deckhouse
user community

meetup/3

Использование Open Source-виртуализации Deckhouse в небольших инсталляциях

Deckhouse
ФЛАНТ

Мария Бочарова

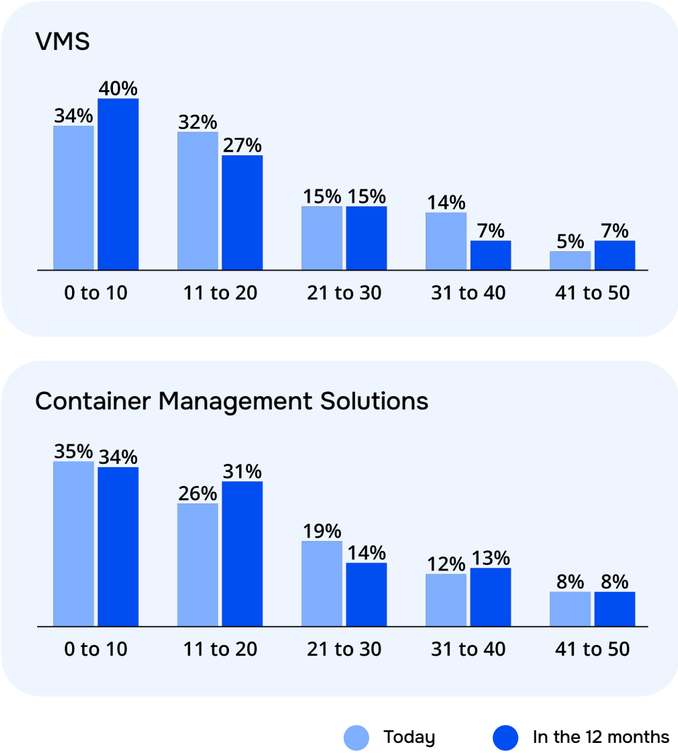
Специалист по решениям Deckhouse

Cloud Native-виртуализация — это неизбежность

Гибридные нагрузки — это **стандарт**, но управление ими стало сложнее.

Реальность: ВМ и контейнеры теперь сосуществуют, создавая операционный хаос. 2/3 организаций в мире имеют 11 и более независимых сред для ВМ и Kubernetes.

Standalone Instances Of VMs And Container Management Solutions Deployed At Organizations Today Versus In The Next 12 Months

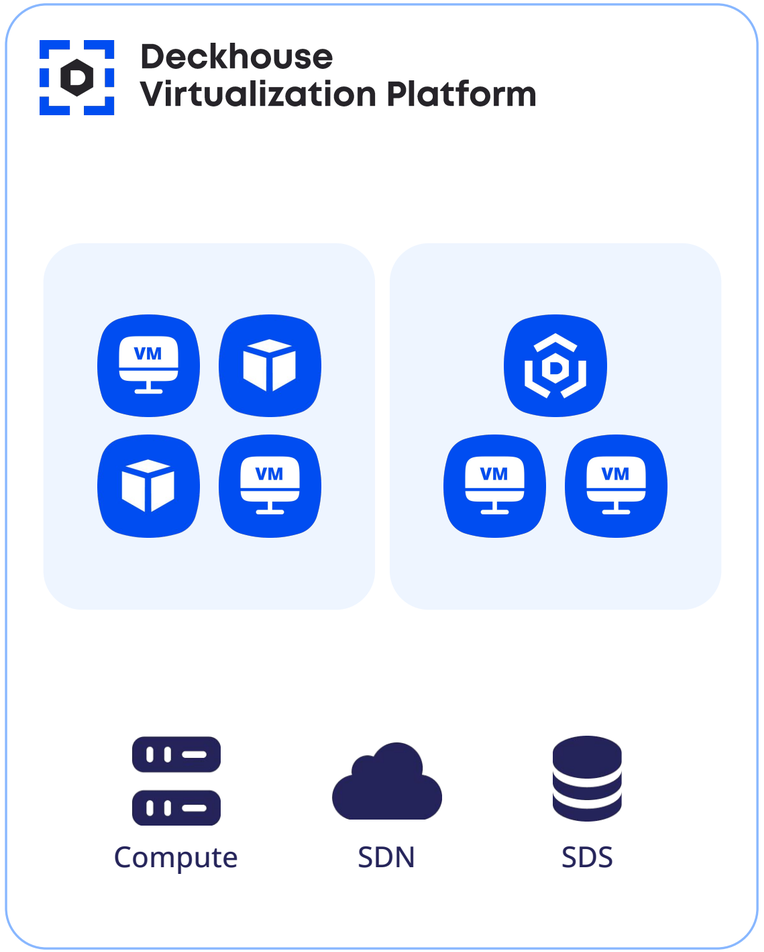


Base: 216 IT decision-makers from the US, EMEA, and APAC with influence over their organization's IT infrastructure strategy and architecture
Source: Forrester's Q2 2025 Cloud Management Solutions Survey [E-63311]

* Отчёт Forrester Modernize Or Fall Behind: Rethinking IT Infrastructure For A Competitive Edge

