

算力控制器监控指标前后端接口文档v1.3-2023

年4月11日

红色：新增

黄色：修改

接口列表

```
1 // 获取整个区域的信息
2 ws.Route(ws.GET("/").
3
4 // 获取整个集群的信息
5 ws.Route(ws.GET("/{cluster}").
6
7 // 获取单个集群的所有node信息
8 ws.Route(ws.GET("/{cluster}/nodes").
9
10 // 获取单个集群的单个node信息
11 ws.Route(ws.GET("/{cluster}/nodes/{node}").
12
13 // 获取单个集群的所有pod信息
14 ws.Route(ws.GET("/{cluster}/pods").
15
16 // 获取单个集群某个命名空间的所有pod的指标
17 ws.Route(ws.GET("/{cluster}/namespaces/{namespace}/pods").
18
19 // 获取单个集群某个命名空间的某个pod的指标
20 ws.Route(ws.GET("/{cluster}/namespaces/{namespace}/pods/{pod}").
21
22 // 获取单个集群某个node的所有pod信息
23 ws.Route(ws.GET("/{cluster}/nodes/{node}/pods").
24
25 // 获取单个集群某个node的某个pod信息
26 ws.Route(ws.GET("/{cluster}/nodes/{node}/pods/{pod}").
27
28 // 获取单个集群的所有services信息
29 ws.Route(ws.GET("/{cluster}/services").
30
31 // 获取单个集群的某命名空间下的所有services信息
32 // ws.Route(ws.GET("/namespaces/{namespace}/services").
```

```
33     ws.Route(ws.GET("/{cluster}/namespaces/{namespace}/services").
34
35
36     // 目前前端可忽略，粒度较细
37     // 获取整个集群某个命名空间的某个pod的容器信息
38     ws.Route(ws.GET("/{cluster}/namespaces/{namespace}/pods/{pod}/containers").
39     // 获取整个集群某个命名空间的某个pod的某个容器信息
40     ws.Route(ws.GET("/{cluster}/namespaces/{namespace}/pods/{pod}/containers/{co
41
42
```

一、集群级别监控指标

接口0:区域的元数据

<http://10.21.5.162:9092/CNCC/metadata>

返回示例

GET <http://10.21.5.162:9092/CNCC/metadata>

```
1 {
2   "Zonename": "华北",
3   "ZoneIP": "10.160.100.100",
4   "ZonePort": 9094,
5   "Clusters": {
6     "cluster1": {
7       "Name": "cluster1",
8       "IP": "10.160.10.101",
9       "UUid": "6a620ef5-bdd4-41b2-868c-66e2ca947ef7",
10      "Provider": "华云",
11      "City": "北京"
12    },
13    "cluster2": {
14      "Name": "cluster2",
15      "IP": "10.160.20.101",
16      "UUid": "75bba120-dd09-43eb-8eae-f6c0a5bb08a5",
17      "Provider": "腾云",
18      "City": "南宁"
19    },
20    "cluster3": {
21      "Name": "cluster3",
22      "IP": "10.63.203.127",
23      "UUid": "03decad2-3f93-4480-9e81-1b1196bad7b9",
```

```
24     "Provider": "阿云",
25     "City": "杭州"
26   }
27 }
28 }
```

接口1:所有集群的所有指标/单个指标

TODO

<http://10.21.5.162:9092/CNCC/>

接口名称:

- 集群指标

接口简要描述:

- 获取集群级别指标（所有集群） 集群元数据
(cluster_name/cluster_ip/cluster_provider/cluster_uuid)

请求URL:

- 控制器ip地址:端口/CNCC/
 - 例如: <http://10.21.5.162:9092/CNCC/>

请求方式:

- GET

请求参数:

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称, 指定要返回的指标, 例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时 群CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
start	否	string	范围查询时 的开始时间, Unix 时间格式的字符串, 例如1559347200, 值。与end、step配合使用。
end	否	string	范围查询时 的结束时间, Unix 时间格式的字符串, 例如1559347400, 值。与start、step配合使用。
step	否	string	范围查询时 的时间间隔, 在start-end之间以固定的间隔返回一组监控 式为[0-9]+[smhdwy], 默认为10m (10分钟)。需要同时提供start和 数。
time	否	string	获取某个时间点的监控指标值, 默认为now 。与start+end+step查询结 斥。

