

Разработка CLI на Golang
или
Автоматизация контура
разработки

Мастер-класс



Немировский Лев Дмитриевич

Руководитель направления по
автоматизации разработки

В Банке с 2020 года

DevOps/Веб-разработка

«Интерфейс командной строки (англ. Command line interface, CLI) – разновидность текстового интерфейса (TUI) между человеком и компьютером, в котором инструкции компьютеру даются в основном путём ввода с клавиатуры текстовых строк».

— Wikipedia

Laravel Framework 8.73.2

Usage:

command [options] [arguments]

Options:

-h, --help Display help for the given command. When no command is given display help for the `list` command
-q, --quiet Do not output any message
-V, --version Display this application version
--ansi|--no-ansi Force (or disable `--no-ansi`) ANSI output
-n, --no-interaction Do not ask any interactive question
--env[=ENV] The environment the command should run under
-v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:

clear-compiled Remove the compiled class file
completion Dump the shell completion script
db Start a new database CLI session
down Put the application into maintenance / demo mode
env Display the current framework environment
help Display help for a command
inspire Display an inspiring quote
list List commands
migrate Run the database migrations
optimize Cache the framework bootstrap files
serve Serve the application on the PHP development server
test Run the application tests
tinker Interact with your application
up Bring the application out of maintenance mode
auth
auth:clear-resets Flush expired password reset tokens
cache
cache:clear Flush the application cache
cache:forget Remove an item from the cache
cache:table Create a migration for the cache database table
config
config:cache Create a cache file for faster configuration loading
config:clear Remove the configuration cache file
db
db:seed Seed the database with records

```
create-nuxt-app v3.7.1
* Generating Nuxt.js project in nuxt-cli
? Project name: nuxt-cli
? Programming language: JavaScript
? Package manager: Npm
? UI framework: Bootstrap Vue
? Nuxt.js modules:
  > Axios - Promise based HTTP client
  ○ Progressive Web App (PWA)
  ○ Content - Git-based headless CMS
```

```
# Get more information before running the command
ghost install --help

# Install in development mode for a staging env
ghost install --development, ghost install -D

# Select the directory to install Ghost in
ghost install --dir path/to/dir

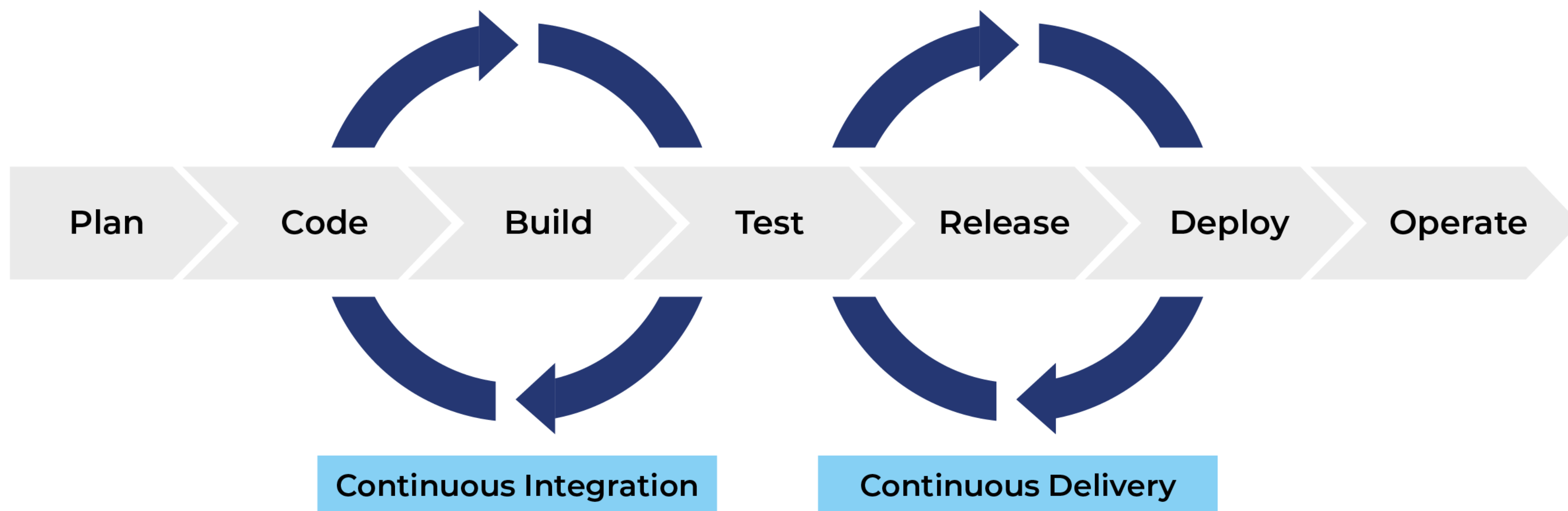
# Install Ghost from a specific zip (useful for testing or custom builds)
ghost install --zip path/to/file.zip

# Disable stack checks
ghost install --no-stack

# Install without running setup
ghost install --no-setup

# Install without starting Ghost
ghost install --no-start
```