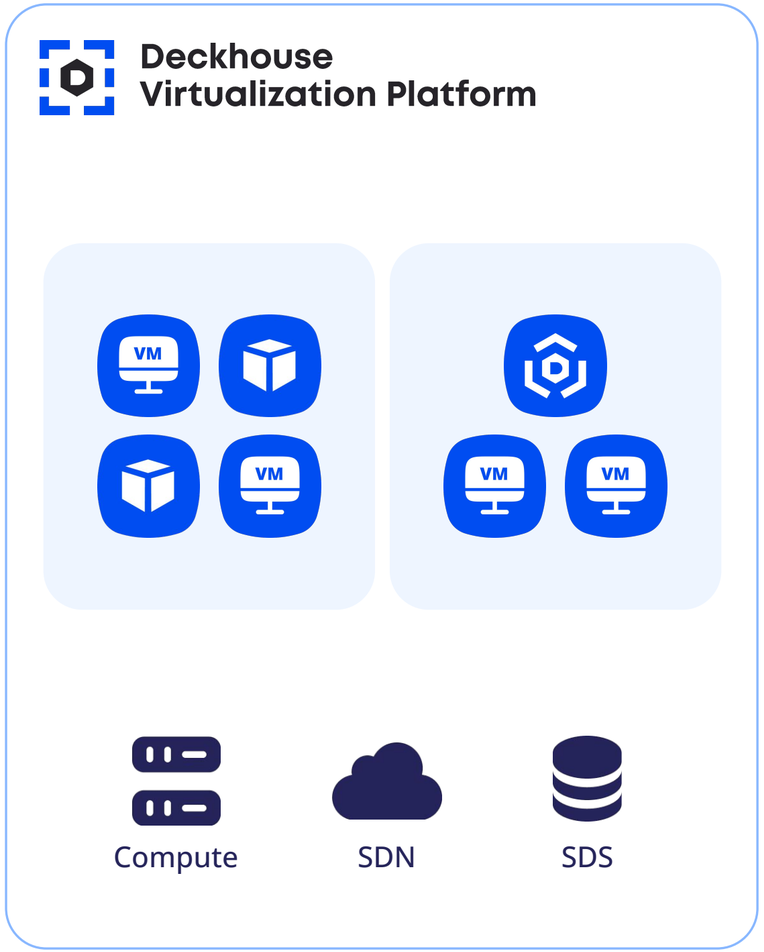
Преимущества платформы:

- **Полная декларативность** и GitOps
- **Живая миграция ВМ** между гипервизорами с разными CPU
- Изоляция и квотирование ресурсов **на уровне проектов**
- **Встроенный SDS** и микросегментация
- Мониторинг и логирование **«из коробки»**
- **Автоматизированное** управление сертификатами



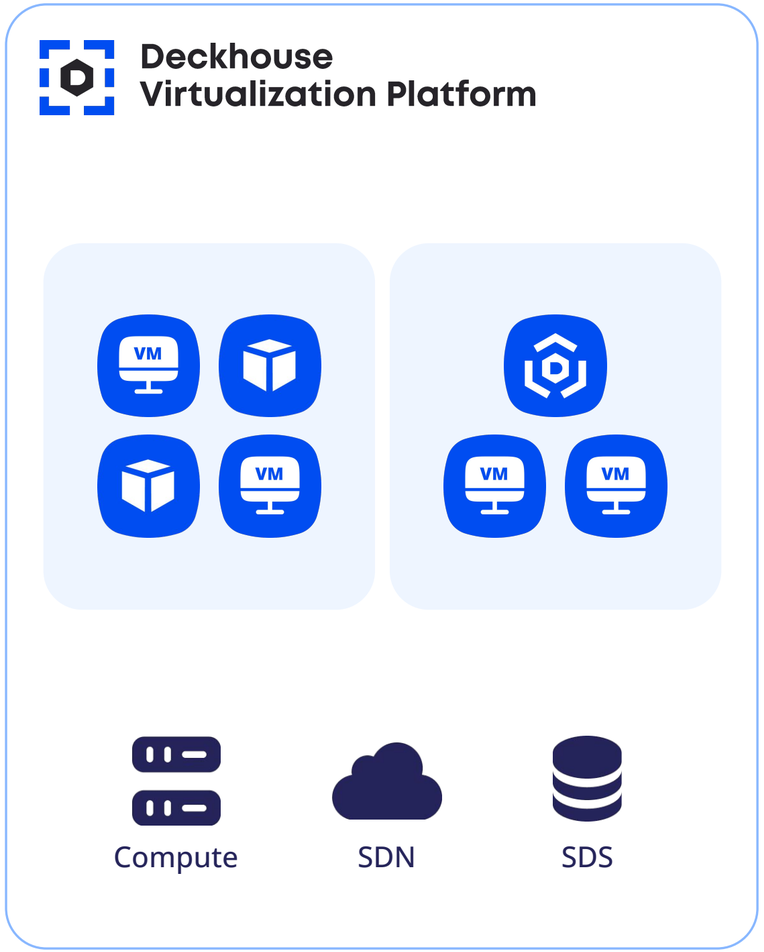
Killer features:

- **Единый Control Plane** для BM и Kubernetes:
  - BM и поды управляются одним оркестратором через общий API
  - Нет разделения на две независимые системы



