Preparatório para AC3

Arquitetura C = 5

Kauan Cavazani Brianez

RA: 02221015

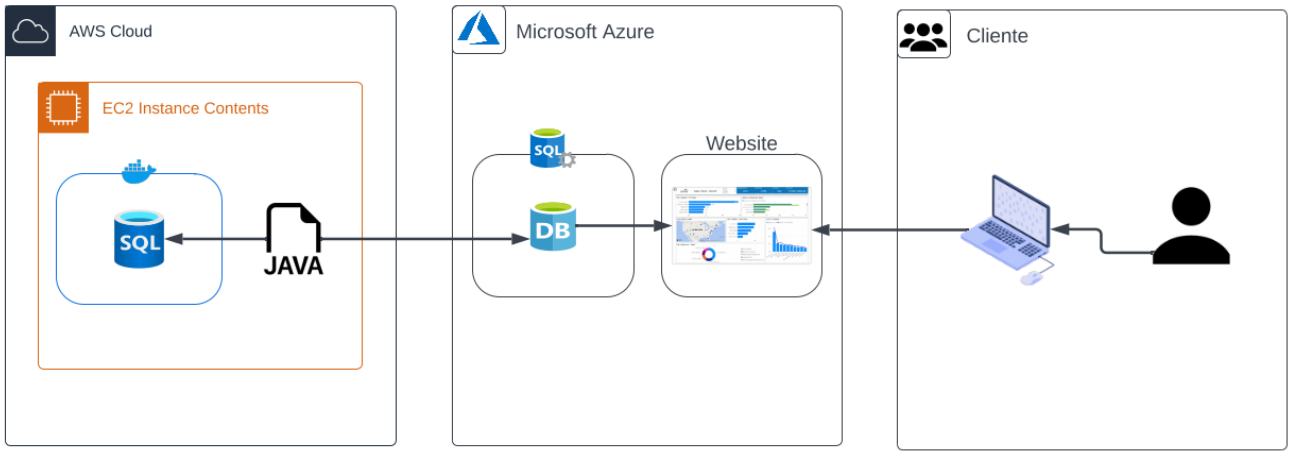
Sistemas Operacionais

São Paulo

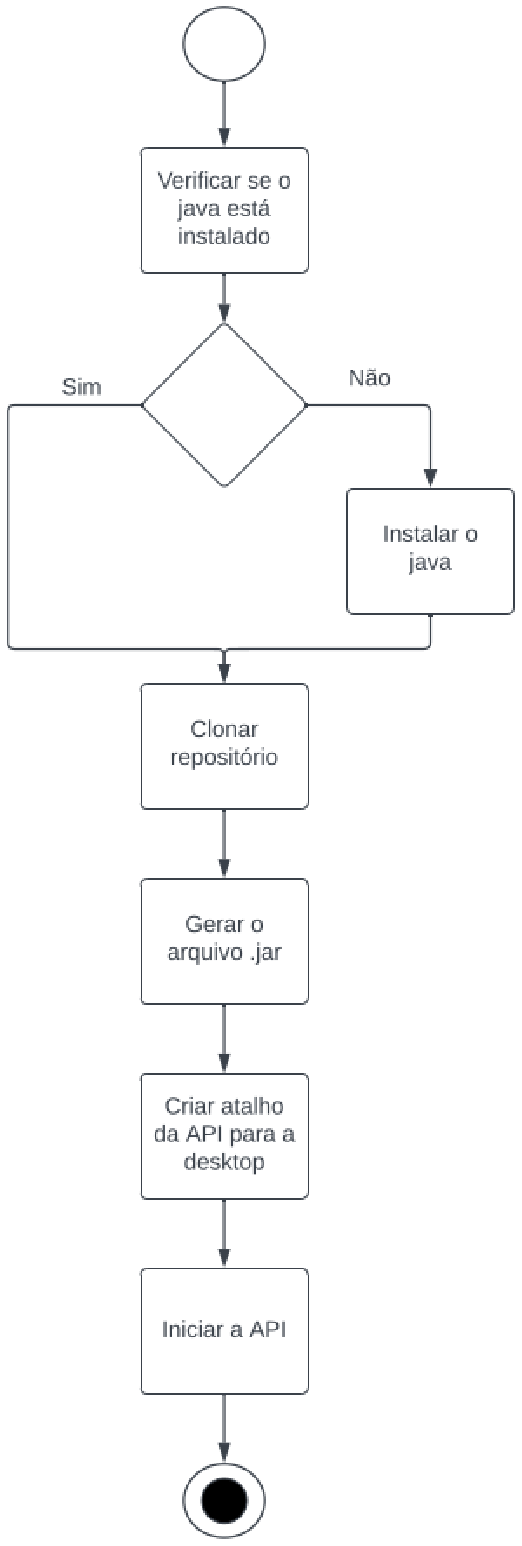
2022

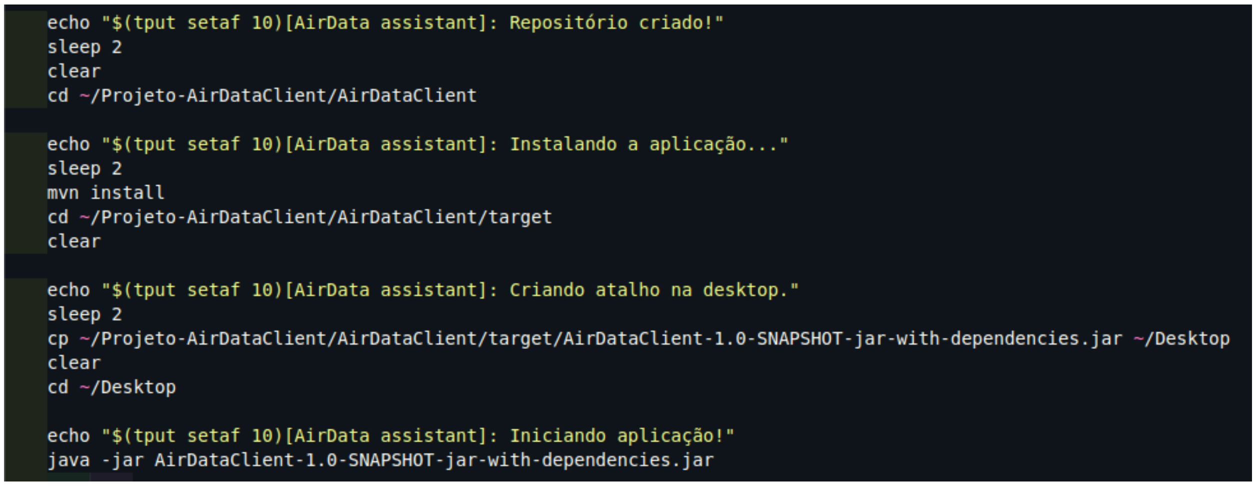
Arquitetura do Projeto

A arquitetura do projeto é formada em 3 partes, AWS Cloud, Microsoft Azure e Cliente, na AWS Cloud, temos uma instância EC2 que possui um container docker com o banco de dados mysql e a API java que captura os dados da máquina e envia para o banco de dados local dentro do docker e para o banco de dados SQL na Microsoft Azure, o cliente através de um computador com wifi, consegue acessar o website que contém uma dashboard que é populada pelo banco de dados da Azure



