

Lab 1: Deploying a Simple Serverless Application

1 hour 30 minutes Free ★★★★★ [Rate Lab](#)

aws training and certification

© 2021 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Corrections, feedback, or other questions? Contact us at [AWS Training and Certification](#).

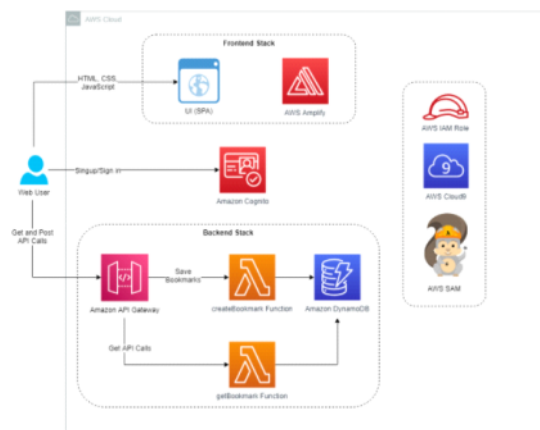
Overview

Your technical support team has a partially homegrown, highly customized ticketing system. Between development team members and tech support, you are often pointing people to links to help resolve issues, but there are no real strategies for managing the resources. You have a team knowledge base, which is maintained by the development managers, but new suggestions must be emailed to a shared mailbox for review before they are published. So submissions, reviews, and updates happen sporadically.

A recent hackathon produced a partial solution for saving resource links with relevant information. Your product manager has asked your team to build the first iteration of this application as an opportunity to address the business need while learning about and piloting a serverless architecture.

In this lab, you deploy the backend application and front-end application in various steps, and update the front-end configurations.

The following diagram shows the architecture components that have been or will be deployed in this lab.



This lab uses the following software stack:

- AWS Amplify
- AWS Serverless Application Model (AWS SAM)

This lab uses the following software stack:

- AWS Amplify
- AWS Serverless Application Model (AWS SAM)
- Amazon Cognito
- Vue JavaScript framework
- AWS Cloud9
- Swagger API
- Amazon DynamoDB

Objectives

After completing this lab, you will be able to:

- Configure authentication through an Amazon Cognito user pool
- Deploy your backend code using AWS SAM
- View API documentation using the Swagger Editor
- Update your front-end configuration and run the build through AWS Cloud9 to test it prior to deployment
- Deploy your front-end application using Amplify

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS X, or Linux (Ubuntu, SUSE, or Red Hat)
- For Microsoft Windows users, administrator access to the computer
- An internet browser such as Chrome, Firefox, or Internet Explorer 9 (previous versions of Internet Explorer are not supported)
- A text editor

⚠ Note The lab environment is not accessible using an iPad or tablet device, but you can use these devices to access the lab guide.

Duration

This lab requires approximately **60 minutes** to complete.

Start Lab

1. At the top of your screen, launch your lab by choosing **Start Lab**

This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

❗ If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing **Open Console**

This opens an AWS Management Console sign-in page.

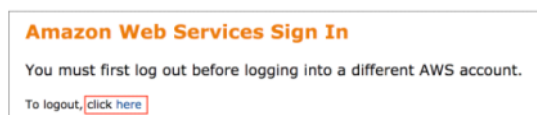
3. On the sign-in page, configure:

- **IAM user name:** `awsstudent`
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose **Sign In**

⚠ Do not change the Region unless instructed.

Common Login Errors

Error: You must first log out



If you see the message, **You must first log out before logging into a**

different AWS account:

- Choose [click here](#)
- Close your browser tab to return to your initial lab window
- Choose [Open Console](#) again

Task 1: Understanding key services and the environment setup

In this task, you look at the different AWS services that are used in this development. Take a few minutes to navigate the console for each service. Next, you download your source code, including both backend and front-end code, from an Amazon Simple Storage Service (Amazon S3) bucket to your AWS Cloud9 environment.

- **AWS Amplify** is a set of tools and services that enables mobile and front-end web developers to build secure, scalable full stack applications powered by AWS. Amplify includes an open-source framework with use-case-centric libraries and a powerful toolchain to create and add cloud-based features to your application, and a web-hosting service to deploy static web applications.
- **AWS SAM** is an open-source framework for building serverless applications. It provides shorthand syntax to express functions, APIs, databases, and event source mappings. With just a few lines per resource, you can define the application you want and model it using YAML. During deployment, AWS SAM transforms and expands the AWS SAM syntax into AWS CloudFormation syntax, enabling you to build serverless applications faster.
- **Amazon Cognito** lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.
- **Vue JavaScript framework** is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed to be incrementally adoptable. The core library focuses on the view layer only and is easy to pick up and integrate with other libraries or existing projects. Vue is also perfectly capable of powering sophisticated single-page applications when used in combination with modern tooling and supporting libraries.
- **AWS Cloud9** is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. AWS Cloud9 makes it easy to write, run, and debug serverless applications. It pre-configures the development environment with all the SDKs, libraries, and plugins needed for serverless development.
- **Swagger API** is an open-source software framework backed by a large ecosystem of tools that help developers design, build, document, and consume RESTful web services. Swagger also allows you to understand and test your backend API specifically.
- **Amazon DynamoDB** is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-Region, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

To set up your environment, open the pre-provisioned AWS Cloud9 environment. From there, download and unzip your source code so that you can deploy the AWS SAM application in an upcoming task.

