链滴

一个使用 shell 脚本例子

作者：SmiteLi
原文链接：https://ld246.com/article/1582420677896
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

```
#!/bin/bash
#-----------------------------------------------
# This script is for:
# 1. to download the scripts/binaries/images needed for installing a k8s cluster with kubeasz
# 2. to run kubeasz in a container
# @author:   gjmzj
# @usage:    ./easzup
# @repo:     https://github.com/easzlab/kubeasz
# @ref:      https://github.com/kubeasz/dockerfiles
#-----------------------------------------------

set -o nounset
# set -o nounset = set -u
set -o errexit
#  set -e 等价于 set -o errexit
#set -o xtrace

# default version, can be overridden by cmd line options
export DOCKER_VER=18.09.7
export KUBEASZ_VER=2.0.2
export K8S_BIN_VER=v1.15.0
export EXT_BIN_VER=0.3.0
export SYS_PKG_VER=0.3.2

function install_docker() {
  # check if a container runtime is already installed
  systemctl status docker|grep Active|grep -q running && { echo "[WARN] docker is already r
nning."; return 0; }
  systemctl status containerd|grep Active|grep -q running && { echo "[ERROR] containerd is r
nning, unsupported."; exit 1; }

  if [[ "$REGISTRY_MIRROR" == CN ]];then
    DOCKER_URL="https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/static/stable/x86_64/
ocker-${DOCKER_VER}.tgz"
  else
    DOCKER_URL="https://download.docker.com/linux/static/stable/x86_64/docker-${DOCKER
VER}.tgz"
  fi

  mkdir -p /opt/kube/bin /etc/docker /etc/ansible/down
  if [[ -f "/etc/ansible/down/docker-${DOCKER_VER}.tgz" ]];then
    echo "[INFO] docker binaries already existed"
  else
    echo -e "[INFO] \033[33mdownloading docker binaries\033[0m $DOCKER_VER"
    if [[ -e /usr/bin/curl ]];then
      curl -C- -O --retry 3 "$DOCKER_URL" || { echo "[ERROR] downloading docker failed"; exit 1;
}
    else
      wget -c "$DOCKER_URL" || { echo "[ERROR] downloading docker failed"; exit 1; }
    fi
    mv ./docker-${DOCKER_VER}.tgz /etc/ansible/down
  fi

  tar zxf /etc/ansible/down/docker-${DOCKER_VER}.tgz -C /etc/ansible/down && \
```

```
  mv /etc/ansible/down/docker/* /opt/kube/bin && \
  ln -sf /opt/kube/bin/docker /bin/docker

  echo "[INFO] generate docker service file"
  cat > /etc/systemd/system/docker.service << EOF
[Unit]
Description=Docker Application Container Engine
Documentation=http://docs.docker.io
[Service]
Environment="PATH=/opt/kube/bin:/bin:/sbin:/usr/bin:/usr/sbin"
ExecStart=/opt/kube/bin/dockerd
ExecStartPost=/sbin/iptables -I FORWARD -s 0.0.0.0/0 -j ACCEPT
ExecReload=/bin/kill -s HUP \$MAINPID
Restart=on-failure
RestartSec=5
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Delegate=yes
KillMode=process
[Install]
WantedBy=multi-user.target
EOF

  # configuration for dockerd
  echo "[INFO] generate docker config file"
  if [[ "$REGISTRY_MIRROR" == CN ]];then
    echo "[INFO] prepare register mirror for $REGISTRY_MIRROR"
    cat > /etc/docker/daemon.json << EOF
{
  "registry-mirrors": [
    "https://dockerhub.azk8s.cn",
    "https://docker.mirrors.ustc.edu.cn",
    "http://hub-mirror.c.163.com"
  ],
  "max-concurrent-downloads": 10,
  "log-driver": "json-file",
  "log-level": "warn",
  "log-opts": {
    "max-size": "10m",
    "max-file": "3"
    },
  "data-root": "/var/lib/docker"
}
EOF
  else
