«Сервис агрегации и хранения справочных ответов поставщиков транспортного контента»

Инструкция по установке

Оглавление

Перечень сокращений

Термин	Определение
АРМ	Автоматизированное рабочее место
АСУ	Автоматизированная система управления
База данных (БД)	Совокупность данных, организованных в соответствии с концептуальной схемой, описывающей характеристики этих данных и связи между соответствующими им объектами, поддерживающая одну или несколько предметных областей
Доступ к информации (Доступ)	Ознакомление с информацией, ее обработка, в частности, копирование, модификация или уничтожение информации
Пользователь	Лицо, сотрудник Заказчика или организаций-агентов, участвующее в функционировании Системы или использующее результаты ее функционирования
ПО	Программное обеспечение
ПК	Персональный компьютер
Система	АСУ «Сервис агрегации и хранения справочных ответов поставщиков транспортного контента»
СУБД	Система управления БД
Ansible	ПО для удаленного управления конфигурациями
Consul	Распределенная, высокодоступная система обнаружения и конфигурирования сервисов
Elasticsearch	Тиражируемая свободная (лицензия SSPL) программная поисковая система
Grafana	Мультиплатформенное веб-приложение для аналитики и интерактивной визуализации с открытым исходным кодом
Kubernetes	Открытое ПО для автоматизации развёртывания, масштабирования и координации контейнеризированных приложений в условиях кластера
Prometheus	ПО используемое для мониторинга событий и оповещения
Zabbix	Свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования

1 Первичные действия

1.1 Специалисты, необходимые для обеспечения работы системы

Минимальное количество специалистов, необходимое для обеспечения работы системы.

Для работы системы минимально необходимы:

- 1 системный администратор;
- 1 администратор БД.

Системный администратор должен обладать высоким уровнем квалификации в следующих областях:

- установка, настройка и администрирование технических средств и программного обеспечения (серверы, системное ПО);
- разработка, управление и реализация эффективной политики информационной безопасности;
- модернизация программных средств.

Администратор баз данных должны обладать высоким уровнем квалификации в следующих областях:

- администрирование СУБД;
- разработка, управление и реализация эффективной политики информационной безопасности.

2 Действия системного администратора при установке и наладке системы

2.1 Требования к рабочему месту системного администратора

Рабочее место (АРМ) системного администратора:

- стандартный офисный ПК, оснащённый браузерами, Интернетом и снабженный источниками бесперебойного питания 220 вольт;
- операционная система не ниже Windows 7/8/10, Linux с ядром 2.6/3.x;
- наличие непрерывного подключения ПК к АСУ посредством сети TCP/IP, не хуже 256 кбит/с и задержкой (ping) не более 50 мсек.

2.2 Параметры программного обеспечения, необходимые для работы с АСУ

Доступ пользователя (системного администратора) к Системе осуществляется в режиме тонкого клиента, функционирующего в различных операционных средах – Microsoft Windows, Unix (Linux), Mac OS.

Доступ к Системе возможен только с АРМ системного администратора в режиме реального времени.

Для работы с Системой необходимо:

- доступ к сети Интернет;
- браузер с поддержкой HTML 4.0, CSS Level 2, JavaScript 1.1. и выше, режима асинхронного взаимодействия JavaScript/XML (XMLHttpRequest и т.п.). Пользовательские интерфейсы Системы совместимы с браузерами: Microsoft Internet Explorer версии 8.0 или выше, Mozilla FireFox версии 6.0 или выше, Google Chrome версии 10.0 или выше;
- сертификат безопасности.

***Примечание**:*

В настоящий момент используется публичный корневой SSL сертификат, подтверждающий валидность доменного имени, по которому доступно приложение для пользователя из сети Интернет.

Доступ к системе выдаётся по логину и паролю с использованием двойной (двухфакторной) аутентификации пользователя. Доступ системного администратора к функциональным сервисам осуществляется по технологии «тонкого клиента» на базе web-обозревателя (браузера) посредством АРМ.

2.3 Общий вид системы

Общий вид системы представлен ниже (Рисунок 1).

