

Модели и методы управления инфраструктурой. Работа с образами VM.

План

- Типы серверов: Snowflake vs Phoenix
- Модели и методы управления инфраструктурой
- Инфраструктурная модель Base-Service-App
- Работа с образами VM: base image, bake vs fry, immutable infrastructure
- Сервисные модели: pets vs cattle
- Работа с Packer

Ручная установка и конфигурация - есть ли проблемы?

- ...
- ...
- ...
- ...
- ...
- ...



Сервера снежинки

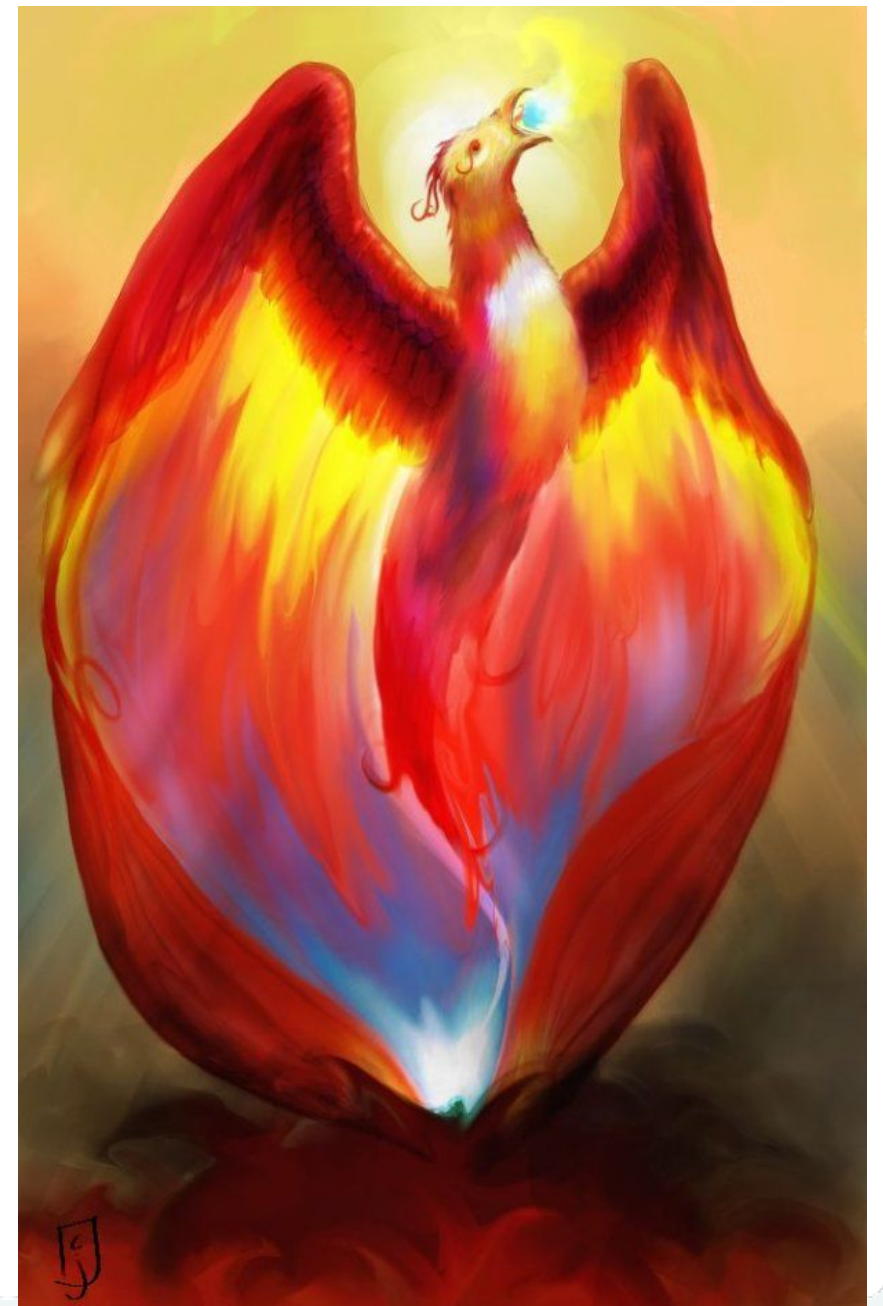
- Уникальные
- Сложны для понимания
- Хрупкие
- Неповторимые



Феникс сервера

Сервер должен быть как феникс, регулярно восставая из пепла.