    echo "[INFO] standard config without registry mirrors"
    cat > /etc/docker/daemon.json << EOF
{
  "max-concurrent-downloads": 10,
  "log-driver": "json-file",
  "log-level": "warn",
  "log-opts": {
    "max-size": "10m",
```

```
    "max-file": "3"
    },
  "data-root": "/var/lib/docker"
}
EOF
  fi

  if [[ -e /etc/centos-release || -e /etc/redhat-release ]]; then
    echo "[INFO] turn off selinux in CentOS/Redhat"
    setenforce 0
    echo "SELINUX=disabled" > /etc/selinux/config
  fi

  echo "[INFO] enable and start docker"
  systemctl enable docker
  systemctl daemon-reload && systemctl restart docker && sleep 8
}

function get_kubeasz() {
  # check if kubeasz already existed
  [[ -d "/etc/ansible/roles/kube-node" ]] && { echo "[WARN] kubeasz already existed"; return 0
 }

  echo -e "[INFO] \033[33mdownloading kubeasz\033[0m $KUBEASZ_VER"
  echo "[INFO] run a temporary container"
  docker run -d --name temp_easz easzlab/kubeasz:${KUBEASZ_VER} || { echo "[ERROR] downl
ad failed."; exit 1; }

  [[ -f "/etc/ansible/down/docker-${DOCKER_VER}.tgz" ]] && mv "/etc/ansible/down/docker-$
DOCKER_VER}.tgz" /tmp

  rm -rf /etc/ansible && \
  echo "[INFO] cp kubeasz code from the temporary container" && \
  docker cp temp_easz:/etc/ansible /etc/ansible && \
  echo "[INFO] stop&remove temporary container" && \
  docker rm -f temp_easz

 [[ -f "/tmp/docker-${DOCKER_VER}.tgz" ]] && mv "/tmp/docker-${DOCKER_VER}.tgz" /etc/an
ible/down
 return 0
}

function get_k8s_bin() {
  [[ -f "/etc/ansible/bin/kubelet" ]] && { echo "[WARN] kubernetes binaries existed"; return 0; }

  echo -e "[INFO] \033[33mdownloading kubernetes\033[0m $K8S_BIN_VER binaries"
  docker pull easzlab/kubeasz-k8s-bin:${K8S_BIN_VER} && \
  echo "[INFO] run a temporary container" && \
  docker run -d --name temp_k8s_bin easzlab/kubeasz-k8s-bin:${K8S_BIN_VER} && \
  echo "[INFO] cp k8s binaries" && \
  docker cp temp_k8s_bin:/k8s /k8s_bin_tmp && \
  mv /k8s_bin_tmp/* /etc/ansible/bin && \
  echo "[INFO] stop&remove temporary container" && \
  docker rm -f temp_k8s_bin && \
```

```
  rm -rf /k8s_bin_tmp
}

function get_ext_bin() {
  [[ -f "/etc/ansible/bin/etcdctl" ]] && { echo "[WARN] extral binaries existed"; return 0; }

  echo -e "[INFO] \033[33mdownloading extral binaries\033[0m kubeasz-ext-bin:$EXT_BIN_VE
"
  docker pull easzlab/kubeasz-ext-bin:${EXT_BIN_VER} && \
  echo "[INFO] run a temporary container" && \
  docker run -d --name temp_ext_bin easzlab/kubeasz-ext-bin:${EXT_BIN_VER} && \
  echo "[INFO] cp extral binaries" && \
  docker cp temp_ext_bin:/extra /extra_bin_tmp && \
  mv /extra_bin_tmp/* /etc/ansible/bin && \
  echo "[INFO] stop&remove temporary container" && \
  docker rm -f temp_ext_bin && \
  rm -rf /extra_bin_tmp
}

function get_sys_pkg() {
  [[ -f "/etc/ansible/down/packages/chrony_xenial.tar.gz" ]] && { echo "[WARN] system packa
es existed"; return 0; }

  echo -e "[INFO] \033[33mdownloading system packages\033[0m kubeasz-sys-pkg:$SYS_PK
_VER"
  docker pull easzlab/kubeasz-sys-pkg:${SYS_PKG_VER} && \
  echo "[INFO] run a temporary container" && \
  docker run -d --name temp_sys_pkg easzlab/kubeasz-sys-pkg:${SYS_PKG_VER} && \
  echo "[INFO] cp system packages" && \
  docker cp temp_sys_pkg:/packages /etc/ansible/down && \
