

Guide to upgrading dependencies

Upgrading Go or Node.js requires making changes in many different files. See below for a list and explanation for each.

Go

- Drone
- grafana/build-container
- Appveyor
- Dockerfile

Node.js

- Drone
- grafana/build-container
- Appveyor
- Dockerfile
- .github/workflows/publish.yml

Go dependencies

The Grafana project uses Go modules to manage dependencies on external packages. This requires a working Go environment with version 1.11 or greater installed.

To add or update a new dependency, use the `go get` command:

```
go get example.com/some/module/pkg
```

```
# Pick a specific version.
```

```
go get example.com/some/module/pkg@vX.Y.Z
```

Tidy up the `go.mod` and `go.sum` files:

```
go mod tidy
```

You have to commit the changes to `go.mod` and `go.sum` before submitting the pull request.

To understand what the actual dependencies of `grafana-server` are, one could run it with `-vv` flag. This might produce an output, different from `go.mod` contents and `-vv` option is the source of truth here. It lists the modules *compiled* into the executable, while `go.mod` lists also test and weak transitive dependencies (modules, used in some package, which is not in use by itself). If you are interested in reporting a vulnerability in a dependency module - please consult `-vv` output, maybe the “dependency” is not a dependency as such.

Upgrading dependencies

If you need to upgrade a direct or indirect dependency, you can do it like so, `$MODULE` being the dependency in question: `go get -u $MODULE`. The corresponding entry in `go.mod` should then have the version you specified; if it's an indirect dependency, the entry should have the `// indirect` comment. Follow this by executing `go mod tidy`, to ensure that `go.mod` and `go.sum` are up to date. If the indirect dependency turns out to not be used (transitively) by any of our packages, `go mod tidy` will actually strip it from `go.mod`. In that case, you can just ignore it since it isn't used in the end.

Node.js dependencies

Updated using `yarn`.

- `package.json`

Where to make changes

Drone

Our CI builds run on Drone.

Files

- `.circleci/config.yml`.

Dependencies

- `nodejs`
- `golang`
- `grafana/build-container` (our custom docker build container)

grafana/build-container

The main build steps (in Drone) happen using a custom Docker image that comes pre-baked with some of the necessary dependencies.

Link: `grafana/build-container`

Dependencies

- `fpm`
- `nodejs`
- `golang`
- `crosscompiling` (several compilers)

Appveyor

Main and release builds trigger test runs on Appveyors build environment so that tests will run on Windows.

Files:

- `appveyor.yml`

Dependencies

- nodejs
- golang

Dockerfile

There is a Docker build for Grafana in the root of the project that allows anyone to build Grafana just using Docker.

Files

- `Dockerfile`

Dependencies

- nodejs
- golang

Local developer environments

Please send out a notice in the grafana-dev slack channel when updating Go or Node.js to make it easier for everyone to update their local developer environments.