



Стачка

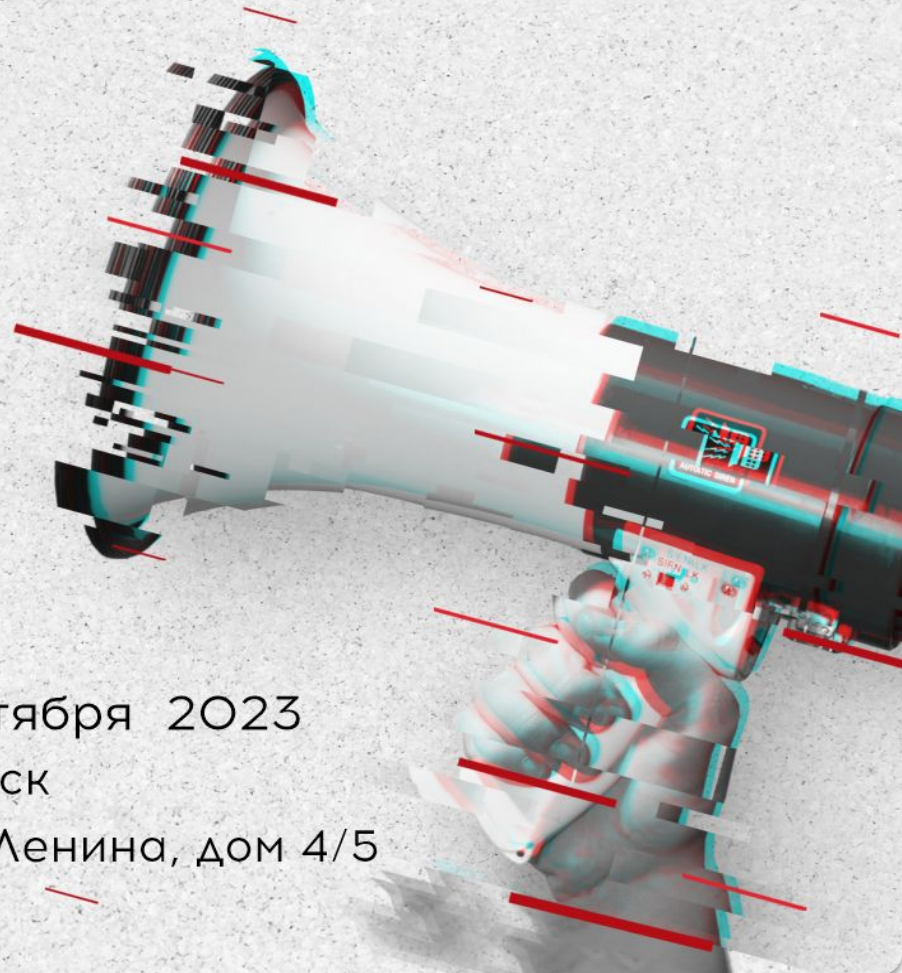
**10 МЕЖДУНАРОДНАЯ
IT-КОНФЕРЕНЦИЯ**

**РЕШЕНИЯ
ЕСТЬ**

15-16 сентября 2023

г. Ульяновск

Площадь Ленина, дом 4/5



Контейнеры и Kubernetes: не боимся, а используем и защищаем

Ганюшкин Андрей





Ганюшкин Андрей



Luntry.ru

Коммерческий директор

Мой опыт: 10 лет в сфере безопасности, как на Российском, так и международном рынке

О себе: Организатор конференции «БЕКОН» — первая в России конференция по БЕзопасности КОНтейнеров и контейнерных сред





