

Сборка образов VM при помощи Packery



Проект *infra* и проверка ДЗ

- Создайте новую ветку в вашей репозитории для выполнения данного ДЗ и назовите ветку **packer-base**.
- Перенесите наработки с предыдущего ДЗ в директорию **config-scripts**.
- Проверка данного ДЗ производится через Approve Pull Request'a одним из преподавателей
- После того, как **любой** из преподавателей сделает approve Pull Request'a, ветку с ДЗ можно смерджить.

Установка Packer

Скачайте версию Packer для вашей ОС, перейдя по [ссылке](#).

Распакуйте скачанный zip архив и поместите бинарный файл в директорию, путь до которой содержится в переменной окружения PATH.

Проверить установку Packer можно командой:

```
$ packer -v
```

Credentials

Для управления ресурсами GCP через сторонние приложения, такие как Packer и Terraform, нам нужно предоставить этим инструментам информацию (credentials) для аутентификации и управлению ресурсами GCP нашего аккаунта.

Application Default Credentials (ADC)

Установка ADC позволяет приложениям, работающим с GCP ресурсами и использующим Google API библиотеки, управлять ресурсами GCP через авторизованные API вызовы, используя credentials вашего пользователя.

Создайте ADC:

```
$ gcloud auth application-default login
```

Создаем Packer template

Создайте в **infra** репозитории директорию **packer**.

Внутри директории **packer** создайте файл **ubuntu16.json**

Это и будет наш Packer шаблон, содержащий описание образа VM, который мы хотим создать.

Для нашего тестового приложения мы соберем образ VM с предустановленными Ruby и MongoDB, так называемый baked-образ.

Packer builder

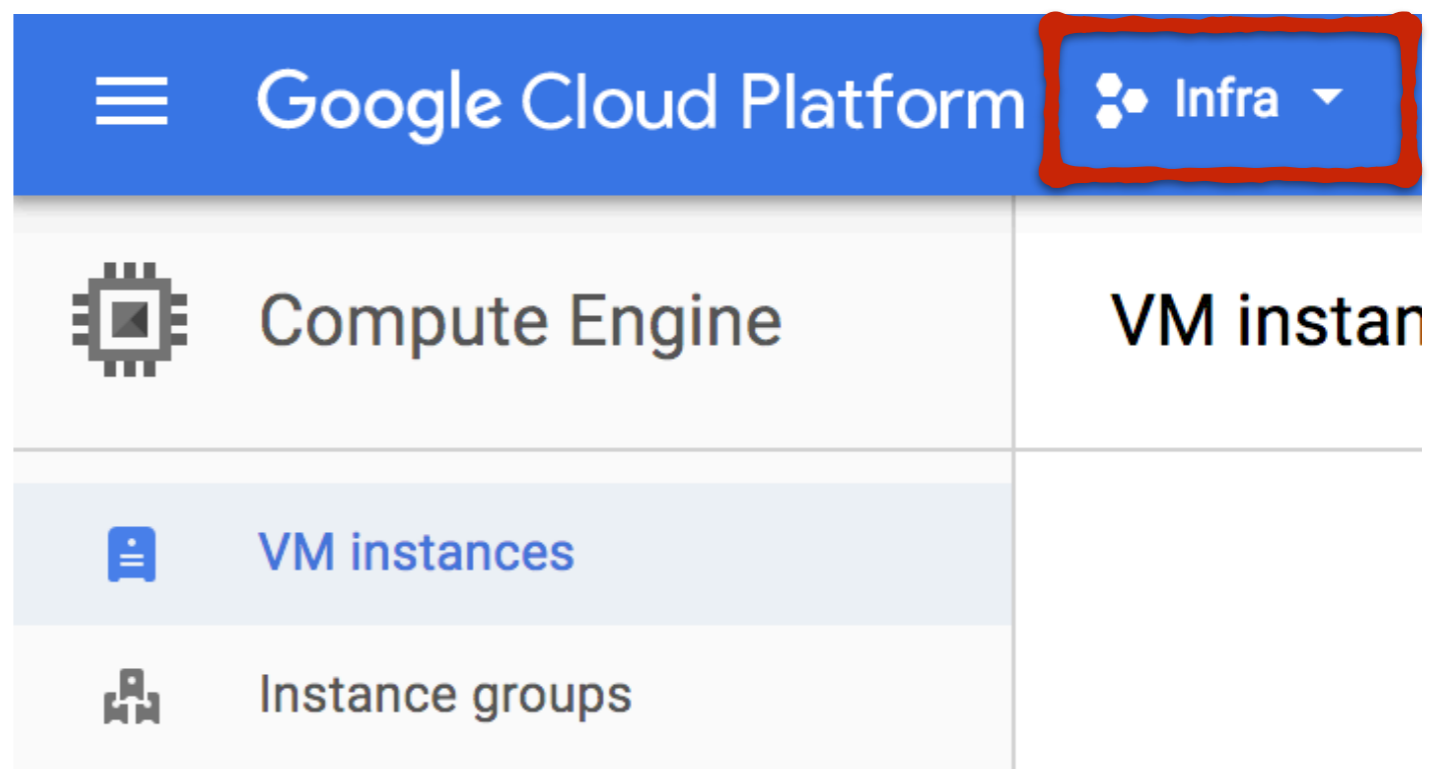
Определим Packer builders в шаблоне
ubuntu16.json ([gist](#)):

