RESTFUL PLUGIN

RESTful plugin offers the REST API access to the status of the cluster over an SSL-secured connection.

ENABLING

The *restful* module is enabled with:

```
ceph mgr module enable restful
```

You will also need to configure an SSL certificate below before the API endpoint is available. By default the module will accept HTTPS requests on port `8003` on all IPv4 and IPv6 addresses on the host.

SECURING

All connections to *restful* are secured with SSL. You can generate a self-signed certificate with the command:

```
ceph restful create-self-signed-cert
```

Note that with a self-signed certificate most clients will need a flag to allow a connection and/or suppress warning messages. For example, if the `ceph-mgr` daemon is on the same host,:

```
curl -k https://localhost:8003/
```

To properly secure a deployment, a certificate that is signed by the organization's certificate authority should be used. For example, a key pair can be generated with a command similar to:

```
openssl req -new -nodes -x509 \
  -subj "/O=IT/CN=ceph-mgr-restful" \
  -days 3650 -keyout restful.key -out restful.crt -extensions v3_ca
```

The `restful.crt` should then be signed by your organization's CA (certificate authority). Once that is done, you can set it with:

```
ceph config-key set mgr/restful/$name/crt -i restful.crt
ceph config-key set mgr/restful/$name/key -i restful.key
```

where `$name` is the name of the `ceph-mgr` instance (usually the hostname). If all manager instances are to share the same certificate, you can leave off the `$name` portion:

```
ceph config-key set mgr/restful/crt -i restful.crt
ceph config-key set mgr/restful/key -i restful.key
```

CONFIGURING IP AND PORT

Like any other RESTful API endpoint, *restful* binds to an IP and port. By default, the currently active `ceph-mgr` daemon will bind to port 8003 and any available IPv4 or IPv6 address on the host.

Since each `ceph-mgr` hosts its own instance of *restful*, it may also be necessary to configure them separately. The IP and port can be changed via the configuration key facility:

```
ceph config-key set mgr/restful/$name/server_addr $IP
ceph config-key set mgr/restful/$name/server_port $PORT
```

where `$name` is the ID of the ceph-mgr daemon (usually the hostname).

These settings can also be configured cluster-wide and not manager specific. For example,:

```
ceph config-key set mgr/restful/server_addr $IP
ceph config-key set mgr/restful/server_port $PORT
```

If the port is not configured, *restful* will bind to port `8003`. If the address it not configured, the *restful* will bind to `::`, which corresponds to all available IPv4 and IPv6 addresses.

LOAD BALANCER

Please note that *restful* will *only* start on the manager which is active at that moment. Query the Ceph cluster status to see which manager is active (e.g., `ceph mgr dump`). In order to make the API available via a consistent URL regardless of which manager daemon is currently active, you may want to set up a load balancer front-end to direct traffic to whichever manager endpoint is available.

AVAILABLE METHODS

You can navigate to the `/doc` endpoint for full list of available endpoints and HTTP methods implemented for each endpoint.

For example, if you want to use the PATCH method of the `/osd/<id>` endpoint to set the state up of the OSD id 1, you can use the following curl command:

```
echo -En '{"up": true}' | curl --request PATCH --data @- --silent --insecure --user <user> 'h
```

or you can use python to do so:

```
$ python
>> import requests
>> result = requests.patch(
       'https://<ceph-mgr>:<port>/osd/1',
       json={"up": True},
       auth=("<user>", "<password>")
   )
>> print result.json()
```

Some of the other endpoints implemented in the *restful* module include

- `/config/cluster`: **GET**
- `/config/osd`: **GET**, **PATCH**
- `/crush/rule`: **GET**
- `/mon`: **GET**
- `/osd`: **GET**
- `/pool`: **GET**, **POST**
- `/pool/<arg>`: **DELETE**, **GET**, **PATCH**
- `/request`: **DELETE**, **GET**, **POST**
- `/request/<arg>`: **DELETE**, **GET**
- `/server`: **GET**

THE /REQUEST ENDPOINT

You can use the `/request` endpoint to poll the state of a request you scheduled with any **DELETE**, **POST** or **PATCH** method. These methods are by default asynchronous since it may take longer for them to finish execution. You can modify this behaviour by appending `?wait=1` to the request url. The returned request will then always be completed.

The **POST** method of the `/request` method provides a passthrough for the ceph mon commands as defined in `src/mon/MonCommands.h`. Let's consider the following command:

```
COMMAND("osd ls " \
        "name=epoch,type=CephInt,range=0,req=false", \
        "show all OSD ids", "osd", "r", "cli,rest")
```

The **prefix** is **osd ls**. The optional argument's name is **epoch** and it is of type `CephInt`, i.e. `integer`. This means that you need to do the following **POST** request to schedule the command:

```
$ python
>> import requests
>> result = requests.post(
        'https://<ceph-mgr>:<port>/request',
       json={'prefix': 'osd ls', 'epoch': 0},
       auth=("<user>", "<password>")
   )
>> print result.json()
```