



Kubernetes 実践入門

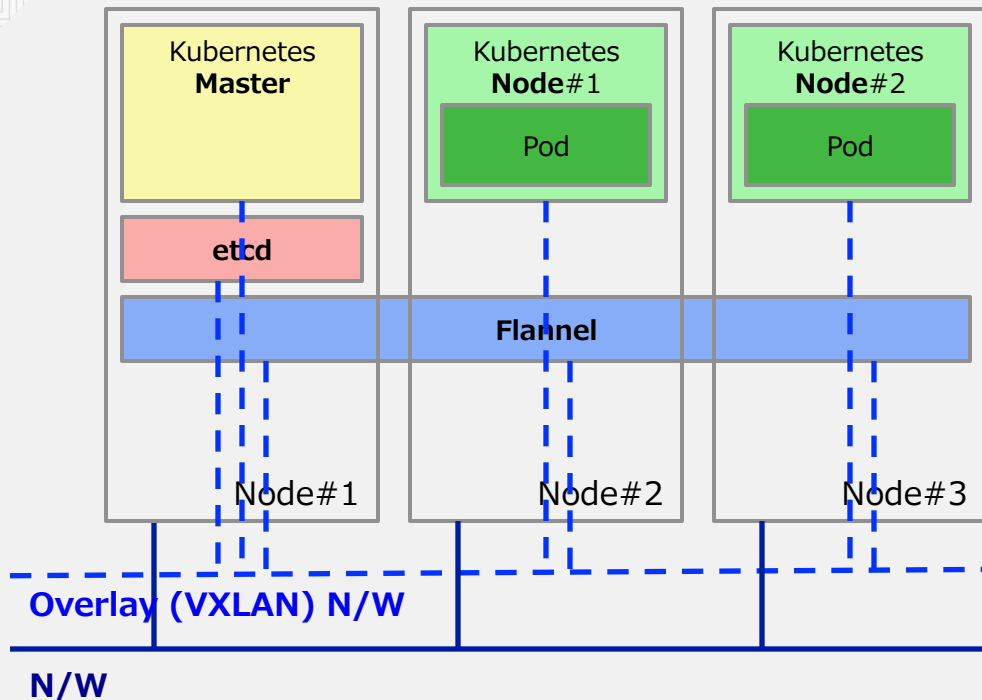
2016/11/09 ver.1.0

Yoshikazu YAMADA <yyamada@redhat.com>
Red Hat K.K. DevOps Lead Senior Architect

Kubernetes クラスタの構築

1. クラスタの構成
2. H/W, N/W, S/W 環境
3. 環境準備
4. データストア (etcd) の構築
5. Docker の構築
6. Overlay N/W の構築
7. Kubernetes Master の構築
8. Kubernetes Node の構築
9. サンプルコンテナの起動

