信誠金融科技股份有限公司
XINCHEN FINTECH CO., LTD.

MultiKF Command Line Manual

馬蒂庫夫使用說明全攻略

Tintin

Machine Learning Platform for Everyone

版本: v1.0.0

修訂日期: 2022/05/10

目錄

1. 簡介

Kubernetes 為一個容器管理平台(Container Orchestration)，從2015年的版本一的發佈至今已經逾十個年頭了，業界的使用與普及率也越來越高，詳細請參考敝司先前的演講內容: 企業界如何使用Kubernetes。

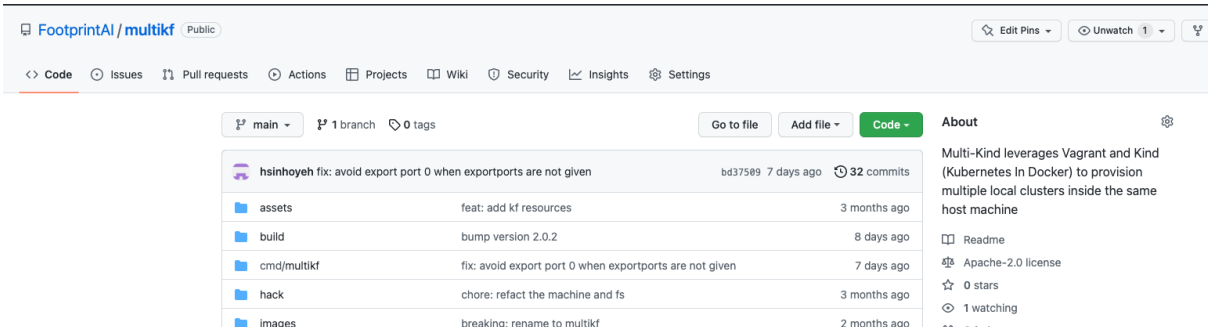
而基於Kubernetes上，Kubeflow想把MLOps(Machine Learning Operations)搬到Kubernetes上使用，讓資料科學家也可以輕易的使用Kuberentes所帶來的便利性。但Kubeflow的安裝上的不便利性以及單一個集群需要相對高的硬體資源(e.g. 4Core+ Cpu 16G+ memory..)使得個人隨身裝置(如筆記型電腦)的負擔相對吃緊。

隨著硬體的資源越來越豐富，一台伺服器等級的機器組成很容易可以擁有數十甚至數百Gbytes的記憶體，若只讓單一台伺服器機器運行單一個Kubernetes群的話相對浪費硬體資源，而且多人操作單一個機器群也容易有資源衝突或矛盾的地方。有鑑於此，馬蒂庫夫(multikf)使用容器化的kubernetes (e.g. kind)將Kubernetes與Kubeflow封裝成一個container image並且配置好該有的設定(如ingress controller等)，讓不同的使用者可以操作伺服器上的個別的容器，這樣既可以不會互相影響，也可以讓安裝過程整體自動化。

馬蒂庫夫(multikf)同時支援Vagrant的虛擬機設定，可用來部署以虛擬機為基礎的VM.

2. 下載與安裝Command Line Tool

- 前往 https://github.com/footprintai/multikf



下載二進位執行檔，我們目前支援Windows/Linux/MacOS等系統

- Windows用戶請使用 *build/multikf.windows.exe*
- Linux用戶請使用 *build/multikf.linux*
- MacOS用戶請使用 *build/multikf.darwin*

用戶端程式會隨著功能新增以及用戶端效能調整而做更新，請不定時上來檢查是否有最新版本。更新的話僅複製二進位執行檔至相對位置即可。

3. 使用規範與注意事項

底下使用說明採用MacOS下的執行模式，若為其他作業系統僅須更換前置執行檔名。其底下範例僅作示意圖示，用戶得就自身使用狀況稍作調整。

使用前請先確認Docker daemon是否有安裝完成，可透過底下指令檢查:

```
docker ps

CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

此為合乎預期訊息。Windows用戶請先確認Docker Desktop是否有安裝以及WSL2是否有啟動，詳細請參考此連結。

4. 參數列表 (--help)

```
./multikf.darwin --help
multikf is a command-line tool which use vagrant and docker to provision Kubernetes and
kubeflow single-node cluster.

