



# Kubernetes LINSTOR ボリューム 構築手順書

サイオステクノロジー株式会社

バージョン 1.1 2019/09/05

# 目次

1. はじめに .....	1
1.1. 前提条件 .....	1
2. ソフトウェア .....	3
3. linstor-csi-controller, linstor-csi-node の配備 .....	4
4. StorageClass, PersistentVolumeClaim の準備 .....	5
5. fedora with DRBD Persistent Volume .....	7
6. 変更履歴 .....	9
6.1. v1.1 2019/09/05 .....	9
6.2. v1.0 2019/02/15 .....	9

# 1. はじめに

DRBD を Kubernetes で使用するには Linstor CSI Plugin を使用します。CSI Plugin は

```
https://github.com/LINBIT/linstor-csi
```

からダウンロードできます。以下に構築例として、Kubernetes で Persistent Volume として DRBD を設定し、fedora pod で DRBD データ領域にアクセスする方法を示します。

## 1.1. 前提条件

前提条件として、

- 3台のホストを用意する。1台を Master(LINSTOR では controller), 残り 2台を Worker (INSTOR では satellite) として使用する。
- すべてのホストに CentOS 7.6、DRBD9, LINSTOR をインストールしてある。
- すべてのホストに Docker, Kubernetes をインストールしてある。
- Kubernetes の Master node, Work node は設定済みである。
- LINSTOR の node は設定済みである。
- LINSTOR の storage-pool は drbdpool として作成済みである。

```
# docker run hello-world
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
:
```

```
# kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
tate-z55	Ready	master	164m	v1.15.3
tate-z56	Ready	<none>	162m	v1.15.3
tate-z57	Ready	<none>	162m	v1.15.3

```
# linstor node list
```

```
+-----+
| Node      | NodeType | Addresses                | State |
+-----+
| tate-z55  | COMBINED | 10.1.12.155:3366 (PLAIN) | Online |
| tate-z56  | SATELLITE | 10.1.12.156:3366 (PLAIN) | Online |
| tate-z57  | SATELLITE | 10.1.12.157:3366 (PLAIN) | Online |
+-----+
```

```
# linstor storage-pool list
```

```
+-----+
|StoragePool|Node      |Driver  |PoolName|FreeCapacity|TotalCapacity|SupportsSnapshots|
+-----+
| drbdpool  | tate-z55 | LvmDriver | drbdpool | 4.88 GiB | 4.88 GiB | false |
| drbdpool  | tate-z56 | LvmDriver | drbdpool | 4.88 GiB | 4.88 GiB | false |
| drbdpool  | tate-z57 | LvmDriver | drbdpool | 4.88 GiB | 4.88 GiB | false |
+-----+
```

- LINSTOR のインストールは DRBD9 LINSTOR 構築手順書(<https://fex.3ware.co.jp/s/iJiSTfimagMDzhn>) を参照ください。パッケージを自分でビルドする方法もあります。

## 2. ソフトウェア

今回使用したソフトウェア情報を以下に記載します。

CentOS 7.6

kmod-drbd-9.0.19\_3.10.0\_957-1

drbd-9.10.0-1

drbd-utils-9.10.0-1

linstor-common-1.0.1-1

linstor-client-1.0.1-1

linstor-controller-1.0.1-1

linstor-satellite-1.0.1-1

python-linstor-1.0.0-1

<https://download.docker.com/linux/centos/docker-ce.repo>

docker-ce-19.03.1-3

docker-ce-cli-19.03.1-3

[https://packages.cloud.google.com/yum/repos/kubernetes-el7-\\$basearch](https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch)

kubeadm-1.15.3-0

kubect1-1.15.3-0

kubelet-1.15.3-0

### 3. linstor-csi-controller, linstor-csi-node の配備

linstor-csi をダウンロードし、linstor-csi-dev.yaml の IP address を書き換えます。

```
# git clone https://github.com/LINBIT/linstor-csi
# cp linstor-csi/examples/k8s/deploy/linstor-csi-dev.yaml .
# diff -u linstor-csi/examples/k8s/deploy/linstor-csi-dev.yaml linstor-csi-dev.yaml
--- linstor-csi/examples/k8s/deploy/linstor-csi-dev.yaml    2019-09-05 04:57:53.806738020
+0900
+++ linstor-csi-dev.yaml    2019-09-05 08:16:28.818746618 +0900
@@ -83,7 +83,7 @@
         fieldRef:
           fieldPath: spec.nodeName
         - name: LINSTOR_IP
-         value: "http://linstor-controller.example.com:3370"
+         value: "http://10.1.12.155:3370"
         imagePullPolicy: "Always"
         volumeMounts:
           - name: socket-dir
@@ -262,7 +262,7 @@
         fieldRef:
           fieldPath: spec.nodeName
         - name: LINSTOR_IP
-         value: "http://linstor-controller.example.com:3370"
+         value: "http://10.1.12.155:3370"
         imagePullPolicy: "Always"
         securityContext:
           privileged: true

# kubectl create -f linstor-csi-dev.yaml
# kubectl describe statefulset linstor-csi-controller --namespace kube-system
# kubectl describe daemonset linstor-csi-node --namespace kube-system
```

正しく動作すると以下のような出力になります。

```
# kubectl get pod --namespace kube-system -o wide |grep linstor
linstor-csi-controller-0          4/4    Running    0           36m    10.244.1.2    tate-
z56    <none>    <none>
linstor-csi-node-5jrdx           2/2    Running    0           36m    10.1.12.157   tate-
z57    <none>    <none>
linstor-csi-node-ph4qb           2/2    Running    0           36m    10.1.12.156   tate-
z56    <none>    <none>
```

## 4. StorageClass, PersistentVolumeClaim の準備

あとは StorageClass に登録し、

```
# vi two-replica-sc.yaml

apiVersion: storage.k8s.io/v1beta1
kind: StorageClass
metadata:
  name: two-replica
provisioner: linstor.csi.linbit.com
parameters:
  autoPlace: "2"
  storagePool: "drbdpool"

# kubectl create -f two-replica-sc.yaml
storageclass.storage.k8s.io/two-replica created
# kubectl describe storageclass two-replica
```

PersistentVolumeClaim でボリュームを割当ます。

```
# vi my-first-volume-pvc.yaml

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-first-volume
  annotations:
    volume.beta.kubernetes.io/storage-class: two-replica
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi

# kubectl create -f my-first-volume-pvc.yaml
# kubectl describe persistentvolumeclaim my-first-volume
```

正しく動作すると resource, storage-pool は以下ようになります。

```
# linstor resource list
```

```
+-----+
| ResourceName          | Node   | Port | Usage | State |
+-----+-----+-----+-----+-----+
| pvc-a31343c5-1002-447e-9b4c-34eb05d53cef | tate-z55 | 7000 | Unused | UpToDate |
| pvc-a31343c5-1002-447e-9b4c-34eb05d53cef | tate-z56 | 7000 | Unused | UpToDate |
+-----+-----+-----+-----+-----+
```

```
# linstor storage-pool list
```

```
+-----+
| StoragePool | Node   | Driver   | PoolName | FreeCapacity | TotalCapacity | SupportsSnapshots |
+-----+-----+-----+-----+-----+-----+-----+
| drbdpool   | tate-z55 | LvmDriver | drbdpool | 4.88 GiB   | 4.88 GiB   | false              |
| drbdpool   | tate-z56 | LvmDriver | drbdpool | 4.88 GiB   | 4.88 GiB   | false              |
| drbdpool   | tate-z57 | LvmDriver | drbdpool | 4.88 GiB   | 4.88 GiB   | false              |
+-----+-----+-----+-----+-----+-----+-----+
```



## 5. fedora with DRBD Persistent Volume

fedoraコンテナを作成し動作確認します。

```
# vi fedora.yaml

apiVersion: v1
kind: Pod
metadata:
  name: fedora
  namespace: default
spec:
  containers:
  - name: fedora
    image: fedora
    command: ["/bin/bash"]
    args: ["-c", "while true; do sleep 10; done"]
    volumeMounts:
    - name: default-my-first-volume
      mountPath: /data
    ports:
    - containerPort: 80
  volumes:
  - name: default-my-first-volume
    persistentVolumeClaim:
      claimName: "my-first-volume"

# kubectl create -f fedora.yaml
# kubectl describe pod fedora
```

FailがでてなければOKです。問題なければfedora podに接続します。

```
# kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED
NODE
fedora        1/1     Running   0          49s   10.244.2.22  tate-z57   <none>
# kubectl exec fedora -it /bin/bash
```

/dataに書き込みます。

```
# echo "Test1" > /data/Test1
# exit
```

fedora podを抜け削除し、tate-z57 意外に割当ます。

```
# kubectl delete pod fedora
# kubectl cordon tate-z57
# kubectl create -f fedora.yaml
# kubectl get pod -o wide
NAME      READY   STATUS    RESTARTS   AGE      IP           NODE      NOMINATED NODE
fedora    1/1     Running   0           1m       10.244.1.4   tate-z56  <none>
```

tate-z56 にアサインされたので、ここに接続します。

```
# kubectl exec fedora -it /bin/bash
[root@fedora /]# ls -la /data/
total 8
drwxr-xr-x 2 root root  19 Feb 18 01:29 .
drwxr-xr-x 1 root root 4096 Feb 18 01:31 ..
-rw-r--r-- 1 root root   6 Feb 18 01:29 Test1
# cat /data/Test1
Test1
```

先程の書き込みが確認できました。

## 6. 変更履歴

### 6.1. v1.1 2019/09/05

- v0.7.2 対応

### 6.2. v1.0 2019/02/15

- 最初のバージョン