Serverless SaaS API Integration Deployment Guide

Summary:

This guide is for independent software vendors (ISVs), selling SaaS products in AWS Marketplace. The guide covers the Serverless SaaS application that automates the SaaS API integration phase. It is a step-by-step guide to set up the Serverless SaaS API integration using AWS Cloud9 service. It provides a web UI based option to deploy the Serverless application.

Pre-requisites:

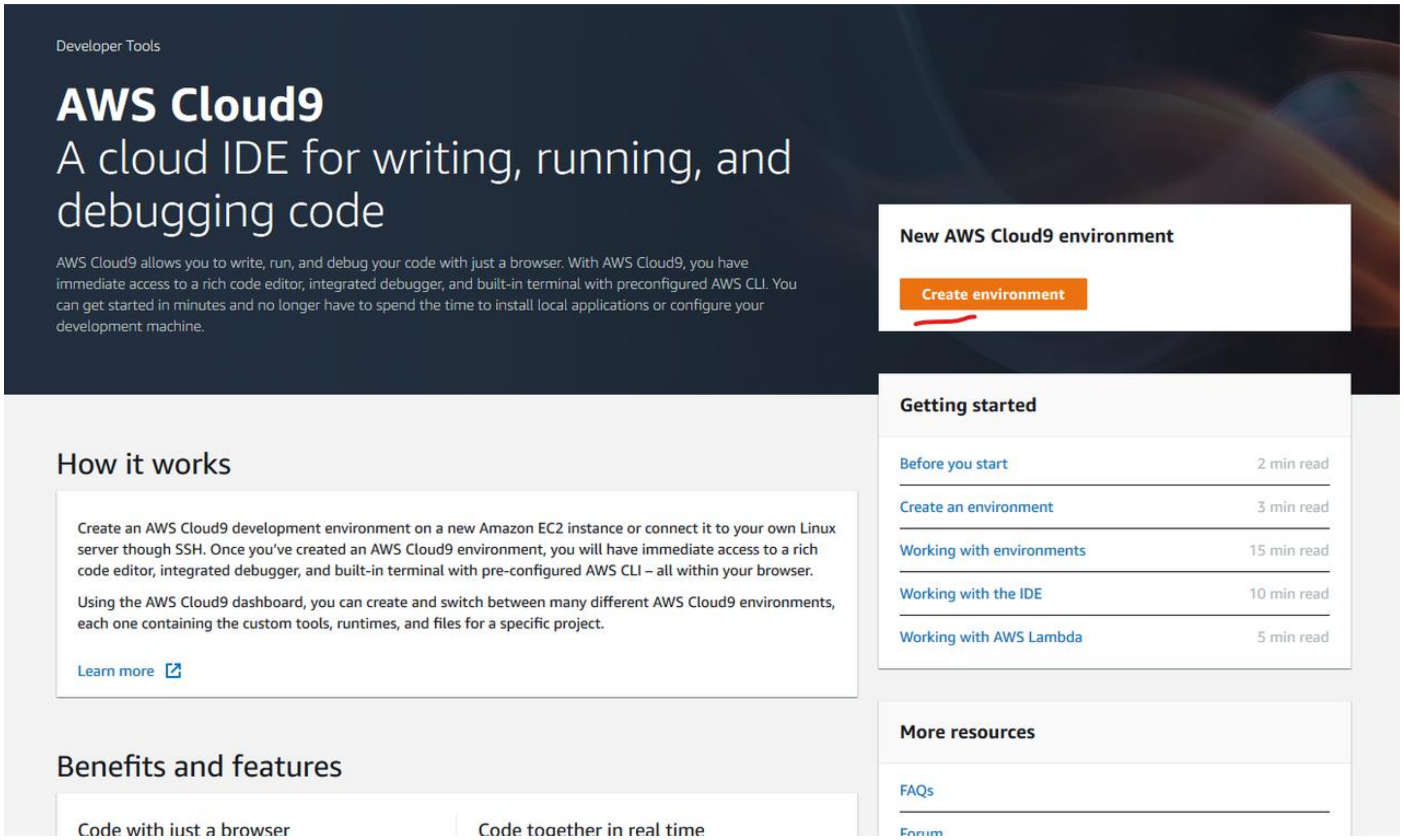
You need to register as a paid Seller on AWS Marketplace, finalize a SaaS pricing model and publish a SaaS product to limited view before you deploy the SaaS API integration. To get started as Seller on AWS Marketplace, click here. You can find information on supported SaaS pricing models and how to list them here.

Below is the information you need from the AWS Marketplace Seller Operations teams before you proceed to next steps:

- Product code - A unique identifier of your AWS Marketplace listing
- SNS topics associated with your listing.

Setting up your Cloud9 environment:

- Log into your seller AWS account. Your user account should have admin or root access to the seller AWS account to create new resources.
- Go to your Cloud9 launch page: https://console.aws.amazon.com/cloud9/home/product#
- Click on “Create environment” button to launch your environment



Developer Tools

AWS Cloud9

A cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

New AWS Cloud9 environment

Create environment

Getting started

Before you start	2 min read
Create an environment	3 min read
Working with environments	15 min read
Working with the IDE	10 min read
Working with AWS Lambda	5 min read

How it works

Create an AWS Cloud9 development environment on a new Amazon EC2 instance or connect it to your own Linux server though SSH. Once you've created an AWS Cloud9 environment, you will have immediate access to a rich code editor, integrated debugger, and built-in terminal with pre-configured AWS CLI – all within your browser.

Using the AWS Cloud9 dashboard, you can create and switch between many different AWS Cloud9 environments, each one containing the custom tools, runtimes, and files for a specific project.

Learn more

More resources

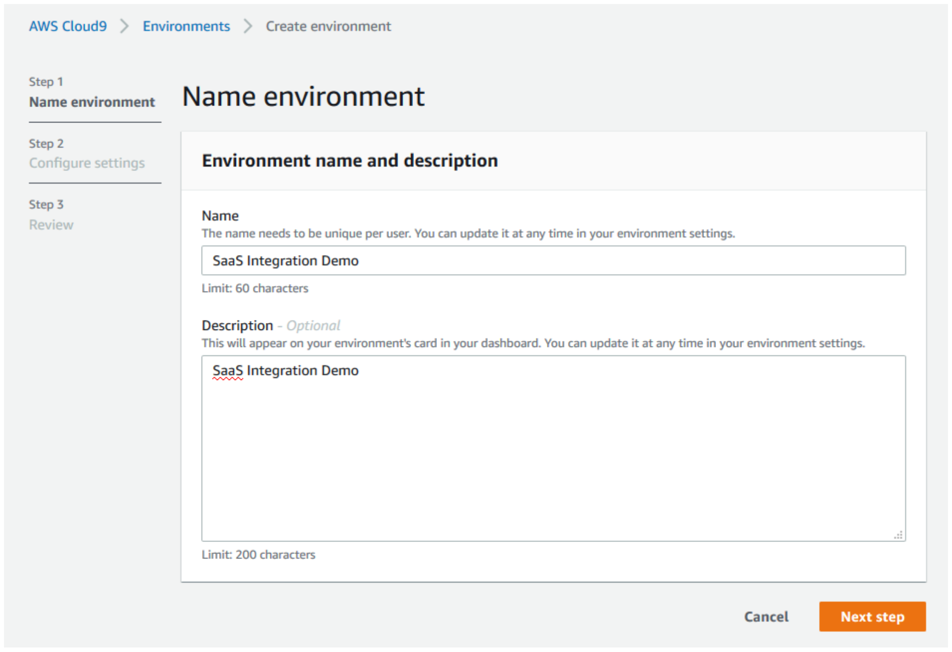
FAQs

Forum

Benefits and features

Code with just a browser

Code together in real time


AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

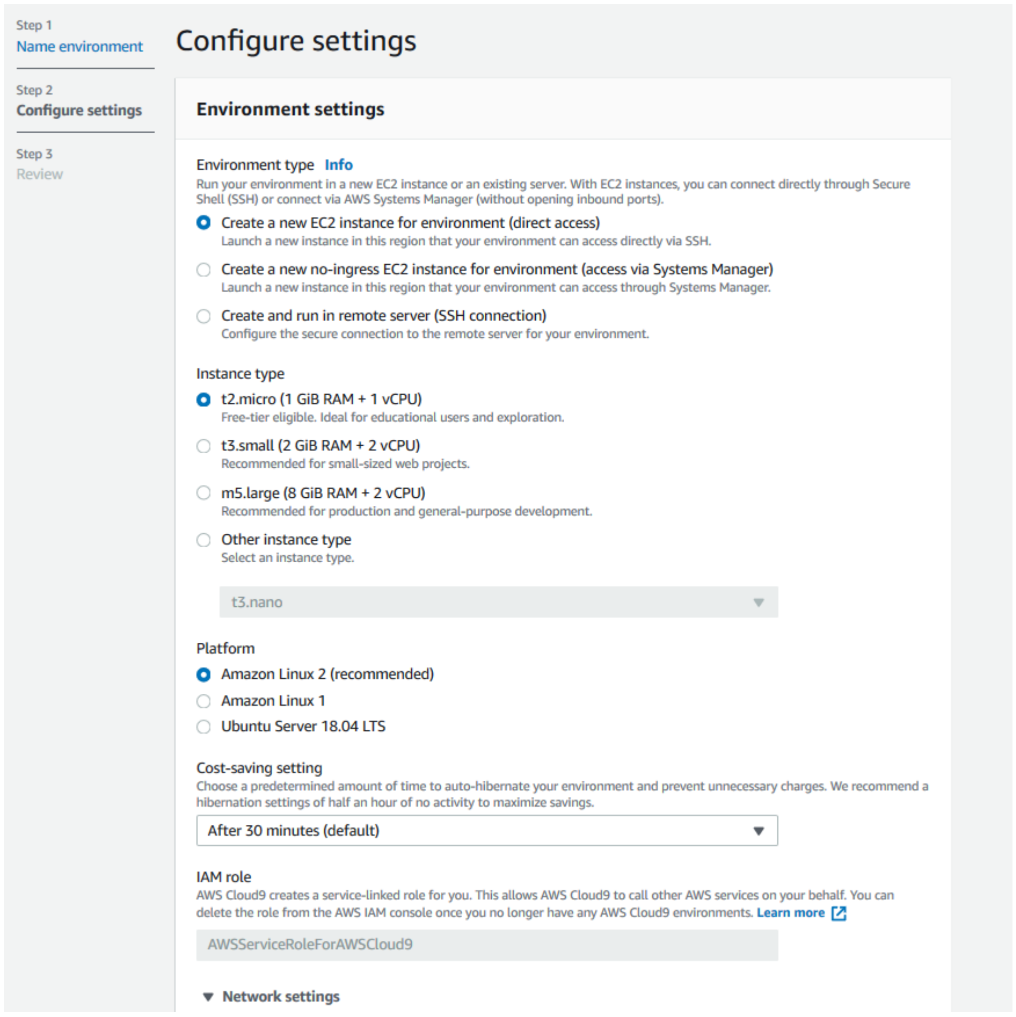
Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.
SaaS Integration Demo
Limit: 60 characters

