ООО «Высокие цифровые технологии»

Инструкция по установке Mimic в Kubernetes

Оглавление

О документе

Данный документ служит руководством для технических специалистов, желающих установить программное обеспечение «Mimic».

1 Установка в Kubernetes

Kubernetes является желательным методом для развертывания Mimic в производственной среде. Для упрощения процесса развертывания и управления приложением Mimic рекомендуется воспользоваться официальным Helm-чартом, который включает в себя все необходимые зависимости и настройки, обеспечивая при этом соответствие всем требованиям вашей среды.

Мы используем Helm для управления конфигурацией сервисов в Kubernetes. В этом процессе у нас есть основной шаблон Helm-чарта, который включает в себя все необходимые настройки для развертывания приложения в кластере.

Параметры, которые могут варьироваться в зависимости от конкретной среды, хранятся в файле values.yaml. Этот файл позволяет конечным пользователям задавать свои уникальные настройки, которые будут применены при развертывании.

Когда мы запускаем Helm, он берет значения из values.yaml и подставляет их в основной шаблон, заменяя стандартные параметры. Если в будущем шаблон сервиса обновляется, например, добавляется новая опция, это изменение будет автоматически применено к развернутой версии приложения при следующем обновлении через Helm.

Программное обеспечение Mimic развертывается с помощью Helm-чарта, который предоставляет стандартный способ установки. Пользователям рекомендуется использовать именно этот метод для упрощения процесса и обеспечения совместимости.

Если же пользователь предпочитает альтернативные подходы для развертывания Mimic, это означает, что пользователь способен настроить такую установку, имея достаточно знаний и опыта. В этом случае конечный пользователь берет на себя полную ответственность за мониторинг изменений в оригинальном Helm-чарте и за обновление своей конфигурации, не полагаясь на поддержку со стороны разработчиков Mimic.

Важно отметить, что при использовании нестандартных методов установки поддержка со стороны команды Mimic будет ограничена.

1.1 Требования к ресурсам

Чтобы гарантировать, что развернутая система сможет эффективно функционировать под нагрузкой в производственной среде, мы советуем учитывать следующие ресурсы, необходимые для правильной работы сервисов Mimic:

№	Сервис	ЦПУ	Оперативная
1	Frontend	0,5 CPU	1Gi
2	Project	0,5 CPU	1Gi
3	Algorithm	0,5 CPU	1Gi
4	EntryPoint	0,5 CPU	1Gi
	Итого:	2 CPU	4 Gi

Окончательные настройки ресурсов подлежат корректировке на основе профиля нагрузки. Мы рекомендуем использовать описанные выше ресурсы по умолчанию и не опускаться ниже указанных значений.

Чтобы убедиться, что в кластере достаточно ресурсов для развертывания Mimic, используйте команду:

```
kubectl describe nodes
```

1.2 Предварительные условия

Перед началом установки убедитесь, что у вас выполнены следующие шаги:

1. Развернут и настроен **кластер Kubernetes**.
2. Установлены инструменты **Helm** и **kubectl**, настроен доступ к вашему кластеру.
3. Развернута и настроена **KeyCloak** для возможности авторизации (если необходимо).
4. Получены **учетные данные** для доступа к приватному реестру образов и репозиторию Helm (https://repo.hd-tech.ru/).
5. Настроена **база данных PostgreSQL** на отдельном сервере.
6. Создан **ImagePullSecret** (детали см. в пунктах ниже).
7. Установлен **Ingress-контроллер** (детали см. в пунктах ниже).

1.2.1 Создание ImagePullSecret

ImagePullSecrets используются для аутентификации при доступе к приватным контейнерным реестрам, из которых необходимо загружать образы контейнеров. Если ваш образ находится в приватном реестре, Kubernetes должен знать, как аутентифицироваться, чтобы получить доступ к этому образу.

Для создания ImagePullSecret выполните следующие шаги:

1. **Подключиться к репозиторию** repo.hd-tech.ru **и создать токен**.

   Для этого нужно войти в систему по полученным в письме учётным данным и в профиле пользователя создать токен.