М. Фаулер



Модели и методы управления инфраструктурой

Модели управления

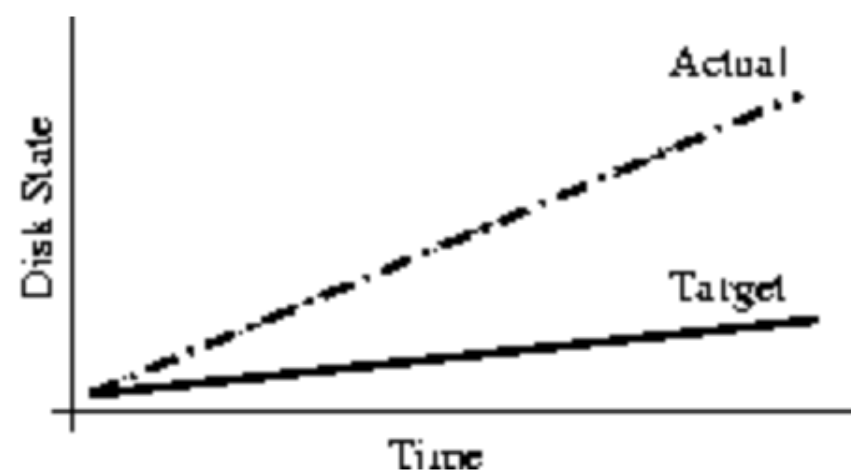


Figure 4.1.1: Divergence

расхождение

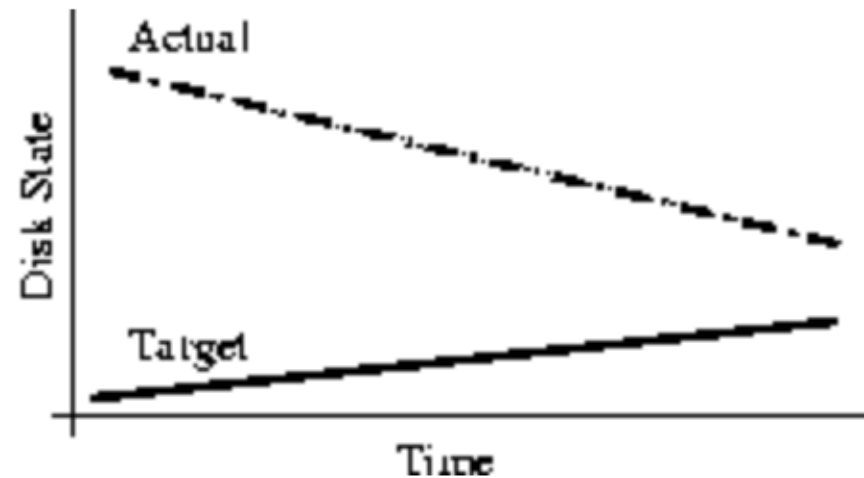


Figure 4.2.1: Convergence

схождение

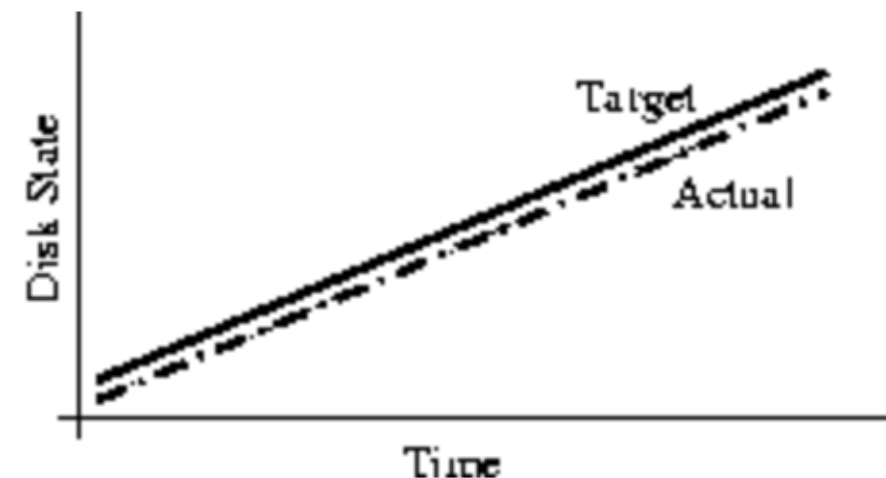


Figure 4.3.1: Congruence

соответствие

Методы управления

- Manual
- Scripts
- Infrastructure as a Code (IaC)
- Immutable Infrastructure
- Immutable Delivery

Взгляд на инфраструктуру

