Builder Session re:Invent 2018

Machine Learning in the Edge - Enhanced by Amazon Sumerian

Simith Nambiar

IoT Architect, AWS Professional Services

INTRODUCTION

In this session we are going to build a demo around **Worker Safety Enforcement**. We will bring together Machine Learning Inference on the Edge using **AWS Greengrass** and use the Predictions or Inferences to provide guidance to Workers using the **AR/VR Service Amazon Sumerian**.

WORKER SAFETY ENFORCEMENT IN FACTORIES - WORKING BACKWARDS FROM THE USE CASE

Worker safety in sites has been the top priority for Mining, Construction, Oil and Gas industries. As there are various contractors involved in Site operations it is becoming increasingly difficult to enforce that the Workers go in with the right safety gear like Helmets and Vests. Security personnel who are entrusted with this Safety Enforcement are tasked with administrative tasks like verifying ID cards, checking schedules etc. and hence cannot be entrusted the task of making sure the Workers are properly geared for the site. We are going to build a demo showing how Security cameras can take on this job of monitoring Workers as they walk into their Sites and have a real life like character advise worker on Worker safety when there is a breach!

WHY MACHINE LEARNING INFERENCE ON THE EDGE?

Applications that combine edge computing and machine learning (ML) are enabling new kinds of experiences and new kinds of opportunities in industries ranging from Mobile and Connected Home, to Security, Surveillance, and Automotive. Products that combine these two technologies can leverage the local compute and storage capabilities of new types of edge devices, and perform a variety of actions on their own. This can save time, improve privacy, reduce network traffic, and enable applications or devices to be optimised for specific environments.

WHAT IS AWS GREENGRASS?

AWS Greengrass is software that lets you run local compute, messaging, data caching, sync, and ML inference capabilities for connected devices in a secure way. With AWS Greengrass, connected devices can run AWS Lambda functions, keep device data in sync, and communicate with other devices securely – even when not connected to the Internet.

WHAT IS AMAZON SUMERIAN?

Amazon Sumerian lets you create and run virtual reality (VR), augmented reality (AR), and 3D applications quickly and easily without requiring any specialised programming or 3D graphics expertise. With Sumerian, you can build highly immersive and interactive scenes that run on popular hardware such as Oculus Go, Oculus Rift, HTC Vive, HTC Vive Pro, Google Daydream, and Lenovo Mirage as well as Android and iOS mobile devices.

WHAT WILL BE BUILD TODAY?

Fig: Detection and enforcement of Vests on Workers

BUILD INSTRUCTIONS:

AWS ACCOUNT, AWS REGION AND LOGIN DETAILS

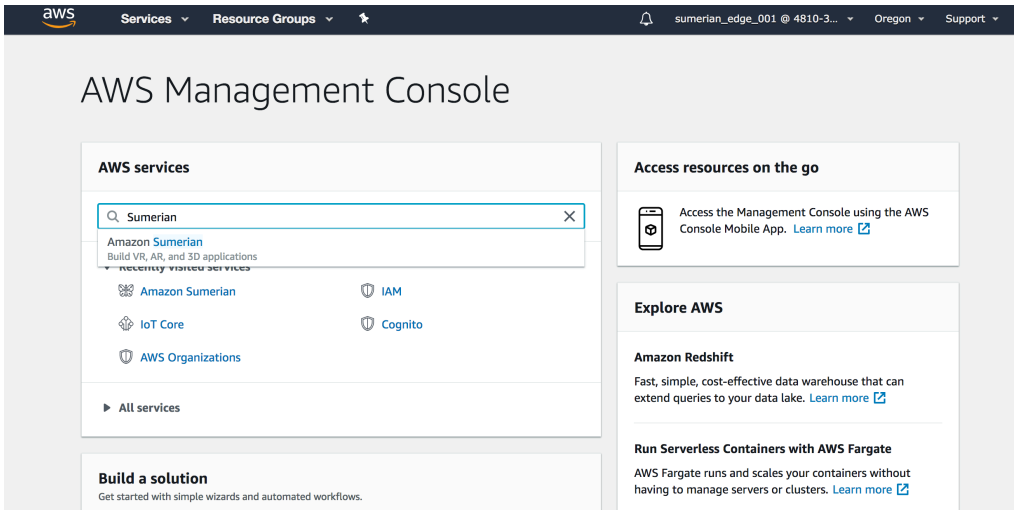
You do not need to use your own AWS Account. An AWS username and password with Login instructions will be provided to you by the Instructor. Region to be used will be **Oregon (us-west-2**.

