**Оглавление**

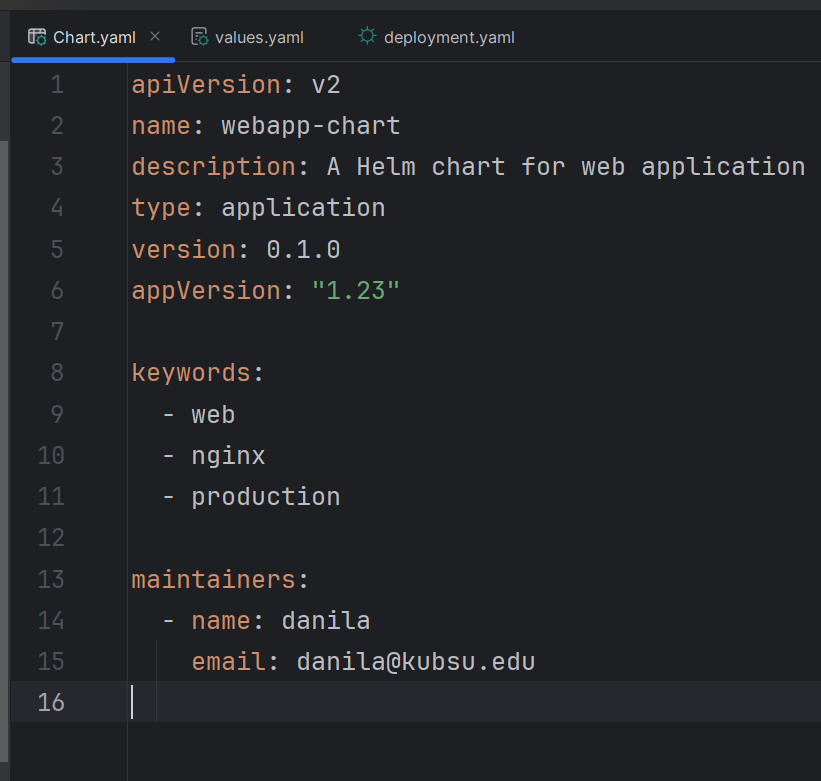
[Задание 11.1 Создание кастомного Helm Chart 1](#_Toc215926994)

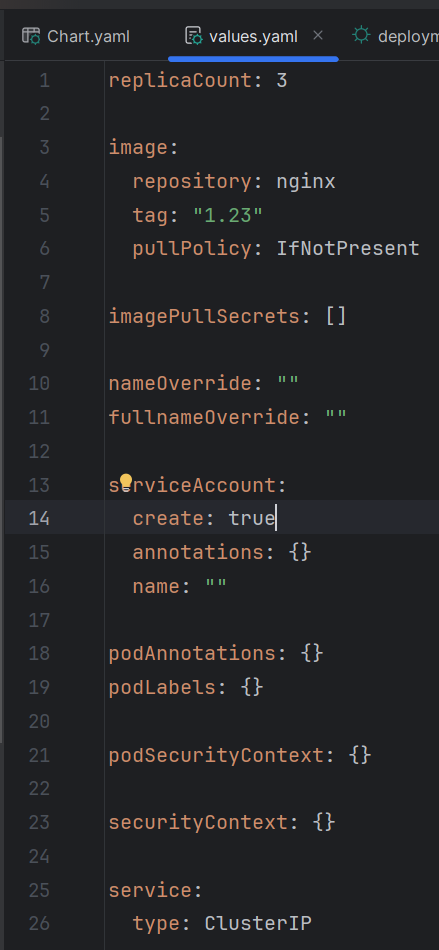
[Задание 11.2. Шаблонизация с условиями и циклами 17](#_Toc215926995)

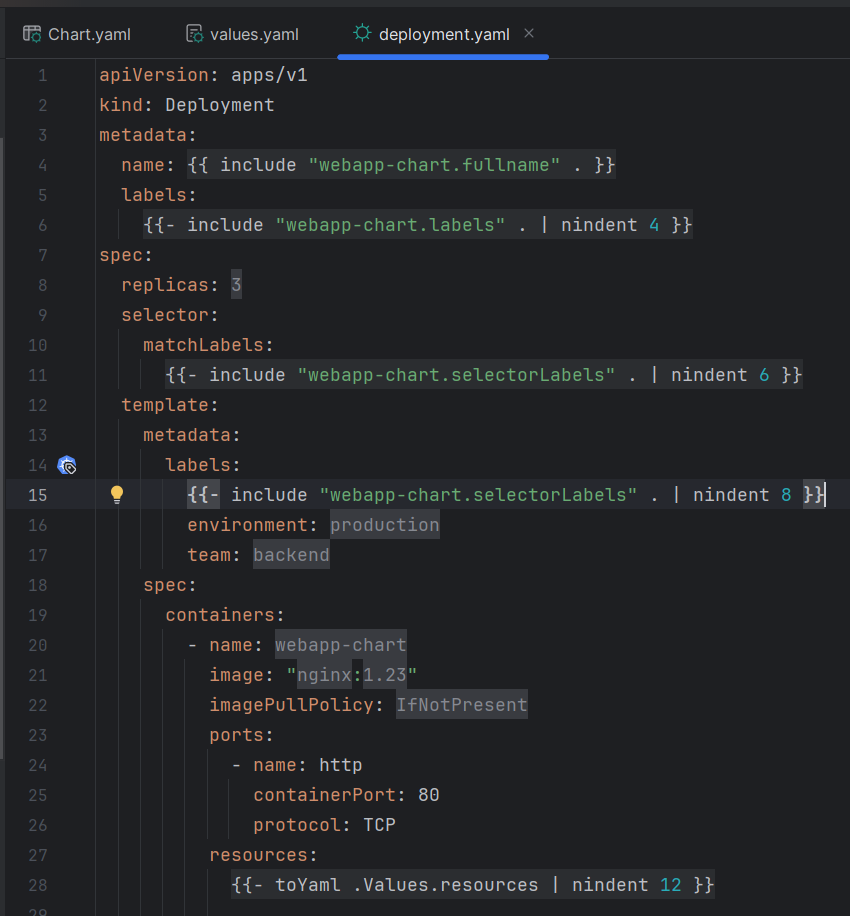
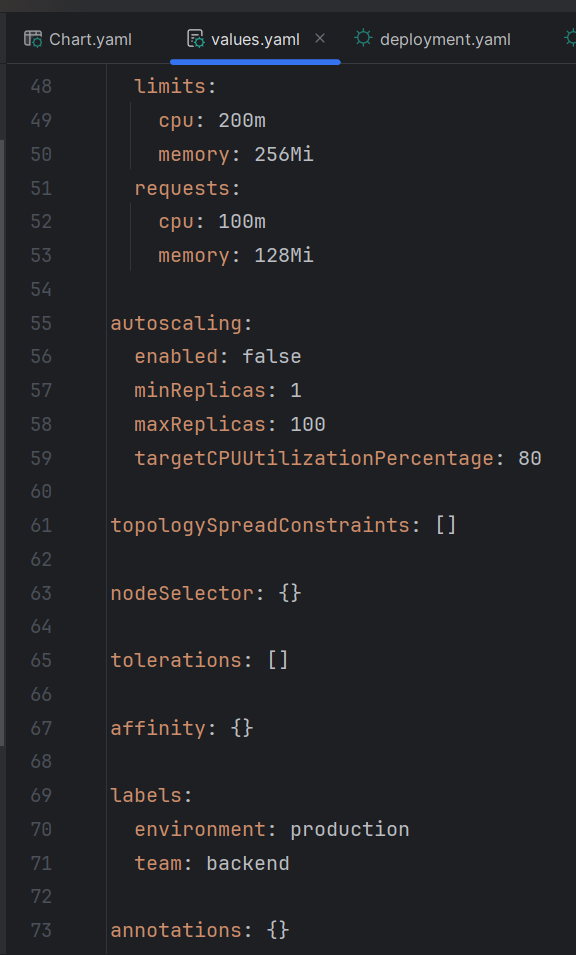
[Задание 11.3. Многофайловая конфигурация и ConfigMap 24](#_Toc215926996)

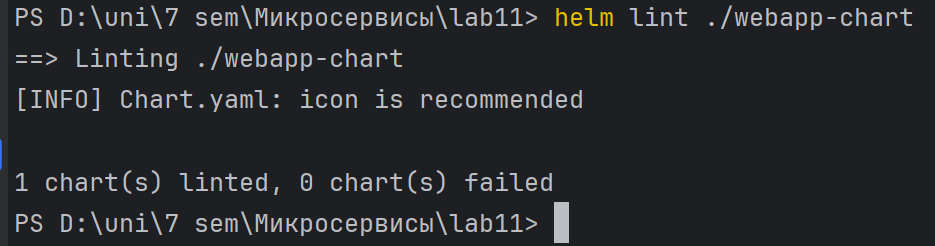
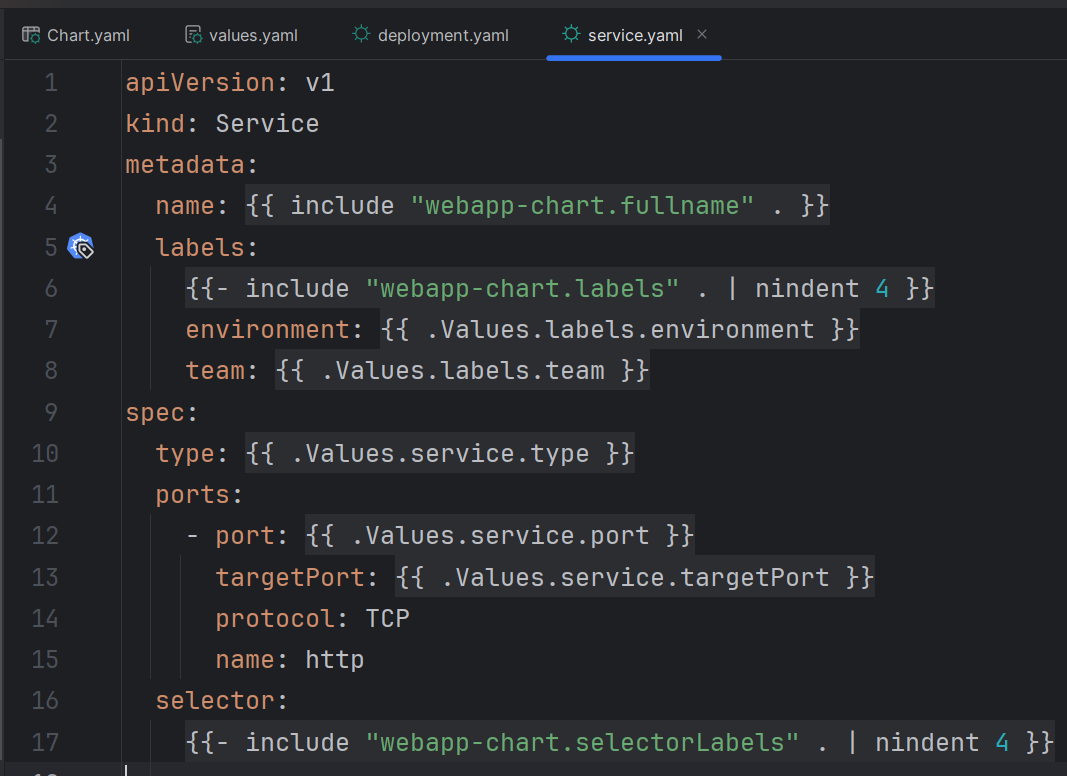
[Задание 11.4. Диагностика и отладка Helm Chart 31](#_Toc215926997)

