

Лабораторная работа №1: Настройка окружения и знакомство с базовыми концепциями распределённых систем

Дисциплина: Распределенная обработка данных **Направление подготовки:** 09.03.03 Прикладная информатика

Цель работы: Получить практический опыт развёртывания базовых компонент стека технологий для распределённой обработки данных. Освоить работу с виртуальными машинами, контейнеризацией (Docker) и оркестрацией (Kubernetes) на примере простого кластера. Закрепить понимание ключевых концепций распределённых систем: отказоустойчивости, масштабируемости и прозрачности.

Стек технологий:

- **Платформа/ОС:** Linux (Ubuntu 22.04 LTS) на виртуальной машине (VirtualBox/VMware)
- **Система контейнеризации:** Docker
- **Система оркестрации:** Kubernetes (режим Minikube)
- **Инструментарий:** Terminal, `kubectl`, `docker cli`

Теоретическая часть (краткое содержание из лекции):

Распределённая система — это совокупность независимых компьютеров, представляющих пользователю единой объединённой системой. Её ключевые свойства:

- **Масштабируемость:** Возможность увеличивать вычислительную мощность путём добавления новых узлов (горизонтальное масштабирование).
- **Отказоустойчивость:** Способность системы работать при сбоях отдельных её компонентов.
- **Прозрачность:** Сокрытие от пользователя и приложения распределённой природы системы (например, прозрачность местоположения и отказов).

Современные подходы к построению таких систем часто основаны на концепции **контейнеризации** (Docker) и **оркестрации** (Kubernetes), которые позволяют эффективно управлять жизненным циклом приложений в распределённом окружении.

Задание на практическую реализацию:

Этап 1: Подготовка базового окружения

1. Установите и настройте программу для виртуализации (VirtualBox или VMware).
2. Создайте виртуальную машину под управлением Ubuntu Server 22.04 LTS.
3. Настройте базовое окружение: обновите систему, установите SSH для удобного доступа.

Этап 2: Установка и настройка Docker

1. Установите Docker Engine на виртуальную машину, следуя официальной документации.
2. Проверьте корректность установки, запустив команду:

```
sudo docker run hello-world
```

3. Создайте свой первый образ. Напишите простой **Dockerfile** для запуска веб-сервера Nginx.

4. Соберите образ и запустите из него контейнер, пробросив порт контейнера (80) на порт хоста (например, 8080). Убедитесь, что веб-сервер доступен по <http://<IP-вашей-VM>:8080>.

Этап 3: Развёртывание одноджузового Kubernetes-кластера с помощью Minikube

1. Установите Minikube и **kubectl** на вашу виртуальную машину.

2. Запустите кластер командой **minikube start**.

3. Проверьте статус кластера и его компонентов:

```
kubectl cluster-info  
kubectl get nodes  
kubectl get pods -A
```

Этап 4: Демонстрация ключевых концепций на практике

1. Масштабируемость и прозрачность:

- Разверните в кластере развёртывание (Deployment) с вашим образом Nginx из Этапа 2.
- Создайте Сервис (Service) для доступа к этим контейнерам.
- Увеличьте количество реплик (копий) вашего приложения до 3 с помощью команды **kubectl scale deployment**.
- Убедитесь, что все поды (pods) перешли в состояние **Running**. Продемонстрируйте, что сервис предоставляет единую точку доступа к вашему приложению, несмотря на то, что оно запущено в нескольких экземплярах.

2. Отказоустойчивость:

- Вручную удалите один из подов вашего развёртывания командой **kubectl delete pod <pod-name>**.
- Наблюдайте в реальном времени (команда **kubectl get pods -w**), как система оркестрации автоматически создаёт новый под взамен удалённого, чтобы поддерживать желаемое количество реплик.
- Убедитесь, что сервис оставался доступным во время этой операции.

Требования к оформлению и отчёту:

1. **Код и конфигурации:** Предоставьте все использованные файлы конфигураций (**Dockerfile**, манифести Kubernetes для Deployment и Service).

2. **Скриншоты:** Приложите скриншоты, подтверждающие выполнение каждого этапа:

- Успешный запуск **hello-world** контейнера.
- Работающий веб-сервер Nginx из контейнера.
- Вывод команд **kubectl cluster-info** и **kubectl get nodes**.
- Вывод команды **kubectl get pods** с тремя репликами вашего приложения.

- Вывод команды `kubectl get pods`, показывающий автоматическое воссоздание удалённого пода.

3. **Отчёт:** Краткий письменный отчёт должен содержать:

- Описание проблем, с которыми вы столкнулись на каждом этапе, и их решение.
- Ответы на вопросы:
 - Какую проблему решает Docker?
 - Какую проблему решает Kubernetes?
 - Как продемонстрированные вами действия (масштабирование, пересоздание пода) относятся к ключевым свойствам распределённых систем (масштабируемость, отказоустойчивость, прозрачность)?

Критерии оценки:

- **Удовлетворительно (3):** Выполнены Этапы 1 и 2. Docker установлен, образ `hello-world` и собственный образ Nginx успешно запущены.
- **Хорошо (4):** Выполнены Этапы 1-3. Успешно развёрнут кластер Minikube. Развёрнуто приложение в кластере с помощью предоставленных манифестов.
- **Отлично (5):** Выполнены все этапы. Продемонстрированы масштабирование приложения и его отказоустойчивость путём удаления пода. В отчёте даны развёрнутые ответы на вопросы о решаемых проблемах и связи с концепциями распределённых систем.