4. Choose [Services](#) and select **Cloud9**.
5. In the left navigation pane, choose **Your environments**.
6. For the **BookmarkAppDevEnv** environment, choose [Open IDE](#)

Within a few seconds, the AWS Cloud9 environment launches. Notice the Linux-style terminal window in the bottom pane.

7. To download your source code, run the following commands:

```
cd ~/environment
wget https://us-west-2-tcprod.s3-us-west-2.amazonaws.com/courses/ILT-TF-200-SVDVSS/v1.0.2/lab-1-Bookmarks/scripts/app-code.zip
unzip app-code.zip
cd app-code
```

Keep the browser tab that's running the AWS Cloud9 IDE open because you will use it throughout this lab.

```
~/environment wget https://us-west-2-tcprod.s3-us-west-2.amazonaws.com/courses/ILT-TF-200-SVDVSS/v1.0.2/lab-1-Bookmarks/scripts/app-code.zip unzip app-code.zip cd app-code
```

Task 2: Setting up authentication with Amazon Cognito

In this task, you set up your Amazon Cognito user pool for authentication.

8. From the AWS Management Console, choose **Services** and open **Amazon Cognito** in a new browser tab.
9. Choose **Manage User Pools**.
10. Choose **Create a user pool**.
11. For **Pool name**, enter **bookmark-app-userpool**.
12. Choose **Review defaults**.

From here, modify a few of the default settings.

13. On the left side of the page, choose **Policies**.
14. Under **What password strength do you want to require?**, clear the check boxes for **Require special character** and **Require uppercase letters**. Leave the other boxes selected to keep the password simple during account creation. Refer to the following screenshot.



15. Choose **Save changes**.
16. On the left side of the page, choose **Triggers**.
17. From the **Pre sign-up** dropdown list, choose the **PresignupLambda- AWS Lambda** function.
- Note** This Lambda function automatically verifies your email and phone number during the sign-up process.
18. Choose **Save changes**.
19. On the left side of the page, choose **App clients**.
20. Choose **Add an app client**.
21. For **App client name**, enter **AppClientForBookmarkUserPool**.
22. Clear the box for **Generate client secret**.
23. Choose **Create app client**.
24. On the left side of the page, choose **Review**, and review your user pool settings.


25. Choose **Create pool**.
26. Copy the **Pool id** and paste it into a text editor for later use.
27. On the left side of the page, choose **App clients**, and then copy and paste the **App client id** into a text editor for later use.

Task 3: Deploying the backend application using AWS SAM

In this task, you run AWS Command Line Interface (AWS CLI) commands to deploy the backend code using AWS SAM.

The following are a few key AWS SAM CLI commands to know:

- **sam build**: Builds a serverless application and prepares it for subsequent steps in your workflow, such as locally testing the application or deploying it to the AWS Cloud
- **sam deploy**: Deploys an AWS SAM application
- **sam init**: Initializes a serverless application with an AWS SAM template

 **Learn more** For more information about AWS SAM CLI commands, see [AWS SAM CLI Command Reference](#).

28. Go back to the AWS Cloud9 browser tab.
29. On the left side of the AWS Cloud9 IDE, choose the dropdown arrow next to the **app-code** folder, and then the **backend** folder.
30. Choose and open the **template.yaml** file.

This file contains the AWS SAM template that defines your application's AWS resources. Take a moment to scroll through and review the anatomy of the architecture.
31. On the left side of the AWS Cloud9 IDE, choose the dropdown arrow next to the **src** folder, and then choose the dropdown arrow next to the **createBookmark** folder.
32. Choose and open the **index.js** file to view the code within this file.

This file contains your actual Lambda handler logic for the **createBookmark** Lambda function. Review the other Lambda functions in the **deleteBookmark**, **getBookmark**, **listBookmarks**, and **updateBookmark** folders.