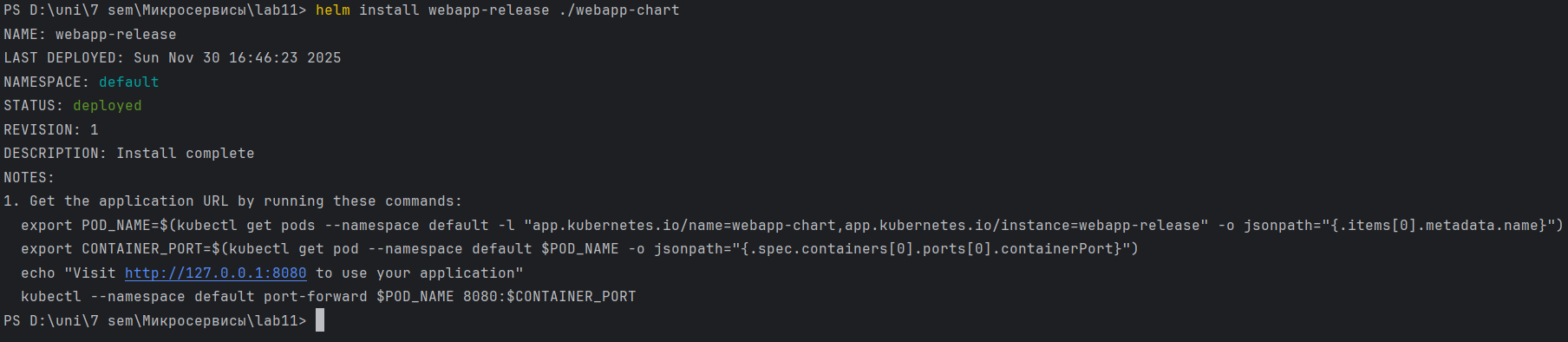
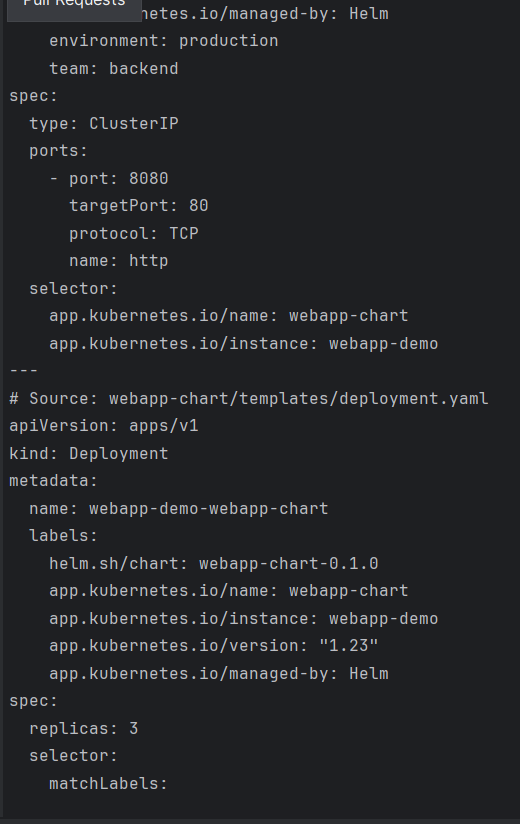
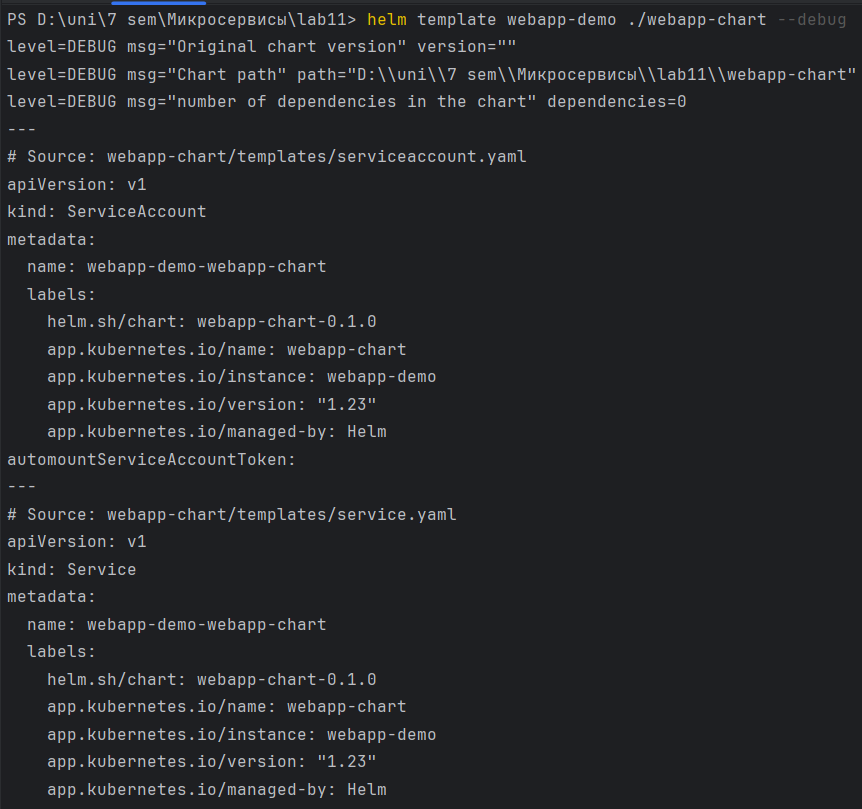
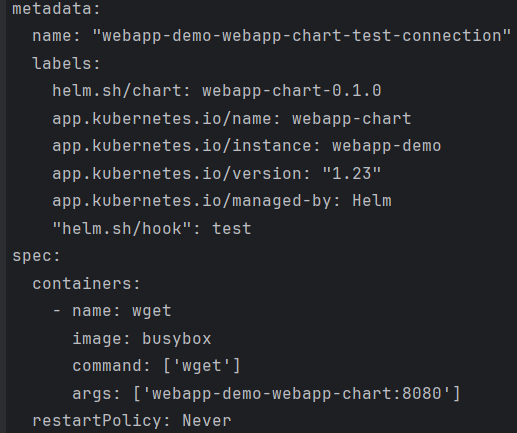
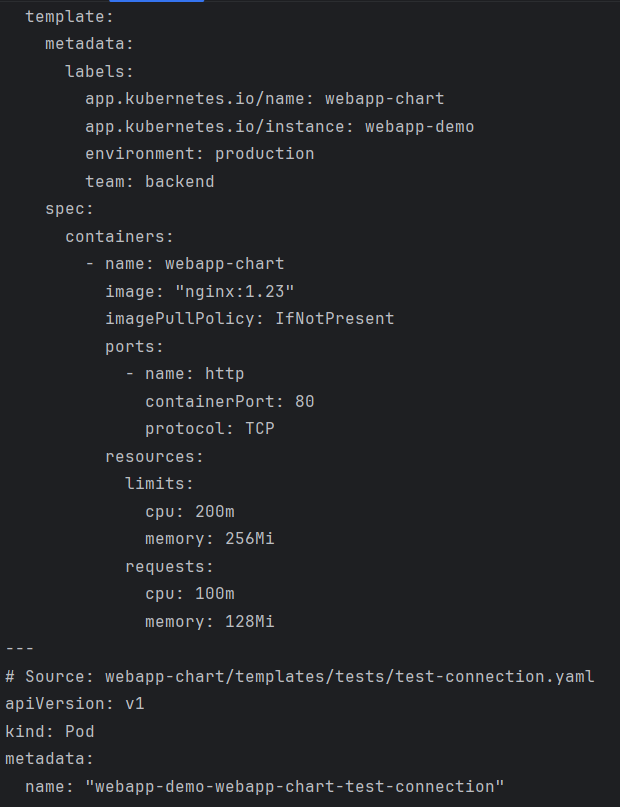
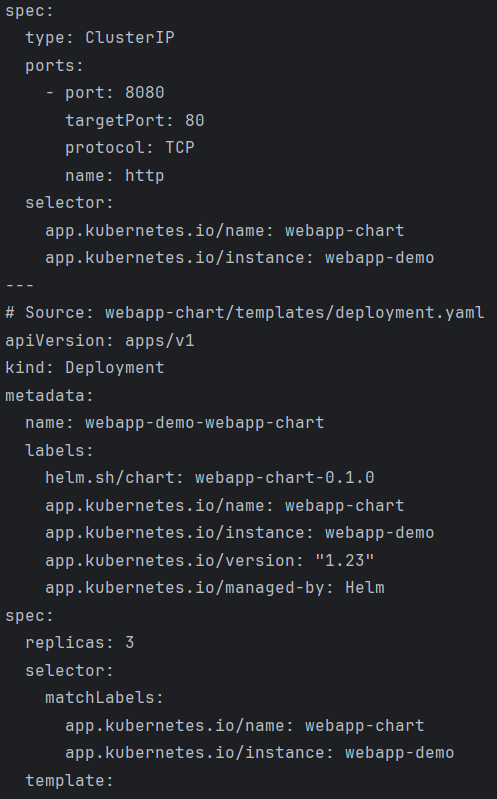
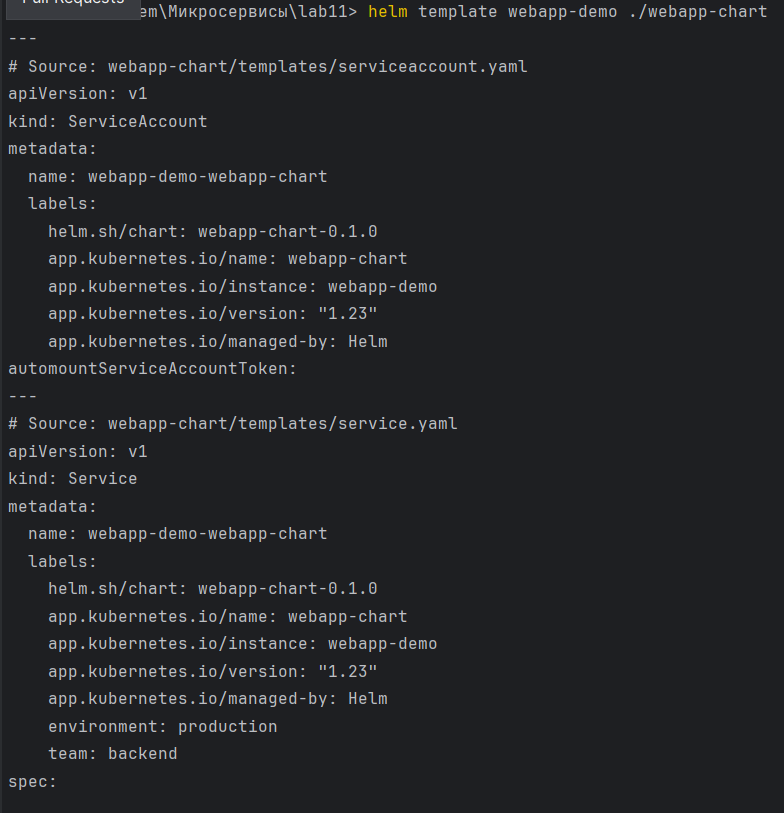
## Задание 11.1 Создание кастомного Helm Chart

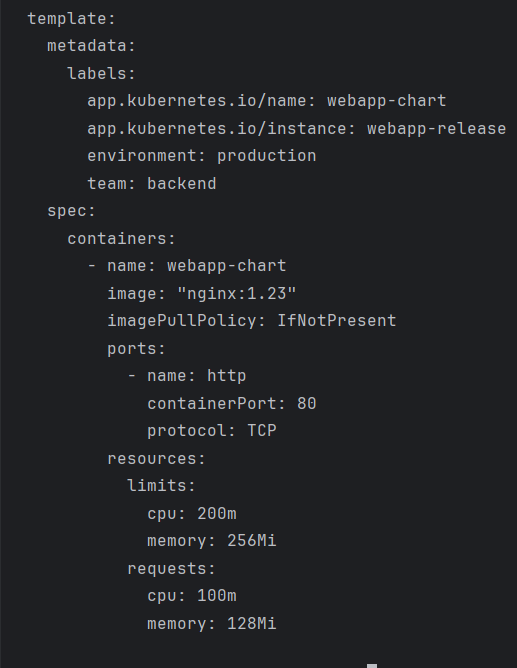
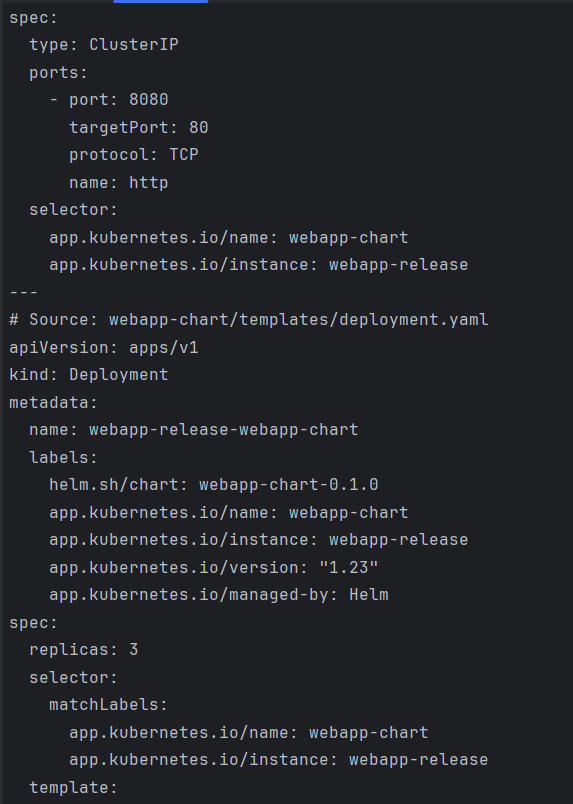
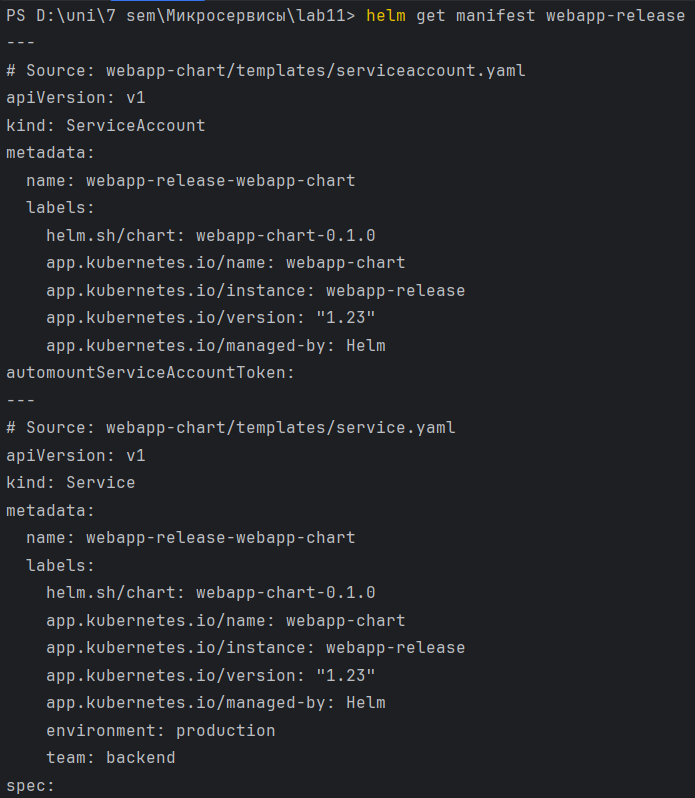