AWS Console Access: https://481033735235.signin.aws.amazon.com

LOGIN WITH THE USERNAME AND PASSWORD PROVIDED TO YOU BY THE INSTRUCTOR

ACCESSING THE AMAZON SUMERIAN CONSOLE

Please type **Sumerian** *in search field of the AWS Console and select Amazon Sumerian*



ⓘ Please note the region - Oregon (us-west-2)

CREATING YOUR FIRST SCENE

DASHBOARD

When you first sign in to Amazon Sumerian, you see the Dashboard home screen. From the home screen, you can:

- Access your account in the top right.
- Access recently edited scenes.
- Create a new scene using a wide selection of templates. The most common template is the **Default Template**.

Below your Recent Scenes is a list of **Scene Templates**. There are several templates available, some of which are used in other tutorials. From the templates below, we will use the **Speech and Gestures** template,

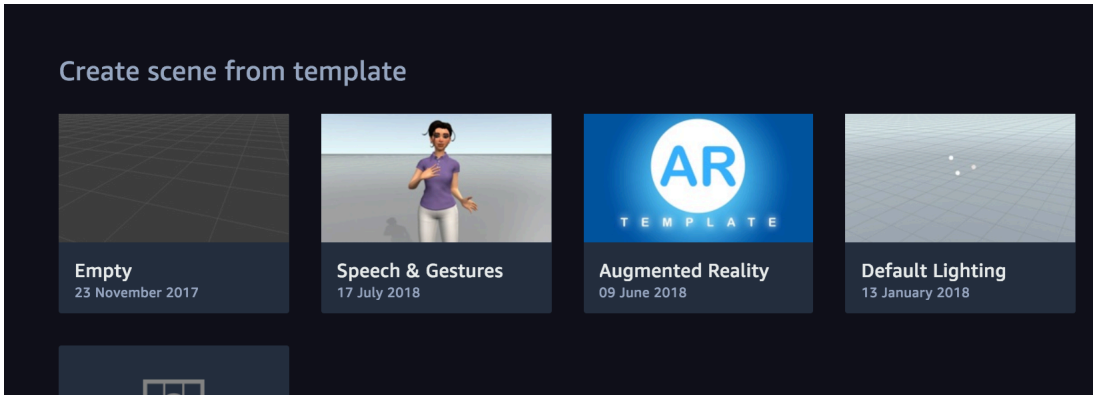
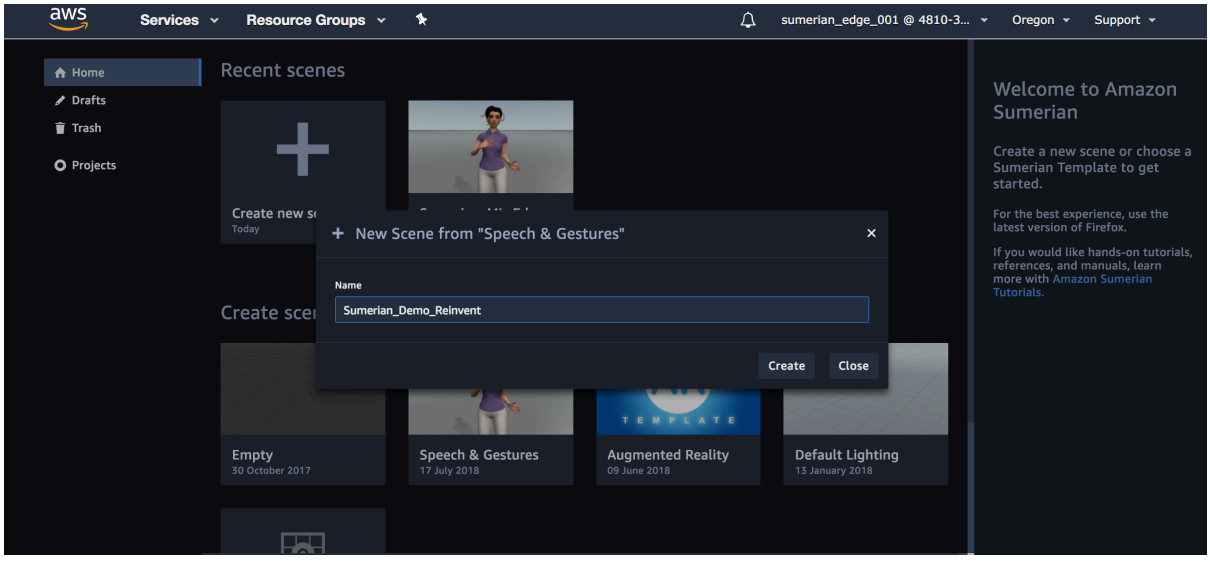