Base-Service-App модель

Application

Производимые продукты, приносящие прибыль компании либо имеющие внутреннюю ценность для ее работы.

Service

Поддерживающие сервисы, требуемые для работы приложений.

Base

Настроенная операционная система, отвечающая предъявленным требованиям, и поддерживающая инфраструктура.

Base-app-service модель

Layer

Responsibility

Application

Dev
Ops*

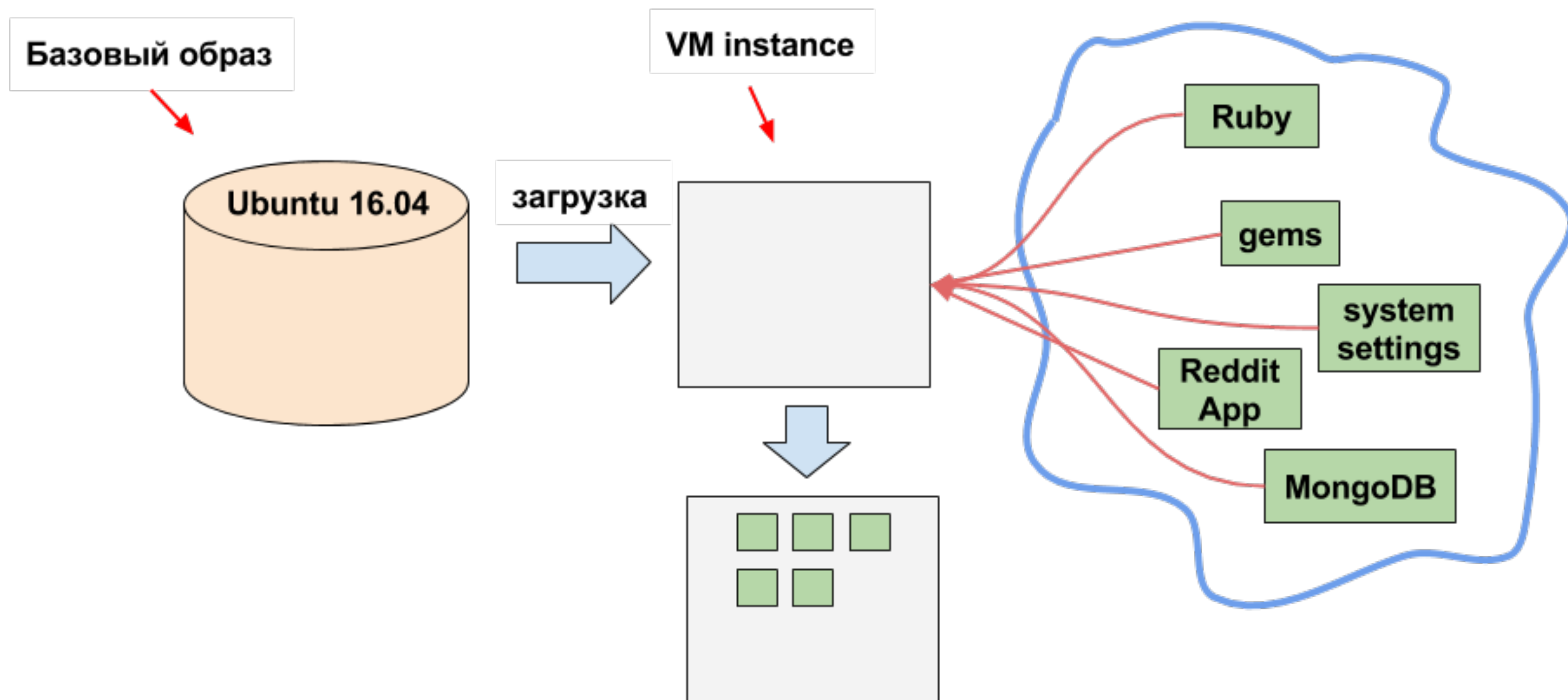
Service (IIS, MS SQL, MongoDB)

