

Информационная система  
«Интеграционная платформа 2.0».  
Руководство по установке

Всего 20 листов

---

# Оглавление

<b>1</b>	<b>ОБЩИЕ СВЕДЕНИЯ</b>	<b>4</b>
1.1	О ДОКУМЕНТЕ	4
1.2	О ПРОДУКТЕ	4
<b>2</b>	<b>СИСТЕМНЫЕ ТРЕБОВАНИЯ</b>	<b>4</b>
2.1	ТРЕБОВАНИЯ К АППАРАТНОМУ ОБЕСПЕЧЕНИЮ	4
2.1.1	<i>Минимальные характеристики и конфигурация серверов</i>	4
2.1.2	<i>Рекомендуемые характеристики и конфигурация серверов</i>	5
2.2	ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	5
2.2.1	<i>Сервер базы данных</i>	5
2.2.2	<i>Сервер приложений</i>	5
2.2.3	<i>Web-сервер</i>	5
2.3	ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ ПОЛЬЗОВАТЕЛЕЙ	5
2.3.1	<i>Рабочая станция web-разработчика</i>	6
2.3.2	<i>Рабочая станция backend разработчика</i>	6
2.3.3	<i>Рабочая станция пользователя</i>	6
<b>3</b>	<b>УСТАНОВКА</b>	<b>6</b>
3.1	СОСТАВ ОСНОВНЫХ КОМПОНЕНТ	6
3.2	УСТАНОВКА KUBERNETES	7
3.2.1	<i>Установка и настройка kubectl</i>	7
3.2.2	<i>Проверка конфигурации kubectl</i>	11
3.2.3	<i>Дополнительная конфигурация kubectl</i>	11
3.3	УСТАНОВКА ELASTICSEARCH С ПОМОЩЬЮ DOCKER	13
3.3.1	<i>Скачивание Docker-образа</i>	13
3.3.2	<i>Запуск и настройка кластера Elasticsearch</i>	13
3.3.3	<i>Проверка работоспособности Elasticsearch</i>	14
3.3.4	<i>Установка elasticsearch-head</i>	15
3.4	УСТАНОВКА KIBANA	16
3.4.1	<i>Требования к версиям Kibana и Elasticsearch</i>	16
3.4.2	<i>Скачивание Docker-образа Kibana</i>	16
3.4.3	<i>Запуск Kibana</i>	16
3.5	УСТАНОВКА LOGSTASH	17
3.5.1	<i>Требования к версии Logstash</i>	17
3.5.2	<i>Скачивание Docker-образа Logstash</i>	17
3.5.3	<i>Хранение файлов конфигурации</i>	17
3.6	УСТАНОВКА FILEBEAT	18
3.6.1	<i>Требования к версии Filebeat</i>	18
3.6.2	<i>Скачивание Docker-образа Filebeat</i>	18
3.6.3	<i>Хранение файлов конфигурации</i>	19
3.6.4	<i>Запуск Filebeat</i>	20

---

---

4	ОБЩАЯ ПРОВЕРКА РАБОТОСПОСОБНОСТИ.....	20
---	---------------------------------------	----

## 1 Общие сведения

### 1.1 О документе

Настоящий документ представляет собой руководство по установке ИП 2.0 с использованием установочных файлов, входящих в комплект поставки.

### 1.2 О продукте

Информационная система «Интеграционная платформа 2.0» (далее, **ИП 2.0**) предназначена для организации обмена данными между распределенными приложениями, а также для поддержки, мониторинга и управления композитными процессами.

Платформа обеспечивает управление контейнеризованными приложениями в среде Kubernetes, выполненными в микросервисной архитектуре.

## 2 Системные требования

### 2.1 Требования к аппаратному обеспечению

В разделе перечислены требования к программному и аппаратному окружению для установки основных модулей системы. Также в разделе представлен перечень внешних программных продуктов и библиотек.

#### 2.1.1 Минимальные характеристики и конфигурация серверов

Производитель заявляет, что для промышленной эксплуатации ИП 2.0 в минимальной конфигурации требуется шесть выделенных серверов:

Таблица 1

Назначение	Количество серверов	Характеристики сервера		
		CPU	RAM, GB	Disk space, GB
Сервер базы данных	2	4	8	Минимум 50 GB HDD (без учета дальнейшего прироста)
Сервер приложений	3	4	8	50 GB HDD
Web-сервер	1	2	4	50 GB HDD

### 2.1.2 Рекомендуемые характеристики и конфигурация серверов

Требования к промышленной нагрузке рассчитываются индивидуально в зависимости от типа и количества используемых модулей и потоков.

