Amazon DocumentDB

Cloud Database Service

Case Study



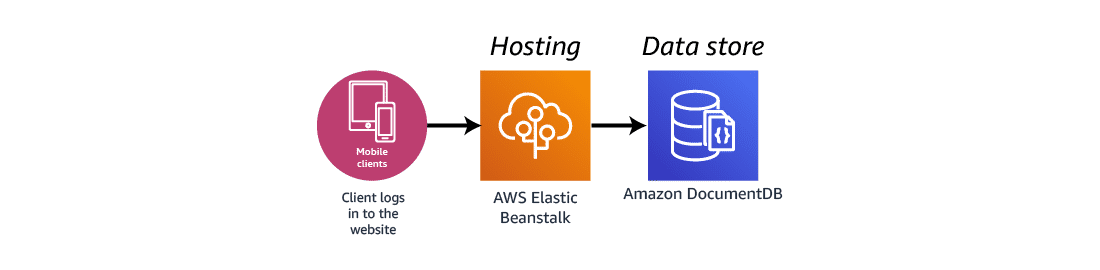
Introduction

Amazon DocumentDB (with MongoDB compatibility) is a fast, reliable, and fully managed database service. Amazon DocumentDB makes it easy to set up, operate, and scale MongoDB-compatible databases in the cloud. With Amazon DocumentDB, you can run the same application code and use the same drivers and tools that you use with MongoDB.


         Diagram showing Amazon DocumentDB endpoints including the cluster, reader, and instance
            endpoints.
      

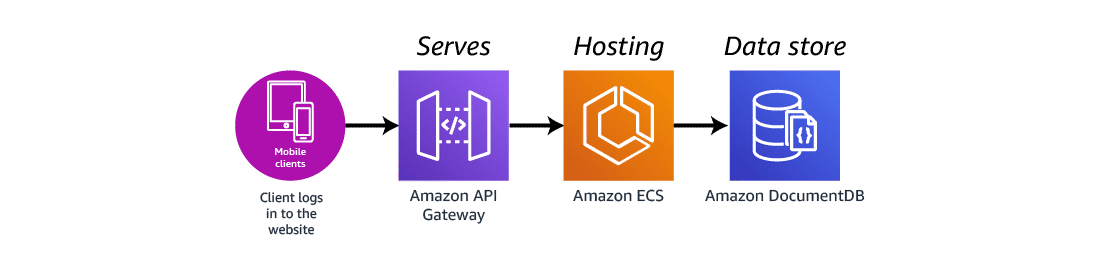
**Online user profile architecture**

Document databases are a great solution to store and query online profiles. This architecture is one way you can provide user profile information for a web application.



**Real-time mobile web application architecture**

Building high-performance mobile applications that scale to process millions of user requests per second with millisecond latency can be challenging. This architecture is one way you could build a mobile application to deliver real-time stock information and recommendations



