CloudStax NoSQL DB for Apache Cassandra on the AWS Cloud

Quick Start Reference Deployment

May 2018

CloudStax, Inc.
AWS Quick Start Team

Contents

Overview ........................................................................................................................................... 2
CloudStax NoSQL DB for Apache Cassandra on AWS ................................................................. 2
Costs and Licenses ......................................................................................................................... 3
Architecture ...................................................................................................................................... 3
Prerequisites .................................................................................................................................. 5
  Specialized Knowledge ............................................................................................................... 5
Deployment Options .................................................................................................................... 6
Deployment Steps .......................................................................................................................... 6
  Step 1. Prepare Your AWS Account .............................................................................................. 6
  Step 2. Launch the Quick Start .................................................................................................... 7
  Step 3. Test the Deployment ....................................................................................................... 15
Best Practices for Using CloudStax NoSQL DB for Apache Cassandra on AWS .................. 18
  Accessing Cassandra ................................................................................................................. 18
  High Availability ....................................................................................................................... 18
  Scaling the Cluster .................................................................................................................... 18
  Cassandra Configuration ........................................................................................................... 19
  Security ..................................................................................................................................... 20
Operating System Security .................................................................................................. 20
Network Security .................................................................................................................. 20
Cassandra Security .................................................................................................................. 20
FAQ ........................................................................................................................................ 20
GitHub Repository .................................................................................................................... 21
Document Revisions .................................................................................................................. 22

This Quick Start deployment guide was created by CloudStax, Inc. in partnership with Amazon Web Services (AWS).

Quick Starts are automated reference deployments that use AWS CloudFormation templates to deploy key technologies on AWS, following AWS best practices.

Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying CloudStax NoSQL DB for Apache Cassandra on the AWS Cloud.

This Quick Start is for users who need an easily deployed and self-governed Apache Cassandra cluster on AWS.

CloudStax NoSQL DB for Apache Cassandra on AWS

CloudStax NoSQL DB is a NoSQL database powered by Apache Cassandra that makes it easy to set up, manage, and scale Apache Cassandra on AWS. Apache Cassandra is a masterless, peer-to-peer, distributed system. It’s a production-level NoSQL database solution that can be deployed across many commodity servers for decentralization, scalability, and high availability. CloudStax NoSQL DB for Cassandra removes the complexity associated with deploying and managing Apache Cassandra. It provides a high-performance, highly scalable, and cost-effective NoSQL database that you can use to manage large amounts of data.

CloudStax NoSQL DB runs Apache Cassandra in a container on AWS. The Quick Start deployment uses Amazon Elastic Container Service (Amazon ECS) for container orchestration and CloudStax FireCamp for stateful service management. Each Cassandra container has one Amazon Elastic Block Store (Amazon EBS) volume for the commit log and one EBS volume for data. Each Cassandra container also has a unique Domain Name System (DNS) name, so an application can access Cassandra just by using the DNS name.
Deploying CloudStax NoSQL DB on AWS helps enhance the reliability of using Cassandra for your production deployments. The benefits of running CloudStax NoSQL DB for Cassandra on AWS include the following:

- Cassandra nodes are deployed across multiple Availability Zones for high availability.
- The Multi-AZ environment on AWS provides automatic failure detection and recovery. If one Cassandra node fails, the AWS Auto Scaling group starts a new node, and the container service (Amazon ECS) automatically starts the service container. FireCamp attaches the original EBS volumes and updates the DNS record. The failover involves no data copy and is seamless to the application.
- AWS helps provide enhanced security and isolation for Cassandra.

Costs and Licenses

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

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

The Quick Start deploys Cassandra version 3.11 and CloudStax FireCamp. Apache Cassandra and CloudStax FireCamp are open-source software. They are distributed under the Apache 2.0 license and are free to use.

