

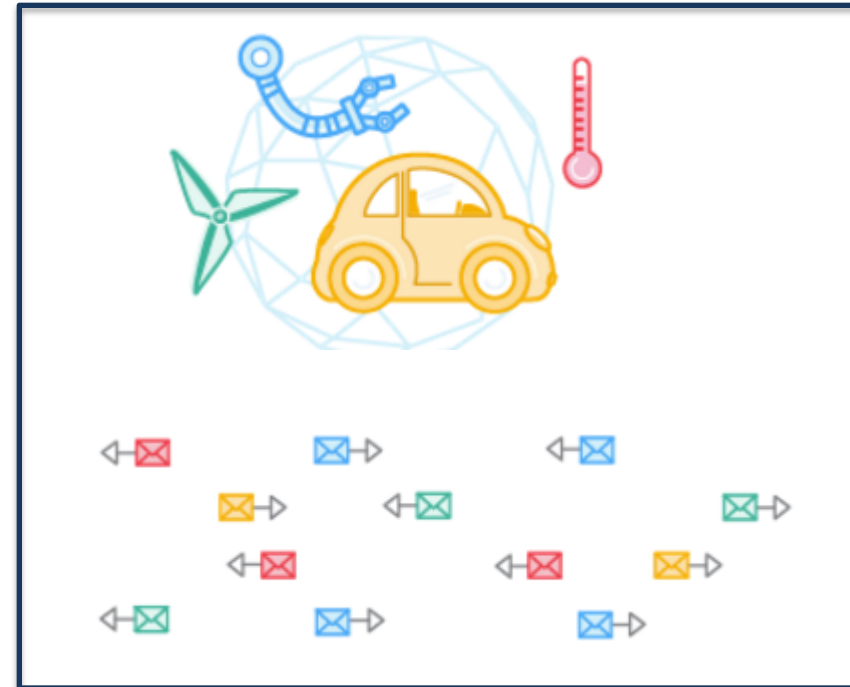
Internet-of-Things

➤ Things (Devices)

- ❑ Many of them
 - Different Types
 - Isolated Systems

❑ Data and Command

- Sensing the world
- Give Response

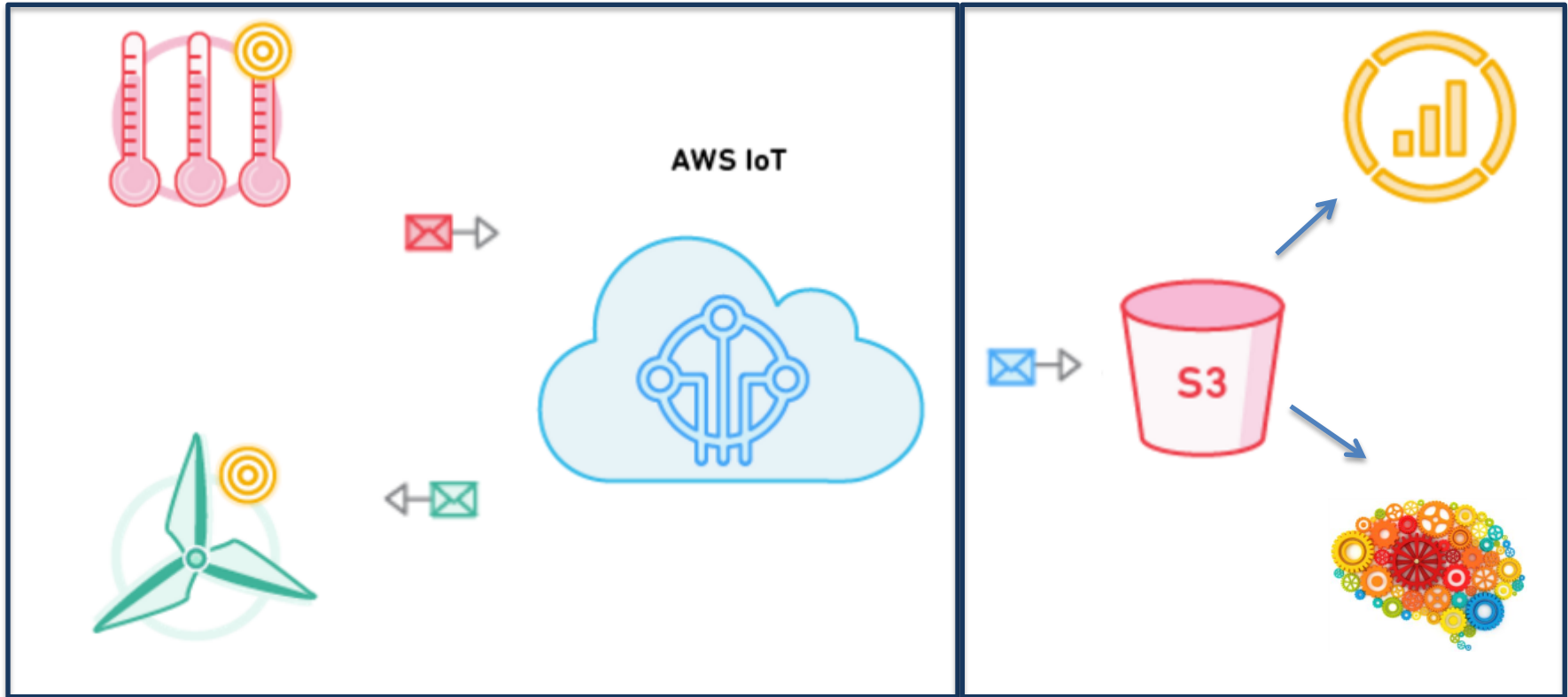


❑ Challenge

- United: Connected + Communication
- Smart: Data Analytics + Strategy



Solution: AWS IoT



United: Connect + Communication

Smart: Other Cloud Service
Data Storage
Machine Learning

Tutorial: Hello AWS IoT!

