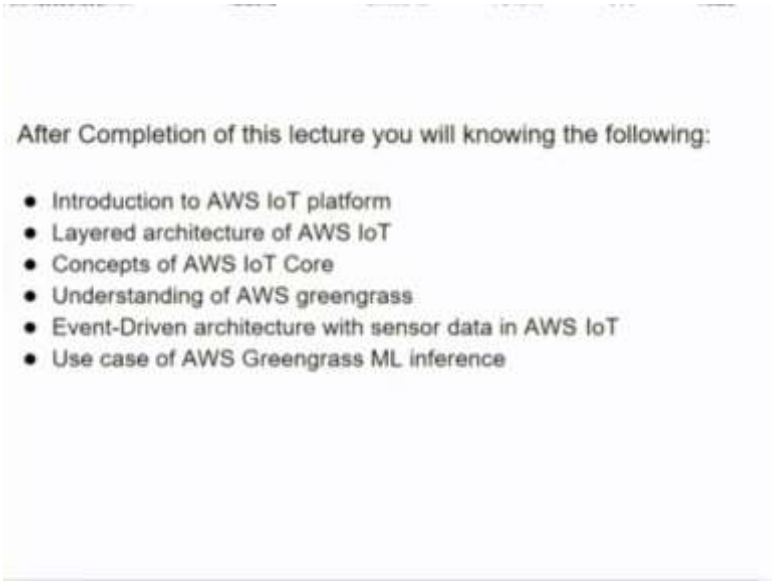
Foundation of Cloud IoT Edge ML
Professor Rajiv Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna
Lecture No. 20
Introduction to Edge ML with AWS IoT platform

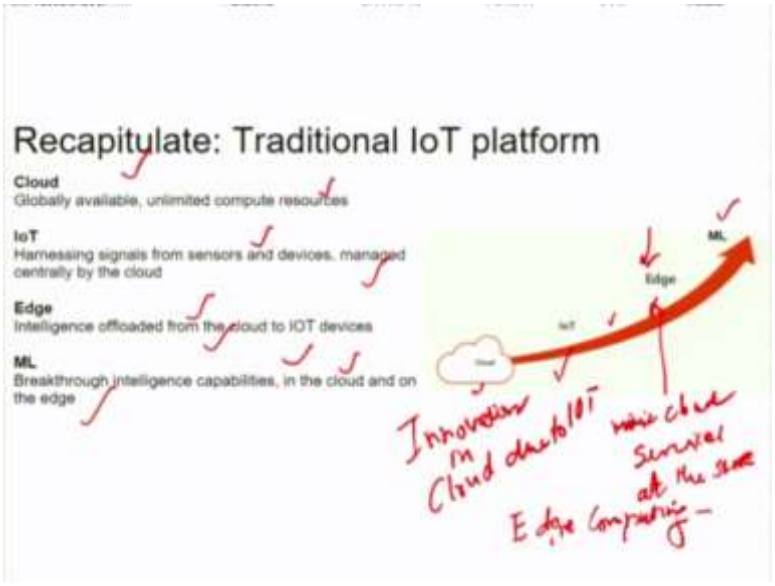
I am Doctor Rajiv Misra from IIT, Patna. The topic of this lecture is Introduction to Edge ML with AWS IoT platform.

(Refer Slide Time: 0:25)



So, in this lecture, we will introduce you to the following concepts. First is that we will introduce you to AWS, that is service for IoT platform. Then we will discuss the layered architecture of AWS IoT and then concepts of AWS IoT core. And then we will do the introduction of AWS Greengrass, which is an edge offering for IoT platforms. Then we will discuss about and use case, which is an event driven architecture with sensor data using AWS IoT. We will also see some of the use case of AWS Greengrass that is doing machine learning at the edge on AWS Greengrass.

(Refer Slide Time: 1:23)



So, let us recapitulate what we have so far discussed is that we have given you the concept of traditional IoT platform, which is being offered by different vendors such as Amazon, which is popularly known as Azure IoT Hub, or Microsoft Azure IoT Hub. And Amazon AWS IoT Greengrass. These are all the offering to do the IoT at the edge. So, looking at the trend, which is shown here in the figure that lot of innovations in the cloud is happening due to the advent of the internet of things.

So, lot of internet of things devices, when they start connecting at the cloud, then there is a demand for the cloud to offer the services to the IoT and that led to the growth of this cloud. Now, there is an issue that IoT on a cloud will require lot of issues like latency. That is that IoT data has to travel up to the cloud for doing the influencing and analytics and then come back to the cloud and come back to the IoT for inferencing.

Therefore, due to the latency, lot of new use cases were not supporting and also since lot of devices start connecting to the cloud. So, that also scalability problem at the level of cloud also is becoming significant. So, due to these requirements, the, so the cloud providers, they have started considering to provide the cloud-like services very close to the IoT devices and that is called the Edge. So, this will mimic the cloud services at the source. So, this becomes an Edge Computing.

Edge Computing makes the cloud fully distributed. And then, the most important use case for doing this IoT and depending upon the cloud is machine learning on this IoT data. So, performing machine learning earlier used to happen at the cloud, but now with the

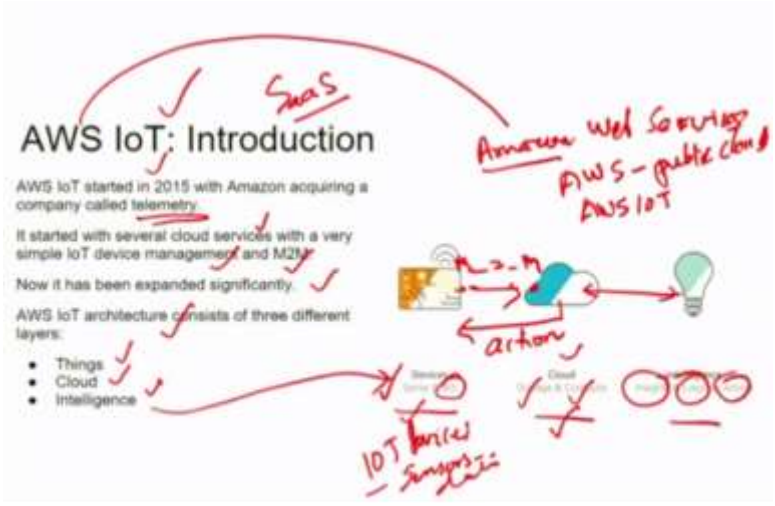
provisioning of edge computing that is at the edge layer of an IoT, this is possible that machine learning inference can be performed at this edge.

So, that is the devices can now do this business intelligence of the logic plate or action taken at the edge that is at the source itself and therefore, paving away not only to the new kind of application such as self-driving car, but also allowing this scalability of these IoT devices to be serviced by the cloud in conjunction with the edge. So, that is what is explained. This is the innovation.

So, we have in this slide have covered that the cloud, although it is globally available and also providing unlimited compute resource, but when comes to the IoT, then these signals which is generated out of the sensors and the devices and earlier, which used to be managed centrally by the cloud, requires more attention, more services. Therefore, edge computing is being innovated out of the cloud due to these requirements to make this intelligence offloaded from cloud to these IoT devices. So, that is what we have explained.

Now coming to the machine learning. So, the breakthrough in the intelligence capabilities, which was earlier being done at the cloud, is now shifted to the edge and thereby this particular notion of doing machine learning at the edge that is very close to the source becomes and requirement and, supporting lot of new applications and also the scalability. Let us go ahead and see how AWS, IoT, Edge can provide this particular support into their platform.

(Refer Slide Time: 6:32)



So, let us start with the introducing the AWS IoT. The full form of AWS is Amazon Web Services. So, this AWS is popularly known as the public cloud. And this, when you say AWS IoT, that means, so Amazon Web Services that is AWS for IoT is a service as software as a service platform, which is provided by the public cloud that is Amazon. So, AWS IoT started in 2015 with Amazon acquiring a company called Telemetry.

