

# Хранилища контейнеров и артефактов поставки, инфраструктура для контроля безопасности артефактов

# Не забудь включить запись!



# План

- Жизненный цикл артефакта
- Хранение артефактов
- Registry
- Security
- DEMO

## Артефакт

- это любой созданный искусственно элемент программной системы

# Жизненный цикл артефакта

- Создание
- Тестирование
- Различного рода проверки
- Эксплуатация
- Обновление
- Удаление

**Нужно ли проводить проверки безопасности в пайплайне?**

**В каком именно месте расположить проверки безопасности?**

## Типы образов

- Образы готовые для развертывания в production окружении
- Образы с различными debug инструментами

# Хранение артефактов

- "Боевые" сервера
- Файл сервера
- Флэшки ^\_^
- Специализированные хранилища

# Хранение артефактов



## Google Container Registry



# Хранение артефактов: GCR

- Безопасное, приватное Docker registry
- Сборка и развертывание в автоматическом режиме
- Встроенное сканирование уязвимостей
- Блокировка потенциально опасных образов
- Нативная поддержка докера
- Быстрый и высокодоступный
- Поддержка триггеров
- API сканирования уязвимостей

# Хранение артефактов: GCR

Resource	Monthly Free Usage Limits
Standard Storage	5 GB-months
Class A Operations	5,000
Class B Operations	50,000
Network Egress	1 GB from North America to each GCP egress destination (Australia and China excluded)

Monthly Usage	Egress to Worldwide Destinations (excluding Asia & Australia) (per GB)	Egress to Asia Destinations (excluding China, but including Hong Kong) (per GB)	Egress to China Destinations (excluding Hong Kong) (per GB)	Egress to Australia Destinations (per GB)	Ingress
0-1 TB	\$0.12	\$0.12	\$0.23	\$0.19	Free
1-10 TB	\$0.11	\$0.11	\$0.22	\$0.18	Free
10+ TB	\$0.08	\$0.08	\$0.20	\$0.15	Free

# Хранение артефактов: GCR

Storage Class <sup>1</sup>	Class A operations (per 10,000 operations)	Class B operations (per 10,000 operations)	Free operations
Standard Storage	\$0.05	\$0.004	Free
Nearline Storage and Durable Reduced Availability (DRA) Storage	\$0.10	\$0.01	Free
Coldline Storage	\$0.10	\$0.05	Free

	Nearline Storage	Coldline Storage
Data retrieval	\$0.01 per GB	\$0.05 per GB
Minimum storage duration	30 days	90 days

# Хранение артефактов: GCR

API or Feature	Class A Operations	Class B Operations	Free Operations
JSON API	storage.*.insert <sup>1</sup> storage.*.patch storage.*.update storage.*.setIamPolicy storage.buckets.list storage.buckets.lockRetentionPolicy storage.notifications.delete storage.objects.compose storage.objects.copy storage.objects.list storage.objects.rewrite storage.objects.watchAll storage.projects.hmacKeys.create storage.projects.hmacKeys.list storage.*.AccessControls.delete	storage.*.get storage.*.getIamPolicy storage.*.testIamPermissions storage.*.AccessControls.list storage.notifications.list Each object notification	storage.channels.stop storage.buckets.delete storage.objects.delete storage.projects.hmacKeys.delete
XML API	GET Service GET Bucket (when listing objects in a bucket) PUT POST	GET Bucket (when retrieving bucket configuration) GET Object HEAD	DELETE

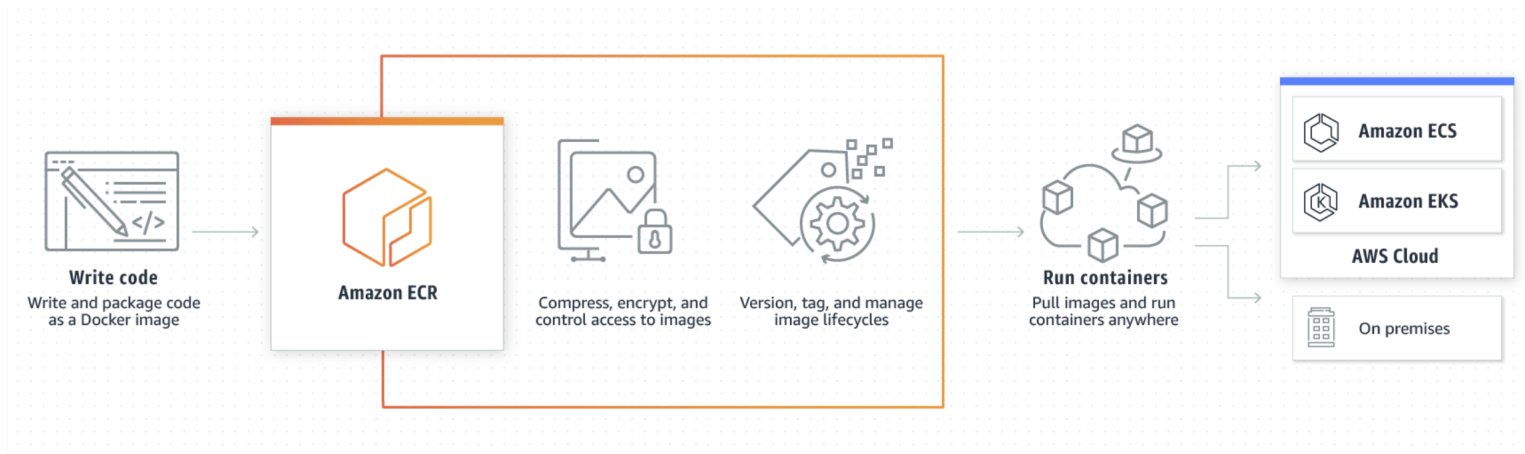
## Amazon Elastic Container Registry



Amazon ECR

# Хранение артефактов: ECR

- Полностью управляемое хранилище
- Высокая доступность
- Безопасность



# Хранение артефактов: ECR

Amazon ECR offers **new** customers **500MB-month** of storage **for** one **year**.

