Infrastructure as Codeを学ぶ 実践的ハンズオン

~大規模インフラのノウハウと春の彩りを添えて~

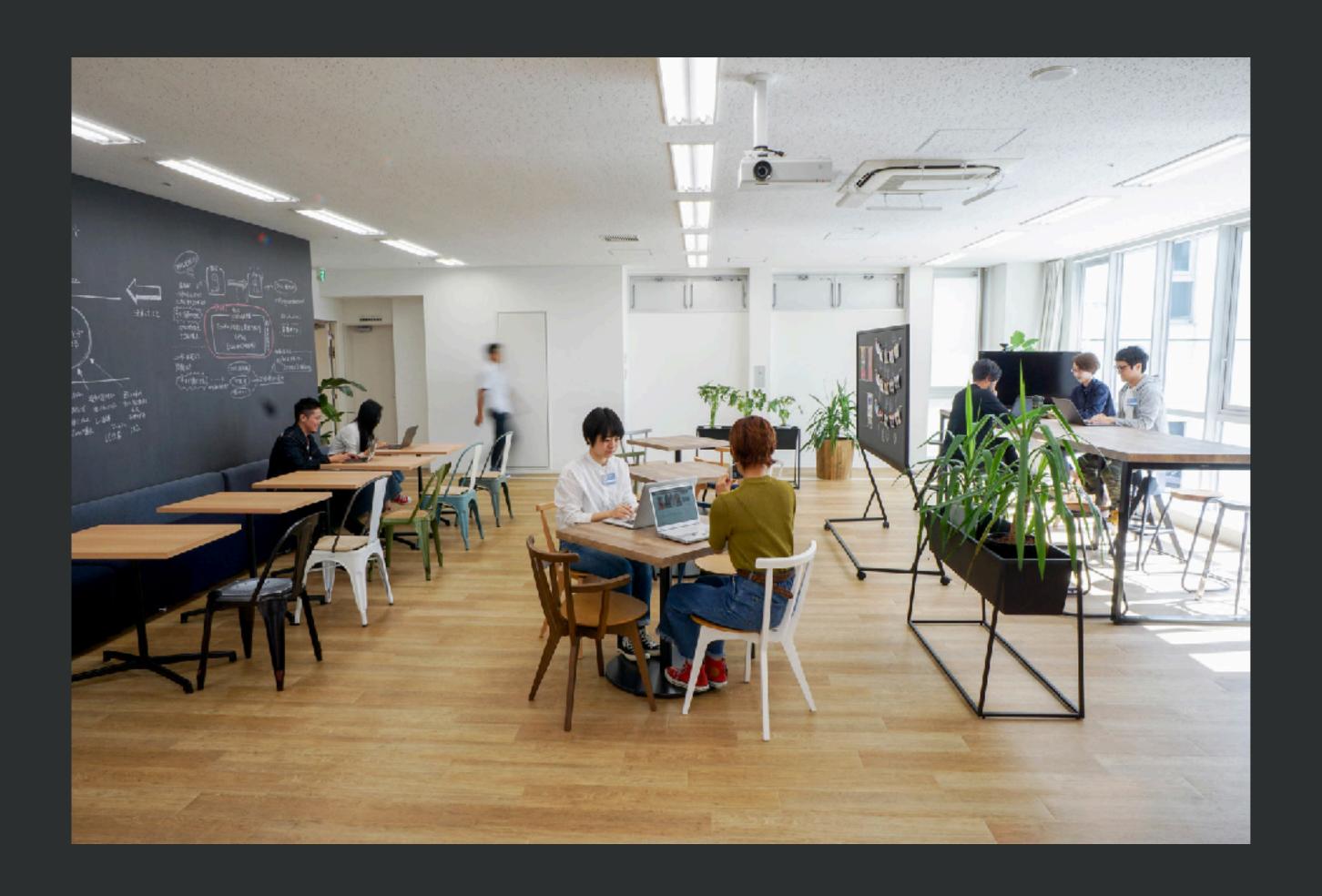
pyama86 / GMO Pepabo, Inc. 2018.04.08 インフラハンズオン