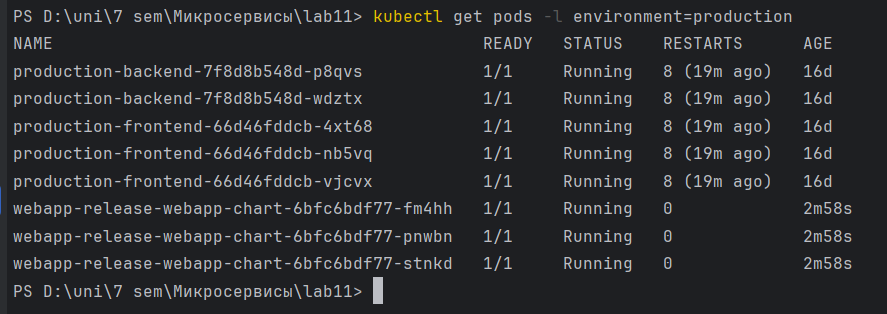
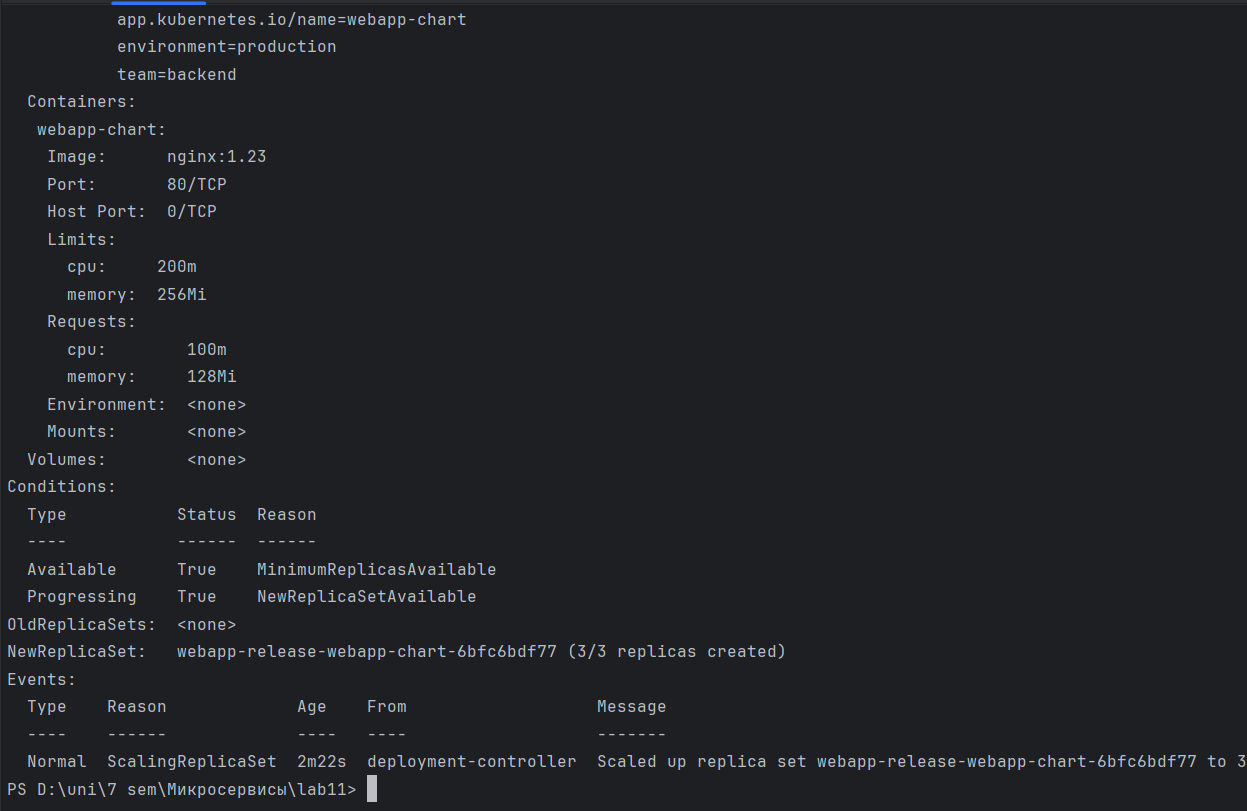
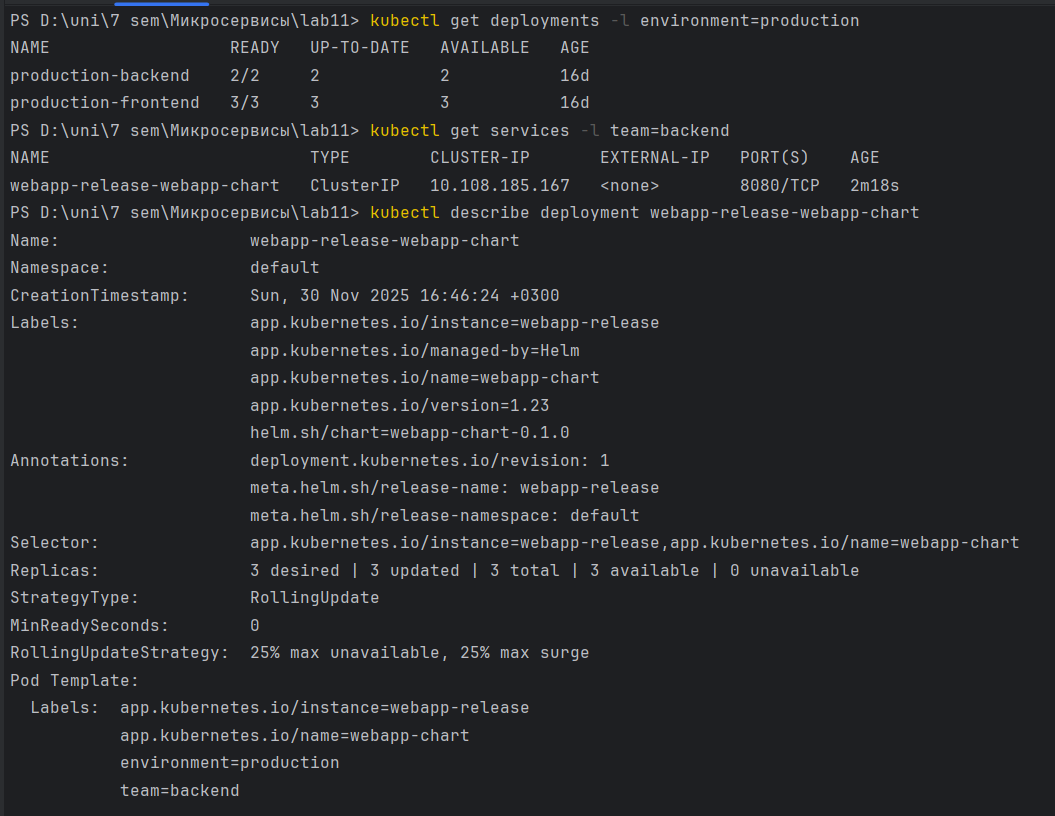




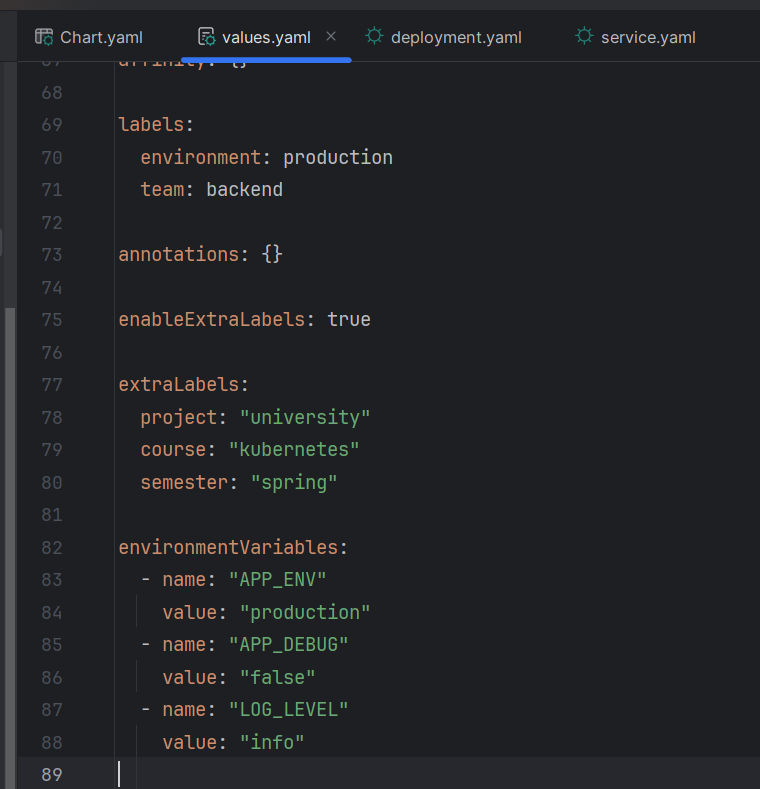


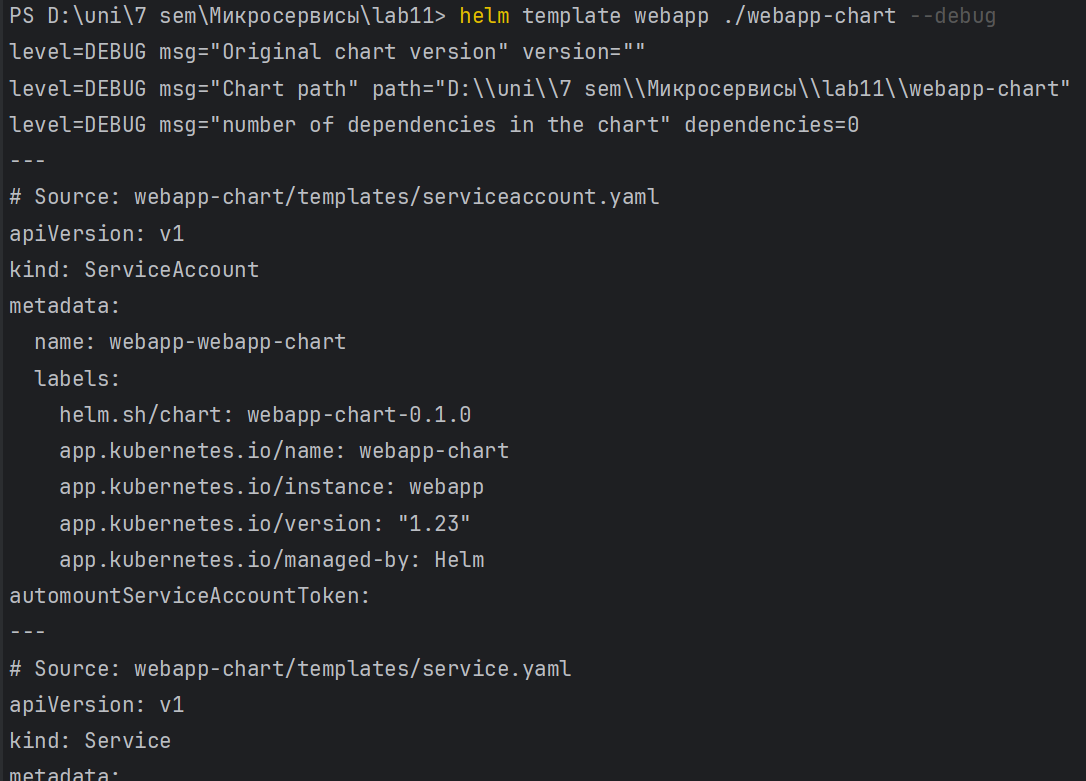
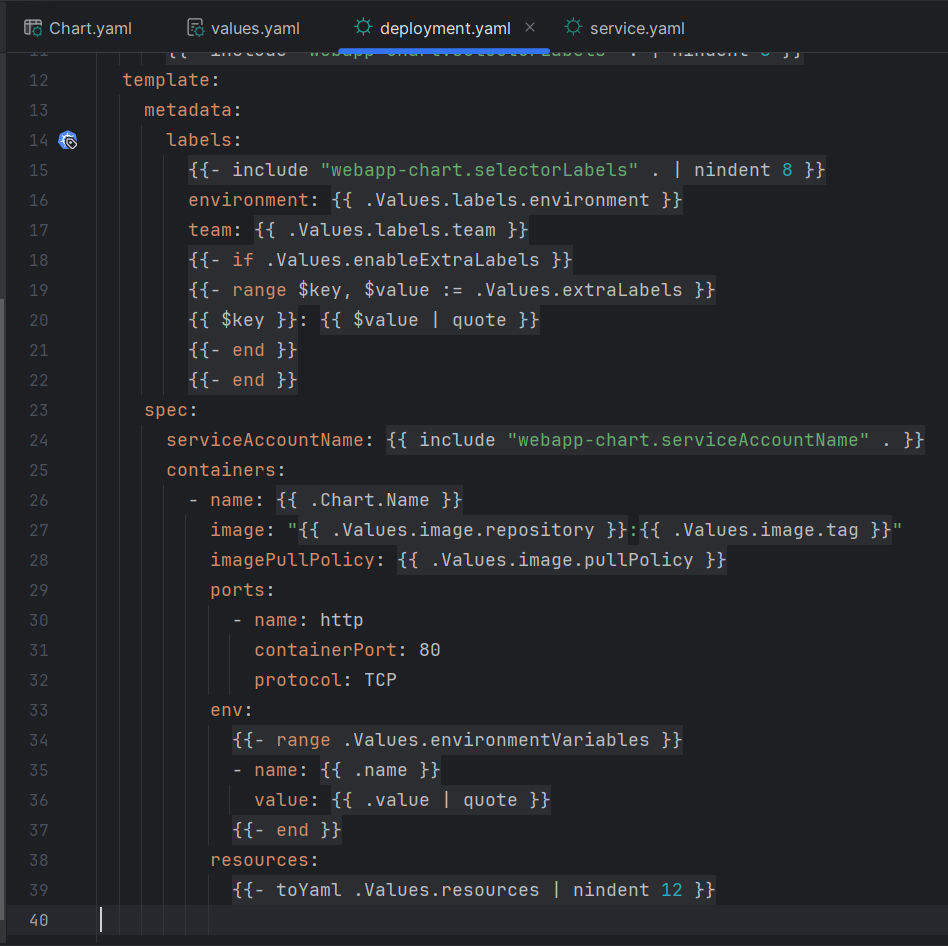