Ops

Base (Windows, Linux)

Ops

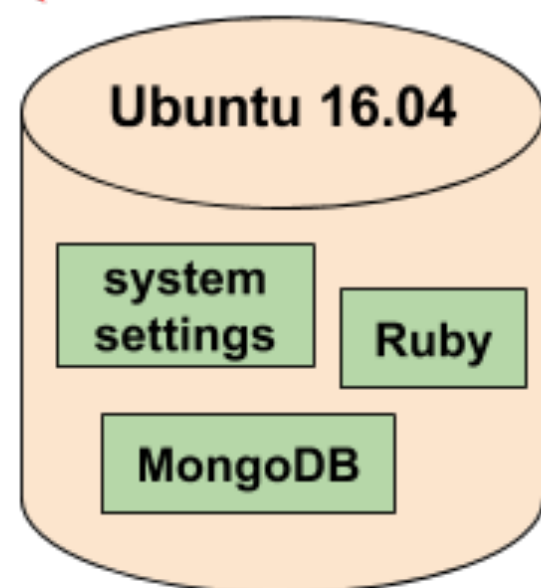
Управление базовым слоем (Base)



Fry

- Минимальный образ VM
- Настройка системы производится после запуска инстанса
- Низкая скорость деплоя
- Возможные проблемы с установкой из сторонних репозиториев