Random
Integer
[1, 100]

LED

>50 : ON
<=50: OFF



Publish

Subscribe

AWS IoT



Forward

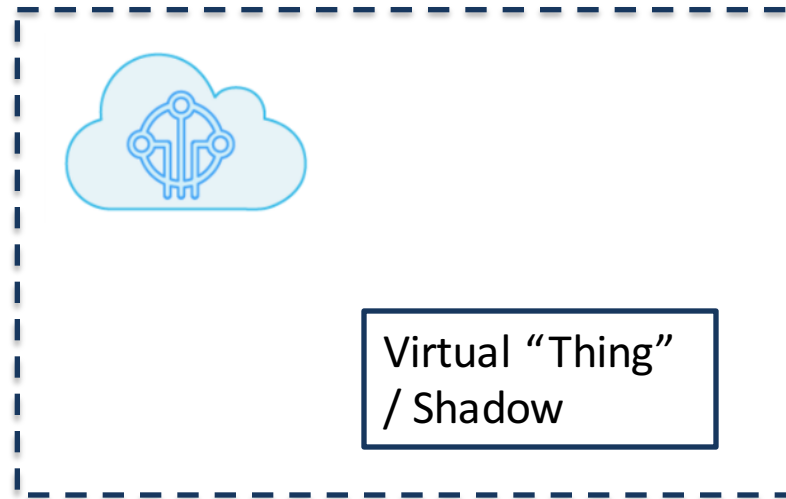


Amazon SNS



Step 1: Create a Virtual "Thing"

AWS IoT



Get into AWS Manage Console

- Create your own AWS account
- Sign In IoT Manage Console
 - ❑ <https://aws.amazon.com/iot/>

AWS services

Find a service by name (for example, EC2, S3, Elastic Beanstalk).

▼ All services



Compute

EC2
EC2 Container Service
Lightsail
Elastic Beanstalk
Lambda
Batch



Developer Tools

CodeCommit
CodeBuild
CodeDeploy
CodePipeline



Management Tools

CloudWatch



Internet of Things

AWS IoT



Game Development

GameLift



Mobile Services

Create a thing

- 1. AWS IoT Menu
 - ❑ Registry
 - Things → Create
- 2. Give a name

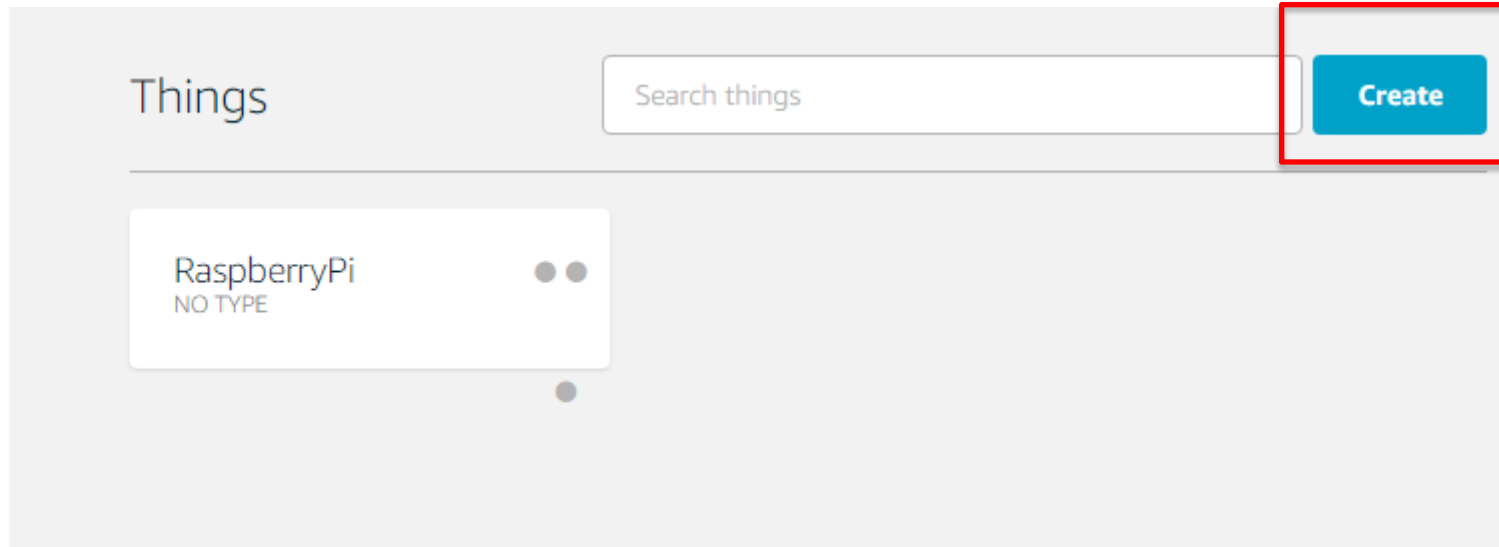


Dashboard

Connect

Registry
Things

Types



Basic Interact: Publish

➤ Using Embedded **MQTT Client** to Test

Security

Rules

Test

Publish

Specify a topic and a message to publish.

Publish to topic

```

1 {
2   "state":
3   {
4     "reported":
5     {
6       "Info": "Hello AWS IoT!"
7     }
8   }
9 }

```

➤ Check the Things Shadow

Shadow

Interact

Activity

```
arn:aws:iot:us-west-2:401317363811:thing/Test
```

Shadow Document

Last update: Jan 17, 2017 10:24:27 PM -0600

Shadow state:

```

1 {
2   "reported": {
3     "Hello": "Hello AWS IoT"
4   }
5 }

```

Basic Interact: Subscribe

-  Dashboard
-  Connect
-  Registry
-  Security
-  Rules
-  Test

Subscriptions

Subscribe to a topic

- \$aws/things/Test/shadow/update/accepted ×

Subscribe to a topic

Devices publish MQTT messages on topics. Subscribe to a topic to view the messages published to it.

Subscription topic

\$aws/things/Test/shadow/update/accepted

Max message capture ?

Quality of Service ? 0 1

Subscribe to topic

MQTT client ?

