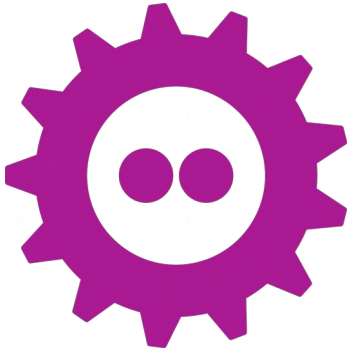


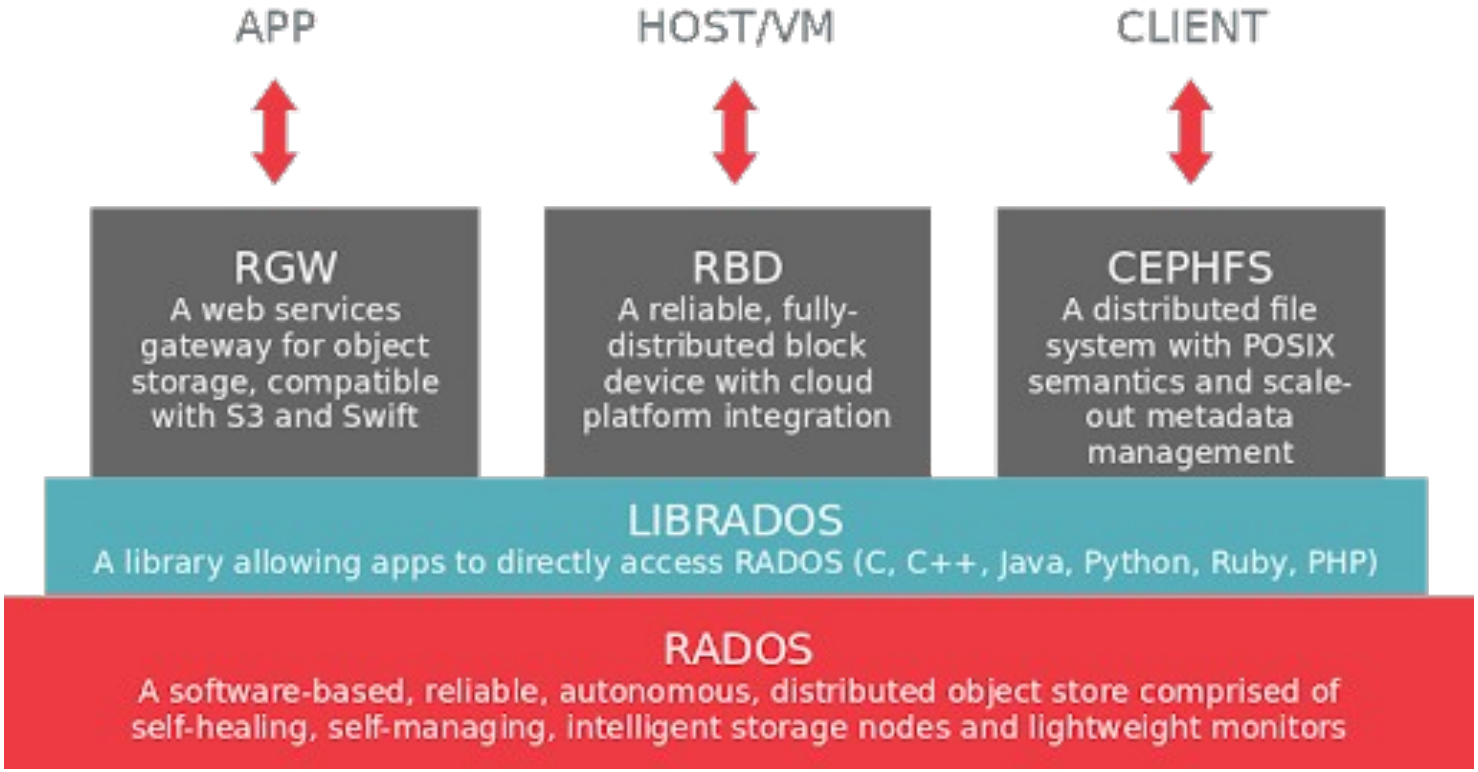
Secure Token Service (STS) in Ceph

Pritha Srivastava
Red Hat

Index

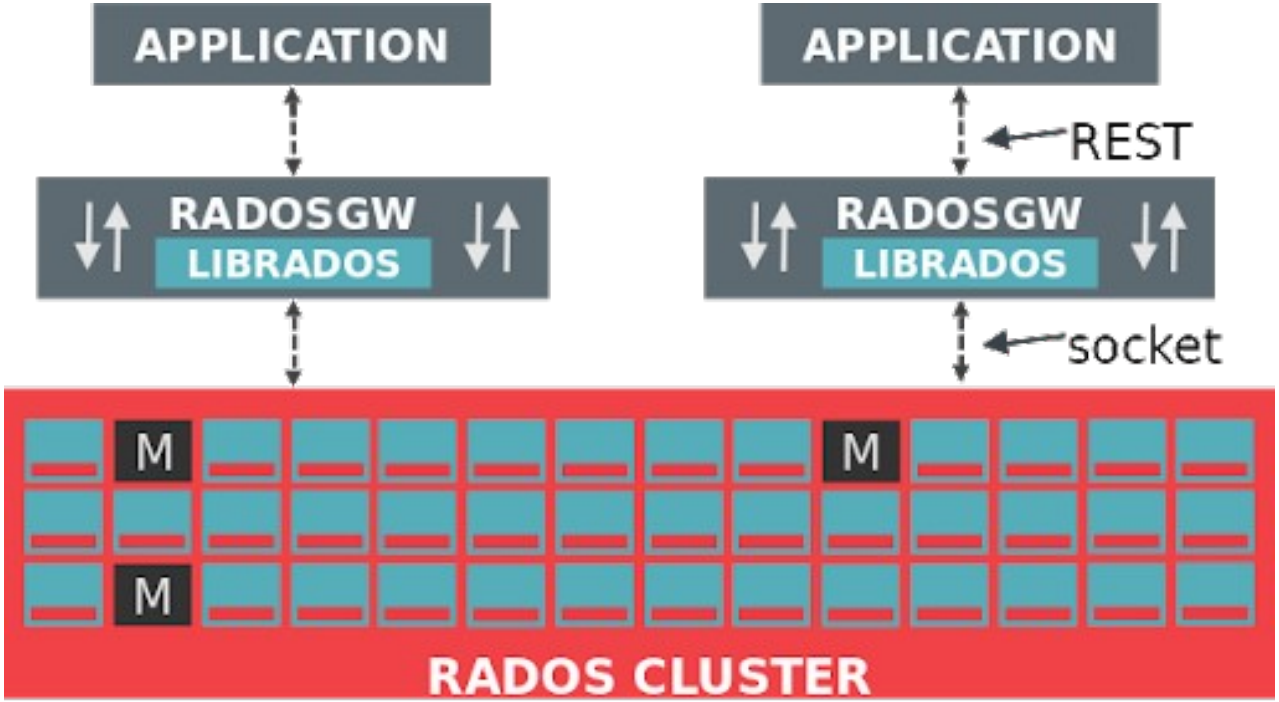
- What is Ceph
- What is Ceph Object Gateway
- AWS STS
- STS in Ceph Object Gateway
- Temporary Credentials
- Role
- AssumeRole
- AssumeRoleWithWebIdentity
- Advantages
- Restricting permissions
- Extending STS to STS Lite
- Future Work

What is Ceph


APP
HOST/VM
CLIENT
RGW
A web services gateway for object storage, compatible with S3 and Swift
