


KubernetesにおけるCSI

- Kubernete Meetup Tokyo #25 -
@bells17

自己紹介

- 名前: 早川 大貴
- 所属: 株式会社IDCフロンティア クラウド本部
- 職業: ソフトウェアエンジニア
- 職務内容: CSIドライバの実装・検証などIDCFクラウドでKubernetesの利用を
いい感じにする諸々をやる
-  @bells17

今日の目標・話すこと

目標

- 今後Kubernetesにおけるボリュームプラグインの標準となる仕様であるCSIについて紹介すること
- KubernetesがCSIとどのように連携を行っているかについての概要について紹介すること

話すこと

- Kubernetesにおけるボリュームプラグインの歴史
- Container Storage Interface(CSI)
- KubernetesとCSIの連携方法

今日話さないこと・注意事項

今日話さないこと

- CSI以前のKubernetesのボリュームプラグインについては概要程度しか触れません
- client-goやKubernetes OperatorパターンなどKubernetes本体の実装などについては詳しく説明しません
 - (時間の都合でこのあたりについては基礎的な知識がある前提で話してしまうかもしれません)

注意事項

- 発表者はストレージの専門家ではないので、ストレージ関連の説明に関しては適切な説明ができないかもしれません

Kubernetesにおける ボリュームプラグインの歴史

Kubernetesにおける ボリュームプラグインの歴史

- v1.8まではボリュームプラグインはKubernetesのリポジトリに含まれるin-treeのプラグインのみだった
- v1.8ではFlexvolumeによる外部プラグインのサポートが開始
 - ただしノードおよびマスターマシンのルートファイルシステムへのアクセスが必要といったいくつかの課題有り
- Kubernetes v1.9でContainer Storage Interface (CSI)対応がAlphaリリース
- Kubernetes v1.13ではCSIがGAに
- 最終的にはすべてのボリュームプラグインをCSIプラグインとして外部に切り出す予定

CSI移行のモチベーション ~ ストレージプロバイダ ~

- ストレージプロバイダはCSIプラグインを実装するだけで各種コンテナオーケストレーター(CO)に対応できるようになる
- in-treeプラグインと比較するとKubernetesのリリースフローに依存しないリリースができるようになる

CSI移行のモチベーション ~ Kubernetes ~

- ボリュームプラグインのソースコードをKubernetesから除外することができる
- ボリュームプラグインをコンテナ化することでホストマシンからマウントツールやファイルシステムツールなどへの依存関係を取り除くことができる
- Kubernetes本体はCSIプラグイン以外の開発が不要となるため開発コストが削減される

v1.16リリース時のCloud Providerを外部化した際の バイナリサイズ比(CSI移行でもこれくらい減るかも...)



bentheelder 10:55 AM

<https://www.diffchecker.com/oAYWVjjk>



diffchecker.com

Diff Checker

Diffchecker is an online diff tool to compare text to find the difference between two text files

kube-apiserver: 203M -> 133M

kube-controller-manager: 151M -> 80M

kubelet: 152M -> 85M



1



1



現状のボリュームプラグインの数

```
$ tree -d -L 1 pkg/volume
```

```
pkg/volume
```

```
├── awsebs  
├── azure_dd  
├── azure_file  
├── cephfs  
├── cinder  
├── configmap  
├── csi  
├── downwardapi  
├── emptydir  
├── fc  
├── flexvolume  
├── flocker  
└── gcepd
```

```
...
```

```
30 directories
```



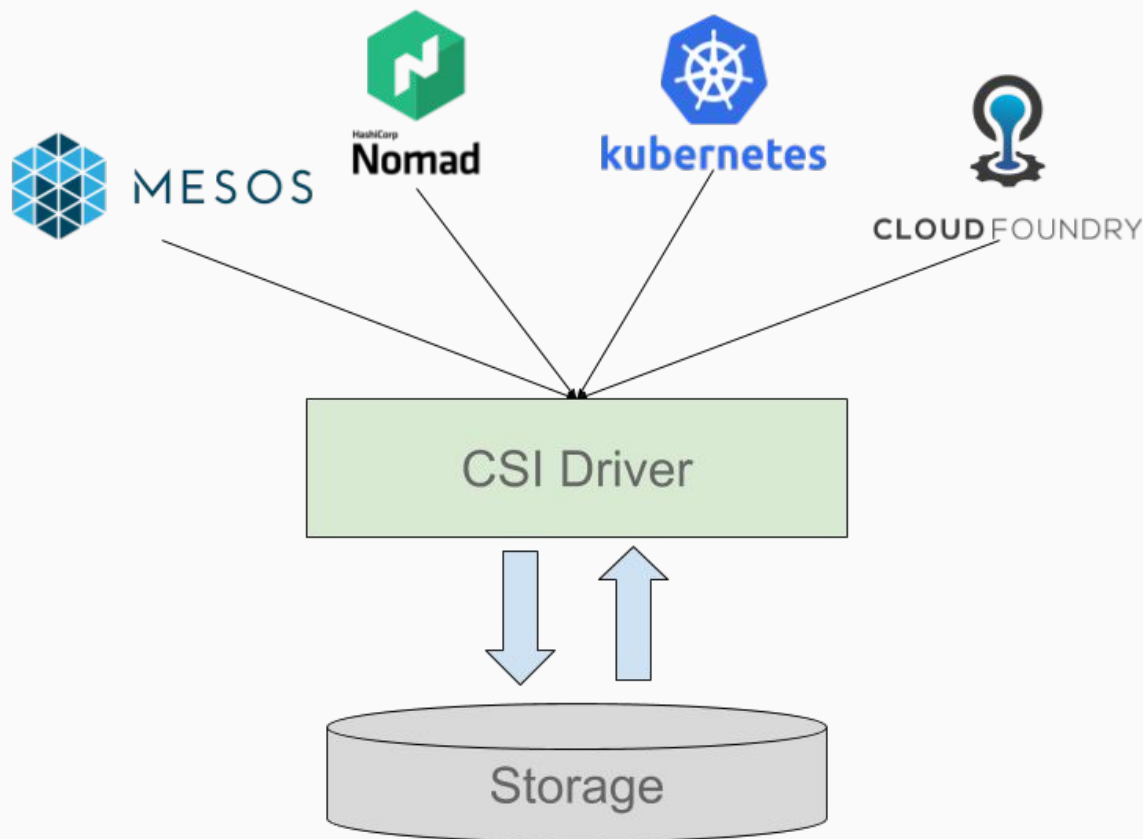
CONTAINER
STORAGE
INTERFACE

Container Storage Interface(CSI)