Fig: Speech and Gesture template

By Selecting the “Speech & Gestures” template let us create our first scene from it. Name your template whatever you like it to be called and then click Create. Please wait for Sumerian to create your scene in the background and display it.



THE AMAZON SUMERIAN INTERFACE

As the new scene is loading, please read through and get acquainted with the Amazon Sumerian user interface: https://docs.sumerian.amazonaws.com/tutorials/create/getting-started/sumerian-interface/

Congratulations! You have just created your first Sumerian scene.

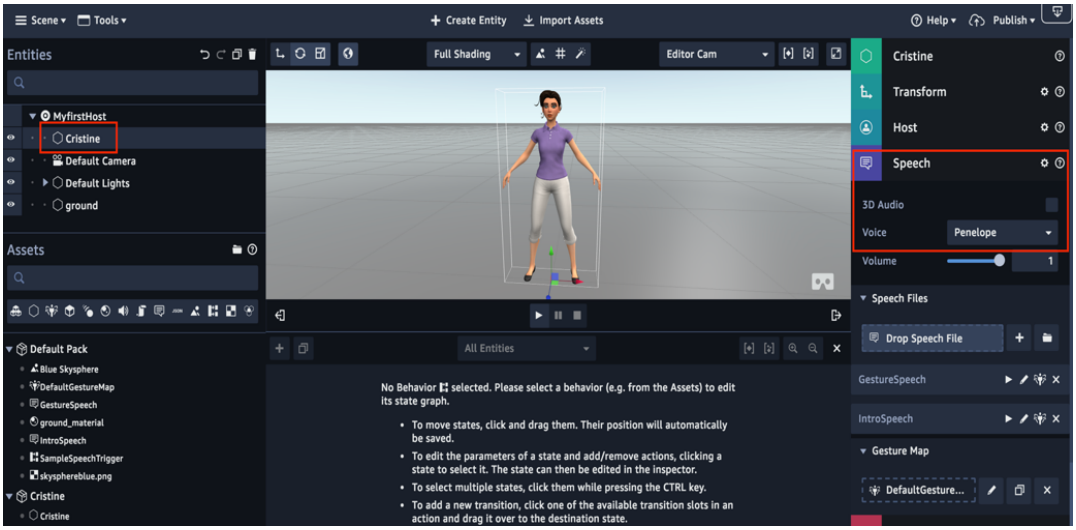
MAKING THE HOST SPEAK

Our Host for this scene is Christine. Christine can gesture and speak. Try playing the scene by clicking on the Play button in the bottom centre of the screen. *Amazon Sumerian would complain with a popup box saying you do not have an Identity Pool Configuration setup*, **this is expected**. An Identity Pool that would give Christine the required permissions to do a Text to Speech conversion via Amazon Poly has already been setup. Let us enter that into the **AWS Configuration**. Click on your scene name on the left-hand side and then, on the AWS configuration menu on the right-hand side of the screen (Please see the below fig.).

Fig: AWS Configuration menu

Enter the Identity Pool Id. **us-west-2:73d6f77b-bf39-4e41-8d42-c2f0930f3fbe**. The Identity Pool Id (Unauthenticated Role) has been configured to provide access to Amazon Polly and AWS IoT Services. Try Playing the scene now, Christine will speak a few sample sentences. Great going, let us now connect the scene to AWS IoT, so that when our Camera detects a missing Vest, it can report directly to Christine, who can give further instructions to the Worker.