Контейнеры и Kubernetes: не боимся, а используем и защищаем



План

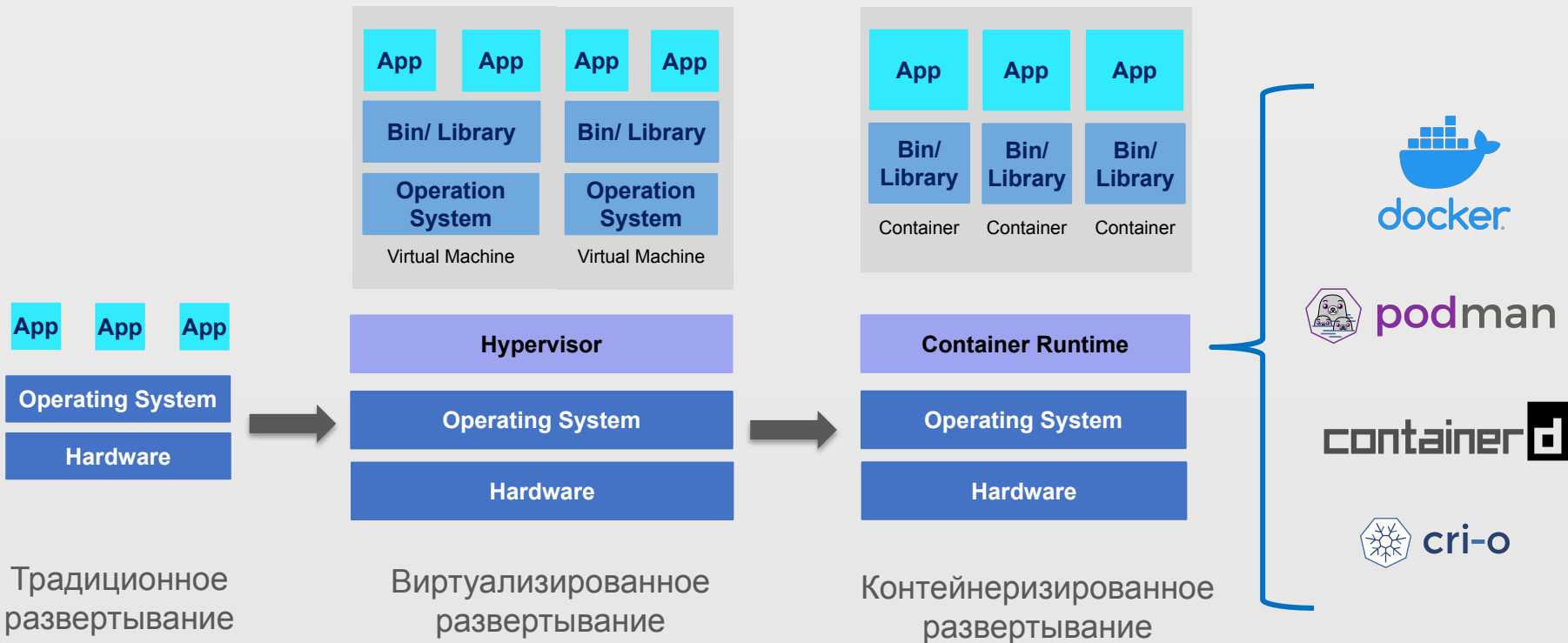
- Контейнеры
Отличие от виртуальных машин
- Оркестрация контейнеров
Kubernetes
- Взгляд регуляторов на контейнерные среды
PCI DSS, 118-й приказ, ГОСТ 57580.1-2017
- Безопасность контейнерных сред
- Специфика безопасности Kubernetes
RBAC, NetworkPolicy, безопасность Kubernetes-ресурсов



Контейнеры



Эволюция к контейнерам



Что такое контейнер?

Container — это Linux process с определёнными свойствами/ограничениями

- Что можно увидеть: namespaces (pid, user, uts, ipc, net, mnt), pivot_root + image)
- Что можно делать: Capabilities, seccomp, LSMs
- Что можно использовать: Control group (процессор, память, устройства, ...)

```

root    2966156  0.0  0.0  110128  5932  ?    Sl   Nov19   0:11  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root    2966174  0.0  0.0    1020     4  ?    Ss   Nov19   0:00  |  \_ /pause
root    2966375  0.0  0.0  108720  6356  ?    Sl   Nov19   0:11  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
sadm   2966394  0.0  0.0  827512  19728  ?    Ssl  Nov19   0:00  |  \_ node /usr/bin/nodemon /src/index.js
sadm   2966421  0.0  0.0    4460     80  ?    S    Nov19   0:00  |  \_ sh -c node /src/index.js
sadm   2966422  0.0  0.0  967396  16596  ?    Sl   Nov19   0:00  |  \_ node /src/index.js
root    2988902  0.0  0.0  108720  5408  ?    Sl   Nov19   0:11  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root    2988922  0.0  0.0    1020     4  ?    Ss   Nov19   0:00  |  \_ /pause
root    2989066  0.0  0.0  108720  5408  ?    Sl   Nov19   0:26  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root    2989099  0.0  0.0   31000  23956  ?    Ss   Nov19   0:42  |  \_ /usr/local/bin/python /usr/local/bin/gunicorn -b :8080 --workers 1 --threads 1 --
root    2989116  0.3  0.1  142092  48964  ?    Sl   Nov19  16:50  |  \_ /usr/local/bin/python /usr/local/bin/gunicorn -b :8080 --workers 1 --threads
root    2989333  0.0  0.0  110128  5404  ?    Sl   Nov19   0:11  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root    2989352  0.0  0.0    1020     4  ?    Ss   Nov19   0:00  |  \_ /pause
root    596808  0.0  0.0  110128  6316  ?    Sl   Nov20   0:06  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root    596827  0.0  0.0    1020     4  ?    Ss   Nov20   0:00  |  \_ /pause
root    598309  0.0  0.0  110128  6224  ?    Sl   Nov20   0:07  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root    598334  1.4  5.5  7236340  1832196  pts/0  Ssl+ Nov20   39:39  \_ /docker-java-home/bin/java -Djava.util.logging.config.file=/opt/atlassian/conflue
root    599854  1.0  1.3  7007820  427956  pts/0  Sl+  Nov20   28:11  |  \_ /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java -classpath /opt/atlassian/conf
root    701694  0.0  0.0    4288    764  ?    Ss+  Nov20   0:00  \_ /bin/sh

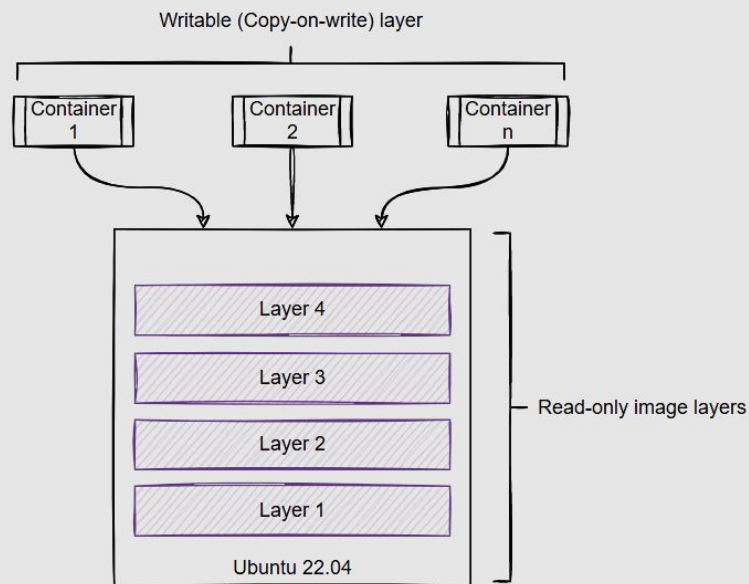
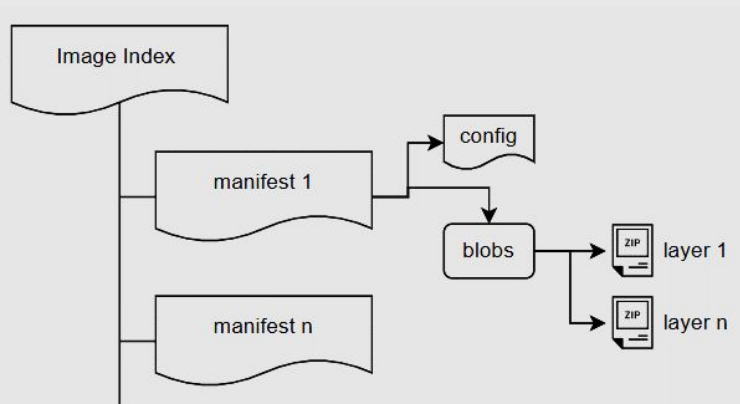
```



