Инструкция по установке программного обеспечения "Программный комплекс «Цифровые гарантии»"

Оглавление

1. ВВЕДЕНИЕ

Программное обеспечение "Программный комплекс «Цифровые гарантии»" разработано для применения банковскими организациями при оказании своим клиентам услуг по выдаче банковских гарантий.

Программное обеспечение предназначено для хранения, учёта и систематизации данных, связанных с выдачей банковских гарантий.

С помощью программного обеспечения банки могут масштабировать свой бизнес по количеству выдаваемых банковских гарантий без дополнительных операционных затрат на обслуживание процессов.

Программное обеспечение "Программный комплекс «Цифровые гарантии»" позволяет Клиенту (его агенту):
- создавать и направлять заявки в банк на выдачу банковских гарантий;
- согласовывать с банком параметры банковских гарантий;
- подписывать электронной подписью документы с банком;
- формировать систематизированные отчёты об операциях, совершенных с банком.

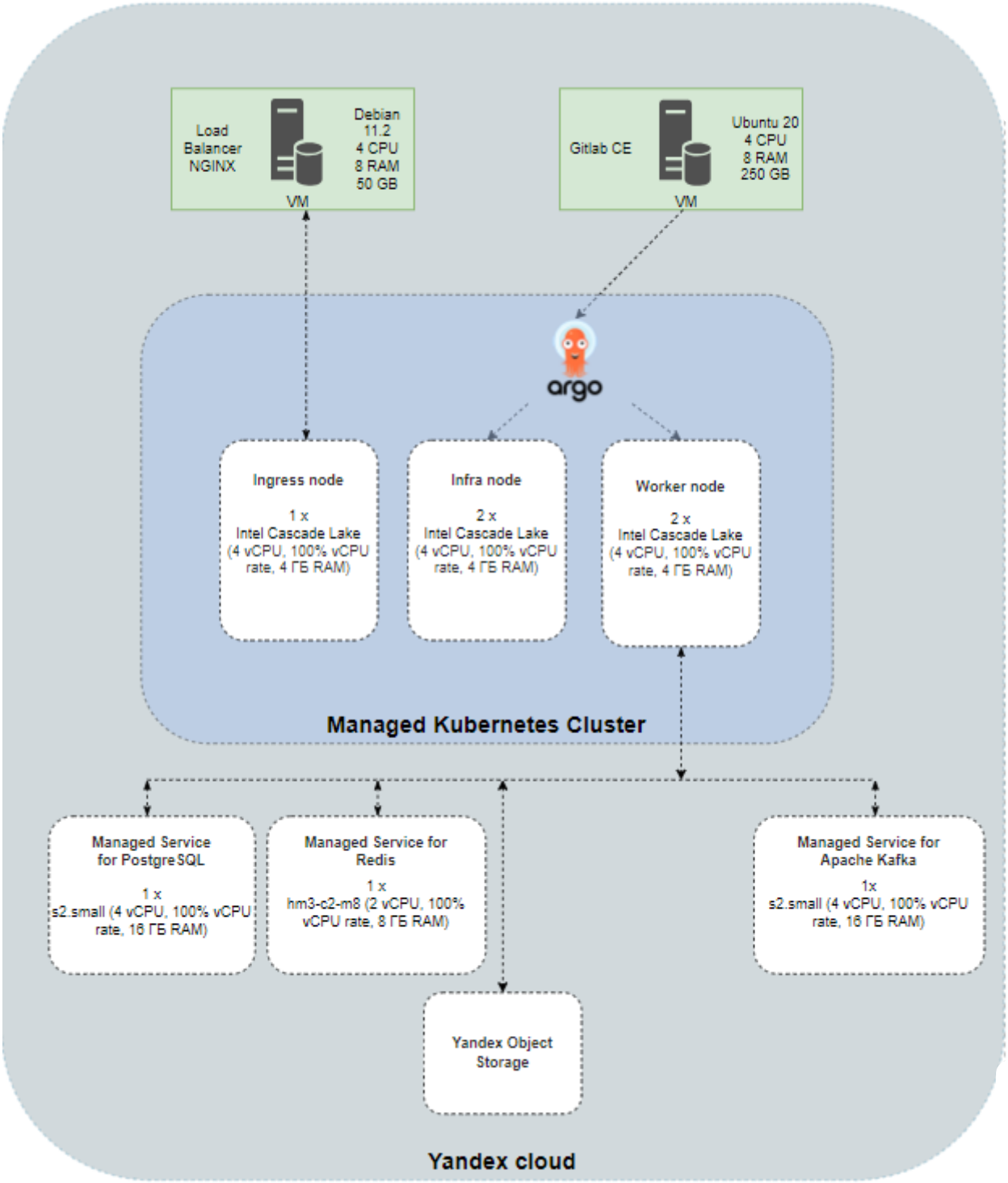
Программное обеспечение "Программный комплекс «Цифровые гарантии»" позволяет банкам:
- организовать выдачу банковских гарантий;
- сопровождать выданные банковские гарантии;
- управлять портфелем выданных банковских гарантий;
- хранить и учитывать информацию о контрагентах, их заявках; суммах задолженности и оплатах счетов;
- формировать систематизированные отчёты с выгрузкой данных в иные программы и/или сервисы.

