S3 - Bucket Setup

Go to AWS console and search for S3

1. Click **Create Bucket**
   Use the following options
   - General Purpose
   - Name the bucket descriptively—abide by these naming rules
   - [optional] Copy Settings from existing buckets already created

     OR

   - Object Ownership – ACLs Disabled
   - Public Access - Block All Public Access
   - Bucket Versioning – Disabled
   - Default Encryption -- Server-side encryption with Amazon S3 managed keys (SSE-S3)
   - Bucket Key – Enabled
   - Advanced Settings: Object Lock – Disabled

If you have an instance outside of the Kubeflow cluster that you want to give S3 and secrets access to, you need to create a new policy in IAM Roles -> follow the S3 - Set up Permissions documentation

This is an example code snippet of how to programmatically connect to the S3 bucket

```
import boto3
import os

# Set Bucket Name and Role ARN
bucket_name = "rag-llm-docs"
role_arn = 'arn:aws:iam:awsaccountnumber:role/aws-s3-access'

# Create an STS client
sts_client = boto3.client('sts')

# Assume the role
response = sts_client.assume_role(RoleArn=role_arn,
RoleSessionName='AssumeRoleSession')
```

```
credentials = response['Credentials']

# Configure AWS SDK with temporary credentials
s3_client = boto3.client('s3',

aws_access_key_id=credentials['AccessKeyId'],

aws_secret_access_key=credentials['SecretAccessKey'],

aws_session_token=credentials['SessionToken'])

def upload_folder_to_s3(local_folder, bucket_name, s3_client):
     for root, dirs, files in os.walk(local_folder):
     for file in files:
            local_path = os.path.join(root, file)
            relative_path = os.path.relpath(local_path,
local_folder)
           s3_path = relative_path.replace("\\", "/")  # Ensure
S3 path uses forward slashes
           try:
                s3_client.upload_file(local_path, bucket_name,
s3_path)
                print(f'Successfully uploaded {local_path} to
s3://{bucket_name}/{s3_path}')
           except Exception as e:
                print(f'Failed to upload {local_path} to
s3://{bucket_name}/{s3_path}: {e}')

# Replace this with local folder w/ files
local_folder = 'path/to/your/local/folder'

# Call the function to upload the folder
upload_folder_to_s3(local_folder, bucket_name, s3_client)
```