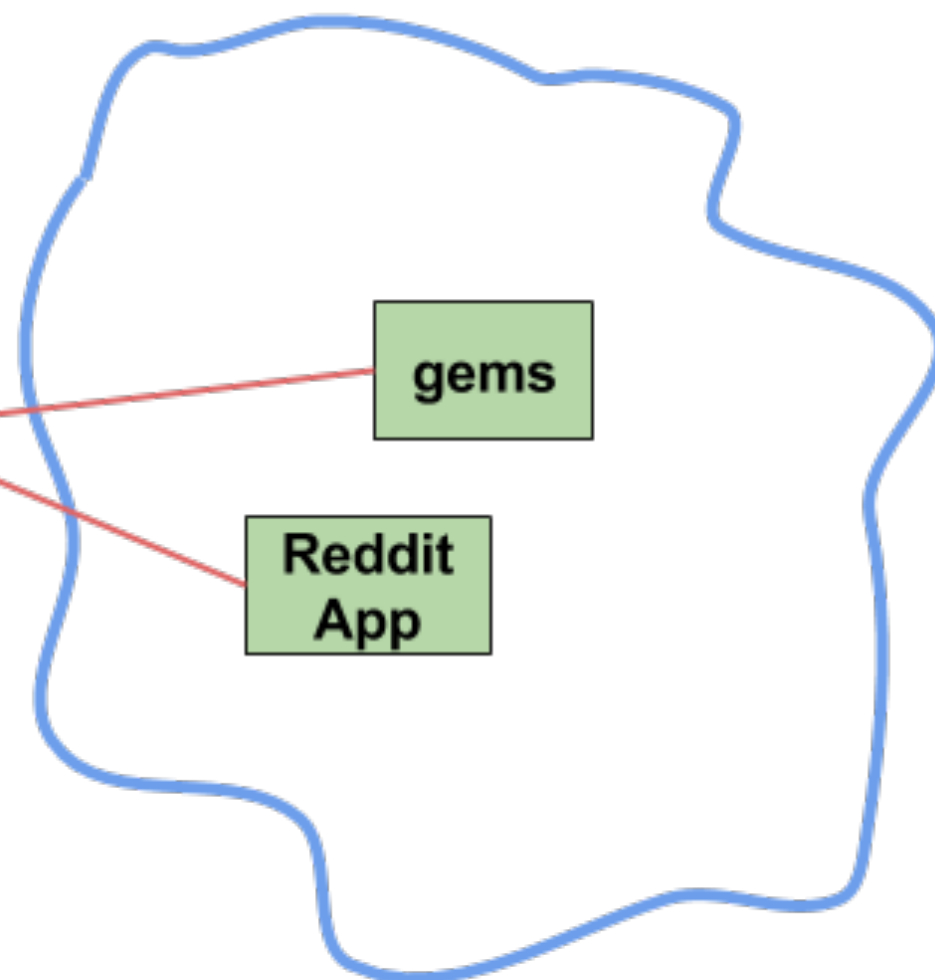
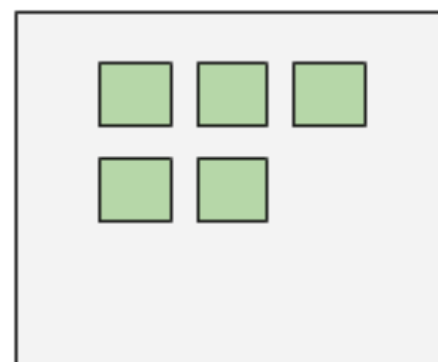
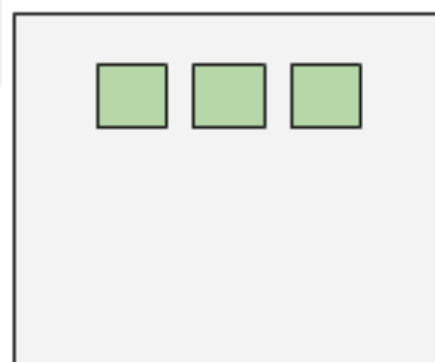
Базовый образ



загрузка



VM instance



Wake

- Образ содержит необходимые настройки системы
- Пакеты и сервисы заранее установлены в образ
- Ускоряется деплой
- Повышается надежность деплоя
- Сложности управления версиями

При создании базового образа, учитываем ...

- Обязательные настройки и ПО
- Скорость установки пакетов и зависимостей
- Доступность удаленных репозиториев
- Частота изменения

