Paper 257-2019

Automating SAS® Viya® Admin tasks using CLI and Chatbot

Sandeep Grande, Core Compete INC

ABSTRACT

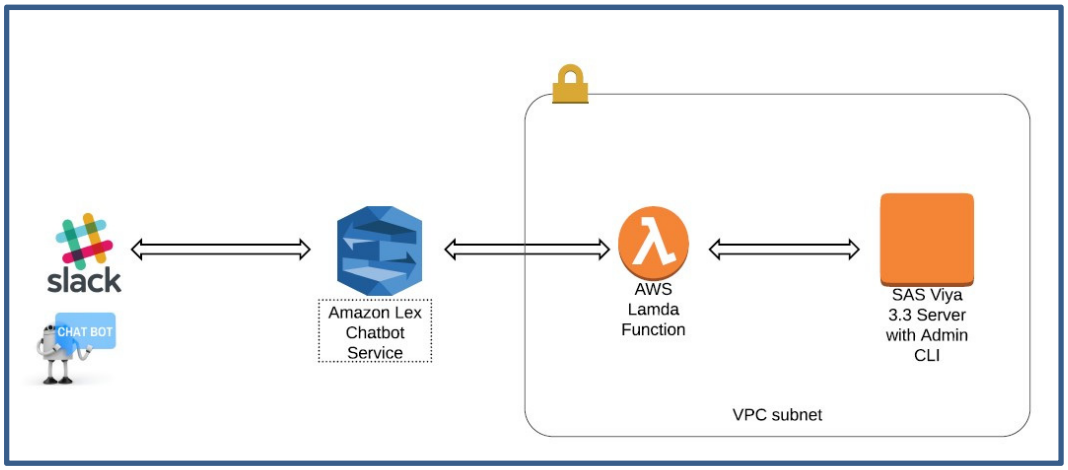
SAS® Viya® platform comes with a new command line interface to interact with microservices. This paper is an attempt to embrace the openness of Viya Platform by creating a Chatbot which helps SAS Administrator in performing his/her day to day tasks. While there are many ways to automate the Admin tasks, this paper explores the latest cloud services such as AWS Lex chatbot service, AWS Lambda which is a serverless computing platform to create user Interactive chat bot with Slack application chatbot being the front end. This chatbot can be easily customized to work at our voice commands. The lambda function uses python runtime environment and we also explore the way we can interact with microservices using python.

INTRODUCTION

SAS Viya is made up of microservices and CAS Server Engine. There is a microservice for every feature, identities manage the connection to LDAP, folder micro service to manage folders and so on. SAS has given sas-admin cli to interact with micro services, as the microservices has REST interface to Interact with , we can use the python modules to interact with the required microservices to get the tasks done This paper is gives an overview of components mentioned in the below Architecture diagram, which discusses mainly about

- Building AWS Lambda python code to manage SAS Viya Platform
- Creating a AWS Lex Chatbot service to use the previously created Lambda code
- Creating a Slack Chatbot App and tie with the AWS Lex chatbot service

OVERVIEW



AWS LAMBDA FUNCTION

AWS Lambda runs the python code which does the SAS admin tasks on a high-availability compute infrastructure. AWS performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code monitoring and logging. All you need to do is supply your code in one of the languages that AWS Lambda supports (currently Node.js, Java, C#, Go and Python).

Here, we have used python 2.7,
They are two ways to interact with micro services through python

Method 1: In this type , we do ssh to one of the SAS Viya server where CommandLine tools were installed and use the sas-admin cli to perform the task. Authentication is done using the “sas-admin auth login “ command , Here we have used Paramiko python module to make an SSH connection to SAS Viya Servers.
Method 1 Code Snippet :

```
import boto3
import paramiko

def build_response(message):
        return {
            "dialogAction": {
                "type": "Close",
                "fulfillmentState": "Fulfilled",
                "message": {
                    "contentType": "PlainText",
                    "content": message
                }
            }
        }

def worker_handler(event, context):
    if 'Fetchhostname' == event['currentIntent']['name']:
        c = paramiko.SSHClient()
        c.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        event = {"IP":"54.[redacted]"}
        host=event['IP']
        print "Connecting to " + host
        c.connect( hostname = host, username = "san[redacted].com", password = "[redacted]" )
        print "Connected to " + host

        commands = ["hostname -f"]
        for command in commands:
            print "Executing {}".format(command)
            stdin , stdout, stderr = c.exec_command(command)
            hostname=str(stdout.read())
            print stderr.read()
        return build_response(hostname)


    elif 'Fetchusername' == event['currentIntent']['name']:
        return build_response("Fetchusernamecalled")
```

