

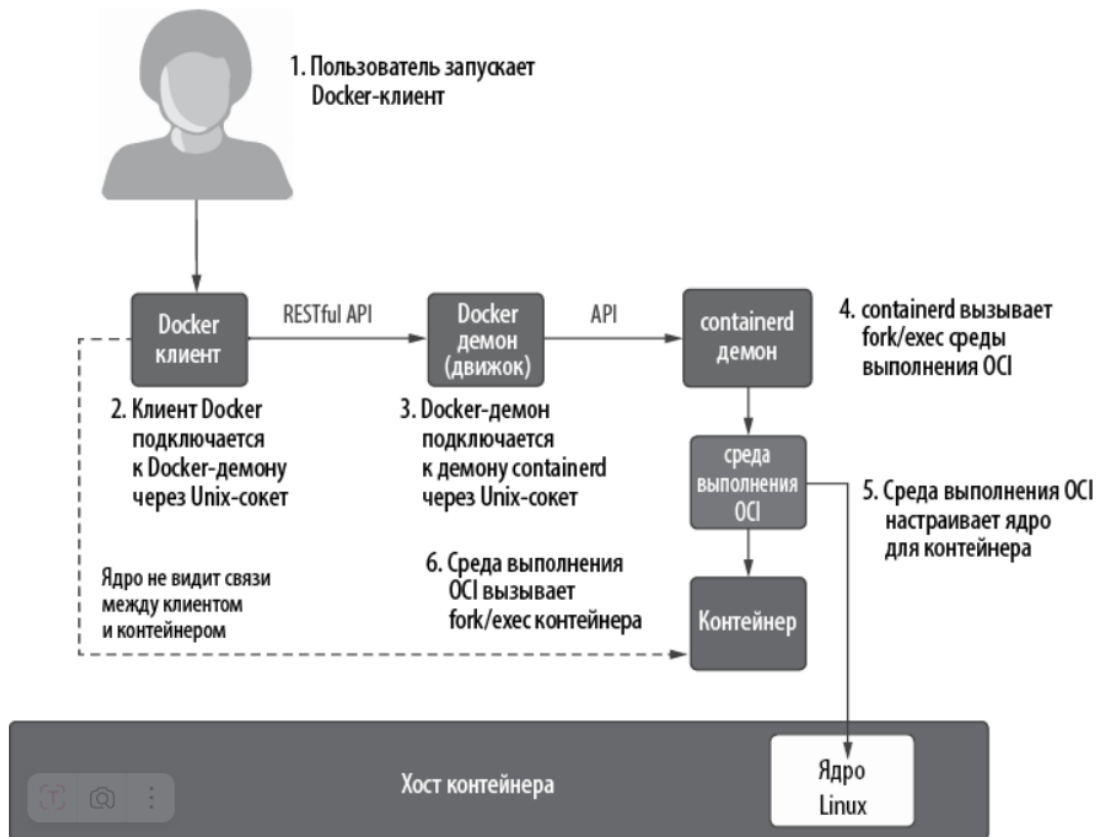


ІІТМО

**Использование Pod'ов в Podman
и средства взаимодействия с
podman**

Артур Валиуллин
Леся Бондаренко

Клиент-серверная архитектура Docker



Архитектура fork/exec в Podman

Как Podman запускает контейнеры



* Источник:

Дэниэл Уолш Podman в действии. — СПб.: Питер, 2024.

Контейнера Astra Linux 1.8



```
podman run --rm -it registry.astralinux.ru/library/astra/ubi18:1.8.1 /bin/bash
```

Источники:

