Replication

The Replication Service handles the transfer of data to a Chronopolis Node. It does this by querying the Ingest Server in order to discover the collections it needs to process and transfer to preservation storage. Once this is complete, it runs an initial audit on an ACE AM server local to the Chronopolis Node.

Links

- Gitlab: Chronopolis-Core

Installation

Prereqs

- ACE AM - The replication service will need to send requests to an ACE AM web application in order to register collections
- rsync - Transfers are done using rsync
- SSH key exchange - ssh is used for authentication so keys must be exchanged to any nodes planning on distributing content so that file transfers can occur
- Preservation Storage - The replication service pulls from the ingest server directly to your preservation area on a POSIX file system. We may expand this In the future, but for now, it is only on local disk.

RPM

Download and install the latest rpm

Running

Running can be done with the provided init scripts

- EL6: service replicationd start
- EL7: systemctl start replicationd

Installation Notes

RHEL6 Installed Files

```
/etc/init.d/replicationd
/usr/local/chronopolis/replication
/usr/local/chronopolis/replication/application.yml
/usr/local/chronopolis/replication/replicationd.jar
```

RHEL7 Installed Files

```
/usr/lib/systemd/system/replicationd.service
/usr/local/chronopolis/replication
/usr/local/chronopolis/replication/application.yml
/usr/local/chronopolis/replication/replicationd-prepare
/usr/local/chronopolis/replication/replicationd.jar
```

Preserved Files

As part of the install process by yum, the following files will not be overwritten

- /usr/local/chronopolis/replication/application.yml

User Creation

A service account is also needed as part of the install process who can write to /var/log/chronopolis and the preservation storage defined in the configuration. This is no longer handled by the rpm installation process and must be done manually. By default, the init scripts will look for a `chronopolis` user, and if it is not found fail. These can be updated in the following places:

- EL6: /etc/init.d/replicationd
- EL7: /usr/lib/systemd/system/replicationd.service

Configuration

The replicationd service reads the configuration file in /usr/local/chronopolis/replication/application.yml

application.yml

```
# Replication Service Configuration

# Replication Cron Job Configuration
# The rate at which to poll the ingest server for replications
replication.cron: 0 0 * * * *

# General Configuration Options
# node: the name to use when sending notification messages
# workDirectory: directory used to store temporary data while processing a replication
# maxFileTransfers: the maximum number of rsyncs which can run at once
# send-on-success: flag to enable sending notification on successful replications
# rsync.profile: the rsync profile to use, SINGLE or CHUNKED
# rsync.arguments: arguments to pass to created rsync processes
chron:
  node: chron
  workDirectory: /tmp/chronopolis
  maxFileTransfers: 2
  smtp.send-on-success: true
  rsync:
    profile: SINGLE
    arguments:
      - "-aL"
      - "--stats"

# ACE-AM Configuration
# am: the endpoint of the Audit Manager application
# username: the username to connect to the Audit Manager with
# password: the password to connect to the Audit Manager with
ace:
  am: http://localhost:8080/ace-am/
  username: user
  password: change-me

# Ingest API Configuration
# endpoint: the endpoint of the Ingest Server
# username: the username to connect to the Ingest Server with
# password: the password tot connect to the Ingest Server with
ingest.api:
  endpoint: https://localhost:8080/ingest/
  username: ingest-user
  password: change-me

# Preservation Storage Configuration: Only posix supported at this time
# posix: a list of Storage Filesystems available
#   id: the id of the Storage Filesystem (optional for replication - Storage does not need to be registered
with the Ingest Server)
#   path: the path on disk to the Storage FS
storage.preservation:
  posix:
    - id: 1
      path: /preservation-isilon/bags/
    - id: 2
      path: /preservation-xfs/bags/

# Replication Space Limit
# By default replication will stop if less than "warn" free space, where warn defaults to 0.1,
# which might seem close to the limit, but when you have petabyte storage, there is still a lot of room left.
# TDL adjusted their warn
storage.preservation:
  posix:
    - id: 1
      path: /chronopolis
      warn: 0.05

# Misc ACE configuration
# timeout: the timeout in Minutes for HTTP communication with the Audit Manager
ace.timeout: 5
```

```
# SMTP Configuration
smtp:
  send: true
  to: chron-support-l@mailman.ucsd.edu
  from: localhost
  host: localhost.localdomain

# Specify the active profile for loading various services, normally production
# Do not need to be changed
spring.profiles.active: production
spring.pid.file: /var/run/replicationd.pid

# Logging properties
# Can be modified if errors occur
# org.chronopolis can be changed to INFO if less logging is wanted
logging.file: /var/log/chronopolis/replication.log
logging.level:
  org.springframework: ERROR
  org.hibernate: ERROR
  org.chronopolis: DEBUG
```

rsync configuration SINCE 3.0.0

The replication service now has the ability to create multiple rsyncs when transferring a single bag. This is done with the `chron.rsync` properties. Currently there are two rsync profiles which can be chosen from, `SINGLE` and `CHUNKED`.

- Choosing `SINGLE` will run the standard flow which runs only one rsync per Bag.
- Choosing `CHUNKED` will run a newer rsync flow which will query the Ingest Server for all files in a Bag, and create batches of transfers to work on. Currently this is done in a naive manner by chunking ~10% of the given file listing, which can be inefficient for smaller collections. In addition, the `chron.maxFileTransfers` property may take some experimentation to find the best value for optimizing saturation of the network link.

The arguments passed to rsync should be edited with care, as the defaults should work for all workflows. In recent versions of rsync, commas have been introduced into the output and can be disabled with `--no-human-readable`.

SMTP Configuration

The replication service sends email to the `smtp.to` when a bag fails to replicate. The `chron.node` value is used in order to add information to the title of the email about which node the email came from.

In the event an email is wanted for all replications, `chron.smtp.send-on-success` can be set to true in order to trigger emails for successful replications as well. If no email is wanted, `stmp.send` can be set to false.

Additional Notes on Configuration

- The replication.cron timer sets how often the replication-shell queries the ingest-server for active replications. It uses a cron style formatting:
    - See http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger for an overview of the formatting used
    - `0 0 * * * *`: The default timer, at the top of every hour
    - `0 */1 * * * *`: A faster timer, once per minute
- The development profile can be used for testing configuration options. It remains in the foreground and has a limited set of commands.
- The mail configuration (smtp) is set to send by default.
    - If you don't want to send mail, or have a server which does not have smtp capabilities you can turn it off by setting smtp.send: false
- If the storage is not set up correctly, the replicationd process will not run. This includes existence, and r/w/e permissions

Errors

This will be filled out as we experience problems. Check `/var/log/chronopolis/replication.log` to see if there are any stack traces.