

k8s Event

1. Event数据结构

Event 结构体定义在 k8s.io/api/core/v1/types.go, 主要字段:

```
type Event struct {
    metav1.TypeMeta
    metav1.ObjectMeta
    InvolvedObject ObjectReference
    Reason string
    Message string
    Source EventSource
    FirstTimestamp metav1.Time
    LastTimestamp metav1.Time
    Count int32
    Type string
}
```

其中TypeMeta和ObjectMeta是k8s资源必备字段

InvolvedObject表示该Event所属于的资源, 具体结构如下:

```
type ObjectReference struct {
    Kind            string
    Namespace       string
    Name            string
    UID             types.UID
    APIVersion      string
    ResourceVersion string
    FieldPath       string
}
```

通过ObjectReference结构体所包含的字段, 我们可以唯一确认Event属于哪一个资源实例。

Source表示Event的来源, 具体结构如下:

```
type EventSource struct {
    Component string
    Host      string
}
```

其中Component表示该Event由哪个组件产生，Host表示所在的节点名字。可以产生Event的组件主要包含DeploymentController, ReplicasetController, DefaultScheduler, Kubelet。

Type表示Event的类型，分别是EventTypeNormal和EventTypeWarning。

2 Event产生原因

产生Event的原因定义在 `kubernetes/pkg/kubelet/events/events.go` 中，主要包含容器事件，如容器的创建、容器的启动、容器创建失败、容器的启动失败等，Pod事件，如杀掉Pod失败、创建Pod的容器失败、网络没准备好等，镜像事件，如镜像拉取、镜像拉取成功、镜像拉取失败等，kubelet事件，如节点有效、节点失效、节点可调度、节点不可调度等。

3 Event聚合和去重缓存

当ControllerManager或者kubelet产生Event，在广播到各个watcher之前，会进行聚合和去重操作。

在k8s中，如果两个Events除了时间戳以外的其他信息均相同，则他们是相同的Event。如果两个Events除了时间戳和消息(message)以外的其他信息均相同，则他们是相似的Event。k8s会过滤频繁发生的相似Events，对于相同的Events，通过只增加Event的Count属性，只保存一个Event。

k8s 保存生成过的Event的历史记录，是通过LRU Cache实现保存最新的4096个Event。k8s聚合和去重缓存分别保存一个LRU cache。

聚合时使用的key叫做aggregateKey，是使用event.Source、event.InvolvedObject、event.Type和event.Reason构成，而一个Event的唯一标示eventKey就是aggregateKey信息加上event.Message信息。

通过aggregateKey将对10分钟内出现10次的Event进行分组，如果每个Event中的Message不同且总数少于10，则使用eventKey作为

缓存的key，来保存新的Event。如果超过10个，则使用aggregateKey作为缓存的key，达到保存只有message不同的一组相似的Events。

缓存时，判断聚合时返回的key是否在缓存中已经存在，如果存在，使用PUT方法来更新etcd里存放的这条记录，仅更新它的LastTimestamp和Count域。同时还会更新Event Cache里对应记录的Count、LastTimestamp、Name和ResourceVersion。

如果新产生的Event的Key是新的，使用POST方法来在etcd里创建该记录，对该Event的记录同样被加入到Event Cache里。

聚合和去重后，Event就会被广播到各个Watcher，同时Event的TTL默认是1小时。

4 Example

如果创建一个deployment，输入的image不存在发生拉取image失败的错误，重新拉取，我们会收到一个PUT操作，下面打印PUT中old event如下，其中name是pod名字加上时间戳构成：

```
& Event {
  ObjectMeta: k8s.io.apimachinery.pkg.apis.meta.v1.ObjectMeta {
    Name: dm123 - 56688 d9756 - zpcgh .158e84 b503059e37,
    GenerateName: ,
    Namespace: testevt,
    SelfLink: /api/v
1 / namespaces / testevt / events / dm123 - 56688 d9756 - zpcgh .158e84 b503059e37,
    UID: 494 dbac4 - 4 d3a - 11e9 - 8 d02 - 52540058e000,
    ResourceVersion: 132601,
    Generation: 0,
    CreationTimestamp: 2019 - 03 - 23 15: 07: 14 + 0800 CST,
    ...
  },
  InvolvedObject: ObjectReference {
    Kind: Pod,
    Namespace: testevt,
    Name: dm123 - 56688 d9756 - zpcgh,
    UID: 46 f84b53 - 4 d3a - 11e9 - 8 d02 - 52540058e000,
    APIVersion: v1,
    ResourceVersion: 132558,
    FieldPath: spec.containers {
      1
    },
  },
  Reason: Failed,
  Message: Error: ErrImagePull,
  Source: EventSource {
    Component: kubelet,
    Host: k8s - worker - 2,
  },
  FirstTimestamp: 2019 - 03 - 23 15: 07: 14 + 0800 CST,
  LastTimestamp: 2019 - 03 - 23 15: 07: 29 + 0800 CST,
  Count: 2,
  Type: Warning,
  ...
}
```

PUT中新的event如下，如图只更新了Name、ResourceVersion、LastTimestamp和Count字段。

```
& Event {
  ObjectMeta: k8s.io.apimachinery.pkg.apis.meta.v1.ObjectMeta {
    Name: dm123 - 56688 d9756 - zpcgh .158e84 b503059e37,
    GenerateName: ,
    Namespace: testevt,
    SelfLink: /api/v1/namespaces / testevt / events / dm123 - 56688 d9756 - zpcgh .158e84 b503059e37,
    UID: 494 dbac4 - 4 d3a - 11e9 - 8 d02 - 52540058e000,
    ResourceVersion: 132574,
    Generation: 0,
    CreationTimestamp: 2019 - 03 - 23 15: 07: 14 + 0800 CST,
    ...
  },
  InvolvedObject: ObjectReference {
    Kind: Pod,
    Namespace: testevt,
    Name: dm123 - 56688 d9756 - zpcgh,
    UID: 46 f84b53 - 4 d3a - 11e9 - 8 d02 - 52540058e000,
    APIVersion: v1,
    ResourceVersion: 132558,
    FieldPath: spec.containers {
      1
    },
  },
  Reason: Failed,
  Message: Error: ErrImagePull,
  Source: EventSource {
    Component: kubelet,
    Host: k8s - worker - 2,
  },
  FirstTimestamp: 2019 - 03 - 23 15: 07: 14 + 0800 CST,
  LastTimestamp: 2019 - 03 - 23 15: 07: 14 + 0800 CST,
  Count: 1,
  Type: Warning,
  ...
}
```