

Инструменты защиты контейнеров в процессе разработки

Тимофей Минин

Менеджер по развитию продуктов облачная и сетевой безопасности, "Лаборатория Касперского"

kaspersky

- Выгоды контейнеризации и роль в разработке ПО
- Основные угрозы в контейнерной инфраструктуре
- Обеспечение безопасности - от хранения до эксплуатации
- Kaspersky Container Security – архитектура и роль

Контейнеризация

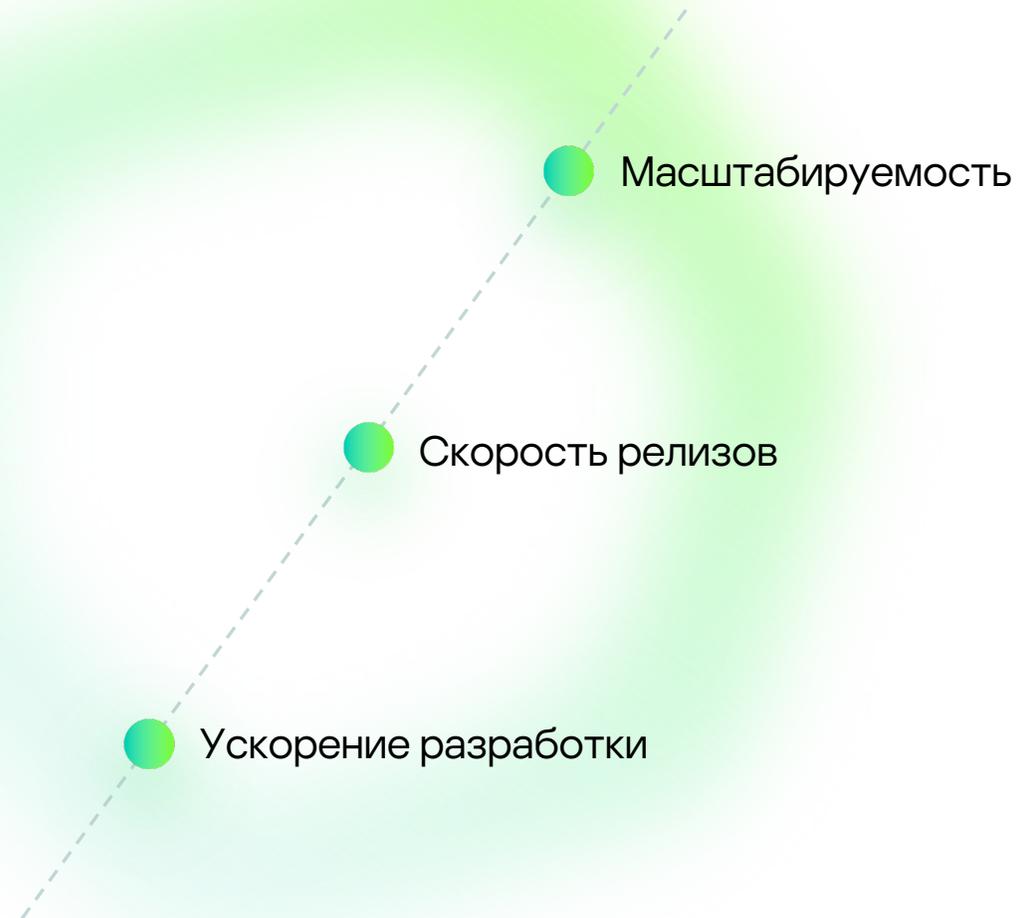


Микросервисы

Вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие небольших, слабо связанных и легко изменяемых модулей — **микросервисов**

Преимущества

Микросервисы - легко развивать и обновлять: добавление или улучшение отдельных функций никак не повлияет на остальные компоненты

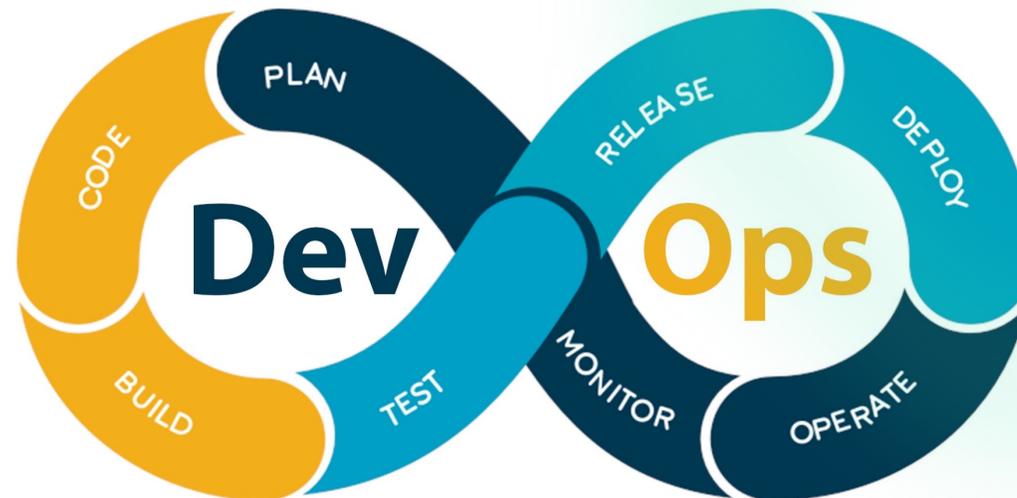


Контейнеризация – расширяет возможности и скорость бизнеса

5

Контейнер – это изолированная среда, в которой запускается единственное приложение. Контейнер легче, чем виртуальная машина, но содержит все необходимое для запуска приложения.