Method 2 : In this method, we directly interact with RESTFUL endpoints of SAS Viya using python module urllib2 or requests and parse the JSON output as per the requirement. This method is more flexible as we can achieve variety of tasks which are not provided by CLI. Prior to communicate to the REST API , we need to authenticate to SAS Logon manager to obtain the OAuth Token.

Usage Note 60180: Using Python Administration Tools in SAS® Viya™

http://support.sas.com/kb/60/180.html

AWS LEX CHATBOT

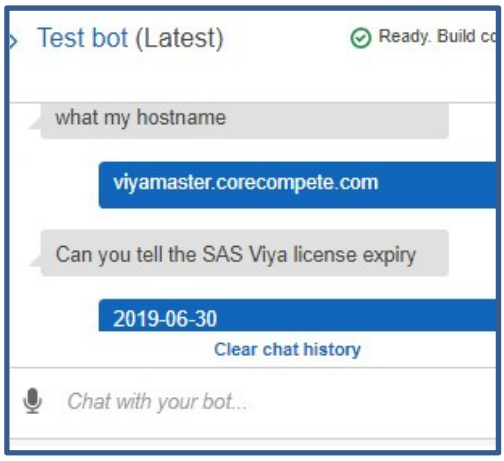
Amazon Lex is a service for building conversational interfaces into any application using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable you to build applications with highly engaging user experiences and lifelike conversational interactions

Intents

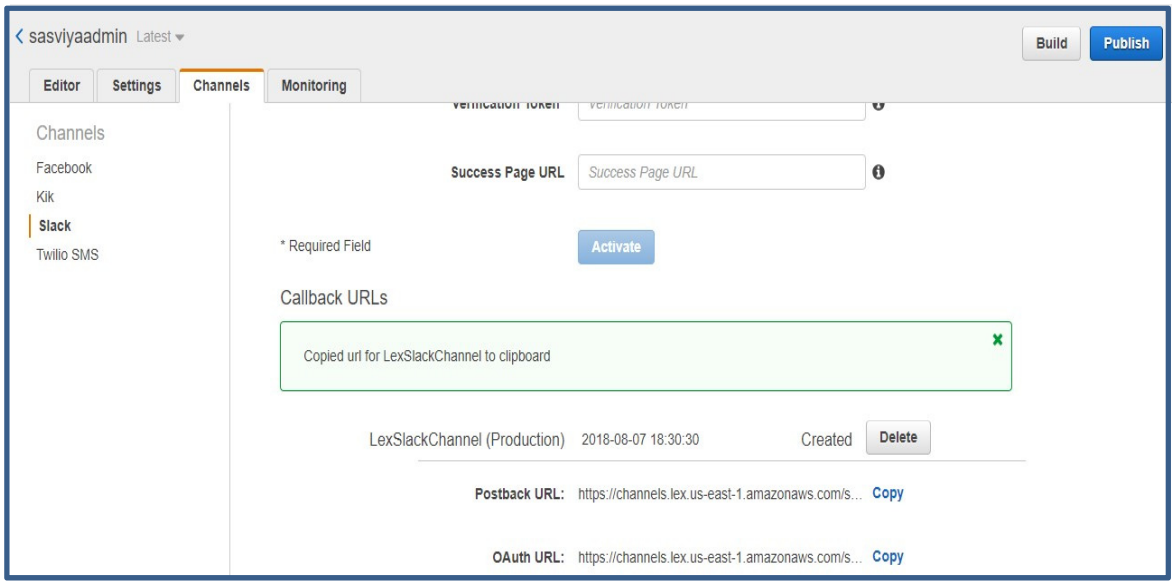
A chatbot is a set of responses that it gives to a certain message. These are stored in Intents which are like talking points.