Architecture

Deploying this Quick Start for a new virtual private cloud (VPC) with default parameters builds the following Cassandra environment in the AWS Cloud.
The Quick Start sets up the following:

- A highly available architecture that spans three Availability Zones.*
- A VPC configured with public and private subnets according to AWS best practices. This provides the network infrastructure for your deployment.*
- An internet gateway to allow access to the internet. The bastion hosts use this gateway to send and receive traffic.*
- In the public subnets, NAT gateways to allow outbound internet connectivity for resources (Cassandra 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. Two bastion host instances are deployed by default, but this number is configurable. (For more information, see the Linux Bastion Quick Start.*)

*Source: AWS Quick Start architecture for CloudStax NoSQL DB for Apache Cassandra on AWS.
- An AWS Identity and Access Management (IAM) instance role with fine-grained permissions for access to AWS services.
- Security groups to enable communication within the VPC and to restrict access to only necessary protocols and ports.
- In the private subnets, an Amazon ECS cluster with CloudStax FireCamp software, and a customizable Cassandra cluster. The Quick Start launches the Cassandra members across all Availability Zones. If you choose an AWS Region that provides only two Availability Zones, the Quick Start reuses one of the zones to create the third subnet.
- An Amazon DynamoDB table to persist FireCamp data.
- An Amazon Route 53 hosted zone for the DNS names of each Cassandra member, and two EBS volumes for each Cassandra member.
- Amazon CloudWatch Logs for collecting Cassandra service logs sent from Cassandra containers.

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

The Quick Start launches all the Cassandra-related nodes in the private subnets, so you can access the nodes by using Secure Shell (SSH) to connect to the bastion hosts. Instead of using a remote access CIDR for each Cassandra instance, the deployment requires a bastion host security group 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 into an existing VPC, you must create a security group for your bastion hosts or use a security group that already exists.

**Prerequisites**

**Specialized Knowledge**

Before you deploy this Quick Start, we recommend that you become familiar with the following AWS services. (If you are new to AWS, see [Getting Started with AWS](#).

- [Amazon DynamoDB](#)
- [Amazon EBS](#)
- [Amazon EC2](#)
- [Amazon ECS](#)
- [Amazon Route 53](#)
- [Amazon VPC](#)
• **AWS Auto Scaling**
• **AWS CloudFormation**

We also recommend that you become familiar with the features and configuration of Apache Cassandra and CloudStax FireCamp:

• **Apache Cassandra**
• **CloudStax FireCamp**

**Deployment Options**

This Quick Start provides two deployment options:

• **Deploy CloudStax NoSQL DB for Apache Cassandra 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 CloudStax NoSQL DB for Apache Cassandra into this new VPC.

• **Deploy CloudStax NoSQL DB for Apache Cassandra into an existing VPC.** This option provisions CloudStax NoSQL DB for Apache Cassandra in your existing AWS infrastructure.

The Quick Start provides separate templates for these options. It also lets you configure CIDR blocks, instance types, and CloudStax NoSQL DB for Apache Cassandra settings, as discussed later in this guide.

**Deployment Steps**

**Step 1. Prepare Your AWS Account**

1. If you don’t already have an AWS account, create one at [https://aws.amazon.com](https://aws.amazon.com) by following the on-screen instructions.

2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy CloudStax NoSQL DB for Apache Cassandra on AWS.

3. Create a **key pair** in your preferred region.

4. If necessary, **request a service limit increase** for the Amazon EC2 **m4.2xlarge** instance type. You might need to do this if you already have an existing deployment that uses this instance type, and you think you might exceed the **default limit** with this deployment.
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.

   ![Option 1: Deploy Quick Start into a new VPC on AWS](Launch)
   ![Option 2: Deploy Quick Start into an existing VPC on AWS](Launch)

   **Important** If you’re deploying CloudStax NoSQL DB for Apache Cassandra into an existing VPC, make sure that your VPC has two private subnets in different Availability Zones for the database instances. These subnets require [NAT gateways or NAT instances](https://aws.amazon.com/nat/) in their route tables, to allow the instances to download packages and software without exposing them to the internet. You will also need the domain name option configured in the DHCP options as explained in the Amazon VPC documentation. You will be prompted for your VPC settings when you launch the Quick Start.

   Each deployment takes about 30 minutes to complete.

