

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра технологий программирования

Лабораторная работа №5
Проектирование модели управления версиями ПО,
конфигурациями и непрерывной поставкой
По дисциплине «Непрерывное интегрирование и сборка
программного обеспечения»

Методические указания по выполнению лабораторной работы

Подготовила:
Давидовская М. И.,
Ст. преподаватель кафедры ТП

Минск, 2025 г.

Содержание

Цель работы.....	3
Задачи работы.....	3
Задания.....	3
Методические указания.....	3
Критерии оценивания.....	3
Содержание отчета.....	3
Задание 1. Основы управления конфигурацией инфраструктуры с Terraform.....	4
Задание 2. Управление конфигурацией серверов с помощью Ansible...	5
Задание 3. Создание кластера с помощью Terraform и Ansible локально.....	5
Контрольные вопросы.....	7

Цель работы

Изучение решений для автоматизации управления инфраструктурой и установки приложений.

Задачи работы

1. Установить и изучить применение Terraform.
2. Установить и изучить Ansible
3. Продемонстрировать развертывание инфраструктуры из трех виртуальных машин и кластера kubernetes, установка необходимых приложений с использованием Terraform и Ansible.
4. Оформить отчёт в формате Markdown и создать Pull Request в git-репозитории.

Задания

Методические указания

Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «Непрерывное интегрирование и сборка программного обеспечения»

Критерии оценивания

Для групп 11-13 выполнить задания 1-3, для группы 14 — задания 1-2

Содержание отчета

1. Цель работы.
2. Вариант задания.
3. Код приложений, конфигурационных файлов.
4. Ответы на контрольные вопросы.

Отчет должен быть опубликован в git-репозитории на github. Все результаты лабораторной работы должны быть опубликованы в git-репозитории, ссылка на который доступна в курсе «Непрерывное интегрирование и сборка программного обеспечения».

В файле Readme проекта на github должна быть ссылка на отчёт. Отчет опубликовать во внешнем хранилище или в репозитории в каталоге /docs. Если в лабораторной работе необходимо написать

программу/ы, то отчёт должен результаты тестов по каждой программе и ответы на контрольные вопросы.

Пример оформления файла Readme может быть таким:

```
# Overview

Report on LabRabota5.

# Usage

// Заменить <<link>> и <<folder>> на соответствующие ссылки/названия
To check, please, preview report by <<link>> and source files
in <<folder>>.

# Author

Your name and group number.

# Additional Notes

// СКОПИРОВАТЬ И ВСТАВИТЬ ССЫЛКУ НА СВОЙ РЕПОЗИТОРИЙ, НАПРИМЕР
https://github.com/maryiad/lab5-task1-gr16-david
```

Каждая лабораторная работа содержит тексты задач и контрольные вопросы, ответы на которые проверяются преподавателем при приёме работы у студента.

Выполнение студентом лабораторной работы и сдача её результатов преподавателю происходит следующим образом:

1. Студент выполняет разработку программ.
2. В ходе разработки студент обязан следовать указаниям к данной задаче (в случае их наличия). Исходные тексты программ следует разрабатывать в соответствии с требованиями к оформлению, приведёнными в приложении.
3. Студент выполняет самостоятельную проверку исходного текста каждой разработанной программы и правильности её работы, а также свои знания по теме лабораторной работы.

Задание 1. Основы управления конфигурацией инфраструктуры с Terraform

1. Изучить официальное руководство Terraform Tutorial и выполнить все 7 руководств в виртуальных лабораториях к каждому руководству:
 - [Get Started - Docker by Terraform](#)

2. Продемонстрировать выполнение каждого из руководств и добавить в отчет.

Задание 2. Управление конфигурацией серверов с помощью Ansible

1. Изучить статьи [How To Set Up Ansible Inventories](#) и [How To Manage Multiple Servers with Ansible Ad Hoc Commands](#)
2. Изучить документацию [Using Ansible command line tools](#)
3. Продемонстрировать подключение к виртуальным машинам, созданным для кластера Kubernetes в лабораторной работе 3 и установкой любого пакета, например zsh, с помощью команд.
4. Изучить статью [How To Write Ansible Playbooks](#) и документацию [Using Ansible playbooks](#).
5. Настроить конфигурации в виде файлов playbook на основе примеров из статьи в п. 4 задания и продемонстрировать их.
6. Выполнить примеры из руководства [Ansible, Playbook](#) и продемонстрировать результаты.
7. Опубликовать конфигурационные файлы в репозиторий.

