

**Платформа DevOps (ИС ТП)**

Программа и методика испытаний

Ответственный исполнитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / ФИО

|  |  |
| --- | --- |
| Дата: |  |
| Версия: |  |

Содержание

[1 Введение 4](#_Toc183096770)

[2 Объект испытаний 4](#_Toc183096771)

[3 Цель тестирования 4](#_Toc183096772)

[4 Объем тестирования 4](#_Toc183096773)

[5 Условия и порядок проведения 4](#_Toc183096774)

[5.1 Порядок проведения 4](#_Toc183096775)

[5.2 Место и продолжительность испытаний 5](#_Toc183096776)

[5.3 Отчетность 5](#_Toc183096777)

[5.4 Требования к техническим средствам 5](#_Toc183096778)

[6 Стратегия тестирования 6](#_Toc183096779)

[6.1 Функциональное тестирование 6](#_Toc183096780)

[6.1.1 Цель тестирования 6](#_Toc183096781)

[6.1.2 Проверяемые требования 6](#_Toc183096782)

[6.1.3 Методика тестирования 8](#_Toc183096793)

[6.1.4 Критерии завершения 13](#_Toc183096794)

[6.2 Тестирование безопасности 14](#_Toc183096796)

[6.2.1 Цель тестирования 14](#_Toc183096797)

[6.2.2 Проверяемые требования 14](#_Toc183096798)

[6.2.3 Методика тестирования 16](#_Toc183096800" \o "#_Toc183096800)

[6.2.4 Критерии завершения 22](#_Toc183096801)

[6.3 Мониторинг 23](#_Toc183096803)

[6.3.1 Цель тестирования 23](#_Toc183096804)

[6.3.2 Проверяемые требования 23](#_Toc183096805)

[6.3.3 Методика тестирования 24](#_Toc183096807)

[6.3.4 Критерий завершения 27](#_Toc183096808)

[6.4 Тестирование масштабируемости 28](#_Toc183096810)

[6.4.1 Цель тестирования 28](#_Toc183096811)

[6.4.2 Проверяемые требования 28](#_Toc183096812)

[6.4.3 Методика тестирования 29](#_Toc183096814)

[6.4.4 Критерий завершения 30](#_Toc183096815)

[6.5 Интеграционное тестирование 31](#_Toc183096817)

[6.6 Прочее тестирование 31](#_Toc183096818)

[6.6.1 Цель тестирования 31](#_Toc183096819)

[6.6.2 Проверяемые требования 31](#_Toc183096820)

[6.6.3 Методика тестирования 32](#_Toc183096822)

[6.6.4 Критерий завершения 32](#_Toc183096823)

# Введение

Настоящий документ представляет собой «Программу и методику испытаний», предназначенную для проведения испытаний ПО «Платформа DevOps (ИС ТП)» (далее – ИС "Платформа") для нужд Блока информационных технологий ПАО «Промсвязьбанк» (далее – Заказчик).

Испытания ИС "Платформа" проводит Заказчик с участием представителя Разработчика Платформы. Заказчик представляется группой экспертов, уполномоченных на подписание итогового «Протокола испытания».

Все компоненты ИС "Платформа" должны быть предварительно локально протестированы Разработчиком. Разработчик должен иметь документ, подтверждающий готовность ИС "Платформа" к тестированию.

Цель создания документа:

* Определить объекты испытания.
* Рекомендовать и описать стратегии тестирования, которые будут применены.
* Определить требуемые ресурсы и предоставить оценку трудозатрат на тестирование.
* Перечислить поставляемые Заказчику компоненты ИС "Платформа".
* Определить условия и порядок проведения испытаний.

# Объект испытаний

Объект испытаний: ИС «Платформа DevOps (ИС ТП)» , версия 1.0

Шифр системы: PSB.00x

Табл. 1. Комплектность ИС

|  |  |
| --- | --- |
| Подсистема/модуль | |
| ИС "Платформа" | 1.0 |
| Документация | |
| Приложение 2 к Договору №\_\_\_\_\_ от " " 2024 г | 1.0 |

# Цель тестирования

Целью тестирования ИС "Платформа" является:

* Проверка соответствие требований «Технического задания» и разработанного ПО.
* Обнаружение и устранения программных ошибок или (и), если это возможно, доказательство отсутствия последних в ПО.

# Объем тестирования

В рамках данных испытаний предполагается провести функциональное тестирование модулей ИС "Платформа" в соответствии с функциональными требованиями, описанными в документе «Техническое задание».

# Условия и порядок проведения

## Порядок проведения

* Сроки проведения испытаний за две недели согласуются Разработчиком с Заказчиком на основании предложений Разработчика.
* Испытания проводятся комиссией, формируемой Разработчиком совместно с Заказчиком.
* Испытания осуществляются на стенде Заказчика, согласно требованиям к техническому и программному обеспечению «Технического задания». Рекомендуется, чтобы тестовая система имитировала производственную среду, масштабированную в сторону уменьшения в тех случаях, когда это допустимо.
* Испытания производятся согласно настоящей «Программа и методика испытаний» в сроки согласованные с Заказчиком.
* Испытания заканчиваются оформлением документа «Протокол испытаний», который подписывается представителем Разработчиков, представителем Заказчика.
* Возникшие в ходе проведения испытаний замечания, устраняются в установленном порядке.
* В случае неудовлетворительных результатов прохождения сценария Программы и методики испытаний (ПМИ) или отсутствия необходимых сценариев, будет разработана следующая версия ПМИ. Она будет учитывать указанные замечания и сценарии. При этом отсутствие отдельных сценариев или их неудовлетворительное выполнение не являются основанием для прекращения опытной эксплуатации

## Место и продолжительность испытаний

* Испытания проводятся на стенде с участием представителей Заказчика и Разработчика, уполномоченных на подписание «Протокол испытаний».
* Плановая дата начала испытаний: \_\_\_\_.\_\_\_\_\_.2024 г. в рабочее время (с 9 до 18). Плановая продолжительность испытаний – \_\_\_\_\_\_дней. Продолжительность испытаний может быть увеличена в случае необходимости проведения дополнительных тестов и проверок.

## Отчетность

* В процессе испытаний ведется «Протокол испытаний».
* По завершении испытаний составляется Акт, в котором излагаются общие результаты испытаний. К Акту прилагается «Протокол испытаний», который ведётся в процессе проведения испытаний.
* Акт составляется в двух экземплярах и подписывается представителями Заказчика и Разработчика.

