Sitespeed User Guide

October 2023

Access and Administration

- Management of all tests is performed from the Jump server
- There are no passwords used for any communication for any server
- For existing Sitespeed installations, contact the Sitespeed administrator to have an account created
- Obtain an SSH private key for the Jump server from the Sitespeed administrator
- The SSH key will allow access to the Jump server only
- A second SSH key pair was installed during the initial system installation, which is used for communication between the Jump server and the other servers

Jump Server Access for New Installations

- During initial installation of the Jump server, an SSH public key is installed on the Jump server
- The SSH private key for the Jump server can be downloaded from https://as.akamai.com/user/sitespeed/jump
- Copy the displayed contents to a file called `jump` and change the file permission to 600 (i.e., `chmod 600 jump`)
- The SSH passphrase for the SSH private key is `!!sitespeed!!`

- To create a new SSH key pair use the following steps:
  - Use `ssh-keygen -t rsa -b 2048 -C "jump" -f jump` to generate a new key pair
  - The name ***jump*** can be changed to anything
  - Copy the public key to `~/.shh/authorized_keys` on the Jump server

Test Workflow

- The following are the high-level tasks required to start collecting data:
  - Create a seed file
  - Deploy the seed file
  - Create a test schedule
  - Deploy the test schedule

- When a test is no longer required, it is recommended you remove the test schedule entry and associated seed file and then either stop all testing or re-deploy the revised schedule if there are other tests running

- All test and server management is performed using a single script called `admin.sh`

- All commands described in this document are performed on the Jump server with one exception. The `graphite.sh` is used on the Graphite server.

Admin Script

- All tasks are performed using the `admin.sh` script
- The script has been designed to avoid and prevent as many errors as possible
- When the script encounters an error, it will print a descriptive reason why the task failed before exiting
- The script contains a help menu that can be displayed using the following methods:

  `./admin.sh`

  `./admin.sh -help` ← double dash before the word help

  `./admin.sh -h`

  `./admin.sh /?`

- The following slide contains a complete summary of all commands and arguments
- This guide will cover will explain what each command does and why and when it should be used

Admin Script

- User access to admin functions is restricted based on account permission
- User accounts are created with either Admin level or User level permission
- All users are permitted to create seed files and schedule tests

```
USAGE admin command [arg1 arg2]

DESCRIPTION Automates the distribution files and execution of scripts across servers

The following commands are available:

        all         Copies customized files across servers

        cert        Checks the certificate renewal date on servers

        core        Checks for core files on servers. Requires:
                    arg1 = check|delete

        cron        Schedules cron jobs on servers. Requires:
                    arg1 = check|update|delete

        docker      Performs various docker functions on servers. Requires:
                    arg1 = check|clean

        grafana     Updates Grafana to the latest version. Requires:
                    arg1 = update|provision

        graphite    Manages the size of graphite.db. Requires:
                    arg1 = check|reduce

        logs        Checks for errors on servers. Requires:
                    arg1 = check|delete

        reset       Deletes Sitespeed data on servers

        seed        Manages the seed files on servers. Requires:
                    arg1 = tld|comp|delete
                    arg2 = seed file

        server      Adds and removes servers. Requires:
                    arg1 = add|delete|names

        storage     Checks the amount of storage used on servers

        update      Updates packages on all servers

        user        Manages user accounts on servers. Requires:
                    arg1 = add|delete|names
```

Admin Permission Matrix

	Admin	User
all	✓	✗
cert	✓	✗
core	✓	✗
cron	✓	✓
docker	✓	✗
grafana	✓	✗
graphite	✓	✗
logs	✓	✗
reset	✓	✗
seed	✓	✓
server add	✓	✗
server delete	✓	✗
server names	✓	✓
storage	✓	✗
update	✓	✗
user	✓	✗

Seed File Information

- A seed file serves two purposes – it represents the name of the test, and it contains the URLs that are associated with a test
- All tests require their own seed file, which contain one or more URL entries, including query parameters
  - All URLs should be tested using a browser to ensure they are valid
- A seed file entry consists of three fields, separated by spaces:
  - URL (including the scheme)
  - Page name
  - Domain name
- The Page and Domain names are used in Grafana dashboards, so it is recommended to use friendly, meaningful names
- A seed file name must include a "txt" extension
- Seed file names are case-insensitive and can contain special characters, such as "–" and "_"
  - The name of the seed file is used in Grafana dashboards, so it is recommended to use friendly, meaningful names

Seed File Deployment

- All servers use the same seed file
- Only one seed file can be deployed at a time
- Use `admin.sh` to deploy a seed file across servers using the following three arguments:
  - seed
  - tld | comp | delete
  - name
- tld (top-level domain) and comp (competitive analysis) tells the system where to store the results in the Graphite database
- delete removes a previously deployed seed file
- name is the seed file that resides in the seeds folder
  - Do not include the “txt” extension when running the admin script
- The admin script checks to ensure the correct number of arguments are used
  - If the script does not find a seed file with a “txt” extension the script will print an error message before exiting

