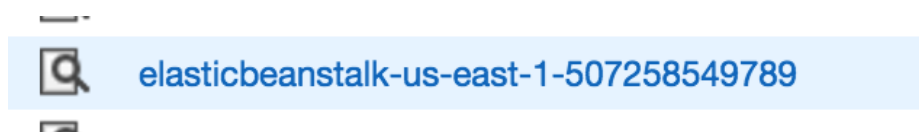


Unable to Delete Empty Elastic Beanstalk S3 Bucket

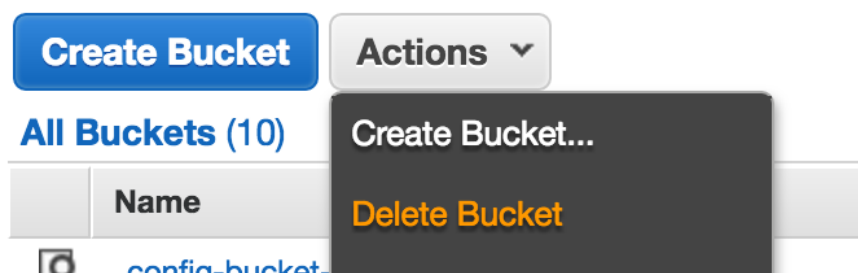
For those who are doing AWS work among the different projects, you will most likely do some storage on S3 (Simple Storage Service) for templates and logs. Each AWS service has the ability to write its configuration and logs to S3 and is usually a part of the setup wizard.

Sometimes the permissions set by the AWS wizard may leave you with some challenges. A common and simple example is when using AWS Elastic Beanstalk. When you clear out an Elastic Beanstalk configuration, the S3 bucket is left behind because it is not deleted as part of the removal process.

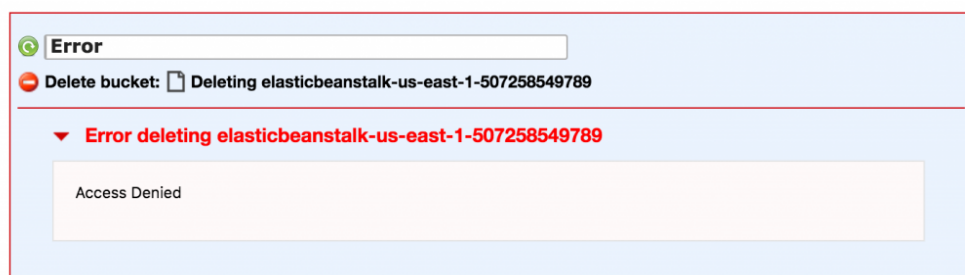
Normally, we just select the bucket and then you can empty it and delete it. This is what happens instead. First, select your bucket:



Once selected, we then choose the **Delete Bucket** option from the **Actions** button:



Then we are disappointed by seeing this error message:



Access Denied?! That shouldn't be the case. I'm using an account that does have enhanced privileges, and have even attempted it using the root level account for my entire AWS environment. NOTE: It's not recommended to use the root account, but I did try it to prove the point.

Fixing the S3 Bucket Access Denied Issue

The issue is a simple one as it turns out. Open up the **properties** for the bucket and click the **Edit bucket policy** button:

Bucket: elasticbeanstalk-us-east-1-507258549789 ×

Bucket: elasticbeanstalk-us-east-1-507258549789
Region: US Standard
Creation Date: Sun Sep 06 13:14:35 GMT-400 2015
Owner: eric

▼ Permissions

You can control access to the bucket and its contents using access policies. [Learn more.](#)

Grantee: eric List Upload/Delete View Permissions Edit Permissions ×

+ Add more permissions

🔑 Edit bucket policy

🔑 Add CORS Configuration

Save

Cancel

When the bucket is created by the system, it is created with a specific bucket policy that has been set to deny the **s3:DeleteBucket** action:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "eb-58950a",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:DeleteBucket",
      "Resource": "arn:aws:s3::ela"
    }
  ],
}
```

That's a safety measure so that we don't accidentally remove the contents which could be driving an active Elastic Beanstalk configuration. Change the **Deny** effect to **Allow** in the JSON editor and save the policy:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "eb-58950a",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:DeleteBucket",
      "Resource": "arn:aws:s3:::ela"
    },
  ],
}
```

Once you've saved the policy, go ahead with the **Delete bucket** process under the **Actions** menu again, and you will see a much more appropriate response. This time you will see a **Done** result in the results window.

 **Done**

 **Delete bucket:**  Deleting elasticbeanstalk-us-east-1-507258549789

This is one of those oddities around saving ourselves from ourselves by making sure we don't accidentally delete things. Sometimes we really do want to delete stuff ☹