Script de Instalação

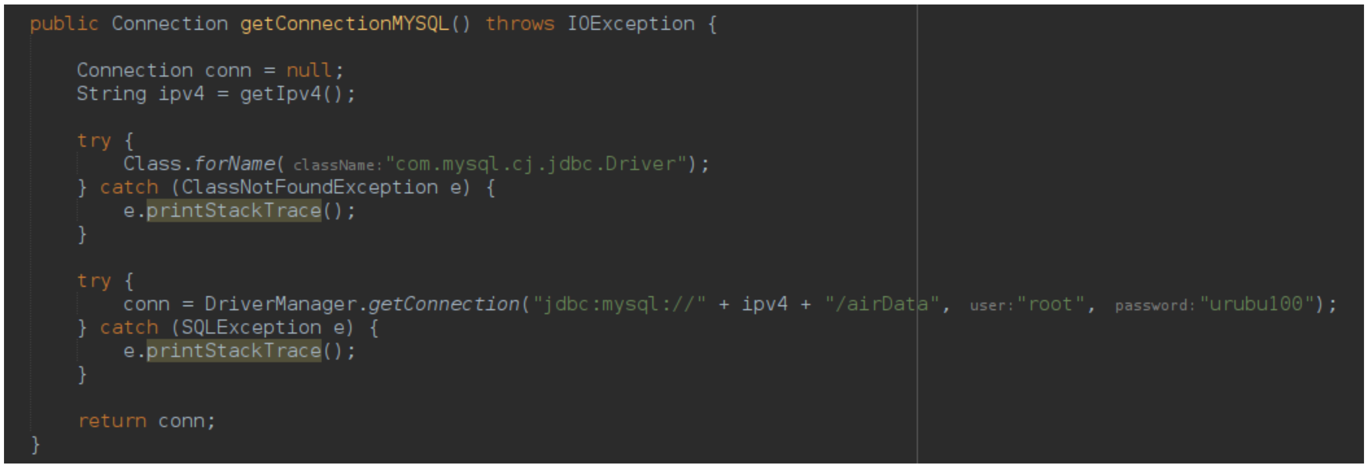




```
echo "$(tput setaf 10)[AirData assistant]: Repositório criado!"
sleep 2
clear
cd ~/Projeto-AirDataClient/AirDataClient

echo "$(tput setaf 10)[AirData assistant]: Instalando a aplicação..."
sleep 2
mvn install
cd ~/Projeto-AirDataClient/AirDataClient/target
clear

echo "$(tput setaf 10)[AirData assistant]: Criando atalho na desktop."
sleep 2
cp ~/Projeto-AirDataClient/AirDataClient/target/AirDataClient-1.0-SNAPSHOT-jar-with-dependencies.jar ~/Desktop
clear
cd ~/Desktop

echo "$(tput setaf 10)[AirData assistant]: Iniciando aplicação!"
java -jar AirDataClient-1.0-SNAPSHOT-jar-with-dependencies.jar
```