Storage:

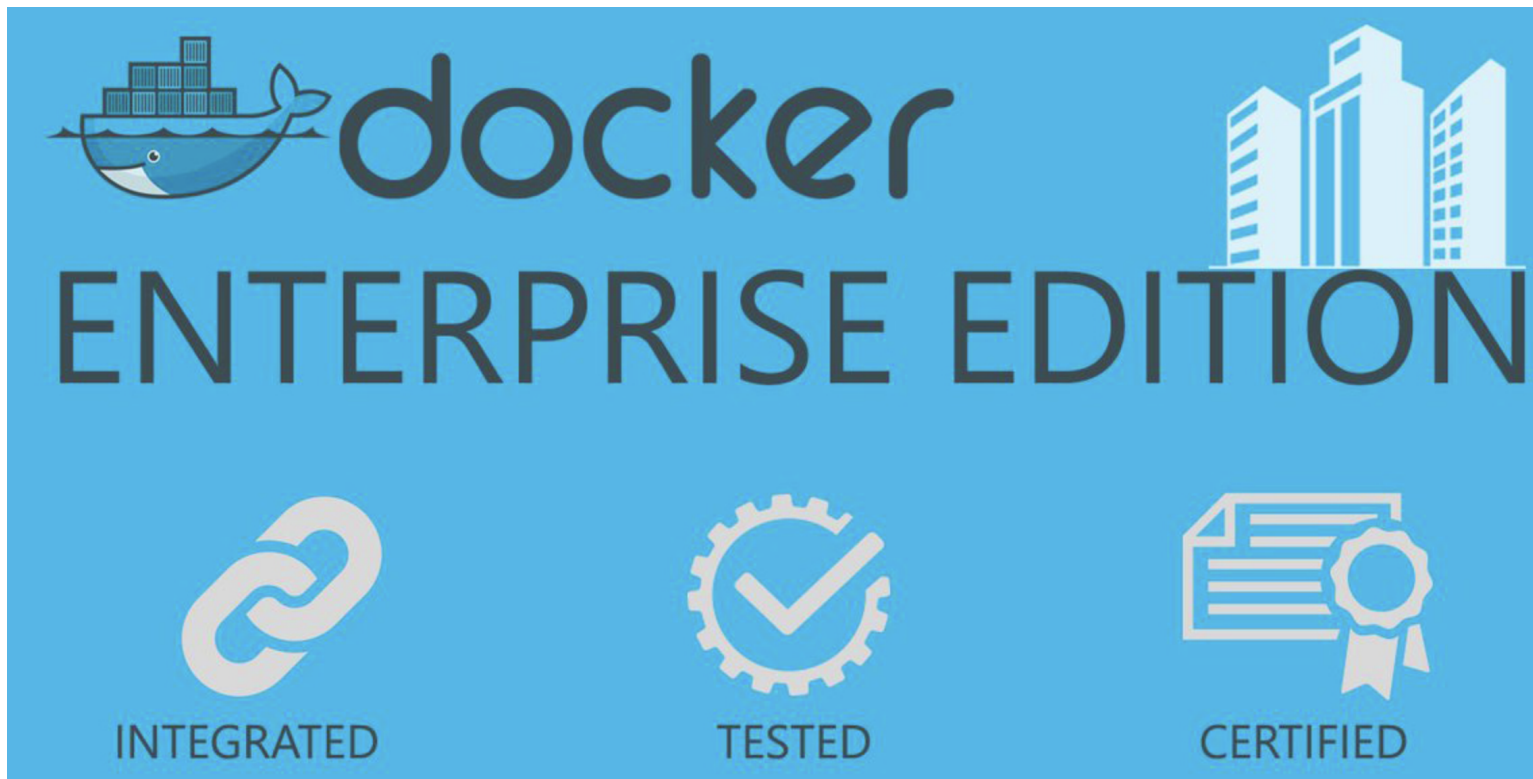
- Storage is \$0.10 per GB-month

Data transfer: \*\*

Region: EU (Frankfurt) ▾

	Pricing
<b>Data Transfer IN</b>	
All data transfer in	\$0.00 per GB
<b>Data Transfer OUT***</b>	
Up to 1 GB / Month	\$0.00 per GB
Next 9.999 TB / Month	\$0.09 per GB
Next 40 TB / Month	\$0.085 per GB
Next 100 TB / Month	\$0.07 per GB
Greater than 150 TB / Month	\$0.05 per GB

# Хранение артефактов: Docker Enterprise



# Хранение артефактов: Docker Enterprise

## Компоненты входящие в поставку

- Docker Desktop Enterprise
- Docker Application
- Docker Kubernetes Service (DKS)
- Lifecycle Automation Tools

