**Deploying a static web application on AWS**

**Overview**

Static web pages are managed by AWS Amplify Console and served via Amazon CloudFront. End user is able to access your site using the public website URL exposed by AWS Amplify console. There are no need to run any web servers or use other services in order to make the site available.

**Pre-requisites**

* AWS Amplify Console
* RESTful APIs
* AWS Lambda
* AWS API Gateway

**Step 1: Select Region**

* AWS Amplify Console
* AWS Codecommit

North America: N. Virginia, Ohio, Oregon

Europe: Ireland, London, Frankfurt

Asia Pacific: Tokyo, Seoul, Singapore, Sydney, Mumbai

**Step 2: Create the git repository - setting up the codecommit repository**

A. Create new CodeCommit repository using the following command

Aws codecommit create-repository –repository-name wild-rydes

B. Clone the existing workshop repository from github

Git clone <https://github.com/aws-samples/aws-serverless-webapp-workshop.git>

C. Change into the workshop repository directory

cd aws-serverless-webapp-workshop

D. Split out the WildRydesVue code into its own branch

sudo yum install git-subtree -y

git subtree split -P resources/code/WildRydesVue -b WildRydesVue

E. Create a new directory for your CodeCommit repo using following command

Mkdir ../wild-rydes && cd ../wild-rydes

F. Initialize a new git repository

Git init

G.Pull the WildRydesVue branch into your new repo

git pull ../aws-serverless-webapp-workshop WildRydesVue

H.Add your codecommit repository as a remote

Git remote add origin codecommit://wild-rydes

I.Push the code to your new CodeCommit repository

git push -u origin master

J.Remove the temporary local repository you created in step b

rm -rf ../aws-serverless-webapp-workshop

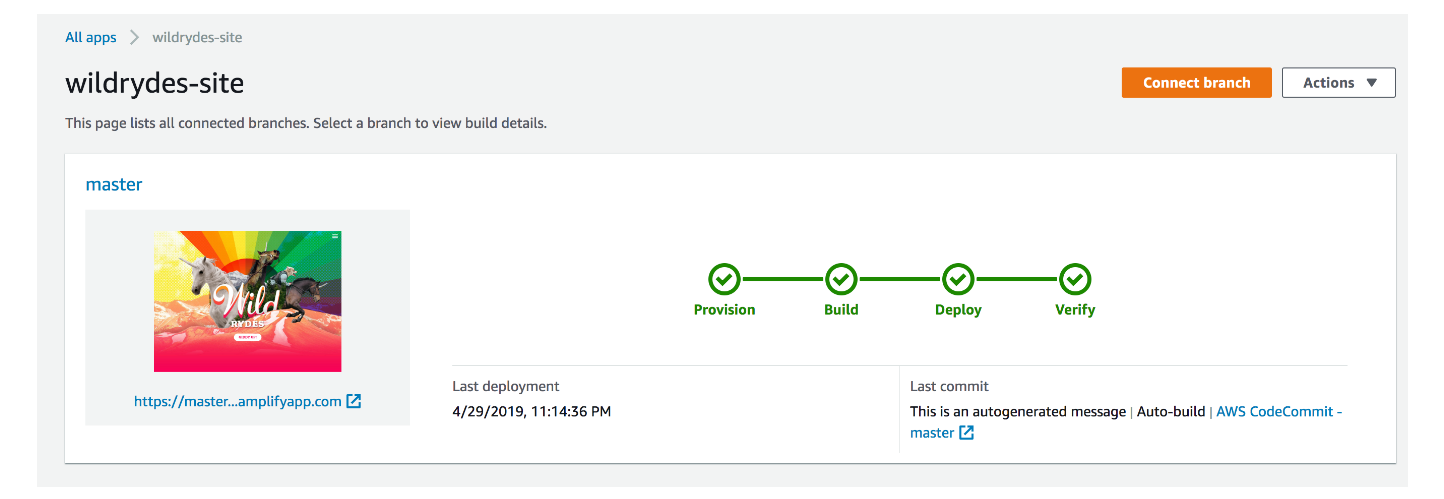
**Step 3: Deploying the site with AWS Amplify Console**

AWS Amplify Console takes care of the work of setting up a place to store your static web application code and provides a number of helpful capabilities to simplify both the lifecycle of that application.

1. Lauch the Amplify console using the following link
2. Select “Host your web app” nu clicking on the Get started button
3. Select AWS CodeCommit and click continue
4. From dropdown select repository and master branch created today and select NEXT
5. Aplify detects existing amplify backend. Select Create New Environment and name it prod in lowercase

**Step 4: Create Service Role with Permissions to deploy the application backend**

1. Click on Create new role, check that Amplify is selected and click Next permissions, click Next: Tags, click Next: Review.
2. Give the Role a new name: wildrydes-backend-role and click Create role.
3. Search for wildrydes-backend-role from the search filter, and click the role name.
4. Click Attach policies under the Permissions tab, search for AWSCodeCommitReadOnly policy, click on the checkbox next to the policy name, and click Attach Policy.
5. Close this tab and return to the AWS Amplify Build configure console.
6. Refresh the role list by clicking on the circular arrow button, and select the role created in the step above.
7. Select Next
8. Review, select Save and Deploy
9. It takes a fdew minutes to load the content of the web application



**Step 5: Modify the website**

AWS Amplify Console will rebuild and redeploy the app when it detects changes to the connected repository

1. Navigate to the Cloud9 environment open the index.html file in the /wild-rydes/public/ directory of the repository
2. Modify the title line:

<title>wildrydes</title>

TO

<title>Wild Rydes - Rydes of the Future!</title>

Save the file

1. Commit again to the git repository using the following command

git add .

git commit -m “updated title”

git push

1. Once completed, re-open the Wild Rydes and notice the title change