2. Check the region that’s displayed in the upper-right corner of the navigation bar, and change it if necessary. This is where the network infrastructure for CloudStax NoSQL DB for Apache Cassandra will be built. The template is launched in the US East (Ohio) 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 the Quick Start into a new VPC
- Parameters for deploying the Quick Start into an existing VPC

• **Option 1: Parameters for deploying CloudStax NoSQL DB for Apache Cassandra into a new VPC**

**Network template**

**Network Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability Zones</td>
<td>Requires input</td>
<td>The list of Availability Zones to use for the subnets in the VPC. This field displays the available zones within your selected region. You can choose two or three Availability Zones from this list (we recommend three zones for a production system). The logical order of your selections is preserved in your deployment. After you make your selections, make sure that the value of the <strong>Number of Availability Zones</strong> parameter matches the number of selections.</td>
</tr>
<tr>
<td>Number of Availability Zones</td>
<td>3</td>
<td>The number of Availability Zones (2 or 3) you want to use in your deployment. This count must match the number of selections in the Availability Zones parameter; otherwise, your deployment will fail with an AWS CloudFormation template validation error. (Note that some AWS Regions provide only two Availability Zones.) For a production environment, we recommend that you use three Availability Zones.</td>
</tr>
<tr>
<td>VPC CIDR</td>
<td>10.0.0.0/16</td>
<td>The CIDR block for the VPC.</td>
</tr>
<tr>
<td>Private Subnet 1 CIDR</td>
<td>10.0.0.0/19</td>
<td>The CIDR block for the private subnet located in Availability Zone 1.</td>
</tr>
<tr>
<td>Private Subnet 2 CIDR</td>
<td>10.0.32.0/19</td>
<td>The CIDR block for the private subnet located in Availability Zone 2.</td>
</tr>
<tr>
<td>Private Subnet 3 CIDR</td>
<td>10.0.64.0/19</td>
<td>The CIDR block for the private subnet located in Availability Zone 3.</td>
</tr>
<tr>
<td>Public Subnet 1 CIDR</td>
<td>10.0.128.0/20</td>
<td>The CIDR block for the public (DMZ) subnet located in Availability Zone 1.</td>
</tr>
</tbody>
</table>
### Parameter label (name) | Default | Description
---|---|---
**Public Subnet 2 CIDR** (PublicSubnet2CIDR) | 10.0.144.0/20 | The CIDR block for the public (DMZ) subnet located in Availability Zone 2.

**Public Subnet 3 CIDR** (PublicSubnet3CIDR) | 10.0.160.0/20 | The CIDR block for the public (DMZ) subnet located in Availability Zone 3.

**Allowed Bastion External Access CIDR** (RemoteAccessCIDR) | Requires input | The CIDR IP range that is allowed external SSH access to the bastion host instances. We recommend that you set this value to a trusted IP range. For example, you might want to grant only your corporate network access to the software. The range you specify must be in the form x.x.x.x/x (for example, 10.2.0.0/24).

### Security Configuration:

### Parameter label (name) | Default | Description
---|---|---
**Key Pair Name** (KeyPairName) | Requires input | An existing public/private key pair, which allows you to connect securely to the bastion hosts and cluster nodes. When you created an AWS account, this is the key pair you created in your preferred region.

### Linux Bastion Amazon EC2 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. If you choose CentOS, make sure that you have a subscription to the [CentOS AMI in AWS Marketplace](https://aws.amazon.com/marketplace/pp/B01E1QH3RY). (BastionAMIOS)

**Bastion Instance Type** (BastionInstanceType) | t2.micro | The EC2 instance type for the bastion host instances.

**Number of Bastion Hosts** (NumBastionHosts) | 2 | The number of Linux bastion hosts to run. Auto Scaling will ensure that you always have this number of bastion hosts running. The maximum is 4 bastion hosts.