So, at that point, telemetry and that it is started with several cloud services with a very simple device management and machine to machine. So, meaning to say that IoT devices are allowed to connect to the cloud that is AWS cloud initial days that is, when they have acquired the complete telemetry and thereby offering this cloud services to these IoT devices.

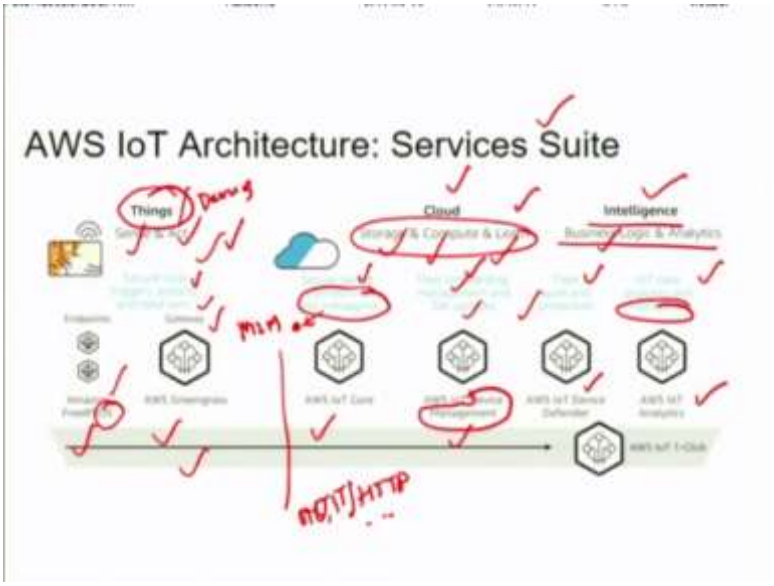
This simple notion has now expanded significantly now, and AWS architecture thereby, consist of three different offering into the layers called as a thing, cloud and intelligence, which is explained here in this particular diagram. So, when we say things that means, the sensors and actuators, we are the IoT devices. So, these IoT devices has the sensors and these sensors are generating the data called sensor data. So, they will sense the environment and generate the data.

Now the other part that this particular data needs to be actioned. So, nowadays with Amazon IoT that is AWS IoT where it grow to that particular level where these devices have become intelligent and therefore, the actions can also be performed there in the device itself with the help of AWS IoT that we are going to see.

The other component is called the cloud. So, these devices, when they are connected to the cloud with the help of telemetry and other protocols like MQTT, HTTP and all and others. So, this particular connectivity provides machine to machine type of offering. So, machine to machine communications, that is one end of that M to M is at the cloud. So, cloud provides this kind of connectivity and thereby offerings to the IoT services such as the storage in the cloud and doing the computation with the help of a cloud for IoT devices was possible.

Now, why these particular machine to machine communications are happening at the cloud for an IoT devices is to get the services of the cloud. The most important service is doing the insights with the IoT data and performing the business logic on it. And both together is called an action. So, therefore, this particular data, which is brought into the cloud to do the action part and to inference the action as per the business logic was called the intelligence. And thereby, now these three different developments are still happening.

(Refer Slide Time: 10:37)



So, this AWS IoT architecture, if we go into the different aspects, that is what are the services suits. So, let us go in more detail about this. Now, these things that is nothing but the devices which we are talking about. So, they will sense and perform an action on that particular sensor or data. So, that is, it will provide for that, it will provide the local triggers on an events to happen. That is called action part. And, this, to do this, the data has to be synchronized with the cloud in this particular case.

So, to do this action on the things or on the device itself, this particular AWS service has given a notion or a software which is called AWS Greengrass, that is on the edge. That software which runs on the edge with all the capabilities which cloud is having now, is being offered at the edge. Now, similarly at the endpoint that is at the device or the sensor level, the endpoints is being created by image and FreeRTOS. That is an operating system to be loaded on those particular devices so that it can connect with the gateway and thereby to the cloud for doing lot of intelligence.

So, the second part in this service suit of IoT AWS IoT architecture is done through the cloud. So, the cloud is giving the support to the IoT services such as storage, compute, and learning. So, let us see one by one how that is all being provided in the cloud for an IoT services. So, for that, I, AWS in the cloud runs a service or a platform or a software that is called AWS IoT core.

This AWS IoT core has lot of functionality or the support for an IoT services, such as it will support the device connectivity and also the messaging. So, I told you that this provides AWS core provides one endpoint of machine to machine communication between the things and

this AWS core that is the cloud. So, this particular connectivity and the messaging is part of this machine to machine communication. And the protocol which is supported here in AWS is called MQTT and HTTP and so on.

Now the next part is called the device management. So, AWS service, which is called AWS IoT device management. So, device management, you know that lot of devices, hundreds and billions of devices are now start connecting to the cloud. So, it requires a particular management into this particular cloud, and that is called AWS IoT device management.

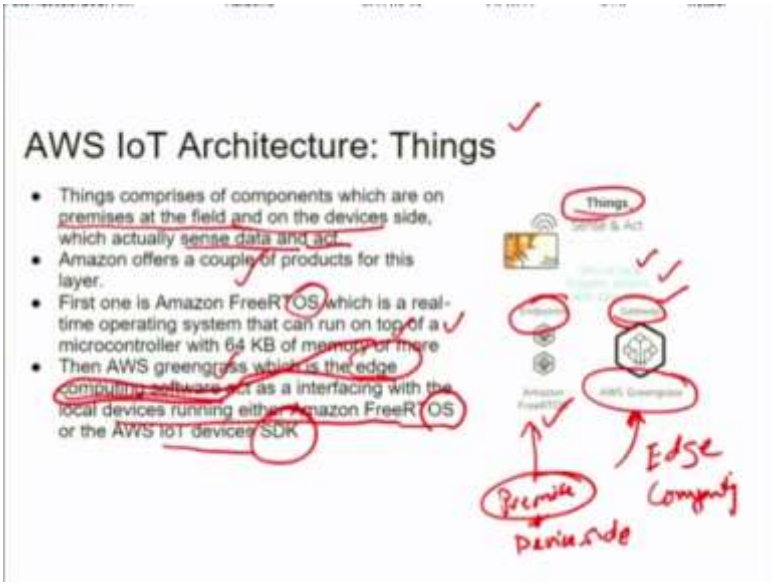
So, therefore, these devices will, that is AWS IoT device management allows this onboarding and management for whenever there is an update at is provided on the fly to these kind of devices which are to be supported through the cloud. So, AWS IoT device management is a service which is there into the cloud, which supports these things without going into more details, using this particular platform.

Now, the important part here nowadays becoming more and more important is called the intelligence, which is now done at the cloud. So, this intelligence on the IoT data is applied to bring up this logic and analytics for that application support. So, therefore, IoT device defender is there to take care of all the audit and protection when so many devices are now onboarding into the cloud for these services.

So, first thing is that it has to be secure and protected while getting these services out of these IoT data. When you say production means privacy, for example, sometimes you do not want your private information to be sent to the cloud, or let us say you do not want, your private information should be available to other party. So, for that, this AWS IoT device defender is available to do all kind of fleet production, fleet audit and production.

So, after passing through this IoT defender, now the data which IoT sends requires to do the analytics and it for the intelligence out of that so that the business logic can be applied and action can be taken. So, for that, there is a service which is called AWS IoT analytics, which is there inside the cloud. So, all this is called AWS IoT architecture and they become the service suit.