## Требования к техническим средствам

Испытания должны проводиться на стенде, с минимальными техническими характеристиками, приведенными в таблице ниже.

|  |  |
| --- | --- |
| Сервер для узлов Deckhouse | |
| Процессор | 4 vCPU на каждый узел кластера |
| Жесткий диск | 100 GB SSD для каждого узла |
| Оперативная память | 16 GB RAM на каждый узел кластера |
| Сетевой адаптер | 1 Gbps Ethernet на каждый узел |
| Операционная система | РЕД ОС 7.3.4 или совместимая операционная система на всех узлах. |
| Bootstrap Сервер | |
| Процессор | 2 vCPU |
| Оперативная память | 8 GB RAM |
| Жесткий диск | 50 GB SSD |
| Монитор | Разрешение не менее 1920x1080 (Full HD) |
| Сетевой адаптер | 1 Gbps Ethernet или Wi-Fi |
| Операционная система | РЕД ОС 7.3.4 или совместимая операционная система |
| Системы, не входящие в состав ИС "Платформа" | |
| Платформа аппаратной виртуализации | |
| LDAP-каталог | |
| Почтовый сервер | |
| Система управления секретами | |
| AHRepo | |
| Система сбора журналов | |
| Система класса SIEM | |
| Средства защиты информации | |

# Стратегия тестирования

Для требований, предъявляемых к ИС "Платформа" установлены следующие категории:

* Решающее – требование, при невыполнении которого результат тестирования оценивается как неуспешный.
* Нерешающее – требование, несоответствие которому не является препятствием для оценки результата тестирования как успешного.

## Функциональное тестирование

### Цель тестирования

Убедиться в корректном функционировании объекта испытаний.

### Проверяемые требования

Таблица 1. Требования к функциональному тестированию

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Код требования в ТЗ** | **Краткое описание** | **Тестовые сценарии в ПМИ** |
| 1 | П. 3.1.2 | Возможность автоматической установки кластеров с заданной конфигурацией | Ф-7 |
| 2 | П. 3.1.5 | Поддержка конфигурации высокой доступности и восстановления после аварий (Disaster Recovery) | Ф-13 |
| 3 | П. 3.1.7 | Возможность управления распределением подов между узлами кластера | Ф-9 |
| 4 | П. 3.1.8 | Механизм назначения меток на узлы кластера | Ф-9 |
| 5 | П. 3.3.11 | Использование актуальной версии Kubernetes с задержкой не более 12 месяцев | Ф-5 |
| 6 | П. 3.3.12 | Возможность использования предыдущей версии Kubernetes | Ф-5 |
| 7 | П. 3.3.16 | Управление функциями Платформы через включение и отключение компонентов по функциональному признаку | Ф-11, Ф-12, Ф-10, Ф-15 |
| 8 | П. 3.3.8 | Возможность автоматического добавления аннотаций и меток к пространствам имен | Ф-14 |
| 9 | П. 3.5.1 | Возможность самостоятельной загрузки пакетов обновления для последующей установки | Ф-3 |
| 10 | П. 3.5.2 | Обновление всех компонентов через единый инструмент управления | Ф-2 |
| 11 | П. 3.5.4 | Поддержка хранения версий используемых компонентов | Ф-2, Ф-3 |
| 12 | П. 4.1.10 | Совместимость Платформы с ОС Astra Linux Special Edition версии 1.7.1 и выше. | Ф-4 |
| 13 | П. 6.1.6 | Поддержка российских операционных систем из ЕРРП на пользовательских устройствах | Ф-4 |
| 14 | П. 6.2.2 | Автономное восстановление после сбоев для высокой доступности системы | Ф-13 |
| 15 | П. 6.3.1 | Поддержка автоматического горизонтального масштабирования без остановки работы | Ф-6 |
| 16 | П. 6.5.2 | Возможность установки системы на оборудовании заказчика (on-premises) | Ф-1 |
| 17 | П. 3.5.5 | Уведомление Платформы о наличии обновления с возможностью отложить обновление на удобное время. | Ф-16 |
| 18 | П. 6.3.3 | Возможность использования предварительно подготовленных узлов в горячем резерве для моментального масштабирования | Ф-6 |
| 19 | П. 3.3.2 | Администрирование Платформы с использованием веб-интерфейса, интерфейса командной строки (CLI) и API. | Ф-17 |
| 21 | П. 3.3.5 | Лимиты ресурсов для пространств имен (квоты). | Ф-14 |
| 22 | П. 6.5.6 | Возможность установки в изолированном контуре без доступа к Интернету | Ф-1 |

### Методика тестирования

Функциональное тестирование предполагает выполнение тестовых процедур, приведенных в таблице ниже.