Что такое образ контейнера?

Container Image — это неизменяемый пакет файлов операционной системы, кода приложения и любых зависимостей приложения

- Union File System
OverlayFS как реализация
- OCI image спецификация

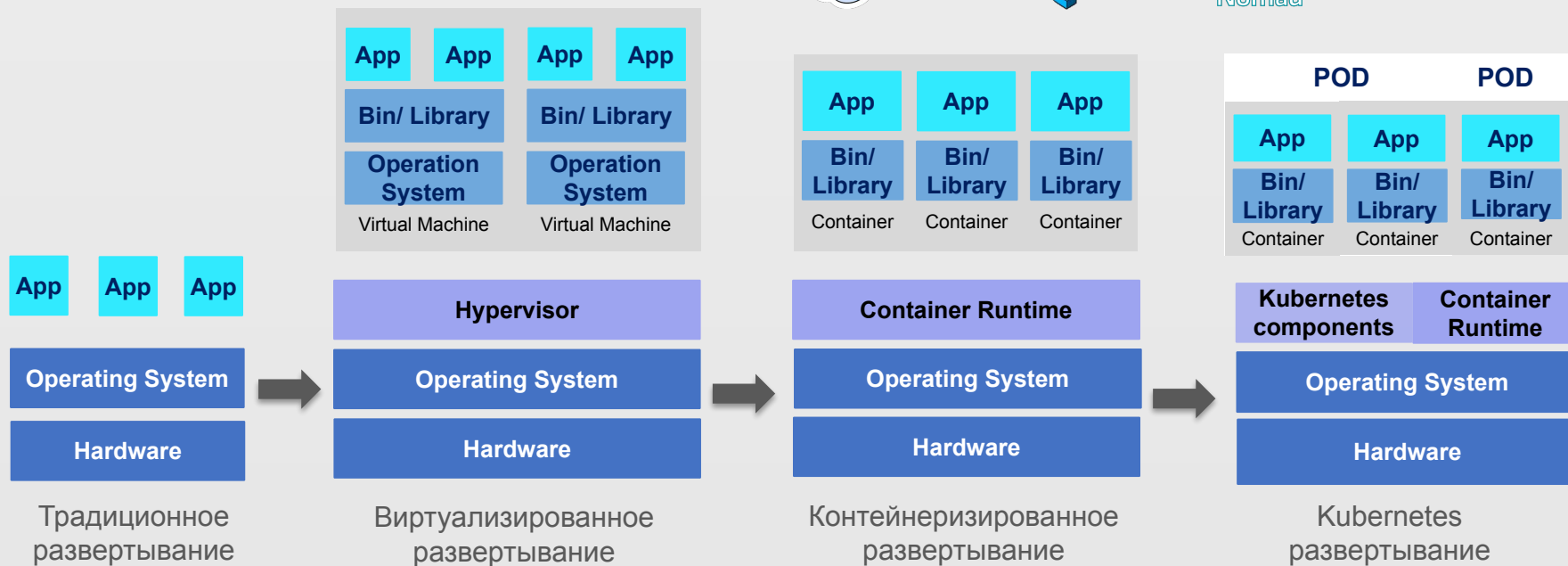


Оркестрация контейнеров



Эволюция от контейнеров к оркестраторам контейнеров

Оркестраторы:



Что такое Kubernetes?