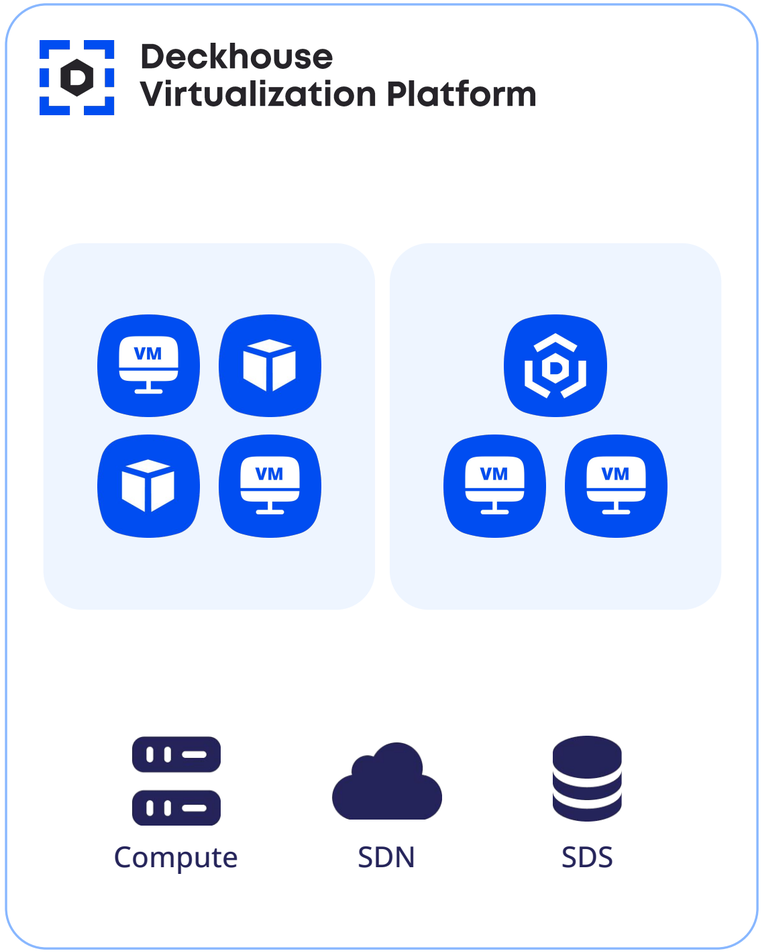
Killer features:

- **Единый Control Plane** для BM и Kubernetes
- **Конвергенция** операционных моделей:
  - Запуск stateful workload'ов любого типа:
    от legacy BM до Cloud Native-приложений в K8s.
  - Общие сети, хранилища и политики
    безопасности для всех типов нагрузок



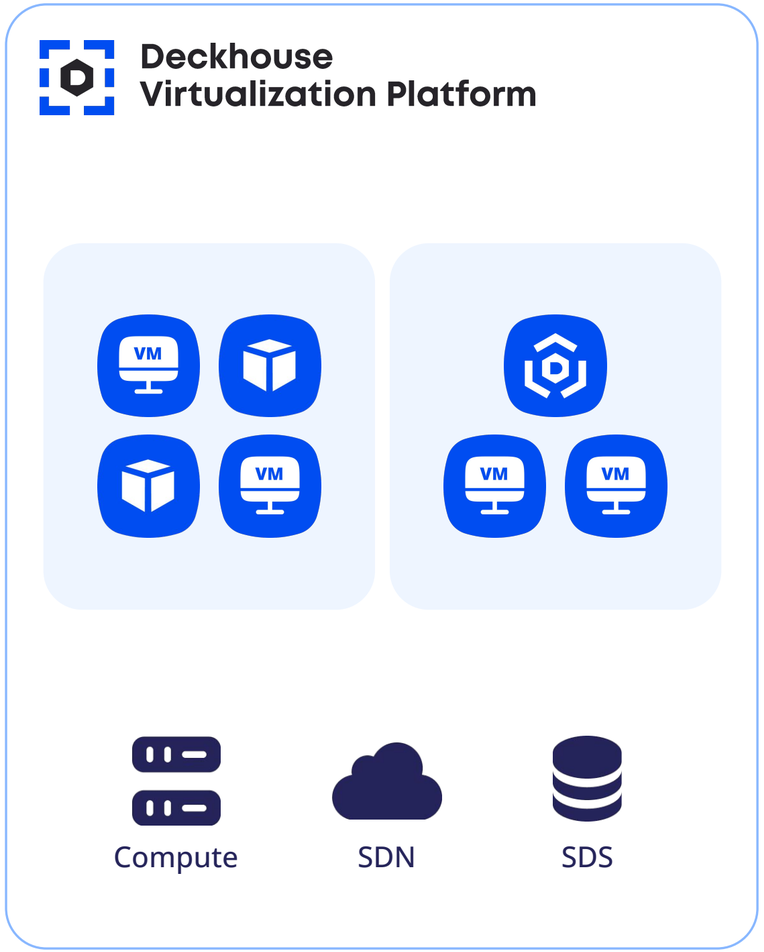
Killer features:

- **Единый Control Plane** для BM и Kubernetes
- **Конвергенция** операционных моделей
- Production-Grade K8s как **основа платформы:**
  - Не Kubernetes поверх виртуализации,
    а виртуализация как часть Kubernetes.
  - Использование всей экосистемы
    CNCF (Istio, Grafana, Prometheus)
    без дополнительной интеграции



Killer features:

- **Единый Control Plane** для BM и Kubernetes
- **Конвергенция** операционных моделей
- Production-Grade K8s как **основа платформы**



Сценарии использования

Тестируем опасное ПО

- Изолированная среда для исследования
- Быстрое восстановление состояния
- Сетевой анализ и мониторинг
- Автоматизация анализа

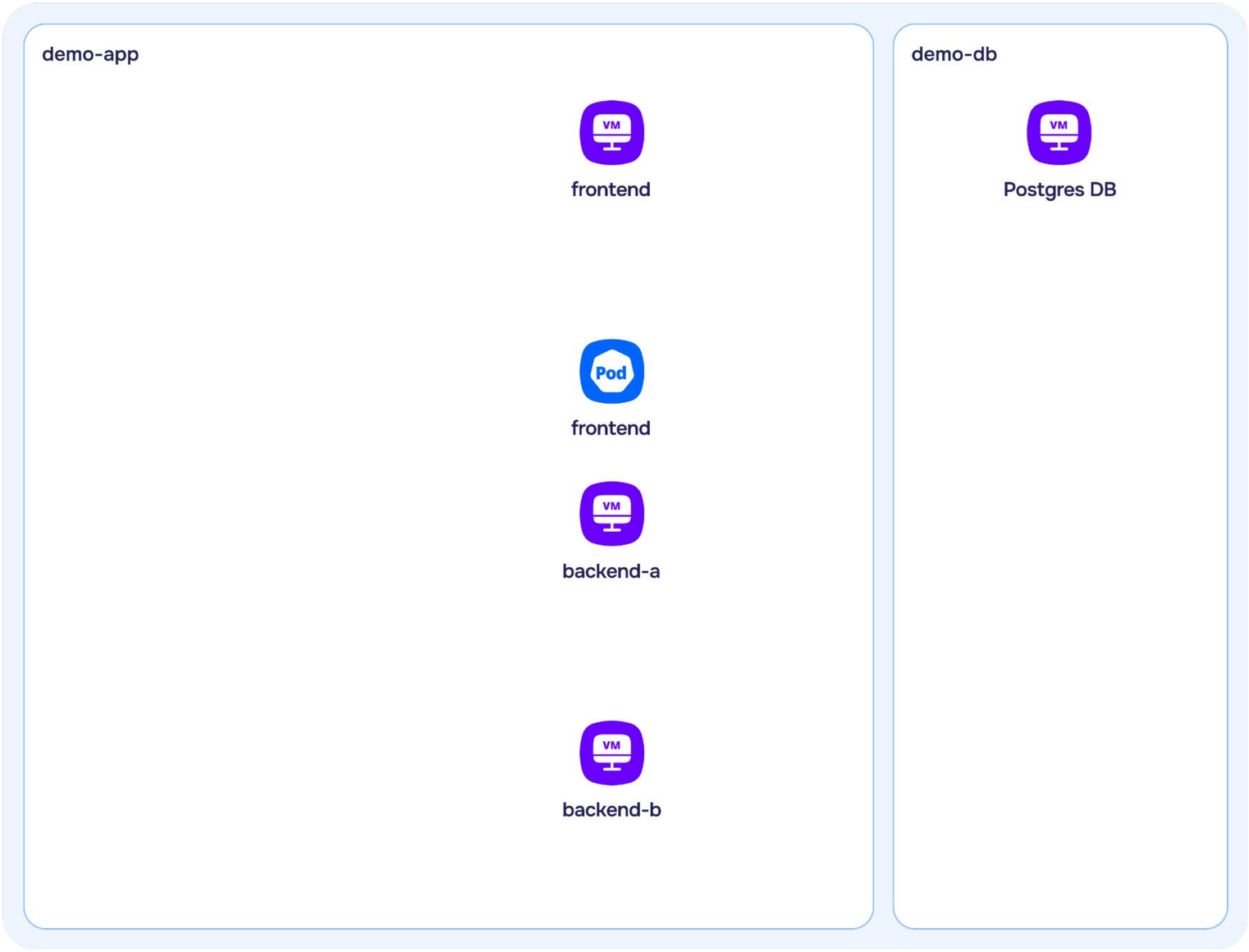
