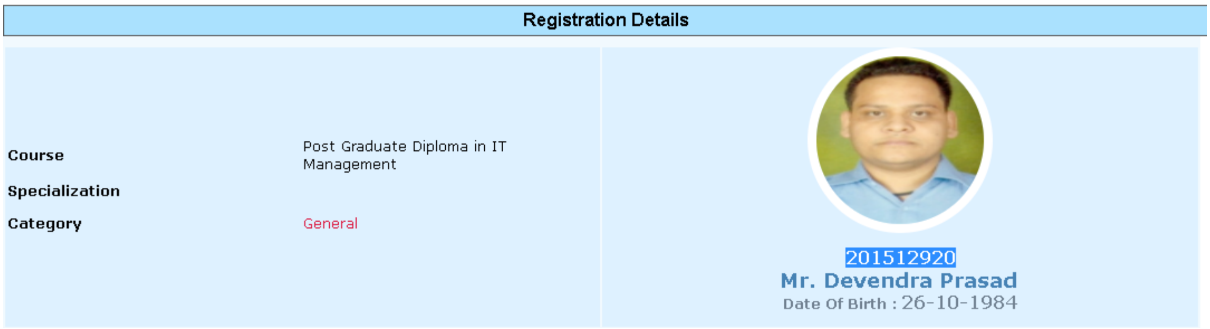
Text to Speech

Machine Learning Project

Using Clound Computing – Amazon Web Services

The Game Changer



Registration Details

Course	Post Graduate Diploma in IT Management
Specialization	
Category	General

201512920

Mr. Devendra Prasad

Date Of Birth : 26-10-1984

Table of contents

Contents

Introduction

Objective

My project is to use the aws toolset to build a simple web service which would convert text to speech on the fly and it would help the people saving text notes as audio and can listern anytime anywhere. This can help a great to students study while listening to audio notes

I should be able to make a portal UI where all such services can be linked together and get a Single Page Application view to the end users

Scope of my project

- **Aws services**
  - Boto client api
  - S3 – Simple Storage Service
  - Polly Client
  - Dynamo DB – the no-sql database
  - Lambda-the serverless calls
  - Python 2.7/3.6
  - Html5
  - Javascript
- **Portal UI**
  - Html5+jquery+angular js
  - Rest service calls
  - Parsing JSON docs and making HTML view ocnfigurable
  - xampp server

Installtion instruction

1. install xampp in c:\
2. install python in c:\
3. place simple-ui folder **in c:\xampp\htdocs** folder
4. open chrome and type http://localhost/simple-ui. also available at http://dpresume.com/aws
5. the other service (**aws-code** folder) is also attached which has been loaded in AWS free tier account and hence no installtion needed

note: in ideal scenarios, angular or jsquery api must not be shipped/deployed with the code rather their web api references must be used, but I am not following that as I downloaded the API and using reference from my nested folder

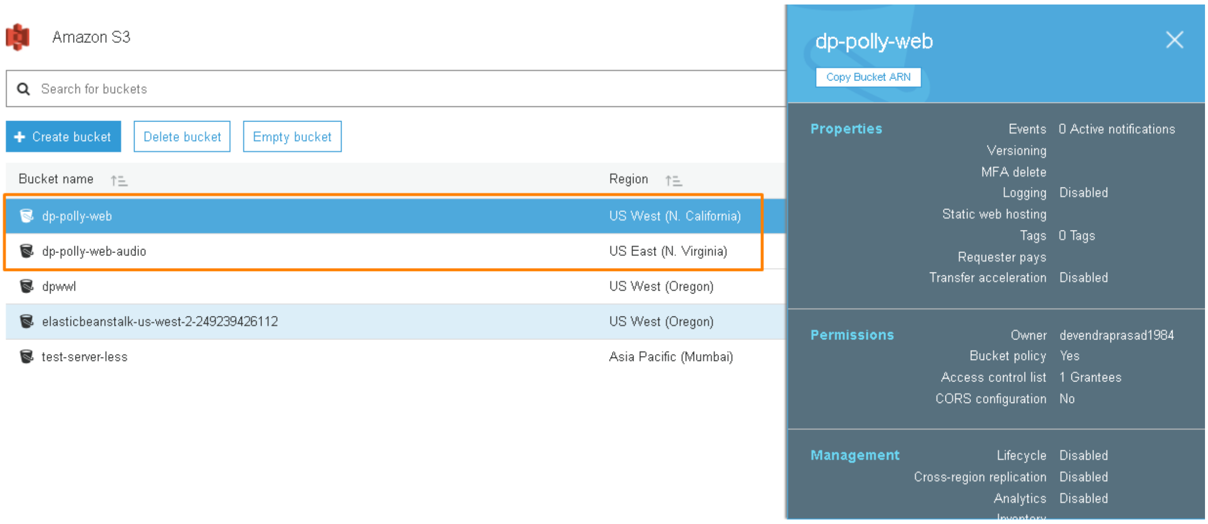
to keep portal UI design a bit of flexible, I am using my own templating model by dividing UI into left,right,top, bottom containers and supplying contents on run time. It gives end-user a consitent feeling and easiness in coding at my end as well