CI/CD





- Несколько разработчиков/тестировщиков
- Сервер Debian 10
- SQLite
- Бекенд на Laravel (PHP 8 + composer) – на каждого пользователя
- Frontend на Nuxt.js (Node.js + Npm) – на каждого пользователя
- Nginx
- Домен третьего уровня, dev.levn.tech
- Основная папка `/var/www/{user_name}`
- Системный пользователь: `www-data`

```
# Добавляем php репу
sudo apt install apt-transport-https lsb-release ca-certificates git gcc g++ make -y
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
sudo sh -c 'echo "deb https://packages.sury.org/php/ \$(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list'
sudo apt update
# Установим php
sudo apt install php8.0-{common,cli,fpm,mysql,bz2,curl,intl,readline,xml,zip} -y
sudo systemctl enable php8.0-fpm
# Установим composer
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8caba5b96ac8b64115ac402c8cd292b8a03482574915d1a8')
{ echo 'Installer verified';
} else { echo 'Installer corrupt'; unlink('composer-setup.php');
} echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/local/bin/composer
# Установим nginx
sudo apt install nginx -y
sudo systemctl enable nginx
# Установим node.js
curl -sL https://deb.nodesource.com/setup_16.x | sudo bash -
sudo apt install nodejs
```



```
# Добавить пользователя  
sudo adduser -G sudo,www-data nemirlev
```

Создание двух площадок

```
# Создаем папку пользователя. Идеально определять по юзеру
sudo mkdir /var/www/nemirlev
# Заходим в нашу папку
cd /var/www/nemirlev
# Дадим права
sudo chown -R www-data:www-data /var/www/
sudo find /var/www/ -type f -exec chmod 2664 {} \;
sudo find /var/www/ -type d -exec chmod 2775 {} \;
# Клонировем репу бекенда
git clone https://github.com/nemirlev/laravel-cli.git /var/www/nemirlev/backend
# Копируем .env
php -r "file_exists('/var/www/nemirlev/backend/.env') || copy('/var/www/nemirlev/backend/.env.example', '/var/www/nemirlev/backend/.env');"
# Устанавливаем зависимости
cd /var/www/nemirlev/backend/
composer update
# Генерируем ключ
php artisan key:generate
# Даем права на папку кэша
chmod -R 777 storage /var/www/nemirlev/bootstrap/cache
# Создаем БД
touch database/database.sqlite
# Заменяем адрес сервера и root директорию
sed -i 's/server_name.*/server_name backend.nemirlev.dev.levn.tech;/g' nginx.conf
sed -i 's/root.*/root \/var\/www\/nemirlev\/backend\/public;/g' nginx.conf
sed -i 's/access_log.*/access_log \/var\/www\/nemirlev\/backend\/storage\/logs\/nginx.access.log;/g' nginx.conf
sed -i 's/error_log.*/error_log \/var\/www\/nemirlev\/backend\/storage\/logs\/nginx.error.log;/g' nginx.conf
# Переносим конфиг nginx
mv /var/www/nemirlev/backend/nginx.conf /etc/nginx/sites-available/backend.nemirlev.dev.levn.tech.conf
ln -s /etc/nginx/sites-available/backend.nemirlev.dev.levn.tech.conf /etc/nginx/sites-enabled/
sudo service nginx restart
# Клонировем frontend
git clone https://github.com/nemirlev/nuxt-cli.git /var/www/nemirlev/frontend
# Устанавливаем зависимости
cd /var/www/nemirlev/frontend/
npm install
# Собираем проект
npm run build
npm run generate
# Заменяем адрес сервера и root директорию
sed -i 's/server_name.*/server_name frontend.nemirlev.dev.levn.tech;/g' nginx.conf
sed -i 's/root.*/root \/var\/www\/nemirlev\/frontend\/dist;/g' nginx.conf
# Переносим конфиг nginx
mv /var/www/nemirlev/frontend/nginx.conf /etc/nginx/sites-available/frontend.nemirlev.dev.levn.tech.conf
ln -s /etc/nginx/sites-available/frontend.nemirlev.dev.levn.tech.conf /etc/nginx/sites-enabled/
sudo service nginx restart
```