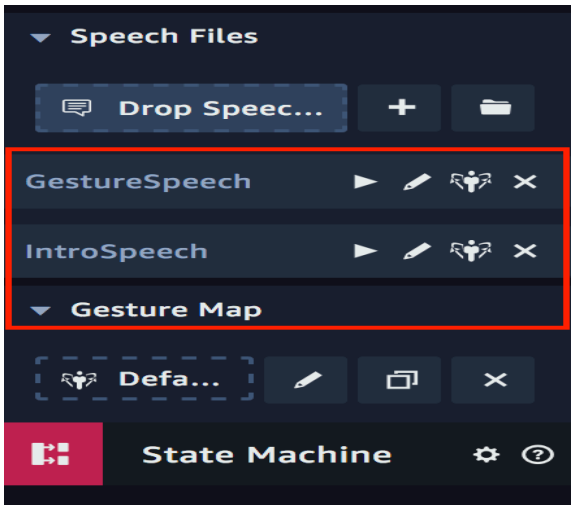
Follow this tutorial to learn how to make a host speak
https://docs.sumerian.amazonaws.com/tutorials/create/beginner/host-speech-component/



You can find from the below document what is the voice name belonging to the language and accordingly you can change the voice in your Sumerian scene.

https://docs.aws.amazon.com/polly/latest/dg/voicelist.html

Please remove the default speeches added to the scene by clicking cancel as shown below



ADDING A DISPLAY AREA TO THE SCENE

Let us import a Television to the screen to show the predictions coming out of our Camera device. To do so, Click on Import Assets on the top centre of the screen, and search for Television as shown in the image below:

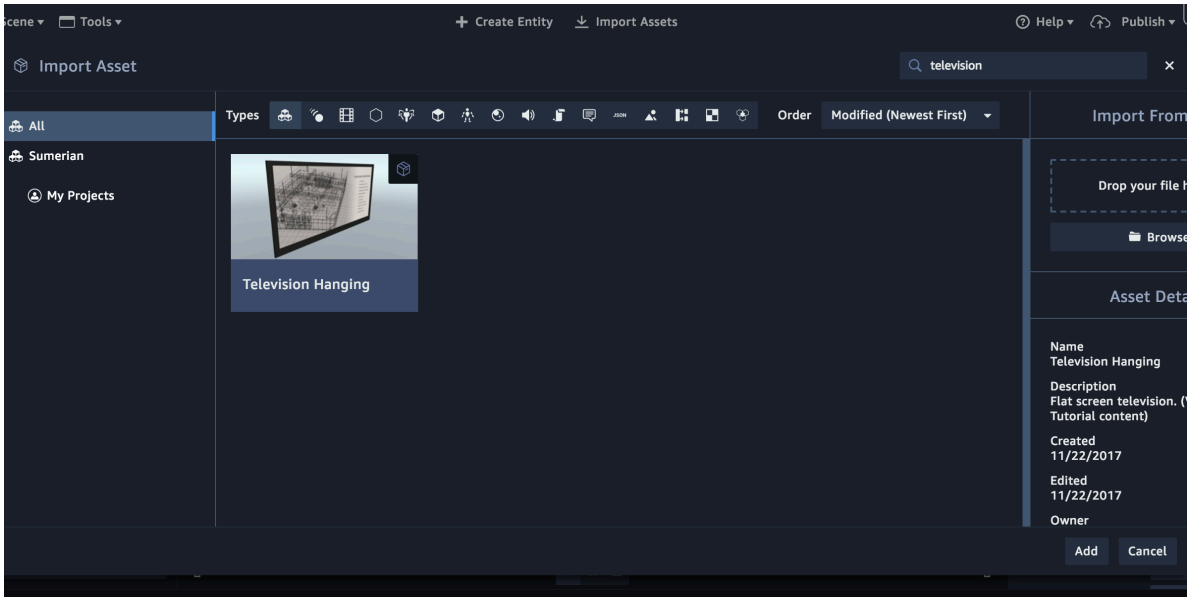


Fig: Importing the Television into the scene

The Television will be added to the Assets Panel. In order to be a part of the scene we need to place it in the Entities Panel. Expanding the **Television Hanging** asset from the Assets Panel, Drag and drop the television_wall_ViewRoom.fbx into the scene as shown below from the Assets Panel.