Description - *Optional*
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.
SaaS Integration Demo
Limit: 200 characters

- Leave the defaults for **Environment type**, **Instance type, Platform, Cost Saving-Setting, and IAM Role.**
- Create a **new** VPC and Subnet or use an **existing** VPC and Subnet for hosting the Cloud9 environment.


Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Configure settings

Environment settings

Environment type Info
Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

● Create a new EC2 instance for environment (direct access)
Launch a new instance in this region that your environment can access directly via SSH.

○ Create a new no-ingress EC2 instance for environment (access via Systems Manager)
Launch a new instance in this region that your environment can access through Systems Manager.

○ Create and run in remote server (SSH connection)
Configure the secure connection to the remote server for your environment.

Instance type

● t2.micro (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

○ t3.small (2 GiB RAM + 2 vCPU)
Recommended for small-sized web projects.

○ m5.large (8 GiB RAM + 2 vCPU)
Recommended for production and general-purpose development.

○ Other instance type
Select an instance type.
t3.nano

Platform

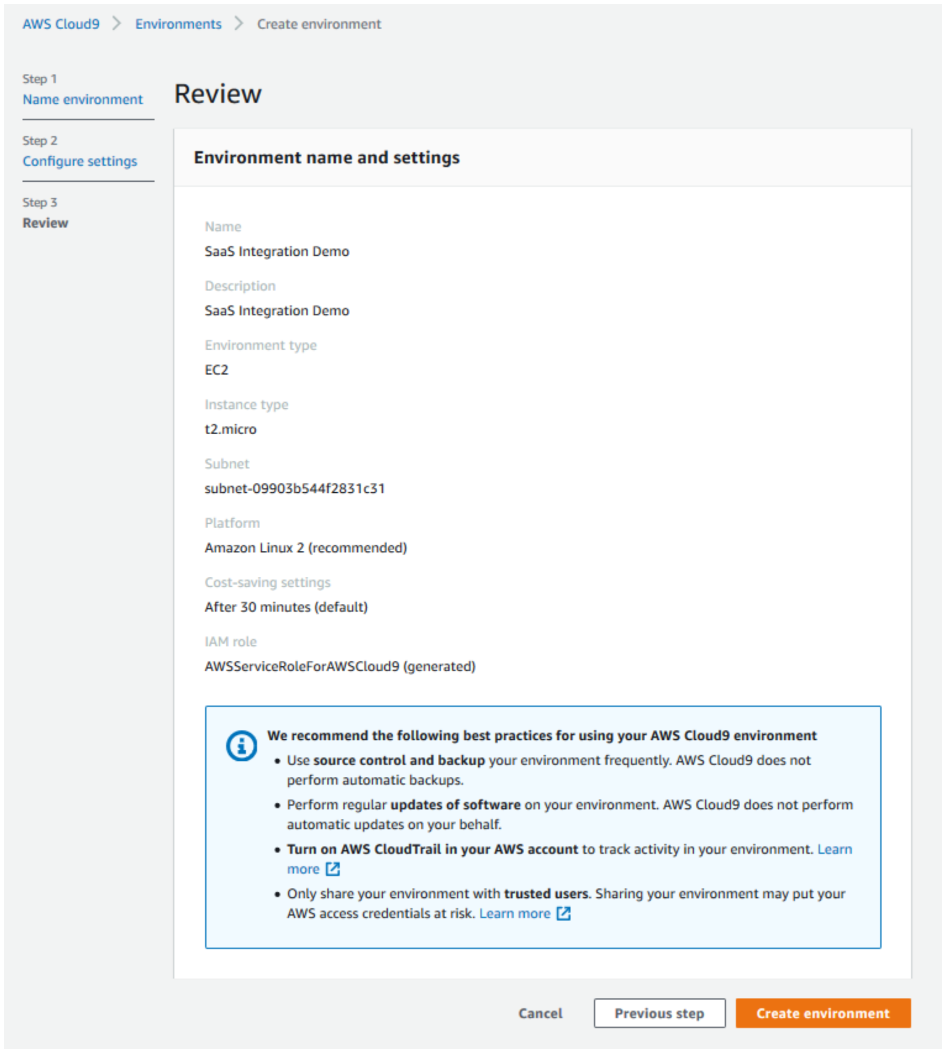
● Amazon Linux 2 (recommended)
○ Amazon Linux 1
○ Ubuntu Server 18.04 LTS

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.
After 30 minutes (default)

IAM role
AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. Learn more
AWSServiceRoleForAWSCloud9

▼ Network settings

- Click **'Next Step'**, click on **'Create Environment'**. It will take about 5 minutes for your environment to be ready.



Review

Environment name and settings

Name
SaaS Integration Demo

Description
SaaS Integration Demo

Environment type
EC2

Instance type
t2.micro

Subnet
subnet-09903b544f2831c31

Platform
Amazon Linux 2 (recommended)

Cost-saving settings
After 30 minutes (default)

IAM role
AWSServiceRoleForAWSCloud9 (generated)

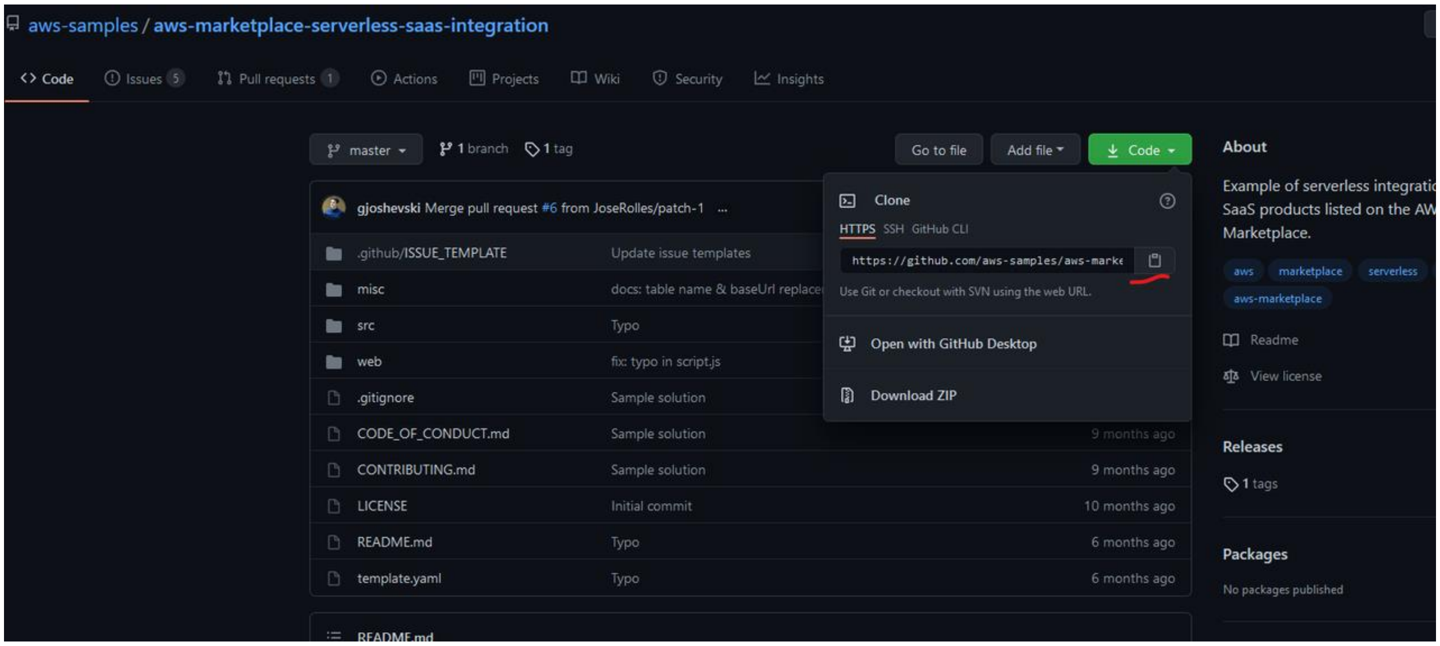
We recommend the following best practices for using your AWS Cloud9 environment
- Use **source control and backup** your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular **updates of software** on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- **Turn on AWS CloudTrail in your AWS account** to track activity in your environment. Learn more
- Only share your environment with **trusted users**. Sharing your environment may put your AWS access credentials at risk. Learn more

Implementation:

Deploying the integration resources:

- Once your IDE is ready, go to the Github repository to clone it. Copy the URL.
  Github Repo URL: https://github.com/aws-samples/aws-marketplace-serverless-saas-integration

- Click on the "Source Control" icon on the left side bar, then select "Clone Repository". Paste the Github repo URL from the previous step and press enter.



gjoshevski Merge pull request #6 from JoseRolles/patch-1 ...

