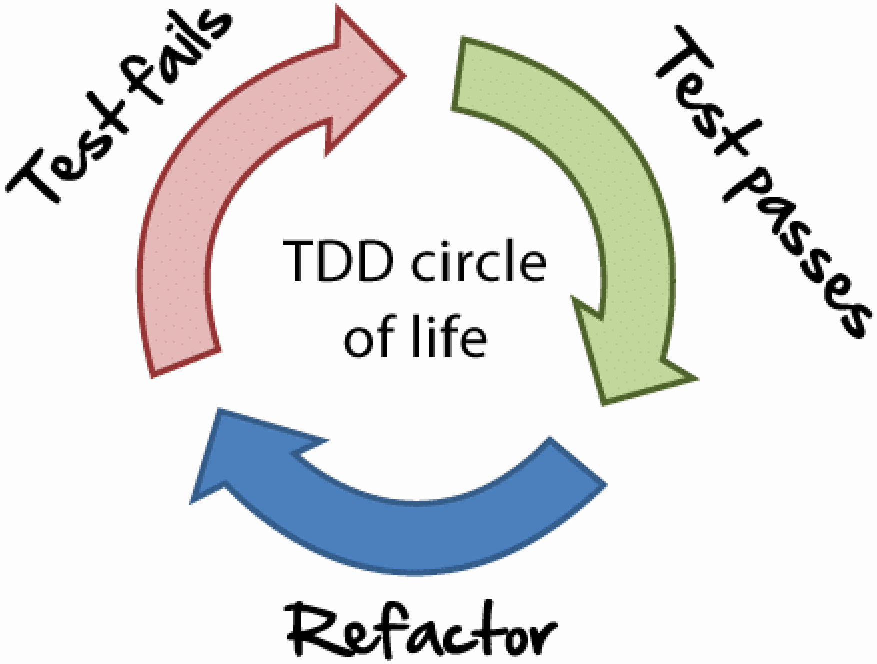
TDD for Infrastructure

Mario Fernandez

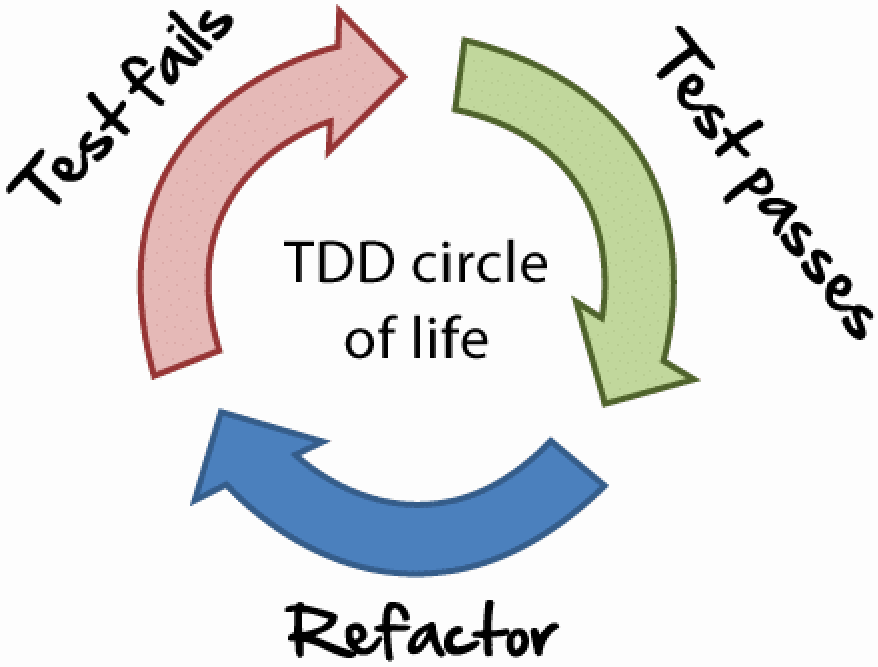
TDD

TDD: Test Driven Development

TDD in a nutshell



TDD in a nutshell

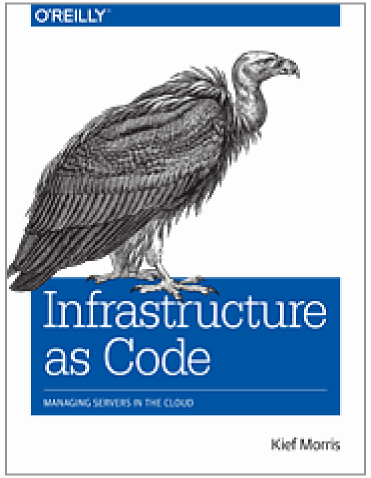


1. Write the only the code that is needed
2. Get quick feedback
3. Build a modular design

You knew that already

Infrastructure

Infrastructure as Code



Infrastructure as Code

IaC means writing code to manage configurations and automate provisioning

How do you test infrastructure?