Container Storage Interface(CSI)とは

- KubernetesやMesos、Cloud Foundryといったコンテナオーケストレーター(CO)でストレージを利用するための共通の仕様を定義したもの
- ストレージプロバイダは一度CSIに対応したドライバを書いておけばあとはKubernetesやMesosでも共通して利用できるようにすることを目指して作られた
- CSI用のストレージプラグインのことをCSIドライバ/CSIプラグインと呼ぶ
- CSIドライバー毎に有効な機能の設定をRPCでCOに通知できるようになっている
 - 各CSIドライバーでどの機能を提供しているのかを選択できる

Container OrchestratorとCSIドライバ



CSIで定義されている仕様

- CSIドライバの通信方法や提供方法など
- CSIドライバが提供する機能
- COがCSIドライバを利用するためのgRPCのProtocol Buffers

CSI Overview

Functions & Protocol Buffers

CSI Driver

Node Plugin

Node Service

- NodeGetCapabilities
- NodeGetInfo
- etc...

Controller Plugin

Controller Service

- ControllerGetCapabilities
- etc...

Identity Service

- GetPluginInfo
- GetPluginCapabilities
- Probe

Rules

Provided in Container Image Format



Use Unix Domain Socket & gRPC



etc...

CSIドライバの通信方法や提供方法について

- コンテナイメージ形式で提供する必要がある
- CSIドライバとCOの通信は
 - UNIXドメインソケットを介して転送され
 - gRPCプロトコルを使用する必要がある
 - (現状ではWindowsコンテナはサポートされていない...?)

CSIが提供する機能

- ボリュームの作成/削除
- ノードへのボリュームのアタッチ/デタッチ
- ボリュームのマウント/アンマウント
- ボリュームのスナップショットの作成/削除
- etc...

Protocol Buffersで定義されているgRPCの インターフェイス

- Controller Plugin
- Node Plugin

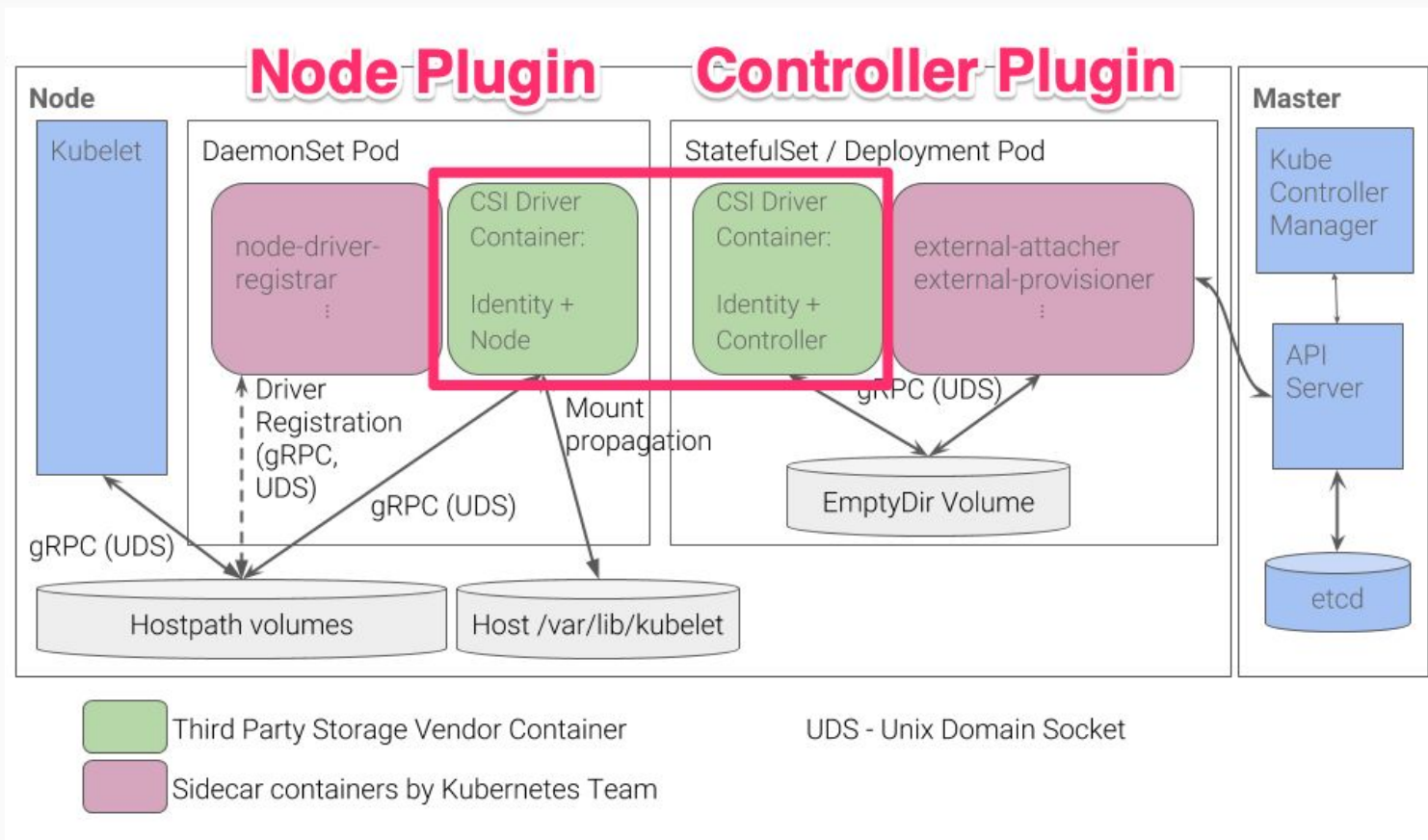
Controller Plugin

- COのmaster nodeで動くgRPCサーバー
- Controller Service/Identity Serviceの2種類のgRPCのサービスを実装したもの
- 主に以下の機能を提供する
 - ボリュームの作成/削除
 - ノードへのボリュームのアタッチ/デタッチ
 - ボリュームのスナップショットの作成/削除

