**Commit容器镜像**

1. **基本原理**

由于容器的重启等操作是很快的，因此没有新加API，而是先把请求发给API server，然后API server作代理讲请求转发给kubelet，kubelet调用docker的接口去重启容器

1. **实现原理**
2. **Pod新加一个ContainerCmd的子资源**

|  |
| --- |
| // ContainerCmdREST implements the Container cmd to a container of Pod  type ContainerCmdREST struct {  store \*etcdgeneric.Etcd  kubeletConn client.ConnectionInfoGetter  }  var \_ = rest.Connecter(&ContainerCmdREST{})  // New creates a new Pod ContainerCmd Resource object  func (r \*ContainerCmdREST) New() runtime.Object {  return &api.Pod{}  }  // Connect returns a handler for the pod ContainerCmd proxy  func (r \*ContainerCmdREST) Connect(ctx api.Context, name string, opts runtime.Object) (rest.ConnectHandler, error) {  CmdOpts, ok := opts.(\*api.PodContainerCmdOptions)  if !ok {  return nil, fmt.Errorf("Invalid options object: %#v", CmdOpts)  }  location, transport, err := pod.ContainerLocation(r.store, r.kubeletConn, ctx, name, CmdOpts)  if err != nil {  return nil, err  }  // upgradeRequired=false, will always use request's query param  return genericrest.NewUpgradeAwareProxyHandler(location, transport, false), nil  }  // NewConnectOptions returns additional parameters  func (r \*ContainerCmdREST) NewConnectOptions() (runtime.Object, bool, string) {  return &api.PodContainerCmdOptions{}, false, ""  }  var CommandMethods = []string{"GET", "POST"}  // ConnectMethods returns the methods supported by ContainerCmd  func (r \*ContainerCmdREST) ConnectMethods() []string {  return CommandMethods  } |

当API server 接收到这个请求后，会将请求转发给kubelet，kubelet接收到以后调用kubelet的接口将容器重启