# SAS® GLOBAL FORUM 2019

## USERS PROGRAM

APRIL 28 – MAY 1, 2019 | DALLAS, TX

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

# Automating SAS® Viya® Administrative Tasks Using Chatbot

Sandeep Grande, Senior SAS Administrator

CORE COMPETE INC

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

Sandeep Grande

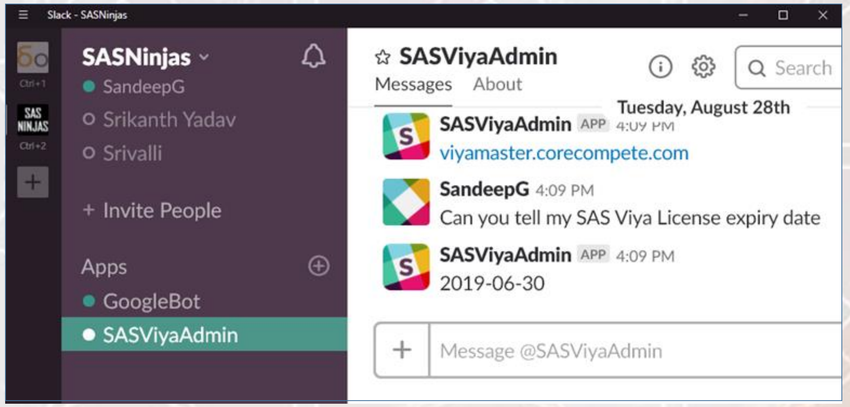
## Index



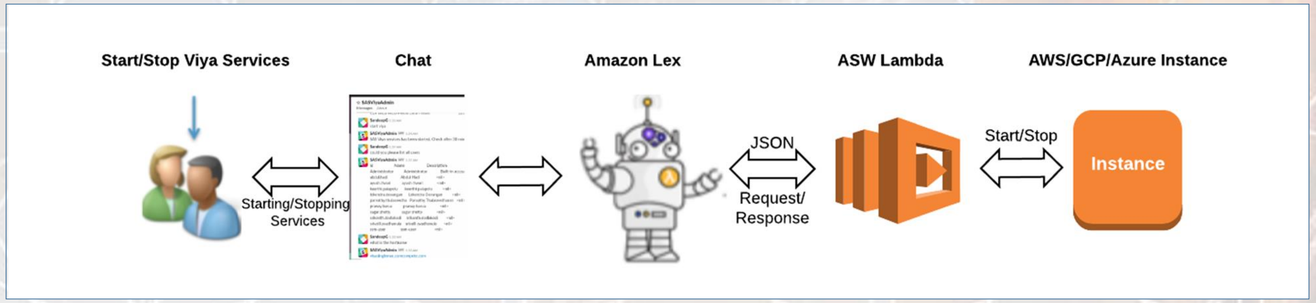
Please use the headings above to navigate through the different sections of the poster

- SAS® Viya® comes with a new command-line interface to interact with microservices. This poster attempts to embrace the openness of SAS Viya by creating a Chatbot that helps the SAS administrator to perform day-to-day tasks. While there are many ways to automate administrative tasks, this poster explores the latest cloud services such as Amazon Web Services (AWS) Lex chatbot service and AWS Lambda, which is a serverless computing platform to create a user-interactive chatbot with Slack App chatbot being the front end. This chatbot can be easily customized to work to our voice commands. The Lambda function uses the Python runtime environment, and we also explore the way we can interact with microservices using Python.

Final Output:




Backend Flow:




CORECOMPETE

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

# Automating SAS® Viya® Administrative Tasks Using Chatbot

Sandeep Grande, Senior SAS Administrator
CORE COMPETE INC

## Index

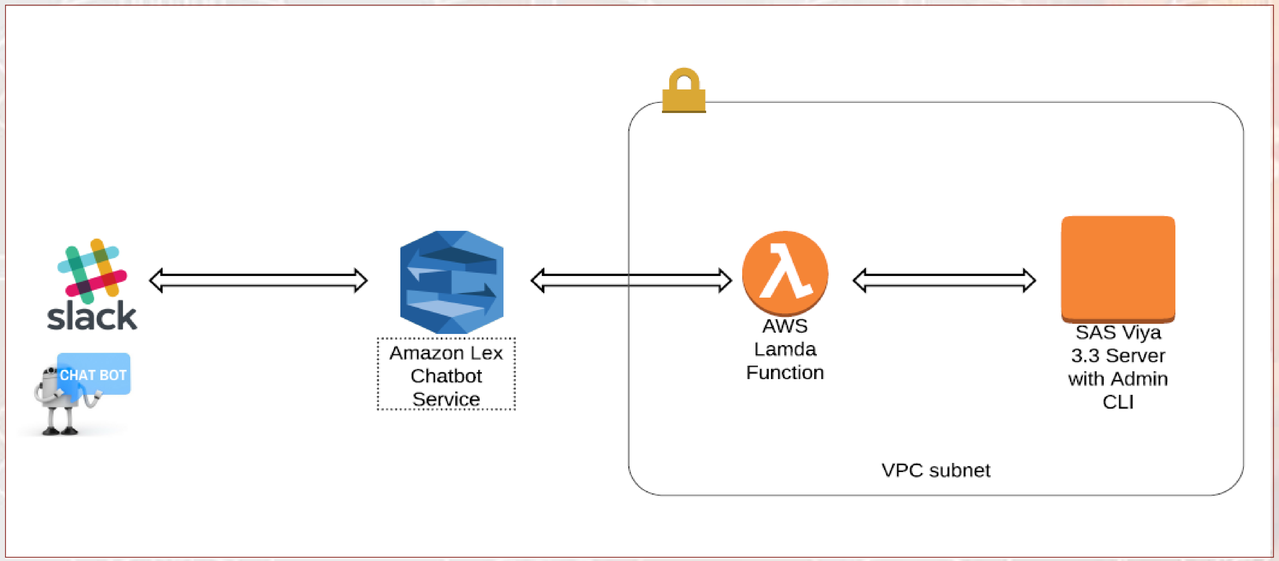


Please use the headings above to navigate through the different sections of the poster

## Intro

- 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 eposter gives an overview of components mentioned in the below Architecture diagram, which discusses mainly about
  - Building AWS Lamda python code to manage SAS Viya Platform
  - Creating a AWS Lex Chatbot service to operate previously created Lamda code
  - Creating a Slack Chatbot App and integrate with the AWS Lex chatbot service

## Objective




CORECOMPETE

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

# Automating SAS® Viya® Administrative Tasks Using Chatbot

Sandeep Grande, Senior SAS Administrator
CORE COMPETE INC

## Index



Please use the headings above to navigate through the different sections of the poster

## Creating AWS LAMDA 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 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

