

06

Para saber mais: Elastic Block Device

O que é EBS?

Nesse momento do treinamento você já viu em algum momento aparecendo EBS. O Elastic Block Device é o lugar onde as instâncias vão salvar seus dados. O EBS é como um HD virtual disponíveis para as instâncias garantindo alta disponibilidade pois automaticamente replica os dados dentro da mesma zona de disponibilidade.

Separando Dados e Processamento

Repare que temos as instâncias (unidades de computação) em um lado e o EBS (HD virtuais) no outro.. Se há algum problema na instância não vai afetar os dados e vice versa. Isso também significa que podemos associar um volume do EBS com uma outra instância a qualquer momento, mas apenas para uma instância ao mesmo tempo.

Novos volumes

Você pode criar EBS Volume separadamente e associar com uma instância depois mas **cuidado esse serviço pode ser cobrado**. A criação é simples, basta ir no menu *Elastic Block Device -> Volumes* e no botão *Create Volume*:

ELASTIC BLOCK STORE

Volumes

Snapshots

Create Volume

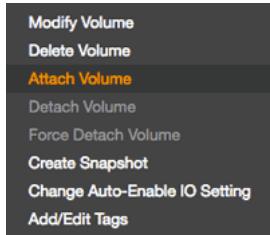
No formulário devemos escolher os detalhes do volume, como o tipo (ssd, provisioned ssd, magnetic), tamanho e zona de disponibilidade:

Create Volume

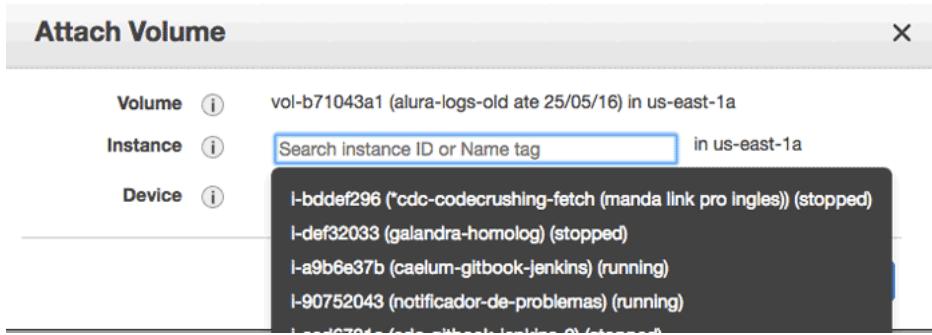
Volume Type	<input type="button" value="General Purpose SSD (GP2)"/>
Size (GiB)	<input type="text" value="100"/> (Min: 1 GiB, Max: 16384 GiB)
IOPS	<input type="text" value="300 / 3000"/> (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
Throughput (MB/s)	<input type="text" value="Not Applicable"/>
Availability Zone	<input type="button" value="us-east-1a"/>
Snapshot ID	<input type="text" value="Search (case-insensitive)"/>
Encryption	<input type="checkbox"/> Encrypt this volume
<input type="button" value="Cancel"/> <input type="button" value="Create"/>	

Associar volume e instância

Uma vez criado podemos associar o volume com uma instância basta clicar com o botão direto no volume:



E depois escolher a instância que usará o volume:



Vamos verificar se conseguimos identificar o volume na máquina através do comando `df -h`:

```
[ubuntu@ip-172-31-28-205:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            488M    0  488M   0% /dev
tmpfs           100M  3.1M   97M   4% /run
/dev/xvda1       7.7G  955M  6.8G  13% /
tmpfs           496M    0  496M   0% /dev/shm
tmpfs            5.0M    0   5.0M   0% /run/lock
tmpfs           496M    0  496M   0% /sys/fs/cgroup
tmpfs           100M    0   100M   0% /run/user/1000
```

Repare que conseguimos ver o *filesystem* default da máquina (`/dev/xvda1`) mas não conseguimos ver o device de 100GiB que acabamos de criar. Isso acontece porque é necessário formatar e montar o volume na instância.

Para isso, vamos executar o comando `sudo fdisk -l` para descobrir o nome real do device (o kernel do Linux pode alterar esse nome automaticamente):

Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes

Repare que há um dispositivo com 100GiB chamado `/dev/xvdf`, diferente do nome original escolhido pelo AWS.

Screenshot of the AWS Management Console showing the EBS Volume list and a detailed view of a volume.

EBS Volume List:

Name	Volume ID	Size	Type	IOPS	Snapshot	Created	Availability Zone	State
vol-02e5c556f961827	vol-02e5c556f961827	100 GiB	gp2	300 / 3000	-	May 30, 2017 at 9:4...	sa-east-1c	in-use

Volume Details:

Size	100 GiB	Snapshot	-
Created	May 30, 2017 at 9:49:01 PM UTC-3	Availability Zone	sa-east-1c
State	in-use	Encrypted	Not Encrypted
Attachment information	i-08b87d652606c2a42:/dev/sdf (attached)	KMS Key ID	
Volume type	gp2	KMS Key Aliases	

Vamos formatar esse volume executando `sudo mkfs -t ext3 /dev/xvdf`.

```
[ubuntu@ip-172-31-28-205:~$ sudo mkfs -t ext3 /dev/xvdf
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 26214400 4k blocks and 6553600 inodes
Filesystem UUID: f3ea08d6-18e3-4164-ab23-096f9f513122
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
ubuntu@ip-172-31-28-205:~$
```

Após isso precisamos montar esse volume e para isso primeiramente crie uma pasta executando `sudo mkdir /dev/ebs`.

Para montar o volume vamos usar o comando `mount <caminho-do-volume> <caminho-da-pasta>`. No nosso caso vamos fazer `sudo mount /dev/xvdf /dev/ebs/` e por último vamos listar os discos com `df -h`:

```
[ubuntu@ip-172-31-28-205:~$ sudo mkdir /dev/ebs
[ubuntu@ip-172-31-28-205:~$ sudo mount /dev/xvdf /dev/ebs/
[ubuntu@ip-172-31-28-205:~$ sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            488M    0  488M   0% /dev
tmpfs           100M  3.1M  97M   4% /run
/dev/xvda1       7.7G  954M  6.8G  13% /
tmpfs           496M    0  496M   0% /dev/shm
tmpfs           5.0M    0  5.0M   0% /run/lock
tmpfs           496M    0  496M   0% /sys/fs/cgroup
tmpfs           100M    0  100M   0% /run/user/1000
/dev/xvdf        99G   60M  94G   1% /dev/ebs
```

Pronto, podemos ver que nosso volume (`/dev/xvdf`) agora é reconhecido e a nossa instância ganhou um "novo HD". Na Alura usamos o EBS separados para gravar os arquivos de *logs* e para o *root template* das instâncias.