Scheduling Tests

- All tests are controlled using a cron job on each server
- The cron folder on the Jump server contains two files:
  - psicron
  - sitecron
- **psicron** controls the collection of Google Chrome User Experience and Lighthouse data
  - Data is collected once per day since Google only updates their database daily
- The following is an example of a psicron entry

```
5 1 * * * /usr/local/sitespeed/google.sh tld ATT &>> /usr/local/sitespeed/logs/YYY.ZZZ.msg.log
```

- **tld** can either be tld (top-level domain) or **comp** (competitive analysis)
- **ATT** (example) is the name of the seed file that was previously created and deployed
- No other portions of the file should be modified

Scheduling Tests

- **sitecron** controls the synthetic testing schedule
- The following is an example of an entry

```
5 * * * * /usr/local/sitespeed/sitespeed.sh tld ATT XXX 3 &>> /usr/local/sitespeed/logs/YYY.ZZZ.msg.log
```

- tld can either be tld (top-level domain) or comp (competitive analysis)
- ATT (example) is the name of the seed file that was previously created and deployed
- 3 represents the number of test iterations
  - If the number if missing, it will default to 3
  - The number must be odd
- No other portions of the file should be modified
- The first five fields of a cron entry represent Minute Hour Day of Month Month Day of Week
  - Multiple test entries should be spaced out using the first two fields. The duration of a test will determine the best amount of time to wait before scheduling the next test
  - Normal testing is done once per hour

Scheduling Tests

- The admin cron command requires a single argument:
  - check Displays the jobs currently running on all servers
  - update Deploys all test entries contained in psicron and sitecron
  - delete Deletes all tests on all servers
- The following is an example deploying a new test

```
./admin.sh cron update
```

- When testing is complete, remove your entries from the psicron and sitecron files and run either one of the following commands:

```
./admin.sh cron update
```

```
./admin.sh cron delete
```

- Since the system is designed for multiple users, it is a good idea to run `./admin.sh cron check` before deleting all tests as a courtesy to other users
- Test duration is subjective and depends on what is trying to be accomplished

Admin Commands

`./admin.sh all`

The `all` command distributes customized scripts and configuration files across servers. This is helpful if a lot of changes have been made and you want to ensure that all servers have a consistent configuration.

`./admin.sh cert`

During installation, the certbot capability was installed, enabling the installation of an SSL certificate using Let's Encrypt. If an SSL certificate was installed after the initial installation, the `cert` command displays the current status of the SSL certificate.

`./admin.sh core check|delete`

The `core` command checks for the existence of core dump files on each server and provides the ability to delete the core dump files. The `check` argument counts the number of core dump files on each server. The `delete` argument deletes the core dump files on all servers. Core dump files are an indication that the testing process is having trouble and should be investigated. A common cause of core dumps is when a URL is being blocked by a WAF rule. Although the test will proceed, the internal node.js application will generate a core dump file. The solution is to examine the logs files, identify the offending URL, remove the URL from the seed file, and then re-deploy the revised seed file.

Admin Commands

`./admin.sh cron check|update|delete`

The `cron` command manages all aspects of scheduling across servers. The `check` argument displays the current cron schedules on each server. The `update` argument deploys the jobs that have been defined in psicron and sitecron to all servers. psicron gets deployed to Google and sitecron gets deployed to all the Sitespeed machines. The `delete` argument removes the test schedule on all servers. The `delete` argument does not cancel any tests that are currently in progress.

`./admin.sh docker check|clean`

The `docker` command checks the installed Docker images and running containers on each server. The `check` argument displays the current Docker images on the server and the current status of any running containers. This is useful to identify any long running, run-a-way Docker containers. If you suspect that containers are not getting shutdown "gracefully" use the `clean` argument to forcibly delete all Docker images and containers. A new Docker image will automatically be installed during the next test cycle.

`./admin.sh grafana update|provision`

The `grafana` command remotely updates Grafana from the Jump server. The `update` argument updates Grafana to the latest Grafana Enterprise version. The `provision` argument serves two purposes. The first is to re-install the default dashboards that were installed during the initial installation. The second is to install either revised or new dashboards that have been created and published.

Admin Commands

`./admin.sh graphite check|reduce`

The `graphite` command checks the current storage usage of the Graphite annotations database and provides the ability to reduce its size if necessary. The `check` argument displays the current size of the Graphite annotations database. Although this process runs on a nightly basis, `check` should be used if it is suspected that the nightly maintenance job is not running. The nightly maintenance job runs on the Jump server and is called `maintenance.sh`. All running jobs can be checked using `./admin.sh cron check`. The `reduce` argument will delete all annotations from the database older than seven days.

`./admin.sh logs check|delete`

The `logs` command displays the number of errors on all the servers and provides the ability to delete the log files on all servers. The `check` argument displays the number of errors that have been logged for tld and comp test on each server. This information is also displayed with the Grafana Sitespeed Monitor dashboard. The `delete` argument deletes log files on all servers.