# Хранение артефактов: Quay



Build, Store and Distribute your Containers

# Хранение артефактов: Quay

- Автоматические билды
- Сканирование безопасности
- Геораспределенная репликация
- Security vulnerability detection integration
- Continuous garbage collection
- Torrent distribution ^\_^
- Поддержка манифестов для различных архитектур

# Хранение артефактов: Harbor



# Хранение артефактов: Harbor

- Multi-tenant
- Расширенный API и Web UI
- Репликация образов между инстансами Harbor
- RBAC
- Анализ безопасности и уязвимостей
- Подпись образов и проверка

# Хранение артефактов: Harbor

## Квоты для проектов

Configuration

Authentication Email System Settings Labels **Project Quotas**

Default artifact count per project unlimited [EDIT](#)

Default disk space per project unlimited [↻](#)

Project	Owner	Count	Storage
temp	admin	0 of unlimited	0Byte of unlimited
otus	admin	2 of unlimited	203.76MB of unlimited
library	admin	0 of unlimited	0Byte of unlimited

1 - 3 of 3 quotas

## Очистка мусора

Garbage Collection

Garbage Collection History

Current Schedule: None

[EDIT](#) [GC NOW](#)

# Хранение артефактов: Harbor

## Создание правил репликации

## Создание endpoint

# Хранение артефактов: Harbor

## Настройка проектов

The screenshot shows the Harbor Configuration page for a project named 'otus'. The page is titled 'otus System Admin' and has a navigation menu with the following items: Summary, Repositories, Helm Charts, Members, Labels, Logs, Robot Accounts, Tag Retention, Webhooks, and Configuration (which is currently selected). The main content area is divided into several sections:

- Project registry:** A checkbox labeled 'Public' is unchecked. Below it, a note states: 'Making a project registry public will make all repositories accessible to everyone.'
- Deployment security:** A checkbox labeled 'Enable content trust' is unchecked. Below it, a note states: 'Allow only verified images to be deployed.' Another checkbox labeled 'Prevent vulnerable images from running.' is also unchecked. Below it, a note states: 'Prevent images with vulnerability severity of Low and above from being deployed.'
- Vulnerability scanning:** A checkbox labeled 'Automatically scan images on push' is unchecked. Below it, a note states: 'Automatically scan images when they are pushed to the project registry.'
- CVE whitelist:** A note states: 'Project whitelist allows vulnerabilities in this list to be ignored in this project when pushing and pulling images. You can either use the default whitelist configured at the system level or click on 'Project whitelist' to create a new whitelist. Add individual CVE IDs before clicking 'COPY FROM SYSTEM' to add system whitelist as well.'

At the bottom of the CVE whitelist section, there are two radio buttons: 'System whitelist' (which is selected) and 'Project whitelist'. Below these are two buttons: 'ADD' and 'COPY FROM SYSTEM'. To the right of these buttons is an 'Expires at' field with a dropdown menu. The dropdown is currently set to 'Never expires' and is checked. Below the dropdown is another checkbox labeled 'Never expires', which is also checked. At the bottom left of the CVE whitelist section, there is a text input field containing the word 'None'.

# Хранение артефактов: Harbor

## Результаты сканирования образа

< Projects < Repositories < otus/nginx  
otus/nginx:latest

Author NGINX Docker Mainta...  
Architecture amd64  
OS linux  
OS Version  
Docker Version 18.06.1-ce  
Scan Completed Oct 16, 2019

1 Package has High vulnerability  
11 Packages have Medium vulnerabilities  
5 Packages have Low vulnerabilities  
1 Package has Unknown vulnerability

Vulnerability Build History

SCAN

Vulnerability	Severity	Package	Current version	Fixed in version
> CVE-2005-2541	Negligible	tar	1.30+dfsg-6	
> CVE-2019-9923	Negligible	tar	1.30+dfsg-6	
> CVE-2007-5686	Negligible	shadow	1:4.5-1.1	
> CVE-2013-4235	Negligible	shadow	1:4.5-1.1	
> CVE-2018-7169	Low	shadow	1:4.5-1.1	
> CVE-2019-17543	Unknown	lz4	1.8.3-1	
> CVE-2019-15718	Low	systemd	241-5	
> CVE-2019-3843	Medium	systemd	241-5	

