# Temperature demo on Raspberry PI部署实例

# 项目概述

## 项目目标：

基于Ubuntu虚拟机、树莓派以及温度传感器实现在云端对边缘侧环境温度的实时采集。

## 项目背景：

温度监测在不同的应用场景下都具备广泛的需求，本项目基于华为开源的边缘计算框架KubeEdge来实现对边缘设备的监测控制，验证功能的可行性，在本实验基础上可以对集群规模进行扩展，实现大规模环境温度实时监测，并根据边缘侧负载、网络情况进行任务的动态调度，在实现云边协同的同时增强边缘节点的离线自治能力，满足多场景应用需求。

## 项目详情：

实验使用kubernetes搭建集群并初始化主节点，并配置golang环境可实现灵活的业务定制，使用keadm工具分别在云端和树莓派节点部署cloudcore服务和edgecore服务，完成云边协同架构的搭建，在云端生成设备实例，边缘侧在业务代码基础上构建镜像，云端通过kube-scheduler服务结合nodeSelector的标签选择机制将温度监测pod调度至树莓派节点，树莓派在本地构建的镜像基础上拉起容器，在云端可实时显示温度状态。

# 本地环境

## 软件包版本

Kernel version: 5.11.0-25-generic

Kubelet version: v1.19.3

KubeEdge version： v1.7.2

Golang version: go1.14.4 linux/amd64 for Ubuntu 21.04，go1.14.4 linux/arm64 for Raspberry PI

Docker version: v20.10.7

## 硬件型号与配置

Raspberry PI 4B

CPU(s): 4

Memory: 8G

Disk: 128G

Vendor ID: ARM

Model name: Cortex-A72

CPU max MHz: 1500.0000

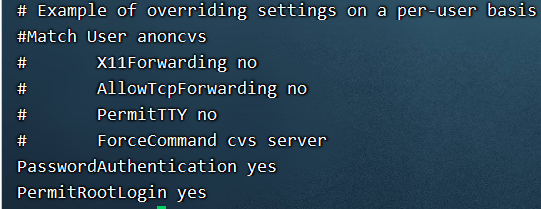
CPU min MHz: 600.0000

温度传感器: DHT11

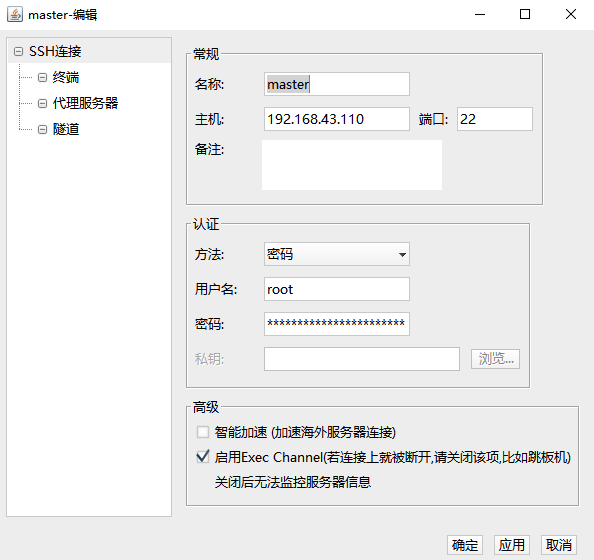
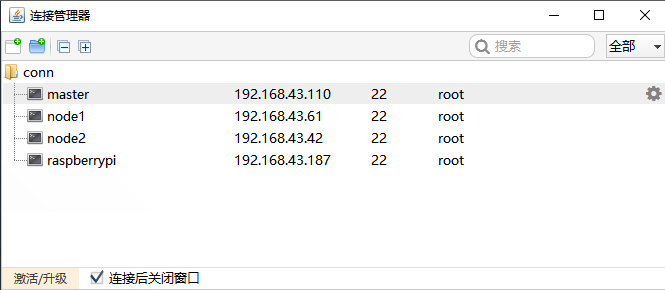
# 依赖工具