Таблица 2. Порядок проведения функционального тестирования

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Код требования в ТЗ** | **Название Тестового сценария** | **Приоритет проверки** | **Шаги** | **Ожидаемый результат** | **Время прохождения** |
| 1 | П.3.3.16 | Ф-11, Установка / добавление элементов интерфейса / модулей (из поставки платформы) | Высокий | Установить платформу Deckhouse с предустановленным набором модулей default. Добавить или удалить необходимые модули, используя конфигурацию управления модулями Deckhouse. Проверить корректность работы новых модулей и убедиться, что состав модулей изменился в соответствии с настройками. | Платформа установлена с набором модулей default, после чего состав модулей был успешно изменен, а новые модули работают корректно. |  |
|  |
|  |
|  |
| 2 | П.6.3.1 | Ф-6, Управление узлами кластера (добавление, удаление) | Высокий | Добавить новый узел в Kubernetes кластер, настроить его и убедиться, что он перешел в статус Ready. Освободить узел от рабочих нагрузок, удалить его из кластера и очистить конфигурацию. Проверить, что узел удален и больше не отображается в списке узлов кластера. | Узел был успешно добавлен в кластер и перешел в статус Ready, затем был освобожден от рабочих нагрузок, удален из кластера и перестал отображаться в списке узлов. |  |
| П. 6.3.3 |
|  |
| 3 | П.3.3.8  П.3.3.5 | Ф-14, Управление namespaces (добавление, удаление, редактирование) | Высокий | Создать новый namespace с помощью команды kubectl create namespace. Добавить или удалить labels на созданном namespace, используя команды kubectl label namespace. Убедиться, что изменения применились, и namespace отображает корректные labels. | Namespace успешно создан, labels были добавлены или удалены, и изменения отобразились корректно. |  |
| 4 | П.3.1.7 | Ф-9, Размещение компонентов Deckhouse на выделенных узлах | Высокий | Добавить выделенный узел в кластер и настроить его для размещения компонентов мониторинга. Применить конфигурацию NodeSelector или Taints/Tolerations для ограничения размещения Prometheus только на выделенном узле. Проверить, что Prometheus успешно запустился на выделенном узле и функционирует корректно. | Выделенный узел был настроен для размещения компонентов мониторинга, Prometheus запустился на этом узле и работает корректно. |  |
| П.3.1.8 |
| 5 | П.4.1.10 | Ф-4, Поддержка РФ операционных систем (РЕДОС, ALT linux, Astra Linux) | Высокий | Установить узел Kubernetes под управлением одной из российских операционных систем (РЕДОС, ALT Linux, Astra Linux) и подготовить его для работы с Deckhouse. Выполнить установку Deckhouse на подготовленный узел, следуя документации по установке в выбранной ОС. Проверить, что все поды запускаются успешно и находятся в статусе Running. | Deckhouse успешно установлена на узел с российской ОС, все поды работают корректно, и статус Running подтвержден для всех компонентов. |  |
| П.6.1.6 |
| 6 | П.3.1.5  П.6.2.2. | Ф-13, Отказоустойчивая конфигурация всех компонентов платформы | Низкий | Установить multi-master кластер Kubernetes и развернуть платформу Deckhouse. Проверить, что служебные компоненты (Prometheus, Grafana, Dex) имеют как минимум две реплики. Убедиться, что реплики распределены по различным узлам кластера для обеспечения отказоустойчивости. | Служебные компоненты Prometheus, Grafana и Dex развернуты в двух репликах и равномерно распределены между узлами кластера. |  |
| 7 | П.3.3.11 | Ф-5, Обновление версии Kubernetes | Высокий | Установить кластер Kubernetes с настройкой версии “Automatic”. Изменить версию Kubernetes на желаемую через конфигурацию кластера и применить изменения. Проверить, что на всех узлах кластера kubelet обновился до необходимой версии. | Версия Kubernetes была изменена на желаемую, и kubelet на всех узлах кластера обновился до необходимой версии. |  |
| П.3.3.12 |
| 8 | П.3.3.16 | Ф-10, Запуск модулей Deckhouse Enterprise версии | Высокий | Включить модуль user-authn с использованием команды управления модулями Deckhouse. Проверить, что в кластере появился namespace d8-user-authn. Убедиться, что в этом namespace запущен под с именем dex. | Модуль user-authn был включен, namespace d8-user-authn создан, а под dex успешно запущен. |  |
|  |
|  |
| 9 | П.6.5.2 | Ф-1, Возможность установки платформы в закрытом контуре | Высокий | Настроить локальные репозитории для Docker-образов и Helm-чартов, загрузив в них все необходимые компоненты. Выполнить установку с использованием локальных ресурсов и проверить корректную работу платформы. Подтвердить успешность установки, убедившись, что все компоненты функционируют автономно без доступа к интернету. | Установка прошла успешно, все компоненты Deckhouse работают в закрытом контуре, все зависимости загружены из локальных репозиториев, и платформа готова к использованию. |  |
| П.6.5.6 |
|  |
|  |
|  |
| 10 | П.3.5.1 | Ф-3, Возможность ручного обновления платформы Deckhouse | Высокий | Установить Deckhouse платформу, указав версию на единицу ниже текущего минорного релиза выбранного канала обновления, и настроить ручной режим обновления. Запустить обновление вручную, используя команду kubectl apply для нового релиза из репозитория канала обновлений на сайте releases.deckhouse.ru. Проверить логи обновления и убедиться, что статус ресурса deckhouserelease изменился на Deployed для новой версии. | Платформа Deckhouse успешно обновляется вручную до актуальной версии выбранного канала, а статус deckhouserelease отображается как Deployed. |  |
| П.3.5.4 |
| 11 | П.3.3.16 | Ф-12, Возможность отключения неиспользуемых модулей платформы | Средний | Отключить модуль upmeter, используя команду управления модулями Deckhouse. Проверить, что модуль успешно отключен и все связанные ресурсы удалены. Убедиться, что namespace d8-upmeter больше не существует в кластере. | Модуль upmeter был отключен, и namespace d8-upmeter удален из кластера. |  |
| 12 | П.3.3.16 | Ф-15, Возможность использования внешних модулей | Низкий | Установить внешний оператор в Deckhouse, используя манифесты или Helm-чарт. Проверить создание необходимых ресурсов оператора, включая namespace и deployment. Убедиться, что поды оператора успешно запущены и находятся в статусе Running. | Внешний оператор установлен, его поды успешно запущены и находятся в статусе Running. |  |
|  |
|  |
| 13 | П. 5.2.5 | Ф-8, Возможность дополнительной конфигурации runtime-компонентов узлов кластера | Высокий | Применить настройку ресурса NodeGroup с необходимыми параметрами для конфигурации runtime-компонентов. Проверить работу systemd-сервиса bashible на узле кластера и его логи для подтверждения успешного выполнения изменений. Убедиться, что параметр на узле кластера изменился на заданное значение. | Настройка ресурса NodeGroup была применена, systemd-сервис bashible успешно отработал, и параметр на узле кластера изменился на необходимое значение. |  |
| 14 | П.3.5.4 | Ф-2, Автоматическое обновление платформы Deckhouse | Высокий | Установить Deckhouse платформу, указав версию на единицу ниже текущего минорного релиза выбранного канала обновления, и настройте автоматический механизм обновления. Убедиься, что платформа настроена для получения обновлений с использованием репозиториев канала на сайте releases.deckhouse.ru. Проверить логи и статус ресурса deckhouserelease, чтобы подтвердить успешное обновление и статус Deployed для нового релиза. | Платформа Deckhouse автоматически обновляется до актуальной версии выбранного канала обновления, а статус deckhouserelease отображается как Deployed. |  |
| П.3.5.2 |
|  |
| 15 | П.3.1.2 | Ф-7, Автоматическая настройка узлов кластера | Высокий | Применить ресурс NodeGroupConfiguration с заданными параметрами для настройки узлов кластера. Проверить работу systemd-сервиса bashible на узле и убедиться, что конфигурация была применена. Проверить, что параметр на узле кластера изменился на указанное значение. | Ресурс NodeGroupConfiguration был применен, systemd-сервис bashible отработал корректно, а параметр на узле изменился на необходимое значение. |  |
|  |
| 16 | П. 3.5.5 | Ф-16,  Уведомление о наличии новой версии | Высокий | На веб-странице по адресу console.<домен-кластера> отображается информации о наличии новой версии | На веб-странице по адресу console.<домен-кластера> отображается информации о наличии новой версии |  |
| 17 | П. 3.3.2 | Ф-17, Администрирование Платформы с использованием веб-интерфейса, интерфейса командной строки (CLI) и API. | Высокий | Открыть веб-страницу по адресу console.<домен-кластера>, выполнить вход в систему, назначить ноде кластера новую роль. Открыть терминал, выполнить команду kubectl edit node <имя-ноды>, назначить ноде кластера новую роль, проверить применение конфигурации. | В веб-интерфейсе и через терминал настройки ноды применяются одинаково. |  |

Таблица 3. Оценка трудозатрат на тестирование

|  |  |
| --- | --- |
| **Количество проверок** | **Общее время прохождения** |
| 17 |  |

### Критерии завершения

Все запланированные тесты были выполнены. Все выявленные ошибки зафиксированы.