ホスティング事業部 チーフテクニカルリード **山下和彦** @pyama86



ペパボ福岡







Infrastructure as Code

Infrastructure as Code

- ・ソフトウェア開発のノウハウをインフラへ
- ・バージョン管理
- ・テスト駆動開発
- ・継続的インテグレーション
- ・継続的デリバリー



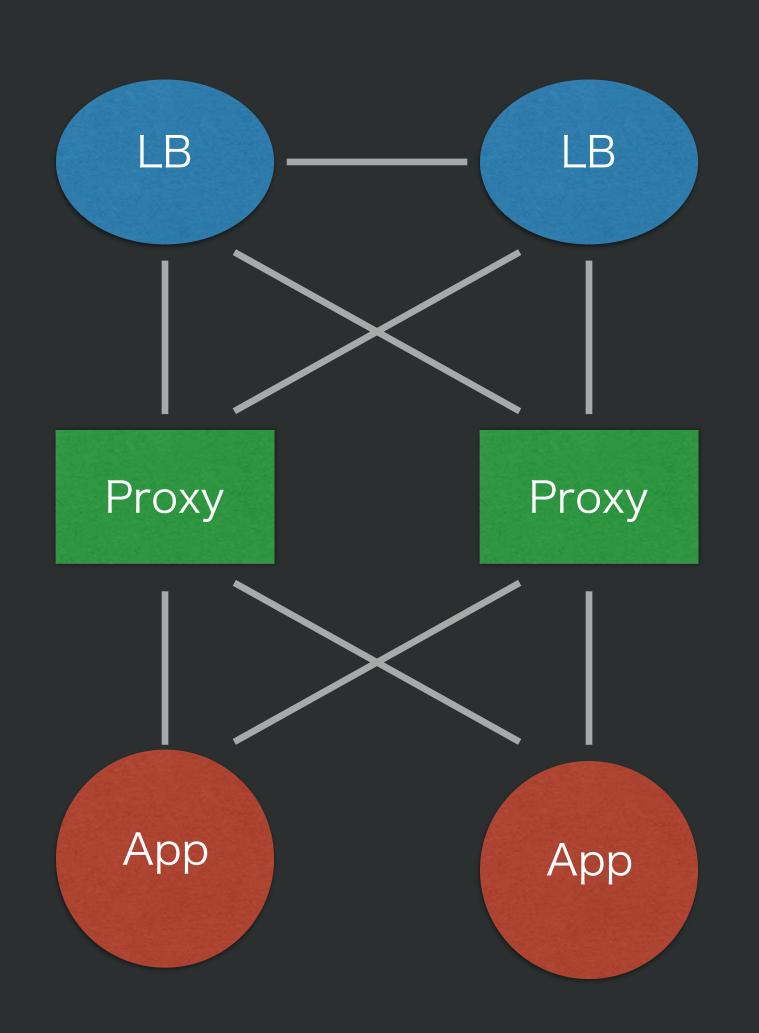
Infrastructure as Code

- ・ソフトウェア開発のノウハウをインフラへ
- ・バージョン管理
- ・テスト駆動開発
- ・継続的インテグレーション
- ・継続的デリバリー



実践的ハンズオン

ハンズオンのゴール



冗長化されたWebシステムを Infrastructure as Codeを体感しながら 開発する

ハンズオンで使用するツール・ミドルウェア

プロダクト名	役割
vagrant	VMマネージメント
itamae	プロビジョニング
Sererspec	テスト
keepalived	ロードバランシング
nginx	WEBプロキシ
httpd	アプリケーション

vagrant

- ・Rubyの記法でVirtualBoxなどのVMを管理できるソフトウェア
- · HashiCorp(https://www.hashicorp.com/)

https://www.vagrantup.com/



itamae

- ・Preferred Networks の @ryot_a_rai が作製したOSS
- · Light chef
- ・シンプルで使いやすく、mruby版はmitamae

https://github.com/itamae-kitchen/itamae



Serverspec

- ・ @mizzy が作製したOSS
- ・RSpecの記法でサーバ環境をテストできる

http://serverspec.org/



ハンズオン環境

·https://github.com/pepabo/infrasturucture_as_code_hands_on

```
— Gemfile // 利用gemの定義
— Gemfile lock // gemのバージョン固定ファイル
  README.md
 — Vagrantfile // VMの定義
— bootstrap.rb // Itamaeの実行スクリプト
  cookbooks // プロビジョニングスクリプトを配置するディレクトリ
         default.rb
      L— templates
```