RBD
A reliable, fully-distributed block device with cloud platform integration
CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management
LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)
RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

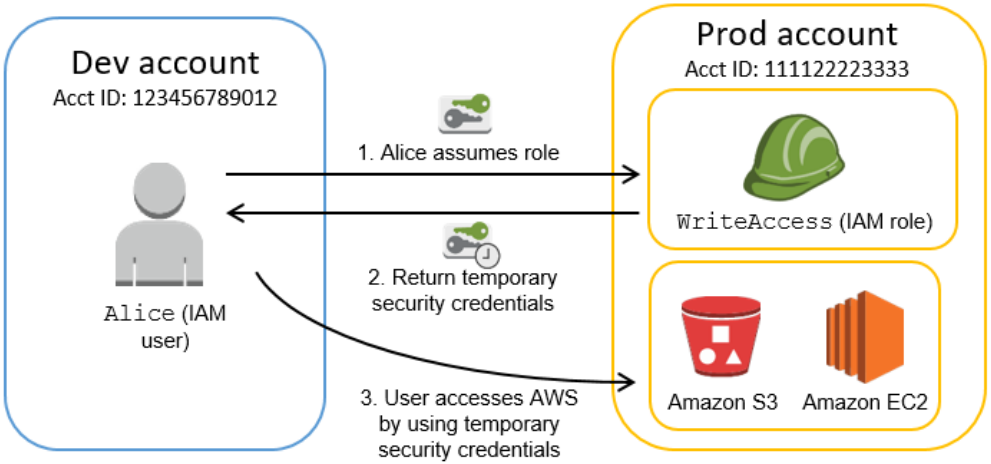
What is Ceph Object Gateway

A web services gateway for object storage, compatible with S3 and Swift



AWS STS

- Web service in AWS that returns temporary and limited-privilege credentials, when a user requests it.



