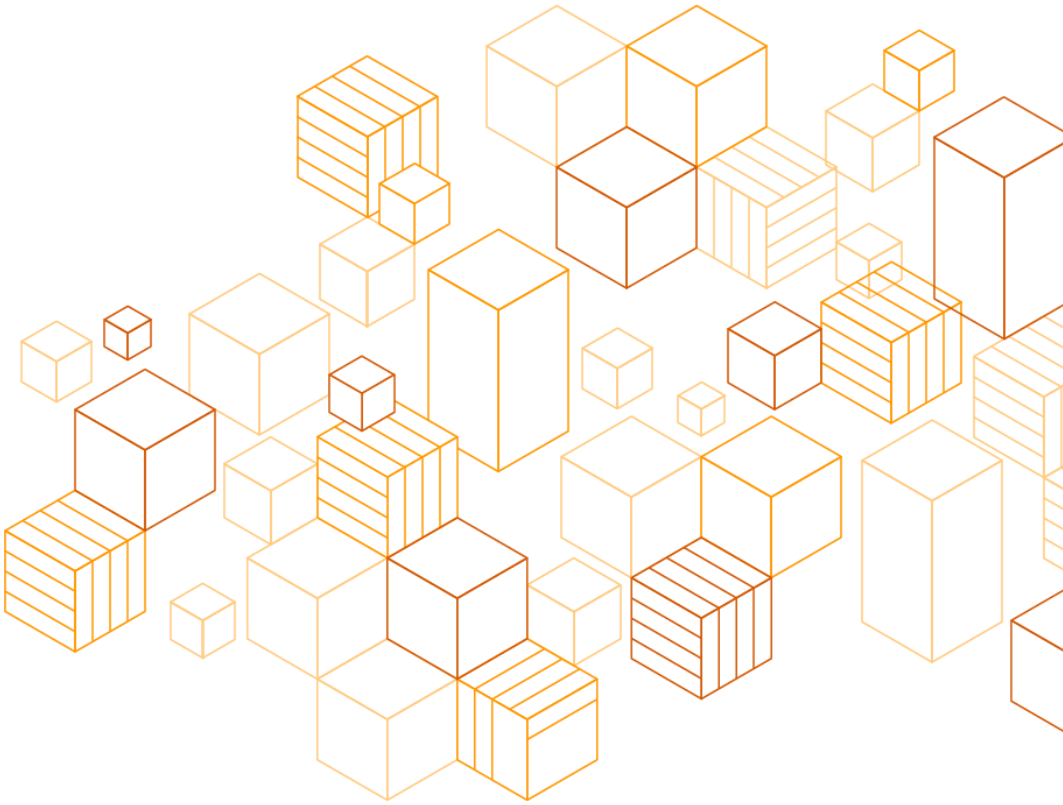
Couchbase Server 6.6 on AWS

Best Practice Guide

October 2020



Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. 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.

Contents

Abstract

This whitepaper provides an overview of implementing Couchbase Server Enterprise Edition 6.6 on the AWS Cloud, including best practices and implementation characteristics such as performance, durability, and security. This paper focuses on AWS services and features relevant to Couchbase Server that help ensure scalability, high availability, and disaster recovery. Note that this paper is not a comprehensive introduction to AWS nor does it cover the full range of capabilities of Couchbase.

Introduction

Couchbase Server combines NoSQL database with the ease and power of SQL in one platform. Couchbase features a memory-first architecture, key value store and document database capabilities making it appropriate for even the most business-critical and high-performance applications.

Couchbase Server can provide consistent high-performance access to information, even during database upgrades and system-level administration tasks. Couchbase Server is nonhierarchical, shared-nothing architecture with a single server type. Because of this flat design, you can easily resize your Couchbase Server system by adding and removing instances, even while the system is running and servicing requests.

Couchbase Server Deployment Options

You can deploy Couchbase Server with the AWS Marketplace, or AWS CloudFormation templates.

Deployment with AWS Marketplace

The recommended and often easiest way to deploy and license Couchbase on AWS is to use AWS Marketplace. This option provides a convenient way for customers to find, purchase, and immediately start using Couchbase Server in the AWS Cloud.

The Couchbase Server and Sync Gateway product listing offers a simple click-to-deploy web interface. The listing can be used to deploy a combination of Couchbase Server or Couchbase Sync Gateway nodes and have them up and running quickly.

Couchbase ships three products on AWS Marketplace:

- Couchbase Server Enterprise Edition is recommended for commercial production systems running Couchbase Server
- Couchbase Sync Gateway, part of Couchbase Mobile stack, is a secure web gateway for data access and synchronization across mobile clients and Couchbase Server
- Couchbase Autonomous Operator enables you to run Couchbase as a stateful database application next to your microservices applications on Amazon EKS

All the three products have two production pricing options: Bring Your Own License (BYOL), where you procure a license from Couchbase directly; and Hourly Pricing, where you're billed through your AWS account via AWS Marketplace.

Apart from these three offerings, Couchbase Cloud is a fully managed Database-as-a-Service (DBaaS) that can be deployed in your own Amazon VPC. Couchbase Cloud includes a control plane, that automates operational tasks like deployment, scaling, recovery, and upgrades.

Additional Deployment options

- AWS CloudFormation Templates - Couchbase provides several AWS CloudFormation templates on GitHub to assist with provisioning of Couchbase Server on AWS.

Couchbase Architecture

The design of your Couchbase Server installation on Amazon EC2 depends mainly on the scale at which you're trying to operate. If you're experimenting with the framework on your own for a private project, we recommend creating a cluster with a minimum of three instances. Beyond three instances, the actual sizing depends on the overall amount of data that you want to store and on your performance requirements. See sizing guidelines in the Couchbase documentation for details.

