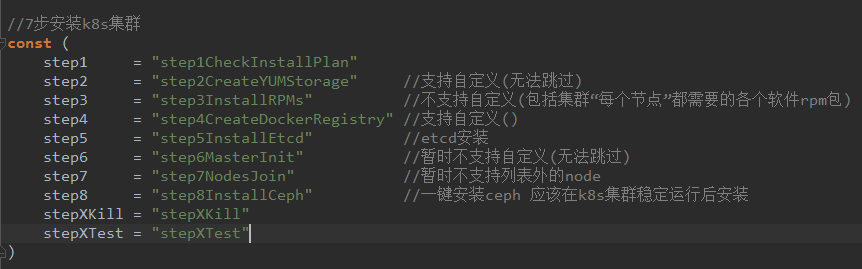
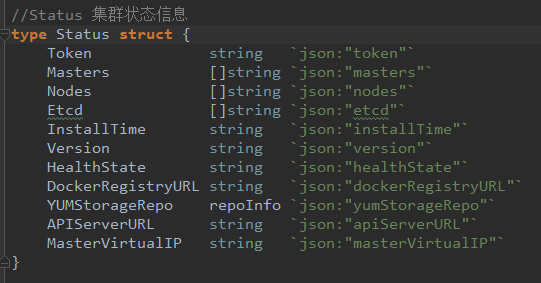
# kubeinstall接口说明文档

用户可以通过POST访问 /cluster/create/{step} 来搭建集群



搭建集群成功后可以GET /cluster查看集群信息 得到如下结构



**提供的POST的参数类型**：统一为如下json,这个结构在最开始的参数检查步骤要用 后续的单个步骤中可能需要只需要填写某几个字段。

//InstallPlan 统一的参数结构  
**type** InstallPlan **struct** {  
 MachineSSHSet **map**[string]kubessh.LoginInfo `json:"machineSSHSet"`  
 EtcdCfg EtcdConf `json:"etcdCfg"`  
 MasterCfg MasterConf `json:"masterCfg"`  
 DockerCfg DockerConf `json:"dockerCfg"`  
 YUMCfg YUMConf `json:"yumCfg"`  
 NodesCfg NodesConf `json:"nodesCfg"`

CephCfg CephConf `json:"cephCfg"`  
}

**type** LoginInfo **struct** {  
 UserName string `json:"userName"`  
 Password string `json:"password"`  
 HostAddr string `json:"hostAddr"`  
 Port int `json:"port"`  
}

//EtcdConf 搭建etcd所需的配置  
**type** EtcdConf **struct** {  
 Hosts []string `json:"hosts"`  
 SSHHosts **map**[string]kubessh.LoginInfo `json:"sshHosts"`  
}

//MasterConf 搭建k8s集群master信息  
**type** MasterConf **struct** {  
 K8sMasterIPSet []string `json:"k8sMasterIPSet"`   
 VirtualIP string `json:"virtualIP"`  
 Port string `json:"port"`  
 NetworkCNI string `json:"networkCNI"`  
 PodNetworkCIDR string `json:"podNetworkCIDR"`  
 AccessIPList **map**[string]kubessh.LoginInfo `json:"accessIPList"`

K8sVersion string `json:"k8sVersion"`  
}

//DockerConf 构建docker仓库所需的信息  
**type** DockerConf **struct** {  
 RegistryIP string `json:"registryIP"`  
 RegistryPort string `json:"registryPort"`  
 UserDockerRegistryURL string `json:"userDockerRegistryURL"`

Storage **map**[string]storagePlan `json:"storage"` // key --- docker ip ;value --- storage plan  
}

//每个docker节点自己拥有一套storage方案  
**type** storagePlan **struct** {  
 DevPath []string `json:"devPath"`  
 VGName string `json:"vgName"`  
 ThinPoolCfg thinPoolConf `json:"thinPoolCfg"`  
}

//thin\_pool 配置  
**type** thinPoolConf **struct** {  
 DataVolumePercent int `json:"dataVolumePercent"` //小于等于95 单位%  
 MetadataVolumePercent int `json:"metadataVolumePercent"` //小于5 单位%  
 LVMVolumeStorageSize int `json:"lvmVolumeStorageSize"` //256~512 单位K  
 AutoExtendCfg autoExtendConf `json:"autoExtendCfg"` //自动扩容配置  
}