实验过程中需安装docker环境并配置golang环境变量，当前KubeEdge支持的kubernets最高版本是v1.19.X以下，建议保持各节点软件包版本一致，本次实验Vmware虚拟机使用的是amd64位架构，树莓派为arm64位架构，需使用对应架构版本的软件。

## 安装Finalshell，配置各节点ssh服务

在安装虚拟机和烧录树莓派系统时可直接选择安装openssh server，配置/etc/ssh/sshd\_config文件，增加配置项PermitRootLogin yes，允许ssh客户端以root权限登录，可以实现文件的灵活传递

Finalshell配置





连接成功后可以看到相关系统信息，后续操作可以全部在Finalshell中实现。

## 安装docker

apt-get install docker.io

建议配置docker镜像加速，使用国内源，本实验采用的是阿里源，可登录阿里云账号获取个人专用镜像加速地址



安装完成后可使用docker version查看，显示版本号说明安装成功。

## 配置golang环境变量

vim /etc/profile文件末尾添加：

# golang envexport

GOROOT=/usr/local/goexport

GOPATH=/data/gopathexport

PATH=$PATH:$GOROOT/bin:$GOPATH/bin

source /etc/profile

mkdir -p /data/gopath && cd /data/gopath

mkdir -p src pkg bin

# 集群搭建

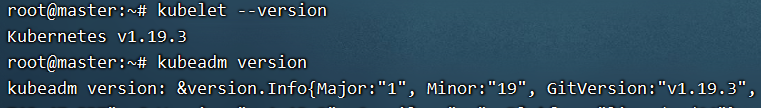
## 安装kubernetes

安装指定版本的kubelet、kubeadm、kubectl三件套：

apt-get install -y kubelet=1.19.3-00 kubeadm=1.19.3-00 kubectl=1.19.3-00

建议使用apt国内源，编辑/etc/apt/sources.list,本实验使用

[http://mirrors.aliyun.com/ubuntu 代替官方源。](http://mirrors.aliyun.com/ubuntu代替官方源。)



可查看版本号说明安装成功。

## 拉取集群所需镜像

由于使用镜像地址拉取镜像，镜像名称与官方名称不同，镜像拉取到本地后需要改成官方的名称，这里使用shell脚本pullk8s.sh写一个循环自动处理：

for  i  in  `kubeadm config images list`;  do

    imageName=${i#k8s.gcr.io/}

    docker pull registry.aliyuncs.com/google\_containers/$imageName

    docker tag registry.aliyuncs.com/google\_containers/$imageName k8s.gcr.io/$imageName

    docker rmi registry.aliyuncs.com/google\_containers/$imageName

done;