Connected as **iotconsole-1484713476597-4** ▾

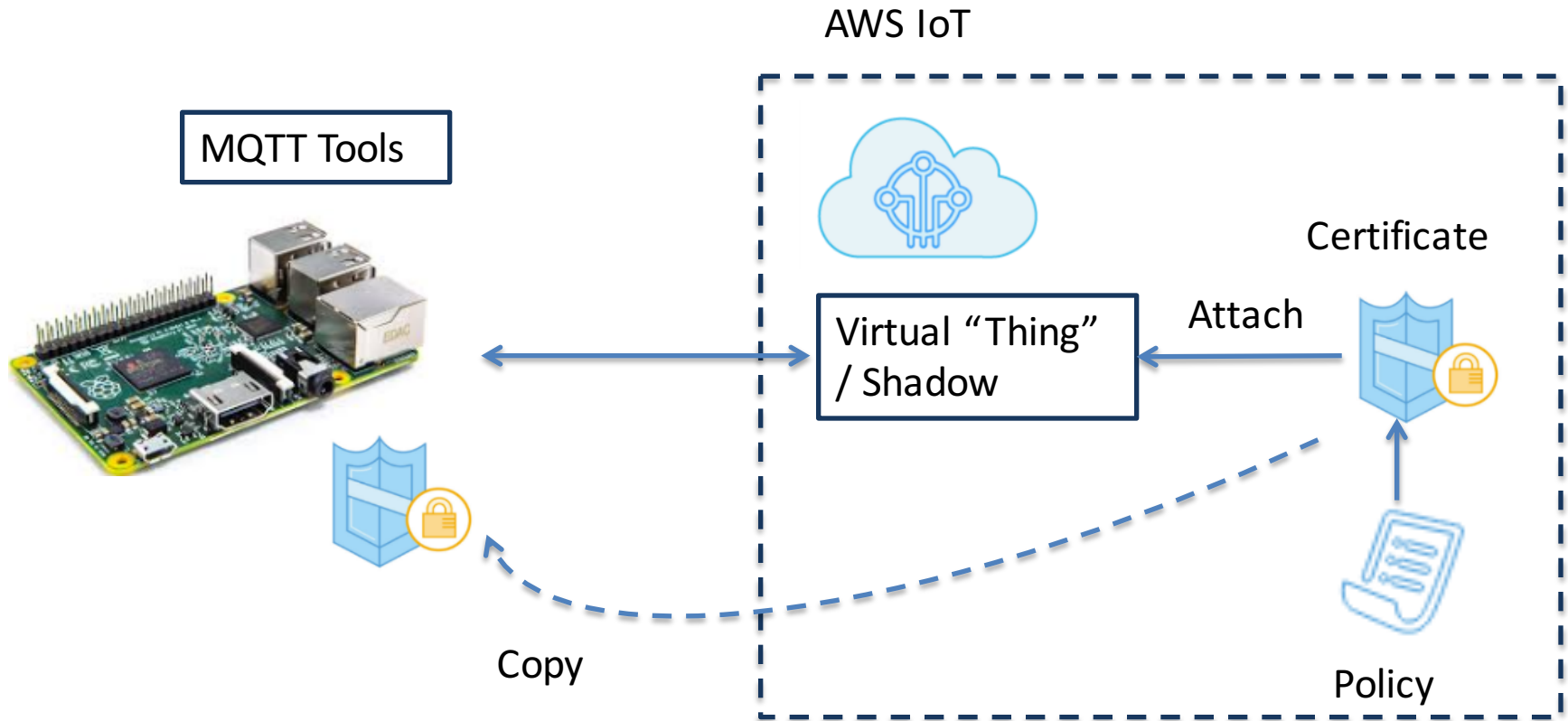
Subscriptions

Subscribe to a topic

- \$aws/things/Test/shadow/update/accepted ×

Subscriptions	\$aws/things/Test/shadow/update/accepted	Clear	Pause
	\$aws/things/Test/shadow/update/accepted		Jan 17, 2017 10:27:34 PM Hide
	<pre> { "state": { "reported": { "Info": "Hello AWS IoT!" } } } </pre>		

Step 2: Connect a Physical Device



Create and get Certificates

➤ Create Certificates

☐ Security → Certificates → Create

➤ Download Cert Files

- 1. public & private key
- 2. thing cert
- 3. Root CA for AWS



Dashboard

Connect

Registry

Security
Certificates

Policies

CAs

In order to connect a device, you need to download the following:

A certificate for this thing	f32c514adc.cert.pem	Download
A public key	f32c514adc.public.key	Download
A private key	f32c514adc.private.key	Download

You also need to download a root CA for AWS IoT from Symantec:

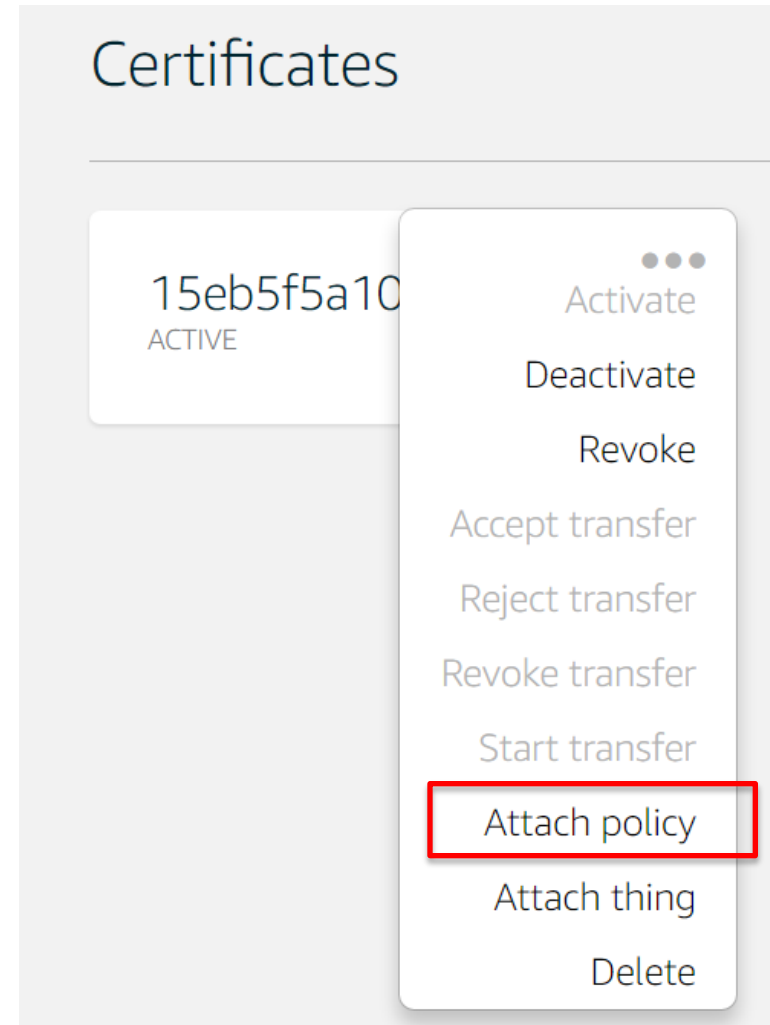
A root CA for AWS IoT [Download](#)

Create Policy and attach it to cert

➤ Create Policy



➤ Attach Policy to Certificates



Connect your Device

- Copy certificates to RP2

- Choose your AWS SDK (support MQTT)
 - Node JS
 - Python
 - Java
 - Embedded C

- You can also use third party MQTT tools
 - Python (paho mqtt library)