クラスタの構成



Node#1

Kubernetes Master + etcd (Kubernetes データストア) が稼働する。

Flannel はオプションだが、Kubernetes UI を使用する場合は必要となる。

Node#2, Node#3

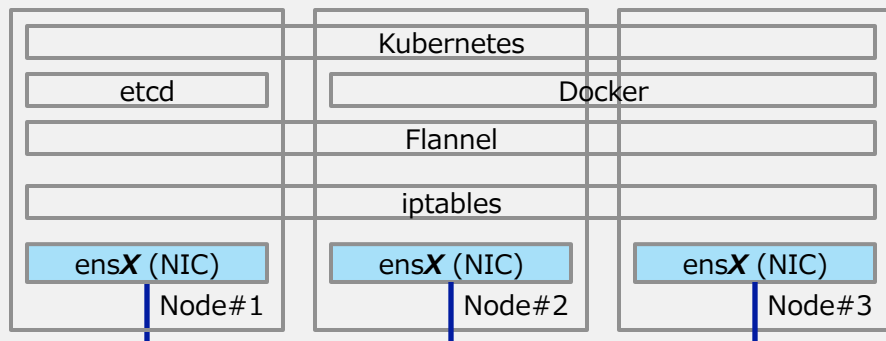
Kubernetes Node + Flannel が稼働する。

その他

- プライベートレジストリの構築は行わない。
- 追加 Node を用意し etcd を同ノードで稼働させる場合も手順は基本的に同じ。

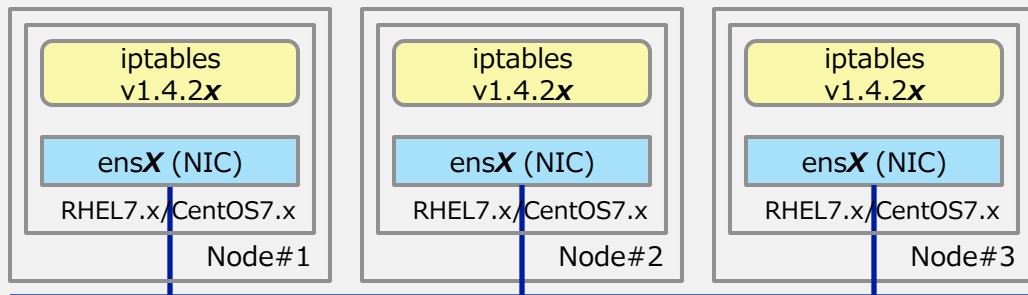
H/W, N/W, S/W 環境

	Node#1	Node#2	Node#3
H/W Spec	CPU 1 Core 以上		
	Memory 2 GB 以上		
OS (Host Linux)	<ul style="list-style-type: none"> Red Hat Enterprise Linux 7 以降 ※ 下記 ※1, ※2 の yum リポジトリ設定が必要 CentOS 7 以降 		
S/W	<ul style="list-style-type: none"> Kubernetes Flannel (※ オプション) etcd 2.3.7 以上 iptables 1.4.21 以上 	<ul style="list-style-type: none"> Kubernetes Flannel Docker iptables 1.4.21 以上 	



※ 1 subscription-manager repos --enable=rhel-7-server-extras-rpms
 ※ 2 subscription-manager repos --enable=rhel-7-server-optional-rpms

環境準備



1. パッケージのアップデート

```
$ sudo yum update -y
```

2. iptables の確認

```
$ iptables -version  
iptables v1.4.21
```

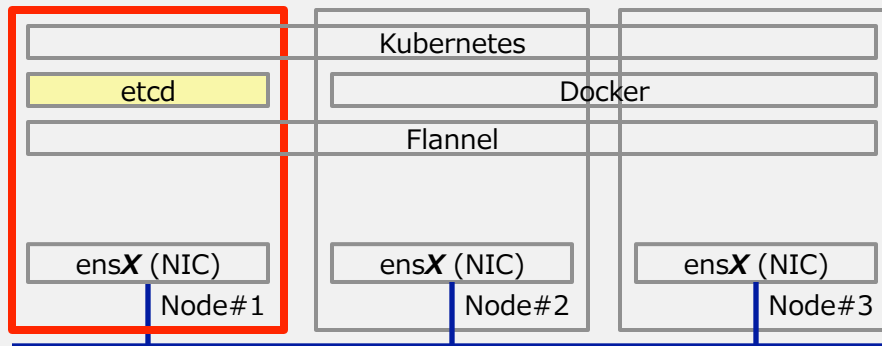
3. ネットワークインターフェースの確認

```
$ ip addr && nmcli
```

4. 1. ~ 3. の手順について node#1 ~ node#3 で実施。

データストア (etcd) の構築概要

1. etcd のインストール と 動作確認
2. 自動起動設定
3. コンフィギュレーション



etcd のインストール と 動作確認

1. etcd のインストール

```
$ sudo yum install -y etcd
```

2. インストールの確認

```
$ etcd --version  
etcd Version: 2.3.7  
Git SHA: fd17c91  
Go Version: go1.6.2  
Go OS/Arch: linux/amd64
```

3. etcd の動作確認

1. etcd の起動

```
$ etcd --name sample-etcd --data-dir /tmp/sample.etcd &
```

2. etcd の動作確認 (プロセス、ポート番号) ※ etcd プロセスが TCP 2379 ポートを listen していることを確認

```
$ ss -natup | grep etcd
```

3. etcdctl による etcd の動作確認

```
$ etcdctl set key0 value0 && etcdctl get key0  
value0  
value0
```

4. curl による REST API 経由での etcd の動作確認

```
$ curl -L http://localhost:2379/v2/keys/key0  
{  
  "action": "get",  
  "node": {  
    "key": "/key0",  
    "value": "value0",  
    "modifiedIndex": 4,  
    "createdIndex": 4  
  }  
}
```

5. etcd プロセスの停止

```
$ ps -ef | grep etcd | grep -v grep | awk '{print $2}' | xargs kill
```

etcd の 自動起動設定

1. Unit 設定ファイル (systemd) の作成 – 以下の URL からファイルを取得

```
https://github.com/yyamada-redhat/training\_kubernetes/blob/master/env/etcd.service
```