Analysis, Design, Guide and some Diagrams

AWS Flow

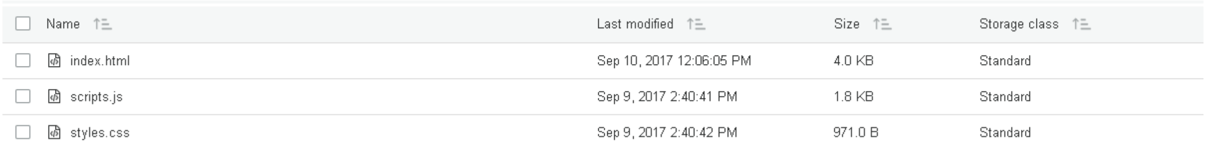
S3 – Simple Storage Service

I have used S3 to storemy web pages and to store polly converted audio files



Bucket name	Region
dp-polly-web	US West (N. California)
dp-polly-web-audio	US East (N. Virginia)
dpwwl	US West (Oregon)
elasticbeanstalk-us-west-2-249239426112	US West (Oregon)
test-server-less	Asia Pacific (Mumbai)

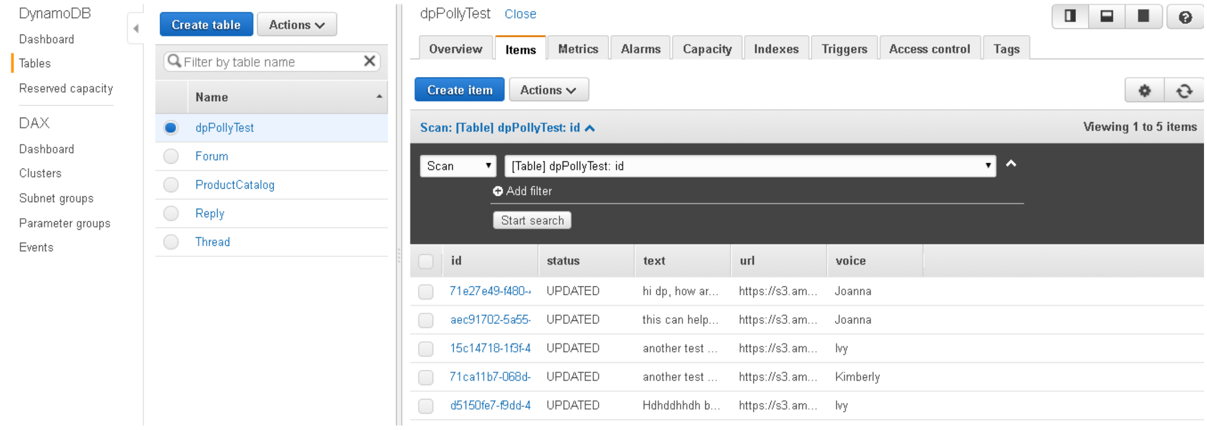
Properties	Events	0 Active notifications
	Versioning	
	MFA delete	
	Logging	Disabled
	Static web hosting	
	Tags	0 Tags
	Requester pays	
	Transfer acceleration	Disabled
Permissions	Owner	devendraprasad1984
	Bucket policy	Yes
	Access control list	1 Grantees
	CORS configuration	No
Management	Lifecycle	Disabled
	Cross-region replication	Disabled
	Analytics	Disabled



□ Name	Last modified	Size	Storage class
□ index.html	Sep 10, 2017 12:06:05 PM	4.0 KB	Standard
□ scripts.js	Sep 9, 2017 2:40:41 PM	1.8 KB	Standard
□ styles.css	Sep 9, 2017 2:40:42 PM	971.0 B	Standard