Some Notes

- 1. You will need these certification when setting up the TLS1.2 verification

```

pi@NaroRP2: ~/Course/CSE521S_2017/1_Connection...
pi@NaroRP2 ~/Course/CSE521S_2017/1_Connection_Test $ ls
15eb5f5a10-certificate.pem.crt  Connect.py
15eb5f5a10-private.pem.key     root-CA.pem
15eb5f5a10-public.pem.key
pi@NaroRP2 ~/Course/CSE521S_2017/1_Connection_Test $
  
```

- 2. You will need the endpoint and port (8883) when connect to AWS IoT Gateway

Security

HTTPS

Shadow

Interact

Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

Activity

a351pfzlksv6kq.iot.us-west-2.amazonaws.com

Publish / Subscribe

➤ Publish

- payload = "{\"state\":{\"reported\":{\"rndnum\":50}}}"

```
pi@NaroRP2: ~/Course/CSE521S_2017/1_Connect
pi@NaroRP2 ~/Course/CSE521S_2017/1_Connect $ ./2_Publish.py
Subscriber Connection status code: 0 | Connection status: successful
```

Shadow Document

Last update: Jan 17, 2017 11:18:50 PM -0600

Shadow state:

```
1 {
2   "reported": {
3     "rndnum": 50
4   }
5 }
```

➤ Subscribe

Publish to topic

```
1 {
2   "state":
3   {
4     "reported":
5     {
6       "rndnum": 60
7     }
8   }
9 }
```

```
pi@NaroRP2: ~/Course/CSE521S_2017/1_Connect
pi@NaroRP2 ~/Course/CSE521S_2017/1_Connect $ ./3_Subscri
Subscriber Connection status code: 0 | Connection status
Subscribed: 1 (0,)dataNone
[Topic] : $aws/things/RaspberryPi/shadow/update/accepted
[Data]  : b'{"state":{"reported":{"rndnum":60}},"metadat
:{"rndnum":{"timestamp":1484716970}},"version":350,"tim
6970}'
[rndnum]: 60
```

Step 3: Push Button and Publish

Random
Integer
[1, 100]

