# Appendix 1: Command Guide and Reference Manual

## kubectl apply

- `kubectl apply -f [filename]`:

  If you want to create api object in a declarative way, you can use this command.

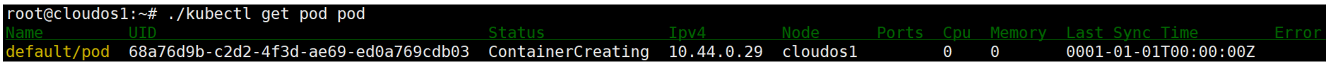
  For example, if you have a pod template yaml file called `pod.yaml` in the current directory, then you can type `kubectl apply -f ./pod.yaml` to create a pod according to your specified template.

## kubectl get

- `kubectl get pod [pod name]`

  This command will show the status of the given pod in a table.

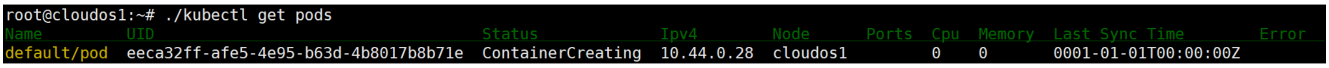
  For example, if you have a pod called `example` in the default namespace, then you can check its status by `kubectl get pod example` (since it's in the default namespace, so the namespace can be omitted. Otherwise, you must type `namespace/name` to specify a pod).



```
root@cloudos1:~# ./kubectl get pod pod
Name         UID                                   Status             Ipv4         Node        Ports  Cpu  Memory  Last Sync Time          Error
default/pod  68a76d9b-c2d2-4f3d-ae69-ed0a769cdb03  ContainerCreating  10.44.0.29   cloudos1           0    0       0001-01-01T00:00:00Z
```

- `kubectl get pods`

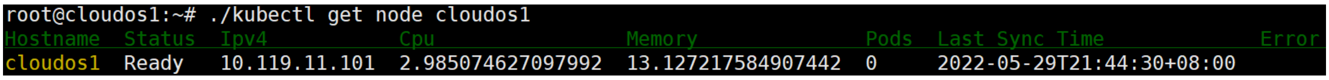
  This command will show the status of all pods(in all namespaces) in a table.



```
root@cloudos1:~# ./kubectl get pods
Name         UID                                   Status             Ipv4         Node        Ports  Cpu  Memory  Last Sync Time          Error
default/pod  eeca32ff-afe5-4e95-b63d-4b8017b8b71e  ContainerCreating  10.44.0.28   cloudos1           0    0       0001-01-01T00:00:00Z
```

- `kubectl get node [node name]`

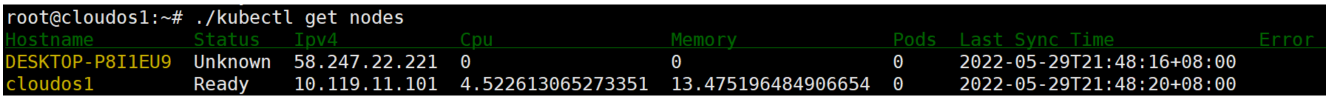
  This command will show the status of the given node in a table.



```
root@cloudos1:~# ./kubectl get node cloudos1
Hostname  Status  Ipv4           Cpu                Memory              Pods  Last Sync Time             Error
cloudos1  Ready   10.119.11.101  2.985074627097992  13.127217584907442  0     2022-05-29T21:44:30+08:00
```

- `kubectl get nodes`

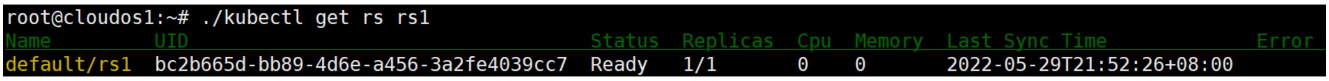
  This command will show the status of all nodes(in all namespaces) in a table.



```
root@cloudos1:~# ./kubectl get nodes
Hostname         Status   Ipv4           Cpu                Memory              Pods  Last Sync Time             Error
DESKTOP-P8I1EU9  Unknown  58.247.22.221  0                  0                   0     2022-05-29T21:48:16+08:00
cloudos1         Ready    10.119.11.101  4.522613065273351  13.475196484906654  0     2022-05-29T21:48:20+08:00
```

