ROUTE 256

OZON tech

Настройка окружения

Егор Стручин

Команда тестирования микросервисов

Егор Стручин

Преподаватель

- Руководитель группы тестирования микросервисов
- В Озон с 2019 года
- Много хобби, мало времени
- Могу рассказать анекдот

На прошлом занятии

- Поговорили про теорию тестирования
- Обсудили, чем занимаются тестировщики в Озоне

План занятия

Цель: развернуть проект и работать в нем

Инструменты:

1. Развернуть проект
2. Пакетные менеджеры
3. Golang
4. IDE GoLand
5. Утилиты учебного проекта
6. environment и конфигурации

1 Развернуть проект

2 Менеджеры пакетов

Как поставить приложения

В чем проблема просто скачать с сайта?
Что может пойти не так?

- Надежность источника?
- Разные архитектуры (M1) или OS
- Third-party libs (сторонние зависимости)
- Периодически надо обновляться

Homebrew

Менеджер недостающих пакетов для macOS

MacOS

> /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)»

Windows or Linux

Windows

> https://chocolatey.org/

example: choko install 7zip

Linux

уже все на борту

Что дает нам пакетный менеджер?

- Integrity — Целостность скачанного пакета
- Security — Доверенный источник пакета
- Простое обновление
- Управление зависимостями — поставит приложение и все, что нужно
- Установка крупных приложений — через - - cask / snap

Вывод:
Используйте
пакетные менеджеры!

3+4

IDE GoLand + Golang

GoLand

https://www.jetbrains.com/help/go/installation-guide.html

MacOS

> brew install --cask goland

brew cask – это расширение для brew, оно описывает установку приложений с графическим интерфейсом

Ubuntu

> sudo snap install goland --classic

GoLand: Go SDK

https://go.dev/doc/install

MacOS

> brew install go – будет установлена последняя стабильная версия

> brew install go@1.18 – будет установлена версия 1.18.X

GoLand: Go SDK

https://www.jetbrains.com/help/go/installation-guide.html

Goland > Preferences > GOROOT

Плюсик на GOROOT > Download

5 Утилиты учебного проекта

Docker

Контейнеризация приложения и сопутствующих сервисов

MacOS: Docker Desktop

> brew install --cask docker

Ubuntu: Docker Engine

> curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \

sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

> sudo apt-get update

> sudo apt-get install docker-ce docker-ce-cli containerd.io

Проверим установку?

> docker run hello-world

make

Лучшее место для начала знакомства с проектом

MacOS

> brew install make

Ubuntu

> sudo apt-get install build-essential

Проверим установку?

> make --version

curl (и jq)

client URL, использует возможности библиотеки libcurl
jq для форматирования и работы с json в терминале

MacOS

> brew install curl

Ubuntu

уже на борту.

Проверим установку?

> curl --version

> jq

golangci-lint

Линтер, быстрый, параллельный, yaml-конфиг

MacOS

> brew install golangci-lint

Ubuntu

> # binary will be $(go env GOPATH)/bin/golangci-lint curl -sSfL https://raw.githubusercontent.com/golangci/golangci-lint/master/install.sh | sh -s -- -b $(go env GOPATH)/bin v1.49.0

Проверим установку?

> golangci-lint --version

protobuf

...think XML, but smaller, faster, and simpler
Сериализация данных, общение между сервисами

MacOS

> brew install protobuf

Ubuntu

> sudo apt install protobuf-compiler

Проверим установку?

> protoc --version

grpc_cli
клиент для grpc

MacOS

> brew install grpc

Ubuntu

> sudo apt install grpc

Проверим установку?

> grpc_cli help

Goose

клиент для миграций postgres

MacOS

> brew install goose

Ubuntu

> sudo apt install goose

Проверим установку?

> goose -version

Extras

Postman

MacOS:

> brew install --cask postman

Ubuntu:

> sudo apt install postman

grpcurl

like curl for gRPC

MacOS: Docker Desktop

> brew install grpcurl

Ubuntu: Docker Engine

> sudo apt install grpcurl

Проверим установку?

> grpcurl --version

dive

dive into images

MacOS: Docker Desktop

```
> brew install dive
```

Ubuntu: Docker Engine

```
> wget \
 https://github.com/wagoodman/dive/releases/download/v0.9.2/dive_0.9.2_linux_amd64.deb
```