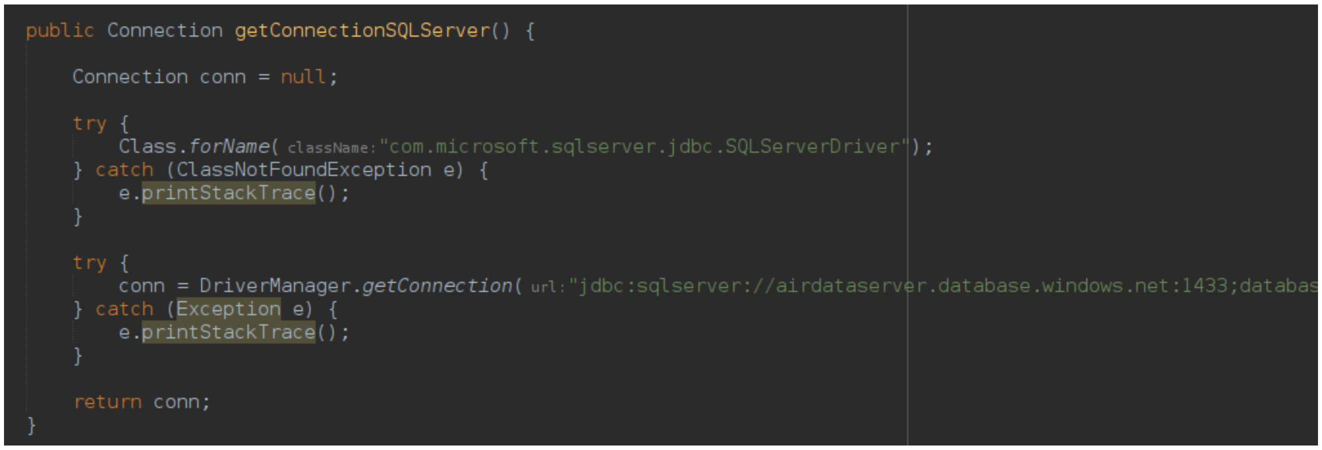
API Java

Método de conexão com o mysql no Docker:



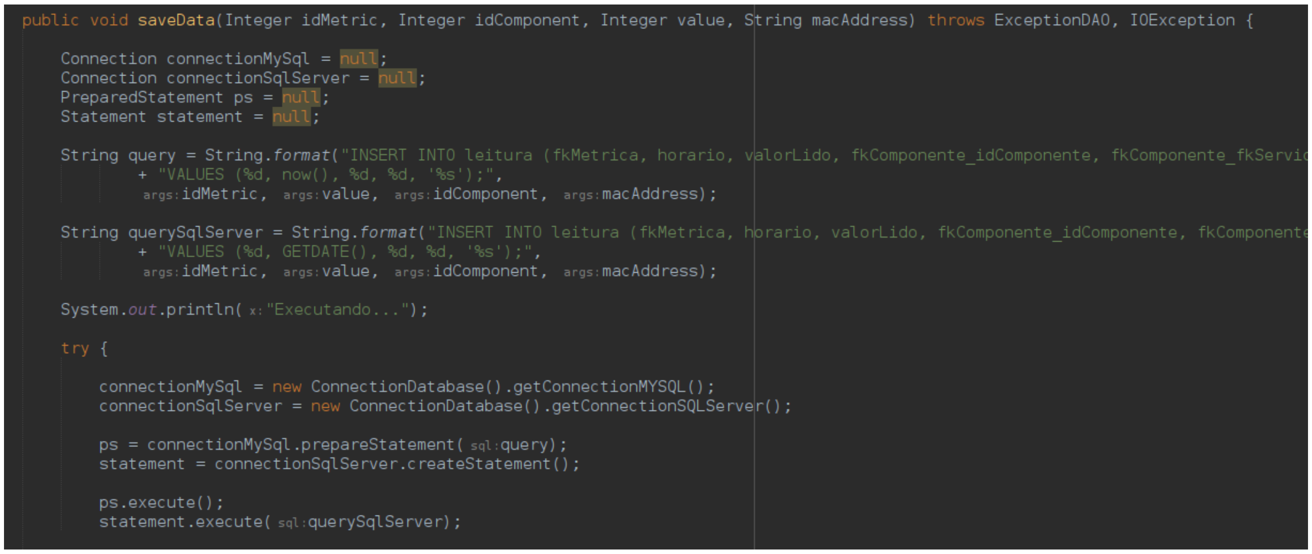
```java
public Connection getConnectionMYSQL() throws IOException {

    Connection conn = null;
    String ipv4 = getIpv4();

    try {
        Class.forName( className: "com.mysql.cj.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }

    try {
        conn = DriverManager.getConnection("jdbc:mysql://" + ipv4 + "/airData",  user: "root",  password: "urubu100");
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return conn;
}
```

