**Qlik Sense Scalability Tools Extension**

**** Qlik Sense Scalability Tools knowledge required ****

**Introduction**

This extends Qlik Sense Scalability Tools and helps to prepare Qlik Sense to perform scalability tests in multi applications scenarios, for example SaaS. This allows you to generate number of QS streams and apps. For example, you can generate 100 QS streams and apps and then run scalability tasks with 1 or more users per QS app.
Before you start, read Scalability Tools documentation and make sure you understand how Scalability Tools work. SaaS scenario requires some specific settings of Scalability Tools described below.

**Prerequisites**

Tested on: Qlik Sense February 2018, Scalability Tools 4.4.0, Node.js v8.11.2 with npm modules: request, fs

**Installation**

Use npm to install as module. This will download app_gen.js script. Alternatively download project from Qlik branch or directly from github.

**Using**

There are two main steps before you start scalability tests execution:

- Qlik Sense environment preparation
- Scalability Tools configuration

## Qlik Sense environment preparation

**Steams and apps**

Develop Qlik Sense application template you want to test, deploy on Qlik Sense server and publish to Everyone stream.

Then use app_gen.js scrip to:

- create number of Qlik Sense streams, script will name them as CUST_1, CUST_2…. CUST_n
- create number of your application copies published to above streams, script changes apps names to ST. Optionally you can reload applications.

Script creates tag ‘Scaltest’ and tags all streams and apps generated.

Type 'node app_gen.js -q' to get help and understand how to run the script. See sample .bat file attached to this package.

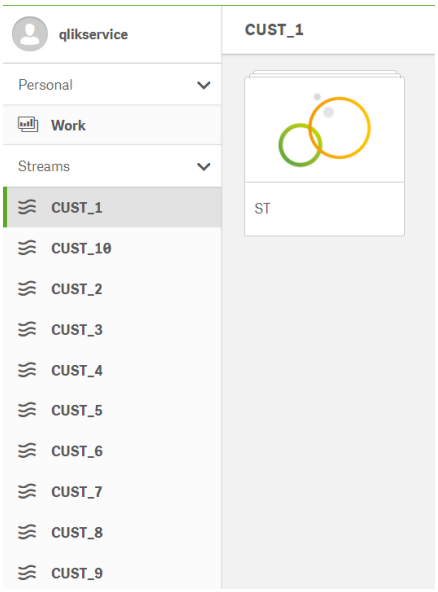
Because script makes app copies and optionally reloads them, in case of large number of apps to create (ex 1000) this takes time and resources. Script uses REST QRS API and simply starts all app copy tasks with 1 … n seconds timeout, where n is number of apps to be created. Depends on hardware performance it is advised to generate 100 – 200 apps per one script execution. Use offset parameter to start streams numbering from any other number for example 101, if you have already generated 100 streams.

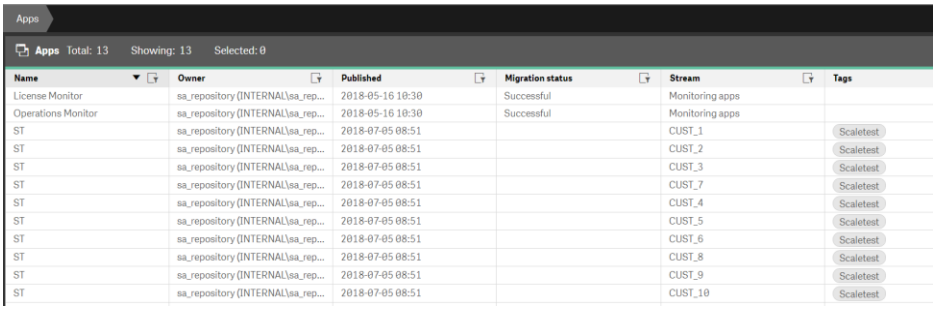
In case of application reloads you can adjust Qlik Sense reload performance by two Scheduler settings in QMC:

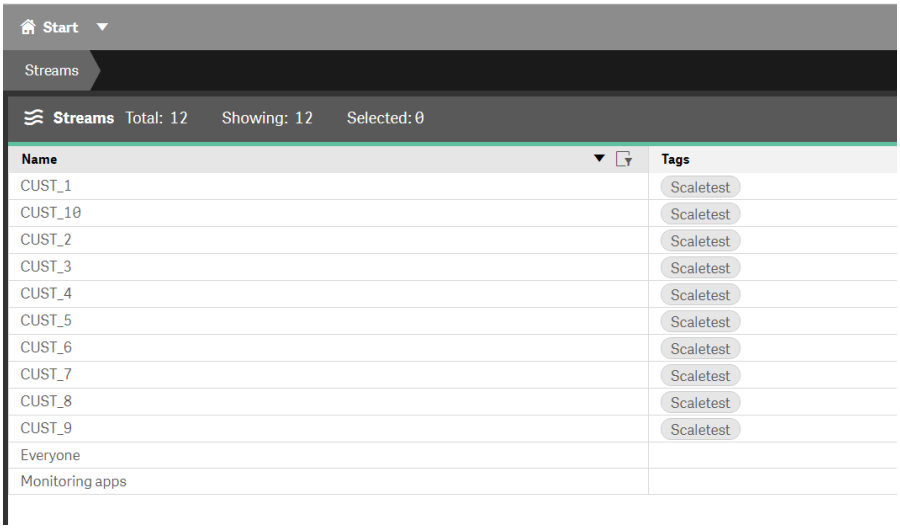
- "Max concurrent reloads" - can be up to n-1 where n is number of CPU cores
- "Engine Timeout" – how long reload task can wait in Queued state

After app_gen.js executions you will have Qlik Sense environment prepared, see below sample screenshots.

When scalability tests are finished you can delete generated Qlik Sense streams and apps by executing 'app_gen.js -d'. This will delete all streams and apps with tag 'Scaltest'






**Authentication**

Configure Qlik Sense header authentication as described in Scalability Tools documentation, but apply following in regards of authentication setup:

- Create Virtual Proxy with Header authentication and dynamic user directory
- Use $ud@$id for dynamic user directory ($ud\\$id is not supported by Scalability Tools 4.4.0)

**IDENTIFICATION**

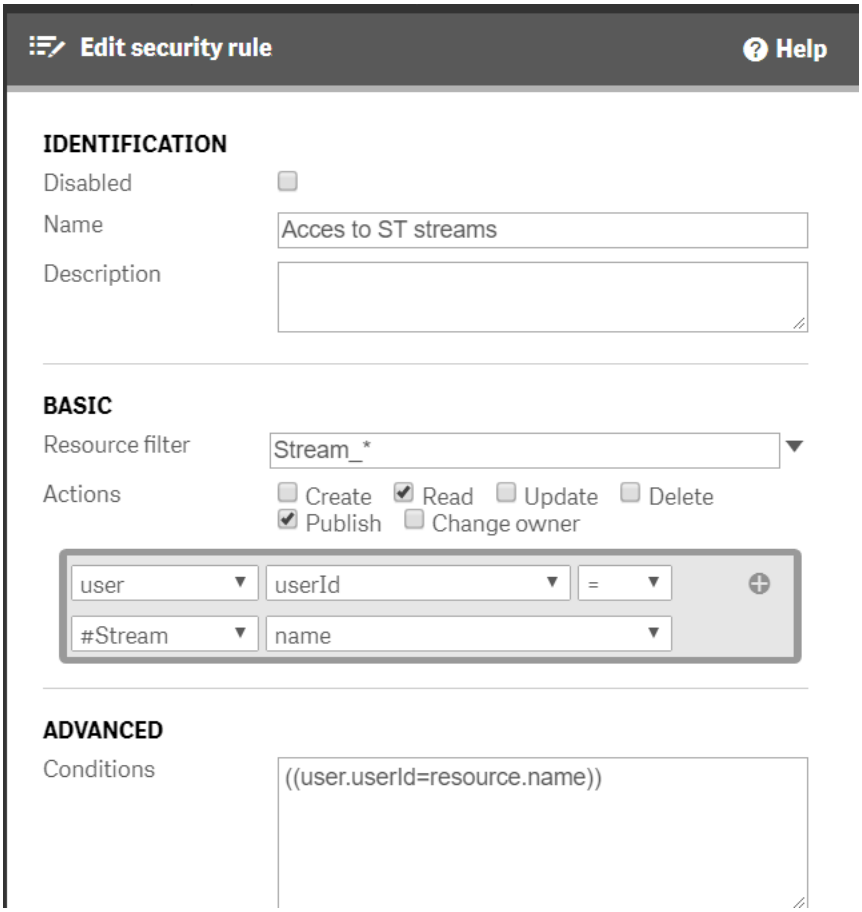
| | |
|---|---|
| Description | Header Auth dyn |
| Prefix | hdd |
| | The prefix must be unique for all virtual proxies used by the same proxy service, as this differentiates the virtual proxies and will be a part of the URL (https://[node]/[prefix]/). Valid characters for prefix are "a-z", "0-9", "-", ".", "_", "~". Slash "/" is also valid, but cannot begin or end the prefix. |
| Session inactivity timeout (minutes) | 30 |
| Session cookie header name | X-Qlik-Session-hdd |
| | The session cookie header name must be unique for all virtual proxies used by the same proxy service. |