//自动扩容配置  
**type** autoExtendConf **struct** {  
 Threshold int `json:"threshold"` //80  
 Percent int `json:"percent"` //20  
}

**type** YUMConf **struct** {  
 IP string `json:"ip"`  
 Repo repoInfo `json:"repo"`  
}

//如果用户自定义了仓库repo 必须保证各个字段的合法性  
**type** repoInfo **struct** {  
 Name string `json:"name"`  
 BaseURL string `json:"baseurl"`  
 Enabled bool `json:"enabled"`  
 GPGCheck bool `json:"gpgcheck"`  
 GPGKey string `json:"gpgkey"`  
}

//NodesConf 节点相关配置  
**type** NodesConf **struct** {  
 NodesIPSet **map**[string]kubessh.LoginInfo `json:"nodesIPSet"`  
}

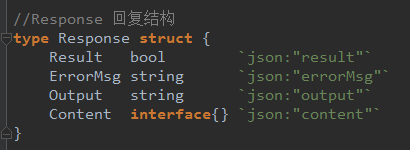
//CephConf 安装ceph所需的用户定制参数  
**type** CephConf **struct** {  
 NodesSSHSet **map**[string]kubessh.LoginInfo `json:"nodesSSHSet"` //ceph集群主机列表  
 RolesSet **map**[string]CephRole `json:"rolesSet"` //角色信息集合  
 GlobalCfg CephGlobalConf `json:"cephGlobalConf"` //ceph 全局配置  
 HostnameSet **map**[string]string `json:"hostnameSet"` //key hostname(cd002) value---(ip 用于查找SSH信息)  
}

//CephGlobalConf ceph 全局配置  
**type** CephGlobalConf **struct** {  
 IsOsdCrushUpdateOnStart bool `json:"isOsdCrushUpdateOnStart"` //false  
 OsdJournalSize int64 `json:"osdJournalSize"` //20480  
 MonPgWarnMaxPerOsd int64 `json:"monPgWarnMaxPerOsd"` //1000  
 MaxOpenFiles int64 `json:"maxOpenFiles"` //131072  
 OsdScrubBeginHour int64 `json:"osdScrubBeginHour"` //1  
 OsdScrubEndHour int64 `json:"osdScrubEndHour"` //7  
}

//CephRole 角色  
**type** CephRole **struct** {  
 Types []string `json:"types"` //角色类型-可以指定多个角色  
 Attribute CephRoleAttribute `json:"attribute"` //角色属性  
}

//CephRoleAttribute ceph角色属性  
**type** CephRoleAttribute **struct** {  
 OSDMediaType string `json:"osdMediaType"` //file or disk  
 OSDMediasPath **map**[string]string `json:"osdMediasPath"` // key---journal value----data 成对出现 且一一对应 不能重复  
}

