Configuring the self-hosted runner application as a service

Host your own runners
3 of 8 in learning path

Next: **Using a proxy server with self-hosted runners**

In this article

You can configure the self-hosted runner application as a service to automatically start the runner application when the machine starts.

Mac Windows Linux

Note: You must add a runner to GitHub Enterprise Cloud before you can configure the self-hosted runner application as a service. For more information, see "Adding self-hosted runners."

For Linux systems that use `systemd`, you can use the `svc.sh` script that is created after successfully adding the runner to install and manage using the application as a service.

On the runner machine, open a shell in the directory where you installed the self-hosted runner application. Use the commands below to install and manage the self-hosted runner service.

Note: Configuring the self-hosted runner application as a service on Windows is part of the application configuration process. If you have already configured the self-hosted runner application but did not choose to configure it as a service, you must remove the runner from GitHub and re-configure the application. When you re-configure the application, choose the option to configure the application as a service.

For more information, see "Removing self-hosted runners" and "Adding self-hosted runners."

You can manage the runner service in the Windows **Services** application, or you can use PowerShell to run the commands below.

Note: You must add a runner to GitHub Enterprise Cloud before you can configure the self-hosted runner application as a service. For more information, see "Adding self-hosted runners."

On the runner machine, open a shell in the directory where you installed the self-hosted runner application. Use the commands below to install and manage the self-hosted runner service.

Installing the service

1. Stop the self-hosted runner application if it is currently running.
2. Install the service with the following command:

   ```
   sudo ./svc.sh install
   ```

3. Alternatively, the command takes an optional `user` argument to install the service as a different user.

   ```
   ./svc.sh install USERNAME
   ```

Installing the service

1. Stop the self-hosted runner application if it is currently running.
2. Install the service with the following command:

   ```
   ./svc.sh install
   ```

Starting the service

Start the service with the following command:

```
sudo ./svc.sh start
```

```
Start-Service "actions.runner.*"
```

```
./svc.sh start
```

Checking the status of the service

Check the status of the service with the following command:

```
sudo ./svc.sh status
```

```
Get-Service "actions.runner.*"
```

```
./svc.sh status
```

For more information on viewing the status of your self-hosted runner, see "Monitoring and troubleshooting self-hosted runners."

Stopping the service

Stop the service with the following command:

```
sudo ./svc.sh stop
```

```
Stop-Service "actions.runner.*"
```

```
./svc.sh stop
```

Uninstalling the service

1. Stop the service if it is currently running.
2. Uninstall the service with the following command:

   ```
   sudo ./svc.sh uninstall
   ```

   ```
   Remove-Service "actions.runner.*"
   ```

   ```
   ./svc.sh uninstall
   ```

Customizing the self-hosted runner service

If you don't want to use the above default `systemd` service configuration, you can create a customized service or use whichever service mechanism you prefer. Consider using the `serviced` template at `actions-runner/bin/actions.runner.service.template` as a reference. If you use a customized service, the self-hosted runner service must always be invoked using the `runsvc.sh` entry point.

Customizing the self-hosted runner service

If you don't want to use the above default launchd service configuration, you can create a customized service or use whichever service mechanism you prefer. Consider using the `plist` template at `actions-runner/bin/actions.runner.plist.template` as a reference. If you use a customized service, the self-hosted runner service must always be invoked using the `runsvc.sh` entry point.