## Тестирование безопасности

### Цель тестирования

Безопасность уровня приложения: проверить, что пользователю каждого типа доступны только те функции / данные, на которые предоставлены права данному типу пользователей.

Безопасность уровня системы: проверить, что только те пользователи, которым предоставлен доступ к системе и приложениям, могут получить к ним доступ.

Средства обеспечения информационной безопасности в средах контейнерной оркестрации: проверить обеспечение соблюдения стандартов безопасности сред контейнерной оркестрации, проверить работоспособность и возможность настройки механизмов обеспечения информационной безопасности.

### Проверяемые требования

Таблица 4. Требования тестирования безопасности

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Код требования в ТЗ** | **Краткое описание** | **Тестовые сценарии в ПМИ** |
| 1 | П. 5.2.11 | Журналирование операций Kubernetes и работы с etcd, выгрузка на syslog-сервер. | Б-1 |
| 2 | П. 6.12.1 | Раздел аудита для протоколирования действий пользователя. | Б-1 |
| 3 | П. 6.12.2 | Модули регистрации событий: логирование, SQL-доступ, аудит. | Б-1 |
| 4 | П. 4.2.1 | Поддержка CNI плагинов с Network Policy. | Б-2, Б-3 |
| 5 | П. 6.10.3 | Поддержка внутрикластерных сетевых политик. | Б-2, Б-3, Б-4 |
| 6 | П. 3.2.1 | CNI с фильтрацией пакетов на уровнях L4, L7 OSI. | Б-2, Б-3 |
| 7 | П. 4.2.5 | Интеграция с LDAPS, OAuth /SAML, двухфакторная аутентификация. | Б-8 |
| 8 | П. 5.3.4 | RBAC for Kubernetes и ролевая модель для ПРОД/не-ПРОД. | Б-7, Б-9, Б-10, Б-11, Б-12 |
| 9 | П. 3.1.10 | ACL и RBAC для управления правами доступа. | Б-7, Б-9, Б-10, Б-11 |
| 10 | П. 6.10.6 | Задание шаблона настроек безопасности Платформы. | Б-17 |
| 11 | П. 6.10.7 | Настройка Pod Security Standards. | Б-13 |
| 12 | П. 3.11.2 | Возможность формирования и применения политик безопасности. | Б-14, Б-15 |
| 13 | П. 3.11.3 | Проверка ресурсов кластера на соответствие политикам безопасности. | Б-14, Б-15 |
| 14 | П. 3.11.4 | Блокировка деплоя на основе проверки политик безопасности. | Б-14, Б-15 |
| 15 | П. 3.11.8 | Запуск контейнера ограничен минимально необходимыми POSIX разрешениями. | Б-15 |
| 16 | П. 3.11.9 | Платформа исключает запуск процессов в контейнерах под привилегированным пользователем root. | Б-15 |
| 17 | П. 3.11.11 | Платформа ограничивает мутабельность контейнеров в ПРОД и не-ПРОД средах с помощью запрета команд и набора ограничивающих правил. | Б-14, Б-15 |
| 18 | П.3.3.2 | Администрирование Платформы с использованием веб-интерфейса, интерфейса командной строки (CLI) и API. | Б-5 |
| 19 | П. 3.3.5 | Лимиты ресурсов для пространств имен (квоты). | Б-16 |
| 20 | П. 3.3.19 | Учет лимитов ресурсов при деплое манифестов. | Б-16 |
| 21 | П. 6.1.11 | Централизованное управление квотированием ресурсов. | Б-16 |
| 22 | П. 6.1.9 | Группировка ресурсов по изолированным средам (ПРОД/не-ПРОД). | Б-17 |
| 23 | П. 5.2.5 | Контроль событий runtime (запуск приложений, сетевые соединения). | Б-18 |
| 24 | П. 5.2.10 | Мониторинг команд и ограничивающих правил для sudo. | Б-18 |
| 25 | П. 3.11.1 | Поддержка CIS Benchmark для Kubernetes. | Б-20 |
| 26 | П. 5.2.8 | Мониторинг шифрования соединений (TLS). | Б-19 |
| 27 | П. 5.2.9 | Контроль TLS для соединений между подами. | Б-19 |
| 28 | П. 3.1.3 | Автоматический выпуск сертификатов кластеров. | Б-19 |
| 29 | П. 3.1.12 | Функционал Service Mesh. | Б-19 |
| 30 | П. 6.10.4 | Контроль конфигурации кластера по CIS Benchmarks. | Б-20 |
| 31 | П. 6.1.8 | Система поддерживает многоарендность с полной изоляцией арендаторов друг от друга. | Б-17 |
| 32 | П. 6.13.1 | Система интегрируется со службой каталогов Банка (AD) для аутентификации и авторизации с использованием доменных учетных данных, а также для автоматического наделения полномочий пользователей. | Б-8 |
| 33 | П. 3.11.12 | Монтирование файловой системы хоста обусловлено общим запретом на монтирование или выделением отдельного раздела с опциями: nodev, nosuid, noexec. | Б-13 |

### Методика тестирования

Тестирование безопасности и контроля доступа предполагает выполнение тестовых процедур, приведенных в таблице ниже.