- Kubernetes (K8s) — это открытое программное обеспечение для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями.
- 5 бинарей

Platform as a
Service (PaaS)

Configuration

Function

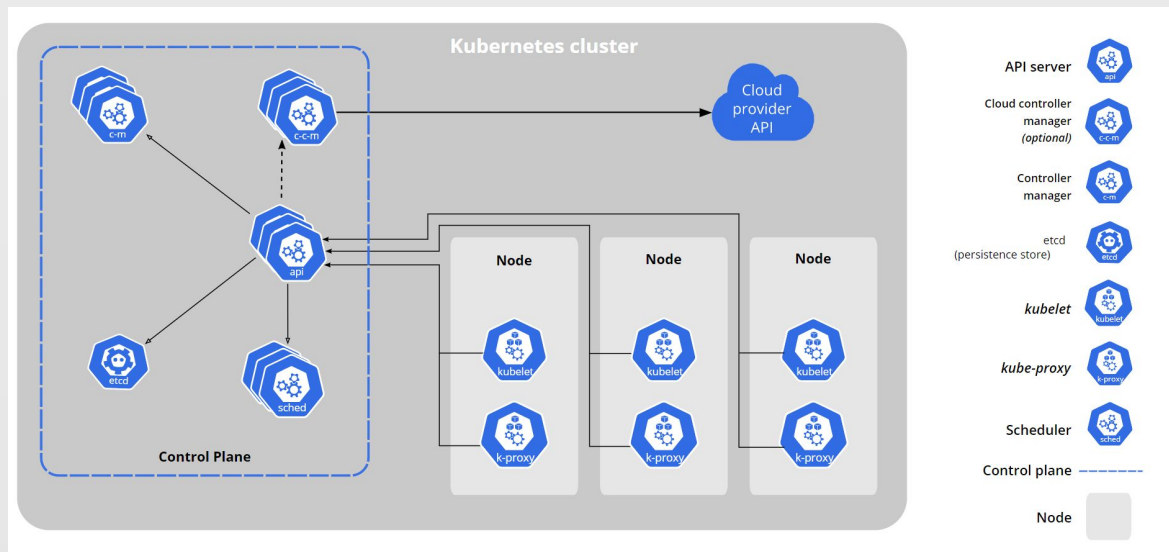
Applications

Runtime

Containers

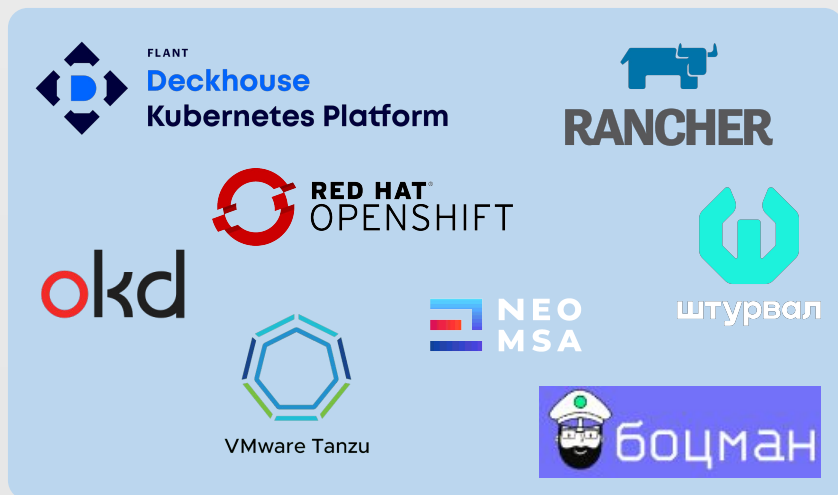
Operation Systems

Hardware



Дистрибутивы Kubernetes

- Это фреймворк
- OnPrem и Managed Kubernetes
- Это ядро Linux 21 века
- На базе Kubernetes делают свои дистрибутивы



