FLANT

Deckhouse
Kubernetes Platform

Кластер Kubernetes,
который просто работает

ФЛАНТ

О компании

Опыт

Работаем с 2008 года

Более 15 лет опыта с Linux

Более 5 лет в DevOps и Kubernetes
занимаемся Kubernetes с момента его появления

Нам доверяют

LEROY MERLIN
делимобиль
10 money
onlinetours
FESCO
ленточка
Альфа-Лизинг
ВТБ
Все инструменты.ру
vc.ru

Команда

170+ человек

10 DevOps-команд

2 команды R&D

АО «Флант» – лидер российского DevOps-рынка

Собственное R&D

Разрабатываем Open Source-инструменты
для экосистемы Kubernetes, получившие признание сообщества

werf
werf/werf

deckhouse
deckhouse/deckhouse

shell-operator
flant/shell-operator

Statusmap
flant/grafana-statusmap

addon-operator
flant/addon-operator

Что такое Deckhouse Kubernetes Platform?

FLANT
Deckhouse
Kubernetes Platform

Квинтэссенция многолетней работы с Linux, контейнерами и Kubernetes в продуктовых окружениях.

Предельно простая процедура установки K8s любую инфраструктуру с наиболее популярными дистрибутивами Linux: Ubuntu, CentOS, Debian-совместимые дистрибутивы

Набор всех необходимых компонентов для эксплуатации «боевого» кластера.

Возможность развертывания и поддержки кластеров в “закрытом контуре”.

Инструментарий для управления жизненным циклом кластеров

Встроенный мониторинг с продуманными метриками “из коробки”

“Вторые руки DevOps”: используя DeckHouse 2 инженера «Фланта» обслуживают более 220 кластеров наших заказчиков обеспечивая SLA.

Развертывание кластеров Kubernetes:

FLANT
Deckhouse
Kubernetes Platform

На “голом железе” или виртуальных машинах

В публичных и частных облаках

Поддержка наиболее популярных дистрибутивов Linux: Ubuntu, CentOS, Debian-совместимые дистрибутивы

Развертывание и поддержка кластеров в “закрытом контуре”, без доступа в Интернет

Предельно простая процедура развертывания: команда CLI и время от 8 минут!

100% ванильный Kubernetes с возможностью выбора версии

Управление жизненным циклом кластеров

FLANT
Deckhouse
Kubernetes Platform

Многоступенчатое тестирование релизов

Пять каналов обновлений с разными уровнями стабильности: alfa, beta, ea, stable, rock solid

Автоматическое обновление Kubernetes и всех компонентов с гибкими настройками

Возможность обновления по расписанию или вручную

Ключевые компоненты Deckhouse

Интегрированный и настроенный Load Balancer

Nginx Ingress + MetalLB

Мониторинг, выверенный на сотнях проектов

Prometheus и Grafana, которые позволяют с первого взгляда провести RCA

Автомасштабирование «во все стороны» по любым метрикам

HPA, VPA, Cluster Autoscaler

Автоуправление сертификатами SSL

Let’s Encrypt (cert-manager)

Сбор и отправка логов из кластера и приложений

На основе Vector

Service Mesh

На основе Istio

CNI Flannel и Cilium

CNI Cilium в режиме tech preview

Функции дополнительных компонентов

FLANT
Deckhouse
Kubernetes Platform

Доступ к компонентам кластера (таким как мониторинг, дашборд, статус-страница) через web-интерфейс

Управление SSL-сертификатами, включая их автоматический выпуск и обновление, для безопасного доступа к кластерным компонентам

Обеспечение сетевой связности между узлами кластера

Создание единого внутрикластерного пространства имён сервисов (DNS)

Предоставление доступа к кластерным ресурсам посредством защищенного сетевого соединения (VPN)

Балансировка входящего трафика между узлами кластера

Тонкая настройка маршрутизации запросов между сервисами в кластере

Мониторинг состояния всех компонентов кластера и уведомление оператора о нештатных ситуациях (алерты)

Расчет и отображение текущего уровня доступности всех компонентов кластера (SLA)

Управление вычислительными ресурсами кластера для нужд сервисных приложений, в том числе и высвобождение недоутилизированных ресурсов

Настройка единой системы аутентификации и авторизации для доступа к ресурсам кластера, управление ролями пользователей в кластере и обеспечение мультиарендного режима работы в кластере

Управление узлами в кластере

FLANT
Deckhouse
Kubernetes Platform

Возможность создания и управления группами узлов, в том числе под определенный вид нагрузки