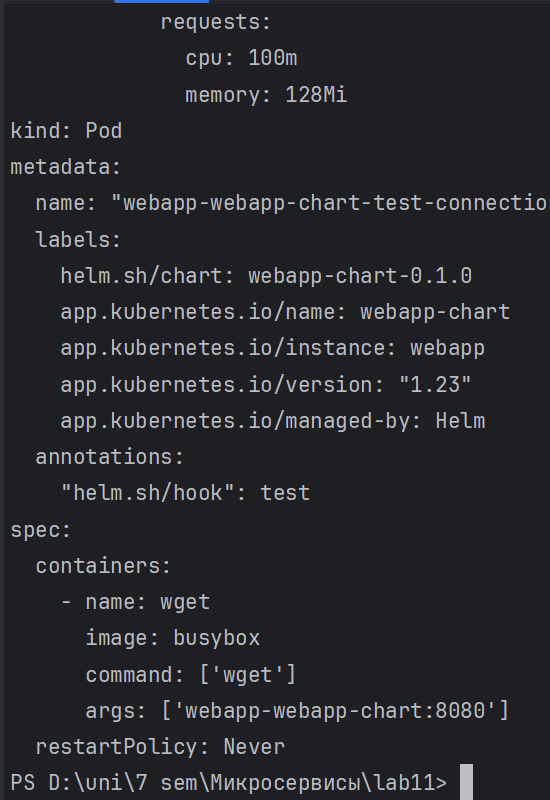
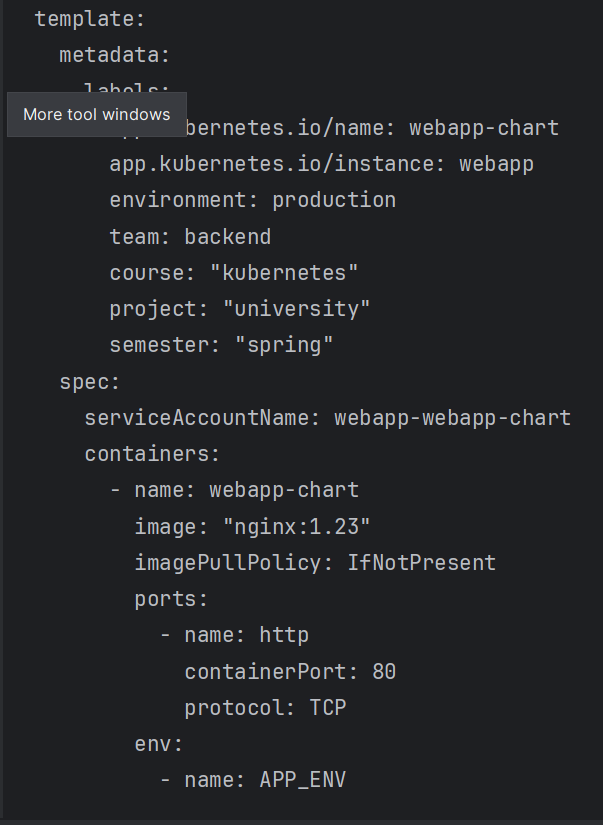
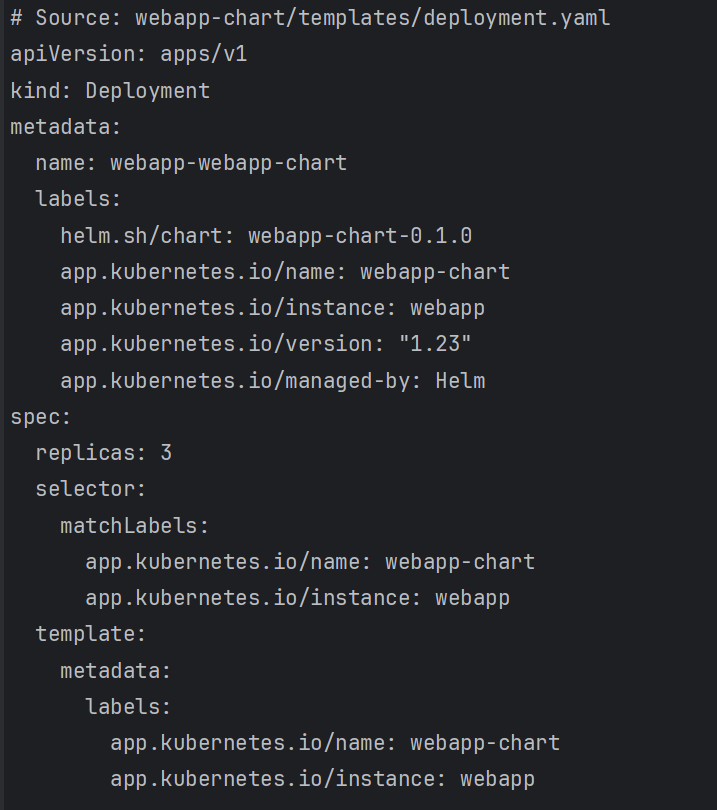
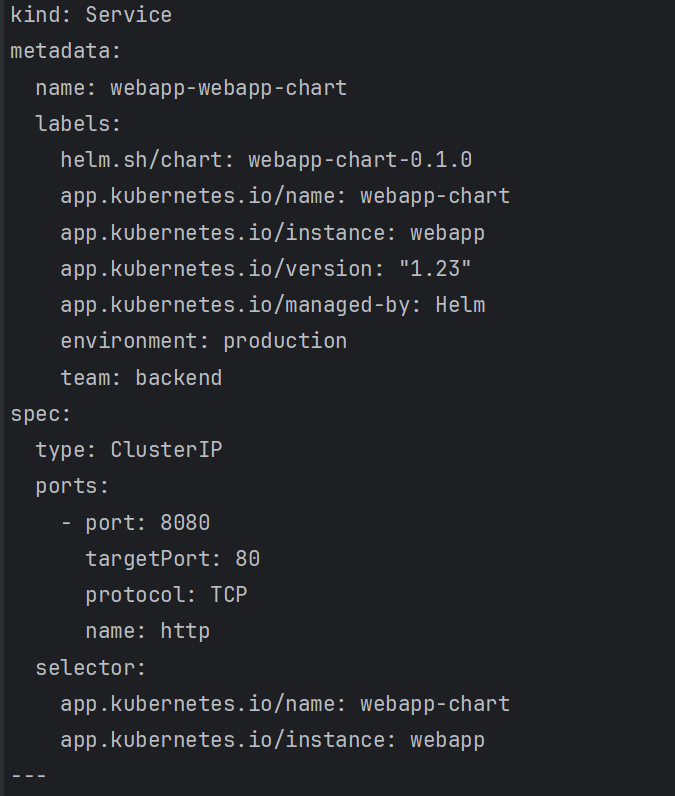