### Cluster Configuration:

### Parameter label (name) | Default | Description
---|---|---
**Cluster Name** (ClusterName) | Requires input | The name of the Amazon ECS cluster. This string consists of 1-54 characters (letters, numbers, or hyphens) and must start with a letter. This string could be the same as the stack name. The cluster domain will be: `ClusterName-firecamp.com`.
<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CloudStax FireCamp Release Version</strong></td>
<td>0.9.5</td>
<td>The version of FireCamp to deploy. For information about each release, see the Releases section of the GitHub repository for CloudStax FireCamp.</td>
</tr>
<tr>
<td><strong>Number of Container Worker Nodes</strong></td>
<td>3</td>
<td>The number of container instances to launch in the cluster. This number must match the number of Apache Cassandra replicas.</td>
</tr>
<tr>
<td><strong>Node AMI Type</strong></td>
<td></td>
<td>The Linux distribution for the AMI to be used for the container service instances.</td>
</tr>
<tr>
<td><strong>Node Instance Type</strong></td>
<td>m4.2xlarge</td>
<td>The EC2 instance type for the container service instances.</td>
</tr>
<tr>
<td><strong>Delete EBS Volumes</strong></td>
<td>false</td>
<td>Set this parameter to true if you want to delete the Apache Cassandra EBS volumes when you delete the stack.</td>
</tr>
</tbody>
</table>

Apache Cassandra Service Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service Name</strong></td>
<td>Requires input</td>
<td>The name of the Apache Cassandra service. This string consists of 1-58 characters (letters, numbers, or hyphens) and must start with a letter. The service DNS name will be ServiceName-index.ClusterName-firecamp.com. Each label of the domain name can be up to 63 bytes long. Reserve 4 bytes for the index.</td>
</tr>
<tr>
<td><strong>Number of Replicas</strong></td>
<td>3</td>
<td>The number of Cassandra replicas to deploy.</td>
</tr>
<tr>
<td><strong>Apache Cassandra JVM Heap Size MB</strong></td>
<td>8192</td>
<td>The heap size for the Java virtual machine (JVM) for Apache Cassandra, in MiB.</td>
</tr>
<tr>
<td><strong>Cassandra Data Volume Type</strong></td>
<td>gp2</td>
<td>The data volume type for Cassandra. The three options are General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Throughput Optimized HDD (st1).</td>
</tr>
<tr>
<td><strong>Data Volume Iops</strong></td>
<td>100</td>
<td>The IOPS of the data volume. This parameter is used only if you choose io1 for the data volume type.</td>
</tr>
<tr>
<td><strong>Data Volume Size GB</strong></td>
<td>400</td>
<td>The size of the data volume, in GiB.</td>
</tr>
<tr>
<td><strong>Encrypt Data Volume</strong></td>
<td>false</td>
<td>Set this parameter to true if you want to encrypt the data volume.</td>
</tr>
</tbody>
</table>
### Parameter label (name) | Default | Description
--- | --- | ---
**Cassandra Journal Volume Type** (JournalVolumeType) | gp2 | The journal volume type for Cassandra. The three options are General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Throughput Optimized HDD (st1).

**Journal Volume Iops** (JournalIops) | 100 | The IOPS of the journal volume. This parameter is used only if you choose io1 for the journal volume type.

**Journal Volume Size GB** (JournalVolumeSizeGB) | 20 | The size of the journal volume, in GiB.

**Encrypt Journal Volume** (EncryptJournalVolume) | false | Set this parameter to **true** if you want to encrypt the journal volume.

**JMX Remote User** (JmxRemoteUser) | jmxuser | The user name of the Java Management Extensions (JMX) remote user for Apache Cassandra. This is a 1-255 character string.

**JMX Remote Password** (JmxRemotePassword) | jmxpassword | The password of the Apache Cassandra JMX remote user. This is a 1-255 character string.

---

### AWS Quick Start Configuration:

| Parameter label (name) | Default | Description |
--- | --- | ---
**Quick Start S3 Bucket Name** (QSS3BucketName) | aws-quickstart | The S3 bucket you have 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-cloudstax-nosql/ | 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 CloudStax NoSQL DB for Apache Cassandra into an existing VPC

