## Grinder Instructions

I create the Grinder stack using my bash script called grinderCF.sh (based off Ray’s cloudform.sh)

To run tests, Grinder needs a test script and a properties file, and for distributed testing, these need to be in the same directory.

So on my computer, I made a folder called grinderscripts which contains grinderCF.sh, httpget.py, and grinder-tile-server.properties

* httpget.py is the test script. It does the HTTP Request to the tileserver (hitting the output URL from the tile server stack that is being load tested) and tells Grinder to record the results
* grinder-tile-server.properties is the file that specifies properties that Grinder needs to run (the test script to run, # of processes, # of threads, # of runs, etc.)

The grinder-load-testing.template is used to make the console and agents for Grinder (as a CF stack). I have this template in the folder above the grinderscripts folder on my computer.

The template takes in a bunch of parameters that’s used for testing (some of which I set with grinderCF.sh). Template’s parameters:

* DesiredIncreasePercentage: % by which the number of agents increase when scaling up
* DesiredIntervalBetweenIncreases: how long to wait between each scale-up of the agents
* DesiredLoad – desired Tests Per Second to get to for load testing
* KeyPairName – Optional, keypair for instances
* InstanceTypeAgent/InstanceTypeConsole – the instance types of agents and console
* MaxAgents – max # of agents to launch for distributed testing
* TestProperties – name of properties file (in TestDirectory)
* TestDirectory – S3 location of directory which contains test script and properties

In S3, in my leidoslabs/cf-templates/image-tile-service-yims2 bucket, I made a grinderscripts folder.

In grinderCF.sh:

* The script’s current directory (grinderscripts folder) on my computer syncs with grinderscripts in S3 to save the test script and properties file
* The grinder-load-testing.template is copied over to S3
* The parameters I specify for the CF template:
  + DesiredLoad – from input
  + KeyPairName – advanced-analytics-05-18-2017
  + MaxAgents – from input
  + TestProperties – grinder-tile-server.properties
  + TestDirectory - s3://leidoslabs/cf-templates/image-tile-service-yims2/grinderscripts

So to actually create the stack, the command to run the bash script is:

./grinderCF.sh 1500 10

* 1500 = desired load (tests/second)
* 10 = max Grinder agents to create