How do you test infrastructure?

Enter **ServerSpec**

SERVERSPEC

How do you test infrastructure?

Enter ServerSpec

SERVERSPEC

ServerSpec is a tool to write tests for infrastructure based on RSpec

Show me the code

Straightforward configuration

```
spec/spec_helper.rb
```

```ruby
require 'serverspec'
require 'net/ssh'

set :backend, :ssh
host = ENV['TARGET_HOST']

options = Net::SSH::Config.for(host)
set :host,        options[:host_name] || host
set :ssh_options, options
```

Users & Groups

```
spec/hostname/users_spec.rb
```

```ruby
describe user('travis') do
  it { is_expected.to exist }
  it { is_expected.to belong_to_group 'docker' }
end

describe group('docker') do
  it { is_expected.to exist }
end
```

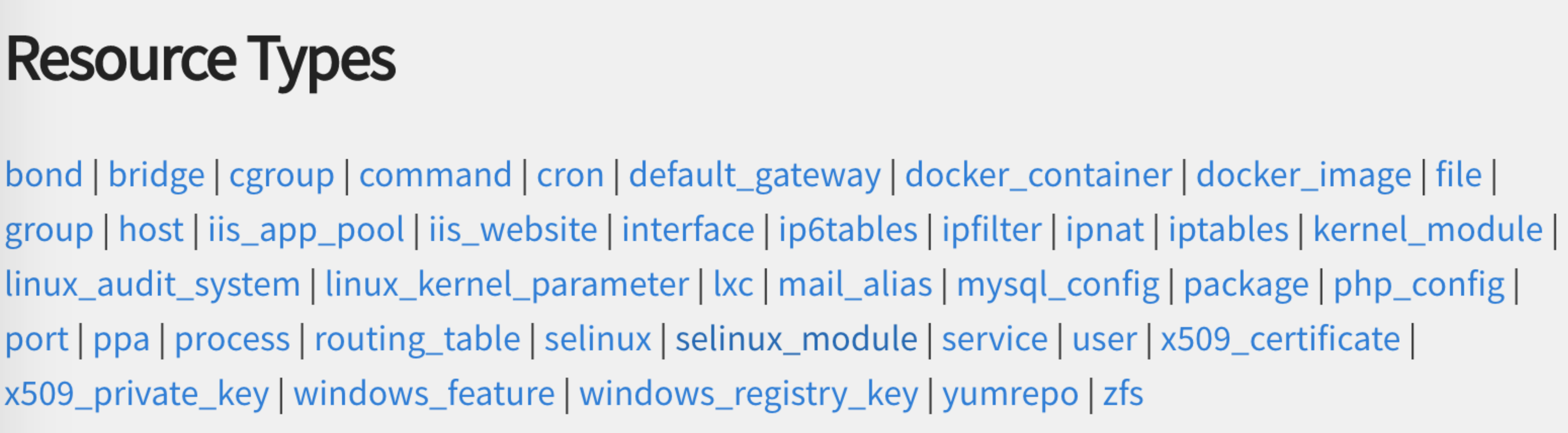
Files

```
spec/hostname/files_spec.rb
```

```ruby
describe file('/vault/file') do
  it { is_expected.to be_directory }
  it { is_expected.to be_owned_by 'vault' }
end

describe file('/bin/vault') do
  it { is_expected.to be_executable }
  it { is_expected.to be_owned_by 'root' }
end
```