Usage:
  multikf [command]

Available Commands:
  add         add a guest machine
  completion  Generate the autocompletion script for the specified shell
  connect     connect
  delete      delete a guest machine
  export      export kubeconfig from a guest machine
  get         get
  help        Help about any command
  list        list guest machines
  version     version of multikf

Flags:
      --alsologtostderr                log to standard error as well as files
      --dir string                     multikf root dir (default ".multikfdir")
  -h, --help                           help for multikf
      --log_backtrace_at traceLocation   when logging hits line file:N, emit a stack trace (default
:0)
      --log_dir string                 If non-empty, write log files in this directory
      --logtostderr                    log to standard error instead of files
      --provisioner string             provisioner, possible value: docker and vagrant (default "docker")
      --stderrthreshold severity       logs at or above this threshold go to stderr (default 2)
  -v, --v Level                        log level for V logs
```

```
    --verbose                   verbose (default: true) (default true)
    --vmodule moduleSpec        comma-separated list of pattern=N settings for file-filtered
logging

Use "multikf [command] --help" for more information about a command
```

5. 查看版本

```
./multikf.darwin version

ver:2.0.2, build time:2022-05-02 16:51:56+08:00,
hashid:e4690f48b74d725abb4205f783a13f0f7c7fd4bc
```

此為該二進位執行檔之版本，可與upstream上的版本做比較。若此版本相對老舊請儘速更新。

6. 虛擬機資料夾

馬蒂庫夫(multikf)預設會在執行黨相對目錄下生成一個資料夾放置各個虛擬機的相關安裝資訊，預設資料夾為**.multikfdir**. 底下透過tree指令顯示該資料夾的檔案配置。

註: 請勿任意修改該檔案內容，除非你知道你正在做哪些事情。

```
tree .multikfdir/
.multikfdir/
├── bin
│   ├── kind
│   └── kubectl
└── node001
```

```
├── kind-config.yaml
├── kubeconfig.yaml
└── kubeflow-manifest-v1.4.1.yaml
```

7. 建立虛擬機

底下範例建立虛擬機(名稱為node001)並使用至多4個Core的Cpu

```
./multikf.linux add node001 —cpus 4
can't found binary from .multikfdir/bin/kind, download from intenrnet...
can't found binary from .multikfdir/bin/kubectl, download from intenrnet...
cmd->[.multikfdir/bin/kind create cluster --config .multikfdir/node001/kind-config.yaml]

cmd->[.multikfdir/bin/kind get kubeconfig --name node001]
this command will keep retry 2 times for every 30s.
cmd->[.multikfdir/bin/kubectl apply -f .multikfdir/node001/kubeflow-manifest-v1.4.1.yaml
--kubeconfig .multikfdir/node001/kubeconfig.yaml]
namespace/auth created
namespace/cert-manager created
namespace/istio-system created
namespace/knative-eventing created
namespace/knative-serving created
namespace/kubeflow created
…
cmd->[.multikfdir/bin/kubectl rollout restart deployment/workflow-controller -n kubeflow
--kubeconfig .multikfdir/node001/kubeconfig.yaml]
deployment.apps/workflow-controller restarted
```

註: 第一次建立時，會從internet上下載對應需要的二進位執行檔，請確保該機器有存取公開網路等能力，並且沒有限制特定阜號(Port number)才能存取(如下載container images時可能會使用阜號5000，請確保該阜號不會被防火牆給限制住)。

預設虛擬機會安裝:

1. Kubernetes V1.20.2
2. Kubeflow V1.4

我們可以透過底下指令檢查其安裝是否完成，一般安裝會需要10-15分鐘才能全部安裝完成。可以透過底下指令查看安裝是否完成

```
.multikfdir/bin/kubectl --kubeconfig=.multikfdir/node001/kubeconfig.yaml get pods -n kubeflow