Персонализированные стенды

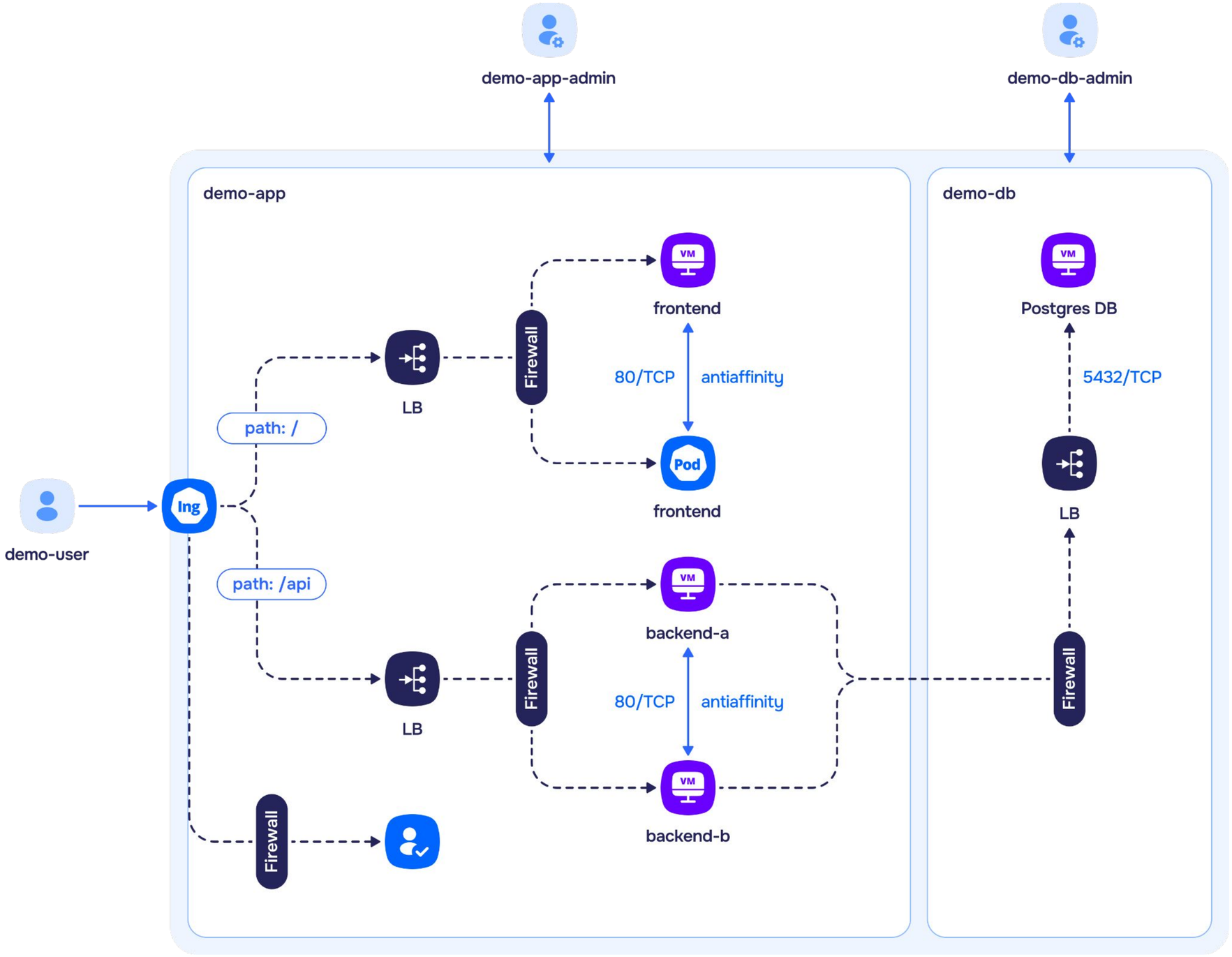
- Можно запускать специфичные ОС, эмулировать разные окружения
- Специализированные ВМ с большими ресурсами для компиляции мобильных приложений, прошивок, ML-моделей
- Запуск PostgreSQL, ClickHouse и других БД с кастомными настройками ОС и специфичными требованиями к I/O