Application servers communicating with Couchbase cluster are aware of the individual node topology for data and services. For example, if a data needs to be read, the application server can access the Couchbase server node where it resides directly because it knows about the overall cluster configuration.

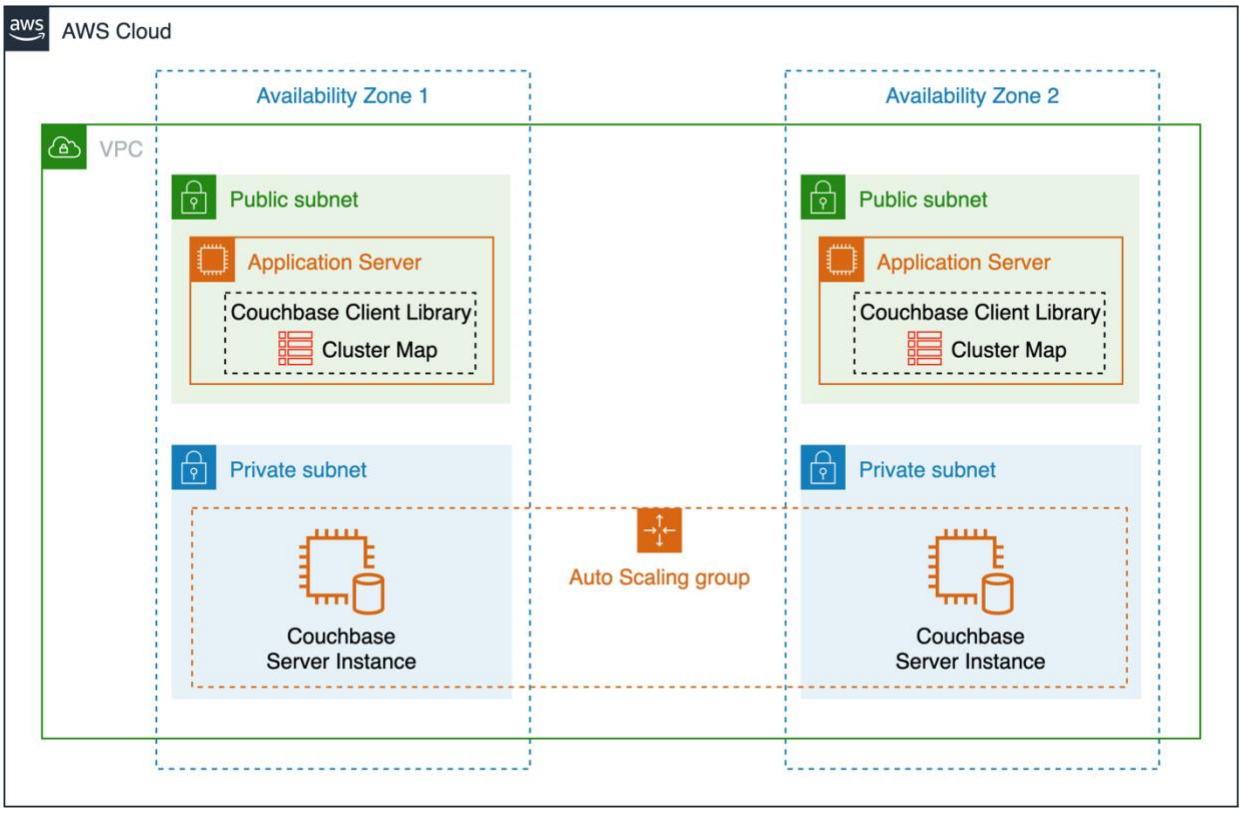


Figure 1: Application Servers are Couchbase Cluster topology aware

As demand for your application increases, you can add instances to your cluster to cope with additional compute, memory or I/O requirements. Moreover, with Couchbase Multi-Dimensional Scaling (MDS), individual database services can be scaled separately on Amazon EC2 instance types optimized for that service.

