CloudStax Cache for Redis on the AWS Cloud

Quick Start Reference Deployment

February 2018

CloudStax Inc.
AWS Quick Start Team

Contents

Overview ................................................................. 2
CloudStax Cache for Redis on AWS ...................................... 2
Costs and Licenses ......................................................... 3
Architecture ............................................................... 3
Prerequisites ............................................................. 5
Specialized Knowledge .................................................. 5
Technical Requirements ................................................. 6
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 Using CloudStax Cache for Redis on AWS .................. 16
  Easy Access .......................................................... 16
  High Availability ..................................................... 17
  Data Persistence ....................................................... 18
  Data Backup .......................................................... 18
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redis Configuration</td>
<td>18</td>
</tr>
<tr>
<td>Security</td>
<td>19</td>
</tr>
<tr>
<td>Operating System Security</td>
<td>19</td>
</tr>
<tr>
<td>Network Security</td>
<td>19</td>
</tr>
<tr>
<td>Redis Security</td>
<td>19</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>20</td>
</tr>
<tr>
<td>GitHub Repository</td>
<td>21</td>
</tr>
<tr>
<td>Additional Resources</td>
<td>21</td>
</tr>
<tr>
<td>Document Revisions</td>
<td>22</td>
</tr>
</tbody>
</table>

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 Cache for Redis on the AWS Cloud.

This Quick Start is for users who need an easily deployed and self-managed Redis cluster on AWS.

**CloudStax Cache for Redis on AWS**

CloudStax Cache is an in-memory data store service for Redis that makes it easy to set up, manage, and scale Redis on AWS. CloudStax Cache for Redis removes the complexity associated with deploying and managing Redis. It provides a high-performance, scalable, and cost-effective in-memory database or cache solution that you can use to improve the performance of your applications.

CloudStax Cache runs Redis in a container on AWS. This deployment uses Amazon Elastic Container Service (Amazon ECS) for container orchestration and CloudStax FireCamp for stateful service management. Each Redis container has one Amazon Elastic Block Store (Amazon EBS) volume and one static IP. Each Redis container also has a unique Domain
Name System (DNS) name that points to the static IP, so an application can simply access Redis by using the DNS name.

Deploying CloudStax Cache on AWS enhances the reliability of using Redis for your production deployments. The benefits of running CloudStax Cache for Redis on AWS include the following:

- Redis nodes are deployed across multiple Availability Zones for high availability.
- The Multi-AZ environment on AWS provides automatic failure detection and recovery. If one Redis 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 volume and takes over the static IP. The failover involves no data copy and is seamless to the application.
- If a primary Redis node fails, AWS automatically promotes a read replica to become the new primary node.
- AWS helps provide enhanced security and isolation for Redis.

**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 Redis version 4.0 and CloudStax FireCamp. Redis is open-source software that’s distributed under the BSD license. CloudStax FireCamp is open-source software that’s distributed under the Apache 2.0 License.

**Architecture**

Deploying this Quick Start for a new virtual private cloud (VPC) with default parameters builds the following Redis cluster 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. This gateway is used by the bastion hosts to send and receive traffic.*
- In the public subnets, NAT gateways to allow outbound internet connectivity for resources (Redis 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.*)

*Note: For more details on the architecture and setup, refer to the provided images and documentation.
• 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 Redis cluster that you can run in stand-alone mode, primary-secondary mode, or cluster mode. The Quick Start launches the members of the primary-secondary mode or cluster in different 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 Redis member, and an EBS volume for each Redis member.

* 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 Redis-related nodes in the private subnets, so you can access the nodes by using SSH to connect to the bastion hosts. Instead of using a remote access CIDR for each Redis 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 in 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](https://aws.amazon.com/gettingStarted)).