返回示例

GET http://10.21.5.162:9092/CNCC/?metrics_filter=cluster_load1|cluster_load5|cluster_load15

```
1  [
2    {
3      "metric_name": "cluster_load1",
4      "data": {
5        "result": [
6          {
7            "metadata": {
8              "cluster_ip": "10.160.10.101",
9              "cluster_name": "cluster1",
10             "cluster_provider": "华云",
11             "cluster_uuid": "22facc81-fe32-404a-bf49-7f33fa8a115d"
12           },
13           "value": [
14             1679991122.602,
15             "0.016666666666666666"
16           ]
17         },
18         {
19           "metadata": {
20             "cluster_ip": "10.160.10.101",
21             "cluster_name": "cluster2",
22             "cluster_provider": "阿外云",
23             "cluster_uuid": "314107d4-6b40-4f8e-9c11-3abcf914d769"
24           },
25           "value": [
26             1679991122.602,
27             "0.016666666666666666"
28           ]
29         }
30       ]
31     }
32   },
33   {
34     ...
35   },
36   {
37     "metric_name": "cluster_load15",
38     "data": {
39       ...
40     }
41   }
```

返回参数说明

参数名	类型	说明
metric_name	string	指标名称
result	-	返回指标结果
value/values	float64[2]	指标样本点(时间戳和实际指标数值)、样本序列（一系列样本点）
value[0]	float64	时间戳
value[1]	float64	指标数值

接口2:单个集群的所有指标/单个指标

接口名称:

- 集群指标

接口简要描述:

- 获取集群级别指标

请求URL:

- 控制器ip地址:端口/CNCC/cluster
 - 例如: <http://10.21.5.162:9092/CNCC/cluster>
 - 后续可能修改为 <http://10.21.5.162:9092/CNCC/{cluster-name}>

请求方式:

- GET

请求参数:

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称，指定要返回的指标，例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时集群CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
start	否	string	范围查询时 的开始时间，Unix 时间格式的字符串，例如1559347200，值。与end、step配合使用。
end	否	string	

			范围查询时 的结束时间，Unix 时间格式的字符串，例如1559347400，值。与start、step配合使用。
step	否	string	范围查询时 的时间间隔，在start-end之间以固定的间隔返回一组监控式为[0-9]+[smhdwy]， 默认为10m （10分钟）。需要同时提供start和数。
time	否	string	获取某个时间点的监控指标值， 默认为now 。与start+end+step查询排斥。

返回示例

GET [http://10.21.5.162:9092/CNCC/cluster?](http://10.21.5.162:9092/CNCC/cluster?metrics_filter=cluster_load1|cluster_load5|cluster_load15)
[metrics_filter=cluster_load1|cluster_load5|cluster_load15](http://10.21.5.162:9092/CNCC/cluster?metrics_filter=cluster_load1|cluster_load5|cluster_load15)

```
1 {
2     "results": [
3         {
4             "metric_name": "cluster_load15",
5             "data": {
6                 "result": [
7                     {
8                         "value": [
9                             1678691585.172,
10                            "0.1575"
11                        ]
12                    }
13                ]
14            }
15        },
16        {
17            "metric_name": "cluster_load1",
18            "data": {
19                "result": [
20                    {
21                        "value": [
22                            1678691585.172,
23                            "0.07999999999999999"
24                        ]
25                    }
26                ]
27            }
28        },
29        {
30            "metric_name": "cluster_load5",
31            "data": {
```

```
32         "result": [  
33             {  
34                 "value": [  
35                     1678691585.172,  
36                     "0.13625"  
37                 ]  
38             }  
39         ]  
40     }  
41 }  
42 ]  
43 }
```

返回参数说明

参数名	类型	说明
metric_name	string	指标名称
result	-	返回指标结果
value/values	float64[2]	指标样本点(时间戳和实际指标数值)、样本序列（一系列样本点）
value[0]	float64	时间戳
value[1]	float64	指标数值

二、节点级别监控指标

接口1:所有节点的所有指标/单个指标

接口名称:

- 所有节点指标

接口简要描述:

- 获取节点级别指标

请求URL:

- 控制器ip地址:端口/CNCC/nodes
 - 例如: <http://10.21.5.162:9092/CNCC/nodes>
 - 后续可能修改为 <http://10.21.5.162:9092/CNCC/{cluster-name}/nodes>

请求方式:

- GET

请求参数：

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称，指定要返回的指标，例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时群CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
start	否	string	范围查询时 的开始时间，Unix 时间格式的字符串，例如1559347200，值。与end、step配合使用。
end	否	string	范围查询时 的结束时间，Unix 时间格式的字符串，例如1559347400，值。与start、step配合使用。
step	否	string	范围查询时 的时间间隔，在start-end之间以固定的间隔返回一组监控式为[0-9]+[smhdwy]， 默认为10m （10分钟）。需要同时提供start和数。
time	否	string	获取某个时间点的监控指标值， 默认为now 。与start+end+step查询斥。

返回示例

GET http://10.21.5.162:9092/CNCC/nodes?metrics_filter=node_load5

```
1 {
2     "results": [
3         {
4             "metric_name": "node_load5",
5             "data": {
6                 "result": [
7                     {
8                         "metadata": {
9                             "node": "k8s-c1m1"
10                        },
11                        "value": [
12                            1678695809.706,
13                            "0.04"
14                        ]
15                    },
16                    {
17                        "metadata": {
18                            "node": "m-k8s-min"
19                        },
```



```
20         "value": [  
21             1678695809.706,  
22             "13.31"  
23         ]  
24     }  
25 ]  
26 }  
27 }  
28 ]  
29 }
```

返回参数说明

参数备注：

metadata:该结构体包含指标的元数据标签（可能的key为namespaces,node,job,pod等）.例如:kube_pod_info{**container**="kube-rbac-proxy-main", ··· **namespace**="xxx", **node**="m-k8s-min", **pod**="xxx", **pod_ip**="xxx", **priority_class**="xxx", **uid**="xxx"}。

参数名	类型	说明
metric_name	string	指标名称
result	-	返回指标结果
metadata	-	在该请求URL下，返回的结果key为node，value为主机名称。
value/values	float64[2]	指标样本点(时间戳和实际指标数值)、样本序列（一系列样本点）
value[0]	float64	该指标值对应的时间戳
value[1]	float64	指标实际数值

接口2:单个节点的所有指标/单个指标

接口名称:

- 某个节点指标

接口简要描述:

- 获取某个节点指标

请求URL:

- 控制器ip地址:端口/CNCC/nodes/{node-name}
 - 例如: <http://10.21.5.162:9092/CNCC/nodes/m-k8s-min>

- 后续可能修改为 <http://10.21.5.162:9092/CNCC/{cluster-name}/nodes/{node-name}>

请求方式:

- GET

请求参数:

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称，指定要返回的指标，例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时群CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
start	否	string	范围查询时 的开始时间，Unix 时间格式的字符串，例如1559347200，值。与end、step配合使用。
end	否	string	范围查询时 的结束时间，Unix 时间格式的字符串，例如1559347400，值。与start、step配合使用。
step	否	string	范围查询时 的时间间隔，在start-end之间以固定的间隔返回一组监控式为[0-9]+[smhdwy]， 默认为10m （10分钟）。需要同时提供start和数。
time	否	string	获取某个时间点的监控指标值， 默认为now 。与start+end+step查询斥。

返回示例

GET http://10.21.5.162:9092/CNCC/nodes/m-k8s-min?metrics_filter=node_load1|node_load15

```
1 {
2     "results": [
3         {
4             "metric_name": "node_load1",
5             "data": {
6                 "result": [
7                     {
8                         "metadata": {
9                             "node": "m-k8s-min"
10                        },
11                        "value": [
12                            1678703700.587,
13                            "7.175"
14                        ]
15                    }
16                ]
17            }
18        }
19    ]
20 }
```

```
17         }
18     },
19     {
20         "metric_name": "node_load15",
21         "data": {
22             "result": [
23                 {
24                     "metadata": {
25                         "node": "m-k8s-min"
26                     },
27                     "value": [
28                         1678703700.587,
29                         "9.525"
30                     ]
31                 }
32             ]
33         }
34     }
35 ]
36 }
```

返回参数说明

参数名	类型	说明
metric_name	string	指标名称
result	-	返回指标结果
metadata	-	在该请求URL下，返回的结果key为node，value为主机名称。
value/values	float64[2]	指标样本点(时间戳和实际指标数值)、样本序列（一系列样本点）
value[0]	float64	该指标值对应的时间戳
value[1]	float64	指标实际数值

三、服务级别监控指标

接口1:所有/单个Pod的所有指标/单个指标

接口名称:

- 所有Pod的指标

接口简要描述:

- 获取所有Pod的指标

请求URL:

- 控制器ip地址:端口/CNCC/pods
 - 例如: <http://10.21.5.162:9092/CNCC/pods>
 - 后续可能修改为 <http://10.21.5.162:9092/CNCC/{cluster-name}/pods>
 - 单个Pod指标查询需使用namespaced_resources_filter参数

请求方式:

- GET

请求参数:

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称，指定要返回的指标，例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时获取CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
namespaced_resources_filter	否	string	过滤获取某个Pod指标信息，等号右边为命名空间名/Pod名称，例如namespxxx=namespaces-name/pod-name
start	否	string	范围查询时 的开始时间，Unix 时间格式的字符串，例如1559347200，值。与end、step配合使用。
end	否	string	范围查询时 的结束时间，Unix 时间格式的字符串，例如1559347400，值。与start、step配合使用。
step	否	string	范围查询时 的时间间隔，在start-end之间以固定的间隔返回一组监控数据，格式为[0-9]+[smhdwy]，默认为 10m （10分钟）。需要同时提供start和end。
time	否	string	获取某个时间点的监控指标值，默认为 now 。与start+end+step查询配合使用。

返回示例-0

查询所有Pods信息

GET <http://10.21.5.162:9092/CNCC/pods>

```
1 {
2     "results": [
```

```

3      {
4          "metric_name": "pod_memory_usage_wo_cache",
5          "data": {
6              "result": [
7                  {
8                      "metadata": {
9                          "namespace": "calico-apiserver",
10                         "node": "m-k8s-min",
11                         "owner_kind": "ReplicaSet",
12                         "owner_name": "calico-apiserver-7f74d87c9b",
13                         "pod": "calico-apiserver-7f74d87c9b-hfgv8"
14                     },
15                     "value": [
16                         1678705695.333,
17                         "31621120"
18                     ]
19                 },
20                 {
21                     "metadata": {
22                         "namespace": "kube-system",
23                         "node": "m-k8s-min",
24                         "owner_kind": "DaemonSet",
25                         "owner_name": "kube-proxy",
26                         "pod": "kube-proxy-pv7df"
27                     },
28                     "value": [
29                         1678705695.333,
30                         "16203776"
31                     ]
32                 },
33                 ...
34                 ...

```

返回示例-1

查询**kube-system**命名空间下名称为**etcd-m-k8s-min**的Pod的网络**pod_net_bytes_received**指标和**pod_net_bytes_transmitted**指标

GET <http://10.21.5.162:9092/CNCC/pods?>

[metrics_filter=pod_net_bytes_received|pod_net_bytes_transmitted&namespaced_resources_filter=kube-system/etcd-m-k8s-min](#)

```

1  {
2      "results": [
3          {
4              "metric_name": "pod_net_bytes_transmitted",

```

```

5      "data": {
6          "result": [
7              {
8                  "metadata": {
9                      "namespace": "kube-system",
10                     "node": "m-k8s-min",
11                     "owner_kind": "Node",
12                     "owner_name": "m-k8s-min",
13                     "pod": "etcd-m-k8s-min"
14                 },
15                 "value": [
16                     1678705301.01,
17                     "11056.533333333333"
18                 ]
19             }
20         ]
21     },
22     {
23         "metric_name": "pod_net_bytes_received",
24         "data": {
25             "result": [
26                 {
27                     "metadata": {
28                         "namespace": "kube-system",
29                         "node": "m-k8s-min",
30                         "owner_kind": "Node",
31                         "owner_name": "m-k8s-min",
32                         "pod": "etcd-m-k8s-min"
33                     },
34                     "value": [
35                         1678705301.01,
36                         "44538.7"
37                     ]
38                 }
39             ]
40         }
41     }
42 ]
43 }
44 }

```

返回参数说明

参数名	类型	说明

metric_name	string	指标名称
result	-	返回指标结果
metadata	-	在该请求URL下，返回的结果各个ke-valuey含义 namespaces :Pod所属命名空间 node :Pod所属主机名称 owner_kind :Pod所属控制器类型(StatefulSet/ReplicaSet/DaemonSet) owner_name :Pod所属控制器名称 pod :当前pod名称
value/values	float64[2]	指标样本点(时间戳和实际指标数值)、样本序列（一系列样本点）
value[0]	float64	该指标值对应的时间戳
value[1]	float64	指标实际数值