(Refer Slide Time: 16:01)



So, let us start one by one all these different components that is three layers, often AWS IoT architecture, the first one is called the things. So, here the things comprises of the components which are on the premise, at the field and on the device side. So, they are at the premise, at the device side. So, there the sensors and actuators are operating along with that environment and they want these actuators wants the action to be enforced or implemented.

So, Amazon offers a couple of products for this layer that is, the layer which is running at the premise that is, at the source. So, the first one, which Amazon offers at this level that is at this end called operating system, which is called Amazon FreeRTOS, which is a realtime operating system and which can run on top of the microcontroller with 64 kb of memory or more. So, this kind of micro controllers are attached with the device and there this operating system are loaded. So, this becomes the endpoint, which is very much essential to connect here in AWS architecture.

Then comes in the premise, very close to premise is an edge computing. So, edge computing is offering the cloud-like services or mimics the cloud services at the edge or at the premise. So, therefore, this service is called AWS Greengrass. So, it becomes a gateway in the sense it will also trigger, do a local triggers and on the events and perform various actions and also do the data sync. How that is all done, we are going to discuss in further slides.

So, AWS Greengrass is the cloud computing software, so is the edge computing software and it acts an interface with the local devices running either the Amazon FreeRTOS or AWS IoT SDKs. So, this is the two services which runs, which becomes the part of the things in AWS IoT architecture.

(Refer Slide Time: 18:44)

Now comes the second part in AWS IoT architecture that is called a cloud. So, there are two important aspects which you will see here in the cloud. One is called AWS IoT core. This is the first component, and the second component in the cloud is called AWS IoT device management. So, when we talk about AWS IoT core, so it is the core building block of this AWS IoT platform and is responsible for registering the device. So, it acts as the device registry.

Now this endpoint for machine to machine communication, with the things part on the other side is done through the protocols called MQTT Web Socket and HTTP for the devices to talk to each other and talk to the cloud, and also becomes a touch point for different application that wants to control the devices running in the field. So, therefore, this becomes a very, very crucial and important service for supporting the IoT applications. So, AWS IoT Core acts as the interface between the applications, for example, the mobile app that is talking to the device is similarly a device that is sending the sensor data to the cloud.

So, this IoT AWS core becomes the interface between the applications who want to control or talk to the devices which are deployed on the field. And also, these devices are sending the information of sensing or or sensor data to the cloud. So, the second part, which is also very important is called an IoT device management, which is a part of the cloud service. So, this particular IoT device management supports the bulk onboarding of the devices.

So, this is to deal with so many hundreds and thousands of the devices, IoT devices when start connecting how that is all done is done through the IoT, AWS IoT device management because registering one device at a time in the industrial use case is not feasible. So,

therefore, you have to support our, this AWS IoT platform supports the bulk onboarding of IoT devices to support the industrial use cases. So, it supports a bulk onboarding and also has the property like over the air software updates, maintenance, performing bulk jobs, operations, and so on.

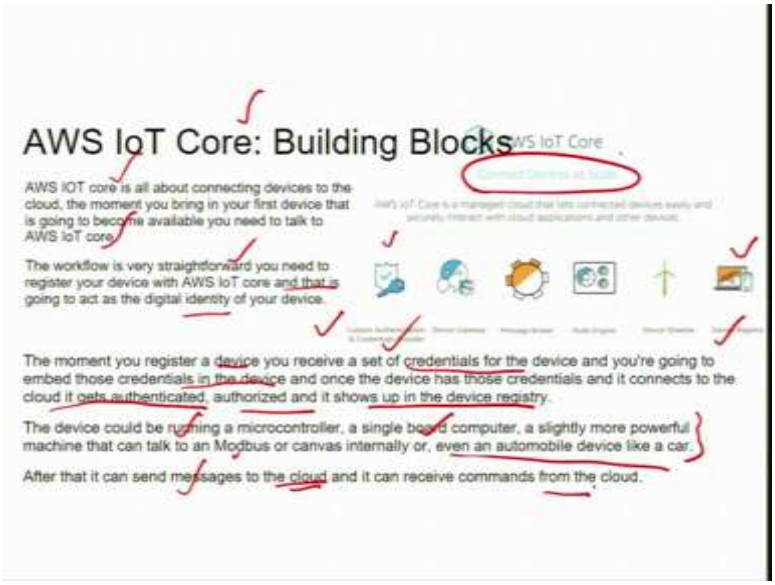
(Refer Slide Time: 21:31)

So, now next important component in AWS IoT architecture is called the intelligence. So, when you say intelligence, that means applying the business logic on that particular sensor data or the devices data, which is being sent and thereby, performing the actions based on those business logics. So, that is why this is called the intelligence.

So, AWS device defender, we have already explained as the important service is about the security and if there is any drift in the preferred configuration and the policy, that automatically raises an alert. So, it also provides highly secure footprints for all the devices, and if there is any anomaly, it raises an alert so that the fleet audit and production services there.

Now let us talk about another service, which is called AWS analytics. So, AWS analytics solution is a part of the intelligence layer and this particular service is responsible for analyzing the trends, visualizing from their feeding to a more powerful systems like the quick site, redshift and so on. So, thereby, this particular IoT AWS IoT analytics supports the IoT data analytics and intelligence either through the services which are being provided by the cloud or by the third party services that we will see in the next further slides.

(Refer Slide Time: 23:09)



So, AWS IoT core, the basic building blocks are summarized over here. So, AWS IoT core is all about connecting the devices to the cloud. The moment you bring the, your first device is going to become the available and you need to talk to AWS IoT core, if you want to connect your device to this platform to the cloud. So, the workflow is very straightforward. You need to register your device with AWS IoT core, and that is going to act as digital identity of your device. The moment you register your device, you receive the credentials for the device and you are going to embed these credentials in the device.

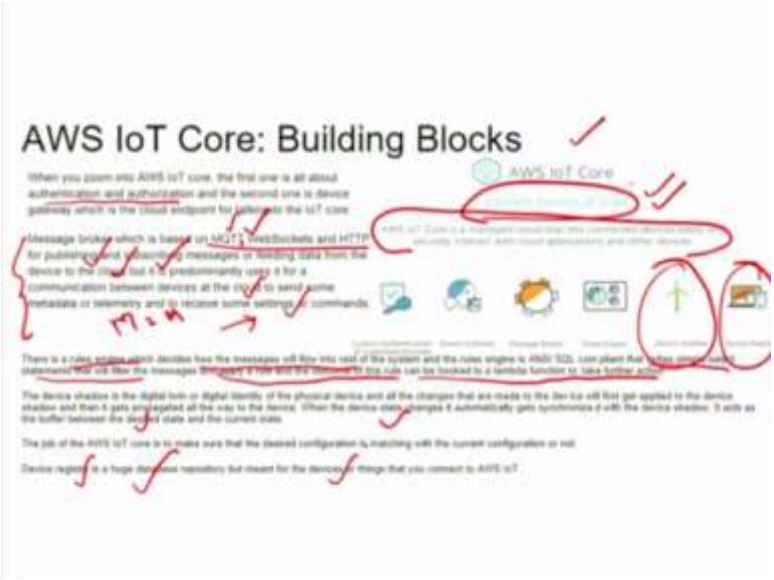
And once the device has those credentials, it connects to the cloud, gets authenticated, authorized, and it shows up in the device registry. So, these two services are very important when you want to connect your device or an IoT device to the cloud, that is called device registry of an IoT core. And then, from that device registry, you will get the CUS authenticated your device. So, customer authentication and credential provider is another service which runs in IoT core.

So, the device could be running in the microcontroller that we have already seen in the things part or on a single board computer and slightly more powerful machine, and therefore, even support the automobile IoT device such as self-driving car and so on or a connected car. So, this particular aspect that the connecting the device at a scale provides the industrial use case and we will go ahead and let us see at the end what are the other specific industrial use case.