ハンズオン環境

·https://github.com/pepabo/infrasturucture_as_code_hands_on

```
├── nodes // ロール毎のアトリビュートファイルを配置するディレクトリ
   — www.yaml
  roles // ロール毎のプロビジョニングスクリプトを配置するディレクトリ
   L--- WWW
      └─ default.rb
  spec // Serverpecのスクリプトを配置するディレクトリ
     - spec_helper.rb
         httpd_spec.rb
  vagrant_properties.yml // vagrantの構成ファイル
```



進め方

- ·WWロール作る
- ・PROXYロール作る
- ・LBロール作る



簡単!

WWWロールの要件

- ・phpinfoが表示できる
- ・PHP7が利用できる



TDD

・テストでまずあるべき状態を定義してから、開発を行う

```
$ vagrant up www-1
```

\$ bin/rake spec:www-1



apache2のインストール

- ・cookbooks/www/apache.rbの作成
- ・cookbooks/www/default.rbからapache.rbを読み込む
- ・roles/www/default.rbからwwwのcookbookを読み込む





package

パッケージのインストールを行うリソース

```
package <name> do
  action :install
  version 1.0
end
```

https://github.com/itamae-kitchen/itamae/wiki/package-resource



cookbooks/www/apache.rb

・レシピはミドルウェア単位で分割し、再利用性を高める

```
%w(
apache2
php7.0
libapache2-mod-php7.0
).each do |n|
package n
end
```

デフォルトアクションが:install なので省略可能



cookbooks/www/default.rbの定義

先ほど定義したレシピを読み込む

include_recipe 'apache.rb'

default.rbは複数のレシピを束ねる役割に使う



roles/www/default.rbの定義

先ほど定義したクックブックを読み込む

include_cookbook 'www'

このようにすると、一つのロールを複数の クックブックを組み合わせて構築できる



Let's Provision!!!]

\$ vagrant provision www-1



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /vor/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- norts_conf is always included from the main configuration file. It is used to determine the

http://www-1.hands-on.pbdev/

ubuntuのデフォルトページが 見えればOK

phpinfoを表示する

spec/www/phpinfo_spec.rb

```
describe file('/var/www/html/index.php') do
  it { should be_file }
  it { should be_mode 755 }
  it { should be_owned_by 'root' }
  it { should be_grouped_into 'root' }
end
```

http://serverspec.org/resource_types.html#file



phpinfoを表示する

· cookbooks/www/phpinfo.rb

```
remote_file '/var/www/html/index.php' do
  owner 'root'
  group 'root'
  mode '755'
end
```

https://github.com/itamae-kitchen/itamae/wiki/remote_file-resource



phpinfoを表示する

· cookbooks/www/files/index.php

```
<?php
echo phpinfo();</pre>
```

※一般的にはプロビジョニングツールでコンテンツ配置はやらない



cookbooks/www/default.rbの定義

先ほど定義したレシピを読み込む

```
include_recipe 'apache.rb'
include_recipe 'phpinfo.rb'
```

default.rbは複数のレシピを束ねる役割に使う



Let's Provision!!!]

\$ vagrant provision www-1



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /vor/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- norts_conf is always included from the main configuration file. It is used to determine the

http://www-1.hands-on.pbdev/

phpinfo見えない・・・

SShログイン

index.htmlを削除する必要がある



index.htmlの削除

spec/www/phpinfo_spec.rb

```
describe file('/var/www/html/index.html') do
  it { should_not exist }
end
```



index.htmlの削除

· cookbooks/www/phpinfo.rb

```
file '/var/www/html/index.html' do
  action :delete
end
```

https://github.com/itamae-kitchen/itamae/wiki/file-resource



Let's Provision!!!