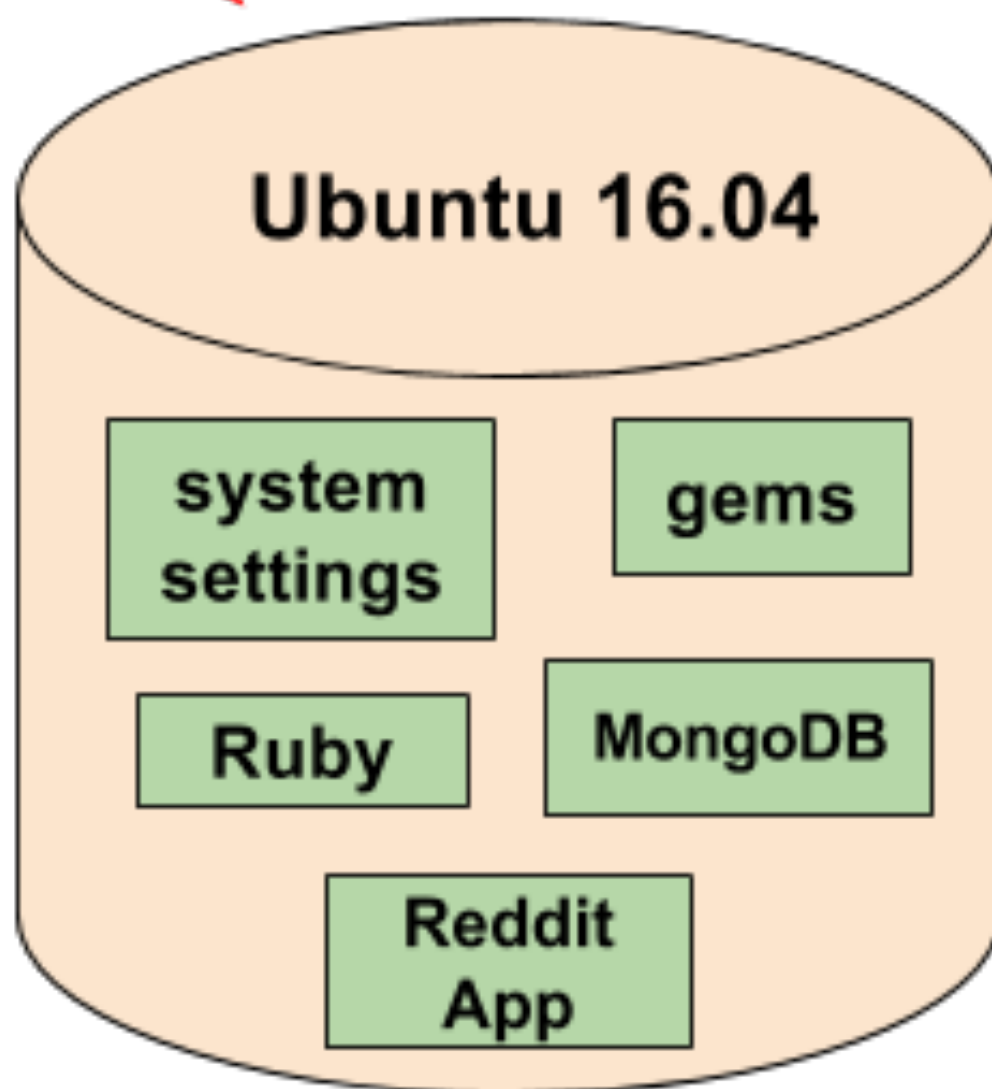
Базовый образ ОС может включать

- Настройки ОС (ядро, сеть, DNS)
- Пакеты, зависимости с низкой частотой изменения
- Агенты мониторинга, логирования, управления конфигурацией

Методы управления инфраструктурой

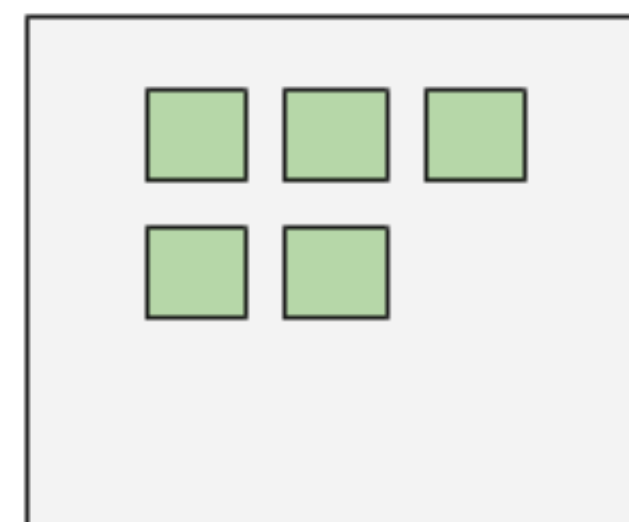
- Manual
- Scripts
- Infrastructure as a Code (IaC)
- **Immutable Infrastructure**
- Immutable Delivery

