Overview

This API provides secure access to a resource using AWS Cognito for user authentication. Users must log in via Cognito to obtain a token, which is then used to authorize API requests.

Endpoints

1. User Login

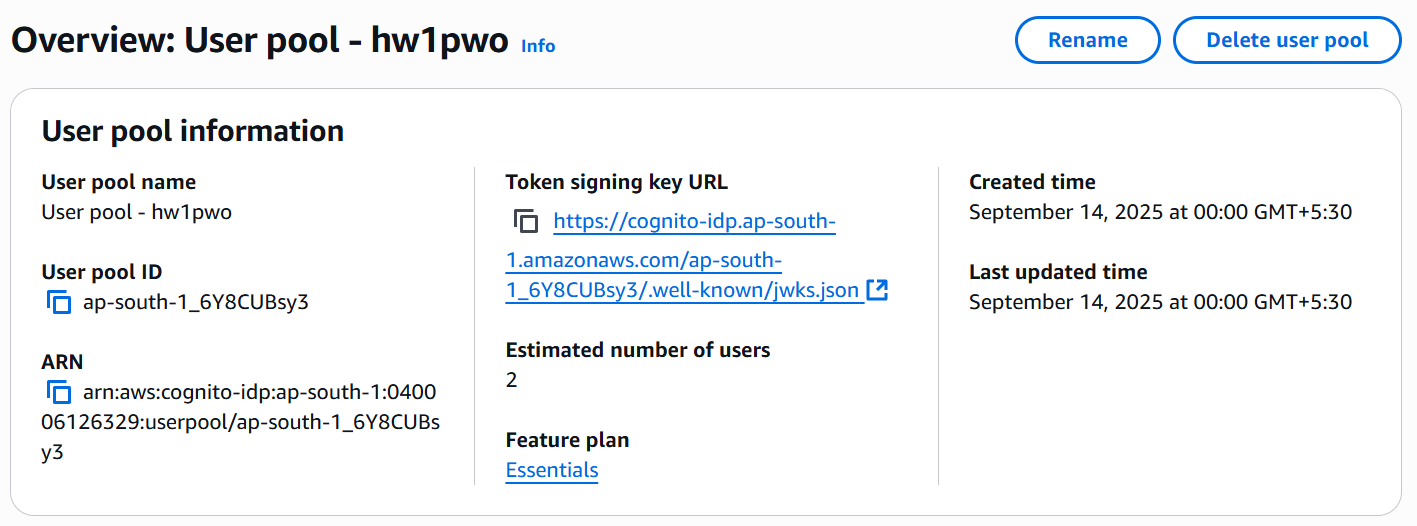
URL: [Sign-in](https://ap-south-16y8cubsy3.auth.ap-south-1.amazoncognito.com/login?client_id=3r3lhge9oulner2ju2rlsuptfs&redirect_uri=https://d84l1y8p4kdic.cloudfront.net&response_type=code&scope=email+openid+phone)

