**MongoDB Atlas with API Gateway and Cognito User Pool**

# Author

Deepti Chugh-AWS , Bharath S - AWS and Babu Srinivasan - MongoDB

# Title

Developer's Guide: Crafting API-Driven Apps with MongoDB Atlas using AWS CDK, API Gateway, and Lambda

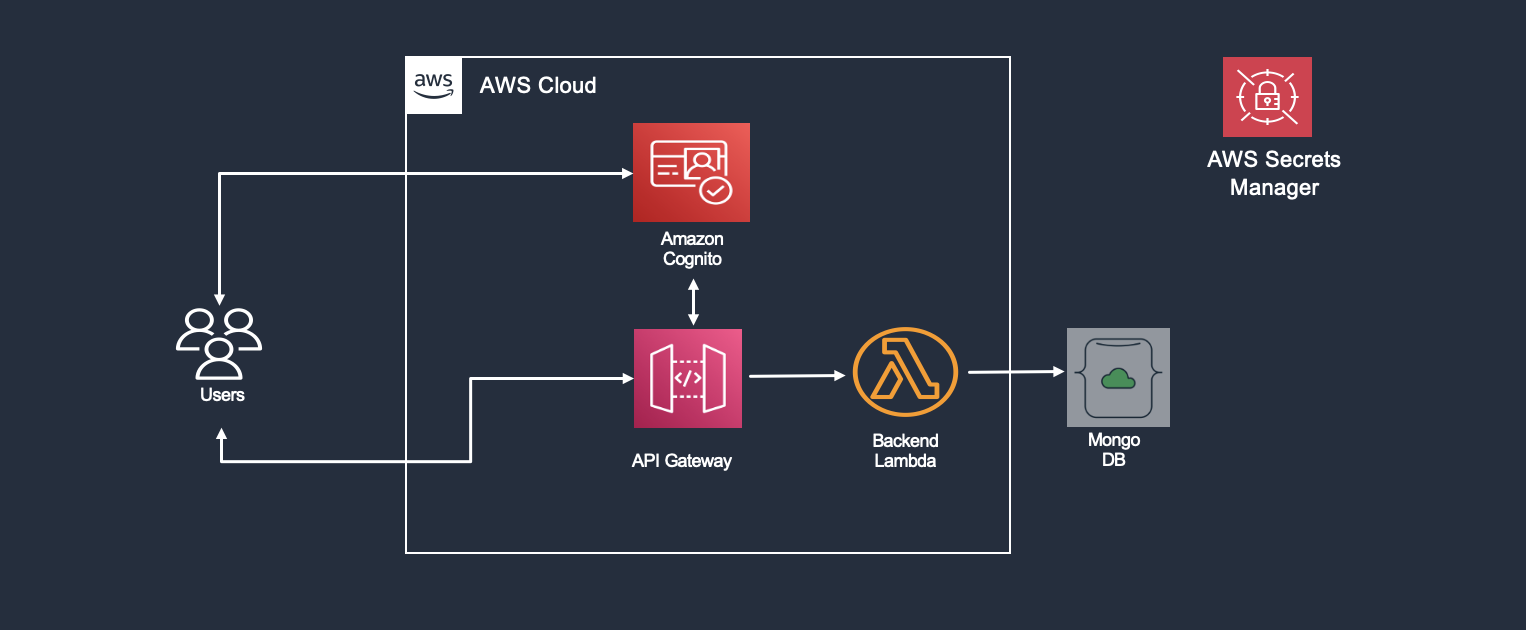
Welcome to our technical blog, where we unveil a step-by-step guide to deploying a robust REST API powered by Lambda functions, expertly bridging the gap between MongoDB Atlas and AWS, all with the added convenience of automation through AWS CDK. Our mission is to empower developers like you to seamlessly integrate MongoDB Atlas database clusters with AWS API Gateway, all while implementing authentication via Cognito User Pools. If you're ready to embark on a journey that not only streamlines the process of building modern API-driven applications but also leverages the power of automation, you're in the right place. Let's dive into the details and unlock the potential of this dynamic integration.

# What you will build?

This Solution comprises of following AWS Services which gets deployed using CDK (Cloud Development Kit):

* In the MongoDB SaaS account:
  + A MongoDB cluster
* In the customer account:
  + Amazon Cognito UserPool - User directory for authentication and authorization
  + AWS Secrets Manager – For keeping Mongo DB Database Credentials
  + API Gateway – Application Programming Interface acts as the "front door" for applications to access data, business logic, or functionality from your backend services
  + Lambda function – Backend Lambda function which connects to the Mongo DB database using PyMongo which is Python driver for MongoDB

# Reference Architecture

****

Deploying this Partner Solution with valid parameters builds the following environment in the AWS Cloud.

 [US East (N. Virginia) us-east-1](https://us-east-1.console.aws.amazon.com/)

# Implementation Steps

This solution uses Cloud Development Kit (CDK) to deploy the solution on AWS. First step involves creating a Mongo DB and then followed by deploying AWS Services.