Node Plugin

- COの各worker nodeで動くgRPCサーバー
- Node Service/Identity Serviceの2種類のgRPCのサービスを実装したもの
- 主に以下の機能を提供する
 - ノードにアタッチされたボリュームのフォーマット
 - ボリュームのマウント/アンマウント

KubernetesにおけるCSIドライバの動作イメージ



RPC Service

- Identity Service
- Controller Service
- Node Service

Identity ServiceのRPC

RPC	Description
GetPluginInfo	プラグイン名やバージョン情報を返す
GetPluginCapabilities	プラグインがサポートしている機能の一覧を返す (e.g. トポロジーをサポートしているかなど)
Probe	ヘルスチェック情報を返す

Controller Serviceの主なRPC

RPC	Description
ControllerGetCapabilities	CSIドライバのController Serviceがサポートしている機能の一覧を返す
CreateVolume/ DeleteVolume	ボリュームの作成/削除を行う
ControllerPublishVolume/ ControllerUnpublishVolume	ノードへのボリュームへのアタッチ/デタッチを行う

Node Serviceの主なRPC

RPC	Description
NodeGetCapabilities	CSIドライバのNode Serviceがサポートしている機能の一覧を返す
NodeGetInfo	CSIドライバがノードを扱う上で必要なノードのIDやアタッチ可能なボリューム数、トポロジー情報を返す
NodeStageVolume/ NodeUnstageVolume	ファイルシステムのフォーマットやNodeへのボリュームのマウントアンマウントを行う
NodePublishVolume/ NodeUnpublishVolume	ファイルシステムのフォーマットやPodへのボリュームのマウントアンマウントを行う

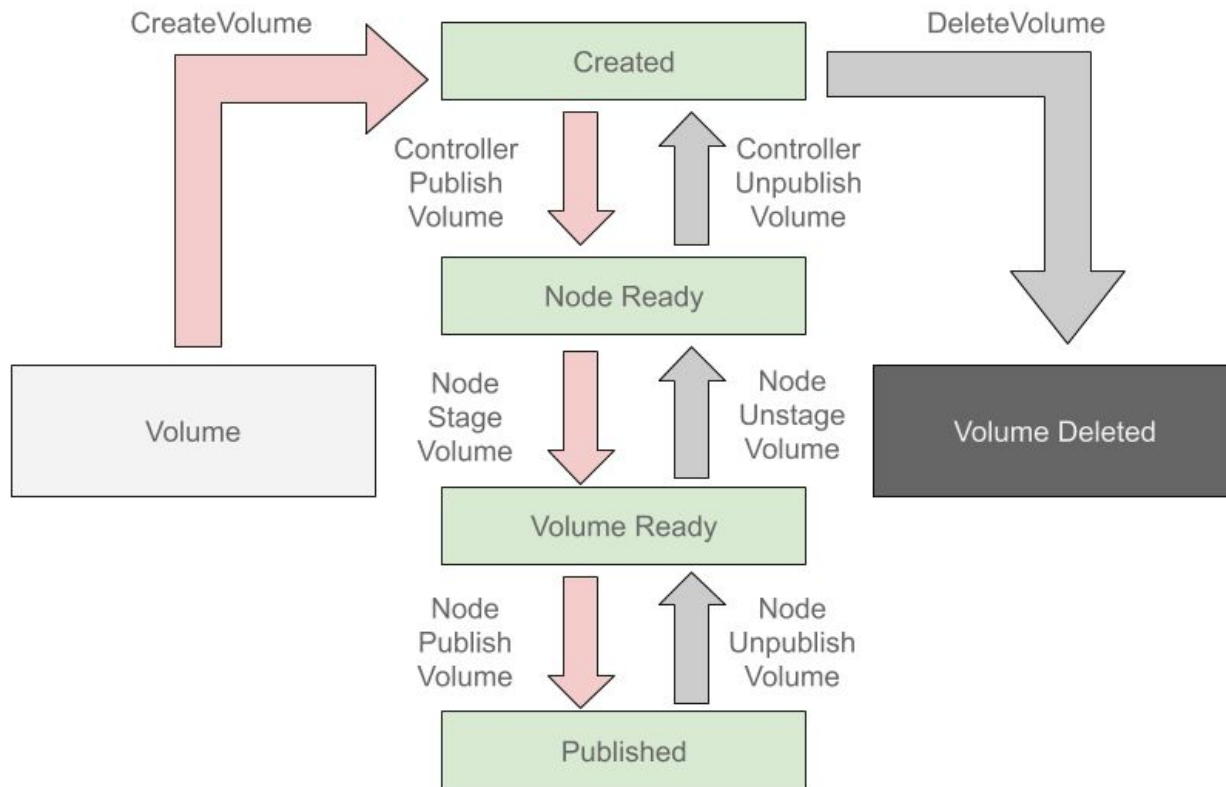
各ServiceのCapability一覧

```
message PluginCapability {
  message Service {
    enum Type {
      UNKNOWN = 0;
      CONTROLLER_SERVICE = 1;
      VOLUME_ACCESSIBILITY_CONSTRAINTS = 2;
    }
    ...
  }
  ...
}
```

```
message NodeServiceCapability {
  message RPC {
    enum Type {
      UNKNOWN = 0;
      STAGE_UNSTAGE_VOLUME = 1;
      GET_VOLUME_STATS = 2;
      EXPAND_VOLUME = 3;
    }
    ...
  }
  ...
}
```

```
message ControllerServiceCapability {
  message RPC {
    enum Type {
      UNKNOWN = 0;
      CREATE_DELETE_VOLUME = 1;
      PUBLISH_UNPUBLISH_VOLUME = 2;
      LIST_VOLUMES = 3;
      GET_CAPACITY = 4;
      CREATE_DELETE_SNAPSHOT = 5;
      LIST_SNAPSHOTS = 6;
      CLONE_VOLUME = 7;
      PUBLISH_READONLY = 8;
      EXPAND_VOLUME = 9;
      LIST_VOLUMES_PUBLISHED_NODES = 10;
    }
    ...
  }
  ...
}
```

ボリュームのライフサイクル



CSIまとめ

- ストレージプロバイダがボリュームプラグインをCOに提供するための共通の仕様を定義したもの
- 以下のようなものを定義している
 - CSIドライバの動作環境や通信方法の定義
 - コンテナイメージ形式
 - UNIXドメインソケット/gRCPプロトコル
 - RPCインターフェイスの定義
 - Controller Plugin
 - Node Plugin
- より詳細な仕様については[Container Storage Interface Spec](#)を参照