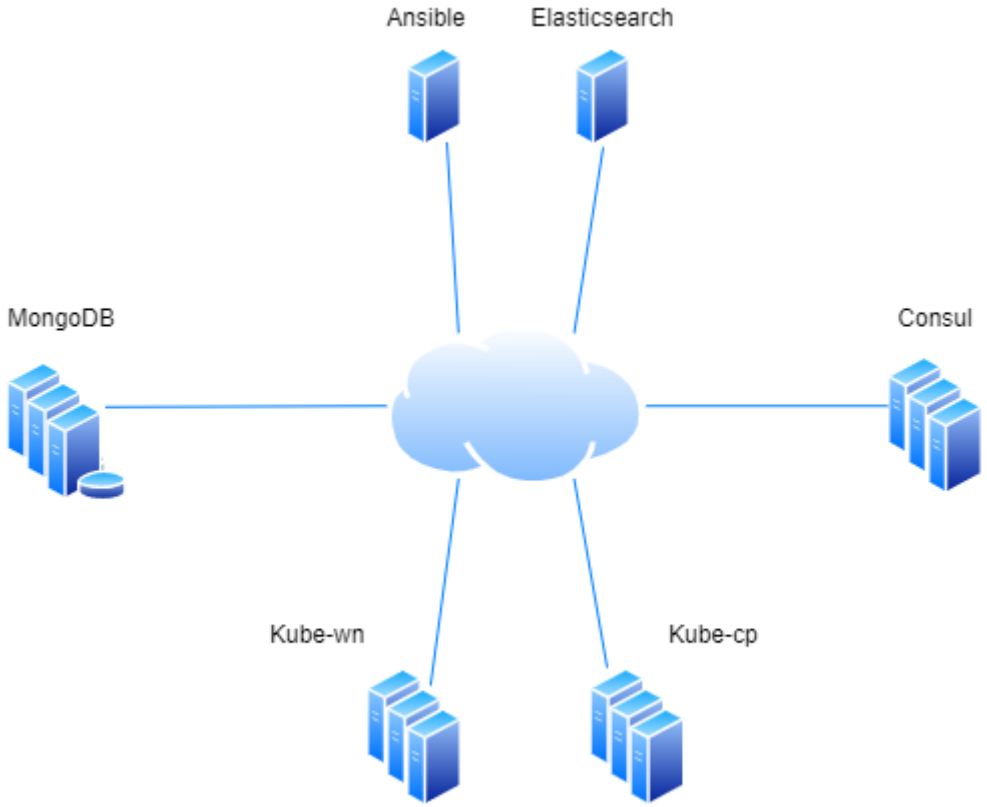

Рисунок 1 – Общий вид системы

Система представляет собой одноранговую сеть из нескольких разнотипных серверов:

- Ansible – сервер системы управления конфигурациями;
- Elasticsearch – сервер поисковой системы на базе платформы аналитики и визуализации Kibana;
- MongoDB – комплект серверов БД;
- Consul – комплект серверов системы обнаружения и конфигурирования сервисов;
- Kube-wn – комплект серверов АСУ (рабочие узлы в кластере Kubernetes);
- Kube-cp – комплект серверов системы управления АСУ (Control Plane кластера Kubernetes).

Рекомендуем состав системы включает в себя по одному серверу Ansible и Elasticsearch, а также комплекты по три сервера MongoDB, Consul, Kube-wn и Kube-cp.

2.4 Требования к серверам системы

Для рассматриваемого варианта системы рекомендуются сервера со следующими техническими характеристиками:

- Consul: (2 CPU / 1 GB RAM / 21 GB Disk)
- Elasticsearch: (4 CPU / 8 GB RAM / 128 GB Disk)
- Kube-cp: (2 CPU / 4 GB RAM / 54 GB Disk)
- Kube-wn: (8 CPU / 16 GB RAM / 120 GB Disk)
- MongoDB: (4 CPU / 8 GB RAM / 520 GB Disk)
- Ansible: (2 CPU / 1 GB RAM / 21 GB Disk)

2.5 Установка MongoDB

MongoDB — это бесплатная БД документов с открытым исходным кодом.

Для установки последней стабильной версии MongoDB на сервере под управлением CentOS требуется выполнить следующие процедуры:

а) Включение репозитория MongoDB

Для добавления репозитория MongoDB в систему, необходимо открыть текстовый редактор и создать новый файл конфигурации репозитория YUM с именем mongodb-org.repo внутри каталога /etc/yum.repos.d/ :

/etc/yum.repos.d/mongodb-org.repo

[mongodb-org-4.0] name = MongoDB Repository baseurl = https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.0/x86_64/ gpgcheck = 1 enabled = 1 gpgkey = https://www.mongodb.org/static/pgp/server-4.0.asc

б) Установка MongoDB

Теперь, когда репозиторий включен, требуется установить мета-пакет mongodb-org с помощью утилиты yum:

sudo yum install mongodb-org

Во время установки yum предложит импортировать GPG-ключ MongoDB. Введите «y» и нажмите клавишу «Enter».

Следующие пакеты будут установлены в системе как часть пакета mongodb-org:

- mongodb-org-server — демон mongod и соответствующие сценарии инициализации и конфигурации.
- mongodb-org-mongos — демон mongos.

- mongodb-org-shell — mongodb-org-shell mongo, интерактивный интерфейс JavaScript для MongoDB, используемый для выполнения административных задач из командной строки.
- mongodb-org-tools — содержит несколько инструментов MongoDB для импорта и экспорта данных, статистики, а также другие утилиты.

в) Запуск MongoDB

После завершения установки требуется запустить демон MongoDB и разрешить ему запускаться при загрузке, набрав:

sudo systemctl start mongod sudo systemctl enable mongod

г) Проверка установки MongoDB

Чтобы проверить установку, требуется подключиться к серверу БД MongoDB с помощью команды mongo, а затем, после загрузки оболочки MongoDB, проверить версию сервера с помощью команды db.version().

Если отображена версия БД (к примеру, 4.0.1), то установка MongoDB прошла успешно.

2.6 Настройка MongoDB

Для настройки MongoDB требуется отредактировать файл конфигурации /etc/mongod.conf, написанный на YAML, и создать одного административного пользователя MongoDB.

В файле требуется раскомментировать раздел безопасности и включить авторизацию:

/etc/mongod.conf

security:

authorization: enabled

Параметр authorization включает управление доступом на основе ролей (RBAC), которое регулирует доступ пользователей к ресурсам и операциям базы данных. Если эта опция отключена, каждый пользователь будет иметь доступ к любой базе данных и сможет выполнять любые действия.

После внесения изменений в файл конфигурации MongoDB требуется перезапустить службу mongod:

sudo systemctl restart mongod

Дополнительная информация о параметрах файла конфигурации доступна на сайте разработчика – https://docs.mongodb.com/manual/reference/configuration-options/.

