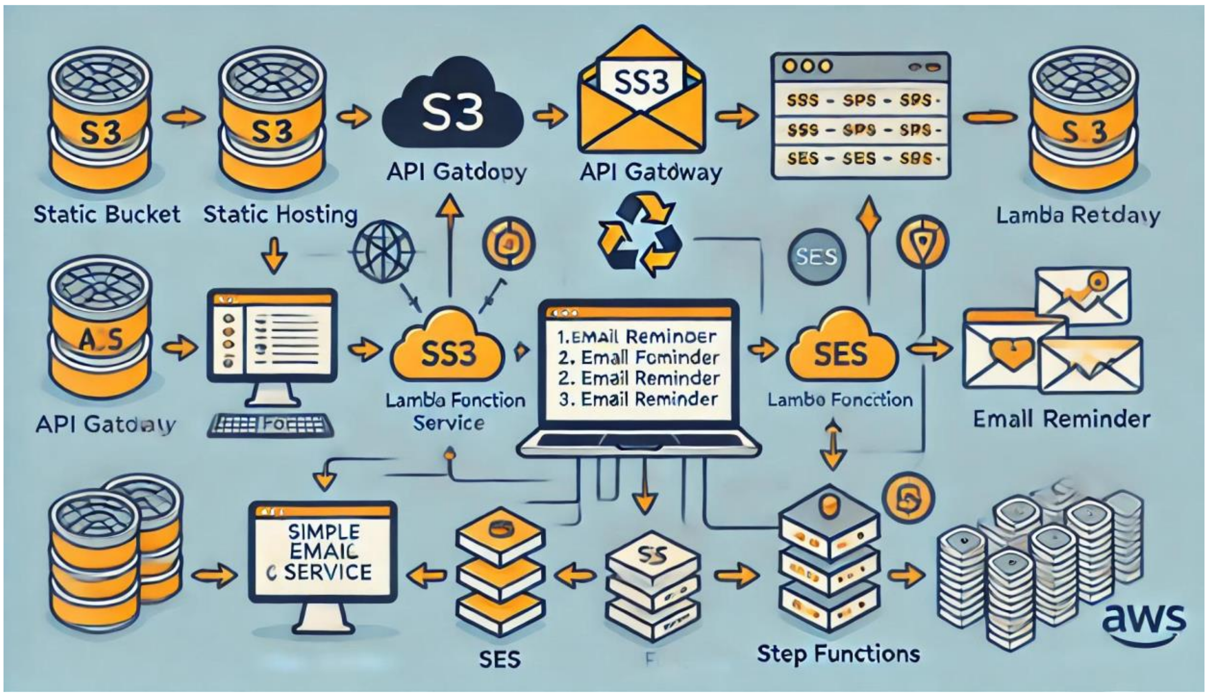


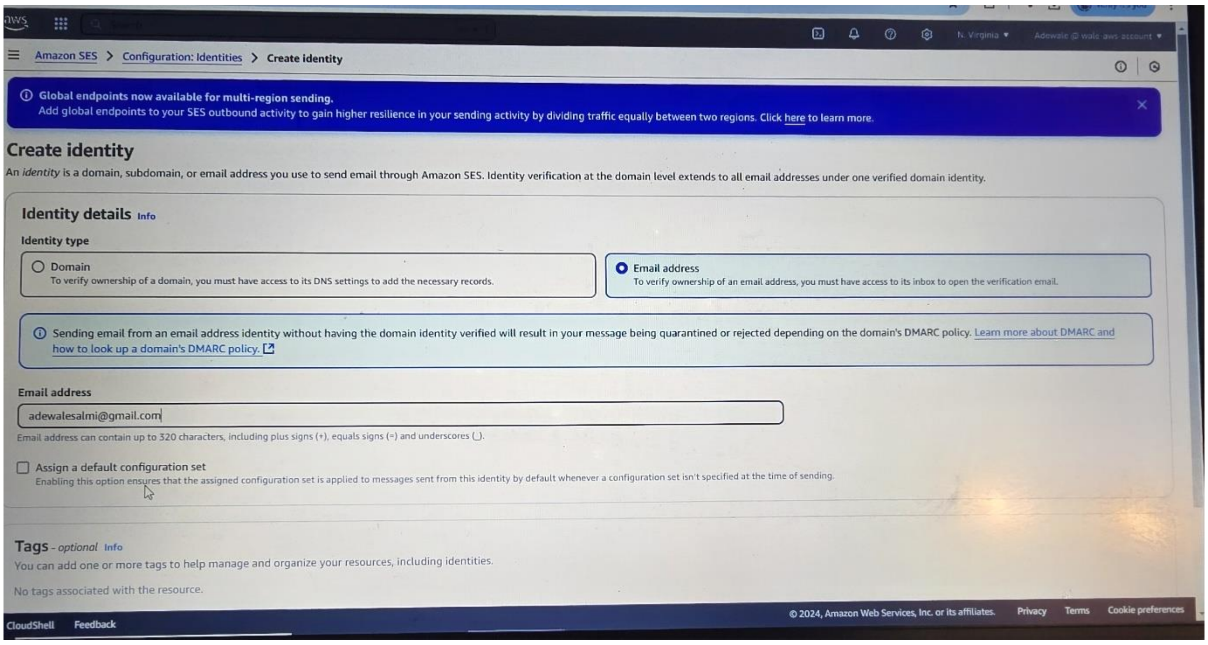
In this project, you will build a serverless reminder application. The application will be hosted in an S3 bucket and executed within a browser, interfacing with Lambda and Step Functions via an API Gateway Endpoint. This app enables users to configure reminders for 'pet cuddles' to be sent via email.

This demo consists of five stages:

STAGE 1: Setting up the Simple Email Service (SES)

Step 1A: Verify the Sender Email Address

1. Ensure you are logged into an AWS account with administrative privileges, located in the us-east-1 region.
2. Navigate to the SES Console and select **Verified Identities** under **Configuration**.
3. Click **Create Identity**, select the **Email Address** checkbox, and enter the email address you wish to use for sending reminders.


Amazon SES > Configuration: Identities > Create identity

Global endpoints now available for multi-region sending.
Add global endpoints to your SES outbound activity to gain higher resilience in your sending activity by dividing traffic equally between two regions. Click here to learn more.

Create identity
An *identity* is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.

Identity details Info
Identity type
Domain — To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.
[✓ Email address] — To verify ownership of an email address, you must have access to its inbox to open the verification email.

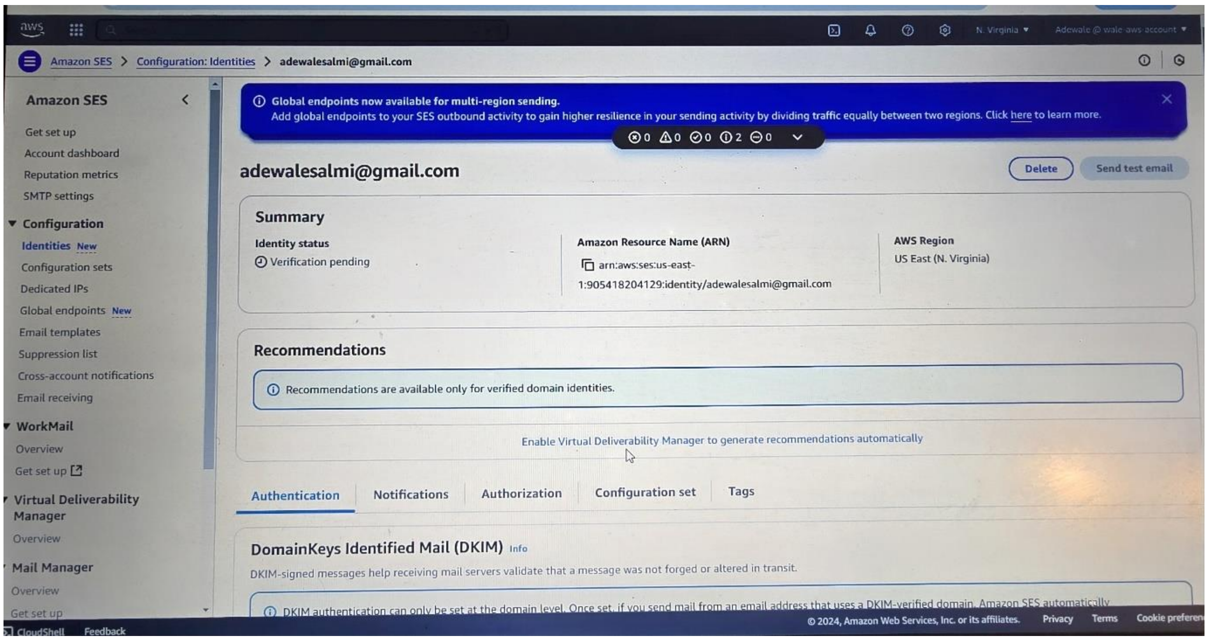
Sending email from an email address identity without having the domain identity verified will result in your message being quarantined or rejected depending on the domain's DMARC policy. Learn more about DMARC and how to look up a domain's DMARC policy.

Email address
adewalesalmi@gmail.com
Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).

Assign a default configuration set
Enabling this option ensures that the assigned configuration set is applied to messages sent from this identity by default whenever a configuration set isn't specified at the time of sending.

Tags - *optional* Info
You can add one or more tags to help manage and organize your resources, including identities.
No tags associated with the resource.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Amazon SES > Configuration: Identities > adewalesalmi@gmail.com

Global endpoints now available for multi-region sending.
Add global endpoints to your SES outbound activity to gain higher resilience in your sending activity by dividing traffic equally between two regions. Click here to learn more.

adewalesalmi@gmail.com

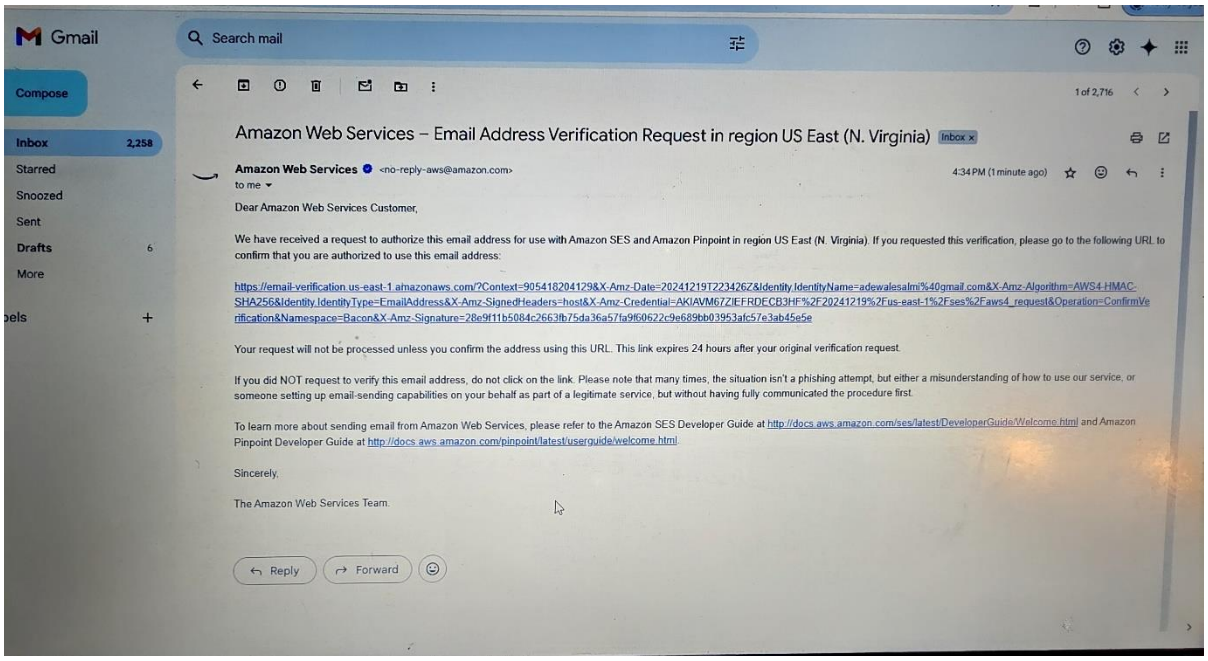
Summary
Identity status: Verification pending
Amazon Resource Name (ARN): arn:aws:ses:us-east-1:905418204129:identity/adewalesalmi@gmail.com
AWS Region: US East (N. Virginia)

Recommendations
Recommendations are available only for verified domain identities.
Enable Virtual Deliverability Manager to generate recommendations automatically

DomainKeys Identified Mail (DKIM) Info
DKIM-signed messages help receiving mail servers validate that a message was not forged or altered in transit.
DKIM authentication can only be set at the domain level. Once set, if you send mail from an email address that uses a DKIM-verified domain, Amazon SES automatically

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

4. Check your email for a verification message, click the confirmation link, and ensure the verification status updates to "verified" in the SES console.


Amazon Web Services – Email Address Verification Request in region US East (N. Virginia)

Amazon Web Services <no-reply-aws@amazon.com>
to me
4:34 PM (1 minute ago)

Dear Amazon Web Services Customer,

We have received a request to authorize this email address for use with Amazon SES and Amazon Pinpoint in region US East (N. Virginia). If you requested this verification, please go to the following URL to confirm that you are authorized to use this email address:

https://email-verification.us-east-1.amazonaws.com/?Context=905418204129&X-Amz-Date=20241219T223426Z&Identity.IdentityName=adewalesalmi%40gmail.com&X-Amz-Algorithm=AWS4-HMAC-SHA256&Identity.IdentityType=EmailAddress&X-Amz-SignedHeaders=host&X-Amz-Credential=AKIAVM67ZIEFRDECB3HF%2F20241219%2Fus-east-1%2Fses%2Faws4_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=28e9f11b5084c2663fb75da36a57fa9f60622c9e689bb03953afc57e3ab45e5e

Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original verification request.

If you did NOT request to verify this email address, do not click on the link. Please note that many times, the situation isn't a phishing attempt, but either a misunderstanding of how to use our service, or someone setting up email-sending capabilities on your behalf as part of a legitimate service, but without having fully communicated the procedure first.

