April 2020

IBM® TS7700 Series Cloud Storage Tier Setup Guide with Public Cloud Service and NGINX

Owner: Sosuke Matsui
e34975@jp.ibm.com
IBM Systems Development, IBM Japan

Contents

1 Introduction

This document shows how to setup TS7700 with IBM Cloud Object Storage Public and RStor Cloud Storage Service (https://rstor.io) using NGINX as a reverse proxy server.

1.1 Summary of Changes

Version 1.0

- Initial version

1.2 Configurations Supported

Cloud Storage Tier with IBM Cloud Object Storage Public and Rstor Cloud Storage Service is supported on TS7700C model at code level 8.42.2.12 or a later release.

2 TS7700C overview

TS7700 has supported physical tape as a backend storage for archiving and backup purpose. Similar to physical tape, TS7700 now supports cloud storage service as a backend storage. This model is called TS7700C, and users can select IBM Cloud Object Storage (ICOS) on-premise or Amazon S3 as a backend cloud storage service.

In addition to ICOS on-premise and Amazon S3, users can now select ICOS public and RStor Cloud Storage Service with support of NGINX reverse proxy server. NGINX is open source software that supports reverse proxying. Linux and Windows versions are available as of today.

In the following sections, we explain how to set up a cloud storage tier with ICOS public and RStor Cloud Storage Service with NGINX.

3 Setting up NGINX server for TS7700C

In this section, we will explain how to set up NGINX reverse proxy server to configure ICOS public or RStor Cloud Storage Service as a cloud storage tier for TS7700C. Complete the following steps:

1. Get a public cloud service account.
2. Install nginx to the Linux server. The example below shows compiling and installing nginx from source.

```
[root@y21c ~]# tar zxvf nginx-1.16.0.tar.gz
[root@y21c ~]# cd nginx-1.16.0
[root@y21c nginx-1.16.0]#  ./configure --sbin-path=/usr/local/nginx/sbin/nginx --conf-
path=/usr/local/nginx/conf/nginx.conf --pid-path=/usr/local/nginx/nginx.pid --with-
http_ssl_module --with-stream --with-mail=dynamic
[root@y21c nginx-1.16.0]#  make
[root@y21c nginx-1.16.0]#  make install
```

3. Add "/usr/local/nginx/sbin to $PATH

```
[root@y21c ~]# tail .bashrc
```

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
PATH=$PATH:/usr/local/nginx/sbin/
```

4. Create a self-signed certificate on the nginx server. First generate a private key. The example below follows the instruction in the link.
   https://docs.oracle.com/cd/E52668_01/E66514/html/ceph-issues-24424028.html

```
[root@y21c ~]# openssl genrsa -out nginx_private.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................+++
.....+++
e is 65537 (0x10001)
```

5. Create a copy of OpenSSL configuration file.

```
[root@y21c ~]# cp /etc/pki/tls/openssl.cnf .
```

6. Update the configuration file at ./openssl.cnf to make the following changes:

- Add "req_extensions = v3_req" to the section [req]
- Add "subjectAltName = @alt_names" to the section [v3_req]
- Add the [alt_names] section as follows:

```
[ alt_names ]
DNS.1 = <IP address of your nginx server>
```

7. Create a certificate signing request.

```
[root@y21c ~]# openssl req -new -key ./nginx_private.key -out nginx.csr -extensions
v3_req -config openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:AZ
Locality Name (eg, city) [Default City]:Tucson
Organization Name (eg, company) [Default Company Ltd]:IBM
Organizational Unit Name (eg, section) []:Systems
Common Name (eg, your name or your server's hostname) []:<IP address of your nginx
server>
Email Address []:<your email address>

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

8. Create a new configuration file v3.cnf to create a self-signed certificate with v3 extension.

```
[root@y21c ~]# cat v3.cnf
[v3_req]
subjectAltName = @alt_names
[alt_names]
DNS.1 = <IP address of your nginx server>
```

9. Create a self-signed certificate.

```
[root@y21c ~]# openssl x509 -req -days 730 -in nginx.csr -signkey ./nginx_private.key -
out nginx.crt -extensions v3_req -extfile v3.cnf
Signature ok
subject=/C=US/ST=AZ/L=Tucson/O=IBM/OU=Systems/CN=9.68.109.25/emailAddress=e34975@jp.ibm.c
om
Getting Private key
[root@y21c ~]#
```