Для создания административного пользователя MongoDB требуется:

а) Загрузить оболочку MongoDB с помощью команды mongo.

б) В оболочке MongoDB ввести набор команд для подключения к базе данных admin:

use admin

switched to db admin

в) Создать нового пользователя с именем mongoAdmin с ролью userAdminAnyDatabase:

```
db.createUser(
  {
    user: "mongoAdmin",
    pwd: "changeMe",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
  }
)
Successfully added user: {
        "user" : "mongoAdmin",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ]
}
```

г) Выйти из оболочки MongoDB с помощью команды quit()

2.7 Установка и настройка системы управления АСУ

Для работы АСУ потребуется механизм контейнеризации. Рекомендуется использовать docker, как самый популярный.

Настройку требуется выполнить на всех узлах системы управления АСУ (Kube-cp на рисунке 1) для формирования кластера АСУ.

Для создания системы управления АСУ требуется:

а) Предварительная настройка

На серверах системы управления АСУ требуется задать имена хоста и дополнить файлы /etc/hosts, к примеру:

```
# hostnamectl set-hostname master-node1
# cat <<EOF>> /etc/hosts
10.128.0.25 master-node1
```

```
10.128.0.26 master-node2
10.128.0.27 master-node3
10.128.0.29 node-1 worker-node-1
10.128.0.30 node-2 worker-node-2
10.128.0.31 node-3 worker-node-3
EOF
```

Рекомендуется выполнить проверку связи с рабочими узлами, чтобы убедиться в правильности работы обновленного файла хоста с помощью команды ping:

```
# ping 10.128.0.29
# ping 10.128.0.30
# ping 10.128.0.31
```

Затем требуется отключить систему принудительного контроля доступа SElinux и обновить правила брандмауэра.

```
# setenforce 0
# sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
# reboot
```

Заключительным этапом требуется установить правила брандмауэра для портов:

```
# firewall-cmd --permanent --add-port=6443/tcp
# firewall-cmd --permanent --add-port=2379-2380/tcp
# firewall-cmd --permanent --add-port=10250/tcp
# firewall-cmd --permanent --add-port=10251/tcp
# firewall-cmd --permanent --add-port=10252/tcp
# firewall-cmd --permanent --add-port=10255/tcp
# firewall-cmd –reload
# modprobe br_netfilter
# echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

б) Настройка Kubernetes REPO

Требуется добавить хранилище Kubernetes, так как оно не установлено по умолчанию в CentOS 7.

```
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/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

в) Установка KubeAdm и Docker

Для установки пакетов KubeAdm и Docker требуется выполнить следующую команду:

```
# yum install kubeadm docker -y
```

После успешного завершения установки требуется включить и запустить обе службы.

```
# systemctl enable kubelet
# systemctl start kubelet
# systemctl enable docker
# systemctl start docker
```

г) Установка Kubernetes Master и настройка пользователя по умолчанию

Перед инициализацией Kubernetes Master требуется отключить файл подкачки:

```
# swapoff –a
```

Инициализация Kubernetes Master на первом узле системы управления - это полностью автоматизированный процесс, управляемый командой kubeadm init (Kube-cp – доменное имя системы управления, 10.128.0.27 – IP адрес первого узла).

```
# kubeadm init --control-plane-endpoint=Kube-cp --apiserver-advertise-address=10.128.0.27
```

После успешного завершения установки требуется скопировать последнюю строку (далее – маркер) и сохранить ее на внешнем носителе, поскольку нужно будет запустить ее на рабочих узлах. При этом рекомендуется отредактировать ее, убрав символ «\» в конце первой строки:

kubeadm join Kube-cp:6443 --token nu06lu.xrsux0ss0ixtnms5 --discovery-token-ca-cert-hash sha256:f996ea3564e6a07fdea2997a1cf8caeddafd6d4360d606dbc82314688425cd41

Инициализация Kubernetes Master на последующих узлах системы управления - это также полностью автоматизированный процесс, управляемый командой kubeadm join и дополненным маркетом (Kube-cp – доменное имя системы управления, 10.128.0.x – IP адрес узла).

kubeadm join Kube-cp:6443 --apiserver-advertise-address=10.128.0.25
--token nu06lu.xrsux0ss0ixtnms5 --discovery-token-ca-cert-hash
sha256:f996ea3564e6a07fdea2997a1cf8caeddafd6d4360d606dbc82314688425cd41 --control-plane

Для настройки системы управления АСУ требуется:

а) Инициализация Kubernetes Master

После успешной установки Kubernetes необходимо разрешить использование кластера. Рекомендуется запустить эту настройку от имени пользователя root:

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

При необходимости запуска от имени пользователя sudo требуется перейти на пользователя с поддержкой sudo и выполнить следующие действия:

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

Для завершения инициализации рекомендуется проверить статус узла системы управления.

kubectl get nodes

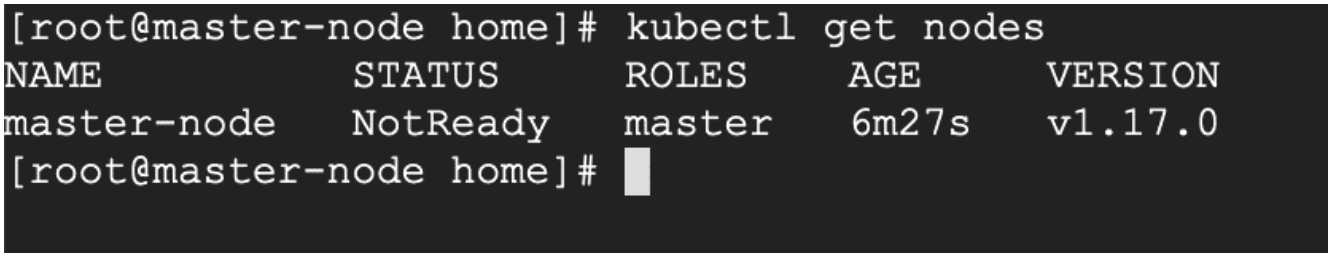

Рисунок 2 – Команда kubectl активирована

На этом этапе система управления имеет статус NotReady. Это связано с тем, что сеть модулей еще не развернута в кластере.

