

The JetBrains logo is located in the top right corner. It consists of a stylized, colorful shape resembling a 'J' or a bracket, with a gradient from purple to yellow. Inside this shape is a black square containing the text 'JET BRAINS' in white, uppercase letters, with a horizontal line below it.

**JET  
BRAINS**

# Infrastructure as Code Testing Framework

---

стажёр: Ёров Собир  
руководитель: Владислав Рассохин

# Глоссарий

---

- Практика “Infrastructure as Code” – автоматизация установки и конфигурации серверов и приложений с помощью скриптов
- 
- Ansible – система управления конфигурациями
- 
- Terraform – инструмент для создания, изменения и версионирования инфраструктуры
- 
- BDD (Behaviour-Driven-Development) тестирование – использование естественных языков для описания тестов

# Актуальность задачи

---

- В Teamcity следуют практике “Infrastructure as Code” с написанием скриптов на Ansible и Terraform
- Автоматические тесты для проверки правильности установки и конфигурации серверов и приложений

# Постановка задачи

---

- Написать фреймворк для тестирования автоматизированной инфраструктуры
  - набор локальных ресурсов (процесс, stdout, файл, порт)
  - Тестирование локальной и удалённой машины (ssh, winrm)
  - test runner для запуска terraform-модулей и ansible-ролей
- Свойства фреймворка
  - в стиле Behaviour-Driven-Development (spek)
  - выразительные и удобные ассерты

# Существующие решения

---

- Chef Inspec, ServerSpec
  - тесты представляют собой код на Ruby
- Goss
  - слишком ограниченный (не все можно протестировать)

# Примеры тестов

---

- describe file('/etc/build') do  
  its(:content) { should match '2.2' }  
end
- describe command('git --version') do  
  its(:exit\_status) { should eq 0 }  
  its(:stdout) { should match 'version 2.1' }  
end

```
package:  
  apache2:  
    installed: true  
service:  
  apache2:  
    enabled: true  
    running: true
```



Goss - Quick and Easy server validation

# Результаты

```
@Environment
```

```
machine {
```

```
    local()
```

```
    // remote(hostname, username, password)
```

```
    // remote(hostname, username, privateKeyPath)
```

```
}
```

```
@Test("test git version")
```

```
command("git --version") {
```

```
    exitCode equalTo 0
```

```
    output matches "version 2\\.7.[2-3]"
```

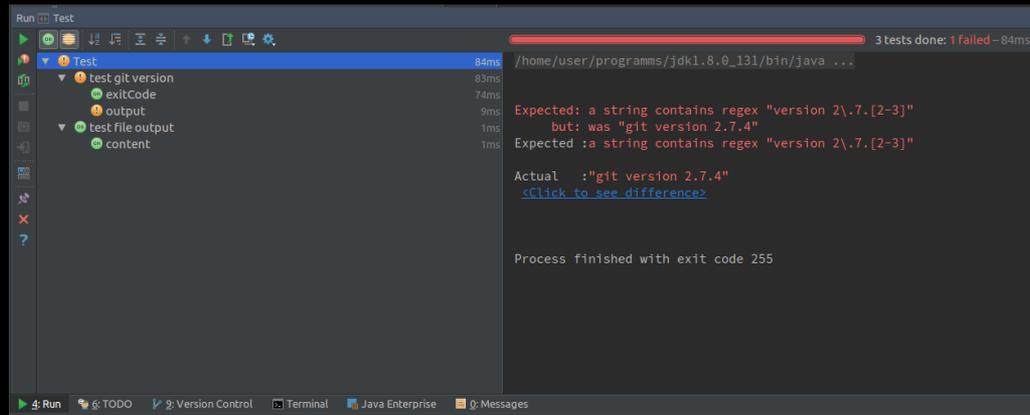
```
}
```

```
@Test("test file output")
```

```
file("./output") {
```

```
    !(content startsWith "not failed") endsWith "everybody"
```

```
}
```



# Основные проблемы в ходе решения задачи

---

- Довольно часто меняющиеся требования к фреймворку
- Библиотека для подключения по ssh
  - jsch. Последняя версия вроде работает)

# Thank you for your attention

---

private repository :(