

Практическое изучение средств контейнерной виртуализации и платформы Kubernetes

Д.А. Костюк, П.Н. Луцюк
Брестский государственный технический университет

- Программная платформа для контейнеризированных приложений
 - автоматизация развёртывания, масштабирования и управления
 - По умолчанию – в контейнерах Docker, но можно и в других (в т.ч. полноценных) VM
 - 1-й релиз в 2014 г.
 - Исходно – детище Google при активном участии Red Hat и др.
- Основной язык программирования системы — Go.
 - К счастью разработчиков приложений его знание не требуется

Применение

- Очень хорошо подходит для
 - систем с горизонтальным масштабированием
 - систем без сохранения состояния
- Популярен в качестве основы для микросервисов
- Используется как основа для разного рода кластеров
 - достижимость
 - Обнаружение
 - Самопочинка в случае проблем
- Нравится тем, кто не осилил особенности командной строки Docker

Специфика курса

- У платформы высокий порог вхождения
 - много слов, все что-то значат, одним словом кошмар
- Слабая адаптируемость к русскому языку
 - под всё это изобилие нет устоявшихся русских терминов
- Целевая аудитория - студенты старших курсов 1-й ступени
 - изучены компьютерные сети, элементы системного администрирования GNU/Linux, архитектуры клиент-серверных приложений, ...
- Студенты 2-й ступени тоже потом что-то изучают
 - там в Kubernetes еще много всего

ОСНОВНЫЕ ПОНЯТИЯ

- Master node
- Worker node
- Pod
- Replication Controller
- Service
- Label, Selector
- Deployment
- Volume
- Load balancer
- Secret
- Probe
- Namespace
- taints/tolerations
- affinity/antiaffinity
- Operator
- ...