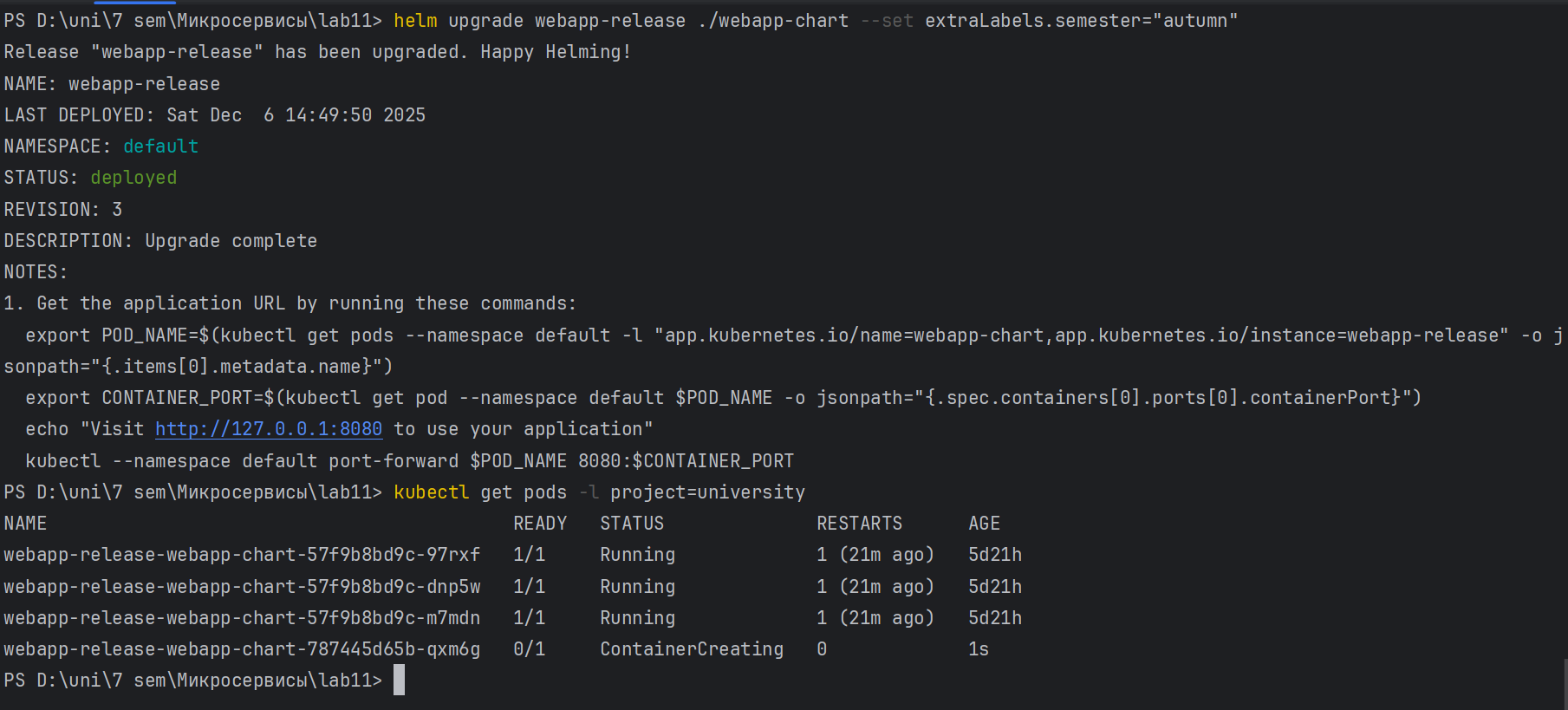


## Задание 11.2. Шаблонизация с условиями и циклами





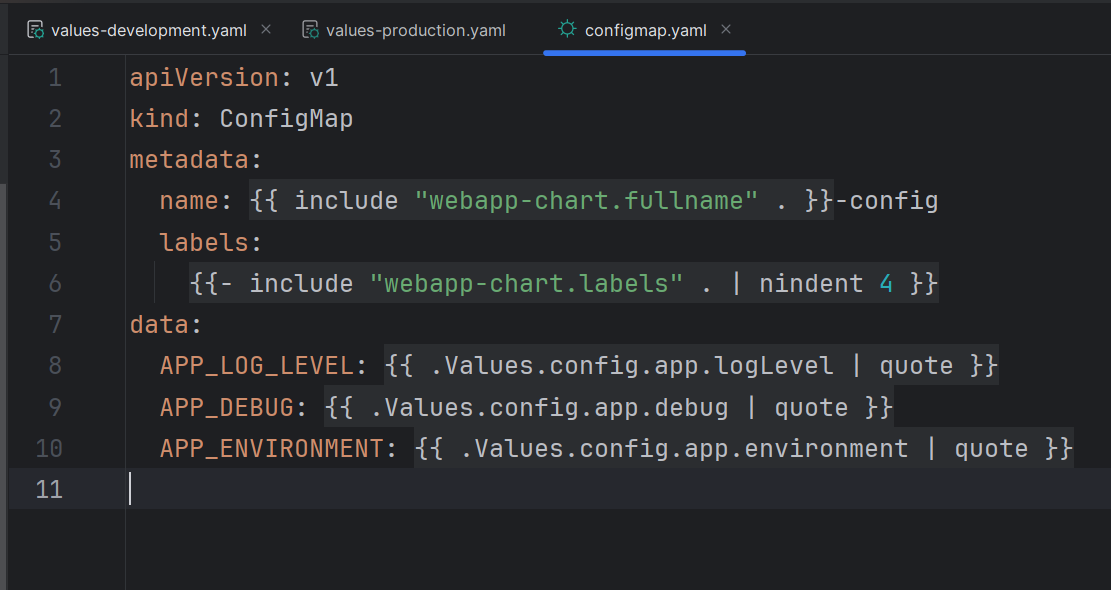


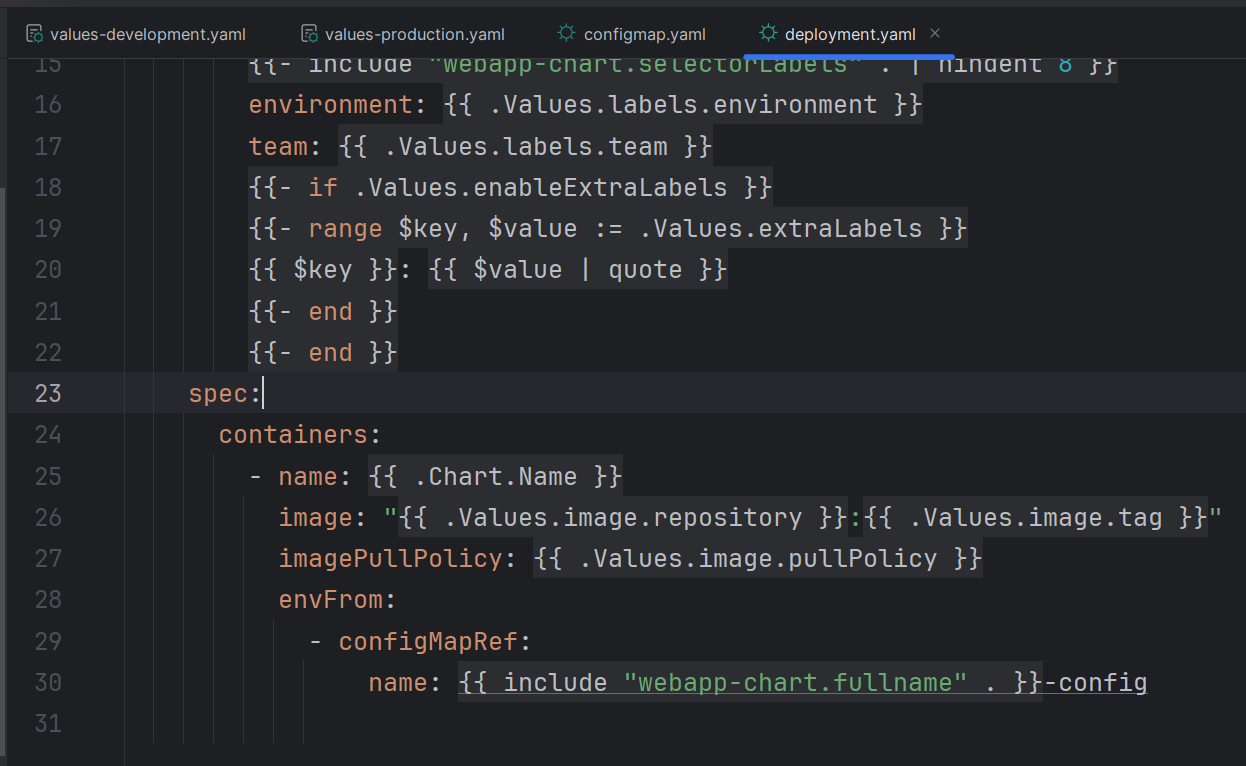


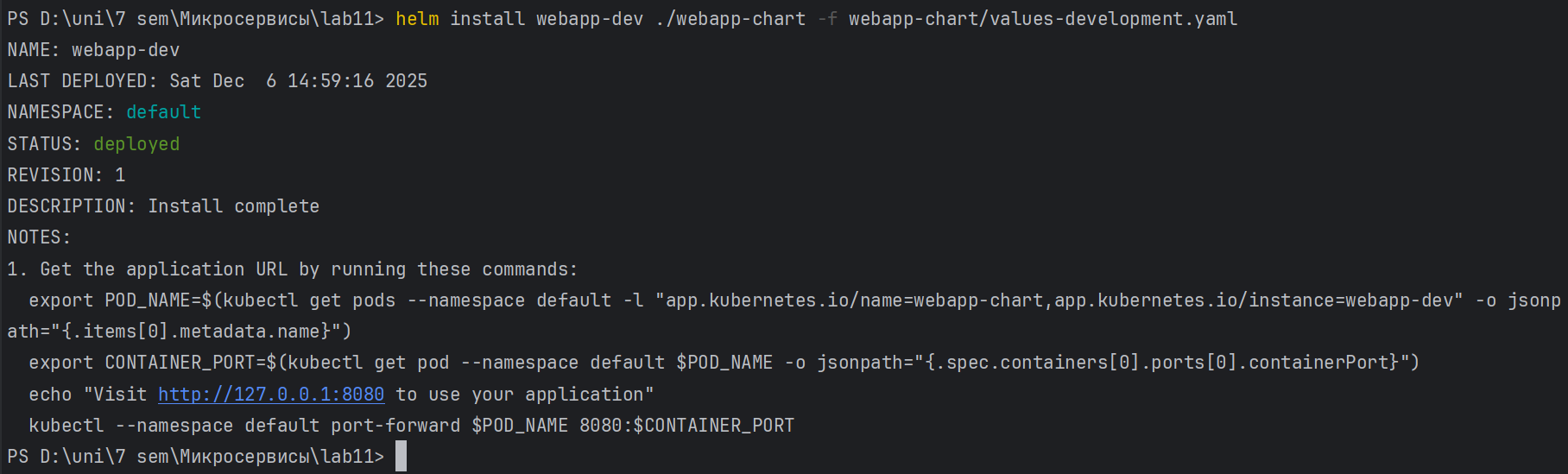
## Задание 11.3. Многофайловая конфигурация и ConfigMap

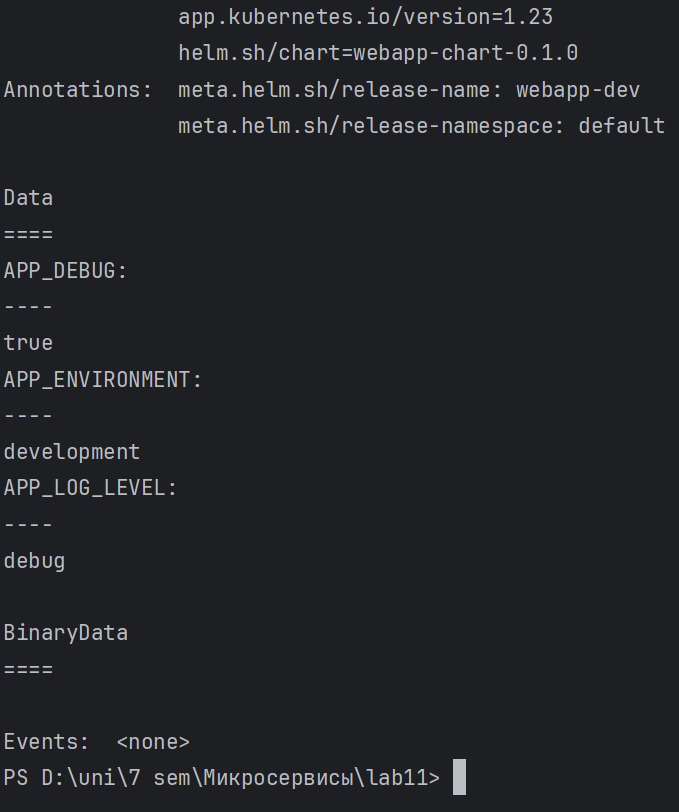
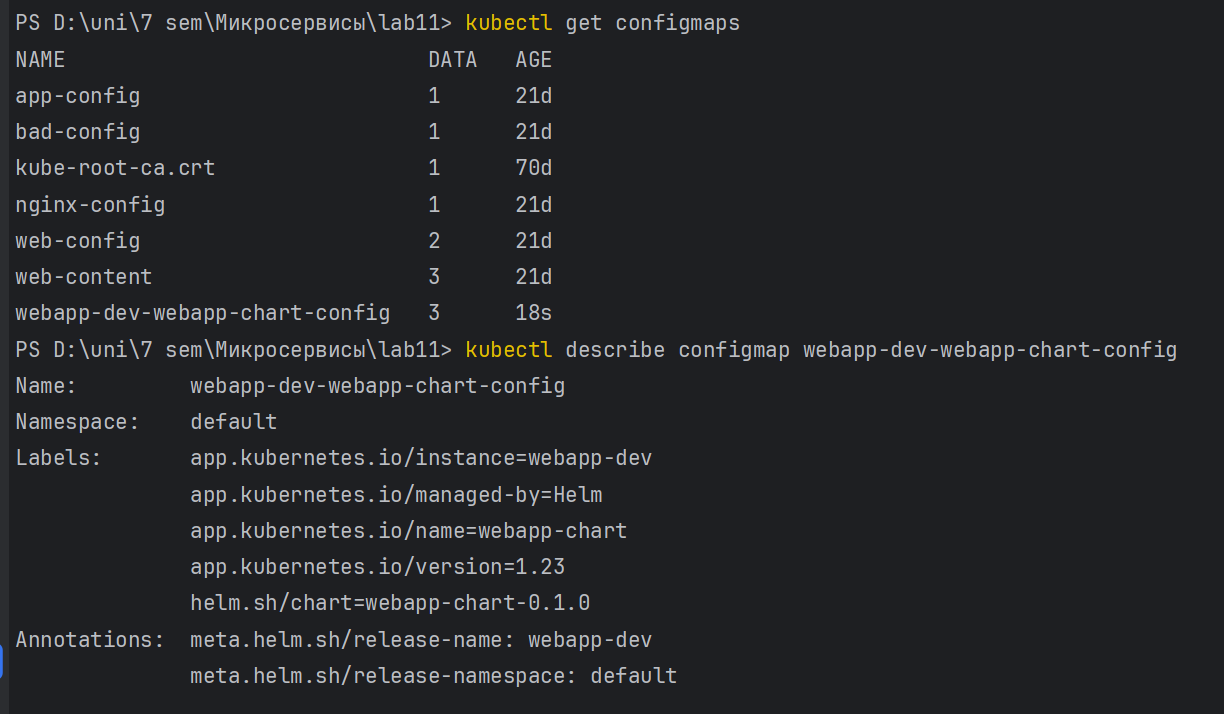
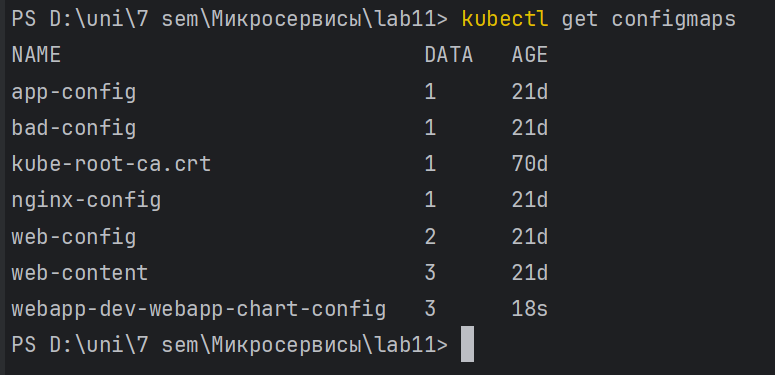