Технологии контейнеризации – ключевой компонент в построении микросервисной архитектуры и эффективного **DevOps** процесса



Компоненты контейнерной инфраструктуры

Хранение

Реестр

Образ контейнера

Образ контейнера

Образ контейнера

Сборка
↔
Хранение

Сборка и запуск

CI / CD
платформа

Запуск
↔
Работа

Эксплуатация

Оркестратор

Контейнер

Контейнер

Контейнер

Контейнер

Среда запуска
контейнеров

Операционная система

Рабочая нода



Основные риски ключевых компонентов контейнерных сред

Образы

- Открытые внешние источники
- Уязвимости ПО
- Ошибки в конфигурациях
- Вредоносное ПО
- Секреты в открытом виде
- Использование недоверенных образов

Реестр образов

- Незащищенное подключение
- Наличие устаревших образов с уязвимостями и вредоносным ПО
- Недостаточные ограничения на аутентификацию и авторизацию

Оркестратор

- Не ограничен административный доступ
- Доступ без авторизации
- Отсутствует или слабое разделение трафика между контейнерами
- Не разнесены по хостам контейнеры с разным уровнями защиты данных
- Ошибки в конфигурации оркестратора

Контейнеры

- Уязвимости среды выполнения
- Неограниченный доступ контейнеров к сети
- Небезопасные конфигурации
- Уязвимости приложений в контейнерах
- Незапланированные контейнеры в среде выполнения

ОС хоста

- Большая площадь атак
- Общее ядро ОС для всех контейнеров
- Уязвимости компонентов ОС
- Некорректная настройка прав доступа пользователей
- Возможность доступа контейнеров к файловой системе

Практики безопасности



Практики безопасности в процесс разработки



VCS & Реестр

Исследование

CI инструменты

**Создание
и тестирование**

CD инструменты

**Доставка
и развертывание**

Оркестратор

Выполнение

Практики безопасности в процесс разработки

VCS & Реестр

Исследование и хранение



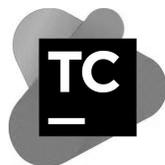
- Проверка образов реестра на актуальность, наличие уязвимостей и вредоносного ПО
 - Регулярное и ручное сканирование
 - Создание чистого реестра
- Проверка конфигурационных файлов (IaC, Dockerfiles) на наличие ошибок, небезопасных настроек, секретов

Практики безопасности в процесс разработки



CI инструменты

**Создание
и тестирование**



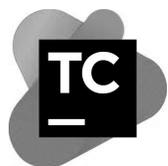
- Контроль целевого качества Dockerfile перед сборкой самого образа
- Проверка собранного образа на все возможные риски
- Подпись (Signing Image) образа после сборки – присваивание цифровой подписи подтверждающее целостность и подлинность



Практики безопасности в процесс разработки

CD инструменты

Доставка и развертывание

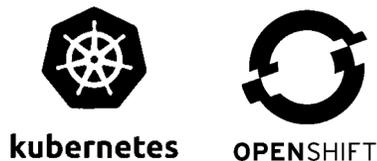


- Реализация Admission Controlling - допускать к развертыванию только образы:
 - которые соответствуют политикам безопасности (compliant)
 - только из внутреннего/чистого реестра
 - только с подписью

Практики безопасности в процесс разработки

Оркестратор

Выполнение



- Анализ конфигураций оркестратора на соответствие лучшим практикам (CIS)
- Контроль сетевого взаимодействия контейнеров
- Контроль целостности контейнера
- Поведенческий анализ для отслеживания аномалий
- Контроль запуска приложений и сервисов внутри контейнера

