```
[ceph: root@node /]# ceph config set client.rgw rgw_dns_name dns_suffix
```

where *dns_suffix* is the fully qualified domain name to be used to create your bucket's name.

In addition to configuring `rgw_dns_name`, you must configure your DNS server with a wildcard DNS record for that domain that points to the RADOS Gateway IP address. Syntax for implementing wildcard DNS entries varies for different DNS servers.

Using Amazon S3 API Clients

Commands from the `awscli` package support bucket and object management by using the S3 API. You can create a bucket by using the `aws mb` command. This example command creates the bucket called `demobucket`.

```
[ceph: root@node /]# aws s3 mb s3://demobucket
```

Upload objects to a bucket using the `aws cp` command. This example command uploads an object called `demoobject` to the `demobucket` bucket, using the local file `/tmp/demoobject`.

```
[ceph: root@node /]# aws --acl=public-read-write s3 cp /tmp/demoobject \
s3://demobucket/demoobject
```

The `radosgw-admin` command supports bucket operations, such as the `radosgw-admin bucket list` and the `radosgw-admin bucket rm` commands.

> **Note**
>
> There are multiple S3 public clients available, such as awscli, cloudberry, cyberduck, and curl, which provide access to object storage supporting the S3 API.

S3 Bucket Versioning, Life Cycle, and Policies

S3 bucket versioning supports storing multiple versions of an object in a bucket. RADOS Gateway supports versioned buckets, adding a version identifier to objects uploaded to the bucket. The bucket owner configures the bucket as a versioned bucket.

RADOS Gateway also supports S3 API object expiration by using rules defined for a set of bucket objects. Each rule has a prefix, which selects the objects, and a number of days after which objects become unavailable.

RADOS Gateway supports only a subset of the Amazon S3 API policy language applied to buckets. No policy support is available for users, groups, or roles. Bucket policies are managed through standard S3 operations rather than using the `radosgw-admin` command.

An S3 policy uses JSON format to define the following elements:

- The `Resource` key defines the resources which permissions the policy modifies. The policy uses the Amazon Resource Name (ARN) associated with the resources to identify it.
- The `Actions` key defines the operations allowed or denied for a resource. Each resource has a set of operations available.
- The `Effect` key indicates if the policy allows or denies the action previously defined for a resource. By default, a policy denies the access to a resource.