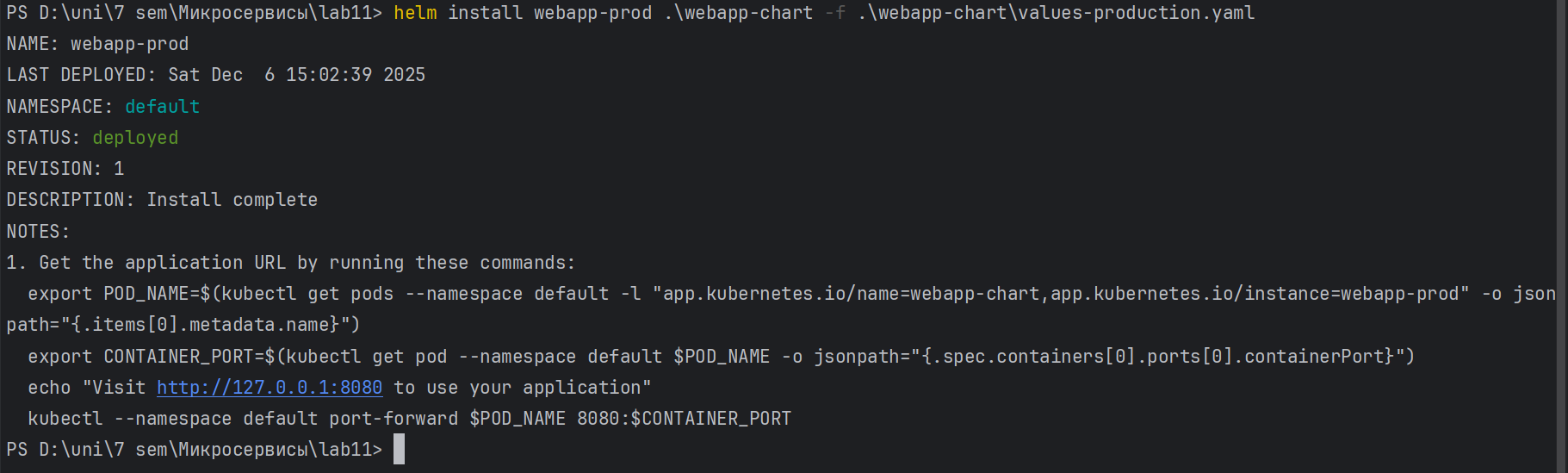


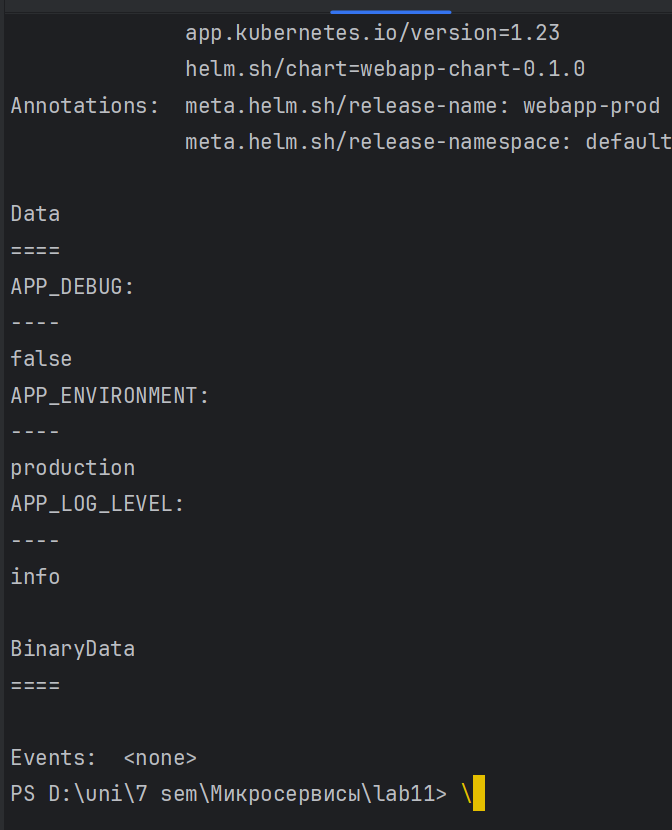
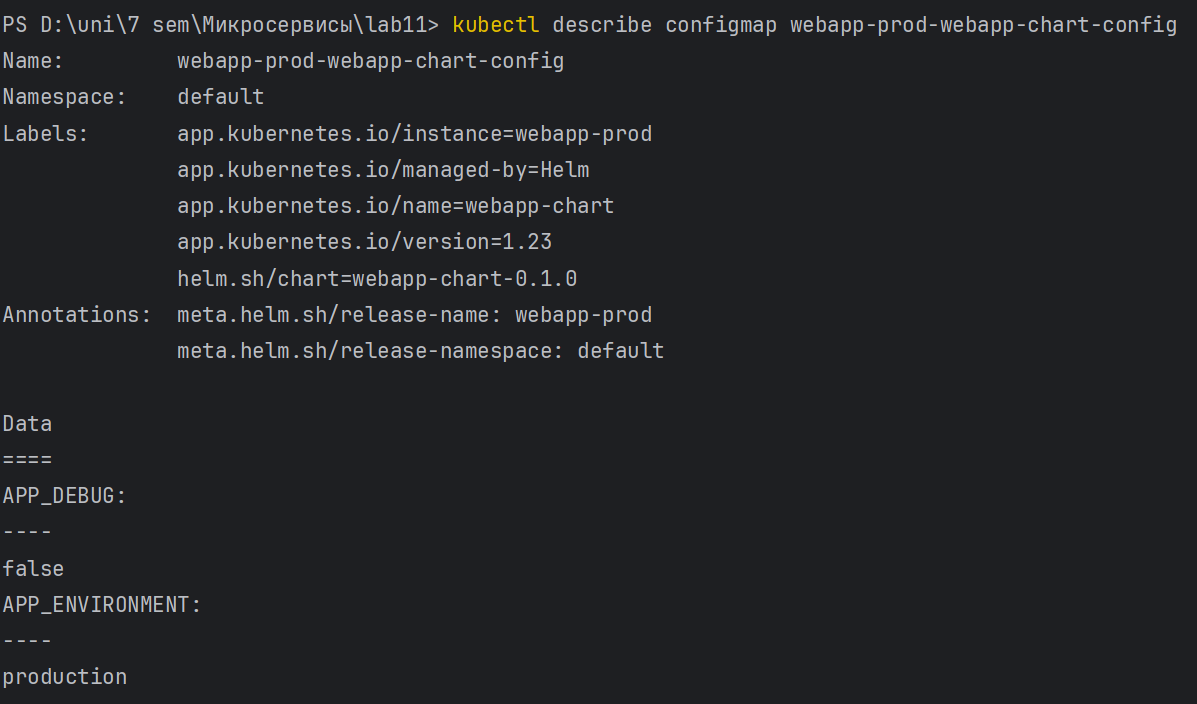
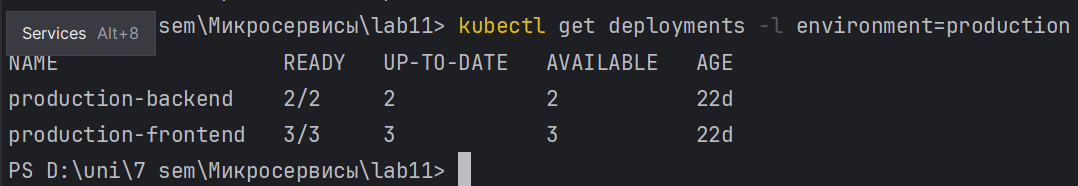




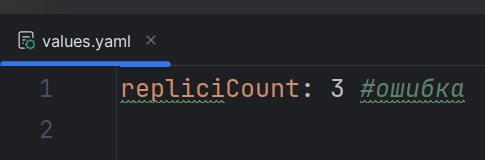


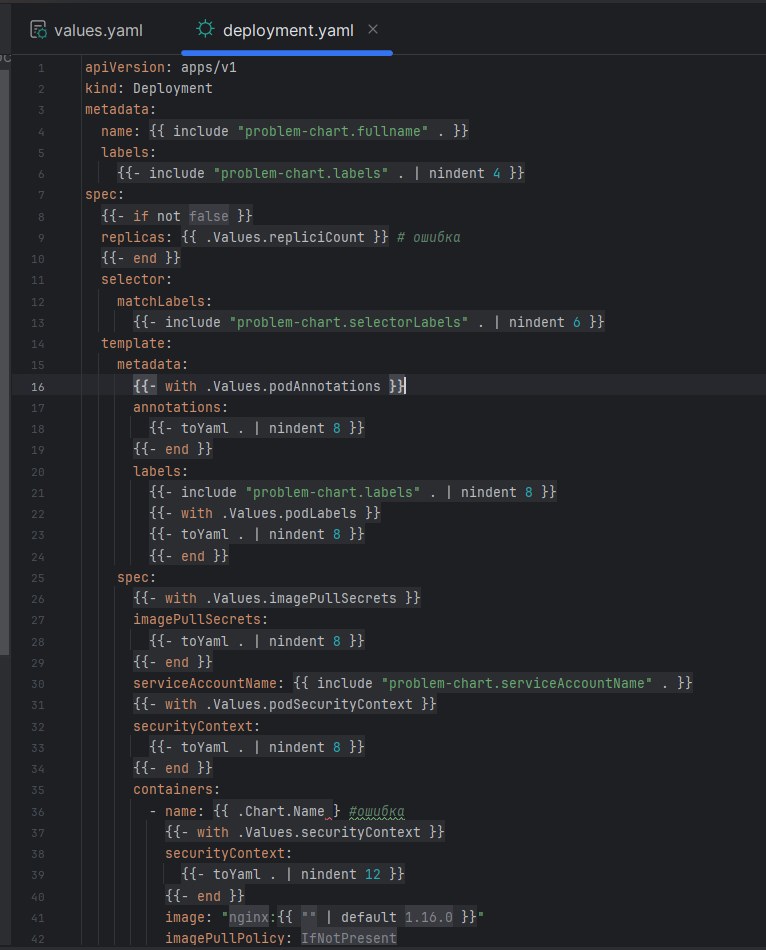


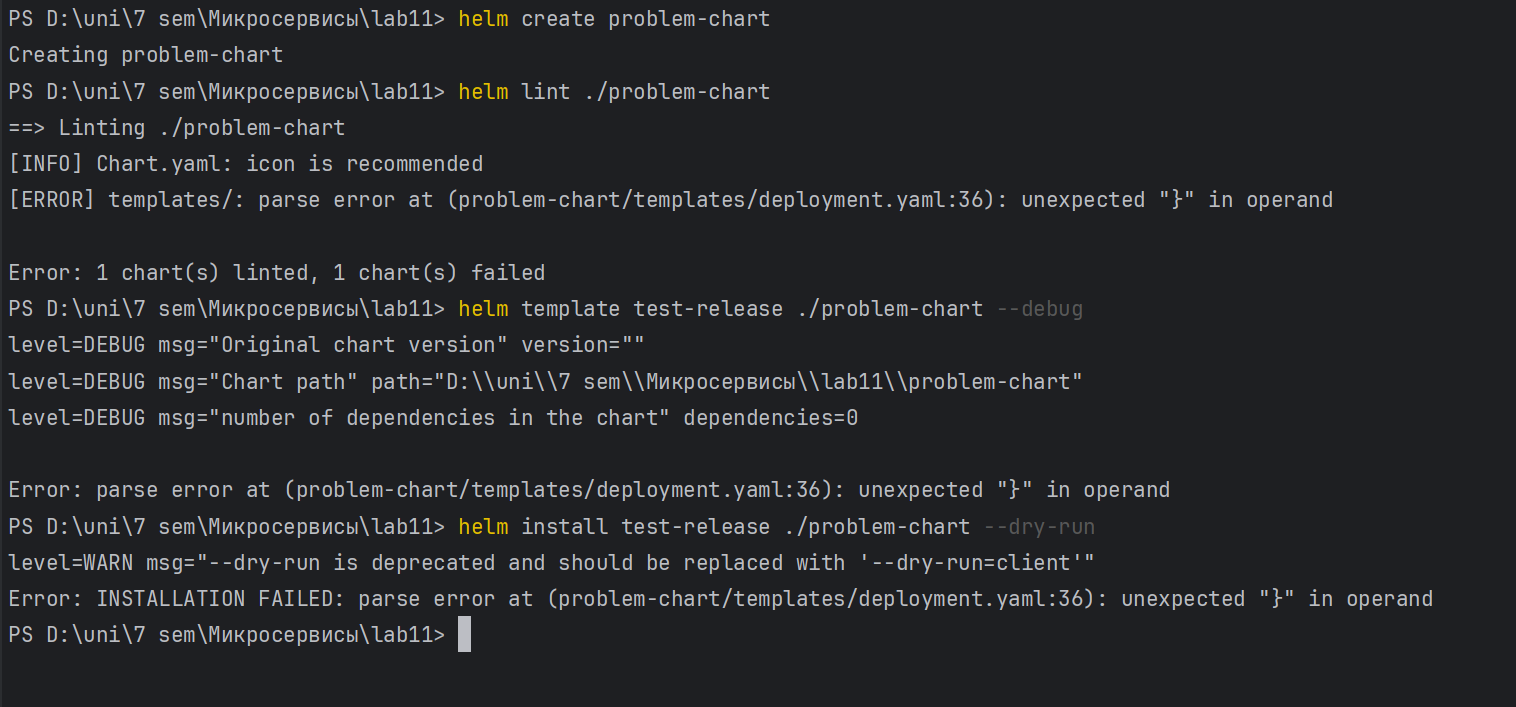




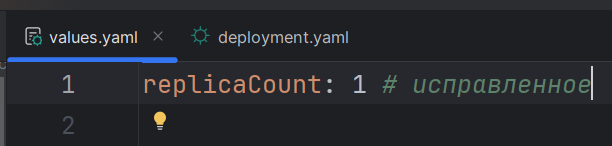
## Задание 11.4. Диагностика и отладка Helm Chart

