

**Сопроводительная документация по развертыванию
Программы для ЭВМ
«Платформа разработки бизнес-приложений «Взаимодействие»**

На 9 листах

2022

СОДЕРЖАНИЕ

1. Общие положения	4
1.1 Полное наименование Программы для ЭВМ, обозначение	4
1.2 Разработчик Программы для ЭВМ	4
1.3 Назначение документа	4
2. Описание требований	4
2.1 Минимальные аппаратные требований	4
2.2 Программное обеспечение платформы	5
3. Развертывание	6
3.1 Развертывание сервера PostgreSQL 13 на сервере с операционной системой Astra Linux CE 2.12.....	6
3.2 Развертывание кластера Kubernetes 1.18 на серверах с операционной системой Astra Linux CE 2.12.....	6

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Термин, определение, сокращение	Определение
СХД, ЭХ	Система хранения данных, электронное хранилище
СЭД, СЭДО	Система электронного документооборота
API, Application Programming Interface	Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой
Camunda BPMN Modeler	Настольное приложение для моделирования рабочих процессов BPMN и решений DMN
HTTP, HyperText Transfer Protocol	Протокол прикладного уровня передачи данных
HTTPS, HyperText Transfer Protocol Secure	Расширение протокола HTTP, для поддержки шифрования в целях повышения безопасности
Kubernetes	Программное обеспечение с открытым кодом для развертывания контейнеров и управления ими в большом масштабе
MS AD, AD (Microsoft Active Directory)	Службы каталогов корпорации Microsoft для операционных систем семейства Windows Server
MongoDB	Документоориентированная система управления базами данных
PostgreSQL	Свободная объектно-реляционная система управления базами данных (СУБД)
WSDL, Web Services Description Language	Язык описания веб-сервисов и доступа к ним, основанный на языке XML
Astra Linux Common Edition	Операционная система специального назначения на базе ядра Linux, созданная для комплексной защиты информации и построения защищённых автоматизированных систем
XML, eXtensible Markup Language	Расширяемый язык разметки - рекомендованный Консорциумом Всемирной паутины (W3C) язык разметки

1. Общие положения

1.1 Полное наименование Программы для ЭВМ, обозначение

Полное наименование Программы для ЭВМ: Платформа разработки бизнес-приложений «Взаимодействие».

Краткое наименование (обозначение) Программы для ЭВМ: Платформа бизнес-приложений «Взаимодействие».

1.2 Разработчик Программы для ЭВМ

Полное наименование: Общество с ограниченной ответственностью «Философия.ИТ».

Сокращенное наименование: ООО «Философия.ИТ».

1.3 Назначение документа

Настоящий документ входит в комплект эксплуатационной документации по Платформе бизнес-приложений «Взаимодействие» (далее – Система) и описывает функциональные характеристики Системы.

2. Описание требований

2.1 Минимальные аппаратные требований

Таблица 1

Требования к серверам по развертыванию платформы

Сервер	Назначение	Кол-во	vCPU, ядер	RAM, Gb	HDD, Gb	Модель CPU	ОС
Infra	LDAP, DNS сервера	1	2	8	50	Intel® Xeon®	Astra Linux CE 2.12
Gitlab	Сервер хранения конфигураций	1	2	8	100	Intel® Xeon®	Astra Linux CE 2.12
Neo4j	БД (графовая) хранение связей между документами	1	4	16	200	Intel® Xeon®	Astra Linux CE 2.12
MongoDB	БД хранения реквизитов документов и справочников	1	4	16	500	Intel® Xeon®	Astra Linux CE 2.12
Kafka	Брокер сообщений	1	2	8	100	Intel® Xeon®	Astra Linux CE 2.12