Программное обеспечение "Программный комплекс «Цифровые гарантии»" поставляется клиенту (заказчику, покупателю) в формате облачного решения: программа "Программный комплекс «Цифровые гарантии»" и её данные размещаются на серверах клиента, но могут размещаться и на серверах компаний, предоставляющих клиенту услуги дата-центра (например, ООО «Селектел», ООО «ВК», ООО «Яндекс.Облако» и т.д.).

Программное обеспечение может также размещаться на серверах правообладателя программного обеспечения "Программный комплекс «Цифровые гарантии»".

2. СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Схема программного обеспечения



ПО состоит из 2-х частей:

- клиента, реализованного с использованием Vue3, TypeScript, HTML/CSS/JavaScript;
- сервера, реализованного с использованием .NET 7.0 и языка программирования C#.

Функционал реализован в клиент-серверной архитектуре.

3. ОПИСАНИЕ СЕРВИСОВ И ИХ НАЗНАЧЕНИЕ

backend
Сервис предоставляет REST интерфейс для получения и обновления данных с целью выдачи банковских гарантий. Сервис реализует бизнес логику и использует другие сервисы для автоматизации процесса выдачи гарантии.

frontend
Сервис предоставляет Клиентам, Агентам возможность взаимодействовать с Личным кабинетом Клиента/Агента. Пользователям предоставляется возможность заводить, заполнять и отправлять в банк на рассмотрение заявки на выдачу банковских гарантий.

frontend-int
Сервис предоставляет сотрудникам возможность взаимодействовать с Личным кабинетом сотрудника. Пользователям предоставляется возможность обслуживать заявки с целью выдачи банковских гарантий.

keycloak
Сервис аутентификации и авторизации пользователей, реализует функционал Single sign-on. Keycloak интегрирован с Active Directory для получения актуального списка пользователей и ролей.

keycloak-ext
Сервис аутентификации и авторизации Клиентов, Агентов, реализует функционал Single sign-on.

messaging
Сервис предоставляет бинарный протокол общения поверх TCP для отправки уведомлений Клиентам и сотрудникам по таким каналам как СМС и Email. Сервис интегрирован с внешними сервисами партнеров:

- для рассылки СМС - SMSЦентр,
- рассылки Email - ExpertSender.

Рассылка Email сотрудникам производится через внутренний SMTP сервер.

focus-api
Сервис предоставляет бинарный протокол общения поверх TCP (обмен сообщениями) для запроса выписок ЕГРЮЛ/ЕГРИП.

credit-registry
Сервис предоставляет REST интерфейс для осуществления запросов в БКИ. Платформенный сервис БКИ интегрирован с сервисом БКИ Банка.

multitender
Сервис предоставляет REST интерфейс для осуществления запросов к открытым источникам по закупкам.

signature-validator
Сервис предоставляет REST интерфейс для осуществления запросов, участвующих в процессе входа пользователя по ЭП. Так же сервис предоставляет бинарный протокол общения поверх TCP (общем сообщениями) для реализации механизма проверки подписи.

scoring
Сервис предоставляет REST интерфейс для осуществления Скоринга Клиента по СТОП-факторам для заведения заявок на выдачу банковской гарантии. Результаты Скоринга хранятся в базе данных Программного обеспечения "Программный комплекс «Цифровые гарантии»".

reports
Сервис предоставляет REST интерфейс для генерации документов из готовых шаблонов. А также реализует непосредственный функционал генерации документов. Сервис генерирует как документы, необходимые для работы над Заявкой Клиента.

dadata
Сервис предоставляет REST интерфейс для подключения Дадата. Реализует функционал адаптера.

