TROUBLESHOOTING

HTTP REQUEST ERRORS

Examining the access and error logs for the web server itself is probably the first step in identifying what is going on. If there is a 500 error, that usually indicates a problem communicating with the `radosgw` daemon. Ensure the daemon is running, its socket path is configured, and that the web server is looking for it in the proper location.

CRASHED `RADOSGW` PROCESS

If the `radosgw` process dies, you will normally see a 500 error from the web server (apache, nginx, etc.). In that situation, simply restarting radosgw will restore service.

To diagnose the cause of the crash, check the log in `/var/log/ceph` and/or the core file (if one was generated).

BLOCKED `RADOSGW` REQUESTS

If some (or all) radosgw requests appear to be blocked, you can get some insight into the internal state of the `radosgw` daemon via its admin socket. By default, there will be a socket configured to reside in `/var/run/ceph`, and the daemon can be queried with:

```
ceph --admin-daemon /var/run/ceph/client.rgw help

help                list available commands
objecter_requests   show in-progress osd requests
perfcounters_dump   dump perfcounters value
perfcounters_schema dump perfcounters schema
version             get protocol version
```

Of particular interest:

```
ceph --admin-daemon /var/run/ceph/client.rgw objecter_requests
...
```

will dump information about current in-progress requests with the RADOS cluster. This allows one to identify if any requests are blocked by a non-responsive ceph-osd. For example, one might see:

```
{ "ops": [
      { "tid": 1858,
        "pg": "2.d2041a48",
        "osd": 1,
        "last_sent": "2012-03-08 14:56:37.949872",
        "attempts": 1,
        "object_id": "fatty_25647_object1857",
        "object_locator": "@2",
        "snapid": "head",
        "snap_context": "0=[]",
        "mtime": "2012-03-08 14:56:37.949813",
        "osd_ops": [
              "write 0~4096"]},
      { "tid": 1873,
        "pg": "2.695e9f8e",
        "osd": 1,
        "last_sent": "2012-03-08 14:56:37.970615",
        "attempts": 1,
        "object_id": "fatty_25647_object1872",
        "object_locator": "@2",
        "snapid": "head",
        "snap_context": "0=[]",
        "mtime": "2012-03-08 14:56:37.970555",
        "osd_ops": [
```

```
              "write 0~4096"]}],
"linger_ops": [],
"pool_ops": [],
"pool_stat_ops": [],
"statfs_ops": []}
```

In this dump, two requests are in progress. The `last_sent` field is the time the RADOS request was sent. If this is a while ago, it suggests that the OSD is not responding. For example, for request 1858, you could check the OSD status with:

```
ceph pg map 2.d2041a48

osdmap e9 pg 2.d2041a48 (2.0) -> up [1,0] acting [1,0]
```

This tells us to look at `osd.1`, the primary copy for this PG:

```
ceph --admin-daemon /var/run/ceph/osd.1.asok
{ "num_ops": 651,
 "ops": [
       { "description": "osd_op(client.4124.0:1858 fatty_25647_object1857 [write 0~4096] 2.d2
         "received_at": "1331247573.344650",
         "age": "25.606449",
         "flag_point": "waiting for sub ops",
         "client_info": { "client": "client.4124",
             "tid": 1858}},
...
```

The `flag_point` field indicates that the OSD is currently waiting for replicas to respond, in this case `osd.0`.

JAVA S3 API TROUBLESHOOTING

PEER NOT AUTHENTICATED

You may receive an error that looks like this:

```
[java] INFO: Unable to execute HTTP request: peer not authenticated
```

The Java SDK for S3 requires a valid certificate from a recognized certificate authority, because it uses HTTPS by default. If you are just testing the Ceph Object Storage services, you can resolve this problem in a few ways:

1. Prepend the IP address or hostname with `http://`. For example, change this:

   ```
   conn.setEndpoint("myserver");
   ```

   To:

   ```
   conn.setEndpoint("http://myserver")
   ```

2. After setting your credentials, add a client configuration and set the protocol to `Protocol.HTTP`.

   ```
   AWSCredentials credentials = new BasicAWSCredentials(accessKey, secretKey);

   ClientConfiguration clientConfig = new ClientConfiguration();
   clientConfig.setProtocol(Protocol.HTTP);

   AmazonS3 conn = new AmazonS3Client(credentials, clientConfig);
   ```

405 METHODNOTALLOWED

If you receive an 405 error, check to see if you have the S3 subdomain set up correctly. You will need to have a wild card setting in your DNS record for subdomain functionality to work properly.

Also, check to ensure that the default site is disabled.

> [java] Exception in thread “main” Status Code: 405, AWS Service: Amazon S3, AWS Request ID: null, AWS Error Code: MethodNotAllowed, AWS Error Message: null, S3 Extended Request ID: null