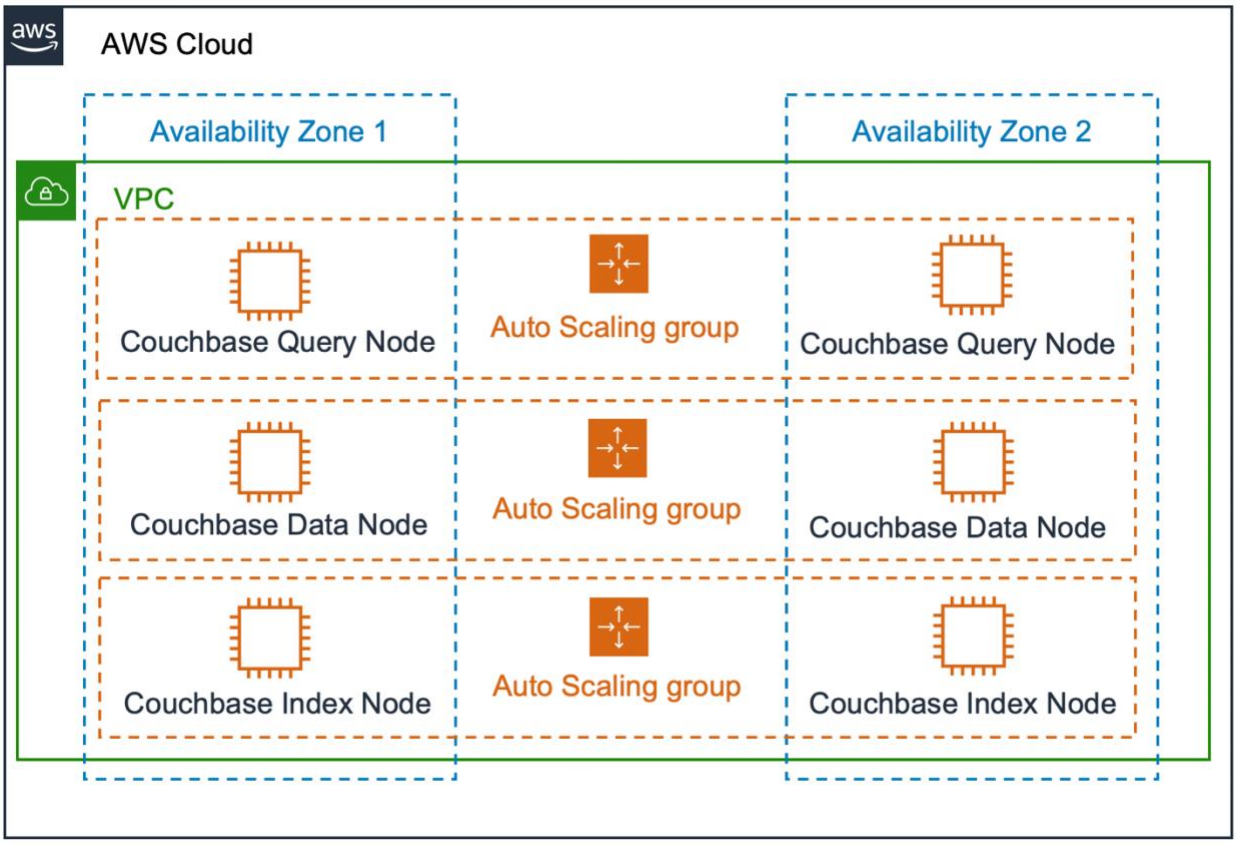


Figure *2*: Multi-Dimensional Scaling example

Building Blocks

This section covers Couchbase Server Cluster components and concepts.

Buckets and vBuckets

Couchbase Server uses buckets as a logical container of data. Not to be confused with Amazon S3 buckets, Couchbase buckets are used to segregate the configuration and operational handling of data in terms of cache allocation, indexing, and replication. For example, you can configure one bucket with replicas for critical data and another bucket for session data where replicas would not be required. The types of bucket most frequently used are *Couchbase* and *Ephemeral* buckets. *Couchbase* buckets exist both in memory and on disk. *Ephemeral* buckets exist only in memory. Buckets can be configured to maintain data in compressed format even in memory, to maximize resource-effectiveness.

Underlying the bucket structure is the vBucket system. The vBucket system is an abstraction layer that allows the information in the bucket to be distributed and sharded across the different instances in the Couchbase Server cluster.

Users can set replication on a bucket-by-bucket basis. Replica data is automatically sharded into replica vBuckets and distributed across the instances in the cluster. If an instance fails, replica vBuckets of the data stored on failed instance are enabled, and clients will communicate directly with the replica instances instead of the original.

Memory First

Couchbase Server provides a fully integrated caching layer to facilitate high-speed data-access. A write associated with *Couchbase* bucket is first saved in memory and then placed on the disk queue and replication queue. Memory quotas are actively managed by the Couchbase server to maintain consistent free space in the caching layer. Infrequently used items are ejected to disk and can be reacquired when needed.

Persistence

Couchbase Server provides persistence, whereby items are stored on disk as well as in memory. This also allows data sets to exceed the size permitted by existing memory resources.

Data is stored on disk in compressed format. Synchronized, multi-threaded readers and writers provide simultaneous, high-performance operations for data on disk.

Cluster Manager

The Cluster Manager runs on all the nodes of a cluster, maintaining essential per-node processes, and coordinating cluster-wide operations. Cluster Manager maintains the mapping of vBuckets and to nodes known as vBucket map. As new vBuckets on new nodes become active, the Cluster Manager ensures that the new vBucket map and cluster topology are communicated to all nodes and clients.

Cluster Manager also maintains the mapping of services to nodes known as cluster map. Couchbase client SDKs automatically load-balance across services, using the cluster map.

Cross Datacenter Replication (XDCR)

Cross datacenter replication provides an easy way to replicate active data to multiple, AWS regions either for disaster recovery or to bring data closer to its users. XDCR can

be set up on a per bucket basis. Depending on your application requirements, you might want to replicate only a subset of the data in Couchbase Server between two clusters. With XDCR you can selectively pick which buckets to replicate between two clusters in a unidirectional or bidirectional fashion.

Deployment Recommendations

Each deployment is unique and depends primarily on the workload assigned to the Couchbase cluster. The following guidelines are meant to address best practices and should be used only as guidance for your Couchbase Server deployment on AWS.

Security

We recommend using security facilities provided by Couchbase Server that cover:

- **Identities**: To access Couchbase Server, administrators and applications must be authenticated. Along with local identity management, Couchbase Server supports LDAP authentication with existing OpenLDAP or AD identity source, so that you can use your existing users and groups. LDAP identity source can be configured using Couchbase web console or Couchbase CLI. Authentication can be done using credentials, or by means of client-authentication using X.509 certificates.

  We recommend that you strengthen the default password-policy using Couchbase CLI command `setting-password-policy` and rotate the passwords on regular basis.

- **Permissions**: Role-Based Access Control (RBAC), whereby access-privileges are assigned to fixed roles that are assigned to administrators, users and applications.

  We recommend to grant only the minimum level of access needed by the specific roles.

