

How to avoid a TooManyApplicationVersion exception on AWS Elastic Beanstalk?

Asked 12 years, 9 months ago Modified 1 year, 6 months ago

Viewed 15k times  Part of [AWS](#) Collective



29

Lately has anyone witnessed the

TooManyApplicationVersions Exception



on [AWS Elastic Beanstalk](#) console while deploying a new application version (war)? It's so annoying to see this message as it appears only after you have finished uploading the war.



I would be interested to know why this exception occurs and what precautions one should take to avoid such situations?



amazon-web-services

amazon-elastic-beanstalk

Share

Improve this question

Follow

edited Mar 6, 2012 at 20:48



Steffen Opel

64.6k ● 11 ● 195 ● 212

asked Mar 6, 2012 at 18:17



A Null Pointer

2,277 ● 3 ● 26 ● 29

- 2 If it helps anyone, there is a nicely written article on how to use the AWS SDK to delete the obsolete/old application versions when the limit is about to exceed.

danmandle.com/blog/... – A Null Pointer Nov 19, 2013 at 19:30

6 Answers

Sorted by:

Highest score (default)



Cause

30

The exception you are seeing stems from reaching your respective account limits for [AWS Elastic Beanstalk](#), see section *Errors* in [CreateApplicationVersion](#) [paraphrased]:



- **TooManyApplicationVersions** - *The caller has exceeded the limit on the number of application versions associated with their account.*
- **TooManyApplications** - *The caller has exceeded the limit on the number of applications associated with their account.*

The current limits are outlined in the respective FAQ [How many applications can I run with AWS Elastic Beanstalk?](#):

You can create up to 25 applications and **500 application versions**. By default you can run up to 10 environments across all of your applications. If you are also using AWS outside of Elastic Beanstalk, you may not be [...] **If you need more resources, complete the AWS Elastic Beanstalk request form and your request will be promptly evaluated.** *[emphasis mine]*

Solution

As emphasized, AWS offers the usual escalation option and allows you to submit a [Request to Increase AWS Elastic Beanstalk Limits](#), if you really need that many application versions to be available for reuse still. Otherwise you might just delete older ones you will not use anymore and the problem should vanish accordingly.

Good luck!

Share Improve this answer

edited Mar 6, 2012 at 20:08

Follow

answered Mar 6, 2012 at 20:03



[Steffen Opel](#)

64.6k ● 11 ● 195 ● 212

-
- 1 I've experienced this too - is there any automated way to delete old versions? Or to set up version deletion policy of

some sort? – petr Feb 19, 2014 at 15:15

@petr - nothing readily available from AWS that I know of, sorry. – Steffen Opel Feb 19, 2014 at 23:05

- 1 Beanstalk now supports automatically deleting old versions based on a lifecycle policy. See docs.aws.amazon.com/elasticbeanstalk/latest/dg/... – Odinodin Aug 31, 2017 at 9:02



Here's a one liner that uses the AWS CLI that will help you clear out old application versions:

21



```
aws elasticbeanstalk describe-application-versions --  
output text --query 'ApplicationVersions[*].  
[ApplicationName,VersionLabel,DateCreated]' | grep  
"2014-02" | while read app ver date; do aws  
elasticbeanstalk delete-application-version --  
application-name $app --version-label $ver --delete-  
source-bundle; done
```



Replace the grep with whatever date, (2013, 2014-01, 2014-02-0, etc) you see fit.

Share Improve this answer

answered Apr 30, 2014 at 3:01

Follow



tomberek

319 ● 2 ● 2

Would be great if this command were improved to delete all applications version no deployed minus the previous applications versions deployed (To keep the last one in emergency case). I'm trying to figure out a way to get the list

of applications version with information if was deployed or not, but describe-application-versions doesn't provide such information. Something similar to Elastic Beanstalk web console that shows the list of applications versions detailing where were deployed each one. – [Jose Nobile](#) Apr 19, 2015 at 23:28



15



As of EB CLI 3.3, you can now run the following to clear out old versions:

```
$ eb labs cleanup-versions
```

By default this will cleanup to the last 10 versions and/or older than 60 days. Adding `--help`, outputs the following:

```
usage: eb labs cleanup-versions [options...]

Cleans up old application versions.

optional arguments:
  --num-to-leave NUM      number of versions to leave
                          DEFAULT=10
  --older-than DAYS      delete only versions older
                          than x days DEFAULT=60
  --force                 don't prompt for
                          confirmation
```

Share Improve this answer

answered Sep 24, 2015 at 8:27

Follow



[Darren Shewry](#)

10.4k ● 4 ● 53 ● 47

-
- 1 Doesn't work in 3.7.8. This is what the AWS rep suggested also, but it simply doesn't work. He didn't know how to make

it work either. Pretty on par for the platform. – [lucian303](#) Dec 2, 2016 at 21:09

Worked OK for me using EB CLI 3.20.6 – [steinkel](#) May 19, 2023 at 8:20



You're approaching maximum number of versions and need to delete old or unused ones.