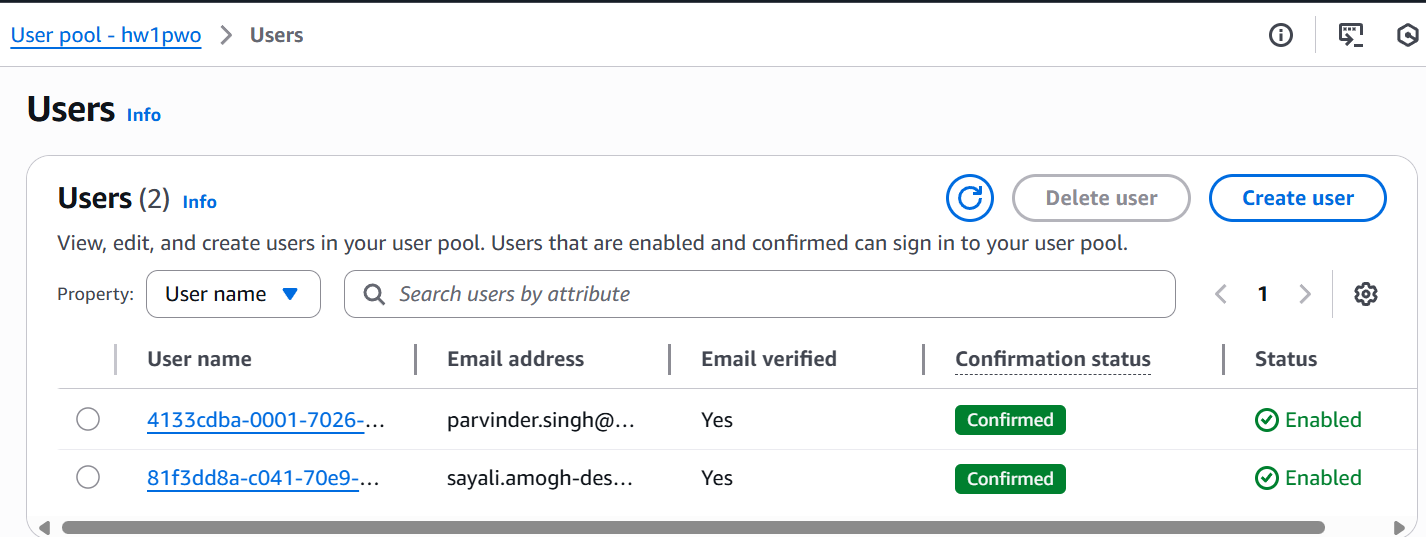
Description: Signup user, authenticates, retrieves a Cognito ID token and add entry in cognito user pool.

A screenshot of a login form

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.





2. API Gateway Lambda Function

Create a Lambda function to handle secure resource requests.

Python import json

def lambda\_handler(event, context):

# Extract token from Authorization header

token = event['headers'].get('Authorization', '').split(' ')[1]

# Validate token (this is a simplified example)

if not token:

return {

"statusCode": 401,

"body": json.dumps({"message": "Unauthorized"})

}

# Return secure resource

return {

"statusCode": 200,

"body": json.dumps({

"message": "This is a secure resource",

"data": {"example": "value"}

})

}

3. API Gateway

Use AWS API Gateway to create a REST API.