33. Choose and open the **samconfig.toml** file.

This is the configuration file for the project. You need to update it with the following code before deploying.
34. To update the code in the **samconfig.toml** file, Replace the following parameters with the values from the left side of the lab instructions:
 - *(REPLACE WITH S3 BUCKET ARN)*: **BookmarkBucket**
 - *(REPLACE WITH REGION)*: **Region**
 - *(REPLACE WITH SAM ROLE ARN)*: **SamDeploymentRole**
35. To save the updated **samconfig.toml** file, on the **File** menu, choose **Save**.

Note This file is usually generated during the **sam deploy --guided** command, but for this lab, the prebuilt file is provided. This file contains the parameters of the Amazon S3 bucket from where the code is deployed, the role needed for AWS SAM to deploy, and the Region. The file contains the parameters for the CloudFormation stack to be deployed.

36. In the AWS Cloud9 terminal, run the following command:

```
cd backend
sam deploy
```

This command deploys your application to the AWS Cloud. The command takes the deployment artifacts you build with the **sam build** command, packages and uploads them to an Amazon S3 bucket created by the AWS SAM CLI, and deploys the application using CloudFormation.

Note The command takes a minute or two to run.

 **Learn more** The AWS SAM CLI comes preinstalled on Cloud9. For more information about installing the AWS SAM CLI, see <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install-linux.html>.

You can see the output for the changeset. Notice the operations, in this case **+ Add**, and look under **ResourceType** to see the AWS resources about to be deployed.

The changeset is now being deployed. You can watch the sequence of events from your AWS Cloud9 instance and from CloudFormation.

37. Go to your browser tab with the AWS Management Console. Choose **Services** and open **CloudFormation** in a new browser tab.

38. Choose the **sam-bookmark-app** stack.

This is the stack that was deployed from your backend code in AWS Cloud9.

39. To view the AWS resources that were created within this stack, choose the **Resources** tab.

40. To open the DynamoDB resource, choose the **Physical ID** of the **bookmarksTable**.

This is the **sambookmark-app-bookmarksTable** table that was created when deploying your backend code via AWS SAM.

41. Choose the **Items** tab.

Nothing is populating the table yet because no bookmarks have been added.

42. Choose **Services** and select **API Gateway**.

This is the **Bookmark App** API that was launched through the AWS SAM template.

43. Choose the **Bookmark App**.

44. On the left side of the page, choose **Stages**.

45. Choose **dev**, and then copy and paste the **Invoke URL** into a text editor for later use.

46. Choose the **Export** tab.

47. Choose **Export as Swagger + API Gateway Extensions**, and select the JSON or YAML option.

This saves the file to your local directory for you to open and view.


View the API documentation using Swagger Editor

48. Choose the following link: <https://editor.swagger.io/>

49. Choose **File** and **Import file** to import the file that you just saved.

The right side of the console displays all deployed APIs.

50. Expand each API to check the parameters and the functionality.

 **Learn more** For more information about integrating Amazon API Gateway with Swagger, see [API Gateway Integration](#).

Task 4: Reviewing the front-end application code and updating the

configuration file

Before deploying the bookmark application through Amplify, you can view the source code and test the front-end application from AWS Cloud9. You can also make changes to the code by reworking some of the source files within AWS Cloud9.

51. Return to the Cloud9 IDE browser tab.

52. To switch to your **frontend** directory and install packages and dependencies, run the following commands:

```
cd ../frontend
npm install
```

53. To run the dev server, run the following command:

```
npm run dev
```

54. On the left side of the AWS Cloud9 IDE, choose the dropdown arrow next to the **frontend** folder, and then choose the dropdown arrow next to the **src** folder.

55. Choose and open the **main.js** file to view the code. The following is an example.

```
import Vue from 'vue'
import App from './App'
import router from './router'

import '@aws-amplify/ui-vue';
import Amplify from '@aws-amplify/core';
import { Auth } from '@aws-amplify/auth';
import awsmobile from './aws-exports';

Amplify.configure(awsmobile);

Auth.configure(awsmobile)

Vue.config.productionTip = false

/* eslint-disable no-new */
new Vue({
  el: '#app',
  router,
  components: { App },
  template: '<App/>'
})
```

Notice that the Vue framework has been imported along with **import { Auth } from '@aws-amplify/auth'**, which is used for Amazon Cognito authentication.

Next, you need to update the **aws-exports.js** file to add the Amazon Cognito user pool that was created earlier.

56. Choose and open the **aws-exports.js** file.

57. Copy and paste the following code block into the file:

```
const awsmobile = {
  "aws_project_region": "(REPLACE WITH REGION)",
  "aws_cognito_region": "(REPLACE WITH REGION)",
  "aws_user_pools_id": "(REPLACE WITH COGNITO ID POOL)",
  "aws_user_pools_web_client_id": "(REPLACE WITH APP CLIENT ID)",
  "oauth": {},
  "aws_cloud_logic_custom": [
    {
      "name": "Bookmark App",
      "endpoint": "(REPLACE WITH API GATEWAY INVOKE URL)",
      "region": "(REPLACE WITH REGION)"
    }
  ]
};

export default awsmobile;
```

58. Replace the following parameters with the values copied from tasks 2 and 3. Find the **Region** value on the left side of the lab instructions.