KubernetesとCSIの連携方法

KubernetesのCSI用Sidecarアプリケーション

KubernetesではController Plugin/Node PluginそれぞれにSidecar形式のアプリケーションを提供することで、CSIドライバとKubernetesを連携できるようになっている

Controller Sidecar

Sidecar	Description
external-provisioner	ボリュームの作成 / 削除を行う
external-attacher	ボリュームのアタッチ / デタッチを行う
external-resizer	ボリュームのリサイズを行う
external-snapshotter	スナップショットの作成 / 削除を行う
livenessprobe	Liveness Probe用HTTPプロキシ

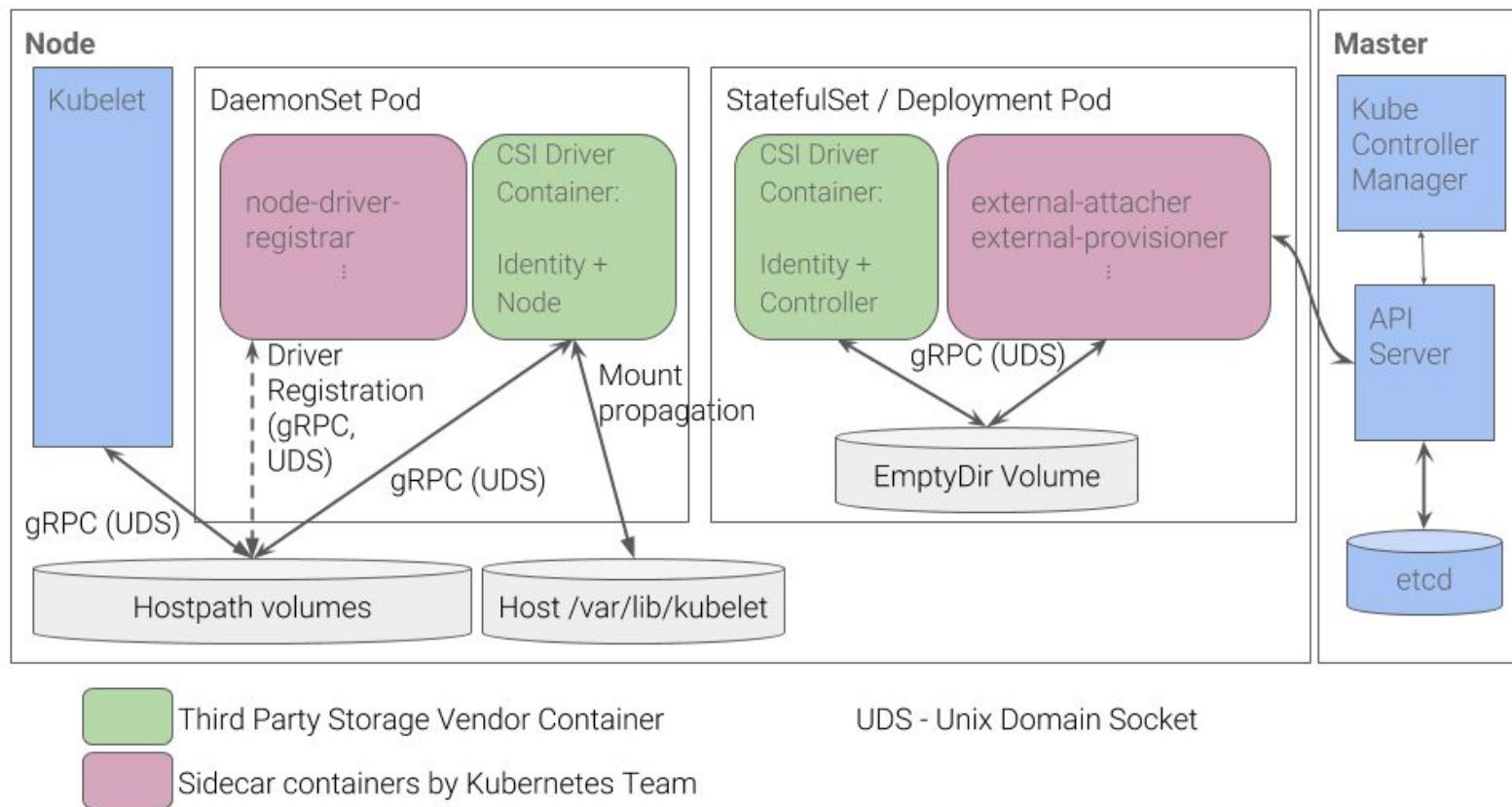
※ Controller Sidecarを統合する提案も上がっているようなので将来的には 1つのアプリケーションに統合されるかもしれません

- https://groups.google.com/d/msg/kubernetes-sig-storage-wg-csi/HB5_8uLiq74/UYzN0jEhAAAJ
- https://groups.google.com/forum/?hl=ja#!topic/kubernetes-sig-storage-wg-csi/HB5_8uLiq74

Node Sidecar

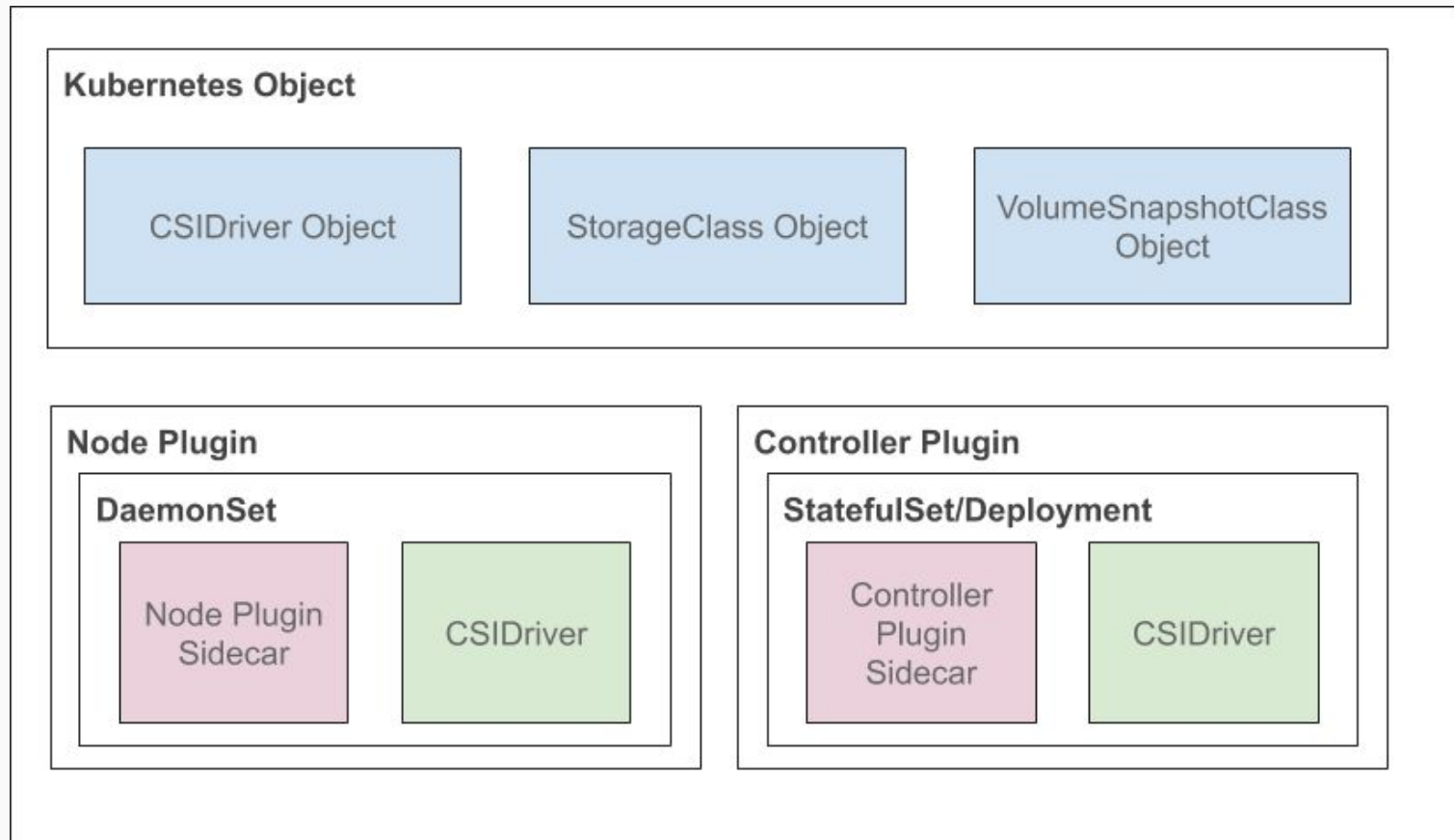
Sidecar	Description
node-driver-registrar	kubeletにあるPlugin Watcherという機能を利用してCSIドライバをkubeletに登録する機能を提供する

KubernetesにおけるCSIドライバの動作イメージ



CSIドライバの登録フロー

CSIDriverを利用するときにリリースする必要があるもの



CSIDriver Object

- CSIDriver Objectは以下のような目的で作成を行うもの
 - `kubectl get CSIDriver` コマンドでユーザーに利用可能なCSIドライバを示すために作成する
 - `podInfoOnMount` などCSIDriverの設定を行うことができるので、設定の変更が必要なCSIドライバの場合には作成する必要がある
- CSIDriverの設定を変更する必要がない場合は作成は必須ではない

```
apiVersion: storage.k8s.io/v1beta1
kind: CSIDriver
metadata:
  name: mycsidriver.example.com
spec:
  attachRequired: true
  podInfoOnMount: true
```

StorageClass Object

- PersistentVolumeClaim(PVC)から指定するStorageClass Object
- PVCが作成された時にボリュームのプロビジョニングを行うprovisionerを設定することでPVCがプロビジョニングを行うボリュームプラグインを指定することができる
- CSIの場合csi.storage.k8s.io/fstypeなどStorage Class単位でCSIドライバのデフォルトの設定を行うことができる

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-csi-storage
provisioner: mycsidriver.example.com
parameters:
  csi.storage.k8s.io/fstype: ext4
  csi.storage.k8s.io/provisioner-secret-name: mysecret
  csi.storage.k8s.io/provisioner-secret-namespace: mynamespace
```

StorageClassを利用してボリュームを作成する

先程作成したStorageClassを利用してボリュームを作成する場合
PersistentVolumeClaimは以下のように設定する

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: csi-pod-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: my-csi-storage
```

← 作成したStorageClassを指定する

VolumeSnapshotClass Object

- VolumeSnapshotClass ObjectはVolumeSnapshotリソースを利用して、ボリュームのスナップショットを作成するためのオブジェクト
- `snapshotter`にはボリュームプラグイン名を指定する
- CSIドライバがボリュームのスナップショット機能をサポートしない場合には作成は不要

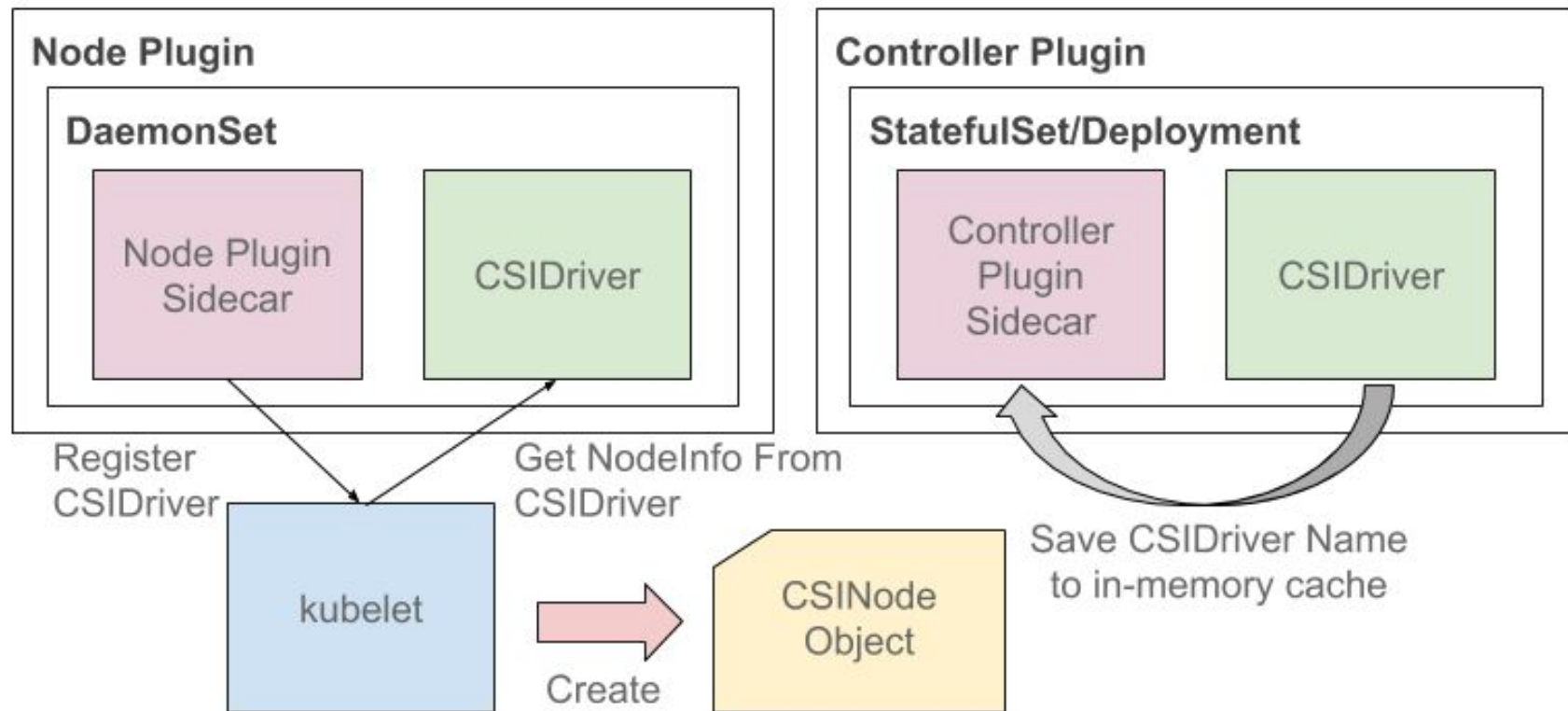
```
apiVersion: snapshot.storage.k8s.io/v1alpha1
kind: VolumeSnapshotClass
metadata:
  name: my-csi-snapclass
snapshotter: mycsidriver.example.com
```

VolumeSnapshotClassを利用してボリュームのスナップショットを作成する

先程作成したVolumeSnapshotClassを利用してボリュームのスナップショットを作成する場合VolumeSnapshotは以下のように設定する

```
apiVersion: snapshot.storage.k8s.io/v1alpha1
kind: VolumeSnapshot
metadata:
  name: snapshot-csi-pod-pvc
spec:
  snapshotClassName: my-csi-snapclass
  source:
    name: csi-pod-pvc
    kind: PersistentVolumeClaim
```


CSIドライバのデプロイ時に起こること



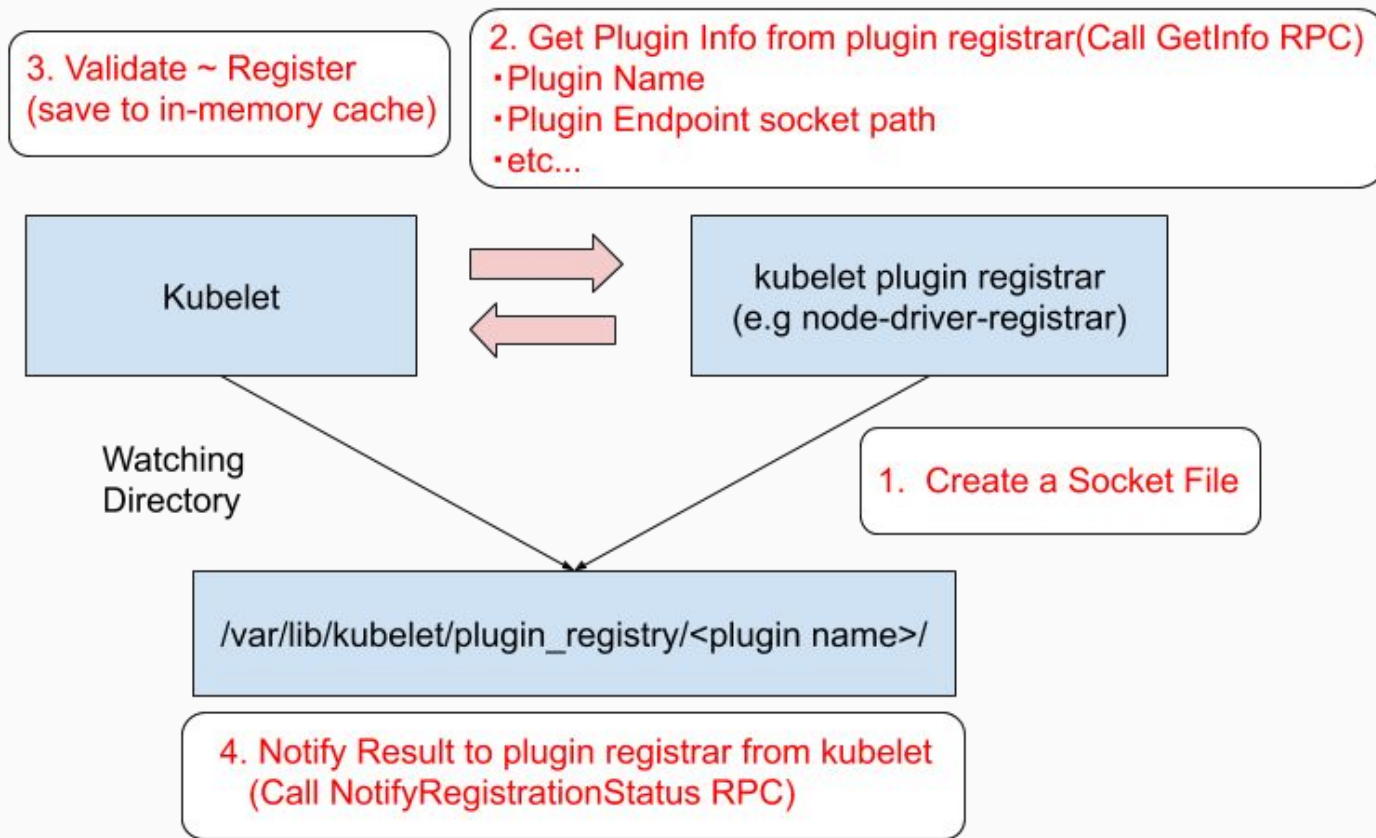
CSINode Object

- ボリュームのアタッチ/デタッチやトポロジーに基づくPod作成先のノードの選定に利用される
- kubeletによってCSIドライバのNodeGetInfo RPCの情報から自動的に作成される
- Node Objectに追加するには情報量が多いのでCSI用の追加情報がCSINode Objectに切り出された