- [Amazon DynamoDB](https://aws.amazon.com/dynamodb/)
- [Amazon EBS](https://aws.amazon.com/ebs/)
- [Amazon EC2](https://aws.amazon.com/ec2/)
- [Amazon ECS](https://aws.amazon.com/ecs/)
- [Amazon Route 53](https://aws.amazon.com/route53/)
- [Amazon VPC](https://aws.amazon.com/vpc/)
- AWS Auto Scaling
- AWS CloudFormation

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

Technical Requirements
This Quick Start uses the Amazon Machine Image (AMI) for Amazon Linux for EC2 instances, and creates an ECS cluster, EBS volumes, a DynamoDB table, and a Route 53 hosted zone. The EC2 instances are included in an Auto Scaling group. The Redis service logs are sent to Amazon CloudWatch. During the deployment, an AWS Lambda function is triggered to initialize the Redis service, and requires a local, temporary Amazon Simple Storage Service (Amazon S3) bucket to copy the Lambda zip file. The account you use to run this Quick Start must have authorization to create these resources.

Deployment Options
This Quick Start provides two deployment options:

- **Deploy CloudStax Cache for Redis 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 Cache for Redis into this new VPC.

- **Deploy CloudStax Cache for Redis into an existing VPC**. This option provisions CloudStax Cache for Redis in your existing AWS infrastructure.

The Quick Start provides separate templates for these options. It also lets you configure CIDR blocks, instance types, and Redis 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 Cache for Redis on AWS.
3. Create a key pair in your preferred region.

4. If necessary, request a service limit increase for the Amazon EC2 r4.large 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.

<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy CloudStax Cache for Redis into a new VPC on AWS</td>
<td>Deploy CloudStax Cache for Redis into an existing VPC</td>
</tr>
<tr>
<td><img src="Launch" alt="Launch" /></td>
<td><img src="Launch" alt="Launch" /></td>
</tr>
</tbody>
</table>

**Important** If you’re deploying CloudStax Cache for Redis into an existing VPC, make sure that your VPC has two or three private subnets in different Availability Zones for the container instances. These subnets require NAT gateways or NAT instances 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 20-30 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 Cache for Redis will be built. The template is launched in the US West (Oregon) 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 CloudStax Cache for Redis into a new VPC**
- **Parameters for deploying CloudStax Cache for Redis into an existing VPC**

**• Option 1: Parameters for deployment into a new VPC**

*View template*

**Network Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability Zones</strong> <em>(AvailabilityZones)</em></td>
<td><strong>Requires input</strong></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. 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><strong>Number of Availability Zones</strong> <em>(NumberOfAZs)</em></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><strong>VPC CIDR</strong> <em>(VPCCIDR)</em></td>
<td>10.0.0.0/16</td>
<td>The CIDR block for the VPC.</td>
</tr>
<tr>
<td><strong>Private Subnet 1 CIDR</strong> <em>(PrivateSubnet1CIDR)</em></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><strong>Private Subnet 2 CIDR</strong> <em>(PrivateSubnet2CIDR)</em></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><strong>Private Subnet 3 CIDR</strong> <em>(PrivateSubnet3CIDR)</em></td>
<td>10.0.64.0/19</td>
<td>The CIDR block for the private subnet located in Availability Zone 3.</td>
</tr>
</tbody>
</table>
### Amazon Web Services – CloudStax Cache for Redis on the AWS Cloud
February 2018

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Public Subnet 1 CIDR (PublicSubnet1CIDR)</strong></td>
<td>10.0.128.0/20</td>
<td>The CIDR block for the public (DMZ) subnet located in Availability Zone 1.</td>
</tr>
<tr>
<td><strong>Public Subnet 2 CIDR (PublicSubnet2CIDR)</strong></td>
<td>10.0.144.0/20</td>
<td>The CIDR block for the public (DMZ) subnet located in Availability Zone 2.</td>
</tr>
<tr>
<td><strong>Public Subnet 3 CIDR (PublicSubnet3CIDR)</strong></td>
<td>10.0.160.0/20</td>
<td>The CIDR block for the public (DMZ) subnet located in Availability Zone 3.</td>
</tr>
<tr>
<td><strong>Allowed Bastion External Access CIDR (RemoteAccessCIDR)</strong></td>
<td>Requires input</td>
<td>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).</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><strong>Key Pair Name (KeyPairName)</strong></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>