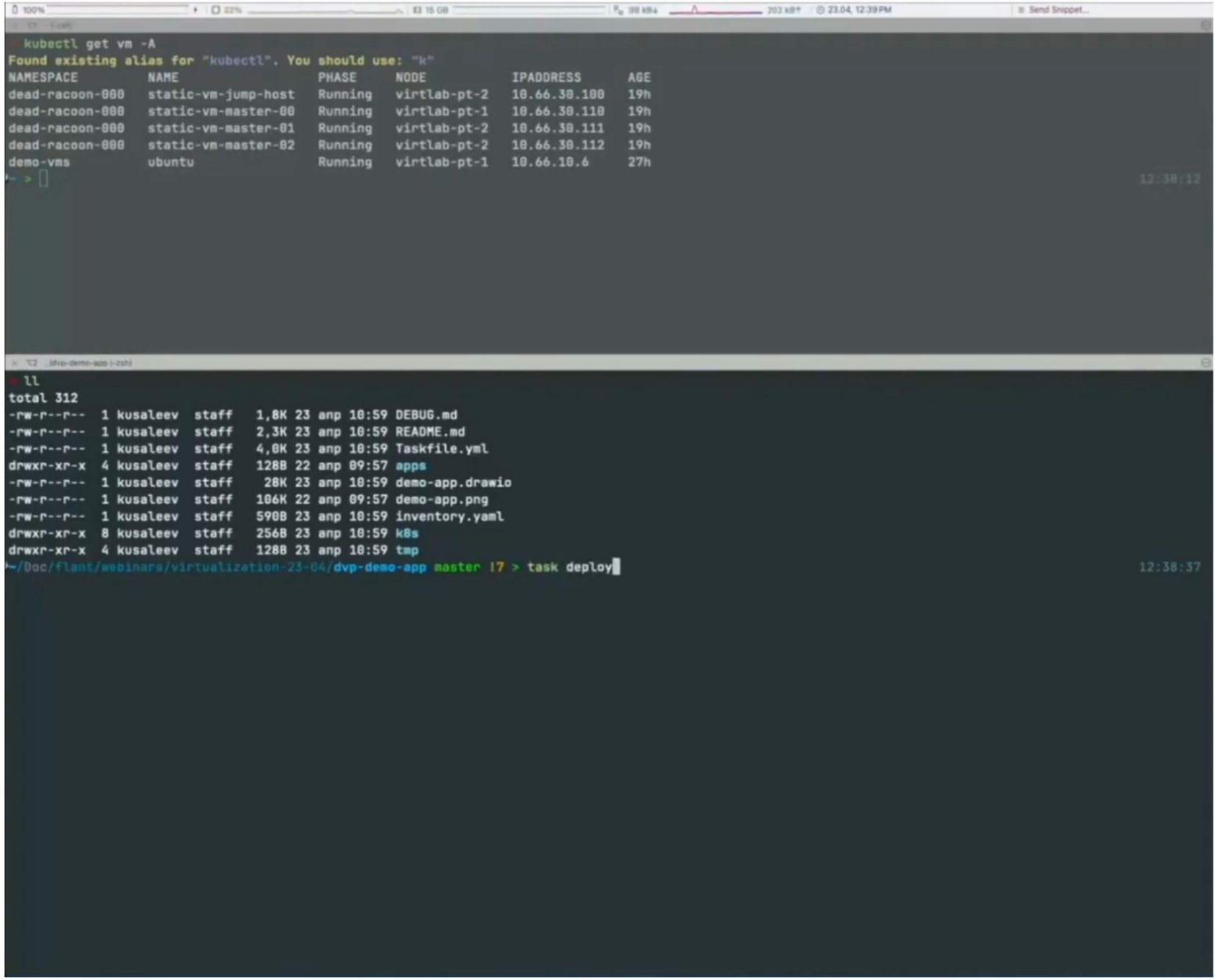
Enterprise-виртуализация дома

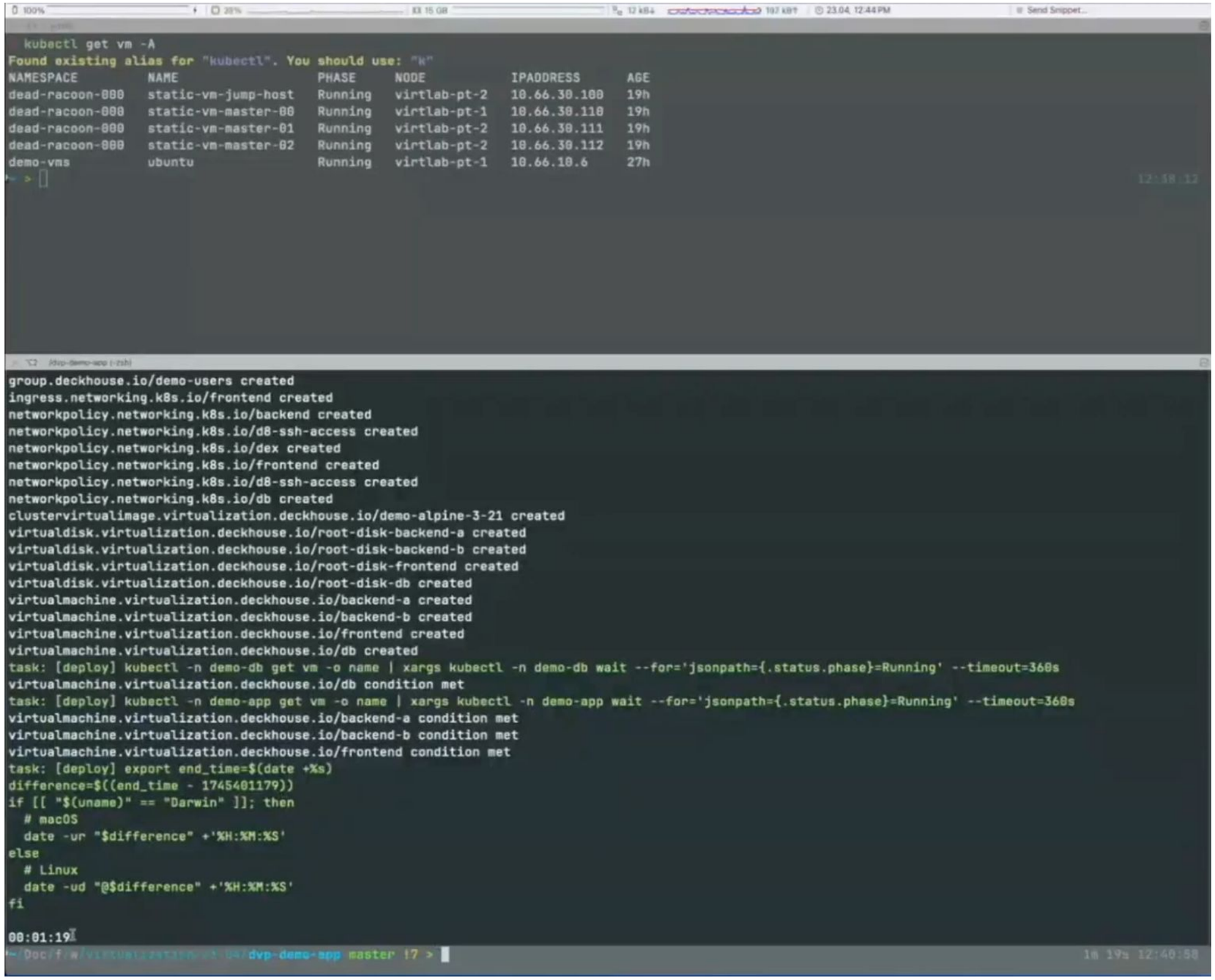
- Абстракция от железа
- Безопасность и изоляция
- Энергоэффективность
- Легко управлять ресурсами и запускать новые проекты

