Sobre um serviço de web storage como o Amazon S3 é correto afirmar que as suas características principais são: (Selecione 3)

Parte superior do formulário

OK  
Alta escalabilidade.

 Com o Amazon S3, você pode armazenar a quantidade de dados desejada e acessá-los quando precisar. Não é mais necessário adivinhar as necessidades futuras de armazenamento.



OK  
Redundância dos dados.  
 O Amazon S3 proporciona infraestrutura durável para armazenar dados importantes e foi projetado para oferecer durabilidade de objetos de 99,999999999%. Seus dados são armazenados com redundância em várias instalações e diversos dispositivos em cada instalação.

   
  
Controle da infraestrutura de perto.  
 Quando usa-se um serviço de armazenamento é justamente o contrário que estamos querendo, ou seja, terceirizar esse controle e as cópias defensivas.

   
  
OK  
Baixo custo.  
 O Amazon S3 permite armazenar grandes quantidades de dados com custo muito baixo. Com o uso de políticas de ciclo de vida, que será visto no final do curso, pode-se definir políticas para migrar automaticamente dados para a categoria S3 Padrão – Acesso ocasional e para o Amazon Glacier.

   
  
Acessível somente pela interface web.  
O S3 disponibiliza APIs REST e SDKs completos para a integração fácil com tecnologias de terceiros.

No estudo do Amazon S3 é importante o conhecimento das características de um Bucket.

Das alternativas a seguir a que **NÃO** temos uma informação correta é:

São usados com unidade de agregação nos relatórios do S3.  
 No relatório de pagamento do S3, por exemplo, é através de um Bucket que se tem uma unidade de agregação de maneira que podemos controlar e verificar mais detalhadamente os custos com o armazenamento.

   
  
São configurados para existir em alguma região geográfica.

Na criação de um bucket temos a necessidade de definir a região de localização dele. Quanto mais próxima do usuário menos latência teremos e por isso é natural que façamos a escolha da região mais próxima

 É um container de objetos dentro do S3.

É dentro dele que salvamos e buscamos os nossos objetos e por isso é chamado de container de objetos.

 Alternativa correta



É o nome dado ao que guardamos e recuperamos do S3.

Objetos são os nomes que são usados para referenciar recursos guardados, recuperados e atualizados no S3.

 Alternativa correta



São únicos e portanto no mundo só existe um bucket com um determinado nome.

Um bucket tem a característica de ser único. Por isso na criação não se pode usar um nome que já tenha sido usado previamente. A necessidade é justamente porque uma vez criado um bucket a URL dos objetos a partir dele tem com única referência o nome usado. <http://alura.s3.amazonaws.com/foto.jpg>.

Além de um bucket ter o propósito de container de objetos deve-se ficar atento ao nome definido nele pois não existe a possibilidade de renomeá-lo. Além disso esse nome fica exposto na URL.

Usando essas informações escolha as alternativas que estejam, **CORRETAS** no que diz respeito às regras usadas:

 Podem seguir o estilo de IP, como: 192.168.5.4

O nome **não** pode ser formatado como um endereço IP como em: 192.168.5.4. Sobe pena de uma URL para um bucket ficar inconsistente, já que é possível configurá-lo no estilo *virtual-hosted–style*: [http://bucket.s3.amazonaws.com](http://bucket.s3.amazonaws.com/)

 Alternativa correta



Deve-se ficar atento a tamanho mínimo e máximo na definição do nome.

O nome deve ter pelo menos 3 e não mais que 63 caracteres.

 Alternativa correta



Pode-se usar todos os caracteres menos: barras(**/**) e pontos(**.**).

Os nomes podem ser seguidos por um um mais palavras, onde essas palavras podem ser separadas por um ponto(.) ou hífen(-). Além disso cada palavra, ou label, deve começar com letras em minúsculo ou um número.

 Alternativa correta



Pode-se usar números, letras e os únicos caracteres especiais aceitos são: **ponto(.)** e **hífen(-)**.