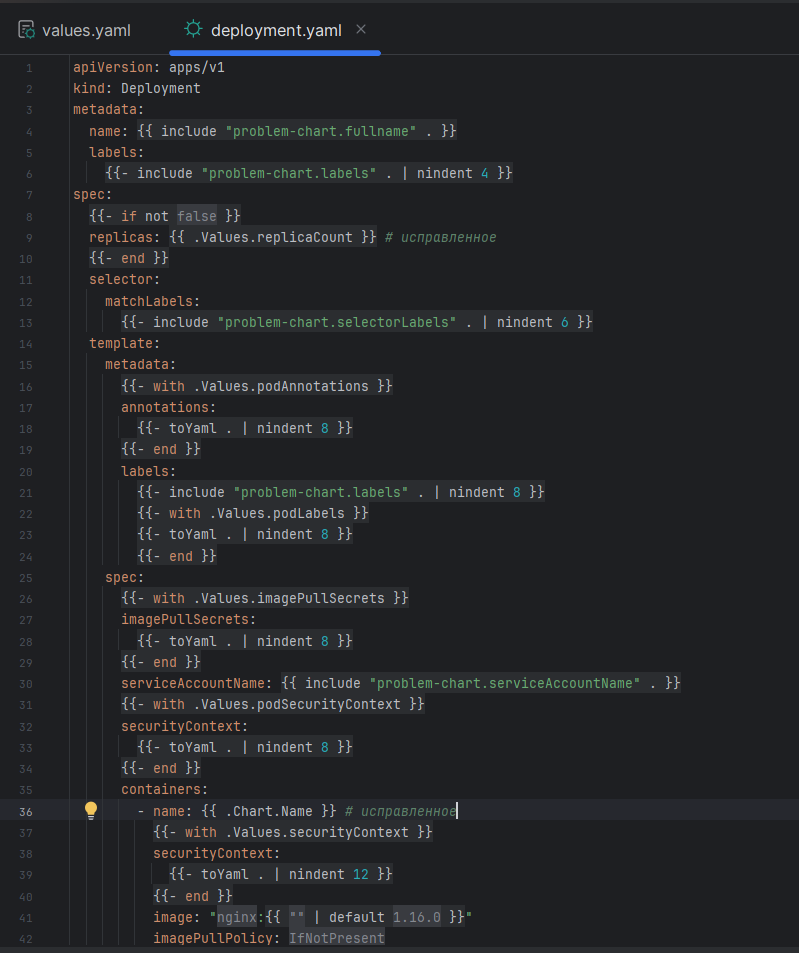


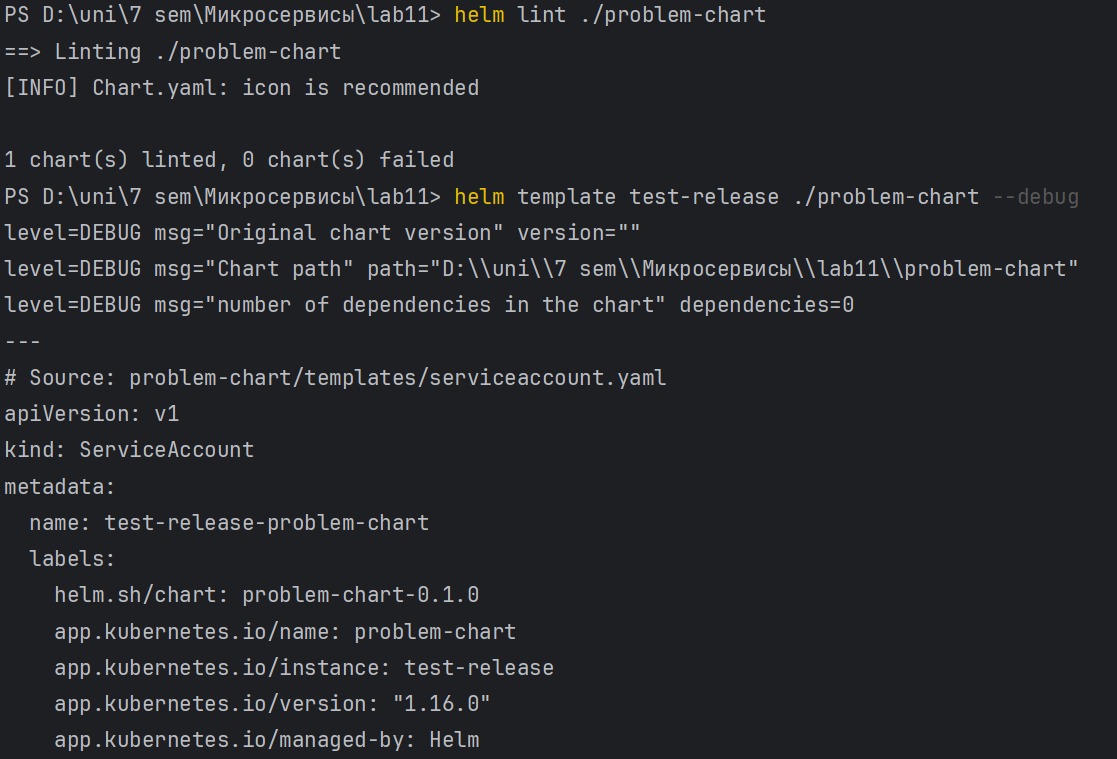




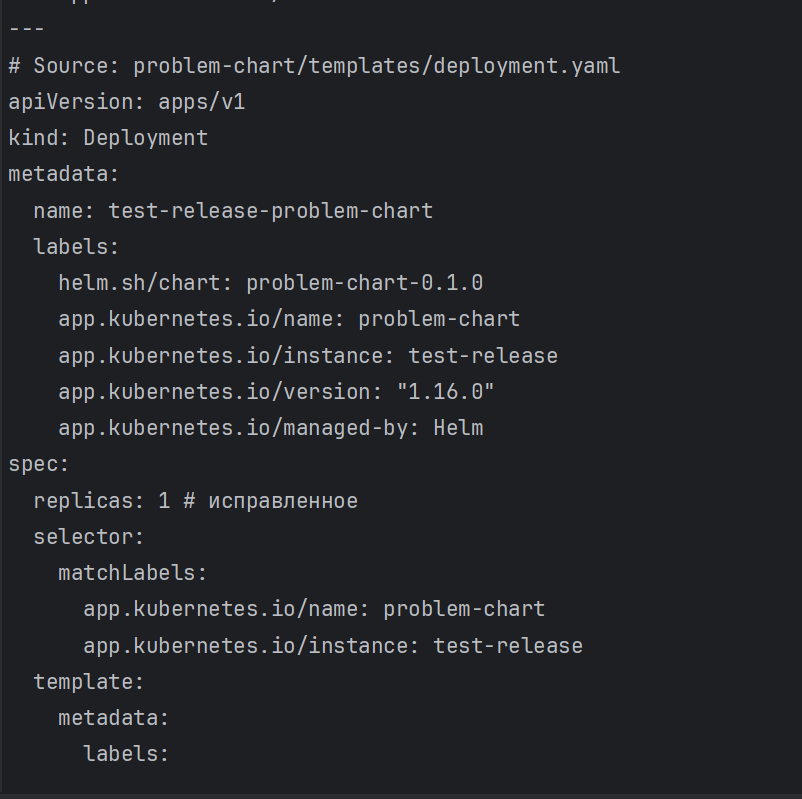
Исправленный вариант:

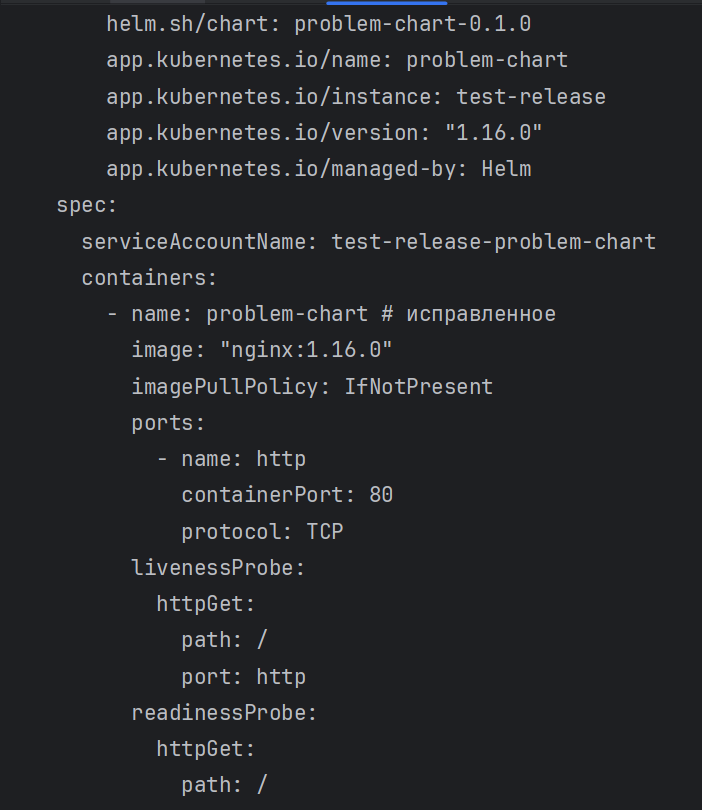
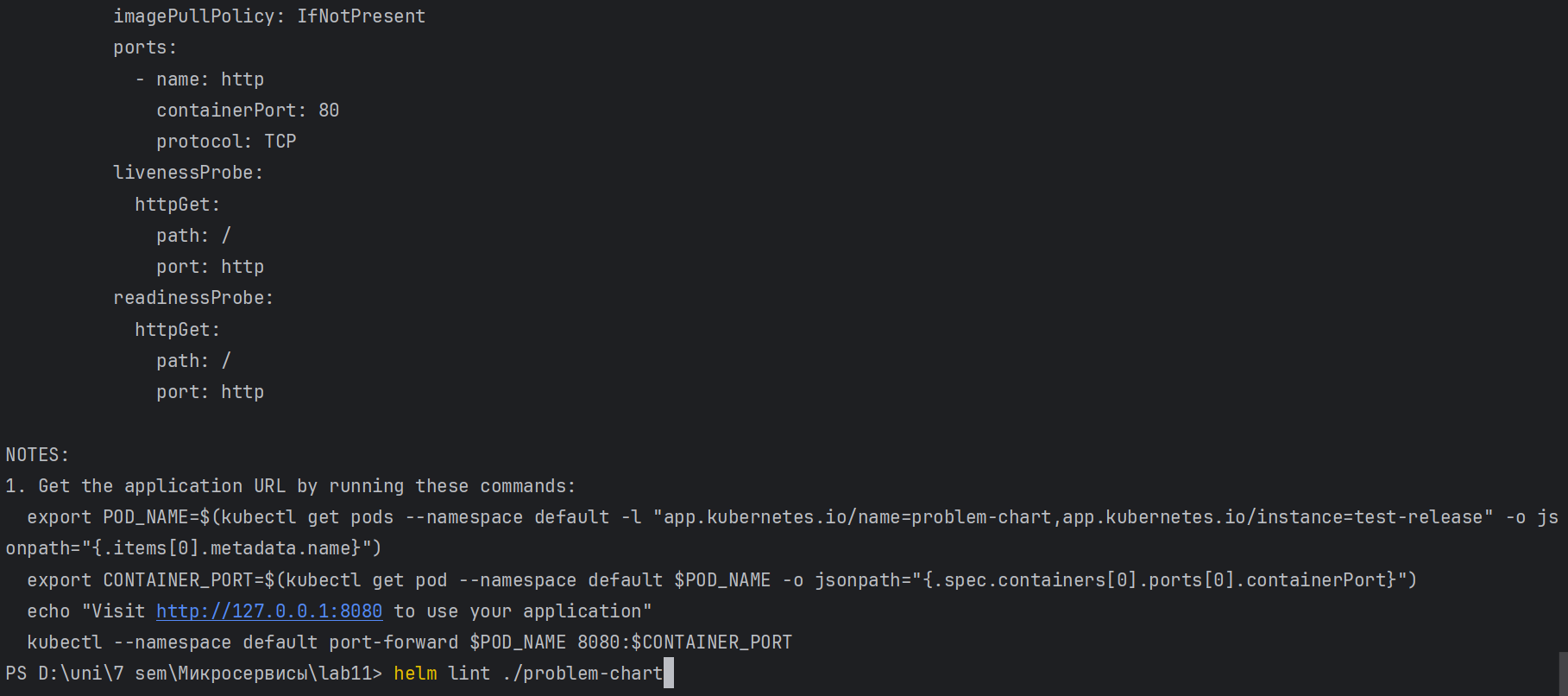
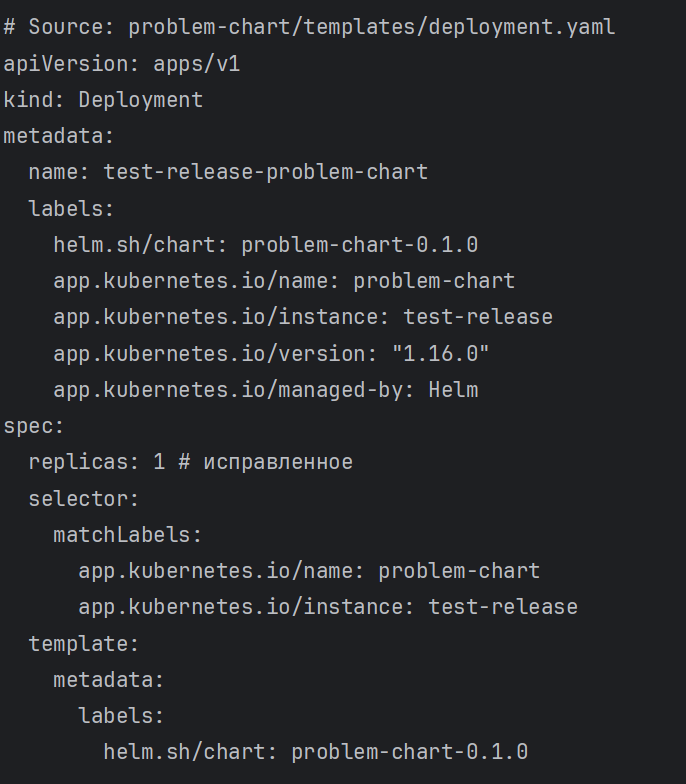
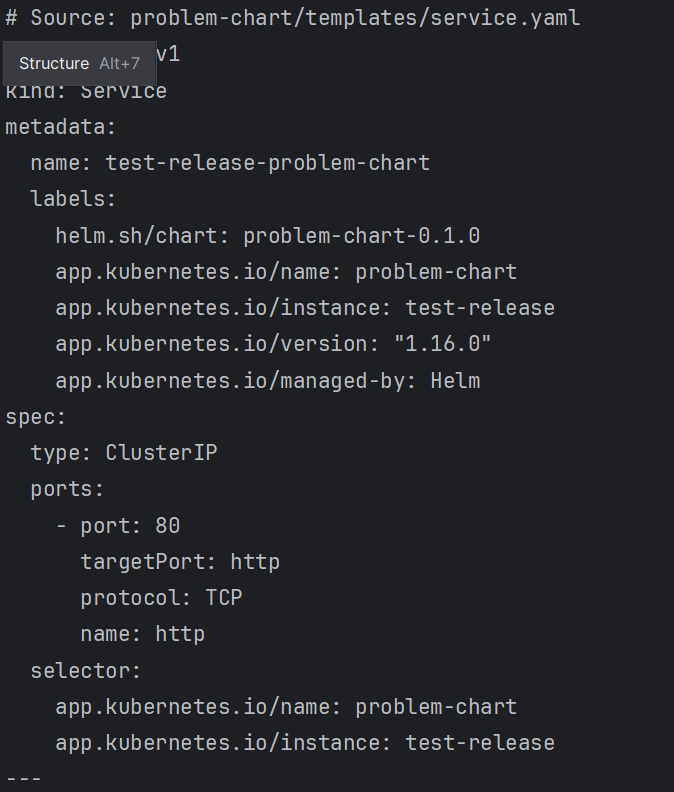
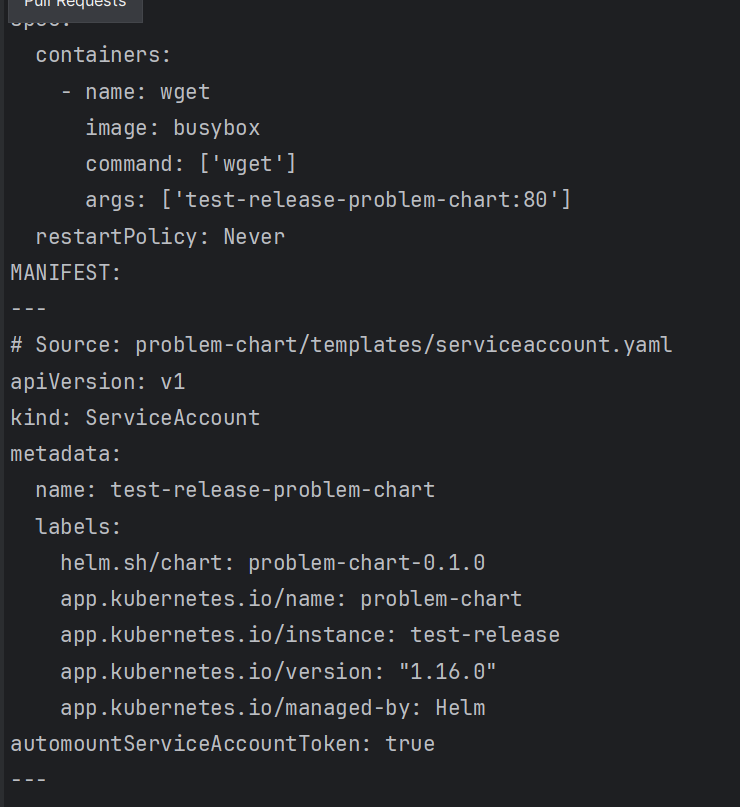
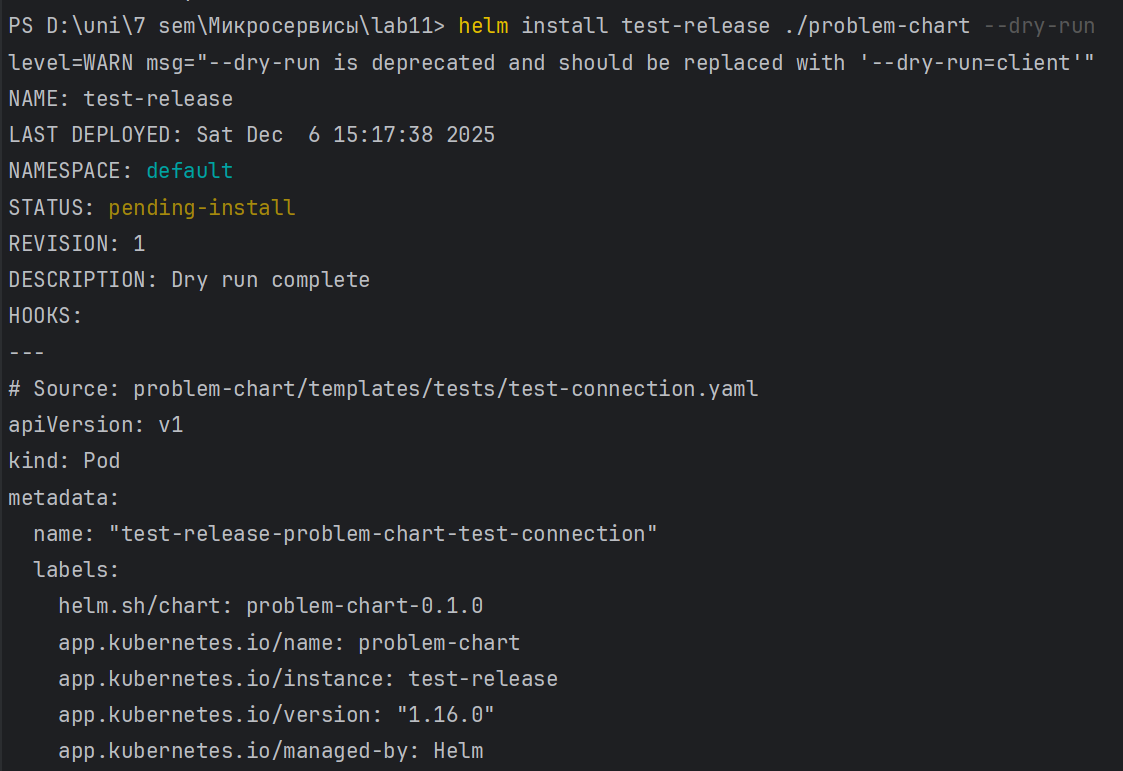
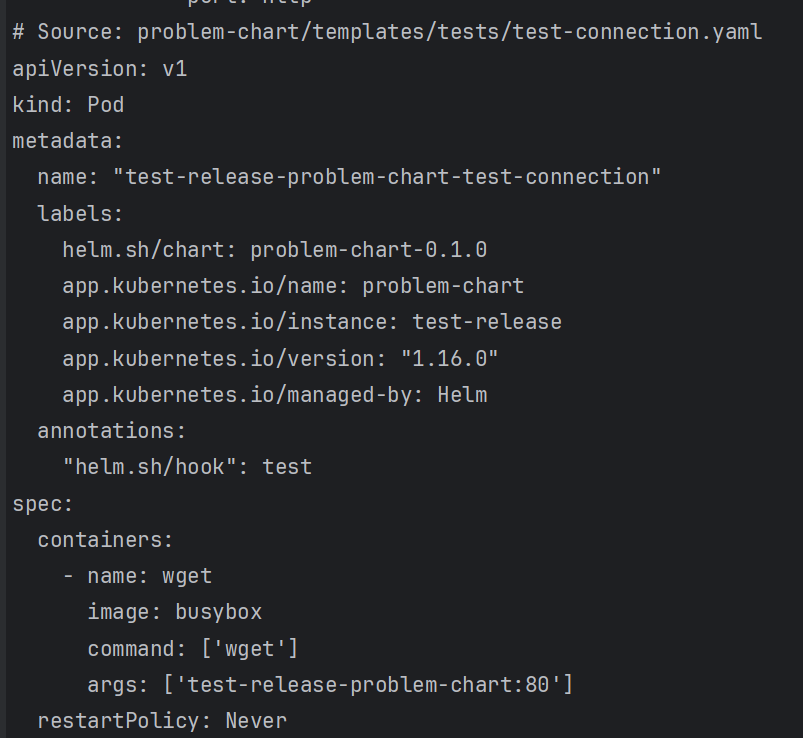


