Setting up a local development environment for an **AWS Cloud Development course** is essential for students to practice and experiment with AWS services without incurring unnecessary costs. This guide covers the installation of necessary tools, configuration of AWS credentials, and setting up a local development environment.

**Self-Paced Setup Document for AWS Cloud Development Course**

**Prerequisites**

1. **Operating System**: Windows, macOS, or Linux.
2. **Internet Connection**: Required for downloading tools and accessing AWS services.
3. **AWS Account**: Students should create a free-tier AWS account at <https://aws.amazon.com/>.

**Step 1: Install Required Tools**

**1. Install Python**

Python is widely used for AWS development, especially with AWS SDKs (Boto3).

1. Download Python from <https://www.python.org/downloads/>.
2. During installation, check the box to **Add Python to PATH**.
3. Verify the installation:

python --version

**2. Install Node.js (Optional)**

If the course involves AWS Lambda or serverless development with Node.js:

1. Download Node.js from <https://nodejs.org/>.
2. Verify the installation:

node --version

npm --version

**3. Install AWS CLI**

The AWS Command Line Interface (CLI) is essential for interacting with AWS services.

1. Download and install the AWS CLI:
   * Windows: Use the MSI installer from <https://aws.amazon.com/cli/>.
   * macOS/Linux: Follow the instructions at <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>.
2. Verify the installation:

aws --version

**4. Install Docker (Optional)**

Docker is useful for running AWS services locally (e.g., LocalStack) or containerized applications.

1. Download Docker from <https://www.docker.com/products/docker-desktop>.
2. Verify the installation:

docker --version

**5. Install Visual Studio Code (VS Code)**

VS Code is a lightweight and powerful code editor.

1. Download VS Code from <https://code.visualstudio.com/>.
2. Install recommended extensions:
   * Python
   * AWS Toolkit
   * Docker
   * ESLint (for Node.js)

**Step 2: Set Up AWS Credentials**

**1. Create an IAM User**

1. Log in to the AWS Management Console.
2. Go to the **IAM** service.
3. Create a new user with **Programmatic Access**.
4. Attach the **AdministratorAccess** policy (for learning purposes; restrict permissions in production).
5. Save the **Access Key ID** and **Secret Access Key**.

**2. Configure AWS CLI**

1. Open a terminal or command prompt.
2. Run the following command and provide the required details:

aws configure

* + **AWS Access Key ID**: Paste the Access Key ID.
  + **AWS Secret Access Key**: Paste the Secret Access Key.
  + **Default Region Name**: Enter your preferred region (e.g., us-east-1).
  + **Default Output Format**: Enter json.

1. Verify the configuration:

aws sts get-caller-identity

**Step 3: Set Up Local AWS Development Tools**

**1. Install LocalStack (Optional)**

LocalStack allows you to run AWS services locally for testing.

1. Install LocalStack using Docker:

docker run -d -p 4566:4566 localstack/localstack

1. Verify it’s running:

curl http://localhost:4566/health

**2. Install AWS SAM CLI (Optional)**

AWS Serverless Application Model (SAM) CLI is used for serverless development.

1. Install AWS SAM CLI:
   * Windows: Use the installer from <https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/install-sam-cli.html>.
   * macOS/Linux: Use Homebrew:

brew tap aws/tap

brew install aws-sam-cli

1. Verify the installation:

sam --version

**Step 4: Set Up Version Control**

**1. Install Git**

1. Download Git from <https://git-scm.com/>.
2. Verify the installation:

git --version

**2. Configure Git**

1. Set your username and email:

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

**Step 5: Practice AWS Development**

**1. Create a Simple Python Script with Boto3**

1. Install Boto3:

pip install boto3

1. Create a Python script (example.py) to list S3 buckets:

python

import boto3

s3 = boto3.client('s3')

response = s3.list\_buckets()

print("S3 Buckets:")

for bucket in response['Buckets']:

print(bucket['Name'])

1. Run the script:

python example.py

**2. Deploy a Serverless Application (Optional)**

1. Use AWS SAM CLI to create a sample application:

sam init

1. Follow the prompts to create a "Hello World" Lambda function.
2. Deploy the application:

sam deploy --guided

**Step 6: Additional Resources**

1. **AWS Free Tier**: <https://aws.amazon.com/free/>
2. **AWS Documentation**: <https://docs.aws.amazon.com/>
3. **AWS Educate**: <https://aws.amazon.com/education/awseducate/>

**Troubleshooting Tips**

1. **AWS CLI Errors**: Ensure your credentials are correct and have the necessary permissions.
2. **LocalStack Issues**: Check if Docker is running and LocalStack is properly configured.
3. **Python Errors**: Verify Python and Boto3 are installed correctly.