1 . <https://registry.astralinux.ru/browse/library/>



Pod в Podman

ІТМО

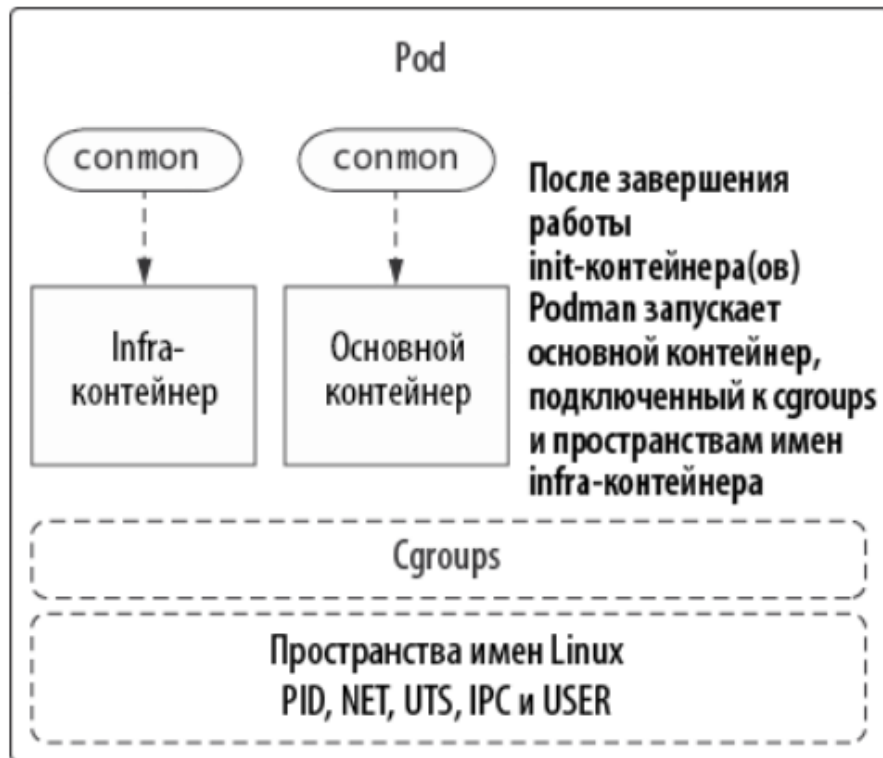
Podman позволяет создавать и управлять подами, аналогично Kubernetes.



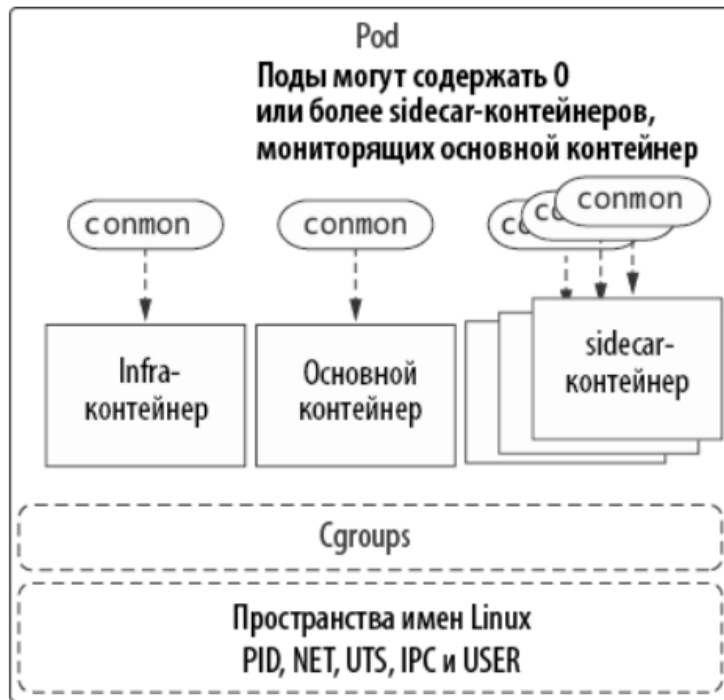
Pod — это единица развертывания, включающая один или несколько контейнеров, взаимодействующих друг с другом.