.github/ISSUE_TEMPLATE	Update issue templates	
misc	docs: table name & baseUrl replace	
src	Typo	
web	fix: typo in script.js	
.gitignore	Sample solution	
CODE_OF_CONDUCT.md	Sample solution	9 months ago
CONTRIBUTING.md	Sample solution	9 months ago
LICENSE	Initial commit	10 months ago
README.md	Typo	6 months ago
template.yaml	Typo	6 months ago

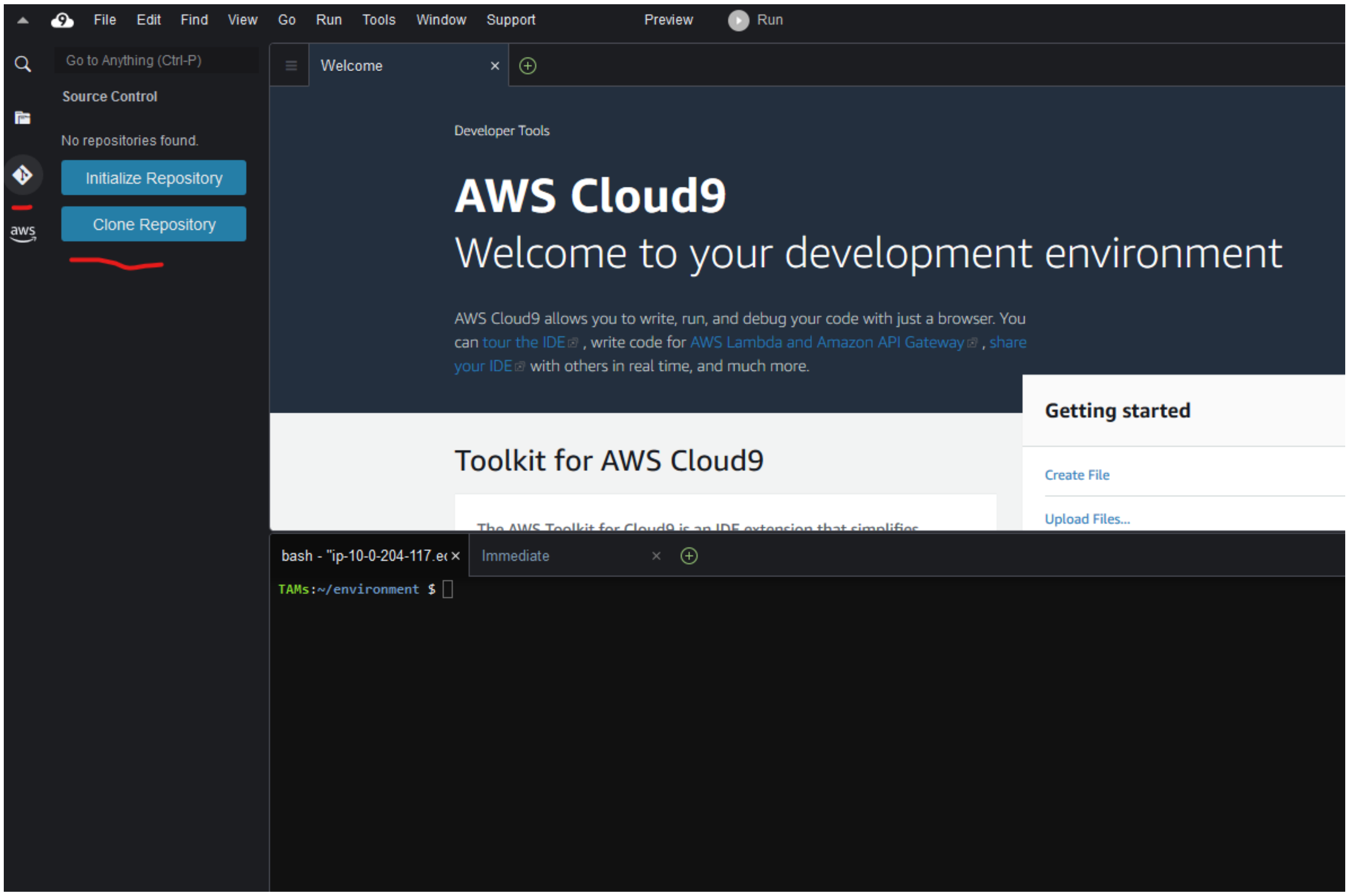
README.md

```
https://github.com/aws-samples/aws-marke
```

Use Git or checkout with SVN using the web URL.

Example of serverless integratio SaaS products listed on the AW Marketplace.

No packages published



No repositories found.

Developer Tools

AWS Cloud9

Welcome to your development environment

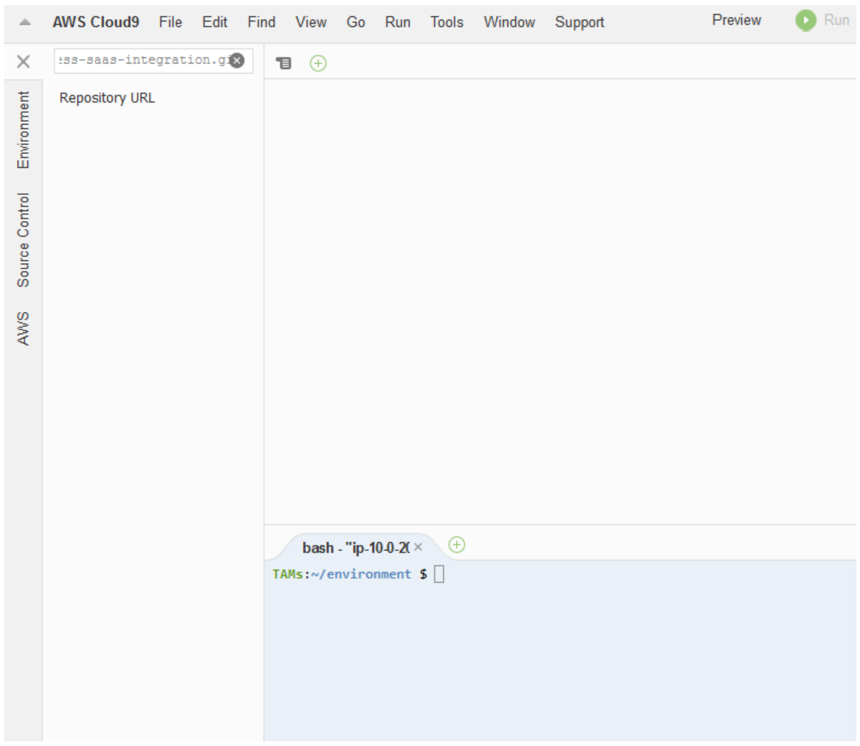
AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can tour the IDE, write code for AWS Lambda and Amazon API Gateway, share your IDE with others in real time, and much more.

Getting started

Toolkit for AWS Cloud9

The AWS Toolkit for Cloud9 is an IDE extension that simplifies

```
bash - "ip-10-0-204-117.ec
TAMs:~/environment $
```

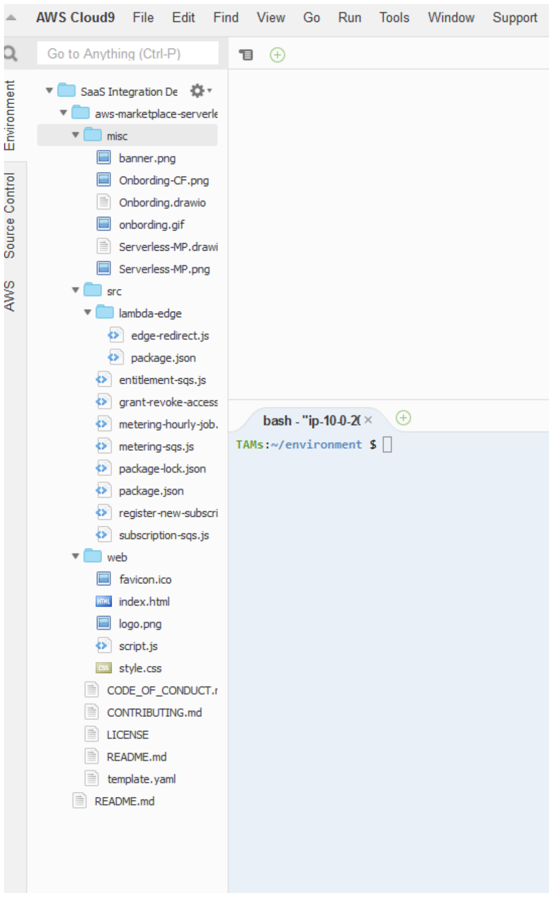


```
:ss-saas-integration.gi
```

```
TAMs:~/environment $
```

