Migrating from GitHub Enterprise 11.10.x to 2.1.23

In this article

To migrate from GitHub Enterprise 11.10.x to 2.1.23, you'll need to set up a new appliance instance and migrate data from the previous instance.

> **Note**: GitHub Enterprise Server 11.10 is an unsupported release from 2014. For a list of supported releases, see "GitHub Enterprise Server releases."

Migrations from GitHub Enterprise 11.10.348 and later are supported. Migrating from GitHub Enterprise 11.10.348 and earlier is not supported. You must first upgrade to 11.10.348 in several upgrades. For more information, see the 11.10.348 upgrading procedure, "Upgrading to the latest release."

To upgrade to the latest version of GitHub Enterprise, you must first migrate to GitHub Enterprise Server 2.1, then you can follow the normal upgrade process. For more information, see "Upgrading GitHub Enterprise Server".

Prepare for the migration

1. Review the Provisioning and Installation guide and check that all prerequisites needed to provision and configure GitHub Enterprise 2.1.23 in your environment are met. For more information, see "Provisioning and Installation."

2. Verify that the current instance is running a supported upgrade version.

3. Set up the latest version of the GitHub Enterprise Server Backup Utilities. For more information, see GitHub Enterprise Server Backup Utilities.

   - If you have already configured scheduled backups using GitHub Enterprise Server Backup Utilities, make sure you have updated to the latest version.
   - If you are not currently running scheduled backups, set up GitHub Enterprise Server Backup Utilities.

4. Take an initial full backup snapshot of the current instance using the `ghe-backup` command. If you have already configured scheduled backups for your current instance, you don't need to take a snapshot of your instance.

   > **Tip:** You can leave the instance online and in active use during the snapshot. You'll take another snapshot during the maintenance portion of the migration. Since backups are incremental, this initial snapshot reduces the amount of data transferred in the final snapshot, which may shorten the maintenance window.

5. Determine the method for switching user network traffic to the new instance. After you've migrated, all HTTP and Git network traffic directs to the new instance.

    - **DNS** - We recommend this method for all environments, as it's simple and works well even when migrating from one datacenter to another. Before starting migration, reduce the existing DNS record's TTL to five minutes or less and allow the change to propagate. Once the migration is complete, update the DNS record(s) to point to the IP address of the new instance.
    - **IP address assignment** - This method is only available on VMware to VMware migration and is not recommended unless the DNS method is unavailable. Before starting the migration, you'll need to shut down the old instance and assign its IP address to the new instance.

6. Schedule a maintenance window. The maintenance window should include enough time to transfer data from the backup host to the new instance and will vary based on the size of the backup snapshot and available network bandwidth. During this time your current instance will be unavailable and in maintenance mode while you migrate to the new instance.

Perform the migration

1. Provision a new GitHub Enterprise 2.1 instance. For more information, see the "Provisioning and Installation" guide for your target platform.
2. In a browser, navigate to the new replica appliance's IP address and upload your GitHub Enterprise license.
3. Set an admin password.
4. Click **Migrate**.
5. In the "Add new SSH key" text field, paste your backup host access SSH key.
6. Click **Add key** and then click **Continue**.
7. Copy the `ghe-restore` command that you'll run on the backup host to migrate data to the new instance.
8. Enable maintenance mode on the old instance and wait for all active processes to complete. For more information, see "Enabling and scheduling maintenance mode."

    > **Note:** The instance will be unavailable for normal use from this point forward.

9. On the backup host, run the `ghe-backup` command to take a final backup snapshot. This ensures that all data from the old instance is captured.
10. On the backup host, run the `ghe-restore` command you copied on the new instance's restore status screen to restore the latest snapshot.

    ```
    $ ghe-restore 169.254.1.1
    The authenticity of host '169.254.1.1:122' can't be established.
    RSA key fingerprint is fe:96:9e:ac:d0:22:7c:cf:22:68:f2:c3:c9:81:53:d1.
    Are you sure you want to continue connecting (yes/no)? yes
    Connect 169.254.1.1:122 OK (v2.0.0)
    Starting restore of 169.254.1.1:122 from snapshot 20141014T141425
    Restoring Git repositories ...
    ```

```
Restoring GitHub Pages ...
Restoring asset attachments ...
Restoring hook deliveries ...
Restoring MySQL database ...
Restoring Redis database ...
Restoring SSH authorized keys ...
Restoring Elasticsearch indices ...
Restoring SSH host keys ...
Completed restore of 169.254.1.1:122 from snapshot 20141014T141425
Visit https://169.254.1.1/setup/settings to review appliance configuration.
```

11. Return to the new instance's restore status screen to see that the restore completed.

12. Click **Continue to settings** to review and adjust the configuration information and settings that were imported from the previous instance.

13. Click **Save settings**.

    > **Note:** You can use the new instance after you've applied configuration settings and restarted the server.

14. Switch user network traffic from the old instance to the new instance using either DNS or IP address assignment.

15. Upgrade to the latest patch release of GitHub Enterprise Server. For more information, see "Upgrading GitHub Enterprise Server."

Legal