



Advanced Administration

Advanced Administration

Section Goal:

Understand Advanced GitHub Enterprise administration topics.



Topics and Agenda:

- **Publish logs**
- **Search auditing for users actions**
- **Suspend users**
- **User impersonation**



Topics and Agenda:

- **Validate git-lfs**
- **Access audit logging via API**
- **Enable GitHub replication**
- **GitHub backup and DR**

Useful Commands

Connect to **SSH** using port 122:

- **ghe-announce** - sets a banner at the top of every GitHub Enterprise page
- **ghe-maintenance** - put the instance into maintenance mode
- **ghe-user-csv** - dumps a list of all the users in the installation in CSV format
- **ghe-system-info** - responds with JSON of instance health stats
- **ghe-license-info** - responds with JSON of the currently installed license
- **ghe-set-password** - set a new password to log into the Management Console

Not available when LDAP sync is enabled:

- **ghe-user-promote**
- **ghe-user-demote**
- **ghe-user-suspend**
- **ghe-user-unsuspend**
- **ghe↔↔** - view all commands

Audit Logs

GitHub Enterprise keeps logs of audited user, repository, and system events. You can use these logs to debug your installation as well as to comply with internal security mandates and external regulations. A number of user-initiated actions are audited, including:

- **creating or deleting a repository**
- **creating or deleting an organization**
- **adding or removing an email address**
- **adding or removing an SSH key**

All audited system events—including all pushes and pulls—are logged to `/var/log/github/audit.log`.

All audit information is logged to the `audit.log` file in the `github-logs` directory of any [Support Bundle](#). If [log forwarding](#) is enabled, you can stream this data to an external syslog stream consumer such as Splunk or logstash. All entries from this log use the "github_audit" keyword and can be filtered using that.

Suspending Users

You can [suspend](#) users GitHub Enterprise accounts to open up seats in your site license while preserving the issues, comments, repositories, gists, and other data they created. Suspended users cannot sign into your instance, nor can they push or pull code. When you suspend a user, the change takes effect immediately with no notification to the user. **If the user attempts to pull or push to a repository, they'll receive this error:**

```
$ git clone git@[hostname]:john-doe/test-repo.git
Cloning into 'test-repo'...
ERROR: Your account is suspended. Please check with your installation administrator.
fatal: The remote end hung up unexpectedly
```

High Availability Cluster Configuration

In this [configuration](#), a fully redundant secondary GitHub Enterprise instance is kept in sync with the primary instance via replication of all major datastores.

- **Fully redundant GitHub Enterprise instance**
- **Automated setup of one-way, asynchronous replication of all datastores**
- **Active/Passive cluster configuration**
- **DNS failover**

The replication and failover features included in GitHub Enterprise 2.4 should **only** be used for:

- **Software crashes**
- **Primary system hardware failures**
- **Virtualization host system failures**
- **Logically or physically severed network at the primary site**

Backups

Before you begin using GitHub Enterprise in a production capacity, you should set up a backup host, schedule automated backups, and develop a recovery plan as part of an overall [automated backups plan](#).

- Download the latest [github/backup-utils](#) release and extract on the backup host
- Copy the included `backup.config-example` file to `backup.config` and open in an editor
- Set the `GHE_HOSTNAME` value and the `GHE_DATA_DIR` value
- Open [https://\[hostname\]/setup/settings](https://[hostname]/setup/settings) and add the backup host's SSH key
- Run `bin/ghe-host-check` to verify SSH connectivity
- Run `bin/ghe-backup` to perform an initial full backup
- Schedule regular backups on the host using `cron(8)` or similar

Disaster Recovery

In the event of prolonged outage or catastrophic event at the primary site, basic operations of your GitHub Enterprise environment can be **restored** by performing a restore from the backup host.

```
$ ghe-restore 169.154.1.1
```

```
Starting restore of 169.154.1.1 from snapshot 20141111T174152
```

```
Connect 169.154.1.1 OK (v2.0.0)
```

```
Enabling maintenance mode on 169.154.1.1 ...
```

```
Restoring Git repositories ...
```

```
Restoring GitHub Pages ...
```

```
Restoring MySQL database ...
```

```
Restoring Redis database ...
```

```
Restoring SSH authorized keys ...
```

```
Restoring Elasticsearch indices ...
```

```
Restoring SSH host keys ...
```

```
Completed restore of 169.154.1.1 from snapshot 20141111T174152
```

```
Visit https://169.154.1.1/setup/settings to configure the recovered appliance.
```