SYNC MODULES

New in version Kraken.

The Multisite functionality of RGW introduced in Jewel allowed the ability to create multiple zones and mirror data and metadata between them. `Sync Modules` are built atop of the multisite framework that allows for forwarding data and metadata to a different external tier. A sync module allows for a set of actions to be performed whenever a change in data occurs (metadata ops like bucket or user creation etc. are also regarded as changes in data). As the rgw multisite changes are eventually consistent at remote sites, changes are propagated asynchronously. This would allow for unlocking use cases such as backing up the object storage to an external cloud cluster or a custom backup solution using tape drives, indexing metadata in ElasticSearch etc.

A sync module configuration is local to a zone. The sync module determines whether the zone exports data or can only consume data that was modified in another zone. As of luminous the supported sync plugins are elasticsearch, `rgw`, which is the default sync plugin that synchronises data between the zones and `log` which is a trivial sync plugin that logs the metadata operation that happens in the remote zones. The following docs are written with the example of a zone using elasticsearch sync module, the process would be similar for configuring any sync plugin

- ElasticSearch Sync Module

REQUIREMENTS AND ASSUMPTIONS

Let us assume a simple multisite configuration as described in the Multisite docs, of 2 zones `us-east` and `us-west`, let's add a third zone `us-east-es` which is a zone that only processes metadata from the other sites. This zone can be in the same or a different ceph cluster as `us-east`. This zone would only consume metadata from other zones and RGWs in this zone will not serve any end user requests directly.

CONFIGURING SYNC MODULES

Create the third zone similar to the Multisite docs, for example

```
# radosgw-admin zone create --rgw-zonegroup=us --rgw-zone=us-east-es \
--access-key={system-key} --secret={secret} --endpoints=http://rgw-es:80
```

A sync module can be configured for this zone via the following

```
# radosgw-admin zone modify --rgw-zone={zone-name} --tier-type={tier-type} --tier-config={set
```

For example in the `elasticsearch` sync module

```
# radosgw-admin zone modify --rgw-zone={zone-name} --tier-type=elasticsearch \
                            --tier-config=endpoint=http://localhost:9200,num_shards=10,num_re
```

For the various supported tier-config options refer to the elasticsearch sync module docs

Finally update the period

```
# radosgw-admin period update --commit
```

Now start the radosgw in the zone

```
# systemctl start ceph-radosgw@rgw.`hostname -s`
# systemctl enable ceph-radosgw@rgw.`hostname -s`
```