Usecase examples of Packer

the best tools for os image creation

GMOペパボ株式会社



self.introduce



=>
{
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  title: “Chief engineer at GMO Pepabo, Inc.”,
  commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “tdiary”, “hiki”, “railsgirls”, “railsgirls-jp”, “jenkins”],
  sites: [“ruby-lang.org”, “rubyci.com”, “railsgirls.com”, “railsgirls.jp”],
}

Do scale-out

Our service issue

Do scale-out

Do scale-out with automation!

Do scale-out with rapid automation!!!

Do scale-out with extremely rapid automation!!!!!1

Concerns of bootstrap time

Typical scenario of server set-up for scale out.

- OS boot
- OS Configuration
- Provisioning with puppet/chef
- Setting up to capistrano
- Deploy rails application
- Added load balancer (= Service in)

Concerns of bootstrap time

Slow operation

- OS boot
- Provisioning with puppet/chef
- Deploy rails application

Fast operation

- OS Configuration
- Setting up to capistrano
- Added load balancer (= Service in)

Check point of Image creation

Slow operation

- OS boot

Step1

- Provisioning with puppet/chef

Step2

- Deploy rails application

Fast operation

- OS Configuration
- Setting up to capistrano
- Added load balancer (= Service in)

2 phase strategy

- Official OS image
  - Provided from platform like AWS, Azure, GCP, OpenStack…
- Minimal image(phase 1)
  - Network, User, Package configuration
  - Installed puppet/chef and platform cli-tools.
- Role specified(phase 2)
  - Only boot OS and Rails application

Tuning tools(cloud-init)

We only use OS configuration. Do not use “run_cmd”

```
#cloud-config
repo_update: true
repo_upgrade: none

packages:
 - git
 - curl
 - unzip

users:
  - default

locale: ja_JP.UTF-8
timezone: Asia/Tokyo
```

Before packer age

We use IaaS API for image creation with cloud-init userdata.

```
puppet agent -t

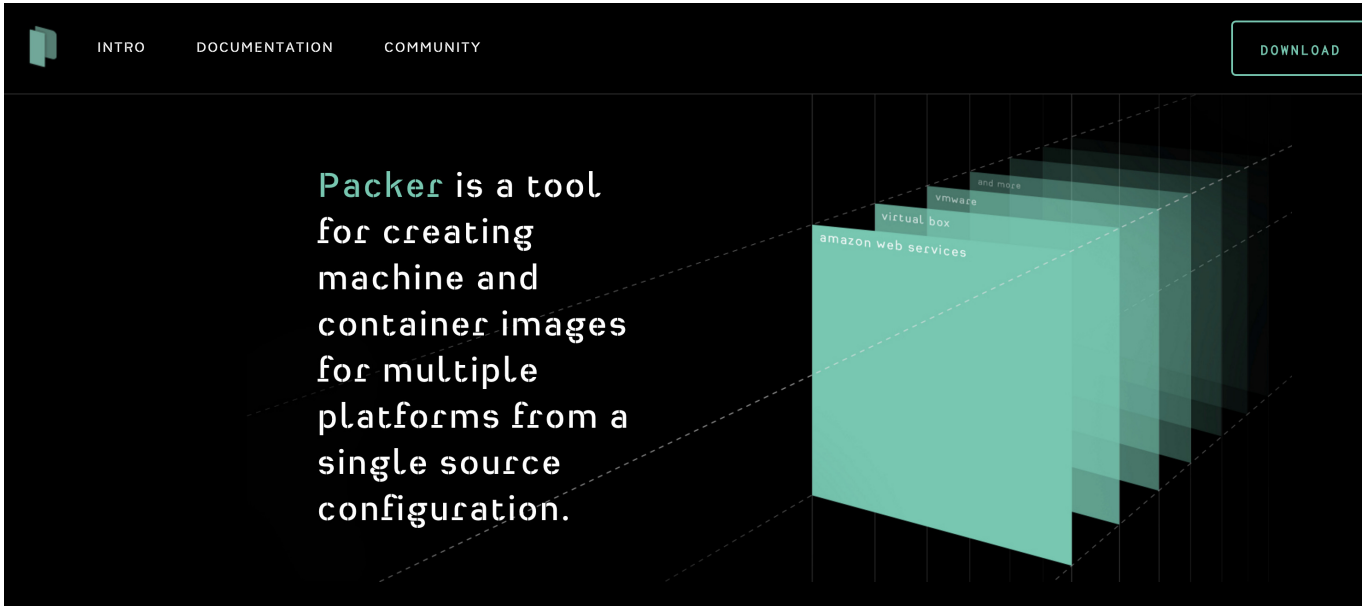
rm -rf /var/lib/cloud/sem /var/lib/cloud/instances/*

aws ec2 create-image --instance-id `cat /var/lib/cloud/data/instance-id` --name
www_base_`date +%Y%m%d%H%M`
```

We can create OS Image using cloud-init and provisioned puppet when boot time of instance.