- `kubectl get rs [replicaSet name]`

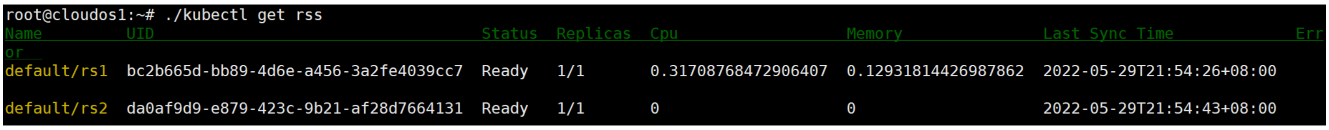
  This command will show the status of the given replicaSet in a table.



```
root@cloudos1:~# ./kubectl get rs rs1
Name         UID                                   Status  Replicas  Cpu  Memory  Last Sync Time             Error
default/rs1  bc2b665d-bb89-4d6e-a456-3a2fe4039cc7  Ready   1/1       0    0       2022-05-29T21:52:26+08:00
```

- `kubectl get rss`

  This command will show the status of all replicaSets(in all namespaces) in a table.

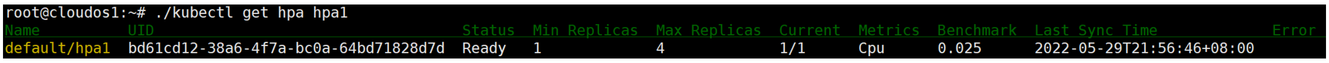


```
root@cloudos1:~# ./kubectl get rss
Name         UID                                   Status  Replicas  Cpu                  Memory               Last Sync Time             Err
or
default/rs1  bc2b665d-bb89-4d6e-a456-3a2fe4039cc7  Ready   1/1       0.31708768472906407  0.12931814426987862  2022-05-29T21:54:26+08:00

default/rs2  da0af9d9-e879-423c-9b21-af28d7664131  Ready   1/1       0                    0                    2022-05-29T21:54:43+08:00
```

- `kubectl get hpa [hpa name]`

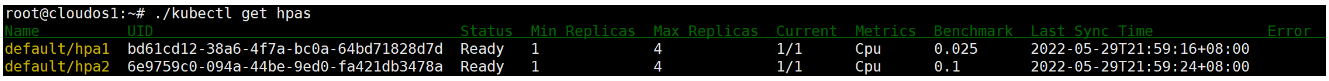
  This command will show the status of the given horizontal pod autoscaler in a table.



```
root@cloudos1:~# ./kubectl get hpa hpa1
Name          UID                                   Status Min Replicas  Max Replicas  Current  Metrics  Benchmark  Last Sync Time                Error
default/hpa1  bd61cd12-38a6-4f7a-bc0a-64bd71828d7d  Ready  1             4             1/1      Cpu      0.025      2022-05-29T21:56:46+08:00
```

- `kubectl get hpas`

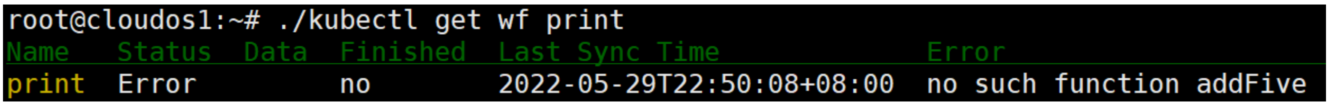
  This command will show the status of all horizontal pod autoscalers(in all namespaces) in a table.



```
root@cloudos1:~# ./kubectl get hpas
Name          UID                                   Status Min Replicas  Max Replicas  Current  Metrics  Benchmark  Last Sync Time                Error
default/hpa1  bd61cd12-38a6-4f7a-bc0a-64bd71828d7d  Ready  1             4             1/1      Cpu      0.025      2022-05-29T21:59:16+08:00
default/hpa2  6e9759c0-094a-44be-9ed0-fa421db3478a  Ready  1             4             1/1      Cpu      0.1        2022-05-29T21:59:24+08:00
```

- `kubectl get wf [workflow name]`

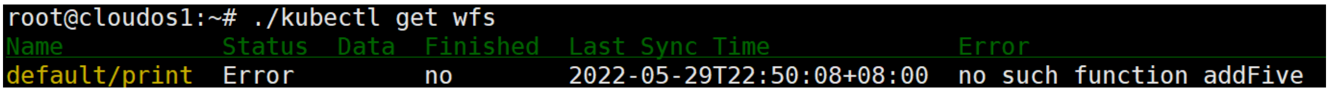
  This command will show the status of the given workflow in a table.



```
root@cloudos1:~# ./kubectl get wf print
Name   Status  Data  Finished  Last Sync Time             Error
print  Error         no        2022-05-29T22:50:08+08:00  no such function addFive
```

