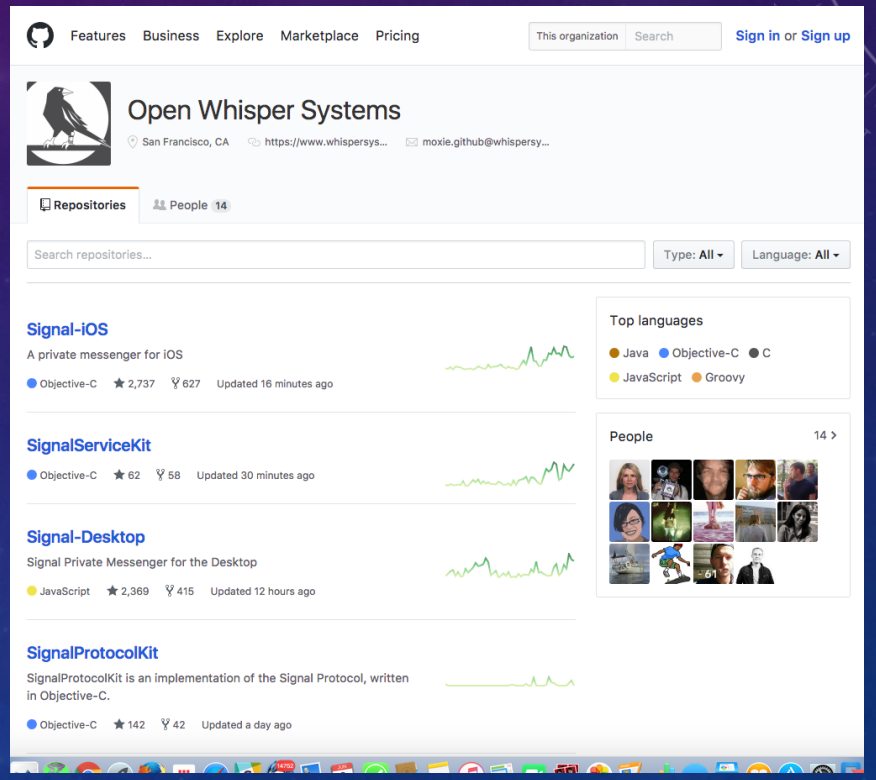
AUTOMATICALLY UPDATE REPOSITORIES FROM GITHUB

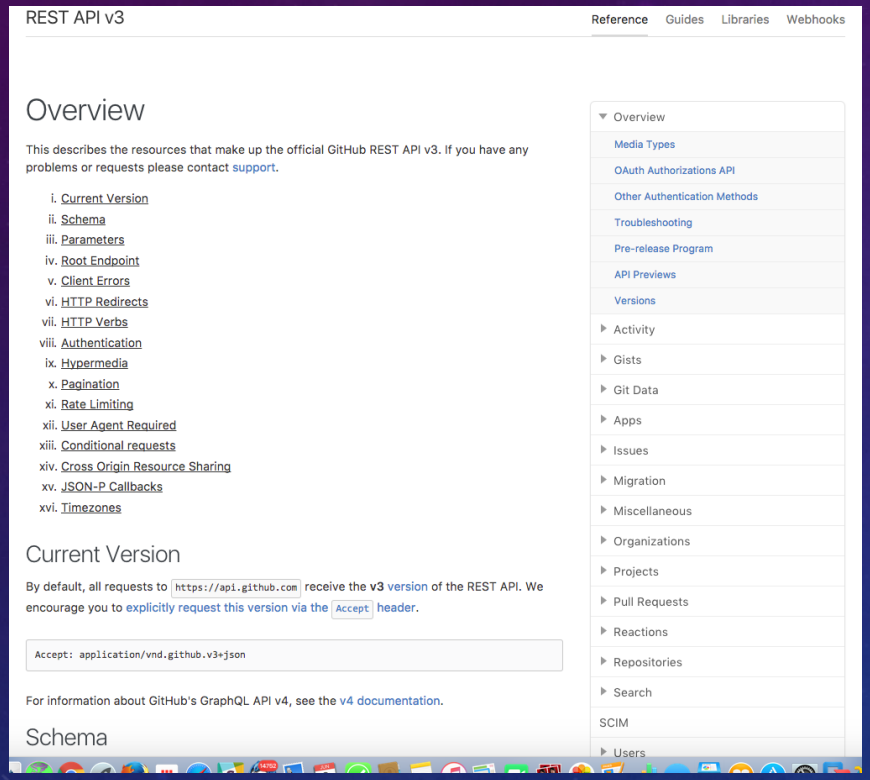
AN ARRAY OF LINUX UTILITIES AND TECHNIQUES