Возможность систематического прерывания работы узлов (chaos monkey) с целью проверки отказоустойчивости

Установка/обновление и настройка ПО узла, подключение узла в кластер

Автомасштабирование кластера с заказом ресурсов для создания нового узла в поддерживаемых облачных средах

Управление пользовательской нагрузкой

FLANT
Deckhouse
Kubernetes Platform

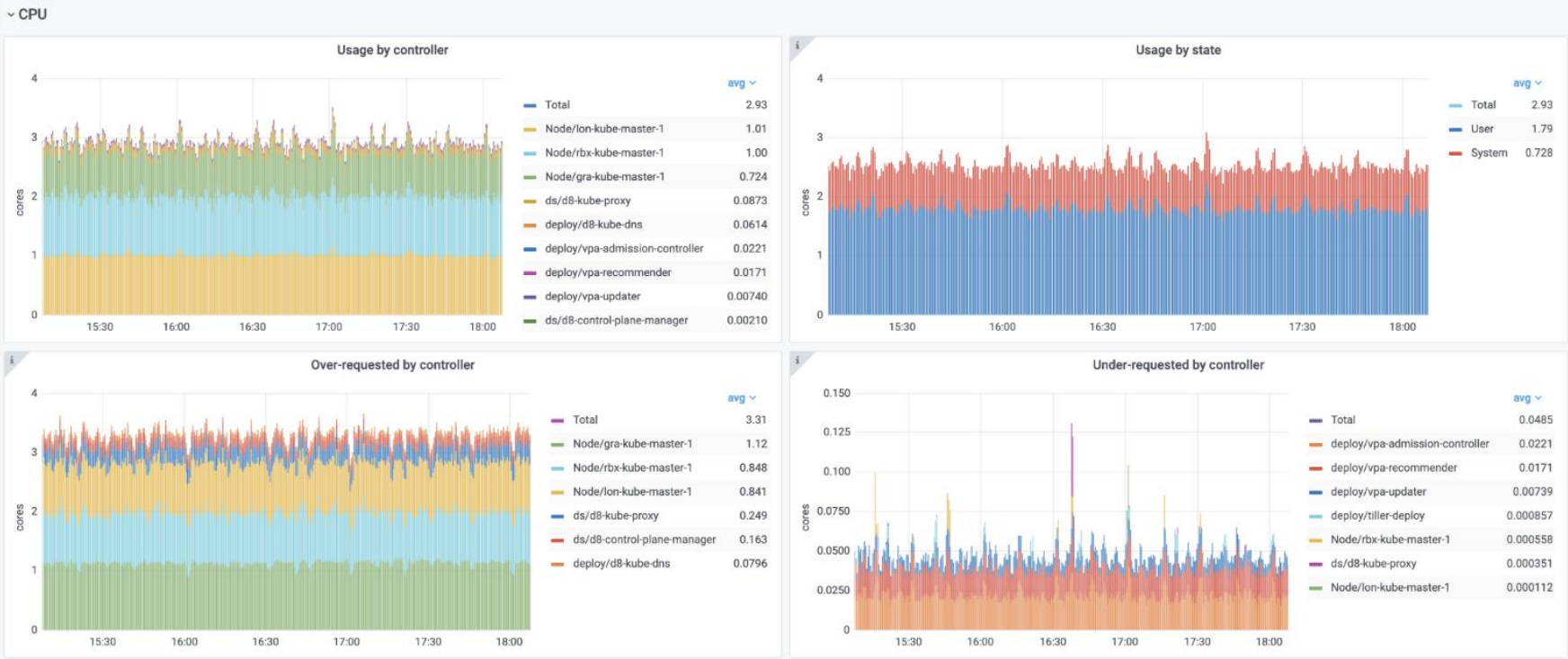
Вертикальное и горизонтальное масштабирование по любым скалярным метрикам Prometheus

Развитая система мониторинга с продуманными графиками и алертами “из коробки” на базе Prometheus

