GoReleaser

Ship Go releases

CONCEPTS

GoReleaser releases the latest git tag. Use semver tags (v1.2.3) and a clean working tree, or add --snapshot to skip the checks.

Everything lands in dist/. Use --clean on every run so old artifacts do not mix with new ones.

SETUP

goreleaser init
Write an example .goreleaser.yaml

goreleaser check
Validate the config file

goreleaser --help
List commands and global flags

BUILD

goreleaser build --snapshot --clean
Build all targets without a tag

goreleaser build --single-target --clean
Build only for this machine

RELEASE

goreleaser release --clean
Publish the release for the current tag

goreleaser release --snapshot --clean
Full local run, nothing published

goreleaser release --skip=publish
Build and package but do not publish

git tag -a v<x.y.z> **-m "**<msg>**"**
Create the release tag

git push origin v<x.y.z>
Push the tag

CONFIG KEYS

version: 2
First line of the config file

builds:
Binaries to compile

archives:
How binaries are packaged

BUILDS

main: ./cmd/<name>
Path of the main package

goos: [linux, darwin, windows]
Target operating systems

goarch: [amd64, arm64]
Target architectures

env: [CGO_ENABLED=0]
Build environment variables

ldflags: [-s -w -X main.version={{.Version}}]
Linker flags with version stamp

ARCHIVES AND CHECKSUMS

formats: [tar.gz]
Archive formats (tar.gz is default)

name_template: "{{ .ProjectName }}_{{ .Version }}_{{ .Os }}_{{ .Arch }}"
Archive file name

format_overrides: [{goos: windows, formats: [zip]}]
Zip for Windows only

CHANGELOG AND RELEASE

changelog: filters: exclude: ["^docs:"]
Hide commits matching a regex

release: prerelease: auto
Mark rc/beta tags as pre-releases

TEMPLATES AND SIGNING

{{ .Version }}
Version without the leading v

RECIPES

Minimal .goreleaser.yaml

```
version: 2
builds:
  - env: [CGO_ENABLED=0]
    goos: [linux, darwin, windows]
    goarch: [amd64, arm64]
    ldflags: [-s -w -X
main.version={{.Version}}]
archives:
  - formats: [tar.gz]
    format_overrides:
      - {goos: windows, formats: [zip]}
```