

Inovação com dados em nuvem

Integrando MongoDB com Autonomous
Database

Thamires Samira Ferreira

30.11.21

Inovação com dados em nuvem



Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Compartilhalgual 4.0 Internacional.
Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-sa/4.0/>.

Introdução

Um banco de dados convergente é um banco de dados com suporte nativo a todos os tipos de dados modernos e os mais recentes paradigmas de desenvolvimento integrados num único produto.

Bancos de dados convergentes suportam dados espaciais para reconhecimento de localização, JSON para armazenamento de documentos, IoT para integração de dispositivos, tecnologias in-memory para análises em tempo real e, claro, os dados relacionais tradicionais.

Ao fornecer suporte a todos esses tipos dados, o Banco de Dados Convergente pode executar todas as espécies de cargas de trabalho, de IoT a Blockchain até a Analytics e a Machine Learning. Ela também consegue operar com qualquer paradigma de desenvolvimento, incluindo Microservices, Events, REST, SaaS e CI/CD, para citar alguns.

MongoDB

Lançado em 2009, o MongoDB é um dos bancos de dados não-relacionais mais comuns do mercado. Ele tem como linguagem C++ e utiliza o Java Script para facilitar os recursos de pesquisas.

Ele também é open source e orientado por documentos (document database) em formato JSON. Por ser não relacional, não demanda a utilização de tabelas com colunas e linhas para fazer a armazenagem de dados. O MongoDB funciona em Windows, Linux e OSX.

Seus dados são armazenados dentro de documentos semelhantes a JSON, porém utilizando uma versão binário do JSON chamada BSON, que retém os dados usando pares de chave/valor.

Por exemplo:

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



Elementos principais no MongoDB:

- **Database:** Este é o elemento de nível superior. Sendo um recipiente físico de uma estrutura chamado de coleção (collections). Cada banco de dados tem o seu próprio conjunto de arquivos no seus sistemas de arquivos, porem o MongoDB tem um único servidor normalmente tem vários banco de dados.
- **Collection:** Este é um conjunto formado por documentos do MongoDB. Somente pode haver uma coleção com esse nome no banco de dados.
- **Documents:** Esta é a unidade básica de dado no MongoDB. Basicamente, ela é composta por conjuntos pares de chave/valor. Documentos são dinâmicos e fazem parte de uma mesma coleção sem precisar ter o mesmo conjunto de campos.

Uma das grandes diferenças entre bancos de dados convergentes como Oracle Autonomous Database e bancos de dados não relacionais como o MongoDB é que bancos de dados não relacionais em geral abrem mão de propriedades ACID para funcionar.

O que é ACID ?

ACID é um conceito que se refere às quatro propriedades de transação de um sistema de banco de dados: Atomicidade, Consistência, Isolamento e Durabilidade.

- **Atomicidade:** Em uma transação envolvendo duas ou mais partes de informações discretas, ou a transação será executada totalmente ou não será executada, garantindo assim que as transações sejam atômicas.
- **Consistência:** A transação cria um novo estado válido dos dados ou em caso de falha retorna todos os dados ao seu estado antes que a transação foi iniciada.
- **Isolamento:** Uma transação em andamento mas ainda não validada deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação concorrente.
- **Durabilidade:** Dados validados são registrados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

As propriedades ACID das transações permitem que você escreva aplicações sem considerar o ambiente complexo em que o aplicativo é executado.

Com transações ACID você pode se concentrar na lógica da aplicação e não na detecção de falhas, recuperação e sincronização do acesso aos dados compartilhados.

O Oracle Autonomous JSON Database é um serviço de banco de dados de documentos em nuvem que simplifica o desenvolvimento de aplicativos centrados em JSON. Ele apresenta APIs de documentos simples, dimensionamento sem servidor, transações ACID de alto desempenho, segurança abrangente e baixo preço de pagamento por uso.

Automatiza o provisionamento, configuração, ajuste, dimensionamento, patching, criptografia e reparo de bancos de dados, eliminando o gerenciamento de banco de dados e oferecendo disponibilidade de 99,995%.