- \$ vagrant provision www-1
- \$ bin/rake spec:www-1

PHP Version 7.0.28-0ubuntu0.16.04.1	php
System	Linux www-1 4.4-0-87-generic #110-Ubuntu SMF Tue Jul 18 12 95:35 UTC 2017 x86_64
Server API	Apache 20 Handler
Virtual Directory Support	disabled
Configuration File (php.int) Path	/etc/php//.0/apache2
Loaded Configuration File	/elo/php/x C/spsche2/php.m
Scan this dir for additional ini files	/sdx/php/7 C/speche?/conflid
Additional ini files parsed	/etc/php/7 0/epache2/conf d/10-opeache init /etc/php/7 0/eccene2/conf d/10-peache1/conf d/10-peache2/conf d/20 celender init /etc/php/7 0/epache2/conf d/20-objective2/conf d/20-
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	APG20161012NIS
PHP Extension Build	AP20151012/X1X
Debug Build	na na
Thread Safety	disahind
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zand Multibyte Support	beldalb
Pv6 Support	enabled
DTrace Support	azailadde, dicadded
Registered PHP Streams	https://fipe.compress.vills.php.file.glob.dute.http://fipe.pher
Registered Stream Socket Transports	tep, udp, unis, udg. sal, ris, riav1.0, risv1.1, riav1.2
Registered Stream Filters	zlib.t, string.rot 13, string toupper, string.tolower, string.strip_tags, convertit, consumed, dechunk, converticent/f

http://www-1.hands-on.pbdev/

YY!!!!



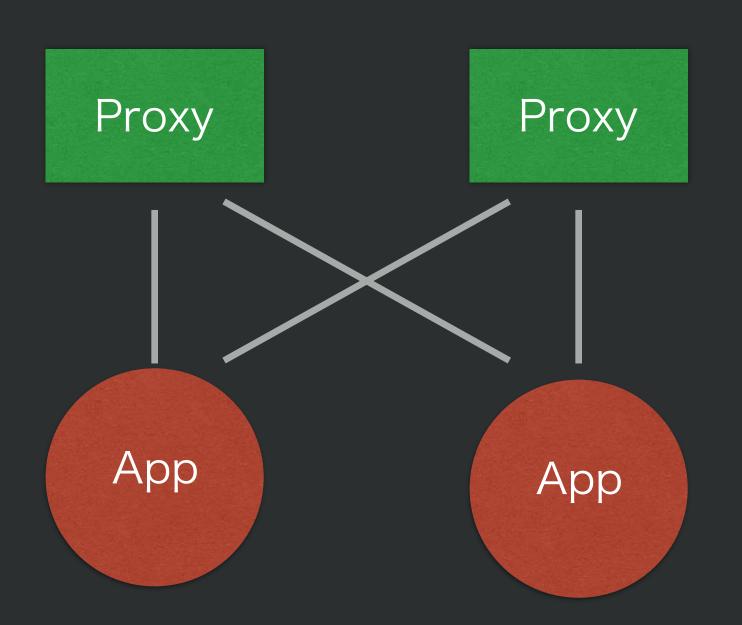
記法チェックとcommit

```
$ bin/rake rubocop -a
$ git add .
$ git commit -m "wwwの構築"
```



PROXYロールの要件

・nginxを利用したHTTPプロキシができる



nginxをインストールする

·spec/proxy/nginx_spec.rb

```
require 'spec_helper'
%W(
  nginx
).each do |n|
  describe package(n) do
    it { should be_installed }
  end
end
describe service('nginx') do
  it { should be_enabled }
  it { should be_running }
end
describe port(80) do
  it { should be_listening }
end
```