NAME                                READY   STATUS    RESTARTS   AGE
admission-webhook-deployment-667bd68d94-2rwnl       1/1     Running   0          101s
cache-deployer-deployment-79fdf9c5c9-mzrnp          2/2     Running   1          9m43s
cache-server-5bdf4f4457-ft5q2                       2/2     Running   0          9m45s
centraldashboard-8fc7d8cc-knr4s                     1/1     Running   0          9m43s
jupyter-web-app-deployment-6f744fbc54-w5qvs         1/1     Running   0          9m43s
katib-controller-68c47fbf8b-r8cp8                   1/1     Running   0          101s
katib-db-manager-6c948b6b76-fjtvd                   1/1     Running   1          9m45s
katib-mysql-7894994f88-wqd78                        1/1     Running   0          9m43s
katib-ui-64bb96d5bf-qf9dc                           1/1     Running   0          9m43s
kfserving-controller-manager-0                      2/2     Running   0          99s
kfserving-models-web-app-7884f597cf-dvtsn           2/2     Running   0          9m42s
kubeflow-pipelines-profile-controller-7b947f4748-dsl6b   1/1     Running   0          9m42s
metacontroller-0                                    1/1     Running   0          9m46s
metadata-envoy-deployment-5b4856dd5-t2w8p           1/1     Running   0          9m44s
metadata-grpc-deployment-6b5685488-d94cx            2/2     Running   3          9m44s
metadata-writer-548bd879bb-74zj7                    2/2     Running   0          9m42s
minio-5b65df66c9-kr4kq                              2/2     Running   0          9m43s
ml-pipeline-8c4b99589-fwbl4                         2/2     Running   1          9m41s
ml-pipeline-persistenceagent-d6bdc77bd-dvfq9        2/2     Running   0          9m46s
ml-pipeline-scheduledworkflow-5db54d75c5-dnd7p      2/2     Running   0          9m42s
ml-pipeline-ui-5bd8d6dc84-fvqhd                     2/2     Running   0          9m41s
ml-pipeline-viewer-crd-68fb5f4d58-8lbzq             2/2     Running   1          9m46s
```

```
ml-pipeline-visualizationserver-8476b5c645-z2kfb        2/2    Running  0       9m43s
mpi-operator-5c55d6cb8f-mrjfd                   1/1    Running  0       9m44s
mysql-f7b9b7dd4-65d8j                           2/2    Running  0       9m45s
notebook-controller-deployment-75b4f7b578-gdq6p         1/1    Running  0       9m44s
profiles-deployment-89f7d88b-dgqkk                 2/2    Running  0       9m44s
tensorboard-controller-controller-manager-954b7c544-dkptq  3/3    Running  2       9m40s
tensorboards-web-app-deployment-6ff79b7f44-lbzqh        1/1    Running  0       9m44s
training-operator-7d98f9dd88-rvkq9                 1/1    Running  0       9m41s
volumes-web-app-deployment-8589d664cc-985lf           1/1    Running  0       9m45s
workflow-controller-79dc4bc649-ggz72               2/2    Running  2       8m50s
```

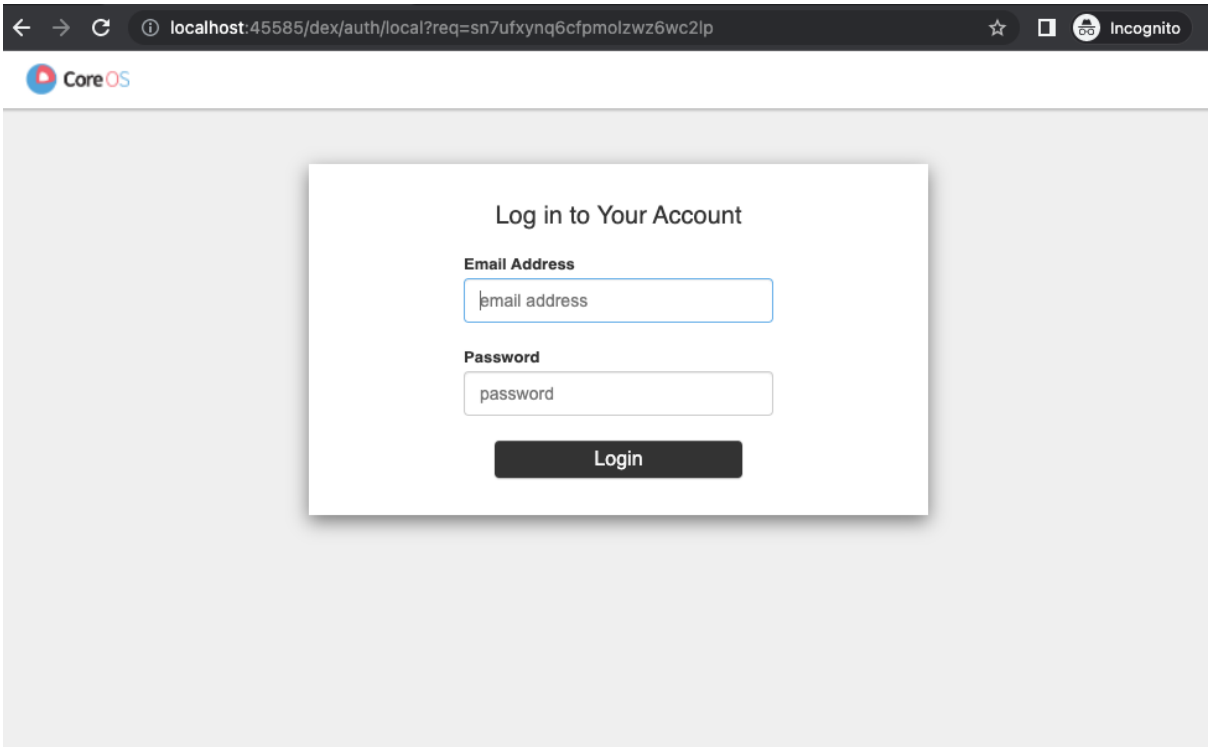
若上述的容器狀態皆為Running時，代表我們已經安裝完成。

8. 連線虛擬機

每台虛擬機皆可以透過底下指令連線