Таблица 5. Порядок проведения тестирования безопасности и контроля доступа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Код требования в ТЗ** | **Название Тестового сценария** | **Приоритет проверки** | **Шаги** | **Ожидаемый результат** | **Время прохождения** |
| 1 | П. 5.2.11,  П. 6.12.1,  П. 6.12.2 | Б-1, Аудит событий Kubernetes API | Средний | Настроить сбор Kubernetes Audit Log, добавив соответствующую конфигурацию в манифест API-сервера. Убедиться, что файлы аудита записываются в указанное хранилище с полным логированием действий в кластере. Проверить содержимое файла аудита, чтобы подтвердить запись всех операций в Kubernetes API, в том числе действия пользователей.  Проверить, что в логах содержится информация об источнике, типе и времени события. | Сбор Kubernetes Audit Log настроен, и в файле аудита корректно записываются все действия, выполненные в Kubernetes. |  |
| 2 | П. 4.2.1,  П. 6.10.3,  П. 3.2.1 | Б-2, Фильтрации трафика внутри кластера (поддержка NetworkPolicy). Только для кластеров с CNI Cilium | Высокий | Настроить CNI Cilium с включением режима policyAuditMode, добавив соответствующие параметры в конфигурацию. Создать NetworkPolicy для ограничения трафика между подами в кластере. Протестировать работу NetworkPolicy, убедившись, что разрешенный и запрещенный трафик соответствует настроенным правилам. Проверить, что политики ограничивают трафик на уроне L4. | Режим policyAuditMode успешно настроен, NetworkPolicy работает корректно, трафик фильтруется в соответствии с заданными правилами. |  |
| 3 | П. 3.2.1,  П. 4.2.1,  П. 6.10.3 | Б-3, Фильтрации трафика на уровне L7 внутри кластера (поддержка CiliumNetworkPolicy). Только для кластеров с CNI Cilium | Высокий | Настроить кластер с CNI Cilium, добавив необходимые параметры для поддержки L7 фильтрации трафика. Создать CiliumNetworkPolicy, задав правила фильтрации на уровне L7 (например, HTTP методы или URI). Протестировать работу политики, проверив, что трафик соответствует заданным правилам, а неразрешенные запросы блокируются. | CiliumNetworkPolicy успешно настроена, фильтрация трафика на уровне L7 работает корректно, разрешенные запросы проходят, а запрещенные блокируются. |  |
| 4 | П. 6.10.3 | Б-4, Отображения действия политик (NetworkPolicy) в веб-интерфейсе. Только для кластеров с CNI Cilium | Низкий | Включить модуль cilium-hubble через конфигурацию Deckhouse, следуя документации. Проверить доступность веб-интерфейса Hubble и убедиться, что интерфейс отображает события, связанные с срабатыванием NetworkPolicy. Протестировать несколько сценариев фильтрации трафика, чтобы убедиться в корректной визуализации действий политик. | Модуль cilium-hubble включен, веб-интерфейс доступен, и в нем отображаются события, связанные с срабатыванием NetworkPolicy. |  |
| 5 | П.3.3.2 | Б-5, Администрирование Платформы с использованием веб-интерфейса, интерфейса командной строки (CLI) и API. | Высокий | Открыть веб-страницу по адресу console.<домен-кластера>, выполнить вход в систему, назначить ноде кластера новую роль. Открыть терминал, выполнить команду kubectl edit node <имя-ноды>, назначить ноде кластера новую роль, проверить применение конфигурации. | В веб-интерфейсе и через терминал настройки ноды применяются одинаково. |  |
| 6 | П. 4.2.5,  П. 6.13.1 | Б-6, Использования временных статических пользователей в кластере | Высокий | Создать временного статического пользователя через конфигурацию Deckhouse, указав его данные в соответствующем ресурсе. Настроить права доступа для созданного пользователя, обеспечив возможность входа в веб-интерфейсы платформы. Проверить успешную авторизацию пользователя в веб-интерфейсах с использованием указанных учетных данных. | Статический пользователь был успешно создан и использован для входа в веб-интерфейсы платформы. |  |
| 7 | П. 5.3.4,  П. 3.1.10 | Б-7, Использование статических групп пользователей в кластере | Высокий | Создать статическую группу пользователей, добавив соответствующую конфигурацию в Deckhouse. Настроить роль (Role) или роль с областью (ClusterRole), а также привязку роли (RoleBinding/ClusterRoleBinding) для предоставления прав доступа этой группе. Проверить успешное использование группы для выполнения операций через Kubernetes API. | Статическая группа пользователей была успешно создана и использована для выдачи прав доступа в Kubernetes API. |  |
| 8 | П. 4.2.5,  П. 6.13.1 | Б-8, Использование внешнего провайдера аутентификации (LDAP/ /OIDC) | Высокий | Включить модуль user-authn и настроить DexProvider для интеграции с внешним провайдером аутентификации (LDAP или OIDC). Настроить параметры подключения к LDAP-серверу, включая URL, базовый DN, фильтры пользователей и групп. Проверить успешную авторизацию пользователя в веб-интерфейсах Deckhouse с использованием учетных данных LDAP. | Модуль user-authn был включен, DexProvider настроен, и вход в веб-интерфейсы Deckhouse с использованием LDAP-учетных данных выполнен успешно. |  |
| 9 | П. 5.3.4,  П. 3.1.10 | Б-9, Настройка ролевой модели доступа на основе групп, атрибутов пользователя | Средний | Настроить ролевую модель доступа в Deckhouse, создав Role/ClusterRole и RoleBinding/ClusterRoleBinding с привязкой к группам пользователей из LDAP или OIDC. В конфигурации DexProvider указать соответствие групп в внешнем провайдере атрибутам пользователя. Проверить, что пользователь, входящий в указанную группу, успешно получает доступ в соответствии с заданными правами. | Доступ пользователю был успешно предоставлен на основе его принадлежности к группе в LDAP или OIDC, а права доступа применены корректно. |  |
| 10 | П. 5.3.4,  П. 3.1.10 | Б-10, Ограничение доступа пользователей к определенным namespace | Высокий | Создать Role и RoleBinding для заданного namespace, привязав их к пользователю/группе. Попробовать выполнить действия в этом namespace от имени пользователя. Проверить, что доступ к другим namespace отсутствует. | Пользователь получил доступ к заданному namespace и не имеет прав на работу с другими namespace. |  |
| 11 | П. 5.3.4,  П. 3.1.10, | Б-11, Возможность расширения прав доступа |  | Обновить роль доступа User, добавив в нее разрешения на работу с секретами (например, get, list, create). Применить обновленную роль через RoleBinding или ClusterRoleBinding для конкретного пользователя. Проверить, что пользователь получил дополнительные права и успешно выполняет операции с секретами. | Роль доступа User успешно расширена правами на работу с секретами, пользователь получил дополнительные права и может выполнять соответствующие операции. |  |
| 12 | П. 5.3.4 | Б-12, Использование сервисной учетной записи для выката прикладного ПО в платформу | Высокий | Создать сервисную учетную запись (ServiceAccount) в заданном namespace. Настроить Role и RoleBinding, предоставив учетной записи права на выкат приложений в этом namespace и ограничив доступ к другим namespace. Выполнить выкат приложения, используя созданную учетную запись, и проверить успешность операции. | Сервисная учетная запись создана с ограниченными правами, приложение успешно выкачено в заданный namespace, доступ к другим namespace отсутствует. |  |
| 13 | П. 6.10.7  П. 3.11.12 | Б-13, Использование политик безопасности Kubernetes (Pod Security Standards) | Высокий | Применить политику безопасности Kubernetes (Pod Security Standards) с уровнем Restricted в заданном namespace. Попробовать создать под с параметром privileged: true, нарушающим требования политики. Проверить, что под не был создан, а в событиях кластера отображается сообщение об ошибке, связанное с нарушением политики безопасности. | Политика Restricted успешно применена, под с параметром privileged: true не создан, а соответствующая ошибка зафиксирована в событиях кластера. |  |
| 14 | П. 3.11.3,  П. 3.11.4,  П. 3.11.2  П. 3.11.11 | Б-14, Использование операционных политик для безопасной работы прикладного ПО | Низкий | Применить OperationPolicy, задающую ограничения на выполнение определенных операций в кластере (например, запрет запуска подов с некорректными настройками). Попробовать создать под, который нарушает правила, установленные в OperationPolicy. Проверить, что под не был создан, а в событиях кластера отображается ошибка о нарушении политики. | OperationPolicy успешно применена, под с нарушением политики не создан, а соответствующая ошибка зафиксирована в событиях кластера. |  |
| 15 | П. 3.11.3,  П. 3.11.4,  П. 3.11.2  П. 3.11.8  П. 3.11.9  П. 3.11.11 | Б-15, Использование политик безопасности для безопасной работы прикладного ПО | Высокий | Создать и применить SecurityPolicy, определяющую ограничения для работы подов (например, запрет привилегированных контейнеров или использование определенных объемов). Попробовать создать под, который нарушает заданные правила безопасности в SecurityPolicy. Проверить, что под не был создан, а в событиях кластера отображается ошибка, связанная с нарушением политики. Прописать в политике минимально необходимые разрешения POSIX Capabilities. | SecurityPolicy успешно применена, под с нарушением правил не создан, а соответствующая ошибка зафиксирована в событиях кластера. |  |
| 16 | П. 3.3.5,  П. 3.3.19,  П. 6.1.11 | Б-16, Возможность использовать квот в рамках namespaces | Высокий | Создать ресурс ResourceQuota в заданном namespace, указав ограничения на использование вычислительных ресурсов (CPU, памяти и т.д.). Запустить поды (выполнить деплой манифестов), которые запрашивают ресурсов больше, чем разрешено квотой. Проверить, что такие поды не созданы, а в событиях кластера отображаются соответствующие ошибки о превышении квоты. | Ресурс ResourceQuota успешно применен, поды с запросами ресурсов, превышающими ограничения, не созданы, а соответствующие ошибки зафиксированы в событиях кластера. |  |
| 17 | П. 6.1.9,  П. 6.1.8  П. 6.10.6 | Б-17, Создание изолированного окружения по заготовленному шаблону | Низкий | Создать проект из заранее подготовленного шаблона, используя команду kubectl apply или соответствующий API. Убедиться, что все ресурсы из шаблона автоматически созданы в указанном namespace. Проверить корректность создания ресурсов, включая их статусы и работу. | Проект успешно создан из шаблона, все ресурсы автоматически появились в заданном namespace и работают корректно. |  |
| 18 | П. 5.2.5,  П. 5.2.10, | Б-18, Обнаружение угроз безопасности анализируя прикладное ПО и контейнеры | Низкий | Настроить модуль безопасности Deckhouse для обнаружения угроз, включая контроль выполнения команд внутри контейнеров и попыток сетевых соединений за пределы контейнера. Запустить shell в контейнере прикладного ПО и выполнить действия, которые должны быть зафиксированы системой безопасности (*nc -zv 8.8.8.8 443*). Проверить, что уведомление о подозрительной активности сработало корректно и отобразилось в логах или системах оповещения. | Уведомление о запуске shell в контейнере было успешно зафиксировано и обработано системой безопасности. |  |
| 19 | П. 3.1.3,  П. 3.1.12,  П. 5.2.8,  П. 5.2.9, | Б-19, Организация mTLS между узлами прикладного ПО | Высокий | Создать два сервиса, включив в их конфигурацию sidecar istio-proxy. Настроить политику безопасности Istio (PeerAuthentication) для включения mTLS между сервисами. Проверить, что сертификаты выпускаются автоматически. Проверить взаимодействие между сервисами, убедившись, что оно происходит с использованием mTLS, через логи или диагностические инструменты Istio. (Kiali). | Два сервиса с istio-proxy взаимодействуют друг с другом через mTLS, подтвержденное настройками и логами. |  |
| 20 | П. 3.11.1,  П. 6.10.4 | Б-20, Сканирование образов прикладного ПО на наличие известных уязвимостей | Низкий | Запустить контейнеры прикладного ПО в платформе Deckhouse, используя доступные манифесты или Helm-чарты. Настроить модуль безопасности Deckhouse для автоматического сканирования контейнерных образов на наличие известных уязвимостей (trivy). Проверить, что результаты сканирования отображаются в CustomResource и на дашбордах Grafana. | Контейнерные образы были успешно отсканированы, результаты сканирования отобразились в CustomResource и дашбордах Grafana. |  |