Sample utterances and Response

Amazon Lex uses Natural Language Understanding (NLU) to work out what the user is trying to say. If they say “What’s your name” instead of “What is your name”, Lex will still match the phrases. Pretty smart!



We use AWS Lex chatbot testbot to check the working condition of your interaction between Lex and Lambda. Once everything is working fine, we publish the Chatbot bot to make it available to slack.
We user Lex Channel service to connect to slack. Once you create Slack bot App, you will get Client ID, Client Secret and Verification token. We supply the credentials of Slack bot to activate the Lex channel.



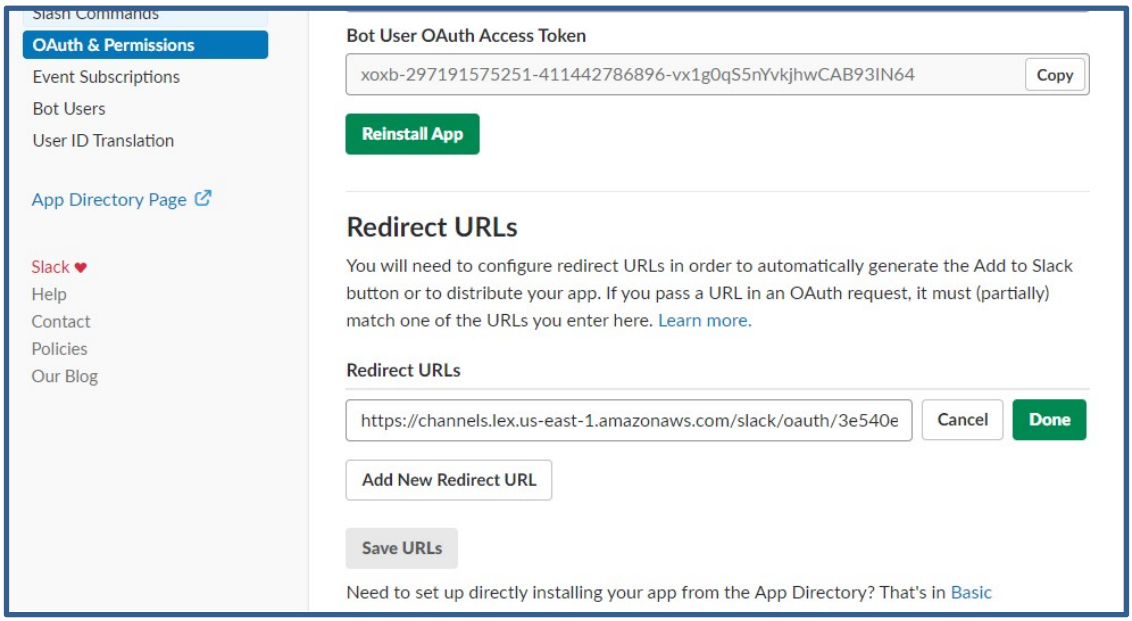
Once the channel is activated, we get two urls PostbackURL, OAuth URL which we use in next section.

SLACK - CHATBOT USER

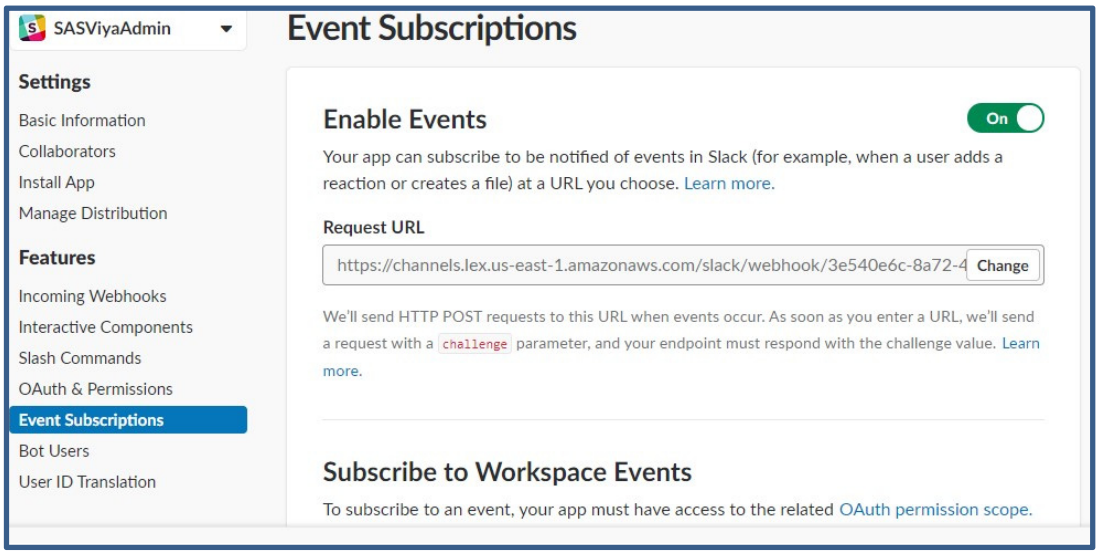
Slack application provides us with a console to create chatbot application at https://api.slack.com/
Here we can create a chatbot user, in our case its SASViyaAdmin The bot user from Slack acts as our front end for our Lex Chatbot Service.