- *(REPLACE WITH REGION):* **Region**
- *(REPLACE WITH APP CLIENT ID):* **App client ID**
- *(REPLACE WITH COGNITO ID POOL):* **Pool ID**

59. Once you have made these changes, save the **aws-exports.js** file.

```
const omnibot = {
  "aws_project_region": "us-west-2",
  "aws_cognito_region": "us-west-2",
  "aws_user_pool_id": "us-west-2_8P7H8888g",
  "aws_user_pool_web_client_id": "4af6phg3l0n0hphed8514q",
  "auth": {},
  "aws_cloud_logic_custom": [
    {
      "name": "Backend App",
      "endpoint": "https://nlm6lzd52.execute-api.us-west-2.amazonaws.com/dev",
      "region": "us-west-2"
    }
  ]
};

export default omnibot;
```

50. To stop the bookmark application from running, press **Ctrl + C**.

```
npm run build
```

52. From the AWS Cloud9 terminal, run the following command:

```
cd dist
ls
```

53. To zip the contents of the build folder, run the following command:

```
zip -r app.zip *
```

```
aws s3 cp app.zip s3://(BOOKMARKBUCKET)
```

- **App name:** Enter `BookmarkApp`
- **Environment name:** Enter `dev`

- **Method:** Choose **Amazon S3**
- **Bucket:** Choose **bookmarkbucket**
- **Zip file:** Select **app.zip** (When the **Bucket** is selected, this dropdown menu auto-populates.)

70. Choose **Save and deploy**

That's it! You just deployed your front-end code to AWS via Amplify. You should see a green **Deployment successfully completed** bar in the middle of your page.

Task 6: Testing the bookmark application

In this task, you test the bookmark application that was deployed in Amplify in the previous task. You start by creating an account, and then you add bookmarks to the application. Verify the addition by checking the DynamoDB table.

71. From the Amplify console, choose the URL under **Domain**.

This opens the bookmark application.

72. From the bookmark application page, choose **Create account**

73. Fill in the fields with your information, and choose **CREATE ACCOUNT**

74. To add a bookmark, choose the plus icon at the top-right corner of the page.

75. For **Add New Bookmark**, configure the following information:

- **Name:** Enter `aws`
- **Description:** Enter `aws cloud training`
- **Bookmark URL:** Enter `aws.training`

76. Choose **ADD BOOKMARK**

77. Add another bookmark of your choice.

When you add these bookmarks on the front end, DynamoDB adds this data to the **bookmarks-app-bookmarksTable** table.


78. From the AWS Management Console, choose **Services ▾** and select **DynamoDB**.

79. On the left side of the page, choose **Tables**.

30. Choose **sam-bookmark-app-bookmarksTable**.

31. Choose the **Items** tab.

This is where the bookmarks that you just added are stored.

32. Go back to the bookmark application browser tab, and choose the trash can  icon to delete all of your bookmarks.


When the bookmarks are deleted, DynamoDB deletes this data from the **sam-bookmark-app-bookmarksTable** table.

33. Go to the DynamoDB console browser tab.

34. Choose the refresh button at the top-right corner of the page.

You can see that the deleted bookmarks are no longer in the table.

Conclusion

 Congratulations! You now have successfully:

- Configured authentication through an Amazon Cognito user pool
- Deployed your backend code using AWS SAM
- Viewed API documentation using the Swagger Editor
- Updated your front-end configuration file and ran the build through AWS Cloud9 to test it prior to deployment
- Deployed your front-end application using Amplify

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

35. Return to the AWS Management Console.

36. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

37. Choose **End Lab**

38. Choose **OK**

39. (Optional):

- Select the applicable number of stars ☆
- Type a comment
- Choose **Submit**
 - 1 star = Very dissatisfied
 - 2 stars = Dissatisfied
 - 3 stars = Neutral
 - 4 stars = Satisfied
 - 5 stars = Very satisfied

You may close the window if you don't want to provide feedback.

For more information about AWS Training and Certification, see

<http://aws.amazon.com/training/>.

Your feedback is welcome and appreciated.

If you would like to share any feedback, suggestions, or corrections, please provide the details in our [AWS Training and Certification Contact Form](#).

Additional resources

- For more information about Swagger, see <https://swagger.io/tools/swagger-ui/>.
- For more information about the Vue JavaScript framework, see <https://vuejs.org/>.