Dynamo DB view –no sql data stoage

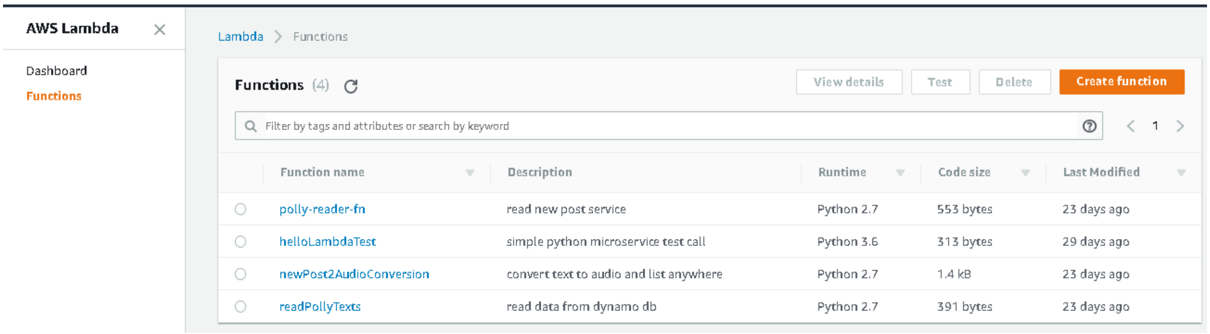
Dynamo db is used to store the notes being submitted from web UI. It stores the text being submitted and runtime generated S3 url which has audio files stored



□	id	status	text	url	voice
□	71e27e49-f480--	UPDATED	hi dp, how ar...	https://s3.am...	Joanna
□	aec91702-5a55-	UPDATED	this can help...	https://s3.am...	Joanna
□	15c14718-1f3f-4	UPDATED	another test ...	https://s3.am...	Ivy
□	71ca11b7-068d-	UPDATED	another test ...	https://s3.am...	Kimberly
□	d5150fe7-f9dd-4	UPDATED	Hdhddhhdh b...	https://s3.am...	Ivy

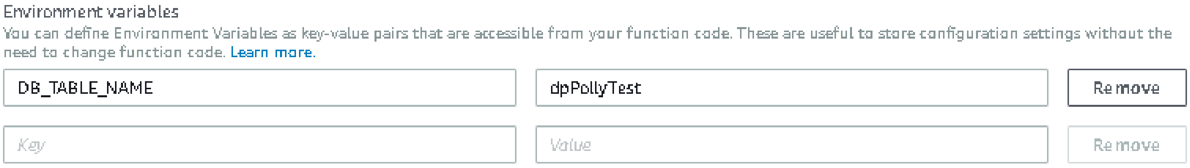
AWS Lambda

Python written api code which acts as a lamba function which gets triggered as soon as new posts come from web page which actually does the conversion of text to speeh engine



Functions (4)

Function name	Description	Runtime	Code size	Last Modified
polly-reader-fn	read new post service	Python 2.7	553 bytes	23 days ago
helloLambdaTest	simple python microservice test call	Python 3.6	313 bytes	29 days ago
newPost2AudioConversion	convert text to audio and list anywhere	Python 2.7	1.4 kB	23 days ago
readPollyTexts	read data from dynamo db	Python 2.7	391 bytes	23 days ago



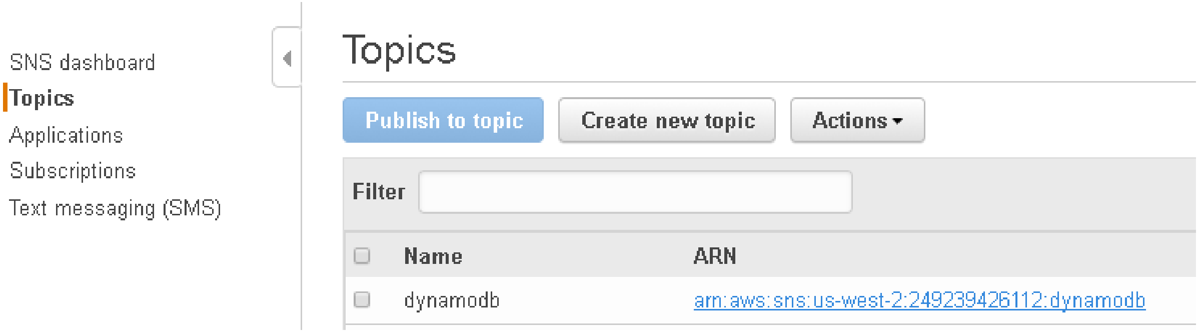
Environment variables

You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more.