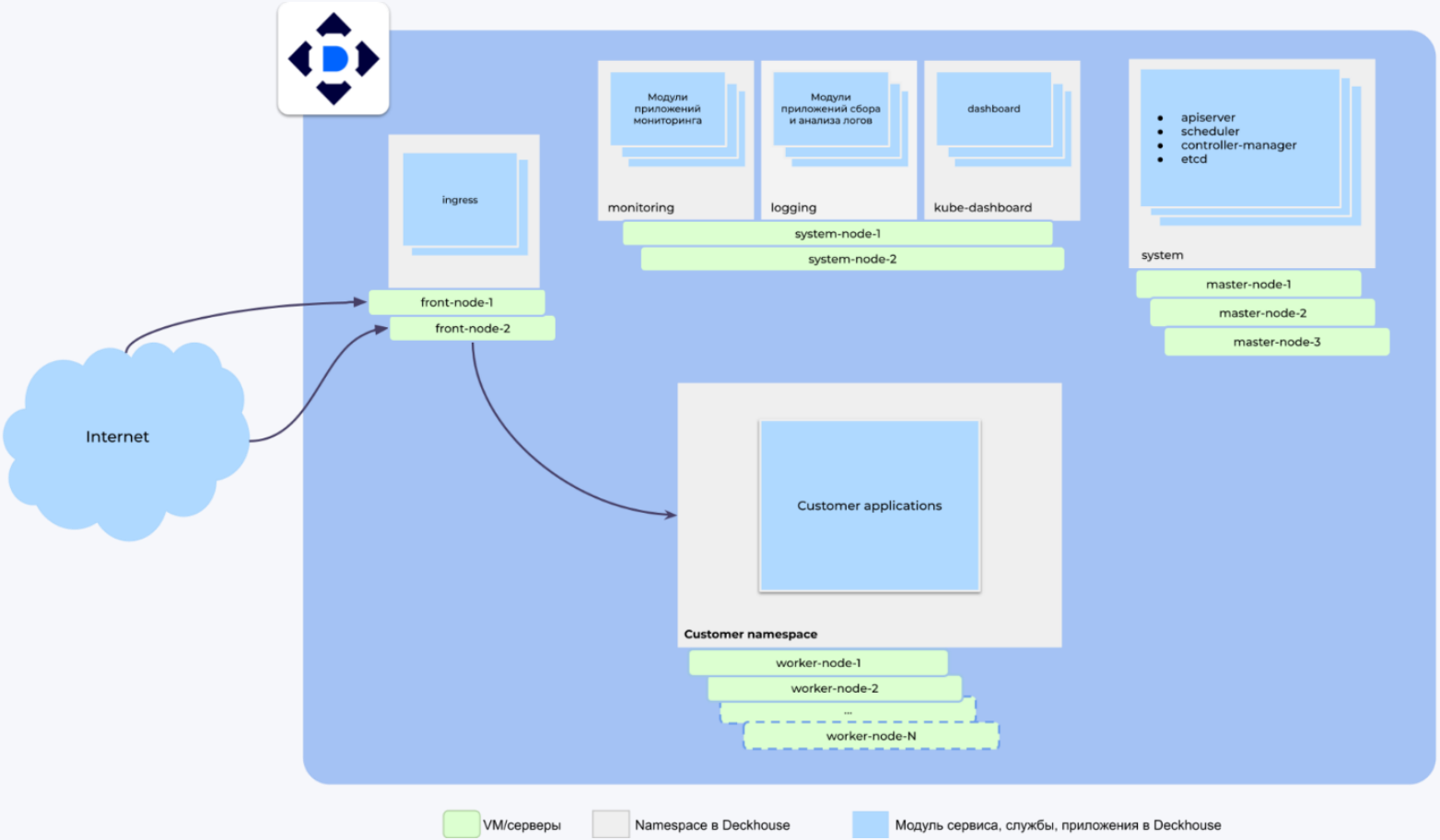
Возможность использования квот на ресурсы с помощью механизма Kubernetes resources quotas

Возможность выбора размещения приложения на определенной ноде с помощью pod node selector

Пример визуализации потребления и рекомендаций по ресурсам в графиках



Типовая архитектура кластера



Требования к ресурсам для развертывания Deckhouse и используемые репозитории

Базовая ОС

Ubuntu 16.04, 18.04, 20.04, CentOS 7,

Ресурсы

Master-node x 3

4 Cores / 8Gb RAM / 30Gb SSD

System node* x 2

4 Cores/8Gb RAM/30Gb SSD + 50Gb SSD
(для хранилища метрик)

Front-node x 2

2 Cores / 4Gb RAM / 30Gb SSD

Worker Nodes

1 CPU / 1GB RAM/10Gb SSD или выше,
в зависимости от ваших приложений.

Репозитории для Ubuntu

Стандартный репозиторий Ubuntu
http://apt.kubernetes.io/
https://nginx.org/packages/ubuntu/
https://download.docker.com/linux/ubuntu/