~~Pod Network - это сеть наложения для кластера, которая развернута поверх текущей сети узла. Она предназначена для обеспечения возможности подключения через модуль.~~

б) Настройка сети модуля

Применение сетевого кластера является очень гибким процессом в зависимости от потребностей пользователя и наличия множества доступных вариантов. Для более простой установки ниже описано использование плагина Weavenet, который не требует никакой конфигурации или дополнительного кода, он предоставляет один IP-адрес на модуль:

export kubever=$(kubectl version | base64 | tr -d '\n')
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"

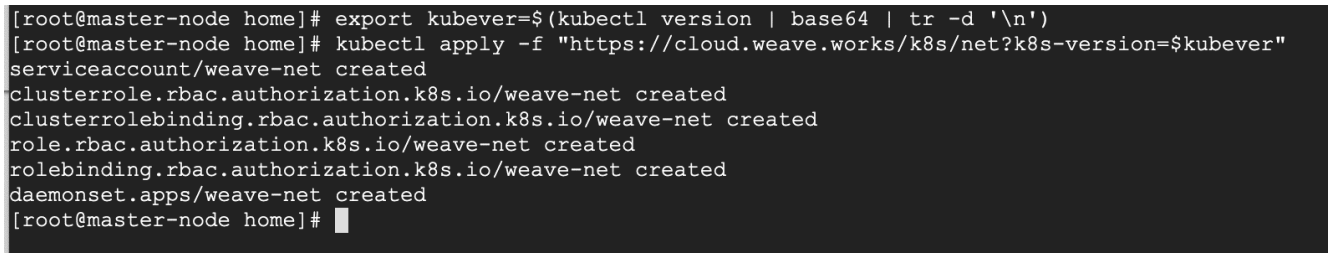

Рисунок 3 – Настройка сети модуля

Для завершения настройки рекомендуется проверить статус узла.

kubectl get nodes

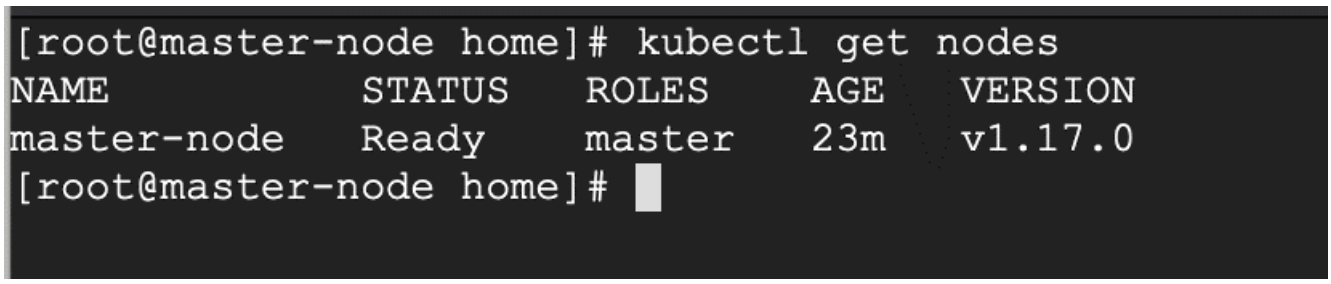

Рисунок 4 – Узел системы управления активирован

2.8 Настройка рабочих узлов АСУ

Настройку требуется выполнить на всех рабочих узлах (Kube-wn на рисунке 1) для присоединения к кластеру АСУ.

а) Предварительная настройка

На рабочих узлах требуется задать имя, а также обновить основные и рабочие узлы в файле /etc/hosts.

```
# hostnamectl set-hostname 'node-1'
# cat <<EOF>> /etc/hosts
10.128.0.25 master-node1
10.128.0.26 master-node2
10.128.0.27 master-node3
10.128.0.29 node-1 worker-node-1
10.128.0.30 node-2 worker-node-2
10.128.0.31 node-3 worker-node-3
EOF
```

Рекомендуется выполнить проверку связи с узлами системы управления, чтобы убедиться в правильности работы обновленного файла хоста с помощью команды ping:

```
# ping 10.128.0.25
# ping 10.128.0.26
# ping 10.128.0.27
```

Затем требуется отключить систему принудительного контроля доступа SElinux и обновить правила брандмауэра.

```
# setenforce 0
# sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
```

Заключительным этапом требуется установить правила брандмауэра для портов:

```
# firewall-cmd --permanent --add-port=6783/tcp
# firewall-cmd --permanent --add-port=10250/tcp
# firewall-cmd --permanent --add-port=10255/tcp
# firewall-cmd --permanent --add-port=30000-32767/tcp
# firewall-cmd  --reload
# echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
```

