AWS Cognito Set-up
Duke TBI

AWS Cognito Username Setup

Run cognito_aws_account_setup.py script attached in the email.
Install flask and authlib packages using the terminal:

```
pip install Flask authlib
```

Run the script:

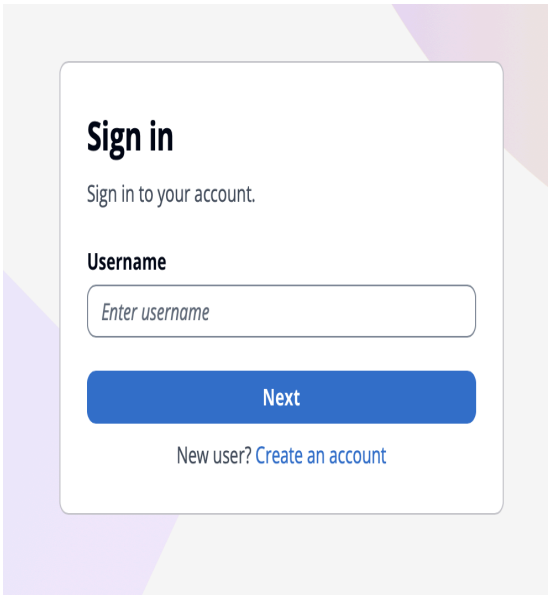
```
Python cognito_aws_account_setup.py
```

python cognito_aws_account_setup.py

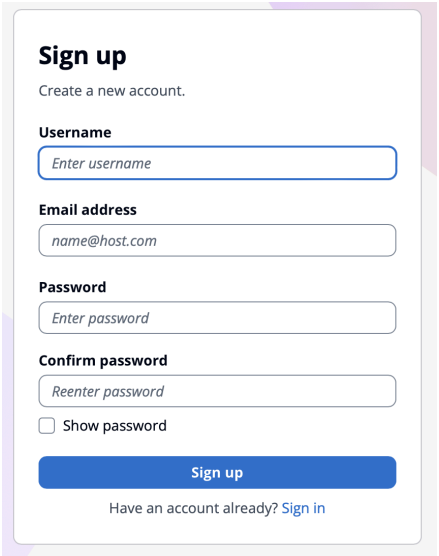
This should open a localhost app. Click on the URL and this should be the window you see:

Welcome! Please Login.

Upon clicking Login, you should get one of two screens



or



If you get the left screen, then please click create an account and you should automatically reach the right screen.

Please create a username, use the email address of your AWS account, create a new password and press sign up.

Please remember the username and the password. That is what we will be using in the next steps. Upon signing up, you should have gotten this landing page. If you see this, you have successfully created a username and password!

Successfully signed in

This is the default redirect page for Amazon Cognito user pools.

You're seeing this page because your Amazon Cognito app client doesn't have a return URL set.

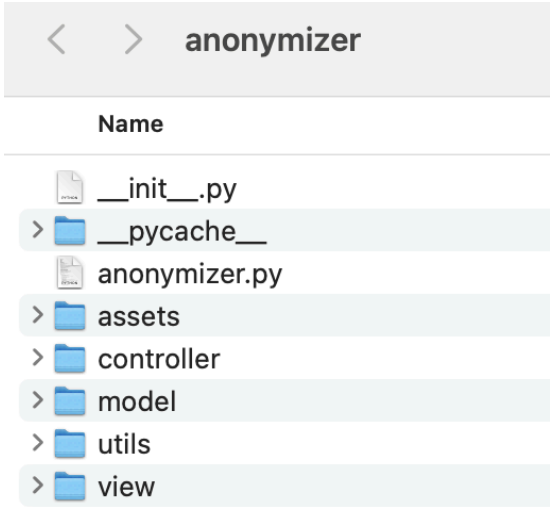
Anonymizer - AWS Integration

Changes to the Anonymizer app

We have to add a secret hash key into the Anonymizer python script for the authentication part. Hence,

Locate the Anonymizer package codebase: a folder named “anonymizer”
Usually in Windows, the folder would be located in these filepaths:
`C:\Users\<YourUserName>\.conda\pkgs (conda installed)`
Or
`C:\Program Files\Python\PythonXX\Lib\site-packages (not conda installed)`

The folder should have these files:



Open the controller folder and open the project.py file.