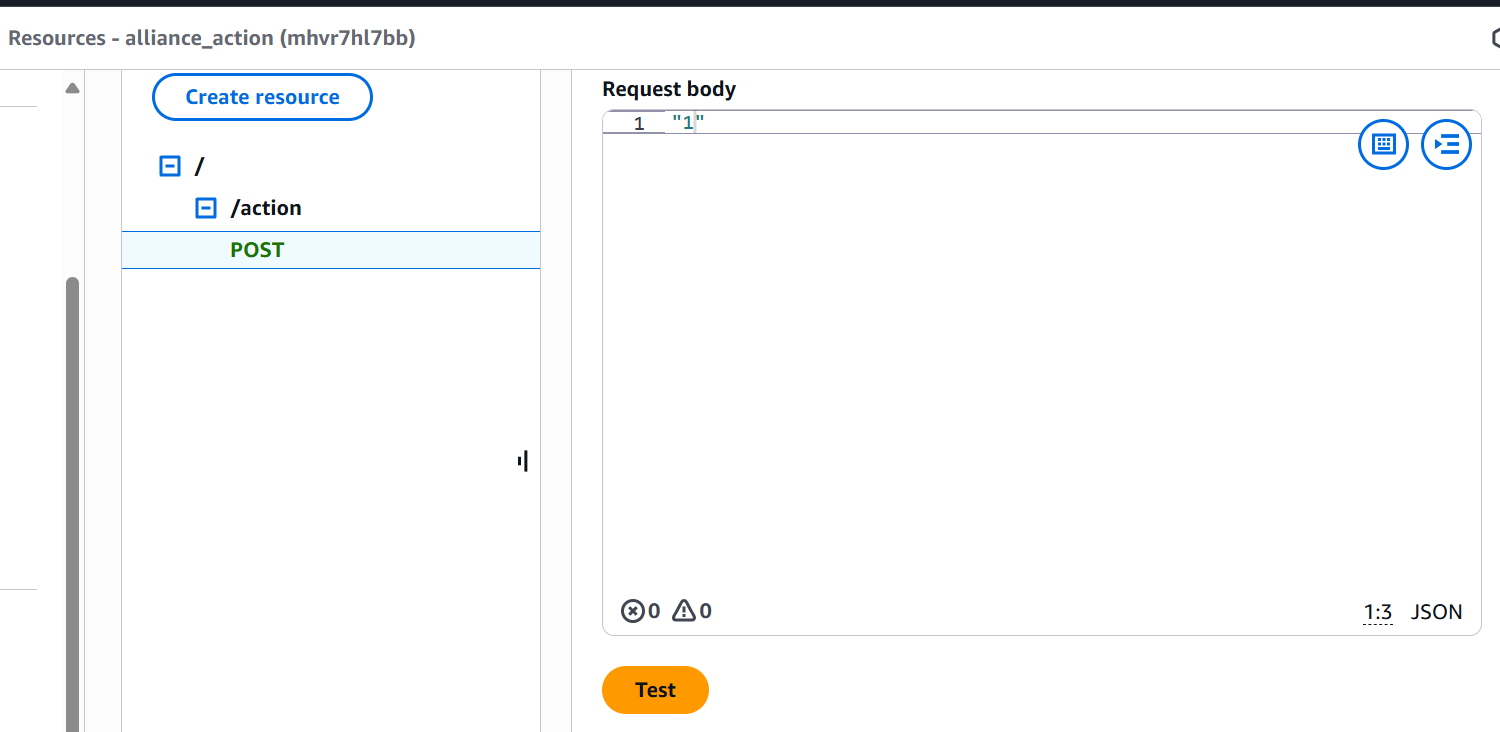
Configure Cognito User Pool Authorizer for the /secure-resource endpoint.

Attach the Lambda function to the /secure-resource endpoint.