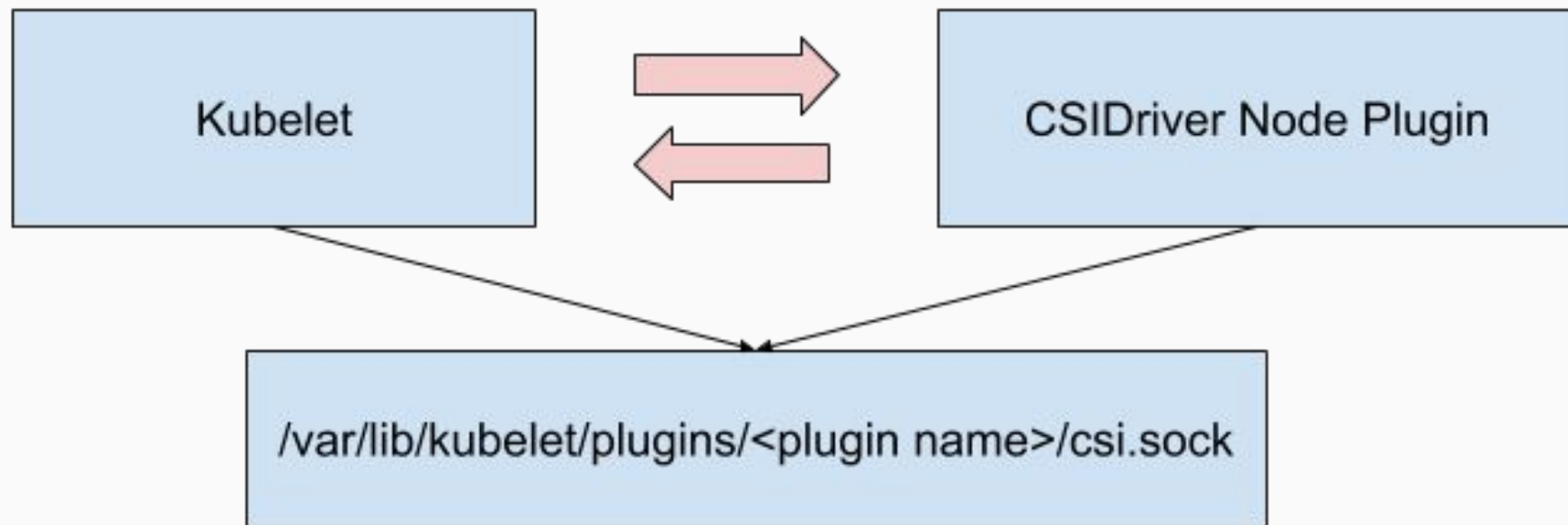
```
apiVersion: storage.k8s.io/v1beta1
kind: CSINodeInfo
metadata:
  name: node1
spec:
  drivers:
  - name: mycsidriver.example.com
    nodeID: storageNodeID1
    topologyKeys:
    - mycsidriver.example.com/regions
    - mycsidriver.example.com/zones
```