## 初始化主节点

kubeadm init \

--apiserver-advertise-address=192.168.43.110 \

--service-cidr=10.96.0.0/12 \

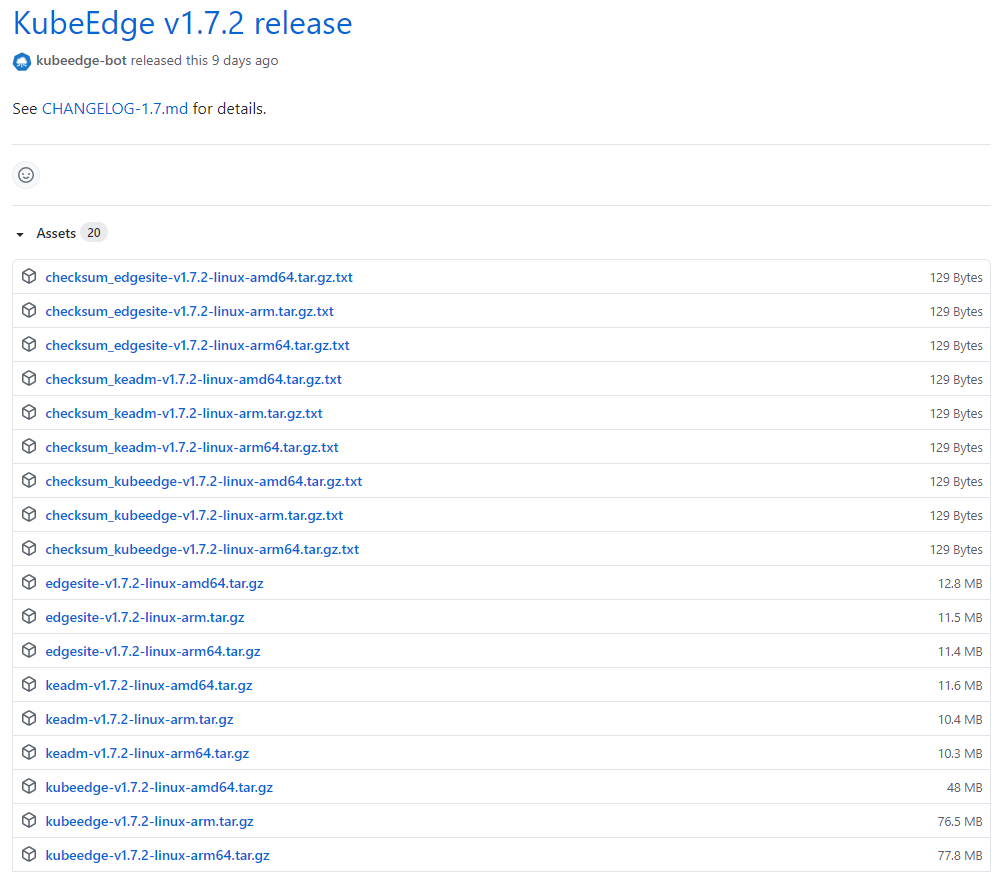
--pod-network-cidr=10.244.0.0/16

初始化完成后，根据提示为集群安装网络：

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

## 部署cloudcore服务

使用华为提供的keadm工具安装云端核心服务



下载相应版到云端节点，解压后到keadm目录下：

./keadm init

成功后为提示cloudcore is running。

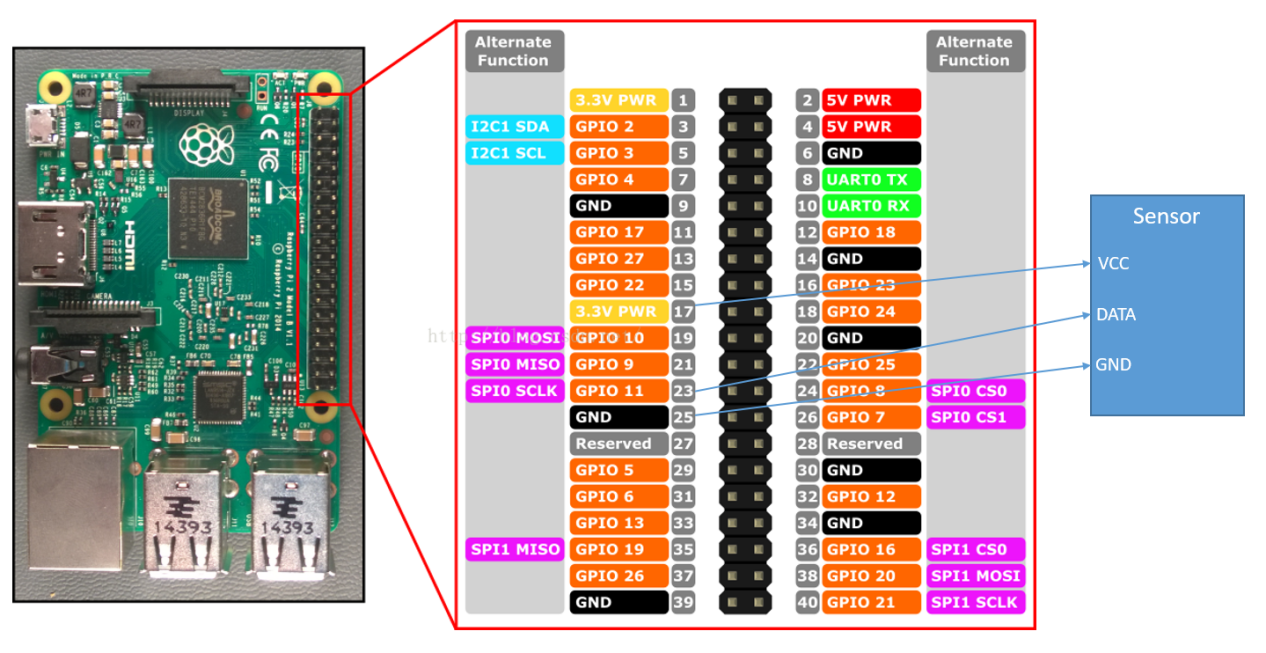
# 业务部署

## 硬件准备

树莓派外接显示器、鼠标和键盘；

温度传感器和杜邦线；