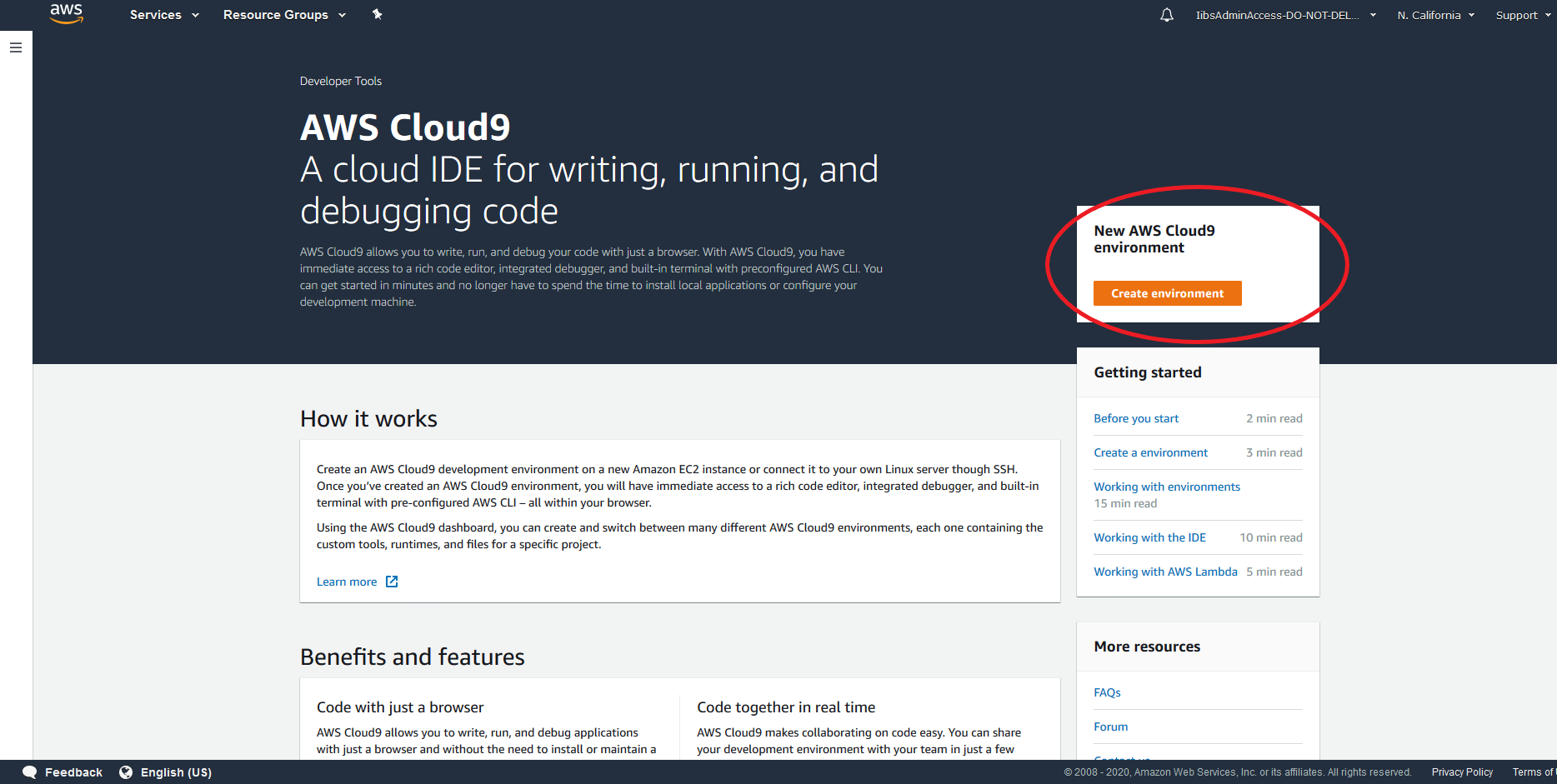
**Features**

* It provides 99.99% availability by copying the cluster's data in three different Availability Zones.
* It helps to scale storage and compute services independently.
* It provides automatic failover either to one of up to 15 replicas created in other Availability Zones or to a new instance if no replicas have been provisioned.
* It provides backup capability and point-in-time recovery for the cluster. It has a backup retention period of up to 35 days.
* It is best suited for TTL and Timeseries Workloads and supports ACID properties based on transactions across one or more documents

## Step 1: Create an AWS Cloud9 environment

AWS Cloud9 provides a web-based terminal that you can use to connect to and query your Amazon DocumentDB cluster using the mongo shell.

1. From the AWS Management Console navigate to the AWS Cloud9 console and choose **Create environment**.

****

In the **Environment name and description** section, in the **Name** field, enter DocumentDBCloud9.

Graphical user interface, text, application, email

Description automatically generated

Choose **Next step**.

Graphical user interface, text, application, email

Description automatically generated

In the **Configure settings section**, choose **Next step**.

Graphical user interface, application

Description automatically generated

In the **Review** section, choose **Create environment**

Graphical user interface, text, application

Description automatically generated

Step 2: Create a security group

This security group will enable you to connect to your Amazon DocumentDB cluster from your AWS Cloud9 environment.

1. On the [Amazon EC2 Management Console](https://console.aws.amazon.com/ec2), under **Network and Security**, choose **Security groups**.

A screenshot of a computer

Description automatically generated

Choose **Create security group**.

Graphical user interface, application, Word

Description automatically generated

In the **Basic details** box, input the following for each field:

* For **Security group name**, enter demoDocDB.
* For **Description**, enter a description.
* For **VPC**, accept the usage of your default VPC.

Graphical user interface, text, application

Description automatically generated

In the **Inbound rules** section, choose **Add rule**.

Graphical user interface, text, application

Description automatically generated

For **Type**, choose **Custom TCP Rule**.