Таблица 6. Оценка трудозатрат на тестирование

|  |  |
| --- | --- |
| **Количество проверок** | **Общее время прохождения** |
| 20 |  |

### Критерии завершения

Все запланированные тесты были выполнены. Все выявленные ошибки зафиксированы.

## Мониторинг

### Цель тестирования

Проверить, что обеспечиваются требования к ИС "Платформа" на предмет производительности, доступности и корректности работы в реальном времени.

### Проверяемые требования

Таблица 7. Требования тестирования мониторинга

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Код требования в ТЗ** | **Краткое описание** | **Тестовые сценарии в ПМИ** |
| 1 | П. 3.3.9 | Встроенная система мониторинга на основе стека Prometheus и Grafana | М-1, М-2, М-3, М-8, М-4, М-7, М-6, М-5, М-9 |
| 2 | П. 3.3.10 | Автоматическая установка устанавливает и компонент мониторинга и визуализации метрик Prometheus и Grafana, а также настройка сбора метрик компонентов кластера и Платформы. | М-1, М-2, М-3, М-5, М-7, М-8 |
| 3 | П. 3.3.13 | Компоненты Платформы экспортируют метрики в формате Prometheus. | М-1, М-5, М-7, М-8 |
| 4 | П. 3.3.14 | Платформа автоматически настраивает панели в Grafana для отображения данных о нагрузке, утилизации и доступности компонентов, балансировщиков, Ingress и Egress. | М-4, М-5 |
| 5 | П. 3.3.15 | Платформа содержит описание событий мониторинга в формате Prometheus и позволяет настраивать отправку уведомлений о событиях. | М-1, М-5, М-6. М-9, М-10 |
| 6 | П. 3.3.17 | Агрегация доступности компонентов кластера по функциональности и всем кластерам. | М-11 |
| 7 | П. 3.4.1 | Интеграция с почтовой системой по протоколу SMTP, и с CommuniGate Pro. | М-10 |

### Методика тестирования

Тестирование отказоустойчивости/восстановления предполагает выполнение тестовых процедур, приведенных в таблице ниже.

