Инженерный вестник Дона, №2 (2026)
ivdon.ru/ru/magazine/archive/n2y2026/10739

Контейнеризация и сборка Android-приложений в условиях сетевой изолированности

В.А. Васькин, С.А. Ямашкин

[1]*Национальный исследовательский Мордовский государственный университет им. Н.П. Огарева, институт электроники и светотехники, Саранск*

Аннотация: Статья посвящена решению проблемы контейнеризации и сборки Android-приложений в условиях сетевой изолированности. Исследование направлено на разработку метода процесса сборки с использованием Docker и организации зеркалирования внешних зависимостей через веб-сервер nginx, посредством прокси репозиториев Nexus - менеджера репозиториев для управления артефактами. Предложенное решение позволит собирать артефакты для мобильной разработки в изолированных сетях без прямого доступа к внешним репозиториям, что повышает безопасность системы.
Ключевые слова: docker, контейнеризация, android, flutter, ci/cd, nginx, проксирование, сетевая изолированность, сборка приложений.

Введение

Современная разработка мобильных приложений и сборка артефактов для их распространения, требует постоянного доступа к многочисленным внешним ресурсам и зависимостям. Однако в условиях корпоративных сетей с ограниченным доступом к интернету традиционные подходы к сборке становятся невозможными [1,2]. Особую актуальность приобретает задача организации изолированной среды сборки с полным зеркалированием всех необходимых зависимостей. Использование контейнеризации позволяет создать воспроизводимую и переносимую среду сборки, проксирование запросов через nginx обеспечивает прозрачную работу с зависимостями, а использования менеджера репозиториев позволит хранить и управлять артефактами, которые будут доступны исключительно во внутренней изолированной сети.

Организация проксирования зависимостей

Одним из первых этапов решения задачи является организация высокодоступного прокси-сервера для обеспечения прозрачного перенаправления запросов к внешним репозиториям на внутренние ресурсы. Для реализации этой задачи был развернут и настроен веб-сервер nginx, который должен был заниматься проксированием зависимостей со следующих ресурсов [3]:

Тип репозитория	Имя репозитория	Remote storage
raw (proxy)	mobile-dl-google-com	https://dl.google.com
raw (proxy)	mobile-repo-maven-apache-org	https://repo.maven.apache.org
raw (proxy)	mobile-maven-google-com	https://maven.google.com
raw (proxy)	mobile-plugins-gradle-org	https://plugins.gradle.org
raw (proxy)	mobile-developer-huawei-com	https://developer.huawei.com
raw (proxy)	mobile-jitpack-io	https://jitpack.io
raw (proxy)	mobile-artifactory-external-vkpartner-ru	https://artifactory-external.vkpartner.ru
raw (proxy)	mobile-services-gradle-org	https://services.gradle.org
raw (proxy)	mobile-repo-gradle-org	https://repo.gradle.org
raw (proxy)	mobile-jcenter-bintray-com	https://jcenter.bintray.com
raw (proxy)	mobile-flutter-storage-proxy	https://storage.flutter-io.cn
dart (proxy)	mobile-dart-proxy	https://pub.dev
maven2 (proxy)	mobile-download-flutter-maven-proxy	https://storage.googleapis.com/download.flutter.io

Рис. 1. – Список зависимостей приложения.

Для каждого из доменов внешних репозиториев были сгенерированы само подписанные SSL-сертификаты, что обеспечило корректное установление защищенных соединений и избежание ошибок сертификации при работе инструментов сборки.

Серверные блоки nginx были настроены для каждого проксируемого домена с детальной конфигурацией параметров передачи данных и заголовками, необходимыми для корректной работы каждого блока [4]. Пример конфигурации для домена dl.google.com демонстрирует комплексный подход:

Пример конфигурации для домена dl.google.com:

```
upstream ag-nexus.vaskinva.ru {
    server 10.6.17.104:8081;
}

server {
    listen 443 ssl;
    server_name dl.google.com;

    ssl_certificate /etc/nginx/conf.d/certs/dl.google.com.crt;
    ssl_certificate_key /etc/nginx/conf.d/certs/dl.google.com.key;

    location / {
        proxy_pass http://ag-nexus.elocont.ru/repository/mobile-dl-google-com/;
        proxy_set_header Host dl.google.com;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

Для каждого из этих доменов в Nexus были созданы соответствующие proxy-репозитории с настроенными политиками кэширования и хранения артефактов. Это обеспечило полное покрытие всех зависимостей, необходимых для сборки артефактов мобильных приложений с использованием Flutter и нативных Android-инструментов [5,6].

Такой подход в совокупности позволяет создать надежную инфраструктуру для сборки приложений в условиях полной сетевой изолированности.

Сборка базового Docker-образа

Для обеспечения воспроизводимости процесса сборки артефактов в условиях сетевой изолированности было решено использовать docker образ, с предустановленным ПО. Данный образ предназначен для последующего использования в конвейерах непрерывной интеграции и доставки (CI/CD), обеспечивая детерминированное выполнение этапов сборки [7].

Критически важным аспектом подготовки образа является предварительное формирование полного набора компонентов Android SDK, включая платформы, инструменты сборки и системные образы, зависимости и предварительно настроенные конфигурационные файлы Gradle. Java Development Kit (JDK 17) служит базовой средой выполнения для Gradle. Для поддержки кроссплатформенной разработки реализуется интеграция с менеджером версий Flutter (FVM), обеспечивающим управление различными версиями Flutter SDK. Все инструкции, которые применены в образе представлены в следующей конфигурации:

```
FROM ubuntu:22.04

COPY Android /AndroidSDK_linux/Android/
COPY ./gradle/* /root/.gradle/

RUN apt-get update && apt-get install curl git unzip xz-utils zip clang cmake
ninja-build pkg-config libgtk-3-dev wget -y \
    libglu1-mesa libc6:amd64 libstdc++6:amd64 libbz2-1.0:amd64 libncurses5:amd64
lib32z1  -y

RUN wget https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.deb  &&
dpkg -i jdk-17_linux-x64_bin.deb && rm jdk-17_linux-x64_bin.deb

RUN curl -fsSL https://fvm.app/install.sh | bash

COPY flutter/ /flutter

ENTRYPOINT [ "tail","-f","/dev/null"]
```

Результатом сборки образа является полностью самодостаточная среда сборки, обеспечивающая детерминированное выполнение процессов компиляции артефактов мобильных приложений независимо от внешних факторов [8].

Настройка CI/CD процесса

Для автоматизации процессов сборки и публикации артефактов был разработан и внедрен CI/CD-процесс на платформе GitLab. Конфигурация

пайплайна включает последовательное выполнение взаимосвязанных этапов, обеспечивающих полную изолированность процесса сборки от внешних зависимостей.

Основой процесса сборки выступает предварительно подготовленный Docker-образ, содержащий полный набор инструментов для сборки артефактов. Ниже представлен job – единица gitlab, представляющая собой описательную инструкцию для сборки артефактов мобильного приложения [9,10]:

```
build-job:
  stage: build
  image: base_image:latest
  script:
    - pwd
    - echo "10.6.17.104 dl.google.com" | tee -a /etc/hosts
    - echo "10.6.17.104 repo.maven.apache.org" | tee -a /etc/hosts
    - echo "10.6.17.104 repo.gradle.org" | tee -a /etc/hosts
    - echo "10.6.17.104 jcenter.bintray.com" | tee -a /etc/hosts
    - echo "10.6.17.104 nexus.proxy.googleapis.ru" | tee -a /etc/hosts
    - cat /etc/hosts
    - |
      ./scripts/configure_services_google.sh
      fvm flutter build apk --profile --flavor internal --build-number=1
  only:
    - main
  tags:
    - ag-gitlab-ci-runner
```

В процессе выполнения пайплайна осуществляется перенаправление DNS-запросов к ключевым доменам внешних ресурсов на прокси-сервер внутри корпоративной сети.

Завершающим этапом является непосредственная сборка APK-файла с использованием инструментария Flutter. Сборка выполняется в профилирующем режиме, что позволяет оптимизировать производительность конечного приложения.

В выводе терминала видно, что flutter использует внешний репозиторий для установки зависимостей:

```
Flutter assets will be downloaded from https://ag-nexus.elocont.ru/repository/mobile-flutter-storage-proxy. Make sure you trust this source!
Downloading android-arm-profile/linux-x64 tools...                500ms
Downloading android-arm-release/linux-x64 tools...                260ms
Downloading android-arm64-profile/linux-x64 tools...              265ms
Downloading android-arm64-release/linux-x64 tools...              210ms
Downloading android-x64-profile/linux-x64 tools...                202ms
Downloading android-x64-release/linux-x64 tools...                184ms
Running Gradle task 'assembleInternalProfile'...
```

Рис. 2. – Установка зависимостей приложения из удаленного репозитория.

Весь процесс сборки выполняется на предварительно настроенных gitlab-runner-ax, обладающих функцией кэширования, которое позволит сократить время выполнения процесса сборки при повторных запусках. После окончания сборки, интерфейс Gitlab представит отчет об успешном выполнении задания с указанием пути сохраненного артефакта мобильного приложения.

Заключение

Разработанное решение демонстрирует эффективный метод организации процесса сборки артефактов Android-приложений в условиях полной сетевой изолированности. Комплексный подход, сочетающий контейнеризацию среды сборки, проксирование внешних зависимостей и автоматизацию CI/CD-процессов, позволяет создать замкнутую систему разработки, не требующую прямого доступа к внешним ресурсам.

Предложенная методика обладает значительным потенциалом для адаптации в различных корпоративных средах с строгими требованиями к информационной безопасности. Важным преимуществом разработанного подхода является его способность обеспечивать непрерывную разработку мобильных приложений без компромиссов в отношении требований сетевой изоляции.

Литература

1. Маркелов К. Д. Использование docker-контейнеров для сборки Android-приложений //Молодой ученый. 2021. № 18. С. 44-47.

2. Топалов Н. К. Контейнеризация приложений: преимущества Docker и Kubernetes //Молодой ученый. 2024. № 51 (550). С. 22.

3. Бобров А. В., Рубашенков А. М. Настройка прокси-сервера Nginx // Academy. 2019. № 5 (44). С. 24-27.

4. Павелкина К. И., Федотова Р. И. Прокси-сервер как средство обеспечения информационной безопасности //Информационные системы и технологии: управление и безопасность. 2013. № 2. С. 298-304.

5. Калиневич Н., Гильванов Р. Г. Разработка кросс-платформенных приложений на языке dart при помощи фреймворка flutter // Интеллектуальные технологии на транспорте. 2021. № 4 (28). С. 21-27.

6. Рудин П. И., Хромых Е. А. Технологии разработки android приложений //Математическое моделирование процессов и систем Материалы XI Международной молодежной научно-практической конференции. 2021. С. 126.

7. Бакшанский В. Д., Серышев А. С., Замотайлова Д. А. Автоматизация развертывания приложений // Цифровизация экономики: направления, методы, инструменты. 2021. С. 264-267.

8. Шатунов А. Е. Моделирование работы серверного программного обеспечения // Политехнический молодежный журнал. 2019. № 7. С. 2-2.

9. Безпятый М. В. Автоматизация и оптимизация процессов разработки и развертывания в DEVOPS: применение современных методов и инструментов // Инновации и инвестиции. 2023. № 7. С. 458-464.

10. Курчевская О. В. Настройка CI/CD для Android проектов с использованием Github Actions // Universum: технические науки. 2023. № 12-1 (117). С. 16-18.

References

1. Markelov K. D. Molodoj ucheny`j. 2021. No. 18. pp. 44-47.
2. Topalov N. K. Molodoj ucheny`j. 2024. No. 51 (550). P. 22.
3. Bobrov A. V., Rubashenkov A. M. Academy. 2019. No. 5 (44). pp. 24-27.
4. Pavelkina K. I., Fedotova R. I. Informacionnye sistemy i texnologii: upravlenie i bezopasnost`. 2013. No. 2. pp. 298-304.
5. Kalinevich N., Gilvanov R. G. Intellektualny`e texnologii na transporte. 2021. No. 4 (28). pp. 21-27.
6. Rudin P. I., Xromyx E. A. Matematicheskoe modelirovanie processov i sistem. Materialy XI Mezhdunarodnoj molodezhnoj nauchno-prakticheskoj konferencii. 2021. P. 126.
7. Bakshanskij V. D., Seryshev A. S., Zamotajlova D. A. Cifrovizaciya ekonomiki: napravleniya, metody, instrumenty`. 2021. pp. 264-267.
8. Shatunov A. E. Politexnicheskij molodezhny`j zhurnal. 2019. No. 7. pp. 2-2.
9. Bezpyaty`j M. V. Innovacii i investicii. 2023. No. 7. pp. 458-464.
10. Kurchevskaya O. V. Universum: texnicheskie nauki. 2023. No. 12-1 (117). pp. 16-18.

Авторы согласны на обработку и хранение персональных данных.

Дата поступления: 23.12.2025

Дата публикации: 24.01.2026