spf13 / cobra Public

Watch 349 Star 24.1k Fork 2.1k

Code Issues 255 Pull requests 100 Actions Projects Wiki Security Insights

master 5 branches 16 tags

Go to file Add file Code

umarcor args_test: add helper functions (#1426) 9e1d6f1 14 days ago 852 commits

.github	ci: test golang 1.16.x and 1.17.x too (#1425)	3 months ago
cobra	fix(diff): use arg '--strip-trailing-cr' (#949)	14 days ago
doc	go.mod: cpuguy83/go-md2man/v2 v2.0.1 (#1460)	4 months ago
.gitignore	leverage makefile to run build tasks (#976)	2 years ago
.golangci.yml	Use golangci-lint (#1044)	10 months ago
.mailmap	Add .mailcap for a more tidy "git shortlog" output	6 years ago
CHANGELOG.md	Fix typo	10 months ago
CONDUCT.md	Cobra User Contract (#1292)	10 months ago
CONTRIBUTING.md	Add CONTRIBUTING.md (#1183)	16 months ago
LICENSE.txt	adding license	8 years ago
Makefile	Use golangci-lint (#1044)	10 months ago
README.md	Updating generator documentation and links	28 days ago
args.go	Fish completion using Go completion (#1048)	2 years ago

About

A Commander for modern Go CLI interactions

cobra.dev

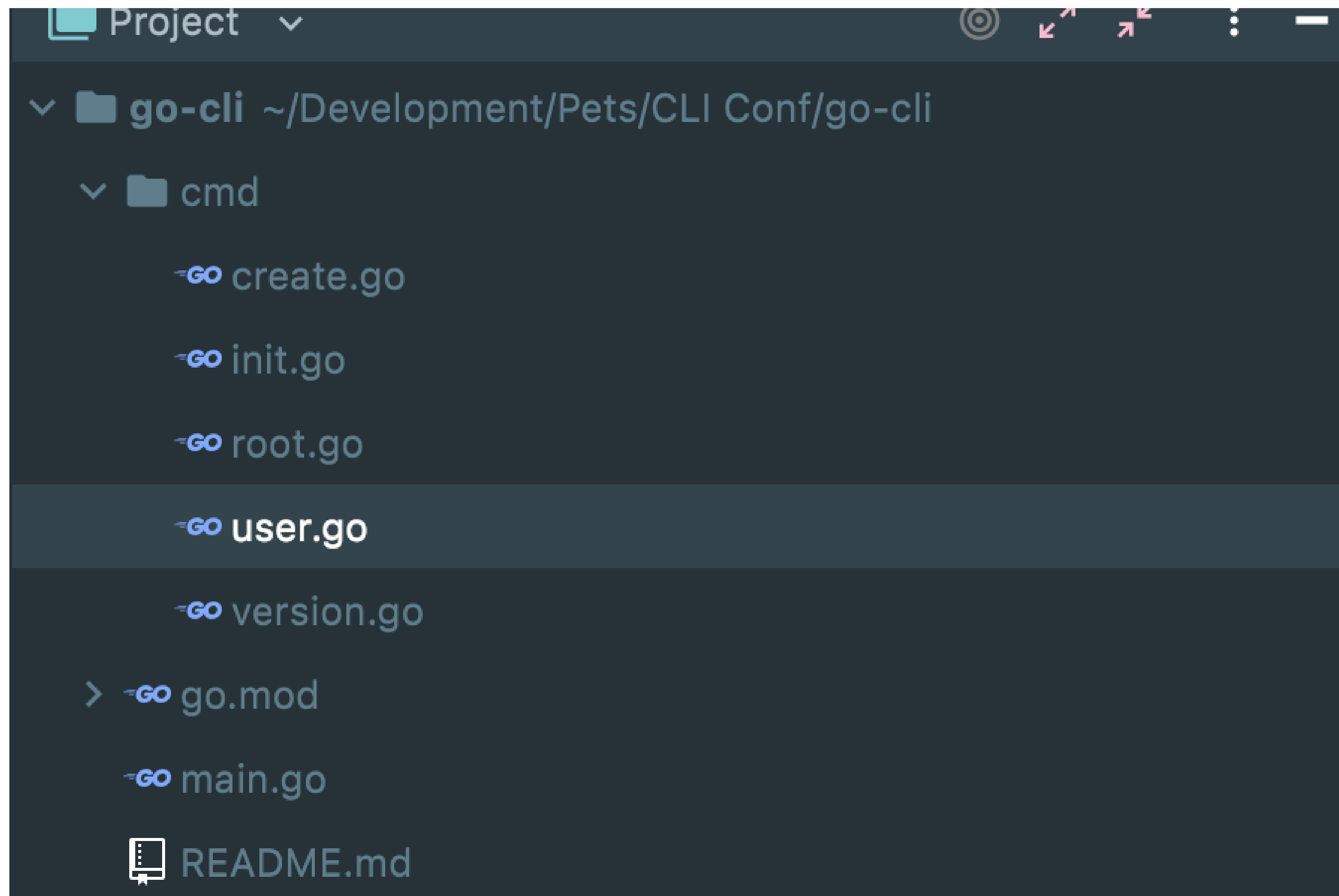
go cli golang commandline command command-line posix subcommands cli-app golang-library cobra golang-application cobra-library cobra-generator posix-compliant-flags command-cobra