2. Unit 設定ファイルの配置

```
$ sudo cp etcd.service /usr/lib/systemd/system/
```

3. 自動起動の有効化

```
$ sudo systemctl enable etcd
```

4. etcd の起動

```
$ sudo systemctl start etcd.service
```

5. etcd プロセスの確認

```
$ systemctl status etcd.service
```

6. etcd プロセスの確認 ※ etcd プロセスが TCP 2379 ポートを listen していることを確認

```
$ sudo ss -tnatp | grep etcd
```


etcd のコンフィギュレーション

1. 以下の etcd 設定ファイルを確認

```
/etc/etcd/etcd.conf
```

2. 以下の項目を設定

設定項目	意味	設定値	備考
ETCD_NAME	インスタンス名	default (デフォルト)	
ETCD_DATA_DIR	データディレクトリ	"/var/lib/etcd/default.etcd" (デフォルト)	
ETCD_LISTEN_CLIENT_URLS	リスン URL	http://0.0.0.0:2379	変更必須
ETCD_ADVERTISE_CLIENT_URLS	Advertise URL	http://0.0.0.0:2379	変更必須

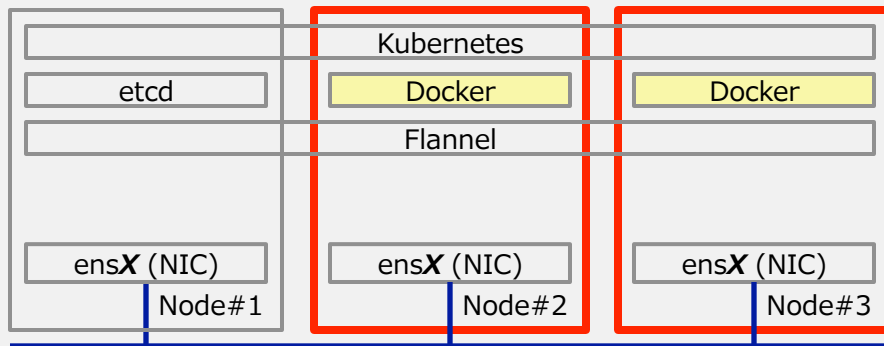
3. 設定変更後 etcd を再起動

```
$ systemctl restart etcd.service
```

※ 本番環境で etcd を使用する場合、上記以外に セキュリティ、冗長構成 の設定が必要

Docker の構築概要

1. インストールと動作確認



Docker のインストール と 動作確認

1. docker のインストール

```
$ sudo yum install -y docker
```

2. インストールの確認

```
$ docker --version  
Docker version 1.10.3, build 79ebcd8-unsupported
```

3. 自動起動の有効化

```
$ sudo systemctl enable docker.service && systemctl is-enabled docker.service
```

4. docker の起動

```
$ sudo systemctl start docker.service && systemctl status docker.service
```

5. 仮想 Bridge (docker0) の起動確認

```
$ ip link show type bridge
```

6. docker の動作確認

```
$ docker run hello-world
```

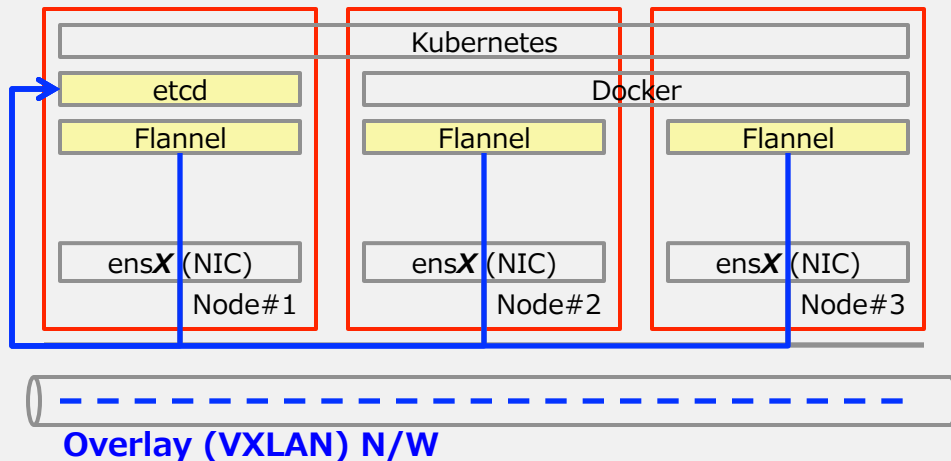
7. docker の停止

```
$ sudo systemctl stop docker.service && systemctl status docker.service
```