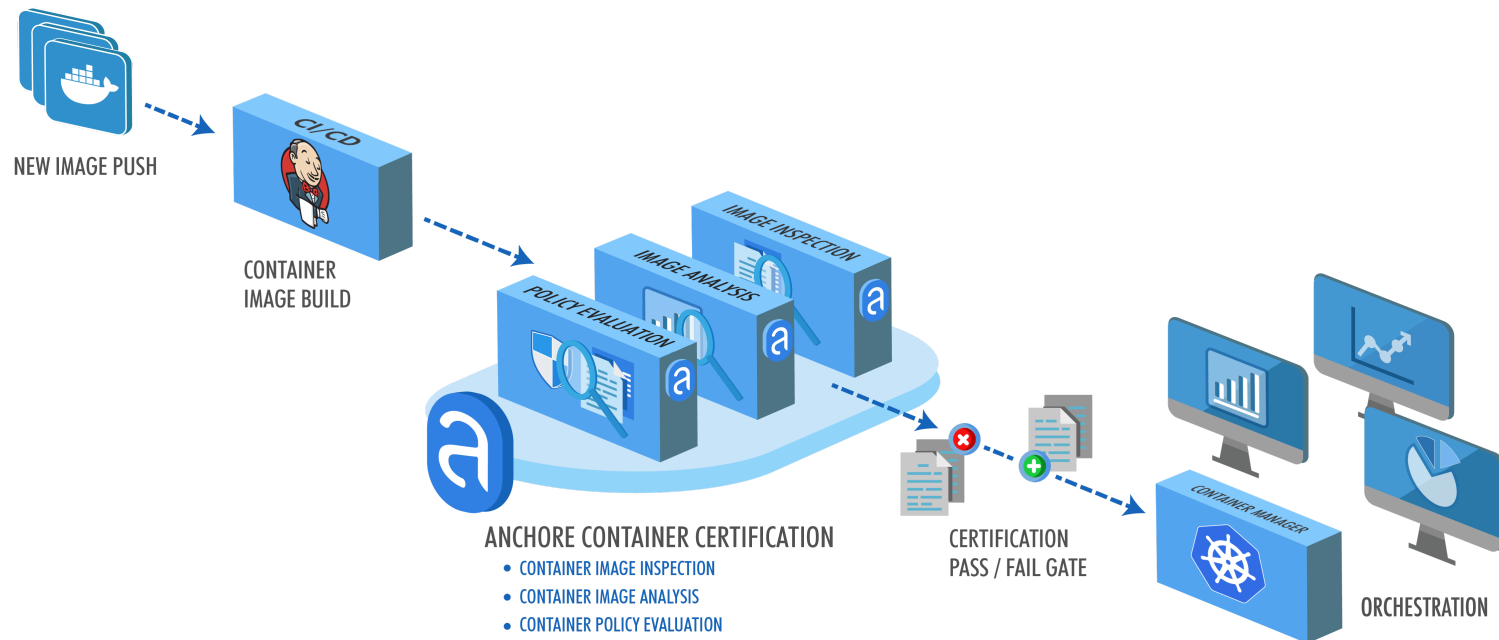
# Хранение артефактов: Harbor

## Результаты сканирования образа

Vulnerability	
Build History	
Create on	Commands
7/10/19, 12:22 AM	ADD file:71ac26257198ecf6a4ea05f45e522de961543a965486dd49a86bd6aca8342026 in /
7/10/19, 12:22 AM	CMD ["bash"]
7/23/19, 10:58 PM	LABEL maintainer=NGINX Docker Maintainers <docker-maint@nginx.com>
7/23/19, 10:58 PM	ENV NGINX_VERSION=1.17.2
7/23/19, 10:58 PM	ENV NJS_VERSION=0.3.3
7/23/19, 10:58 PM	ENV PKG_RELEASE=1-buster
7/23/19, 10:59 PM	<pre>RUN set -x &amp;&amp; addgroup --system --gid 101 nginx &amp;&amp; adduser --system --disabled-login --ingroup nginx --no-create-home --home /nonexistent --gecos "nginx user" --shell /bin/false --uid 101 nginx &amp;&amp; apt-get update &amp;&amp; apt-get install --no-install-recommends --no-install-suggests -y gnupg1 ca-certificates &amp;&amp; NGINX_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; found=""; for server in ha.pool.sks-keyservers.net hkp://keyserver.ubuntu.com:80 hkp://p80.pool.sks-keyservers.net:80 pgp.mit.edu ; do echo "Fetching GPG key \$NGINX_GPGKEY from \$server"; apt-key adv --keyserver "\$server" --keyserver-options timeout=10 --recv-keys "\$NGINX_GPGKEY" &amp;&amp; found=yes &amp;&amp; break; done; test -z "\$found" &amp;&amp; echo &gt;&amp;2 "error: failed to fetch GPG key \$NGINX_GPGKEY" &amp;&amp; exit 1; apt-get remove --purge --auto-remove -y gnupg1 &amp;&amp; rm -rf /var/lib/apt/lists/* &amp;&amp; dpkgArch="\$(dpkg --print-architecture)" &amp;&amp; nginxPackages=" nginx=\${NGINX_VERSION}-\${PKG_RELEASE} nginx-module-xslt=\${NGINX_VERSION}-\${PKG_RELEASE} nginx-module-geoip=\${NGINX_VERSION}-\${PKG_RELEASE} nginx-module-image-filter=\${NGINX_VERSION}-\${PKG_RELEASE} nginx-module-njs=\${NGINX_VERSION}-\${NJS_VERSION}-\${PKG_RELEASE}" &amp;&amp; case "\$dpkgArch" in amd64 i386) echo "deb https://nginx.org/packages/mainline/debian/ buster nginx" &gt;&gt; /etc/apt/sources.list.d/nginx.list &amp;&amp; apt-get update ;; *) echo "deb-src https://nginx.org/packages/mainline/debian/ buster nginx" &gt;&gt; /etc/apt/sources.list.d/nginx.list &amp;&amp; tempDir="\$(mktemp -d)" &amp;&amp; chmod 777 "\$tempDir" &amp;&amp; savedAptMark="\$(apt-mark showmanual)" &amp;&amp; apt-get update &amp;&amp; apt-get build-dep -y \$nginxPackages &amp;&amp; ( cd "\$tempDir" &amp;&amp; DEB_BUILD_OPTIONS="nocheck parallel=\$(nproc)" apt-get source --compile \$nginxPackages ) &amp;&amp; apt-mark showmanual   xargs apt-mark auto &gt; /dev/null &amp;&amp; { [ -z "\$savedAptMark" ]    apt-mark manual \$savedAptMark; } &amp;&amp; ls -lAFh "\$tempDir" &amp;&amp; ( cd "\$tempDir" &amp;&amp; dpkg-scanpackages . &gt; Packages ) &amp;&amp; grep "Package: !" "\$tempDir/Packages" &amp;&amp; echo "deb [ trusted=yes ] file://\$tempDir ./" &gt; /etc/apt/sources.list.d/temp.list &amp;&amp; apt-get -o Acquire::GzipIndexes=false update ;; esac &amp;&amp; apt-get install --no-install-recommends --no-install-suggests -y \$nginxPackages gettext-base &amp;&amp; apt-get remove --purge --auto-remove -y ca-certificates &amp;&amp; rm -rf /var/lib/apt/lists/* /etc/apt/sources.list.d/nginx.list &amp;&amp; if [ -n "\$tempDir" ]; then apt-get purge -y --auto-remove &amp;&amp; rm -rf "\$tempDir" /etc/apt/sources.list.d/temp.list; fi</pre>
7/23/19, 10:59 PM	RUN ln -sf /dev/stdout /var/log/nginx/access.log && ln -sf /dev/stderr /var/log/nginx/error.log
7/23/19, 10:59 PM	EXPOSE 80
7/23/19, 10:59 PM	STOPSIGNAL SIGTERM
7/23/19, 10:59 PM	CMD ["nginx" "-g" "daemon off;"]