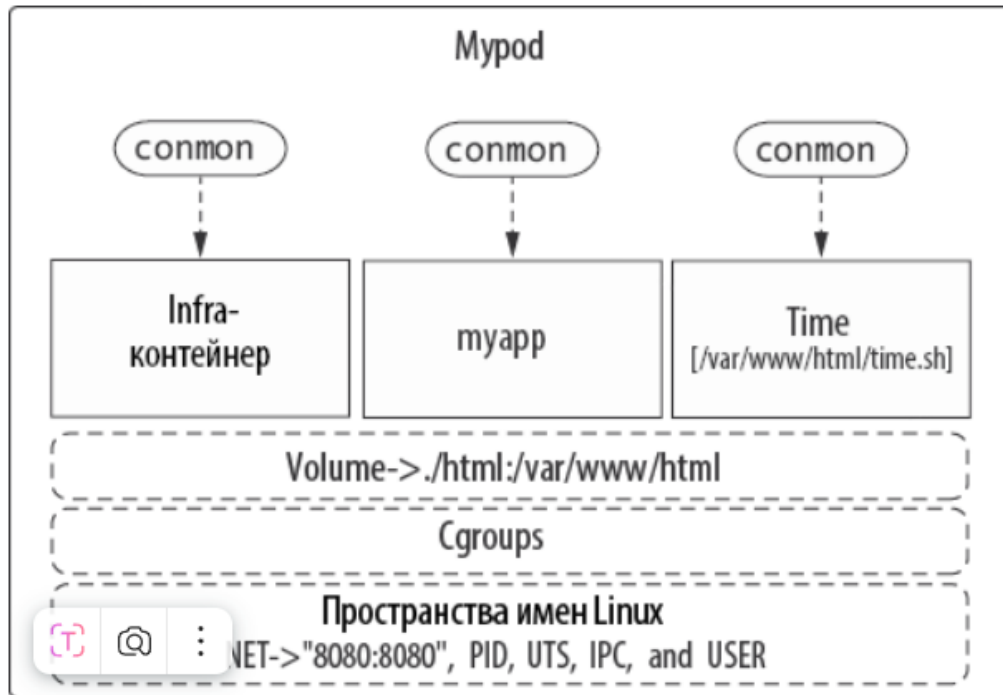
Pod в Podman



Pod в Podman



Пример



Пример:

<https://github.com/arvaliullin/pod-demo.git>

Пример

1. Создадим поду

```
podman pod create -p 8080:80 --name mypod --volume ./html:/usr/share/nginx/html:z
```

2. Добавим контейнеры в поду

```
podman create --pod mypod --name myapp nginx
```

```
podman create --pod mypod --name time --workdir /usr/share/nginx/html ubi8 ./time.sh
```

3. Запустим поду

```
podman pod start mypod
```

4. Запуск поды из mypod.yaml

```
podman play kube mypod.yaml
```

Sidcar-контейнер с именем time



```
1  #!/bin/sh
2
3  data() {
4      echo "<html><head></head><body><h1>"; date; echo "Hello World </h1></body></html>"
5      sleep 1
6  }
7
8  while true; do
9      data > index.html
10 done
11
```