Now it can send the messages to the cloud. That means once the device is registered at the AWS IoT core using device registry and it gets authenticated and using its credentials, it can

send the message to the cloud cloud and it will also receive the commands that is the actions from the cloud.

(Refer Slide Time: 25:25)



So, let us see moving forward. So, AWS IoT core, you know that it first authentication, authorization and the second one is the device gateway. So, which is the cloud endpoint for talking to the IoT core. So, the device using this device gateway will do this using the message broker service. And this message broker service will perform this telemetry using the protocols called MQTT web socket and HTTP four publishing and subscribing the messages or feeding data from the device to the cloud.

And it is predominantly uses a communication between the device and the cloud to send some metadata or telemetry or to receive some settings or the command. So, all these things are explained very much and therefore the connectivity with the device, an IoT device many times needed and to receive some kind of setting, some kind of commands. So, that is going back and forth between an IoT device and the cloud and uses the one end point at the cloud, which is also called as MQTT.

So, this kind of communication is called machine to machine communication because at one side that is called the devices or the things, the other side is the cloud. So, that communication is called machine to machine communication and is supported by various protocols such as MQTT, web Socket and HTTP that is all supported by AWS IoT core. Therefore, AWS IoT Core provides, uses all these standard protocols. You do not have to now go into into more detail about it. It is already used up here in AWS IoT core and thereby allowing to connect the devices at a scale.

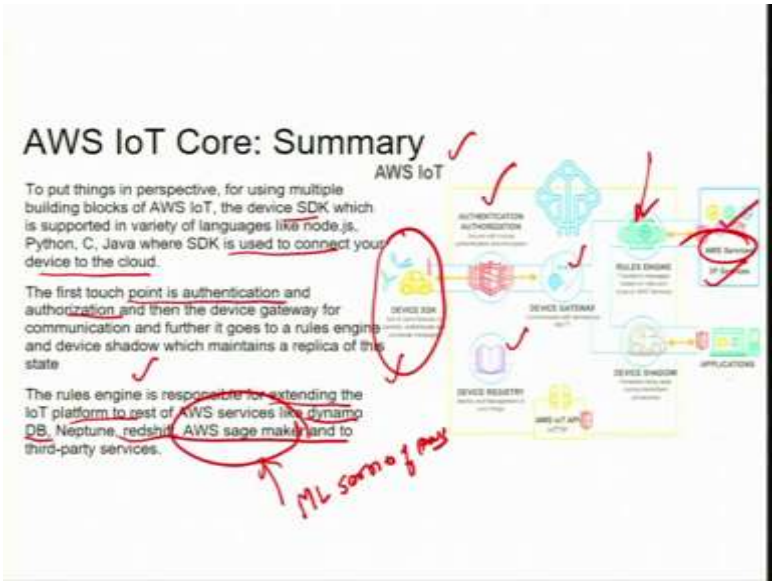
So, therefore, AWS IoT code is a managed cloud that lets connected devices easily and securely and interact with the cloud application and other and other devices. So, this is an important definition or component explanation for an AWS IoT building blocks. So, now coming to this particular rules engine. So, this, there is another service inside AWS IoT core called, called rules engine, which decides how the messages will flow in the rest of the system.

And the rules engine is ANSI SQL compliant that writes the simple select statements that will filter the message and apply the rule and the outcome of this rule can be hooked to the Lambda function and that Lambda function will perform the action part using these rules engine.

Now to maintain the current status of the device and configuration, there is a service which is called a device shadow. So, device shadow is nothing but a kind of digital twin or a digital identity of this physical device. And all the changes that are made to the device will first be applied to that digital shadow or digital twin of that particular device and then gets propagated all the way to the device in this case. So, when a device state changes, it automatically gets synchronized with the device shadow and it acts as a buffer between the desired estate and the current estate.

So, this is very important concept which will allow these devices to operate not only connect, but to operate at the scale with the help of device shadow. So, the job of AWS IoT core is to make sure that the desired configuration is matching with the current configuration or not. So, the device registry, when you talk about this aspect or functionality of the AWS IoT core, so device registry is a huge database repository, maybe key value stores that we have discussed in another lecture, but it meant for the devices or the things that you connect to the AWS IoT.

(Refer Slide Time: 29:55)

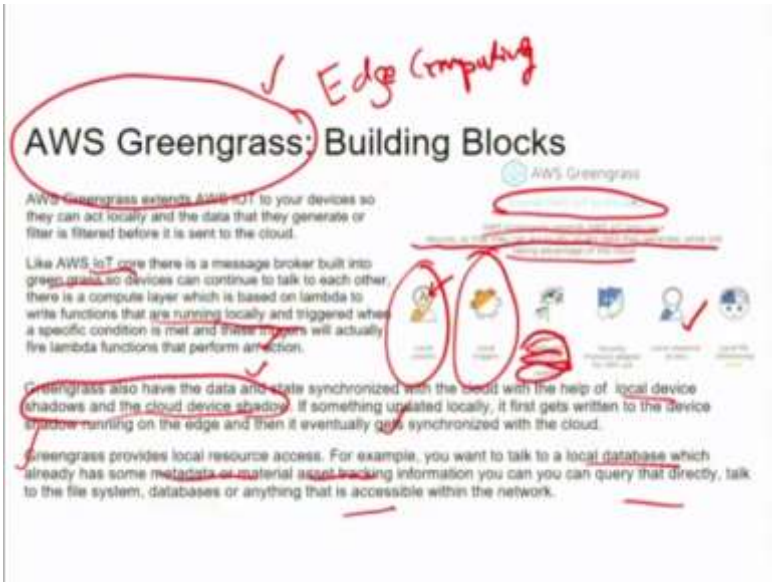


So, let us summarize about AWS IoT core architecture. So, put the things in the perspective for using the multiple building blocks of AWS IoT core. That is the device SDK, which is supported in a variety of languages, and it is used to write a application for the devices. And these devices allows the SDKs to connect to the cloud.

So, the first touchpoint is the authentication and authorization here in this case. So, this we have explained that devices SDK are being supported and it is nothing but the set of client libraries to connect and authenticate and exchange the message. So, here this example is shown showing that the physical objects, which are embedded with these things or the devices is also running the SDK that is device SDK. And this particular device SDK becomes the device of an IoT device or a thing which will be connected to the AWS IoT.

So, the rules engine is then responsible for extending the IoT platform. So, this authentication and authorization we have discussed. Device registry also we have discussed and the device gateway also we have covered. Then coming the rules engine. The rules engine is responsible for extending the IoT platform to the rest of AWS services, such as to the rest of AWS services like Dynamo db, that is no SQL data store or key value store that we have covered in the previous lecture. Redshift, AWS sage maker, sage maker is a machine learning service of AWS. It can also use the third party services here through the rules engine.

(Refer Slide Time: 32:07)



So, let us see another component which is called AWS Greengrass. This is to support the edge computing through the service that is called AWS Greengrass. So, AWS Greengrass, as I have told that it extend the AWS IoT to the edge. So, AWS Greengrass extends a AWS IoT onto your devices so that they can locally on the device, they gen on the data they generate while still taking the advantage of the cloud. So, this is an important development in IoT and the cloud. So, this development is given by the or is an innovation of the cloud that is called edge computing. And this AWS cloud is providing this service with the help of AWS Greengrass.

