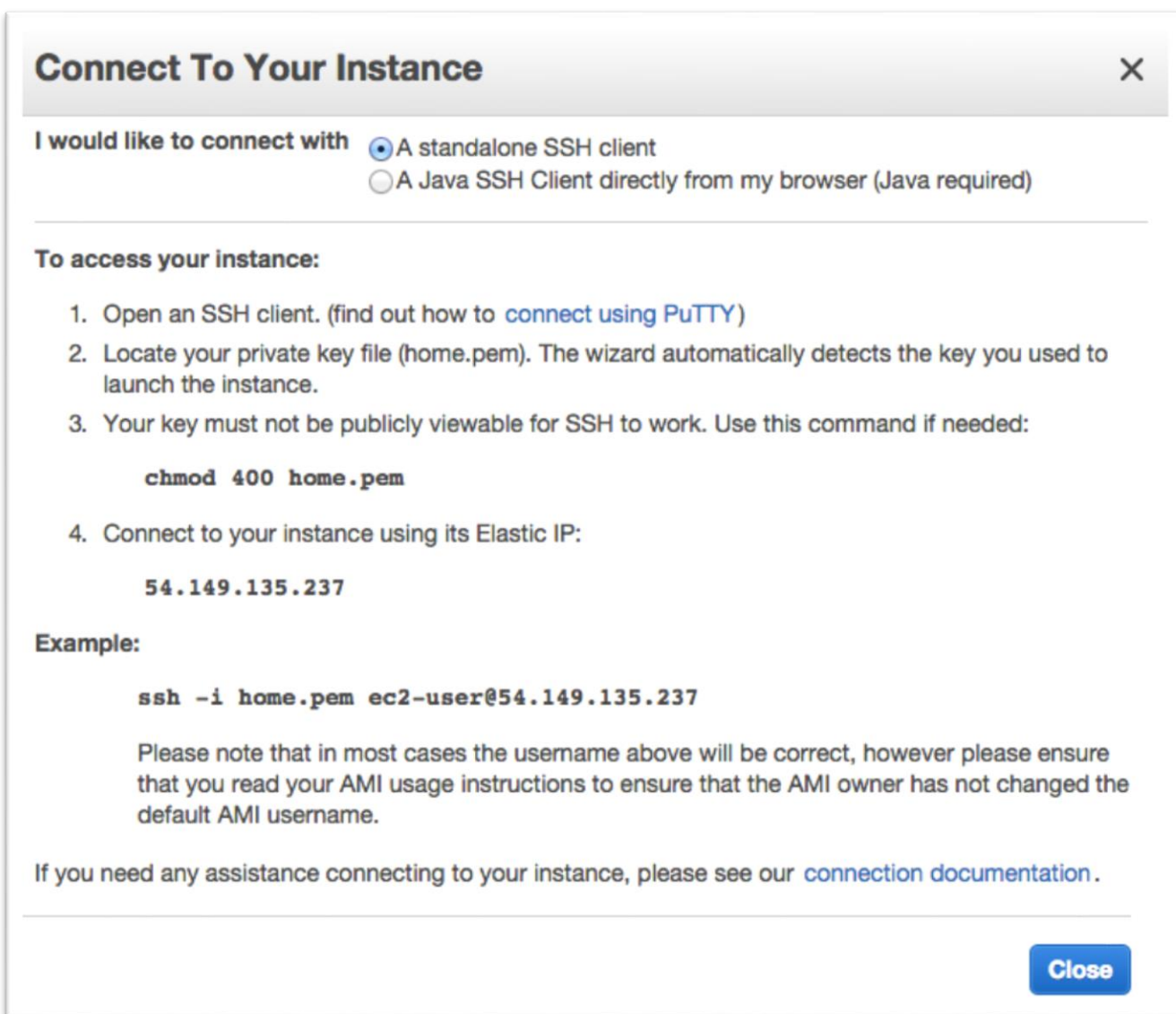
Stack Name	Created Time	Status	Description
✓ AWS-MongoDB-Infrastructure	2015-03-30 15:19:21 UTC-0700	CREATE_COMPLETE	(000F) Deploy MongoDB on AWS