- **Auditing**: Audit logs are optional on Couchbase Server. You can enable audit logs to capture detailed record of actions performed on Couchbase Server. Audit logs are stored on per node basis on the node filesystem. Based on your auditing needs, manually consolidate these logs on Amazon CloudWatch or Amazon S3 and plan for how long you need to store the logs.

- **Encryption**: Couchbase Server supports encryption of data in transit, for communication between cluster-nodes, clusters, and external clients, using TLS. We recommend using TLS 1.2 version and plan for certificate rotation well before certificates expire.

  Amazon EBS encryption is recommended for data at rest, it provides a straight-forward encryption solution. Amazon EBS volumes support Encryption with same IOPS performance on encrypted volumes as unencrypted volumes, with a minimal effect on latency. By default, Amazon EBS uses the AWS managed customer master key. However, you can specify a customer managed CMK in AWS Key Management service (KMS) that you create and manage.

Least Privileges Access Policies

There are two types of permissions required to operate Couchbase cluster securely on AWS:

- The IAM role for Amazon EC2 which is assigned to the EC2 instances to perform internal scaling and maintenance operations.
- The IAM role for the Couchbase administrators to manage and deploy changes.

We recommend that you create these IAM roles based on the least privilege access policy and that you avoid adding an IAM policy to an existing IAM user or group.

The policy should be scoped down to the specific actions required and not allow all actions. For example, Avoid using `ec2:*` , and use specific EC2 actions. Scope actions by resource wherever possible, specify the resource ARN or a prefix instead of using `Resource:"*"`.

Network Security

Amazon VPC enables you to create an isolated portion of the AWS Cloud. You can further segment your VPC into subnets of IP ranges grouping similar instances together for simplifying your design and security. Each subnet is associated with a network access control list (network ACL) and route table.

You can create the Couchbase Cluster in private subnets and use the network ACL as a stateless firewall to narrow the scope of traffic allowed between layers.

AWS provides another level of network security at the instance level with VPC security groups. Security groups restrict access to ingress and egress traffic at protocol and port level. Make sure to create appropriate security groups for your Couchbase cluster.

See Couchbase Server Ports in the Couchbase documentation for details on ports.

You can create the Couchbase cluster security group and add ingress rules for custom TCP with self-reference to the security group ID for node-to-node communication ports and additional ingress rules for custom TCP with application server security group ID for node-to-client communication ports.

Table 1: Couchbase Security Group Inbound Rules

Type	Protocol	Ports	Source
SSH	TCP	22	Bastion Host CIDR or Remote Access CIDR
Custom TCP	TCP	4369	self-reference to security group ID
Custom TCP	TCP	8091-8094	self-reference to security group ID
Custom TCP	TCP	9100-9105	self-reference to security group ID
Custom TCP	TCP	9110-9118	self-reference to security group ID
Custom TCP	TCP	9120-9122	self-reference to security group ID
Custom TCP	TCP	9130	self-reference to security group ID
Custom TCP	TCP	9999	self-reference to security group ID
Custom TCP	TCP	11209-11210	self-reference to security group ID
Custom TCP	TCP	21100	self-reference to security group ID
Custom TCP	TCP	11207	self-reference to security group ID
Custom TCP	TCP	18091-18094	self-reference to security group ID

Type	Protocol	Ports	Source
Custom TCP	TCP	19130	self-reference to security group ID
Custom TCP	TCP	21150	self-reference to security group ID
Custom TCP	TCP	8091-8096	application server security group ID
Custom TCP	TCP	11210	application server security group ID
Custom TCP	TCP	11207	application server security group ID
Custom TCP	TCP	18091-18096	application server security group ID

Cost and Performance optimization

The cost of Couchbase Server on AWS varies depending on your deployment. The following guidelines and design considerations can help you keep your costs low while also meeting your business requirements.

Storage

Couchbase Server uses storage disk for persistence. Amazon Elastic Block Store (Amazon EBS) is the recommended persistence storage for Couchbase Server on AWS. Couchbase require high IOPS in most cases, you should optimize your EBS volumes to the IOPS you expect your system to require. AWS offers tools that can help you optimize block storage. Amazon CloudWatch automatically collects a range of data points for EBS volumes and lets you set alarms on volume behavior. You can then, use the elastic volumes feature to change the size or volume type, or adjust IOPS performance without detaching the volume.

Amazon EBS

The two main Amazon EBS volume types recommended for Couchbase are **General Purpose SSD (gp2)** and **Provisioned IOPS SSD (io1 & io2)**.

General Purpose SSD (gp2) volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods. These volumes have a consistent baseline performance of 3 IOPS/GiB (minimum 100 IOPS) to max IOPS of 16,000 per volume (based on 16 KiB I/O size), along with 250-MiB/s throughput. This option provides a good balance of performance and cost for most deployments.

> Note: It is not recommended that you exceed 1TB for data volumes. Large volumes can lead to overly dense nodes that suffer from long rebuild times. Instead, scale horizontally by adding Couchbase data nodes.
>
> EBS gp2 volume of 1000 GiB has max IOPS of 3,000 (16 KiB I/O size)
>
> EBS io1 volume of 1000 GiB has max IOPS of 50,000 (16 KiB I/O size)
>
> EBS io2 volume of 1000 GiB has max IOPS of 64,000 (16 KiB I/O size)

A provisioned IOPS SSD volume is the best choice for maximizing performance with consistent and predictable results on a single volume for higher IOPS requirement, up to 64,000 IOPS (based on 16 KiB I/O size), along with 1000-MiB/s throughput.

