Building a Serverless Web Application by Blake Gerold

(This work is strongly based of the “Build a Serverless Web Application” tutorial from the AWS website. <https://aws.amazon.com/getting-started/hands-on/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>)

Introduction: This tutorial requires a variety for AWS services that have not been used in class before. This includes Amazon API Gateway, AWS Amplify, Amazon Cognito and an ArcGIS account to add mapping to the app.

Section Zero: Adding all the new permission for IAM. However, there is not enough space in the permission policies to add them all directly! Create a new group and call “Project2”. Add the follow permissions to this group and then add it to the Cloud9Admisistrators policies:

[AWSCodeCommitFullAccess](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAWSCodeCommitFullAccess)

[AmazonAPIGatewayAdministrator](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonAPIGatewayAdministrator)

[AmazonAPIGatewayInvokeFullAccess](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonAPIGatewayInvokeFullAccess)

[AmazonCognitoDeveloperAuthenticatedIdentities](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonCognitoDeveloperAuthenticatedIdentities)

[AmazonCognitoPowerUser](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonCognitoPowerUser)

[AmazonESCognitoAccess](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonESCognitoAccess)

[AWSLambda\_FullAccess](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAWSLambda_FullAccess)

[AmazonDynamoDBFullAccess](https://us-east-1.console.aws.amazon.com/iamv2/home#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonDynamoDBFullAccess)

**Section One: Host a Static Website**

**Part one: Create GitHub Depository**

1. Select the region in the top right-hand corner. Make sure that it is N. Virginia!
2. Log into your Cloud9Admisistrators account.
3. Enter the [AWS CodeCommit](https://aws.amazon.com/codecommit/) app either by clicking on the previous link or by searching for it on the AWS website.
4. Select **Create Repository** and name it “Project2” and hit **Create**.
5. Log back into your root account.

**Part two: Cloning the Git Project and Entering Website Files**

1. Navigate to IAM app and select the Cloud9Administrators user.
2. 
                   Generating Git credentials in the IAM console
               Click on the **Security Credentials** tab and scroll down to the **HTTPS Git credentials for AWS CodeCommit** and select **Generate.**
3. Copy both the username and password that is generated. **IMPORTANT! THIS MUST BE DONE NOW! Once the password is generated, it is impossible to view again.**
4. Go to the pull-down bar labelled **Clone URL** and select **Clone HTTPS**
5. Go to your Cloud9 IDE and start a terminal. Enter the following:

git clone YOUR CLONED URL HERE

If this works, it will ask for a username and password. Entered the generated username and password generated from step 8.

1. The following cod will obtain files from the S3 bucket used in this tutorial. Enter the following that is highlighted and hit enter after every paste:

*This opens your Project2 folder in the terminal.*

cd Project2

*This enters the static website file to the folder.*

aws s3 cp s3://wildrydes-us-east-1/WebApplication/1\_StaticWebHosting/website ./ --recursive

*Use the following Git commands to commit all of these files to the Project2 folder.*

git add .

git commit -m 'new'

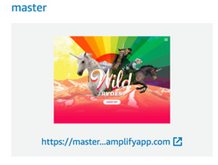
git push

The following show output in the terminal if it works:

**Counting objects: 95, done.  
Compressing objects: 100% (94/94), done.  
Writing objects: 100% (95/95), 9.44 MiB | 14.87 MiB/s, done.  
Total 95 (delta 2), reused 0 (delta 0)  
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Project2  
\* [new branch] master -> master**

**Part 3: Web Hosting with Amplify Console**

1. Either click on this [AWS Amplify Console](https://aws.amazon.com/amplify/console/) or search the AWS website for the app.
2. Select **Get Started**
3. Select **New App** and then **Host Web App**
4. Select **CodeCommit** under **Get started with Amplify Hosting**
5. Select **AWS CodeCommit** as the service provider and hit **Continue**
6. Select your “Project2” repository from the dropped down menu and hit **Next**
7. Under the Configure build settings tab, select **Allow AWS Amplify to automatically deploy all files hosted in your project root directory** and hit the **Next** button.
8. Hit **Save and deploy**
9. Click on the picture to launch the website.



**Part 4: Editing Files from CodeCommit**

1. Go back to CodeCommit
2. Click **Project2**
3. Click **index.html**
4. Click **Edit**
5. Use CTRL + F “Wild Rydes” and change it to “Wild Rydes – Rydes of the Future!”. Do this for the other “Wild Rydes” in this file as well.

Section Two: Manage Users

Section Three: Build a Serverless Backend

Section Four: Deploy a RESTful API