2. **Создайте секрет** для доступа к вашему реестру:

```
kubectl create secret docker-registry docker-registry \
  --docker-server= https://repo.hd-tech.ru \
  --docker-username=<имя-пользователя> \
  --docker-password=<созданный токен из репозитория> \
  --namespace=Mimic
```

3. Проверьте, что **секрет создан и работает корректно**:

```
kubectl describe secret docker-registry -n Mimic
```

Если возникают проблемы с созданием ImagePullSecret, то необходимо сбросить настройки доступа и проверить права пользователя.

1.2.2 Установка Ingress-контроллера

Для доступа к веб-интерфейсу Mimic необходимо установить Ingress-контроллер, который необходим для маршрутизации внешних HTTP/HTTPS-запросов в кластер Kubernetes. Рекомендуется использовать nginx. Пример:

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-
nginx/main/deploy/static/provider/cloud/deploy.yaml
```

Проверить установку Ingress-контроллера:

```
kubectl get pods -n ingress-nginx
```

1.3 Конфигурация

Для развертывания с использованием нашего Helm-чарта вам необходимо **настроить файл values.yaml**, который содержит настройки, специфичные для вашей среды. Обратите внимание на следующие пункты:

1. Все настройки выполняются, используя файл **values.yaml**.
2. Вам следует **создать свою конфигурацию**, основываясь на файле values.yaml, который вы найдете в архиве (https://repo.hd-tech.ru/repository/octo-helm/mimic-<версия>.tgz).

Для получения доступа к архиву, вам понадобятся имя пользователя и пароль (те же, что используются для загрузки образов). Эти данные будут предоставлены вам при получении ознакомительной или коммерческой лицензии.

Helm будет использовать ваши настройки из values.yaml и применит их к чарту перед тем, как развернуть Mimic.

1.3.1 Подготовка файла конфигурации values.yaml

Файл **values.yaml** используется для настройки развертывания с помощью Helm-чарта Mimic. Вам потребуется настроить этот файл в соответствии с вашими требованиями.

Основные параметры:

- **localChart** – в данном блоке можно задать установку **postgres** при развертывании Mimic или указать значение **false**, если будут использоваться уже имеющиеся ресурсы.
  Postgresql: true используется для развертывания ознакомительных версий и не рекомендуется ставить для промышленной эксплуатации, т.к. есть риск потери данных.

```
# если значение true включает локальное использование чарта
localChart:
  postgresql: false
```

- **imagePullSecrets** – в данном блоке указывается имя созданного ранее секрета для аутентификации при загрузке образов из приватного реестра (docker-registry);
- **database** - параметры подключения к серверу баз данных PostgreSQL версии 15 или выше. На сервере необходимо заранее создать пустые БД:

  - ✔ mimic_frontend
  - ✔ mimic_project
  - ✔ mimic_algorithm

  Создать пользователя Mimic и выдать ему права владельца на базы выше;

  В блоке указать **host, port** сервера БД**.** Заполнить поля **user** и **password** данными созданного пользователя **Mimic**.

  Если выбрана установка **localChart.postgresql: true**, то пользователь и БД будут созданы автоматически при развертывании PostgreSQL, и данный блок необходимо заполнить пересекающиеся значения из блока **postgresql**.

- ✔ database. host = “postgresql”
- ✔ database.port = “5432”
- ✔ database. user = значение из <postgresql .auth.username>
- ✔ database. password = значение из <postgresql .auth. password>

- **admin** – в данном блоке указывается данные администратора системы Mimic:
  - ✔ **Name** - логин для входа
  - ✔ **Email** – почта администратора
  - ✔ **Password** - пароль для входа в систему

- **keycloak –** блок заполняется данными KeyCloak
  - ✔ **enable**: "true" – признак использования авторизации через KeyCloak
  - ✔ **usersAuth. url –** адрес сервиса KeyCloak
  - ✔ **usersAuth.realm** – realm используемый в KeyCloak
  - ✔ **version** – используемая версия KeyCloak

- **frontend, project, algorithm, entrypoint –** в данных блоках указывается настройки поднимаемых сервисов.
  - ✔ **Frontend –** интерфейс системы. Его хост используется для работы пользователей с системой.
  - ✔ **Project –** сервис проектов хранит все настройки и информацию о проектах.
  - ✔ **Algorithm –** сервис отвечает за логику срабатывания правил и формирование ответов.
  - ✔ **EntryPoint –** точка входа для подключения проверяемого сервиса к имитации
    - Для каждой тестовой среды поднимается свой EntryPoint
    - Возможные значения параметра **Stand:** «Dev», «Test», «Stage». Для точки, развернутой в одном пространстве с сервисом Mimic и не имеющей стенда тестирования можно указать любое из допустимых значений.

- **mimic_admins** – список e-mail адресов администраторов, которым будет предоставлен доступ к подам в Kubernates (необязательный блок)