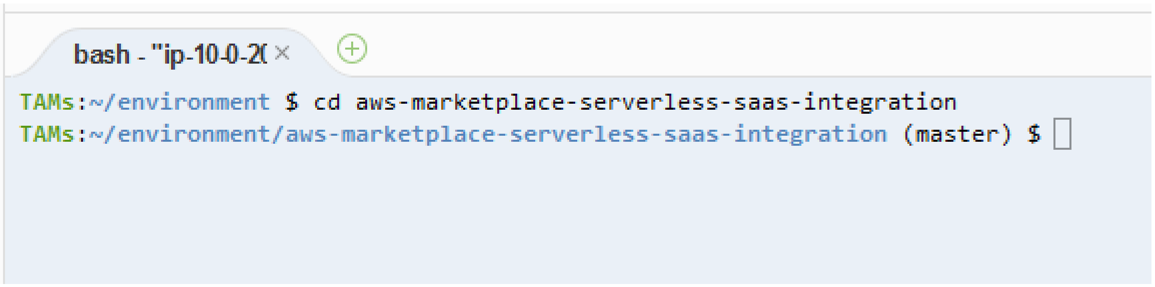
- Select “Environment” from the left side bar and expand the cloned repository folder. You should now see all your files. Change your shell directory to the folder by entering the following command in the terminal window:

  cd aws-marketplace-serverless-saas-integration

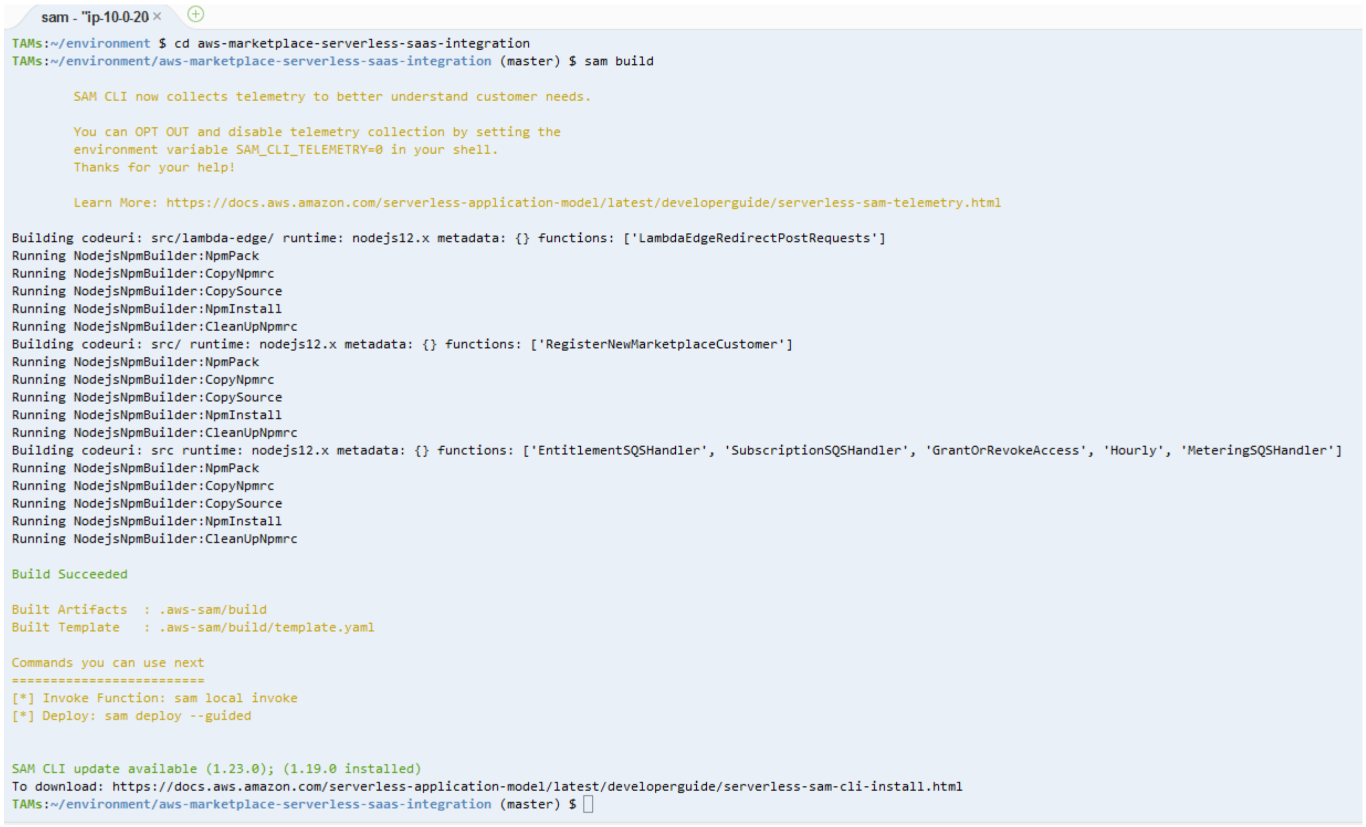
  ```
  TAMs:~/environment $
  ```

-



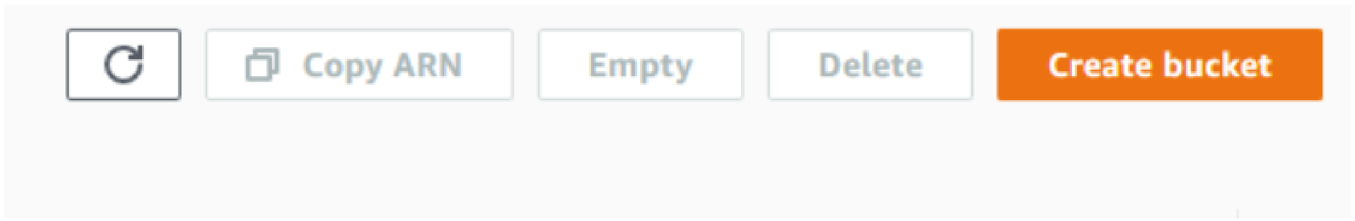
```
TAMs:~/environment $ cd aws-marketplace-serverless-saas-integration
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $
```

- Run the command ' *sam build* 'to build your application from the "template.yaml" file.



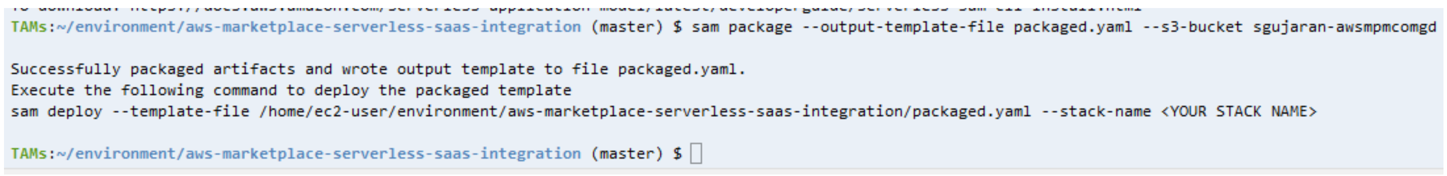
```
TAMs:~/environment $ cd aws-marketplace-serverless-saas-integration
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $ sam build

        SAM CLI now collects telemetry to better understand customer needs.

        You can OPT OUT and disable telemetry collection by setting the
        environment variable SAM_CLI_TELEMETRY=0 in your shell.
        Thanks for your help!

        Learn More: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-telemetry.html

Building codeuri: src/lambda-edge/ runtime: nodejs12.x metadata: {} functions: ['LambdaEdgeRedirectPostRequests']
Running NodejsNpmBuilder:NpmPack
Running NodejsNpmBuilder:CopyNpmrc
Running NodejsNpmBuilder:CopySource
Running NodejsNpmBuilder:NpmInstall
Running NodejsNpmBuilder:CleanUpNpmrc
Building codeuri: src/ runtime: nodejs12.x metadata: {} functions: ['RegisterNewMarketplaceCustomer']
Running NodejsNpmBuilder:NpmPack
Running NodejsNpmBuilder:CopyNpmrc
Running NodejsNpmBuilder:CopySource
Running NodejsNpmBuilder:NpmInstall
Running NodejsNpmBuilder:CleanUpNpmrc
Building codeuri: src runtime: nodejs12.x metadata: {} functions: ['EntitlementSQSHandler', 'SubscriptionSQSHandler', 'GrantOrRevokeAccess', 'Hourly', 'MeteringSQSHandler']
Running NodejsNpmBuilder:NpmPack
Running NodejsNpmBuilder:CopyNpmrc
Running NodejsNpmBuilder:CopySource
Running NodejsNpmBuilder:NpmInstall
Running NodejsNpmBuilder:CleanUpNpmrc

Build Succeeded

Built Artifacts  : .aws-sam/build
Built Template   : .aws-sam/build/template.yaml

Commands you can use next
=========================
[*] Invoke Function: sam local invoke
[*] Deploy: sam deploy --guided


SAM CLI update available (1.23.0); (1.19.0 installed)
To download: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install.html
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $
```

- Once your application is built, you need an S3 bucket to package and deploy it. Go to the S3 console to create a bucket: https://s3.console.aws.amazon.com/s3/home?region=us-east-1 . Choose the "Create bucket" option and use the default settings to create a new S3 bucket.



- Run the following command to package the application and upload it to the newly created S3 bucket. Replace the **placeholder value** with your S3 bucket name.

sam package --output-template-file packaged.yaml --s3-bucket <DEPLOYMENT_S3BUCKET_PLACEHOLDER_VALUE>



```
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $ sam package --output-template-file packaged.yaml --s3-bucket sgujaran-awsmpmcomgd

Successfully packaged artifacts and wrote output template to file packaged.yaml.
Execute the following command to deploy the packaged template
sam deploy --template-file /home/ec2-user/environment/aws-marketplace-serverless-saas-integration/packaged.yaml --stack-name <YOUR STACK NAME>

TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $
```

- To deploy the application, replace the placeholder values in the following command with the values relevant to your SaaS listing. It may take about 10 minutes for the stack to be created. You will see a Success message on completion.

sam deploy --template-file packaged.yaml --stack-name <STACK_NAME> --capabilities CAPABILITY_IAM --region us-east-1 --parameter-overrides ParameterKey=WebsiteS3BucketName,ParameterValue=<WEBSITE_BUCKET_NAME> ParameterKey=ProductCode,ParameterValue=<MARKETPLACE_PRODUCT_CODE> ParameterKey=EntitlementSNSTopic,ParameterValue=<MARKETPLACE_ENTITLEMENT_SNS_TOPIC> ParameterKey=SubscriptionSNSTopic,ParameterValue=<MARKETPLACE_SUBSCRIPTION_SNS_TOPIC> ParameterKey=MarketplaceTechAdminEmail,ParameterValue=<MARKETPLACE_TECH_ADMIN_EMAIL>

