Визитка

СЕРГЕЙ ГОЛОВАШОВ,
ведущий инженер DevOps Bell Integrator

Визитка

НИКОЛАЙ СИТНИКОВ,
инженер DevOps Bell Integrator

Openshift и все вокруг него
Часть 4: Кластеры

Продолжение серии статей про Openshift, его администрирование и работу в нем. В данной статье будет рассмотрено управление кластером Openshift, добавление и удаление нод в состав кластера, методы управления узлами, логирование и журналирование, а также обновление узлов.

OpenShift использует два метода установки для настройки кластера OpenShift.

- Метод быстрой установки
- Расширенный метод настройки

Метод быстрой установки

Этот метод используется для запуска быстрой настройки конфигурации незавершенного кластера. Чтобы использовать его, нам нужно сначала установить установщик. Это можно сделать так:

Интерактивный метод

```
$ atomic-openshift-installer install
```

Это полезно, когда вы хотите запустить интерактивную установку.

Метод автоматической установки

Этот метод используется, когда требуется установить метод автоматической установки, при котором пользователь может определить файл конфигурации yaml и поместить его в **~ / .config / openshift /** с именем **installer.cfg.yml**. Затем можно выполнить следующую команду для установки тега **–u**:

```
$ atomic-openshift-installer -u install
```

По умолчанию он использует файл конфигурации, расположенный в **~ / .config / openshift /** . Ansible, с другой стороны, используется в качестве резервной копии установки.

```
version: v2
variant: openshift-enterprise
variant_version: 3.1
ansible_log_path: /tmp/ansible.log

deployment:
   ansible_ssh_user: root
   hosts:
   - ip: 172.10.10.1
   hostname: vklnld908.int.example.com
   public_ip: 24.222.0.1
   public_hostname: master.example.com
   roles:
      - master
      - node
   containerized: true
   connect_to: 24.222.0.1

   - ip: 172.10.10.2
   hostname: vklnld1446.int.example.com
   public_ip: 24.222.0.2
   public_hostname: node1.example.com
   roles:
      - node
   connect_to: 10.0.0.2

   - ip: 172.10.10.3
   hostname: vklnld1447.int.example.com
   public_ip: 10..22.2.3
   public_hostname: node2.example.com
   roles:
      - node
   connect_to: 10.0.0.3

roles:
   master:
      <variable_name1>: "<value1>"
      <variable_name2>: "<value2>"
   node:
      <variable_name1>: "<value1>"
```

Здесь у нас есть ролевая переменная, которую можно определить, если вы хотите установить какую-то конкретную переменную. После этого мы можем проверить установку с помощью следующей команды:

```
$ oc get nodes
NAME                  STATUS  AGE
master.example.com     Ready   10d
node1.example.com     Ready     10d
node2.example.com     Ready     10d
```

Расширенная установка

Расширенная установка полностью основана на конфигурации Ansible, в которой присутствует полная конфигурация хоста и определение переменных, относящихся к конфигурации мастера и узла. Она содержит все детали, касающиеся

конфигурации. Как только у нас есть настройки, и книга воспроизведения готова, мы можем просто запустить следующую команду для настройки кластера:

```
$ ansible-playbook -i inventry/hosts ~/openshift-ansible/
playbooks/byo/config.yml
```

Добавление хостов в кластер

Мы можем добавить хост в кластер, используя инструмент быстрой установки или расширенный метод настройки.

Инструмент быстрой установки работает как в интерактивном, так и в неинтерактивном режиме. Используйте следующую команду:

```
$ atomic-openshift-installer -u -c </path/to/file> scaleup
```

Формат масштабирования внешнего вида конфигурационного файла приложения можно использовать для добавления как главного, так и узла.

Расширенный метод настройки

В этом методе мы обновляем файл хоста Ansible, а затем добавляем новый узел или сведения о сервере в этот файл. Файл конфигурации выглядит следующим образом.

```
[OSEv3:children]
masters
nodes
new_nodes
new_master
```

В том же файле Ansible hosts добавьте переменные детали относительно нового узла, как показано ниже:

```
[new_nodes]
vklnld1448.int.example.com openshift_node_labels =
"{'region': 'primary', 'zone': 'east'}"
```

Наконец, используя обновленный файл хоста, запустите новую конфигурацию и вызовите файл конфигурации, чтобы выполнить настройку, используя следующую команду:

```
$ ansible-playbook -i /inventory/hosts /usr/share/ansible/
openshift-ansible/playbooks/test/openshift-node/scaleup.yml
```

Управление кластерными журналами

Журнал кластера OpenShift — это не что иное, как журналы, генерируемые с главного и узлового компьютеров кластера. Они могут управлять любым видом журнала, начиная с журнала сервера, главного журнала, журнала контейнера, журнала модуля и т.д. Для управления журналом контейнера имеется несколько технологий и приложений. Некоторые из них перечислены ниже:

- Fluentd
- ELK
- Kabna
- Nagios
- Splunk

Стек ELK – полезен при попытке собрать журналы со всех узлов и представить их в систематическом формате. Стек ELK в основном делится на три основные категории.

ElasticSearch — ответственен за сбор информации из всех контейнеров и размещение ее в центральном месте.

Fluentd – используется для подачи собранных бревен в двигатель контейнера поиска эластичного материала.

Kibana – графический интерфейс, используемый для представления собранных данных в виде полезной информации в графическом интерфейсе.