Método de conexão com o SQL Server:



```java
public Connection getConnectionSQLServer() {

    Connection conn = null;

    try {
        Class.forName( className: "com.microsoft.sqlserver.jdbc.SQLServerDriver");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }

    try {
        conn = DriverManager.getConnection( url: "jdbc:sqlserver://airdataserver.database.windows.net:1433;databas
    } catch (Exception e) {
        e.printStackTrace();
    }

    return conn;
}
```

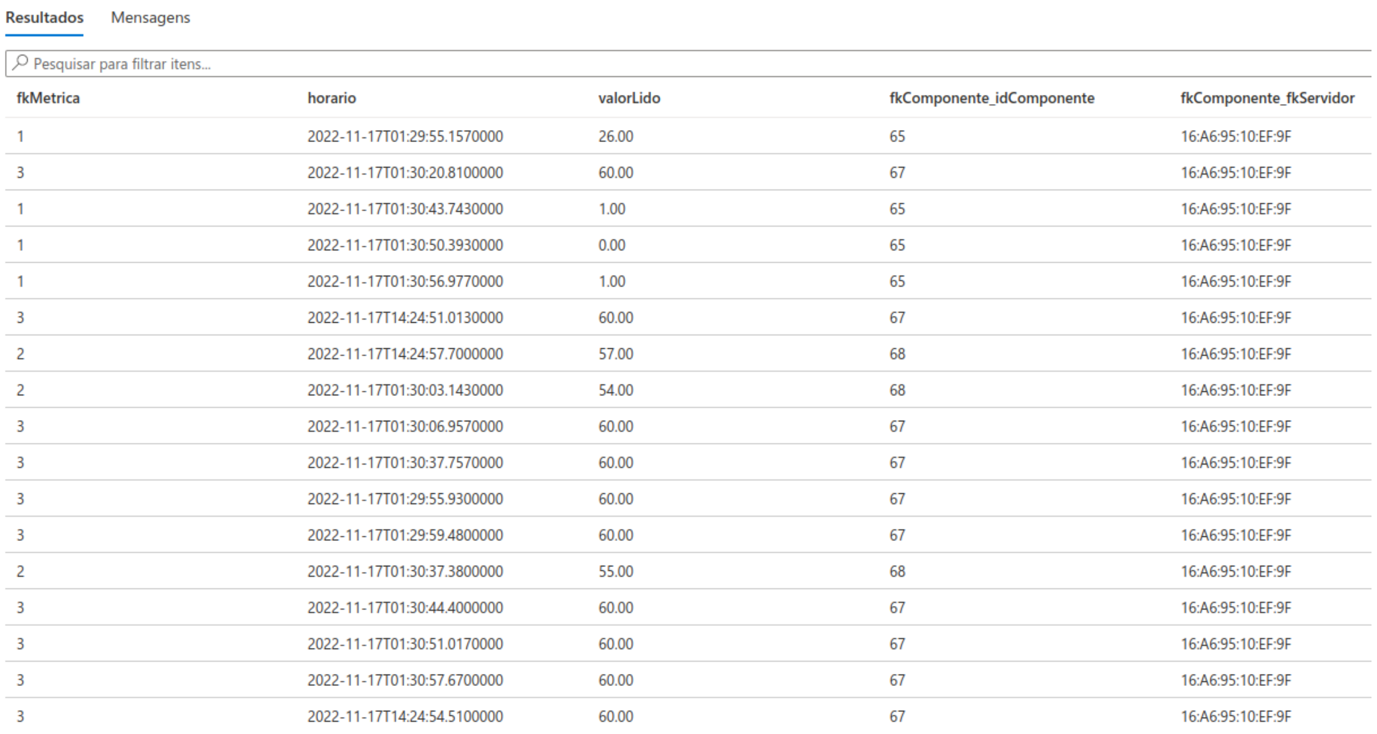
Método que salva os dados das leituras no banco de dados no Docker e no Banco de dados da Azure:



```
public void saveData(Integer idMetric, Integer idComponent, Integer value, String macAddress) throws ExceptionDAO, IOException {

    Connection connectionMySql = null;
    Connection connectionSqlServer = null;
    PreparedStatement ps = null;
    Statement statement = null;

    String query = String.format("INSERT INTO leitura (fkMetrica, horario, valorLido, fkComponente_idComponente, fkComponente_fkServic
            + "VALUES (%d, now(), %d, %d, '%s');",
             args: idMetric,  args: value,  args: idComponent,  args: macAddress);

    String querySqlServer = String.format("INSERT INTO leitura (fkMetrica, horario, valorLido, fkComponente_idComponente, fkComponente
            + "VALUES (%d, GETDATE(), %d, %d, '%s');",
             args: idMetric,  args: value,  args: idComponent,  args: macAddress);

    System.out.println( x: "Executando...");

    try {

        connectionMySql = new ConnectionDatabase().getConnectionMYSQL();
        connectionSqlServer = new ConnectionDatabase().getConnectionSQLServer();

        ps = connectionMySql.prepareStatement( sql: query);
        statement = connectionSqlServer.createStatement();

        ps.execute();
        statement.execute( sql: querySqlServer);
```