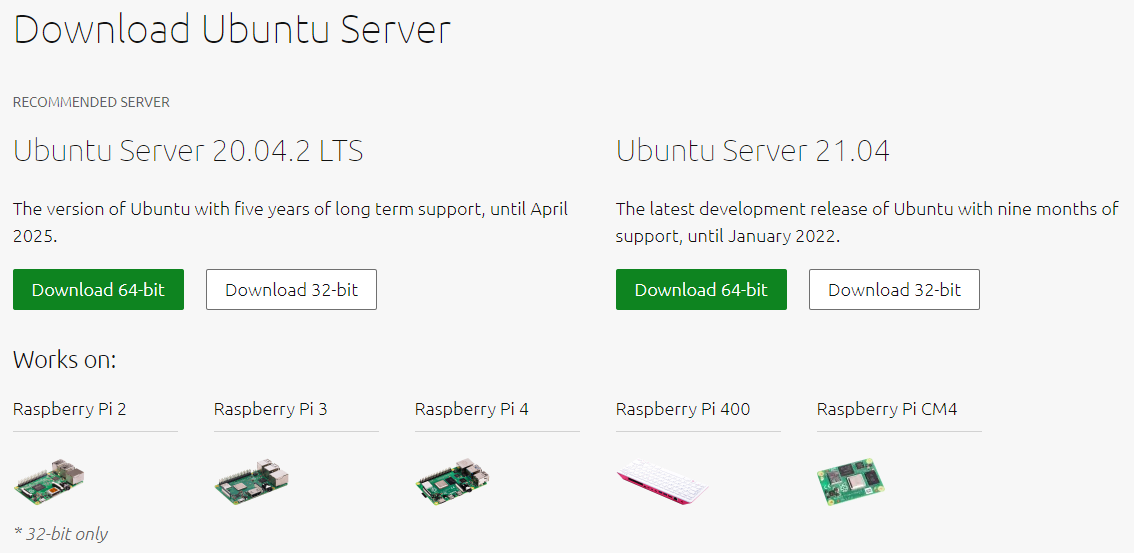




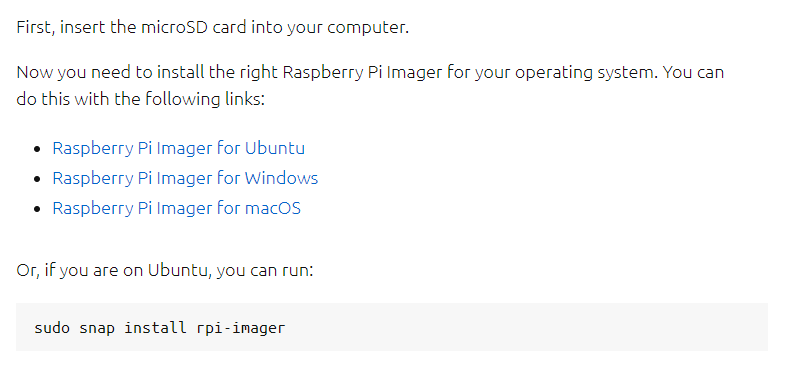
读卡器、SD卡（烧录树莓派操作系统）。

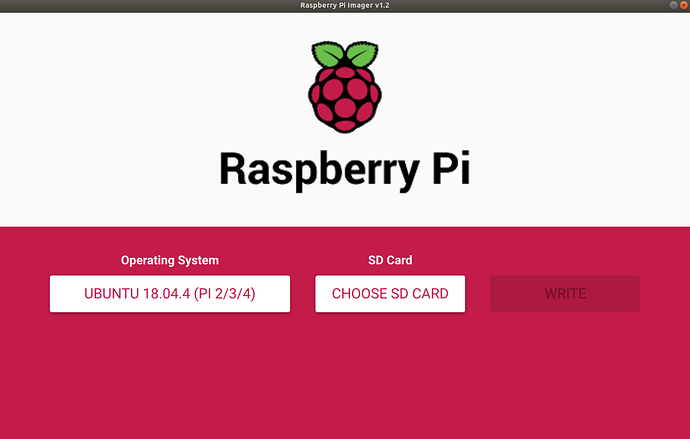
## 系统烧录

在树莓派官网根据硬件型号下载专用镜像



使用官方提供的Raspberry Pi Imager工具烧录镜像至树莓派SD卡，工具可在官方文档中下载





系统安装完成后根据上述方法安装docker及golang环境。

## 部署edgecore服务

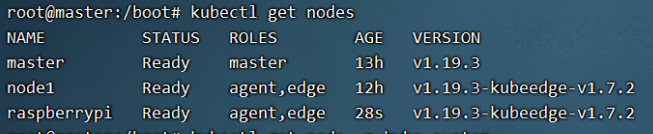
边缘节点要使用keadm工具加入集群，加入时需指定云端节点的token，可在云端节点获取：

./keadm gettoken

边缘端使用下面命令加入集群：

./keadm join --cloudcore-ipport=192.168.43.110:10000 --token=