- `kubectl get wfs`

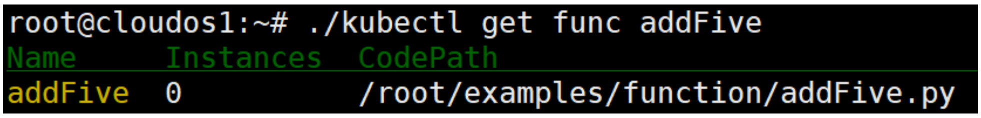
  This command will show the status of all workflows(in all namespaces) in a table.



```
root@cloudos1:~# ./kubectl get wfs
Name           Status  Data  Finished  Last Sync Time             Error
default/print  Error         no        2022-05-29T22:50:08+08:00  no such function addFive
```

- `kubectl get func [function name]`

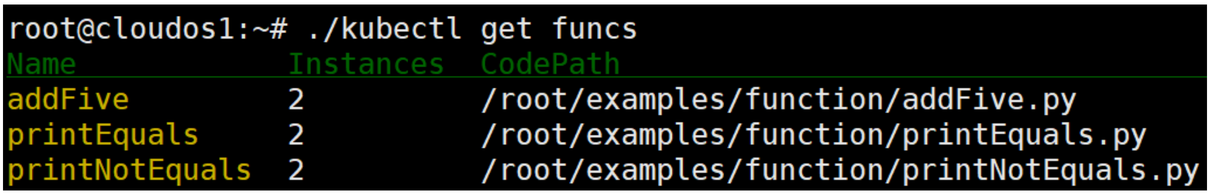
  This command will show the status of the given function in a table.



```
root@cloudos1:~# ./kubectl get func addFive
Name     Instances  CodePath
addFive  0          /root/examples/function/addFive.py
```

- `kubectl get funcs`

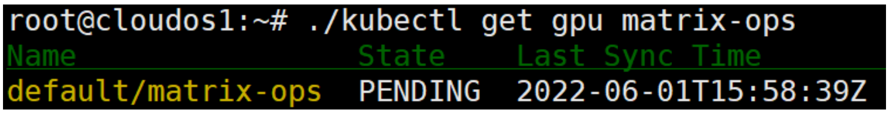
  This command will show the status of all functions in a table.



```
root@cloudos1:~# ./kubectl get funcs
Name            Instances  CodePath
addFive         2          /root/examples/function/addFive.py
printEquals     2          /root/examples/function/printEquals.py
printNotEquals  2          /root/examples/function/printNotEquals.py
```

- `kubectl get gpu [gpu job name]`

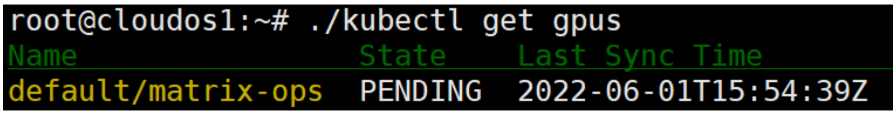
  This command will show the status of the given gpu job in a table.



```
root@cloudos1:~# ./kubectl get gpu matrix-ops
Name                State    Last Sync Time
default/matrix-ops  PENDING  2022-06-01T15:58:39Z
```

- `kubectl get gpus`

  This command will show the status of all gpu jobs in a table.



```
root@cloudos1:~# ./kubectl get gpus
Name                State    Last Sync Time
default/matrix-ops  PENDING  2022-06-01T15:54:39Z
```

- `kubectl get service [service name]`

  This command will show the status of the given service in a table.

- `kubectl get services`

This command will show the status of all services in a table.

- `kubectl get dns [dns name]`

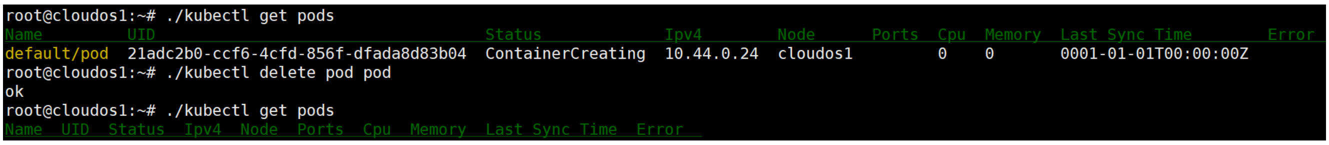
  This command will show the status of the given dns in a table.

- `kubectl get dnses`

  This command will show the status of all dnses in a table.