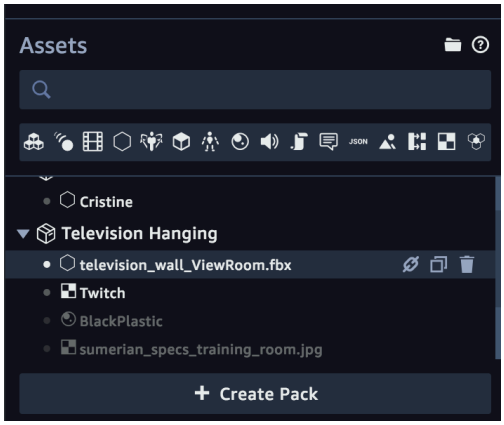


Fig: Choosing the Entity from the Television Hanging Asset

Add an HTML 3D entity to the scene to display messages on the TV. To do that, click on Create Entity on the Top centre of the screen, and choose HTML 3D under the Others section. Finally, set the Transform values of Christine, Television and 3D HTML Entity to the below values as shown in the figure below.

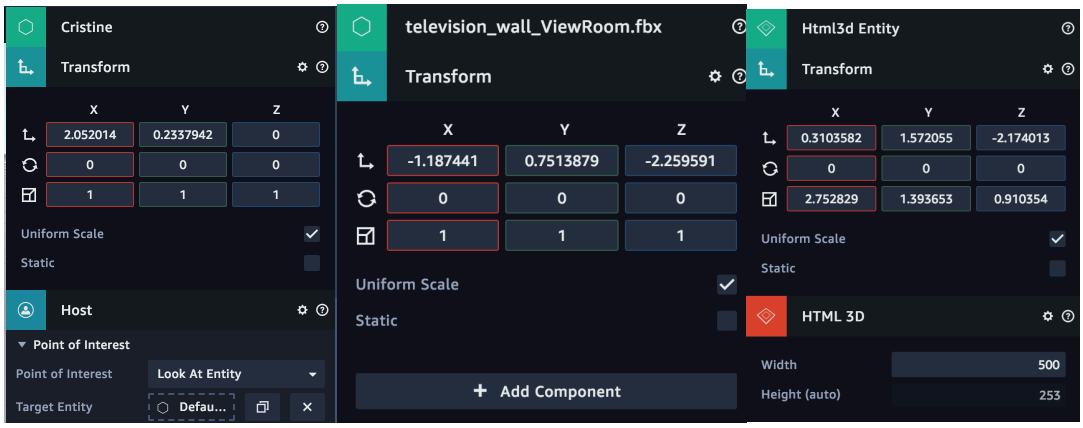


Fig: Transform values for the various entities

Your scene should now look like the below:

Fig: The scene

CONNECTING THE SUMERIAN SCENE TO AWS IOT

To connect the scene to the AWS IoT Service to receive predictions, let us attach a script file to the Host entity. The script will connect to AWS IoT and subscribe to a particular MQTT

topic to receive Predictions. To add the script, **click on the Cristine entity** on the Entities Panel and then at the bottom right corner, click on + Add Component and select Script. Please see below for visual instructions on adding a script.

Fig: Adding a script to Cristine Entity

The Script component is now added on the Right-hand Side of the screen to the Cristine entity. Now to add a script file, please see the below figure:

A new script is created. To view the Script, just hit **J** on your keyboard. Sumerian Text editor will open in a new window. Click on the newly created Script named "Script". Notice that there are various functions like setup, clean-up, enter, exit etc. These functions are fired at different stages of the Scene and Entity lifecycle. Clear the content of the Script file and then Copy and Paste the script.js file from the following GitHub link,

GitHub link for script file:
https://github.com/simith/reinvent_edge_sumerian_workshop/blob/master/script.js

into the window in the Sumerian editor. Replace the **AWS_IOT_Endpoint** provided to you by the instructor in the placeholders. **AWS_IOT_ENDPOINT_REPLACE_ME** placeholder in the Script file. Please see below figure.

