Geostreaming Api V3 Deployment

To set up geostreaming as a service in a ubuntu machine use the following service file in /lib/systemd/system/geostreams.service

```
#
# This file is centrally managed and will be replaced every 30 minutes
# To make changes to this file contact Rob Kooper (kooper@illinois.edu)
#
[Unit]
Description=GEO TEMPORAL API V3
After=network.target
[Service]
User=clowder
Group=users
Restart=on-failure
WorkingDirectory=/home/clowder
ExecStartPre=/bin/bash -c "if [ -d /home/clowder/geo-temporal-api-v3 ]; then if [ -f /home/clowder/geo-temporal-
api-v3/RUNNING_PID ]; then if ps -p `cat /home/clowder/geo-temporal-api-v3/RUNNING_PID` > /dev/null ; then echo
\"Found running version, killing old version\"; kill `cat /home/clowder/geo-temporal-api-v3/RUNNING_PID`; fi;
rm /home/clowder/geo-temporal-api-v3/RUNNING_PID; fi; fi"
ExecStart=/home/clowder/geo-temporal-api-v3/bin/geostreams -Dhttp.port=9002 -Dplay.http.secret.key="
SOMESECRETKEY" -Dapplication.context=/geostreams/ -mem 2048 -Dconfig.file=/home/clowder/geo-temporal-api-v3
/custom/application.conf

[Install]
WantedBy=multi-user.target
~
```

Use the following script to update the geostreaming api to the latest **develop** branch. Update the name of the branch if you want to get the latest production one. I am not sure on how to get this file to puppet

```
#!/bin/bash
GS3_BRANCH=${GS3_BRANCH:-"GEOD-GS0"}
GS3_BUILD=${GS3_BUILD:-"latestSuccessful"}
# Slack token for notifications
SLACK_TOKEN=**Get from other script**
SLACK_CHANNEL="isda-software"
# change to folder where script is installed
cd /home/clowder

# fetch software
if [[ ${GS3_BUILD} == latest* ]]; then

   BB="${GS3_BRANCH}/${GS3_BUILD}"

else
   BB="${GS3_BRANCH}-${GS3_BUILD}"

fi
URL="https://opensource.ncsa.illinois.edu/bamboo/browse/${BB}/artifact/shared/dist/"
/usr/bin/wget -q -e robots=off -A "geostreams-*.zip" -nd -r -N -l1 ${URL}

LATEST=$( /bin/ls -1rt geostreams-*.zip | tail -1 )

if [ -s ${LATEST} ]; then

  if [ "$1" == "--force" -o ${LATEST} -nt geostreams ]; then

    exec 3>&1

    exec &> "/tmp/$$.txt"
```

```
 echo "UPDATING GEOSTREAMING API v3 on ${HOSTNAME}"
 echo " bamboo branch = ${GS3_BRANCH}"
 echo " bamboo build = ${GS3_BUILD}"
 # stop geostreams
 /usr/sbin/service geostreams stop
# Save local modifications
if [ -d geo-temporal-api-v3/custom ]; then
     mv geo-temporal-api-v3/custom geo-temporal-api-v3.custom
fi
if [ -d geo-temporal-api-v3/logs ]; then
     mv geo-temporal-api-v3/logs geo-temporal-api-v3.logs
fi
# install new version
/bin/rm -rf geo-temporal-api-v3 $( basename ${LATEST} .zip)
/usr/bin/unzip -q ${LATEST}
/bin/mv -v $( basename ${LATEST} .zip) geo-temporal-api-v3
/usr/bin/touch geo-temporal-api-v3

# restore local modifications
if [ -d geo-temporal-api-v3.custom ]; then
     mv geo-temporal-api-v3.custom geo-temporal-api-v3/custom
fi
if [ -d geo-temporal-api-v3.logs ]; then
     mv geo-temporal-api-v3.logs geo-temporal-api-v3/logs
fi
if [ -f geo-temporal-api-v3/custom/messages.en ] &&[ -d geo-temporal-api-v3/conf ]; then
     /bin/cp -f geo-temporal-api-v3/custom/messages.en geo-temporal-api-v3/conf
fi
# change permissions
/bin/chown -R clowder geo-temporal-api-v3
# start geostreams again
/usr/sbin/service geostreams start
# Send message to slack
if [ "${SLACK_TOKEN}" != "" -a "${SLACK_CHANNEL}" != "" ]; then
   url="https://hooks.slack.com/services/${SLACK_TOKEN}"
   txt=$(cat /tmp/$$.txt | sed 's/"/\\"/g;s/$/\\/g' | tr '\n' 'n' )
```

```
      payload="payload={\"channel\": \"${SLACK_CHANNEL}\", \"username\": \"geostreams\", \"text\": \"${txt}\",
\"icon_url\": \"https://opensource.ncsa.illinois.edu/projects/artifacts/GEOD/logo.png\"}"

      result=$(curl -s -X POST --data-urlencode "${payload}" $url)

   fi

   if [ "${STDOUT}" != "" ]; then

      cat /tmp/$$.txt >&3

   fi

    rm /tmp/$$.txt

  fi

fi
```

Note: You need to have inside /home/clowder/geo-temporal-api-v3 a custom folder with the application.conf file and the messages.en

There is a sample application.conf in seagrant-dev machine under Yan's username.

If this is the first time deploying, remember the setting up steps for the region trend tables (if needed) and the parameters (needed for all). More details in Geostreams-api-v3 - Local

Add to your nginx a route for /geostreams

```
  rewrite ^/geostreams$ /geostreams/ permanent;

  location /geostreams {

    proxy_pass http://localhost:9002;

  }
```