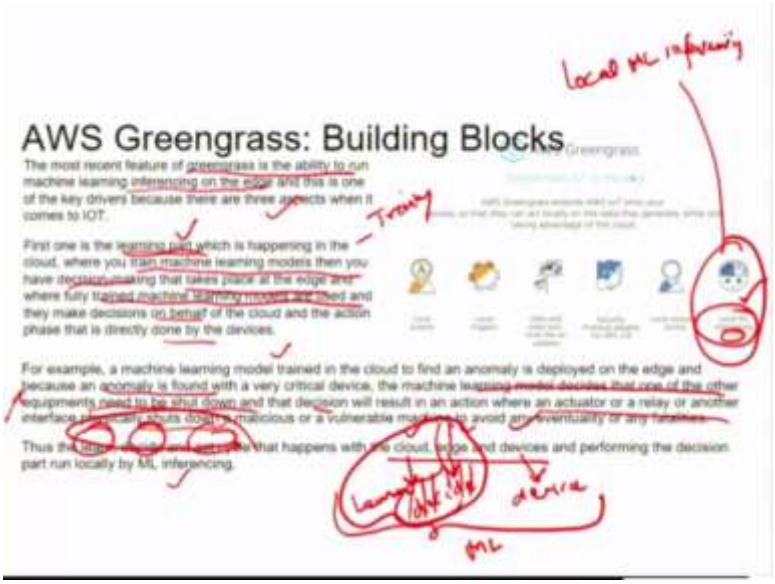
Like AWS IoT core, there are, there is a message broker built in the, in the Greengrass so the devices can talk to each other, so devices can talk to each other. And there is a compute layer, which is based on the lambda function to write the function that are running locally and triggered when a specific condition is met. These triggers will actually fire the Lambda function and perform the actions.

So, these are the local actions and this is the function. So, Lambda function will allow this particular local triggers to take this particular action. Now, Greengrass also have the data and the state, which is synchronized with the cloud, with the help of local device shadow. So, local device shadow and the cloud device shadow, if something updated locally, we have already told that this device shadow is running on the edge and then it eventually gets synchronized with the cloud. So, data and state synchronization is done through this particular service, which is called the device shadow.

Now finally, the Greengrass also provides the local resource access. For example, if you want to talk to local databases, which is already has some metadata or method or, material asset

tracking information, then you can query and talk to the file system databases. And these are all called local resource access.

(Refer Slide Time: 34:46)



Now, most important part, which is very new is called the local machine learning inferencing. So, this is the most recent feature in the Greengrass, is the ability to run machine learning inferencing at the edge. Now, this is one of the key drivers because these three aspects when it comes to an IoT doing machine learning. So, the first one is when you do the machine learning, so that is called the learning part and which is happening in the cloud.

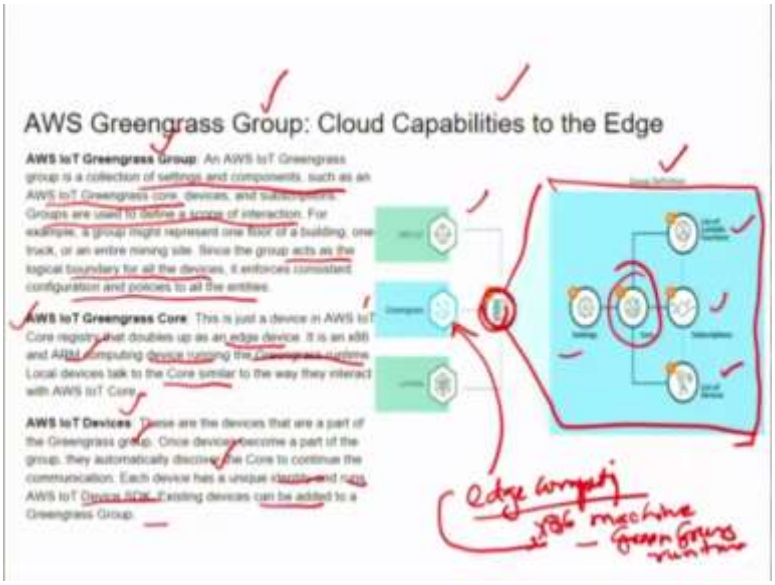
So, learning is sometimes also called the training is happen in the cloud. And when you train the machine learning model, then you would, you would like to use it for making the decisions. And that particular in you want to take place at the edge that is locally and where the fully trained machines models are used and they make the decisions on behalf of the cloud and the action phase that is directly done by these devices.

So, for example, a machine learning model, which is trained in the cloud to find some anomaly is deployed at the edge. And because this anomaly is found and with a, with a very critical device, so the machine learning model decides one of the other equipment to be shut down and paving the way to work that particular device and the decision will be taken locally where the actuators or the relay will become the interface and the follow the instructions which is derived automatically at the local level.

Thus, the learn, decide, and act cycle. These three we have explained in the intelligence part, which was happened earlier with the, with the cloud edge and the device at three different

levels and the decision part locally on the inferencing. So, let me explain again. So, when you say learn, so learning or the training will be done at the cloud. When you say decide, so, this particular logic that is called influencing is now happening at the edge. And when you say act, that action has to be taken at the level of the device. So, this is to be implemented with the help of machine learning. So, machine learning is supporting using the AWS Greengrass, these particular aspects, and this is also called local machine learning in inferencing

(Refer Slide Time: 37:32)



Now AWS Greengrass. When we talk about the group of services, for example, if, let us say in an organization the sensors are deployed and doing a particular task, for example, a smart city of a particular city. So, therefore, AWS manage those related devices or using that related service called a group. So, AWS Greengrass group is nothing but bringing the cloud capabilities to the edge.

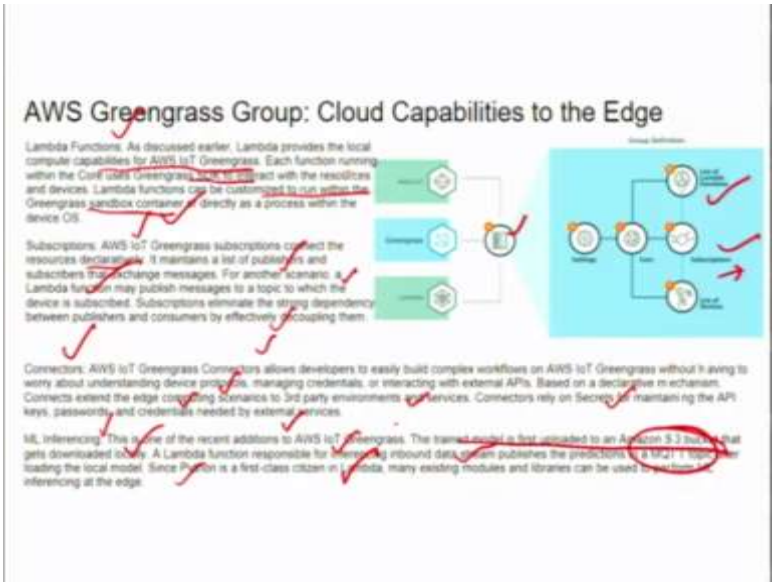
Let us see about this. So, AWS IoT Greengrass group is a collection of the settings and the components such as AWS IoT Greengrass core. So, this particular Greengrass will have this particular group of services. So, it will start with the group definition. So, the groups are used to define the scope of interaction. For example, the group might represent one floor, a building or a truck, or the, the entire mining side. Since the groups acts as the local boundaries for all the devices, it enforces the consistent configuration and policy to all the entities. So, this becomes a very, very important service.

So, this AWS IoT Greengrass core, this is just a device in AWS IoT core registry that doubles up as an as device. So, it is a 3 86 arm computing that is running, that is the device that is called an edge computing. And this is enabled with the help of this particular support that is

386 machine or a device. And this particular machine runs the Greengrass and this brings into the edge computing. So, this runs a Greengrass runtime. So, runtime is like an operating system service. So, local devices can talk to the core similar to the way they interact with AWS IoT core.

Now AWS IoT devices that are the part of the Greengrass group. These, once the device becomes the part of the group, they automatically discover the core to continue the communication. Each device has the unique identity entrance, IoT SDK, and existing devices can be added to this particular group. So, all these functions are used here as a part of the group definition, which is being provided by IoT Greengrass group capabilities.

