

# Mongoose OS + ESP8266 + AWS

## IoT 制作智能加热器

Phodal Huang

October 24, 2017

目录

步骤 0: 材料准备 .....	3
系统架构 .....	3
构建指南 (Mac/Linux) .....	4

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/mongoose-os-esp8266-aws-iot-build-iot-smart-h>

这里所说的『智能加热器』(Smart Heater)是指:加热器设备能报告当前温度、响应状态请求以及接受开关加热器的命令。

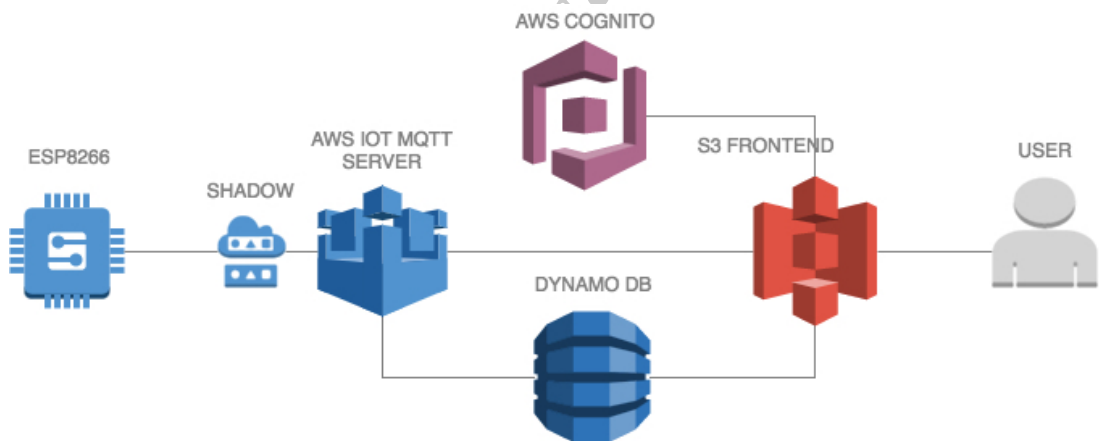
## 步骤 0: 材料准备

在这个玩法里,我们所需要的东西有:

- 硬件,一个 ESP8266 NodeMCU
- Amazon AWS 账户
- Amazon 的 aws 管理实用程序(请参见 <https://aws.amazon.com/cli>)
- Mongoose OS ([www.mongoose-os.com](http://www.mongoose-os.com))

## 系统架构

系统架构如下图所示:



其数据流如下所示:

- 一个设备使用 I2C 温度传感器,并定期向 AWS IoT 发送数据
- 一个 AWS IoT 规则 (rule) 拦截温度数据,并将其存储到 DynamoDB 表中
- 一个设备与其 AWS IoT Shadow 同步
- 使用 OAuth 授权的前端,运行在 AWS S3 上运行;与 Device Shadow 通讯以获取设备状态,或设置设备状态(打开/关闭)

## 构建指南 (Mac/Linux)

首先，您需要创建 **Google** 和者 **Facebook OAuth2** 客户端，以使用户能够登录到加热器应用程序。

对于使用 **Google** 来说：请访问 [Google Cloud Console](#)，单击创建凭据 (**credentials**) -> **OAuth Client ID** -> **Web** 应用程序，然后输入名称，如 “**AWS Heater**”，然后单击 “**Create**”。它将显示您的客户端 **ID** 和密钥 (**secret**)；请复制一下客户端 **ID**，因为它很快会用到。暂不要关闭标签：当您的栈 (**stack**) 被实例化时，您需要返回到此处，并输入 **Authorized JavaScript origin**。

对于使用 **Facebook** 来说：访问 [Facebook Apps](#)，点击 “**Add a New App**”，然后输入应用名称，如 “**My Heater**”，选择一个分类，然后点击 “**Create App ID**”。应用创建完成后，您将在应用程序信息中心看到。暂时不要关闭标签：当您的栈 (**stack**) 被实例化时，您需要返回此处，并输入网站 **URL**。

现在，按下面的步骤执行：

### 1. 克隆代码

```
1 git clone https://github.com/mongoose-os-apps/aws-iot-heater # Clone repo
2 cd aws-iot-heater
```

### 2. 编译、安装、配置 Mongoose OS

```
1 mos flash aws-esp8266 # Install Mongoose OS
2 DEVICE_ID=$(mos config-get device.id) # Get device ID
3 mos put init.js # Copy init.js on the device
4 mos config-set mqtt.enable=true # Enable MQTT
5 mos wifi WIFI_SSID WIFI_PASSWORD # Setup WiFi
6 mos aws-iot-setup --aws-iot-policy=mos-default # Provision on AWS IoT
```

### 3. 安装 helpers/cloudformation-helpers

```
1 npm --prefix helpers/cloudformation-helpers install
   helpers/cloudformation-helpers
```

### 4. 创建 AWS S3 Bucket

```
1 # We'll also need to create a separate S3 bucket for helper functions:
2 aws s3 mb s3://my-cf-helpers
```

### 5. 获取端口地址

```
1 # Get the endpoint address for your AWS account, you'll need to provide it
  as
2 # a parameter for your stack:AWS_IOT_ENDPOINT=$(aws iot describe-endpoint
  --output text)
```