The size of I/O operations affects IOPS. If the I/O chunks are very large, you might get fewer IOPS than you provisioned because of the 1000 MiB/s per volume throughput limit. If your I/O size is consistently large, consider joining multiple EBS volumes together in a RAID 0 configuration to increase available throughput. For more information, see RAID Configuration on Linux. Be sure to use an EBS-optimized instance that supports enough IOPS and throughput for all the attached EBS volumes.

When you architect your cluster, make sure you know which nodes will require higher IOPS and which nodes can utilize burst IOPS with gp2 to reduce the overall cost of the cluster.

> **Note:** Amazon EBS volumes can also benefit from the Elastic Volumes. Elastic Volumes allow you to change the type, size and IOPS of your EBS volumes with no downtime.

Amazon EC2 Instance Store

Instance Store devices are ephemeral in nature and are attached to the Amazon EC2 instance as long as it is not terminated, stopped, or experienced a failure in the underlying hardware. The NVMe-based SSD Instance Store devices provide very low latency and higher IOPS. Using Instance Store requires the cluster to be built in such a

way that data loss is either mitigated or not critical – this is an edge case and should not be your standard configuration option.

Compute

Couchbase Server works best with multi-core high memory EC2 instances. Typically, Amazon EC2 M5 instances, the General Purpose EC2 instances built on the AWS Nitro System, provide the best balance between price and performance for most production environment.

Memory-optimized Amazon EC2 R5 instances and compute-optimized Amazon EC2 C5 instances, built on the AWS Nitro System, also align well with Couchbase Server for production environment. For testing and development, consider General Purpose - Burstable Amazon EC2 T3 instances.

Amazon EC2 M5 Instances are general purpose instances that provide a good mix of CPU cores and Memory with an Amazon EBS backed storage. The M5 instance types include Intel based instances, AMD based instances and also instances that are optimized for networking and storage. These instances provide the best cost and performance for common deployments where the I/O performance is not high.

Amazon EC2 C5 Instances are optimized for compute and provide cost-effective high performance at a low price per compute ratio. They are backed by Amazon EBS volumes similar to M5 instances and support enhance networking. These instances are recommended for smaller deployments of Couchbase, testing environments and high CPU bound databases.

Amazon EC2 R5 Instances are memory optimized instances that are ideal for memory-bound workloads, including high performance databases, and provide more memory for a lower cost for a given vCPU. These types of instances are recommended for workloads heavily utilizing the in-memory feature of Couchbase.

Amazon EC2 M5n, C5n and R5n instance types are optimized for network throughput and are recommended for workloads that require high network throughput.

You can optimize Couchbase Cluster with the Couchbase Multi-Dimensional Scaling (MDS) feature, and then isolate individual database services and scale separately on Amazon EC2 instance type optimized for that service. For example, the compute-optimized Amazon EC2 C5 instance is an ideal choice for Couchbase Query Service which requires more processing capacity whereas the memory-optimized Amazon R5 instance is a good choice for Couchbase Data Service due to the increased memory.

Couchbase sizing guidelines and the table of Couchbase components and their resource usage under normal operations can be used as a starting point in optimizing Amazon EC2 instance types and Amazon EBS storage.

Table 2: Couchbase component resource usage

Couchbase component	Resource Usage			
	CPU	**Memory**	**Disk**	**Network**
Data service	Low	High	Workload Dependent	Workload Dependent
Query	High	Medium	Low	Workload Dependent
Indexing	Medium	High	High	Workload Dependent
Views	High	Low	High	Low
Analytics	High	High	Workload Dependent	High
Eventing	High	Low	Low	Workload Dependent
XDCR	Low	Low	Low	High
Rebalancing	Medium	High	High	High

Reliability

Couchbase on AWS is highly scalable and reliable. Follow these reliability design practices for your Couchbase Server on AWS deployment.

Durability

Data Durability refers to the fault tolerance and persistence of data in the face of software or hardware failure. Couchbase clients can specify durability requirements for data-write, which ensures the specified document is updated on multiple nodes in memory and/or disk locations across the cluster; before considering the write to be committed.

Based on the application needs, Couchbase clients must choose between the higher throughput using asynchronous regular writes and greater data-protection provided by synchronous durable writes. To optimize performance, its recommended to select durable writes only for critical data where data loss is not acceptable.

High availability

To meet the demands of high availability requirements, all Couchbase operations can be done while the system remains online, without requiring modifications or interrupting running applications. Built-in fault tolerance mechanisms protect against downtime caused by arbitrary unplanned incidents, including server failures. Replication is an important mechanism that increases system availability.

Data Replication

Up to three replica buckets can be defined for every bucket. Each replica itself is implemented as 1024 replica vBuckets, like the original bucket implementation of 1024 active vBuckets. To ensure maximum availability of data in case of node failures, the master services for the cluster calculate and implement the optimal vBucket distribution across available nodes. Replica vBuckets receive a continuous stream of mutations from the active vBucket, and are thereby kept constantly up to date. Typically, only active vBuckets are accessed for read and write operations. In case of a node failure, automatic failover is triggered and replica vBuckets, corresponding to the active vBuckets on the failed node, are promoted to be active.

As Couchbase Server Nodes can tolerate node failures, a common approach to is to deploy the server nodes in AWS Auto Scaling group with a set desired capacity.

Server group awareness

Server group awareness provides enhanced availability using Availability Zones. Specifically, it protects a cluster from large-scale infrastructure failure such as an Availability Zone failure, through the definition of *groups*. We recommend using Couchbase Web Console to define Couchbase server group to contain all the subset of nodes in a single Availability Zone. Following group definition and rebalance, the active vBuckets for any defined bucket are located on one *group,* while the corresponding replicas are located on another *group*. This allows group failover to be enabled, so that if an entire Availability Zone (for example us-east-1a) goes offline, the associated groups replica vBuckets, which remain available on another Availability Zone (for example us-east-1b), can be automatically promoted to active status.