### 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")
```

CORECOMPETE

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

# Automating SAS® Viya® Administrative Tasks Using Chatbot

Sandeep Grande, Senior SAS Administrator
CORE COMPETE INC

## Index



Please use the headings above to navigate through the different sections of the poster


## AWS Lex Chatbot Service

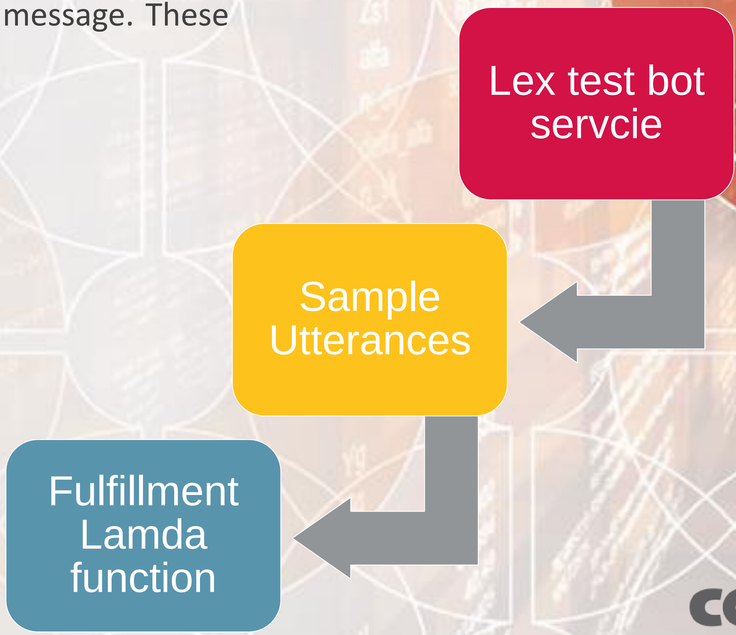
- Amazon Lex is a service for building conversational interfaces into any application using voice and text.
- Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent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

## Intent

- 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

The smart bit about Amazon Lex is that it 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!




CORECOMPETE

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

# Automating SAS® Viya® Administrative Tasks Using Chatbot

Sandeep Grande, Senior SAS Administrator
CORE COMPETE INC

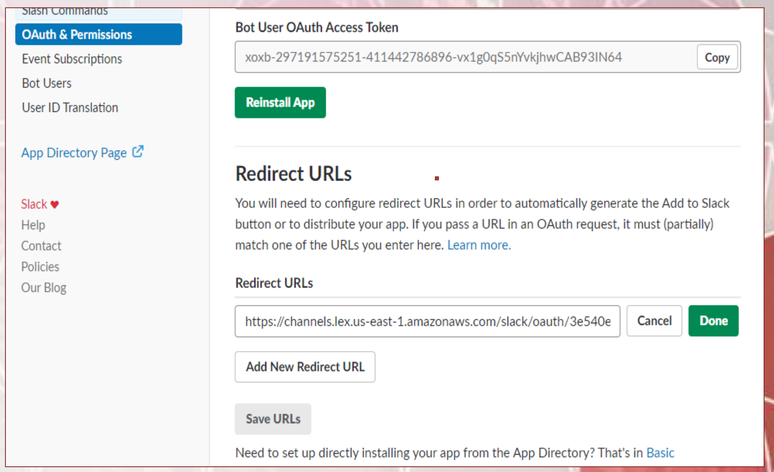
## Index

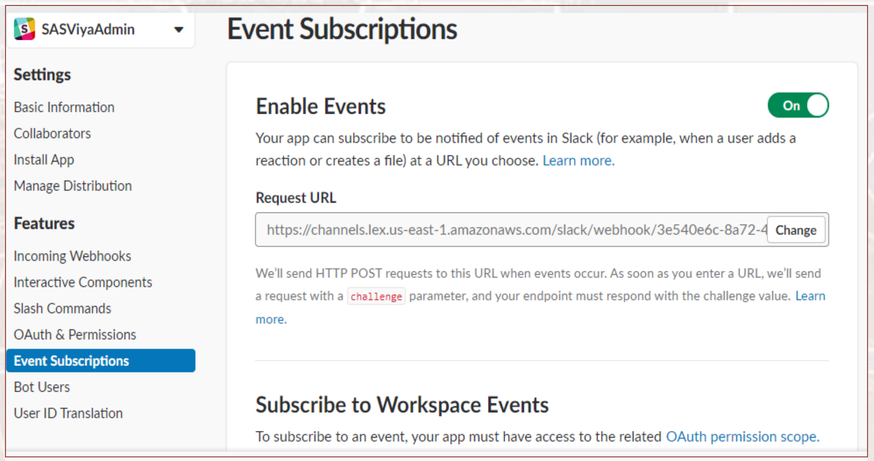


Please use the headings above to navigate through the different sections of the poster

## Creating 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.




## Integrating with Lex

- 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

CORECOMPETE

CORECOMPETE
DELIVERING VALUE FROM BIG DATA

# Automating SAS® Viya® Administrative Tasks Using Chatbot

Sandeep Grande, Senior SAS Administrator
CORE COMPETE INC

## Index



Please use the headings above to navigate through the different sections of the poster

## Thank you …

When ever 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.

Your comments and questions are valued and encouraged. Contact the author at:
Sandeep.Grande@corecompete.com
Technical Consultant
CORE COMPETE INC
Durham, NC

## 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-for-administration/
Amazon Web Services : Lamda . 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

CORECOMPETE