Таблица 8. Порядок проведения тестирования мониторинга

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Код требования в ТЗ** | **Название Тестового сценария** | **Приоритет проверки** | **Шаги** | **Ожидаемый результат** | **Время прохождения** |
| **1** | П. 3.3.9 | М-1, Встроенный мониторинг состояния служебных компонент кластера | Высокий | Убедиться, что модуль мониторинга Deckhouse включен и корректно настроен. Проверить наличие в Grafana предустановленных дашбордов и метрик, связанных с состоянием служебных компонентов кластера, включая Deckhouse. Открыть соответствующие дашборды и убедиться в отображении актуальных метрик. | В Grafana присутствуют дашборды с метриками по Deckhouse, отображающими состояние служебных компонентов кластера, и данные обновляются корректно. |  |
| П.3.3.10 |
| П.3.3.13 |
| П.3.3.15 |
|  |
| 2 | П. 3.3.9 | М-2, Мониторинг аппаратных ресурсов платформы | Высокий | Убедиться, что модуль мониторинга аппаратных ресурсов (например, node-exporter) включен и настроен в Deckhouse. Проверить наличие в Grafana предустановленных дашбордов и метрик, связанных с мониторингом аппаратных ресурсов узлов кластера. Открыть соответствующие дашборды и убедиться, что метрики по CPU, памяти, диску и сети отображаются корректно. | В Grafana присутствуют дашборды и метрики по аппаратным ресурсам (node-exporter), данные отображаются и обновляются корректно. |  |
| П. 3.3.10 |
|  |
|  |
| 3 | П. 3.3.9 | М-3, Мониторинг Kubernetes в составе платформы | Высокий | Убедиться, что модуль мониторинга Kubernetes control plane включен и настроен в Deckhouse. Проверить наличие в Grafana предустановленных дашбордов и метрик, отображающих состояние компонентов Kubernetes control plane (Kube-API, Kube-scheduler, Kube-controller-manager и др.). Открыть соответствующие дашборды и убедиться, что метрики отображаются корректно и обновляются в реальном времени. | В Grafana присутствуют дашборды с метриками по Kubernetes control plane, данные отображаются и обновляются корректно. |  |
| П.3.3.10 |
|  |
|  |
| 4 | П. 3.3.9 | М-4, Встроенный мониторинг входящего трафика | Низкий | Убедиться, что модуль мониторинга Ingress Nginx включен и настроен в Deckhouse. Проверить наличие в Grafana дашбордов и метрик, связанных с входящим трафиком, включая состояние и производительность Ingress Nginx. Открыть дашборды и убедиться, что метрики по входящему трафику отображаются корректно. | В Grafana присутствуют дашборды с метриками по Ingress Nginx, данные о входящем трафике отображаются и обновляются корректно. |  |
| П. 3.3.14 |
|  |
|  |
|  |
| 5 | П.3.3.9 | М-5, Оценка использования ресурсов | Высокий | Убедиться, что модуль мониторинга ресурсов и Capacity Planning включен и настроен в Deckhouse. Проверить наличие в Grafana дашбордов и метрик, связанных с анализом использования ресурсов кластера (CPU, память, диск). Открыть дашборды и убедиться, что данные по текущей загрузке и прогнозированию использования ресурсов отображаются корректно. | В Grafana присутствуют дашборды с метриками по Capacity Planning, данные по использованию ресурсов отображаются и обновляются корректно. |  |
| П.3.3.10  П.3.3.13  П.3.3.14  П.3.3.15 |
| 6 | П. 3.3.9 | М-6, Уведомления (alerts) по нагрузке серверов кластера, количество ошибочных запросов ingress и пр. | Высокий | Проверить настройки Prometheus в Deckhouse, убедившись, что включены предустановленные правила для уведомлений (alerts). Просмотреть список доступных правил оповещений в Prometheus, связанных с нагрузкой серверов, ошибочными запросами Ingress и другими метриками. Смоделировать условия, вызывающие срабатывание правил, и убедиться, что уведомления генерируются корректно. | В платформе настроен обширный набор уведомлений в Prometheus, правила оповещения работают корректно и охватывают метрики нагрузки серверов, ошибочных запросов Ingress и другие показатели. |  |
| П. 3.3.15 |
| П. 3.4.1 |
|  |
| 7 | П. 3.3.9 | М-7, Расширенный мониторинг состояния прикладных сервисов | Средний | Применить лейбл для активации мониторинга на выбранный namespace. Создать заведомо неработающий Deployment в этом namespace (например, указав недоступный образ контейнера). Убедиться, что в Prometheus появляются уведомления (alerts) о проблемах с состоянием Deployment, и проверить отображение соответствующих метрик в Grafana. | После применения лейбла и создания неработающего Deployment в namespace система мониторинга генерирует уведомления о проблемах, и метрики отображаются в Grafana. |  |
| П. 3.3.13 |
| П. 3.3.10 |
|  |
|  |
|  |
| 8 | П. 3.3.9 | М-8, Мониторинг прикладных сервисов | Высокий | Развернуть приложение в кластере, которое генерирует метрики в формате Prometheus. Настроить сбор метрик, создав ServiceMonitor или аналогичный объект для интеграции с Prometheus. Убедиться, что метрики приложения появились в интерфейсе Prometheus и отображаются корректно. | Приложение успешно развернуто, метрики в формате Prometheus собраны и отображаются в интерфейсе Prometheus. |  |
| П. 3.3.10 |
| П. 3.3.13 |
| 9 | П.3.3.15 | М-9, Возможность добавления своего набора уведомлений (alerts) | Высокий | Создать ресурс CustomPrometheusRule, указав в нем пользовательские правила уведомлений (alerts), включая их условия и действия. Применить созданный ресурс в кластере с помощью команды kubectl apply. Проверить в интерфейсе Prometheus, что новое правило успешно загружено и отображается в списке активных правил. | CustomPrometheusRule успешно создан, пользовательские уведомления подгрузились в Prometheus и отображаются в списке правил. |  |
| П.3.3.9 |
| 10 | П.3.3.15 | М-10, Возможность отправки уведомлений (alerts) во внешнюю систему | Высокий | Создать и настроить ресурс CustomAlertmanager, указав конфигурацию для отправки уведомлений во внешнюю систему (например, Slack, email или PagerDuty). Применить конфигурацию в кластере с помощью команды kubectl apply. Создать тестовое условие для срабатывания уведомления и убедиться, что оно успешно отправлено во внешнюю систему. | CustomAlertmanager успешно настроен, уведомления из кластера отправляются во внешнюю систему, и тестовое уведомление было доставлено. |  |
| П.3.4.1 |
|  |
|  |
| 11 | П. 3.3.17 | М-11, Агрегация доступности компонентов кластера | Высокий | Перейти по адресу upmeter.<домен-кластера>, выполнить авторизацию. Убедиться в наличии отображения метрик доступности компонентов кластера. | По адресу upmeter.<домен-кластера> отображаются метрики доступности компонентов кластера |  |

Таблица 9. Оценка трудозатрат на тестирование

|  |  |
| --- | --- |
| **Количество проверок** | **Общее время прохождения** |
| 11 |  |

### Критерий завершения

Все запланированные тесты были выполнены. Все выявленные ошибки зафиксированы.