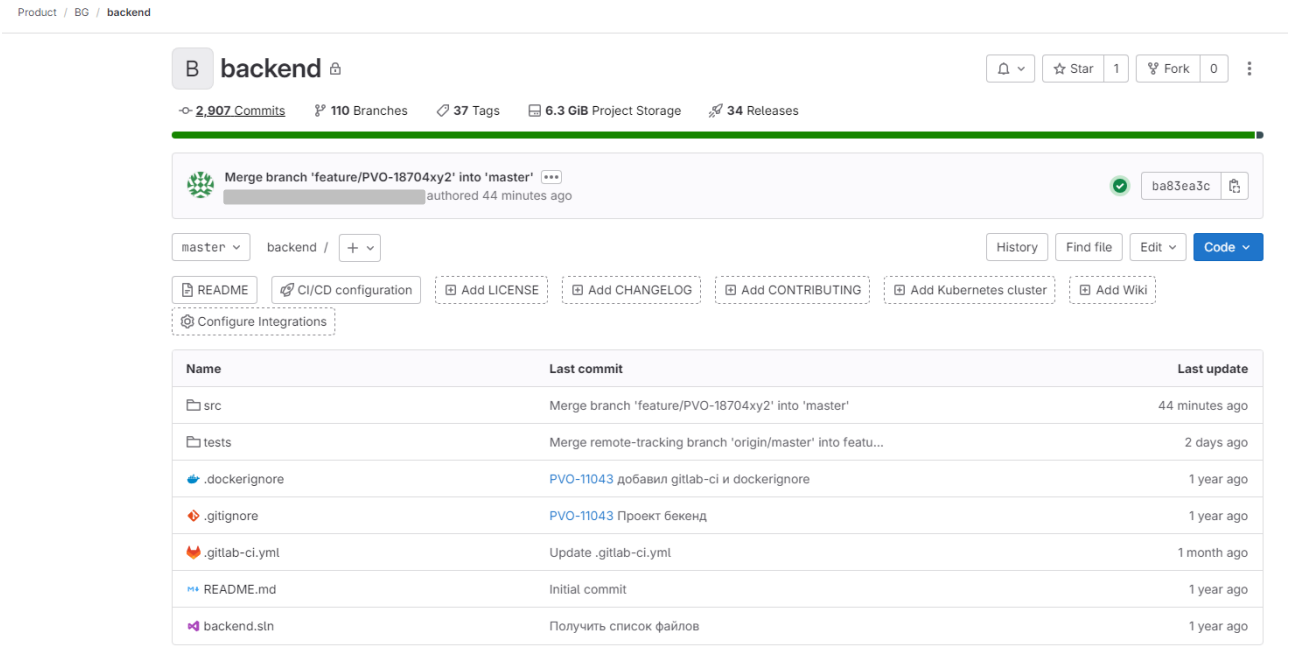
4. ПРОЦЕСС УСТАНОВКИ ПО

После приобретения заказчику предоставляется доступ к репозиторию с экземпляром ПО или предоставляется ссылка для загрузки файлов для установки (развертывания ПО) в том числе с использованием развернутого репозитория на мощностях заказчика.

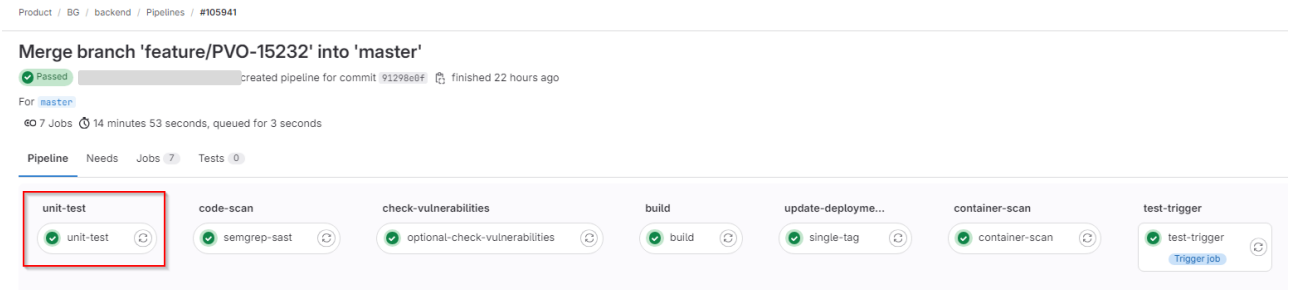
Далее расписаны шаги сборки и “деплоя” сервиса в кластер k8s на примере - backend, применимые к остальным частям ПО.