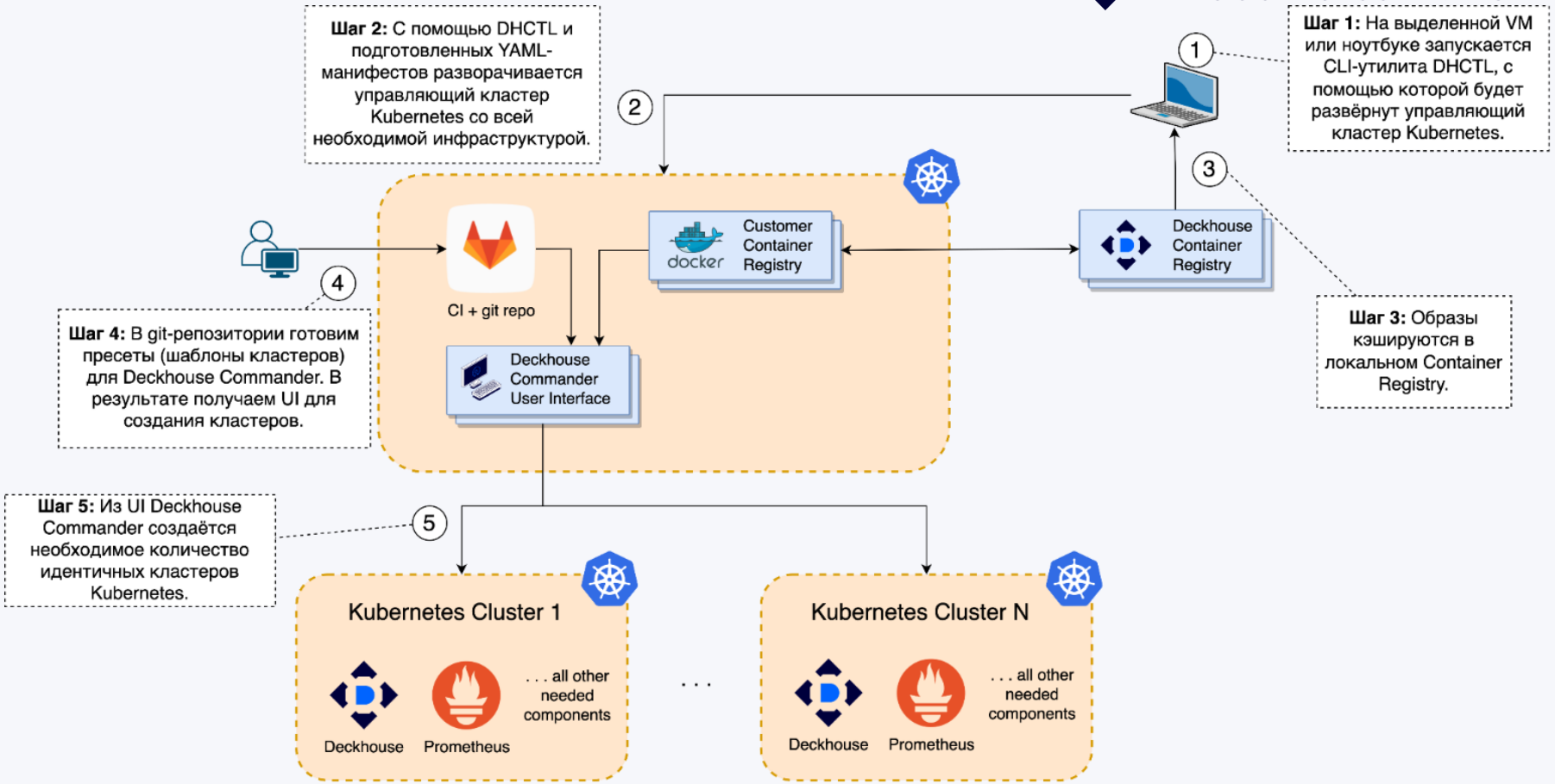
Репозитории для CentOS

https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
https://download.docker.com/linux/centos/7/$basearch/stable
http://nginx.org/packages/centos/$releasever/$basearch/

**System Nodes нужны для размещения отказоустойчивого Prometheus и др. компонент мониторинга.*

Развертывание в закрытом контуре

FLANT
Deckhouse
Kubernetes Platform



Persistent storage средствами Linstor (tech. preview)