To learn more about sending email from Amazon Web Services, please refer to the Amazon SES Developer Guide at http://docs.aws.amazon.com/ses/latest/DeveloperGuide/Welcome.html and Amazon Pinpoint Developer Guide at http://docs.aws.amazon.com/pinpoint/latest/userguide/welcome.html.

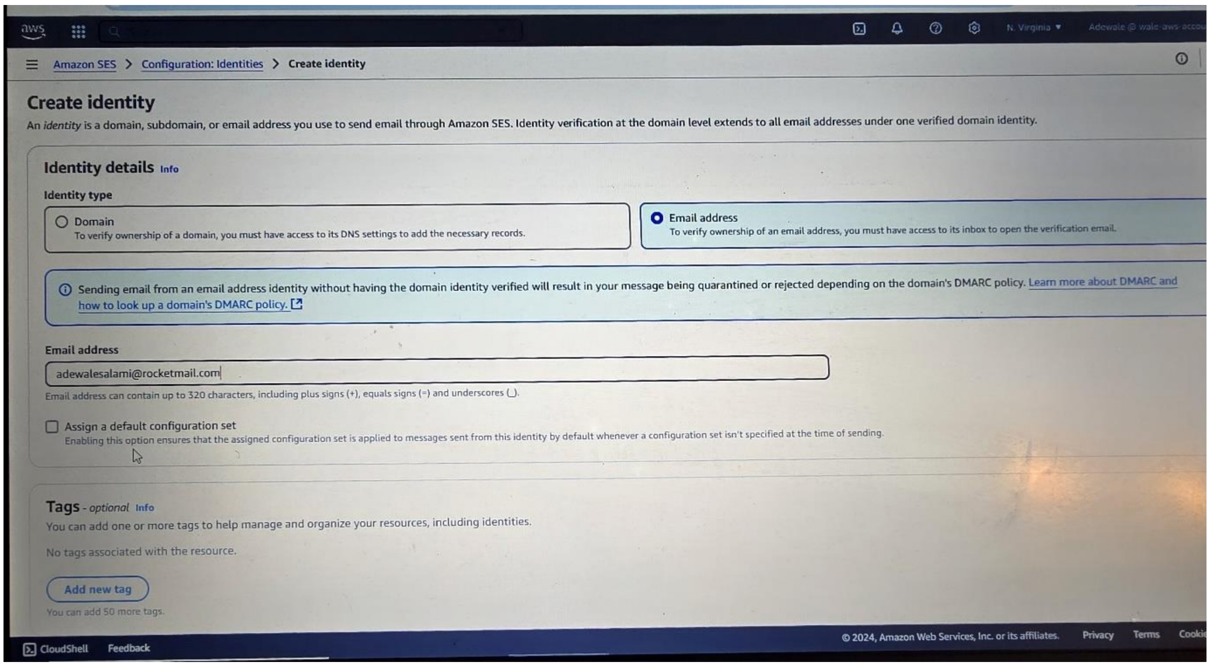
Sincerely,

The Amazon Web Services Team.

5. Record this address as the **Sending Address** for the application.

Step 1B: Verify the Recipient Email Address

1. Repeat the above steps for the email address you wish to use as the recipient.
2. Record this address as the **Customer Address** for the application.


Amazon SES > Configuration: Identities > Create identity

Create identity

An *identity* is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.

Identity details

Identity type

( ) Domain
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.

(●) Email address
To verify ownership of an email address, you must have access to its inbox to open the verification email.

Sending email from an email address identity without having the domain identity verified will result in your message being quarantined or rejected depending on the domain's DMARC policy. Learn more about DMARC and how to look up a domain's DMARC policy.

Email address: adewalesalami@rocketmail.com

Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).

[ ] Assign a default configuration set
Enabling this option ensures that the assigned configuration set is applied to messages sent from this identity by default whenever a configuration set isn't specified at the time of sending.

Tags - *optional*
You can add one or more tags to help manage and organize your resources, including identities.

No tags associated with the resource.

You can add 50 more tags.

© 2024, Amazon Web Services, Inc. or its affiliates.

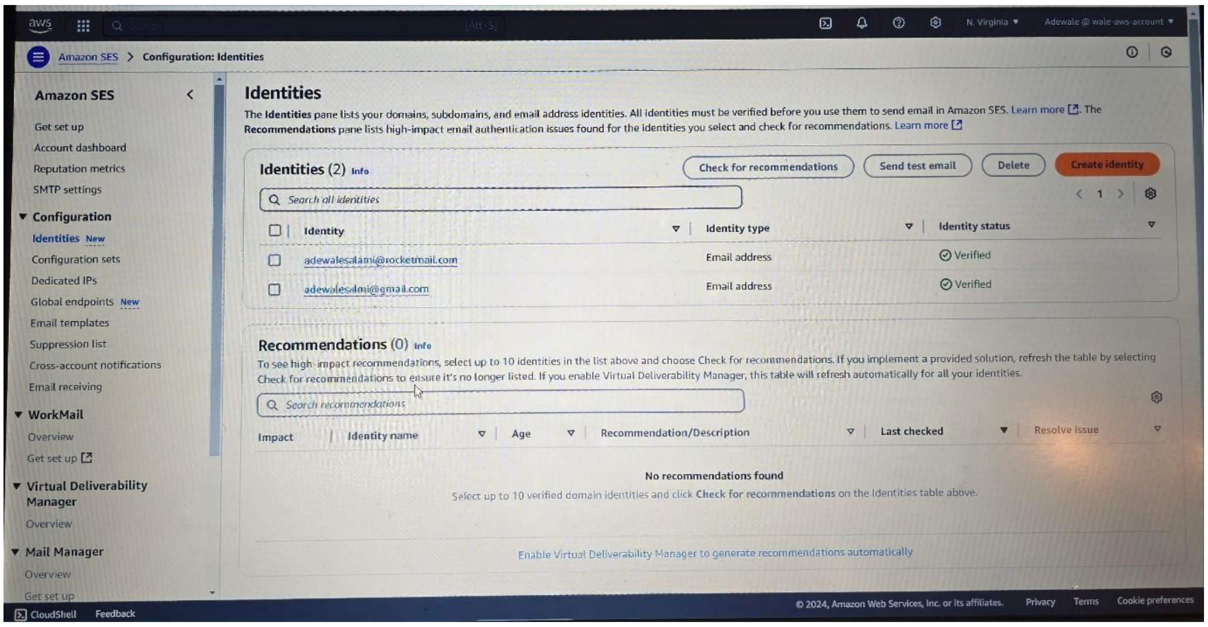
Completion of Stage 1

At this point, two email addresses are whitelisted in SES:

- **PetCuddleOTron Sending Address**

- **PetCuddleOTron Customer Address**

These will be used in subsequent stages of the project.


Amazon SES > Configuration: Identities

Identities

The **Identities** pane lists your domains, subdomains, and email address identities. All identities must be verified before you use them to send email in Amazon SES. Learn more. The **Recommendations** pane lists high-impact email authentication issues found for the identities you select and check for recommendations. Learn more

Identities (2)

Identity	Identity type	Identity status
adewalesalami@rocketmail.com	Email address	Verified
adewalesalmi@gmail.com	Email address	Verified

Recommendations (0)

To see high-impact recommendations, select up to 10 identities in the list above and choose Check for recommendations. If you implement a provided solution, refresh the table by selecting Check for recommendations to ensure it's no longer listed. If you enable Virtual Deliverability Manager, this table will refresh automatically for all your identities.

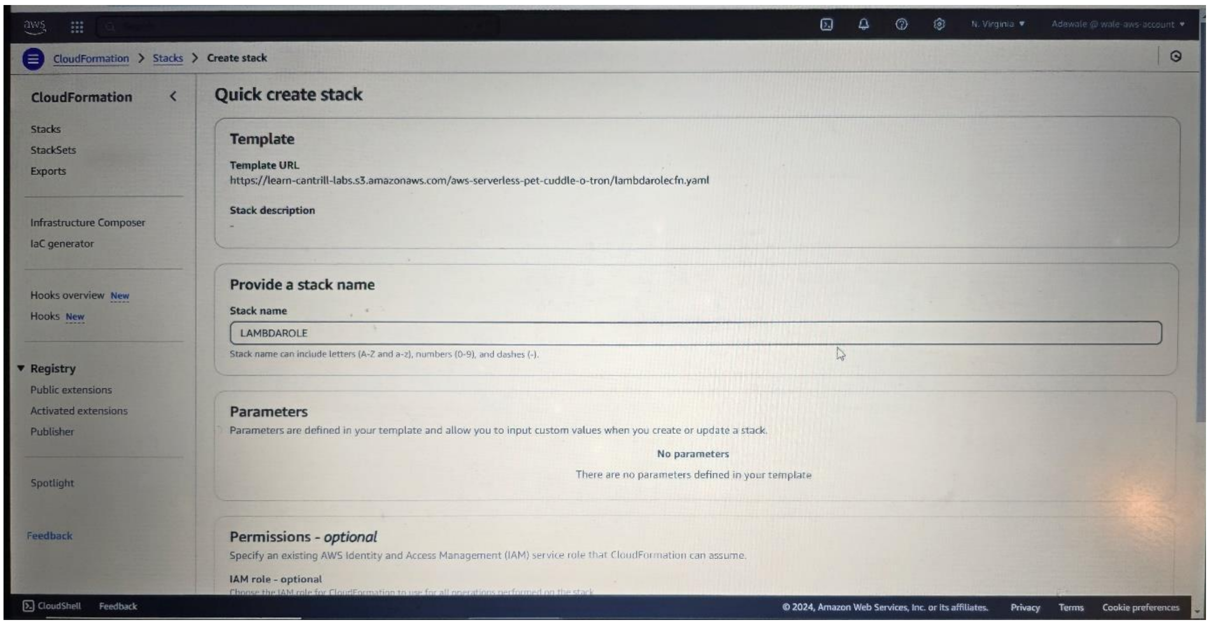
No recommendations found

Select up to 10 verified domain identities and click **Check for recommendations** on the Identities table above.

STAGE 2: Creating the Lambda Email Function

Step 2A: Create the Lambda Execution Role

1. Use CloudFormation for quick setup by clicking this link to create the necessary IAM role.


CloudFormation > Stacks > Create stack

Quick create stack

Template

Template URL
https://learn-cantrill-labs.s3.amazonaws.com/aws-serverless-pet-cuddle-o-tron/lambdarolecfn.yaml

Stack description
-

Provide a stack name

Stack name
LAMBDAROLE

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters

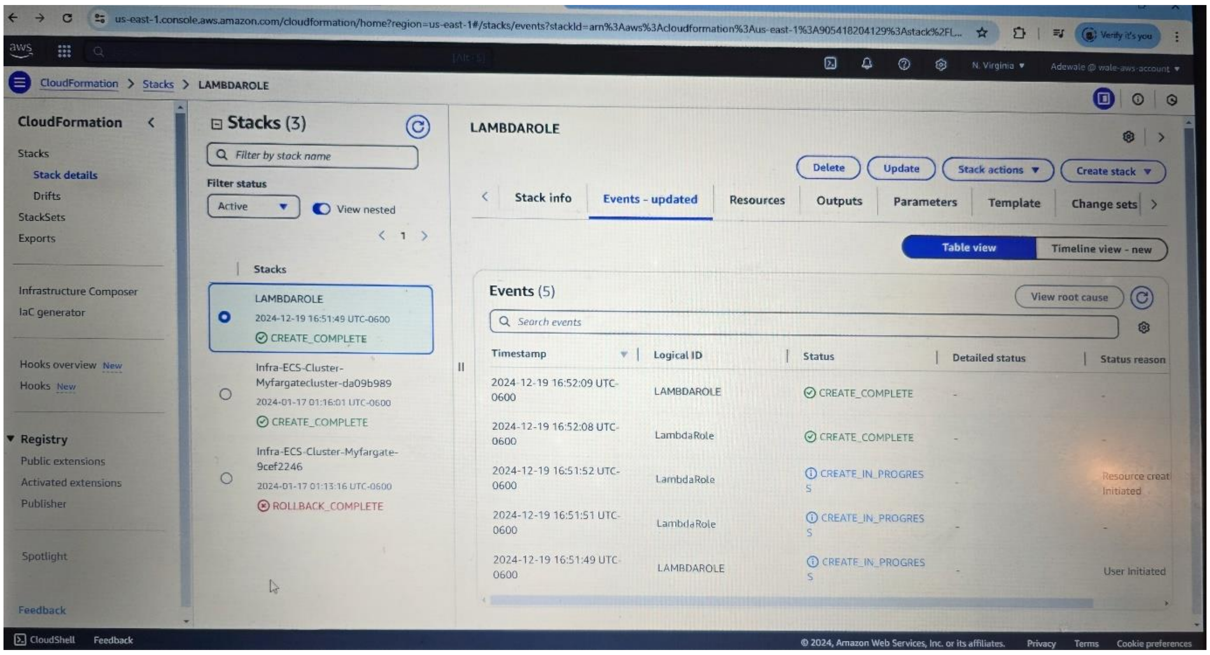
There are no parameters defined in your template

Permissions - *optional*

Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.