(Refer Slide Time: 40:43)



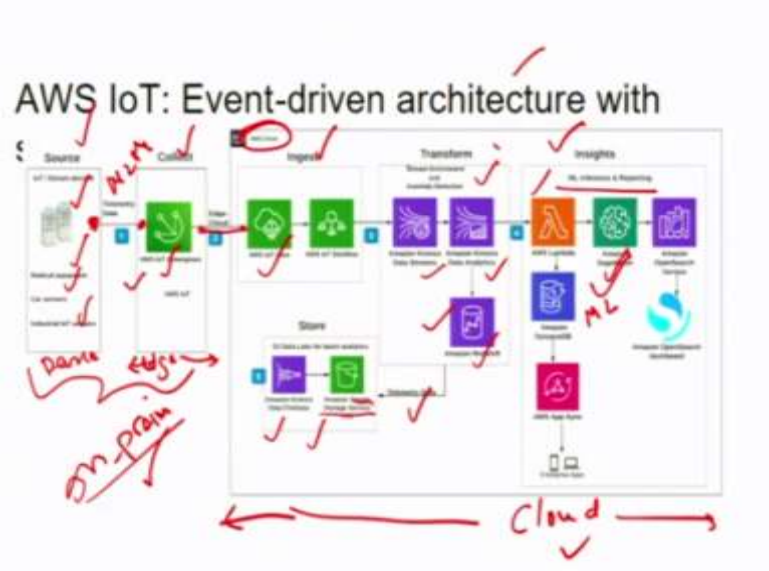
Now then comes the Lambda function. So, Lambda function provides the local computing capabilities for IoT Greengrass that is at the edge. And these functions uses the Greengrass SDK to interact with the resources and the devices. So, lambda function can be customized to run within the Greengrass using the containers or directly as a process as the device operating system. Now coming to the subscriptions, that is AWS IoT Greengrass subscription connects the devices and it maintains the list of publisher and subscriber that exchange the messages.

So, you know that the protocols like MQTT operates in this manner of the publisher and subscriber to exchange the messages, and they sometimes uses the topic for this kind of connectivity. So, the subscription eliminate the strong dependency between the publisher and consumer by effectively decoupling that.

Now coming the various connectors. So, IoT Greengrass connectors allows the developer to build easily complex workflow without having to worry about understanding of these protocols, managing credentials and so on using external API. So, this connect extends this edge computing scenario to the third-party environment, and the services and connects rely secret for maintaining APIs.

So, machine learning inferencing, if you talk about here in supported in IoT AWS Greengrass. So, this is one of the recent additions that we have already covered in AWS Greengrass, the train model is first uploaded to AWS S 3 bucket, simply storage bucket gets downloaded locally. And the lambda function is responsible for influencing of an inbound data streams, publishes the predictions using MQTT topic and then loading that local model. So, the Python is used in in the lambda functions and the libraries to perform this inferencing at the machine learning for the machine learning.

(Refer Slide Time: 43:06)



So, this is the architecture for an AWS IoT, which is an event-driven architecture. So, here you know that the source at the source level are one end of this pipeline that is nothing but the deployment of IoT devices. They sometimes also call the extreme devices, so such as the medical equipment, the connected car sensors or industrial sensors. Now these source or the devices they use this, the database generate. This is the telemetry data. And using the protocols, it will connect to the edge, which is, which is on onsite on preim.

So, this service is called AWS IoT Greengrass will take one endpoint. This is M to M communication and this particular action is called to collect the data. So, this data is now further sent to this AWS IoT core. So, AWS IoT Greengrass is now having the proper

connector for connectivity through the protocols, either web socket or HTTP or MQTT, depending upon different applications. So, it will use the service, which is called ingest data ingestion.

So, that data is brought into the AWS cloud. So, this entire aspect is the cloud. This is an edge and this is the device. So, both are in on pre. This is the cloud. So, data ingestion will bring the data into this particular AWS IoT code. And then data is given to these services. Let us say that anomaly detection and various analytics will be performed in that data stream and then use the database.

Let us say that Redshift is one such example shown here in this architecture. And this kind of telemetry data is now made persistent in some AWS storage service called Simply Storage Service are using some database, which we have al already seen in the previous classes. So, once this transformation is done or a pre-processing is done, then it is given to the machine learning model for making the inferencing. So, machine learning, inferencing and reporting is done in the following manner.

So, when the data comes, it will be applied on an insight for an insight using AWS lambda function and, for that it has to now perform this machine learning inferencing or machine learning using Amazon Sage Maker. So, Amazon Sage Maker will allow the machine learning on this particular data, which is available by this particular source. So, this shows all that details.

(Refer Slide Time: 46:43)

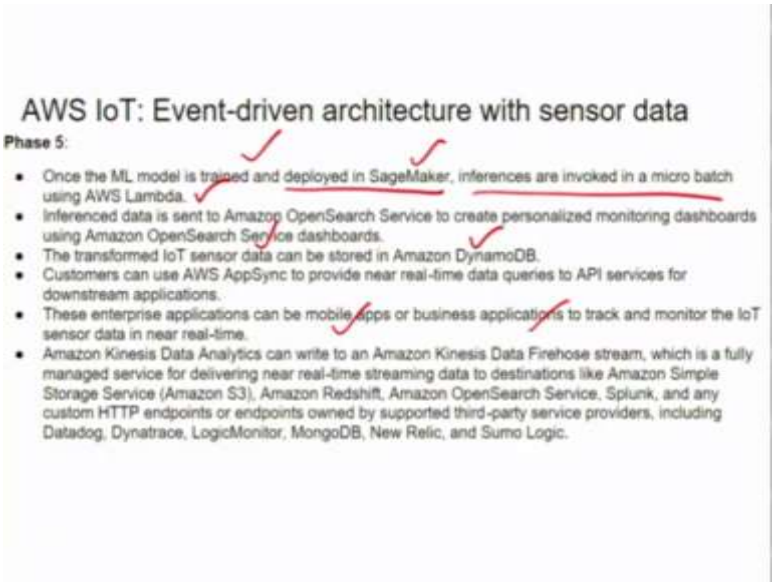
In this particular part. So, here, these pages we have explained in three different phases. So, data is originated from an IoT devices and this particular telemetry data is collected using IoT Greengrass. And then, when the data arrives, then it will react autonomously to the local event filters and aggregate the device. That is, it will be able to perform the influencing part using influencing part of a machine learning model.

When the data is ingested to the cloud that is, as to the cloud interface services such as AWS IoT core, and it connects and manage and scales easily and security as well. So, AWS core here will interact with the cloud services and other devices. So, for that we have already explained all that.

(Refer Slide Time: 47:42)

Now, phase 3 uses this particular stream ingestion so, so data is ingested and is available in the form of data streams. So, data is ingested to perform near realtime analytics. Various tools are also available to do this. Now, to do this analytics, you sometimes require to store it and therefore it requires the use of storage system. So, we have already explained that some kind of storage is very much needed in phase 4 and, this particular storage is now used for doing the machine learning model. So, machine learning uses the data, which is available for training the model and using the end-to-end machine learning thereafter.