※ 上記 1. ～ 6. の手順を Node#2, Node#3 で実施

Overlay N/W の構築概要

1. Flannel コンフィギュレーションの etcd への登録
2. Flannel のインストールとコンフィギュレーション
3. 自動起動設定
4. 動作確認



Flannel コンフィギュレーションの etcd への登録

1. docker 仮想ブリッジ (docker0) の削除

```
$ ip link delete docker0
```

2. Flannel コンフィギュレーション (.json) の取得 - 以下の URL からファイルを取得

```
https://github.com/yyamada-redhat/training\_kubernetes/blob/master/env/flannel-config-vxlan.json
```

3. コンフィギュレーションの確認

```
$ cat flannel-conig-vxlan.json
{
  "Network": "10.1.0.0/16",
  "SubnetLen": 24,
  "Backend": {
    "Type": "vxlan",
    "VNI": 1
  }
}
```

設定項目	意味	設定値
Network	Flannel が Overlay N/W に使用する IPv4 ネットワーク	10.1.0.0
SubnetLen	各 Linux Host に割り当てられるサブネットマスクの長さ	24
Backend:Type	パケット転送方式 (udp, vxlan)	vxlan

4. Flannel コンフィギュレーションの etcd への登録

```
$ etcdctl set /atomic.io/network/config < flannel-config-vxlan.json
```

5. etcd 上の Flannel コンフィギュレーション の確認

```
$ etcdctl get atomic.io/network/config
{
  "Network": "10.1.0.0/16",
  "SubnetLen": 24,
  "Backend": {
    "Type": "vxlan",
    "VNI": 1
  }
}
```

Flannel のインストールとコンフィギュレーション

1. flannel のインストール

```
$ sudo yum install -y flannel
```

2. 以下の flannel 設定ファイルを確認

```
/etc/sysconfig/flanneld
```

3. 以下の項目を設定

設定項目	意味	設定値
FLANNEL_ETCD_ENDPOINTS	etcd URL	(例) ="http://192.168.140.144:2379"
FLANNEL_ETCD_PREFIX	コンフィギュレーション key	(例) ="/atomic.io/network"
FLANNEL_OPTIONS	オプション	

4. 以下のファイルを確認

```
/usr/lib/systemd/system/flanneld.service
```

flannel の 自動起動設定

1. Unit 設定ファイル (systemd) の作成 – 以下の URL からファイルを取得
`https://github.com/yyamada-redhat/training-kubernetes/blob/master/prv/flannel.service`

2. Unit 設定ファイルの配置

```
$ sudo cp flannel.service /usr/lib/systemd/system/
```

3. 自動起動の有効化

```
$ sudo systemctl enable flanneld
```

4. flanneld の起動

```
$ sudo systemctl start flanneld.service
```

5. N/W インターフェースの確認

```
$ ip addr
```

本手順は実行しない

flannel の 動作確認

1. flannel が構成する仮想ブリッジ (flannel0) の確認

```
$ ip a | grep flannel | grep inet
```

2. flannel の サブネット環境設定ファイル の確認

```
/run/flannel/subnet.env
```

3. etcd コンフィギュレーションの確認

```
$ etcdctl get /atomic.io/network
```

Appendix

1. 一般ユーザ権限による docker の実行

一般ユーザ権限による docker の実行

1. /etc/sysconfig/docker の設定

```
OPTIONS='--selinux-enabled --log-driver=journald -G dockerroot'
```

2. dockerroot グループへの一般ユーザの追加

```
$ sudo usermod dockerroot $USER_NAME
```

3. docker (daemon) の再起動

```
$ sudo systemctl restart docker.service
```

4. 変更対象ユーザのシステムの login & logout

5. docker の動作確認

```
$ docker run hello-world
```

※ 上記の設定は開発環境のみで行うこと。

References

- [1] [kubernetes] (<http://kubernetes.io>)
- [2] [CoreOS] (<https://coreos.com>)
- [3] [flannel] (<https://github.com/coreos/flannel/>)
- [4] [rkt] (<https://coreos.com/rkt/>)
- [5] [etcd] (<https://coreos.com/etcd/>)
- [6] [docker] (<https://www.docker.com>)



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos