
KNOWLEDGE BASE ARTICLE

GLServer Command Line Help

The article provides a detailed guide on using the GLServer command line interface to manage Glitch services, including options for starting, stopping, exporting, and deploying units. It also covers backup, upgrade, and publishing commands.

PUBLISHED

9 Jun 2022

DOCUMENT

InfoHub — Knowledge, news, and updates from EssingtonITS

GLServer Command Line Help

9 Jun 2022

Reference for every `gl` command in GLServer v0.20. All commands are run as `gl --<option> [args]` on the host server.

Contents

- [Units](#)
 - `--list` - List units running on this server.
 - `--start` - Start the specified unit.
 - `--stop` - Stop the specified unit.
 - `--restart` - Restart a unit (or all running units).
 - `--attach` - Attach a shell to a running unit.
 - `--deploy` - Deploy a new unit from the registry.
 - `--cloneu` - Clone an existing unit.
 - `--register` - Register a unit to this host server.
 - `--unregister` - Unregister a unit from this host.
 - `--claim` - Claim a unit for this host.
 - `--unitupdate` - Update a unit's core code from the registry.
- [Backup](#)
 - `--backup` - Backup a unit (or all units).
 - `--exportc` - Export a unit's container to a .tar file.

- `--exportu` - Export a unit (container + data) to .tar.gz.
- `--importc` - Import a container from a .tar file.
- `--importu` - Import a unit from a .tar.gz file.
- [Network](#)
 - `--publish` - Publish a unit via the LB or Kemp.
 - `--unpublish` - Unpublish a unit from the LB or Kemp.
 - `--lb` - Manage the glitch-lb load balancer unit.
 - `--kemp` - Configure glitch.conf for Kemp integration.
- [Services](#)
 - `--api` - Deploy the glitch-api service unit.
 - `--webui` - Deploy the glitch-webuidb service unit.
 - `--sai` - Launch the SophiaAI interactive agent.
- [Server](#)
 - `--upgrade` - Upgrade glitch-cli, or run pending upgrade tasks.
 - `--cache` - List units available from the registry cache.
 - `--info` - Display version and system information.
 - `--version` - Print the GLServer version number.
 - `--help` - Show CLI help.
- [Internal](#)
 - `--test` - Internal: developer test command.

Units

`--list`

Usage: `gl --list [all] [cluster] [json]`

List units running on this server.

Lists the units running on the host. By default only running units are shown.

Modifiers can be combined.

`all`

Include offline (stopped) units.

`cluster`

Include units from every host in the cluster.

`json`

Emit machine-readable JSON instead of the human table.

```
gl --list
gl --list all
gl --list cluster json
```

`--start`

Usage: `gl --start <unit>`

Start the specified unit.

Starts the named unit's container. If the container fails to start, it is removed cleanly so the next attempt is from a clean state.

```
gl --start glitch-moodle
```

`--stop`

Usage: `gl --stop <unit>`

Stop the specified unit.

Stops the named unit's container and removes it. The unit's data on disk is untouched.

```
gl --stop glitch-moodle
```

--restart

Usage: `gl --restart <unit>|all`

Restart a unit (or all running units).

Restart a single unit, or every running unit on this host.

```
gl --restart glitch-moodle
gl --restart all
```

--attach

Usage: `gl --attach <unit>`

Attach a shell to a running unit.

Opens an interactive shell inside the running unit's container (equivalent to `docker exec -it <unit> /bin/bash`).

```
gl --attach glitch-moodle
```

--deploy

Usage: `gl --deploy <unit-name> [service]`

Deploy a new unit from the registry.

Deploys a new unit by pulling its definition from the registry. Requires internet access to the registry host.

service

Optional flag indicating the deployment is a service unit (different lifecycle).

```
gl --deploy glitch-moodle
gl --deploy glitch-api service
```

--cloneu

Usage: `gl --cloneu <unit>`

Clone an existing unit.

Clones a unit's container and data into a new unit. The new unit is automatically claimed by this host server.

Caveat: the CLI clone does not run pre/post clone scripts. For full clone semantics use the WebUI.

```
gl --cloneu glitch-moodle
```

--register

Usage: `gl --register <unit>`

Register a unit to this host server.

Registers the specified unit to this host. Use after manually placing a unit on disk so the host claims ownership.

```
gl --register glitch-moodle
```

--unregister

Usage: `gl --unregister <unit>`

Unregister a unit from this host.

Removes the host's claim on a unit by deleting its server.conf. The unit must be offline before unregistering.

```
gl --stop glitch-moodle  
gl --unregister glitch-moodle
```

--claim

Usage: `gl --claim <unit>`

Claim a unit for this host.

Switches the host server recorded against a unit to this server. Useful when migrating units between hosts.

```
gl --claim glitch-moodle
```

--unitupdate

Usage: `gl --unitupdate <unit>`

Update a unit's core code from the registry.

Pulls the latest core code for a deployed unit from the registry, leaving the unit's data alone.

```
gl --unitupdate glitch-moodle
```

Backup

--backup

Usage: `gl --backup <unit>|all`

Backup a unit (or all units).

Backs up a unit's container and data to `backup_volume`. Pass `all` to back up every unit on this host.

Requires:

- `backup_volume` in `glitch.conf`
- `backup_expire_days` in `glitch.conf`

```
gl --backup glitch-moodle  
gl --backup all
```

`--exportc`

Usage: `gl --exportc <unit>`

Export a unit's container to a .tar file.

Exports the unit's running container to a `.tar` archive in the unit's directory. Useful for moving a container to another host or archiving its image state.

```
gl --exportc glitch-moodle
```

`--exportu`

Usage: `gl --exportu <unit>`

Export a unit (container + data) to .tar.gz.

Exports the unit's container *and* data into a single `.tar.gz` placed in the `exports/` directory of the glitch volume.

```
gl --exportu glitch-moodle
```

`--importc`

Usage: `gl --importc <path/to/file.tar>`

Import a container from a .tar file.

Imports a container previously exported with `--exportc`.

```
gl --importc /tmp/glitch-moodle.tar
```

`--importu`

Usage: `gl --importu <path/to/file.tar.gz>`

Import a unit from a .tar.gz file.

Imports a complete unit (container and data) previously exported with `--exportu`.

```
gl --importu /tmp/glitch-moodle.tar.gz
```

Network

--publish

Usage: `gl --publish <http|https> <unit> [<port>]`

Publish a unit via the LB or Kemp.

Publishes a unit so it is reachable through the configured load balancer (glitch-lb) and/or Kemp load master.

`http|https`

Protocol to publish.

`unit`

The unit name to publish.

`port`

Optional. Defaults to 80 for http or 443 for https.

For HTTPS the unit must contain `cert.crt` and `cert.key`.

```
gl --publish https glitch-moodle
gl --publish http glitch-moodle 8080
```

--unpublish

Usage: `gl --unpublish <http|https> <unit>`

Unpublish a unit from the LB or Kemp.

Removes the load balancer publication for a unit from glitch-lb and/or Kemp.

```
gl --unpublish https glitch-moodle
```

--lb

Usage: `gl --lb deploy|reload`

Manage the glitch-lb load balancer unit.

Manage the glitch-lb (NGINX-based) load balancer unit.

`deploy`

Deploy and start glitch-lb. Persists `glitch_lb=` in `glitch.conf` so other commands can find it.

`reload`

Force the running glitch-lb to reload its configuration.

```
gl --lb deploy
gl --lb reload
```

`--kemp`

Usage: `gl --kemp enable`

Configure `glitch.conf` for Kemp integration.

Appends placeholder Kemp configuration entries to `glitch.conf` :

- `kemp_ip`
- `kemp_api_user`
- `kemp_api_pass`
- `kemp_vs_ip`

Edit the values to match your Kemp Load Master, then publish/unpublish commands will route through Kemp.

```
gl --kemp enable
```

Services

`--api`

Usage: `gl --api deploy`

Deploy the glitch-api service unit.

Deploys the `glitch-api` unit, which exposes the GLServer REST API and OpenAPI docs on port 9998.

```
gl --api deploy
```

After deploy, browse `https://<host>:9998/docs` .

`--webui`

Usage: `gl --webui deploy`

Deploy the glitch-webuidb service unit.

Deploys the `glitch-webuidb` unit, which provides the database backend for the GLServer WebUI on port 9997.

```
gl --webui deploy
```

`--sai`

Usage: `gl --sai`

Launch the SophiaAI interactive agent.

Launches SophiaAI, an interactive command-line agent (delegates to `bin/sai.sh`).

```
gl --sai
```

Server

`--upgrade`

Usage: `gl --upgrade [<source>|tasks]`

Upgrade glitch-cli, or run pending upgrade tasks.

Upgrades the glitch-cli core code.

(no argument)

Fetches the latest release from the remote registry.

`<local-path>`

Uses a local directory as the source - useful for development.

`tasks`

Runs any pending upgrade/migration tasks without downloading code. Run after a manual deploy.

```
gl --upgrade
gl --upgrade /tmp/glserver
gl --upgrade tasks
```

After upgrade, run `gl --info` to verify required PHP modules are present.

`--cache`

Usage: `gl --cache`

List units available from the registry cache.

Lists every unit available for deploy from the local registry cache.

```
gl --cache
```

`--info`

Usage: `gl --info`

Display version and system information.

Prints the version table from `version.php` plus relevant system information for diagnostics.

```
gl --info
```

`--version`

Usage: `gl --version`

Print the GLServer version number.

Prints just the version number with no formatting - useful for shell scripts.

```
gl --version
```

--help

Usage: `gl --help`

Show CLI help.

Shows the CLI help text - the same content as this knowledge base in plain-text form.

```
gl --help
```

Managing modern applications across servers, networks, and environments can quickly become complex and costly.

[EssingtonITS GL Server](#) is a powerful platform designed to simplify the deployment and management of containerised application Units. Whether running a single server or a large multi-environment infrastructure, GL Server helps organisations deliver secure, scalable, and resilient services without requiring in-house DevOps expertise.