  echo "[INFO] stop&remove temporary container" && \
  docker rm -f temp_sys_pkg
}

function get_offline_image() {
  # images needed by k8s cluster
  calicoVer=v3.4.4
  corednsVer=1.5.0
  dashboardVer=v1.10.1
  flannelVer=v0.11.0-amd64
  heapsterVer=v1.5.4
  metricsVer=v0.3.3
  pauseVer=3.1
  traefikVer=v1.7.12

  imageDir=/etc/ansible/down
  [[ -d "$imageDir" ]] || { echo "[ERROR] $imageDir not existed!"; exit 1; }

  echo -e "[INFO] \033[33mdownloading offline images\033[0m"

  if [[ ! -f "$imageDir/calico_$calicoVer.tar" ]];then
    docker pull calico/cni:${calicoVer} && \
    docker pull calico/kube-controllers:${calicoVer} && \
    docker pull calico/node:${calicoVer} && \
```

```
    docker save -o ${imageDir}/calico_${calicoVer}.tar calico/cni:${calicoVer} calico/kube-controlers:${calicoVer} calico/node:${calicoVer}
  fi
  if [[ ! -f "$imageDir/coredns_$corednsVer.tar" ]];then
    docker pull coredns/coredns:${corednsVer} && \
    docker save -o ${imageDir}/coredns_${corednsVer}.tar coredns/coredns:${corednsVer}
  fi
  if [[ ! -f "$imageDir/dashboard_$dashboardVer.tar" ]];then
    docker pull mirrorgooglecontainers/kubernetes-dashboard-amd64:${dashboardVer} && \
    docker save -o ${imageDir}/dashboard_${dashboardVer}.tar mirrorgooglecontainers/kubernetes-dashboard-amd64:${dashboardVer}
  fi
  if [[ ! -f "$imageDir/flannel_$flannelVer.tar" ]];then
    docker pull easzlab/flannel:${flannelVer} && \
    docker save -o ${imageDir}/flannel_${flannelVer}.tar easzlab/flannel:${flannelVer}
  fi
  if [[ ! -f "$imageDir/heapster_$heapsterVer.tar" ]];then
    docker pull mirrorgooglecontainers/heapster-amd64:${heapsterVer} && \
    docker save -o ${imageDir}/heapster_${heapsterVer}.tar mirrorgooglecontainers/heapster-amd64:${heapsterVer}
  fi
  if [[ ! -f "$imageDir/metrics-server_$metricsVer.tar" ]];then
    docker pull mirrorgooglecontainers/metrics-server-amd64:${metricsVer} && \
    docker save -o ${imageDir}/metrics-server_${metricsVer}.tar mirrorgooglecontainers/metric-server-amd64:${metricsVer}
  fi
  if [[ ! -f "$imageDir/pause_$pauseVer.tar" ]];then
    docker pull mirrorgooglecontainers/pause-amd64:${pauseVer} && \
    docker save -o ${imageDir}/pause_${pauseVer}.tar mirrorgooglecontainers/pause-amd64:$pauseVer}
  fi
  if [[ ! -f "$imageDir/traefik_$traefikVer.tar" ]];then
    docker pull traefik:${traefikVer} && \
    docker save -o ${imageDir}/traefik_${traefikVer}.tar traefik:${traefikVer}
  fi
  if [[ ! -f "$imageDir/kubeasz_$KUBEASZ_VER.tar" ]];then
    docker pull easzlab/kubeasz:${KUBEASZ_VER} && \
    docker save -o ${imageDir}/kubeasz_${KUBEASZ_VER}.tar easzlab/kubeasz:${KUBEASZ_VER}
  fi
}

function download_all() {
  install_docker && \
  get_kubeasz && \
  get_k8s_bin && \
  get_ext_bin && \
  get_sys_pkg && \
  get_offline_image
}

function start_kubeasz_docker() {
  [[ -d "/etc/ansible/roles/kube-node" ]] || { echo "[ERROR] not initialized. try 'easzup -D' first."; exit 1; }
```

```
  # get host's IP
  host_if=$(ip route|grep default|cut -d' ' -f5)
  host_ip=$(ip a|grep "$host_if$"|awk '{print $2}'|cut -d'/' -f1)
  echo "[INFO] get host IP: $host_ip"

  # allow ssh login using key locally
  if [[ ! -e /root/.ssh/id_rsa ]]; then
    echo "[INFO] generate ssh key pair"
