**Storing temporary data in an Alexa skill in python**

You can store temporary data using session attributes. For more permanent data you can use dynamoDb or S3 storage.

But there is a quicker and easier way if you don’t want to do that. In fact, you get 500MB of storage.

<https://aws.amazon.com/lambda/faqs/> says:

*Q: What if I need scratch space on disk for my AWS Lambda function?*

*Each Lambda function receives 500MB of non-persistent disk space* ***in its own /tmp directory.***

So, let’s quickly see how to do that

Start a new Alexa Hosted Skill. I gave it the name **temp storage**, choose python skill, and use start from scratch. If you don’t know how to do that look at one of my other videos.

Click the code button and change the LaunchRequestHandler(AbstractRequestHandler), handler code to the following:

class LaunchRequestHandler(AbstractRequestHandler):

"""Handler for Skill Launch."""

def can\_handle(self, handler\_input):

# type: (HandlerInput) -> bool

return ask\_utils.is\_request\_type("LaunchRequest")(handler\_input)

def handle(self, handler\_input):

# type: (HandlerInput) -> Response

#"a" - Append - Opens a file for appending, creates the file if it does not exist

f = open("/tmp/tempfile.txt", "a")

f.write("This is how we create and save a file")

f.close()

#open and read the file after the appending:

f = open("/tmp/tempfile.txt", "r")

readFile = (f.read())

f.close()

speak\_output = readFile + " and this is how we read it back"

return (

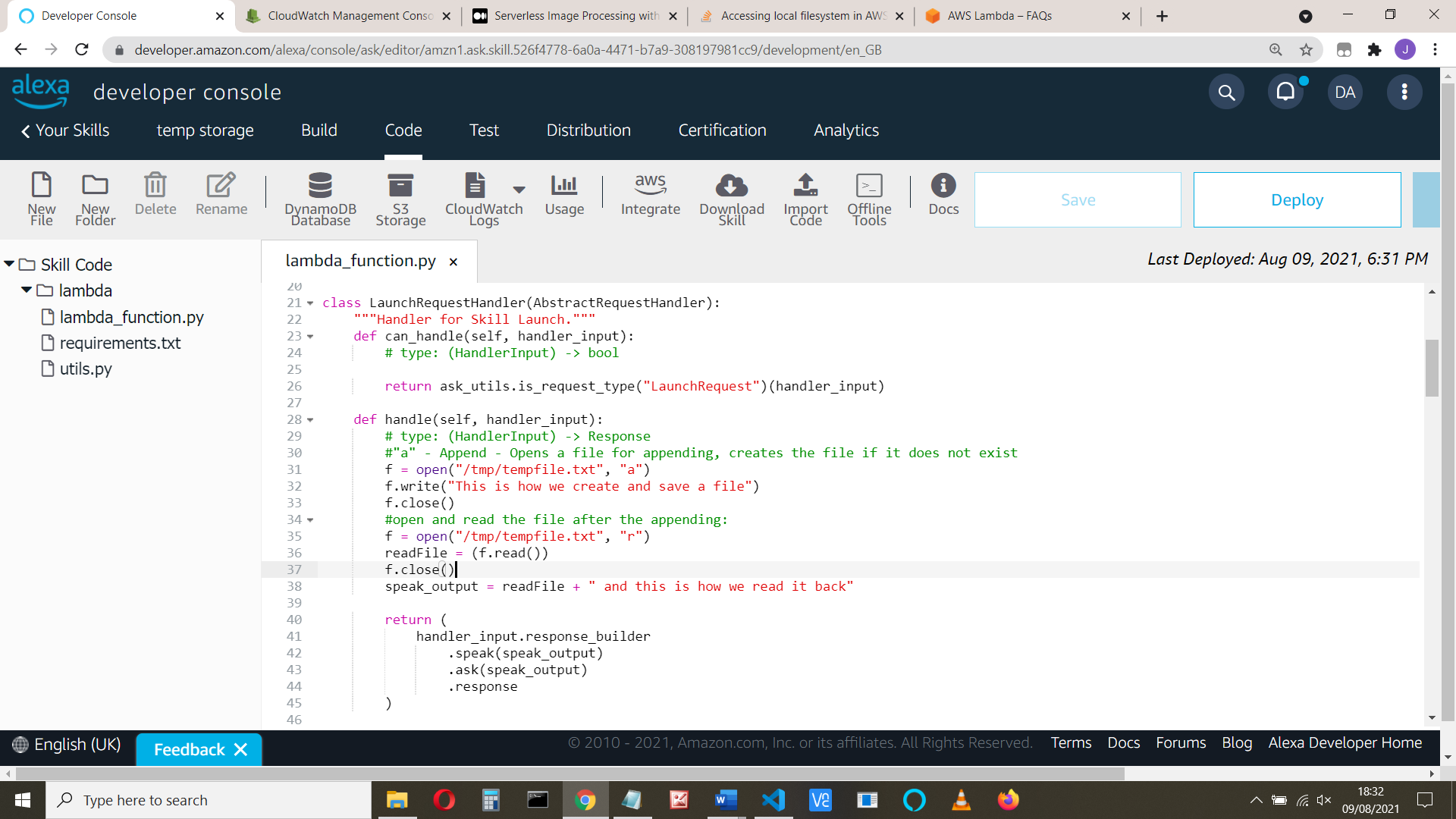
handler\_input.response\_builder

.speak(speak\_output)