# kubectl delete

- `kubectl delete [api object type] [name]`



```
root@cloudos1:~# ./kubectl get pods
Name         UID                                   Status             Ipv4         Node       Ports  Cpu   Memory   Last Sync Time         Error
default/pod  21adc2b0-ccf6-4cfd-856f-dfada8d83b04  ContainerCreating  10.44.0.24   cloudos1          0     0        0001-01-01T00:00:00Z
root@cloudos1:~# ./kubectl delete pod pod
ok
root@cloudos1:~# ./kubectl get pods
Name  UID  Status  Ipv4  Node  Ports  Cpu  Memory  Last Sync Time  Error
```

# kubectl autoscale

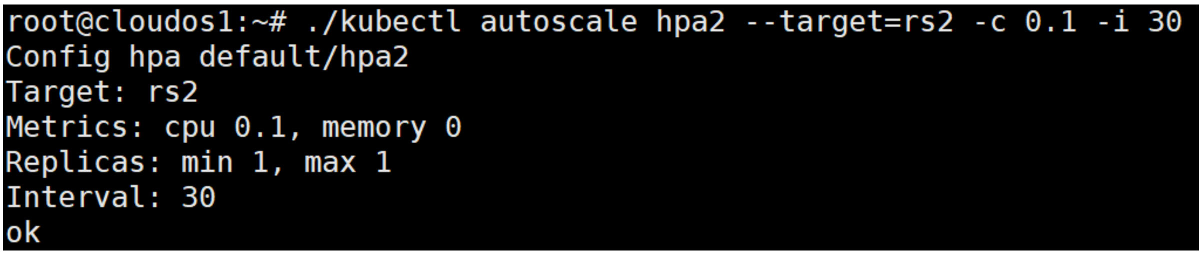
- `kubectl autoscale [hpa name]`

| Parameter | Description | Example |
|---|---|---|
| target | Required. Specify the target of hpa | --target=rs1 |
| min | Optional, default value is 1. Specify the minimum number of replicas | --min=1 |
| max | Optional, default value is 1. Specify the maximum number of replicas | --max=4 |
| c | Optional. Choose CPU utilization as metrics, and the given value as its benchmark | -c 0.5 |
| m | Optional. Choose memory utilization as metrics, and the given value as its benchmark | -m 0.5 |
| i | Optional, the default value is 15. Specify the interval of scaling. | -i 30 |

This command will create a hpa according to the given parameters. Notice that if you specify both CPU and memory metrics,
the hpa controller will choose CPU(CPU has higher priority).

You can also create a hpa in a declarative way: `kubectl apply -f hpa.yaml`.

We recommend you to use `kubectl autoscale` command, for it is more obvious.



```
root@cloudos1:~# ./kubectl autoscale hpa2 --target=rs2 -c 0.1 -i 30
Config hpa default/hpa2
Target: rs2
Metrics: cpu 0.1, memory 0
Replicas: min 1, max 1
Interval: 30
ok
```

# kubectl reset

This command is for test only(of course you can also feel free to use it). It will remove all the K-V pairs stored in `etcd`, thus resetting the status of the whole system.

# kubectl gpu

- `kubectl gpu [gpu job name] -d [directory] -f [file to download]`

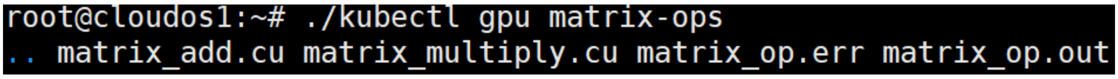
  This command will list or download the files throw the `nginx-fileserver`. Files in blue colors are directories.

  `-d` flag stands for the directory to hold the download files(only works when flag `-f` is used).

  `-f` flag stands for the file you want to download. It will be downloaded through `GET` http request into the directory you have specified in flag `-d`.

  If you don't use any flags, this command will list the files, just like `ls` command.

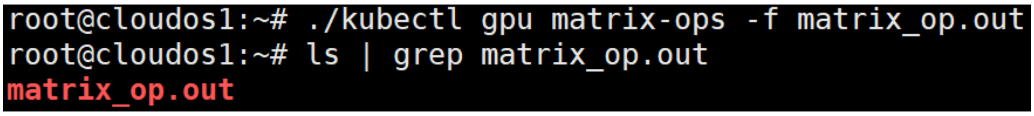
  - List files example

    ```
    root@cloudos1:~# ./kubectl gpu matrix-ops
    .. matrix_add.cu matrix_multiply.cu matrix_op.err matrix_op.out
    ```

  - Download files example

    You can see that the file `matrix-op.out` has been successfully downloaded to the current directory.