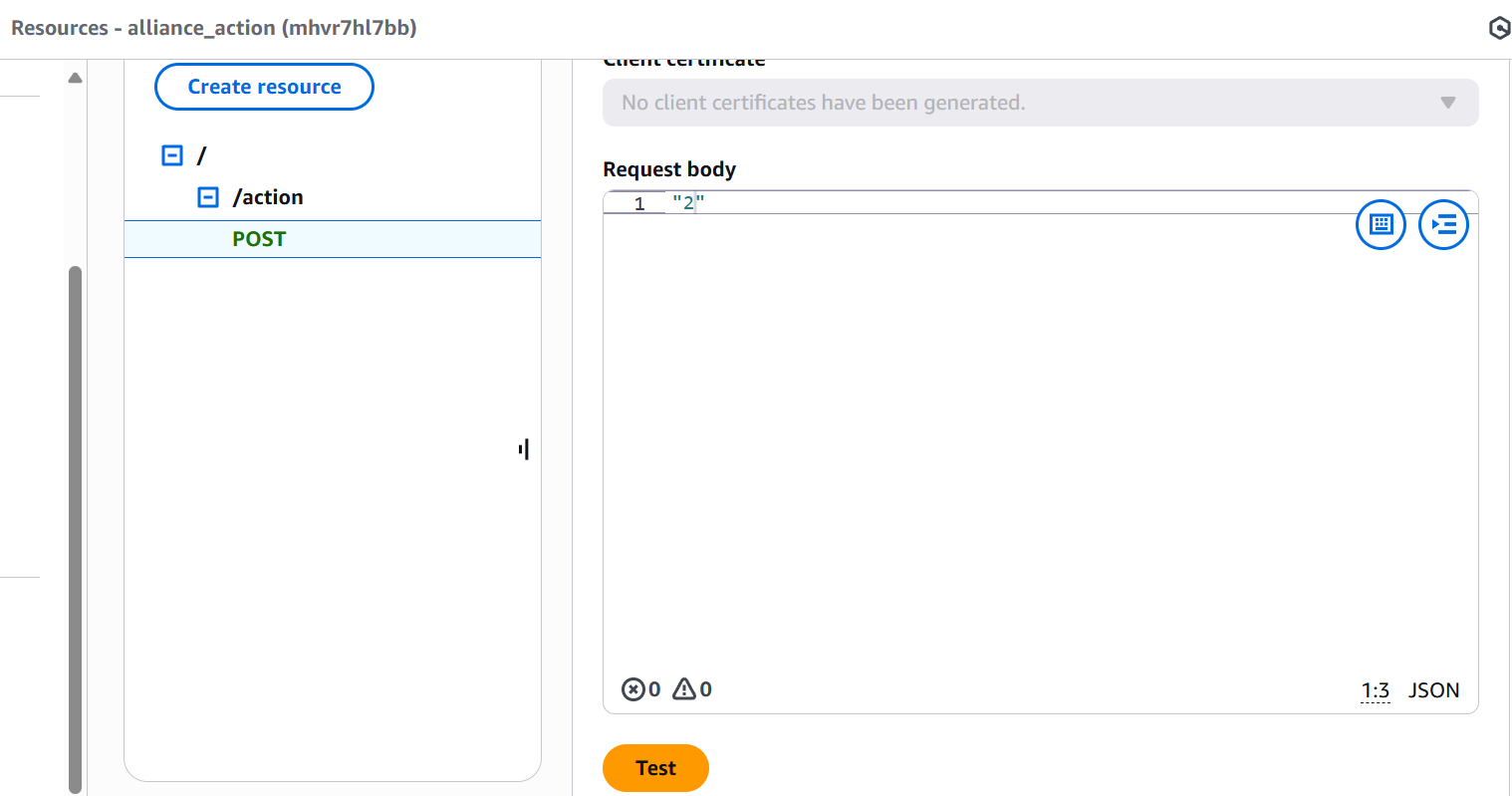
This method will not require separate lambda authorizer and validates user using cognito authorizer.

Testing the API:

For AWS we can send action event as “1” for create and start instance and “2” for stop and delete instance.



If testing from console Auth token is called internally from cognito if authorizer is attached or we need to write separate lambda authorizer.



