



# Infrastructure as a code

Infrastructure as a code

Длительность курса: 82 академических часа

# 1 Введение в Infrastructure as Code

## 1 Infrastructure as Code и практики разработки

### Цели занятия:

после занятия вы получите представление о различных подходах и инструментах методологии Infrastructure as Code.

### Краткое содержание:

введение в IaC как в одну из практик организации инфраструктуры: подготовка инфраструктурных репозиторий, тестирование кода и построение пайплайнов.

---

## 2 Packer. Создание "Золотых образов" // ДЗ

### Цели занятия:

понять принципы IaC;  
понять подходы Fry/Bake и принципы Immutable Infrastructure;  
научиться писать простейшие шаблоны для Packer.

### Краткое содержание:

AS-IS: ад "ручного" конфигурирования;  
TO-BE: Infrastructure as Code;  
Immutable Infrastructure;  
Hashicorp Stack & Packer.

### Домашние задания

#### 1 Подготовка базового образа VM при помощи Packer

Цель: В данном дз студент произведет сборку готового образа с установленным nginx при помощи Packer.

В данном задании тренируются навыки: работы с Packer, работы с Yandex Compute Engine.

1. Установите Yandex.CLI:  
<https://cloud.yandex.ru/docs/cli/operations/install-cli>

2. Установите Packer: <https://www.packer.io/>

3. Подключитесь при помощи Yandex.CLI к облаку и просмотрите конфигурацию:

```
yc init
```

```
yc config list
```

4. Склонируйте к себе репозиторий

<https://github.com/timurb/otus-packer> и создайте у себя в облаке образ:

```
packer build json/template.json
```

(понадобится установить переменные)

5 (\*). Создайте сервисную учетную запись для packer:

```
yc iam service-account create --name packer --  
folder-id XXXXXXXXXXXXXXXX
```

```
yc iam service-account get
```

```
yc resource-manager folder add-access-binding --  
id XXXXXXXXXXXXXXXX --role editor --service-  
account-id XXXXXXXXXXXXXXXX
```

```
yc iam key create --service-account-id  
XXXXXXXXXXXXXXXXXX --output key.json
```

6 (\*). Вставьте путь к вашему ключу в `template.json` в `"service_account_key_file"` в блок `builders` и создайте из этого шаблона образ (вероятно понадобится удалить из шаблона параметр `token`)

7 (\*). Параметризируйте сборку, например добавьте установку `image_description` пользователем

8. Удалите созданные образы:

```
yc compute image list
```

```
yc compute image delete
```

9. Пришлите последние строчки вывода `packer build` (те, в которых указан `id`) и получившийся в финале `template.json`

Как вариант задания со (\*) реализуйте это не на Yandex.Cloud, а на любом другом облаке или системе виртуализацию

---

### 3 Terraform как инструмент для декларативного описания инфраструктуры // ДЗ

#### Цели занятия:

поднимать инфраструктуру при помощи инструмента Terraform.

#### Краткое содержание:

обзор декларативного инструмента описания инфраструктуры и сравнение с ближайшими аналогами.

#### Домашние задания

- 1 Развернуть при помощи Terraform тестовую среду, включающую в себя хосты для front-end, back-end и базы данных.

Цель: Развернуть при помощи Terraform тестовую среду, включающую в себя front-end, back-end и базы данных.

Методичка по выполнению ДЗ:  
<https://hackmd.io/@otus/rkCER8MfK>

---

### 4 Дополнительные уровни абстракции в Terraform

#### Краткое содержание:

Обзор функционала для создания повторно используемых компонентов Terraform.

---

### 5 Избегаем дублирования кода манифестов в Terraform

#### Краткое содержание:

Расширяем функционал Terraform при помощи Terragrunt, а также модулей Terraform.

---

## 6 Тестирование инфраструктурного кода на Terraform // ДЗ

### Цели занятия:

подготовить тесты для проверки корректности описания инфраструктуры.

### Краткое содержание:

описание инструментов Terragrunt, TerraTest и tfswitch для автоматизации тестирования кода Terraform.

### Домашние задания

- 1 Написать тесты для тестирования кода Terraform из ДЗ1

Цель: Написать тесты для тестирования кода Terraform из ДЗ1

Методичка по выполнению ДЗ:  
<https://hackmd.io/@otus/B1TK3dUMK>

---

## 7 Pulumi. Описание инфраструктуры языком программирования

### Цели занятия:

понять, зачем нужен Pulumi;  
научиться создавать новый проект;  
научиться хранить состояние и тестировать инфраструктуру.

### Краткое содержание:

узнаем, что такое Pulumi;  
практика на Pulumi;  
написание тестов.

### 1 Введение в Ansible

#### Цели занятия:

объяснить основные возможности и особенности инструмента управления инфраструктурой Ansible.