## 2.2 Требования к программному обеспечению

Для успешной установки ИП 2.0 на выделенных серверах должно быть установлено соответствующее программное обеспечение. Для каждого типа сервера указан свой набор ПО и требования к нему.

### 2.2.1 Сервер базы данных

Таблица 2

Компонент	Версия
Linux	Версия ядра не ниже 2.6.x
PostgreSQL	Не ниже 9.5

### 2.2.2 Сервер приложений

Таблица 3

Компонент	Версия
Linux	Версия ядра не ниже 2.6.x
Nginx/HAProxy	Не ниже 1.18.0

### 2.2.3 Web-сервер

Таблица 4

Компонент	Версия
Linux	Версия ядра не ниже 2.6.x
Nginx	Не ниже 1.18.0
Stunnel	Не ниже 5.3.0

## 2.3 Требования к программному обеспечению пользователей

При разработке и обновлении продукта могут изменяться как frontend, так и backend решения. К рабочим станциям таких специалистов предъявляются отдельные требования к установленному ПО.

### 2.3.1 Рабочая станция web-разработчика

Таблица 5

Компонент	Версия
Операционная система	Windows: 7/8/10 MacOS Ubuntu — не ниже 16.04.6 LTS
Браузер (совместимость)	Google Chrome, Yandex Browser — Latest

### 2.3.2 Рабочая станция backend разработчика

Таблица 6

Компонент	Версия
Операционная система	Windows: 7/8/10 Mac OS Ubuntu — не ниже 16.04.6 LTS
Браузер (совместимость)	Google Chrome, Yandex Browser — Latest

### 2.3.3 Рабочая станция пользователя

Таблица 7

Компонент	Версия
Операционная система	Windows: 7/8/10 Mac OS Ubuntu — не ниже 16.04.6 LTS
Браузер (совместимость)	Google Chrome, Yandex Browser — Latest

## 3 Установка

### 3.1 Состав основных компонент

- Модуль Kubernetes
- Модуль Elasticsearch
- Модуль Kibana
- Модуль Logstash

## 3.2 Установка Kubernetes

### 3.2.1 Установка и настройка kubectl

Инструмент командной строки Kubernetes [kubectl](#) позволяет запускать команды для кластеров Kubernetes. Вы можете использовать kubectl для развертывания приложений, проверки и управления ресурсами кластера, а также для просмотра логов. Полный список операций kubectl смотрите в [Overview of kubectl](#).

**Внимание!** Используемая вами мажорная версия kubectl не должна отличаться от той, которая используется в кластере. Например, версия v1.2 может работать с версиями v1.1, v1.2 и v1.3. Использование последней версии kubectl поможет избежать непредвиденных проблем.

#### 3.2.1.1 Установка kubectl в Linux

Kubectl в Linux можно установить несколькими способами:

- С помощью утилиты cURL
- С помощью встроенного в Linux пакетного менеджера
- С помощью стороннего пакетного менеджера.

Для установки kubectl с помощью cURL выполните следующие действия:

- Загрузите последнюю версию с помощью команды:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
```

*Примечание: Чтобы загрузить нужную версию, укажите ее в ссылке.*

*Например, команда для загрузки версии v1.25.0 для Linux будет выглядеть следующим образом:*

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.25.0/bin/linux/amd64/kubectl
```

- Сделайте двоичный файл kubectl исполняемым:

```
chmod +x ./kubectl
```

- Переместите двоичный файл в директорию из переменной окружения PATH:

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

- Убедитесь, что установлена нужная вам версия:

```
kubectl version --client
```

Для установки с помощью встроенного пакетного менеджера:

- В ОС на базе Ubuntu, Debian или HyperOS выполните команды

```
sudo apt-get update && sudo apt-get install -y apt-transport-https
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key
add -
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a
/etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update
sudo apt-get install -y kubectl
```

- В ОС на базе CentOS, RHEL или Fedora выполните команды

```
sudo cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
sudo yum install -y kubectl
```

Для установки с помощью стороннего пакетного менеджера:

- Если вы используете Ubuntu или другой Linux-дистрибутив, в котором есть пакетный менеджер [snap](#), kubectl доступен в виде приложения [snap](#).

```
snap install kubectl --classic
kubectl version
```

- Если вы работаете в Linux и используете пакетный менеджер [Homebrew](#), то установите kubectl с помощью [Homebrew](#)

```
brew install kubectl
kubectl version
```

### 3.2.1.2 Установка kubectl в macOS

Kubectl в macOS можно установить несколькими способами

- С помощью утилиты cURL
- С помощью пакетного менеджера Homebrew
- С помощью Macports

Для установки kubectl с помощью cURL выполните следующие действия:

- Загрузите последнюю версию с помощью команды

```
curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/darwin/amd64/kubectl"
```

*Примечание: Чтобы загрузить нужную версию, укажите ее в ссылке.*

*Например, команда для загрузки версии v1.25.0 для Linux будет выглядеть следующим образом:*

```
curl -LO https://storage.googleapis.com/kubernetes-
release/release/v1.25.0/bin/darwin/amd64/kubectl
```

- Сделайте двоичный файл kubectl исполняемым:

```
chmod +x ./kubectl
```



- Переместите двоичный файл в директорию из переменной окружения PATH:

```
sudo mv ./kubectl /usr/local/bin/kubectl
```

- Убедитесь, что установлена нужная вам версия

```
kubectl version --client
```

Для установки с помощью встроенного пакетного менеджера Homebrew:

- Установите kubectl

```
brew install kubectl
```

Или

```
brew install kubernetes-cli
```

- Убедитесь, что установлена последняя версия

```
kubectl version --client
```

Для установки с помощью Macports:

- Выполните команды установки

```
sudo port selfupdate  
sudo port install kubectl
```

- Убедитесь, что установлена последняя версия

```
kubectl version --client
```

### 3.2.1.3 Установка kubectl в Windows

Kubectl в Windows можно установить несколькими способами

- Скачать и установить утилиту по [ссылке](#) (для версии v1.25.0)
- С помощью утилиты cURL
- С помощью Powershell из PSGallery
- С помощью пакетного менеджера Chocolatey или установщика Scoop.

Для установки kubectl с помощью cURL выполните следующие действия:

- Загрузите последнюю версию с помощью команды

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.25.0/bin/windows/amd64/kubectl.exe
```

*Примечание: Последнюю стабильную версию (например, при написании скриптов) вы можете узнать из файла по [ссылке](#).*

- Переместите двоичный файл в директорию из переменной окружения PATH:
- Убедитесь, что установленная версия kubectl совпадает с загруженной

```
kubectl version --client
```

---

*Примечание: [Docker Desktop for Windows](#) добавляет собственную версию `kubectl` в переменную окружения `PATH`. Если у вас установлен `Docker Desktop`, то задайте путь к установленному двоичному файлу `kubectl` перед записью, добавленной установщиком `Docker Desktop`, либо удалите `kubectl`, поставляемый вместе с `Docker Desktop`.*

Для установки `kubectl` с помощью Powershell из PSGallery выполните следующие действия:

- Выполните установку, задав параметр для `DownloadLocation`

```
Install-Script -Name install-kubectl -Scope CurrentUser -Force  
install-kubectl.ps1 [-DownloadLocation <path>]
```

**Внимание!** Если вы не укажете `DownloadLocation`, то `kubectl` будет установлен во временную директорию пользователя.

- Установщик создаст `$HOME/.kube` вместе с конфигурационным файлом.
- Убедитесь, что установленная версия `kubectl` совпадает с загруженной

```
kubectl version --client
```

*Примечание: Чтобы обновить `kubectl`, повторите следующие команды:*

```
Install-Script -Name install-kubectl -Scope CurrentUser -Force  
install-kubectl.ps1 [-DownloadLocation <path>]
```

Для установки `kubectl` с помощью Chocolatey выполните следующие действия:

- Выполните установку

```
choco install kubernetes-cli
```

- Убедитесь, что установлена последняя версия

```
kubectl version --client
```

- Перейдите в домашнюю директорию

```
cd %USERPROFILE%
```

- Создайте директорию `.kube`:

```
mkdir .kube
```

- Перейдите в созданную директорию `.kube`:

```
cd .kube
```

- Настройте `kubectl` для использования удаленного кластера Kubernetes

```
New-Item config -type file
```

*Примечание: Отредактировать конфигурационный файл можно с помощью текстового редактора, например `Notepad`.*

### 3.2.1.4 Установка `kubectl` из SDK Google Cloud

Вы можете использовать `kubectl` из SDK Google Cloud, который использует этот CLI-инструмент. Для этого:

- Установите [Google Cloud SDK](#)
- Установите `kubectl`

```
gcloud components install kubectl
```

- Убедитесь, что установленная версия `kubectl` совпадает с загруженной

```
kubectl version --client
```

### 3.2.2 Проверка конфигурации `kubectl`

Чтобы `kubectl` мог найти и получить доступ к кластеру Kubernetes, нужен файл *kubeconfig*, который создается автоматически при создании кластера с помощью скрипта *kube-up.sh* или при успешном развертывании кластера Minikube.

По умолчанию конфигурация `kubectl` находится в `~/.kube/config`.

Посмотрите на состояние кластера, чтобы убедиться, что `kubectl` правильно сконфигурирован:

```
kubectl cluster-info
```

Если вы видите URL-ответ, значит `kubectl` корректно настроен для работы с вашим кластером.

Если вы видите сообщение следующего содержания, то значит `kubectl` настроен некорректно или не может подключиться к кластеру Kubernetes:

```
The connection to the server <server-name:port> was refused - did you specify the right host or port?
```

Например, если вы собираетесь запустить кластер Kubernetes на своем ноутбуке (локально), вам потребуется сначала установить специальный для этого инструмент, например Minikube, а затем снова выполнить указанные выше команды.

Если команда `kubectl cluster-info` возвращает URL-ответ, но вы не можете подключиться к своему кластеру, чтобы убедиться, что он правильно настроен, воспользуйтесь этой командой:

```
kubectl cluster-info dump
```

### 3.2.3 Дополнительная конфигурация `kubectl`

#### 3.2.3.1 Включение автодополнения ввода *shell*

`Kubectl` поддерживает автодополнение (автозаполнение) ввода в Bash и Zsh, которое сэкономит вам много времени на набор команд.

Для настройки автодополнения для Bash (для Linux и macOS) и Zsh воспользуйтесь инструкциями по [ссылке](#).

Скрипт дополнения ввода `kubectl` для Bash может быть сгенерирован с помощью команды `kubectl completion bash`. Подключение скрипта дополнения ввода в вашу оболочку включает поддержку автозаполнения ввода для `kubectl`.

---

Однако скрипт дополнения ввода зависит от [bash-completion](#), поэтому вам нужно сначала установить этот пакет (вы можете выполнить команду `type _init_completion`, чтобы проверить, установлен ли у вас уже `bash-completion`).

`bash-completion` можно установить через многие менеджеры пакеты (см. [ссылку](#)). Вы можете установить его с помощью:

- `apt-get install bash-completion`
- `yum install bash-completion` и других команд.

Приведенные выше команды создадут файл `/usr/share/bash-completion/bash_completion`, который является основным скриптом `bash-completion`. В зависимости от используемого менеджера пакетов, вы можете подключить этот файл в файле `~/.bashrc`.

Чтобы убедиться, что этот скрипт выполняется, перезагрузите оболочку и выполните команду:

```
type _init_completion
```

Если команда отработала успешно, установка сделана правильно, в противном случае добавьте следующее содержимое в файл `~/.bashrc`:

```
source /usr/share/bash-completion/bash_completion
```

Перезагрузите вашу оболочку и убедитесь, что `bash-completion` правильно установлен, напечатав в терминале:

```
type _init_completion
```

Теперь нужно убедиться, что скрипт дополнения ввода `kubectl` выполняется во всех сессиях командной оболочки. Есть два способа сделать это:

- Добавьте запуск скрипта дополнения ввода в файл `~/.bashrc`

```
echo 'source <(kubectl completion bash)' >>~/.bashrc
```

- Добавьте скрипт дополнения ввода в директорию `/etc/bash_completion.d`:

```
kubectl completion bash >/etc/bash_completion.d/kubectl
```

- Если у вас определен псевдоним для `kubectl`, вы можете интегрировать его с автодополнением оболочки:

```
echo 'alias k=kubectl' >>~/.bashrc
```

```
echo 'complete -F __start_kubectl k' >>~/.bashrc
```

*Примечание: Все скрипты дополнения ввода `bash-completion` находятся в `/etc/bash_completion.d`. Оба подхода эквивалентны. После перезагрузки вашей оболочки, должны появиться дополнения ввода `kubectl`.*

## 3.3 Установка Elasticsearch с помощью Docker

### 3.3.1 Скачивание Docker-образа

Используя Docker можно быстро запустить Elasticsearch для сред тестирования/разработки и продуктивного окружения. Для начала работы скачайте нужный Docker-образ:

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:7.1.1
```

### 3.3.2 Запуск и настройка кластера Elasticsearch

Чтобы запустить Elasticsearch на сервере, используя Docker, должны быть установлены Docker ([см. инструкцию](#)) и надстройка docker-compose ([см. инструкцию](#)) выполните команды:

- Для DEV/TEST-среды

```
docker run -p 9200:9200 -e "http.host=0.0.0.0" -e "transport.host=127.0.0.1"
docker.elastic.co/elasticsearch/elasticsearch:7.1.1
```

- Для PROD-среды

Для серверов на базе Linux для параметра ядра `vm_max_map_count` задайте значение 262144 или выше:

```
sysctl -w vm.max_map_count=262144
```

После установки проверьте, что в файле `/etc/sysctl.conf` внесено соответствующее изменение:

```
grep vm.max_map_count /etc/sysctl.conf
```

Если `sysctl -w` недействителен, то измените непосредственно сам файл:

```
vi /etc/sysctl.conf
```

Для развертывания кластера Elasticsearch для продуктивных сред обычно используется надстройка [docker-compose](#). Настройки будущего кластера задаются в специальном файле `docker-compose.yml`.

Для настройки кластера Elasticsearch, состоящего из двух узлов, используйте следующий `docker-compose.yml`:

```
version: '2.2'
services:
  es1:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.1.1
    container_name: es1
    environment:
      - cluster.name=docker-cluster
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
    ulimits:
      memlock:
        soft: -1
```

```

    hard: -1
  mem_limit: 1g
  volumes:
    - esdata1:/usr/share/elasticsearch/data
  ports:
    - 9200:9200
  networks:
    - esnet
es2:
  image: docker.elastic.co/elasticsearch/elasticsearch:7.1.1
  container_name: es2
  environment:
    - cluster.name=docker-cluster
    - bootstrap.memory_lock=true
    - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
    - "discovery.zen.ping.unicast.hosts=es1"
  ulimits:
    memlock:
      soft: -1
      hard: -1
  mem_limit: 1g
  volumes:
    - esdata2:/usr/share/elasticsearch/data
  networks:
    - esnet

volumes:
  esdata1:
    driver: local
  esdata2:
    driver: local

networks:
  esnet:

```

Чтобы запустить кластер, выполните команду:

```
docker-compose up
```

Рисунок 1

```

[]# docker-compose up -d
Starting elasticsearch1 ...
Starting elasticsearch1
Starting elasticsearch1 ... done
[]# docker-compose ps

```

Name	Command	State	Ports
elasticsearch1	/bin/bash bin/es-docker	Up	0 0.0.0:9200->9200/tcp, 9300/tcp
elasticsearch_elasticsearch2_1	/bin/bash bin/es-docker	Up	9200/tcp, 9300/tcp

Чтобы остановить кластер, выполните команду:

```
docker-compose down
```

При этом тома с данными останутся целыми.

Если требуется удалить кластер и данные, то выполните команду:

```
docker-compose down -v
```

### 3.3.3 Проверка работоспособности Elasticsearch

*Примечание:* По умолчанию имя пользователя/пароль задаются как *elastic/changeme*.