    ssh-keygen -t rsa -b 2048 -N '' -f /root/.ssh/id_rsa > /dev/null
    cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
    ssh-keyscan -t ecdsa -H "$host_ip" >> /root/.ssh/known_hosts
  fi

  # create a link '/usr/bin/python' in Ubuntu1604
  if [[ ! -e /usr/bin/python && -e /etc/debian_version ]]; then
    echo "[INFO] create a soft link '/usr/bin/python'"
    ln -s /usr/bin/python3 /usr/bin/python
  fi

  #
  docker load -i /etc/ansible/down/kubeasz_${KUBEASZ_VER}.tar

  # run kubeasz docker container
  echo "[INFO] run kubeasz in a container"
  docker run --detach \
      --name kubeasz \
      --restart always \
      --env HOST_IP="$host_ip" \
      --volume /etc/ansible:/etc/ansible \
      --volume /root/.kube:/root/.kube \
      --volume /root/.ssh/id_rsa:/root/.ssh/id_rsa:ro \
      --volume /root/.ssh/id_rsa.pub:/root/.ssh/id_rsa.pub:ro \
      --volume /root/.ssh/known_hosts:/root/.ssh/known_hosts:ro \
      easzlab/kubeasz:${KUBEASZ_VER} sleep 36000
}

function clean_container() {
 echo "[INFO] clean all running containers"
 docker ps -a|awk 'NR>1{print $1}'|xargs docker rm -f
}

function usage() {
  cat <<EOF
Usage: easzup [options] [args]
  option: -{DdekSz}
    -C         stop&clean all local containers
    -D         download all into /etc/ansible
    -S         start kubeasz in a container
    -d <ver>   set docker-ce version, default "$DOCKER_VER"
    -e <ver>   set kubeasz-ext-bin version, default "$EXT_BIN_VER"
    -k <ver>   set kubeasz-k8s-bin version, default "$K8S_BIN_VER"
    -m <str>   set docker registry mirrors, default "CN"(used in Mainland,China)
    -p <ver>   set kubeasz-sys-pkg version, default "$SYS_PKG_VER"
    -z <ver>   set kubeasz version, default "$KUBEASZ_VER"
```

```
see more at https://github.com/kubeasz/dockerfiles
EOF
}

### Main Lines ##################################################
function main() {
  # check if use bash shell
  readlink /proc/$$/exe|grep -q "dash" && { echo "[ERROR] you should use bash shell, not sh"; exit 1; }
  # check if use with root
  [[ "$EUID" -ne 0 ]] && { echo "[ERROR] you should run this script as root"; exit 1; }

  [[ "$#" -eq 0 ]] && { usage >&2; exit 1; }

  export REGISTRY_MIRROR="CN"
  ACTION=""
  while getopts "CDSd:e:k:m:p:z:" OPTION; do
      case "$OPTION" in
        C)
          ACTION="clean_container"
          ;;
        D)
          ACTION="download_all"
          ;;
        S)
          ACTION="start_kubeasz_docker"
          ;;
        d)
          export DOCKER_VER="$OPTARG"
          ;;
        e)
          export EXT_BIN_VER="$OPTARG"
          ;;
        k)
          export K8S_BIN_VER="$OPTARG"
          ;;
        m)
          export REGISTRY_MIRROR="$OPTARG"
          ;;
        p)
          export SYS_PKG_VER="$OPTARG"
          ;;
        z)
          export KUBEASZ_VER="$OPTARG"
          ;;
        ?)
          usage
          exit 1
          ;;
      esac
  done

  [[ "$ACTION" == "" ]] && { echo "[ERROR] illegal option"; usage; exit 1; }
```

```
  # excute cmd "$ACTION"
  echo -e "[INFO] \033[33mAction begin\033[0m : $ACTION"
  ${ACTION} || { echo -e "[ERROR] \033[31mAction failed\033[0m : $ACTION"; return 1; }
  echo -e "[INFO] \033[32mAction successed\033[0m : $ACTION"
}

main "$@"
```

原文链接：一个使用 shell 脚本例子