Resource Name	Physical ID	Type	Status
PrimaryReplicaNode10...	i-2a7ff8fd	AWS::EC2::Instance	CREATE_COMPLETE
SecondaryReplicaNode...	i-eb21d316	AWS::EC2::Instance	CREATE_COMPLETE
ConfigServer2NodeInst...	i-d521d328	AWS::EC2::Instance	CREATE_COMPLETE
SecondaryReplicaNode...	i-cb21d336	AWS::EC2::Instance	CREATE_COMPLETE
ConfigServer1NodeInst...	i-cc21d331	AWS::EC2::Instance	CREATE_COMPLETE
ConfigServer0NodeInst...	i-ca7ef91d	AWS::EC2::Instance	CREATE_COMPLETE
PrimaryReplicaNode00...	i-c77ef910	AWS::EC2::Instance	CREATE_COMPLETE
SecondaryReplicaNode...	i-c921d334	AWS::EC2::Instance	CREATE_COMPLETE
SecondaryReplicaNode...	i-d321d32e	AWS::EC2::Instance	CREATE_COMPLETE

Figure 6: Successful creation of the MongoDB cluster

Step 3. Connect to MongoDB Nodes

Once the AWS CloudFormation template has successfully created the stack, all the MongoDB nodes will be running with the software installed in your AWS account. To connect to any of the MongoDB nodes, use SSH to connect to the bastion host instance. In the Amazon EC2 console, choose the instance, and then choose **Connect**.



Connect To Your Instance ✕

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

To access your instance:

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

```
chmod 400 home.pem
```
4. Connect to your instance using its Elastic IP:

```
54.149.135.237
```

Example:

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

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

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

Figure 7: Connecting to a MongoDB node

Once you connect to the bastion host instance by using SSH, you can connect to any of the MongoDB nodes in a similar fashion (choose the node, and then choose **Connect** to find the SSH command).

Important You need the private key (.pem) file to connect to MongoDB nodes. Copy the private key (.pem) file into the bastion host instance; for example:

```
scp -i mykey.pem mykey.pem ec2-user@Bastion-public-ip:/home/ec2-user/mykey.pem
```

Note that all the MongoDB nodes are launched with an IAM role that grants them privileges to create and delete Amazon DynamoDB tables, to access Amazon Simple Storage Service (Amazon S3), to create and delete Amazon EC2 instances, and so on. You can modify the policy by using the IAM console. For details about the benefits of IAM roles, see [Using IAM Roles to Delegate Permissions to Applications that Run on Amazon EC2](#) in the AWS documentation.

Testing MongoDB

After the AWS CloudFormation template has completed successfully, the system will have a *mongod* instance running on each of the primary replica set nodes. To validate the system and verify the configuration, follow these steps:

1. Use SSH to log in to one of the primary instances created by the Quick Start template.
2. Execute the following commands from the terminal:

```
mongo
use admin
db.auth("admin", "YourAdminPassword")
rs.printReplicationInfo()
rs.status()
```

3. Verify that the mongo shell connects to the local host on the default TCP port (27017), and that the output reflects the configuration that you specified for the Quick Start template.

For additional information on testing the MongoDB server, see the [MongoDB documentation](#).

Backing Up Your Data

For backup, we recommend using Amazon S3 to keep a copy of your MongoDB data. Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. You can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures.

For other backup strategies, see the [MongoDB 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 (IAM)

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

OS Security

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

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

Network Security

The default network security setup of this solution follows AWS security best practices. The provisioned MongoDB instances are deployed in private subnets and can be accessed in three ways:

- By connecting to the bastion host instance by using an SSH terminal.
- From AWS resources (such as Amazon EC2) that you might have in the `MongoDBServerAccessSecurityGroup` security group, or that you might launch using the security group. You may include your application instance in this security group.
- By including new rules in `MongoDBServerSecurityGroup` to allow access to your database from a known IP block CIDR. For example, you might add an inbound rule to enable the VLAN 10.50.10.0/24 in your data center to connect through a VPN or AWS Direct Connect.