STS in Ceph Object Gateway

- Implements AWS STS APIs related to cross account access and web identity federation.
- Supports authentication of temporary credentials.
- Implements some AWS IAM APIs related to ‘Role’ and its manipulation.
- Implements some AWS IAM APIs for attaching and validating IAM User Policy.
- STS and IAM APIs co-exist in same namespace as S3 APIs.
- Part of Nautilus release.

Temporary Credentials

- Access key id, secret access key, session token
- Signature v4/ v2 validation now takes into account session token also (x-amz-security-token)
- Session token
  - Opaque to end user
  - Encrypted using AES 128
  - Authentication and authorization information

Role

- Entity similar to a user, but can be ‘assumed’ by multiple users to get temporary credentials
- Trust policy – IAM policy that describes who is allowed to ‘assume’ a role.

```
{
  "Version": "2012-10-17",
  "Statement": {
  "Effect": "Allow",
  "Principal": { "AWS": "arn:aws:iam:::tester" },
  "Action": "sts:AssumeRole",
  }
}
```

Role

- Permission policy – IAM policy that describes what a role can or can not do.

```
{
   "Version": "2012-10-17",
   "Statement": {
   "Effect": "Allow",
   "Action": "s3:*",
   "Resource": "*"
   }
 }
```

- Support for inline permission policies only

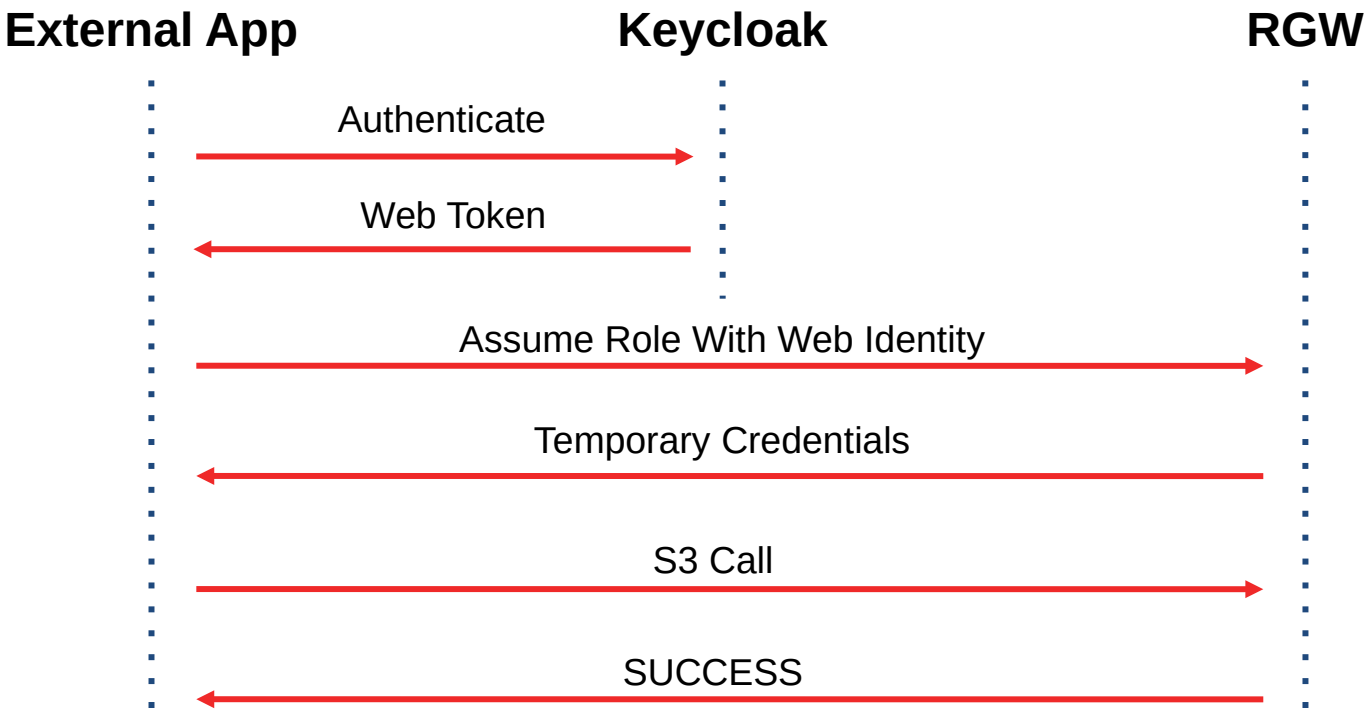
AssumeRole

- Provides cross account access to users.
- User ‘A’ can access a bucket owned by User ‘B’ by assuming a role ‘R’ that allows access to buckets owned by User ‘B’.

AssumeRoleWithWebIdentity

- Provides web identity federation.
- Users in an external application can access S3 resources without owning permanent credentials.
- Users need to authenticate with external OpenID Connect/ OAuth 2.0 compliant IDP.
- Keycloak supported currently.