Os nomes podem ser seguidos por um um mais palavras, onde essas palavras podem ser separadas por um ponto(.) ou hífen(-). Além disso cada palavra, ou label, deve começar com letras em minúsculo ou um número.

 Alternativa correta



**Não** podem seguir o estilo de IP, como: 192.168.5.4

O nome **não** pode ser formatado como um endereço IP como em: 192.168.5.4. Sobe pena de uma URL para um bucket ficar inconsistente, já que é possível configurá-lo no estilo *virtual-hosted–style*: [http://bucket.s3.amazonaws.com](http://bucket.s3.amazonaws.com/)

Um bucket no S3 nos permite ter uma nova opção acessada através de Properties, no caso, Static Website Hosting. Sobre essa opção temos as seguintes afirmativas:

**a)** Permite que um bucket sirva com um simples servidor, sem a necessidade de implementar uma tecnologia de backend.

**b)** Permite que o usuário implemente uma tecnologia de backend para servir arquivos estáticos.

**c)** É através da opção Use this bucket to host a website que indicamos qual será a página HTML ponto de entrada e qual página de erro será devolvida em caso algum problema aconteça.

**d)** Permite gerar HTML dinâmicos

 A, B são falsas

A é verdadeira!

 Alternativa correta



B , D são falsas

Correto!

 Alternativa correta



A, D são falsas

A é verdadeira!

 Alternativa correta



A, C são verdadeiras!

Correto

Karen, após criar um Static Web Host, não consegue acessar o endpoint criado recebendo o código de erro 403 (Forbidden). Marque a opção que esclarecer o problema enfrentado por Karen.

Parte superior do formulário

* Alternativa correta



Quando mandamos um objeto para o bucket, por padrão ele não é público, por isso o erro 403 (forbiden).

Correto! Para resolver, ela precisa tornar público o acesso aos arquivos.

* Alternativa correta



Ela se enganou, pois não criou ainda um Static Web Host.

Falso, porque esta mensagem é para um Static Web Host sem permissão de acesso.

* Alternativa correta



Ela não fez upload do arquivo que esta tentando acessar.

Mesmo que estivesse feito, sem configurar o acesso, não poderá acessá-lo continuando a receber 403.

Parte inferior do formulário

Marque apenas as opções que necessitam permissões especiais de compartilhamento, caso contrário não serão acessíveis por usuários externos.

Parte superior do formulário

* Alternativa correta



Bucket

Correto!

* Alternativa correta



Static Web host

Correto!

* Alternativa correta



Objeto dentro de um bucket

Correto! O padrão é o acesso ser privado.

* Alternativa correta



Nenhuma das anteriores.

Na verdade, todas as opções citadas necessitam de permissão!

Parte inferior do formulário

Só podemos remover um bucket do nosso S3, tanto pelo CLI quanto pela interface web, se o mesmo estiver vazio. Mas pelo CLI, podemos forçar a remoção, fazendo com que, ao executar o comando **rb**, o mesmo remova todos os objetos do bucket, além dele em si.

Para isso, devemos adicionar o seguinte argumento ao comando:

Parte superior do formulário

* Alternativa correta



**--forced**

Alternativa errada! Quase lá, hein? O argumento **--forced** não existe! O argumento correto é o **--force**, que faz com que o comando remova tanto os objetos dentro do bucket quanto o próprio bucket.

* Alternativa correta



**--force**

Alternativa correta! Adicionando a opção, o argumento **--force**, o comando irá remover tanto os objetos dentro do bucket quanto o próprio bucket.

* Alternativa correta



**--purge**

Alternativa errada! O argumento **--purge** não existe! O argumento correto é o **--force**, que faz com que o comando remova tanto os objetos dentro do bucket quanto o próprio bucket.

* Alternativa correta



**mb**

Alternativa errada! O comando **mb** **cria** um bucket, e não o remove. O argumento correto é o **--force**, que faz com que o comando **rb** remova tanto os objetos dentro do bucket quanto o próprio bucket.