IAM role - optional

2. Check the acknowledgment box and click **Create Stack**.


CloudFormation > Stacks > LAMBDAROLE

Stacks (3)

Stacks
LAMBDAROLE 2024-12-19 16:51:49 UTC-0600 CREATE_COMPLETE
Infra-ECS-Cluster-Myfargatecluster-da09b989 2024-01-17 01:16:01 UTC-0600 CREATE_COMPLETE
Infra-ECS-Cluster-Myfargate-9cef2246 2024-01-17 01:13:16 UTC-0600 ROLLBACK_COMPLETE

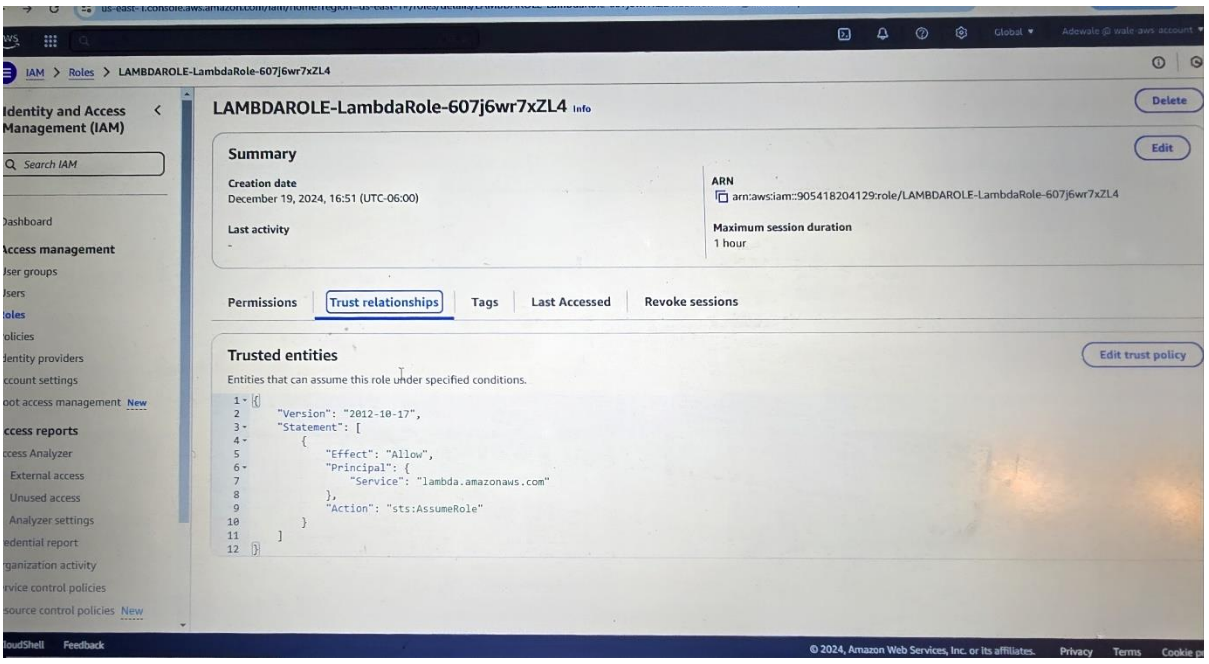
LAMBDAROLE

Events (5)

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-12-19 16:52:09 UTC-0600	LAMBDAROLE	CREATE_COMPLETE	-	-
2024-12-19 16:52:08 UTC-0600	LambdaRole	CREATE_COMPLETE	-	-
2024-12-19 16:51:52 UTC-0600	LambdaRole	CREATE_IN_PROGRESS	-	Resource creation Initiated
2024-12-19 16:51:51 UTC-0600	LambdaRole	CREATE_IN_PROGRESS	-	-
2024-12-19 16:51:49 UTC-0600	LAMBDAROLE	CREATE_IN_PROGRESS	-	User Initiated

3. Wait until the stack reaches the **CREATE_COMPLETE** status.

4. Review the execution role in the IAM console, noting its permissions for SES, SNS, and logging.


IAM > Roles > LAMBDAROLE-LambdaRole-607j6wr7xZL4

LAMBDAROLE-LambdaRole-607j6wr7xZL4

Summary

Creation date: December 19, 2024, 16:51 (UTC-06:00)

ARN: arn:aws:iam::905418204129:role/LAMBDAROLE-LambdaRole-607j6wr7xZL4

Last activity: -

Maximum session duration: 1 hour

Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

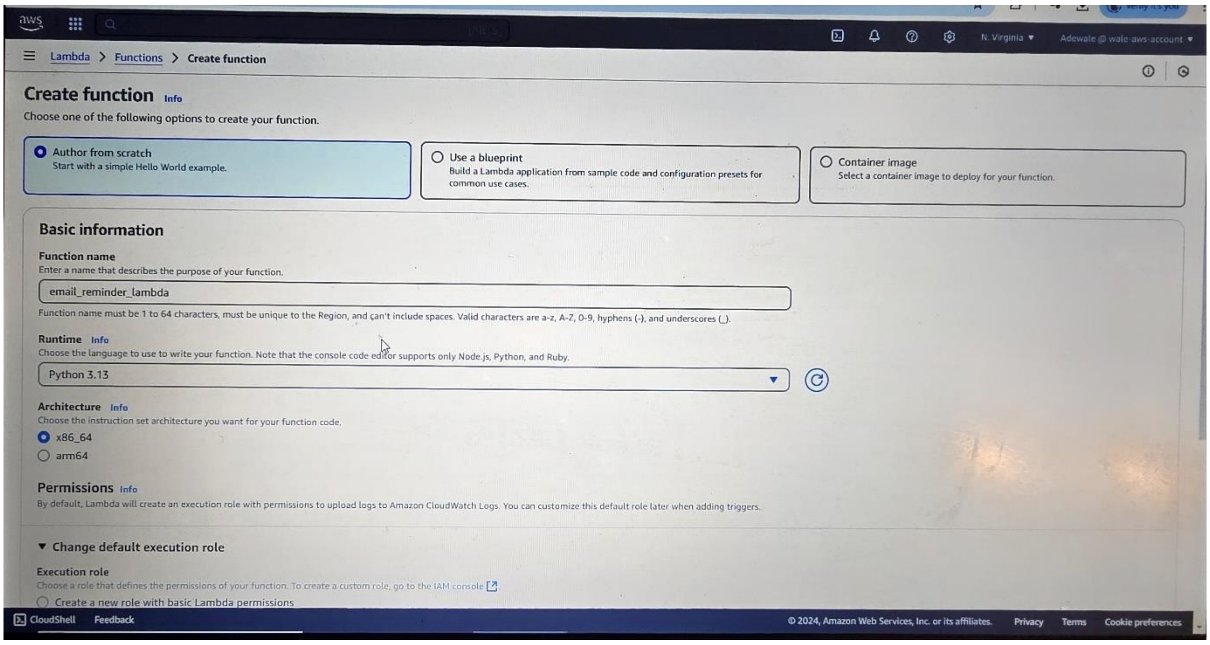
Step 2B: Create the Lambda Function

1. Open the Lambda Console and click **Create Function**.

2. Choose **Author from scratch**, name the function email_reminder_lambda, and set the runtime to **Python 3.9**.

3. Expand **Change default execution role**, select **Use an existing role**, and choose the LambdaRole created earlier.

4. Click **Create Function**.



Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

- [✓ Author from scratch] Start with a simple Hello World example.
- Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image: Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
email_reminder_lambda
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

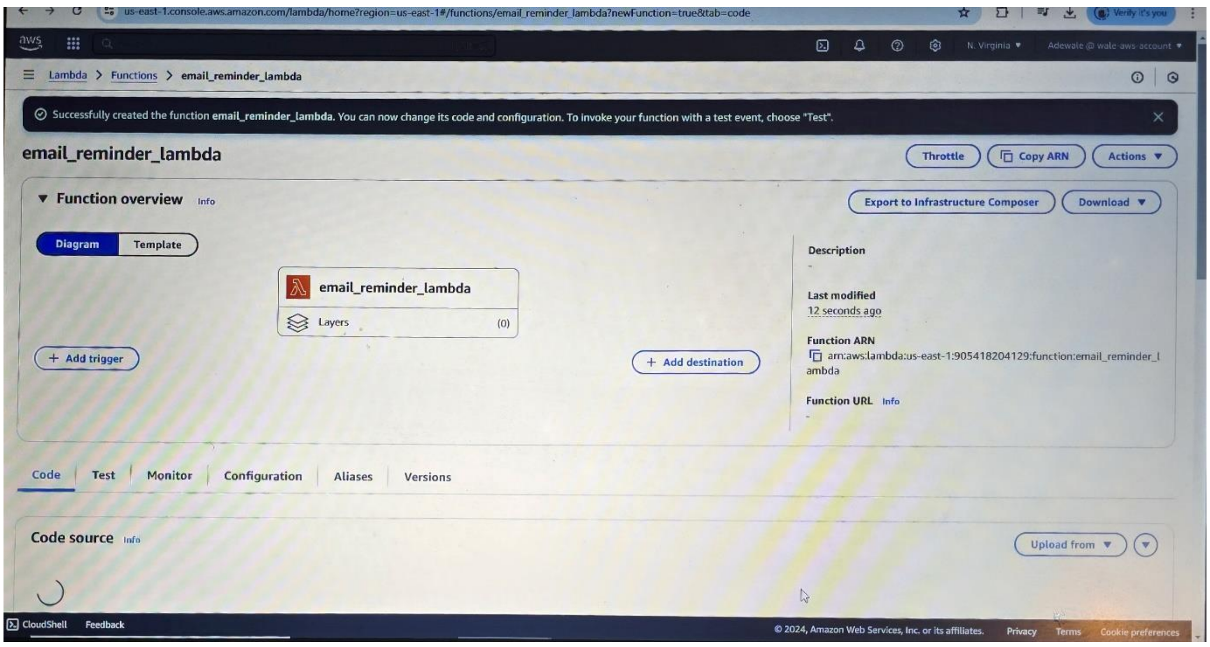
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console
Create a new role with basic Lambda permissions



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/email_reminder_lambda?newFunction=true&tab=code

Lambda > Functions > email_reminder_lambda

Successfully created the function email_reminder_lambda. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

email_reminder_lambda

▼ Function overview Info

Description
-

Last modified
12 seconds ago

Function ARN
arn:aws:lambda:us-east-1:905418204129:function:email_reminder_lambda

Function URL Info
-

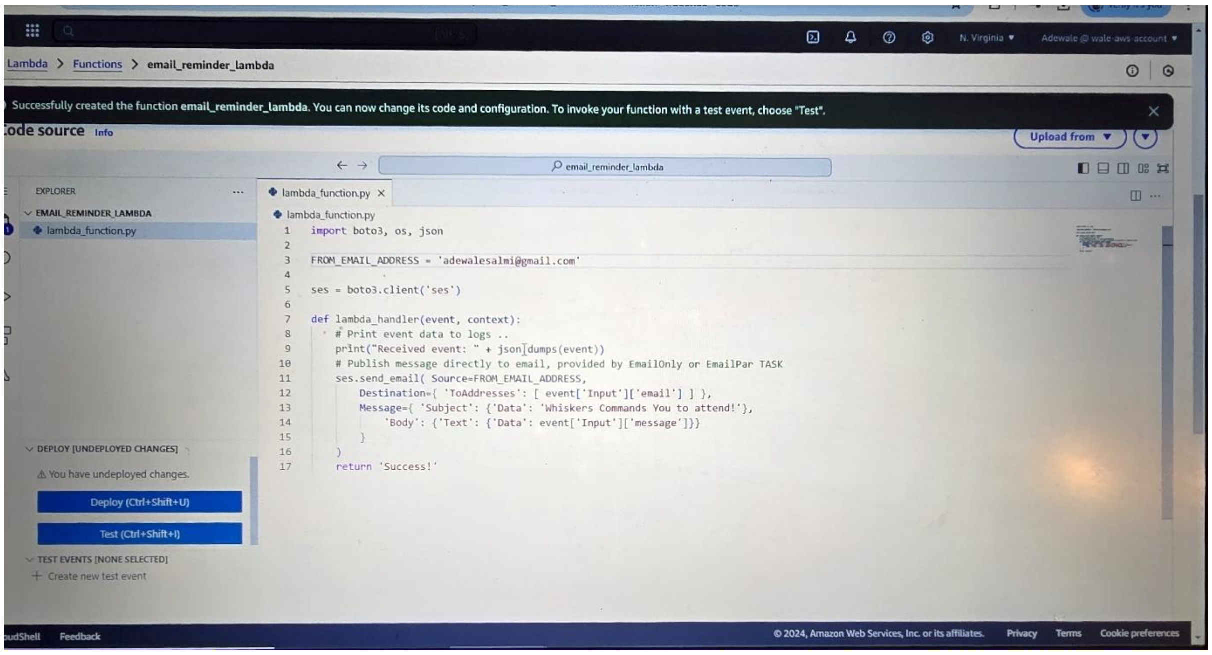
Code source Info

Step 2C: Configure the Lambda Function

1. Replace the default code with the following:

```
import boto3, os, json

FROM_EMAIL_ADDRESS = 'REPLACE_ME'

ses = boto3.client('ses')
```

```
def lambda_handler(event, context):
    print("Received event: " + json.dumps(event))
    ses.send_email(
        Source=FROM_EMAIL_ADDRESS,
        Destination={'ToAddresses': [event['Input']['email']]},
        Message={
            'Subject': {'Data': 'Whiskers Commands You to Attend!'},
            'Body': {'Text': {'Data': event['Input']['message']}}
        }
    )
    return 'Success!'
```



Successfully created the function **email_reminder_lambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

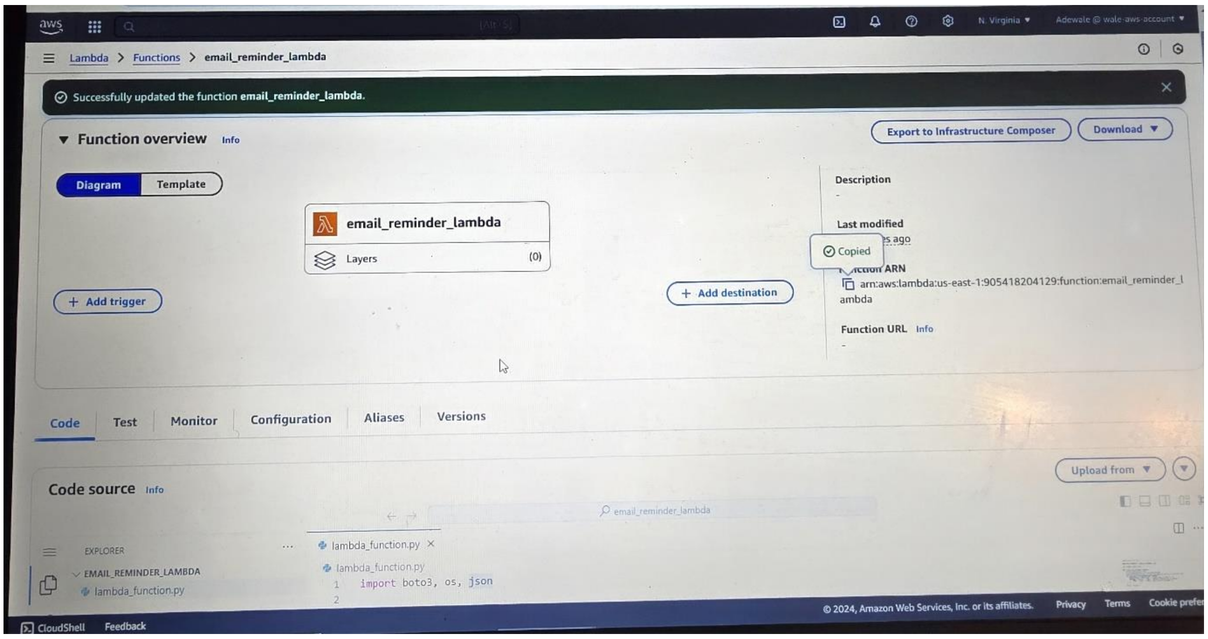
⚠ You have undeployed changes.

```
import boto3, os, json

FROM_EMAIL_ADDRESS = 'adewalesalmi@gmail.com'

ses = boto3.client('ses')

def lambda_handler(event, context):
    # Print event data to logs ..
    print("Received event: " + json.dumps(event))
    # Publish message directly to email, provided by EmailOnly or EmailPar TASK
    ses.send_email( Source=FROM_EMAIL_ADDRESS,
        Destination={ 'ToAddresses': [ event['Input']['email'] ] },
        Message={ 'Subject': {'Data': 'Whiskers Commands You to attend!'},
            'Body': {'Text': {'Data': event['Input']['message']}}
        }
    )
    return 'Success!'
```

© 2024, Amazon Web Services, Inc. or its affiliates.