б) Настройка Kubernetes REPO

Требуется добавить хранилище Kubernetes, так как оно не установлено по умолчанию в CentOS 7.

```
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/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

в) Установка KubeAdm и Docker

Для установки пакетов KubeAdm и Docker требуется выполнить следующую команду:

```
# yum install kubeadm docker -y
```

После успешного завершения установки требуется включить и запустить обе службы.

systemctl enable kubelet

systemctl start kubelet

systemctl enable docker

systemctl start docker

г) Присоединение рабочего узла к кластеру АСУ

Для присоединения к кластеру требуется маркер, созданный kubeadm init на узле системы управления (см. рис. **Error! Reference source not found.**). Его требуется скопировать и вставить во все рабочие узлы.

kubeadm join Kube-cp:6443 --token nu06lu.xrsux0ss0ixtnms5 --discovery-token-ca-cert-hash sha256:f996ea3564e6a07fdea2997a1cf8caeddafd6d4360d606dbc82314688425cd41

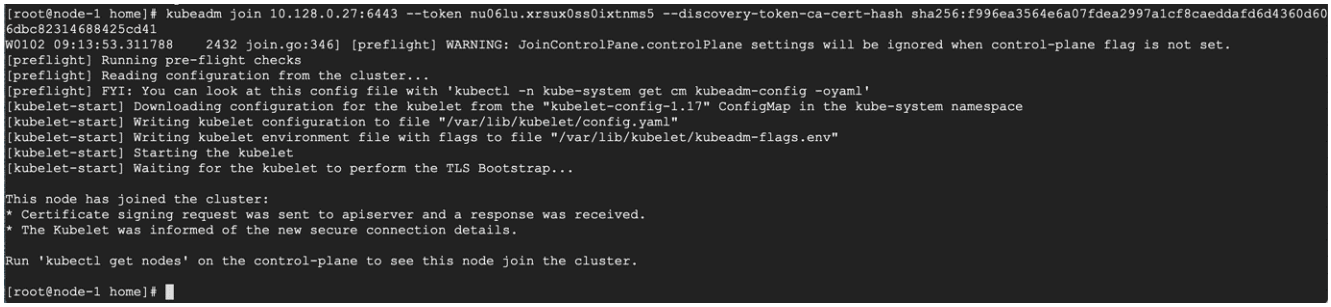

Рисунок 5 – Рабочий узел подключен к кластеру АСУ

После подключения всех рабочих узлов с любого узла системы управления АСУ требуется выполнить команду

kubectl get nodes

При успешной настройке будут показаны все узлы АСУ в состоянии готовности (параметр «STATUS» в значении «Ready»).

2.9 Установка и настройка Consul

Для каждой из машин комплекта необходимо выполнить следующие действия под права администратора root:

а) Установка дополнительных пакетов

Выполнить команду:

yum install unzip wget

где:
unzip — необходим для распаковки архивов.
wget — утилита для загрузки файлов по сети.

б) Настройка времени
Требуется задать часовой пояс:

timedatectl set-timezone Europe/Moscow

П р и м е ч а н и е : допустимо указать любой часовой пояс, но единый на всех серверах Consul.

Требуется установить утилиту для синхронизации времени:

yum install chrony

Затем запустить ее в качестве сервиса:

systemctl enable chronyd --now

в) Настройка имени серверов

Для сервера Consul важна доступность по имени.

hostnamectl set-hostname consul01.dmosk.local

hostnamectl set-hostname consul02.dmosk.local

hostnamectl set-hostname consul03.dmosk.local

Примечание: предполагается, что трем серверам будут заданы имена consul01, consul02 и consul03. Они будут в домене dmosk.local.

Чтобы серверы могли обращаться друг к другу по именам, их надо либо зарегистрировать в локальной системе DNS, либо добавить на каждом сервере следующие записи в файл hosts:

10.128.0.15 consul01.dmosk.local
10.128.0.17 consul02.dmosk.local
10.128.0.19 consul03.dmosk.local

г) Настройка безопасности

Требуется открыть порты в брандмауэре:

firewall-cmd --add-port={8300,8301,8302,8500,8600}/tcp --permanent

firewall-cmd --add-port={8301,8302,8600}/udp --permanent

firewall-cmd --reload

д) Установка и запуск Consul

Установка будет выполнена путем копирования бинарного файла. Для начала требуется перейти на страницу загрузки программного продукта и определить его последнюю версию.

После требуется подставить нужную версию в переменную:

export VER="1.10.2"

Примечание: в описанном примере будет устанавливаться версия 1.10.2.

После требуется загрузить и распаковать требуемый архив в каталог /usr/bin:

```
wget https://releases.hashicorp.com/consul/${VER}/consul_${VER}_linux_amd64.zip
unzip consul_${VER}_linux_amd64.zip
```

Для проверки установки приложения требуется выполнить команду:

```
consul -v
```

На экране будет отображена информация, подобная следующей:

```
Consul v1.10.2
Revision 3cb6eeedb
Protocol 2 spoken by default, understands 2 to 3 (agent will automatically use protocol >2 when speaking to compatible agents)
```

e) Настройка

Требуется создать системную учетную запись, от которой будет работать консул:

```
useradd -r -c 'Consul DCS service' consul
```

Требуется создать каталоги для приложения и определить на них права:

```
mkdir -p /var/lib/consul /etc/consul.d
chown consul:consul /var/lib/consul /etc/consul.d
chmod 775 /var/lib/consul /etc/consul.d
```

П р и м е ч а н и е : в примере указано, что владельцем данных каталогов будет созданная учетная запись consul. Права будут полные у владельца, остальные смогут читать данные.

Требуется сгенерировать ключ для консула на любой из нод кластера:

```
consul keygen
```

zpjf5a4reDbJFpT6FeaF0LGxD0zBRPSRbIoUkLBt0ps=

Полученный ключ требуется сохранить. Он будет использован при настройке всех узлов кластера.

Требуется создать конфигурационный файл для консула:

vi /etc/consul.d/config.json

```
{
    "bind_addr": "0.0.0.0",
    "bootstrap_expect": 3,
    "client_addr": "0.0.0.0",
    "datacenter": "dc1",
    "data_dir": "/var/lib/consul",
    "domain": "consul",
    "enable_script_checks": true,
    "dns_config": {
        "enable_truncate": true,
        "only_passing": true
    },
    "enable_syslog": true,
    "encrypt": "zpjf5a4reDbJFpT6FeaF0LGxD0zBRPSRbIoUkLBt0ps=",
    "leave_on_terminate": true,
    "log_level": "INFO",
    "rejoin_after_leave": true,
    "retry_join": [
        "consul01.dmosk.local",
        "consul02.dmosk.local",
        "consul03.dmosk.local"
    ],
    "server": true,
    "start_join": [
        "consul01.dmosk.local",
```

```
        "consul02.dmosk.local",
        "consul03.dmosk.local"
    ],
   "ui_config": { "enabled": true }
}
```

где:

bind_addr — адрес сервера Consul. Это может быть IP любого из сетевых интерфейсов.

bootstrap_expect — ожидаемое количество серверов в кластере.

client_addr — адрес, к которому будут привязаны клиентские интерфейсы.

datacenter — привязка сервера к конкретному датацентру. Нужен для логического разделения. Серверы с одинаковым датацентром должны находиться в одной локальной сети.

data_dir — каталог для хранения данных.

domain — домен, в котором будет зарегистрирован сервис.

enable_script_checks — разрешает на агенте проверку работоспособности.

dns_config — параметры для настройки DNS.

enable_syslog — разрешение на ведение лога.

encrypt — ключ для шифрования сетевого трафика. В качестве значения требуется использовать сгенерированный ранее.

leave_on_terminate — при получении сигнала на остановку процесса консула, корректно отключать ноду от кластера.

log_level — минимальный уровень события для отображения в логе. Возможны варианты "trace", "debug", "info", "warn", and "err".

rejoin_after_leave — по умолчанию, нода покидающая кластер не присоединяется к нему автоматически. Данная опция позволяет управлять данным поведением.

retry_join — узлы, к которым можно присоединить кластер. Процесс будет повторяться, пока не завершится успешно.

server — режим работы сервера.

start_join — список узлов кластера, к которым пробуем присоединиться при загрузке сервера.

ui_config — конфигурация для графического веб-интерфейса.

Проверку корректности конфигурационного файла можно осуществить выполнением команды:

consul validate /etc/consul.d/config.json

В случае успешной проверки на экране отобразится надпись:

...

Configuration is valid!

В завершение настройки рекомендуется создать командный файл в systemd для возможности автоматического запуска сервиса:

vi /etc/systemd/system/consul.service

```
[Unit]
Description=Consul Service Discovery Agent
Documentation=https://www.consul.io/
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=consul
Group=consul
ExecStart=/usr/bin/consul agent \
   -node=consul01.dmosk.local \
   -config-dir=/etc/consul.d
ExecReload=/bin/kill -HUP $MAINPID
KillSignal=SIGINT
TimeoutStopSec=5
Restart=on-failure
SyslogIdentifier=consul

[Install]
WantedBy=multi-user.target
```

В указанном примере указан сервер consul01.dmosk.local. Для каждого из настраиваемых серверов это имя сервера соответственно меняется. В результате будет сформировано три файла с разными значениями для опции node.

Далее требуется перезагрузить конфигурацию systemd:

```
systemctl daemon-reload
```

Система установлена, настроена и готова к запуску.

ж) Запуск и проверка

Запуск сервиса осуществляется командой:

```
systemctl start consul
```

Также рекомендуетмся разрешить автоматический старт при запуске сервера:

```
systemctl enable consul
```

Текущее состояние работы сервиса можно узнать командой:

```
systemctl status consul
```

В нормальном режиме отображается состояние:

```
...
Active: active (running) since ...
...
```

Состояние нод кластера можно узнать командой:

```
consul members
```

Дополнительную информацию можно отобразить командой:

consul members -detailed

Также доступен веб-интерфейс по адресу:
http://<IP-адрес любого сервера консул>:8500.
На странице по умолчанию отображается статус кластера.

з) Добавление аутентификации

При входе на веб-интерфейс или работе из командной строки, система позволяет выполнять операции без запроса идентификационных данных. Для защиты системы требуется настроить ACL и обновить сервис.

В конфигурационных файлах на всех узлах кластера Consul требуется добавить в файл /etc/consul.d/config.json следующие строки:

```
{
  ...
    "ui_config": { "enabled": true },
    "acl": {
        "enabled": true,
        "default_policy": "deny",
        "enable_token_persistence": true
    }
}
```

П р и м е ч а н и е : Добавленная настройка acl запрещает по умолчанию доступ и требует ввода токена безопасности.

Проверка корректности внесенных настроек выполняется командой:

consul validate /etc/consul.d/config.json

Если ошибок нет, то требуется перезапустить сервис:

systemctl restart consul

Для получения токена необходимо выполнить команду:

consul acl bootstrap

Примечание: если команда нам вернула ошибку Failed ACL bootstrapping: Unexpected response code: 500 (The ACL system is currently in legacy mode.), то значит, что мы не на всех нодах применили новую настройку. Проверяем, на всех ли серверах кластера внесены соответствующие изменения по добавлению ACL.

На экране будет отображен ответ следующего типа:

AccessorID: af5eaac1-4f0b-d46a-58ba-64ec857dfc4c
SecretID: 59ac7fa8-dca6-e066-ff33-0bf9bb6f466a
Description: Bootstrap Token (Global Management)
Local: false
Create Time: 2021-09-01 16:28:58.710545082 +0300 MSK
Policies:
00000000-0000-0000-0000-000000000001 - global-management

В данном примере 59ac7fa8-dca6-e066-ff33-0bf9bb6f466a — нужный для аутентификации в веб-интерфейсе SecretID.

Для работы из командной строки теперь также необходимо авторизоваться. Для упрощения авторизации рекомендуется создать переменную в системном окружении:

export CONSUL_HTTP_TOKEN=59ac7fa8-dca6-e066-ff33-0bf9bb6f466a

где 59ac7fa8-dca6-e066-ff33-0bf9bb6f466a – полученный SecretID.

Создать новый токен можно командой:

consul acl token create -policy-name global-management

Чтобы увидеть список токенов, требуется ввести:

consul acl token list

Удалить токен можно по его AccessorID:

consul acl token delete -id 54b5f2bb-1a57-3884-f0ea-1284f84186f5

П р и м е ч а н и е : в примере будет удален токен с AccessorID 54b5f2bb-1a57-3884-f0ea-1284f84186f5.

и) Настройка политики для DNS

После включения ACL система перестанет отвечать на запросы DNS. Это связано с политикой блокировки по умолчанию. Для разрешения запросов требуется создать политику с разрешением данных запросов, получить для нее токен и применить данный токен на всех серверах кластера Consul.

На одном из серверов требуется создать файл с политикой:

cd /etc/consul.d/

vi dns-request-policy.txt

```
node_prefix "" {
  policy = "read"
}
service_prefix "" {
  policy = "read"
}
```

Далее требуется создать политику в консуле, а затем токен безопасности на основе политики:

consul acl policy create -name "dns-requests" -rules @dns-request-policy.txt

consul acl token create -description "Token for DNS Requests" -policy-name dns-requests

На экране будет отображен ответ следующего типа:

AccessorID: b53741e2-7663-eag6-fd67-a64dbd32feb5
SecretID: 42bd65e5-42a5-356b-a81b-60eff20f657
Description: Token for DNS Requests
Local: false
Create Time: 2021-09-02 12:37:42.342511655 +0300 MSK
Policies:
89e42b6b-bbec-5263-bc7b-60f3a604a0d6 - dns-requests

* где SecretID — это нужный нам токен.

На всех серверах Consul выполняем команду:

export CONSUL_HTTP_TOKEN=59ac7fa8-dca6-e066-ff33-0bf9bb6f466a

* где 59ac7fa8-dca6-e066-ff33-0bf9bb6f466a — ранее созданный токен для получения доступа к управлению консулом.

А затем:

consul acl set-agent-token default 42bd65e5-42a5-356b-a81b-60eff20f657

* где 42bd65e5-42a5-356b-a81b-60eff20f657 — SecretID, полученный на основе политики, разрешающей запросы DNS.

2.10 Установка и настройка Elasticsearch

Пакет Elasticsearch используется для хранения, анализа, поиска информации по логам.

а) Установка Elasticsearch
Предварительно необходимо скопировать публичный ключ репозитория:
rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

Затем требуется подключить репозиторий Elasticsearch:

mcedit /etc/yum.repos.d/elasticsearch.repo

[elasticsearch]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=0
autorefresh=1
type=rpm-md

Установка пакета Elasticsearch осуществляется командой:

yum install --enablerepo=elasticsearch elasticsearch

В завершении установки требуется добавить Elasticsearch в автозагрузку и запустить его с дефолтными настройками:

systemctl daemon-reload
systemctl enable elasticsearch.service
systemctl start elasticsearch.service

Проверку работы службы можно осуществить командой:
systemctl status elasticsearch.service

б) Настройка Elasticsearch

Настройки Elasticsearch находятся в файле /etc/elasticsearch/elasticsearch.yml. Интерес представляют следующие параметры:

path.data: /var/lib/elasticsearch # директория для хранения данных;
network.host: 127.0.0.1 # слушаем только локальный интерфейс.

По умолчанию Elasticsearch слушает localhost. Так как необходимо, чтобы Elasticsearch слушал все сетевые интерфейсы, то требуется изменить параметр network.host:

network.host: 0.0.0.0

Директории с данными будут занимать значительное место, поэтому следует продумать их размещение заранее.

К списку параметров в файле /etc/elasticsearch/elasticsearch.yml требуется добавить:

discovery.seed_hosts: ["127.0.0.1", "[::1]"]

Параметр указывает, что хосты кластера следует искать только локально.

Для применения измененных настроек требуется перезапустить службу:

systemctl restart elasticsearch.service

2.11 Установка и настройка Kibana

Пакет Kibana используется для визуализации данных, полученных из Elasticsearch. Репозитории и публичный ключ для установки Kibana будут такими же, как в установке Elasticsearch.

а) Установка Kibana

Предварительно необходимо скопировать публичный ключ репозитория:

rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

Затем требуется подключить репозиторий Kibana:

mcedit /etc/yum.repos.d/kibana.repo

```
[kibana-7.x]
name=Kibana repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

Установка пакета Kibana осуществляется командой:

yum install kibana

В завершении установки требуется добавить Elasticsearch в автозагрузку и запустить его с дефолтными настройками:

systemctl daemon-reload

systemctl enable kibana.service

systemctl start kibana.service

Проверку работы службы можно осуществить командой:

systemctl status kibana.service

По умолчанию, Kibana слушает порт 5601.