Parte inferior do formulário

Adicionar a opção **--force** faz com que o comando remova tanto os objetos dentro do bucket quanto o próprio bucket. Por exemplo, para remover o bucket **teste-alura**, incluindo os seus objetos, fazemos:

aws s3 rb s3://teste-alura --force

Os comandos do AWS CLI são bem parecidos com os comandos do mundo Unix. Dentre os comandos abaixo, qual **NÃO** funciona no CLI?

Parte superior do formulário

* Alternativa correta



ls

O comando **ls** existe. Ele serve para listar os buckets, por exemplo:

aws s3 lsCOPIAR CÓDIGO

Ou para listar os objetos que estão dentro de um bucket. Por exemplo, para listar os objetos que estão dentro do bucket **alura-bucket**:

aws s3 ls s3://alura-bucketCOPIAR CÓDIGO

* Alternativa correta



rm

O comando **rm** existe. Ele serve para remover algum objeto específico de um bucket.

Por exemplo, para remover o objeto **imagem.png**, que está dentro do bucket **alura-bucket**, fazemos:

aws s3 rm s3://alura-bucket/imagem.pngCOPIAR CÓDIGO

* Alternativa correta



mkdir

O comando **mkdir** **NÃO** existe. No mundo Unix, ele serve para criar uma pasta.

Por exemplo, para criar a pasta **aws** no **Desktop** do usuário **fabio**, fazemos:

mkdir /home/fabio/Desktop/awsCOPIAR CÓDIGO

* Alternativa correta



cp

O comando **cp** existe. Ele serve para copiar arquivos do S3 para nossa máquina local, ou o contrário, para copiar arquivos da nossa máquina local para o S3.

Por exemplo, para copiar o arquivo **imagem.png** do bucket **alura-bucket**, para o **Desktop** do usuário **fabio**, fazemos:

aws s3 cp s3://alura-bucket/imagem.png /home/fabio/Desktop/COPIAR CÓDIGO

E para fazer o contrário, para copiar o arquivo **imagem.png** do **Desktop** do usuário **fabio**, para o bucket **alura-bucket**, fazemos:

aws s3 cp /home/fabio/Desktop/imagem.png s3://alura-bucketCOPIAR CÓDIGO

* Alternativa correta



mv

O comando **mv** existe. Ele serve para mover ou renomear os arquivos do S3.

Por exemplo, no bucket **alura-bucket**, para renomear o arquivo **imagem.png** para **figura.png**, fazemos:

aws s3 mv s3://alura-bucket/imagem.png s3://alura-bucket/figura.png

Parte inferior do formulário

Sobre a sincronização entre um ambiente local e um bucket, julgue as afirmativas abaixo:

**1)** Para realizar a sincronização, é utilizado o comando **sync**.

**2)** Pode-se realizar a sincronização de um bucket com um ambiente local, trazendo todo o seu conteúdo para a nossa máquina local. Por exemplo:

aws s3 sync s3://alura-s3 /home/fabio/alura-s3COPIAR CÓDIGO

**3)** Se criarmos algum novo arquivo no nosso ambiente local, podemos fazer a sua sincronização com o bucket, fazendo assim o upload de todos os arquivos da nossa máquina local que não estão no bucket. Por exemplo:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3

upload: /home/fabio/alura-s3/novo-arquivo.txt to s3://alura-s3/novo-arquivo.txtCOPIAR CÓDIGO

**4)** Do mesmo jeito, se deletarmos algum arquivo no nosso ambiente local, podemos fazer a sua sincronização com o bucket, removendo os arquivos do bucket que não estão na nossa máquina local. Por exemplo:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3

delete: s3://alura-s3/novo-arquivo.txtCOPIAR CÓDIGO

Parte superior do formulário

* Alternativa correta



Nenhuma afirmativa é falsa.

Alternativa errada! A afirmativa **4** é falsa! Se deletarmos algum arquivo no nosso ambiente local, para fazer a sua sincronização com o bucket, removendo os arquivos do bucket que não estão na nossa máquina local, devemos adicionar a opção **--delete**:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3 --delete

delete: s3://alura-s3/novo-arquivo.txtCOPIAR CÓDIGO

* Alternativa correta



A afirmativa **2** é falsa.

