

Полная программа

Infrastructure as a code

Infrastructure as a code

Длительность курса: 94 часа

Модуль 1. Введение в Infrastructure as Code

Тема 1 Packer. Infrastructure as Code и практики разработки

Цель занятия
знать и понимать различные подходы и инструменты методологии Infrastructure as Code.

Краткое содержание
введение в IaC как в одну из практик организации инфраструктуры; подготовка инфраструктурных репозиториев; тестирование кода; построение пайплайнов.

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

Цель занятия
понять принципы IaC, понять подходы Fly/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

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.

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

Цель занятия
поднимать инфраструктуру при помощи инструмента Terraform.

Краткое содержание
обзор декларативного инструмента описания инфраструктуры и сравнение с ближайшими аналогами.

Домашние задания

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.

Тема 4 Дополнительные уровни абстракции в Terraform

Цель занятия
знать и понимать функционал для создания повторно используемых компонентов Terraform.

Краткое содержание
компоненты Terraform.

Тема 5 Избегаем дублирования кода манифестов в Terraform

Цель занятия
понимать, как расширить функционал Terraform при помощи Terragrunt, а также модулей Terraform.

Краткое содержание
Terraform и его модули; Terragrunt.

Тема 6 Тестирование инфраструктурного кода на Terraform // ДЗ

Цель занятия
подготовить тесты для проверки корректности описания инфраструктуры.

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

Домашние задания

Написать тесты для тестирования кода Terraform из ДЗ1
Цель
Написать тесты для тестирования кода Terraform из ДЗ1

Модуль 2. Инструменты SCM

Тема 1 Сравнение с GitOps

Цель занятия
понимать методологии gitops, их особенности, различия и способы сочетания.

Краткое содержание
краткое описание методологии gitops; сравнительная таблица различий; способы сочетания двух методологий.

Тема 2 Введение в Ansible

Цель занятия
понимать 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.

Тема 3 Переиспользование кода Ansible и работа с ролями // ДЗ

Цель занятия
знать и понимать способы 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

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

Тема 4 Работа с динамическими инвенторными и интеграция с облаками

Цель занятия
уметь использовать Ansible для автоматизации управления инфраструктурой в облаках и создавать динамические инвенторизации.

Краткое содержание
создание и настройка динамических инвентарей для облачных окружений с использованием инструментов, таких как ansible-inventory и соответствующих файлов; применение динамических инвентарей для автоматизации развертывания и конфигурации ресурсов в облаке с помощью Ansible playbooks;

работа с переменными и шаблонами для более гибкого управления инфраструктурой в облаках с помощью Ansible.

Тема 5 Работа с Ansible Vault: шифрование данных и настройка безопасности

Цель занятия
знать и понимать шифрование конфиденциальных данных и настройку безопасности при работе с Ansible Vault.

Краткое содержание
шифрование файлов, переменных, секретов и других конфиденциальных данных; работа с Ansible Vault в плейбуках и ролях для безопасной обработки конфиденциальных данных; настройки безопасности.

Тема 6 Chef. Управление конфигурациями

Цель занятия
знать и понимать основные концепции Chef; уметь писать несложные рецепты.

Краткое содержание
экосистема Chef; Chef Infra; темы для изучения.

Тема 7 Эволюция Ansible после версии 2.x.

Цель занятия
знать и понимать новые функции Ansible после версии 2.x и способы их применения в проектах.

Краткое содержание
различия между версиями Ansible до 3.0 и после.

Тема 8 Puppet. Управление конфигурациями

Цель занятия
знать и понимать основные концепции Puppet; уметь писать несложные рецепты.

Краткое содержание
основные концепции Puppet; примеры написания несложных рецептов и их применения на практике.

Модуль 3. Пайплайн для инфраструктуры как код

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

Цель занятия
знать и понимать особенности построения инфраструктурного пайплайна в Gitlab; уметь делать его описание.

Краткое содержание
методика организации инфраструктурного пайплайна на примере Gitlab.

Домашние задания

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

Тема 2 Построение инфраструктурных пайплайнов в Jenkins

Цель занятия
разработать инфраструктурный пайплайн в Jenkins.

Краткое содержание
особенности DSL Jenkins при написании инфраструктурных пайплайнов.

Тема 3 Интеграция Ansible в пайплайны GitLab. Работа с секретами

Цель занятия
знать возможности интеграции Ansible в пайплайны GitLab и принципы работы с секретами в GitLab.

Краткое содержание
интеграция Ansible в пайплайны GitLab: методика, возможности; работа с секретами в GitLab: основные принципы; примеры.

Тема 4 ConcourseCI

Цель занятия
знать и понимать процесс работы с пайплайнами; понимать различия с GitLab; объяснить возможности интеграции с k8s и использования ansible.

Краткое содержание
пайплайны; GitLab; интеграция с k8s; ansible.

Тема 5 Создание виртуальной инфраструктуры на рабочей станции при помощи Vagrant

Цель занятия
подготовить и развернуть виртуальную инфраструктуру при помощи Vagrant.

Краткое содержание
Vagrant.

Тема 6 SaltStack. Управление конфигурациями

Цель занятия
понимать работу системы управления конфигурациями Saltstack; знать и понимать, 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.

Тема 7 Puppet. Управление конфигурациями

Цель занятия
знать и понимать основные концепции Puppet; уметь писать несложные рецепты.

Краткое содержание
основные концепции Puppet; примеры написания несложных рецептов и их применения на практике.

Модуль 4. Everything as Code

Тема 1 Documentations as code

Цель занятия
объяснить понятие documentation as code, самописываемый код и как к нему прийти; знать необходимые инструменты для dac.

Краткое содержание
понятие documentation as code; самописываемый код.

Тема 2 Использование hiera

Цель занятия
знать и понимать принцип работы hiera; объяснить применение этого механизма.

Краткое содержание
использование hiera как механизма отделения данных от логики; использование модулей из puppet forge; написание собственных модулей.

Тема 3 Dashboards as Code

Цель занятия
знать и понимать возможности переиспользования дашбордов и генерации с помощью шаблонизации (рассмотрение на примере grafana).

Краткое содержание
grafana; переиспользование дашбордов; шаблонизация.

Тема 4 Pulumi. Описание инфраструктуры языком программирования

Цель 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.

Тема 5 Скрипты миграции для баз данных

Цель занятия
понимать создание скриптов миграции для баз данных и сферы их использования.

Краткое содержание
два наиболее популярных инструмента для версионирования изменений в базах данных.

Модуль 5. Проектная работа

Тема 1 Выбор темы и организация проектной работы

Цель 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.

Тема 2 SCM в Облаке. AWS CloudFormation

Цель занятия
понимать возможности использования инфраструктуры системы контроля версий (SCM) в облачных средах.

Краткое содержание
обзор AWS CloudFormation и его преимуществ при автоматизации развертывания инфраструктуры на основе кода; создание и управление шаблонами AWS CloudFormation для определения инфраструктуры как кода; использование SCM (например, Git) для управления версиями и историей изменений шаблонов AWS CloudFormation; работа с ветвлением и слиянием в SCM для организации и координации разработки и развертывания инфраструктуры в облачной среде с помощью AWS CloudFormation.

Тема 3 Консультация по проектам и домашним заданиям

Цель 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.

Тема 4 Защита проектной работы

Цель занятия
защитить проект и получить рекомендации экспертов.

Краткое содержание
презентация проектов перед комиссией; вопросы и комментарии по проектам.

Домашние задания

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