

## Introdução a Backup e Disaster Recovery

Muitos profissionais acreditam que o **backup** e **disaster recovery** são a mesma coisa. Porém, não é bem assim. Mesmo que as duas rotinas possam ser utilizadas de forma até mesmo parecida, existem algumas diferenças básicas entre as duas abordagens

### Backup

A principal forma de evitar a perda de dados é realizando o backup.

Essa técnica consiste em realizar a cópia dos dados, de forma automática. Desse modo, armazenando os dados fora do ambiente de trabalho, mantendo registros e logs de todas as transações, além de garantir amplo versionamento dos dados.

Isso permite que, a qualquer tempo, caso os dados originais, sejam afetados de alguma maneira, possam ser restaurados o mais rápido possível.

Quais são esses tipos diferentes de soluções de backup?

- **Backup completo (ou Full)**

O backup completo é aquele no qual se realiza a cópia de todos os arquivos de determinado ambiente ou servidor, para outro local de armazenamento, seja local, virtual ou em nuvem.

Sua principal vantagem é a existência de uma reprodução fiel e completa de todas as informações do ambiente em questão.

Dos tipos de backup, é o que demanda maior espaço de armazenamento e demora mais para ser completado, justamente pelo volume de dados ser maior.

Porém, garante segurança de que todos os dados existentes estarão protegidos e principalmente é a base para que você tenha um **Disaster Recovery** efetivo, ou seja, conseguir recuperar todo seu ambiente em caso de perda total dos dados.

- **Backup incremental**

O backup incremental faz uma cópia dos dados alterados desde o último backup incremental, portanto copia uma quantidade menor de dados para o local escolhido, necessitando de menor espaço de armazenamento.

Entre os tipos de backup, é o mais rápido, já que não é feita uma cópia de todos os arquivos. Seu tempo de restauração também é mais ágil.

Porém, para que não haja chances de perdas de dados, é necessário utilizar o backup completo na mesma política de backup (*completo + incremental*), além de executar testes recorrentes após sua realização, conferindo se ele foi concluído corretamente e principalmente se está recuperável.

Vale lembrar que o teste recorrente, na verdade, pode e deve ser implementado para qualquer tipo de backup.

- **Backup diferencial**

O diferencial é semelhante ao incremental. Porém, os dados copiados são os alterados em relação ao último Backup Completo, armazenando mais dados do que o incremental.

Em questão de velocidade de realização do procedimento, ele é o intermediário entre os três tipos de backup. Pois requer mais espaço de armazenamento do que o incremental.

Devido sua maior facilidade de recuperação, a grosso modo, há uma maior segurança dos dados armazenados, diminuindo as chances de perda no processo.

- **Backup incremental contínuo**

Esta é uma nova modalidade de backup e que tem sido bastante utilizada pelas empresas mais atualizadas. Ele é bem semelhante ao incremental, porém, seu diferencial está na disponibilidade dos dados.

Ele permite a automatização do processo de recuperação de dados, de forma que você não necessita verificar quais bancos de dados necessitam ser recuperados.

Diferente do backup incremental comum, que necessita do backup completo e todos os subsequentes para resgatar as informações desejadas.

Funciona como uma espécie de backup completo, com agilidade e transparência em seu uso, sendo mais rápida a recuperação.

Além disso, por precisar apenas de um backup full, a necessidade de espaço de armazenamento é menor do que a necessidade dos demais tipos.

Este tipo de backup ganhou grande espaço nas empresas, principalmente pelo diferencial competitivo de ser mais moderno, eficiente e ágil no dia a dia.

## AWS Backup

Saber os tipos de backup é importante, porém, é preciso ir além, sabendo como fazê-lo de forma eficiente.

O **AWS Backup** é um serviço de backup totalmente gerenciado que facilita a centralização e a automação do backup de dados entre todos os serviços da AWS na nuvem e no local.

Com o AWS Backup, é possível configurar políticas de backup e monitorar as atividades de backup de seus recursos da AWS em um único lugar.

O AWS Backup automatiza e consolida tarefas de backup que foram executadas anteriormente serviço por serviço e remove a necessidade de criar scripts personalizados e processos manuais.

Para ativar o backup na nossa instancia RDS PostgreSQL é necessário realizar o ajuste nos parâmetros **BkpRetentionPeriod** e **BkpWindows**.