`./admin.sh reset`

The `reset` command deletes all logs, test result images and videos, tld and comp seed files, and the Sitespeed HTML-based results on each server. This function is a good way to clean each server to get it back to its initial state. This function does not delete any data from the Graphite database.

Admin Commands

`./admin.sh seed tld|comp|delete name`

The `seed` command manages the distribution and deletion of seed files on all servers. The `tld` and `comp` arguments specify where the seed files gets stored on the Sitespeed machines. The `tld` argument refers to a top-level domain test and the `comp` argument refers to a competitive analysis test. The `delete` argument deletes the named seed file on all servers. The `name` argument is the name of the seed file that either gets deployed or deleted on all servers. All seed files reside in the seeds folder on the Jump server. When a seed file is created it must use a "txt" extension. However, when the name of the seed file is used with the `seed` command, the "txt" extension should not be included; the admin script will check for the correct format.

`./admin.sh server add|delete|names`

The `server` command manages the adding and deleting of servers from the testing. The `add` argument adds a server and the `delete` argument removes a server. A prerequisite for using the `add` argument is that the server must be online, and its name must be resolvable. The `delete` argument only deletes a server if it already exists; `delete` does not delete the actual server. The `name` argument displays the name of the current servers. If you delete a server, be sure to run `./admin.sh cron delete` first since there may be jobs already running on the server that is to be deleted.

Admin Commands

`./admin.sh storage`

The `storage` command displays how much disk space is used by the tld and comp tests on each server, and the associated images stored on each server. This information is also displayed with the Grafana Sitespeed Monitor dashboard.

`./admin.sh update`

The `update` command updates all the installed O/S packages on all servers.

`./admin.sh user add|delete|names`

The `user` command manages the adding and deleting of users that can create and schedule tests. All new users are assigned the same permissions as the user that installed the system, which is the Administrator. The Administrator is different than the root user of the O/S. The `add` argument adds a new user to all servers only if the user does not exist. The `delete` argument deletes a user from all server only if the user does exist. Some caveats when using the `delete` argument are you cannot delete your own account or the Administrator's account.

Miscellaneous Commands

`jump name`

The `jump` command is a custom Bash function that enables easy access to other servers from the Jump server. The `name` argument is the name of the server that you "jump" over to via SSH. To display the names of all servers run `./admin.sh server names`

`./graphite.sh start|stop|status`

The `graphite.sh` is used to `start` or `stop` the Docker container that runs the Graphite database. The `status` argument displays the status of the Graphite Docker container. This script does not need to be used unless there is an issue with the Docker container or if any of the underlying Graphite configuration files have been modified (i.e., changing the frequency of testing, which is initially set to 60 minutes). This script should be run directly on the Graphite server.

Akamai CDN

- Sitespeed uses TCP ports 22, 2003, and 8888 for intra-system communication, which will cause the system to fail across Akamai
- During installation, the origin domain name should be used
- Akamai could be used to deliver HTML-based results, which are on the Sitespeed portal
- The following files on the Jump server should be changed to deliver HTML-based results across Akamai
  - `/usr/local/sitespeed/admin.sh` → set the CDN variable to the FQDN that traverses Akamai
  - `/usr/local/sitespeed/portal/index.html` → change the domain for Grafana to the FQDN that traverses Akamai
  - `/usr/local/sitespeed/portal/error.html` → change the domain for Sitespeed and Grafana to the FQDN that traverses Akamai
- Once the changes are done distribute the changes using `./admin.sh all`
- Modify the following file on the Graphite/Grafana server
  - `/var/lib/grafana/dashboards/sitespeed/Page Metrics.json` → change the domain to the FQDN that traverses Akamai

Converting to HTTPS

- Sitespeed is installed using HTTP
- During installation certbot was installed on every server that uses a web server
- Use the following steps from to convert Sitespeed to HTTPS
  - Run `certbot –nginx` on each server to install a Let's Encrypt SSL certificate
  - Change HTTP to HTTPS on the following files on the Jump server
    - `/usr/local/sitespeed/admin.sh`
    - `/usr/local/sitespeed/sitespeed.sh`
    - `/usr/local/sitespeed/portal/index.html`
    - `/usr/local/sitespeed/portal/error.html`
  - Run `./admin.sh all` to distribute all changes
  - Change HTTP to HTTPS in `/var/lib/grafana/dashboards/sitespeed/Page Metrics.json,` on the Graphite/Grafana server

Converting to HTTPS

- The following steps should be done on the Graphite/Grafana server
  - `systemctl stop grafana-server`
  - `certbot certonly` → use option #1
  - `chown -R grafana /etc/letsencrypt/live`
  - `chown -R grafana /etc/letsencrypt/archive`
  - `chmod -R 755 /etc/letsencrypt/live`
  - `chmod -R 755 /etc/letsencrypt/archive`
  - `systemctl start grafana-server`
- Make the following changes to `/etc/grafana/grafana.ini` in the Server section
  - `protocol = https`
  - `http_port = 443`
  - `cert_file = abc`
  - `cert_key = xyz`

where abc and xyz are the names of the files that were installed during the `certbot` process