FLANT
Deckhouse
Kubernetes Platform

OpenZFS

LVM

Реализация модели DRBD поверх известных технологий LVM и XFS

Обеспечивает высокодоступное и отказоустойчивое распределенное хранилище

Специально разработан как хранилище для контейнерных технологий

LINSTOR

CI/CD средствами ArgoCD (tech preview)

FLANT
Deckhouse
Kubernetes Platform

Инструмент CI/CD с декларативным синтаксисом

gitops

Реализует модель GitOps
Подробнее о подходе https://www.gitops.tech/

ArgoCD входит в CNCF

argo

Безопасный кластер корпоративного уровня

FLANT
Deckhouse
Kubernetes Platform

Преднастройки для обеспечения безопасности кластера

Интеграция с LDAP, AD, OIDC

Управление правами доступа по ролям (RBAC)

SSL для взаимодействия всех компонентов платформы Deckhouse

Доступ через OpenVPN

mTLS для взаимодействия пользовательских приложений на основе Istio

Сравнение с Vanilla Kubernetes

	Vanilla Kubernetes	*Deckhouse*
Что такое?	Платформа контейнеризации	Платформа контейнеризации + дополнительные компоненты
Сертифицираван в CNCF	ДА	ДА
Поддерживаемые ОС	Не занимается управлением нодами	Ubuntu, CentOS, Debian, Astra*
Безопасность	Нужно разбираться самостоятельно, как сделать кластер безопасным	Преднастроенная безопасность на уровне платформы
Запуск контейнеров под ROOT	ДА	ДА
Аутентификация и Авторизация	Нужно делать самому	Готовый компонент платформы
Решение для организации сети внутри кластера	Нужно выбрать самому и разобраться в нем	Преднастроенный Flannel или Cilium Cilium поставляется в режиме tech preview
Решение для балансировки входящего трафика	Нужно делать самому	Готовое решение для любой инфраструктуры: Преднастроенный и пропатченный nginx Ingress, управление load balancers в облаке, metalLB для bare metal.
Мониторинг	Нужно делать самому	Преднастроенные метрики, алерты, дашборды Grafana+Prometheus
Управление Хостами	Ручное, либо разрабатывать самому	Встроенное в платформу
Поддержка	Комьюнити	Комьюнити (все инструменты Open Source) + Флант

Сравнение с OpenShift (не под запись)

	OpenShift	*Deckhouse*
Компоненты CI/CD	Платформа контейнеризации	Используются внешние. Работаем над ArgoCD. Возможна настройка Флантом.
Шаблоны	OpenShift templates или Helm	Helm
Графический Интерфейс	Есть	Kubernetes Dashboard Deckhouse Commander Tech preview, предоставляется в рамках пакета поддержки Platinum
Image-Registry	Встроены в платформу	Используются внешние Возможна настройка Флантом.
Поддерживаемые ОС	RHEL	Ubuntu, CentOS, Debian-compatible Linux
Безопасность	ДА	Безопасность на уровне платформы Не для клиентских приложений
Возможность работы с Persistent Volumes	ДА	Linstor Tech preview, предоставляется в рамках пакета поддержки Platinum
Поддержка облаков 152-ФЗ	НЕТ	ДА
Наличие в реестре российского ПО	НЕТ	ДА
Vendor lock-in	ДА Нельзя съехать на OKD, т.к. там другая операционка	НЕТ Ванильные компоненты, перепуш образов в клиентский registry, кластер остается работоспособен без Deckhouse
Цена	$$$$$	$

Пакеты поддержки Deckhouse

	Standard*	Platinum
Время оказания поддержки	8×5	24×7
Время реакции	Один бизнес-день	5 минут
Время решения	Не регламентировано	Минимально возможное
Каналы взаимодействия	Email, Slack	Email, Slack, удалённый доступ
SLA для кластеров**		99,95 Простой не более 22 минут в месяц
SLA для приложений	Нет	Да***

** Входит в стоимость лицензии.*

*** Подробнее про SLA на кластера можно узнать в условиях оказания услуги Managed Kubernetes: https://legal.flant.ru/mk8s_terms_ru*

**** Согласовывается с заказчиком отдельно..*

Спасибо!