Опции, которые нужно
заменить (см. след слайд)

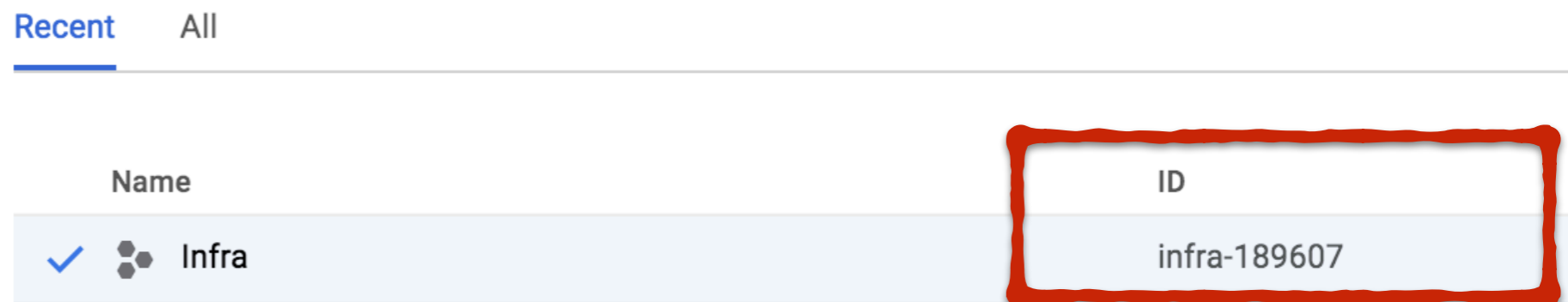
```
{
  "builders": [
    {
      "type": "googlecompute",
      "project_id": "infra-189607",
      "image_name": "reddit-base-{{timestamp}}",
      "image_family": "reddit-base",
      "source_image_family": "ubuntu-1604-lts",
      "zone": "europe-west1-b",
      "ssh_username": "appuser",
      "machine_type": "f1-micro"
    }
  ]
}
```



Чтобы найти ID вашего проекта, нажмите в консоли на имя проекта



Скопируйте ID и вставьте в ваш Rascker шаблон



Packer builder

Или воспользуйтесь следующей командой в командной строке

```
$ gcloud projects list
```

PROJECT_ID	NAME	PROJECT_NUMBER
<code>infra-189607</code>	Infra	529004887562

Пояснения Packer builder

Разберем основные настройки секции builder нашего шаблона:

- **type: "googlecompute"** - что будет создавать виртуальную машину для билда образа (в нашем случае Google Compute Engine)
- **project_id: "infra-189607"** - id вашего проекта
- **image_family: "reddit-base"** - семейство образов к которому будет принадлежать новый образ
- **image_name: "reddit-base-{{timestamp}}"** - имя создаваемого образа

Пояснения Packer builder

Продолжение:

- **source_image_family: "ubuntu-1604-lts"** - что взять за базовый образ для нашего билда
- **zone: "europe-west1-b"** - зона, в которой запускать VM для билда образа
- **ssh_username: "appuser"** - временный пользователь, который будет создан для подключения к VM во время билда и выполнения команд провижинера (о нем поговорим ниже)
- **machine_type: "f1-micro"** - тип инстанса, который запускается для билда

Provisioners

Если **builders** секция отвечает за создание виртуальной машины для билда и создание машинного образа в GCP, то секция **provisioners** позволяет устанавливать нужное ПО, производить настройки системы и конфигурацию приложений на созданной VM.

Используя скрипты для установки Ruby и MongoDB из предыдущего ДЗ, определим два провижинера.

Provisioners

Используем shell provisioner, который позволяет запускать `bash` команды на запущенном инстансе.

После секции “builders” определим провижинер внутри нашего шаблона для установки Ruby (gist).

```
"provisioners": [  
  {  
    "type": "shell",  
    "script": "scripts/install_ruby.sh",  
    "execute_command": "sudo {{.Path}}"  
  }  
]
```

Добавим провижинер для установки MongoDB ([gist](#)).

```
"provisioners": [  
  {  
    "type": "shell",  
    "script": "scripts/install_ruby.sh",  
    "execute_command": "sudo {{.Path}}"  
  },  
  {  
    "type": "shell",  
    "script": "scripts/install_mongodb.sh",  
    "execute_command": "sudo {{.Path}}"  
  }  
]
```

Опция `execute_command` позволяет указать, каким способом будет запускаться скрипт. Т.к. команды по установке требуют `sudo`, то мы указываем, что запускать скрипт следует с `sudo`. Из скриптов **`install_mongodb.sh`** и **`install_ruby.sh`** `sudo` можно будет удалить.

В итоге наш packer шаблон должен выглядеть примерно вот так ([gist](#)):

```
{
  "builders": [
    {
      "type": "googlecompute",
      "project_id": "infra-189607",
      "image_name": "reddit-base-{{timestamp}}",
      "image_family": "reddit-base",
      "source_image_family": "ubuntu-1604-lts",
      "zone": "europe-west1-b",
      "ssh_username": "appuser",
      "machine_type": "f1-micro"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "script": "scripts/install_ruby.sh",
      "execute_command": "sudo {{.Path}}"
    },
    {
      "type": "shell",
      "script": "scripts/install_mongodb.sh",
      "execute_command": "sudo {{.Path}}"
    }
  ]
}
```

Скрипты для провижининга

Внутри директории **packer** создайте директорию **scripts** для скриптов, которые будут использовать провижинерами. Скопируйте в эту директорию скрипты **install_ruby.sh** и **install_mongodb.sh** из предыдущего ДЗ.

Скрипты [install_ruby.sh](#) и [install_mongodb.sh](#) также доступны для скачивания (нажмите на имена скриптов, чтобы скачать). Убедитесь, что ваши скрипты не имеют существенных отличий с теми, что доступны по ссылкам. Поправьте ваши скрипты при необходимости.

Проверка на ошибки

Проверьте, не допустили ли вы ошибок при создании шаблона, используя команду `packer validate`:

```
$ packer validate ./ubuntu16.json
```

