# LAB 5 – Application Layer

Name – Sati, Ankit Date – 10/13/2021

Section - 001

Total in points (Maximum 100 points)–

Professors Comments –

Affirmation of Independent Effort – Ankit Sati

Question 1-

1. This questions wants us to walk through the edge computing that is being developed and take a look at it from the different perspectives listed below.
2. NEC - Network edge compute
3. MEC - Multi access edge compute
4. Help of Robotics
5. Impact on mission critical systems.
6. Implementation and use of private networks.

**Main Goal** – The primary goal is to deploy everything at the end for the users so that they can reduce the time of data on the networks. We try to provide the below services at the end of the network, which makes it faster/accurate with less chances of frame drops.

* Compute Power
* Cloud space
* High Availability
* Service Space

**Main Objective** - Everything for the end user need to remain the same from code creation to deployment/maintenance but the end user will not know where it is being used from. It can either be on the edge of the network or the azure platforms as it has been in the past.

**NEC - Network edge compute**

Network Edge Compute (NEC) is the network carrier equivalent, placing the edge computing platform within their network. Instead of needing to access applications and games running in the public cloud, software providers can bring their solutions physically closer to their end-users. At AT&T’s Business Summit we gave an augmented reality demonstration, working with Taqtile, and showed how to perform maintenance on an aircraft landing gear.

Prime features

* Closer to end user
* One hop communications
* Compute power at the edge.
* Deployment of services at the end.

**MEC - Multi access edge compute**

Through the combination of local compute resources and private mobile connectivity (private LTE), we can enable many new scenarios. For instance, in the smart factory example used earlier customers are now able to run their robotic control logic, highly available and independent of connectivity to the public cloud. MEC helps ensure that operations and any associated critical first-stage data processing remain up and production can continue uninterrupted. Advantage of near-infinite compute and storage, the cloud is ideal for large data-intensive and computational tasks, such as machine learning jobs for predictive maintenance analytics.

Prime Features

* Combination of local compute resources and private mobile connectivity
* Communication over private network.
* Complex logic like robotics can be accessed.
* **Near infinite compute storage.**

**Prime advantages of EDGE COMPUTE**

* Single hop computations
* Frames spend less time over the network.
* Reduce the chance of frame drops.
* Infinite compute storage at the end of network
* Complex logic can be deployed at the end of the network.
* Privatized network for mission critical projects.
* Very high speeds.
* High Availability of resources at the end.
* Cloud space with specific features as per demand of end user.

**Examples of technologies used in field.**

* Enterprise level – mission critical projects.
* Private LTE networks.
* Multiple cloud space
* Smart agriculture and services.
* Robotics in enterprise products.
* Product development.
* Resource deployment for projects.

# Deploy your first IoT Edge module to a Windows device

**Part 1 – Create your IOT HUB.**

Graphical user interface, text, application

Description automatically generated

**Part 2 – Register an IoT Edge device (Screenshot attached)**

**hub name - as141286**

**resource group as14128**

**Connection Key - HostName=as141286.azure-devices.net;DeviceId=myEdgeDevice;SharedAccessKey=DUQ43decm2Rqt8D5u01ZEVDDZ52gq7elK8qdiUKkwBs=**

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Part 3- Install and start the IoT Edge runtime**

Text

Description automatically generated

**Part 4- Deploy a module**

Graphical user interface, text, application, email

Description automatically generated

**Select routes**

{

"modulesContent": {

"$edgeAgent": {

"properties.desired": {

"modules": {

"SimulatedTemperatureSensor": {

"settings": {

"image": "mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0",

"createOptions": ""

},

"type": "docker",

"status": "running",

"restartPolicy": "always",

"version": "1.0"

}

},

"runtime": {

"settings": {

"minDockerVersion": "v1.25"

},

"type": "docker"

},

"schemaVersion": "1.1",

"systemModules": {

"edgeAgent": {

"settings": {

"image": "mcr.microsoft.com/azureiotedge-agent:1.1",

"createOptions": ""