Alternativa errada! A afirmativa **4** é falsa! Se deletarmos algum arquivo no nosso ambiente local, para fazer a sua sincronização com o bucket, removendo os arquivos do bucket que não estão na nossa máquina local, devemos adicionar a opção **--delete**:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3 --delete

delete: s3://alura-s3/novo-arquivo.txtCOPIAR CÓDIGO

* Alternativa correta



A afirmativa **4** é falsa.

Alternativa certa! Se deletarmos algum arquivo no nosso ambiente local, para fazer a sua sincronização com o bucket, removendo os arquivos do bucket que não estão na nossa máquina local, devemos adicionar a opção **--delete**:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3 --delete

delete: s3://alura-s3/novo-arquivo.txtCOPIAR CÓDIGO

* Alternativa correta



A afirmativa **1** é falsa.

Alternativa errada! Somente a afirmativa **4** é falsa! Se deletarmos algum arquivo no nosso ambiente local, para fazer a sua sincronização com o bucket, removendo os arquivos do bucket que não estão na nossa máquina local, devemos adicionar a opção **--delete**:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3 --delete

delete: s3://alura-s3/novo-arquivo.txtCOPIAR CÓDIGO

* Alternativa correta



A afirmativa **3** é falsa.

Alternativa errada! A afirmativa **4** é falsa! Se deletarmos algum arquivo no nosso ambiente local, para fazer a sua sincronização com o bucket, removendo os arquivos do bucket que não estão na nossa máquina local, devemos adicionar a opção **--delete**:

aws s3 sync /home/fabio/alura-s3 s3://alura-s3 --delete

delete: s3://alura-s3/novo-arquivo.txt

Parte inferior do formulário

Para criar um usuário com login gabriel, com tipo de acesso restrito ao uso do Console de Gerenciamento AWS eu preciso marcar a seguinte opção na página de inclusão de usuários do serviço IAM

Parte superior do formulário

* Alternativa correta



SDK Development Tool

Essa é uma das opções disponíveis quando o acesso programático do usuário está habilitado.

* Alternativa correta



Programmatic Access

Esse checkbox deve ficar desmarcado, uma vez que o tipo de acesso ao gabriel deve ser restrito ao console de gerenciamento do AWS.

* Alternativa correta



CLI Development Tool

Essa é uma das opções disponíveis quando o acesso programático do usuário está habilitado.

* Alternativa correta



AWS Management Console access

Isso aí!

Parte inferior do formulário

O usuário gabriel está incluído no grupo instrutores com política de total acesso ao S3 (AmazonS3FullAccess). Quando ele se logar no console, ele terá acesso a quais buckets?

Parte superior do formulário

* Alternativa correta



A todos os buckets.

Isso aí!

* Alternativa correta



Apenas aos buckets criados pelo grupo instrutores.

Como o acesso do grupo instrutores é total, ele herdará essa mesma permissão, dando acesso a todos os buckets.

* Alternativa correta



Apenas aos buckets criados por ele.

Parte inferior do formulário

Veja o código JSON abaixo:

{

"Version" : "2012-10-17",

"Statement" : [

{

"Sid": "LiberacaoDeAcessoAosBuckets",

"Effect": "Allow",

"Action": [

"s3:listAllMyBuckets"

],

"Resource": [

"arn:aws:s3:::\*"

]

}

]

}COPIAR CÓDIGO

Esse código corresponde a uma política:

Parte superior do formulário

* Alternativa correta



Com recurso que identifica todos os buckets identificados por LiberacaoDeAcessoAosBuckets

O recurso definido pelo arquivo relaciona todos os buckets ("arn:aws:s3:::\*")

* Alternativa correta



Com efeito de bloquear a listagem de todos os buckets

O efeito definido no arquivo é o de **Permitir**

* Alternativa correta



Com identificador LiberacaoDeAcessoAosBuckets

Isso aí!