Readme Apache-2.0 License

Releases 14

v1.2.1 Latest on 2 Jul

+ 13 releases



```
package main

import (
    "go-cli/cmd"
)

func main() {
    cmd.Execute()
}
```

```
package cmd

import (
    "fmt"
    "os"

    homedir "github.com/mitchellh/go-homedir"
    "github.com/spf13/cobra"
    "github.com/spf13/viper"
)

var (
    // This is used for config file
    configFile string

    rootCmd = &cobra.Command{
        Use:     "psb",
        Short:   "psb: A CLI for Debian 10 developer server",
        Long:    `psb: A sample CLI for deploy laravel and nuxt.js for each developer.`,
    }
)

// Execute executes the root command.
func Execute() error {
    return rootCmd.Execute()
}

func init() {
    cobra.OnInitialize(initConfig)
    rootCmd.PersistentFlags().StringVar(&configFile, "config", "", "config file (default is $HOME/.psb.yaml)")
}
```

```
package cmd

import (
    "fmt"
    "github.com/spf13/cobra"
)

func init() {
    rootCmd.AddCommand(versionCmd)
}

var versionCmd = &cobra.Command{
    Use:     "version",
    Short:   "Print version number of psb tool",
    Long:    `This command can be used get the version number of psb tool`,
    Run: func(cmd *cobra.Command, args []string) {
        fmt.Println("psb v0.0.1-alpha")
    },
}
```

```
import (  
    "fmt"  
    "github.com/AlecAivazis/survey/v2"  
    "github.com/spf13/cobra"  
    "log"  
    "os/exec"  
)  
  
func init() {  
    rootCmd.AddCommand(initCmd)  
}  
  
var initCmd = &cobra.Command{  
    Use:     "init",  
    Aliases: []string{"int"},  
    Short:   "Prepare server (php, nginx, node)",  
    Long:    `This command can be used for settings up server – install software`,  
    Run: func(cmd *cobra.Command, args []string) {  
        // *** add code to invoke automation end points below ***  
        freshInstallInteractive()  
    },  
}
```



```
func freshInstallInteractive() {
    sure := false
    prompt := &survey.Confirm{
        Message: "Вы уверены что хотите установить?",
    }
    err := survey.AskOne(prompt, &sure)
    if err != nil {
        fmt.Println(err.Error())
        return
    }

    if sure {
        out, err := exec.Command("apt", "install", "apt-transport-https", "lsb-release", "ca-certificates", "git", "gcc", "g++", "make", "-y").Output()
        if err != nil {
            log.Fatal(err)
        }
        fmt.Println(string(out))
        fmt.Println("Вы успешно установили ПО")
        return
    }

    fmt.Println("Вы отказались от установки ПО")
}
```

```
func freshInstallInteractive() {
    sure := false
    prompt := &survey.Confirm{
        Message: "Вы уверены что хотите установить?",
    }
    err := survey.AskOne(prompt, &sure)
    if err != nil {