Cookbook、roleの雛形を作る

\$ bin/itamae generate cookbook proxy
\$ bin/itamae generate role proxy
\$ echo 'role: proxy' > nodes/proxy.yaml
\$ vagrant up proxy-1
\$ bin/rake spec:proxy-1



nginxのインストール

· cookbooks/proxy/nginx.rb

```
package 'nginx'
service 'nginx' do
  action %w(enable start)
end
```



nginxのインストール

· cookbooks/proxy/default.rb

```
include_recipe 'nginx.rb'
```

·roles/proxy/default.rb

```
include_cookbook 'proxy'
```



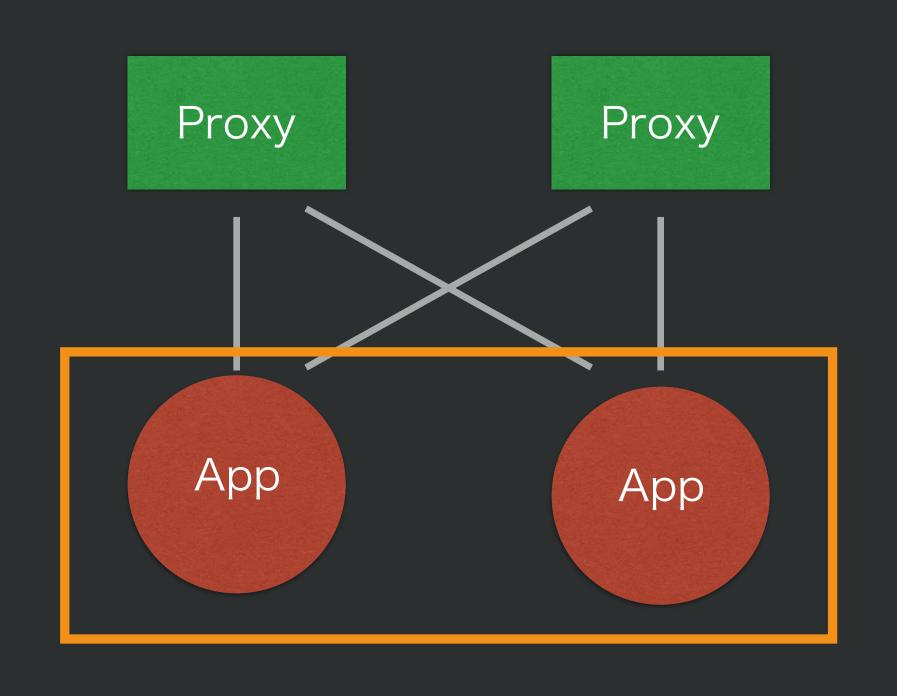
nginxの設定をする

・proxyサーバの中を覗く

```
% vagrant ssh proxy-1
ubuntu@proxy-1:~$ sudo su -
root@proxy-1:~# cd /etc/nginx/
root@proxy-1:/etc/nginx# ls -ltr
root@proxy-1:/etc/nginx# more nginx.conf
...
include /etc/nginx/conf.d/*.conf; # nginxのconfigにはinclude機構がある
...
```



upstreamの定義を行う



proxyサーバから見て、プロキシ先のサーバをupstreamと定義



upstreamの定義を行う

·spec/proxy/nginx_spec.rb

```
describe file('/etc/nginx/conf.d/www.conf') do
  its(:content) { should match /server 172.18.1.31/ }
  its(:content) { should match /server 172.18.1.32/ }
end
describe file('/etc/nginx/sites-enabled') do
 it { should_not exist }
end
describe file('/etc/nginx/sites-available') do
  it { should_not exist }
end
```



templateを利用する

```
template '/etc/nginx/conf.d/www.conf' do
  owner 'root'
  group 'root'
  notifies :restart, 'service[nginx]'
end
```

https://github.com/itamae-kitchen/itamae/wiki/template-resource



templateを利用する

cookbooks/proxy/templates/www.conf.erb

```
upstream app {
 <% node['proxy']['app']['servers'].each do |s| %>
  server <%= s %>;
  <% end %>
server {
  listen
           80;
  server_name localhost;
  location / {
    proxy_pass http://app/;
```



環境ごとのファイルはnodesで管理

nodes/proxy.yaml

```
proxy:
    app:
    servers:
        - 172.18.1.31
        - 172.18.1.32
```

複数環境の場合は、proxy-**production**.yamlや proxy-**development**.yamlなどを作成する



課題

/etc/nginx/sites-enabled /etc/nginx/sites-available 上記のディレクトリを

削除してください



正答例

```
%W(
  enabled
  available
each do |n|
  directory "/etc/nginx/sites-#{n}" do
    action :delete
    notifies :restart, 'service[nginx]'
  end
end
```