Key	Value
DB_TABLE_NAME	dpPollyTest
Key	Value

SNS topic-a trigger to new posts

Lambda call and SNS(simple notification service) collaboratively act as a trigger to new posts

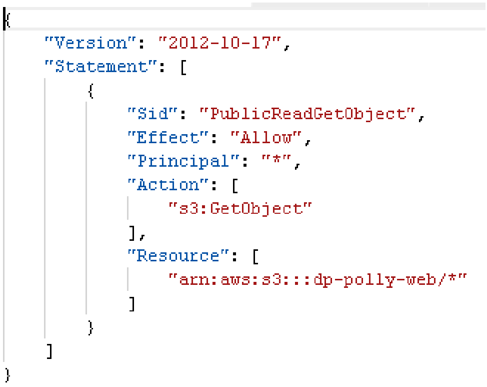


Topics

□	Name	ARN
□	dynamodb	arn:aws:sns:us-west-2:249239426112:dynamodb

S3 Bucket Policy

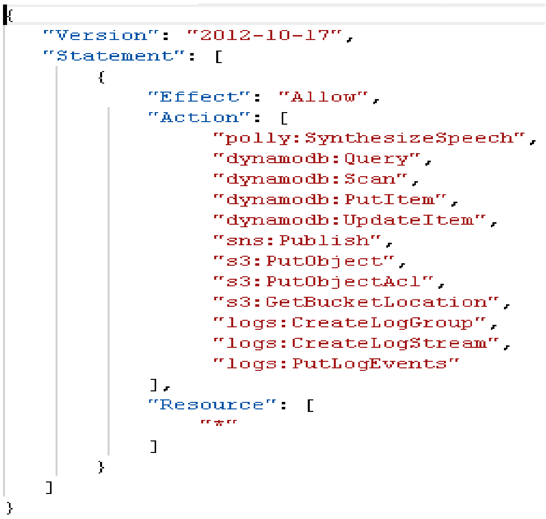
Aws policy document which acts a permission rule without which access and services are not possible over the web