2. Replace 'REPLACE_ME' with the **PetCuddleOTron Sending Address** and deploy the code.

3. Record the ARN of the email_reminder_lambda function for use in Stage 3.


Lambda > Functions > email_reminder_lambda

Successfully updated the function **email_reminder_lambda**.

Function overview Info

Description
-

Last modified
...s ago

Function ARN
arn:aws:lambda:us-east-1:905418204129:function:email_reminder_lambda

Function URL Info
-

Code source Info

lambda_function.py

```
import boto3, os, json
```

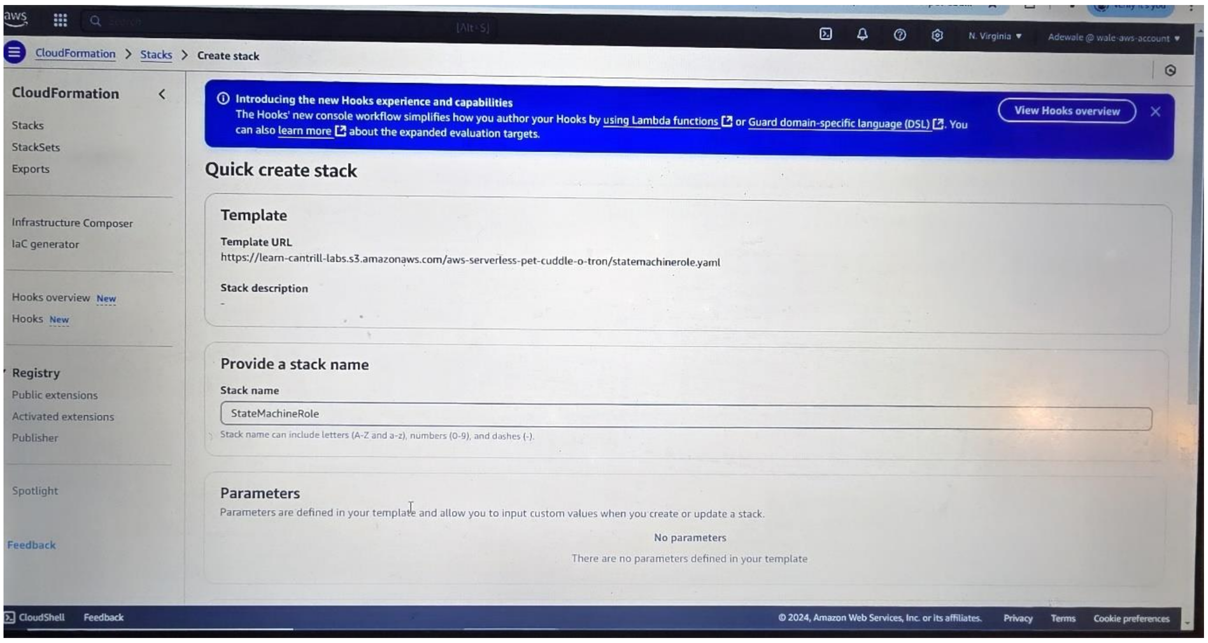
Completion of Stage 2

The Lambda function is now ready to send emails using SES.

STAGE 3: Setting up the State Machine

Step 3A: Create the State Machine Role

1. Use CloudFormation to create the necessary IAM role by clicking this link.
2. Wait until the stack reaches the **CREATE_COMPLETE** status.


CloudFormation > Stacks > Create stack

Introducing the new Hooks experience and capabilities
The Hooks' new console workflow simplifies how you author your Hooks by using Lambda functions or Guard domain-specific language (DSL). You can also learn more about the expanded evaluation targets.

Quick create stack

Template

Template URL
https://learn-cantrill-labs.s3.amazonaws.com/aws-serverless-pet-cuddle-o-tron/statemachinerole.yaml

Stack description
-

Provide a stack name

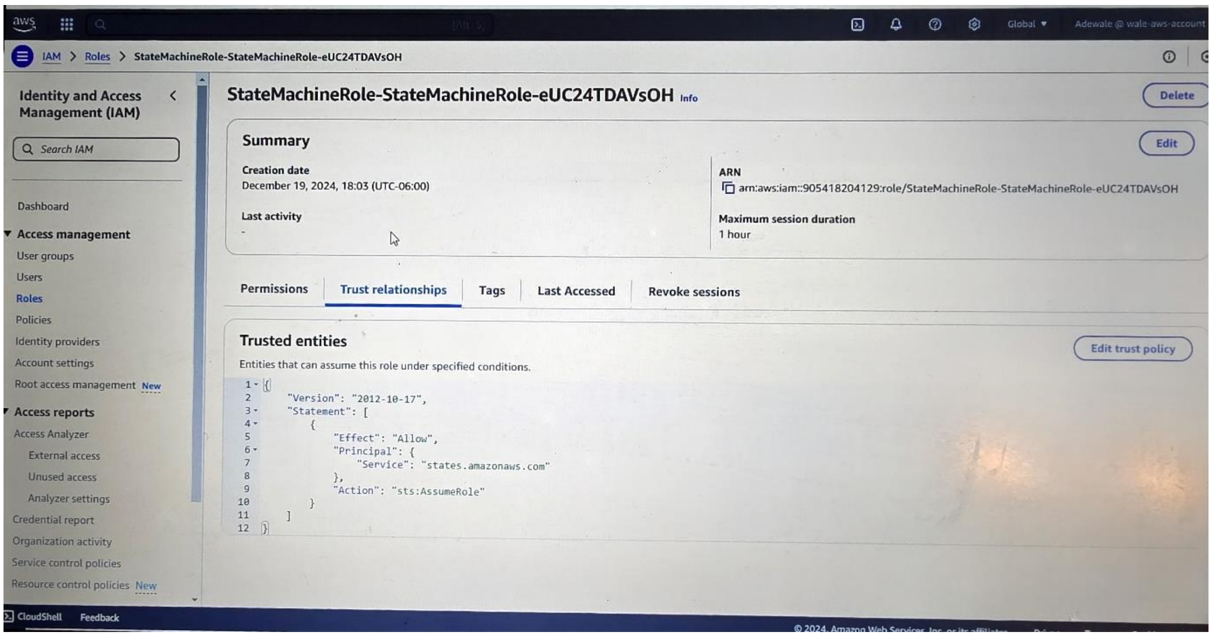
Stack name
StateMachineRole

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters
There are no parameters defined in your template

3. Review the permissions, which include logging, invoking the Lambda function, and sending SMS via SNS.



StateMachineRole-StateMachineRole-eUC24TDAVsOH

Summary

Creation date
December 19, 2024, 18:03 (UTC-06:00)

Last activity
-

ARN
arn:aws:iam::905418204129:role/StateMachineRole-StateMachineRole-eUC24TDAVsOH

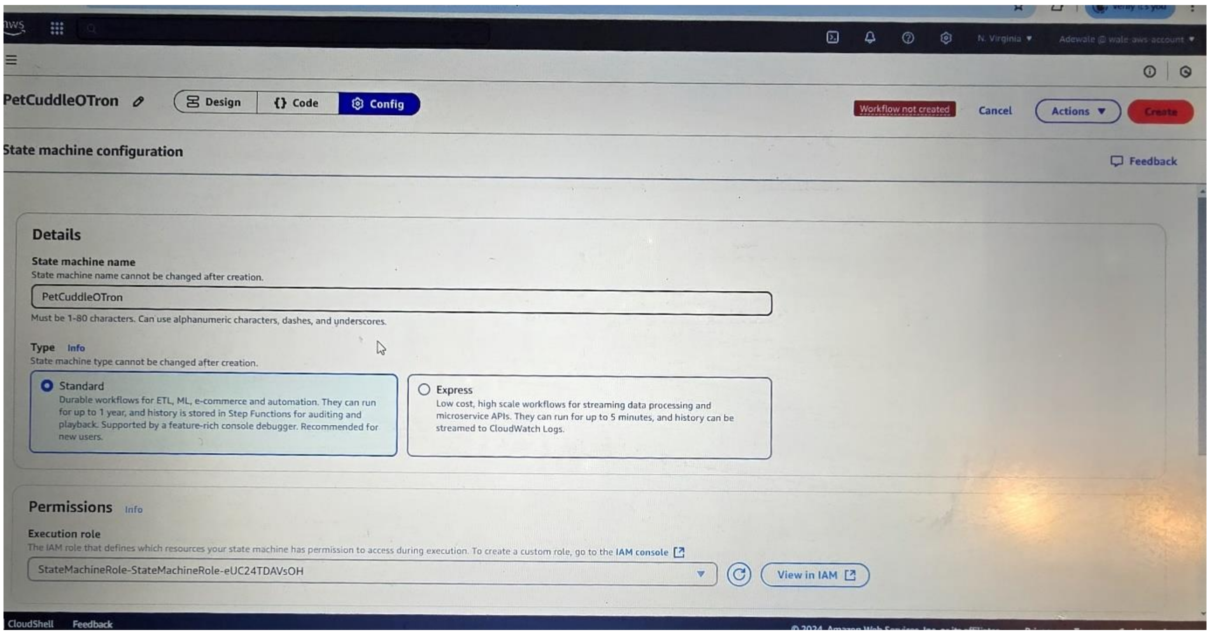
Maximum session duration
1 hour

Trusted entities
Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "states.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Step 3B: Create the State Machine

1. Open the Step Functions Console and create a new state machine.
2. Select **Write your workflow in code**, choose **Standard** for the type, and copy the JSON definition from this link.



State machine configuration

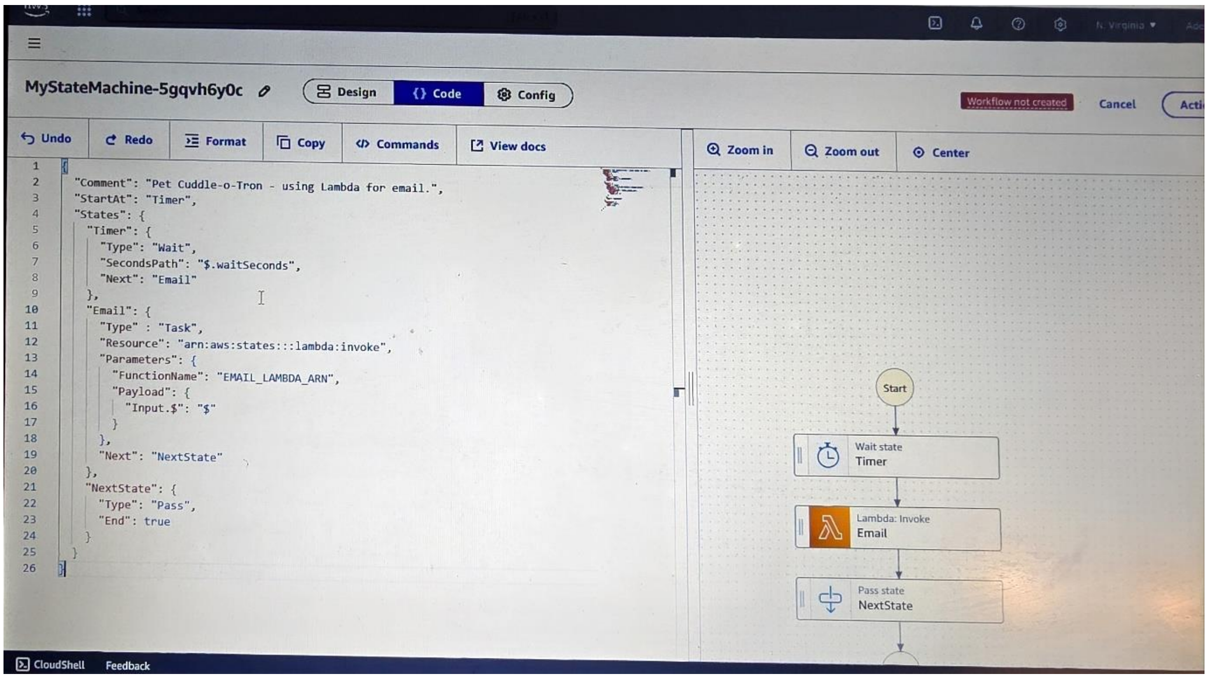
Details

State machine name
State machine name cannot be changed after creation.
PetCuddleOTron
Must be 1-80 characters. Can use alphanumeric characters, dashes, and underscores.

Type
State machine type cannot be changed after creation.
[✓ Standard] Durable workflows for ETL, ML, e-commerce and automation. They can run for up to 1 year, and history is stored in Step Functions for auditing and playback. Supported by a feature-rich console debugger. Recommended for new users.
Express: Low cost, high scale workflows for streaming data processing and microservice APIs. They can run for up to 5 minutes, and history can be streamed to CloudWatch Logs.