        fmt.Println(err.Error())
        return
    }

    if sure {
        installDefaultPrograms()
        fmt.Println("Вы успешно установили ПО")
        return
    }

    fmt.Println("Вы отказались от установки ПО")
}
```

```
func installDefaultPrograms() {
    out, err := exec.Command("apt", "install", "apt-transport-https", "lsb-release", "ca-certificates", "git", "gcc", "g++", "make", "-y").Output()
    if err != nil {
        log.Fatal(err)
    }
    fmt.Println("Идет установка ПО")
    fmt.Println(string(out))
}
```

```
go build -o build/psb
```

```
root@webserver:/var/www/go-cli/build# ./psb
psb: A sample CLI for deploy laravel and nuxt.js for each developer.

Usage:
  psb [command]

Available Commands:
  completion  generate the autocompletion script for the specified shell
  create      Create laravel and Nuxt project for user
  help        Help about any command
  init        Prepare server (php, nginx, node)
  user        Add user and group
  version     Print version number of psb tool

Flags:
  --config string  config file (default is $HOME/.psb.yaml)
  -h, --help      help for psb

Use "psb [command] --help" for more information about a command.
root@webserver:/var/www/go-cli/build# ./psb init --help
This command can be used for settings up server - install software

Usage:
  psb init [flags]

Aliases:
  init, int

Flags:
  -h, --help  help for init

Global Flags:
  --config string  config file (default is $HOME/.psb.yaml)
```

```
root@webserver:/var/www/go-cli/build# ./psb int --help
This command can be used for settings up server - install software

Usage:
  psb init [flags]

Aliases:
  init, int

Flags:
  -h, --help  help for init

Global Flags:
  --config string  config file (default is $HOME/.psb.yaml)
```

```
root@webserver:/var/www/go-cli/build# ./psb init
? Вы уверены что хотите установить? (y/N) █
```

```
root@webserver:/var/www/go-cli/build# ./psb init
? Вы уверены что хотите установить? Yes
Идет установка ПО
Reading package lists...
Building dependency tree...
Reading state information...
apt-transport-https is already the newest version (1.8.2.3).
ca-certificates is already the newest version (20200601~deb10u2).
g++ is already the newest version (4:8.3.0-1).
gcc is already the newest version (4:8.3.0-1).
git is already the newest version (1:2.20.1-2+deb10u3).
lsb-release is already the newest version (10.2019051400).
make is already the newest version (4.2.1-1.2).
The following packages were automatically installed and are no longer required:
  libbrotli1 libc-ares2 libuv1
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.

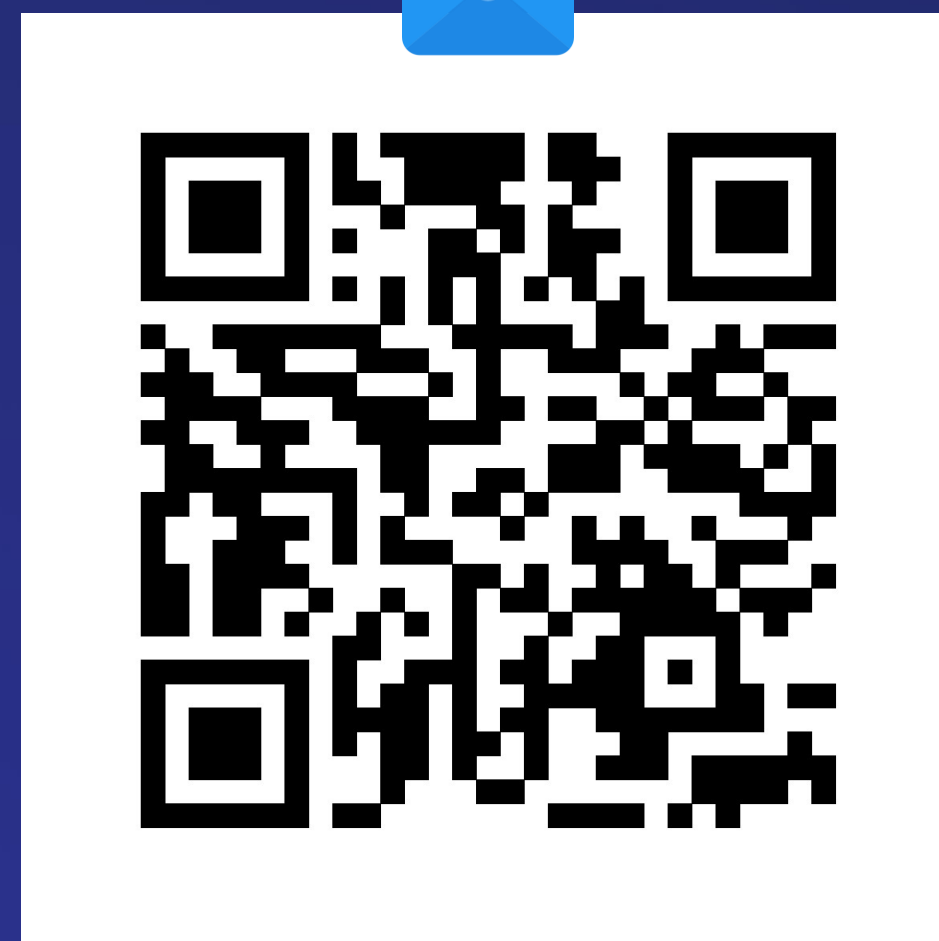
Вы успешно установили ПО
root@webserver:/var/www/go-cli/build# █
```