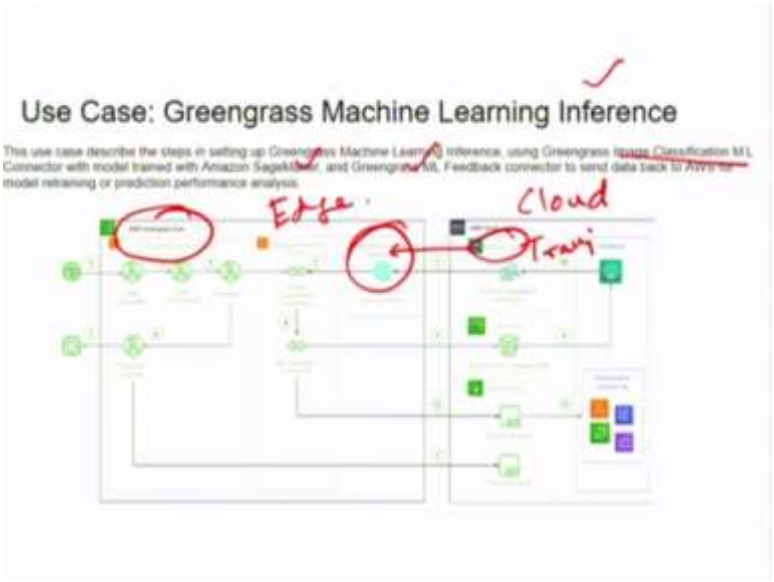
(Refer Slide Time: 48:41)



Fifth phase is, is more on this doing machine learning training at the cloud, and then that model will be deployed for making the influencing using the AWS lambda function on the edge that is. So, this sage maker will prepare the model and deploy to the AWS Greengrass. There, the it edge computing will perform the influencing part. So, for that, you know that it will be using the services often DynamoDB, that is the key value store which will store the sensor data, which is ingested into the cloud system, and then applied for using into the machine learning using the sage maker.

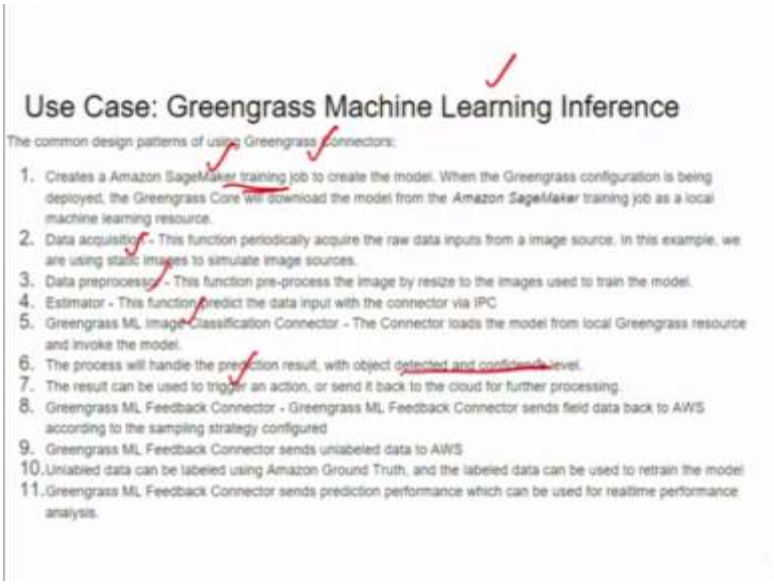
So, these applications may be many, it can run through the mobile app or business application to track and monitor the IoT sensor data in the real time possible. So, there are various public service available through public cloud or maybe third-party vendor. Some of them are written over here.

(Refer Slide Time: 49:56)



So, Greengrass machine learning inference, if you talk about, so this particular setting describes about some image classification, machine learning which is trained by, using the Amazon sage maker. And Greengrass will get this is the Greengrass that is the edge will be performing the influencing part. So, this is the machine learning influencing model, which is trained who are here in the cloud. So, the training is done over here. So, this is the train model which is being deployed. So, this is the Amazon sage maker which will perform the training, and that model will be deployed on the edge where the inferencing will be done.

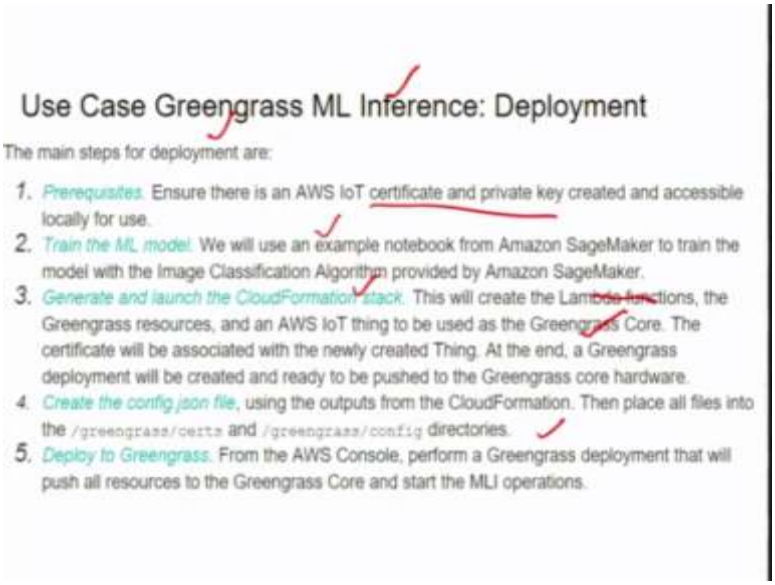
(Refer Slide Time: 50:45)



So, let us see the Greengrass machine learning inferences. This is the common design pattern of using the Greengrass connectors. It will create the Amazon sage maker for doing the

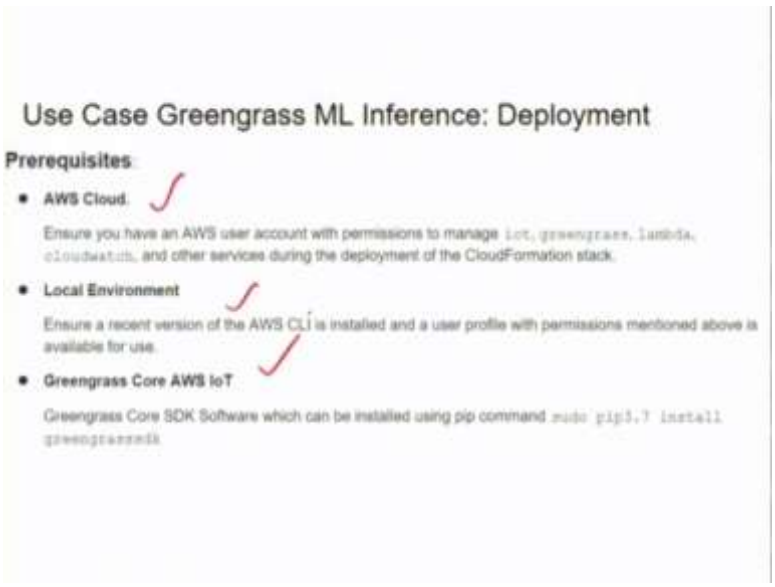
training. And once the training is done, then for training, it will do the data acquisition, data pre-processing, estimator, and the process will be able to handle the prediction results when the objects are detected with a particular confidence.

(Refer Slide Time: 51:15)



So, use case for the Greengrass machine learning inferences after the deployment. So, so the main step for the deployment are is that it requires the AWS IoT certificate and the private key to access locally. Then, it will train the machine learning model using SageMaker and then generate and launch the cloud formation stack using lambda functions at the Greengrass core. Then it will create various functions configurations in the directory, and then it will deploy to the Greengrass. So, these steps are written over here.