### Linux Bastion Amazon EC2 Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bastion AMI Operating System (BastionAMIOS)</strong></td>
<td>Amazon-Linux-HVM</td>
<td>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 <a href="https://aws.amazon.com/marketplace/seller-volume/">CentOS AMI in AWS Marketplace</a>.</td>
</tr>
<tr>
<td><strong>Bastion Instance Type (BastionInstanceType)</strong></td>
<td>t2.micro</td>
<td>The EC2 instance type for the bastion host instances.</td>
</tr>
<tr>
<td><strong>Number of Bastion Hosts (NumBastionHosts)</strong></td>
<td>2</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Redis Cluster Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cluster Name (ClusterName)</strong></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.</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><strong>The cluster domain will be</strong> ClusterName-firecamp.com. Each label in the domain name can be up to 63 bytes long.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CloudStax FireCamp Release Version</strong> (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><strong>Number of Container Worker Nodes</strong> (NumberOfNodes)</td>
<td>6</td>
<td>The number of container instances to launch in the cluster. This parameter must match the number of shards multiplied by the number of replicas per shard (Shards * ReplicasPerShard), based on the Redis settings in the next parameter group.</td>
</tr>
<tr>
<td><strong>Node AMI Type</strong> (NodeAMIOS)</td>
<td>Amazon-Linux-HVM</td>
<td>The Linux distribution for the AMI to be used for the service instances.</td>
</tr>
<tr>
<td><strong>Node Instance Type</strong> (NodeInstanceType)</td>
<td>r4.large</td>
<td>The EC2 instance type for the service instances.</td>
</tr>
<tr>
<td><strong>Delete Redis EBS Volumes</strong> (DeleteVolume)</td>
<td>false</td>
<td>Set this parameter to true if you want to delete the EBS volumes for Redis when you delete the stack.</td>
</tr>
</tbody>
</table>

**Redis Service Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Redis Release Version</strong> (RedisRelease)</td>
<td>4.0</td>
<td>The version of Redis to deploy. For details about the default release, see the Redis 4.0 release notes.</td>
</tr>
<tr>
<td><strong>Service Name</strong> (ServiceName)</td>
<td>Requires input</td>
<td>The Redis service name. This string consists of 1-58 characters (letters, numbers, or hyphens) and must start with a letter. The DNS name for the service will be ServiceName-\text{-index.ClusterName-firecamp.com}. Each label in the domain name can be up to 63 bytes long. Reserve 4 bytes for the index. ServiceName can be up to 58 bytes.</td>
</tr>
<tr>
<td><strong>Number of Redis Shards</strong> (Shards)</td>
<td>3</td>
<td>The number of Redis shards. The value of this parameter is also used to determine the number of container worker nodes. (See the NumberOfNodes parameter in the previous parameter group.)</td>
</tr>
<tr>
<td><strong>Number of Replicas Per Shard</strong> (ReplicasPerShard)</td>
<td>2</td>
<td>The number of replicas per shard. The value of this parameter is also used to determine the number of container worker nodes. (See the NumberOfNodes parameter in the previous parameter group.)</td>
</tr>
<tr>
<td><strong>Redis Node Memory Cache Size MB</strong> (MemoryCacheSizeMB)</td>
<td>256</td>
<td>The memory cache size of each Redis node, in MiB.</td>
</tr>
</tbody>
</table>
### Parameter label (name) | Default | Description
--- | --- | ---
**Redis Volume Type** (VolumeType) | gp2 | The EBS volume type for the Redis nodes. Options are: General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Throughput Optimized HDD (st1).
**Redis Volume Iops** (Iops) | 100 | The IOPS of the Redis volume. This parameter is used only when the **Redis Volume Type** parameter is set to io1.
**Redis Volume Size GB** (VolumeSizeGB) | 1 | The Redis volume size. If the Redis append-only file (AOF) is enabled, the volume size should be the memory cache size multiplied by 6. If AOF is disabled, the volume size could be the memory cache size multiplied by 2.
**Disable Append Only File** (DisableAOF) | false | Set this parameter to true to enable the Redis append-only file (AOF). For more information about the AOF file, see the Redis Security section.
**Redis AUTH Password** (AuthPass) | — | The Redis authentication (AUTH) password, which is a 1-255 character string. If you want to disable AUTH, leave the password field empty.
**Redis Eviction Policy** (MaxMemoryPolicy) | allkeys-lru | The eviction policy that will be used when the memory limit is reached. For more information about these policies, see Eviction Policies on the Redis website.
**New Name of Redis CONFIG Command** (ConfigCommandName) | — | The new name for the Redis **CONFIG** command, which reconfigures the server at run time. This string consists of 0-255 uppercase or lowercase letters. By default, the name is an empty string, which disables the **CONFIG** command. For more information, see the Redis Security section.