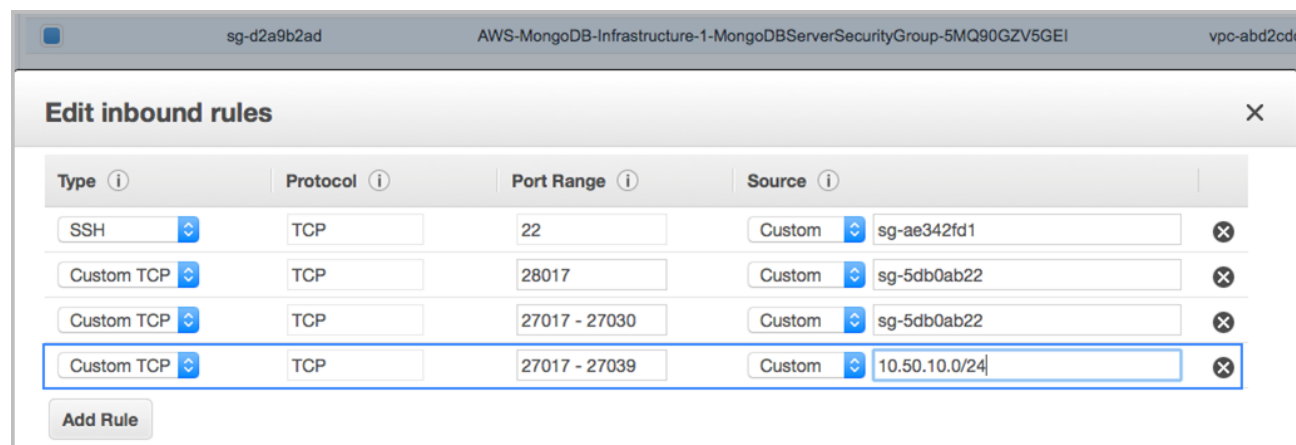


Figure 8: Adding inbound rules to your security group

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 creates three security groups:

- `MongoDBServerSecurityGroup` is used to grant the bastion hosts access to port 22 of the MongoDB instances.

- `MongoDBServersSecurityGroup` is used for communications between MongoDB instances: primary and replica instances on database ports and SSH ports.
- `MongoDBServerAccessSecurityGroup` gives EC2 instances access to your database on the port you set up for database listeners.

After the Quick Start deployment, you are responsible for maintaining these security groups and including or excluding rules.

Database Security

The solution sets up a new root user with a specified administrator user name (by default, “admin”) and an administrator password. Unauthorized database access is not allowed. In addition, an [internal keyfile authentication](#) is set up between replica set nodes.

Additional Resources

AWS services

- Getting Started
<http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/intro.html>
- AWS CloudFormation
<http://aws.amazon.com/documentation/cloudformation/>
- Amazon EC2
 - User’s guide:
<http://aws.amazon.com/documentation/ec2/>
 - Regions and Availability Zones:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>
 - Key pairs:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>
 - Instance stores:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#instance-storage-concepts>
 - FAQ:
<http://aws.amazon.com/ec2/faqs>

- Amazon EBS
 - Overview:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>
 - Volume types:
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>
- Amazon Identity and Access Management
 - User's guide:
<http://aws.amazon.com/documentation/iam/>
 - Benefits of the IAM role:
<http://docs.aws.amazon.com/IAM/latest/UserGuide/role-usecase-ec2app.html>
- Amazon VPC
 - Documentation:
<http://aws.amazon.com/documentation/vpc/>
 - High availability for NAT instances
<https://aws.amazon.com/articles/2781451301784570>
- AWS Security Center
<http://aws.amazon.com/security/>

MongoDB

- MongoDB on AWS: Guidelines and Best Practices
http://do.awsstatic.com/whitepapers/AWS_NoSQL_MongoDB.pdf
- MongoDB documentation
<https://docs.mongodb.com/>
- MongoDB production notes
<https://docs.mongodb.com/master/administration/production-notes/>
MongoDB security checklist
<https://docs.mongodb.com/master/administration/security-checklist/>
- MongoDB Atlas documentation
<https://docs.atlas.mongodb.com/>
- MongoDB Architecture Guide
<https://www.mongodb.com/collateral/mongodb-architecture-guide>
- Performance Best Practices for MongoDB
<https://www.mongodb.com/collateral/mongodb-performance-best-practices>

- MongoDB Multi-Data Center Deployments
<https://www.mongodb.com/collateral/mongodb-multi-data-center-deployments>
- Testing the MongoDB Server
<https://github.com/mongodb/mongo/wiki/Test-The-Mongodb-Server>
- MongoDB Backup Methods
<https://docs.mongodb.com/manual/core/backups/>
- mongodump reference
<https://docs.mongodb.com/manual/reference/program/mongodump/>

Additional Quick Start Reference Deployments

- AWS Quick Start home page
<https://aws.amazon.com/quickstart/>
- Amazon VPC Quick Start
<https://aws.amazon.com/quickstart/architecture/vpc/>
- Linux Bastion Hosts Quick Start
<http://aws.amazon.com/quickstart/architecture/linux-bastion/>

Send Us Feedback

We welcome your questions and comments. Please post your feedback on the [AWS Quick Start Discussion Forum](#).

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

Document Revisions

Date	Change	In sections
May 2017	Upgraded MongoDB to version 3.4; removed sharding configuration; updated security groups and added database security; updated parameters	Changes in templates, architecture , and throughout guide
August 2016	Updated templates to change default instance types and to add Availability Zone parameters	Template updates and changes to parameter table
April 2015	Initial publication	—

© 2017, 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.