After all the import statements, add these new import statements and function:

```
import hmac
import hashlib
import base64
def get_secret_hash(username, client_id, client_secret):
    message = username + client_id
    dig = hmac.new(
        key=client_secret.encode('utf-8'),
        msg=message.encode('utf-8'),
        digestmod=hashlib.sha256
    ).digest()
    return base64.b64encode(dig).decode()
```

Find the `cognito_idp_client = boto3.client("cognito-idp", region_name=self.model.aws_cognito.region_name)` line of code and right after it, add this code chunk:

```
            secret_hash = get_secret_hash(username=self.model.aws_cognito.username,
                client_id=self.model.aws_cognito.app_client_id,
                client_secret="[redacted]")
```

Immediately after, you should see

```
            response = cognito_idp_client.initiate_auth(
                ClientId=self.model.aws_cognito.app_client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": self.model.aws_cognito.username,
                    "PASSWORD": self.model.aws_cognito.password,
                },
            )
```

Add the secret_hash as another parameter to AuthParameters, as shown below:

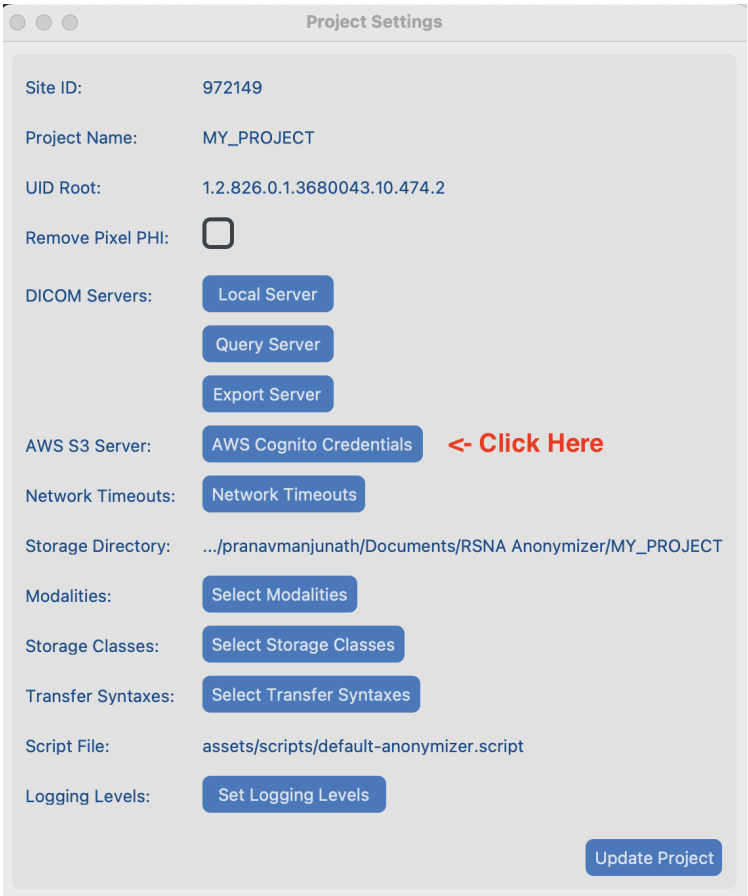
```
            response = cognito_idp_client.initiate_auth(
                ClientId=self.model.aws_cognito.app_client_id,
                AuthFlow="USER_PASSWORD_AUTH",
                AuthParameters={
                    "USERNAME": self.model.aws_cognito.username,
                    "PASSWORD": self.model.aws_cognito.password,
                    "SECRET_HASH":secret_hash,
                },
            )
```

Find this line of code: `user_attribute_1 = response["UserAttributes"][0]`
And change it to `user_attribute_1 = response["UserAttributes"][2]`

Save the changes made to the script. Now load the Anonymizer GUI.

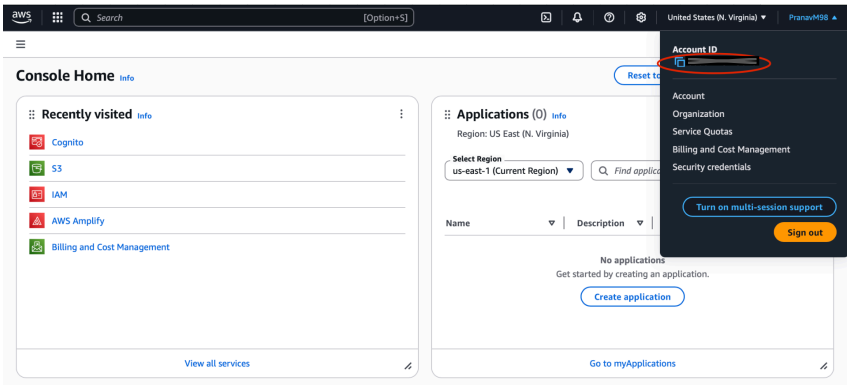
Anonymizer GUI

When creating a new project, go to project settings and Click on AWS Cognito Credentials



And use these credentials:

AWS Account ID: Usually found here (circled):



Region Name: us-east-1

Cognito Application Client ID: [redacted]
Cognito User Pool ID: [redacted]
Cognito Identity Pool ID: [redacted]
S3 bucket: amplify-cognitioapp-dev-fc8c3-deployment
Username: Use username created in step 1
Password: Use password created in step 1

Check Export to AWS and press Ok

It should look like this:

Continue to upload the DICOM files and press export and it should work! 😀