Пример развёртывания приложения

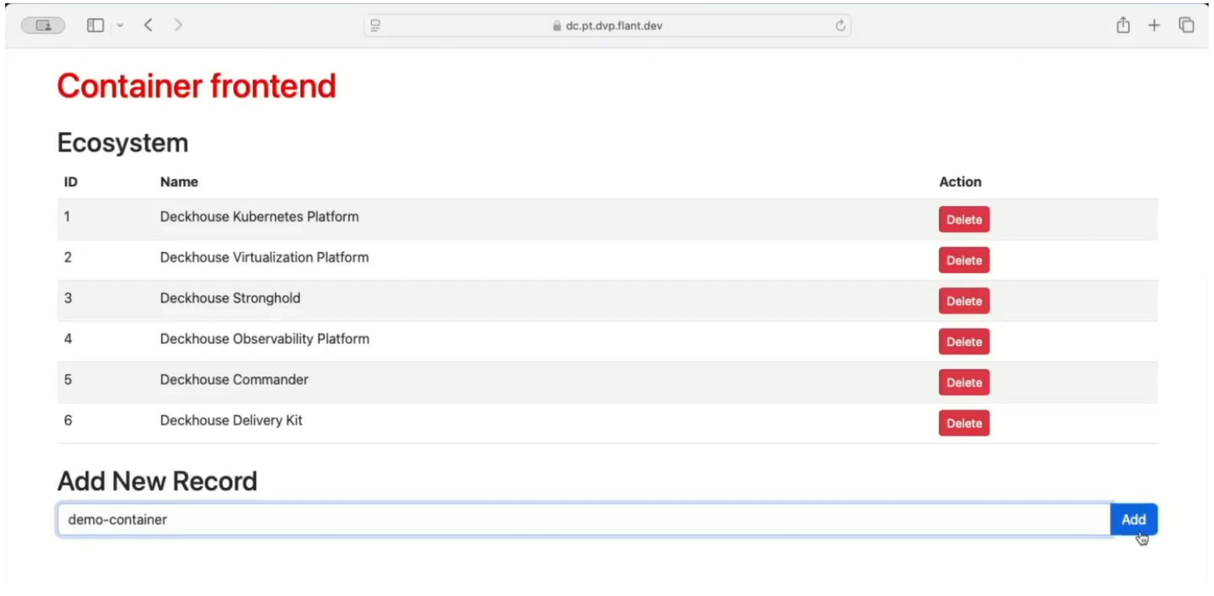

demo-app
VM
frontend
Pod
frontend
VM
backend-a
VM
backend-b
demo-db
VM
Postgres DB

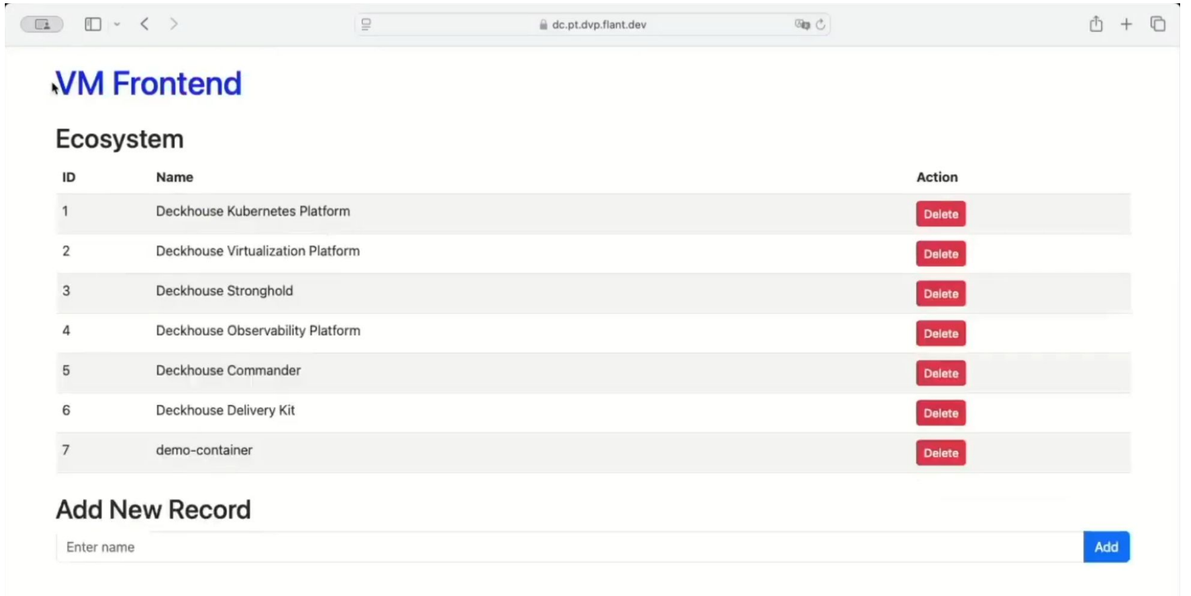

demo-app-admin
demo-db-admin
demo-app
demo-db
VM
frontend
Postgres DB
80/TCP
antiaffinity
5432/TCP
Firewall
LB
path: /
Pod
frontend
demo-user
Ing
path: /api
backend-a
backend-b
Firewall
LB
Firewall
80/TCP
antiaffinity
Firewall
LB
Firewall