**AUTHENTICATION**

| | |
|---|---|
| Anonymous access mode | No anonymous user |
| Authentication method | Header authentication dynamic user directory |
| Header authentication header name | Qlik-hdd |
| Header authentication dynamic user directory | $ud@$id |
| | Use $ud[sep]$id or $id[sep]$ud where [sep] is one or more characters. Backslashes (\) must be escaped with an additional \. Example 1: $ud\\$id; example 2: $id@$ud; example 3: $ud:::$id. |

## Secuirty rules

Create security rule giving access to stream if userid equals stream name:




**Licensing**

Do not forget to configure user access allocation rule in licensing in case you are using token or user based license. User id and user directory will be generated by Scalability Tools and you will configure the pattern, so you can create token allocation rule based on this.

**Qlik Scalability Tools configuration**

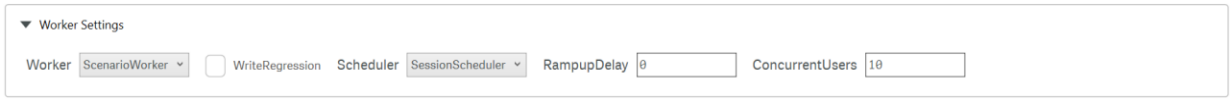
Download https://community.qlik.com/docs/DOC-8878, install Qlik Scalability Tools and create normal scenario script with settings described below.

Start scalability tools and create script in Scenario Editor with following settings:

- Connection settings:
    - Header authentication in Qlik Sense (use virtual proxy configured in QS)
    - In App filed put app name (not GUID), default app name generated by script is ST. There will be a lot of ST apps available on server, but because of security rules each user will see only one.


- Test settings:
    - Configure UserNamePrefix with pattern that match dynamic user directory in virtual proxy. Scalability Tool will be generating users with specified directory and userid_1, userid_2, userid_n where n is number of concurrent users specified in worker setting below. In below case user directory will be HDD0 and user name CUST_1, CUST_2 etc.


- Worker settings
    - Specify number of concurrent users




**How does it work?**

Scalability Tools creates concurrent users executing configured scenario. Each user has asccess to stream with the same name as user name. There is only one application in all streams. So each simulated user works with different application and there is one user per application.

What if you need 2 or more users per one application?

Create two or more scenario scripts with the same settings as describe above, just change user directory in UserNamePrefix in Test Settings section, for example HDD1@CUST. Then execute both script in parallel. In such case Scalability Tools creates users with same names but different directories (for example HDD0\CUST_1, HDD0\CUST_2, HHD1\CUST_1, HDD1\CUST_2 etc. From Qlik server perspective they are distinct users, so this simulates more than 1 user per application.