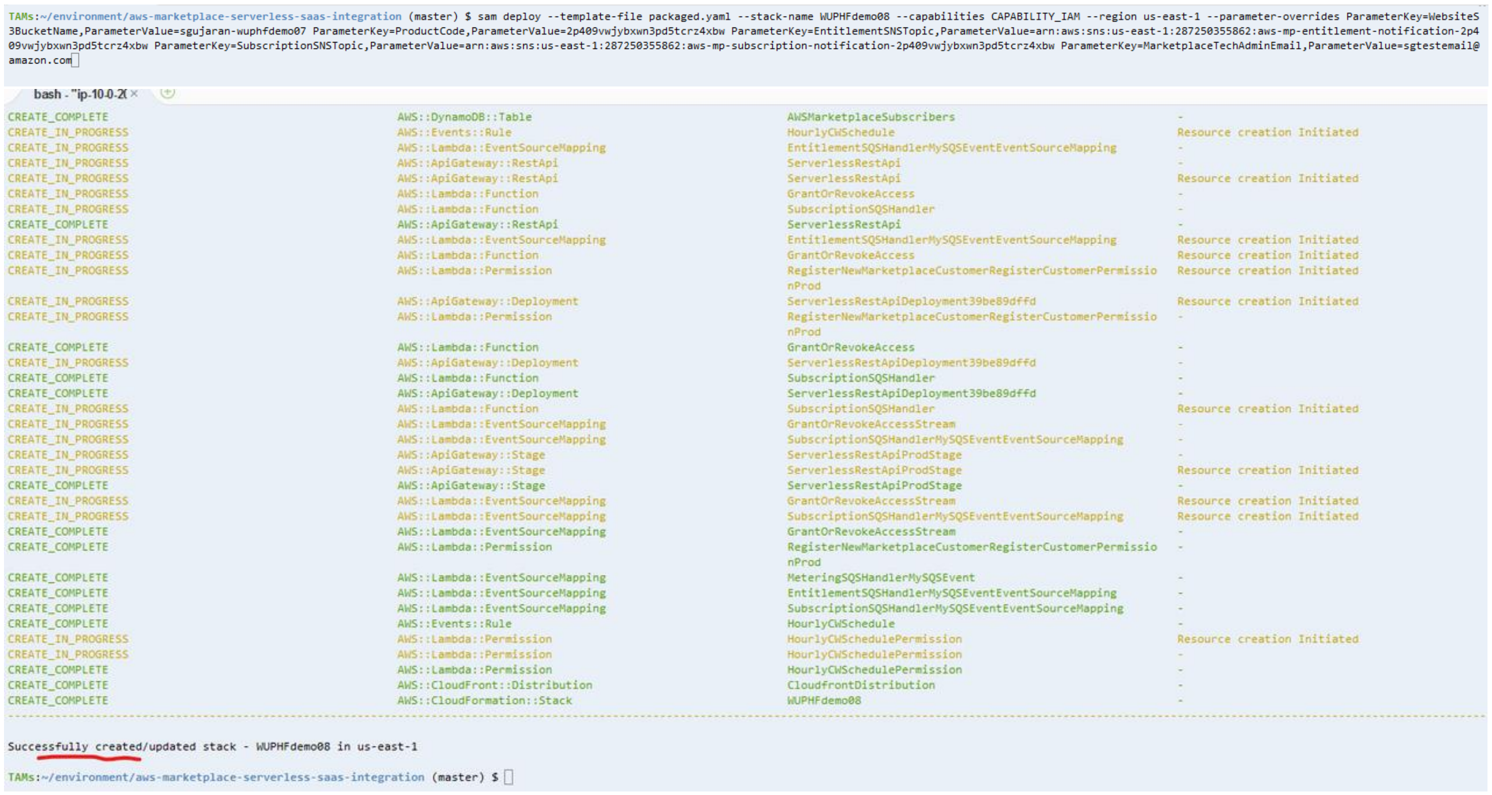
Note1:*The green placeholder value are user-defined. Choose a unique name for stack and website S3 bucket. The Marketplace Tech Admin email will be the email you want the notifications to be sent. The red placeholder values are provided by the AWS Marketplace Seller Operations team.*

Note1: *This sample command is for a SaaS Contract with consumption pricing model. It creates a static landing page and is the default option. You can customize the input parameters based on your pricing model and requirements. A list of all input parameter options is available at end of this document. The command should be executed in one line.*

Note3**:** The WebsiteS3BucketName should be unique and there should not be an existing bucket with the same name. The Website S3 bucket is different than the Deployment S3 Bucket you created earlier for packaging the application. The Website S3 bucket is created automatically when you launch the stack.

Note4**:** Stack names are unique. If your stack fails, make sure to update the stack name before you launch a new one.

Example:
sam deploy --template-file packaged.yaml --stack-name WUPHFdemo08 --capabilities CAPABILITY_IAM --region us-east-1 --parameter-overrides ParameterKey=WebsiteS3BucketName,ParameterValue=sgujaran-wuphfdemo07 ParameterKey=ProductCode,ParameterValue=2p409vwjybxwn3pd5tcrz4xbw ParameterKey=EntitlementSNSTopic,ParameterValue=arn:aws:sns:us-east-1:287250355862:aws-mp-entitlement-notification-2p409vwjybxwn3pd5tcrz4xbw ParameterKey=SubscriptionSNSTopic,ParameterValue=arn:aws:sns:us-east-1:287250355862:aws-mp-subscription-notification-2p409vwjybxwn3pd5tcrz4xbw ParameterKey=MarketplaceTechAdminEmail,ParameterValue=sgtestemail@amazon.com

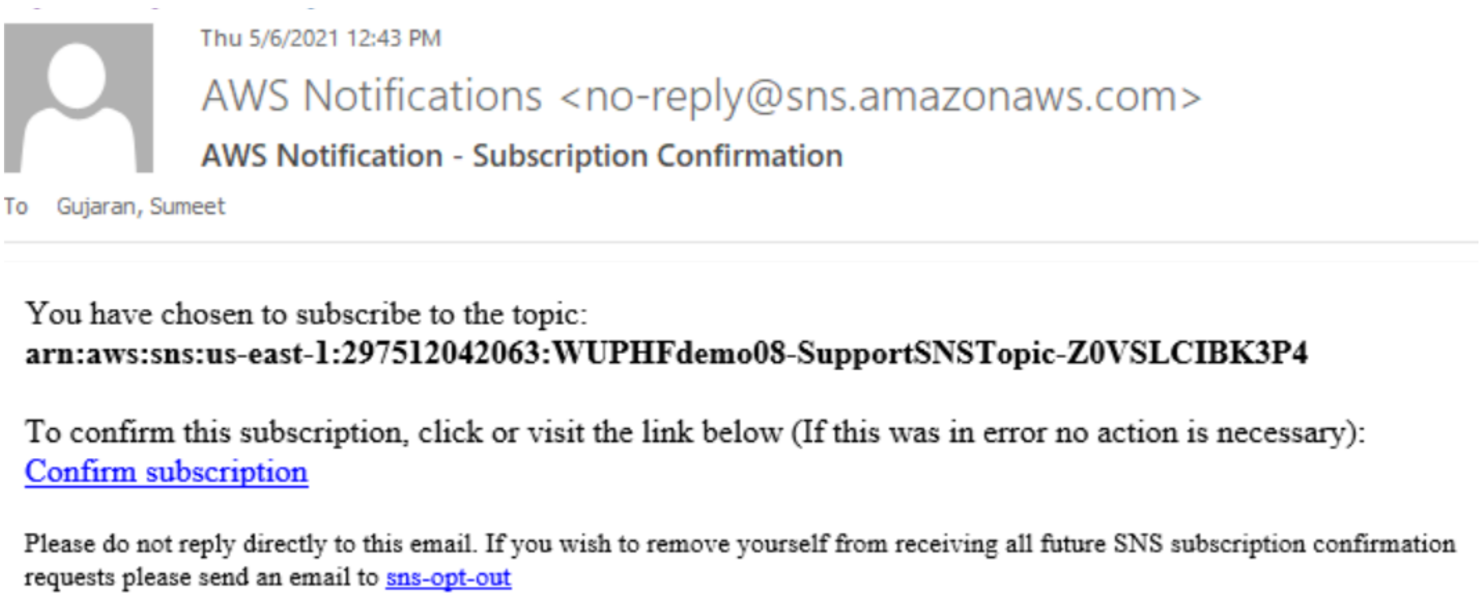


```
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $ sam deploy --template-file packaged.yaml --stack-name WUPHFdemo08 --capabilities CAPABILITY_IAM --region us-east-1 --parameter-overrides ParameterKey=WebsiteS
3BucketName,ParameterValue=sgujaran-wuphfdemo07 ParameterKey=ProductCode,ParameterValue=2p409vwjybxwn3pd5tcrz4xbw ParameterKey=EntitlementSNSTopic,ParameterValue=arn:aws:sns:us-east-1:287250355862:aws-mp-entitlement-notification-2p4
09vwjybxwn3pd5tcrz4xbw ParameterKey=SubscriptionSNSTopic,ParameterValue=arn:aws:sns:us-east-1:287250355862:aws-mp-subscription-notification-2p409vwjybxwn3pd5tcrz4xbw ParameterKey=MarketplaceTechAdminEmail,ParameterValue=sgtestemail@
amazon.com
```