加入成功后会自动安装edgecore服务，提示edgecore is running说明安装成功。



此时在云端节点可以查看集群状态，如上图所示。

## 创建设备实例

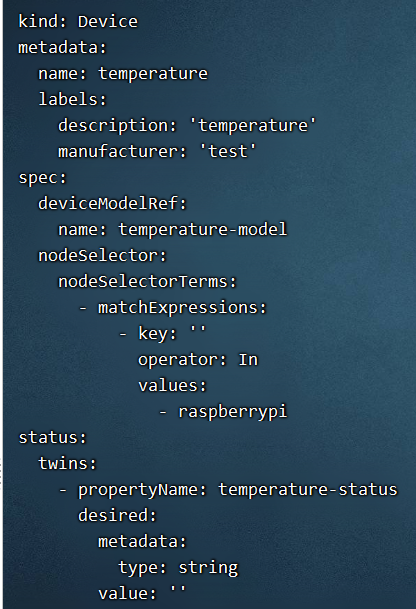
cd $GOPATH/src/github.com/kubeedge/examples/temperature-demo/crds

kubectl apply -f model.yaml

sed -i "s#edge-node#raspberrypi#g" instance.yaml

kubectl apply -f instance.yaml

修改后的instance.yaml文件为



## 构建本地业务Docker镜像

在树莓派节点执行一下操作

cd $GOPATH/src/github.com/kubeedge/examples/temperature-demo

docker build -t kubeedge/temperature-mapper-demo:arm64 .