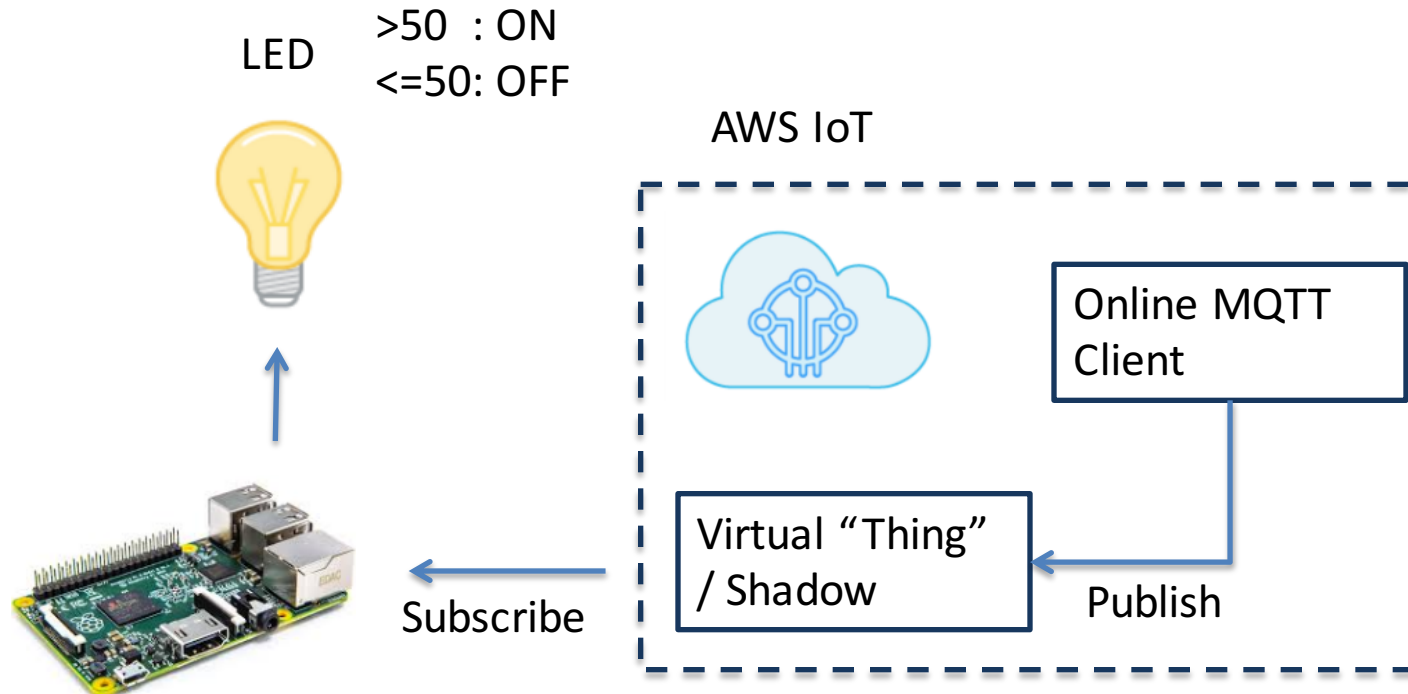


Publish

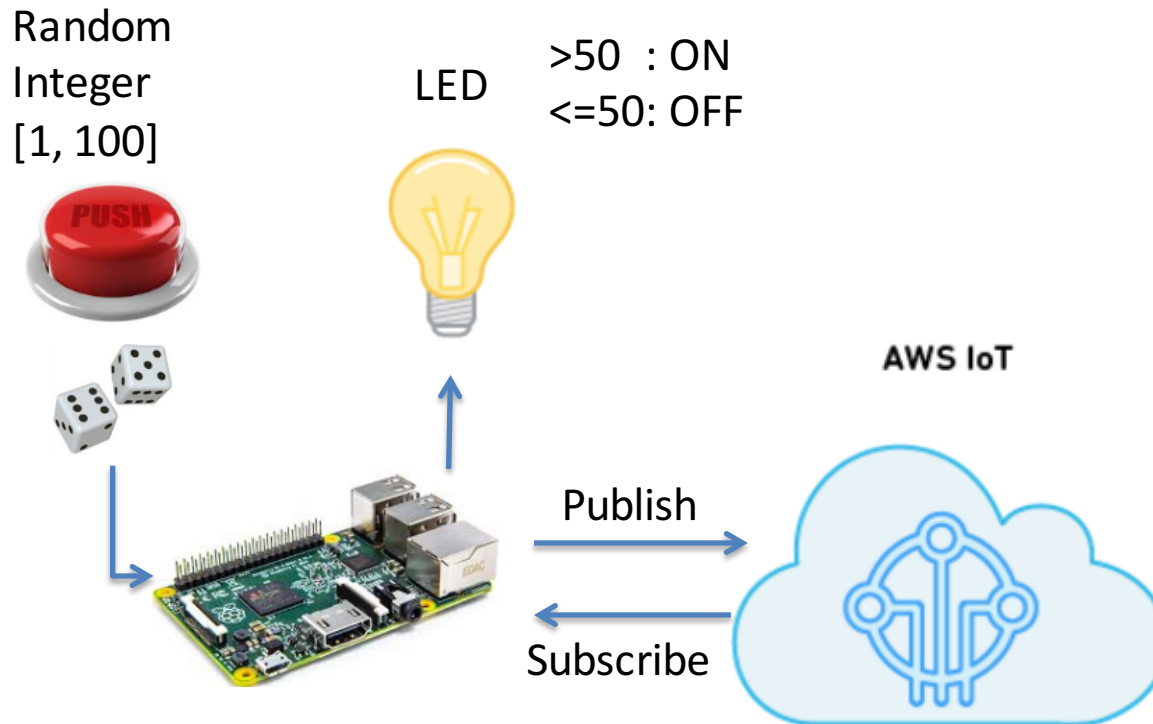
AWS IoT



Step 4: Subscribe and Lit up LED

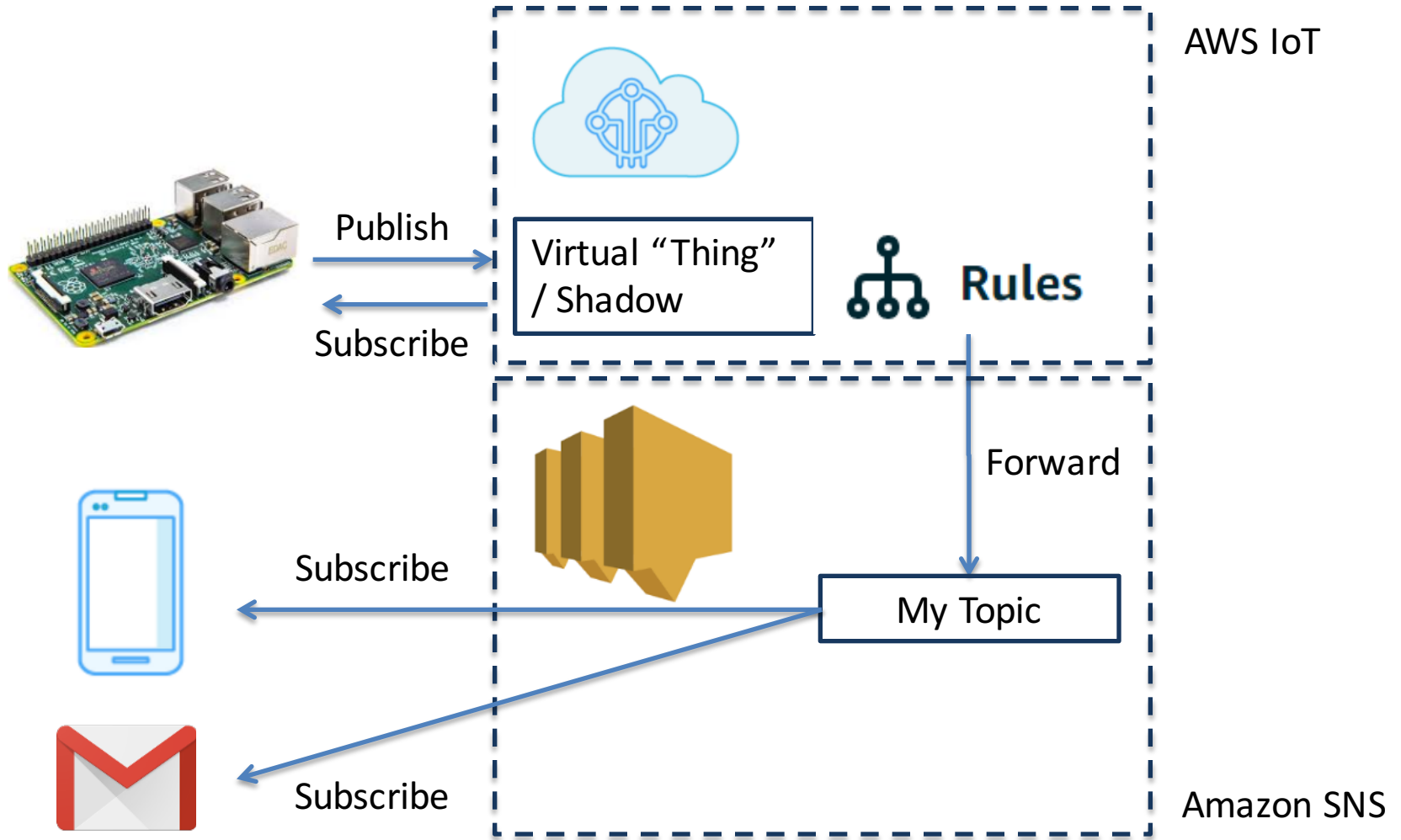


Step 5: Combine Them Together



More Fancy: SNS services

➤ Simple Notification Service



Amazon SNS

- Create a Topic
 - ARN will be used later

Topic details: LED_Litup

Publish to topic

Other topic actions ▾

Topic ARN	arn:aws:sns:us-west-2:401317363811:LED_Litup
Topic owner	401317363811
Region	us-west-2
Display name	LED_Litup

Subscriptions

Create subscription

Request confirmations

Confirm subscription

Other subscription actions ▾

Filter

<input type="checkbox"/>	Subscription ID	Protocol	Endpoint
<input type="checkbox"/>	arn:aws:sns:us-west-2:401317363811:LED_Litup:9d1e4c16-4316-47c3-a8f1-763c72152...	sms	+1929 [REDACTED]
<input type="checkbox"/>	arn:aws:sns:us-west-2:401317363811:LED_Litup:975dbe42-cde3-4b3a-80fc-a404e6930...	email	[REDACTED]@gmail.com