Узнать состояние кластера можно несколькими способами:

- С помощью браузера.  
Для этого откройте браузер, в адресной строке введите [http://127.0.0.1:9200/\\_cat/health](http://127.0.0.1:9200/_cat/health)  
В открывшемся окне будет выведена информация о состоянии кластера
- С помощью утилиты cURL:  
Для этого в командной строке выполните команду:

```
curl http://127.0.0.1:9200/_cat/health
1472225929 15:38:49 docker-cluster green 2 2 4 2 0 0 0 0 - 100.0%
```

Подробные логи могут выводиться в консоль средствами Docker.  
Чтобы просмотреть логи, выполните команду:

```
docker logs
```

### 3.3.4 Установка elasticsearch-head

Далее установите *elasticsearch-head*. Elasticsearch-head — это веб-интерфейс для Elasticsearch. Так как браузерные плагины для elasticsearch-head больше не поддерживаются, запустите *elasticsearch-head* на отдельном узле. Для запуска используйте команду Docker:

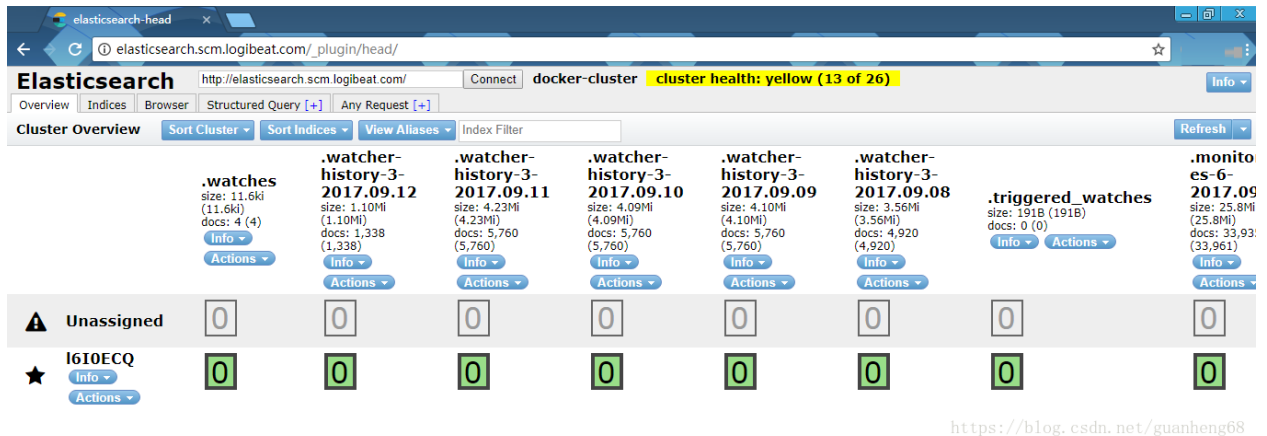
```
docker run -p 9100:9100 -d --name es-head mobz/elasticsearch-head:5
```

Elasticsearch-head вызывает API Elasticsearch. Поэтому доменное имя должно быть единообразным. Настройте проксирование с помощью nginx проху-pass:

```
location /_plugin/head {
    proxy_pass http://devops1-internal:9100/;
    proxy_redirect off;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $host;
    client_max_body_size 10m;
    client_body_buffer_size 128k;
    proxy_connect_timeout 360;
    proxy_send_timeout 180;
    proxy_read_timeout 180;
    proxy_buffer_size 8k;
    proxy_buffers 8 64k;
    proxy_busy_buffers_size 128k;
    proxy_temp_file_write_size 128k;
}
```

В случае успешной настройки по адресу [http://localhost/\\_plugin/head/](http://localhost/_plugin/head/) будет выведена информация о состоянии кластера Elasticsearch

Рисунок 2



## 3.4 Установка Kibana

### 3.4.1 Требования к версиям Kibana и Elasticsearch

Сервис визуализации логов Kibana устанавливается также с помощью Docker. Для этого используется официальное зеркало: [https://hub.docker.com/\\_/kibana](https://hub.docker.com/_/kibana).

**Внимание!** Версии устанавливаемых Elasticsearch и Kibana должны быть идентичными. Разница в версиях может вызвать проблемы.

### 3.4.2 Скачивание Docker-образа Kibana

Для того, чтобы скачать Kibana с сайта Docker, выполните команду:

```
docker pull kibana:7.1.1
```

Для контроля версий Docker-образов используйте команду:

```
docker images
```

Рисунок 3

```
$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
logstash        7.1.1       b0cb1543380d 2 weeks ago  847MB
kibana          7.1.1       67f17df6ca3e 2 weeks ago  746MB
elasticsearch    7.1.1       b0e9f9f047e6 2 weeks ago  894MB
```

*Примечание.* В этом примере Kibana использует конфигурацию по умолчанию и подключается к работающему экземпляру Elasticsearch по адресу `http://localhost:9200`.