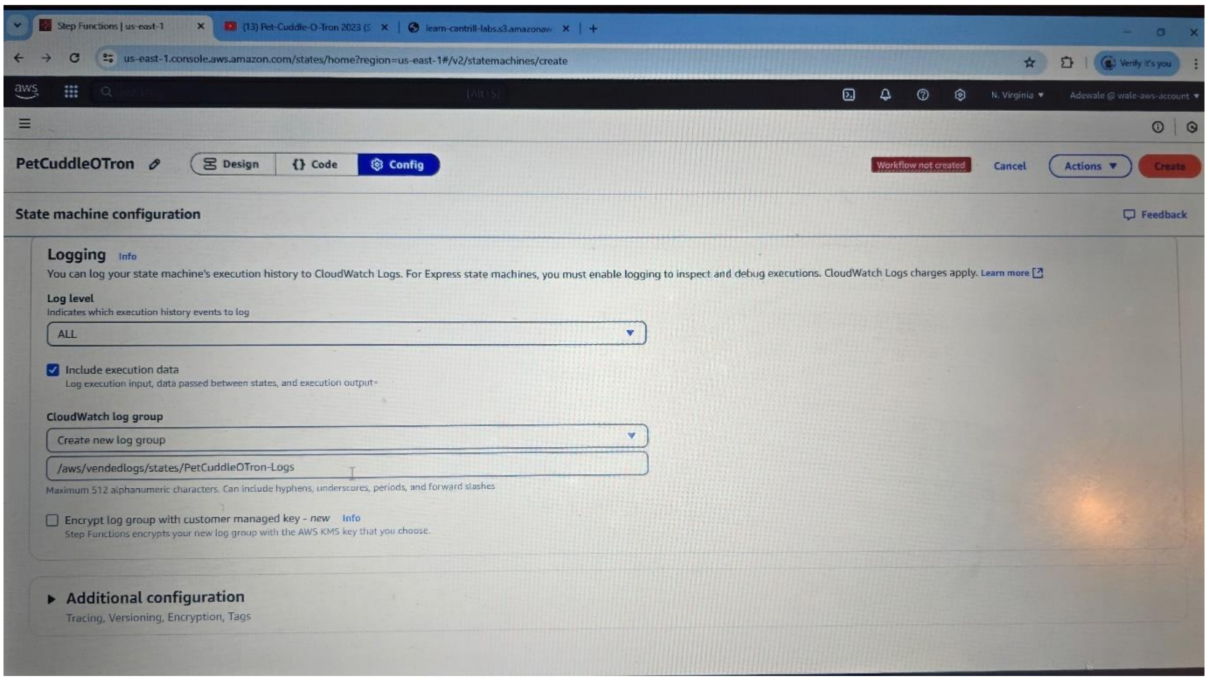
Permissions

Execution role
The IAM role that defines which resources your state machine has permission to access during execution. To create a custom role, go to the IAM console
StateMachineRole-StateMachineRole-eUC24TDAVsOH



```
{
  "Comment": "Pet Cuddle-o-Tron - using Lambda for email.",
  "StartAt": "Timer",
  "States": {
    "Timer": {
      "Type": "Wait",
      "SecondsPath": "$.waitSeconds",
      "Next": "Email"
    },
    "Email": {
      "Type" : "Task",
      "Resource": "arn:aws:states:::lambda:invoke",
      "Parameters": {
        "FunctionName": "EMAIL_LAMBDA_ARN",
        "Payload": {
          "Input.$": "$"
        }
      },
      "Next": "NextState"
    },
    "NextState": {
      "Type": "Pass",
      "End": true
    }
  }
}
```

3. Replace the EMAIL_LAMBDA_ARN placeholder with the ARN of the email_reminder_lambda function.
4. Name the state machine PetCuddleOTron, assign the **StateMachineRole**, and enable detailed logging.


Step Functions | us-east-1
(13) Pet-Cuddle-O-Tron 2023 (S
learn-cantrill-labs.s3.amazonaw
us-east-1.console.aws.amazon.com/states/home?region=us-east-1#/v2/statemachines/create
Verify it's you
N. Virginia
PetCuddleOTron
Design
Code
Config
Workflow not created
Cancel
Actions
Create
Feedback
Info

State machine configuration

Logging

You can log your state machine's execution history to CloudWatch Logs. For Express state machines, you must enable logging to inspect and debug executions. CloudWatch Logs charges apply. Learn more

Log level
Indicates which execution history events to log
ALL

☑ Include execution data
Log execution input, data passed between states, and execution output

CloudWatch log group
Create new log group
/aws/vendedlogs/states/PetCuddleOTron-Logs
Maximum 512 alphanumeric characters. Can include hyphens, underscores, periods, and forward slashes

☐ Encrypt log group with customer managed key - new
Step Functions encrypts your new log group with the AWS KMS key that you choose.

Additional configuration
Tracing, Versioning, Encryption, Tags

5. Record the ARN of the state machine for use in Stage 4.

Completion of Stage 3

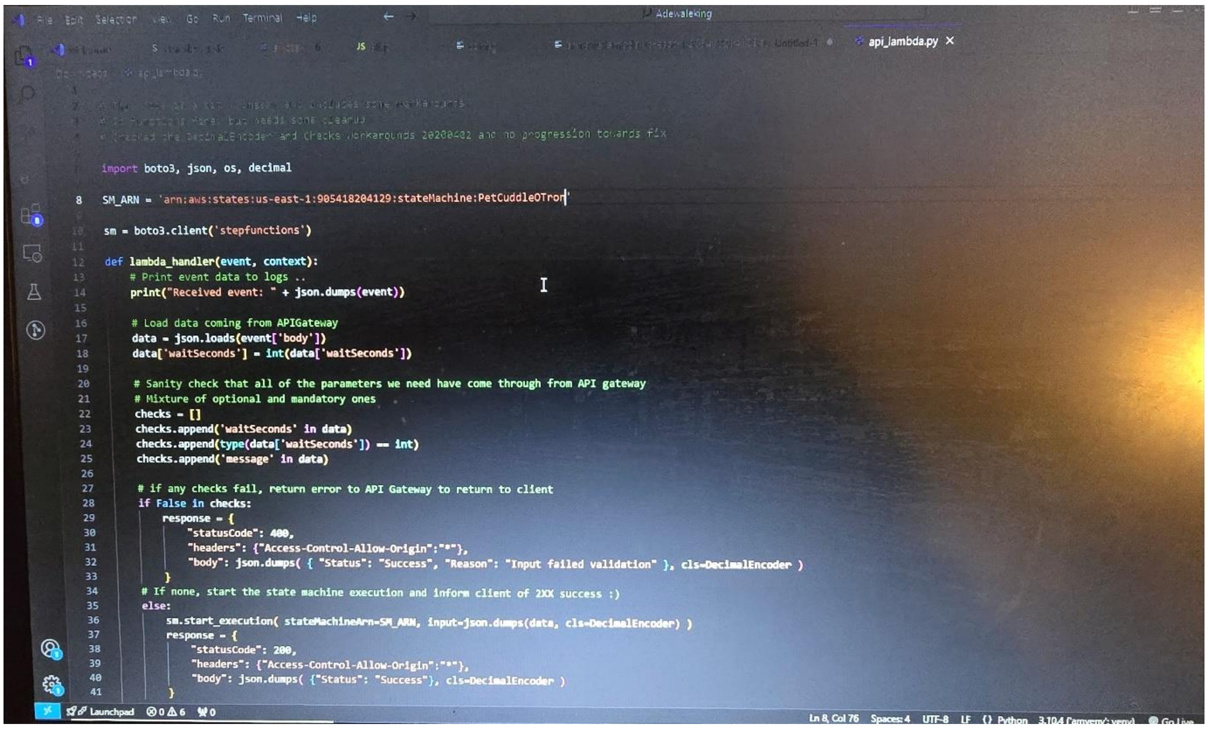
The state machine is now configured to manage the application's workflow.

STAGE 4: Building the API Gateway

Step 4A: Create the API Lambda Function

1. Create a new Lambda function named api_lambda, set the runtime to **Python 3.9**, and assign it the LambdaRole.

2. Replace the default code with the contents from [my github](#).



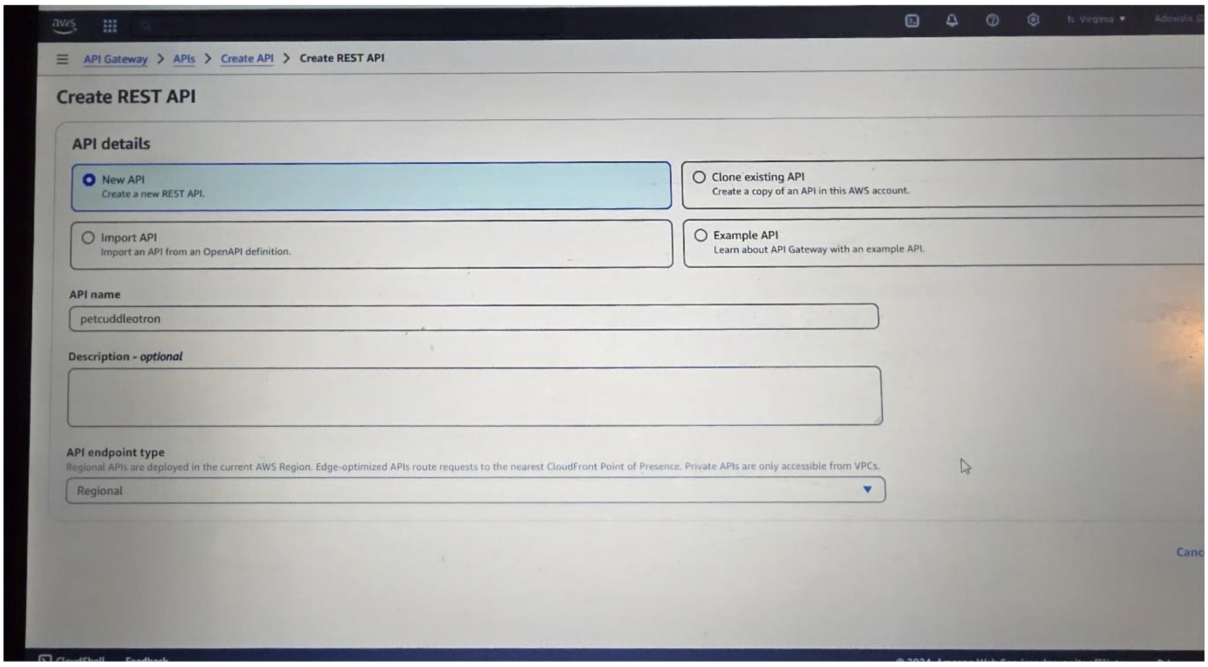
```python
# [illegible]
# [illegible]
# [illegible] the DecimalEncoder and Checks workarounds 20200402 and no progression towards fix

import boto3, json, os, decimal

SM_ARN = 'arn:aws:states:us-east-1:905418204129:stateMachine:PetCuddleOTron'

sm = boto3.client('stepfunctions')

def lambda_handler(event, context):
    # Print event data to logs ..
    print("Received event: " + json.dumps(event))

    # Load data coming from APIGateway
    data = json.loads(event['body'])
    data['waitSeconds'] = int(data['waitSeconds'])

    # Sanity check that all of the parameters we need have come through from API gateway
    # Mixture of optional and mandatory ones
    checks = []
    checks.append('waitSeconds' in data)
    checks.append(type(data['waitSeconds']) == int)
    checks.append('message' in data)

    # if any checks fail, return error to API Gateway to return to client
    if False in checks:
        response = {
            "statusCode": 400,
            "headers": {"Access-Control-Allow-Origin":"*"},
            "body": json.dumps( { "Status": "Success", "Reason": "Input failed validation" }, cls=DecimalEncoder )
        }
    # If none, start the state machine execution and inform client of 2XX success :)
    else:
        sm.start_execution( stateMachineArn=SM_ARN, input=json.dumps(data, cls=DecimalEncoder) )
        response = {
            "statusCode": 200,
            "headers": {"Access-Control-Allow-Origin":"*"},
            "body": json.dumps( {"Status": "Success"}, cls=DecimalEncoder )
        }
```

3. Replace the YOUR_STATEMACHINE_ARN placeholder with the state machine ARN.

4. Deploy the function.

Step 4B: Create the API Gateway

1. Open the [API Gateway Console](#) and create a new REST API named petcuddleotron.



API Gateway > APIs > Create API > Create REST API

Create REST API

API details

[✓ New API] Create a new REST API.

Clone existing API — Create a copy of an API in this AWS account.

Import API — Import an API from an OpenAPI definition.

Example API — Learn about API Gateway with an example API.