Graphical user interface, text, application

Description automatically generated

For **Port range**, enter 27017.

Graphical user interface, text, application

Description automatically generated

The source is the security group for the AWS Cloud9 environment you just created. To see a list of available security groups, enter cloud9 in the destination field. Choose the security group with the name aws-cloud9-<*environment name*>.

Graphical user interface, application

Description automatically generated

Accept all other defaults and choose **Create security group**.

A screenshot of a computer

Description automatically generated

Step 3: Create an Amazon DocumentDB cluster

* **Type DocumentDB in Search box and click on it**
* **This is Amazon DocumentDB dashboard**
* **Click on launch Amazon DocumnetDB**

Graphical user interface, text, chat or text message, website

Description automatically generated

**Search box and click on it**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email, Teams

Description automatically generated

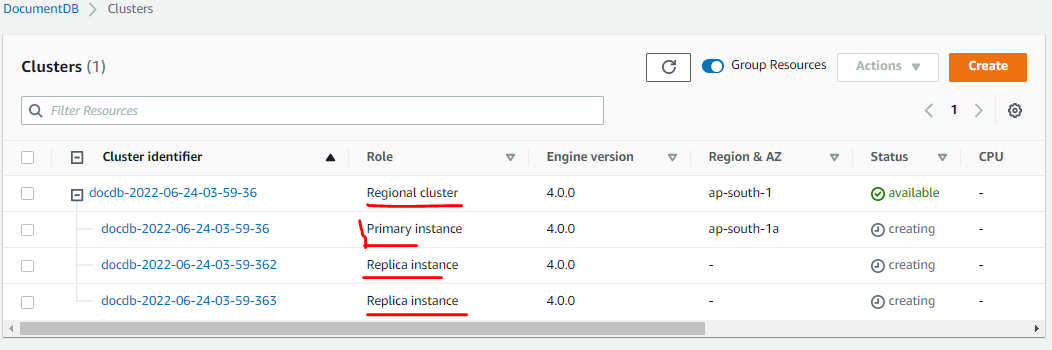
Text, application, timeline

Description automatically generated

**Cluster is Created**

Graphical user interface, application

Description automatically generated



Step 4: Install the mongo shell

You will now install the mongo shell in your AWS Cloud9 environment that you created in Step 1. The mongo shell is a command-line utility that you use to connect and query your Amazon DocumentDB cluster.

1. If your AWS Cloud9 environment is still open from Step 1, go back to that environment and skip to instruction 3. If you navigated away from you AWS Cloud9 environment, in the AWS Cloud9 management console, under **Your environments**, find the environment labeled**DocumentDBCloud9**. Choose **Open IDE**.

Graphical user interface, text, application

Description automatically generated

1. At the command prompt, create the repository file with the following command:

***echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86\_64/\ngpgcheck=1 \nenabled=1 \ngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/yum.repos.d/mongodb-org-4.0.repo***

Text

Description automatically generated

1. When it is complete, install the mongo shell with the following command:

*sudo yum install -y mongodb-org-shell*

1. To encrypt data in transit, download the public key for Amazon DocumentDB from <https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem>. This operation downloads a file named rds-combined-ca-bundle.pem.

*wget* [*https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem*](https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem)

Text

Description automatically generated

Step 5: Connect to your Amazon DocumentDB cluster

You will now connect to your Amazon DocumentDB cluster using the mongo shell that you installed in Step 4.

1. On the Amazon DocumentDB management console, under **Clusters**, locate your cluster. Choose the cluster you created by clicking on the cluster identifier.
2. Graphical user interface, text, application

   Description automatically generated
3. In the **Connectivity and Security** tab, under **Connect to this cluster with the mongo shell**, copy the connection string provided. Omit copying <insertYourPassword> so that you are prompted for the password by the mongo shell when you connect.

Graphical user interface, application

Description automatically generated

Step 6: Insert and query data

Now that you are connected to your cluster, you can run a few queries to get familiar with using a document database.

1. To insert a single document, enter the following:

db.collection.insert({"hello":"DocumentDB"})

1. You get the following output:

WriteResult({ "nInserted" : 1 })

1. You can read the document that you wrote with the findOne() command (because it only returns a single document). Input the following:

db.collection.findOne()

Text

Description automatically generated

Step 7: Explore

Congratulations! You have successfully completed the Amazon DocumentDB Case study.