記法チェックとcommit

```
$ bin/rake rubocop -a
$ git add .
$ git status
$ git commit -m "proxyの構築"
```



課題

/etc/nginx/conf.d/www.conf

このレシピをtemplateの

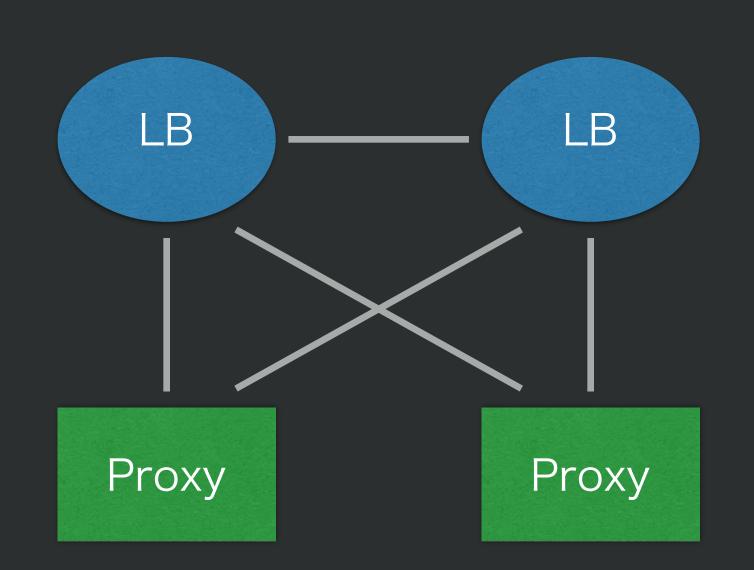
variablesを利用してより

再利用性を高める



LBロールの要件

- ・VIPを管理できる
- ・VIPで受けたトラフィックをProxyサーバに バランシングできる





VIPELA

keepalivedをインストールする

·spec/lb/keepalived_spec.rb

```
require 'spec_helper'
%W(
  keepalived
).each do |n|
  describe package(n) do
    it { should be_installed }
  end
end
describe service('keepalived') do
  it { should be_enabled }
  it { should be_running }
end
describe port(80) do
  it { should be_listening }
end
```



リアルサーバもテスト

·spec/lb/keepalived_spec.rb

```
describe file('/etc/keepalived/keepalived.conf') do
  its(:content) { should match /real_server 172.18.1.21 80/ }
  its(:content) { should match /real_server 172.18.1.22 80/ }
end
```



Cookbook、roleの雛形を作る

\$ bin/itamae generate cookbook lb
\$ bin/itamae generate role lb
\$ echo 'role: lb' > nodes/lb.yaml
\$ vagrant up lb-1
\$ bin/rake spec:lb-1



keepalivedのインストール

· cookbooks/lb/keepalived.rb

```
package 'keepalived'
service 'keepalived' do
 %w(enable start)
end
template '/etc/keepalived/keepalived.conf' do
  owner 'root'
  group 'root'
  notifies :restart, 'service[keepalived]'
end
```



keepalivedのインストール

· cookbooks/lb/default.rb

```
include_recipe 'keepalived.rb'
```

roles/lb/default.rb

```
include_cookbook 'lb'
```



VIPの定義

· cookbooks/lb/templates/keepalived.conf.erb

```
vrrp_instance vrrp_int {
 interface <%= node['lb']['keepalived']['if'] %>
  virtual_router_id <%= node['lb']['keepalived']['router_id'] %>
  nopreempt
  state BACKUP
  priority 100
  advert_int 3
  garp_master_delay 5
  authentication {
   auth_type PASS
   auth_pass hands_on
  virtual_ipaddress {
   <%= node['lb']['keepalived']['vip'] %>
```