11 commands

# Хранение артефактов: Anchore



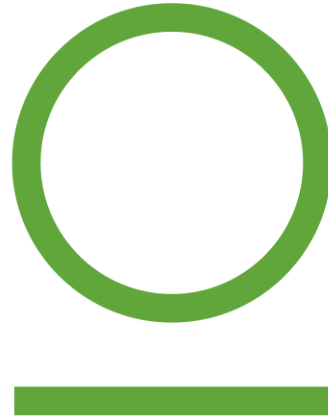
# Хранение артефактов: Anchore

- Анализ образов
- Управление на основе политик
- Оповещения
- Интеграция с системами CI/CD
- Высокая кастомизация
- Оркестрация

# Хранение артефактов: Anchore

- Сканирование уязвимостей
- Сканирование на наличие чувствительных данных
- Выявление всех известных пакетов ОС
- Сканирование 3d-party libraries
- Whitelist
- Blacklist
- Проверка Dockerfile
- Другие проверки ^\_^

# Хранение артефактов: Artifactory



# JFrog Artifactory

# Хранение артефактов: Artifactory

- Автоматизация и интеграция с большинством систем CI
- Высокая доступность
- Репликация
- Восстановление в случае аварии
- Расширяемость
- Внешние хранилища (Filestore Sharding, Checksum-based Storage, Hybrid, Redundancy)
- Поддержка

# Хранение артефактов: Artifactory

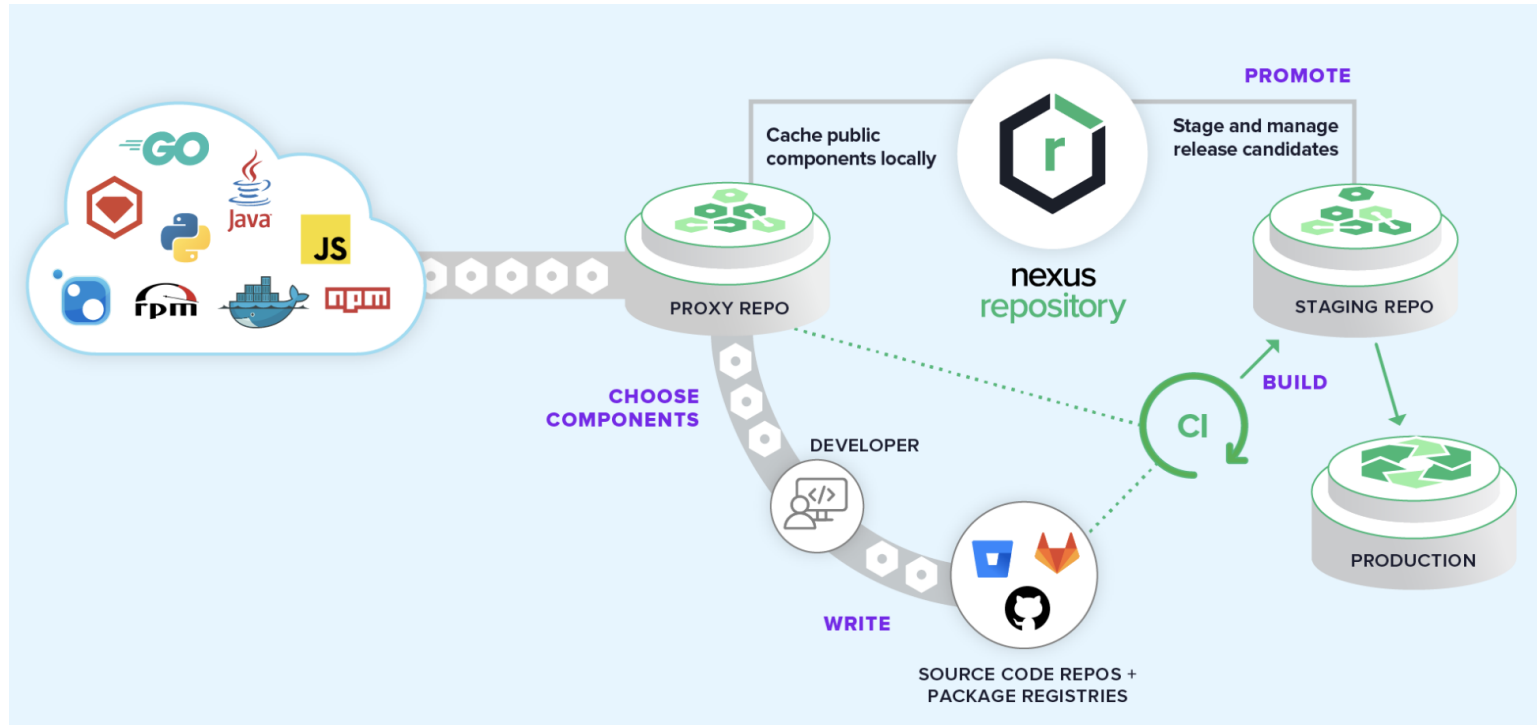
The screenshot displays the JFrog Artifactory web interface. The top navigation bar is green with the JFrog logo and the text "JFrog Artifactory". A dark sidebar on the left contains a vertical menu with icons and labels: Home, Artifacts, Packages, Search, Release Bundles, Builds, and Admin. The main content area is dark grey and features a search bar at the top with the placeholder text "Filter Menu...". Below the search bar, the content is organized into three columns of menu items:

- Repositories**
  - Local
  - Remote
  - Virtual
  - Distribution
  - Layouts
- Configuration**
  - General Configuration
  - JFrog Xray
  - Licenses
  - Property Sets
  - Proxies
  - HTTP Settings
  - Mail
  - High Availability
  - Artifactory Licenses
- Security**
  - Security Configuration
  - Users
  - Groups
  - Permissions
  - Access Tokens
  - LDAP
  - Crowd / JIRA
  - SAML SSO
  - OAuth SSO
  - HTTP SSO
  - SSH Server
  - Signing Keys
  - Trusted Keys
  - Certificates
- Services**
  - Backups
  - Maven Indexer
- Import & Export**
  - Repositories
  - System
- Advanced**
  - Support Zone
  - Log Analytics
  - System Logs
  - System Info
  - Maintenance
  - Storage
  - Config Descriptor
  - Security Descriptor

# Хранение артефактов: Nexus Repository



# Хранение артефактов: Nexus Repository



# Хранение артефактов: Nexus Repository

- Поддержка всех популярных форматов
- Расширенная поддержка Java (JVM)
- Поддержка дополнительных инструментов (Ansible, Docker, Chef, etc.)
- Безопасность
- Repository Health Check (RHC)
- Сканирование на безопасность и лицензионную чистоту
- Интеграция с LDAP, Atlassian Crowd, etc.

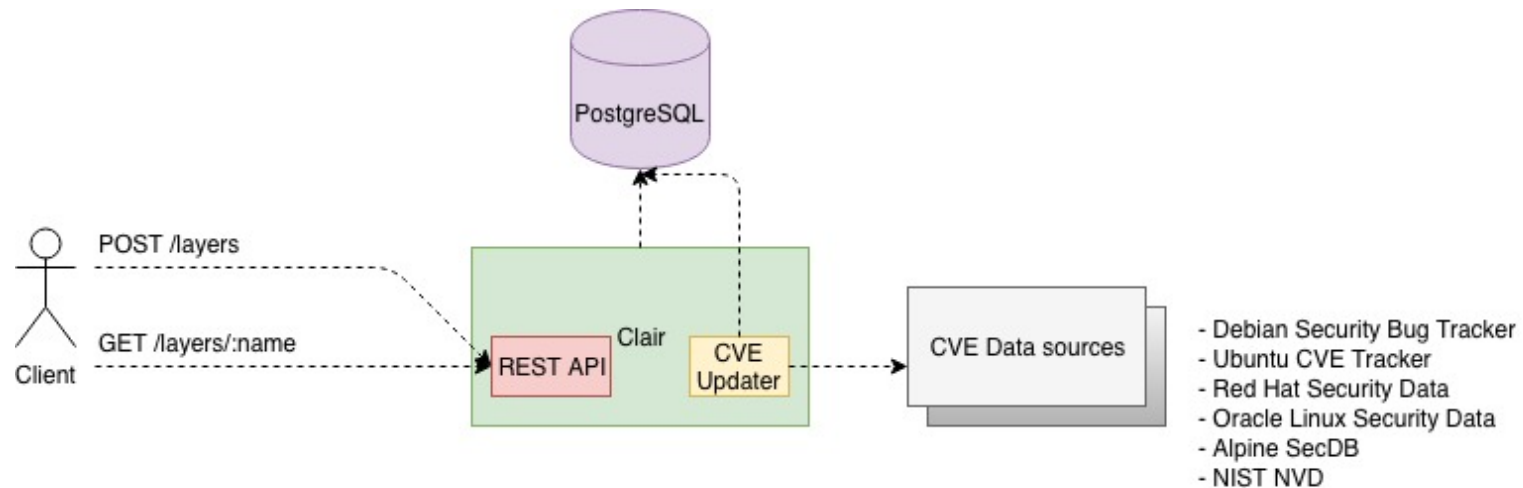
# Security: Clair



# Security: Clair

- REST API
- Статический анализ на уязвимости
- HTML отчеты

# Security: Clair



# Security: Snyk



**snyk**

# Security: Snyk

- Интеграция с Git
- Интеграция с инструментами CI/CD
- Сканирование в удаленных репозиториях (Scan image stored in ECR, Artifactory)
- BOM report
- Copyright info
- Поддерживаемые языки (Java, Python, Net, Ruby, JS)

# Security: Snyk

The screenshot displays the Snyk interface with the 'Issues' tab selected, showing 298 issues. The interface includes search bars for issues and projects, filters for 'Issue filters' and 'Last 90 days', and an 'Export' button. A table lists various security issues with their identifiers and the number of projects affected.