    ```
    root@cloudos1:~# ./kubectl gpu matrix-ops -f matrix_op.out
    root@cloudos1:~# ls | grep matrix_op.out
    matrix_op.out
    ```

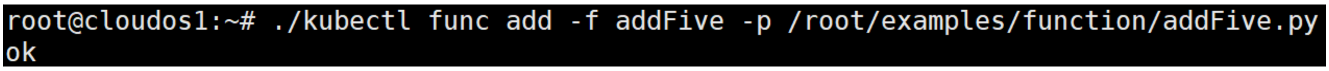
# kubectl func

- `kubectl func add -f [function name] -p [function code path]`

  This command will register a function to the `knative`.

  `-f` flag stands for the name of the function.

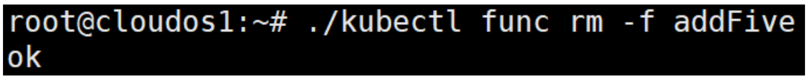
  `-p` flag stands for the file path of the code.

  ```
  root@cloudos1:~# ./kubectl func add -f addFive -p /root/examples/function/addFive.py
  ok
  ```

- `kubectl func rm -f [function name]`

  This command will remove the function and its instances(pods and replicaSet).

  ```
  root@cloudos1:~# ./kubectl func rm -f addFive
  ok
  ```

- `kubectl func update -f [function name] -p [function code path]`

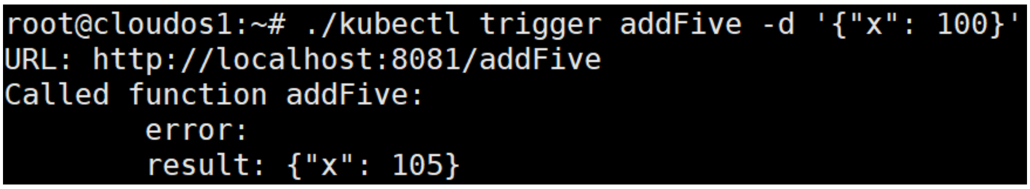
  This command will update a function to the `Knative`.

# kubectl trigger

- `kubectl trigger [function name] -d [function params json]`

  This command will call a serverless function through `http trigger`.

  `-d` flag stands for the JSON form of parameters the function needs.



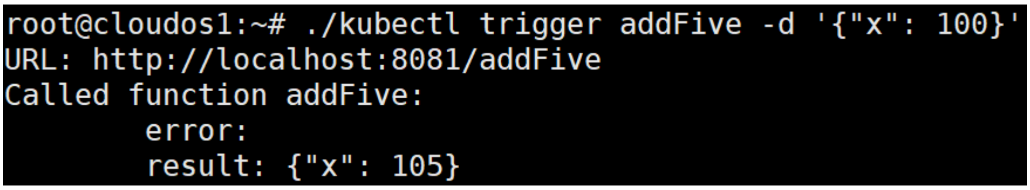
```
root@cloudos1:~# ./kubectl trigger addFive -d '{"x": 100}'
URL: http://localhost:8081/addFive
Called function addFive:
        error:
        result: {"x": 105}
```

# kubectl wf

- `kubectl wf apply -f [workflow.json filepath]`

  This command will create a workflow in a declarative way.

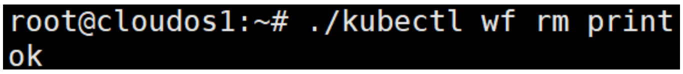
  `-f` flag stands for the `${workflow_name}.json` filepath.



```
root@cloudos1:~# ./kubectl trigger addFive -d '{"x": 100}'
URL: http://localhost:8081/addFive
Called function addFive:
        error:
        result: {"x": 105}
```

- `kubectl wf rm [workflow name]`

  This command will remove the specified workflow.

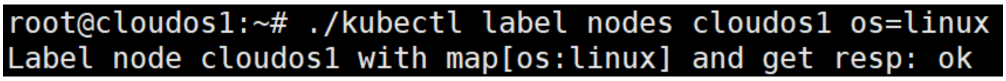


```
root@cloudos1:~# ./kubectl wf rm print
ok
```

# kubectl label

- `kubectl label nodes [node name] [label]`

  This command can label a node with given labels.



```
root@cloudos1:~# ./kubectl label nodes cloudos1 os=linux
Label node cloudos1 with map[os:linux] and get resp: ok
```

# kubectl cfg

- `kubectl cfg sched=[schedule strategy]`

  This command can dynamically change the schedule strategy.

| strategy | Description |
|---|---|
| min-pods | Schedule pod to the node with the minimum number of pods |
| max-pods | Schedule pod to the node with the maximum number of pods |
| min-cpu | Schedule pod to the node with minimum CPU utilization |
| min-mem | Schedule pod to the node with minimum memory utilization |
| random | Schedule pod to a random node |