10. Move the self-signed certificate and the private key to the usable location on the nginx server.

```
[root@y21c ~]# mv nginx.crt /etc/ssl/certs/
[root@y21c ~]# mv nginx_private.key /etc/ssl/private/
```

11. Update nginx.conf file to set up a reverse proxy. Set <Public cloud storage service endpoint> to your endpoint URL. If you are using IBM COS public, endpoint URL can be found in your service credential as shown in this page. If you are using RStor, endpoint URL can be found in this link.

```
[root@y21c ~]# cat /usr/local/nginx/conf/nginx.conf
user root;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
worker_rlimit_nofile 10240;

events {
    worker_connections 10240;
    accept_mutex off;
    multi_accept off;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    log_format upstreamlog '[$time_local] $remote_addr $host $upstream_addr '
                           '$upstream_cache_status $upstream_status `
                           `$upstream_http_location $request $http_authorization';

    access_log  /var/log/nginx/access.log  main;
    access_log /var/log/nginx/upstream.log upstreamlog;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   300;
    types_hash_max_size 2048;

    include             /usr/local/nginx/conf/mime.types;
    default_type        application/octet-stream;
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;
    proxy_http_version 1.1;
    proxy_buffering off;
    proxy_intercept_errors on;

    # IBM COS Endpoints
    upstream backend {
        server <Public cloud storage service endpoint>:443 max_fails=5 fail_timeout=60s;
    }
```

```
    server {
        listen       443 ssl;
        ssl_certificate "/etc/ssl/certs/nginx.crt";
        ssl_certificate_key "/etc/ssl/private/nginx_private.key";
        ssl_session_tickets off;
        ssl_session_cache off;
        ssl_ciphers AES128-SHA256;
        ssl_prefer_server_ciphers on;
        client_max_body_size 30G;

        keepalive_timeout 300s;
        keepalive_requests 1000000;

        server_name $host;
        proxy_buffering off;

        ignore_invalid_headers off;
        etag off;
        location / {
            proxy_pass https://backend;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_http_version 1.1;
            proxy_set_header Connection "";
            etag off;
        }
    }
}
```

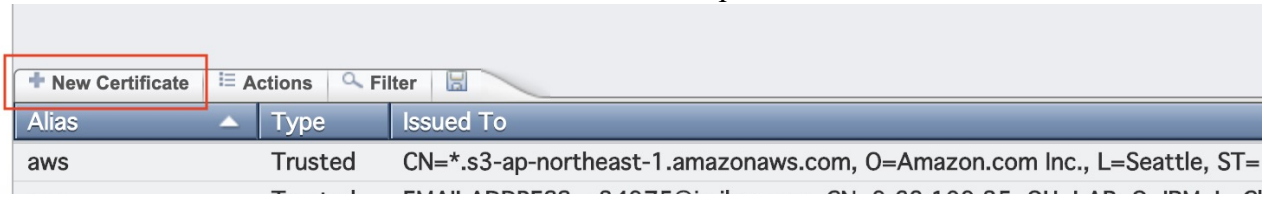
12. Verify the nginx server is set up correctly.

```
[root@y21c ~]# nginx -t
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
[root@y21c ~]#
```

13. Restart the nginx server.

```
[root@y21c ~]# service nginx restart
Redirecting to /bin/systemctl restart nginx.service
[root@y21c ~]#
```

14. On TS7700, select "Access" -> "SSL Certificates" and press "New Certificate".



15. Select "Upload a certificate file" and press "Next".

16. Download /etc/ssl/certs/nginx.crt from the nginx server to your laptop.

17. Press "Upload" on TS7700 MI, select the certificate file you downloaded, and press "Next".

18. Enter an Alias and press "Finish".

4 Setting up a cloud storage tier with IBM Cloud Object Storage public

This section shows how to set up a cloud storage tier with IBM Cloud Object Storage public using the NGINX reverse proxy server you have created in the previous section. Please refer to "IBM TS7700 R5.0 Cloud Storage Tier Guide" for planning and prerequisites. The steps to set up a cloud storage tier is shown below:

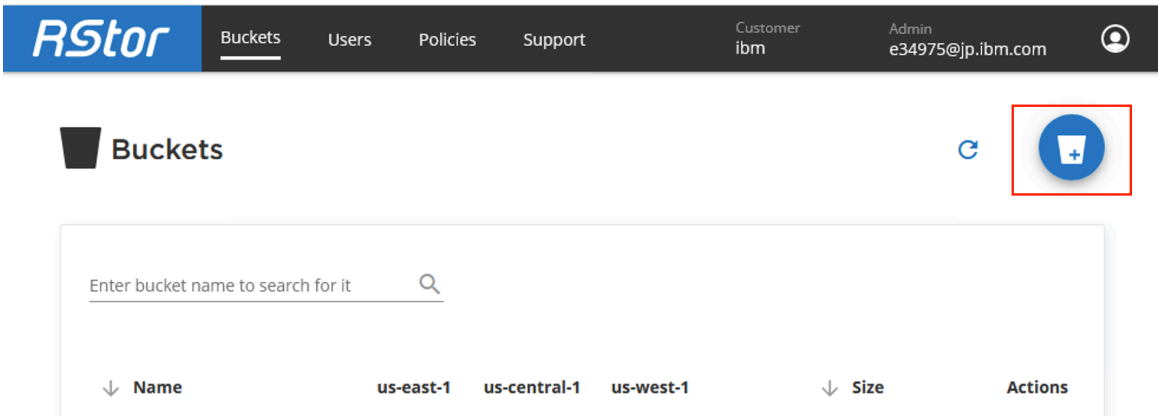
1. Create a vault in IBM COS public. If AWS Command Line Interface installed to your laptop, you can follow the steps below to create a vault:
    1. Update "<Your home directory>/.aws/credentials" and add your access key ID and secret access key like below:
    ```
    [cospublic]
    aws_access_key_id = <your access key ID>
    aws_secret_access_key = <your secret access key>
    ```
    2. Run the command below to create a vault in IBM COS public:
    ```
    $ aws --profile=cospublic --endpoint-url=<your NGINX URL>
    s3api create-bucket --bucket <your IBM COS public vault name>
    ```

2. Follow the steps show in Chapter 9 "Configuring TS7700 for cloud storage tier" to set up cloud storage tier with IBM COS public.
3. Enter the name of the vault you created above when creating a container using MI.
4. Enter the URL of the NGINX server and select a SSL certificate you created in the previous section when creating a cloud URL using MI.

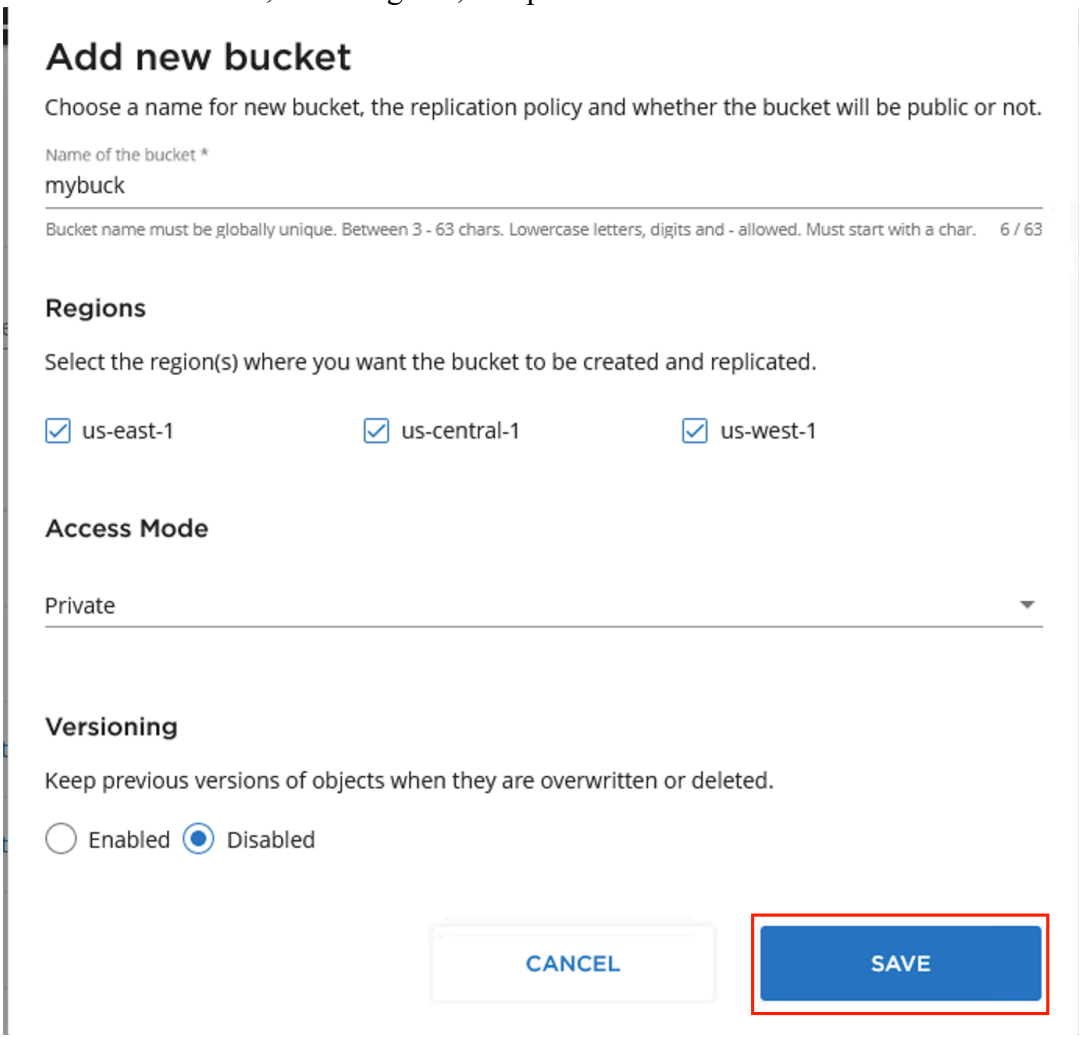
5 Setting up a cloud storage tier with RStor Cloud Storage Service