### 3.4.3 Запуск Kibana

Поскольку Kibana запускается в контейнере, то выполните команду:

```
docker run -d --name kibana --net somenetwork -p 5601:5601 kibana:7.1.1
```

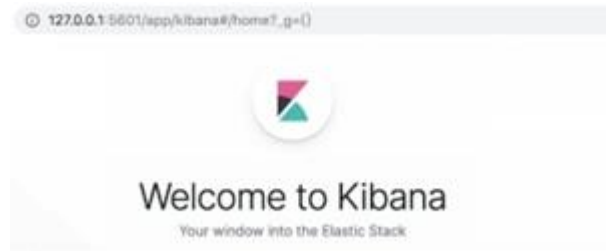
Чтобы просмотреть статус запуска контейнера, выполните команду `docker ps`

```
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
7a1d94279a29  kibana:7.1.1  "/usr/local/bin/kiba..." About an hour ago  Up 3 seconds  0.0.0.0:5601->5601/tcp
e8cc33557add  elasticsearch:7.1.1  "/usr/local/bin/dock..." About an hour ago  Up About an hour  0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp
```



В случае успешного запуска станет доступным веб-интерфейс Kibana. Чтобы его открыть, перейдите в браузере по адресу <http://127.0.0.1:5601>

Рисунок 4



## 3.5 Установка Logstash

### 3.5.1 Требования к версии Logstash

Logstash устанавливается также с помощью Docker. Для этого используется официальное зеркало: [https://hub.docker.com/\\_/logstash](https://hub.docker.com/_/logstash).

**Внимание!** Версии устанавливаемых Elasticsearch, Kibana и Logstash должны быть идентичными. Разница в версиях может вызвать проблемы.

### 3.5.2 Скачивание Docker-образа Logstash

Для того, чтобы скачать Logstash с сайта Docker, выполните команду:

```
docker pull logstash:7.1.1
```

Для контроля версий Docker-образов используйте команду:

```
docker images
```

Рисунок 5

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
logstash	7.1.1	b0cb1543380d	2 weeks ago	847MB
kibana	7.1.1	67f17df6ca3e	2 weeks ago	746MB
elasticsearch	7.1.1	b0e9f9f047e6	2 weeks ago	894MB

*Примечание.* В этом примере Kibana использует конфигурацию по умолчанию и подключается к работающему экземпляру Elasticsearch по адресу `http://localhost:9200`.

### 3.5.3 Хранение файлов конфигурации

Для хранения всех конфигураций Logstash создайте соответствующий каталог в каталоге Docker

Рисунок 6

```
$ pwd
/Users/enle/docker
$ ls
filebeat      logstash
```

В каталоге `/docker/logstash` создайте файл конфигурации `logstash.yml`:

```
path.config: /usr/share/logstash/conf.d/*.conf
path.logs: /var/log/logstash
conf.d / test.conf (содержимое файла)

input {
  beats {
    port => 5044
    codec => "json"
  }
}

output {
  elasticsearch { hosts => ["elasticsearch:9200"] }
  stdout { codec => rubydebug }
}
```

*Примечание:* при конфигурировании уточните значения параметров `path.config` и `path.logs` для вашего сервера.

Запустите Logstash:

```
docker run -it -d -p 5044:5044 --name logstash --net somenetwork -v
{path}/logstash.yml:/usr/share/logstash/config/logstash.yml -v
{path}/conf.d/:/usr/share/logstash/conf.d/ logstash:7.1.1
```

Чтобы просмотреть текущее состояние контейнера, выполните команду `docker ps`:

Рисунок 7



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1575e34c782	logstash:7.1.1	"/usr/local/bin/dock..."	5 seconds ago	Up 4 seconds	0.0.0.0:5044->5044/tcp, 9200/tcp	logstash
4eb19c1c38a	store/elastic/filebeat:7.1.1	"/usr/local/bin/dock..."	54 seconds ago	Up 54 seconds		filebeat
2e1d94275e29	kibana:7.1.1	"/usr/local/bin/kiba..."	3 days ago	Up 3 hours	0.0.0.0:5601->5601/tcp	kibana
68cc339576d4	elasticsearch:7.1.1	"/usr/local/bin/dock..."	3 days ago	Up 3 hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp	elasticsearch

## 3.6 Установка Filebeat

### 3.6.1 Требования к версии Filebeat

Filebeat устанавливается также с помощью Docker. Для этого используется официальное зеркало: [https://hub.docker.com/\\_/filebeat](https://hub.docker.com/_/filebeat).

