

Life Cycle Policies

Transcrição

Nesse vídeo falaremos sobre **Gerenciamento de ciclo de vida** de um *bucket*, que se relaciona às classes de armazenamento. Poderemos determinar ciclos de tempo para que determinado objeto vá para uma classe específica. Ou para que um objeto expire e seja removido de um *bucket*.

Relembrando as nossas classes, vimos no vídeo anterior que temos três classes: *Standard*, a classe padrão, *Standard - IA*, classe padrão de acesso infrequente e *Glacier*, classe de armazenamento para objetos raramente acessados, porém esta ainda não está disponível no Brasil. E todas elas têm preços bem definidos por GB armazenado.

Imaginemos que temos salvos os logs de determinados sites no *alura-s3*, então os instrutores, além de carregar imagens no *bucket*, salvam os logs do nosso servidor internamente.

O que acontece hoje, no dia zero, é que um objeto, um arquivo de log, é criado. Daqui a 30 dias, como ele não será mais tão acessado (existem logs que nem abrimos mais, e ficam mais para uma auditoria, por exemplo), queremos fazer uma transição. Colocaremos este arquivo no *Standard - IA* para pagarmos menos, já que o acesso a ele será menos frequente.

Após 60 dias, colocaremos este arquivo no *Glacier*, pois provavelmente depois deste período o log estará basicamente servindo como backup. E se alguém precisar acessá-lo um dia, basta ir ao *Glacier*. Passado um ano da criação desse objeto, não faz mais sentido ficarmos pagando pelo seu armazenamento, portanto cabe eliminarmos o arquivo de vez.

Poderíamos fazer isso manualmente, pedindo para alguém - algum instrutor de desenvolvimento poderia ficar monitorando, mas isso também não faz sentido. No caso do S3 já existe algo implementado nesse sentido.

A seguir, veremos como implementar essas políticas de ciclo de vida que acabamos de ver, que se relacionam tanto à fase de transição de uma classe para outra quanto à fase de expiração, quando queremos remover um determinado objeto ou versão do mesmo.