### 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-cache/ | 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 deployment 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><strong>Availability Zones</strong> (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. 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><strong>Number of Availability Zones</strong> (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><strong>VPC ID</strong> (VPCID)</td>
<td>Requires input</td>
<td>The ID of your existing VPC (e.g., vpc-0343606e).</td>
</tr>
<tr>
<td><strong>Private Subnet 1 ID</strong> (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><strong>Private Subnet 2 ID</strong> (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><strong>Private Subnet 3 ID</strong> (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><strong>Bastion Security Group ID</strong> (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><strong>Lambda Subnet CIDR</strong> (LambdaSubnetCIDR)</td>
<td>Requires input</td>
<td>The CIDR of the subnet where the Lambda function for Redis creation will be run. The range you specify must be in the form x.x.x.x/x (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><strong>Key Pair Name</strong> (KeyPairName)</td>
<td>Requires input</td>
<td>A 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.</td>
</tr>
</tbody>
</table>
### Redis 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. The cluster domain will be ClusterName-firecamp.com. Each label in the domain name can be up to 63 bytes long.</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>Number of Container Worker Nodes (NumberOfNodes)</td>
<td>6</td>
<td>The number of container instances to launch in the cluster. This parameter must match the number of shards multiplied by the number of replicas per shard (Shards * ReplicasPerShard), based on the Redis settings in the next parameter group.</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 service instances.</td>
</tr>
<tr>
<td>Node Instance Type (NodeInstanceType)</td>
<td>r4.large</td>
<td>The EC2 instance type for the service instances.</td>
</tr>
<tr>
<td>Delete Redis EBS Volumes (DeleteVolume)</td>
<td>false</td>
<td>Set this parameter to true if you want to delete the EBS volumes for Redis when you delete the stack.</td>
</tr>
</tbody>
</table>