#### Краткое содержание:

плейбуки, факты, базовые переменные, инвентори, конфигурирование ansible/ssh, принципы работы; разработка и запуск базового плейбука.

---

## 2 Переиспользование кода Ansible и работа с ролями // ДЗ

### Цели занятия:

рассказать о способах совместной работы над ролями Ansible.

### Краткое содержание:

роль в Ansible;  
алгоритм работы с ролями из Ansible Galaxy;  
когда стоит взять роль из Galaxy, а когда стоит написать самому;  
Ansible Vault.

### Домашние задания

- 1 Первичная настройка хостов, созданных в предыдущем ДЗ, при помощи ролей Ansible.

Цель: Первичная настройка хостов, созданных в предыдущем ДЗ, при помощи ролей Ansible.

У вас есть хосты wp-app-1 и wp-app-2, а так же кластер базы данных, развернутые терраформом.

С помощью ansible на wp-app-1 и wp-app-2 сделать следующее:

1. Поставить wordpress
2. Настроить wordpress для коннекта к БД

Зайти по IP-адресу loadbalancer'a и произвести окончательную установку Wordpress.

Описание работы можно найти тут:  
<https://hackmd.io/@otus/HJ7nJqKmK>

---

## 3 Эволюция Ansible после версии 2.x.

### Краткое содержание:

Рассмотрим, чем отличаются версии Ansible до 3.0 и после.

---

4 **Chef. Управление конфигурациями**

**Цели занятия:**

изучить основные концепции Chef;  
научиться писать несложные рецепты.

**Краткое содержание:**

экосистема Chef;  
Chef Infra;  
темы для изучения.

---

5 **SaltStack. Управление конфигурациями**

**Цели занятия:**

поработать с системой управления конфигурациями Saltstack;  
разобраться в чем ее отличие от прочих аналогичных систем.

**Краткое содержание:**

работа с salt-ssh;  
salt-master и salt-minion;  
основные аспекты построения структуры states;  
применение изменения к хостам.

---

6 **Puppet. Управление конфигурациями**

---

7 **SCM в Облаке. AWS CloudFormation**

## 1 Процесс работы с инфраструктурными пайплайнами в Gitlab // ДЗ

### Цели занятия:

описать особенности построения инфраструктурного пайплайна в Gitlab.

### Краткое содержание:

методика организации инфраструктурного пайплайна на примере Gitlab.

### Домашние задания

- 1 Реализовать в Gitlab инфраструктурный пайплайн, включающий в себя развертывание и тестирование инфраструктуры.

Цель: Реализовать в Gitlab инфраструктурный пайплайн, включающий в себя развертывание инфраструктуры.

Методичка по выполнению ДЗ:  
<https://hackmd.io/@otus/rymWuinmY>

---

## 2 Построение инфраструктурных пайплайнов в Jenkins

### Цели занятия:

разработать инфраструктурный пайплайн в Jenkins.

### Краткое содержание:

особенности DSL Jenkins при написании инфраструктурных пайплайнов.

---

## 3 Скрипты миграции для баз данных

### Краткое содержание:

Изучаем два наиболее популярных инструмента для версионирования изменений в базах данных.

## 1 Documentations as code

### Цели занятия:

описать понятие documentation as code, самоописываемый код и как к нему прийти.

### Краткое содержание:

понятие documentation as code; самоописываемый код.

---

## 2 Dashboards as Code

---

## 3 Создание виртуальной инфраструктуры на рабочей станции при помощи Vagrant

### Цели занятия:

подготовить и развернуть виртуальную инфраструктуру при помощи Vagrant.

### Краткое содержание:

Vagrant.

## 1 Выбор темы и организация проектной работы

### Цели занятия:

выбрать и обсудить тему проектной работы;  
спланировать работу над проектом;  
ознакомиться с регламентом работы над проектом.

### Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;  
требования к результату проекта и итоговой документации.

---

## 2 Консультация по проектам и домашним заданиям

### Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу.

### Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;  
затруднения при выполнении ДЗ;  
вопросы по программе.

---

### 3 **Защита проектной работы**

#### **Цели занятия:**

защитить проект и получить рекомендации экспертов.

#### **Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.

#### **Домашние задания**

##### 1 **Защита проектной работы**

Цель: Создание инфраструктурного проекта с развертыванием нескольких сред.

Отдельная среда предусматривает наличие нескольких хостов различной функциональной направленности (к примеру, фронт/бэк/база данных)

Отвечает следующим требованиям:

- код проекта хранится в git-репозитории
- предусмотрено тестирование инфраструктурного кода
- развертывание сред происходит в соответствии с практиками CI/CD

Рекомендуемые к использованию инструменты:  
Jenkins, Gitlab, Terraform, Ansible, Puppet, Chef, SaltStack