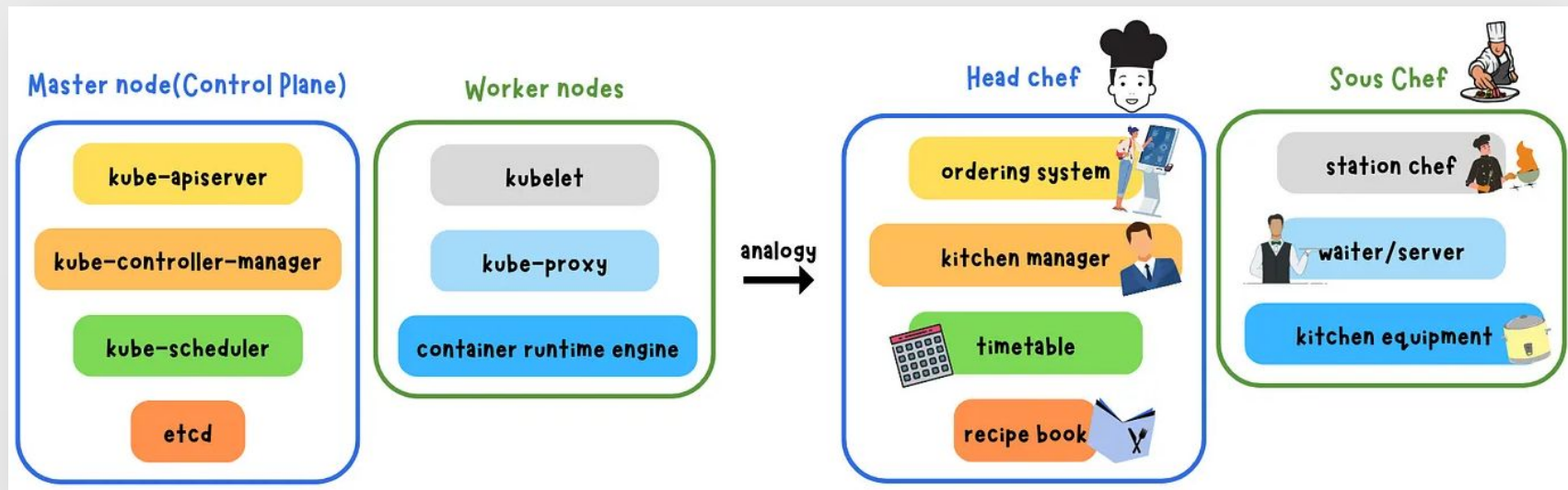
Kubernetes на простом языке



- [Kubernetes в переводе на детский](#)



Kubernetes на простом языке



- Понимание архитектуры Kubernetes через аналогию с шеф-поваром ресторана



Взгляд регуляторов на контейнерные среды



ГОСТ Р 57580.1-2017 от ЦБ РФ

Примеры

выполнения
требований ГОСТ
Р 57580.1-2017
в средах
контейнерной
оркестрации
на базе Kubernetes

№	ПРОЦЕСС	ОБЪЕКТ ПРИМЕНЕНИЯ МЕР		
		УРОВЕНЬ NODE	УРОВЕНЬ ORCHESTRATOR	УРОВЕНЬ POD/CONTAINER
1	Обеспечение защиты информации при управлении доступом	Запрет подключения к Worker Node, контроль кластером через Master Node	Интеграция с Identity Provider (например, LDAP) для получения доступа к кластеру	Использование RBAC для реализации принципа least privilege
2	Обеспечение защиты вычислительных сетей	<ul style="list-style-type: none"> Помещение кластера в защищенный сегмент Реализация TLS при общении между nodes 	<ul style="list-style-type: none"> Контроль запуска pods/containers только на определенных nodes Использование наложенных СЗИ для контроля трафика (L3/L4 firewalling) 	<ul style="list-style-type: none"> Использование Network Policy Использование mTLS
3	Контроль целостности и защищенности информационной инфраструктуры	Принятый в организации подход к реализации процесса управления уязвимостями	Обновление версии оркестратора	<ul style="list-style-type: none"> Подпись образов, верификация подписи перед запуском контейнера при помощи внешних решений Использование внешних решения для идентификации уязвимостей в образе
4	Защита от вредоносного кода	Использование SELinux/AppArmor (компенсирующие меры)	Неприменимо	Компенсирующие меры защиты: <ul style="list-style-type: none"> Использование SecurityContext Использование внешних решений для защиты контейнеров в runtime
5	Предотвращение утечек информации	Неприменимо (используется принятый в организации процесс предотвращения утечек информации)		
6	Управление инцидентами защиты информации	Использование принятого в организации подхода к мониторингу элементов ИТ-инфраструктуры	Разработка Audit Policy с последующим направлением журналов в SIEM для идентификации инцидентов ИБ	<ul style="list-style-type: none"> Разработка Audit Policy с последующим направлением журналов в SIEM для идентификации инцидентов ИБ Использование внешних решений, которые позволяют идентифицировать инциденты ИБ Отправка журналов внешних средств защиты в SIEM (нарушение политик ИБ)
7	Защита среды виртуализации	Реализация практик, указанных в процессах [1], [2], [3], [4], [6] (согласно нумерации, используемой в таблице)	Реализация практик, указанных в процессах [1], [2], [3], [4], [6]	Реализация практик, указанных в процессах [1], [2], [3], [4], [6]
8	Защита при осуществлении удаленного доступа с использованием мобильных (переносных) устройств	Неприменимо (используется принятый в организации процесс защиты при осуществлении удаленного доступа с использованием мобильных (переносных) устройств)		



PCI DSS

- НСПК требует соответствия PCI DSS
- Существует руководство по безопасности для контейнеров и оркестраторов контейнеров

Стандарт PCI DSS

В ПС «Мир» для обеспечения безопасности данных карт «Мир» используется международный индустриальный стандарт PCI Data Security Standard (PCI DSS)

Этот стандарт должен применяться всеми организациями, которые хранят, обрабатывают и передают данные карт «Мир». К таким организациям относятся и торгово-сервисные предприятия, которые принимают к оплате карты «Мир».

Стандарт PCI DSS — это международный стандарт безопасности, созданный специально для защиты данных платежных карт. Он позволяет защитить организацию от инцидентов безопасности и обеспечить необходимый уровень защищенности во всей платежной системе.



