

Symptom

A Pachyderm repo was accidentally named starting with a dash (-) and the repository is treated as a command flag instead of a repository.

Recourse

Pachyderm supports standard `bash` utilities that you can use to resolve this and similar problems. For example, in this case, you can specify double dashes (`--`) to delete the repository. Double dashes signify the end of options and tell the shell to process the rest arguments as filenames and objects.

For more information, see `man bash` .

Failed Uploads

Symptom

A file upload, particularly a recursive one of many files, fails. You may see log messages containing the following in either pipeline logs, `pachd` logs, or from the `pachctl` command locally:

- `pachctl error:` an error occurred forwarding XXXXXX -> 650: error forwarding port 650
- `pachctl error:` EOF

- `pachd` or `worker`: all SubConns are in `TransientFailure`, latest connection error: connection error: desc = `\"transport: Error while dialing dial tcp 127.0.0.1:653: connect: connection refused\"`; retrying in `XXXX.XXXXXs`}]

Recourse

Either `pachd` or your pipeline's worker sidecar may be getting OOM killed as it grows while getting data from object storage.

You can give the storage container more resources by increasing the `cache_size` parameter in your pipeline spec. Increase it to what you can afford; its default is 64M. (If you're using a release prior to 1.10.0 and you have cluster-wide or namespace policies on resource limits, you may need to manually edit the pipeline RC.)

If it still gets OOM killed by k8s, there are a couple of environment variables you can set in your `pachd` deployment to limit the amount of memory the sidecar and `pachd` use.

- `STORAGE_UPLOAD_CONCURRENCY_LIMIT` limits the parallelism to put files into the storage backend. Default is 100.
- `STORAGE_PUT_FILE_CONCURRENCY_LIMIT` limits the number of parallel downloads `pachd` will initiate. Default is also 100.

You may use a binary search technique to hone in on a value

appropriate for a production pipeline:

for `cache_size` , max it out. If it works, halve it. If its OOM killed, increase the value by 50%. and so on for the

`CONCURRENCY_LIMITS` , halve and increase by 50% until you get a value that works.

Debug Pipelines