|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Инструментального и прикладного программного обеспечения (ИиППО) | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 10**  «Использование кластера Kubernetes для оркестрации docker-контейнеров» | |
| **по дисциплине** | |
| **«**Настройка и администрирование сервисного программного обеспечения**»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Сидоров С.Д. |
| Принял преподаватель | Хозяинов А.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_ г. | *(подпись руководителя)* |

Москва 2023

Цель работы

Kubernetes управляет и запускает контейнеры Docker на большом количестве хостов, а также обеспечивает совместное размещение и репликацию большого количества контейнеров. K8s – сокращенное название Kubernetes. K[ubernete]s - в квадратных скобках 8 символов.

Термины Kubernetes.

Nodes (node.md): Нода это машина в кластере Kubernetes.

Pods (pods.md): Pod это группа контейнеров с общими разделами, запускаемых как единое целое.

Replication Controllers (replication-controller.md): replication controller гарантирует, что определенное количество «реплик» pod'ы будут запущены в любой момент времени.

Services (services.md): Сервис в Kubernetes это абстракция, которая определяет логический объединённый набор pod и политику доступа к ним.

Volumes (volumes.md): Volume(раздел) это директория, возможно, с данными в ней, которая доступна в контейнере.

Labels (labels.md): Label'ы это пары ключ/значение которые прикрепляются к объектам, например pod'ам. Label'ы могут быть использованы для создания и выбора наборов объектов.

Kubectl Command Line Interface (kubectl.md): kubectl интерфейс командной строки для управления Kubernetes.

Работающий кластер Kubernetes включает в себя агента, запущенного на нодах (kubelet) и компоненты мастера (APIs, scheduler, etc), поверх решения с распределённым хранилищем.

На каждой ноде запускается Docker. Docker обеспечивает загрузку образов и запуск контейнеров.

Kubelet управляет pod'ами их контейнерами, образами, разделами.

Выполнение работы

Развернуть кластер Kubernetes. Создать несколько докер контейнеров. Можно взять, например, созданные контейнеры в предыдущих работах (например, где контейнеры передавали друг-другу данные) или любые другие контейнеры из других работ. Можно создать и новые контейнеры. Запустить докер-контейнеры внутри кластера Kubernetes.

В качестве контейнеров были использованы контейнеры из работы 9.

Результат представлен на рисунке 1 и листингах 1, 2

Листинг 1 – код container1-manifest.yaml

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: container1-deployment  spec:  selector:  matchLabels:  app: container1  replicas: 1  template:  metadata:  labels:  app: container1  spec:  containers:  - name: container1  image: docker.io/autumnleafff/pract10:v1.0  ports:  - containerPort: 8000  ---  apiVersion: v1  kind: Service  metadata:  name: container1-service  spec:  selector:  app: container1  ports:  - protocol: TCP  port: 8000  targetPort: 8000  type: ClusterIP |

Листинг 2 – код container2-manifest.yaml

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: container2-deployment  spec:  selector:  matchLabels:  app: container2  replicas: 1  template:  metadata:  labels:  app: container2  spec:  containers:  - name: container2  image: docker.io/autumnleafff/pract9:v1.0  ports:  - containerPort: 8001  ---  apiVersion: v1  kind: Service  metadata:  name: container2-service  spec:  selector:  app: container2  ports:  - protocol: TCP  port: 8001  targetPort: 8001  type: ClusterIP |

Листинг 3 - вводимые команды для проброса

|  |
| --- |
| kubectl apply -f container2-manifest.yaml  kubectl apply -f container1-manifest.yaml  kubectl port-forward service/container1-service 8000:8000 &  Forwarding from 127.0.0.1:8000 -> 8000 |

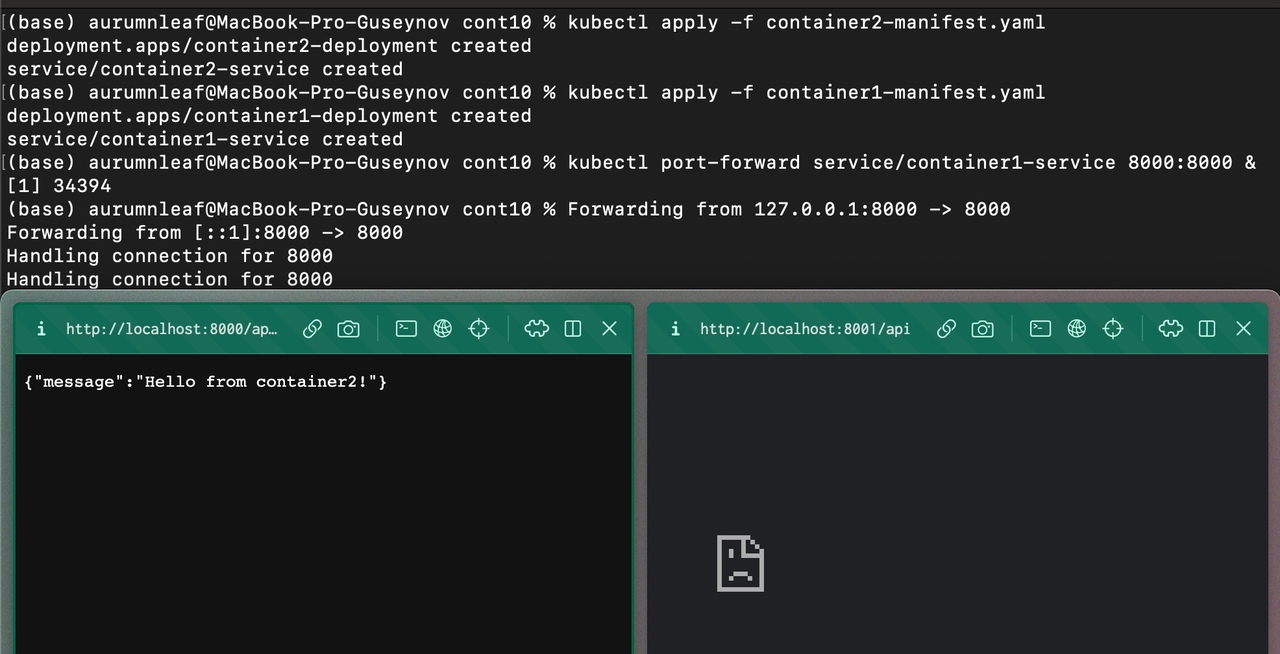


Рисунок 1 – Результат проброса к порту 8000 сервиса контейнер 1

**ВЫВОД**

В ходе данной работы был получен опыт развертывания кластера Kubernetes и разворачивания докер контейнеров в нем.