Grouping Together:

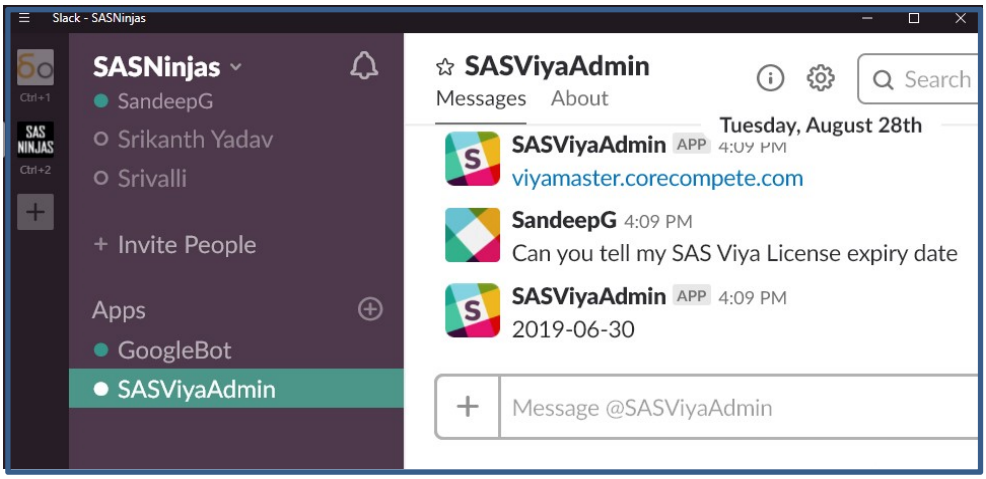
The OAuth Url to be copied as Redirect URL in our OAuth section of SlackApp.



The postback URL from previous section is copied to EventSubcription Request URL



The below is final output using the desktop slack application. The same application id available on mobile, where you can utilize the voice to text service of you mobile to give voice commands to slack bot



CONCLUSION

Whenever SAS Admin performs a task repeatedly, he can have the same task in the form of python code and call it using chatbot or over voice command. With SAS Viya , this kind of chatbot assistant can help SAS Admins in managing the Viya platform by saving their valuable time.

REFERENCES

SAS Institute Inc. 2017. SAS® Viya® 3.3 Administration. Cary, NC: SAS Institute Inc.Available at https://support.sas.com/documentation/onlinedoc/viya/3.3/ViyaAdmin33.pdf
https://blogs.sas.com/content/sgf/2018/03/08/sas-viya-3-3-command-line-interfaces-foradministration/
Amazon Web Services : Lambda . Available at https://docs.aws.amazon.com/lambda/latest/dg/welcome.html Amazon Lex – Build Conversation Bots - Amazon AWS. Available at https://docs.aws.amazon.com/lex/latest/dg/what-is.html
Tutorials | Slack – Chatbot Frontend : https://api.slack.com/tutorials

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

sandeep.grande@corecompete.com

Sandeep Grande
Sr. Technical Consultant
CORE COMPETE INC
5001 S Miami Blvd, Durham, NC 27703