6.“打包”模板。打包包含 **helper** 的源代码，从本地机器复制到上面创建的 **s3 bucket**，并适当地调整模板。这一切都在一步完成：

```
1 aws cloudformation package \
2   --template-file aws_iot_heater_template.yaml \
3   --s3-bucket my-cf-helpers \
4   --output-template-file packaged_template.yaml
5
6 # Generate Oauth IDs on Google and Facebook
7 GOOGLE_CLIENT_ID=YOUR_GOOGLE_CLIENT_ID
8 FACEBOOK_CLIENT_ID=$YOUR_FACEBOOK_CLIENT_ID
9
10 STACK_NAME=my-heater
11
12 # The command above has created a new template file: packaged-template.yaml.
13 # Now, instantiate AWS stack using this template. Feel free to choose
  another
14 # stack name.
15 aws cloudformation create-stack \
16   --stack-name $STACK_NAME \
17   --parameters \
18     ParameterKey=DeviceID,ParameterValue=$DEVICE_ID \
19     ParameterKey=EndpointAddress,ParameterValue=$AWS_IOT_ENDPOINT \
20     ParameterKey=GoogleClientID,ParameterValue=$GOOGLE_CLIENT_ID \
21     ParameterKey=FacebookClientID,ParameterValue=$FACEBOOK_CLIENT_ID \
22   --capabilities CAPABILITY_IAM \
23   --template-body file:///packaged_template.yaml
24
25 # Wait until the stack creation is completed (it may take a few minutes).
26 aws cloudformation wait stack-create-complete --stack-name my-heater
27
28 # Alternatively, you can use the web UI to check the status and read event
```

```
29 # details: https://console.aws.amazon.com/cloudformation/home
30 # When the stack is created, get the name of the created S3 bucket:aws
    cloudformation describe-stacks --stack-name $STACK_NAME
31
32 # look for the following:
33 # ...
34 # {
35 #     "Description": "Name of the newly created S3 bucket",
36 #     "OutputKey": "S3BucketName",
37 #     "OutputValue": "S3_BUCKET_NAME"
38 # },
39 # {
40 #     "Description": "URL of the s3 bucket",
41 #     "OutputKey": "S3BucketWebsite",
42 #     "OutputValue": "APP_URL"
43 # }
44 # ...
45
46 S3_BUCKET_NAME=GENERATED_S3_BUCKET_NAME
47 APP_URL=GENERATED_APP_URL
48
49
50 # $$S3_BUCKET_NAME is the name of the bucket, and $APP_URL is the URL at
51 # which your files can be accessed.
52 #
53 # Copy the actual value of "$APP_URL", and then enter it in the Google
    and/or
54 # Facebook app settings: For Google: go back to the Google Console, and add
    the
55 # URL as an Authorized JavaScript origin. For Facebook: go back to the
    app's
56 # dashboard, click "Settings" in the sidebar, then click "Add Platform" at
    the
57 # bottom, select "Website", and enter Site URL.
58 #
59 # Then, copy the actual value of "$S3_BUCKET_NAME" (from the describe-stacks
```

```
60 # output), and use it to put two files on the S3 bucket:
61 aws s3 cp bucket/index.html s3://$S3_BUCKET_NAME --acl public-readaws s3 cp
    bucket/index.js s3://$S3_BUCKET_NAME --acl public-read
62
63 # Download two files of Cognito SDK, and also put them on the S3 bucket:
64 curl -O
    https://raw.githubusercontent.com/aws/amazon-cognito-identity-js/master/dist/aws-cognito-
65 curl -O
    https://raw.githubusercontent.com/aws/amazon-cognito-identity-js/master/dist/amazon-cogni
66 aws s3 cp aws-cognito-sdk.min.js s3://$S3_BUCKET_NAME --acl public-read
67 aws s3 cp amazon-cognito-identity.min.js s3://$S3_BUCKET_NAME --acl
    public-read
68
69 # Now, navigate to the index page of your app $APP_URL.
```

您将看到从设备上传的温度，当前加热器状态（开/关）、开关的最新图表。只有经认证和授权的用户才能切换加热器；点击“**Sign in with Google**”。

注意：如果它“抱怨”（报错）重定向的 **URI** 不匹配，只需要等待几分钟：**Google Console** 中的设置可能需要一些时间才能生效。

如果您尝试切换加热器状态，您将收到消息说：你没有授权这样做。现在，您需要授权您的用户，来管理加热器。

为此，需要导航到 **AWS Cognito Console**，单击“**Manage Federated Identities**”，选择“**identity\_pool\_for\_DEVICE\_ID**”，单击“**Edit identity pool**”，展开“**Authentication providers**”，点击“**Google+**”标签，并在“**Authenticated role**”区域，将“**Use default role**”“**Choose role with rules**”。

在这里，你可以使用你想要的任何规则。例如，为了授权一些特定用户，您可以指定 **Claim** 为：“**email**”，匹配类型为“**Equals**”，值：“**addr@domain.com**”，并选择一个角色“**my-heater-myHeaterAdminRole-XXXXXXXXXX**”。

之后，您可以从加热器应用程序中注销，登录后，切换加热器将导致更改状态。

原文链接：<https://www.wandianshenme.com/play/mongoose-os-esp8266-aws-iot-build-iot-smart-h>