接口2:单个集群的所有服务的所有指标/单个指标

TODO

接口名称:

- 所有service的指标

接口简要描述:

- 获取所有service的指标
- 目前共4个指标
 - service_cpu_usage
 - service_memory_usage_wo_cache
 - service_net_bytes_received
 - service_net_bytes_transmitted
 - service 对应的pod数量(metadata)
 - service 所属的namespaces()

请求URL:

- 控制器ip地址:端口/CNCC/{cluster-name}/services
 - <http://10.21.5.162:9092/CNCC/{cluster-name}/services>
 - 例如<http://10.21.5.162:9092/CNCC/cluster1/services>

- 事实上 <http://10.21.5.162:9092/CNCC/{cluster-name}/namespaces/{namespaces-name}/services> 更符合k8s的场景时，即查询某个命名空间下的所有service，因为同一个集群会存在同名的service。（如下图所示，只使用名称无法区分services）

```
ubuntu@k8s-make:~/service-test$ kubectl get svc -A | grep "my-service"
namespace-a      my-service      ClusterIP      10.104.44.65      <none>      80/TCP      4m23s
namespace-b      my-service      ClusterIP      10.110.207.74      <none>      80/TCP      4m23s
ubuntu@k8s-make:~/service-test$ kubectl get svc -n namespace-b
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
my-service ClusterIP      10.110.207.74      <none>      80/TCP      4m28s
ubuntu@k8s-make:~/service-test$ kubectl get svc -n namespace-a
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
my-service ClusterIP      10.104.44.65      <none>      80/TCP      4m32s
ubuntu@k8s-make:~/service-test$ kubectl get svc -A
NAMESPACE      NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
calico-apiserver      calico-api      ClusterIP      10.104.213.125      <none>      443/TCP      13d
calico-system      calico-kube-controllers-metrics      ClusterIP      10.99.33.101      <none>      9094/TCP      13d
calico-system      calico-typha      ClusterIP      10.96.50.47      <none>      5473/TCP      13d
default      kubernetes      ClusterIP      10.96.0.1      <none>      443/TCP      13d
default      my-webapp      NodePort      10.105.235.181      <none>      80:30001/TCP      13d
default      my-webapp2      NodePort      10.103.7.169      <none>      80:30002/TCP      13d
ingress-nginx      ingress-nginx-controller      LoadBalancer      10.103.15.179      <pending>      80:31290/TCP,443:32259/TCP      13d
ingress-nginx      ingress-nginx-controller-admission      ClusterIP      10.107.4.187      <none>      443/TCP      13d
kube-system      kube-dns      ClusterIP      10.96.0.10      <none>      53/UDP,53/TCP,9153/TCP      13d
namespace-a      my-service      ClusterIP      10.104.44.65      <none>      80/TCP      4m36s
namespace-b      my-service      ClusterIP      10.110.207.74      <none>      80/TCP      4m36s
openebs      openebs-nam-cluster-exporter-service      ClusterIP      None      <none>      9100/TCP      13d
openebs      openebs-ndm-node-exporter-service      ClusterIP      None      <none>      9101/TCP      13d
ubuntu@k8s-make:~/service-test$
```

请求方式:

- GET

请求参数:

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称，指定要返回的指标，例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时群CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
start	否	string	范围查询时的开始时间，Unix 时间格式的字符串，例如1559347200，值。与end、step配合使用。
end	否	string	范围查询时的结束时间，Unix 时间格式的字符串，例如1559347400，值。与start、step配合使用。
step	否	string	范围查询时的时间间隔，在start-end之间以固定的间隔返回一组监控式为[0-9]+[smhdwy]，默认为10m（10分钟）。需要同时提供start和数。
time	否	string	获取某个时间点的监控指标值，默认为now。与start+end+step查询斥。