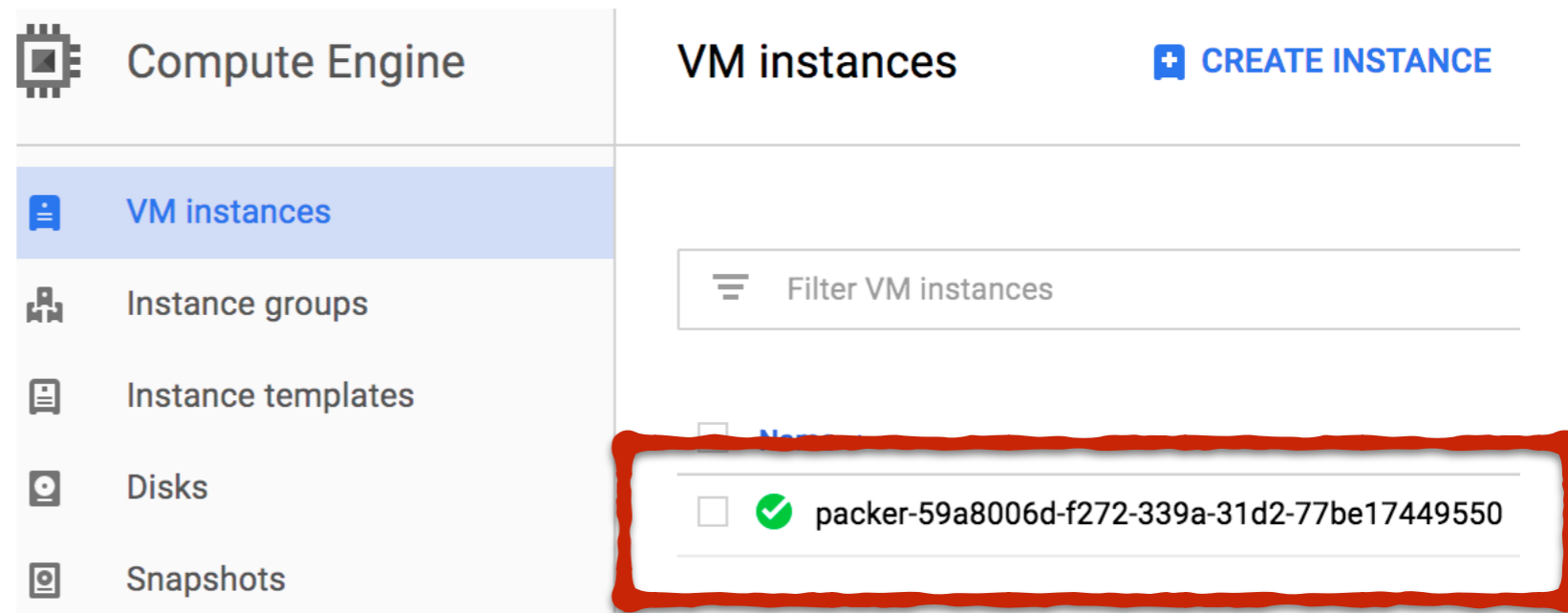
Поправьте ошибки, если они есть.

Packer build

Если проверка на ошибки прошла успешно, то запустите build образа:

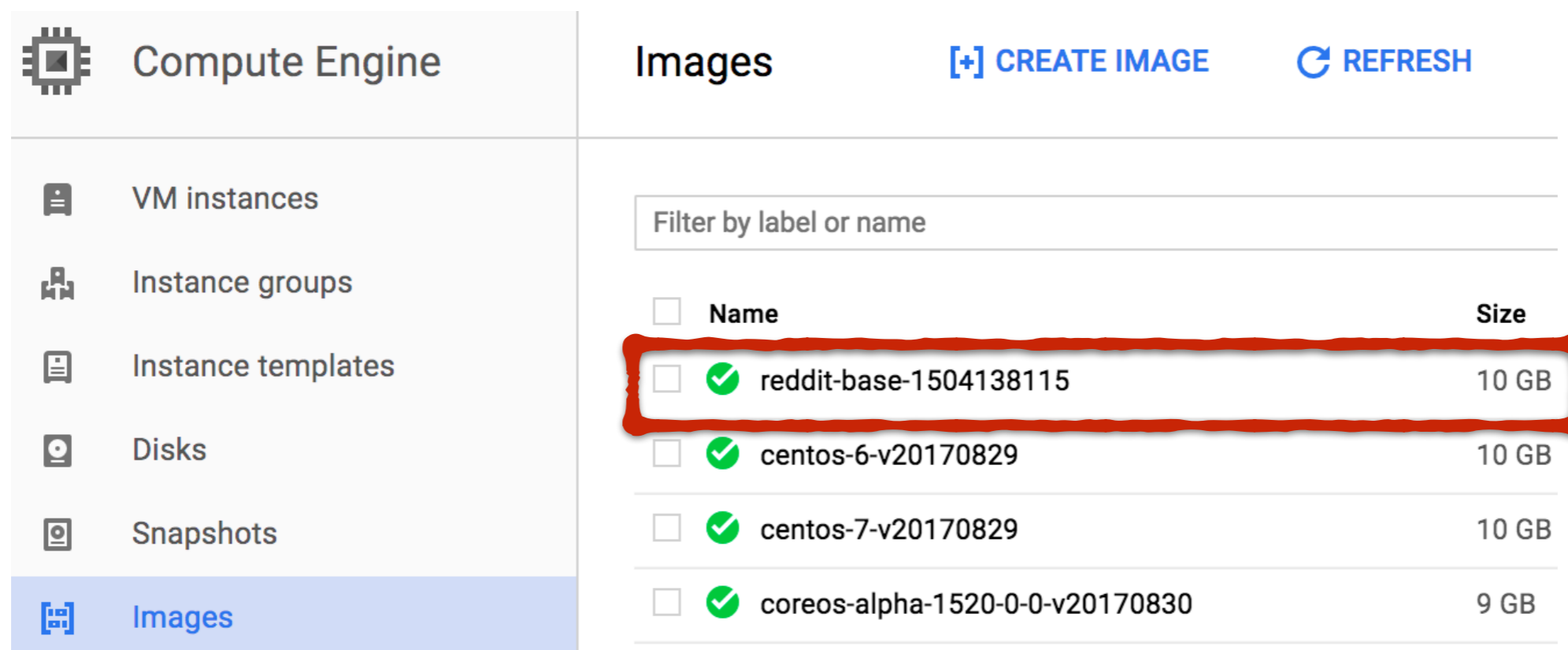
```
$ packer build ubuntu16.json
```

В браузерной консоли можно увидеть, как Packer запустил инстанс VM.



Проверяем созданный образ

В браузерной консоли перейдите по пути Compute Engine -> Images. Найдите свой образ.



Compute Engine

Images [\[+\] CREATE IMAGE](#) [REFRESH](#)

Filter by label or name

<input type="checkbox"/>	Name	Size
<input type="checkbox"/>	✓ reddit-base-1504138115	10 GB
<input type="checkbox"/>	✓ centos-6-v20170829	10 GB
<input type="checkbox"/>	✓ centos-7-v20170829	10 GB
<input type="checkbox"/>	✓ coreos-alpha-1520-0-0-v20170830	9 GB

Деплоим приложение

Как и в прошлый раз, задеплоим наше тестовое приложение.


Но на этот раз нам нужно будет проделать меньше работы, т.к. часть пакетов уже содержится в образе VM, который мы создали.

Задаем нужные
характеристики
машины (тип инстанса
не больше g1-small)

Name ?
reddit-app

Zone ?
europe-west1-b

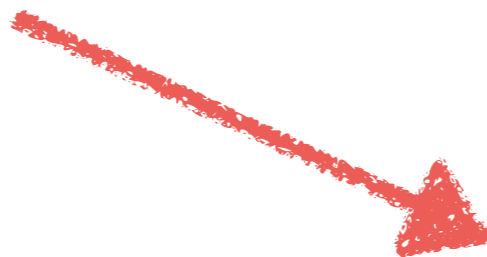
Machine type
small (1 shared... 1.7 GB memory [Customize](#)
[Upgrade your account](#) to create instances with up to 64 cores

Boot disk ?
 New 10 GB standard persistent disk
Image
Debian GNU/Linux 9 (stretch) [Change](#)