Следует отметить один ключевой момент: когда эта система развернута в кластере, она начинает собирать журналы со всех узлов.

Лог-диагностика

OpenShift имеет встроенную команду oc adm dignostics с OC, которую можно использовать для анализа множественных ошибок. Этот инструмент можно использовать от мастера в качестве администратора кластера. Эта утилита очень полезна для устранения неполадок и выявления известных проблем. Это работает на главном клиенте и узлах. При запуске без каких-либо агентов или флагов, он будет искать файлы конфигурации клиента, сервера и узлов и использовать их для диагностики. Диагностику можно запустить индивидуально, передав следующие аргументы:

- AggregatedLogging
- AnalyzeLogs
- ClusterRegistry
- ClusterRoleBindings
- ClusterRoles
- ClusterRouter
- ConfigContexts
- DiagnosticPod
- MasterConfigCheck
- MasterNode
- MetricsApiProxy
- NetworkCheck
- NodeConfigCheck
- NodeDefinitions
- ServiceExternalIPs
- UnitStatus

Можно просто запустить их с помощью следующей команды:

```
$ oc adm diagnostics <DiagnosticName>
```

Обновление кластера

Обновление кластера включает в себя обновление нескольких компонентов внутри кластера и получение кластера, обновленного новыми компонентами и обновлениями. Это включает в себя:

- обновление основных компонентов
- обновление узловых компонентов
- обновление политик
- модернизация маршрутов
- обновление потока изображений

Для того, чтобы выполнить все эти обновления, нам нужно сначала получить быстрые установщики или утилиты на месте. Для этого нам нужно обновить следующие утилиты:

- атомно-OpenShift-Utils
- атомно-OpenShift-Excluder

> атомно-OpenShift-докер-Excluder
> пакет etcd

Перед началом обновления нам нужно сделать резервную копию etcd на главном компьютере, что можно сделать с помощью следующих команд:

```
$ ETCD_DATA_DIR = /var/lib/origin/openshift.local.etcd
$ etcdctl backup \
   --data-dir $ETCD_DATA_DIR \
   --backup-dir $ETCD_DATA_DIR.bak. <date>
```

Обновление основных компонентов

В OpenShift master мы запускаем обновление, обновляя файл etcd, а затем переходим к Docker. Наконец, мы запускаем автоматический исполнитель, чтобы получить кластер в нужном положении. Однако перед началом обновления нам нужно сначала активировать атомарные пакеты openshift на каждом из мастеров. Это можно сделать с помощью следующих команд:

Шаг 1 — Удалить пакеты atomic-openshift.

```
$ atomic-openshift-excluder unexclude
```

Шаг 2 — Обновить etcd на всех мастерах.

```
$ yum update etcd
```

Шаг 3 — Перезапустить службу etcd и проверить, успешно ли она началась.

```
$ systemctl restart etcd
$ journalctl -r -u etcd
```

Шаг 4 — Обновить пакет Docker.

```
$ yum update docker
```

Шаг 5 — Перезапустить сервис Docker и проверить, правильно ли он работает.

```
$ systemctl restart docker
$ journalctl -r -u docker
```

Шаг 6 — После завершения перезагрузить систему.

```
$ systemctl reboot
$ journalctl -r -u docker
```

Шаг 7. Наконец, запустить atomic-execute, чтобы вернуть пакеты в список исключений yum.

```
$ atomic-openshift-excluder exclude
```

Нет такого принуждения для обновления политики, его нужно обновить, только если рекомендуется, что можно проверить с помощью следующей команды:

```
$ oadm policy reconcile-cluster-roles
```

В большинстве случаев нам не нужно обновлять определение политики.

Обновление узловых компонентов

После завершения основного обновления мы можем приступить к обновлению узлов. Следует иметь в виду, что период обновления должен быть коротким, чтобы избежать каких-либо проблем в кластере.

Шаг 1 — Удалите все атомарные пакеты OpenShift со всех узлов, на которых вы хотите выполнить обновление.

```
$ atomic-openshift-excluder unexclude
```

Шаг 2 — Затем отключите планирование узлов перед обновлением.

```
$ oadm manage-node <node name> --schedulable = false
```

Шаг 3 — Реплицируйте все узлы с текущего хоста на другой хост.

```
$ oadm drain <node name> --force --delete-local-data
--ignore-daemonsets
```

Шаг 4 — Обновите настройку Docker на хосте.

```
$ yum update docker
```

Шаг 5 — Перезапустите службу Docker, а затем запустите узел службы Docker.

```
$systemctl restart docker
$ systemctl restart atomic-openshift-node
```

Шаг 6 — Проверьте, правильно ли они запущены.

```
$ journalctl -r -u atomic-openshift-node
```

Шаг 7 — После завершения обновления перезагрузите узел компьютера.

```
$ systemctl reboot
$ journalctl -r -u docker
```

Шаг 8 — Снова включите планирование на узлах.

```
$ oadm manage-node <node> --schedulable.
```

Шаг 9 — Запустите исполнитель atomic-openshift, чтобы вернуть пакет OpenShift на узел.

```
$ atomic-openshift-excluder exclude
```

Шаг 10 — Наконец, проверьте, все ли узлы доступны.

• • •

В следующей статье мы перейдем внутрь самого кластера и поговорим про горизонтальное и вертикальное масштабирование, которое реализовано в Openshift на базе Кубернетиса. metric-server'а. EOF

Ключевые слова: Управление кластером, добавление и удаление воркеров и управление этой группой серверов