View template

Network Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability Zones (AvailabilityZones)</td>
<td>Requires input</td>
<td>The list of Availability Zones to use for the subnets in the VPC. This field displays the available zones within your selected region. You can choose two or three Availability Zones from this list (we recommend three zones for a production system). The logical order of your selections is preserved in your deployment. After you make your selections, make sure that the value of the <strong>Number of Availability Zones</strong> parameter matches the number of selections.</td>
</tr>
<tr>
<td>Number of Availability Zones (NumberOfAZs)</td>
<td>3</td>
<td>The number of Availability Zones (2 or 3) you want to use in your deployment. This count must match the number of selections in the <strong>Availability Zones</strong> parameter; otherwise, your deployment will fail with an AWS CloudFormation template validation error. (Note that some AWS Regions provide only two Availability Zones.) For a production environment, we recommend that you use three Availability Zones.</td>
</tr>
<tr>
<td>VPC ID (VPCID)</td>
<td>Requires input</td>
<td>The ID of your existing VPC (e.g., vpc-0343606e).</td>
</tr>
<tr>
<td>Private Subnet 1 ID (PrivateSubnet1ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 1 in your existing VPC (e.g., subnet-a0246dcd).</td>
</tr>
<tr>
<td>Private Subnet 2 ID (PrivateSubnet2ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 2 in your existing VPC.</td>
</tr>
<tr>
<td>Private Subnet 3 ID (PrivateSubnet3ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 3 in your existing VPC.</td>
</tr>
<tr>
<td>Bastion Security Group ID (BastionSecurityGroupID)</td>
<td>Requires input</td>
<td>The ID of the bastion host security group for accessing the cluster nodes through SSH.</td>
</tr>
<tr>
<td>Lambda Subnet CIDR (LambdaSubnetCIDR)</td>
<td>Requires input</td>
<td>The subnet CIDR range for the Lambda function that initializes the Cassandra service. The range you specify must be in the form x.x.x.x/1 (for example, 10.2.0.0/24).</td>
</tr>
</tbody>
</table>
Security Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Pair Name (KeyPairName)</td>
<td>Requires input</td>
<td>An existing public/private key pair, which allows you to connect securely to the bastion hosts and cluster nodes. When you created an AWS account, this is the key pair you created in your preferred region.</td>
</tr>
</tbody>
</table>

FireCamp Cluster Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Name (ClusterName)</td>
<td>Requires input</td>
<td>The name of the Amazon ECS cluster. This string consists of 1-54 characters (letters, numbers, or hyphens) and must start with a letter. This string could be the same as the stack name. The cluster domain will be: ClusterName-firecamp.com.</td>
</tr>
<tr>
<td>CloudStax FireCamp Release Version (FireCampRelease)</td>
<td>0.9.5</td>
<td>The version of FireCamp to deploy. For information about each release, see the Releases section of the GitHub repository for CloudStax FireCamp.</td>
</tr>
<tr>
<td>Node AMI Type (NodeAMIOS)</td>
<td>Amazon-Linux-HVM</td>
<td>The Linux distribution for the AMI to be used for the container service instances.</td>
</tr>
<tr>
<td>Number of Container Worker Nodes (NumberOfNodes)</td>
<td>3</td>
<td>The number of container instances to launch in the cluster. This number must match the number of Apache Cassandra replicas.</td>
</tr>
<tr>
<td>Node Instance Type (NodeInstanceType)</td>
<td>m4.2xlarge</td>
<td>The EC2 instance type for the container service instances.</td>
</tr>
<tr>
<td>Delete Apache Cassandra EBS Volumes (DeleteVolume)</td>
<td>false</td>
<td>Set this parameter to true if you want to delete the Apache Cassandra EBS volumes when you delete the stack.</td>
</tr>
</tbody>
</table>