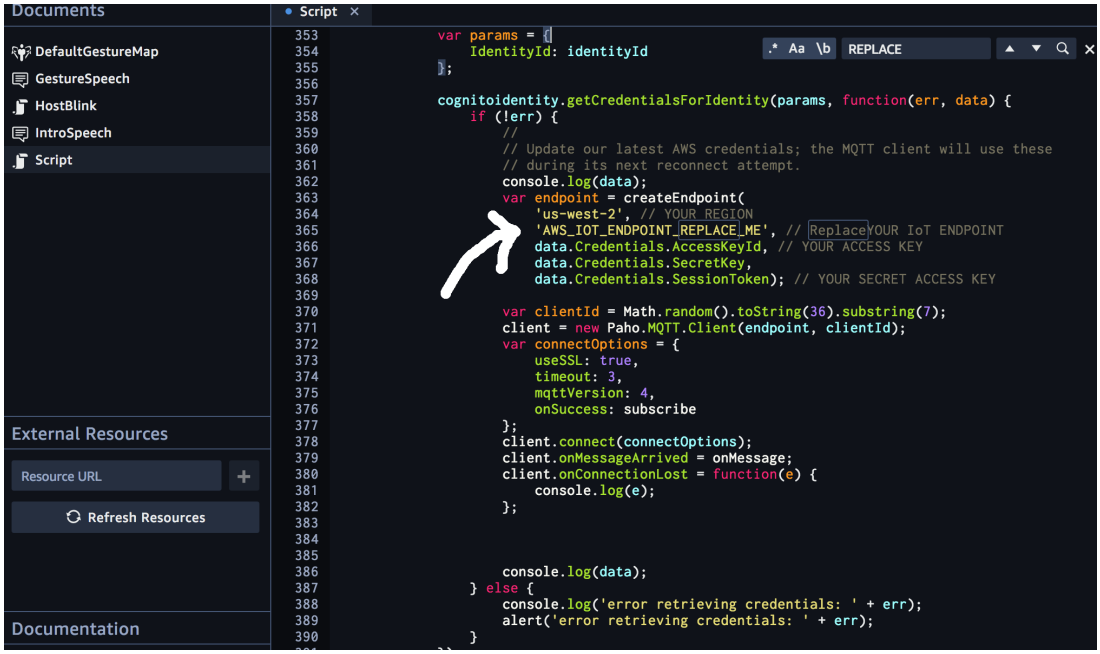


Fig: Replace the placeholder ‘AWS_IOT_ENDPOINT_REPLACE_ME’

Note: *Scripts cannot be edited only during edit mode and not during play mode*

The Script we just added has several dependencies for connecting to AWS IoT via web sockets, like the MQTT Paho Client library, HMAC, Crypto and others, let us add them in as dependencies to the Script. The following external file need to be added for everything to work, you can get all the file links from this location: https://github.com/simith/reinvent_edge_sumerian_workshop/blob/master/external_resources.txt

Note: *Copy and paste of URL’s from this document might not work well with new lines and other characters, hence please download from the above GitHub link.*

Name	URL
MQTT	https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js
Moment	https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.11.2/moment.min.js
HMAC	https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.2/components/hmac-min.js
SHA256	https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.2/components/sha256-min.js
Crypto	https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.2/components/core-min.js