This section shows how to set up a cloud storage tier with RStor Cloud Storage Service using the NGINX server. Please refer to “IBM TS7760 R4.2 Cloud Storage Tier Guide” for planning and prerequisites. The steps to set up a cloud storage tier is shown below:

1. Press “Add new bucket” in the top right corner and create a bucket using RStor web interface.



2. Enter a bucket name, select regions, and press “SAVE”.



5. Follow the steps show in Chapter 9 “Configuring TS7700 for cloud storage tier” to set up cloud storage tier with RStor.
6. Enter the name of the bucket you created above when creating a container using MI.
7. Enter the URL of the NGINX server and select a SSL certificate you created in the previous section when creating a cloud URL using MI.

6 Considerations on settings

This section shows considerations on TS7700 and NGINX settings.

6.1 TS7700 setting

TS7700 may not be able to recall a logical volume from the cloud if the network bandwidth between nginx and COS public is too narrow. By default, the number of cloud recall processes (CRCCNT) is 20 and a cloud recall timeout value to recall 1 GiB of data (CRCTOUT) is 1800 seconds. If you follow these default settings, you can estimate minimum required bandwidth between nginx and COS public.

For example:

"Minimum required network bandwidth" = 1 GiB * 20 processes / 1800 seconds = 11.38 MB/sec

Please check your network bandwidth and TS7700's setting to see if cloud recall does not cause a timeout error. You can check your cloud recall settings by using the LIBRARY REQUEST CLDSET command.

```
"CLOUD SETTINGS V1 .0                                                    "
" CPMCNTH  =        40    CPMCNTL  =         0                          "
" CLDPRIOR =         0                                                   "
" CRCCNT   =        20                                                   "
" CDELCNT  =         0                                                   "
" CPMTOUT  =      1800                                                   "
" CRCTOUT  =      1800                                                   "
" CDELTOUT =      3600                                                   "
" ------------------------------------------------------------------------"
" CENABLMT Controls                                                      "
"   CLDPM    =   ENABLED                                                 "
"   CLDRCALL =   ENABLED                                                 "
"   CLDDEL   =   ENABLED                                                 "
```

If your network bandwidth cannot meet the minimum requirement, you can change cloud recall timeout value (CRCTOUT) by the following library request command:

LIBRARY REQUEST,*distributed_library*,CLDSET,CRCTOUT,*value*

6.2 NGINX setting

In the nginx configuration file, there is a setting called client_max_body_size which defines maximum allowed size of client request body. By default, this value is set to 1 MB. http://nginx.org/en/docs/http/ngx_http_core_module.html#client_max_body_size Make sure your logical volume size is below this value if you want to pre-migrate it to public cloud service. Follow

the steps below to change client_max_body_size value:

1. Add a client_max_body_size setting to your nginx.conf file in http, server, or location context.

   ```
   client_max_body_size 30GB;
   ```

2. If nginx server is already running, reload the new setting by running "nginx -s reload" command. If nginx server is not started yet, run "/usr/bin/nginx" to start the server.

Refer to "NGINX - IBM COS Solution Guide" for other nginx settings. https://cdn-1.wp.nginx.com/wp-content/files/nginx-pdfs/NGINX-IBM-COS-Solution-Guide.pdf