Para saber mais acesse e aqui um comparativo completo entre MongoDB e Autonomous Database:

<https://www.oracle.com/autonomous-database/autonomous-json-database/oracle-json-vs-mongodb-atlas/>

O laboratório de hoje têm como intuito ensinar como habilitar uma API para o seu Autonomous Database Always Free ou não que permite o acesso e manipulação de databases MongoDB com SQL diretamente no banco Oracle.

Recursos usados:

- Conta na OCI – Oracle Cloud Infrastructure – Always Free

Provisionar um Autonomous Database

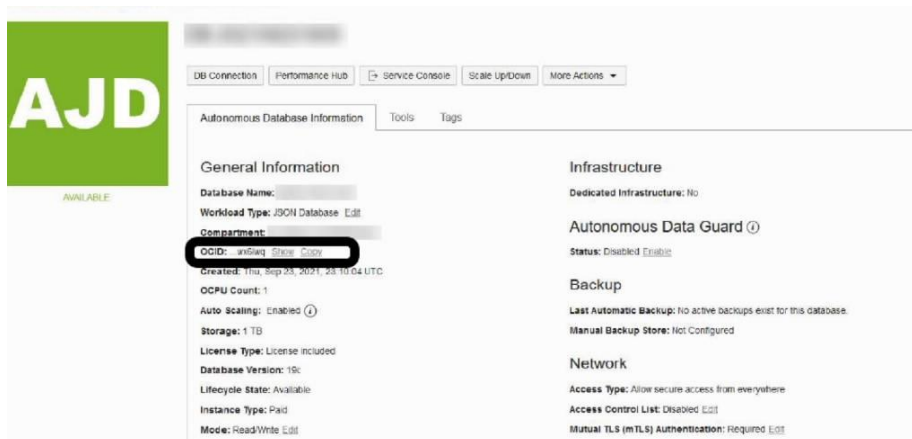
1. Abra o menu de navegação. Em **Oracle Database**, clique em **Autonomous Data Warehouse, Autonomous Transaction Processing** ou **Autonomous JSON Database**. Para provisionar o APEX Application Development, você pode clicar nos tipos de carga de trabalho mencionados anteriormente ou navegar para **Serviços do Desenvolvedor** e, em seguida, clicar em **APEX Application Development**.
2. Forneça as seguintes informações do Autonomous Database:
 - **Compartimento:** Selecione o compartimento do Autonomous Database.
 - **Nome para exibição:** Uma descrição amigável do usuário ou outras informações que ajudam a identificar facilmente o recurso. O nome para exibição não precisa ser exclusivo. Evite inserir informações confidenciais.
 - **Nome do Banco de Dados:** O nome do banco de dados deve consistir apenas em letras e números, começando por uma letra. O tamanho máximo é de 14 caracteres.
3. Escolha **Autonomous Transaction Processing**
4. Escolha o tipo de implantação da **Infraestrutura Compartilhada**.
5. Configure o banco de dados:
 - **Always Free:** Mova esse seletor para a direita para que o workflow de provisionamento mostre apenas as opções de configuração Always Free. Observe que os campos de configuração **Contagem de núcleos de CPU** e **Armazenamento** estão desativados ao provisionar um Autonomous Database Always Free. Seu banco de dados terá 1 OCPU, 8 GB de memória e 20 GB de armazenamento.
 - **Escolher versão do banco de dados:** Selecione uma versão do banco de dados entre as versões disponíveis.
6. **Criar credenciais do administrador:** Defina a senha do usuário ADMIN do Autonomous Database digitando uma senha que atenda aos critérios a seguir. Essa senha é usada ao acessar a console de serviço do Autonomous Database e ao usar uma ferramenta cliente SQL.