ISSUE	IDENTIFIERS	PROJECTS
<b>H</b> XML External Entity (XXE) Injection nokogiri 1.7.0.1	CWE-611	9 projects
<b>H</b> XML External Entity (XXE) Injection org.springframework:spring-web 3.2.6.RELEASE	CVE-2014-0225, CWE-611	2 projects
<b>H</b> Use of vulnerable libxml2 nokogiri 1.7.0.1	CVE-2017-0663, CVE-2017-7375, CVE-2017-7376, CVE-2017-9047, CVE-2017-9048, CVE-2017-9049, CVE-2017-9050, CWE-126, CWE-200, CWE-399, CWE-89	9 projects
<b>H</b> Uninitialized Memory Exposure npmconf 0.0.24	CWE-201	10 projects
<b>H</b> Server-Side Request Forgery (SSRF) paperclip 5.1.0	CVE-2017-0889, CWE-918	2 projects
<b>H</b> Sandbox Bypass jinja2 2.7.2	CVE-2016-10745, CWE-234	1 project

[Ссылка](#)

# Security: Snyk

```
dependancy_scanning:
  image: node:latest
  stage: test
  script:
# Install npm, snyk, and snyk-to-html
- npm install -g npm@latest
- npm install -g snyk
- npm install snyk-to-html -g
# Run snyk help, snyk auth, snyk monitor, snyk test to break build and out report
- snyk --help
- snyk auth $SNYK_TOKEN
- snyk monitor --project-name=goof-gitlab
- snyk test --json | snyk-to-html -o snyk_results.html

# Output report
artifacts:
when: always
paths:
- snyk_results.html
```