FLANT
Deckhouse
Kubernetes Platform

Готовая Open Source-платформа для управления всеми вашими K8s-кластерами в любых облаках, на «железе» и гибридной инфраструктуре

deckhouse.io

Сергей Дужий
Зам. директора по развитию бизнеса
s.duzhiy@olly.ru

ОЛЛИ ДИСТРИБУЦИЯ

Olly дистрибуция
Ваш надежный партнер

OLLYIT.RU
disti@olly.ru

Приложения

Platform Level

Monitoring

Prometheus Grafana

Ready to use

Metrics Dashboards Alerts

Autoscaling & Managing Resources

Priority classes

Metrics API integration

Scaling by any custom metric

Vertical Pod Autoscaler

Descheduler

Accessing Cluster

Dashboard OpenVPN

Network Load Balancing

Ingress nginx Istio service mesh

Little Things

Basic auth Secret copier Logshipper

Security

Users authentication Users authorization Network policy engine Audit External API access

Cluster Level

On-node software management (bashible)

Control plane management (kubeadm-based)

CNI

flannel

cilium (soon)

kube-proxy

kube-dns

Node Local DNS

Infrastructure management (Terraform based)

CSI (cloud specific)

Machine controller manager

Cloud controller manager

Cluster autoscaler

Bare-Metal Compatibility

MetalLB keepalived network-gateway

Infrastructure Level

Cloud-like

aws Google Cloud Platform Azure Yandex Cloud openstack. vmware vSphere

Non-cloud

Bare-metal servers Virtual machines (static)

Platform Level

Monitoring

Prometheus Grafana

Ready to use

Metrics Dashboards Alerts

Autoscaling & Managing Resources

Priority classes

Metrics API integration

Scaling by any custom metric

Vertical Pod Autoscaler

Descheduler

Accessing Cluster

Dashboard OpenVPN

Network Load Balancing

Ingress nginx Istio service mesh

Little Things

Basic auth Secret copier Logshipper

Security

Users authentication Users authorization Network policy engine Audit External API access

Cluster Level

On-node software management
bashible

Control plane management
kubeadm based

CNI

flannel

cilium
soon

kube-proxy

kube-dns

Node Local DNS

Infrastructure management
Terraform based

CSI
cloud specific

Machine controller manager

Cloud controller manager

Cluster autoscaler

Bare-Metal Compatibility

MetalLB keepalived network-gateway

Infrastructure Level

Cloud-like

aws Google Cloud Platform Azure Yandex Cloud openstack vmware vSphere

Non-cloud

Bare-metal servers Virtual machines static

Platform Level

Monitoring

Prometheus Grafana

Ready to use

Metrics Dashboards Alerts

Autoscaling & Managing Resources

Priority classes

Metrics API integration

Scaling by any custom metric

Vertical Pod Autoscaler

Descheduler

Accessing Cluster

Dashboard OpenVPN

Network Load Balancing

Ingress nginx Istio service mesh

Little Things

Basic auth Secret copier Logshipper

Security

Users authentication Users authorization Network policy engine Audit External API access

Cluster Level

On-node software management (bashible)

Control plane management (kubeadm-based)

CNI

flannel

cilium (soon)

kube-proxy

kube-dns

Node Local DNS

Infrastructure management (Terraform based)

CSI (cloud specific)

Machine controller manager

Cloud controller manager

Cluster autoscaler

Bare-Metal Compatibility

MetalLB keepalived network-gateway

Infrastructure Level

Cloud-like

aws Google Cloud Platform Azure Yandex Cloud openstack vmware vSphere

Non-cloud

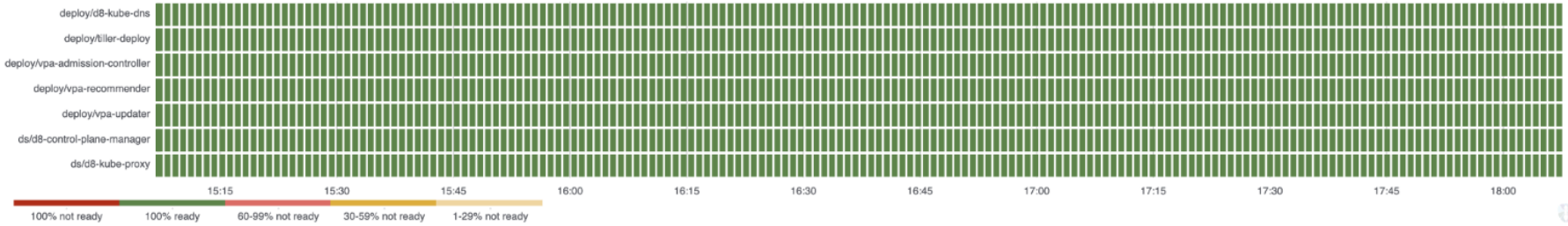
Bare-metal servers Virtual machines (static)