Table: List of script files to be added

The below figure shows how to add External Resources/Dependencies to a Script,

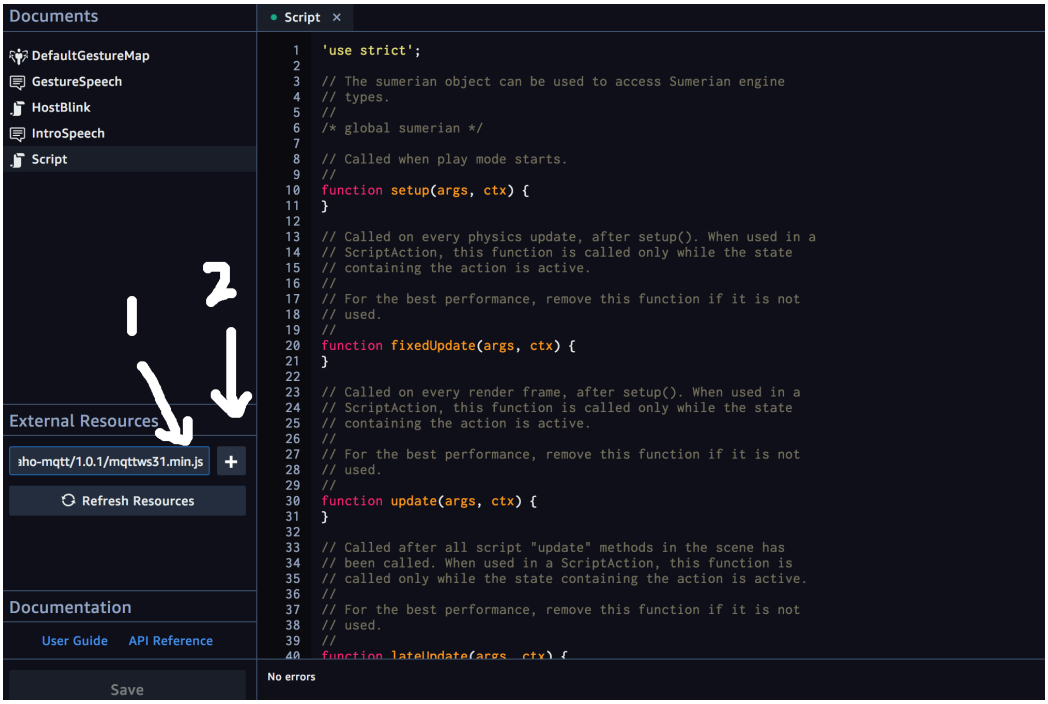


Fig: Adding external Resources/Dependencies to a Script file

Add each file in the above table, and hit the + button to add more. **Please do not forget to Hit *Save* (at the left-hand bottom of the page) when you have added all the scripts. Adding External resource is not auto-saved.**

HTML ELEMENTS FOR THE DISPLAY

We need to add a HTML placeholder for showing the content. From the Scene, press J to open the Text editor. Click on HTML 3D Entity and then enter the following HTML content.

```
<div id="canvas_div" style="width:100%;height:100%">
</div>
```

Note: The script file will load the prediction into this HTML DIV tag

GREENGRASS ML INFERENCE (ALREADY DEPLOYED ON THE DEVICE)

A Greengrass Group has already been created and deployed to a physical device for this demo and is continuously inferring. There is a ML model already deployed and a Lambda function continuously capturing frames from the Camera and inferencing/predicting and sending the inferences up to the AWS IoT Service as a JSON message. You will send some JSON messages down to your Sumerian scene to simulate this condition in the next section.

TESTING VIA AWS IOT CORE

To test the scene we would need to simulate some data coming through from the Camera. Let us simulate this using the AWS IoT service. Head to the AWS IoT Core service on us-west-2, *the same region where we are developing the Sumerian scene*. Once on the dashboard, click on **Test** on the Menu. You will have 2 options *Subscribe to a topic* and *Publish to a topic.*The message we are going to send to the Scene are raw inferences that would be coming out during inferencing out of the ML model. When the ML model is predicting that it is just a Person, the Person prediction message is sent, when it detects a person wearing a vest the Person_vest message is sent. Both the messages are sent on the same MQTT topic.

Choose *Publish to a topic*, and enter the following,

Topic	Prediction	Message

security/gate_00/prediction	Person	{
		"Object0": {
		"confidence": 1,
		"ymax": 1,
		"label": 1,
		"xmax": 0.7,
		"xmin": 0.43,
		"ymin": 0.3,
		"class": ""
		},
		"timestamp": "2018-11-26T16:06:32.538057",
		"frame_id": -1,
		"inference_fps": 33.80834432392243
		}

security/gate_00/prediction	person_vest	{
		"Object0": {
		"confidence": 1,
		"ymax": 1,
		"label": 1,
		"xmax": 0.7,
		"xmin": 0.43,
		"ymin": 0.3,
		"class": ""
		},
		"Object1": {
		"confidence": 0.75,
		"ymax": 0.96,
		"label": 2,
		"xmax": 0.67,
		"xmin": 0.44,
		"ymin": 0.5,
		"class": ""
		},
		"timestamp": "2018-11-26T16:06:32.538057",
		"frame_id": -1,
		"inference_fps": 33.80834432392243
		}

MORE SCENES

Enhancing the scene:

For more tutorials on various Amazon Sumerian concepts and scenes, please go to https://docs.sumerian.amazonaws.com/