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::dp-polly-web/*"
            ]
        }
    ]
}
```

Lambda policy

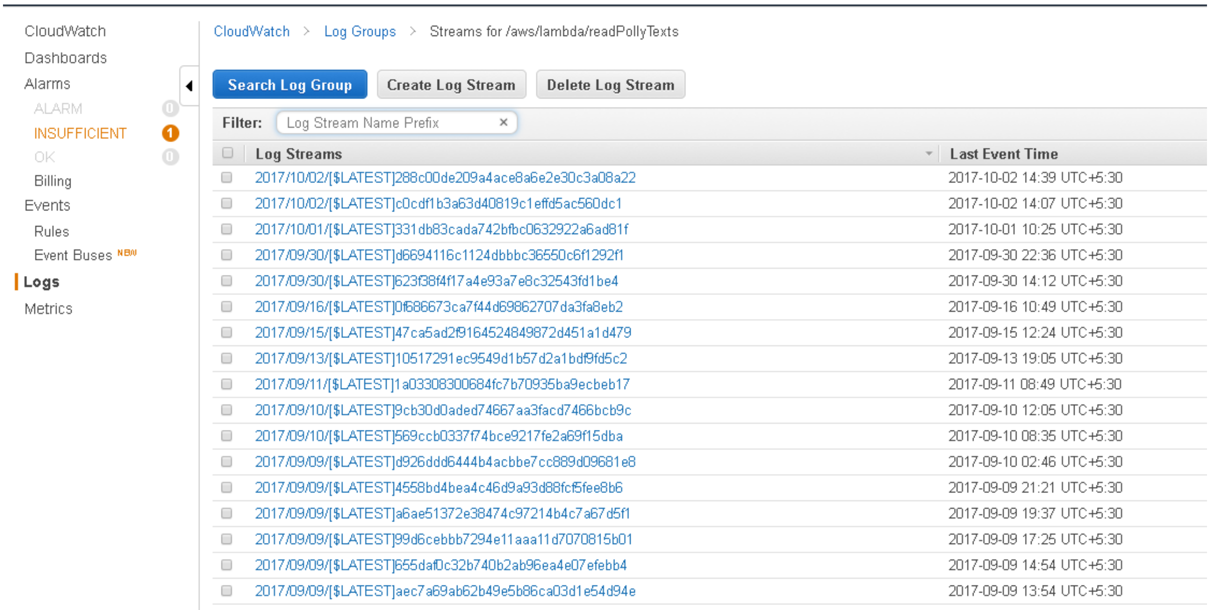
Lambda policy service document which gets lambda calls in place i.e.its a permission to the API



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "polly:SynthesizeSpeech",
                "dynamodb:Query",
                "dynamodb:Scan",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem",
                "sns:Publish",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:GetBucketLocation",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Cloud Watch-monitoriing service

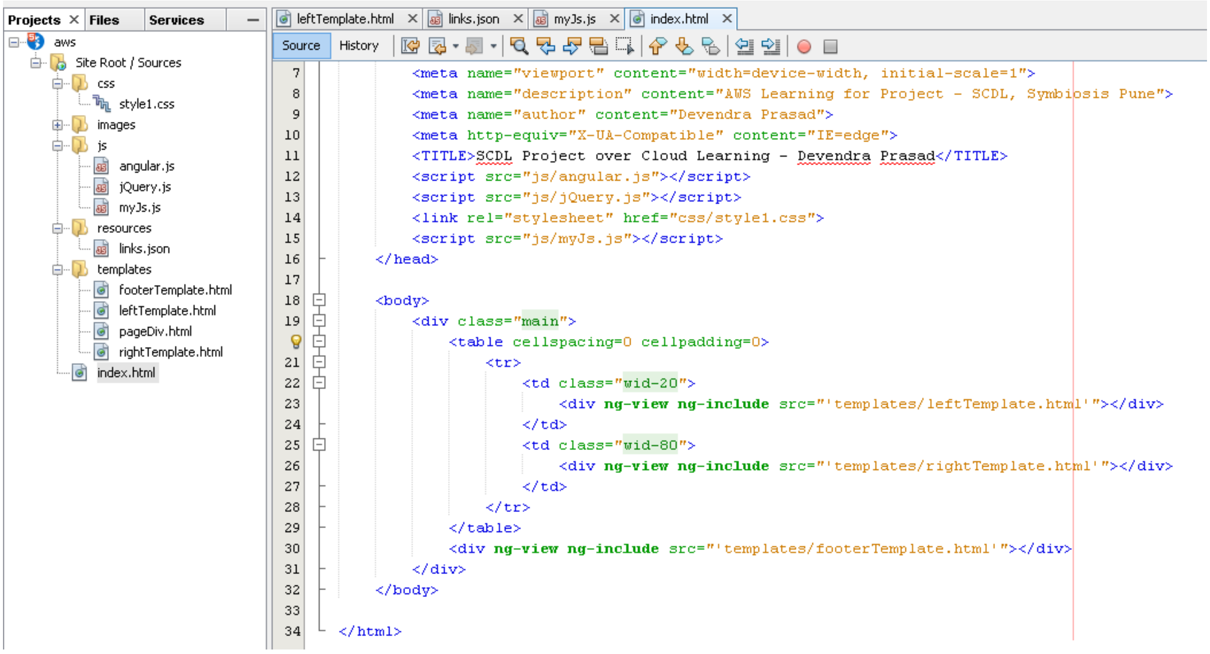
The monitoring service to the calls being made fromwww and helps admin to monitor the traffic



Log Streams	Last Event Time
2017/10/02/[$LATEST]288c00de209a4ace8a6e2e30c3a08a22	2017-10-02 14:39 UTC+5:30
2017/10/02/[$LATEST]c0cdf1b3a63d40819c1effd5ac560dc1	2017-10-02 14:07 UTC+5:30
2017/10/01/[$LATEST]331db83cada742bfbc0632922a6ad81f	2017-10-01 10:25 UTC+5:30
2017/09/30/[$LATEST]d6694116c1124dbbbc36550c6f1292f1	2017-09-30 22:36 UTC+5:30
2017/09/30/[$LATEST]623f38f4f17a4e93a7e8c32543fd1be4	2017-09-30 14:12 UTC+5:30
2017/09/16/[$LATEST]0f686673ca7f44d69862707da3fa8eb2	2017-09-16 10:49 UTC+5:30
2017/09/15/[$LATEST]47ca5ad2f9164524849872d451a1d479	2017-09-15 12:24 UTC+5:30
2017/09/13/[$LATEST]10517291ec9549d1b57d2a1bdf9fd5c2	2017-09-13 19:05 UTC+5:30