Средняя утилизация и рекомендации по VPA

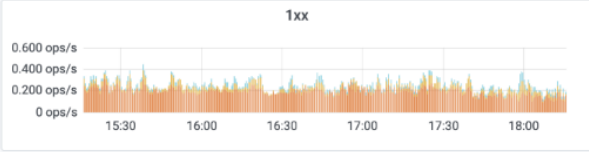
Controllers

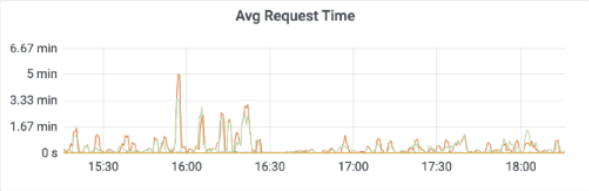
Controller	VPA Mode	CPU	Req CPU	VPA CPU	Over-req CPU	Under-req CPU	Throttling cores	Memory	Req Memory	VPA Memory	Over-req Memory	Under-req Memory	RX Network	TX Network	Read IOPS	Write IOPS	Total replicas	Not ready replicas	Pod Restarts
ds/d8-kube-proxy	Auto	0.089	0.34	0.34	0.25	0	0	326.55 MiB	555.14 MiB	555.14 MiB	228.59 MiB	0 B	0 b/s	0 b/s	0.01 io/s	0	14	0	0
deploy/d8-kube-dns	Auto	0.062	0.14	0.14	0.08	0	0	45.11 MiB	150.00 MiB	150.00 MiB	104.89 MiB	0 B	16.95 kb/s	23.50 kb/s	0 io/s	0	3	0	0
deploy/vpa-admission-controller	Initial	0.022	0	0.08	0	0.02	0	174.10 MiB	0 B	223.79 MiB	0 B	174.10 MiB	65.58 kb/s	3.34 kb/s	0 io/s	0	3	0	0

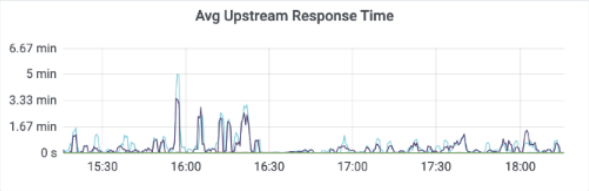
Status

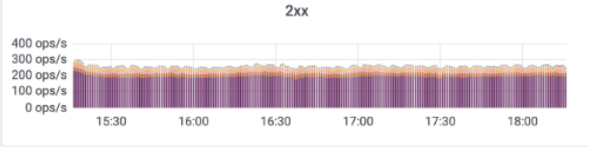


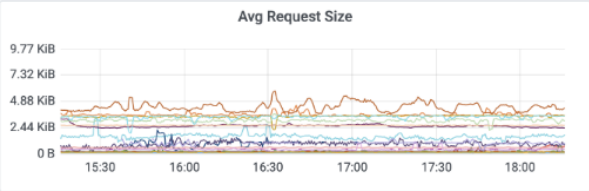
Request/Response статистика по Ingress’ам