Проверим установку?

```
> dive --version
```

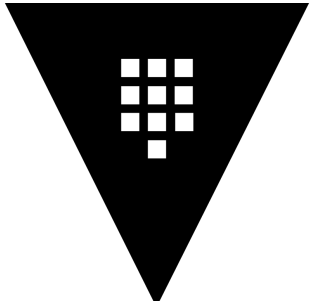
6 environment

Конфиг приложения

Аргументы командной строки

Файлы конфигов

etcd realtime config



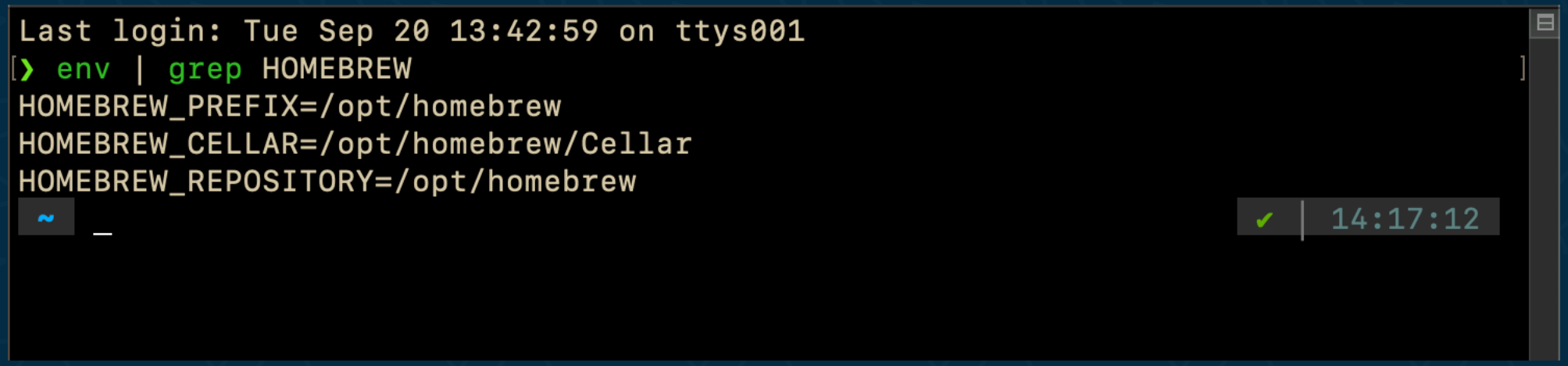
Vault секретные секреты

Переменные окружения

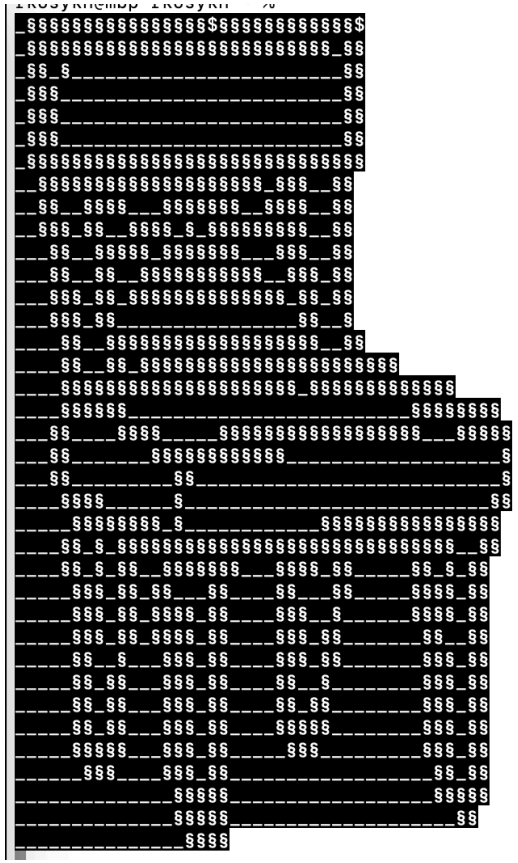
Переменные окружения – это переменные, которые:

- определены оболочкой
- используются программами во время выполнения

Есть системные и пользовательские.



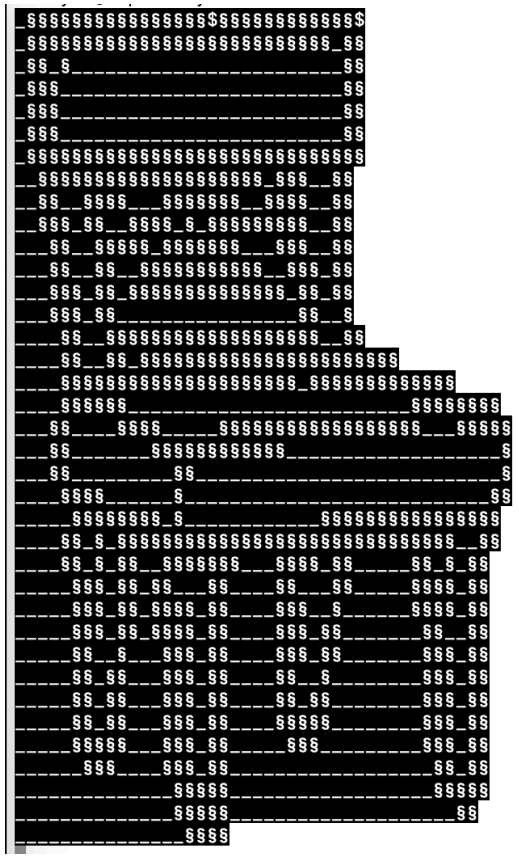
Два терминала – два конфига?



Запись переменной
route256=nice

Чтение переменной
echo $route256

Область хранения



go env
окружение go

1. Большинство из них – это переменные по умолчанию для вашей платформы

2. Можно переопределить
   > go env -w GOPROXY="https://goproxy.s.o3.ru"

3. Переопределенные значения хранятся в конфигах
   - MacOS: "$HOME/Library/Application Support/go/env»
   - Ubuntu: "$HOME/.config/go/env"

Мы рассмотрели

- Пакетные менеджеры – что это такое и зачем нужно
- Установку IDE GoLand
- Установку Golang
- Установку утилит
- Что такое environment и зачем нужно

На следующем занятии

- Docker
- HTTP
- gRPC

…

Вопросы

Спасибо за внимание!

Егор Стручин

Команда тестирования микросервисов