2017/09/11/[$LATEST]1a03308300684fc7b70935ba9ecbeb17	2017-09-11 08:49 UTC+5:30
2017/09/10/[$LATEST]9cb30d0aded74667aa3facd7466bcb9c	2017-09-10 12:05 UTC+5:30
2017/09/10/[$LATEST]569ccb0337f74bce9217fe2a69f15dba	2017-09-10 08:35 UTC+5:30
2017/09/09/[$LATEST]d926ddd6444b4acbbe7cc889d09681e8	2017-09-10 02:46 UTC+5:30
2017/09/09/[$LATEST]4558bd4bea4c46d9a93d88fcf5fee8b6	2017-09-09 21:21 UTC+5:30
2017/09/09/[$LATEST]a6ae51372e38474c97214b4c7a67d5f1	2017-09-09 19:37 UTC+5:30
2017/09/09/[$LATEST]99d6cebbb7294e11aaa11d7070815b01	2017-09-09 17:25 UTC+5:30
2017/09/09/[$LATEST]655daf0c32b740b2ab96ea4e07efebb4	2017-09-09 14:54 UTC+5:30
2017/09/09/[$LATEST]aec7a69ab62b49e5b86ca03d1e54d94e	2017-09-09 13:54 UTC+5:30

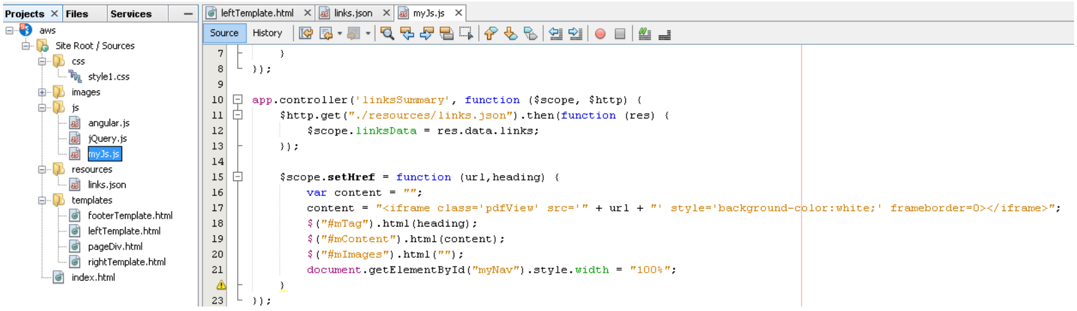
Simple Service Portal UI Flow

Index.html– act a initial point to the UI, where I make the flow to left,right,footer containers and defining which part of my application will fit into that container



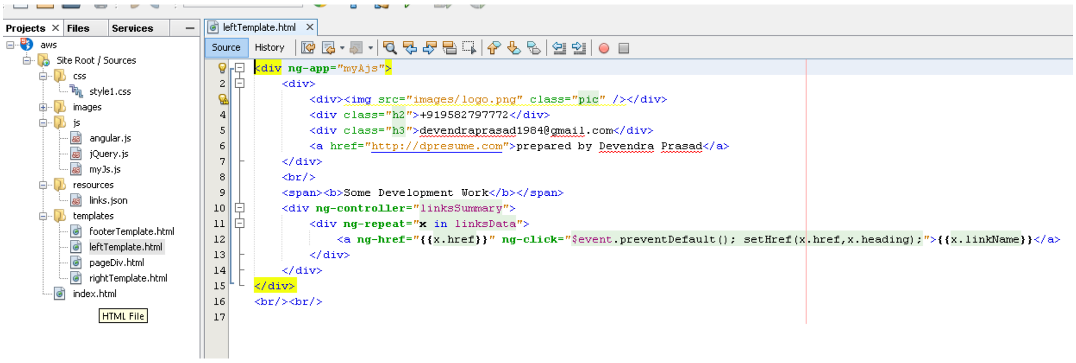
```
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="description" content="AWS Learning for Project - SCDL, Symbiosis Pune">
        <meta name="author" content="Devendra Prasad">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <TITLE>SCDL Project over Cloud Learning - Devendra Prasad</TITLE>
        <script src="js/angular.js"></script>
        <script src="js/jQuery.js"></script>
        <link rel="stylesheet" href="css/style1.css">
        <script src="js/myJs.js"></script>
    </head>