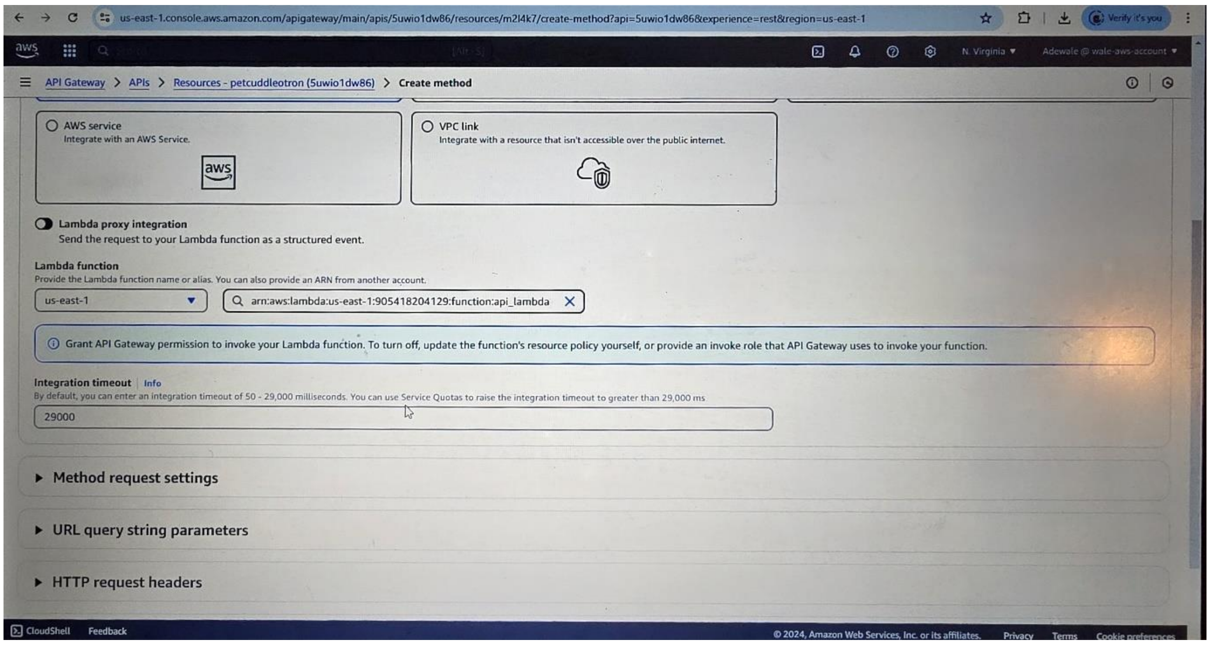
API name: petcuddleotron

Description - *optional*

API endpoint type: Regional

Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

2. Add a resource named /petcuddleotron with CORS enabled.



us-east-1.console.aws.amazon.com/apigateway/main/apis/5uwio1dw86/resources/m2l4k7/create-method?api=5uwio1dw86&experience=rest&region=us-east-1

API Gateway > APIs > Resources - petcuddleotron (5uwio1dw86) > Create method

AWS service
Integrate with an AWS Service.

VPC link
Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration
Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.
us-east-1
arn:aws:lambda:us-east-1:905418204129:function:api_lambda

Grant API Gateway permission to invoke your Lambda function. To turn off, update the function's resource policy yourself, or provide an invoke role that API Gateway uses to invoke your function.

Integration timeout | Info
By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms
29000

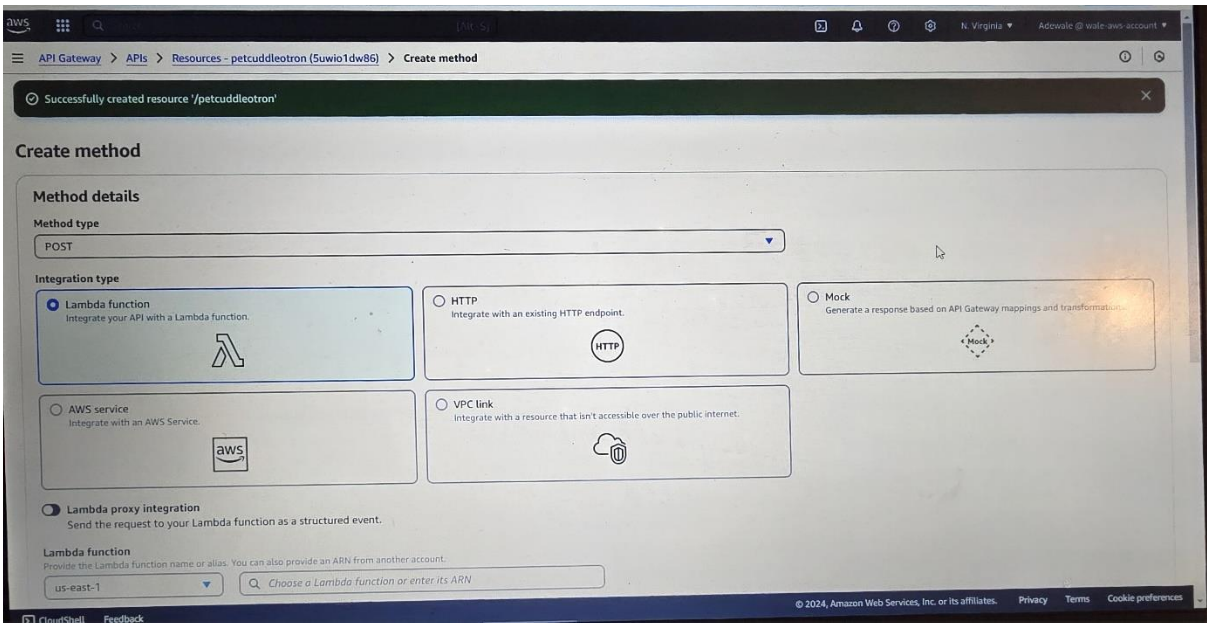
▸ Method request settings

▸ URL query string parameters

▸ HTTP request headers

© 2024, Amazon Web Services, Inc. or its affiliates.

3. Add a POST method to the resource, integrate it with the api_lambda function, and deploy it to a new stage named prod.



API Gateway > APIs > Resources - petcuddleotron (5uwio1dw86) > Create method

Successfully created resource '/petcuddleotron'

Create method

Method details

Method type
POST

Integration type

Lambda function
Integrate your API with a Lambda function.

HTTP
Integrate with an existing HTTP endpoint.

Mock
Generate a response based on API Gateway mappings and transformations.

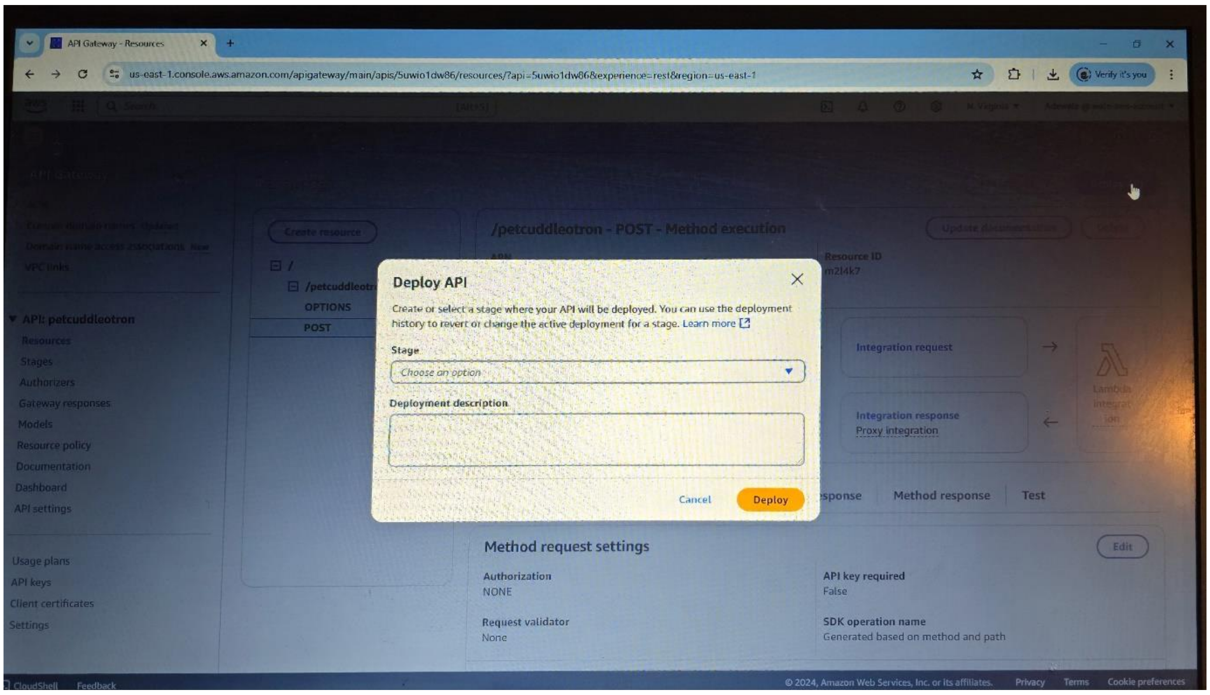
AWS service
Integrate with an AWS Service.

VPC link
Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration
Send the request to your Lambda function as a structured event.

Lambda function
Provide the Lambda function name or alias. You can also provide an ARN from another account.
us-east-1
Choose a Lambda function or enter its ARN

© 2024, Amazon Web Services, Inc. or its affiliates.


us-east-1.console.aws.amazon.com/apigateway/main/apis/5uwio1dw86/resources?api=5uwio1dw86&experience=rest&region=us-east-1

API: petcuddleotron

/petcuddleotron - POST - Method execution

Deploy API

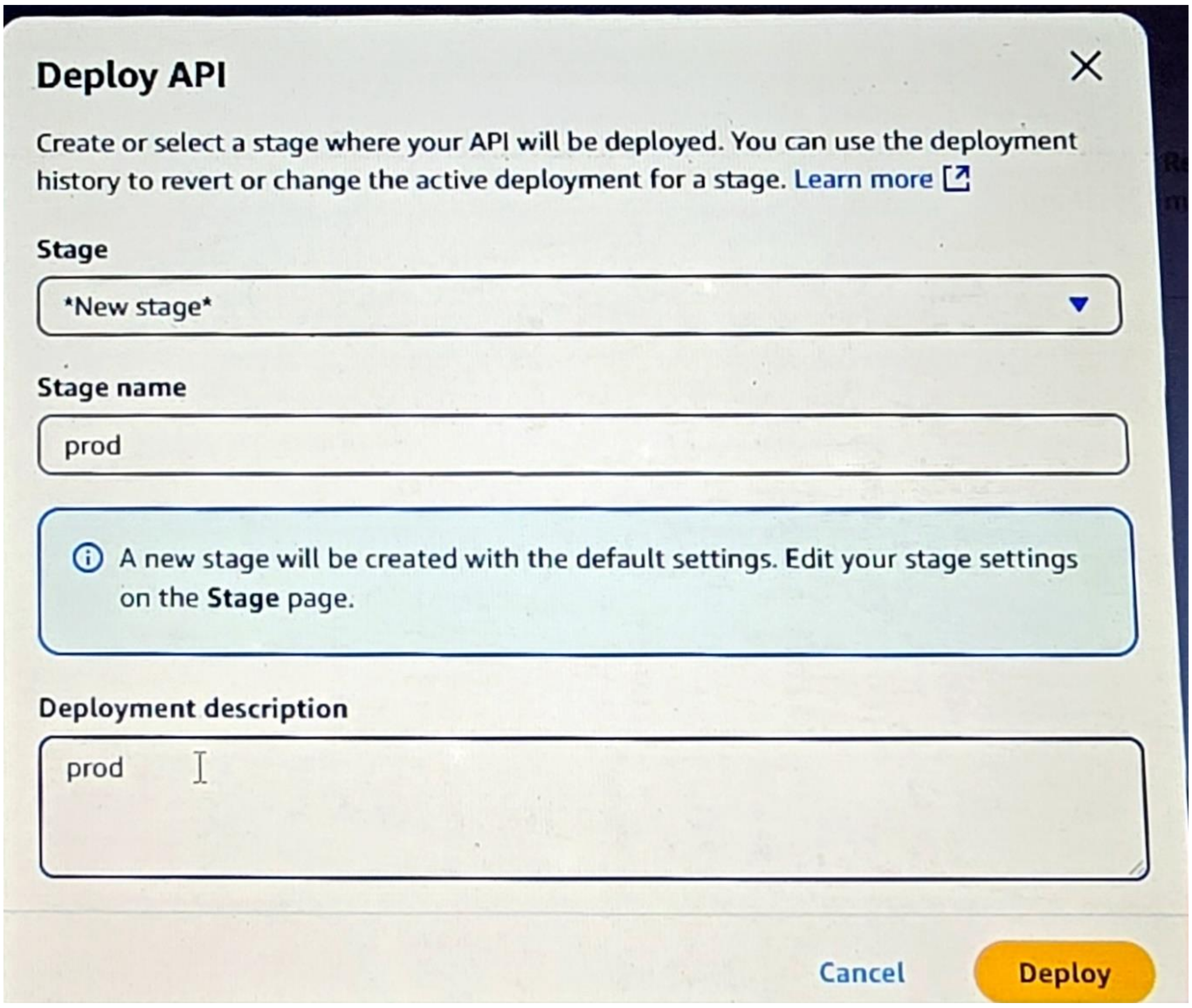
Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. Learn more

Stage: Choose an option

Deployment description:

Method request settings

Authorization: NONE	API key required: False
Request validator: None	SDK operation name: Generated based on method and path


Deploy API

Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. Learn more

Stage: *New stage*

Stage name: prod

A new stage will be created with the default settings. Edit your stage settings on the **Stage** page.

Deployment description: prod

4. Record the **Invoke URL** for use in Stage 5.

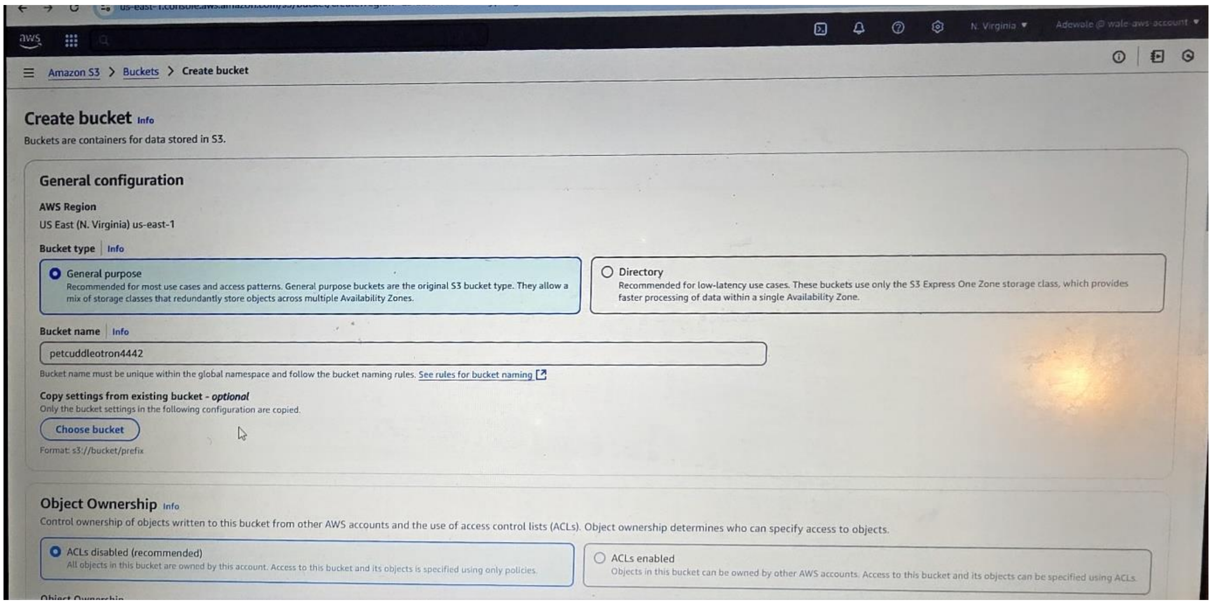
Completion of Stage 4

The API Gateway is ready to handle requests from the application's front end.