Kaspersky Container Security



Закрывает проблемы
безопасности контейнерных
сред на всех этапах

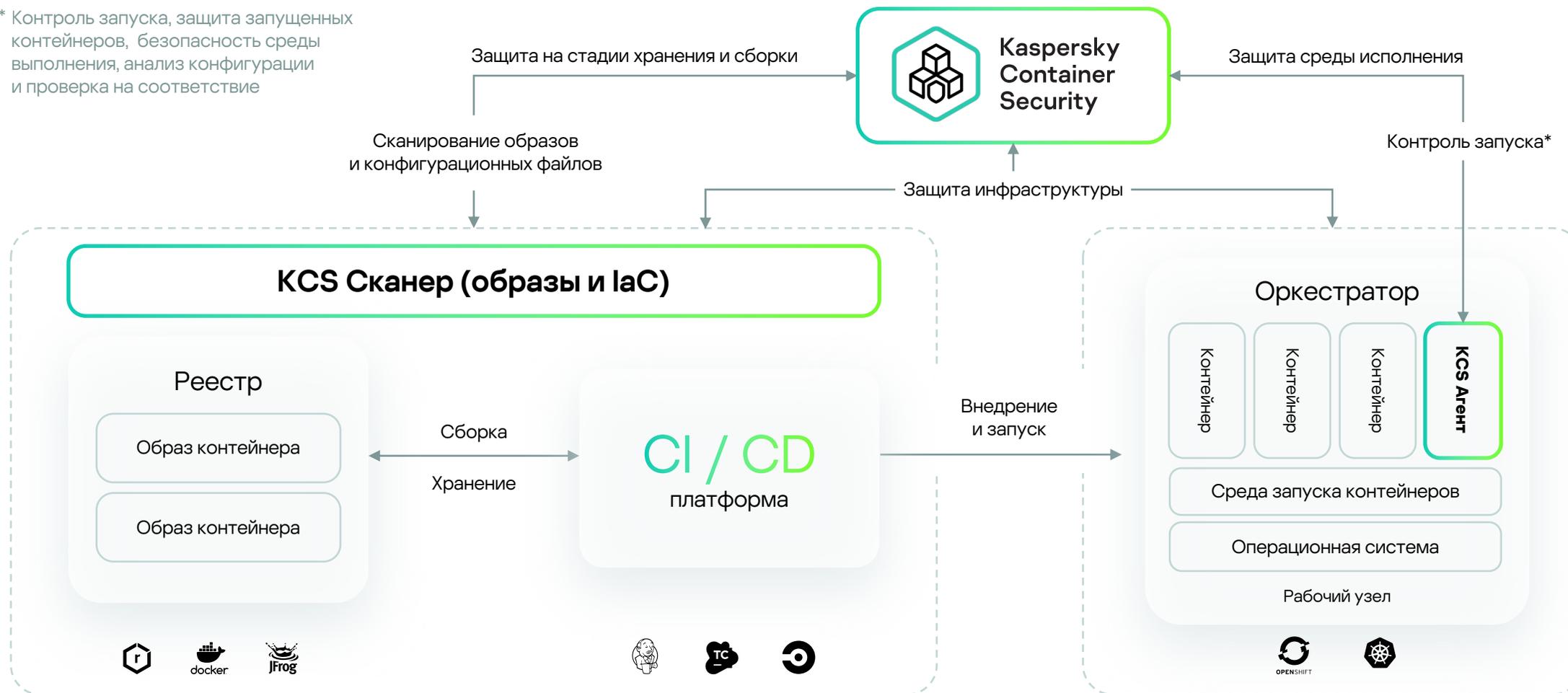
Продукт обеспечивает безопасность всех
компонентов контейнерных платформ:
образы, реестры образов, оркестраторы,
контейнеры, ОС хоста

Позволяет интегрироваться
в процессы безопасной
разработки

Встраивается в CI pipelines
и интегрируется в инфраструктуру

Архитектура Kaspersky Container Security – защита на всех этапах

* Контроль запуска, защита запущенных контейнеров, безопасность среды выполнения, анализ конфигурации и проверка на соответствие



При разработке приложений на микросервисной архитектуре

Безопасность приложений/сервисов в контейнерах, среды выполнения и платформ оркестрации

При выстраивании процессов DevSecOps

Добавление «quality gate» требует проверки собираемых контейнеров

При необходимости соблюдения Compliance

KCS позволяет автоматизировать процесс проверки на соответствие стандартам и требованиям регуляторов

Для инвентаризации и визуализации

Компонентов контейнерной инфраструктуры и ресурсов в кластерах

The screenshot displays the Kaspersky Container Security (KCS) web interface. The top navigation bar includes 'Registry images' and the specific image path 'jfrog.tronsec.ru/tron/nginx:1-alpine'. The main content area is divided into several sections:

- Risk Rating:** A prominent red indicator shows 'Image is not compliant' with a 'Rescan image' button. The risk rating is 'Critical'.
- Image Assurance:** A summary of scan results: Vulnerabilities (failed), Malware (passed), Sensitive data (passed), and Misconfigurations (passed).
- Policy Summary:** A table showing the status of various policies:

Policy	Status
All controls	failed
Policy: 1	passed
Policy: qa Vulnerability control	failed
Policy: qa Vulnerability control	failed
- Vulnerability Scan Details:** A circular gauge shows '105 Total' vulnerabilities. A legend indicates: 10 Critical, 40 High, 33 Medium, 2 Low, and 0 Negligible.
- Compliance Section:** Shows 'CIS Kubernetes Benchmarks' for 'openshift.ru-central1.internal'. A progress bar indicates 'Total' results: 12 Failed, 39 Risk acceptance cancellation, 2 Passed, and 0 Skipped. Below this, a list of 53 controls across 8 categories is shown, with specific control details and pass/fail status.

Спасибо!

kaspersky