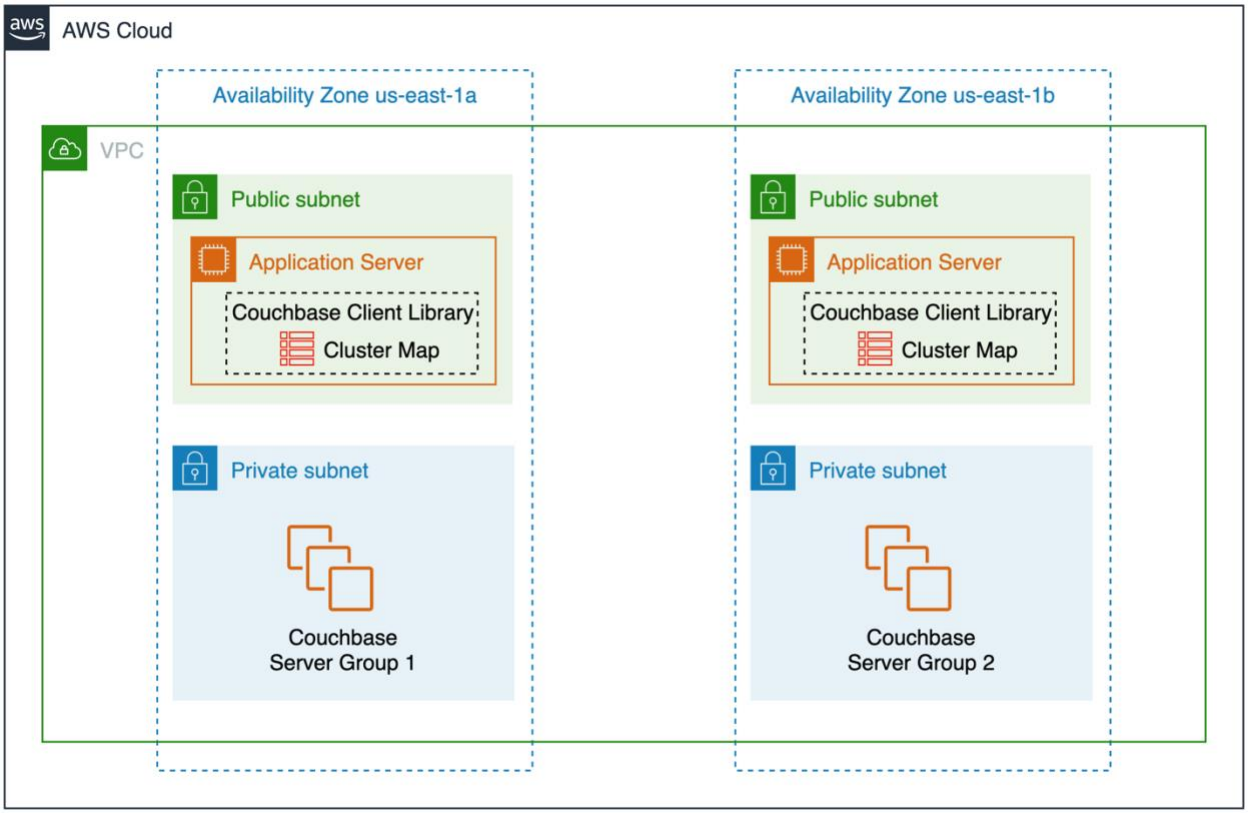


Figure 3: Couchbase Cluster Multi-AZ deployment

Even for single Availability Zone deployments of Couchbase Server on AWS, you can use the Placement groups - Partition option, to help reduce the likelihood of correlated hardware failures for your application.