### Redis Service Configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redis Release Version (RedisRelease)</td>
<td>4.0</td>
<td>The version of Redis to deploy. For details about the default release, see the Redis 4.0 release notes.</td>
</tr>
<tr>
<td>Service Name (ServiceName)</td>
<td>Requires input</td>
<td>The Redis service name. This string consists of 1-58 characters (letters, numbers, or hyphens) and must start with a letter. The DNS name for the service will be ServiceName-index.ClusterName-firecamp.com. Each label in the domain name can be up to 63 bytes long. Reserve 4 bytes for the index. ClusterName can be up to 58 bytes.</td>
</tr>
<tr>
<td>Number of Redis Shards (Shards)</td>
<td>3</td>
<td>The number of Redis shards. The value of this parameter is also used to determine the number of container worker nodes. (See the NumberOfNodes parameter in the previous parameter group.)</td>
</tr>
<tr>
<td>Number of Replicas Per Shard (ReplicasPerShard)</td>
<td>2</td>
<td>The number of replicas per shard. The value of this parameter is also used to determine the number of container worker nodes. (See the NumberOfNodes parameter in the previous parameter group.)</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>Redis Node Memory Cache Size MB</td>
<td>256</td>
<td>The memory cache size of each Redis node, in MiB.</td>
</tr>
<tr>
<td>(MemoryCacheSizeMB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redis Volume Type (VolumeType)</td>
<td>gp2</td>
<td>The EBS volume type for the Redis nodes. Options are: General Purpose SSD (gp2), Provisioned IOPS SSD (io1), and Throughput Optimized HDD (st1).</td>
</tr>
<tr>
<td>Redis Volume Iops (iops)</td>
<td>100</td>
<td>The IOPS of the Redis volume. This parameter is used only when the Redis Volume Type parameter is set to io1.</td>
</tr>
<tr>
<td>Redis Volume Size GB (VolumeSizeGB)</td>
<td>1</td>
<td>The Redis volume size. If the Redis append-only file (AOF) is enabled, the volume size should be the memory cache size multiplied by 6. If AOF is disabled, the volume size could be the memory cache size multiplied by 2.</td>
</tr>
<tr>
<td>Disable Append Only File (DisableAOF)</td>
<td>false</td>
<td>Set this parameter to true to enable the Redis append-only file (AOF). For more information about the AOF file, see the Redis Security section.</td>
</tr>
<tr>
<td>Redis AUTH Password (AuthPass)</td>
<td>—</td>
<td>The Redis authentication (AUTH) password, which is a 1-255 character string. If you want to disable AUTH, leave the password field empty.</td>
</tr>
<tr>
<td>Redis Eviction Policy (MaxMemoryPolicy)</td>
<td>allkeys-lru</td>
<td>The eviction policy that will be used when the memory limit is reached. For more information about these policies, see Eviction Policies on the Redis website.</td>
</tr>
<tr>
<td>New Name of Redis CONFIG Command (ConfigCommandName)</td>
<td>—</td>
<td>The new name for the Redis CONFIG command, which reconfigures the server at run time. This string consists of 0-255 uppercase or lowercase letters. By default, the name is an empty string, which disables the CONFIG command. For more information, see the Redis Security section.</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>Quick Start S3 Key Prefix (QSS3KeyPrefix)</td>
<td>quickstart-cloudstax-cache/</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 Cache for Redis cluster is ready.

9. Use the information displayed in the **Outputs** tab for the stack to view the resources that were created, as shown in Figure 2.

![Figure 2: Stack outputs](image)

**Step 3. Test the Deployment**

After you deploy the Redis 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 Redis service and the CloudStax FireCamp management service running. For example, if you set the **Cluster Name** parameter to `t1` and the **Service Name** parameter to `myredis`, the Amazon ECS console displays a screen similar to Figure 3.

![Figure 3](image)
2. Deploy an EC2 instance in the **RedisAccessSecurityGroup** security group in the same VPC, download **Redis version 4.0**, and test the set/get key with the Redis command line interface (`redis-cli`) as follows. (In this example code, the cluster name is `mycluster` and the Redis service name is `myredis`.)

```bash
# set key
redis-cli -a $password -h myredis-0.mycluster-firecamp.com -c set key1 value1

# get key
redis-cli -a $password -h myredis-0.mycluster-firecamp.com -c get key1
```

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

**Best Practices Using CloudStax Cache for Redis on AWS**

**Easy Access**

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

Let’s assume that the cluster name is mycluster, the Redis service name is myredis, and the Redis cluster has three shards and two replicas per shard. Redis will use myredis-0.mycluster-firecmap.com to myredis-5.mycluster-firecmap.com as DNS names, and an application can access Redis by using these DNS names.