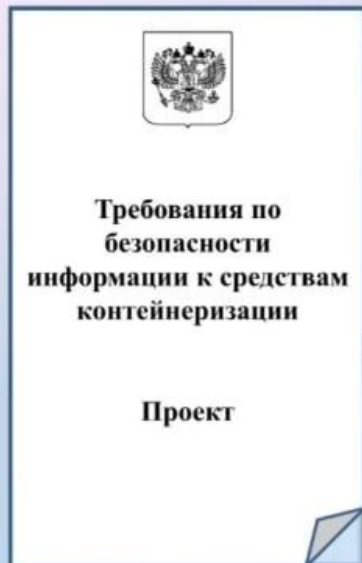
Date: September 2022
Author: Best Practices for Container Orchestration Special Interest Group
PCI Security Standards Council

Information Supplement:
Guidance for Containers and
Container Orchestration Tools



118-й приказ ФСТЭК

Требования по безопасности информации к средствам контейнеризации



Функциональные возможности:

формирование среды выполнения контейнеров и обеспечения выполнения их процессов

запуск контейнера и управление данным контейнером

создание образов контейнеров

распространение образов контейнеров

централизованное управление контейнерами и организацией взаимодействия между ними

Функции безопасности:

управление доступом

идентификация и аутентификация пользователей

изоляция контейнеров

выявление уязвимостей в образах контейнеров

проверка корректности конфигурации контейнеров

контроль целостности контейнеров и их образов

централизованное управление образами контейнеров и контейнерами

регистрация событий безопасности

ИСП РАН

PT POSITIVE TECHNOLOGIES

Яндекс

РусБИТех
Одесские институты информатики

BELLSOFT

R-Vision

ideco

ЦБИ
Центр компетенций информатики

Сбербанк-Технологии

РЕДСОФТ

Эшелон
Информационные технологии



118-й приказ ФСТЭК

- Выписка из Требований по безопасности информации, утвержденных приказом ФСТЭК России от 4 июля 2022 г. N 118
- Сертифицированная ОС
- Сертифицированный оркестратор
- Сертифицированное наложенное средство безопасности

УТВЕРЖДЕНЫ
приказом ФСТЭК России
от 4 июля 2022 г. № 118

Требования по безопасности информации к средствам контейнеризации (выписка)

1. Настоящие Требования являются обязательными требованиями в области технического регулирования к продукции, используемой в целях защиты сведений, составляющих государственную тайну или относимых к охраняемой в соответствии с законодательством Российской Федерации иной информации ограниченного доступа¹ (далее – требования по безопасности информации), предъявляемыми к программным средствам, обеспечивающим создание и функционирование изолированных программных сред на основе ядра хостовой операционной системы (далее – контейнеры) в информационной (автоматизированной) системе (далее – средства контейнеризации).

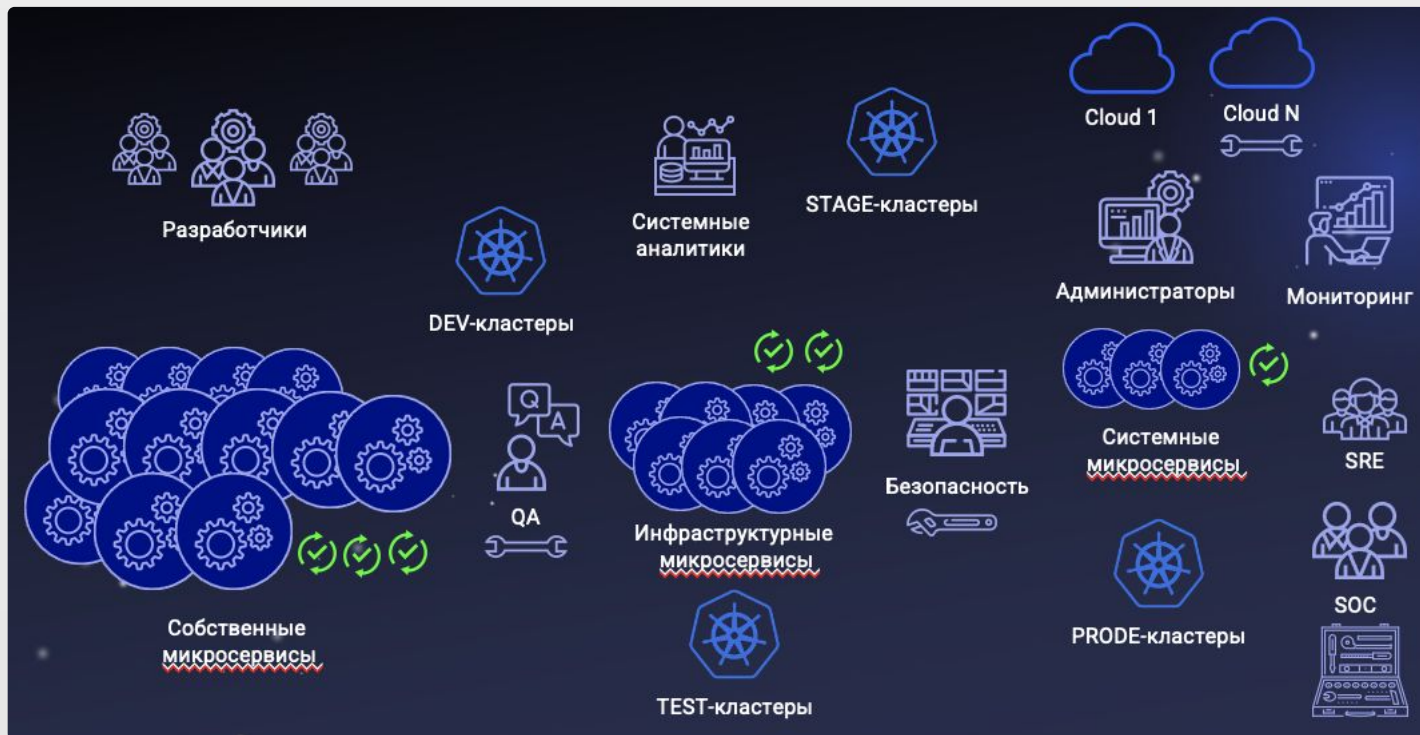
2. Выполнение настоящих Требований является обязательным при проведении работ по оценке соответствия (включая работы по сертификации) средств технической защиты информации и средств обеспечения безопасности информационных технологий, организуемых ФСТЭК России в пределах своих