Сервер	Назначение	Кол-во	vCPU, ядер	RAM, Gb	HDD, Gb	Модель CPU	ОС
PostgreSQL	БД для отчетности	1	4	16	300	Intel® Xeon®	Astra Linux CE 2.12
MinIO	Система хранения вложений	1	4	8	300*	Intel® Xeon®	Astra Linux CE 2.12
KeyCloak	Система для управления доступом и ролевой моделью	1	2	8	100	Intel® Xeon®	Astra Linux CE 2.12
Jasper	Система генерации отчетности	1	2	8	300	Intel® Xeon®	Astra Linux CE 2.12
K8s	Мастер	1	2	8	100	Intel® Xeon®	Astra Linux CE 2.12
	Инфраструктура	1	2	8	100	Intel® Xeon®	Astra Linux CE 2.12
	Вычислительные	2	8	32	200	Intel® Xeon®	Astra Linux CE 2.12

2.2 Программное обеспечение платформы

В данном разделе в виде таблицы приводится состав комплекса программных средств платформы с отображением версий используемого ПО Системы.

Таблица 2

Состав комплекса средств и версий ПО Системы

№ п.п	Программный компонент	Версия
1	Camunda Open Source Community Edition	7.15
2	Camunda Open Source Modeler	4.9
3	Astra Linux Common Edition	2.12
4	Keycloak	14.00
5	Docker	19.03
6	Neo4j Graph Database	4.3
7	JasperReports® Server Community Edition	7.8.0
8	Jaspersoft Studio	6.17.0
9	MongoDB Community Server	5.0.1
10	GitLab Community Edition	12.07.2
11	Kubernetes	1.21
12	Apache Tomcat	08.05

№ п.п	Программный компонент	Версия
13	Nginx	1.20.1
14	Nodejs	14.17.3
15	Apache Kafka	2.8.0
16	MinIO Community	8.0.0
17	OpenJDK	8
18	PostgreSQL	13.3

3. Развертывание

3.1 Развертывание сервера PostgreSQL 13 на сервере с операционной системой Astra Linux CE 2.12

1. Подготовка сервера

1.1. Установить корректный часовой пояс (например, MSK):

```
sudo timedatectl set-timezone Europe/Moscow
```

1.2. Обновить список репозиторийев и текущие версии установленного ПО:

```
sudo apt-get update && apt-get upgrade
```

1.3. В настройках межсетевого экрана разрешить входящие подключения к порту TCP/5432.

2. Установка PostgreSQL

2.1. Добавить репозиторий PostgreSQL и установить ПО:

```
apt-get install -y apt-transport-https ca-certificates curl software-properties-common gnupg2
```

```
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
sudo add-apt-repository " deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main"
```

```
sudo apt-get install -y postgresql-13
```

2.2. Настроить прослушиваемые интерфейсы. Для этого в файле `/etc/postgresql/13/main/postgresql.conf` необходимо внести изменения в параметре «listen_addresses»:

```
listen_addresses = "*"

```

2.3. Настроить доступ к базам данных. Для этого в файле `/etc/postgresql/13/main/pg_hba.conf` необходимо добавить строку:

```
host all all 0.0.0.0/0 md5

```

Данная директива разрешает доступ любым пользователям к любым базам данных (к которым у пользователя есть доступ) с любых хостов с аутентификацией по паролю. При необходимости можно ужесточить правила доступа, для этого в файле `pg_hba.conf` есть примеры.

2.4. Создать необходимых пользователей и базы данных:

```
sudo su - postgres
createuser -l -P newuser
createdb -E UTF8 -O newuser newdb
exit

```

2.5. Перезапустить сервис PostgreSQL:

```
sudo systemctl restart postgresql

```

Развертывание сервера PostgreSQL завершено.

3.2 Развертывание кластера Kubernetes 1.18 на серверах с операционной системой Astra Linux CE 2.12

1. Подготовка серверов

1.1. Перед установкой необходимо убедиться, что все узлы, предназначенные для работы в кластере, удовлетворяют следующим условиям:

- объем оперативной памяти – не менее 2ГБ;
- количество ядер процессора – не менее 2;
- уникальные IP-адрес, имя хоста, MAC-адрес.
- swapon отключен.

1.2. На мастер узле необходимо в настройках межсетевого экрана разрешить входящие подключения к следующим портам: TCP/6443, TCP/2379-2380, TCP/10250-10252.

1.3. На рабочих узлах необходимо в настройках межсетевого экрана разрешить входящие подключения к следующим портам: TCP/10250, TCP/32000-32767.