## 部署业务容器

在云端节点部署业务容器，云端通过kube-scheduler服务结合nodeSelector的标签选择机制将温度监测pod调度至树莓派节点。

cd $GOPATH/src/github.com/kubeedge/examples/temperature-demo/

kubectl create -f deployment.yaml

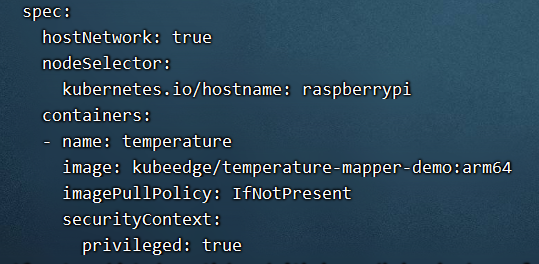
遇到的问题及解决办法：

问题一，云端创建pod后无法将pod调度到指定的树莓派节点，查看日志提示集群中没有符合nodeSelector条件的节点。

解决办法：查看节点标签

kubectl get node --show-labels

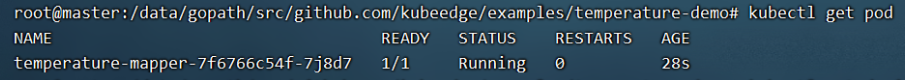
发现树莓派节点标签格式为kubernetes.io/hostname=raspberrypi，deployment.yaml文件中nodeSelector条件为name:raspberryp，两者格式不统一，修改yaml文件后成功将pod调度到边缘节点。修改后的文件为：



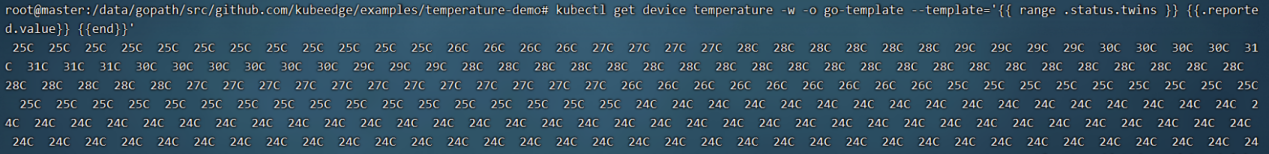
问题二，pod成功调度到边缘节点，但容器无法启动，一直处于Error状态。

解决办法：在镜像构建环节最初是在云端节点构建Docker镜像，通过scp方式传到边缘节点并加载镜像，从云端构建的镜像传到边缘端后无法成功创建容器，pod处于Error状态。

分析后发现问题出在云、边节点架构不同，云端Ubuntu节点为amd64位架构，边缘raspberrypi节点为arm64架构，所以导致镜像在异构节点上无法创建容器。直接在raspberrypi节点创建镜像，并保证镜像名称和tag与deployment.yaml文件中一致，否则创建的pod将无法识别镜像，重新创建镜像后成功拉起容器，问题解决。



## 获取传感器数据

kubectl get device temperature -w -o go-template --template='{{ range .status.twins }} {{.reported.value}} {{end}}'

# 参考文献：

<https://blog.51cto.com/u_10006690/2727626>

<https://learnku.com/articles/29209>

<https://www.freesion.com/article/11691054943/>

<https://blog.csdn.net/weixin_43168190/article/details/107223600>

<https://blog.csdn.net/weixin_43168190/article/details/107227550>

<https://www.yuque.com/sunxiaping/yg511q/gqxqgg>

<https://www.it610.com/article/1281557696918601728.htm>

<https://cloud.tencent.com/developer/article/1661687>

https://github.com/kubeedge/examples/tree/master/temperature-demo

https://blog.csdn.net/Obese\_Tiger/article/details/104802926?utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-18.control&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-18.control

https://mp.weixin.qq.com/s?search\_click\_id=8336705840306415996-1628507703106-740319&sub=&\_\_biz=MzIzNzU5NTYzMA==&mid=2247490514&idx=1&sn=66e8500384a5c2a36c1de23ea3b32f93&chksm=e8c76553dfb0ec458e1fd89dc5b506e2f47277553b564537de017b88103e793a017b41e36ccf&scene=3&subscene=10000&clicktime=1628507703&enterid=1628507703&ascene=0&devicetype=android-29&version=2800093b&nettype=WIFI&abtest\_cookie=AAACAA%3D%3D&lang=zh\_CN&exportkey=ASqKN4QWA8wMMvUa%2B0bx1f0%3D&pass\_ticket=1EpYoHipQ3rxuVT0lY9l4VSM22UXsT4OAvXxoWBm%2B%2Bdlewg%2F%2BalUT%2B7QT1tXKv8S&wx\_header=1