Внимание! Кибана стартует длительное время. Подождите примерно минуту и проверяйте.

Проверку работы Kibana можно осуществить командой:

netstat -tulnp | grep 5601

На экране отобразится информация, подобная следующей:

tcp 0 0 127.0.0.1:5601 0.0.0.0:* LISTEN 20746/node

б) Настройка Kibana

Файл с настройками Kibana располагается по пути - /etc/kibana/kibana.yml.

Для разрешения прослушки внешнего интерфейса и внешних подключений требуется изменить параметр server.host, указав IP адрес сервера:

server.host: "10.128.0.29"

Для прослушки всех интерфейсов требуется указать в качестве адреса 0.0.0.0.

После изменения настроек Kibana сервис требуется перезапустить:

systemctl restart kibana.service

Теперь можно зайти в веб интерфейс по адресу http://10.128.0.29:5601.

2.12 Установка и настройка Ansible

Пакет Ansible используется для централизованного управления остальными машинами системы.

а) Установка Ansible

Предварительно необходимо установить репозиторий EPEL командой:

sudo yum install epel-release

Затем требуется установить Ansible командой:

sudo yum install ansible

б) Настройка хостов Ansible

Ansible взаимодействует с серверами по SSH. При этом система не должна запрашивать паролей. Потому нужно настроить аутентификацию на основе SSH-ключей.

Чтобы получить доступ к серверам системы с сервера Ansible, следует ввести команду:

ssh root@your_server_ip

Ansible отслеживает все серверы, указанные в файле hosts. Чтобы система Ansible могла взаимодействовать с остальными машинами, нужно соответственно отредактировать этот файл, к примеру в текстовом редакторе sudo:

sudo vi /etc/ansible/hosts

В файле хранится множество настроек и примеров конфигурации. Все они закомментированы. Не стоит удалять из файла примеры конфигураций, поскольку они могут пригодиться в дальнейшей работе.

Для добавления серверов системы следует использовать такой синтаксис:

[group_name] alias ansible_ssh_host=your_server_ip

Тег group_name позволяет ссылаться сразу на несколько серверов; alias задаёт имя сервера.

Серверы можно разделить на несколько групп, каждая из которых будет использовать разные параметры настроек.

В файле /etc/ansible/group_vars/all можно указать параметры конфигурации для каждого сервера независимо от группы. Индивидуальные настройки хостов можно поместить в каталог /etc/ansible/host_vars.

Для проверки подключения серверов системы к серверу Ansible можно осуществить командой ping:

ansible -m ping all

При успешной проверке на экране будет отображено:

host1 | SUCCESS => {

"changed": false,

"ping": "…"

}

host3 | SUCCESS => {

"changed": false,

"ping": "…"

}

host2 | SUCCESS => {

"changed": false,

"ping": "…"

}

3 Эксплуатация, техническое обслуживание, ремонт и хранение компонентов системы

Специального регламентного обслуживания Системы не требует.

Контроль над АСУ осуществляется штатными средствами администрирования Kubernetes. При необходимости расширенного мониторинга системы рекомендуется использовать систему мониторинга Zabbix, а также её плагин *powa*, позволяющий собирать статистические данные для облегчения аналитических задач.

Zabbix – свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования. Рекомендуется к установке в системе на отдельный сервер с расположением клиентского ПО, работающего в постоянном фоновом режиме, на каждом сервере АСУ.

Контроль за изменением БД осуществляет администратор БД с помощью одной из систем мониторинга. Рекомендуется использовать систему мониторинга Prometheus.

Prometheus – система мониторинга серверов и программ с открытым исходным кодом. Prometheus работает по принципу сбора требуемой информации с клиентских серверов и устройств, обращаясь к целевым объектам при помощи языка PromQL. Сервер Prometheus считывает параметры целевых объектов с интервалами, которые задаёт пользователь. Данные от целевых объектов передаются на сервер Prometheus в формате http и хранятся в базе данных временных рядов. Серверная часть (Prometheus-server) рекомендуется к установке в системе на один из серверов Kube-wn с расположением клиентского ПО (Exporter), работающего по запросу сервера Prometheus, на каждом сервере БД.

Для просмотра данных системы мониторинга Prometheus рекомендуется к установке программа Grafana — платформа с открытым исходным кодом для визуализации, мониторинга и анализа данных. Grafana позволяет создавать дашборды с панелями, каждая из которых отображает определенные показатели в течение установленного периода времени. Каждый дашборд универсален, его можно настроить для конкретной задачи с учетом любых потребностей.

4 Действия при возникновении ошибок и неполадок

В случае возникновения любой критической ошибки при использовании системы требуется:

- ввести систему в аварийный режим;
- связаться с разработчиком.

Критической ошибкой в Системе является любая невозможность использования основного функционала.