СИСТЕМА TEST IT 5.0

Инструкция по установке

2024

Аннотация

В настоящем документе приведено руководство по установке, обновлению, перезапуску и удалению цифровой системе управления тестированием программного обеспечения «Test IT 5.0» (далее – системы).

Программа предназначена для реализации широкого комплекса задач по управлению тестированием программного обеспечения, а также для сбора аналитических данных и построения отчетности.

Содержание

1 Общие сведения

1.1 Обозначение и наименование программы

Полное наименование – Система Test IT 5.0.

Условное обозначение – Test IT.

1.2 ПО, необходимое для функционирования программы

Для функционирования клиентской части программы требуется веб-браузер (рекомендуется Chrome 81+).

Для установки системы требуются установочные файлы, поддерживается Docker Compose и Kubernetes. Файлы доступны для загрузки на официальном сайте Test IT: https://testit.software/versions/lynx

Минимальные требования:

CPU: 4 ядра серверного класса с поддержкой виртуализации и тактовой частотой 2.2 ГГц и выше

RAM: 16 GB

Network: 100 Mbit/s

SSD: минимум 100 GB

SWAP: отключен

2 Функциональное назначение

Программа Test IT предназначена для управления тестированием ПО.

Область применения программы – тестирование ПО.

Выполняемые функции:

- создание проектов, позволяющих тестировать программный продукт или его часть вручную и автоматически;
- создание тест-кейсов и чек-листов и их хранения в библиотеке тестов;
- выделение повторяющихся в тест-кейсах действий в общие шаги;
- создание пользовательских атрибутов и наполнение тестовой документации тестовыми данными;
- создание параметров, позволяющих избежать дублирования тест-кейсов для разных итераций;
- создание тестовых наборов для различных конфигураций устройств, ОС и браузеров;
- получение системных уведомлений при назначении пользователя на выполнение тестов, упоминании пользователя, либо назначении пользователя в рамках атрибутов тестов;
- запуск автотестов из UI с помощью вебхуков, отслеживание результатов их прогона и анализ причин ошибок автотестов;
- получение отчётов по автоматизированным и ручным тестам в едином формате;
- визуализация аналитики с помощью дашбордов.

3 Установка системы

3.1 Установка в Docker Compose

В качестве примера в этой инструкции используется проект с именем testit. Вы можете использовать другое название.

Требования

Docker Engine 17.12.0 и выше

Docker Compose V2 и выше

Состав поставки

- .env — конфигурационный файл, содержащий переменные, используемые для обращения к контейнерам Test IT
- docker-compose.yml — конфигурационный файл Docker Compose
- docker-compose.elk.yml — конфигурационный файл Docker Compose с базами данных Elasticsearch, Logstash и Kibana
- backup.sh — скрипт запуска резервного копирования
- restore.sh — скрипт восстановления из резервной копии
- images.tar.gz — архив с образами (только в архиве для автономной установки)
- images.tar.gz — архив с образами (только в архиве для автономной установки)
- postgres-init.sql — инициализационный файл для контейнера базы данных

Подготовка

1. Измените значения переменных по умолчанию в .env-файле.
2. Задайте параметры vm.max_map_count=262144 и vm.overcommit_memory=1:

```
echo 'vm.max_map_count=262144' >> /etc/sysctl.conf
echo 'vm.overcommit_memory = 1' >> /etc/sysctl.conf
sysctl -p
```

3. Заблокируйте все порты, кроме порта 80, необходимого для доступа к пользовательскому интерфейсу.

4. Опционально: для обслуживания системы посредством протокола SSH, необходимо открыть порт 22 (может быть переназначено на конкретной конфигурации). Для работы по HTTPS необходимо открыть порт 443. Пример открытия доступа к портам для CentOS 7:

```
firewall-cmd --zone=public --add-port=80/tcp --permanent
firewall-cmd --zone=public --add-port=22/tcp --permanent
firewall-cmd --zone=public --add-port=443/tcp --permanent
firewall-cmd --reload
```

5. Опционально: включите логирование пользовательских действий. По умолчанию оно отключено.

Автономная установка

Данный тип установки поможет установить продукт, если сервер изолирован от сети Internet и нет возможности получить Docker образы с публичных репозиториев.

1. Скачайте дистрибутив со страницы загрузок.

2. Распакуйте содержимое архива автономной установки, например в папку ~/testit.

3. Выполните следующие команды:

```
cd ~/testit
docker load -i images.tar.gz
docker network create yoonion_network
docker compose -f docker-compose.yml --project-name testit up --detach --timeout 120
```

Установка онлайн

1. Скачайте файлы online-установки со страницы загрузок.

2. Распакуйте содержимое архива online-установки, например в папку ~/testit.

3. Выполните следующие команды:

4. cd ~/testit

5. docker network create yoonion_network

6. docker compose -f docker-compose.yml --project-name testit up --detach --timeout 120

3.2 Установка в Kubernetes

Требования

- Установленный в кластере ingress-контроллер, например Nginx Ingress Controller
- Настроенный поставщик Persistent Volumes
- Наличие Kubectl
- Наличие Helm

Состав поставки

Содержимое сборки:

- testit_backend/ — Helm-чарт для внутреннего интерфейса (backend)
- testit_frontend/ — Helm-чарт для внешнего интерфейса (frontend)
- scripts/ — папка, содержащая вспомогательные скрипты
- jobs/ — папка, содержащая вспомогательные объекты K8s

Содержимое чарта:

- templates/ — шаблон манифеста K8s
- Chart.yaml — основной конфигурационный файл чарта
- values.yaml — настройки и переменные базовых шаблонов
- values-override.yaml — файл переопределения настроек и переменных шаблона
- values-ssl.yaml— пример переопределения для включения внутреннего SSL

Установка приложения

1. Перед установкой проверьте наличие файла values.yaml в helm-чартах testit_backend и testit_frontend.