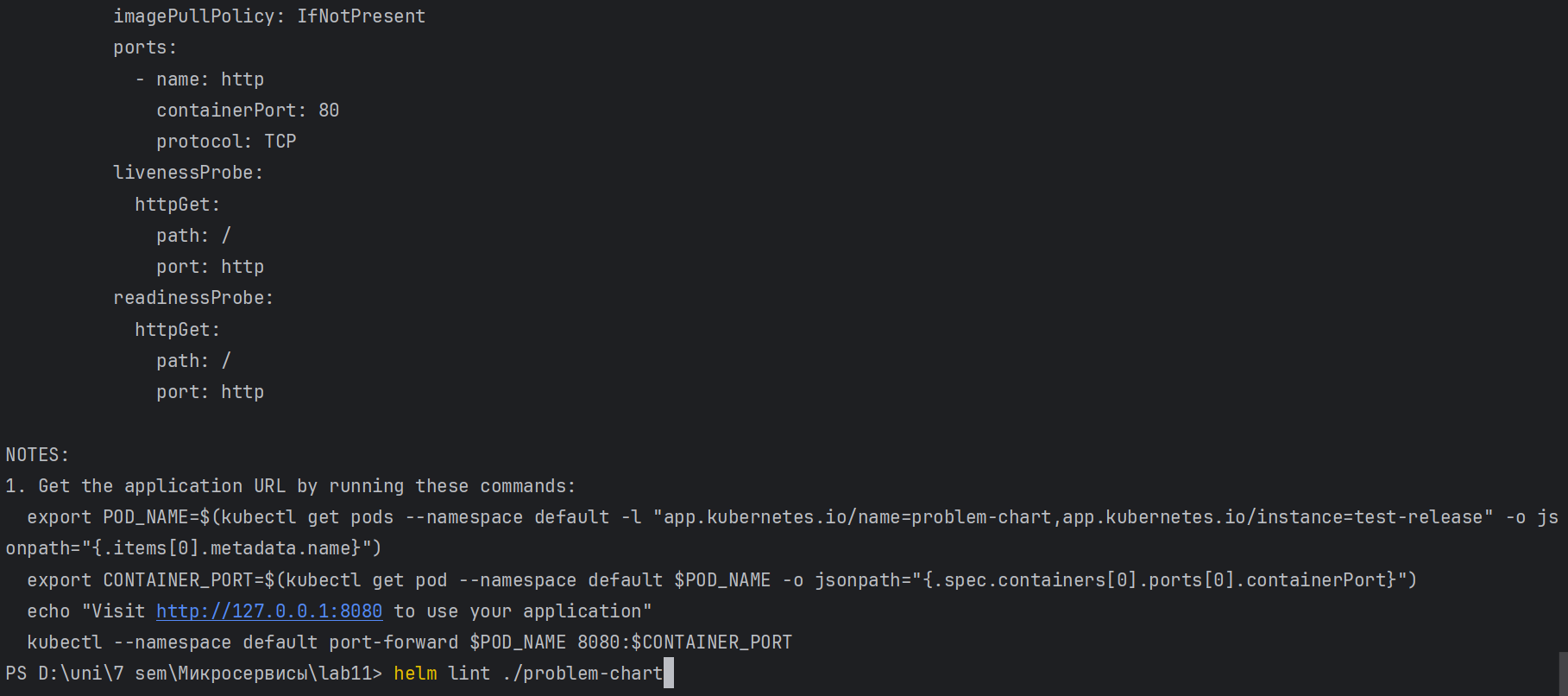












**1. В чем основное преимущество Helm перед прямым применением kubectl apply?**

Helm позволяет описывать приложение в виде пакета (Chart) и управлять его версиями как единым целым: установка, обновление, откат и удаление выполняются одной командой. Values и шаблоны позволяют переиспользовать один и тот же Chart в разных окружениях, тогда как при kubectl apply приходится вручную поддерживать множество YAML-файлов и следить за их версиями.

**2. Какие основные файлы входят в структуру Helm Chart?**

Chart.yaml — метаданные Chart

values.yaml — значения по умолчанию.

templates/ — шаблоны ресурсов Kubernetes

charts/ — зависимости (вложенные Charts).

.helmignore — список файлов, игнорируемых при упаковке Chart.

**3. Как работает шаблонизация в Helm?**

Шаблоны пишутся в формате YAML c вкраплениями Go-шаблонов ({{ ... }}). Внутри доступны объекты .Values (значения из values.yaml и параметров --set / -f), .Chart, .Release и вспомогательные функции (include, toYaml, nindent, range, if и т.д.). При установке Helm подставляет значения и генерирует обычные YAML-манифесты, которые затем отправляются в API Kubernetes.

**4. В чем разница между Chart и Release?**

**Chart** — статический пакет шаблонов и значений (как «шаблон приложения»).

**Release** — конкретный установленный экземпляр Chart в кластере с определённым набором значений и собственной историей версий.

**5. Как можно передать пользовательские значения при установке Chart?**

через файл со значениями: helm install <release> <chart> -f my-values.yaml;

через несколько файлов: -f values1.yaml -f values2.yaml;

через параметр --set key=value или --set-string key=value для единичных значений;

комбинацией этих способов (приоритет у --set поверх файлов).

**6. Что такое зависимости в Helm?**

Зависимости позволяют одному Chart использовать другие Charts (например, база данных как под-Chart). Они описываются в разделе зависимостей (для Helm 3 — в Chart.yaml), а сами под-Charts хранятся в каталоге charts/. При установке главный Chart автоматически устанавливает и конфигурирует все зависимости.

**7. Как можно отладить Helm Chart перед установкой?**

helm lint ./chart — статическая проверка синтаксиса и структуры Chart.

helm template <release> ./chart --debug — рендеринг шаблонов в локальные YAML-манифесты с подробным выводом.

helm install <release> ./chart --dry-run --debug — «сухой» запуск установки: проявляются ошибки рендеринга, но ресурсы не создаются в кластере.

**8. Какие команды используются для управления жизненным циклом Helm Release?**

helm install — установка нового Release.

helm upgrade — обновление существующего Release новой версией Chart или значений.

helm rollback — откат к предыдущей ревизии Release.

helm list — просмотр списка Releases.

helm status — информация о состоянии конкретного Release.

helm uninstall (или helm delete в старых версиях) — удаление Release и всех связанных ресурсов из кластера.