1.4. Проверить состояние модуля ядра `br_netfilter` и загрузить его в случае отсутствия в списке загруженных:

```
lsmod | grep br_netfilter
sudo modprobe br_netfilter
```

2.1. Добавить настройки модуля `br_netfilter` в `sysctl`:

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
```

1.5. Установить на всех узлах корректный часовой пояс (например, MSK):

```
sudo timedatectl set-timezone Europe/Moscow
```

1.6. Внести IP-адреса и имена всех узлов кластера в DNS либо в файл `/etc/hosts` на каждом узле.

1.7. Обновить список репозиторийев и текущие версии установленного ПО:

```
sudo apt-get update && apt-get upgrade
```

1.8. Установить Docker CE:

```
apt-get install -y apt-transport-https ca-certificates curl software-properties-common
gnupg2
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

```
sudo apt-get install -y containerd.io=1.2.13-2 docker-ce=5:19.03.11~3-0~ubuntu-
$(lsb_release -cs) docker-ce-cli=5:19.03.11~3-0~ubuntu-$(lsb_release -cs)
```

```
sudo cat > /etc/docker/daemon.json <<EOF
```

```
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
```

```
EOF
```

```
sudo mkdir -p /etc/systemd/system/docker.service.d
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable docker
```

```
sudo systemctl restart docker
```

2. Установка компонентов Kubernetes

2.1. На всех узлах установить пакеты `kubeadm`, `kubelet`, `kubectl` и зафиксировать их версии (чтобы пакеты не были обновлены в будущем при выполнении `apt-get upgrade`):

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

```

cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
sudo apt-get update
sudo apt-get install -y kubeadm=1.18.20-00 kubelet=1.18.20-00 kubectl=1.18.20-00
sudo apt-mark hold kubelet kubeadm kubectl

```

3. Создание кластера Kubernetes

3.1. Для создания кластера необходимо в первую очередь выполнить развертывание мастер-узла кластера. Для этого на мастер-узле необходимо выполнить следующие команды:

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

При наличии ошибок в процессе установки необходимо исправить их и запустить команду `kubeadm init` заново. В случае успеха на экране будет показано сообщение «Your Kubernetes control-plane has initialized successfully!», и сообщение «You can now join any number of machines by running the following on each node as root:», и строка, начинающаяся с «`kubeadm join`». Это команда для добавления в кластер рабочих узлов, ее необходимо скопировать и использовать на остальных узлах.

3.2. Чтобы управлять кластером не от учетной записи `root`, необходимо скопировать файл конфигурации для подключения к кластеру в домашний каталог пользователя:

```

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

```

3.3. Установить сетевой модуль для обеспечения внутрикластерного сетевого взаимодействия:

```
kubectl apply -f
```

<https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml>

3.4. Установить балансировщик MetalLB:

```
kubectl apply -f
```

<https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/namespace.yaml>

```
kubectl apply -f
```

<https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/metallb.yaml>

Для корректной работы балансировщика, необходимо сконфигурировать его, указав диапазон адресов (в примере ниже это 172.17.70.80-172.17.70.80):

```

cat <<EOF | tee metallb-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
    addresses:
    - 172.17.70.80-172.17.70.80
EOF

```

EOF

```
kubectl apply -f metallb-configmap.yaml
```

3.5. Установить Ingress-контроллер:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml
```


3.6. Добавить в кластер рабочие узлы. Для этого необходимо на каждом рабочем узле выполнить команду `kubeadm join` из п. 3.1.

3.7. Проверить состояние кластера и текущий статус узлов (выполнить команду на мастер-узле):

```
kubectl get nodes
```

Команда выводит список узлов с указанием имени хоста (NAME), текущего статуса узла (STATUS), роли узла (ROLE), времени с момента ввода в кластер (AGE) и текущей версии установленных компонентов (VERSION). У всех узлов должен быть статус «Ready», у мастер-узла должна быть указана роль «master», у рабочих узлов роль «<none>».

Развертывание кластера Kubernetes завершено.