Critérios de senha:

 - Contém de 12 a 30 caracteres e inclui pelo menos uma letra maiúscula, uma letra minúscula e um caractere numérico.
 - Não contém a string "admin", independentemente de maiúsculas e minúsculas
 - Não é uma das quatro últimas senhas usadas para o usuário ADMIN
 - Não contém aspas duplas ("")
 - Não pode ser a mesma senha definida há menos de 24 horas
7. Escolha o tipo de acesso à rede.
 - **Permitir acesso seguro de qualquer lugar:** Por padrão, conexões seguras são permitidas de todos os lugares. Com essa opção, a autenticação TLS (mTLS) mútua é necessária para se conectar ao banco de dados.

Habilitar a API

Acesse o Autonomous Transaction Processing:



Selecione Service Console



Siga para “**Database Action**” e execute:

```
SQL> CREATE TESTUSER IDENTIFIED BY <sua senha>;
```

```
SQL> GRANT SODA_APP to TESTUSER;
```

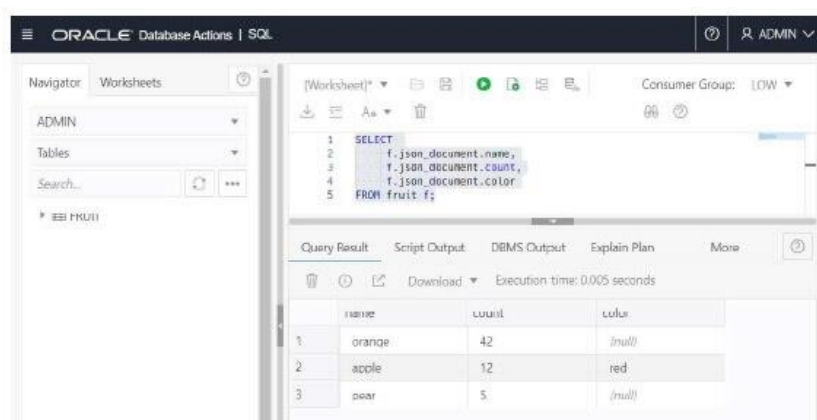
Acesse o MongoDB e utilize o commando abaixo para conectar o Autonomous ao seu MongoDB

```
$ mongosh --tls --tlsAllowInvalidCertificates 'mongodb://
TESTUSER:<PASSWORD>@<database URL>.
<OCI region>.oraclecloudapps.com:27016/admin?
authMechanism=PLAIN&authSource=$external&ssl=true&loadBalanced=false'
Current Mongosh Log ID: 614c9e2a01e3575c8c0b2ec7
Connecting to:      mongodb://<credentials>@<database
URL>.<OCIregion>.oraclecloudapps.com:27016/admin?
authMechanism=PLAIN&authSource=$external&tls=true&loadBalanced=false
Using MongoDB:      3.6.2
Using Mongosh:      1.0.7
For mongosh info see: https://docs.mongodb.com/mongodb-shell/admin
> show dbs
testuser    0 B
>
```

Crie uma coleção:

```
testuser> db.createCollection( 'fruit' )
{ ok: 1 }
testuser> show collections
fruit
testuser> db.fruit.insertOne( {name:"orange", count:42} )
{
  acknowledged: true,
  insertedId: ObjectId("614ca31fdab254f63e4c6b47")
}
testuser> db.fruit.insertOne( {name:"apple", count:12, color:
"red"} )
{
  acknowledged: true,
  insertedId: ObjectId("614ca340dab254f63e4c6b48")
}
testuser> db.fruit.insertOne( {name:"pear", count:5} )
{
  acknowledged: true,
  insertedId: ObjectId("614ca351dab254f63e4c6b49")
}
```

Volte para database actions e faça a query na coleção que acabou de criar:



A partir desse momento você poderá utilizar SQL para fazer queries e ter todos os benefícios de um banco de dados convergente e Oracle Autonomous Database.

Para maiores informações acesse:

1. <https://blogs.oracle.com/database/post/what-is-a-converged-database>
2. <https://www.oracle.com/autonomous-database/autonomous-json-database/oracle-json-vs-mongodb-atlas/>