    <body>
        <div class="main">
            <table cellspacing=0 cellpadding=0>
                <tr>
                    <td class="wid-20">
                        <div ng-view ng-include src="'templates/leftTemplate.html'"></div>
                    </td>
                    <td class="wid-80">
                        <div ng-view ng-include src="'templates/rightTemplate.html'"></div>
                    </td>
                </tr>
            </table>
            <div ng-view ng-include src="'templates/footerTemplate.html'"></div>
        </div>
    </body>

</html>
```

myJs.js: angular js controller which will be bound to the controls over the UI. This actually reads the links.json document whihc is a mimic of REST service and generate left side of anchors tags at run time. The doc supplies all necessary information about the anchor tag and subject/header which helps complete the UI



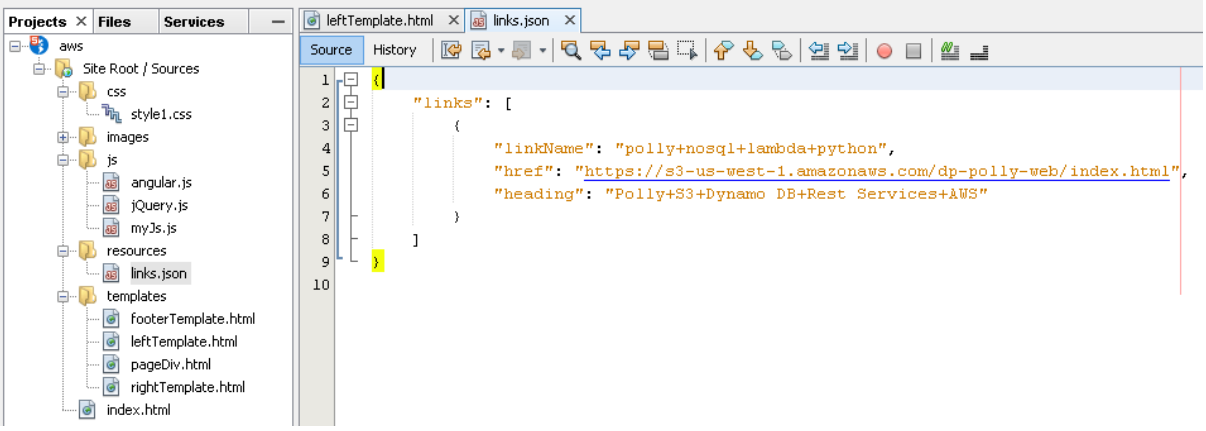
```
    }
});

app.controller('linksSummary', function ($scope, $http) {
    $http.get("./resources/links.json").then(function (res) {
        $scope.linksData = res.data.links;
    });

    $scope.setHref = function (url,heading) {
        var content = "";
        content = "<iframe class='pdfView' src='" + url + "' style='background-color:white;' frameborder=0></iframe>";
        $("#mTag").html(heading);
        $("#mContent").html(content);
        $("#mImages").html("");
        document.getElementById("myNav").style.width = "100%";
    }
});
```

leftTemplate.html: the template having ng-repeat element to render the json/rest doc and help control the click event and display the service in right pane