Сборка, сканирование и “деплой” осуществляются средствами ci-cd с использованием git-репозитория, например, git-репозитория «Gitlab Comminuty Edition», на примере которого приведено дальнейшее описание.

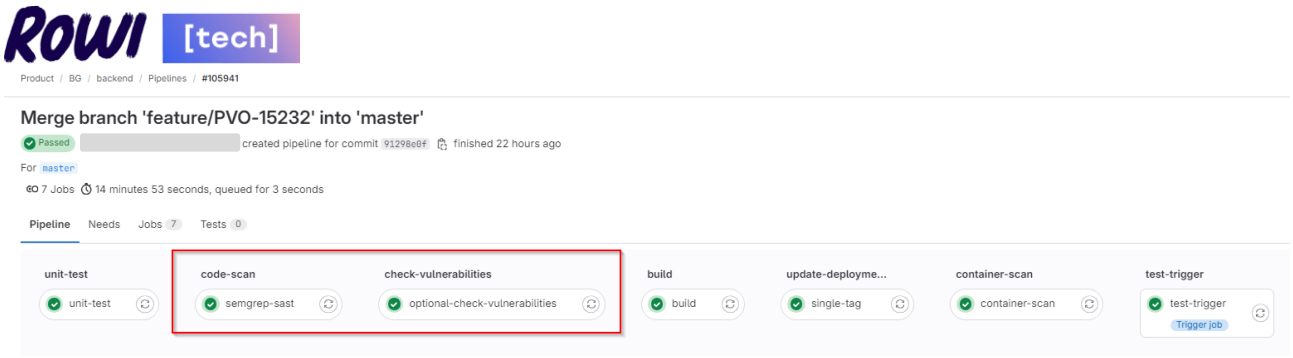
Файлы, необходимые для установки (развертывания) ПО размещаются в упомянутом репозитории.



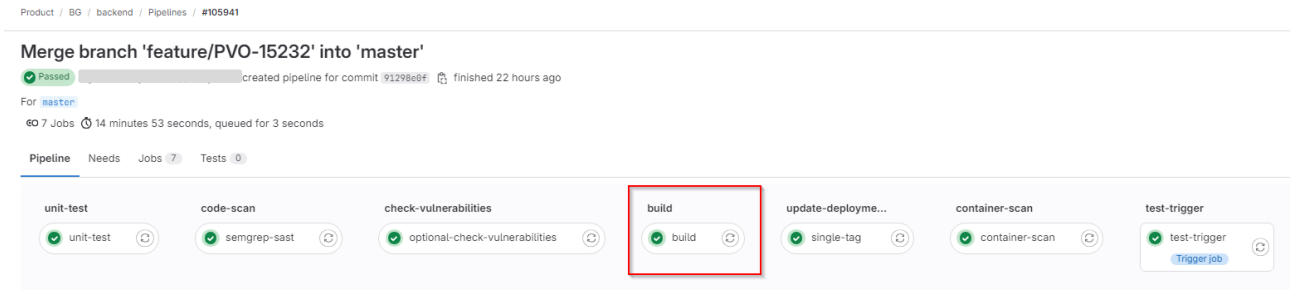
На первом этапе запускается шаг “unit-test“, выполняющий тестирование кода на наличие ошибок.



Следующим этапом после тестирования – запускаются шаги “code-scan” и “check-vulnerabilities”, в котором выполняется сканирование исходного кода различными методами в зависимости от языка, на котором написаны исходный код и расширения манифестов кода.



Следующим этапом следует сборка кода средствами Gitlab.



Результатом сборки являет docker-образ, который размещается в container_registry самого проекта.