**Prerequisites :-**

* [AWS CDK](https://docs.aws.amazon.com/cdk/v2/guide/getting_started.html)
* [NPM](https://docs.npmjs.com/downloading-and-installing-node-js-and-npm)
* [MongoDB Atlas](https://account.mongodb.com/account/login) Account
* AWS Account + [AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html) Installed and Configured
* Activate MongoDb Atlas Cloudformation resources in your AWS account with sufficient permissions
* Store MongoDB Atlas Programmatic API Keys in AWS Secrets Manager.You can find more details about these over [here](https://github.com/mongodb/awscdk-resources-mongodbatlas)

## Step 0 – Initialize the CDK project

1. Open IDE of your choice - Cloud9, VS Code or any other IDE

2. Execute below commands to initialize the environment

#Get the application code

git clone https://github.com/mongodb-partners/Microservice\_Application\_with\_MongoDBAtlas\_AWSCDK\_APIGW\_Lambda.git

cd aws\_mongodb\_sample\_dir

# If you DONT have cdk installed

npm install -g aws-cdk

# Make sure you in root directory

python3 -m venv .venv

source .venv/bin/activate

pip3 install -r requirements.txt

## Step 1 – Deploy MongoDB Atlas and AWS Resources

1. Set up the [AWS CLI and connect to the](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html) session.
2. Run below commands to install the python dependencies included with this sample.

#Install Dependencies for Lambda Function

cd aws\_mongodb\_sample

pip install --target ./dependencies awscdk-resources-mongodbatlas

pip install --target ./dependencies pymongo

cd ..

# Set Environment Variables

export ORG\_ID="<ORG\_ID>"

export MONGODB\_USER="<MONGODB\_USER>"

export MONGODB\_PASSWORD="<MONGODB\_PASSWORD>"

cdk bootstrap aws://<ACCOUNT\_NUMBER>/<AWS-REGION>

1. Run below commands to deploy CDK template.

cdk synth

cdk deploy -all

Copy the API Gateway output Endpoint from terminal as you will be needing this while testing API Gateway or alternatively you can copy from the stack output from cloud formation in console

## Step 3- Explore the deployed resources

Once the CDK is deployed, go to AWS Console and verify the resources

1. MongoDB::Atlas::Cluster
2. MongoDB::Atlas::Project
3. MongoDB::Atlas::DatabaseUser
4. MongoDB::Atlas::ProjectIpAccessList
5. Secret for storing ATLAS DB URI
6. Cognito User Pool
7. Lambda
8. API Gateway

## Step 4- Test Resources

1. Navigate to Cognito user pool and copy the User Pool Id and Client id(App Integration tab) from Cognito User pool
2. Open Cloud Shell and create a user with the command mentioned below

aws congnito-idp admin-create-user –user-pool-id <YOUR\_USER\_POOL\_ID> –username <USERNAME>

1. Once user is created, since it’s created by admin we will have to force change password by running below command

aws congnito-idp admin-set-user-password –user-pool-id <YOUR\_USER\_POOL\_ID> –username <USERNAME> –password<PASSWORD> –permanent

1. Replace the user pool id and client id copied in above step and also replace user name and password of the user created above

aws congnito-idp admin-initiate-auth –user-pool-id <YOUR\_USER\_POOL\_ID> –client-id <CLIENT\_ID> –auth-flow ADMIN\_NO\_SRP\_AUTH –auth-parameters USERNAME=<USERNAME>, PASSWORD=<PASSWORD>

1. Copy the ID Token created from the above step and run the below command to test the API

curl –location –request GET ‘<API\_GATEWAY\_ENDPOINT>’ –header ‘Content-Type: application/json’ –header ‘Authorization: <ID\_TOKEN>’

# Conclusion:

As we wrap up our journey into the world of modern API-driven applications, we hope this blog has illuminated the path to seamless integration. With AWS CDK, MongoDB Atlas, Cognito, and Lambda at your disposal, you're armed with the tools to craft dynamic, efficient, and scalable applications. The power of these technologies lies in your hands, and we encourage you to roll up your sleeves, dig into the code, and embark on your development adventure. The possibilities are boundless, and your next innovative application could be just a few lines of code away. So, go ahead, explore, experiment, and turn your ideas into reality with the combination of AWS CDK, MongoDB Atlas, Cognito, and Lambda. Your journey is just beginning, and the future of application development is at your fingertips

# Try Out:

[AWS CDK for MongoDB Atlas](https://github.com/mongodb/awscdk-resources-mongodbatlas), [Amazon Cognito](https://aws.amazon.com/pm/cognito/), and [Lambda](https://aws.amazon.com/lambda/)

# Rollback:

cdk destroy –all

# Cost and Licenses

There is no cost to use this Partner Solution, but you will be billed for any AWS services or resources that this Partner Solution deploys. For more information, refer to the [AWS Partner Solution General Information Guide](https://fwd.aws/rA69w?).

This Partner Solution deploys MongoDB Atlas resources with the latest stable MongoDB enterprise version, which is licensed and distributed under the [Server Side Public License (SSPL)](https://www.mongodb.com/licensing/server-side-public-license).