POST接口返回值统一为：



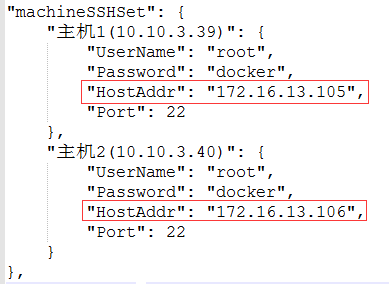
依次为：操作结果、错误信息、屏幕输出、原始请求内容

**Step1:参数检查**

需要将界面上用户选择的**所有参数填**写到InstallPlan结构对应的字段

1. **MachineSSHSet** 包含所有主机的SSH登录信息

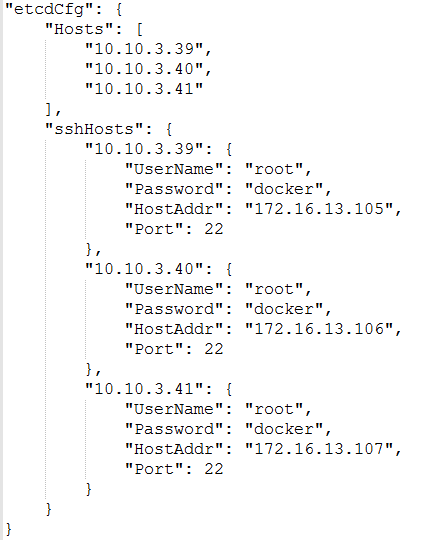
例：



1. **EtcdCfg** 包含创建etcd的相关配置信息 Hosts是etcd搭建在k8s集群内部哪几个IP

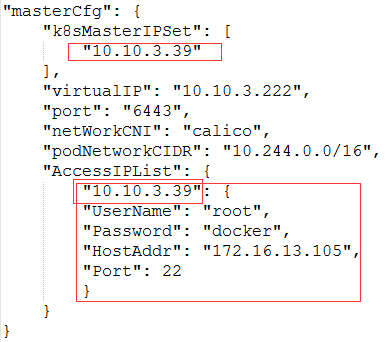
SSHHosts分别是这几个主机的登录信息

例：



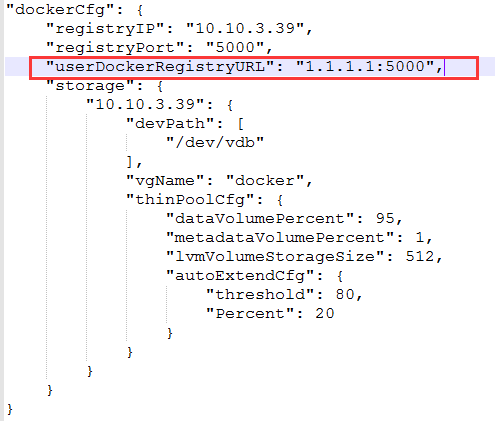
1. **MasterCfg** 主节点的相关配置 k8sMasterIPSet包含了将来在k8s集群中的IP，与SSH登录信息map中的key值应一一对应。**如：10.10.3.39将来是k8s集群的master节点IP，但是此主机对应的SSH登录信息应该是172.16.13.105 root/docker。**

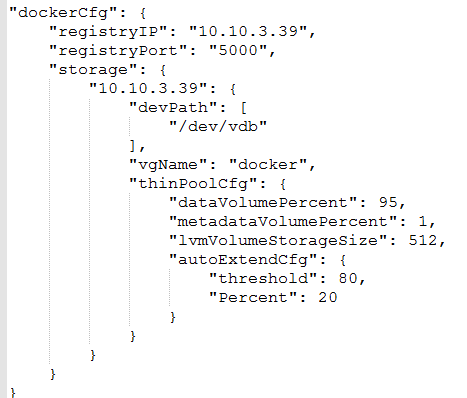
例：



1. **DockerCfg** 安装默认/自定义docker仓库所需的配置

例：如果userDockerRegistryURL字段有值 则认为是自定义的docker仓库，如果为空则使用默认docker仓库; storage字段表示了某个docker节点的存储方案

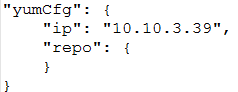




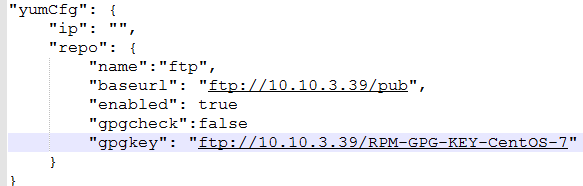
1. **YumCfg** 创建默认/自定义的yum仓库所需的配置

例：在10.10.3.39上创建yum源 其他主机都通过10.10.3.39来访问yum源

//ip字段不为空则在该ip主机上创建默认的yum源

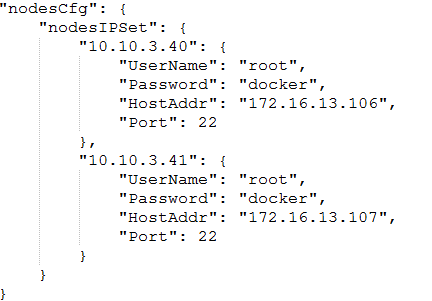


//ip 字段为空则使用用户自定义的yum源



1. **NodesCfg** node节点配置信息 SSH登录信息

例：key是k8s集群ip ，value中的HostAddr是登录IP



1. ceph安装配置检查:

例: 10.10.3.39 40 41 三台主机作为ceph集群

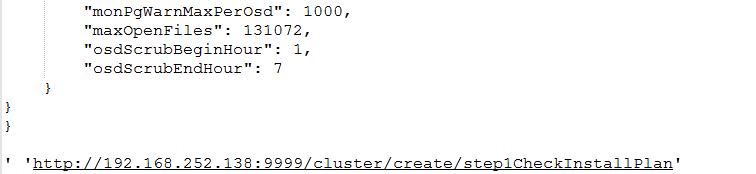
10.10.3.39的角色分别为：osd rgw mds mon;