### BkpRetentionPeriod:

Description: Período dos backups a serem retidos, deve ser um valor de 0 a 35. 0 desativa backups automatizados

Type: Number

Default: '7'

### BkpWindows:

Description: O intervalo de tempo diário durante o qual os backups automatizados serão criados. Deve estar no formato hh24:mi-hh24:mi

Type: String

Default: "03:15-04:15"

The screenshot displays the 'Maintenance & backups' configuration page. It includes sections for 'Maintenance' (Auto minor version upgrade: Enabled, Maintenance window: wed:07:30-wed:08:00 UTC (GMT), Pending maintenance: none, Pending modifications: Master User Password: \*\*\*\*), 'Pending maintenance (0)' (with a filter input and buttons for 'Apply now' and 'Apply at next maintenance window'), and 'Backup' (Automated backups: Enabled (7 Days), Latest restore time: February 7th 2021, 6:06:56 pm UTC, Copy tags to snapshots: Disabled, Backup window: 03:15-04:15 UTC (GMT)).

## Disaster Recovery

O disaster recovery é uma abordagem muito mais profunda, que não se apoia em apenas restaurar os dados perdidos por conta de problemas ou ataques de cibercriminosos.

Geralmente, empresas que não podem parar suas operações, criam planos de disaster recovery para evitar que uma situação adversa leve a paralisação de qualquer de suas atividades.

O plano de recuperação de desastres é um processo documentado, composto por uma série de medidas de contenção, com o objetivo de manter em funcionamento todas as operações de seu cliente, independentemente do que possa vir a ocorrer.

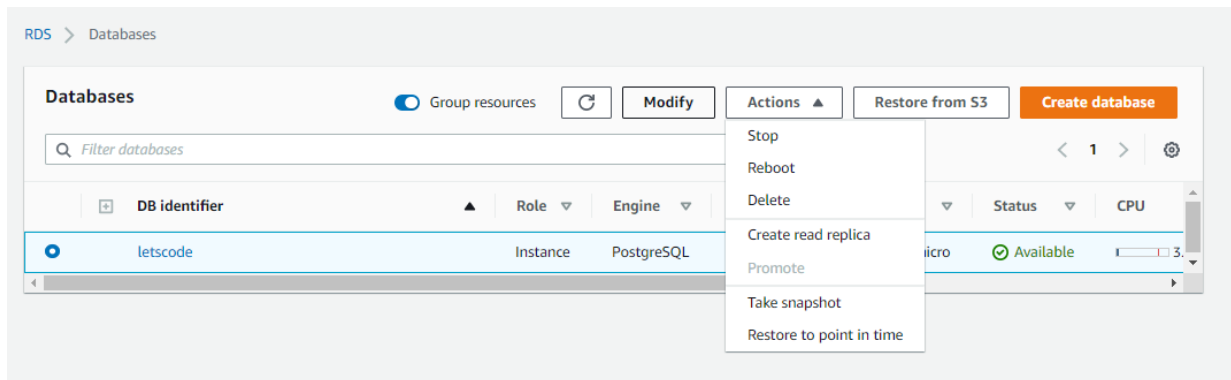
A sua criação não é simples e deve levar em conta uma série de fatores. Por exemplo, a infraestrutura de TI de seu cliente, os sistemas em utilização, rotinas de backup, entre outros pontos.

Um plano de disaster recovery traz muito mais confiança ao seu cliente, que sabe que suas atividades podem ser restauradas no menor tempo possível e você e sua equipe poderão agir a qualquer momento para evitar essas situações.

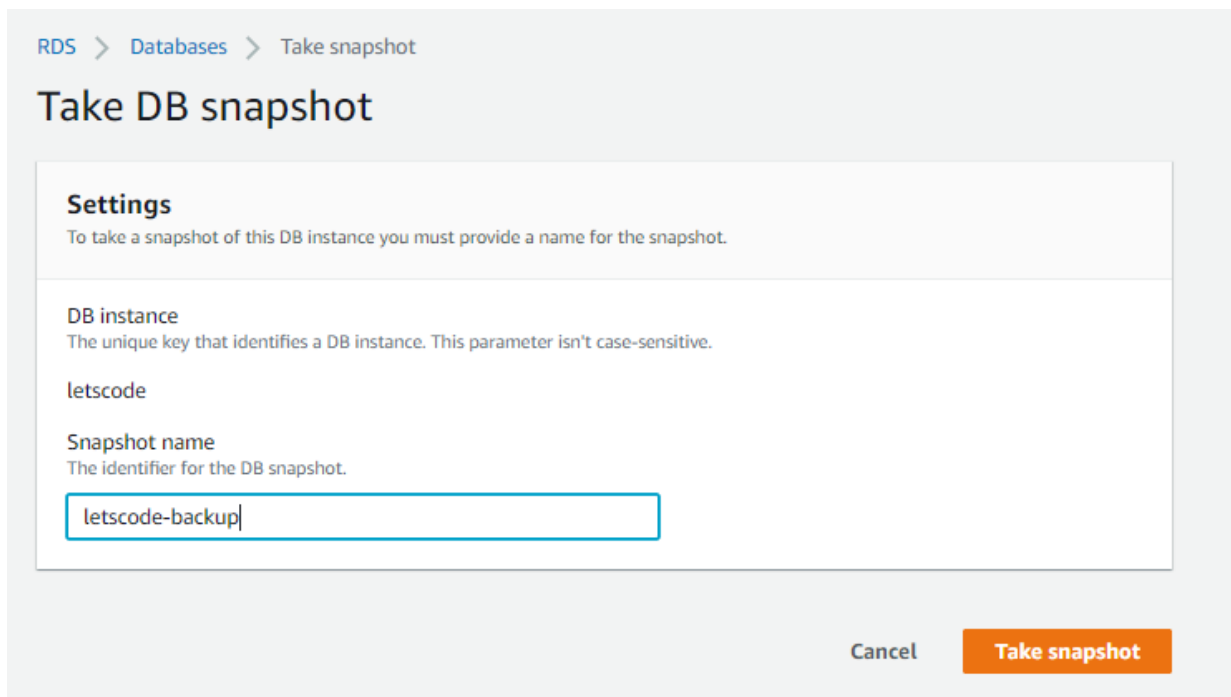
## Criar um snapshot do banco de dados PostgreSQL