```
./multikf.linux connect kubeflow node001
now you can open http://localhost:45585
cmd->[.multikfdir/bin/kubectl port-forward svc/istio-ingressgateway -n istio-system 45585:80
--kubeconfig .multikfdir/node001/kubeconfig.yaml]
```

之後可以打開瀏覽器並存取指定位置，以此範例我們存取 http://localhost:45585



之後可透過預設帳號密碼: user@example.com // 12341234 登入此系統

若您使用伺服器機器，而該機器放置在遠端機房的話，則可以使用SSH port-forwarding來達成，例如底下指令

```
ssh -i <your-ssh-private-key> username@ssh-hostname -L 45585:0.0.0.0:45585
```

可將本地端的阜號 40085與遠端的40085聯通再一起。

9. 表列虛擬機

您可以透過底下指令查看目前該伺服器機器上有運行哪些虛擬機以及其資源使用狀況:

```
./multikf.linux list

+---------+--------------------+---------+------+---------+------+------------------------+
```

```
| NAME  |       DIR       | STATUS  | GPUS | KUBEAPI | CPUS |       MEMORY       |
+---------+--------------------+---------+------+---------+------+------------------------+
| node001 | .multikfdir/node001 | running |      |         |    4 | 7048.75 Mib/32104.70 Mib |
+---------+--------------------+---------+------+---------+------+------------------------+
```

此範例顯示僅有一台虛擬機，其資料夾放置位置(.multikfdir/node001)以及資源使用狀態(使用4個CPU以及7Gbytes的記憶體）。

10. 刪除虛擬機

您可以透過底下指令刪除虛擬機已釋出更多資源:

```
./multikf.linux delete node001
cmd->[.multikfdir/bin/kind delete cluster --name node001]


./multikf.linux list
cmd->[.multikfdir/bin/kind get clusters]
+------+-----+--------+------+---------+------+--------+
| NAME | DIR | STATUS | GPUS | KUBEAPI | CPUS | MEMORY |
+------+-----+--------+------+---------+------+--------+
+------+-----+--------+------+---------+------+--------+
```

11. 連線GPU虛擬機

您可以透過底下指令來建立起能使用Gpu的虛擬機，此機制使用Docker的Gpu Passthough技術來達成。若使用Vagrant的話則不支援Gpu Passthough機制。

建立Gpu虛擬機之前請先確認伺服器上是否安裝:

1. Gpu driver (可透過nvidia-smi指令來檢查是否驅動程式有安裝成功)
2. Nvidia container toolkit: 可參考此連結進行安裝配置

之後可透過底下指令進行配置

```
./multikf.linux add node002 --use_gpus=1 —cpus 4
```

12. 問題回報

如您在使用上有遇到任何問題或有功能需求，請至回報系統上填寫。我們會盡快處理。