After packer age

I couldn’t understand use-case of packer. Is it Provision tool? Deployment tool?



We created cli tool with thor

We can run packer over thor code with advanced options.

```
module SomeCliTool
  class Ami < Thor
    method_option :ami_id,        type: :string,  aliases: "-a"
    method_option :init,          type: :boolean
    desc 'build-www', 'wwwの最新イメージをビルドします'
    def build_www
      …
    end
  end
end
```

```
$ some_cli_tool ami build-minimal
$ some_cli_tool ami build-www
$ some_cli_tool ami build-www —init
$ some_cli_tool ami build-www -a ami-id
```

inside image creation with Packer

- Packer configuration
  - JSON format
  - select instance size, block volume,
- cloud-init
  - Basic configuration of OS
  - only default module of cloud-init
- provisioner
  - shell script :)

minimal image

cloud-init

```
#cloud-config
repo_update: true
repo_upgrade: none

packages:
 - git
 - curl
 - unzip

users:
  - default

locale: ja_JP.UTF-8
timezone: Asia/Tokyo
```

provisioner

```
rpm -ivh http://yum.puppetlabs.com/
puppetlabs-release-el-7.noarch.rpm

yum -y update
yum -y install puppet
yum -y install python-pip
pip install awscli

sed -i 's/name: centos/name: cloud-user/' /etc/
cloud/cloud.cfg
echo 'preserve_hostname: true' >> /etc/cloud/
cloud.cfg
```

www image

cloud-init

```
#cloud-config
preserve_hostname: false
```

provisioner

```
puppet agent -t
set -e
monit stop unicorn
/usr/local/bin/globefish -w
rm -rf /var/www/deploys/minne/releases/*
rm -f /var/www/deploys/minne/current

# tar xf するだけで動くRails アプリケーションを取得

(snip)

# mackerel のホスト設定が packer 実行時のものとかぶらないように初期化

rm /var/lib/mackerel-agent/id
# cloud-init をもう一度動かすようにする準備

rm -rf /var/lib/cloud/sem /var/lib/cloud/instances/*
```

scale out with cli command

We can scale out with one command via our cli tool

```
$ some_cli_tool instances launch -c 10 …
```

One more thing…

```
$ some_cli_tool mackerel fixrole
$ some_cli_tool scale up
$ some_cli_tool deploy blue-green
```

Integration tests with Packer

We can tests results of Packer running. (Impl by @udzura)

packer configuration

```
 "provisioners": [
(snip)
 {
    "type": "shell",
    "script": "{{user `project_root`}}packer/minimal/provisioners/run-serverspec.sh",
    "execute_command": "{{ .Vars }} sudo -E sh '{{ .Path }}'"
  }
]
```

run-serverspec.sh

```
yum -y -q install rubygem-bundler
cd /tmp/serverspec
bundle install --path vendor/bundle
bundle exec rake spec
```

Next step of image creation with packer

- Automated all of test with image creation and launching
- Fail over with fragile API status
- Sync deployment with image creation

Appendix

Working

- vagrant + openstack plugin
- packer + openstack builder
- consul + consul-alerts
- nginx + consul-templates

Evaluating

- vault
- terraform

エンジニア

技術基盤エンジニア

職種名	勤務地
技術基盤整備エンジニア (正社員)	東京

WEBアプリケーションエンジニア

モバイルアプリケーションエンジニア

インフラエンジニア

情報システムエンジニア



ペパランチ



http://pepabo.com/recruit/career/

To be continued

I will full details of this talk at YAPC::Asia 2015



Talk Information



SHIBATA Hiroshi

github

Advanced technic for OS upgrading in 3 minutes Accepted

63 ツイート | 14 Like | 0 +1 | イイネ! | 56 Bookmark