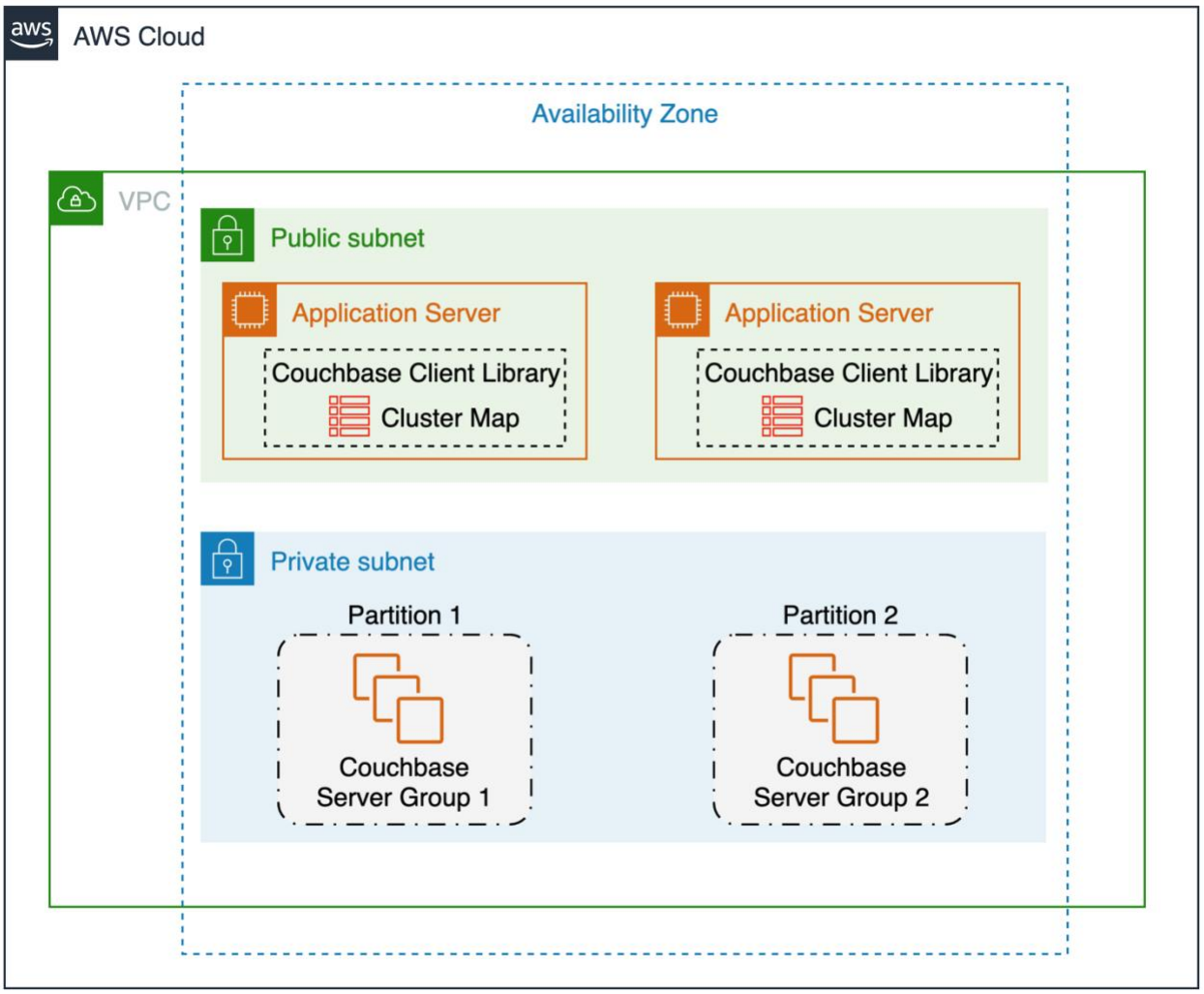


Figure 4: Couchbase Cluster Single AZ deployment using Placement group

You should consider your business requirements when deploying in a single Availability Zone as it can introduce single point of failure if the Availability Zone fails or degrades in performance. The recommended practice is to partition Couchbase Cluster across Availability Zones, to better isolate and protect from issues such as power outages, lightning strikes, tornadoes, earthquakes, and more.

Note: To ensure proper failover, use at least three nodes for the data service and two functional nodes for the query and index services.

Cross Datacenter Replication (XDCR)

You can take the high availability of multi-AZ deployment and expand it to Multi-Region deployment. The recommended cluster configuration to isolate and protect from Region failure is to have separate Couchbase clusters in each Region and have XDCR replication set up between them.

In this configuration, each cluster is managed independently but uses XDCR replication to achieve the highest redundancy between clusters. Connectivity can be provided via XDCR over the public internet using mutual TLS.

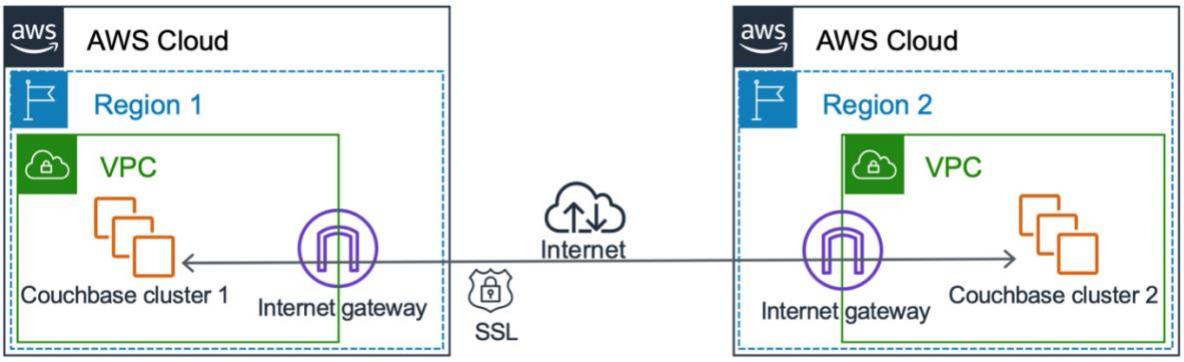


Figure 5: XDCR between Regions over public internet with mutual TLS

In Regions where AWS Transit Gateway is available, we recommend using AWS Transit Gateway so that XDCR traffic never passes through public internet. XDCR can seamlessly use AWS Transit Gateway when specific network routes are properly configured to traverse between the two regions.