Banco de Dados Cliente e Servidor

Tabela onde fica armazenada as leituras no banco de dados da Microsoft Azure:

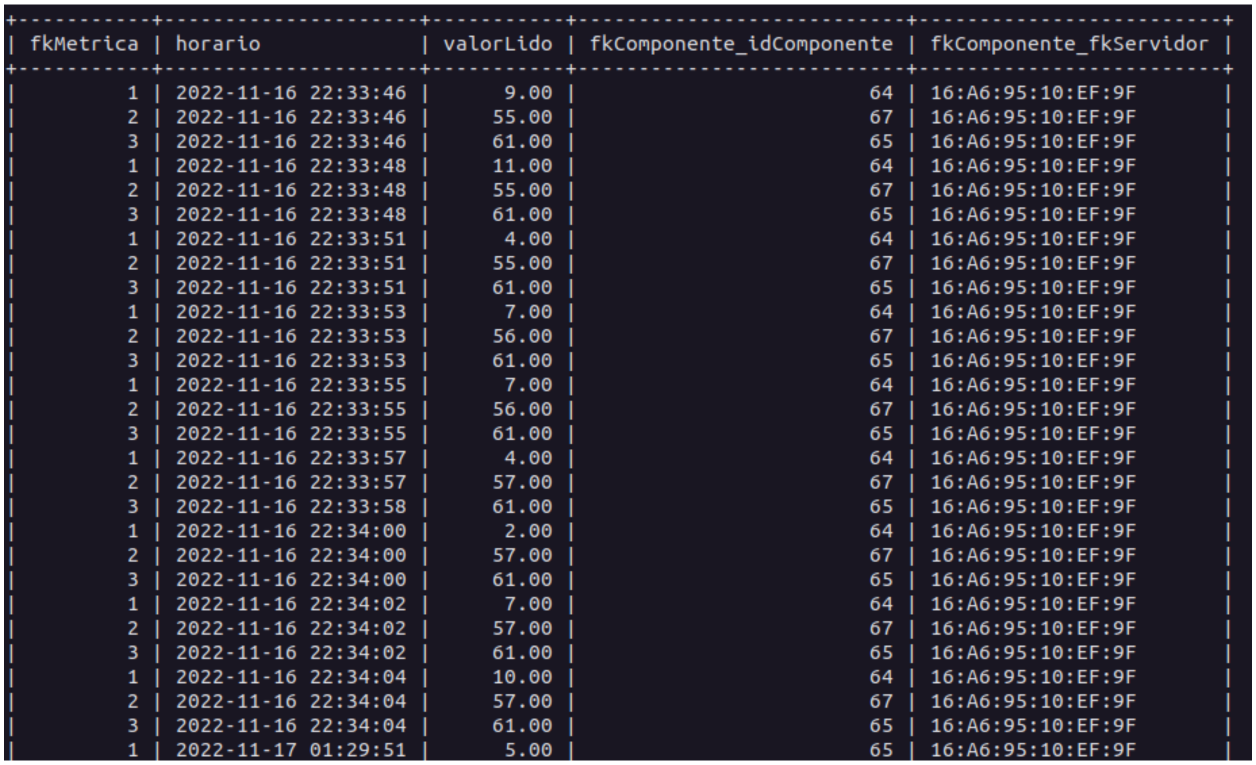


Resultados Mesagens

Pesquisar para filtrar itens...

