Contents

Chapter 1

Recommended Deployment Configurations: Client, Server Software, Server Hardware

Graphistry ships as a Docker container that runs in a variety of Linux + Nvidia GPU environments:

Contents

- Overview
- Client
- Server Software: Cloud, OS, Docker, Avoiding Root Users

Overview

- **Client**: Chrome/Firefox from the last 3 years, WebGL enabled, and 100KB/s download ability
- **Server**:
- x86 Linux server with 4+ CPU cores, 16+ GB CPU RAM (3GB per concurrent user), and 1+ Nvidia GPUs (K80 onwards) with 4+ GB RAM each (1+ GB per concurrent user)
- Recommend Ubuntu 16.04, 4+ CPU cores, 64GB+ CPU RAM, Nvidia Tesla or later

- Docker / CUDA 9.2 / nvidia-docker-2

Client

A user's environment should support Graphistry if it supports Youtube, and even better, Netflix.

The Graphistry client runs in standard browser configurations:

- **Browser**: Chrome and Firefox from the last 3 years, and users regularly report success with other browsers like Safari.

- **WebGL**: WebGL 1.0 is required. It is 7+ years old, so most client devices, including phones and tablets, support it. Graphistry runs fine on both integrated and discrete graphic cards, with especially large graphs working better on better GPUs.

- **Network**: 100KB+/s download speeds, and we recommend 1MB+/s if often using graphs with 100K+ nodes and edges.

- **Operating System**: All.

Recommended: Chrome from last 2 years on a device from the last 4 years and a 1MB+/s network connection

Server Software: Cloud, OS, Docker, Avoiding Root Users

Graphistry can run both on-premises and in the cloud on Amazon EC2, Google GCP, and Microsoft Azure.

Cloud

Tested AWS Instances:

- P2
- G3 *Recommended for testing and initial workloads*
- P3

Tested Azure Instances:

- NV6 *Recommended for testing and initial workloads*
- NC6

See the hardware provisioning section to pick the right configuration for you.

OS & Docker

Graphistry regularly runs on:

- Ubuntu Xenial 16.04 LTS *Recommended*
- RedHat RHEL 7.3

Both support nvidia-docker-2:

- Docker
- nvidia-docker-2
- CUDA 9.2

For cloud users, we maintain bootstrap scripts, and they are a useful reference for on-premises users.

User: Root vs. Not, Permissions

Installing Docker, Nvidia drivers, and nvidia-docker currently all require root user permissions.

Graphistry can be installed and run as an unprivileged user as long as it have access to nvidia-docker.

Storage

We recommend using backed-up network attached storage for persisting visualizations and investigations. Data volumes are negligible in practice, e.g., < $10/mo on AWS S3.

Server: Hardware Capacity Planning

Graphistry utilization increases with the number of concurrent visualizations and the sizes of their datasets. Most teams will only have a few concurrent users and a few concurrent sessions per user. So, one primary server, and one spillover or dev server, gets a team far.

For teams doing single-purpose multi-year purchases, we generally recommend more GPUs and more memory: As Graphistry adds further scaling features, users will be able to upload more data and burst to more devices.

Network

A Graphistry server must support 1MB+/s per expected concurrent user. A moderately used team server may stream a few hundred GB / month.

- The server should allow http/https access by users and ssh by the administrator.
- TLS certificates can be installed (nginx)
- The Graphistry server may invoke various database connectors: Those systems should enable firewall and credential access that authorizes authenticated remote invocations from the Graphistry server.

GPUs & GPU RAM

The following Nvidia GPUs are known to work with Graphistry:

- Tesla: K40, K80, M40
- DGX: P100, V100 ***Recommended***

The GPU should provide 1+ GB of memory per concurrent user.

CPU Cores & CPU RAM

CPU cores & CPU RAM should be provisioned in proportion to the number of GPUs and users:

- CPU Cores: We recommend 4-6 x86 CPU cores per GPU
- CPU RAM: We recommend 6 GB base memory and at least 16 GB total memory for a single GPU system. For balanced scaling, 3 GB per concurrent user or 3X the GPU RAM.

Multi-GPU, Multi-Node, and Multi-Tenancy

Graphistry 1.0 virtualizes a single GPU for shared use by multiple users.

- When Graphistry is on a shared system, it is especially crucial to determine whether the system environment is ready for nvidia-docker-2, or needs potentially disruptive patching updates. Likewise, the CPU, GPU, and network resources assigned to the Graphistry instance (such as via Docker) should not be contended with from sibling applications. Such software is often not as isolatable.

- Multitenancy via multiple GPUs: You can use more GPUs to handle more users and give more performance isolation between users. We recommend separating a few heavy users from many light users, and developers from non-developers.

- Acceleration via multiple GPUs: Graphistry is investigating how to achieve higher speeds via multi-GPU acceleration, but the current benefits are only for multitenancy.