},

"type": "docker"

},

"edgeHub": {

"settings": {

"image": "mcr.microsoft.com/azureiotedge-hub:1.1",

"createOptions": "{\"HostConfig\":{\"PortBindings\":{\"443/tcp\":[{\"HostPort\":\"443\"}],\"5671/tcp\":[{\"HostPort\":\"5671\"}],\"8883/tcp\":[{\"HostPort\":\"8883\"}]}}}"

},

"type": "docker",

"status": "running",

"restartPolicy": "always"

}

}

}

},

"$edgeHub": {

"properties.desired": {

"routes": {

"route": "FROM /messages/\* INTO $upstream",

"SimulatedTemperatureSensorToIoTHub": "FROM /messages/modules/SimulatedTemperatureSensor/\* INTO $upstream"

},

"schemaVersion": "1.1",

"storeAndForwardConfiguration": {

"timeToLiveSecs": 7200

}

}

},

"SimulatedTemperatureSensor": {

"properties.desired": {

"SendData": true,

"SendInterval": 5

}

}

}

}

**REVIEW AND CREATE**

A picture containing background pattern

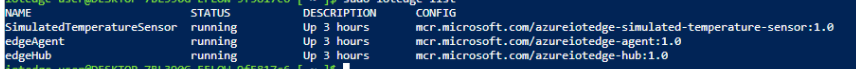
Description automatically generated

**PART 5 – Viewing the device and the data**

Graphical user interface, application

Description automatically generated

**Monitor the data**



**FINAL PART \_ CLEANING UP THE RESOURCES**

**Final screenshot before deletion**

Text

Description automatically generated

**Post Deletion**

Graphical user interface, text, application

Description automatically generated

**Activity log snippet**

Graphical user interface, text, application, email

Description automatically generated

**EXTRA CREDIT**

**Tutorial 1 - Monitor IoT Edge devices**

**Step 1 - Create a Log Analytics workspace**

**Workspace ID : b2050b98-4b47-4ee2-8a19-7c6d7b6f65b1**

Graphical user interface, application

Description automatically generated

**Step 2 - Create a Log Analytics workspace**

**ID -** b7f18d8e-271b-442d-8714-ee5d59e0159e

Graphical user interface, text, application, email

Description automatically generated

**Step 3 - Deploy the metrics collector module**

Graphical user interface, text, application, email

Description automatically generated

**Step 4 - Explore the fleet view and health snapshot workbooks**

Graphical user interface, chart

Description automatically generated

**Tutorial 2 - Develop IoT Edge modules using Windows containers**

**Step 1 - Set up VS**

**Installing environment and reps.**

**Step -2 – Create a container registry**

Graphical user interface, text, application

Description automatically generated

**Step 3 – Edge Runtime**

Graphical user interface, text

Description automatically generated

**Step 4 - Provide your registry credentials to the IoT Edge agent**

"registryCredentials": {

"<registry name>": {

"username": "$CONTAINER\_REGISTRY\_USERNAME\_<registry name>",

"password": "$CONTAINER\_REGISTRY\_PASSWORD\_<registry name>",

"address": "<registry name>.azurecr.io"

}

}

**Step 5 - Changes on Device**

A screenshot of a computer screen

Description automatically generated with medium confidence

**STEP 6 –**

**Clean up resource**

* **Resources – Done**
* **Modules – Done**
* **Devices done**

**Tutorial 3 – Custom code**

**Step 1 - Set up Java**

**Installing environment and reps.**

**Step 2 – Code applet**

Text

Description automatically generated

**Step 3 – Push the custom modules**

**Step 4 - Deploy modules to device**

Graphical user interface, application

Description automatically generated

**Step 5 - Edit the module twin**

Text

Description automatically generated

**Step 6 – Delete all IOT edge modules.**

* **Resources – Done**
* **Modules – Done**
* **Devices done**