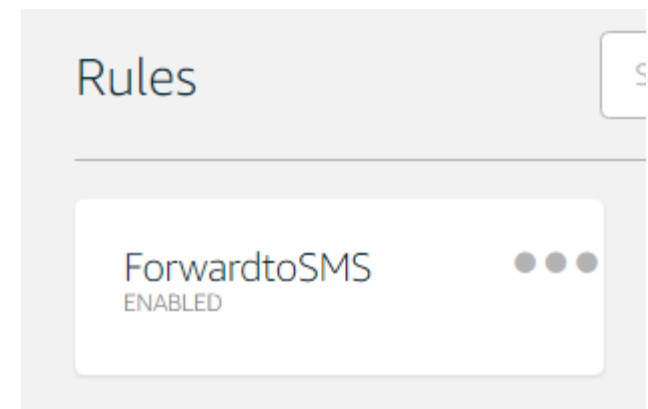
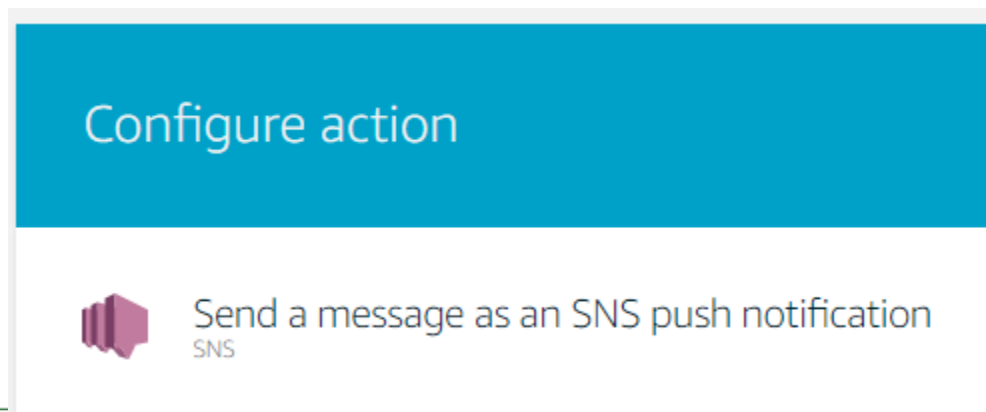
Create a Rule in Amazon IoT

- Add a query to filter your interesting topic (event)

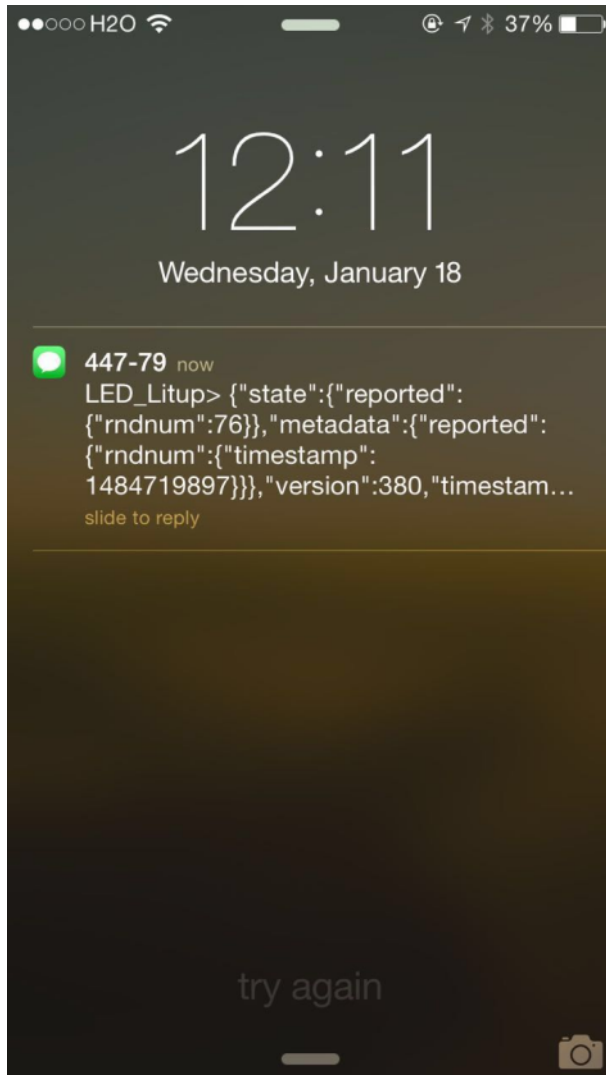
Rule query statement

```
SELECT * FROM '$aws/things/RaspberryPi/shadow/update/accepted'
```

- Add an Action:
 - Forward this message to SNS
 - Specify Dest ARN
 - Enable Rule



Notification on SMS & Email



AWS Notification Message

Inbox x

LED_Litup no-reply@sns.amazona 12:11 AM (1 minute ago) ☆

to me ▾

```
{\"state\":{\"reported\":{\"rndnum\":76}},\"metadata\":{\"reported\":{\"rndnum\":
{\"timestamp\":1484719897}}},\"version\":380,\"timestamp\":1484719897}
```

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:

https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:401317363811:LED_Litup:975dbe42-cde3-4b3a-80fc-a404e6930687&Endpoint=narohlee@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

Recap: Hello AWS IoT!

Random
Integer
[1, 100]