kubeletのPluginWatcherと CSI Plugin

Kubelet Plugin Watcher

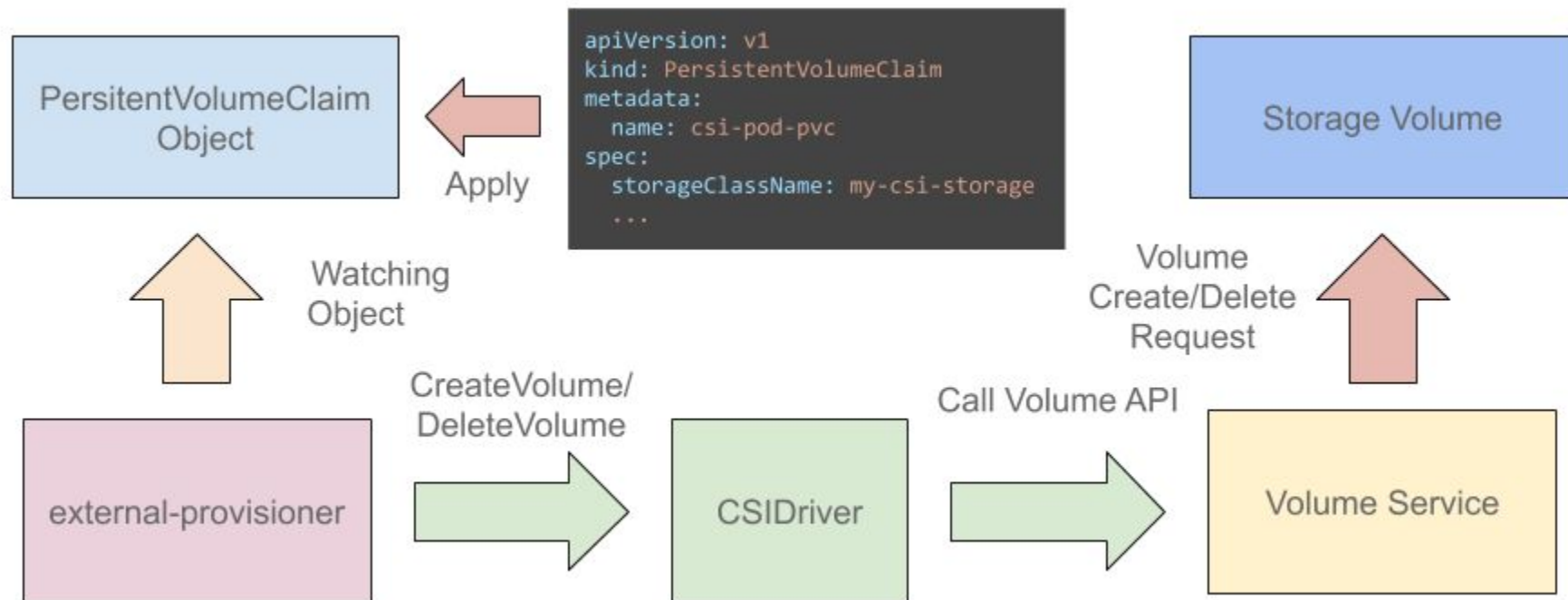


Call some gRPC request directly from a kubelet

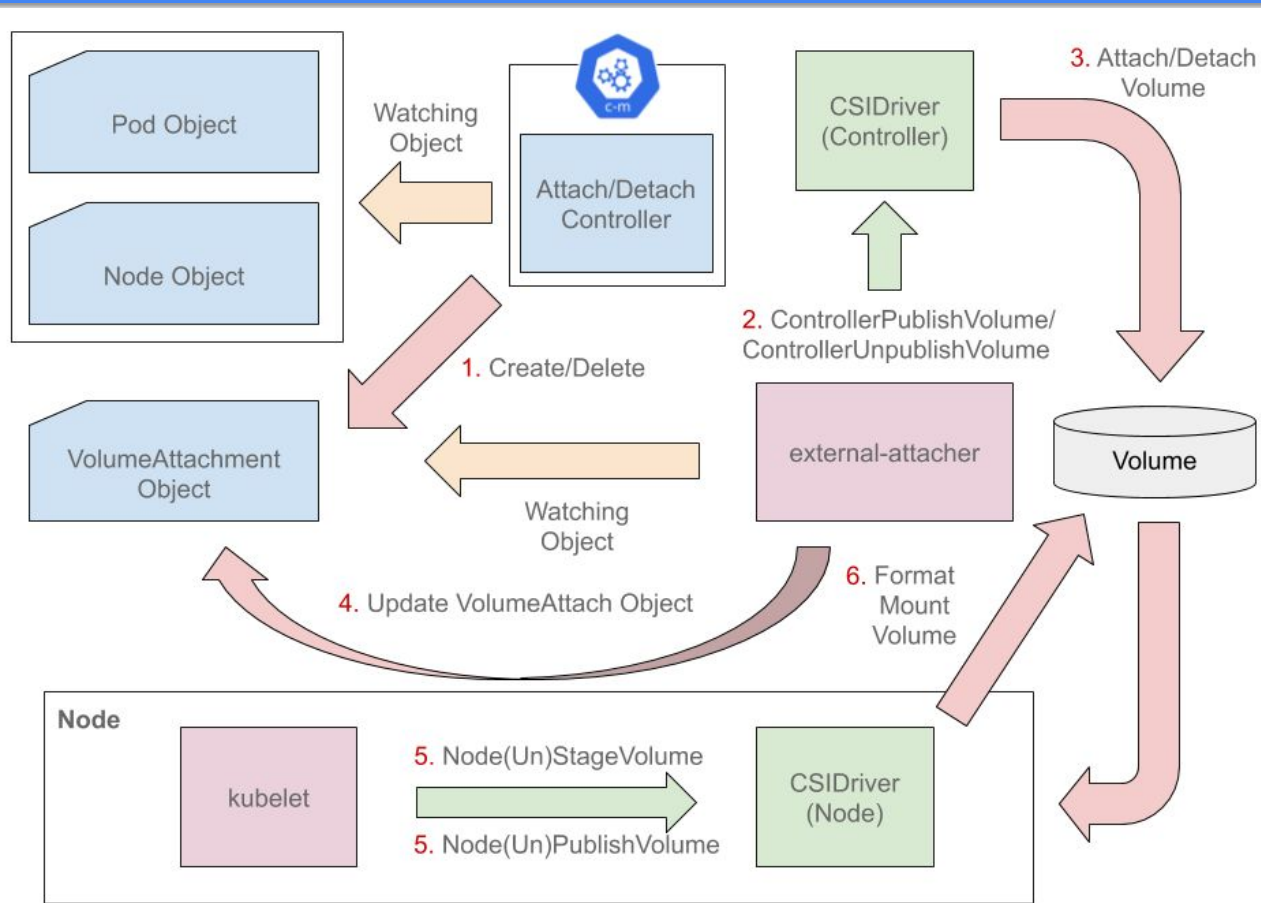


ボリューム作成 ~ マウントまでの
フローについて

ボリューム作成時の処理フロー



Pod作成時のボリュームマウントまでのフロー



KubernetesとCSIの連携方法まとめ

- Kubernetesはサイドカーアプリケーションを提供することでCSIドライバとの連携を行っている
- CSIドライバを利用するために専用のKubernetesリソースを追加している
 - e.g. CSIDriver Object
- CSIドライバはそれぞれ以下のような方法で連携されている

Controller Plugin	サイドカーアプリケーションが Kubernetes Operatorパターンで実装されており、イベントが発生するとサイドカーアプリケーションからController Pluginにリクエストが送られる
Node Plugin	kubeletにプラグインとして登録され、ボリュームを Podにアタッチする際にkubeletからNode Pluginに直接リクエストが送られる

おまけ

その他CSIドライバを自分で開発する際に知っておくと 便利なツール

Name	Description
csi-sanity	CSIドライバが正しく動作するかをチェックすることのできる Kubernetesの公式が出しているe2eテストングツール
Kubernetes E2E Tests	Kubernetes本体のe2eテストを行うためのツール
CSC	CSIドライバのRPCコールを行えるCLIツール あるとデバッグが捗る
Kubernetes mount util	Kubernetesリポジトリの/pkg/util/mountにあるマウント関連のライブラリ Node Pluginのフォーマット処理やマウント処理はこのライブラリを使うと楽
Kubernetes CSI Drivers	KubernetesのCSIドライバー一覧 ここにあるドライバーの実装をいくつか読んでおくとあまり迷わず実装できる

- [Kubernetes Container Storage Interface \(CSI\) Documentation](#)
- [CSI Volume Plugins in Kubernetes Design Doc](#)
- [Container Storage Interface \(CSI\) Spec](#)
- [Container Storage Interface: Present and Future](#)
- [What Is The Container Storage Interface \(CSI\)](#)
- [Introducing Container Storage Interface \(CSI\) Alpha for Kubernetes](#)
- [Plugin Watcher Utility - Model 2: Kubelet watches new plugins under a canonical path through inotify \(Preferred one and current implementation\)](#)
- Kubernetes CSI Sidecar Applications
 - [external-provisioner](#)
 - [external-attacher](#)
- [Attach/Detach Controller](#)

Thank you!