(Refer Slide Time: 51:54)

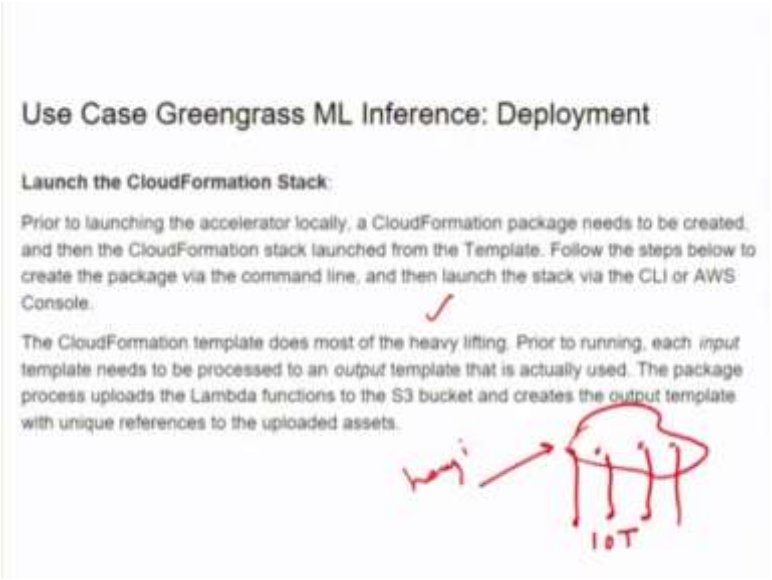


So, to do this, to understand this deployment, you require AWS cloud local environment, Greengrass, AWS IoT core is written in this particular slide.

(Refer Slide Time: 52:05)

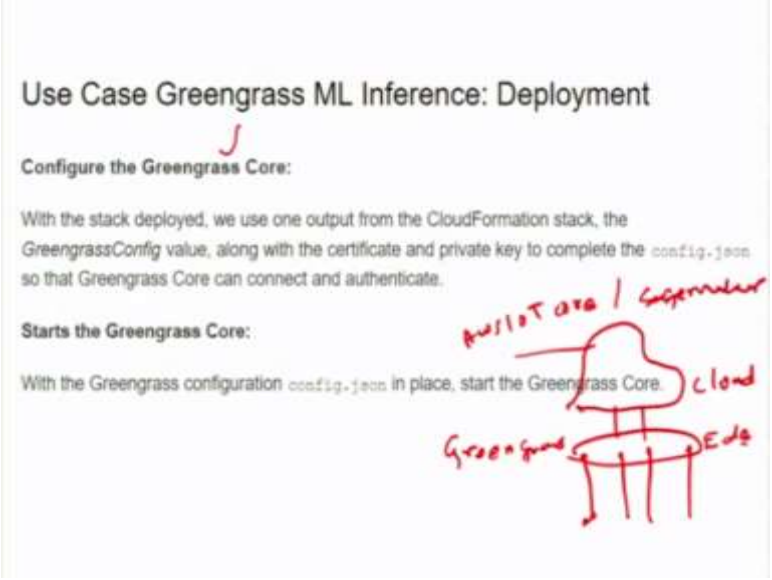
Then the training of a model will be done at the sage maker. And for that different steps are very much needed to connect to the sage maker and use the data set which is there to train the model. And then you select a particular machine learning algorithm for training this particular model. In this example, since image classifier is to be built or is to be trained, therefore, it uses some kind of machine learning.

(Refer Slide Time: 52:42)



Let us say that CNN convolution neural network, we have already discussed in the previous sessions, then it will launch this cloud formation stack. And you can see that once this particular cloud formation template is used for heavy lifting. So, heavy lifting means if this is the cloud and the deployment requires so many devices which are connected to the cloud, maybe that is smart city across a particular country, several smart cities now they start connecting, sending their data to the cloud. So, it requires this cloud formation, a heavy lifting of the IoT data.

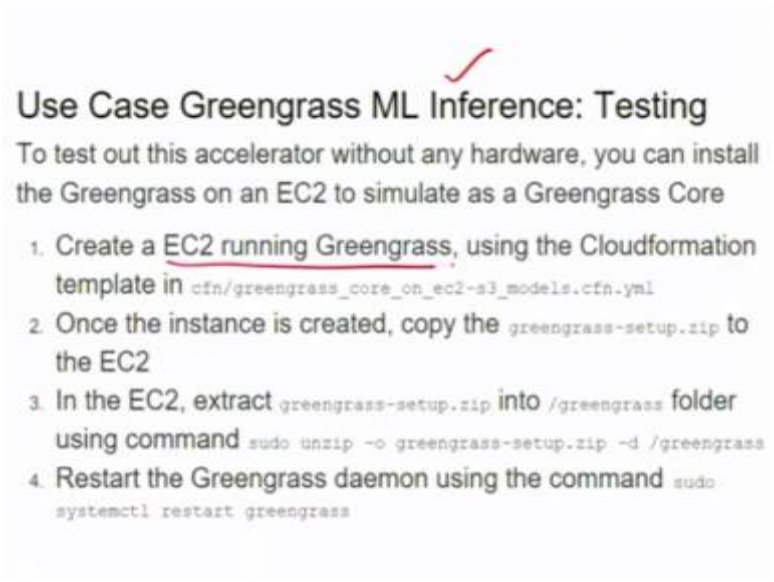
(Refer Slide Time: 53:24)



So, not to send all the heavy lifting data. So, the cloud services are already provided by the, by the edge. So, you can see that there is a layer, which is called an edge. And out of the edge very few data will go to the cloud. So, that edge is in the form of a Greengrass that is edge

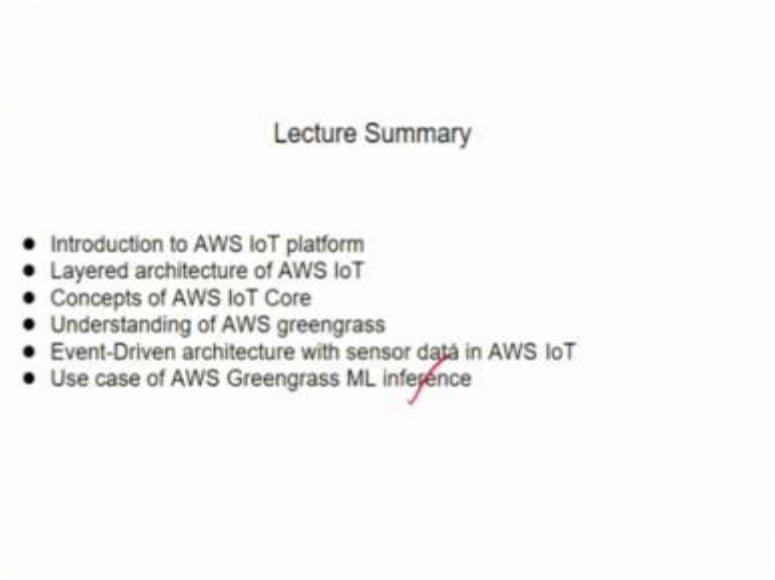
layer. And here you can see that it is an AWS IoT core and it uses the machine learning called sage maker for training the model.

(Refer Slide Time: 54:16)



So, then once the model is trained in the sage maker or in the, in the cloud AWS cloud, then it will be deployed using the Greengrass group into the, the Greengrass core. So, this is the use case of a Greengrass machine learning inference: testing. So, you have to create the easy two running instance running the Greengrass, and then it will deploy all that command.

(Refer Slide Time: 54:46)



So, let us summarize what we have covered in the lecture. We have covered another platform that is provided by the cloud AWS to do the IoT that is called AWS IoT platform. We have seen the layered architecture of AWS, IoT. We have also explained the concepts of AWS IoT core. Then, this particular newer development, which is called edge computing using this AWS IoT that is called AWS Greengrass.

So, then we have used these concept IoT core and a Greengrass to form an event driven architecture for a sensor data to be ingested and being processed into the cloud. So, we have also shown you that how you can use the machine learning inference trained model using the cloud in sage maker. And then the train model is deployed on the edge that is using AWS Greengrass and then how the influencing is done and various applications, they are being supported. So, that is all for this lecture. Thank you.