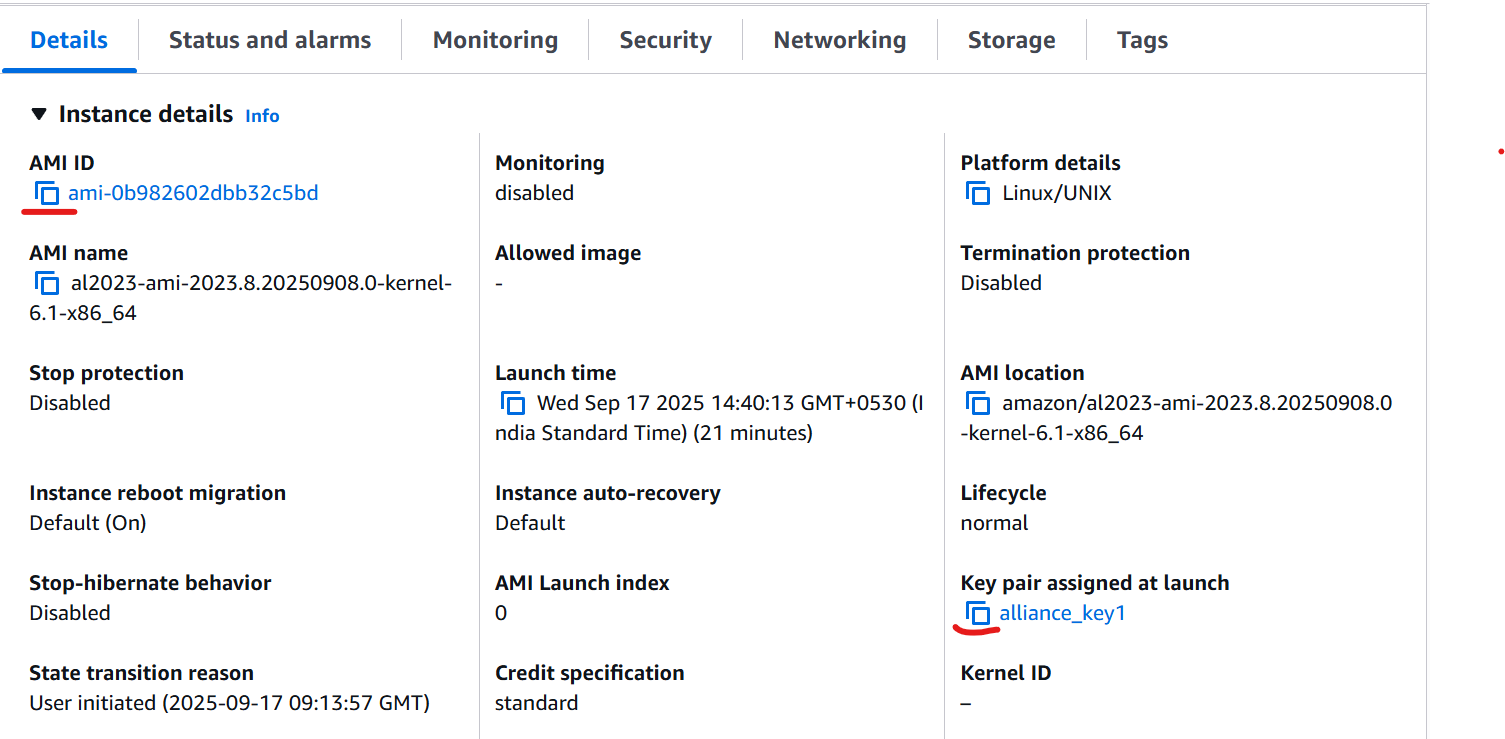
Login: Send a POST request to /login with valid credentials to retrieve the id\_token.

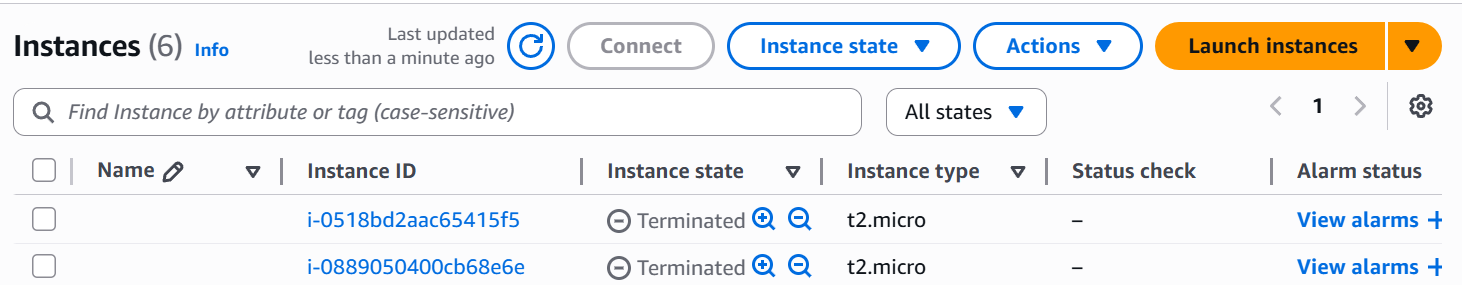
Access Secure Resource:

Use the id\_token in the Authorization header to access /secure-resource.

This setup ensures that only authenticated users can access the secure resource, leveraging AWS Cognito for robust user management and authentication.

We can see instances created and terminated in EC2 console with instance details.





Cloudwatch Logs:

We can see logs generated with key details, instance id, ip address and ssh command.

User must be asked to store this key with keyname mentioned in ssh command or can be asked to choose key name and lambda function can be modified accordingly.

In Current code key pair is generated using random numbers as id with fix header.