* Alternativa correta



Que foi criada pelo usuário da Amazon em 2012-10-17

O campo versão no JSON se refere a versão da API criada pela Amazon e não a do usuário. Atualmente a API em uso é de 2012.

Parte inferior do formulário

Para definir uma política de acesso público de leitura ao bucket alura-s3, qual arquivo JSON abaixo é o mais adequado?

Parte superior do formulário

* Alternativa correta



{

"Id": "Policy123"

"Version" : "2012-10-17",

"Statement" : [

{

"Sid": "PoliticaDeBucket",

"Effect": "Deny",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::alura-s3/\*"

],

"Principal": "instrutores"

}

]

}

Opa, não é bem assim! Neste caso, estamos bloqueando o acesso de leitura para o grupo instrutores.

* Alternativa correta



{

"Id": "Policy123"

"Version" : "2012-10-17",

"Statement" : [

{

"Sid": "PoliticaDeBucket",

"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::\*"

],

"Principal": "\*"

}

]

}

Opa, não é bem assim! Nesse caso, estamos permitindo a leitura para todos os buckets.

* Alternativa correta



{

"Id": "Policy123"

"Version" : "2012-10-17",

"Statement" : [

{

"Sid": "PoliticaDeBucket",

"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": [

"arn:aws:s3:::alura-s3/\*"

],

"Principal": "\*"

}

]

}

Muito bem!

Parte inferior do formulário

Temos as seguintes afirmações a respeito do SDK do AWS:

**A)** O SDK é voltado exclusivamente para a linguagem Java.

**B)** Permite executar localmente, através do terminal, uma série de operações que executaríamos no painel de controle do S3.

**C)** Pode ser usado para automatizar o processo de upload de arquivos.

Podemos afirmar que:

Parte superior do formulário

* Alternativa correta



Apenas A é falsa.

Correto. Há suporte há várias linguagens e plataformas.

* Alternativa correta



C é falsa.

C é verdadeira.

* Alternativa correta



B é falsa.

B é verdadeira.

Parte inferior do formulário

Usando o SDK com Java

Em Java, a classe AmazonS3Builder é aquela que nos permite acessar o S3 de acordo com nossas necessidades. Para termos um acesso padrão, encadeamos uma chamada ao método standard(). Marque as alternativas verdadeiras a respeito do seu uso.

Parte superior do formulário

* Alternativa correta



Uma instância de BasicAWSCredentials recebe como parâmetro o acessKey e o secretKey da nossa conta da Amazon.

* Alternativa correta



Um AWSStaticCredentialProvider pode receber como parâmetro um BasicAWSCredentials.

* Alternativa correta



Precisamos das credenciais de acesso (accessKey e secretKey) para podemos nos conectar ao S3 através do SDK.

* Alternativa correta



Com um cliente criado através do AmazonS3Builder, podemos, por exemplo, obter todos os nosso buckets através do método getBuckets().

Errado, o método correto é listBuckets().

Parte inferior do formulário

Todas as operações que realizaremos com nossa conta do S3 são feitas com a instância criada a partir do AmazonS3ClientBuilder. Vimos como como listar buckets.

Temos as seguintes afirmações:

A) O método createBucket() é um dos métodos responsáveis pela criação de um bucket. Ele recebe uma String com o nome do bucket.

B) Com a representação de um arquivo através da classe java.io.File, podemos enviá-lo com o método putObject.

C) O método putObject recebe três parâmetros. O primeiro é o nome do bucket que receberá nosso arquivo. O segundo, o nome do arquivo que será adotado com o upload concluído. Por fim, o terceiro parâmetro é a instância de java.io.File que representa o arquivo que desejamos enviar.

D) O método listObject retorna o nome de todos os objetos salvos dentro de um bucket, por isso ele recebe como único parâmetro o nome do bucket.