Безопасность контейнерных сред



Постоянные, сложные и быстрые изменения



Проблемы и вызовы для ИБ в Kubernetes

- Окружение становится все сложнее
"Complexity is the worst enemy of security, and our systems are getting more complex all the time", Bruce Schneier
"The only thing that ever yielded real security gains was controlling complexity", Thomas Dullien/"Halvar Flake"
- Разработка стремительно развивается. Старые подходы к безопасности не работают. Департаменты разработки, поддержки и безопасности должны работать вместе
- Kubernetes — это фреймворк. Уникальные модели нарушителя, модель угроз и поверхность атаки
- Атакующий по-прежнему на шаг впереди: 0-day, unfixed 1-day, backdoors, APT и т.д.



Как подходить к безопасности в контейнерных средах?

- Develop
- Distribute
- Deploy
- Runtime

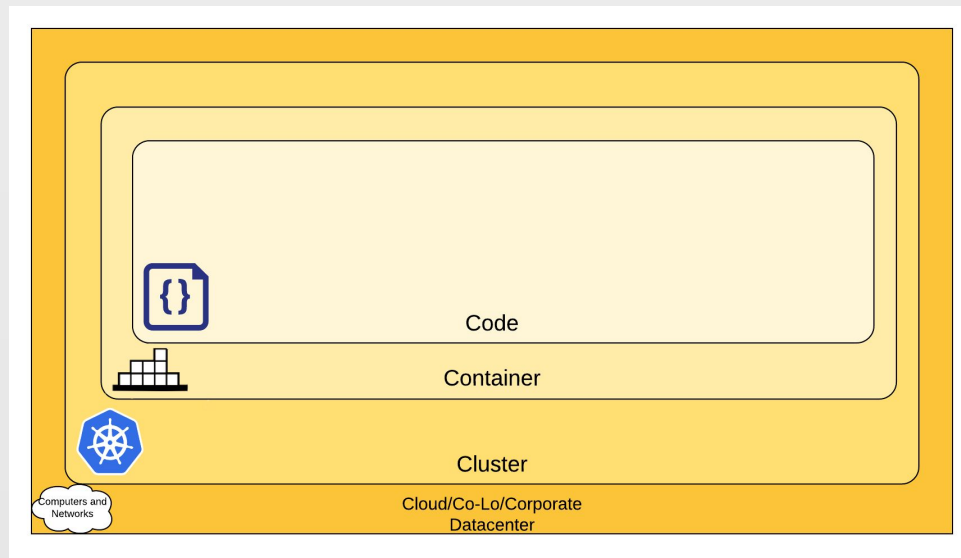
[“CNCF Cloud Native Security Whitepaper”](#)



Как подходить к безопасности в контейнерных средах?

- Cloud/Co-Lo/Corporate Datacenter
- Cluster
- Container
- Code

"The 4C's of Cloud Native security"



Как подходить к безопасности в контейнерных средах?

- Identify
- Protect
- Detect
- Respond
- Recover
- Deception*



"NIST CYBERSECURITY FRAMEWORK"



Как подходить к безопасности в контейнерных средах?

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access Kubernetes API server	Access cloud resources	Images from a private registry	Data destruction
Compromised image in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Collecting data from pod	Resource hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from proxy server	Application credentials in configuration files	Exposed sensitive interfaces	Application credentials in configuration files		
Exposed sensitive interfaces	SSH server running inside container	Container service account			Access managed identity credentials	Instance Metadata API	Writable hostPath mount		
	Sidecar injection	Static pods			Malicious admission controller		CoreDNS poisoning		
							ARP poisoning and IP spoofing		

Threat Matrix for Kubernetes от Microsoft



Специфика безопасности Kubernetes



К чему все привыкли?

- Firewall
- Patch Management
- IPS/IDS
- Vulnerability Assessment
- WAF
- Antimalware
- SIEM
- FIM
- DLP
- DCAP
- Key Management
- EDR
- IAM
- SOAR
- PAM
- Deception Platform



**А как дела с этим
в Kubernetes?**



Основные аспекты ИБ в Kubernetes

- Multitenancy
- Аутентификация пользователей
- Анализатор прав доступа
- Контроль Kubernetes ресурсов
- Логирование
- Контроль безопасности образов
- Управление секретами
- Безопасность Runtime
- Безопасность хоста/ноды
- Сетевая безопасность
- Контроль соответствия

Доклад "Классификация и систематизация средств безопасности для Kubernetes"