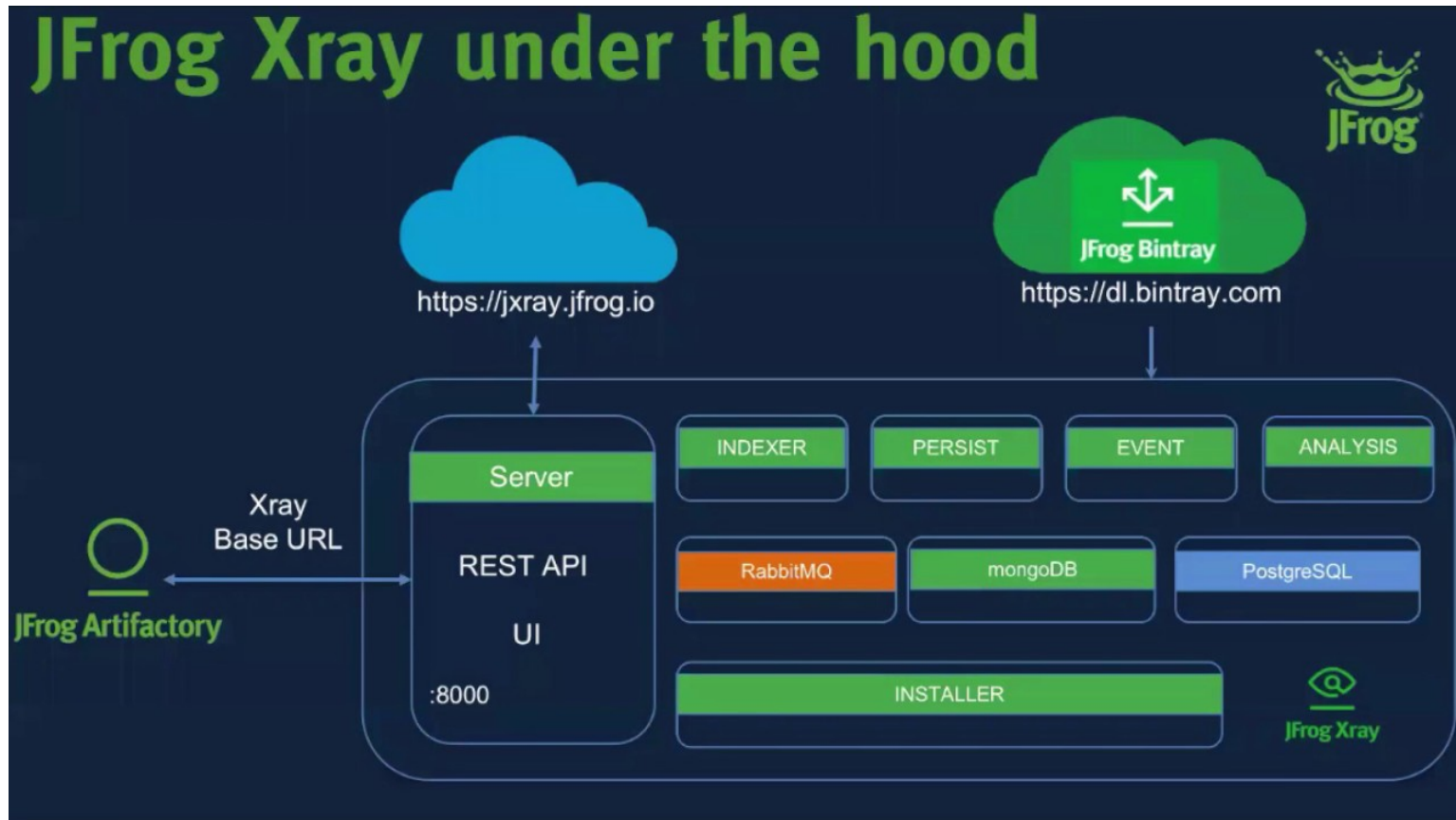


# JFrog Xray

# Security: Xray

- Универсальный инструмент для различных артефактов
- Высокая расширяемость
- Нативная поддержка Artifactory
- Готовая цепочка поставки артефактов
- Платная поддержка

# Security: Xray



# Security: Trivy



# Security: Trivy

- Обнаружение комплексных уязвимостей
- Простота использования и простота установки
- Высокая точность
- DevSecOps ^\_^

# Security: Trivy

```
bash-3.2$ trivy knqyf263/test-image:1.2.3
2019-05-13T15:19:03.912+0900 INFO Updating vulnerability database...
2019-05-13T15:19:05.983+0900 INFO Detecting Alpine vulnerabilities...
2019-05-13T15:19:05.987+0900 INFO Updating npm Security DB...
2019-05-13T15:19:07.048+0900 INFO Detecting npm vulnerabilities...
2019-05-13T15:19:07.048+0900 INFO Updating pipenv Security DB...
2019-05-13T15:19:08.507+0900 INFO Detecting pipenv vulnerabilities...
2019-05-13T15:19:08.508+0900 INFO Updating bundler Security DB...
2019-05-13T15:19:09.574+0900 INFO Detecting bundler vulnerabilities...
2019-05-13T15:19:09.575+0900 INFO Updating cargo Security DB...
2019-05-13T15:19:10.441+0900 INFO Detecting cargo vulnerabilities...
2019-05-13T15:19:10.441+0900 INFO Updating composer Security DB...
2019-05-13T15:19:11.649+0900 INFO Detecting composer vulnerabilities...

knqyf263/test-image:1.2.3 (alpine 3.7.1)
=====
Total: 26 (UNKNOWN: 0, LOW: 3, MEDIUM: 16, HIGH: 5, CRITICAL: 2)

+-----+-----+-----+-----+-----+-----+
| LIBRARY | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION | TITLE |
+-----+-----+-----+-----+-----+-----+
| curl    | CVE-2018-14618   | CRITICAL | 7.61.0-r0         | 7.61.1-r0     | curl: NTLM password overflow |
|         |                 |         |                   |               | via integer overflow         |
+-----+-----+-----+-----+-----+-----+
|         | CVE-2018-16839   | HIGH     |                   | 7.61.1-r1     | curl: Integer overflow leading |
|         |                 |         |                   |               | to heap-based buffer overflow |
|         |                 |         |                   |               | in Curl_sasl_create_plain_message() |
+-----+-----+-----+-----+-----+-----+
|         | CVE-2019-3822    |         |                   | 7.61.1-r2     | curl: NTLMv2 type-3 header   |
|         |                 |         |                   |               | stack buffer overflow       |
+-----+-----+-----+-----+-----+-----+
|         | CVE-2018-16840   |         |                   | 7.61.1-r1     | curl: Use-after-free when    |
|         |                 |         |                   |               | closing "easy" handle in    |
|         |                 |         |                   |               | Curl_close()                 |
+-----+-----+-----+-----+-----+-----+
|         | CVE-2018-16890   | MEDIUM  |                   | 7.61.1-r2     | curl: NTLM type-2 heap      |
|         |                 |         |                   |               | out-of-bounds buffer read   |
+-----+-----+-----+-----+-----+-----+
|         | CVE-2019-3823    |         |                   |               | curl: SMTP end-of-response  |
|         |                 |         |                   |               | out-of-bounds read         |
+-----+-----+-----+-----+-----+-----+
|         | CVE-2018-16842   |         |                   | 7.61.1-r1     | curl: Heap-based buffer     |
|         |                 |         |                   |               | over-read in the curl tool  |
|         |                 |         |                   |               | warning formatting         |
+-----+-----+-----+-----+-----+-----+
| git     | CVE-2018-19486   | HIGH     | 2.15.2-r0        | 2.15.3-r0     | git: Improper handling of    |
|         |                 |         |                   |               | PATH allows for commands to |
|         |                 |         |                   |               | be executed from...         |
+-----+-----+-----+-----+-----+-----+
```

# Security: Sonarqube



# Security: Sonarqube

- Поддержка языков (Kotlin, Scala, Ruby, Golang, etc.)
- Code Reliability
- Technical Debt
- Безопасность

# Security: Notary



# Security: TUF



The Update Framework



# Security: Portieris

- IBM Cloud Container Registry
- Quay.io
- Docker Hub

# Security: Portieris

```
apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
kind: ImagePolicy
metadata:
  name: allow-custom
spec:
  repositories:
    - name: "icr.io/*"
      policy:
        trust:
          enabled: true
          trustServer: "https://icr.io:4443" # Optional, custom trust server for
repository
```

# Security: Binary Authorization for GKE