## Тестирование масштабируемости

### Цель тестирования

Цель тестирования масштабируемости - определение способности системы эффективно справляться с увеличением нагрузки и расширением ресурсов, сохраняя при этом стабильность и удовлетворительное время отклика.

### Проверяемые требования

Таблица 10. Требования тестирования масштабируемости

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Код требования в ТЗ** | **Краткое описание** | **Тестовые сценарии в ПМИ** |
| 1 | П. 3.1.7 | Платформа должна позволять управлять механизмом распределения подов между узлами кластера. | МШ-1 |
| 2 | П. 6.3.1 | Возможность автоматического горизонтального масштабирования Кластера (путем добавления или изъятия узлов) без остановки сервиса. | МШ-4 |
| 3 | П. 6.3.3 | Возможность использования предварительно подготовленных узлов в горячем резерве для моментального масштабирования | МШ-4 |
| 4 | П. 6.6.1 | Наличие механизма автоматического распределения нагрузки. | МШ-1, МШ-2, МШ-3 |

### Методика тестирования

Тестирование масштабируемости предполагает выполнение тестовых процедур, приведенных в таблице ниже.

Таблица 11. Порядок проведения тестирования масштабируемости

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Код требования в ТЗ** | **Название Тестового сценария** | **Шаги** | **Ожидаемый результат** | **Время прохождения** |
| 1 | П. 3.1.7 | МШ-1, Балансировка нагрузки контейнеров между узлами кластера | Создать Deployment с настройкой podAntiAffinity. Проверить, что поды распределились на разные узлы с помощью команды kubectl get pods -o wide. | Поды Deployment распределились между узлами кластера, соблюдая правила podAntiAffinity. |  |
| П. 6.6.1 |
|  |
|  |
|  |
|  |
| 2 | П. 6.3.3 | МШ-2, Масштабирование прикладных сервисов на основе бизнес метрик | Выкатить приложение с метриками в формате Prometheus. Настроить сбор метрик, создать ресурсы IngressMetric/ServiceMetric/PodMetric и HorizontalPodAutoscaler. Подать нагрузку и наблюдать масштабирование через kubectl get hpa. | При увеличении нагрузки сервис автоматически масштабируется, создавая новые поды. |  |
| П. 6.3.1 |
| П. 6.6.1 |
| П. 6.3.2 |
|  |
| 3 | П. 6.3.1 | МШ-3, Масштабирование прикладных сервисов на основе потребления ресурсов | Развернуть приложение, создать ресурс VerticalPodAutoscaler в одном из режимов (Auto, Recreate). Подать нагрузку и проверить изменение запросов ресурсов с помощью kubectl describe pods. | Поды пересозданы или ресурсы обновлены в соответствии с нагрузкой, в зависимости от выбранного режима VerticalPodAutoscaler. |  |
| П. 6.3.2 |
| П. 6.3.3 |
| П. 6.6.1 |
|  |
| 4 | П. 6.3.1,  П. 6.3.3 | МШ-4, Управление узлами кластера (добавление, удаление) | Добавить новый узел в Kubernetes кластер, настроить его и убедиться, что он перешел в статус Ready. Освободить узел от рабочих нагрузок, удалить его из кластера и очистить конфигурацию. Проверить, что узел удален и больше не отображается в списке узлов кластера. | Узел был успешно добавлен в кластер и перешел в статус Ready, затем был освобожден от рабочих нагрузок, удален из кластера и перестал отображаться в списке узлов. |  |

Таблица 12. Оценка трудозатрат на тестирование

|  |  |
| --- | --- |
| **Количество проверок** | **Общее время прохождения** |
| 4 |  |

### Критерий завершения

Все запланированные тесты были выполнены. Все выявленные ошибки зафиксированы.

## Интеграционное тестирование

Интеграционные тесты не предусмотрены, так как система не содержит интеграций с внешними компонентами или системами.

## Прочее тестирование

### Цель тестирования

Цель тестирования – убедиться в корректной работе встроенных механизмов безопасности и управления, включая автоматическое распространение секретов, перезапуск приложений при изменении конфигураций, настройку Ingress для входящего трафика, удалённый сбор и кратковременное хранение логов, а также безопасный доступ к кластеру

### Проверяемые требования

Таблица 13. Требования к прочему тестированию

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Код требования в ТЗ** | **Краткое описание** | **Тестовые сценарии в ПМИ** |
| 1 | П. 3.3.18 | Платформа поддерживает сбор логов с использованием Vector, Opensearch и Logstash. | ПР-1, ПР-2 |
| 2 | П. 5.2.11 | Платформа позволяет журналировать действия и выгружать логи на удаленный сервер. | ПР-1 |

### Методика тестирования

Таблица 14. Порядок проведения прочего тестирования

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Код требования в ТЗ** | **Название Тестового сценария** | **Приоритет проверки** | **Шаги** | **Ожидаемый результат** | **Время прохождения** |
| 1 | П. 3.3.18, 5.2.11 | ПР-1, Встроенные инструменты удаленного ведения и агрегации журналов (логов) | Низкий | Настроить модуль логирования для отправки журналов в удаленное хранилище (например, Elasticsearch или syslog). Проверить, что журналы отправляются и доступны в удаленном хранилище. | Журналы собираются и отправляются в удаленное хранилище, где доступны для анализа. |  |
| 2 | П. 3.3.18 | ПР-2, Встроенная система кратковременного хранения логов | Средний | Установить модуль loki и настроить log-shipper для отправки логов платформы. Убедиться, что логи доступны в интерфейсе Loki. | Логи платформы успешно отправляются и доступны в интерфейсе Loki. |  |

Таблица 15. Оценка трудозатрат на тестирование

|  |  |
| --- | --- |
| **Количество проверок** | **Общее время прохождения** |
| 2 |  |

### Критерий завершения

Все запланированные тесты были выполнены. Все выявленные ошибки зафиксированы.