AWS IoT QuickStart

Kirk S. Kalvar

5/12/2017

# Create AWS IoT Device

1. AWS IoT/Get Started/Registry/Things/Register a thing
2. Name <myiotdevice>
3. Click on “Create thing”
4. Update Android application MY\_IOT\_DEVICE with <myiotdevice>
5. AWS IoT/Registry/<myiotdevice>/Interact
6. Update Android application CUSTOMER\_SPECIFIC\_ENDPOINT with HTTPS Rest API Endpoint

# Create AWS Cognito Federated Identity

1. Cognito/ Manage Federated Identities
2. Identity pool name: <myiotdevice\_identity>
3. Unauthenticated identities: Check “Enable access to unauthenticated identities”
4. Click on “Create Pool”
5. Click on “Allow”
6. Update Android application COGNITO\_POOL\_ID with highlighted Identity from Pool ID in “Get AWS Credentials” section

# Update AWS IAM Cognito Roles

1. Update IAM/Roles/Cognito\_<myiotdevice\_identity>Unauth\_Role
2. Permissions/Attach Policy
3. Add “AWSIoTFullAccess”
4. Click on “Attach Policy”

# Test <myiotdevice> Connectivity

1. AWS IoT/Registry/Things/<myiotdevice>/Activity
2. Send a few packets through to AWS IoT via Mobile Application

# Create AWS IoT Device Rule

1. AWS IoT/Rules/Create a rule
2. Name: <myiotdevice\_rule>
3. Attribute: \*
4. Topic filter: $aws/things/<myiotdevice>/shadow/update/accepted

# Add AWS IoT Device Rule S3 Bucket

1. Click on “Add Action”
2. Select “Store messages in an Amazon S3 bucket”
3. Click on “Configure action”
4. Click on “Create a new resource” # this will open a new browser tab to Amazon S3
5. Clock on “Create bucket”
6. Bucket Name: <myiotdevice-s3-bucket> # must be lowercase, only allows underscores
7. Next
8. Next
9. Next
10. Click on “Create bucket”
11. Go back to AWS IoT “Configure action” browser tab
12. Click “refresh” icon between S3 bucket and “Create a new resource” button
13. Using S3 bucket pull down, Select < myiotdevice-s3-bucket >
14. Key: ${topic()}/${timestamp()}.txt
15. Click on “Create a new role”
16. IAM Role name: <myiotdevice\_role\_s3>
17. Click on “Create a new role”
18. Using IAM role name pull down, Select <myiotdevice\_role\_s3>
19. Click on “Update role”
20. Click on “Add action”
21. Click on “Create rule”

# Test <myiotdevice\_rule> S3 bucket

# Add AWS IoT Device Rule Lambda

1. AWS IoT/Rules/<myiotdevice\_rule>
2. Click on “Add action”
3. Select “Invoke a Lambda function passing the message data”
4. Click on “Configure action”
5. Click on “Create a new resource” # will take you to new browser tab
6. In Blueprints filter: hello-world <return>
7. Click on “hello-world” nodejs6.10
8. Configure triggers: Click on dotted box and Select “AWS IoT”
9. IoT Type Pull down: Custom IoT Rule
10. Rule name: <myiotdevice\_rule> # enter complete rule name before fields will auto populate
11. Enable trigger: click on checkbox to enable
12. Next
13. Name: <myiotdevice\_lambda>
14. Runtime: Node.js 6.10
15. Code entry type: Edit code inline
16. **Cut and Paste lambda.js from Android Project into code pane**
17. Update lambda.js variable endpointAddress: CUSTOMER\_SPECIFIC\_ENDPOINT
18. Update lambda.js variable thingName: <myiotdevice>
19. In “Lambda function handler and role” section
20. Role: Create a custom role # will take you to a new browser tab
21. Role name: <myiotdevice\_role\_lambda>
22. Allow
23. Back on AWS IoT “Configure function” browser tab
24. In “Advanced Settings” section
25. Timeout: 10 secs
26. Next
27. Click on “Create function”
28. Back on AWS IoT “Configure action” browser tab click “refresh” icon between “Function name” and “Create a new resource” button
29. Using “Function name” pull down, Select < myiotdevice\_role\_lambda >
30. Click on “Add action”

# Update AWS IAM Lambda Roles

1. IAM/Roles/< myiotdevice\_role\_lambda >
2. Permissions/Attach Policy
3. Add “AWSIoTFullAccess”
4. Click on “Attach Policy”

# Test <myiotdevice\_rule> Lambda function

1. Using Android Application Open Window set Temperature > 72 Degrees and set window to “OPEN” # NOTE: the rule deployment is not immediate, it may take a minute or two

# Add AWS IoT Device Rule Elasticsearch

1. AWS IoT/Rules/<myiotdevice\_rule>
2. Click on “Add action”
3. Select “Send messages to the Amazon Elasticsearch Service”
4. Click on “Configure action”
5. Click on “Create new resource” # will open browser to new tab
6. Click on “Create a new domain”
7. Elasticsearch domain name: <myiotdevice-es> # must be lowercase, no underscores allowed
8. Next
9. Instance count: 1
10. Instance type: t2.small.elasticsearch
11. Next
12. In “Set up access policy: section
13. Select a template: allow open access to the domain
14. Confirm “Policy Risk Statement”
15. Next
16. Click on “Confirm”
17. Domain status: loading # NOTE: ***allow completion before proceeding***, it takes a few minutes!
18. Back on AWS IoT “Configure action” browser tab
19. Click “refresh” between Domain name and “Create a new resource” button
20. Using Domain name pull down, Select <myiotdevice-es>
21. Be sure Endpoint is pre-populated before continuing
22. Id: ${newuuid()}
23. Index: <myiotdevices>
24. Type: < myiotdevice >
25. Click on “Create a new role”
26. Enter <myiotdevice\_role\_es>
27. Click on “Create a new role”
28. Click “refresh” icon between “IAM role name” and “Create new role” button
29. Using IAM role name pull down, Select <myiotdevice\_role\_es>
30. Click on “Update role”
31. Click on “Add action”

# Access Elasticsearch dashboard

1. AWS Elasticsearch/Dashboards
2. Click on <myiotdevice-es>
3. Click on Kibana endpoint

# Prep <myiotdevice-es> Elasticsearch

1. Send a few packets through to AWS IoT via the Mobile Application so Elasticsearch has some records to work with.

# Configure Kabana Index Pattern

1. Index name or pattern: \*
2. Uncheck “Index contains time-based events “
3. Click on “Create”
4. Set filter: state
5. Click on Star graphic to set as default index

# Configure Kabana Vertical Bar Chart

Kabana is a little more involved to configure so I’ll let you figure out the basics. AWS has online documentation as well for Kabana. I’ve included snapshots of the configuration values to “Visualize” an “Vertical Bar Chart” for the demo. See below. NOTE: Full size graphic is included in the .assets directory of project.