mypod.yaml

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   annotations:
5     bind-mount-options: /mnt/w/projects/itmo/pyweb2024/pod-demo/html:z
6     creationTimestamp: "2024-11-13T21:00:18Z"
7   labels:
8     app: mypod
9     name: mypod
10 spec:
11   containers:
12   - args:
13     - nginx
14     - -g
15     - daemon off;
16     image: docker.io/library/nginx:latest
17     name: myapp
18     ports:
19     - containerPort: 80
20       hostPort: 8080
21     volumeMounts:
22     - mountPath: /usr/share/nginx/html
23       name: mnt-w-projects-itmo-pyweb2024-pod-demo-html-host-0
24   - command:
25     - ./time.sh
26     image: registry.access.redhat.com/ubi8:latest
27     name: time
28     volumeMounts:
29     - mountPath: /usr/share/nginx/html
30       name: mnt-w-projects-itmo-pyweb2024-pod-demo-html-host-0
31     workingDir: /usr/share/nginx/html
32   volumes:
33   - hostPath:
34     path: /mnt/w/projects/itmo/pyweb2024/pod-demo/html
35     type: Directory
36     name: mnt-w-projects-itmo-pyweb2024-pod-demo-html-host-0
37
```

С помощью команды сгенерируем **mypod.yaml**
podman generate kube mypod > mypod.yaml

PS W:\projects\itmo\pyweb2024\pod-demo>
podman play kube mypod.yaml

Pod:

6d07fa060f76e480a21b149179d6b47857d517db
109d8462d86ada2f354b99bc

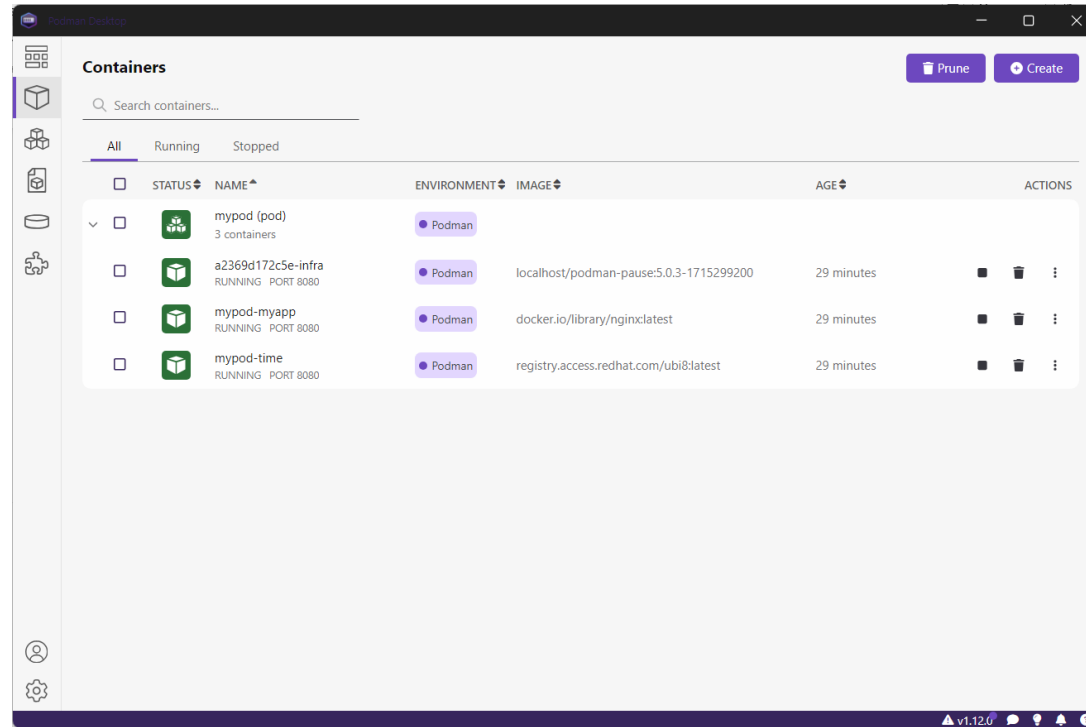
Containers:

8cda503527847fed852140a2fb2cc51d18b2accd
1966fc2a3baf4ee218e67e4b
89138992c8aecc273fe25b3f85ac580f1f3ac7be
345b70a0920f8208cf8b1699



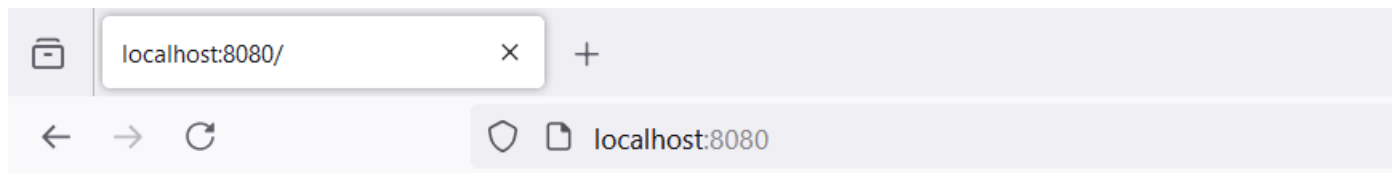
Podman Desktop

VITMO



Результат работы

ИТМО



Wed Nov 13 21:37:19 UTC 2024 Hello World



Podman & systemd



Чтобы запустить systemd в контейнере, Podman должен установить определённые монтирования, необходимые для корректной работы systemd.

Например, tmpfs монтируется в /run, /run/lock, /tmp и /var/log/journald. Также необходимо настроить /sys/fs/cgroup, что зависит от режима cgroup.

Podman делает всё это автоматически, если точкой входа в контейнер является /usr/sbin/init или /usr/sbin/systemd с параметром --systemd=always.

Использование **systemd** для запуска и работы контейнеров

ІІТМО

systemd запускает контейнер и управляет им на протяжении его жизненного цикла.

Команда **podman generate systemd** генерирует юнит-файл **systemd** для конкретного контейнера или пода.

Создание контейнера и службы