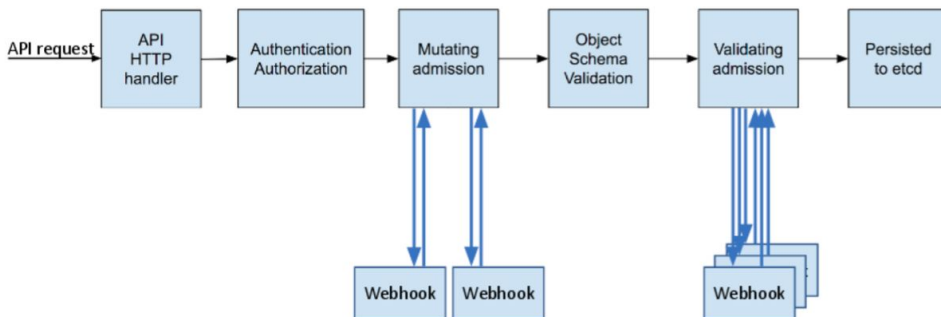


Контроль Kubernetes ресурсов

ВСЕ, ЧТО ЕСТЬ В KUBERNETES, – ЭТО YAML!

Задачи:

- Мутация – добавление, обогащение Kubernetes ресурса или исключение нежелательного
- Валидация – контроль соответствия в режимах аудита и предотвращения
- Генерация – автоматическое создание дополнительных Kubernetes ресурсов



Анализ прав доступа

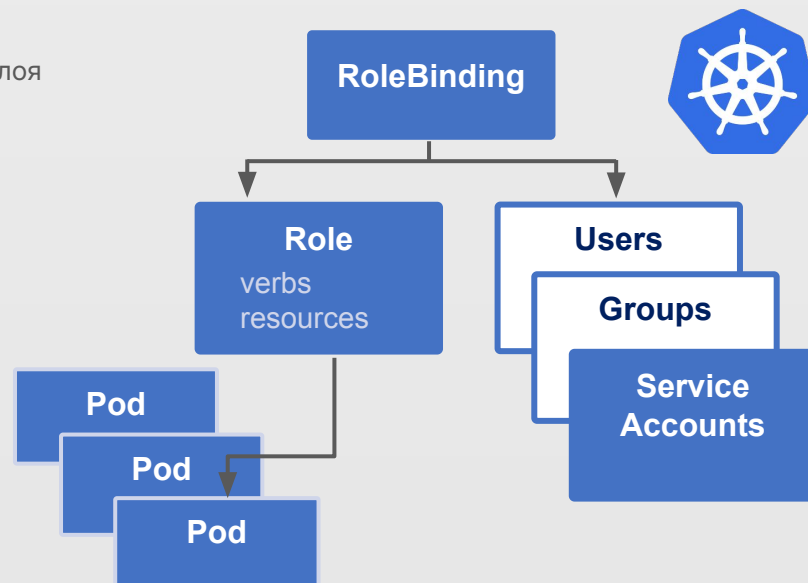
Анализ прав доступа (RBAC) к Kubernetes ресурсам на соответствие принципу наименьших привилегий и на опасные права.

Реализация:

■ Периодическая проверка в Kubernetes

■ На этапе деплоя

Authorization: Mechanisms		
Mechanism	Decision Source	Usage
Node	API Server built-in	Internal use (kubelets)
ABAC	Static file	Insecure, deprecated
RBAC	API Objects	User and administrators
WebHook	External services	Integration
AlwaysDeny AlwaysAllow	API Server built-in	Testing



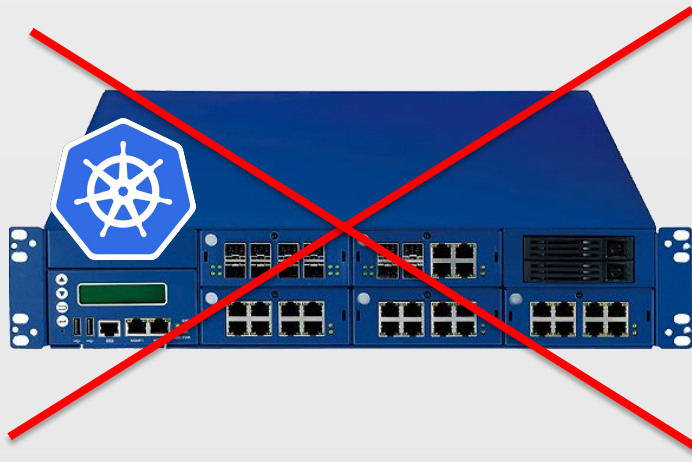
Сетевая безопасность

МОЖНО РАЗДЕЛИТЬ НА 2
УРОВНЯ:

- Host Network Security
- Pod Network Security

ЗАДАЧИ:

- Сегментация, • Микросегментация
- Взаимная аутентификация
- Шифрование
- Контроль входящего и исходящего трафика



ПОДХОДЫ:

- Whitelist
- Blacklist

**IP-адрес меняется/переходит от запуска
к запуску микросервиса!**



Заключение



Заключение

- Если у вас еще нет контейнеров, то они обязательно появятся
- Так или иначе вы придёте к использованию оркестратора
- Kubernetes — стандарт де-факто среди оркестраторов
- Обеспечение безопасности контейнеров и оркестратора — новый вызов для ИБ
- Сегодня регуляторы уже подготовились к этому вызову
- Обеспечение безопасности контейнеров и оркестратора требует отдельных подходов и инструментов



Спасибо за внимание!

Андрей Ганюшкин

avg@luntry.ru

Telegram-канал [@k8security](https://t.me/k8security)

www.luntry.ru