Marque a alternativa verdadeira:

Parte superior do formulário

* Alternativa correta



A é falsa

* Alternativa correta



B é falsa

* Alternativa correta



A e C são falsas

* Alternativa correta



Apenas a D é falsa

A letra D é falsa! Ele recebe como parâmetro uma instância de ListObjectRequest. É a partir dessa instância que indicamos o bucket que será buscado através de withBucketName()

Parte inferior do formulário

Como vimos no vídeo do capítulo, por padrão o versionamento à nível de bucket se encontra desabilitado. Quando optamos pelo seu uso, todos os objetos ganham automaticamente um campo chamado Version ID.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Baseado no que foi visto no capítulo, assinale a alternativa correta:

Parte superior do formulário

* Alternativa correta



Com o uso de várias versões, é possível deixar o usuário livre pra escolher a versão que melhor lhe agrada.

O mecanismo de versionamento permite que o usuário final visualize apenas a última versão enviada.

* Alternativa correta



Caso alguém crie um objeto com o mesmo nome, é possível facilmente voltar à ultima versão que usávamos.

Caso alguém crie um objeto com o mesmo nome, é possível facilmente voltar à ultima versão que usávamos.

* Alternativa correta



As versões servem, no entanto, para coibir o risco de inconsistência em caso de alterações concorrentes.

Apesar de ser comum usar o versionamento para isso, não é possível **alterar** um objeto do bucket.

Parte inferior do formulário

No caso de uma sobrescrita (envio de um objeto de mesmo nome), é possível facilmente voltar à ultima versão que usávamos.

Ao remover um objeto em um bucket versionado, não significa que necessariamente ele foi removido do bucket. Por quê?

Parte superior do formulário

* Alternativa correta



Após a remoção, versionado ou não, é impossível acessar o objeto.

O **objeto** é apenas **marcado para remoção**, o que significa que podemos recuperar qualquer uma das suas versões à qualquer momento.

* Alternativa correta



A **versão mais atual** do objeto é apenas **marcado para remoção**, o que significa que o usuário pode ver qualquer uma das versões anteriores.

O **objeto** é apenas **marcado para remoção**, o que significa que podemos recuperar qualquer uma das suas versões à qualquer momento.

* Alternativa correta



O objeto é apenas **marcado para remoção**, o que significa que podemos recuperar qualquer uma das suas versões à qualquer momento.

O **objeto** é apenas **marcado para remoção**, o que significa que podemos recuperar qualquer uma das suas versões à qualquer momento.

Parte inferior do formulário

O **objeto** é apenas **marcado para remoção**, o que significa que podemos recuperar qualquer uma das suas versões à qualquer momento. Podemos fazer isso através do botão **Deleted Objects**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Vimos no último vídeo as classes de armazenamento, que são três: **Amazon S3 Standard**, **Amazon S3 Standard – IA** e **Amazon Glacier**. Elas estão diretamente relacionadas ao custo que teremos com o S3 e à disponibilidade dos nossos objetos armazenados.

Todas elas possuem uma grande durabilidade, tornando praticamente nula a chance de perdermos o nosso objeto. Mas uma dessas classes possui uma latência do primeiro byte mais alta, isto é, o tempo que leva para o byte trafegar na rede é maior que a das outras classes.

Essa classe é a:

Parte superior do formulário

* Alternativa correta



**Amazon S3 Standard**

Alternativa errada! A classe **Amazon S3 Standard** possui a latência do primeiro byte na casa dos milissegundos.

* Alternativa correta



**Amazon S3 Standard - IA**

Alternativa errada! A classe **Amazon S3 Standard - IA** possui a latência do primeiro byte na casa dos milissegundos.

* Alternativa correta



**Amazon Glacier**

Alternativa correta! A classe **Amazon Glacier** possui a latência do primeiro byte na casa dos minutos ou até horas.

Parte inferior do formulário

A alta latência do primeiro byte da classe **Amazon Glacier** faz com que essa ela seja mais focada em armazenar arquivos que quase não serão acessados, algo como um backup de servidor, ou algum arquivo necessário para um caso de problema, algo que não precisemos acessar rotineiramente.

Em compensação, o seu curso é consideravelmente menor comparado com as classes **Amazon S3 Standard** e **Amazon S3 Standard - IA**.