And many others

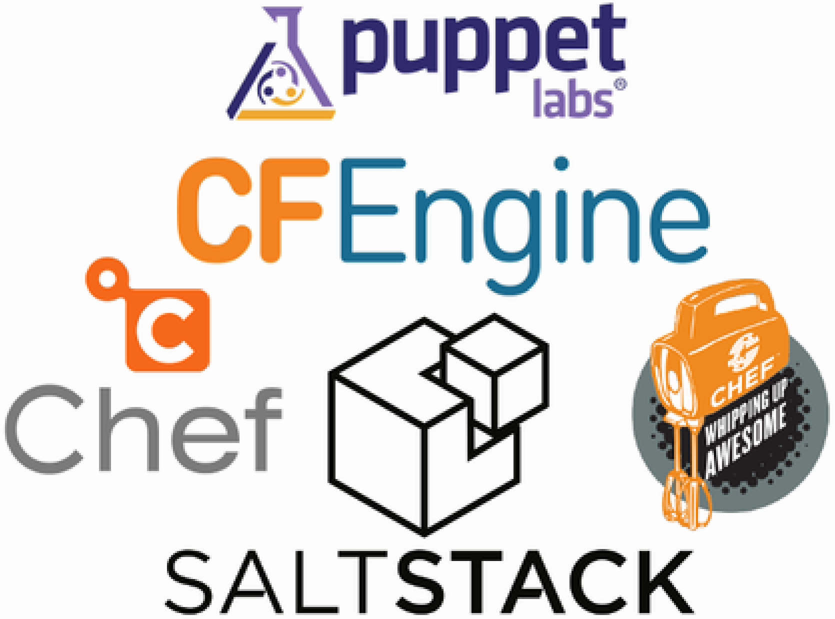


Resource Types

bond | bridge | cgroup | command | cron | default_gateway | docker_container | docker_image | file | group | host | iis_app_pool | iis_website | interface | ip6tables | ipfilter | ipnat | iptables | kernel_module | linux_audit_system | linux_kernel_parameter | lxc | mail_alias | mysql_config | package | php_config | port | ppa | process | routing_table | selinux | selinux_module | service | user | x509_certificate | x509_private_key | windows_feature | windows_registry_key | yumrepo | zfs

But wait ...

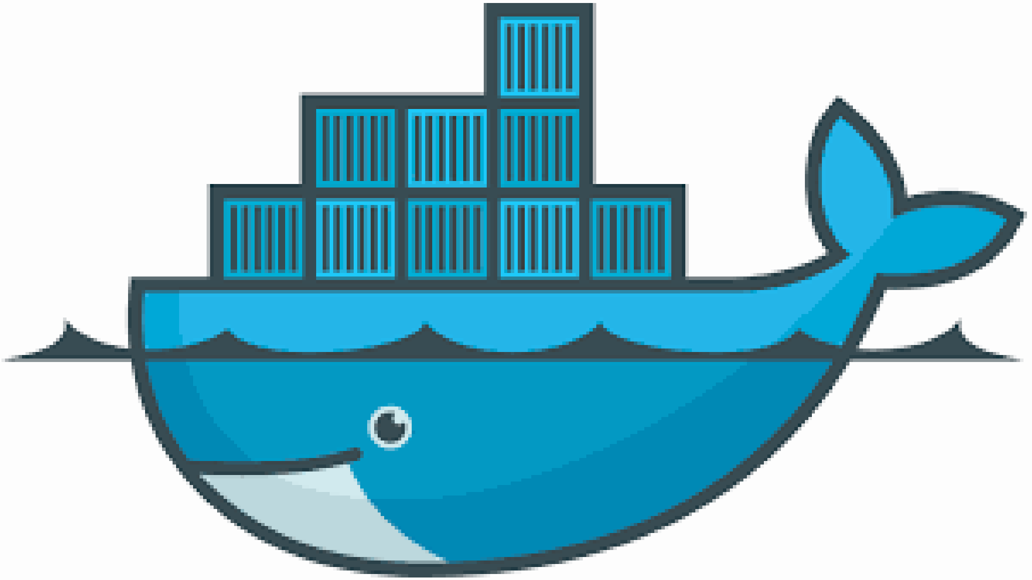
Isn't this covered by provisioning tools?



Isn't this covered by provisioning tools?

HashiCorp
Terraform

Containers



A sample Dockerfile

```
FROM java-openjdk-build-image as builder

WORKDIR /app
COPY . /app/

RUN gradle assemble

FROM java-openjdk-base-image

ENV PROFILE="in-memory"
COPY --from=builder /app/build/libs/*.jar .
EXPOSE 8080

CMD java -jar -Dspring.profiles.active=${PROFILE} app.jar
```

Basic packages

```
describe 'OpenJDK Image' do
  describe package(:openjdk) do
    it { is_expected.to be_installed }
  end

  describe file("/usr/bin/java") do
    it { is_expected.to be_executable }
  end

  describe command("java -version") do
    its(:stderr) { is_expected.to match(/8.151.12/) }
  end
end
```

Running Application

```
describe 'SpringBoot Applicationn' do
  describe file('/app/application.jar') do
    it { is_expected.to be_file }
  end

  describe process('java') do
    it { is_expected.to be_running }
    its(:args) { is_expected.to match(%r{/app/application.jar}) }
  end

  describe 'port of the app' do
    it { wait_for(port(8080)).to be_listening.with('tcp') }
  end
end
```