STAGE 5: Setting Up the Front End

Step 5A: Create and configure an S3 Bucket

1. Create a new S3 bucket with public access enabled.



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type | Info

[✓ General purpose] Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory — Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name | Info
petcuddleotron4442
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix

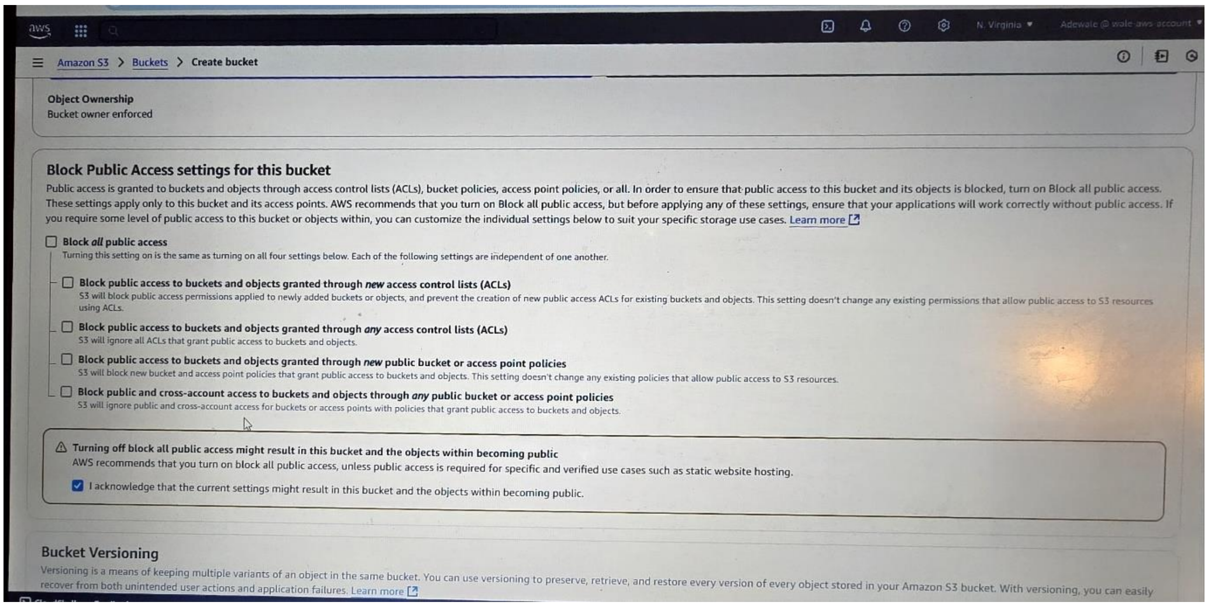
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

[✓ ACLs disabled (recommended)] All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled — Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

2. Add a bucket policy to allow public reads, replacing the placeholder ARN with your bucket's ARN.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"PublicRead",
      "Effect":"Allow",
      "Principal": "*",
      "Action":["s3:GetObject"],
      "Resource":["REPLACEME_PET_CUDDLE_O_TRON_BUCKET_ARN/*"]
    }
  ]
}
```



Amazon S3 > Buckets > Create bucket

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

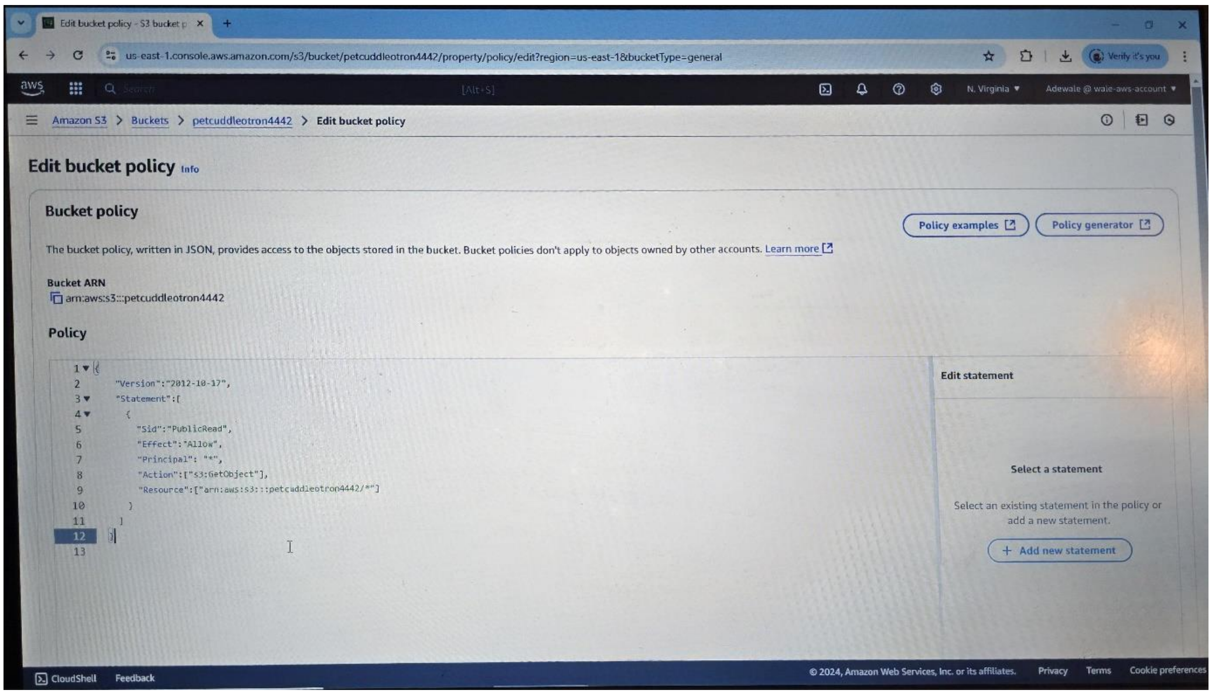
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- [ ] **Block *all* public access**
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
  - [ ] **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - [ ] **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - [ ] **Block public access to buckets and objects granted through *new* public bucket or access point policies**
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - [ ] **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
- [x] I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more



us-east-1.console.aws.amazon.com/s3/bucket/petcuddleotron4442/property/policy/edit?region=us-east-1&bucketType=general

Amazon S3 > Buckets > petcuddleotron4442 > Edit bucket policy

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN
arn:aws:s3:::petcuddleotron4442

Policy

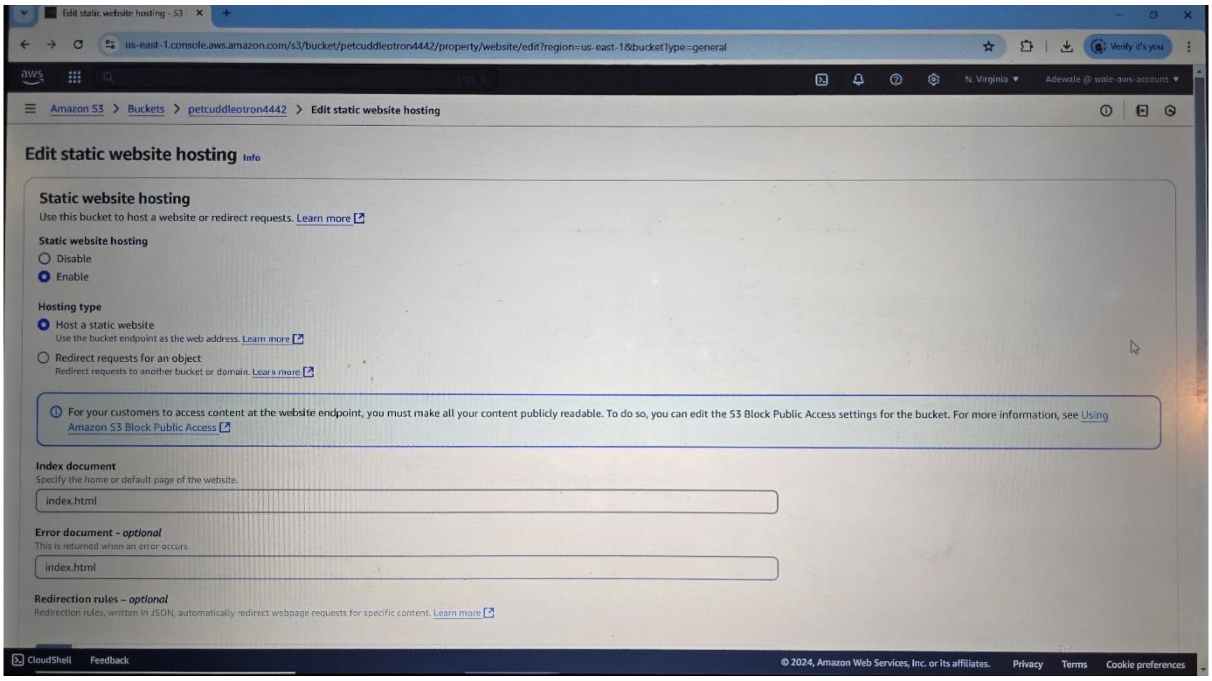
```
{
    "Version":"2012-10-17",
    "Statement":[
      {
        "Sid":"PublicRead",
        "Effect":"Allow",
        "Principal": "*",
        "Action":["s3:GetObject"],
        "Resource":["arn:aws:s3:::petcuddleotron4442/*"]
      }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

3. Enable static website hosting and note the bucket endpoint.


Amazon S3 > Buckets > petcuddleotron4442 > Edit static website hosting

Edit static website hosting

Static website hosting

Use this bucket to host a website or redirect requests.

Static website hosting

Disable [✓ Enable]

Hosting type

[✓ Host a static website]
Use the bucket endpoint as the web address.

Redirect requests for an object
Redirect requests to another bucket or domain.

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

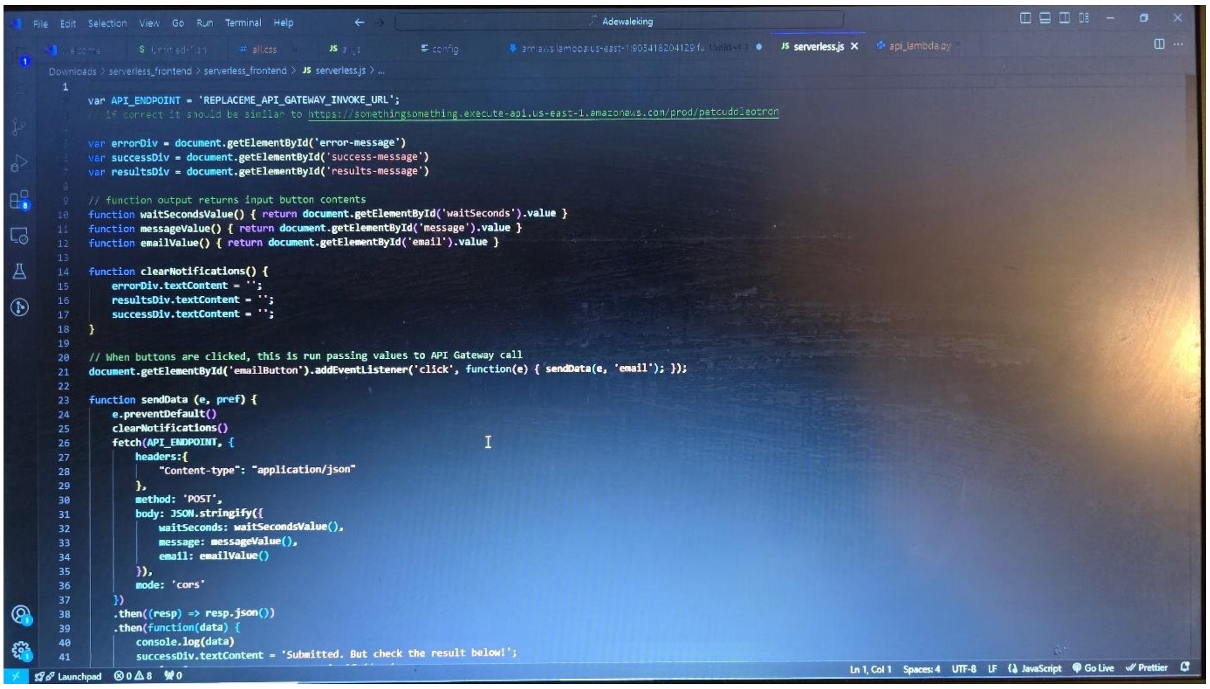
Index document
Specify the home or default page of the website.
index.html

Error document - *optional*
This is returned when an error occurs.
index.html

Redirection rules – *optional*
Redirection rules, written in JSON, automatically redirect webpage requests for specific content.