10.10.3.40的角色分别为: osd rgw mds mon;

10.10.3.41的角色分别为: osd mon;







**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"machineSSHSet": {

"主机1(10.10.3.39)": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"主机2(10.10.3.40)": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

}

},

"masterCfg": {

"ipSet" :[

"10.10.3.39"

],

"virtualIP": "",

"port": "6443"

},

"yumCfg": {

"ip": "",

"repo": {

}

},

"dockerCfg": {

"registryIP": "",

"registryPort": "",

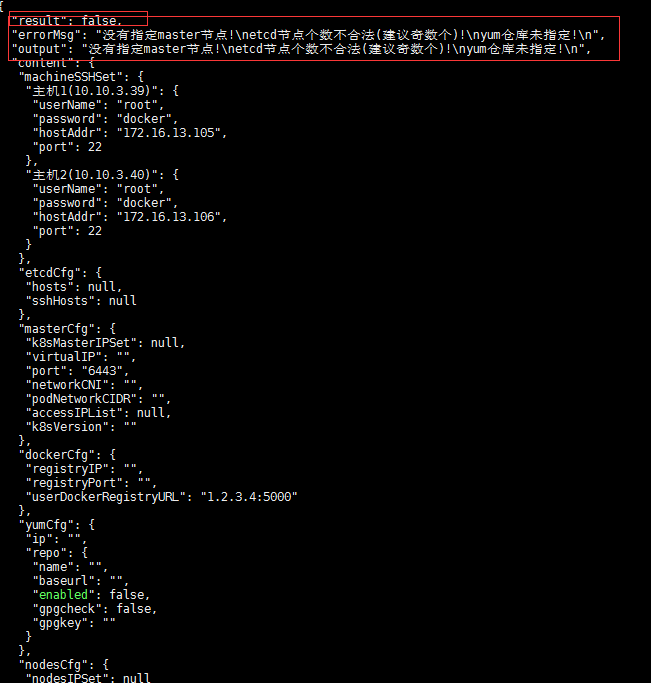
"UserDockerRegistryURL": "1.2.3.4:5000"

}

#####此处可以拓展nodeCfg cephConf#####

}

' 'http://192.168.252.138:9999/cluster/create/step1CheckInstallPlan'



**Step2:创建yum源**

用户应该提供yum源的配置结构和应用该yum源的主机SSH信息(通常是所有主机)

**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

},

"10.10.3.42":{

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.104",

"Port": 22

}

},

"yumCfg": {

"ip": "10.10.3.39",

"repo": {

}

}

}

' 'http://localhost:9999/cluster/create/step2CreateYUMStorage'

**反操作实例**：

####################################反操作 移除yum源(提供所有主机SSH)#######################################

curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '

{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

},

"10.10.3.42":{

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.104",

"Port": 22

}

}

}

' 'http://localhost:9999/cluster/remove/step2CreateYUMStorage'

**Step3:安装必要的RPM包（各个节点均需要安装）**

用户需要提供集群所有主机的SSH登录信息，最终会在所有主机上安装k8s相关组件docker相关组件的rpm包

**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

},

"10.10.3.42":{

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.104",

"Port": 22

}

}

}

' 'http://localhost:9999/cluster/create/step3InstallRPMs'

**反操作实例**：

####################################反操作 移除核心rpm包(提供所有主机SSH)#######################################

curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '

{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

},

"10.10.3.42":{

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.104",

"Port": 22

}

}

}

' 'http://localhost:9999/cluster/remove/step3InstallRPMs'

**Step4:安装docker仓库**

用户需要提供所有主机的SSH信息，docker仓库相关配置，docker存储方案

**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

},

"dockerCfg": {

"registryIP": "10.10.3.39",

"registryPort": "5000",

"storage": {

"10.10.3.39": {

"devPath": [

"/dev/vdb"

],

"vgName": "docker",

"thinPoolCfg": {

"dataVolumePercent": 95,

"metadataVolumePercent": 1,

"lvmVolumeStorageSize": 512,

"autoExtendCfg": {

"threshold": 80,

"Percent": 20

}

}

}

}

}

}

' 'http://localhost:9999/cluster/create/step4CreateDockerRegistry'