13



In the current web console you can simply do it on the Application Versions tab of your Beanstalk environment.



Elastic Beanstalk test [Create New Application](#)

[All Applications](#) > test

Environments

Application versions

<input type="checkbox"/>	Version Label	Description	Date Create	Source	Deployed To
<input type="checkbox"/>	Sample Application		2019-04-09	Sample Application	Test-env

Actions Settings Upload Refresh

Delete Deploy Manage tags

Share Improve this answer

[edited Apr 9, 2019 at 5:54](#)

Follow

answered Nov 10, 2012 at 15:24



[inmyth](#)

9,040 ● 4 ● 49 ● 52



This is the piece of code that we use in our deploy script to delete the oldest application version.

1



```
console.log('Deleting oldest application
version.');
```

```
params = {};
local.waitFor(function(done) {
    eb.describeApplicationVersions(params,
function(err, data) {
    if (err) {
        console.error(err, err.stack);
        local.abort('Could not retrieve the
list of application version.');
```

```
    } else {
        // This is probably not needed as the
list is already sorted but it is
        // not written anywhere that this will
always be the case
        function compare(a,b) {
            if (a.DateCreated > b.DateCreated)
                return -1;
            if (a.DateCreated < b.DateCreated)
                return 1;
            return 0;
        }
        var applicationsVersion =
data['ApplicationVersions'].sort(compare),
            oldestApplication =
applicationsVersion[applicationsVersion.length -
1],
                applicationName =
oldestApplication['ApplicationName'],
                versionLabel =
oldestApplication['VersionLabel'];
        params = {
            ApplicationName: applicationName,
/* required */
            VersionLabel:    versionLabel,
/* required */
            DeleteSourceBundle: false /* Do
not delete source bundle from S3 */
        };
    }
});
}
```

Documentation for the Elastic Beantalk API (js):

<http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/ElasticBeanstalk.html>

Share Improve this answer

answered Sep 2, 2015 at 13:51

Follow



null

4,340 ● 1 ● 22 ● 30



0



In my case:

>> The error only occurred with specific environments and/or applications. Other apps had no problems with the deploy.

>> I did not exceed the default beanstalk quotas. I validated it by counting them and comparing with the default values

environments count

```
aws elasticbeanstalk describe-environments --  
output text --query 'Environments[*].  
[EnvironmentName]' | wc -l
```

versions count

```
aws elasticbeanstalk describe-application-versions  
--output text --query 'ApplicationVersions[*].  
[ApplicationName]' | wc -l
```

default values

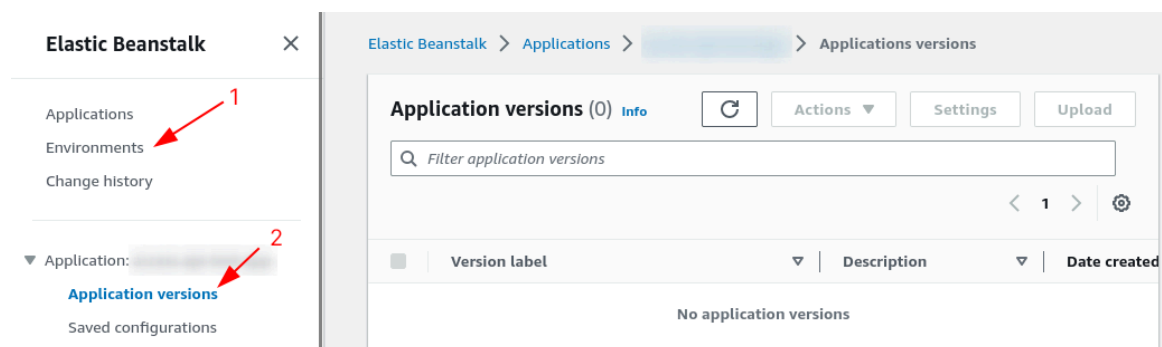
AWS Elastic Beanstalk

Service quotas			
<input type="text" value="Find quotas"/>			
	Quota name ▲	Applied quota value	AWS default quota value
<input type="radio"/>	Application versions	Not available	
<input type="radio"/>	Applications	Not available	
<input type="radio"/>	Configuration templates	Not available	
<input type="radio"/>	Custom platform versions	Not available	
<input type="radio"/>	Environments	Not available	

>> I also implemented an automated recycle like [update-application-resource-lifecycle](#)

>> I deleted several environments/versions but the error persisted

If you reach this line the only solution for me was to delete manually the environment and its versions. In my case the environment had only one version.



Share Improve this answer

answered May 29, 2023 at 17:49

Follow



JRichardsz

16.3k ● 7 ● 68 ● 107