fkMetrica	horario	valorLido	fkComponente_idComponente	fkComponente_fkServidor
1	2022-11-17T01:29:55.1570000	26.00	65	16:A6:95:10:EF:9F
3	2022-11-17T01:30:20.8100000	60.00	67	16:A6:95:10:EF:9F
1	2022-11-17T01:30:43.7430000	1.00	65	16:A6:95:10:EF:9F
1	2022-11-17T01:30:50.3930000	0.00	65	16:A6:95:10:EF:9F
1	2022-11-17T01:30:56.9770000	1.00	65	16:A6:95:10:EF:9F
3	2022-11-17T14:24:51.0130000	60.00	67	16:A6:95:10:EF:9F
2	2022-11-17T14:24:57.7000000	57.00	68	16:A6:95:10:EF:9F
2	2022-11-17T01:30:03.1430000	54.00	68	16:A6:95:10:EF:9F
3	2022-11-17T01:30:06.9570000	60.00	67	16:A6:95:10:EF:9F
3	2022-11-17T01:30:37.7570000	60.00	67	16:A6:95:10:EF:9F
3	2022-11-17T01:29:55.9300000	60.00	67	16:A6:95:10:EF:9F
3	2022-11-17T01:29:59.4800000	60.00	67	16:A6:95:10:EF:9F
2	2022-11-17T01:30:37.3800000	55.00	68	16:A6:95:10:EF:9F
3	2022-11-17T01:30:44.4000000	60.00	67	16:A6:95:10:EF:9F
3	2022-11-17T01:30:51.0170000	60.00	67	16:A6:95:10:EF:9F
3	2022-11-17T01:30:57.6700000	60.00	67	16:A6:95:10:EF:9F
3	2022-11-17T14:24:54.5100000	60.00	67	16:A6:95:10:EF:9F

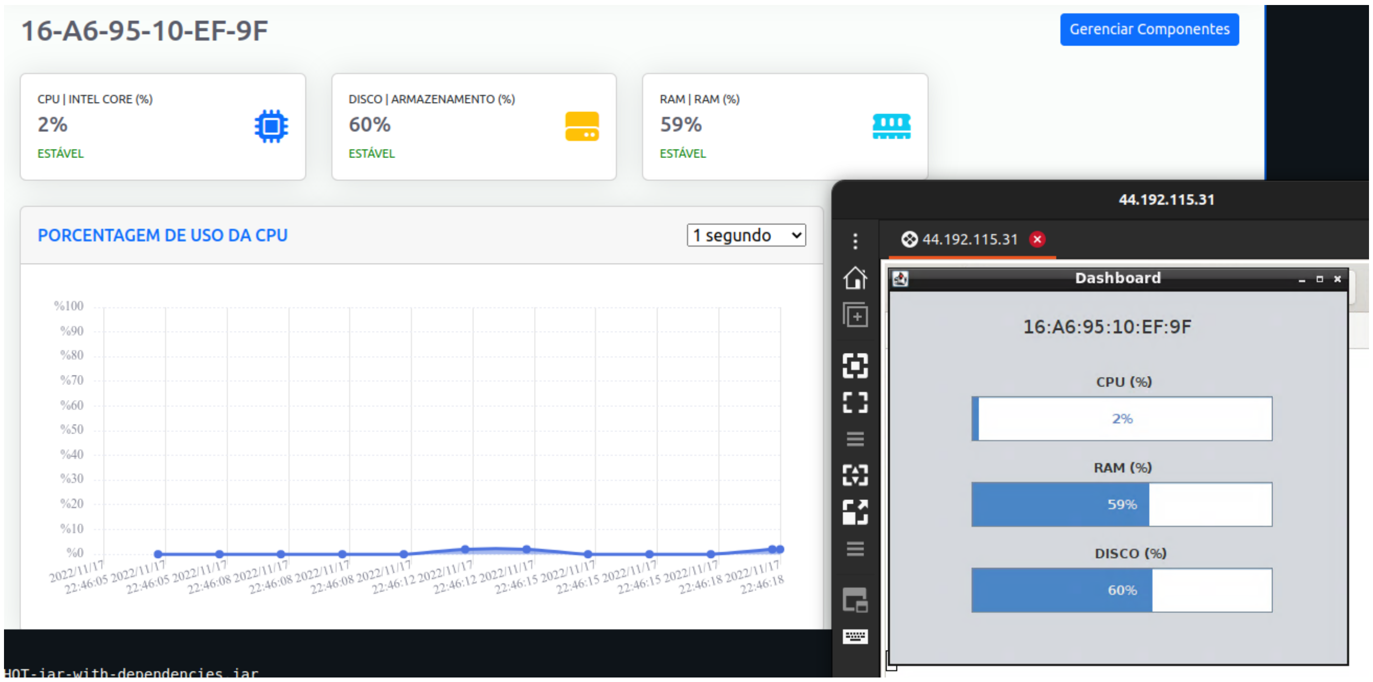
Tabela leitura do banco de dados mysql do Docker:



```
+-----------+---------------------+-----------+---------------------------+--------------------------+
| fkMetrica | horario             | valorLido | fkComponente_idComponente | fkComponente_fkServidor |
+-----------+---------------------+-----------+---------------------------+--------------------------+
|         1 | 2022-11-16 22:33:46 |      9.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:33:46 |     55.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:33:46 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:33:48 |     11.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:33:48 |     55.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:33:48 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:33:51 |      4.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:33:51 |     55.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:33:51 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:33:53 |      7.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:33:53 |     56.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:33:53 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:33:55 |      7.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:33:55 |     56.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:33:55 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:33:57 |      4.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:33:57 |     57.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:33:58 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:34:00 |      2.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:34:00 |     57.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:34:00 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:34:02 |      7.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:34:02 |     57.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:34:02 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-16 22:34:04 |     10.00 |                        64 | 16:A6:95:10:EF:9F        |
|         2 | 2022-11-16 22:34:04 |     57.00 |                        67 | 16:A6:95:10:EF:9F        |
|         3 | 2022-11-16 22:34:04 |     61.00 |                        65 | 16:A6:95:10:EF:9F        |
|         1 | 2022-11-17 01:29:51 |      5.00 |                        65 | 16:A6:95:10:EF:9F        |
```