2. В случае необходимости переназначьте переменные в файле values-override.yaml, используя такие же отступы и иерархию, что и в файле values.yaml. В случае переопределения переменных добавьте флаг -f values-override.yaml во все команды helm, как показано в примере ниже.

3. Распакуйте файлы приложений с помощью команды:

unzip <testit_archive_name> -d <destination_folder>

4. Установите приложение с помощью команды:

cd ~/testit

Установите приложения бэкенда.

helm upgrade --install -f testit_backend/values-override.yaml -n <namespace> --create-namespace testit-backend testit_backend/

Дождитесь начала работы всех модулей бэкенда.

watch -n 1 kubectl -n <namespace> get pods

Установите внешний интерфейс (фронтенд).

helm upgrade --install -f testit_frontend/values-override.yaml -n <namespace> --create-namespace testit-frontend testit_frontend/

Дождитесь начала работы всех модулей внешнего интерфейса

watch -n 1 kubectl -n <namespace> get pods -l app=frontend

5. Перейдите в приложение, используя адрес, указанный в .Values.ingress.host в testit_frontend/values.yaml или testit_frontend/values-override.yaml :

testit_frontend/values.yaml или testit_frontend/values-override.yaml

ingress:

  host: "my-testit.example.com"

4 Обновление системы

4.1 Обновление в Docker Compose

При автономном обновлении предварительно загрузите образы из архива images.tar.gz в выбранный вами локальный репозиторий. Убедитесь, что кластер имеет доступ к этому репозиторию.

1. Создайте новую директорию, скачайте и распакуйте в ней архив с новой поставкой Helm-чарта.

2. Сравните содержимое файлов values.yaml и values-override.yaml для внешнего и внутреннего интерфейса (frontend и backend) и перенесите пользовательские параметры в соответствующие файлы новой версии.

3. В командной строке перейдите в директорию с новой версией и выполните следующие команды:

Обновите приложения бэкенда.

helm upgrade --install --reset-values -f testit_backend/values-override.yaml -n <namespace> --create-namespace testit-backend testit_backend/

Дождитесь начала работы всех модулей бэкенда (все поды в статусе Running).

watch -n 1 kubectl -n <namespace> get pods

Обновите внешний интерфейс (фронтенд).

helm upgrade --install --reset-values -f testit_frontend/values-override.yaml -n <namespace> --create-namespace testit-frontend testit_frontend/

Дождитесь начала работы всех модулей внешнего интерфейса

watch -n 1 kubectl -n <namespace> get pods -l app=frontend

4.2 Обновление в Kubernetes

При автономном обновлении предварительно загрузите образы из архива images.tar.gz в выбранный вами локальный репозиторий. Убедитесь, что кластер имеет доступ к этому репозиторию.

1. Создайте новую директорию, скачайте и распакуйте в ней архив с новой поставкой Helm-чарта.

2. Сравните содержимое файлов values.yaml и values-override.yaml для внешнего и внутреннего интерфейса (frontend и backend) и перенесите пользовательские параметры в соответствующие файлы новой версии.

3. В командной строке перейдите в директорию с новой версией и выполните следующие команды:

Обновите приложения бэкенда.

helm upgrade --install --reset-values -f testit_backend/values-override.yaml -n <namespace> --create-namespace testit-backend testit_backend/

Дождитесь начала работы всех модулей бэкенда (все поды в статусе Running).

watch -n 1 kubectl -n <namespace> get pods

Обновите внешний интерфейс (фронтенд).

helm upgrade --install --reset-values -f testit_frontend/values-override.yaml -n <namespace> --create-namespace testit-frontend testit_frontend/

Дождитесь начала работы всех модулей внешнего интерфейса

watch -n 1 kubectl -n <namespace> get pods -l app=frontend Перезапуск системы

5 Перезапуск системы

5.1 Перезапуск в Docker Compose

В качестве примера в этой инструкции используется проект с именем testit. Вы можете использовать другое название.

- Для перезапуска системы воспользуйтесь следующей командой:

docker compose -f docker-compose.yml --project-name testit restart --timeout 120

5.2 Перезапуск в Kubernetes

- Для полного перезапуска системы используйте команду:

kubectl delete pods --all -n <namespace_name>

6 Удаление системы

6.1 Удаление в Docker Compose

В качестве примера в этой инструкции используется проект с именем testit. Вы можете использовать другое название.

Вы можете удалить систему Test IT с сохранением информации (тома и содержащиеся в них данные будут сохранены) или с потерей всех данных (полное удаление: все тома и содержащиеся в них данные будут потеряны).

Чтобы удалить систему с сохранением информации:

- Выполните команду:

docker compose -f docker-compose.yml --project-name testit down --timeout 120

Чтобы удалить систему полностью (с потерей данных):

Выполните команду:

- docker compose -f docker-compose.yml --project-name testit down --volumes --timeout 120

6.2 Удаление в Kubernetes

Эта инструкция описывает полное удаление системы с потерей всех данных. Чтобы сохранить данные, перед удалением создайте резервную копию.

Чтобы полностью удалить Test IT:

- Используйте набор команд:

Проверьте, имеются ли в пространстве имен установленные чарты.

helm -n <namespace> list

Удалите чарты внешнего интерфейса (frontend) и внутреннего интерфейса (backend).

helm -n <namespace> uninstall testit-frontend

helm -n <namespace> uninstall testit-backend

```
# Опционально: Дождитесь остановки подов.
kubectl -n <namespace> get pods --watch
```