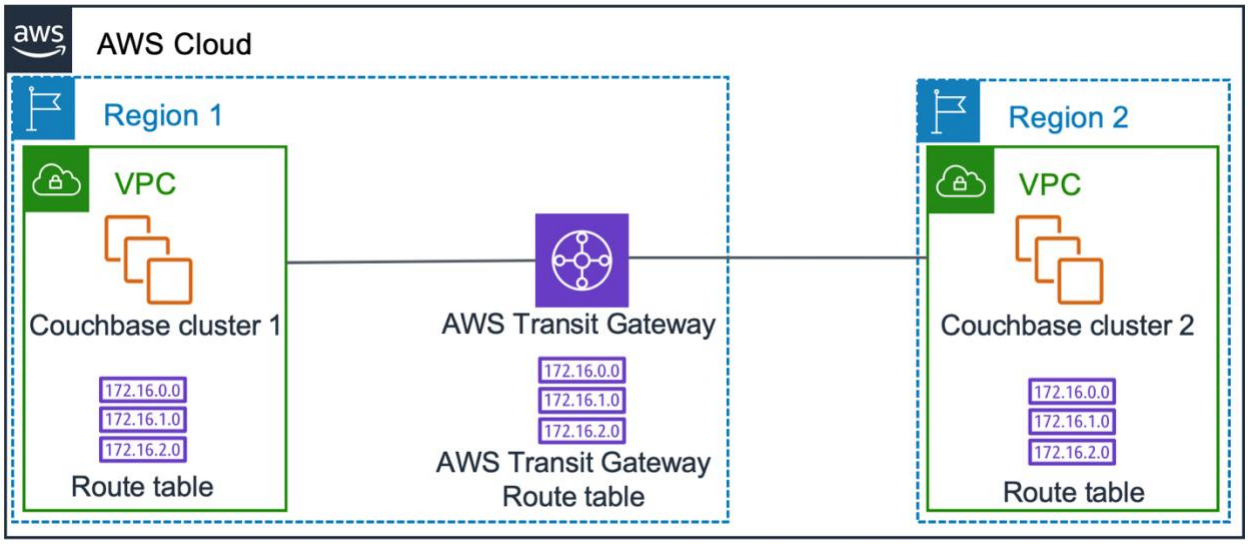


Figure 6: XDCR between multiple regions with AWS Transit Gateway

Some benefits of XDCR include the following:

High Throughput: Couchbase splits all data in a bucket into 1024 vBuckets. When moving data, vBuckets in source and destination pair up and independently move data without any centralized funnel. This architecture allows greater throughput and lower latency for data movement between clusters both for the initial synchronization as well as the continuous streaming of data.

Powerful Topology Support: XDCR provides a building block in the form of unidirectional replication. You can use unidirectional replications to construct more complex topologies including circular, cascading/chained, and known topologies like hub-and-spoke for data aggregation scenarios.

Backup and Restore

In a production environment, as part of an overall disaster recovery (DR) plan, you must plan to back up your entire cluster periodically. This plan helps ensure that you do not lose information in the event of a serious failure and minimizes data inconsistency when a restore is required. We recommend Couchbase Backup Manager cbbackupmgr, an enterprise-grade backup and restore utility. It enables backup and restore of data, indexes, and bucket configurations; and the restoration of such state to a running cluster.

We recommend Couchbase Backup Manager to do one full backup and then subsequent incremental backups over time directly to Amazon S3. Amazon S3 provides a highly durable storage infrastructure to store and manage the backup archive files.

Although it's possible to have cbbackupmgr coexist in the same S3 bucket as other files, we recommend using a bucket to which cbbackupmgr has exclusive access.

> **Note:** Amazon EBS snapshot is not recommended for Couchbase backup and restore.

Operational Excellence

Infrastructure-as-Code (IaC)

When using code to deploy Couchbase, whether it be AWS CloudFormation, Terraform, AWS Cloud Development Kit (AWS CDK) or any other IaC solution, each change to the deployment characteristics should be reflected in the code and treated the same way as any other code with a clear continuous integration/continuous deployment (CI/CD) process. Some examples:

- Test changes before deploying them in production
- In case of failures, keep versions of the code to track changes and revert back to older code
- Automate processes around code testing, deployment, and storage

AWS Developer Tools include AWS CodePipeline, AWS CodeCommit, AWS CodeBuild, and AWS CodeDeploy to manage the IaC deployment lifecycle.

Performance Efficiency

Monitoring

Monitoring is a key element in every cloud deployment, you should never deploy a production environment without proper monitoring and alerting in place first.

Couchbase Server provides different methods for collecting monitoring data about the state of a running cluster:

- Web-based administration console for monitoring data collected from running deployments and providing visual and email-based alerts based on that data.
- REST interface that can be queried for access to current and historic statistics that the cluster stores.

- Per node statistics storage that can be polled through a client that uses the binary protocol or with the cbstats utility.

In addition to Couchbase built-in monitoring, AWS provides several services for monitoring and managing your infrastructure.

- Amazon CloudWatch provides a log repository and monitoring dashboard for your services and application, including CPU, storage I/O and custom metrics. For Couchbase Instance metrics, we recommend monitoring the *CPUUtilization*, *VolumeReadOps*, *VolumeWriteOps*, and *VolumeQueueLength*. Amazon CloudWatch can send an alarm, by either SMS or email, when a user-defined threshold is reached. You can write a custom metric to CloudWatch, such as current free memory on your instances, and to trigger an alarm or an automatic response when the specified threshold is exceeded.
- AWS CloudTrail provides logs related to infrastructure and system management, such as APIs used, entities executing the commands, time stamps and related information.
- VPC Flow Logs provide information about traffic running within your VPC, including the 5 tuple, packet size and more.
- Amazon SNS is a managed pub/sub service where you can send notifications to different consumers, such as other services, endpoints (HTTP/S), or email recipients.

Conclusion

AWS provides a unique set of services for running and managing NoSQL applications, including Couchbase. Pairing Couchbase's ability to expand a running cluster and fleet management features of Amazon EC2 Auto Scaling to maintain the health and availability of your fleet, provide users the ability to build highly scalable enterprise class NoSQL database. This whitepaper has provided an overview of best practices for implementing Couchbase Server on Amazon EC2, paying particular attention to performance, durability, and security.

Contributors

Contributors to this document include:

- Anil Kumar, Director of Product Management, Couchbase

- Roy Rodan, Solutions Architect, Amazon Web Services
- Saurabh Shanbhag, Solution Architect, Amazon Web Services

Document Revisions

Date	Description
October 2020	Updated to Couchbase Server 6.6 on AWS
July 2013	First publication Couchbase Server 2.0 on AWS