OBJECTIVE

- I wanted to automatically and periodically clone and pull GitHub repositories
- Requires a variety of techniques covered in the following slides



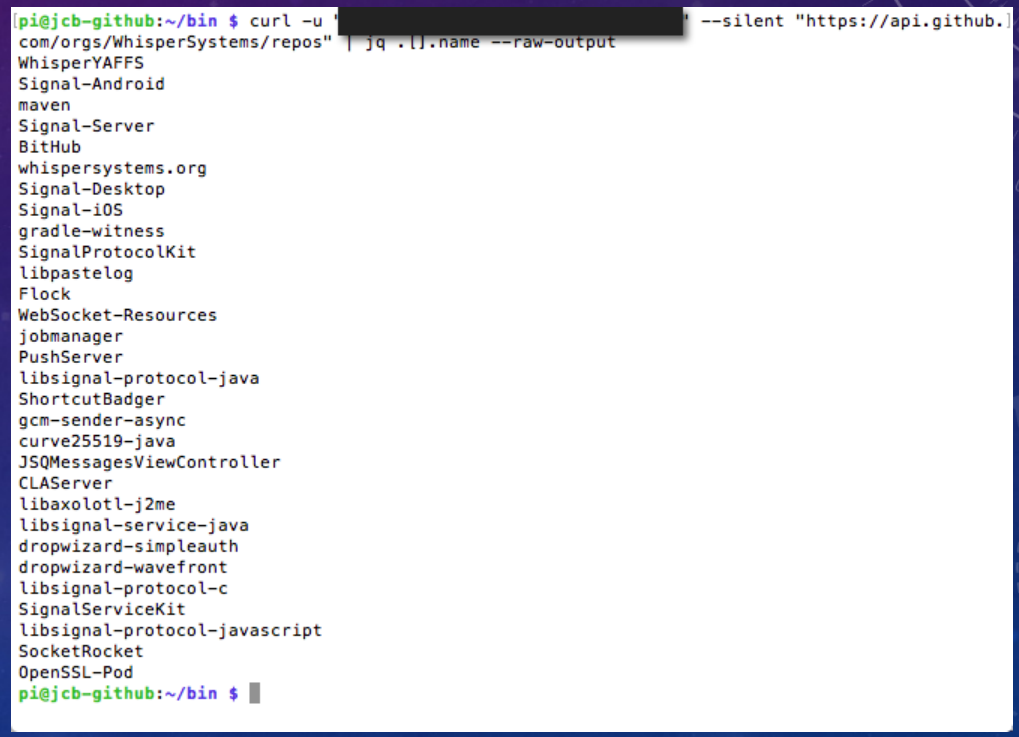
GITHUB API



- GitHub has an API
- Available via REST
  - Check out https://developer.github.com/v3/
  - Returns JSON formatted information
- Some useful verbs
  - curl -u "username:password" --silent "https://api.github.com/orgs/WhisperSystems/repos" # Authenticate and obtain information on the Whisper Systems organization
  - curl -u "username:password" --silent "https://api.github.com/users/Anonabox/repos" # Authenticate and obtain information on the Anonabox user

JQ

- jq is a handy utility to parse JSON formatted information
- Examples
  - curl -u "username:password" --silent "https://api.github.com/orgs/WhisperSystems" | jq .public_repos --raw-output # Obtain the number of repositories
  - curl -u ”username:password" --silent "https://api.github.com/orgs/WhisperSystems/repos" | jq .[].name --raw-output # Obtain the list of repositories