```
Mimic_admins:  - administrator@example.ru
```

- **certificates –** перечисляются сертификаты для работы по SSL в виде текста.

Пример:

```
certificates:
  cert: |
-----BEGIN CERTIFICATE-----
      .....
      ......
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
      …..
-----END CERTIFICATE-----
```

1.4 Установка Mimic с помощью Helm

После выполнения предварительных условий и конфигурирования фала values.yaml, можно приступать к установке Mimic. Для этого:

1. **Добавьте репозиторий Helm следующей командой**

```
helm repo add Mimic https://repo.hd-tech.ru/repository/octo-helm
--username <ваше-имя-пользователя> --password <созданный токен>
```

2. **Обновите репозиторий Helm**

```
helm repo update
```

3. **Установите Mimic используя настроенный ранее ваш файл** values.yaml

```
helm upgrade --install mimic ./mimic-chart/ -n mimic -f
values.yaml --atomic
```

Опция **--atomic** обеспечивает возможность отката до предыдущего состояния в случае ошибки.

1.5 Доступ к интерфейсу Mimic

После завершения развертывания, Mimic будет доступен через Ingress по указанному хосту в конфигурации. Например:

```
frontend:
  ingress:
    host: "mimic.apps.hd-tech.ru"
```

Пожалуйста, **подождите около 5 минут** после установки перед первым входом в веб-интерфейс системы.

Первый вход можно выполнить указав имя и пароль из используемого при развертывании файла values.yaml **admin.name** и **admin.password**.

Работоспособность сервиса после развертывания можно проверить следующей командой:

```
kubectl get ingress -n Mimic
```

Для проверки состояния запущенных подов:

```
kubectl get pods -n Mimic
```

1.6 Обновление Mimic до новой версии

Перед обновлением системы рекомендуется сделать резервные копии всех БД Mimic.

Для обновления Mimic выполните следующие шаги:

1. **Скачайте архив Helm-чарта** нужной версии. Версия чарта равна версии системы Mimic. https://repo.hd-tech.ru/repository/octo-helm/mimic-<версия>.tgz и распакуйте его

2. **Измените параметры в своем файле** values.yaml, если это необходимо (например, если в новой версии появились дополнительные параметры в шаблоне helm).

3. **Выполните команду обновления**:

```
helm upgrade --install mimic ./mimic-chart/ -n mimic -f
```

Дополнение:

Если требуется откат к предыдущей версии, используйте команду:

```
helm rollback
```

Следуя этим инструкциям, вы сможете успешно развернуть и настроить Mimic в вашем кластере Kubernetes. Если у вас возникнут дополнительные вопросы, пожалуйста, обратитесь к команде поддержки Mimic.

1.7 Установка и настройка дополнительных EntryPoint Mimic

Для каждой тестовой среды необходимо развернуть свой **EntryPoint** в соответствующем контуре с доступом к host сервиса **project** и **algorithm** указанных в файле values**.** Также нужно указать соответствующее значение параметра среды **Stand** (Dev, Test, Stage).

```
entrypoint:
  replicas: 1
  stand: "Stage"
```

1.8 Диагностика и отладка

Если возникают проблемы при развертывании, используйте следующие команды для получения информации и логов:

Получение информации обо всех ресурсах в пространстве имен Mimic:

```
kubectl get all -n Mimic
```

Просмотр логов сервиса пользовательского интерфейса:

```
kubectl logs -f -n Mimic -l app=Mimic-frontend
```

Получения подробной информации о подах:

```
kubectl describe pod <pod-name> -n Mimic
```

Проверки ресурсов кластера:

```
kubectl top pod -n Mimic
```

Полное удаление инстанса Mimic

```
helm delete Mimic -n Mimic
```