LED

>50 : ON
<=50: OFF



Publish

AWS IoT

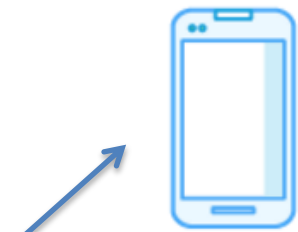
Subscribe



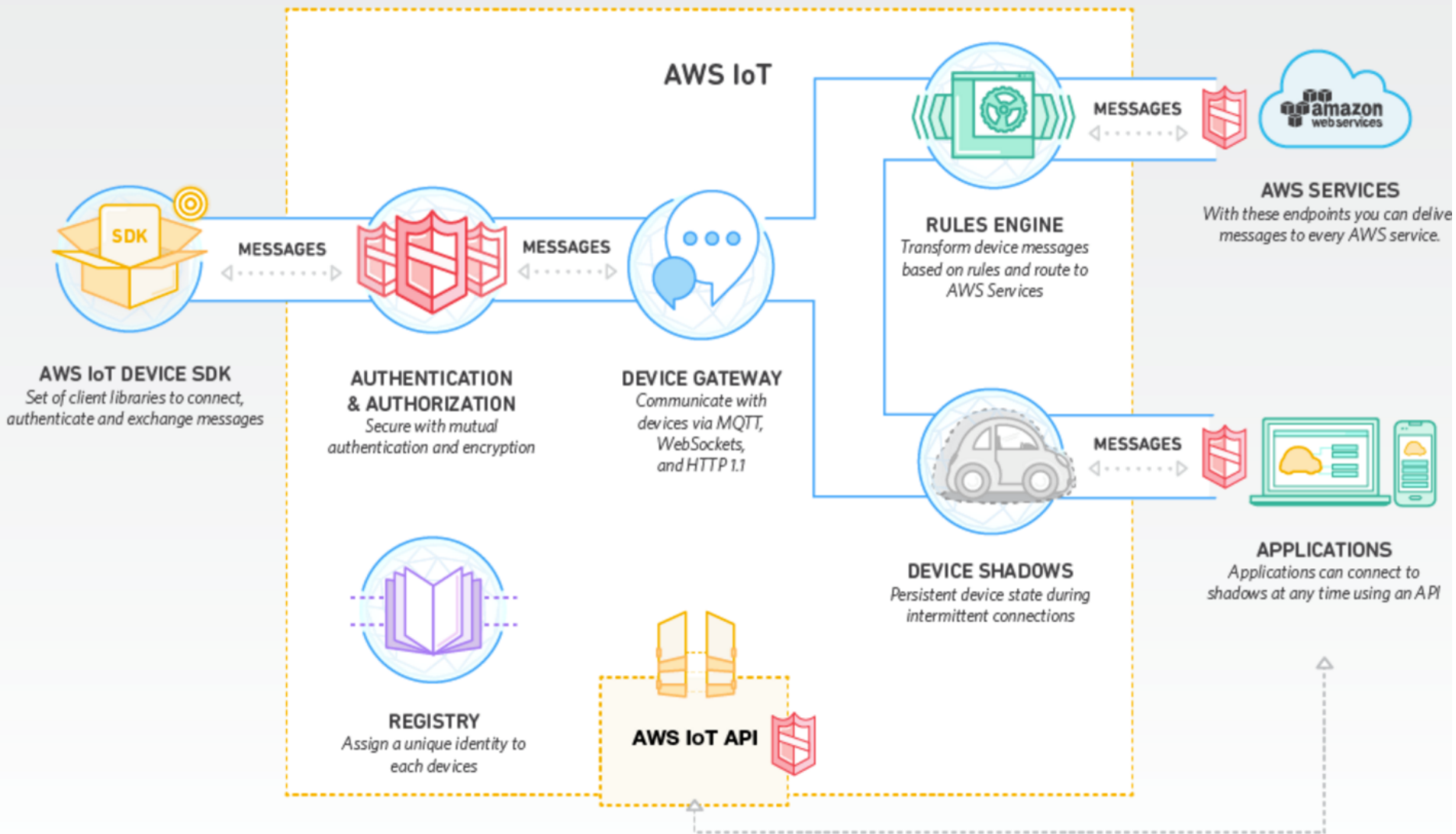
Forward



Amazon SNS



Recap: Amazon IoT Architecture



Be Creative!

➤ Bunch of Services

➤ **Embedded systems + Cloud Services...**


➤ IoT!


AWS services


Find a service by name (for example, EC2, S3, Elastic Beanstalk)


▼ All services

 **Compute**
EC2
EC2 Container Service
Lightsail
Elastic Beanstalk
Lambda
Batch


 **Storage**
S3
EFS
Glacier
Storage Gateway

 **Database**
RDS
DynamoDB
ElastiCache
Redshift


 **Networking & Content Delivery**
VPC
CloudFront
Direct Connect
Route 53

 **Migration**
DMS
Server Migration
Snowball

 **Developer Tools**
CodeCommit
CodeBuild
CodeDeploy
CodePipeline


 **Management Tools**
CloudWatch
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Trusted Advisor
Managed Services
Application Discovery Service


 **Security, Identity & Compliance**
IAM
Inspector
Certificate Manager
Directory Service
WAF & Shield
Compliance Reports


 **Analytics**
Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
Data Pipeline
QuickSight


 **Artificial Intelligence**
Lex
Polly
Rekognition
Machine Learning


 **Internet of Things**
AWS IoT


 **Game Development**
GameLift

 **Mobile Services**
Mobile Hub
Cognito
Device Farm
Mobile Analytics
Pinpoint

 **Application Services**
Step Functions
SWF
API Gateway
Elastic Transcoder

 **Messaging**
SQS
SNS
SES

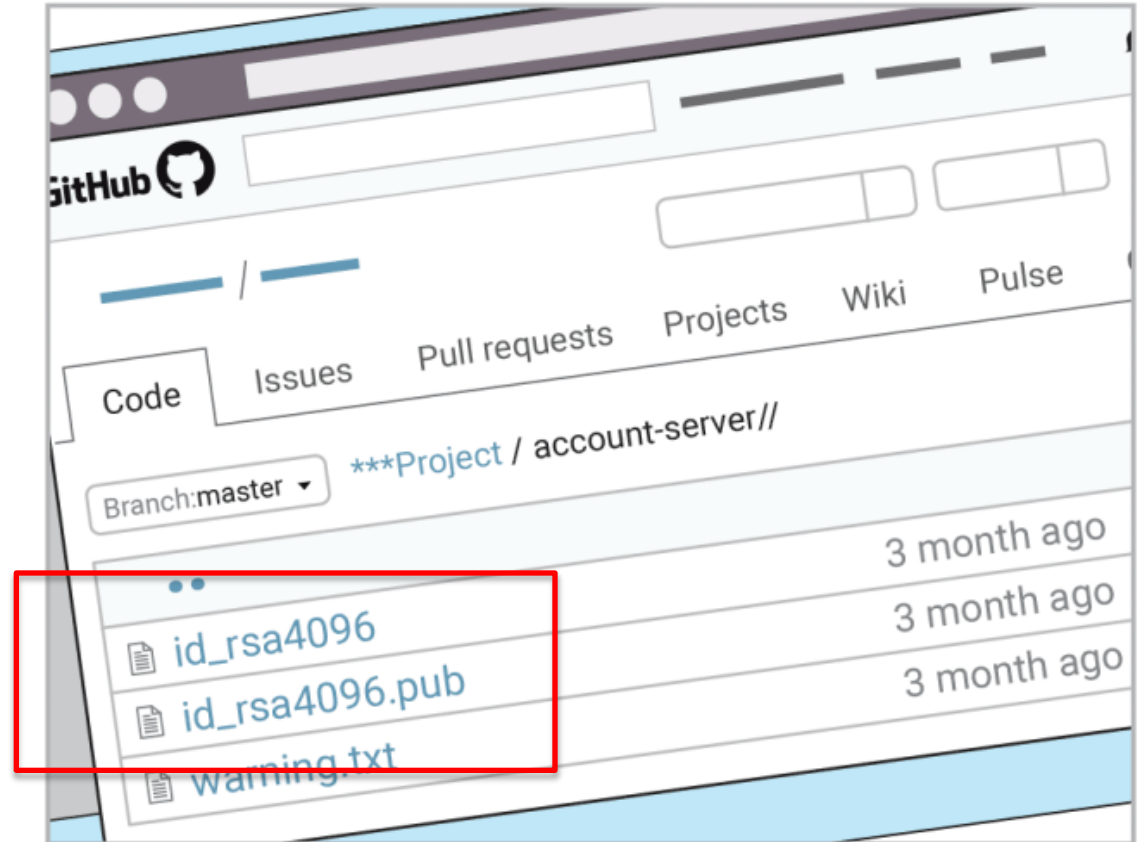
 **Business Productivity**
WorkDocs
WorkMail

 **Desktop & App Streaming**
WorkSpaces
AppStream 2.0

One More Thing: Security

➤ DON'T **UPLOAD** YOUR PUBLIC KEY!!!