Базовый образ



VM instance

загрузка



Immutable Infrastructure

- Образ VM - артефакт для деплоя
- Не делаем изменений на запущенном инстансе
- Любые изменения сопровождаются билдом нового образа
- Старый инстанс сменяется новым

Преимущества II

- Быстрая скорость развертывания
- Минимальный дрейф конфигурации

Недостатки II

- Скорость развертывания уступает Immutable Delivery
- Not Immutable с точки зрения разработчика
- Требуется использование IaC инструментов
- Нужна хорошая модель управления версиями образов

Что почитать про II

- Building with Legos
- ImmutableServer



Pets vs Cattle

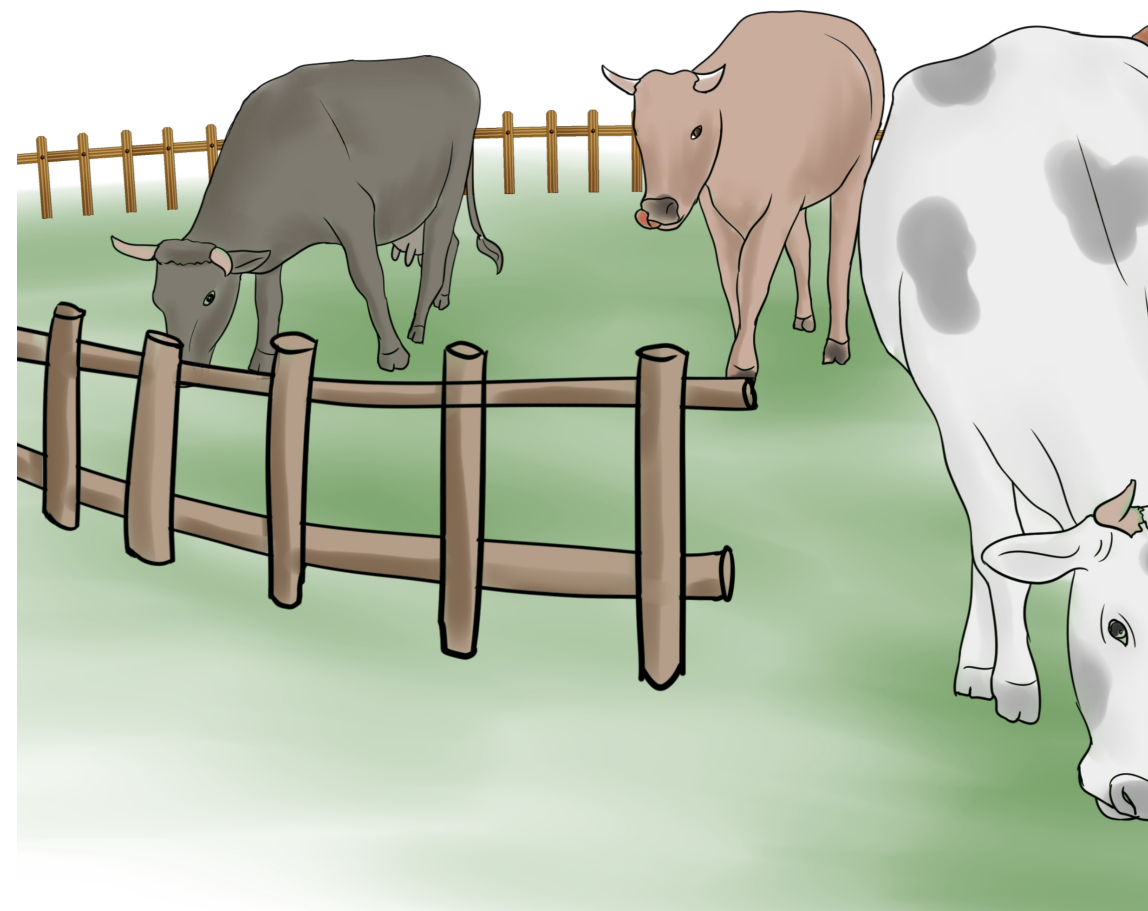
Pets

- Уникальны и неповторимы
- Имеют особые имена
- Не могут заболеть
- Если болеют, то пытаемся лечить