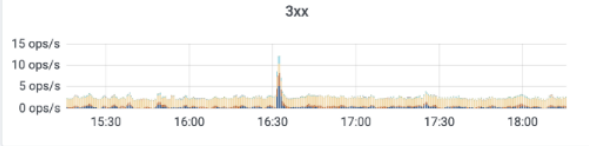


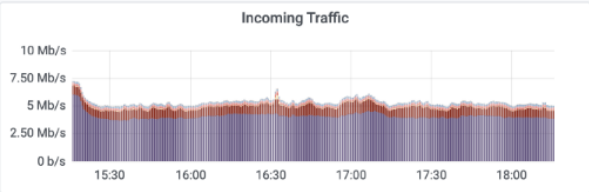


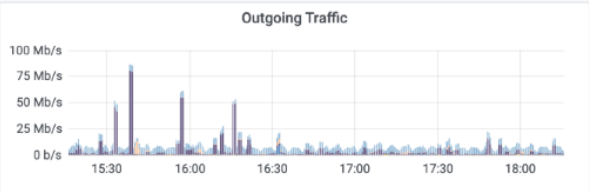


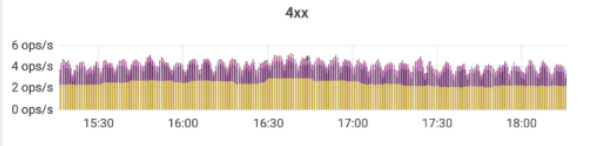


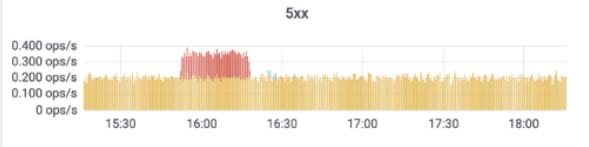






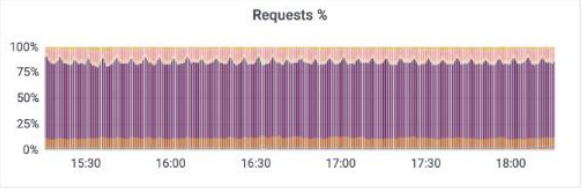
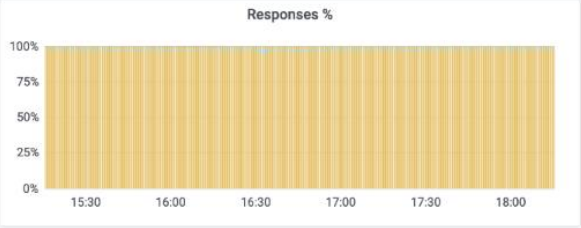
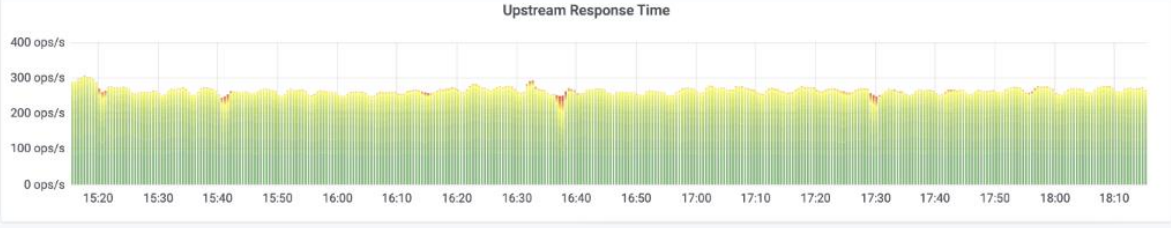
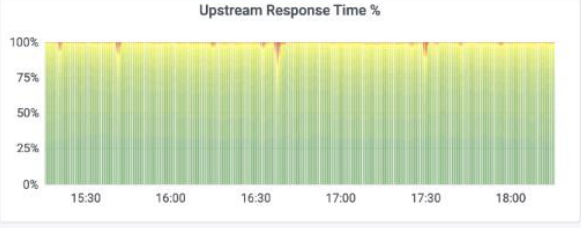






Request/Response статистика по Ingress’ам

Average

Namespace	RPS	Retried Req	Req Time	Upstr Time	In Traffic	Out Traffic	Req Size	Resp Size	1xx	2xx	3xx	4xx	5xx	HTTPS	GET	POST	HEAD	PUT	DELETE	OPTIONS	PATCH
madison-production	194.77 ops/s	0.02%	172.50 ms	137.76 ms	4.08 Mb/s	4.32 Mb/s	2.55 KiB	2.70 KiB	0.01%	98.58%	0.14%	1.26%	0.01%	100.00%	29.15%	70.82%	0%	0.02%	0.00%	0%	0.01%
infra-production	22.89 ops/s	0%	44.89 ms	38.55 ms	816.48 kb/s	79.30 kb/s	4.35 KiB	433.05 B	0%	97.86%	0.22%	1.92%	0.00%	100.00%	0.45%	99.55%	0%	0%	0%	0%	0%
pult-production	20.27 ops/s	0%	119.00 ms	118.28 ms	87.90 kb/s	4.71 Mb/s	541.94 B	28.36 KiB	0%	93.94%	0.04%	6.02%	0%	3.72%	3.70%	96.30%	0%	0%	0%	0%	0%
promscale-production	7.57 ops/s	0%	5.46 ms	5.05 ms	162.96 kb/s	6.15 kb/s	2.63 KiB	101.50 B	0%	99.96%	0.00%	0.03%	0%	100.00%	1.34%	98.66%	0%	0%	0%	0%	0%
werfio-production	6.94 ops/s	0%	5.49 ms	5.25 ms	6.77 kb/s	286.57 kb/s	121.88 B	5.04 KiB	0%	99.56%	0.30%	0.14%	0%	100.00%	13.57%	86.42%	0.00%	0%	0%	0%	0%

1 2 3 4 5