Sobre as classes de armazenamento do S3, pode-se dizer que estão **INCORRETAS** as alternativas:

Parte superior do formulário

* Alternativa correta



Temos a classe Glacier, que só está disponível no Brasil e serve para dados que são raramente acessados, já que a latência do primeiro byte pode ser até de horas.

Esta alternativa está incorreta, pois a classe Glacier ainda não está disponível no Brasil, mas estão corretas as informações da latência demorada e o objetivo dele

* Alternativa correta



Temos a classe Standard - IA, que é a indicada para objetos que são usados com baixa frequência e tem o custo mais alto do S3.

Esta alternativa está incorreta, pois a classe Standard-IA é usada para objetos com baixa frequência e tem um custo reduzido de armazenamento

* Alternativa correta



Temos 3 classes, onde a Standard é a mais indicada para objetos com baixa frequência de acesso.

Esta alternativa está incorreta, pois a classe Standard é indicada para objetos com muita frequência de acesso.

* Alternativa correta



Temos 3 classes, onde Standard - IA é indicada para objetos com menores frequências de acesso, e tem um custo reduzido de armazenamento se comparado ao Standard.

Esta alternativa está incorreta, pois a classe Standard - IA é realmente uma vantagem para objetos com baixa frequência de acesso e possui um custo vantajoso.

Parte inferior do formulário

Sobre as políticas de ciclo de vida, julgue as afirmativas abaixo:

1. Com o uso de políticas de ciclo de vida, podemos definir políticas para migrar automaticamente dados para a categoria S3 Standard - IA e para o Amazon Glacier (se disponível), conforme se aproximarem do fim da vida útil, definida por nós.
2. O uso de políticas de ciclo de vida reduz os nossos custos.
3. Quando uma política é definida, nossos dados são migrados automaticamente para a categoria de armazenamento escolhida, sem que nenhuma alteração seja feita à nossa aplicação.

Parte superior do formulário

* Alternativa correta



A afirmativa 1 é falsa.

Alternativa errada! Nenhuma afirmativa é falsa!

Com o uso de políticas de ciclo de vida, podemos definir políticas para migrar automaticamente dados para a categoria S3 Standard - IA e para o Amazon Glacier (se disponível), conforme se aproximarem do fim da vida útil, definida por nós, reduzindo assim os nossos custos.

Nossos dados são migrados automaticamente para a categoria de armazenamento escolhida, sem que nenhuma alteração seja feita à nossa aplicação.

* Alternativa correta



A afirmativa 2 é falsa.

Alternativa errada! Nenhuma afirmativa é falsa!

Com o uso de políticas de ciclo de vida, podemos definir políticas para migrar automaticamente dados para a categoria S3 Standard - IA e para o Amazon Glacier (se disponível), conforme se aproximarem do fim da vida útil, definida por nós, reduzindo assim os nossos custos.

Nossos dados são migrados automaticamente para a categoria de armazenamento escolhida, sem que nenhuma alteração seja feita à nossa aplicação.

* Alternativa correta



A afirmativa 3 é falsa.

Alternativa errada! Nenhuma afirmativa é falsa!

Com o uso de políticas de ciclo de vida, podemos definir políticas para migrar automaticamente dados para a categoria S3 Standard - IA e para o Amazon Glacier (se disponível), conforme se aproximarem do fim da vida útil, definida por nós, reduzindo assim os nossos custos.

Nossos dados são migrados automaticamente para a categoria de armazenamento escolhida, sem que nenhuma alteração seja feita à nossa aplicação.

* Alternativa correta



Nenhuma afirmativa é falsa.

Alternativa correta! Com o uso de políticas de ciclo de vida, podemos definir políticas para migrar automaticamente dados para a categoria S3 Standard - IA e para o Amazon Glacier (se disponível), conforme se aproximarem do fim da vida útil, definida por nós, reduzindo assim os nossos custos.

Nossos dados são migrados automaticamente para a categoria de armazenamento escolhida, sem que nenhuma alteração seja feita à nossa aplicação.

Parte inferior do formulário