bash - "ip-10-0-2(

```
CREATE_COMPLETE                  AWS::DynamoDB::Table                   AWSMarketplaceSubscribers                               -
CREATE_IN_PROGRESS               AWS::Events::Rule                      HourlyCWSchedule                                        Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::EventSourceMapping        EntitlementSQSHandlerMySQSEventEventSourceMapping       -
CREATE_IN_PROGRESS               AWS::ApiGateway::RestApi               ServerlessRestApi                                       -
CREATE_IN_PROGRESS               AWS::ApiGateway::RestApi               ServerlessRestApi                                       Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::Function                  GrantOrRevokeAccess                                     -
CREATE_IN_PROGRESS               AWS::Lambda::Function                  SubscriptionSQSHandler                                  -
CREATE_COMPLETE                  AWS::ApiGateway::RestApi               ServerlessRestApi                                       -
CREATE_IN_PROGRESS               AWS::Lambda::EventSourceMapping        EntitlementSQSHandlerMySQSEventEventSourceMapping       Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::Function                  GrantOrRevokeAccess                                     Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::Permission                RegisterNewMarketplaceCustomerRegisterCustomerPermissio Resource creation Initiated
                                                                        nProd
CREATE_IN_PROGRESS               AWS::ApiGateway::Deployment            ServerlessRestApiDeployment39be89dffd                   Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::Permission                RegisterNewMarketplaceCustomerRegisterCustomerPermissio -
                                                                        nProd
CREATE_COMPLETE                  AWS::Lambda::Function                  GrantOrRevokeAccess                                     -
CREATE_IN_PROGRESS               AWS::ApiGateway::Deployment            ServerlessRestApiDeployment39be89dffd                   -
CREATE_COMPLETE                  AWS::Lambda::Function                  SubscriptionSQSHandler                                  -
CREATE_COMPLETE                  AWS::ApiGateway::Deployment            ServerlessRestApiDeployment39be89dffd                   -
CREATE_IN_PROGRESS               AWS::Lambda::Function                  SubscriptionSQSHandler                                  Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::EventSourceMapping        GrantOrRevokeAccessStream                               -
CREATE_IN_PROGRESS               AWS::Lambda::EventSourceMapping        SubscriptionSQSHandlerMySQSEventEventSourceMapping      -
CREATE_IN_PROGRESS               AWS::ApiGateway::Stage                 ServerlessRestApiProdStage                              -
CREATE_IN_PROGRESS               AWS::ApiGateway::Stage                 ServerlessRestApiProdStage                              Resource creation Initiated
CREATE_COMPLETE                  AWS::ApiGateway::Stage                 ServerlessRestApiProdStage                              -
CREATE_IN_PROGRESS               AWS::Lambda::EventSourceMapping        GrantOrRevokeAccessStream                               Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::EventSourceMapping        SubscriptionSQSHandlerMySQSEventEventSourceMapping      Resource creation Initiated
CREATE_COMPLETE                  AWS::Lambda::EventSourceMapping        GrantOrRevokeAccessStream                               -
CREATE_COMPLETE                  AWS::Lambda::Permission                RegisterNewMarketplaceCustomerRegisterCustomerPermissio -
                                                                        nProd
CREATE_COMPLETE                  AWS::Lambda::EventSourceMapping        MeteringSQSHandlerMySQSEvent                            -
CREATE_COMPLETE                  AWS::Lambda::EventSourceMapping        EntitlementSQSHandlerMySQSEventEventSourceMapping       -
CREATE_COMPLETE                  AWS::Lambda::EventSourceMapping        SubscriptionSQSHandlerMySQSEventEventSourceMapping      -
CREATE_COMPLETE                  AWS::Events::Rule                      HourlyCWSchedule                                        -
CREATE_IN_PROGRESS               AWS::Lambda::Permission                HourlyCWSchedulePermission                              Resource creation Initiated
CREATE_IN_PROGRESS               AWS::Lambda::Permission                HourlyCWSchedulePermission                              -
CREATE_COMPLETE                  AWS::Lambda::Permission                HourlyCWSchedulePermission                              -
CREATE_COMPLETE                  AWS::CloudFront::Distribution          CloudfrontDistribution                                  -
CREATE_COMPLETE                  AWS::CloudFormation::Stack             WUPHFdemo08                                             -
-----------------------------------------------------------------------------------------------------------------------------------------------------

Successfully created/updated stack - WUPHFdemo08 in us-east-1

TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $
```

- The MarketplaceTechAdmin email address you provided above will receive an email notification to confirm subscription to the Support SNS topic. This SNS topic will send out information on new subscribers and subscription/entitlement updates. Confirm your subscription to the SNS topic.



Thu 5/6/2021 12:43 PM

AWS Notifications <no-reply@sns.amazonaws.com>

AWS Notification - Subscription Confirmation

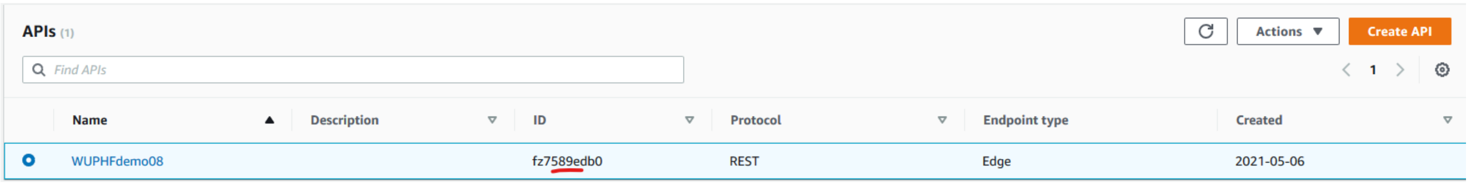
To Gujaran, Sumeet

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:297512042063:WUPHFdemo08-SupportSNSTopic-Z0VSLCIBK3P4

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out

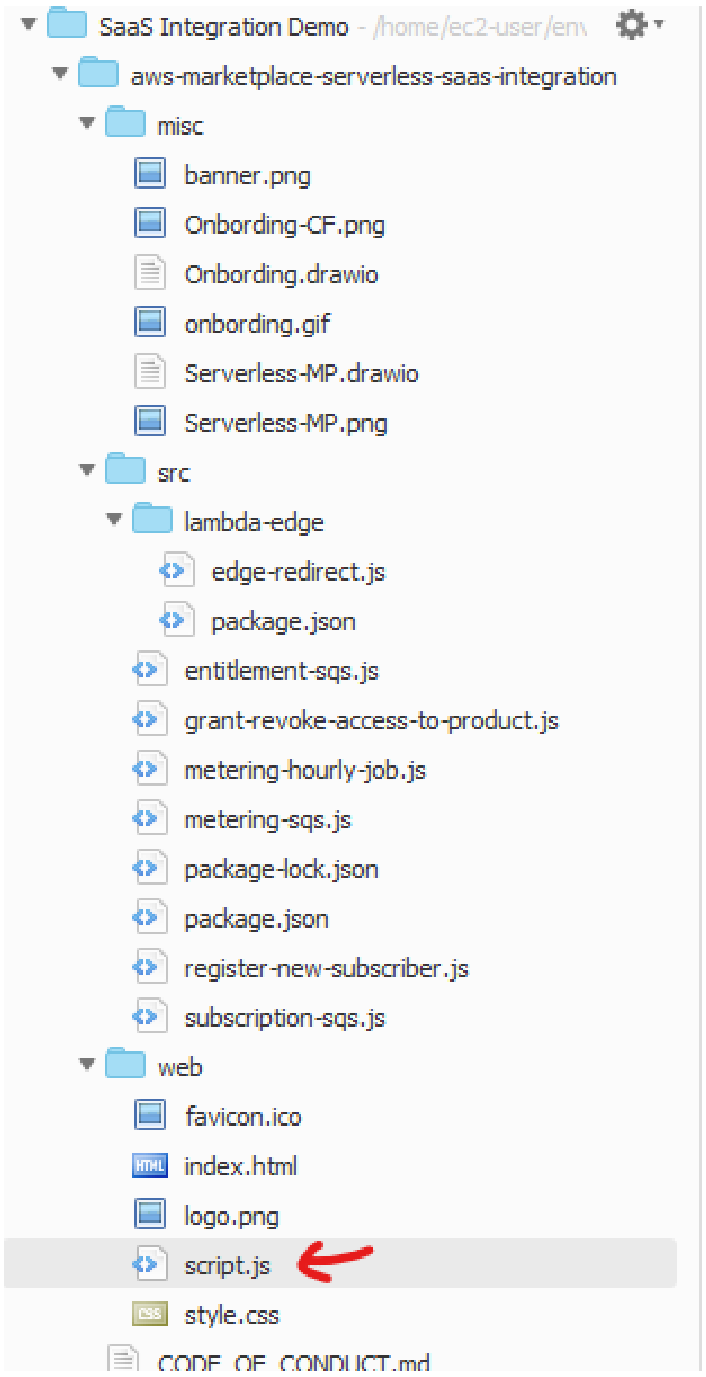
- Once the stack has been created, you will proceed to set up your landing page. Obtain the API gateway endpoint that was created by the stack. Go to: https://console.aws.amazon.com/apigateway/main/apis?region=us-east-1 and copy the API Gateway ID.