GITHUB PAGINATION

```
   # Determine the number of repos
   REPO_COUNT=`curl -u "username:password" --silent "https://api.github.com/orgs/${ORGANIZATIO
N}" | jq .public_repos --raw-output`
   echo This organization has ${REPO_COUNT} repos

   # Determine the number of pages
   PAGES=$(((${REPO_COUNT}+99)/100))
   echo This organization has ${PAGES} pages of repos

   # For each page of repos...
   for CURRENT_PAGE in `seq 1 ${PAGES}`;
   do
      # For each repository owned by that organization
```

- GitHub will return at most 100 results in a response.
- But, some organizations and users have more than 100 repositories
- So, must paginate the results

GIT

- Use git clone to create a repository that isn’t yet present
- Use git pull to update a repository that has already been cloned

```
   for CURRENT_PAGE in `seq 1 ${PAGES}`;
   do
      # For each repository owned by that organization
      for PROJECT in `curl -u "username:password" --silent "https://api.github.com/orgs/${ORGA
NIZATION}/repos?page=${CURRENT_PAGE}&per_page=100" | jq .[].name --raw-output`
      do
         echo Project: ${PROJECT}

         # if the folder doesn't exist...
         if [ ! -d "${PROJECT}" ]; then
            # Clone the project
            git clone https://github.com/${ORGANIZATION}/${PROJECT}

            FILE_COUNT=`ls ${PROJECT} | wc -l`
            echo Repository has ${FILE_COUNT} files
            if [ ${FILE_COUNT} -eq 0 ]; then
               echo Repository is empty... deleting it
               rm -rf ${PROJECT}
            fi

         else
            # Change into the project directory
            cd ${PROJECT}
            echo Current Working Directory: `pwd`

            # Update the repository
            git pull

            cd ..

            FILE_COUNT=`ls ${PROJECT} | wc -l`
            echo Repository has ${FILE_COUNT} files
            if [ ${FILE_COUNT} -eq 0 ]; then
               echo Repository is empty... deleting it
               rm -rf ${PROJECT}
            fi
         fi

      done
   done
```

NILFS

```
# Take a NILFS snapshot
sync
sleep 1
sync
sleep 1
chcp ss `lscp -rn 1 | sed -n 2p | sed -e 's/^[ \t]*//' | cut -f 1 -d ' '`
~
```

- After cloning/updating all the repositories of interest, create a snapshot of the file system
- A series of commands
  - chsp ss # change a file system checkpoint into a snapshot
  - lscp -rn 1 # get the most recent checkpoint
  - sed -n 2p # filter to eliminate the header line
  - sed -e 's/^[\t]*//' # convert consecutive spaces to a tab to help with parsing
  - cut -f 1 -d ' '` # get only the checkpoint number

CRON

- To update the repositories each day, rely on cron
- Create a shell script for use by cron
- Create an link in /etc/crontab.daily to this script

```
pi@jcb-github:~/bin $ sudo cat grabgit.cron.sh
#!/bin/bash

cd ~pi/bin
./grabgit.sh >> grabgit.log 2>&1
pi@jcb-github:~/bin $
```

```
pi@jcb-github:/etc/cron.daily $ ls -l
total 44
-rwxr-xr-x 1 root root 15000 Dec 13 16:38 apt
-rwxr-xr-x 1 root root   314 Jun 15  2014 aptitude
-rwxr-xr-x 1 root root   355 Oct 28  2014 bsdmainutils
-rwxr-xr-x 1 root root  1597 Jun  5  2016 dpkg
lrwxrwxrwx 1 root root    28 Jun  3 20:40 grabgit -> /home/pi/bin/grabgit.cron.sh
-rwxr-xr-x 1 root root    89 Jan 28  2014 logrotate
-rwxr-xr-x 1 root root  1293 Jan  5  2015 man-db
-rwxr-xr-x 1 root root  1110 Jul 25  2016 ntp
-rwxr-xr-x 1 root root   249 Feb 24 16:49 passwd
pi@jcb-github:/etc/cron.daily $
```

LOGROTATE

```
pi@jcb-github:/etc/logrotate.d $ cat grabgit
/home/pi/bin/*log {
    rotate 30
    daily
    compress
    delaycompress
    missingok
    notifempty
}
pi@jcb-github:/etc/logrotate.d $
```

- To keep the log file from growing indefinitely, rely on logrotate