Cattle







- Все подобны друг другу
- Не имеют специальных имен, обычно нумеруются
- Если болеют, то убиваем, заменяем другими таким же



The background is a light blue gradient with a subtle pattern of small white dots representing stars. There are also several small, stylized planets with rings and comets scattered across the scene. The word "Packer" is centered in a large, bold, orange font.

Packer

Hashicorp Ecosystem

DEVELOP	PROVISION		SECURE		RUN
					
Vagrant Create and configure portable development environments	Packer Create platform specific machine images from a single source	Terraform Create, combine and manage infrastructure across multiple providers	Vault Centrally store, secure and control access to distributed secrets	Nomad Cluster manager and scheduler to deploy applications across any infrastructure	Consul Distributed highly available tool for service discovery, configuration and orchestration

Packer

- Описание машинных образов в виде кода
- Файлы с описанием называются шаблонами (templates)
- Создание образов для AWS, GCP, Docker, VirtualBox, etc.

Пример шаблона

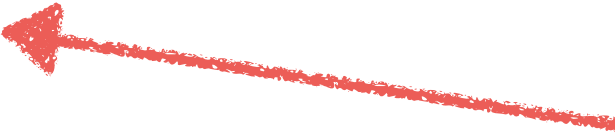
```
{
  "variables": {
    "proj_id": "steam-strategy-174408",
    "machine_type": null
  },
  "builders": [
    {
      "type": "googlecompute",
      "project_id": "{{user `proj_id`}}",
      "image_name": "reddit-base-{{timestamp}}",
      "source_image": "ubuntu-1604-xenial-v20170815a",
      "zone": "europe-west1-b",
      "ssh_username": "appuser",
      "machine_type": "{{user `machine_type`}}"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "script": "install_mongodb.sh",
      "execute_command": "sudo {{.Path}}"
    }
  ]
}
```

User variables

Объявляем переменные и задаем значение по умолчанию:

```
"variables": {  
  "proj_id": "default-project",  
  "machine_type": null  
}
```

Означает, что
переменная
обязательна для
определения



Используем в шаблоне:

```
"builders": [  
  {  
    ...  
    "project_id": "{{user `proj_id`}}",  
    "machine_type": "{{user `machine_type`}}"  
  }  
]
```


User variables

Определение переменных из командной строки

```
$ packer build \  
    -var 'project_id=steam-strategy-174408' \  
    -var 'machine_type=f1-micro' \  
    ubuntu16.json
```

Определение переменных через файл

```
$ packer build -var-file=variables.json ubuntu16.json
```

Builders

- Отвечают за запуск виртуальной машины из заданного базового образа и создание нового образа на ее основе.

Builder definition

```
"builders": [  
  {  
    "type": "googlecompute",  
    "project_id": "{{user `proj_id`}}",  
    "image_name": "reddit-base-{{timestamp}}",  
    "source_image": "ubuntu-1604-xenial-v20170815a",  
    "zone": "europe-west1-b",  
    "ssh_username": "appuser",  
    "machine_type": "{{user `machine_type`}}"  
  }  
]
```

Provisioners

- Позволяют производить настройки ОС и установку ПО на созданной билдером машине.

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

Основные команды

Проверка синтаксиса и конфигурации:

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

Проверка основных секций шаблона

```
$ packer inspect ubuntu16.json
```

Билд образа:

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