Задание 3. Создание кластера с помощью Terraform и Ansible локально

Чтобы развернуть готовый кластер Kubernetes, можно использовать разные способы — например, самому подготовить виртуальные машины или выделенные серверы, настроить операционную систему и установить необходимые программы.

Если создание кластера из виртуальных машине локально или в облаке окажется трудоемким, можно использовать для построения кластера вместо виртуальных машин метод docker-IN-docker и утилиту kind.

Можно выбрать любой из 2 вариантов, предложенных ниже, но первый вариант — предпочтительнее. Обязательное условие — для развертывания использовать Terraform, Ansible, Kubernetes и Docker в независимости от варианта.

Вариант 1) Создание кластера из локальных виртуальных машин или виртуальных серверов в облаке.

Для применения Terraform необходимо настроить провайдера, в качестве которого может выступать облачный провайдер, например

Google Cloud, Amazon AWS, Yandex Cloud и др., или локальный провайдер, реализованный в виде подключения к приложению виртуализации, например Virtualbox, Proxmox, VMware, Libvirt/KVM и др.

В задании необходимо описать в конфигурационных файлах Terraform создаваемую инфраструктуру из 3-х виртуальных машин, например на основе провайдера Virtualbox или Libvirt/KVM, установку требуемого программного обеспечения в каждой из них с помощью Ansible и развертывание кластера Kubernetes на основе виртуальных машин.

При наличии учетной записи в любом облачном провайдере задание может быть выполнено и во внешней инфраструктуре облачного провайдера.

Вариант 2) Создание кластера на основе контейнеров Docker:

kind (Kubernetes in Docker) — это инструмент, при помощи которого можно развернуть готовый кластер Kubernetes на своем устройстве, например стационарном компьютере или ноутбуке. Kind полагается на систему контейнеризации Docker. Вместо Docker можно использовать podman или nerdctl. kind представляют ноды (узлы) кластера как запущенные Docker-контейнеры, так как использует метод Docker-in-Docker (DinD).

1. Вариант 1: Изучить статью «[How to Create a Local Kubernetes Cluster: Terraform and Ansible](#)».

✓ Вместо Proxmox использовать провайдер для Virtualbox или Libvirt/KVM:

1. [Создание виртуальной машины с помощью Terraform](#) с провайдером Virtualbox для terraform <https://github.com/terraform-providers/terraform-provider-virtualbox>
2. Другой провайдер Virtualbox для terraform — <https://github.com/mixdone/terraform-provider-virtualbox>
3. Еще один пример [Using terraform for automation creating virtualbox virtual machine](#)
4. [Использование Terraform на серверах с KVM. Часть 1](#) и [Использование Terraform на серверах с KVM. Часть 2](#)
5. [Local Terraform Experimentation: A Beginner's Guide](#) на основе Libvirt/KVM

6. [libvirt-k8s-provisioner - Automate your cluster provisioning from 0 to k8s!](#)
- ✓ **Вариант 2:** Использование kind для создания кластера с помощью метода Docker-IN-Docker:
7. [Делаем домашнюю DevOps-лабораторию на ноутбуке без затрат](#) — использовать как основное руководство для данного способа
8. [Как установить и использовать кластер Kubernetes при помощи утилиты kind](#)
9. [Kind Quick Start](#)
10. [Deploying Kubernetes Cluster Using Terraform](#)
11. [Docker & Kubernetes: Multi-node Local Kubernetes cluster - Kubeadm-kind\(K8s in Docker\)](#)
12. [Ansible - Deploy Nginx to Docker](#)
2. Продемонстрировать результат развертывания инфраструктуры.
3. Код всех конфигурационных файлов опубликовать в репозиторий.

Контрольные вопросы

1. Для каких задач применяется Ansible?
2. Какие способы управления серверами применяются в Ansible?
3. Что входит в состав файла ansible playbook?
4. Что такое группы узлов (хостов) в Ansible?
5. Как одной командой выполнить ping группы узлов (хостов)?
6. Что такое роль Ansible?
7. Что такое файл инвентаризации Ansible и как он используется?
8. Что такое специальная команда Ansible?
9. Как перезапустить службу с помощью Ansible?
10. Что такое Hashicorp Terraform?
11. Что такое инфраструктура как код (Infrastructure as code — IaC)?
12. Как установить Terraform?
13. Что такое провайдер (поставщик) в Terraform?
14. Объясните базовую структуру файла конфигурации Terraform.
15. Что такое файл состояния Terraform?
16. Для чего нужна terraform destroy команда?
17. Что такое состояние в terraform.
18. Как определить зависимости в Terraform?