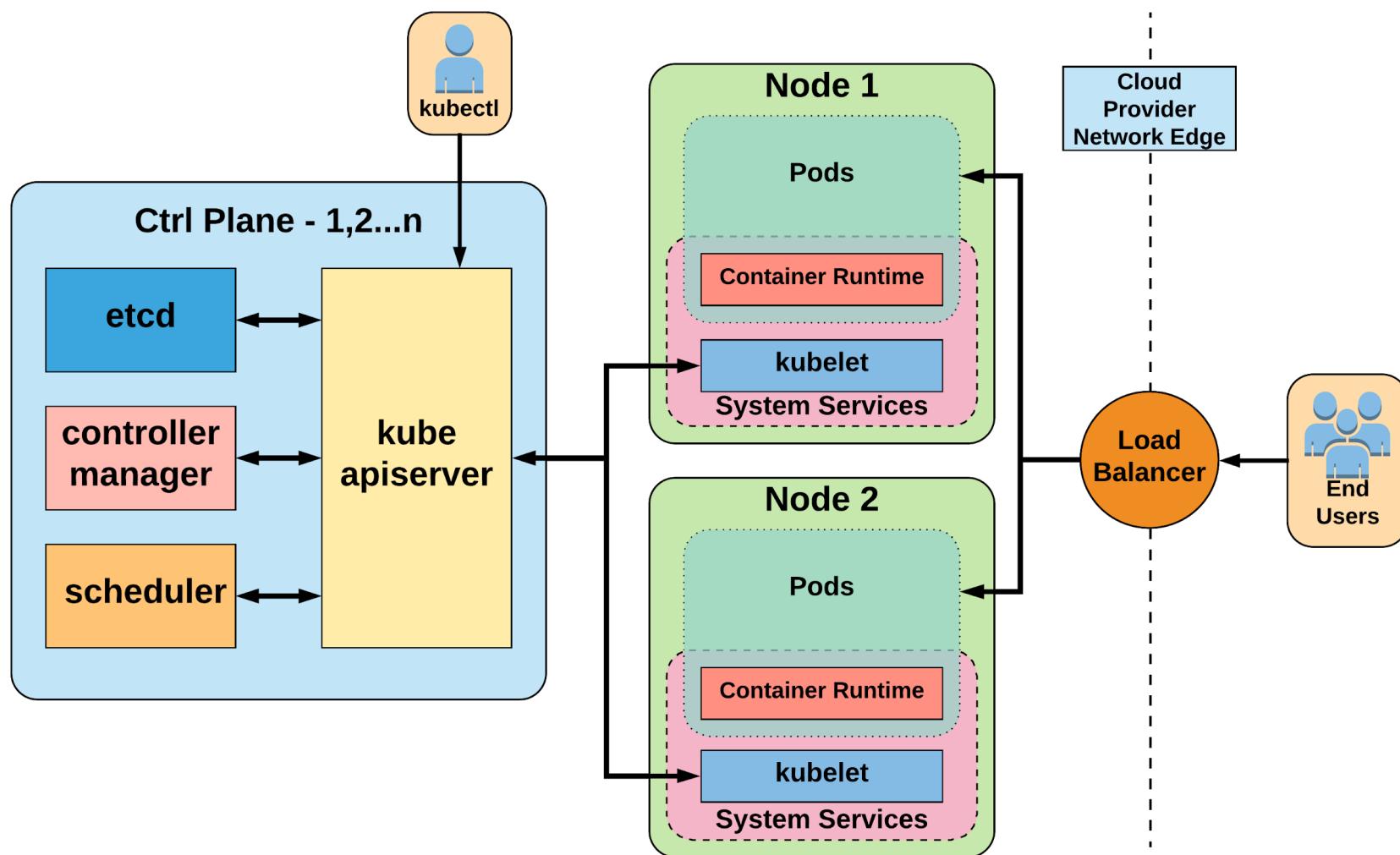
Совсем основные понятия

- Node - отдельная машина, на которой развёрнуты и выполняются контейнеры
- Pod - отдельная единица, которой оперирует шедулер для запуска какой-то работы
 - Содержит хотя бы 1 контейнер
- Replication Controller - старается поддерживать заданное число копий Пода
 - в соответствии с label selector
- Service - определяет, как те кому надо находят нужные им Поды

kube-apiserver

- Предоставляет прямой REST-интерфейс к Control Plane и хранилищу данных Kubernetes.
- Все клиенты и другие приложения взаимодействуют с Kubernetes строго через API-сервер.
- Играет роль “привратника” кластера: выполняет аутентификацию и авторизацию, валидирует запросы, контроль доступа, а еще работает фронтэндом хранилища.

Что работает на узлах



Примеры (YAML, командная строка)

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rs-example
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
      env: prod
  template:
    metadata:
      labels:
        app: nginx
        env: prod
    spec:
      containers:
        - name: nginx
          image: nginx:stable-alpine
          ports:
            - containerPort: 80
```

```
$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
rs-example-9l4dt    1/1     Running   0           1h
rs-example-b7bcg    1/1     Running   0           1h
rs-example-mk1l2    1/1     Running   0           1h
```