返回示例

查询所有service的信息。

GET <http://10.21.5.162:9092/CNCC/cluster1/services>

GET [http://10.21.5.162:9092/CNCC/cluster1/services?
metrics_filter=service_net_bytes_transmitted](http://10.21.5.162:9092/CNCC/cluster1/services?metrics_filter=service_net_bytes_transmitted)

```
1 {
2   "results": [
3     {
4       "metric_name": "service_net_bytes_transmitted",
5       "data": {
6         "result": [
7           {
8             "metadata": {
9               "namespace": "calico-apiserver",
10              "pod-count": "2",
11              "pods": "calico-apiserver-7d7d4fd5dc-rl4qj,calico-apiserver-7d7d4fd5dc-zx",
12              "service": "calico-api"
13            },
14            "value": [
15              1681122383.293,
16              "7053446"
17            ]
18          },
19          {
20            "metadata": {
21              "namespace": "calico-system",
22              "pod-count": "1",
23              "pods": "calico-kube-controllers-78687bb75f-ppwg8",
24              "service": "calico-kube-controllers-metrics"
25            },
26            "value": [
27              1681122383.293,
28              "212793"
29            ]
30          },
31          {
32            "metadata": {
33              "namespace": "calico-system",
34              "pod-count": "2",
35              "pods": "calico-typha-867f5c6f64-rsflv,calico-typha-867f5c6f64-w8p8b",
36              "service": "calico-typha"
37            },
38            "value": [
39              1681122383.293,
40              "112018972"
41            ]
42          }
43        ]
44      }
45    ]
46  }
```

```
42     },
43     ...
```

—(目前还存在bug，data中应该包含多个service的信息，每个service的**metadata**中包含**service_name:xxx**这一键值对)—

- services的元数据:services 的 **namespace名称,services名称**
 - "metadata": {
 "namespace": "ns-1",
 "service": "my-service1"
 },

返回参数说明

参数名	类型	说明
metric_name	string	指标名称
result	-	返回指标结果
metadata	-	在该请求URL下，返回的结果各个ke-valuey含义 namespaces :service所属命名空间 service :service名称
value/values	float64[2]	指标样本点(时间戳和实际指标数值)、样本序列（一系列样本点）
value[0]	float64	该指标值对应的时间戳
value[1]	float64	指标实际数值

接口3:单个命名空间的所有服务的所有指标/单个指标

TODO

接口名称:

- 某个命名空间的所有service的指标

接口简要描述:

- 获取所有service的指标
- 目前共4个指标
 - service_cpu_usage
 - service_memory_usage_wo_cache

- service_net_bytes_received
- service_net_bytes_transmitted

请求URL:

- <http://10.21.5.162:9092/CNCC/{cluster-name}/namespaces/{namespaces-name}/services>
 - 例如<http://10.21.5.162:9092/CNCC/cluster1/namespaces/monitoring/services> 获取cluster1 集群的monitoring命名空间的所有服务及其指标

请求方式:

- GET

请求参数:

参数名	必选	类型	说明
metrics_filter	否	string	过滤指标名称，指定要返回的指标，例如"metrics_filter=cluster_cpu_usage cluster_disk_size_usage"同时群CPU使用和磁盘使用指标数据。所有可用指标见《监控指标》文档。
start	否	string	范围查询时 的开始时间，Unix 时间格式的字符串，例如1559347200，值。与end、step配合使用。
end	否	string	范围查询时 的结束时间，Unix 时间格式的字符串，例如1559347400，值。与start、step配合使用。
step	否	string	范围查询时 的时间间隔，在start-end之间以固定的间隔返回一组监控式为[0-9]+[smhdwy]， 默认为10m （10分钟）。需要同时提供start和数。
time	否	string	获取某个时间点的监控指标值， 默认为now 。与start+end+step查询斥。

返回示例

GET <http://10.21.5.162:9092/CNCC/cluster1/namespaces/monitoring/services>

```
1 {
2   "results": [
3     {
4       "metric_name": "service_net_bytes_transmitted",
5       "data": {
6         "result": [
7           {
8             "metadata": {
9               "namespace": "monitoring",
```

```
10     "pod-count": "1",
11     "pods": "kube-state-metrics-74964b6cd4-8c6fm",
12     "service": "kube-state-metrics"
13 },
14     "value": [
15         1681122058.152,
16         "85100360"
17     ]
18 },
19     ...
```

接口4:单个命名空间的单个服务的所有指标/单个指标

接口名称:

- 某个service的指标

接口简要描述:

- 获取某个service的指标
- TODO后端过滤服务名称

请求URL:

- 控制器ip地址:端口/CNCC/{cluster-name}/services/{service-name}
 - <http://10.21.5.162:9092/CNCC/{cluster-name}/services/service1>
 - <http://10.21.5.162:9092/CNCC/{cluster-name}/{namespaces-name}/services>