Apache Cassandra Service Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Cassandra Release Version (CassandraRelease)</td>
<td>3.11</td>
<td>The version of Apache Cassandra to deploy. For information about each release, see <a href="http://cassandra.apache.org">http://cassandra.apache.org</a>.</td>
</tr>
<tr>
<td>Service Name (ServiceName)</td>
<td>Requires input</td>
<td>The name of the Apache Cassandra service. This string consists of 1-58 characters (letters, numbers, or hyphens) and must start with a letter. The service DNS name will be ServiceName-index.ClusterName-firecamp.com. Each label of the domain name can be up to 63 bytes long. Reserve 4 bytes for the index.</td>
</tr>
<tr>
<td>Parameter label (name)</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Number of Replicas (Replicas)</td>
<td>3</td>
<td>The number of Cassandra replicas to deploy.</td>
</tr>
<tr>
<td>Apache Cassandra JVM Heap Size MB (HeapSizeMB)</td>
<td>8192</td>
<td>The heap size for the Java virtual machine (JVM) for Apache Cassandra, in MiB.</td>
</tr>
<tr>
<td>Cassandra Data Volume Type (VolumeType)</td>
<td>gp2</td>
<td>The data volume type for Cassandra. The three options are General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Throughput Optimized HDD (st1).</td>
</tr>
<tr>
<td>Data Volume Iops (Iops)</td>
<td>100</td>
<td>The IOPS of the data volume. This parameter is used only if you choose io1 for the data volume type.</td>
</tr>
<tr>
<td>Data Volume Size GB (VolumeSizeGB)</td>
<td>400</td>
<td>The size of the data volume, in GiB.</td>
</tr>
<tr>
<td>Encrypt Data Volume (EncryptVolume)</td>
<td>false</td>
<td>Set this parameter to <strong>true</strong> if you want to encrypt the data volume.</td>
</tr>
<tr>
<td>Cassandra Journal Volume Type (JournalVolumeType)</td>
<td>gp2</td>
<td>The journal volume type for Cassandra. The three options are General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Throughput Optimized HDD (st1).</td>
</tr>
<tr>
<td>Journal Volume Iops (JournalIops)</td>
<td>100</td>
<td>The IOPS of the journal volume. This parameter is used only if you choose io1 for the journal volume type.</td>
</tr>
<tr>
<td>Journal Volume Size GB (JournalVolumeSizeGB)</td>
<td>20</td>
<td>The size of the journal volume, in GiB.</td>
</tr>
<tr>
<td>Encrypt Journal Volume (EncryptJournalVolume)</td>
<td>false</td>
<td>Set this parameter to <strong>true</strong> if you want to encrypt the journal volume.</td>
</tr>
<tr>
<td>JMX Remote User (JmxRemoteUser)</td>
<td>jmxuser</td>
<td>The user name of the Java Management Extensions (JMX) remote user for Apache Cassandra. This is a 1-255 character string.</td>
</tr>
<tr>
<td>JMX Remote Password (JmxRemotePassword)</td>
<td>jmxpassword</td>
<td>The password of the Apache Cassandra JMX remote user. This is a 1-255 character string.</td>
</tr>
</tbody>
</table>

**AWS Quick Start Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Start S3 Bucket Name (QSS3BucketName)</td>
<td>aws-quickstart</td>
<td>The S3 bucket you have 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.</td>
</tr>
<tr>
<td>Parameter label (name)</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Quick Start S3 Key Prefix</td>
<td>quickstart-cloudstax-nosql/</td>
<td>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.</td>
</tr>
</tbody>
</table>

5. 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**.

6. 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.

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

8. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the CloudStax NoSQL DB for Apache Cassandra cluster is ready.

9. Use the information displayed in the **Outputs** tab for the stack to view the resources that were created.

![Figure 2: Outputs tab after deployment](image)

**Step 3. Test the Deployment**

After you deploy the Cassandra cluster, you can check the status of the cluster.

1. Open the Amazon ECS console at [https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).

You should see all tasks of the Cassandra service and the CloudStax FireCamp management service running. For example, if you set the **Cluster Name** parameter to **t1** and the **ServiceName** parameter to **mycas**, the Amazon ECS console displays a screen similar to Figure 3.
Figure 3: Checking the status of the Cassandra cluster in the Amazon ECS console

2. Deploy an EC2 instance in the **CassandraAccessSecurityGroup** security group in the same VPC, and install Cassandra version 3.11 or run a Cassandra container to get the Cassandra administrative tools that are available on the instance.

3. To access **CassandraAccessSecurityGroup** and the VPC, look in the **Outputs** tab for the stack in the AWS CloudFormation console, as shown previously in Figure 2.

4. Use the Cassandra Query Language (CQL) to create a new superuser and disable the default **cassandra** superuser. (In this example code, the cluster name is **t1** and the Cassandra service name is **mycas**.)

   a. Log in to the cluster:

   ```
cqlsh mycas-0.t1-firecamp.com -u cassandra -p cassandra
   ```

   b. Create a new superuser:

   ```
   CREATE ROLE newsuperuser WITH SUPERUSER = true AND LOGIN = true AND PASSWORD = 'super';
   ```

   c. Log out, and log back in to the cluster with the new superuser:

   ```
cqlsh mycas-0.t1-firecamp.com -u newsuperuser -p super
   ```
d. Disable the default superuser:

```
ALTER ROLE cassandra WITH SUPERUSER = false AND LOGIN = false;
```

e. Set up the roles and credentials for your application users by using CREATE ROLE statements.

4. You can then create a Cassandra keyspace and user, and start inserting the records:

a. Log in with the new superuser:

```
cqlsh mycas-0.t1-firecamp.com -u newsuperuser -p super
```

b. Create a keyspace and a table:

```
CREATE KEYSPACE test WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy', 'us-east-1' : 3 };
use test;
CREATE TABLE users (userid text PRIMARY KEY, first_name text, last_name text);
```

c. Create a role for the keyspace and table:

```
CREATE ROLE supervisor;
GRANT MODIFY ON test.users TO supervisor;
GRANT SELECT ON test.users TO supervisor;
CREATE ROLE pam WITH PASSWORD = ’password’ AND LOGIN = true;
GRANT supervisor TO pam;
LIST ALL PERMISSIONS OF pam;
```

d. Log in by using the new pam role:

```
cqlsh mycas-2.t1-firecamp.com -u pam -p password
```

e. Insert new records:

```
use test;
inset into users (userid, first_name, last_name) values(’user1’, ’a1’, ’b1’);
select * from users;
```
Best Practices for Using CloudStax NoSQL DB for Apache Cassandra on AWS

Accessing Cassandra

CloudStax uses DNS names for easy access to Cassandra nodes. CloudStax binds each DNS name to one Cassandra node. If a Cassandra container moves to another node, the DNS record is updated to point to the new node, so an application can seamlessly access Cassandra with the DNS name.

By default, the Java virtual machine (JVM) caches a successful DNS lookup forever. If you use the Cassandra Java CQL driver, we recommend that you set the JVM TTL to a reasonable value such as 60 seconds. This ensures that the Java CQL driver can look up the new IP address from the domain name when a Cassandra container moves to another node.

Let’s assume that the cluster name is t1, the Cassandra service name is mycas, and the Cassandra cluster has three replicas. Cassandra will use mycas-0.t1-firecamp.com to mycas-2.t1-firecamp.com as DNS names, and an application can access Cassandra by using these DNS names.

High Availability

Cassandra replicas are distributed to the Availability Zones. The GossipingPropertyFileSnitch is used to teach Cassandra about the topology. Each replica will record the AWS Region as dc, and the Availability Zone as rack.

Scaling the Cluster

You could start with three Cassandra replicas in three Availability Zones and scale out the Cassandra replicas (for example, from three replicas to six) to accommodate data or node increases. In the following examples, the Cassandra service name is mycas, and the cluster name is t1.

1. Scale the cluster nodes. For example, on AWS, navigate to the Auto Scaling section of the Amazon EC2 console and scale the Cassandra cluster nodes. (For instructions, see the AWS documentation.)

2. Scale the Cassandra replicas by using the FireCamp CLI:

   ```
   # firecamp-service-cli -op=scale-service -service-type=cassandra -region=us-east-1 -cluster=t1 -service-name=mycas -replicas=6
   ```

3. Use nodetool status to verify that the new replicas are fully bootstrapped and all other nodes are up (status U or up, state N or normal):
4. Run `nodetool cleanup` on each of the previously existing nodes to remove the keys that no longer belong to those nodes. Wait for cleanup to complete on one node before running `nodetool cleanup` on the next node. Cleanup can be safely postponed until low-usage hours.

```plaintext
# nodetool -h mycas-3.t1-firecamp.com -u cassandrajmx -pw changeme
status
# nodetool -h mycas-4.t1-firecamp.com -u cassandrajmx -pw changeme
status
# nodetool -h mycas-5.t1-firecamp.com -u cassandrajmx -pw changeme
status
```

You can start from three replicas, and scale up three replicas at a time. Scaling up from a single replica and scaling down aren’t currently supported.

If you want to stop the Cassandra containers when they’re idle, run:

```plaintext
firecamp-service-cli -op=stop-service -service-name=mycas
```

This will stop the Cassandra containers without affecting the volumes. To restart the containers, run:

```plaintext
firecamp-service-cli -op=start-service -service-name=mycas
```

**Cassandra Configuration**

- **Cache**: The Cassandra key cache is enabled, and the row cache is disabled. The key cache size is automatically calculated based on the total memory of the node.

- **System parameters**: This deployment implements the recommended configuration settings for Cassandra that are documented in the *DataStax Developer Guide*. For example, the number of file descriptors and threads are increased, TCP tunings are applied, memory swapping is disabled, `memlock` is set to unlimited, and `vm_map_count`
is increased. For more information, see the Readme file in the GitHub repository for CloudStax FireCamp.

Security

The AWS Cloud provides scalable, highly reliable services that help customers deploy applications and data quickly and securely.

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.

Network Security

The default network security setup of this solution follows AWS security best practices. The provisioned Cassandra instances are deployed in private subnets and can be accessed in two 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 CassandraAccessSecurityGroup security group, or that you might launch by using the security group. You may include your application instance in this security group.

Cassandra Security

This deployment enables Cassandra authentication and authorization, as discussed in the Cassandra security guide.

FAQ

Q. I encountered a CREATE_FAILED error when I launched the Quick Start.
A. If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with Rollback on failure set to No. (This setting is under Advanced in the AWS CloudFormation console, Options page.) With this setting, the stack’s state will be retained and the instance will be left running, so you can troubleshoot the issue. (Look at the log files in %ProgramFiles%\Amazon\EC2ConfigService and C:\cfn\log.)
Important When you set Rollback on failure to No, you will continue to incur AWS charges for this stack. Please make sure to delete the stack when you finish troubleshooting.

For additional information, see Troubleshooting AWS CloudFormation on the AWS website.

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

**A.** We recommend that you launch the Quick Start templates from the links in this guide or from another S3 bucket. If you deploy the templates from a local copy on your computer or from a non-S3 location, you might encounter template size limitations when you create the stack. For more information about AWS CloudFormation limits, see the AWS documentation.

**Q.** I need help using the environment set up by this Quick Start.

**A.** For questions about using CloudStax NoSQL DB for Apache Cassandra, use the Issues section of the GitHub repository for CloudStax FireCamp or the FireCamp mailing list.

If the issue is associated with Cassandra, get help from the Cassandra User mailing list and the #cassandra IRC channel.

**GitHub Repository**

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

**AWS services**

- Amazon DynamoDB
  https://aws.amazon.com/documentation/dynamodb/

- Amazon EBS

- Amazon EC2
  https://aws.amazon.com/documentation/ec2/

- Amazon ECS
  https://aws.amazon.com/documentation/ecs/
• Amazon Route 53
  https://aws.amazon.com/documentation/route53/

• Amazon VPC
  https://aws.amazon.com/documentation/vpc/

• AWS Auto Scaling
  https://aws.amazon.com/documentation/autoscaling/

• AWS CloudFormation
  https://aws.amazon.com/documentation/cloudformation/

Apache Cassandra
• Cassandra documentation
  http://cassandra.apache.org/doc/latest/

CloudStax
• CloudStax FireCamp
  https://github.com/cloudstax/firecamp

• CloudStax NoSQL DB for Apache Cassandra
  https://github.com/cloudstax/firecamp/tree/master/catalog/cassandra

Quick Start reference deployments
• AWS Quick Start home page
  https://aws.amazon.com/quickstart/

Document Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
<th>In sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 2018</td>
<td>Initial publication</td>
<td>—</td>
</tr>
</tbody>
</table>
© 2018, Amazon Web Services, Inc. or its affiliates, and and CloudStax Inc. 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.