**Внимание!** Версия Filebeat должна быть идентична версия компонентов из Elastic-стека. Разница в версиях может вызвать проблемы.

### 3.6.2 Скачивание Docker-образа Filebeat

Для того, чтобы скачать Logstash с сайта Docker, выполните команду:

```
docker pull store/elastic/filebeat:7.1.1
```

Для контроля версий Docker-образов используйте команду:

```
docker images
```

Рисунок 8

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
logstash	7.1.1	b0cb1543380d	2 weeks ago	847MB
kibana	7.1.1	67f17df6ca3e	2 weeks ago	746MB
elasticsearch	7.1.1	b0e9f9f047e6	2 weeks ago	894MB
store/elastic/filebeat	7.1.1	0bd69a03e199	2 weeks ago	288MB

### 3.6.3 Хранение файлов конфигурации

Для хранения конфигурации Filebeat используется *filebeat.docker.yml*.

В текущем каталоге конфигураций Docker создайте временно папку, в котором будет храниться файл *filebeat.docker.yml*.

Скачайте файл с оф. сайта:

```
curl -L -O
https://raw.githubusercontent.com/elastic/beats/7.1/deploy/docker/filebeat.docker.yml
```

Содержимое файла:

```
filebeat.config:
  modules:
    path: ${path.config}/modules.d/*.yml
    reload.enabled: false

filebeat.autodiscover:
  providers:
    - type: docker
      hints.enabled: true

processors:
- add_cloud_metadata: ~

output.elasticsearch:
  hosts: '${ELASTICSEARCH_HOSTS:elasticsearch:9200}'
  username: '${ELASTICSEARCH_USERNAME:}'
  password: '${ELASTICSEARCH_PASSWORD:}'
```

В каталоге */docker/filebeat* отредактируйте *filebeat.docker.yml*, добавив следующую конфигурацию:

```
vim filebeat.docker.yml
```

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
  - /var/log/nginx/*.log

output.logstash:
  hosts: ['{ip}:5044']
```

Где **{ip}** — адрес Elasticsearch.

### 3.6.4 Запуск Filebeat

Сервис запускается также с помощью Docker. Для запуска выполните команду:

```
docker run --name filebeat --user=root -d --net somenetwork --volume="{nginx-path}:/var/log/nginx/" --
volume="{path}/filebeat.docker.yml:/usr/share/filebeat/filebeat.yml:ro" --
volume="/var/lib/docker/containers:/var/lib/docker/containers:ro" --
volume="/var/run/docker.sock:/var/run/docker.sock:ro"
store/elastic/filebeat:7.1.1
```

**Внимание!** В соответствии с вашими настройками сервера задайте корректные значения для параметров `{nginx-path}`, `{path}`, где:

- `{nginx-path}` — локальный каталог файла журнала (сопоставлен с контейнером в качестве источника данных).
- `{path}` — путь к файлу конфигурации Filebeat.

Чтобы просмотреть текущее состояние контейнера, выполните команду `docker ps`:

Рисунок 9

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8515e24e782	looptash:7.1.1	"/usr/local/bin/dock..."	5 seconds ago	Up 4 seconds	0.0.0.0:5044->5044/tcp, 9000/tcp	looptash
6eb76cfe98b	store/elastic/filebeat:7.1.1	"/usr/local/bin/dock..."	54 seconds ago	Up 52 seconds		filebeat
2a169475629	#kibana:7.1.1	"/usr/local/bin/kiba..."	1 days ago	Up 3 hours	0.0.0.0:5601->5601/tcp	kibana
d8cc39576dd	elasticsearch:7.1.1	"/usr/local/bin/dock..."	1 days ago	Up 3 hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp	elasticsearch

## 4 Общая проверка работоспособности

В случае успешной установки компонент работает сетевое взаимодействие между компонентами. Это означает, что в Kibana будут попадать логи сервисов. Для проверки того, что стек настроен корректно, откройте веб-интерфейс Kibana и проанализируйте имеющиеся данные.