**反操作实例**：

####################################反操作 移除docker仓库(提供所有主机SSH)#######################################

curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '

{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

},

"10.10.3.42":{

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.104",

"Port": 22

}

}

}

' 'http://localhost:9999/cluster/remove/step4CreateDockerRegistry'

**Step5:安装etcd(集群 仅支持3个节点)**

用户需要提供etcd配置

**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"etcdCfg": {

"Hosts": [

"10.10.3.39",

"10.10.3.40",

"10.10.3.41"

],

"sshHosts": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

}

}

}

' 'http://localhost:9999/cluster/create/step5InstallEtcd'

**反操作实例**：

####################################反操作 移除etcd(提供之前安装的ETCD集群主机SSH)#######################################

curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '

{

"etcdCfg": {

"Hosts": [

"10.10.3.39",

"10.10.3.40",

"10.10.3.41"

],

"sshHosts": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

}

}

}

' 'http://localhost:9999/cluster/remove/step5InstallEtcd'

**Step6:初始化master节点(目前仅支持单节点)**

用户需要提供master节点的配置

networkCNI:目前仅支持calico和flannel

**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"masterCfg": {

"k8sMasterIPSet": [

"10.10.3.39"

],

"virtualIP": "10.10.3.222",

"port": "6443",

"netWorkCNI": "calico",

"podNetworkCIDR": "10.244.0.0/16",

"AccessIPList": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

}

}

}

}

' 'http://localhost:9999/cluster/create/step6MasterInit'

**反操作实例**：

####################################反操作 移除master(提供之前安装ha的3个节点ssh)#######################################

curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"machineSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

}

}

' 'http://localhost:9999/cluster/remove/step6MasterInit'

**Step7:node节点加入集群(暂时不支持前几部操作参数列表以外的node加入)**

用户需要提供node节点的访问配置

**操作实例**：

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"nodesCfg": {

"nodesIPSet": {

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

}

}

}

' 'http://localhost:9999/cluster/create/step7NodesJoin'

**反操作实例**：

####################################反操作 移除node(提供想要移除的node SSH信息)#######################################

curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"nodesCfg": {

"nodesIPSet": {

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

},

"nodesNameMap": {

"10.10.3.40": "cd003.novalocal",

"10.10.3.41": "cd004.novalocal"

}

}

}

' 'http://localhost:9999/cluster/remove/step7NodesJoin'

**Step8:一键安装ceph集群**

用户需要提供node节点的访问配置,ceph角色相关配置，ceph全局配置等参数

**操作实例**:

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{

"cephCfg": {

"hostnameSet": {

"cd002": "10.10.3.39",

"cd003": "10.10.3.40",

"cd004": "10.10.3.41"

},

"nodesSSHSet": {

"10.10.3.39": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.105",

"Port": 22

},

"10.10.3.40": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.106",

"Port": 22

},

"10.10.3.41": {

"UserName": "root",

"Password": "docker",

"HostAddr": "172.16.13.107",

"Port": 22

}

},

"rolesSet": {

"10.10.3.39": {

"types": [

"osd",

"rgw",

"mds",

"mon"

],

"attribute": {

"osdMediaType": "file",

"osdMediasPath": {

"/home/ceph\_journal\_test1": "/home/ceph\_data\_test1",

"/home/ceph\_journal\_test2": "/home/ceph\_data\_test2"

}

}

},

"10.10.3.40": {

"types": [

"osd",

"rgw",

"mds",

"mon"

],

"attribute": {

"osdMediaType": "file",

"osdMediasPath": {

"/home/ceph\_journal\_test1": "/home/ceph\_data\_test1",

"/home/ceph\_journal\_test2": "/home/ceph\_data\_test2"

}

}

},

"10.10.3.41": {

"types": [

"osd",

"mon"

],

"attribute": {

"osdMediaType": "disk",

"osdMediasPath": {

"/home/ceph\_journal\_test1": "/home/ceph\_data\_test1",

"/home/ceph\_journal\_test2": "/home/ceph\_data\_test2"

}

}

}

},

"GlobalCfg": {

"isOsdCrushUpdateOnStart": false,

"osdJournalSize": 20480,

"monPgWarnMaxPerOsd": 1000,

"maxOpenFiles": 131072,

"osdScrubBeginHour": 1,

"osdScrubEndHour": 7

}

}

}' 'http://localhost:9999/cluster/create/step8InstallCeph'