```
$ kubectl describe rs rs-example
Name:                rs-example
Namespace:           default
Selector:             app=nginx,env=prod
Labels:               app=nginx
                     env=prod
Annotations:          <none>
Replicas:             3 current / 3 desired
Pods Status:         3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:             app=nginx
                     env=prod
  Containers:
    nginx:
      Image:           nginx:stable-alpine
      Port:            80/TCP
      Environment:    <none>
      Mounts:          <none>
      Volumes:         <none>
Events:
  Type    Reason            Age   From                    Message
  ----    -
  Normal  SuccessfulCreate  16s   replicaset-controller   Created pod: rs-example-mk1l2
  Normal  SuccessfulCreate  16s   replicaset-controller   Created pod: rs-example-b7bcg
  Normal  SuccessfulCreate  16s   replicaset-controller   Created pod: rs-example-9l4dt
```

Kubernetes Dashboard

☰ **kubernetes** Workloads + CREATE

Admin

- Namespaces
- Nodes
- Persistent Volumes
- Storage Classes

Namespace

default ▾

Workloads

- Deployments
- Replica Sets
- Replication Controllers
- Daemon Sets
- Stateful Sets
- Jobs
- Pods

Services and discovery

- Services
- Ingresses

Deployments

Name	Labels	Pods	Age	Images	
✓ webserver	app: webserver version: alpine	3 / 3	-	nginx:alpine	⋮

Replica Sets

Name	Labels	Pods	Age	Images	
✓ webserver-3101375161	app: webserver pod-template-hash: 310137... version: alpine	3 / 3	-	nginx:alpine	⋮

Pods

Name	Status	Restarts	Age	
✓ webserver-3101375161-jzk57	Running	0	-	☰ ⋮
✓ webserver-3101375161-vxw2g	Running	0	-	☰ ⋮
✓ webserver-3101375161-w1flz	Running	0	-	☰ ⋮

Лабораторный практикум

- Программное обеспечение
 - GNU/Linux
 - VirtualBox
 - Minikube
 - Собственно больше ничего :)

Возможности и ограничения Minikube

- Поддерживается большая часть возможностей Kubernetes
 - Доступ к сервисам через NodePort
 - Монтируемые тома (Persistent Volumes)
 - Ingress
 - Контроль доступа на основе ролей (RBAC)
 - Упрощенный запуск Dashboard
 - ...
- Не поддерживается функционал, требующий наличия нескольких узлов
 - affinity/anti-affinity
 - ...

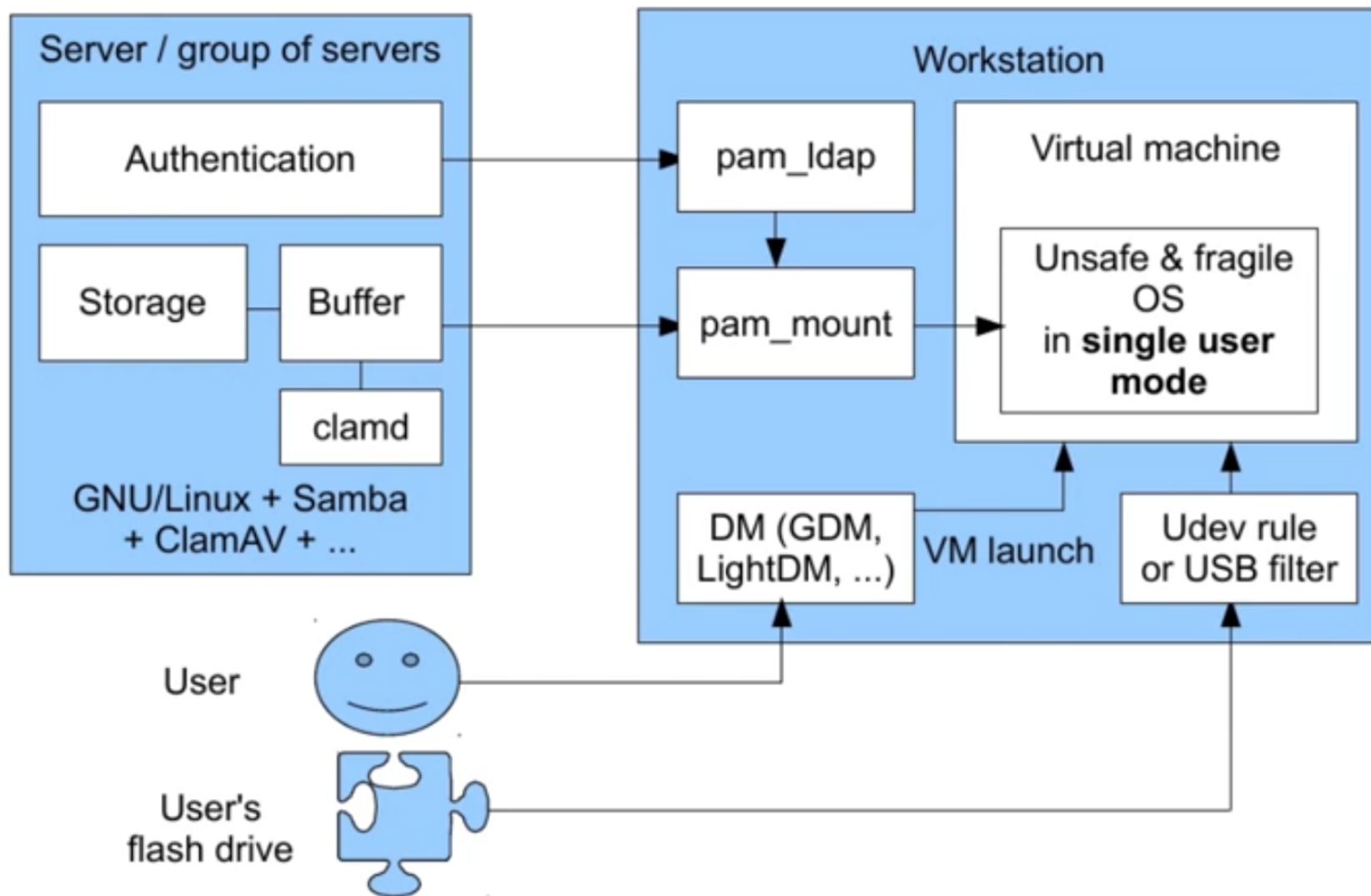
Структура практикума

- Знакомство Minikube
 - системы виртуализации
 - установка
- Доступ к Minikube
 - Использование kubectl
 - Использование kubectl proxy
 - Kubernetes Dashboard
- Развертывание отдельного приложения
 - Поды, ReplicaSet, Deployment
 - Service с доступом через NodePort
- Развертывание многокомпонентного приложения...
 - ...из готовых backend и frontend

Kubernetes без внешнего трафика

- Зачем нужен внешний трафик:
 - Minikube скачивает установочный образ
 - с Docker и компонентами Kubernetes
 - Скачиваются kubelet и kubectl
 - Docker скачивает контейнеры развёртываемого в кластере приложения
- Способы без этого обойтись
 - Использовать приватный реестр Docker и приватный репозиторий для образов Minikube
 - Использовать клонирование окружения с нужными образами в локальном кэше

Почему нам проще клонирование образа



Клонирование

- запуск Minikube и пробное развёртывание учебных приложений
- клонирование полученного профиля на рабочие станции учебного класса
- стандартная политика обновления контейнеров Kubernetes
 - IfNotPresent - скачивать образ контейнера в случае его отсутствия в локальном кэше
 - Never — никогда не скачивать
 - понадобилось чтобы обеспечить полную воспроизводимость использования локальных образов без обновления

Ссылки

- <https://ru.wikipedia.org/wiki/Kubernetes> -
вводная информация
- <https://kubernetes.io/docs/home/> –
официальная документация по Kubernetes
- <https://minikube.sigs.k8s.io/docs/> –
официальная документация по Minikube