O Amazon RDS cria um snapshot do volume de armazenamento de sua instância de banco de dados, fazendo o backup de toda a instância de banco de dados, não apenas dos bancos de dados individuais.

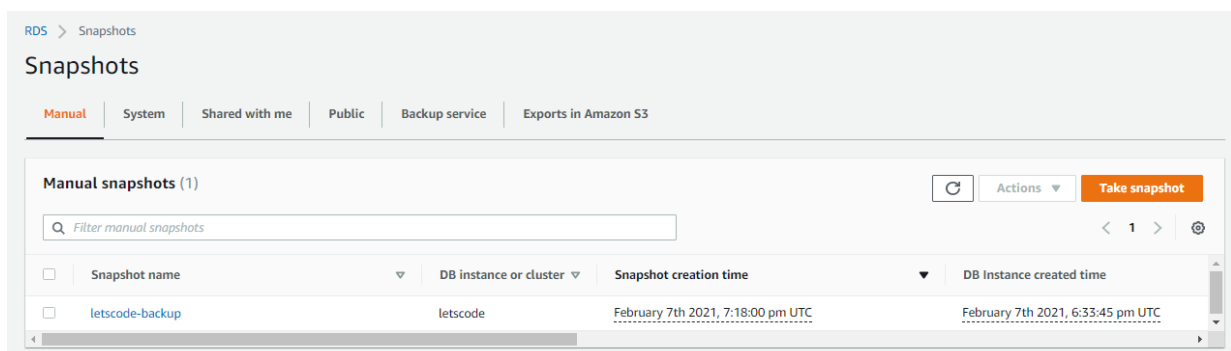
Selecione a instância RDS, **Actions**, **Take snapshot**:



Informe um **Snapshot name**:



Consulte criação **Snapshot**:

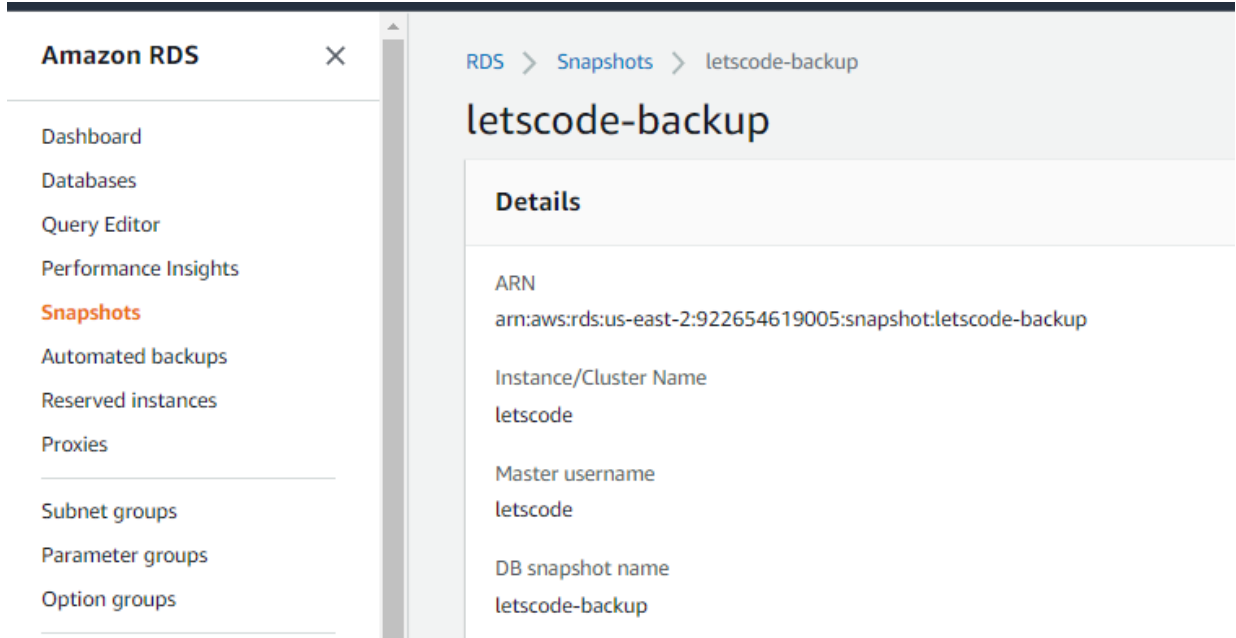


Restaurar o banco de dados PostgreSQL a partir do snapshot utilizando CloudFormation

Utilizaremos o snapshot **letscode-backup** para realizar o restore do banco de dados PostgreSQL. A propriedade no Cloudformation responsável por um restore a partir de um snapshot é o **DBSnapshotIdentifier**.

O valor da propriedade **DBSnapshotIdentifier** é o nome de recurso da [Amazon \(ARN\)](#) do DB snapshot usado para restaurar a instância do banco de dados.

*arn:aws:rds:us-east-2:922654619005:snapshot:letscode-backup*



Ao especificar esta propriedade, você pode criar uma instância de banco de dados a partir do snapshot de banco de dados especificado.

Algumas propriedades de instância de banco de dados não são válidas ao restaurar de um snapshot, como as propriedades **MasterUsername** e **MasterUserPassword**. Para obter informações sobre as propriedades que você pode especificar, consulte a ação [RestoreDBInstanceFromDBSnapshot](#).

```
AWSTemplateFormatVersion: '2010-09-09'
```

```
Description: Restore Banco de Dados Postgres - RDS
```

```
#####
```

```
#           Parameters           #
```

```
#####
```

```
Parameters:
```

```
  RDSName:
```

```
    Description: Nome do DB
```

```
    Type: String
```

```
    Default: letscode-backup
```

```
  RDSInstanceType:
```

```
    Type: String
```

```
    Description: Tipo instancia RDS
```

```
    Default: db.t2.micro
```

```
  Engine:
```

```
    Description: Tipo Sistema de Gerenciamento de Banco de Dados
```

```
    Type: 'String'
```

```

    Default: postgres
EngineVersion:
    Description: Versão do Sistema de Gerenciamento de Banco de Dados
    Type: 'String'
    Default: "12.5"
VPCID:
    Description: VPCID
    Type: 'String'
    Default: "vpc-fe7fc095"
DBAllocatedStorage:
    Default: '20'
    Description: Disco do banco de dados
    Type: Number
StorageType:
    Description: Tipo de armazenamento, valores válidos standard | gp2 | io1
    Type: 'String'
    Default: gp2
MultiAZ:
    Description: Multi-AZ
    Type: String
    Default: 'false'
    AllowedValues:
        - 'true'
        - 'false'
MonitoringInterval:
    Description: Especificar o valor entre 0, 1, 5, 10, 15, 30, 60
    Type: Number
    Default: '0'
BkpRetentionPeriod:
    Description: Período dos backups a serem retidos, deve ser um valor de 0
a 35. 0 desativa backups automatizados
    Type: Number
    Default: '7'
BkpWindows:
    Description: O intervalo de tempo diário durante o qual os backups
automatizados serão criados. Deve estar no formato Deve estar no formato
hh24:mi-hh24:mi
    Type: String
    Default: "03:15-04:15"
DBSnapshotIdentifier:
    Description: Criar instância RDS a partir de um snapshot
    Type: String
    Default: "arn:aws:rds:us-east-2:922654619005:snapshot:letscode-backup"

```

```
#####
```

```
#           Resources           #
```

#####

#### Resources:

##### RDSInstance:

Type: AWS::RDS::DBInstance

##### Properties:

DBSnapshotIdentifier: !Ref DBSnapshotIdentifier

DBInstanceIdentifier: !Ref RDSName

AllocatedStorage: !Ref 'DBAllocatedStorage'

StorageType: !Ref StorageType

DBInstanceClass: !Ref RDSInstanceType

Engine: !Ref Engine

EngineVersion: !Ref EngineVersion

MultiAZ: !Ref 'MultiAZ'

PubliclyAccessible: true

StorageEncrypted: false

AutoMinorVersionUpgrade: true

BackupRetentionPeriod: !Ref 'BkpRetentionPeriod'

PreferredBackupWindow: !Ref 'BkpWindows'

MonitoringInterval: !Ref 'MonitoringInterval'

#### Referências

[O que é o AWS Backup?](#)

[AWS::RDS::DBInstance](#)

[RestoreDBInstanceFromDBSnapshot](#)

[< Tópico anterior](#)

[Próximo Tópico >](#)