Нажимаем изменить образ



Жмем custom images и
выбираем созданный
нами образ



Boot disk

Select an image or snapshot to create a boot disk; or attach an ex

OS images Application images **Custom images** Snaps

Show images from

My Project 4

- reddit-base-1504138115
Ubuntu 16.04 for reddit app. Has mongodb and some gems preinstal
Created from My Project 4 on Aug 31, 2017, 3:10:56 AM

Подключение по SSH

После того как инстанс запустился, вам необходимо подключиться к инстансу по SSH, используя ключи пользователя `appuser`, которые вы сгенерировали на прошлом занятии.

```
$ ssh appuser@<instance_public_ip>
```

Установка зависимостей и запуск приложения

Для деплоя приложения можно использовать созданный вами скрипт `deploy.sh` или перечисленные ниже команды:

```
$ git clone -b monolith https://github.com/express42/reddit.git  
$ cd reddit && bundle install  
$ puma -d
```

Проверяем, что сервер приложения запустился:

```
$ ps aux | grep puma
```

Проверка работы приложения

Предварительно убедитесь, что вам доступен порт сервера приложения в правилах firewall

146.148.56.222:9292

Monolith Reddit

User created

Menu

[All posts](#)

[New post](#)

Закоммитьте результаты

В случае успешного создания образа VM при помощи Packer, закоммитьте результаты вашей работы в созданную ранее ветку.

Самостоятельные задания

1. Необходимо параметризовать созданный шаблон, используя пользовательские переменные (см. лекцию и [документацию](#)).

Должны быть параметризованы:

- ID проекта (обязательно)
- source_image_family (обязательно)
- machine_type

"**Обязательно**" означает, что пользовательская переменная должна быть обязательна для определения и не иметь значения по умолчанию.

2. Пользовательские переменные **определяются в самом шаблоне**, в файле **variables.json** задаются обязательные переменные, либо переопределяются.

Самостоятельные задания

3. Исследовать другие опции builder для GCP ([ссылка](#)).

Какие опции точно хотелось бы увидеть:

- Описание образа
- Размер и тип диска
- Название сети
- Теги

Файл с переменными **variables.json**, нужно внести в **.gitignore**, а в репозиторий добавить файл **variables.json.example** с примером заполнения, используя вымышленные значения, т.к. ваш репозиторий публичный.



Задание со *

Чтобы попрактиковать подход к управлению инфраструктурой Immutable infrastructure, о котором говорили на вебинаре, попробуйте “запечь” (bake) в образ VM все зависимости приложения и сам код приложения. Результат должен быть таким: запускаем инстанс из созданного образа и на нем сразу же имеем запущенное приложение.

Созданный шаблон должен называться **immutable.json** и содержаться в директории **packer**, image_family у получившегося образа должен быть **reddit-full**. Дополнительные файлы можно положить в директорию **packer/files**. Для запуска приложения при старте инстанса необходимо использовать **systemd unit**.

P.S. Этот образ можно строить "поверх" базового.

Задание со *

Для ускорения работы можно запускать виртуальную машину с помощью командной строки и утилиты **gcloud**.

Создайте shell-скрипт с названием **create-reddit-vm.sh** в директории **config-scripts**. Запишите в него команду которая запустит виртуальную машину из образа подготовленного вами в рамках этого ДЗ, из семейства **reddit-full**, если вы выполнили первое задание со звездочкой, или **reddit-base**, если не выполняли.

Проверка ДЗ

- Результаты вашей работы находятся в ветке **packer-base** вашего инфраструктурного репозитория.
- В README внесите описание того, что сделано.
- Создайте Pull Request к ветке master (**описание PR нужно заполнять**);
- В ревьюеры можно никого не добавлять;
- Добавьте "Labels" **packer-base** и **Packer** к вашему Pull Request;
- После того, как один из преподавателей сделает approve пул реквеста, ветку с ДЗ можно смерджить и закрыть PR.
- P.S. TravisCI валидирует шаблон пакера и проверяет наличие необходимых файлов и пустой строки в их конце.