AssumeRoleWithWebIdentity



Advantages

- Credentials are temporary – automatically expire after a duration.
- Provides limited access to S3 resources.
- Credentials are non-persistent in nature.

Restricting permissions further...

- Role may have permissions generic to all entities that may want to assume to it
- IAM Policy passed as parameter to AssumeRole*, can restrict permission specific to entity assuming role
- Final permission - result that is allowed by policy in STS API and permission policies attached to the role

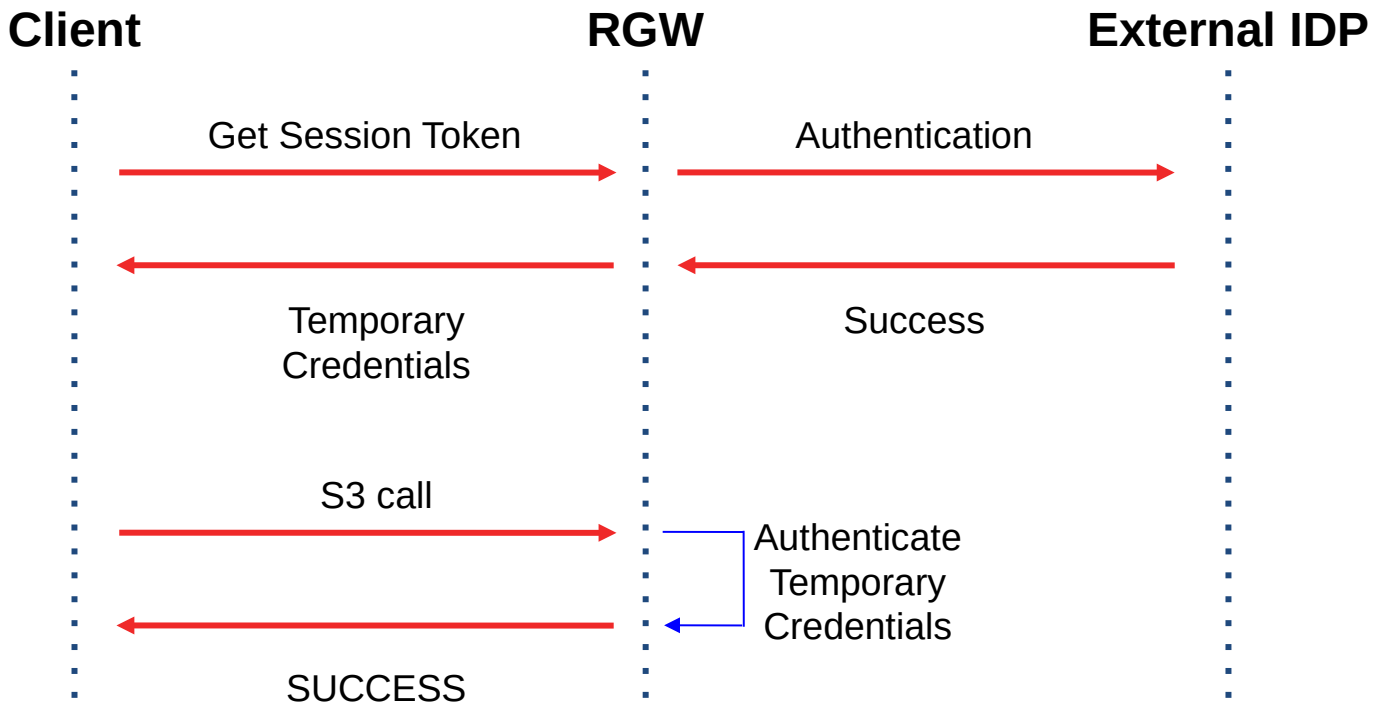
Restricting permissions further ...

- A Role can have a generic permission policy as follows:
  - {"Version":"2012-10-17","Statement": {"Effect":"Allow","Action":"s3:*","Resource":"*"}}
- A user assuming the above role can restrict permission to suits its needs like:
  - {"Version":"2012-10-17","Statement": {"Effect":"Allow","Action":"s3:ListAllMyBuckets","Resource":"*"}}

Extending STS to STS Lite

- In case of external authentication, each S3 call goes to an external IDP.
- STS Lite builds upon GetSessionToken
- Reduces latency and load on external IDPs

Extending STS to STS Lite



STS Lite

- Mainly implemented for Keystone.
- Works with LDAP and local authentication also.

Future Work

- AssumeRoleWithSAML
- Integration with other OpenID Connect/ OAuth 2.0 IDPs
- Integrate MFA with STS APIs
- Improve STS Key

Links

- https://docs.ceph.com/docs/master/radosgw/STSLite/
- https://docs.ceph.com/docs/master/radosgw/role/

Questions