После того, как готов docker- image, средствами Gitlab происходит коммит в проект с helm-чартами нужного проекта.

demo / bg / backend / **Deployment.yml** — Find file | Blame | History | Permalink

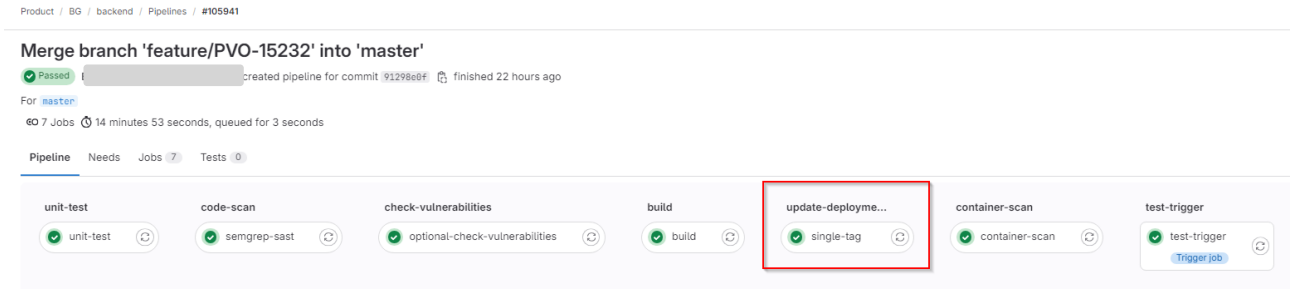
Update file Deployment.yml
[redacted] authored 1 month ago — 2c4cd1fb

Code owners Assign users and groups as approvers for specific file changes. Learn more.

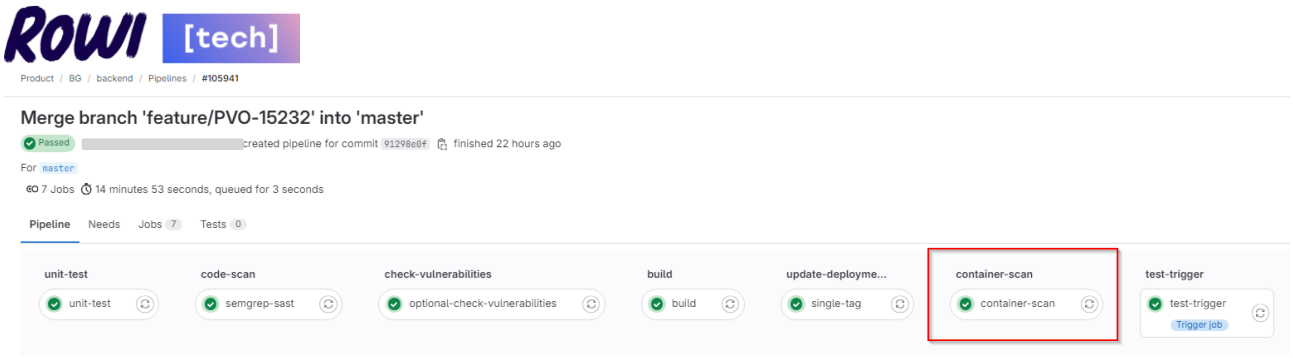
Deployment.yml 1.54 KiB — Blame | Edit | Lock | Replace | Delete

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
  labels:
    app.kubernetes.io/name: backend
spec:
  replicas: 1
  revisionHistoryLimit: 0
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 0
      maxSurge: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: backend
  template:
    metadata:
      labels:
        app.kubernetes.io/name: backend
    spec:
      volumes:
        - name: backend
          configMap:
            name: backend
      imagePullSecrets:
        - name: registry-credentials
      containers:
        - name: backend
          env:
            - name: ASPNETCORE_URLS
              value: 'http://+:80'
            - name: ASPNETCORE_ENVIRONMENT
              value: 'stage'
          image: gitlab.service.rowi.tech:80/product/bg/backend:stage.master.6740c616
          imagePullPolicy: Always
          volumeMounts:
            - name: backend
```

Ниже выделен данный шаг процесса ci-cd.



Container scan – это сканирование каждого слоя docker images на известные уязвимости. Уязвимости делятся на категории: critical, high, medium и low.



Финальным шагом “деплоя” сервиса в кластер k8s – является непосредственная синхронизация состояния сущностей (манифестов) с использованием GitOps инструмента ArgoCD.