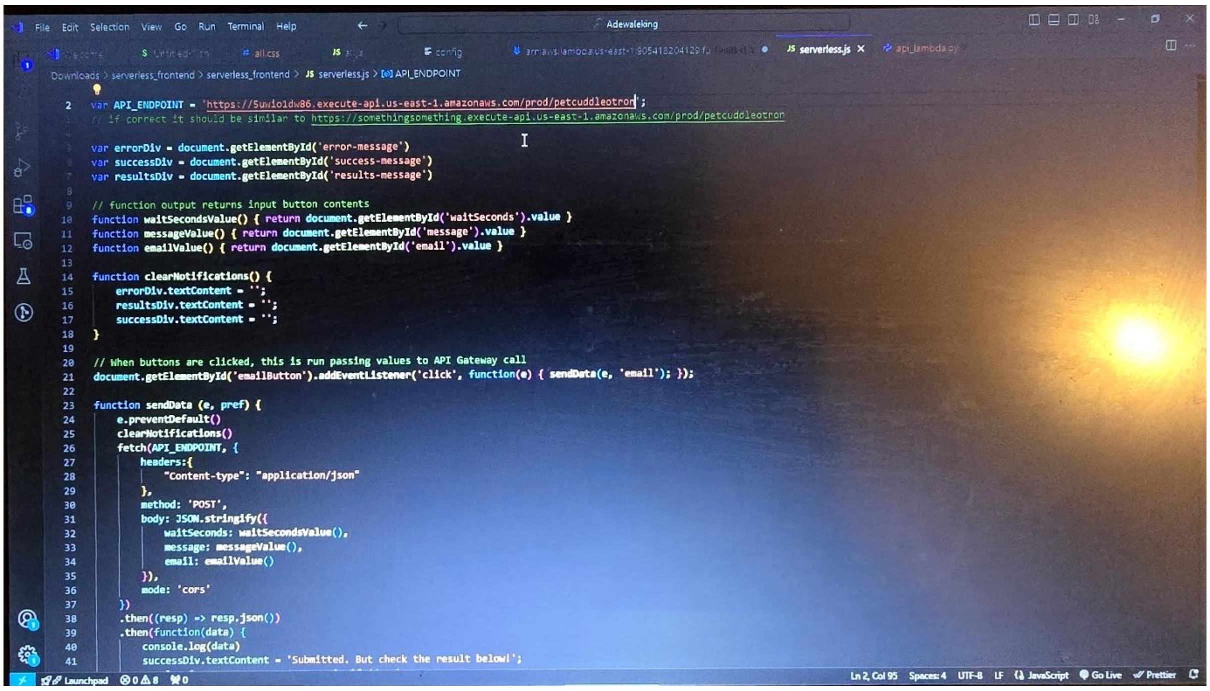
Step 5B: Configure and Upload Front End Files

1. Download and edit the front-end files from my github.


Downloads > serverless_frontend > serverless_frontend > serverless.js > ...

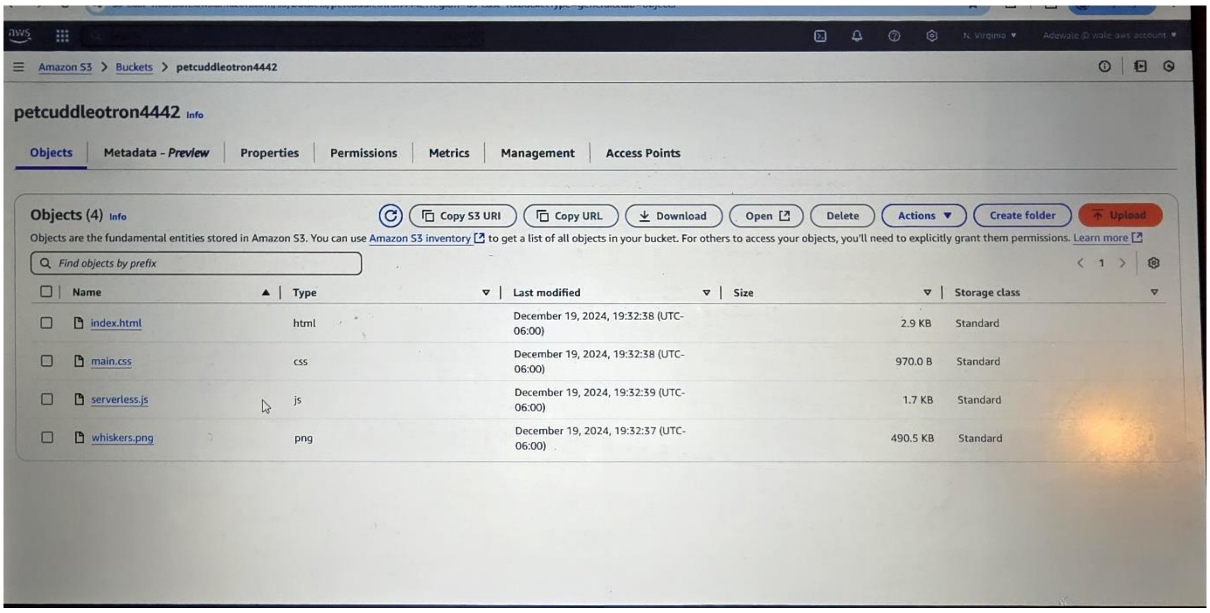
```
var API_ENDPOINT = 'REPLACEME_API_GATEWAY_INVOKE_URL';
// if correct it should be similar to https://somethingsomething.execute-api.us-east-1.amazonaws.com/prod/petcuddleotron

var errorDiv = document.getElementById('error-message')
var successDiv = document.getElementById('success-message')
var resultsDiv = document.getElementById('results-message')

// function output returns input button contents
function waitSecondsValue() { return document.getElementById('waitSeconds').value }
function messageValue() { return document.getElementById('message').value }
function emailValue() { return document.getElementById('email').value }

function clearNotifications() {
    errorDiv.textContent = '';
    resultsDiv.textContent = '';
    successDiv.textContent = '';
}

// When buttons are clicked, this is run passing values to API Gateway call
document.getElementById('emailButton').addEventListener('click', function(e) { sendData(e, 'email'); });

function sendData (e, pref) {
    e.preventDefault()
    clearNotifications()
    fetch(API_ENDPOINT, {
        headers:{
            "Content-type": "application/json"
        },
        method: 'POST',
        body: JSON.stringify({
            waitSeconds: waitSecondsValue(),
            message: messageValue(),
            email: emailValue()
        }),
        mode: 'cors'
    })
    .then((resp) => resp.json())
    .then(function(data) {
        console.log(data)
        successDiv.textContent = 'Submitted. But check the result below!';
```

2. Replace the API URL placeholder in serverless.js with your API Gateway Invoke URL.



```
var API_ENDPOINT = 'https://5uwio1dw86.execute-api.us-east-1.amazonaws.com/prod/petcuddleotron';
// if correct it should be similar to https://somethingsomething.execute-api.us-east-1.amazonaws.com/prod/petcuddleotron

var errorDiv = document.getElementById('error-message')
var successDiv = document.getElementById('success-message')
var resultsDiv = document.getElementById('results-message')

// function output returns input button contents
function waitSecondsValue() { return document.getElementById('waitSeconds').value }
function messageValue() { return document.getElementById('message').value }
function emailValue() { return document.getElementById('email').value }

function clearNotifications() {
    errorDiv.textContent = '';
    resultsDiv.textContent = '';
    successDiv.textContent = '';
}

// When buttons are clicked, this is run passing values to API Gateway call
document.getElementById('emailButton').addEventListener('click', function(e) { sendData(e, 'email'); });

function sendData (e, pref) {
    e.preventDefault()
    clearNotifications()
    fetch(API_ENDPOINT, {
        headers:{
            "Content-type": "application/json"
        },
        method: 'POST',
        body: JSON.stringify({
            waitSeconds: waitSecondsValue(),
            message: messageValue(),
            email: emailValue()
        }),
        mode: 'cors'
    })
    .then((resp) => resp.json())
    .then(function(data) {
        console.log(data)
        successDiv.textContent = 'Submitted. But check the result below!';
```

3. Upload the files to the S3 bucket.



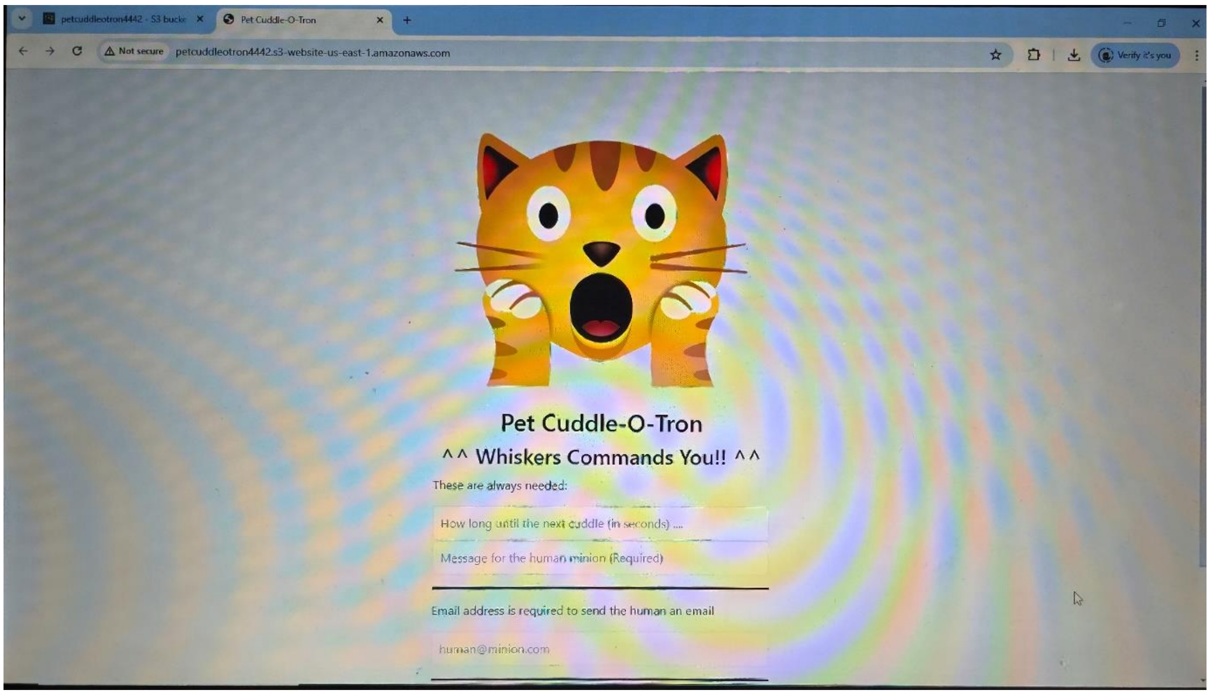
Amazon S3 > Buckets > petcuddleotron4442

petcuddleotron4442

Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
index.html	html	December 19, 2024, 19:32:38 (UTC-06:00)	2.9 KB	Standard
main.css	css	December 19, 2024, 19:32:38 (UTC-06:00)	970.0 B	Standard
serverless.js	js	December 19, 2024, 19:32:39 (UTC-06:00)	1.7 KB	Standard
whiskers.png	png	December 19, 2024, 19:32:37 (UTC-06:00)	490.5 KB	Standard


petcuddleotron4442.s3-website-us-east-1.amazonaws.com

Pet Cuddle-O-Tron

^^ Whiskers Commands You!! ^^

These are always needed:

How long until the next cuddle (in seconds) ...

Message for the human minion (Required)

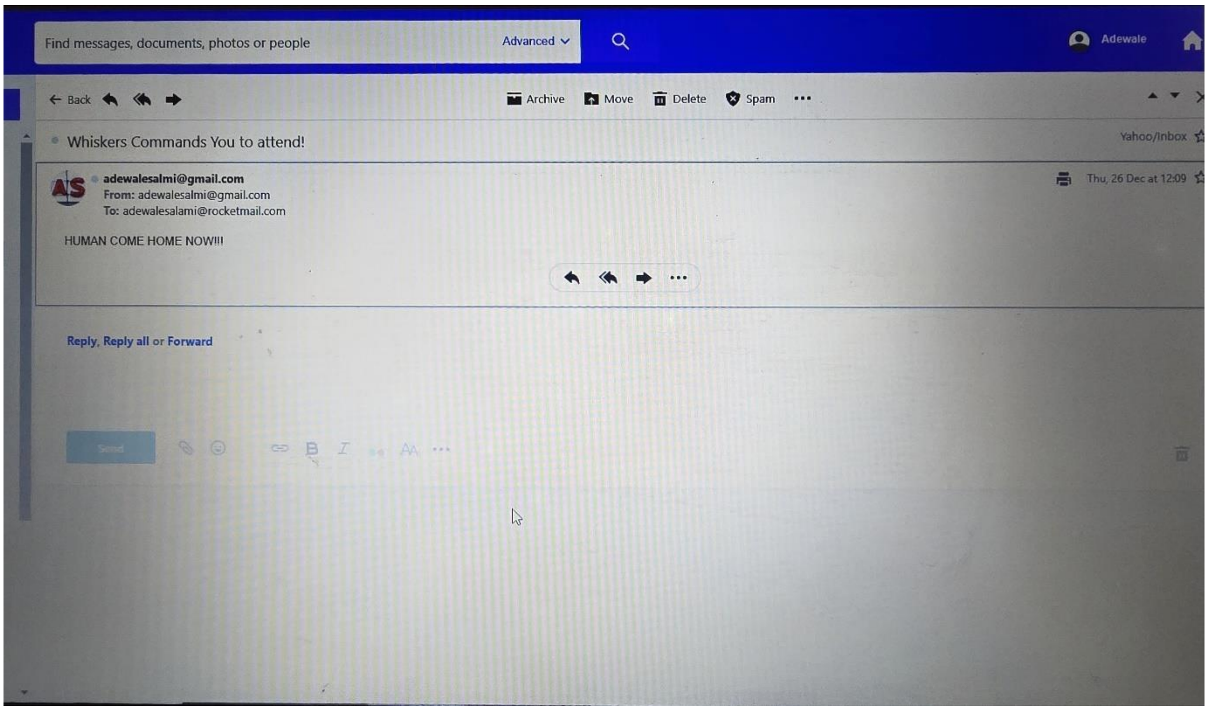
Email address is required to send the human an email

human@minion.com

Ok to test the application Enter an amount of time until the next cuddle. I suggest 120 seconds Enter a message; I Suggest HUMAN COME HOME NOW!!!

then enter the PetCuddleOTron Customer Address in the email box, this is the email which you verified right at the start as the customer for this application.

then click on Email Minion Button to send an email.


Whiskers Commands You to attend!

adewalesalmi@gmail.com
From: adewalesalmi@gmail.com
To: adewalesalami@rocketmail.com

Thu, 26 Dec at 12:09

HUMAN COME HOME NOW!!!

Completion of Stage 5

The front end is now accessible and communicates with the API Gateway.

Remember to delete all AWS resources created during this demo to avoid incurring charges.