Load Testing Jitsi Meet

Link: https://meetrix.io/blog/webrtc/jitsi/jitsi-meet-load-testing.html

Make sure your Jitsi Meet infrastructure is ready for production

By deploying a horizontally scalable Jitsi Meet setup, you can scale your Jitsi Meet conferencing infrastructure to cater thousands of concurrent users. But, before you go live you might want to make sure that your infrastructure is capable of handling the desired number of concurrent users.

Jitsi Meet Torture is a tool that can be used to run tests against a Jitsi Instance and it is capable of load testing Jitsi Meet.

> Jitsi Meet Load Testing
> Jitsi Meet Load Testing with Torture

Selenium Grid

To simulate hundrads of concurrent users, we need to deploy Jitsi Meet Torture agains a selenium grid setup. There would be a `selenium hub` which accepts commands from Jitsi Meet Torture and pass them to `selenium nodes`. You can have hundrads of `selenium nodes` to simulate users.

> Jitsi Meet Load Testing
> Jitsi Meet Load Testing with Torture

Minimal setup with docker-compose

1. Install Docker. You can use following scripts on Ubuntu 18.04

```
sudo apt-get update


sudo apt-get -y install apt-transport-https ca-certificates curl software-properties-
common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository -y \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"

sudo apt-get -y update
sudo apt-get -y install docker-ce

sudo usermod -a -G docker $(whoami)
```

2. Install the latest version of Docker Compose
3. Create a `docker-compose.yml` file with following content.

```
version: "3.3"
services:
  torture:
    image: meetrix/jitsi-meet-torture
  hub:
    image: selenium/hub:3.141.59
  node:
    build: ./node
    image: meetrix/jitsi-meet-torture-selenium-node
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
      - hub
    environment:
      HUB_HOST: hub
```

4. Please note that `node` are configured to run only one chrome instance at a time. Run the setup with `docker-compose up -d --scale node=2` to run two nodes.
5. `docker-compose exec torture /bin/bash`
6. Change `<YOUR_JITSI_MEET_INSTANCE_URL>` to your Jitsi Meet installation (for example `https://meet.example.com`) in following script and run to perform the test.

```
./scripts/malleus.sh --conferences=1 --participants=2 --senders=2 --audio-senders=2 --
duration=300 --room-name-prefix=test --hub-url=http://hub:4444/wd/hub --instance-
```

```
url=<YOUR_JITSI_MEET_INSTANCE_URL>
```

Running the test with large number of nodes

You can scale the number of nodes with `docker-compose up --scale node=<NUMBER>` or you can create a VM using same docker images for `selenium hub` and `selenium-node`. Then you can use an autoscaling mechanism (Such as Autoscaling groups on AWS) to scale the number of nodes.

Looking for commercial support for Jitsi Meet ? Please contact us via hello@meetrix.io

Updated: June 19, 2020

Revision #1
Created 6 July 2024 11:09:28 by Administrador
Updated 6 July 2024 11:11:38 by Administrador