Dashboard

Dashboard recebendo os dados do SQL Server enviados pela API java na EC2:



16-A6-95-10-EF-9F

CPU | INTEL CORE (%)
2%
ESTÁVEL

DISCO | ARMAZENAMENTO (%)
60%
ESTÁVEL

RAM | RAM (%)
59%
ESTÁVEL

PORCENTAGEM DE USO DA CPU

1 segundo

Dashboard

16:A6:95:10:EF:9F

CPU (%)
2%

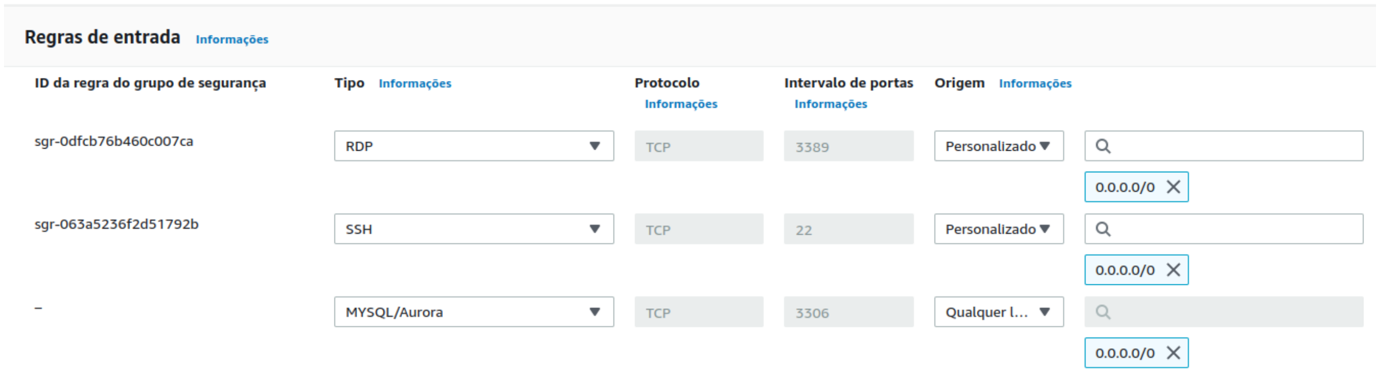
RAM (%)
59%

DISCO (%)
60%

```
HOT-jar-with-dependencies.jar
```

Configurações do Firewall/Portas SQL Server

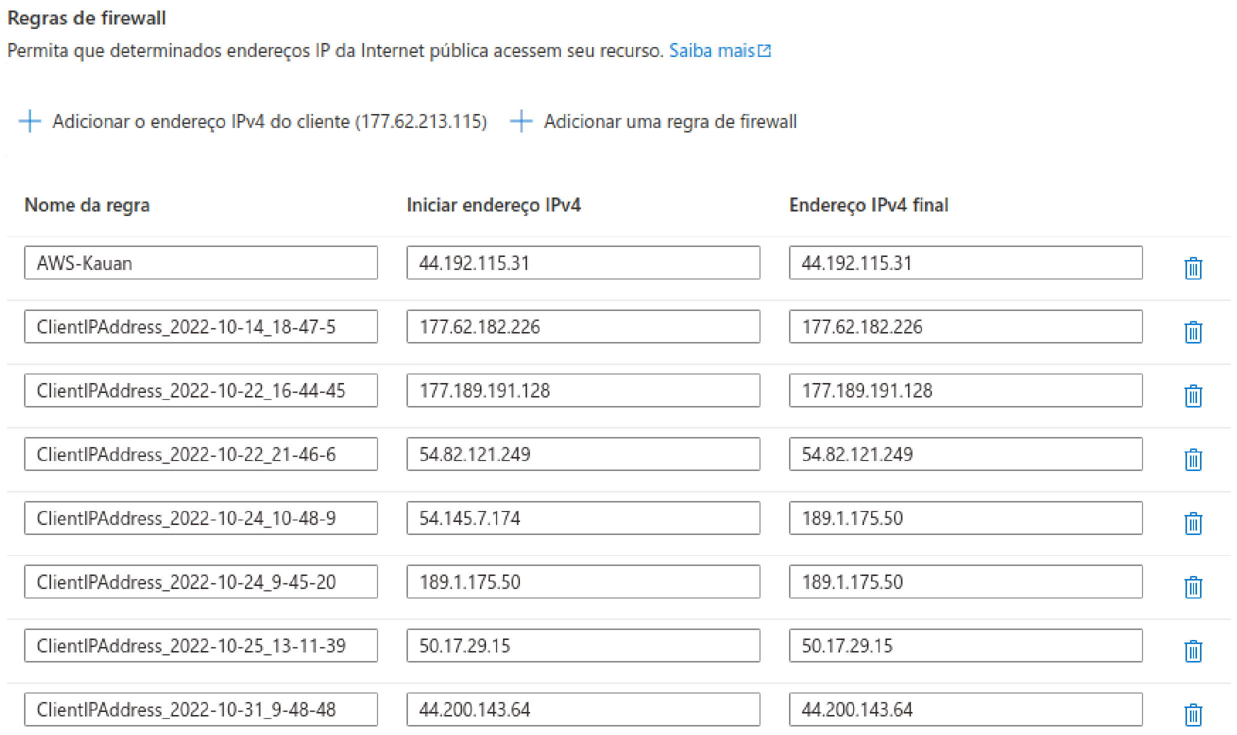
Regras de entrada da EC2:



Regras de entrada Informações

ID da regra do grupo de segurança	Tipo	Protocolo	Intervalo de portas	Origem	
sgr-0dfcb76b460c007ca	RDP	TCP	3389	Personalizado	0.0.0.0/0
sgr-063a5236f2d51792b	SSH	TCP	22	Personalizado	0.0.0.0/0
–	MYSQL/Aurora	TCP	3306	Qualquer l...	0.0.0.0/0

Configurações do Firewall do SQL Server:



Regras de firewall

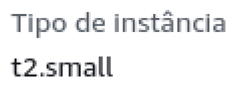
Permita que determinados endereços IP da Internet pública acessem seu recurso. Saiba mais

+ Adicionar o endereço IPv4 do cliente (177.62.213.115) + Adicionar uma regra de firewall

Nome da regra	Iniciar endereço IPv4	Endereço IPv4 final
AWS-Kauan	44.192.115.31	44.192.115.31
ClientIPAddress_2022-10-14_18-47-5	177.62.182.226	177.62.182.226
ClientIPAddress_2022-10-22_16-44-45	177.189.191.128	177.189.191.128
ClientIPAddress_2022-10-22_21-46-6	54.82.121.249	54.82.121.249
ClientIPAddress_2022-10-24_10-48-9	54.145.7.174	189.1.175.50
ClientIPAddress_2022-10-24_9-45-20	189.1.175.50	189.1.175.50
ClientIPAddress_2022-10-25_13-11-39	50.17.29.15	50.17.29.15
ClientIPAddress_2022-10-31_9-48-48	44.200.143.64	44.200.143.64

Configurações da EC2

Para realizar o upgrade da EC2, foi necessário trocar o tipo da instância de t2.micro para t2.small.



Tipo de instância

t2.small