

## Теоретическое введение

Kubernetes — это система с открытым исходным кодом для развертывания, масштабирования и управления контейнеризованными приложениями.

Kubernetes, по сути, является не просто системой оркестрации. Технически оркестрация является выполнением определенного рабочего процесса: сначала сделай А, затем В, затем С.

Kubernetes же устраняет прямую необходимость в этом. В нем есть процессы управления, которые независимы и компонуемы. Главная задача процессов управления перевести текущее состояние к нужному следующему состоянию. Теперь нам неважно какой будет маршрут от А до С, что исключает централизованный контроль.

Благодаря этому система теперь более простая в использовании, мощная, надежная, а также устойчивая и расширяемая.

Kubernetes предоставляет:

- Быструю и автоматическую масштабируемость. При росте нагрузки можно быстро добавить необходимые узлы приложения, а также быстро их вывести, чтобы не тратить драгоценные ресурсы.

- Гибкий подход к эксплуатации. Мы можем быстро и легко построить структуру приложения, так как вся структура описывается в конфигурационных файлах – манифестах.

- Гибкий подход в управлении. Kubernetes не потребует перестройки инфраструктуры и прочего, если вы захотели провести тестирование, внедрить новый сервис или сделать деплой по методологии blue-green.

- Универсальность. С помощью манифестов легко переехать, если вы захотели поменять провайдера или переезжаете в свой собственный кластер.

— Низкий порог вхождения в использование. Kubernetes довольно легок в освоении манифестов, потому что большую часть работы он делает за вас.

Сам кластер K8S состоит из рабочих узлов. В узлах или нодах (Nodes, Worker nodes), помимо контейнеров компонентов самого кластера, размещаются контейнеры наших проектов и сервисов.

Minikube — это инструмент, позволяющий легко запускать Kubernetes на локальной машине. Для тестирования Kubernetes на локальной машине это является хорошим вариантом, потому что он запускает одноузловой кластер Kubernetes внутри виртуальной машины (VM) на компьютере пользователя.

Kubectl — это инструмент командной строки для управления кластерами Kubernetes. kubectl ищет файл config в директории \$HOME/.kube. Вы можете указать другие файлы kubeconfig, установив переменную окружения KUBECONFIG или флаг --kubeconfig.

## Полезные ссылки

1 Установка Kubernetes с помощью Minikube — Текст: электронный [сайт]. — URL: <https://kubernetes.io/ru/docs/setup/learning-environment/minikube/>

2 K8S для начинающих. Первая часть — Текст: электронный [сайт]. — URL: <https://habr.com/ru/post/589415/>

3 Kubernetes или с чего начать, чтобы понять что это и зачем он нужен — Текст: электронный [сайт]. — URL: <https://habr.com/ru/company/otus/blog/537162/>

4 Основы Kubernetes — Текст: электронный [сайт]. — URL: <https://habr.com/ru/post/258443/>

## Практическая часть

Вам необходимо выполнить все указанные в задании пункты и отразить в отчете в формате снимков экрана.

Для начала работы необходимо установить и запустить minikube в соответствии с установленной ОС с официального сайта Kubernetes:

<https://kubernetes.io/ru/docs/tasks/tools/install-minikube/>

Необходимо создать deployment при помощи файла deployment.yaml используя локальный docker образ с сервером:

- название deployment: Фамилия-НомерГруппы (ivanov-ikbo-99-99)
- используемый образ: Фамилия-НомерГруппы-Образ (ivanov-ikbo-99-99-obraz)

Файл server.js:

```
var http = require('http');

var handleRequest = function(request, response) {
  console.log('Получен запрос на URL: ' + request.url);
  response.writeHead(200);
  response.end('Hello world!');
};

var www = http.createServer(handleRequest);
www.listen(8080);
```

Dockerfile:

```
FROM node:6.14.2
EXPOSE 8080
COPY server.js .
CMD [ "node", "server.js" ]
```

Необходимо посмотреть информацию о Deployment при помощи команды:

```
kubectl get deployments
```

Далее необходимо посмотреть информацию о поде при помощи команды:

```
kubectl get pods
```

После этого нужно посмотреть события кластера при помощи команды:

```
kubectl get events
```

Затем необходимо посмотреть kubectl конфигурацию при помощи команды:

```
kubectl config view
```

Потом нужно сделать под с deployment Фамилия-НомерГруппы доступным для публичной сети Интернет с помощью команды **kubectl expose:**

- сервис должен быть виден вне кластера;
- порт: 8080.

После чего необходимо посмотреть только что созданный сервис

```
kubectl get services
```

Далее            нужно            запустить            сервис            hello-node:

```
minikube service ivanov-ikbo-99-99
```

Затем требуется отобразить текущие поддерживаемые дополнения и включить дополнение, например ingress:

```
minikube addons enable ingress
```

После этого нужно посмотреть Pod и Service, которые вы только что создали и отключить ingress.

```
kubectl get pod,svc -n kube-system
```

```
minikube addons disable ingress
```

После отключения необходимо включить dashboard:

```
minikube dashboard
```

Далее откройте во вкладке deployments созданный под и опишите в отчете отображаемые параметры.

После выполнения освободите ресурсы созданного вами кластера и остановите Minikube.

```
kubectl delete service ivanov-ikbo-99-99  
kubectl delete deployment ivanov-ikbo-99-99
```

```
minikube stop
```

## Вопросы к практической работе

1. Назовите виды контроллеров в Kubernetes.
2. Как называется командная строка в Kubernetes?
3. Что такое под?
4. Назовите 2 типа ресурсов, из которых состоит кластер Kubernetes.
5. Чем Kubernetes отличается от Docker Swarm?

## Критерии оценки

- Показана работоспособность Minikube в консоли.
- Показана работоспособность Kubernetes Dashboard.
- Показана работоспособность запущенного узла.
- Сделан отчет с описанием и скриншотами выполненных заданий.

За выполнение данной практической работы можно максимально получить 2 балла.

Критерии на выставление 2 баллов:

— Соблюдены общие требования выполнения практических работ, представленные в документе “Требования к выполнению практических работ”.

— Показана работоспособность Minikube в консоли.

— Показана работоспособность Kubernetes Dashboard.

— Показана работоспособность запущенного узла.

— Сделан отчет с описанием и скриншотами выполненных заданий.

— Дан полный и развернутый ответ на все вопросы преподавателя, как по вопросам к практике, так и по дополнительным вопросам к выполненному заданию.

Критерии на выставление 1 балла:

— Соблюдены общие требования выполнения практических работ, представленные в документе “Требования к выполнению практических работ”.

— Показана работоспособность Minikube в консоли.

— Не показана работоспособность Kubernetes Dashboard.

— Показана работоспособность запущенного узла.

— Сделан отчет с описанием и скриншотами выполненных заданий.

— Дан полный и развернутый ответ на все вопросы преподавателя на вопросы к практической работе, но дополнительные вопросы остались не отвечены: студент не смог полностью описать и аргументированно устно

объяснить ход проделанной работы, все шаги, студент не может объяснить и описать используемые технологии.

Критерии на выставление 0 баллов:

— Не соблюдены общие требования выполнения практических работ, представленные в документе “Требования к выполнению практических работ”.

— Не показана работоспособность Minikube в консоли.

— Не показана работоспособность Kubernetes Dashboard.

— Не показана работоспособность запущенного узла.

— Сделан отчет с описанием и скриншотами выполненных заданий.

— Студент не смог ответить ни на вопросы к практической работе, ни на вопросы к ходу выполнения работы.