**High Availability**

We recommend that you deploy the Redis cluster with at least three shards (which is the default). The single shard works in primary-secondary mode, which includes one primary node and multiple read-only nodes (read replicas). Currently, CloudStax Cache for Redis primary-secondary mode doesn’t support automatic failover. If the primary node goes down, the read replica isn’t promoted to primary node automatically.

A Redis cluster requires at least three shards, as discussed in the Redis documentation. Each shard should have at least one primary node and one read replica to tolerate a shard primary node failure. A Redis cluster can handle a node failure when at least a majority of primary nodes exist, and a read replica exists for every unreachable primary node. (For details, see the Redis documentation.) If the majority of primary nodes are down, the read replicas will not become primary nodes automatically—you have to manually run the cluster failover command to promote a read replica to become the new primary node.

The Redis cluster should be deployed in three Availability Zones. CloudStax Cache for Redis distributes the primary nodes to all Availability Zones, so when one zone goes down, the Redis cluster will still have the majority of primary nodes and can promote a read replica to primary node automatically.

| Important | You might have to run a [manual cluster failover command](#) in some scenarios. Currently, a Redis cluster doesn’t automatically fail back when the original primary node rejoins the cluster after a node failover (see a [discussion about this behavior](#)). So if one Availability Zone (such as us-west-2a) goes down, the primary node of one shard will fail. The read replica of this shard in another Availability Zone (such as us-west-2b) will be promoted as the new primary node. When us-west-2a comes back online, the original primary node will join back as a read replica, so us-west-2b will have two primary nodes. If us-west-2b goes down, the Redis cluster will lose the majority of primary nodes and become read-only. You would have to manually run the cluster failover command to promote the read replica to a primary node. |
Data Persistence
Redis periodically saves snapshots of a database (.rdb file) to disk and uses default save configurations. For example a database will be saved:

- After 900 sec (15 minutes) if at least 1 key is changed
- After 300 sec (5 minutes) if at least 10 keys are changed
- After 60 sec (1 minute) if at least 10,000 keys are changed

If the append-only file option (AOF) is enabled, the key change will always be written to the AOF file, and Redis will fsync the AOF file every second. By default, CloudStax Cache for Redis enables the AOF to minimize the chance of data loss in case Redis stops working.

If you are using Redis as a cache, you could disable AOF when creating the Redis service. Each Redis member will still have one volume to store data. If the system reboots, Redis could quickly rebuild the cache by reading from the volume.

If AOF is disabled, the storage size could be twice the memory size. If AOF is enabled, much more storage is required. For standard usage scenarios, Redis recommends the storage size to be six times the node’s RAM size for Redis Enterprise Software (Redis Pack). Even more storage is required for heavy write scenarios (see details).

Data Backup
If AOF is disabled, you can connect to the primary node or read replica of all shards. Run the BGSAVE command, wait until the save operations are complete, and then back up the data by taking the snapshot for Redis EBS volumes manually. If AOF is enabled, you can take the snapshot for Redis EBS volumes directly.

Redis Configuration
Memory
If you estimate that the total size of all your items will be 12 GB in a three-shard Redis cluster, each shard will serve 4 GB of data. The Redis replication buffer is set to 512 MB, and 1 GB is reserved for the operating system. The Redis node should have at least 5.5 GB of memory. When Redis persists the memory data to disk, it may take up to 4 GB of memory to serve the incoming write operations during data persistence. If your application is write-heavy, you should double the per-node Redis memory to at least 8 GB, so the node memory is at least 9.5 GB.

You should always set a maximum limit to memory usage when you create the Redis service. Otherwise, Redis will allocate memory as long as the operating system allows it to, which might cause memory to get swapped and might slow down Redis unexpectedly.
System Parameters
CloudStax follows Redis best practices and sets system parameters accordingly. For example, somaxconn and tcp_max_syn_backlog are increased to 512, and vm.swappiness and vm.overcommit_memory are set to 1. 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 Redis instances are deployed in private subnets and can be accessed in two ways:

- By connecting to the bastion host instance through an SSH terminal.
- From AWS resources (such as Amazon EC2) that you might have in the RedisAccessSecurityGroup security group, or that you might launch by using the security group. You may include your application instance in this security group.