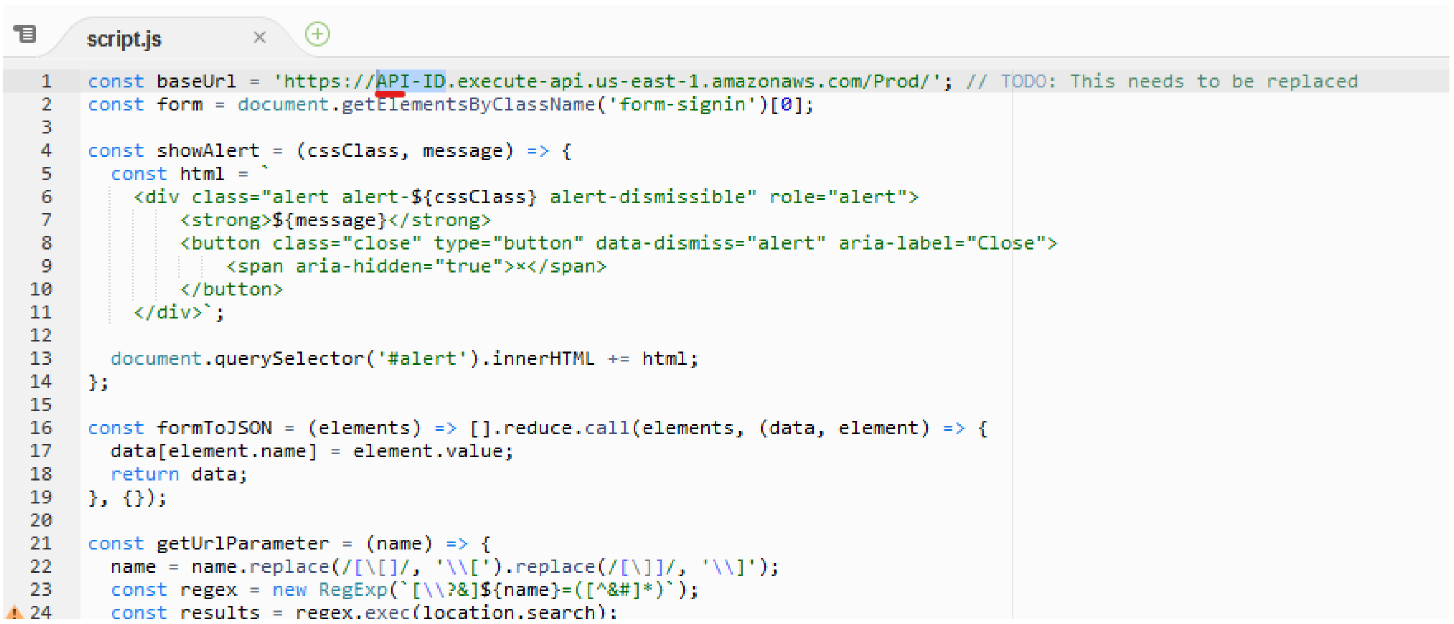
APIs (1)

	Name	Description	ID	Protocol	Endpoint type	Created
●	WUPHFdemo08		fz7589edb0	REST	Edge	2021-05-06

- Replace the ***API-ID*** for the baseURL in Line 1 of the "script.js" file (landing page) in the "web" folder of the repository. Save the changes to the script.js file by using the File tab → Save or CTRL+S keys.



```
SaaS Integration Demo - /home/ec2-user/env
  aws-marketplace-serverless-saas-integration
    misc
      banner.png
      Onbording-CF.png
      Onbording.drawio
      onbording.gif
      Serverless-MP.drawio
      Serverless-MP.png
    src
      lambda-edge
        edge-redirect.js
        package.json
      entitlement-sqs.js
      grant-revoke-access-to-product.js
      metering-hourly-job.js
      metering-sqs.js
      package-lock.json
      package.json
      register-new-subscriber.js
      subscription-sqs.js
    web
      favicon.ico
      index.html
      logo.png
      script.js
      style.css
    CODE_OF_CONDUCT.md
```

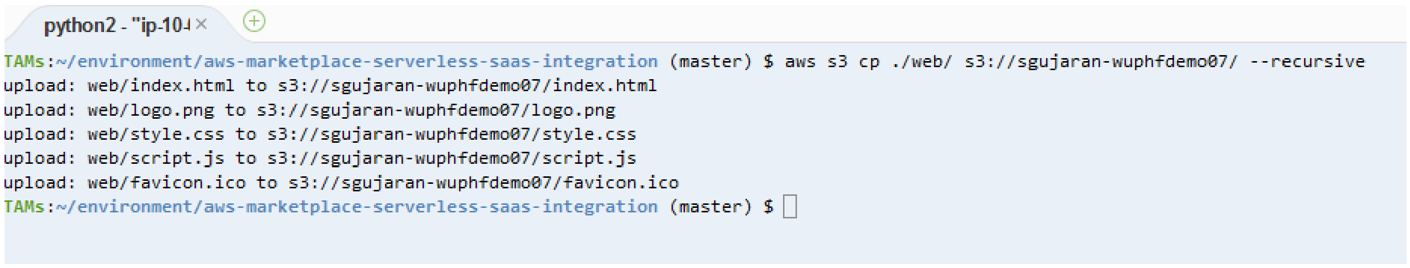


```
const baseUrl = 'https://API-ID.execute-api.us-east-1.amazonaws.com/Prod/'; // TODO: This needs to be replaced
const form = document.getElementsByClassName('form-signin')[0];

const showAlert = (cssClass, message) => {
  const html = `
    <div class="alert alert-${cssClass} alert-dismissible" role="alert">
        <strong>${message}</strong>
        <button class="close" type="button" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">×</span>
        </button>
    </div>`;

  document.querySelector('#alert').innerHTML += html;
};

const formToJSON = (elements) => [].reduce.call(elements, (data, element) => {
  data[element.name] = element.value;
  return data;
}, {});

const getUrlParameter = (name) => {
  name = name.replace(/[\[]/, '\\[').replace(/[\]]/, '\\]');
  const regex = new RegExp(`[\\?&]${name}=([^&#]*)`);
  const results = regex.exec(location.search);
```

- Copy the landing page files to the website S3 bucket used by the stack using the following command:

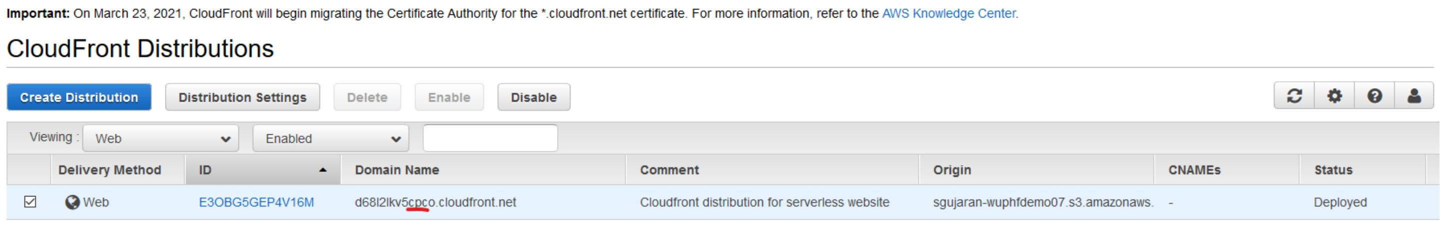
  aws s3 cp ./web/ s3://<WEBSITE_BUCKET_NAME>/ --recursive

Replace the placeholder value with the website S3 bucket name you provided during stack creation.



```
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $ aws s3 cp ./web/ s3://sgujaran-wuphfdemo07/ --recursive
upload: web/index.html to s3://sgujaran-wuphfdemo07/index.html
upload: web/logo.png to s3://sgujaran-wuphfdemo07/logo.png
upload: web/style.css to s3://sgujaran-wuphfdemo07/style.css
upload: web/script.js to s3://sgujaran-wuphfdemo07/script.js
upload: web/favicon.ico to s3://sgujaran-wuphfdemo07/favicon.ico
TAMs:~/environment/aws-marketplace-serverless-saas-integration (master) $
```

- Your integration is now ready for use. Since we use Cloudfront distribution to ensure low latency for the landing page, the SaaS fulfilment URL is the Cloudfront CName (domain name) of the distribution created by the stack.
- Go to the Cloudfront console and get the domain name here: https://console.aws.amazon.com/cloudfront/home?region=us-east-1 .
- Provide the domain name to the AWS Marketplace Seller Operations team to publish to your limited listing.



Important: On March 23, 2021, CloudFront will begin migrating the Certificate Authority for the *.cloudfront.net certificate. For more information, refer to the AWS Knowledge Center.

CloudFront Distributions

	Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status
☑	Web	E3OBG5GEP4V16M	d68l2lkv5cpco.cloudfront.net	Cloudfront distribution for serverless website	sgujaran-wuphfdemo07.s3.amazonaws.	-	Deployed

SaaS URL format: https://<domain name>
Example: https://d142rocbcrghws.cloudfront.net

Grant and revoke access to your product:

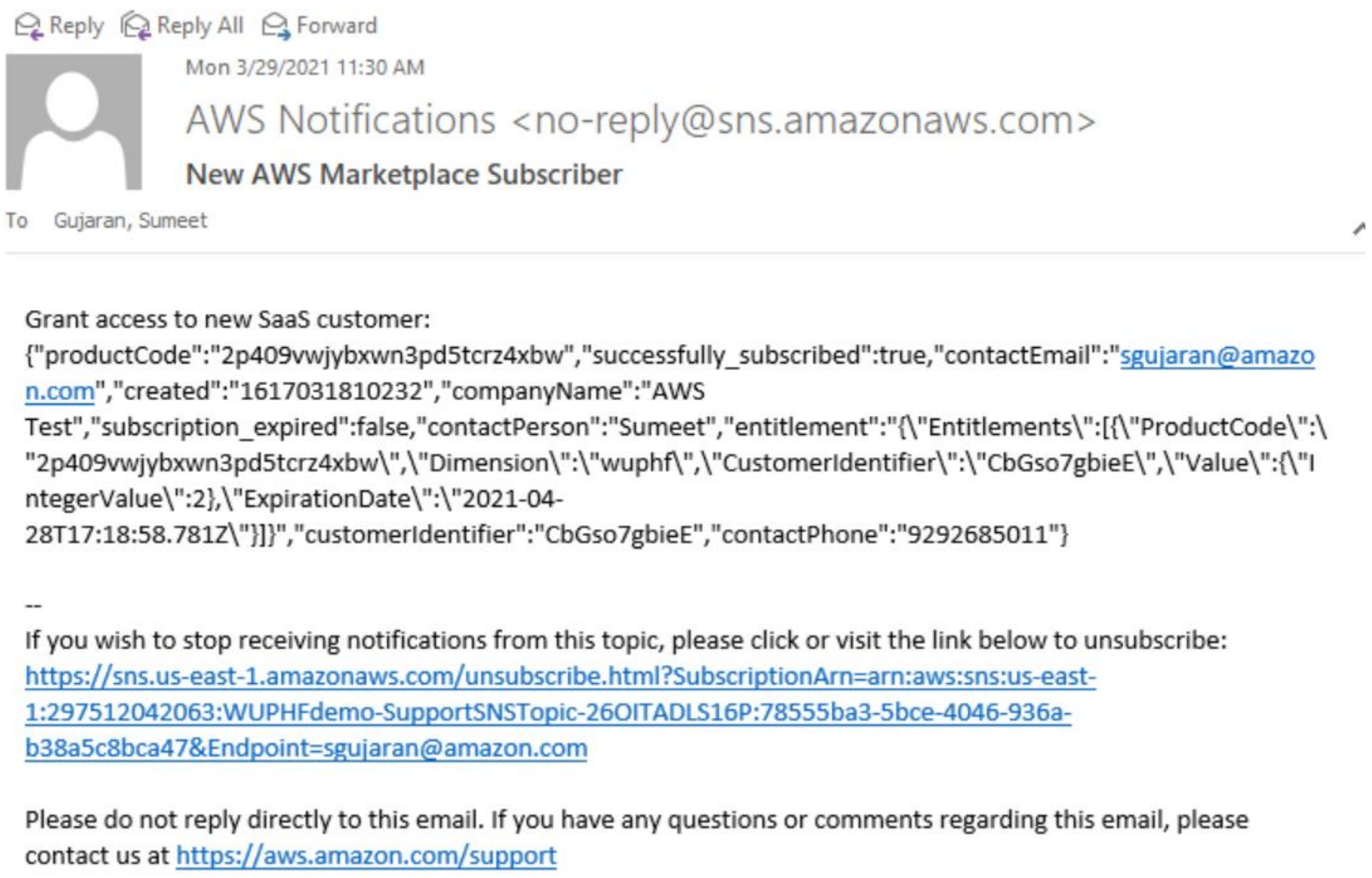
Grant access to new subscribers:

Once the resolveCustomer endpoint returns a successful response, the SaaS vendor must provide the new subscriber with access to the solution. Depending on the SaaS pricing model, we have defined different conditions in the grant-revoke-access-to-product.js stream handler that is executed on adding new or updating existing rows in AWSMarketplaceSubscribers DynamoDB table.

When a new environment needs to be provisioned or an existing environment needs to be updated, the MarketplaceTechAdmin email address (The email address you provided during deployment) will receive an email notification.

When successful response is received for the GetEntitlementAPI call for SaaS Contract pricing model or after receiving subscribe-success message from the Subscription SNS Topic for SaaS subscriptions pricing model in the subscription-sqs-handler.js., the condition “success” is met which will execute the grant-revoke-access-to-product.js to add or update rows in the DynamoDB table.

Sample New Subscriber notification:



Mon 3/29/2021 11:30 AM

AWS Notifications <no-reply@sns.amazonaws.com>

New AWS Marketplace Subscriber

To Gujaran, Sumeet

Grant access to new SaaS customer:
{"productCode":"2p409vwjybxwn3pd5tcrz4xbw","successfully_subscribed":true,"contactEmail":"sgujaran@amazon.com","created":"1617031810232","companyName":"AWS Test","subscription_expired":false,"contactPerson":"Sumeet","entitlement":"{\"Entitlements\":[{\"ProductCode\":\"2p409vwjybxwn3pd5tcrz4xbw\",\"Dimension\":\"wuphf\",\"CustomerIdentifier\":\"CbGso7gbieE\",\"Value\":{\"IntegerValue\":2},\"ExpirationDate\":\"2021-04-28T17:18:58.781Z\"}]}","customerIdentifier":"CbGso7gbieE","contactPhone":"9292685011"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:297512042063:WUPHFdemo-SupportSNSTopic-26OITADLS16P:78555ba3-5bce-4046-936a-b38a5c8bca47&Endpoint=sgujaran@amazon.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Managed SaaS Onboarding:

If you provide a managed onboarding experience, you should update the form submission 'Thank you' page using register-new-subscriber.js as per your onboarding next steps. The file is in the '*src'* folder.

Default Success message: *'Thank you for registering. Please check your email for a confirmation!'*
Default Error message: *'Registration data not valid. Please try again, or contact support!'*

Note: Buyers do not receive an email notification of landing page form submission. The email should be sent by you to initiate the onboarding process.

Automated SaaS application provisioning:

To provide automated access to your application on signup, you can redirect the user from the default landing page to your application page, or you can use your own registration landing page.

To use your existing SaaS registration page, you should call the register new subscriber endpoint after collecting the data.
The registration landing page should identify and accept the *x-amzn-marketplace-token* token in the form data from AWS Marketplace with the customer's identifier for billing. It should then pass that token value to the AWS Marketplace Metering Service and AWS Marketplace Entitlement Service APIs to resolve for the unique customer identifier and corresponding product code.

In this solution, we created a CloudFront Distribution, which can be configured to use domain/CNAME by your choice. The POST request coming from AWS Marketplace is intercepted by the Edge src/lambda-edge/edge-redirect.js file, which transforms the POST request to GET request, and passes the *x-amzn-marketplace-token* in the query string.

We have created a static HTML landing page hosted on S3 which takes the user inputs collected by the HTML form and submits them to marketplace/customer endpoint. The handler for the marketplace/customer endpoint is defined in the src/register-new-subscriber.js file, where we call the resolveCustomer and validate the token. If the token is valid the customer record is created in the AWSMarketplaceSubscribers DynamoDB table and the new customer data is stored.

Update entitlement levels to new subscribers (SaaS Contracts and CCP only):

Each time the entitlement is updated, we receive a message on the SNS topic. The lambda function entitlement-sqs.js on each message is calling the marketplaceEntitlementService and storing the response in the AWSMarketplaceSubscribers dynamoDB.
We are using the same DynamoDB stream to detect changes in the entitlement for SaaS contracts. When the entitlement is updated, an email notification is sent to the MarketplaceTechAdmin.

Revoke access to customers with expired contracts and cancelled subscriptions:

The revoke access logic is implemented in a similar manner as the grant access logic. When the contract expires or the subscription is cancelled, the MarketplaceTechAdmin email address will receive an email notification.

AWS Marketplace strongly recommends automating the access and environment management which can be achieved by modifying the grant-revoke-access-to-product.js function.

Reporting Usage:

For SaaS subscriptions, the SaaS provider must meter for all usage, and then, customers are billed by AWS based on the metering records provided. For SaaS contract with consumption, you only meter for usage beyond a customer's contract entitlements. When your application meters usage for a customer, your application is providing AWS with a quantity of usage accrued. Your application meters for the pricing dimensions that you defined when you created your product, such as gigabytes transferred or hosts scanned in a given hour.

We have created MeteringSchedule CloudWatch Event rule that is **triggered every hour**. The metering-hourly-job.js gets triggered by this rule and it's querying all of the pending/unreported metering records from the AWSMarketplaceMeteringRecords table using the PendingMeteringRecordsIndex. All of the pending records are aggregated based on the customerIdentifier and dimension name, and sent to the SQSMetering queue.

For SaaS subscription and SaaS Contract with consumption pricing model, the records in the AWSMarketplaceMeteringRecords table are expected to be inserted programmatically by your SaaS application. In this case you will have to give permissions to the service in charge of collecting usage data in your existing SaaS product to be able to write to AWSMarketplaceMeteringRecords table.

The lambda function metering-sqs.js is sending all of the queued metering records to the AWS Marketplace Metering Service. After every call to the batchMeterUsage endpoint the rows are updated in the AWSMarketplaceMeteringRecords table, with the response returned from the Metering Service, which can be found in the metering_response field. If the request was unsuccessful the metering_failed value with be set to true and you will have to investigate the issue the error will be also stored in the metering_response field.

The new records in the AWSMarketplaceMeteringRecords table should be stored in the following format:

```
{
"create_timestamp": 113123,
"customerIdentifier": "ifAPi5AcF3",
"dimension_usage": [
{
"dimension": "users",
"value": 3
},
{
"dimension": "admin_users",
"value": 1
}
],
"metering_pending": "true"
}
```

The create_timestamp is the sort key and customerIdentifier is the partition key, and they are both forming the Primary key in the AWSMarketplaceMeteringRecords table.
After the metering record is submitted to AWS Marketplace Metering Service, it will be updated and will look like this:

```
{
"create_timestamp": 113123,
"customerIdentifier": "ifAPi5AcF3",
"dimension_usage": [
{
"dimension": "admin_users",
"value": 3
}
],
"metering_failed": false,
"metering_response": "{\"Results\":[{\"UsageRecord\":{\"Timestamp\":\"2020-06-24T04:04:53.776Z\",\"CustomerIdentifier\":\"ifAPi5AcF3\",\"Dimension\":\"admin_users\",\"Quantity\":3},\"MeteringRecordId\":\"35155d37-56cb-423f-8554-5c4f3e3ff56d\",\"Status\":\"Success\"}],\"UnprocessedRecords\":[]}"
}
```

List of parameters:

Parameter name	**Description**
WebsiteS3BucketName	S3 bucket to store the HTML files; Mandatory if CreateRegistrationWebPage is set to true;
NewSubscribersTableName	Use custom name for the New Subscribers Table; Default value: AWSMarketplaceSubscribers
AWSMarketplaceMeteringRecordsTableName	Use custom name for the Metering Records Table; Default value: AWSMarketplaceMeteringRecords
TypeOfSaaSListing	allowed values: contracts_with_subscription, contracts, subscriptions; Default value: contracts_with_subscription
ProductCode	Product code provided from AWS Marketplace
EntitlementSNSTopic	SNS topic ARN provided from AWS Marketplace
SubscriptionSNSTopic	SNS topic ARN provided from AWS Marketplace
CreateRegistrationWebPage	true or false; Default value: true
MarketplaceTechAdminEmail	Email to be notified on changes requiring action