VIPの定義

· cookbooks/lb/templates/keepalived.conf.erb

```
virtual_server <%= node['lb']['keepalived']['vip'] %> 80 {
  delay_loop 10
  lvs_sched lc
  lvs_method NAT
  protocol TCP
  <% node['lb']['keepalived']['servers'].each do |s| %>
   real_server <%= s %> 80 {
   weight 1
   TCP_CHECK {
     connect_port 80
     connect_timeout 30
 <% end %>
```

YAMLでアトリビュートを定義する

· nodes/lb.yaml

```
lb:
  keepalived:
    vip: 172.18.1.10
    router_id: 100
    if: enp0s8
    servers:
      - 172.18.1.21
      - 172.18.1.22
```



Let's Provision!!!1

- \$ vagrant provision lb-1
- \$ bin/rake spec:lb-1



サーバの中を見てみましょう

ipvsの状態を見る

```
# vipを保持しているか

$ ip a
# real serverの状態を見る

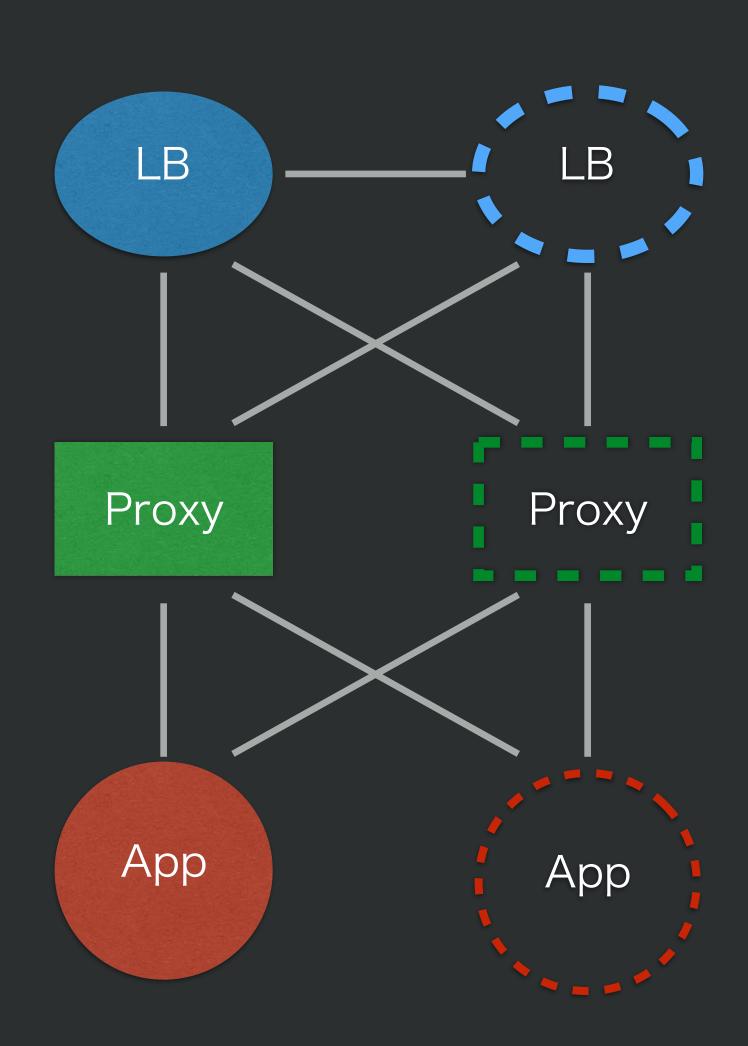
$ ipvsadm -L -n
```



記法チェックとcommit

- \$ bin/rake rubocop -a
- \$ git commit -m "lbの構築"

待機系を起動する



\$ vagrant up



ipvsの状態を見る

```
# real serverの状態を見る
$ ipvsadm -L -n
```



おもむるにサーバを落とす

- \$ vagrant halt www-1
- \$ vagrant halt proxy-1
- \$ vagrant halt lb-1

サービスが無停止であること



今日学んだこと

- ・Infrastructure as Codeはソフトウェア開発のノウハウをインフラの 世界へ持ってきたもの
- ・インフラにおいてもテスト駆動開発
- ・再利用しやすい粒度でレシピを管理
- ・インフラは楽しい!!!1