```
<div ng-app="myAjs">
    <div>
        <div><img src="images/logo.png" class="pic" /></div>
        <div class="h2">+919582797772</div>
        <div class="h3">devendraprasad1984@gmail.com</div>
        <a href="http://dpresume.com">prepared by Devendra Prasad</a>
    </div>
    <br/>
    <span><b>Some Development Work</b></span>
    <div ng-controller="linksSummary">
        <div ng-repeat="x in linksData">
            <a ng-href="{{x.href}}" ng-click="$event.preventDefault(); setHref(x.href,x.heading);">{{x.linkName}}</a>
        </div>
    </div>
</div>
<br/><br/>
```

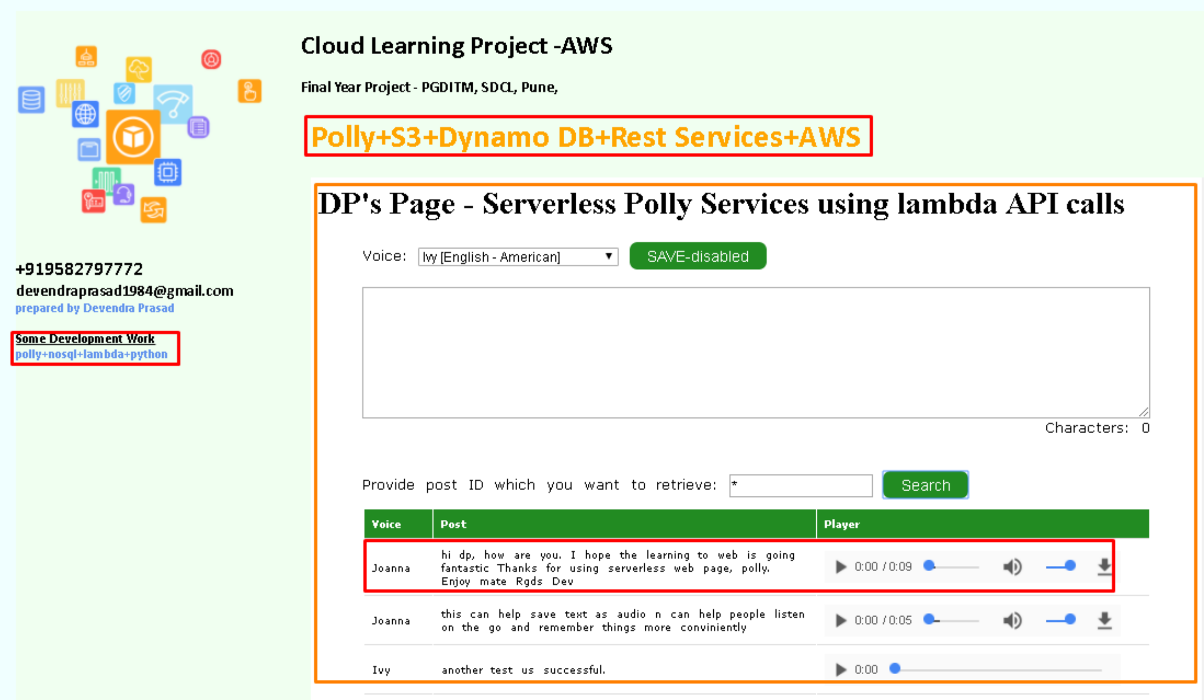
Links.json:a mimic to rest service returning the information that needs to be rendered on left side of the application



```
{
    "links": [
        {
            "linkName": "polly+nosql+lambda+python",
            "href": "https://s3-us-west-1.amazonaws.com/dp-polly-web/index.html",
            "heading": "Polly+S3+Dynamo DB+Rest Services+AWS"
        }
    ]
}
```

Final UI after Compilation

Save is disabled in the deployement just to avoid any hits over the service which could cause me billing which I dont want right now to happen but if project would demand to actually have it demonstrated, kindly let me know via email and I will enable it for 48 hours for the viewers to run through it



Cloud Learning Project -AWS

Final Year Project - PGDITM, SDCL, Pune,

Polly+S3+Dynamo DB+Rest Services+AWS

+919582797772
devendraprasad1984@gmail.com
prepared by Devendra Prasad

Some Development Work
polly+nosql+lambda+python

DP's Page - Serverless Polly Services using lambda API calls

Voice: Ivy [English - American] SAVE-disabled

Characters: 0

Provide post ID which you want to retrieve: * Search

Voice	Post	Player
Joanna	hi dp, how are you. I hope the learning to web is going fantastic Thanks for using serverless web page, polly. Enjoy mate Rgds Dev	▶ 0:00 / 0:09
Joanna	this can help save text as audio n can help people listen on the go and remember things more conviniently	▶ 0:00 / 0:05
Ivy	another test us successful.	▶ 0:00

Learning Experience

This project has helped me learn a lot by doing some real stuff which has broaden my technological perspective wrt cloud computing. Not only I did create the service but also learned on EC2 instance, NAT gateways, VPCs etc which would definately help be architecting on cloud in near fuuture.

Web used in building the service

- AWS API documentation: https://aws.amazon.com/documentation/
- Google search: www.google.com
- acloud.guru forums: https://acloud.guru/forums/home
- youtube tutorials
- boto client api: https://boto3.readthedocs.io/en/latest/
- netbeans IDE: https://netbeans.org/downloads/
- vsCode IDE: https://code.visualstudio.com/
- xampp: https://www.apachefriends.org/download.html
- free tier aws console: https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2#