```
$ podman create -d --name foo busybox:latest top
54502f309f3092d32b4c496ef3d099b270b2af7b5464e7cb4887bc16a4d38597
$ podman generate systemd --name foo

[Unit]
Description=Podman container-foo.service
Documentation=man:podman-generate-systemd(1)

[Service]
Restart=on-failure
ExecStart=/usr/bin/podman start foo
ExecStop=/usr/bin/podman stop -t 10 foo
KillMode=none
Type=forking
PIDFile=/run/user/1000/overlay-
containers/54502f309f3092d32b4c496ef3d099b270b2af7b5464e7cb4887bc16a4d38597/userdata/common.pid

[Install]
WantedBy=multi-user.target
```


Файл сервиса, доступен не из рута

```
$ cat ~/.config/systemd/user/container.service
[Unit]
Description=Podman in Systemd

[Service]
Restart=on-failure
ExecStartPre=/usr/bin/rm -f /%t/%n-pid /%t/%n-cid
ExecStart=/usr/bin/podman run --common-pidfile /%t/%n-pid --cidfile /%t/%n-cid -d busybox:latest top
ExecStop=/usr/bin/sh -c "/usr/bin/podman rm -f `cat /%t/%n-cid`"
KillMode=none
Type=forking
PIDFile=%t/%n-pid

[Install]
WantedBy=multi-user.target
```

Запуск службы, проверка контейнера

```
$ systemctl --user daemon-reload
$ systemctl --user start container.service
$ systemctl --user status container.service
● container.service - Podman in Systemd
   Loaded: loaded (/home/valentin/.config/systemd/user/container.service; disabled; vendor preset:
   enabled)
   Active: active (running) since Mon 2019-11-18 15:32:56 CET; 1min 5s ago
     Process: 189705 ExecStartPre=/usr/bin/rm -f //run/user/1000/container.service-pid
//run/user/1000/container.service-cid (code=exited, status=0/SUCCESS)
     Process: 189706 ExecStart=/usr/bin/podman run --common-pidfile //run/user/1000/container.service-pid
--cidfile //run/user/1000/container.service-cid -d busybox:latest top (code=exited, status=0/SUCCESS)
    Main PID: 189731 (common)
      CGroup: /user.slice/user-1000.slice/user@1000.service/container.service
              └─189724 /usr/bin/fuse-overlayfs [...]
                 └─189726 /usr/bin/slirp4netns [...]
                    └─189731 /usr/bin/common [...]
                       └─189737 top

$ podman ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS          NAMES
f20988d59920   docker.io/library/busybox:latest    top                     12 seconds ago  Up 11 seconds ago  funny_zhukovsky
```

kill top PID и перезапуск



```
$ kill -9 189731
$ systemctl --user status container.service
● container.service - Podman in Systemd
   Loaded: loaded (/home/valentin/.config/systemd/user/container.service; disabled; vendor preset:
   enabled)
   Active: active (running) since Mon 2019-11-18 16:09:38 CET; 1min 3s ago [...]
   Main PID: 191263 (common)
```

kill top PID и перезапуск

```
from podman import PodmanClient
```

```
# Локальное подключение через Unix сокет
```

```
client = PodmanClient(base_url="unix:///run/podman/podman.sock")
```

```
# Подключение через TCP
```

```
# client = PodmanClient(base_url="http://localhost:8080")
```

Podman-ру: Получение списка контейнеров



```
containers = client.containers.list(all=True)
for container in containers:
    print(f"ID: {container.id}, Name: {container.name}, Status: {container.status}")
```

Podman-ру: Создание и запуск контейнера

```
container = client.containers.run(  
    image="nginx",  
    detach=True,  
    name="my-nginx",  
    ports={"8080/tcp": 80}  
)  
print(f"Контейнер запущен: ID {container.id}")
```

**Спасибо
за внимание!**

it'sMO *re than a*
UNIVERSITY

Артур Валиуллин
Леся Бондаренко