Time to Open Source!



What if... 50,000 AWS Bill!

Quora

Ask or Search Quora

Ask Question

Fraud

Amazon Web Services

Amazon.com (product)

Hackers

+3



My AWS account was hacked and I have a \$50,000 bill, how can I reduce the amount I need to pay?

For years, my bill was never above \$350/month on my single AWS instance. Then over the weekend someone got hold of my private key and launched hundreds of instances and racked up a \$50,000 bill before I found out about it on Tuesday. Amazon had sent a warning by email at \$15,000 saying they had found **our key posted publicly**, but I didn't see it. Naturally, this is a devastating amount of money to pay. I'm not saying I shouldn't pay anything, but this just a crazy amount in context. Amazon knew the account was compromised, that is why they sent an email, they knew the account history and I had only spent \$213 the previous month. I almost feel they deliberately let it ride to try to earn more money. Does anyone have any experience with this sort of problem?

Pointers

- Amazon IoT
 - ❑ <http://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>
- Amazon SNS
 - ❑ <http://docs.aws.amazon.com/sns/latest/dg/welcome.html>
- **AWS Resource list** for course projects
 - ❑ http://cps.cse.wustl.edu/index.php/AWS_Resources
- Apply for \$40 credits for Amazon AWS
 - ❑ <https://aws.amazon.com/education/awseducate/apply/>

Project Requirements

- Run in public cloud

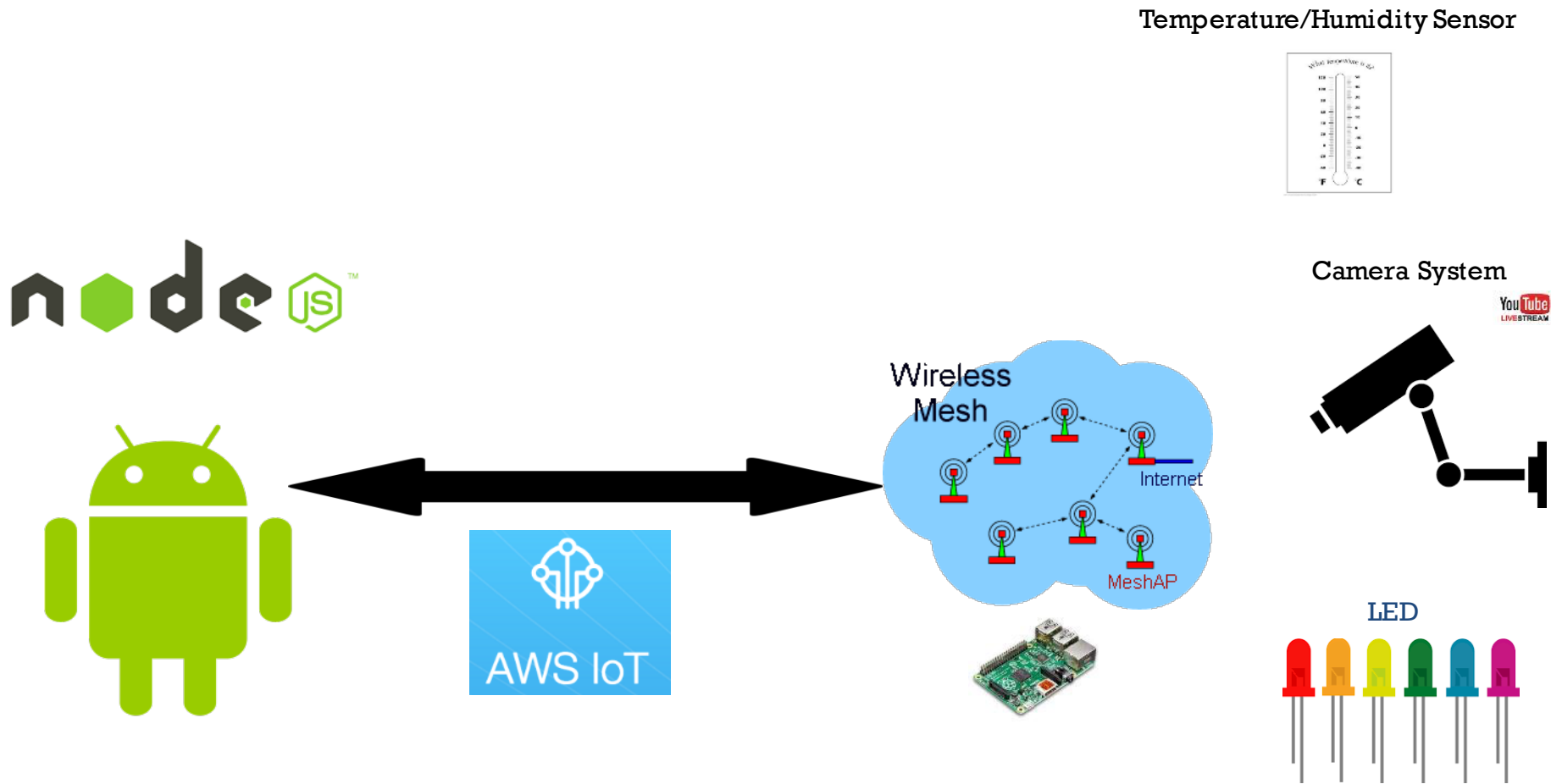
- Difficulty varies for listed candidates - will take difficulty into consideration when grading.

- Will grade based on
 - ❑ project difficulty
 - ❑ quality and depth of work
 - ❑ workload distribution among team members

- Milestones: proposal, demo 1, demo 2, final demo, report.

- **Start early! Discuss with us and Dr. Lu**

Example: Smart House Keeper



Final Project

Integrated everything with webpage dashboard

- RealTime Live Stream
- RealTime Temperature and Humidity Monitor
- Remotely Light Control
- Virtual Control Over Door
- Android App: Live Streaming and Light Control