.ask(speak\_output)

.response

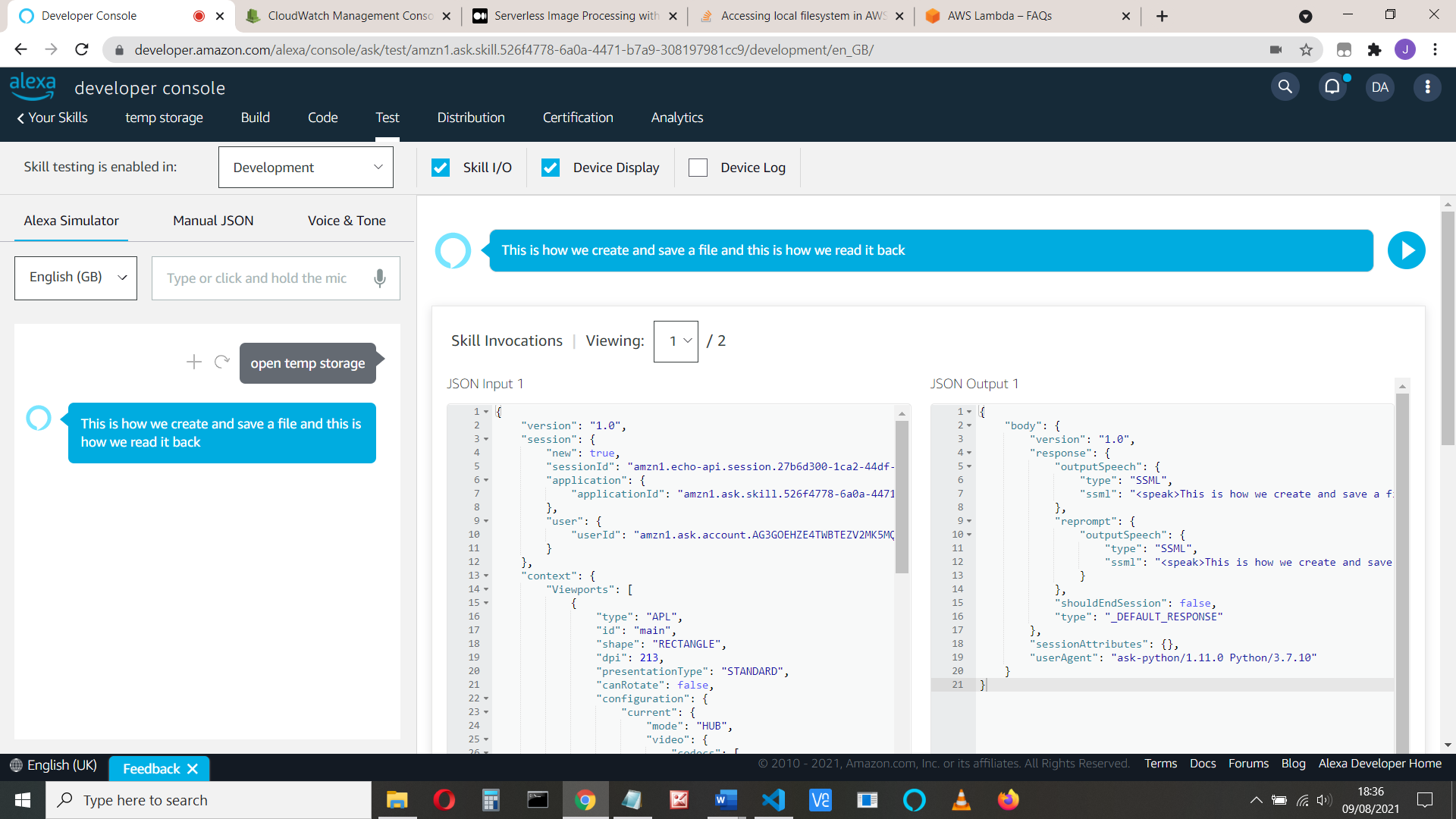
)



Save and deploy the code

Check your skill invocation name (Sometimes this isn’t what you expect).

Click the Test tab, change the skill to Development and invoke the skill:



It works.

Simples!