- Настройку сервера вынести в Ansible
- Настройку пользователя повесить на хуки
- Список веток через API и вывод при создании
- Вынести из `init` каждое ПО в аргументы
- Настройка пакетов при установке php
- Создать аргументы `frontend` и `backend` для независимого развертывания
- Сделать статус выполнения на долгие задачи

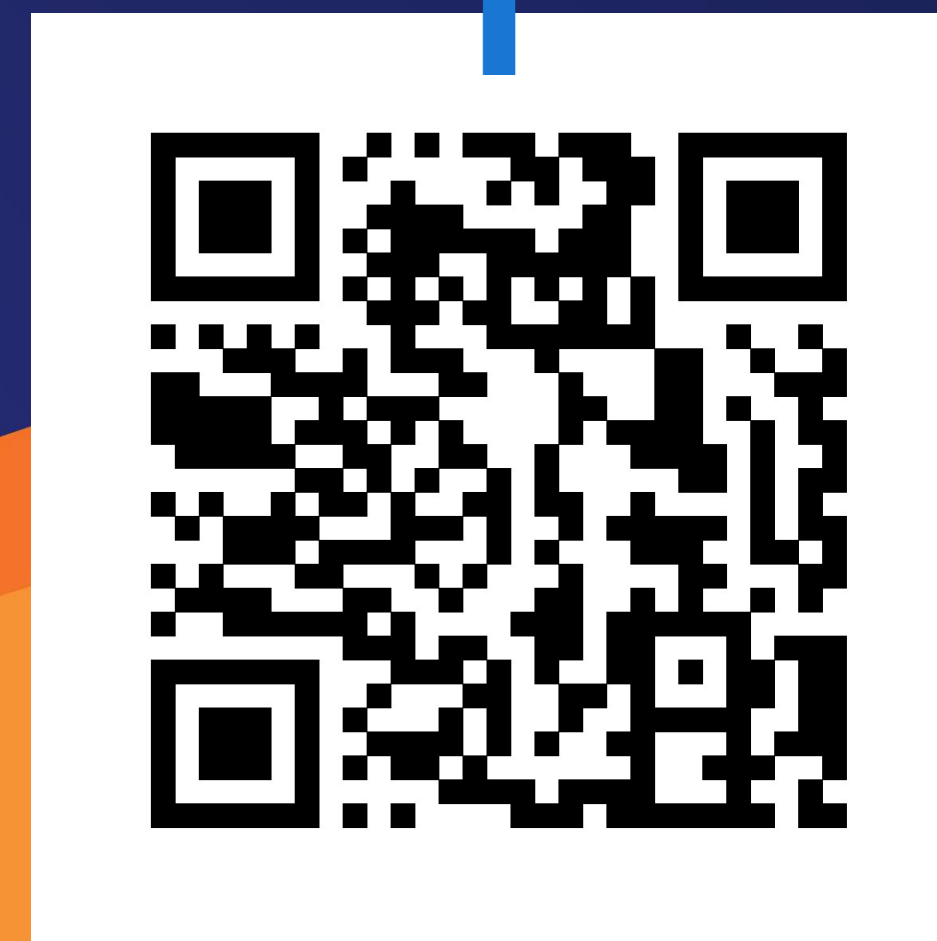
СПАСИБО ЗА ВНИМАНИЕ!



@nemirlev



nemirovskiyld@psbank.ru



lnemirovskij