kubectl get vm -A
Found existing alias for "kubectl". You should use: "k"
NAMESPACE NAME PHASE NODE IPADDRESS AGE
dead-racoon-000 static-vm-jump-host Running virtlab-pt-2 10.66.30.100 19h
dead-racoon-000 static-vm-master-00 Running virtlab-pt-1 10.66.30.110 19h
dead-racoon-000 static-vm-master-01 Running virtlab-pt-2 10.66.30.111 19h
dead-racoon-000 static-vm-master-02 Running virtlab-pt-2 10.66.30.112 19h
demo-vms ubuntu Running virtlab-pt-1 10.66.10.6 27h
12:38:12
ll
total 312
-rw-r--r-- 1 kusaleev staff 1,8K 23 апр 10:59 DEBUG.md
-rw-r--r-- 1 kusaleev staff 2,3K 23 апр 10:59 README.md
-rw-r--r-- 1 kusaleev staff 4,0K 23 апр 10:59 Taskfile.yml
drwxr-xr-x 4 kusaleev staff 128B 22 апр 09:57 apps
-rw-r--r-- 1 kusaleev staff 28K 23 апр 10:59 demo-app.drawio
-rw-r--r-- 1 kusaleev staff 106K 22 апр 09:57 demo-app.png
-rw-r--r-- 1 kusaleev staff 590B 23 апр 10:59 inventory.yaml
drwxr-xr-x 8 kusaleev staff 256B 23 апр 10:59 k8s
drwxr-xr-x 4 kusaleev staff 128B 23 апр 10:59 tmp
~/Doc/flant/webinars/virtualization-23-04/dvp-demo-app master !7 > task deploy
12:38:37


kubectl get vm -A
Found existing alias for "kubectl". You should use: "k"
NAMESPACE NAME PHASE NODE IPADDRESS AGE
dead-racoon-000 static-vm-jump-host Running virtlab-pt-2 10.66.30.100 19h
dead-racoon-000 static-vm-master-00 Running virtlab-pt-1 10.66.30.110 19h
dead-racoon-000 static-vm-master-01 Running virtlab-pt-2 10.66.30.111 19h
dead-racoon-000 static-vm-master-02 Running virtlab-pt-2 10.66.30.112 19h
demo-vms ubuntu Running virtlab-pt-1 10.66.10.6 27h
group.deckhouse.io/demo-users created
ingress.networking.k8s.io/frontend created
networkpolicy.networking.k8s.io/backend created
networkpolicy.networking.k8s.io/d8-ssh-access created
networkpolicy.networking.k8s.io/dex created
networkpolicy.networking.k8s.io/frontend created
networkpolicy.networking.k8s.io/d8-ssh-access created
networkpolicy.networking.k8s.io/db created
clustervirtualimage.virtualization.deckhouse.io/demo-alpine-3-21 created
virtualdisk.virtualization.deckhouse.io/root-disk-backend-a created
virtualdisk.virtualization.deckhouse.io/root-disk-backend-b created
virtualdisk.virtualization.deckhouse.io/root-disk-frontend created
virtualdisk.virtualization.deckhouse.io/root-disk-db created
virtualmachine.virtualization.deckhouse.io/backend-a created
virtualmachine.virtualization.deckhouse.io/backend-b created
virtualmachine.virtualization.deckhouse.io/frontend created
virtualmachine.virtualization.deckhouse.io/db created
task: [deploy] kubectl -n demo-db get vm -o name | xargs kubectl -n demo-db wait --for='jsonpath={.status.phase}=Running' --timeout=360s
virtualmachine.virtualization.deckhouse.io/db condition met
task: [deploy] kubectl -n demo-app get vm -o name | xargs kubectl -n demo-app wait --for='jsonpath={.status.phase}=Running' --timeout=360s
virtualmachine.virtualization.deckhouse.io/backend-a condition met
virtualmachine.virtualization.deckhouse.io/backend-b condition met
virtualmachine.virtualization.deckhouse.io/frontend condition met
task: [deploy] export end_time=$(date +%s)
difference=$((end_time - 1745401179))
if [["$(uname)" == "Darwin"]]; then
# macOS
date -ur "$difference" +'%H:%M:%S'
else
# Linux
date -ud "@$difference" +'%H:%M:%S'
fi
00:01:19
dvp-demo-app master !7 >

Полностью готовое приложение





Будущее за инженерами, которые умеют проектировать и управлять гибридными нагрузками. Присоединяйтесь!

Вы можете самостоятельно создать приложение из примера:

- Пройдите **бесплатный** тренинг «Установка Deckhouse Virtualization Platform»
- **Разверните DVP**, используя быстрый старт
- **Зайдите на GitHub** и склонируйте проект
- Прочитайте readme.md и запустите **task deploy**

Хотите попробовать?

deckhouse
user community

meetup/3

оцените доклад

Мария Бочарова

maria.bocharova@flant.ru

19:30 | следующий доклад