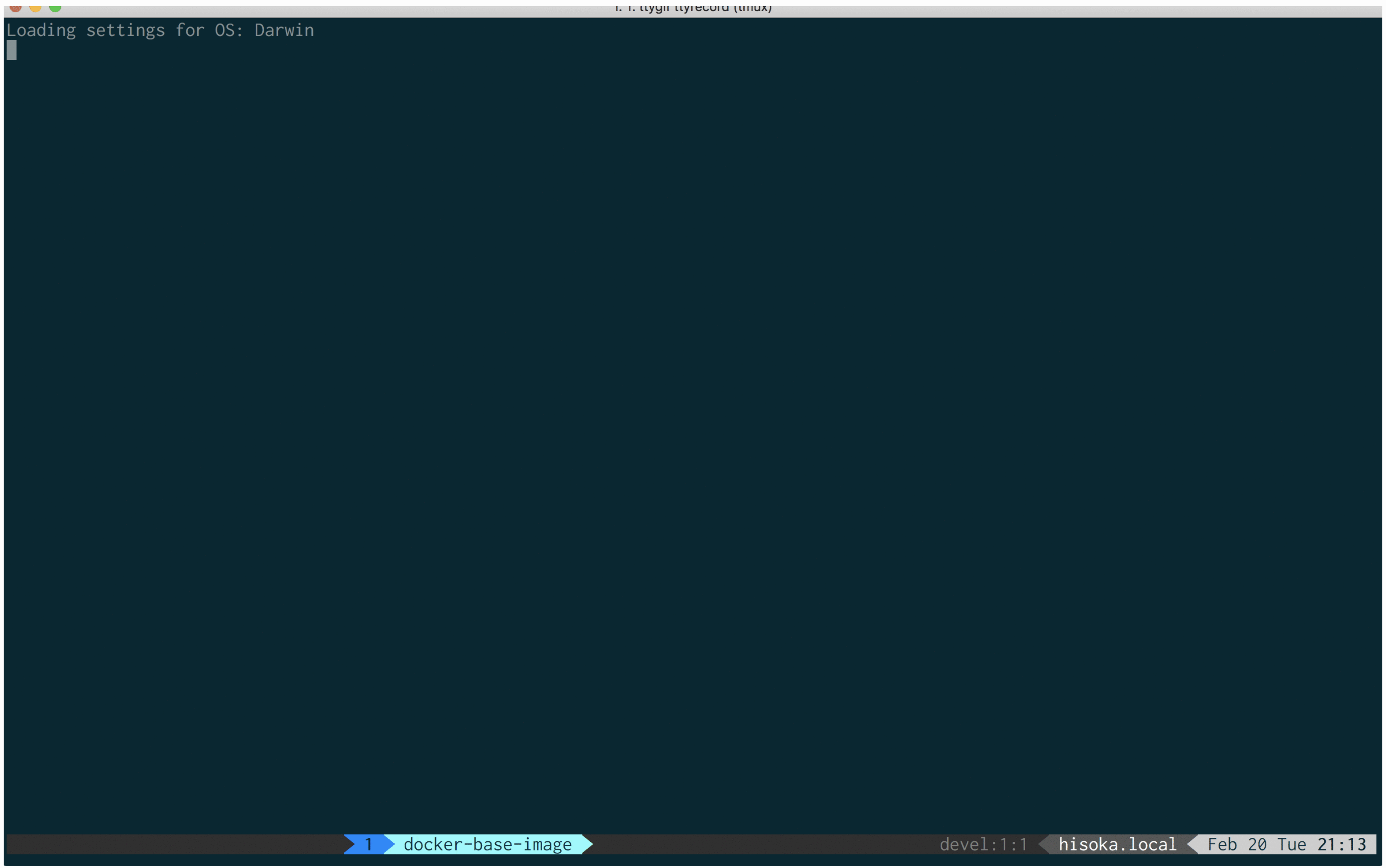
You can go crazy with this stuff

External dependencies via docker-compose

```
describe 'SpringBoot Application' do
  set :docker_container, 'app-springboot'

  before(:all) do
    compose.up(VAULT_CONTAINER_NAME, detached: true)
    vault.logical.write('secret/app/worspace', foo: :bar)
    compose.up('app-springboot', detached: true)
  end

  describe process('java') do
    it { is_expected.to be_running }
  end
end
```



```
Loading settings for OS: Darwin
```

```
 1  docker-base-image                     devel:1:1  hisoka.local  Feb 20 Tue 21:13
```

Summary

ServerSpec

ServerSpec is a good way to automate the process of testing infrastructure

Available backends

- local
- ssh
- Docker
- Even *Windows*

Links

- http://infrastructure-as-code.com/
- http://serverspec.org
- Alternative: https://github.com/aelsabbahy/goss

Thank you

Questions?