Redis Security
This deployment supports Redis authentication (AUTH). To enable AUTH, set a password by using the Redis AUTH Password (AuthPass) parameter in the template. If AUTH is enabled, clients must send the AUTH command followed by the password before processing any other commands. If you don’t want to enable AUTH, leave the AuthPass parameter empty when you deploy the Quick Start.

In this deployment, possibly harmful commands are disabled or can be renamed. The following commands are disabled:

- **FLUSHALL** (removes all keys from all databases)
• **FLUSHDB** (removes all keys from the current database)

• **SHUTDOWN**

You can rename the **CONFIG** command (which reconfigures the server at run time) by using the **New Name of Redis CONFIG Command** (**ConfigCommandName**) parameter during deployment. Setting a new name might be useful in some conditions. For example, if you hit a latency issue, you could enable the latency monitor to collect data by using the command **CONFIG SET latency-monitor-threshold <milliseconds>**. Setting the **ConfigCommandName** parameter to an empty string will disable the **CONFIG** command.

**Troubleshooting**

**Q.** I encountered a CREATE FAILED error when I launched the Quick Start.

**A.** If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack’s state will be retained and the instance will be left running, so you can troubleshoot the issue. (Look at the 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](https://aws.amazon.com/roaming-support/troubleshooting/) 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](https://aws.amazon.com/roaming-support/troubleshooting/).

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

**A.** For questions about using CloudStax Cache for Redis, use the **Issues** section of the [GitHub repository for CloudStax FireCamp](https://github.com/cloudstax/firecamp) or the [FireCamp mailing list](https://groups.google.com/forum/#!forum/firecamp).
If the issue is associated with Redis, get help from the [GitHub repository for Redis](https://github.com/redis) or the [Redis mailing list](https://lists.redislabs.com/listinfo/redis-discuss).

**GitHub Repository**

You can visit the [GitHub repository](https://github.com/redis) for this Quick Start to download the templates and scripts, to post your comments, and to share your customizations with others.

**Additional Resources**

**AWS services**

- Amazon DynamoDB
  [https://aws.amazon.com/documentation/dynamodb/](https://aws.amazon.com/documentation/dynamodb/)
- Amazon EBS
- Amazon EC2
  [https://aws.amazon.com/documentation/ec2/](https://aws.amazon.com/documentation/ec2/)
- Amazon ECS
  [https://aws.amazon.com/documentation/ecs/](https://aws.amazon.com/documentation/ecs/)
- Amazon Route 53
- Amazon VPC
  [https://aws.amazon.com/documentation/vpc/](https://aws.amazon.com/documentation/vpc/)
- AWS Auto Scaling
  [https://aws.amazon.com/documentation/autoscaling/](https://aws.amazon.com/documentation/autoscaling/)
- AWS CloudFormation
  [https://aws.amazon.com/documentation/cloudformation/](https://aws.amazon.com/documentation/cloudformation/)

**Redis**

- Redis Documentation
  [https://redis.io/documentation](https://redis.io/documentation)
- Redis Cluster Tutorial
  [https://redis.io/topics/cluster-tutorial](https://redis.io/topics/cluster-tutorial)
- Redis Cluster Spec
  [https://redis.io/topics/cluster-spec](https://redis.io/topics/cluster-spec)
• Redis Data Persistence
  https://redis.io/topics/persistence

• Redis Security
  https://redis.io/topics/security

CloudStax

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

• CloudStax Cache for Redis
  https://github.com/cloudstax/firecamp/tree/master/catalog/redis

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><strong>February 2018</strong></td>
<td>Initial publication</td>
<td>—</td>
</tr>
</tbody>
</table>
© 2018, Amazon Web Services, Inc. or its affiliates, 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/](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.