Prerequisites & Setup Guide — aws-nasmatchstockprofile-mcp

1) Scope & Audience

This guide walks you through setting up a brand-new macOS laptop (no prior tooling) to develop, deploy, and operate the aws-nasmatchstockprofile-mcp project. It covers AWS account setup, Node.js & CDK, GitHub SSH, Docker (for CDK bundling), Bedrock model access, and end-to-end deployment & validation.

2) System Requirements (macOS)

- Admin user on macOS - Internet access - A GitHub account with repo access - An AWS account (you can use an IAM user or IAM Identity Center/SSO)

3) Install Homebrew & Core Tools

Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Command Line Tools (git, jq, unzip, wget)

```
brew install git jq unzip wget
```

Docker Desktop (required for CDK bundling of Python Lambdas)

Download & install from Docker (open and leave it running). Verify:

```
docker info
```

4) Install Node.js & AWS CDK

Install nvm & Node LTS

```
brew install nvm && mkdir -p ~/.nvm
```

Add to shell (zsh):

```
echo 'export NVM_DIR="$HOME/.nvm"' >> ~/.zshrc
echo '[ -s "/opt/homebrew/opt/nvm/nvm.sh" ] && . "/opt/homebrew/opt/nvm/nvm.sh"' >> ~/.zshrc
source ~/.zshrc && nvm install --lts && nvm use --lts
```

Global CDK (optional; repo also has local CDK)

```
npm i -g aws-cdk@2
```

5) GitHub SSH Setup (push/pull)

Generate SSH key

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Press Enter to accept defaults, set a passphrase if desired.

Add key to ssh-agent

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

Add public key to GitHub

`pbcopy < ~/.ssh/id_ed25519.pub` (paste into GitHub → Settings → SSH and GPG keys → New SSH key)

Test
`ssh -T git@github.com` (type 'yes' if prompted)

6) Clone the Repository

```
git clone git@github.com:aburmd/aws-nasmatchstockprofile-mcp.git
cd aws-nasmatchstockprofile-mcp
```

`npm install` (or `npm ci` if package-lock.json exists)

7) Install & Configure AWS CLI

Install

```
brew install awscli
```

Option A — IAM user access keys

```
aws configure
```

Enter Access Key ID, Secret, region `us-east-1`, output json.

Option B — IAM Identity Center (SSO)

```
aws configure sso
```

Follow prompts, choose account and role, set region `us-east-1`.

Verify

```
aws sts get-caller-identity
```

8) CDK Bootstrap (first time per account/region)

CDK needs a bootstrap stack for assets and roles.

```
cdk bootstrap aws://<ACCOUNT_ID>/us-east-1
```

9) Amazon Bedrock Model Access & IAM

In AWS Console (Region: N. Virginia / us-east-1)
Amazon Bedrock → Model access → Enable:
• Amazon Titan Embeddings G1 – Text (modelId: *amazon.titan-embed-text-v1*)
• Anthropic Claude 3.5 Sonnet (modelId: *anthropic.claude-3-5-sonnet-20240620-v1:0*)

IAM permissions (example least privilege)

```
{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Action":["bedrock:InvokeModel","bedrock:InvokeModelWithResponseStream"],"Resource":["arn:aws:bedrock:us-east-1::foundation-model/amazon.titan-embed-text-v1","arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-3-5-sonnet-20240620-v1:0"]}]}
```

10) Build & Deploy

```
npm run build
npx cdk deploy BaseInfra
```

Outputs will include the S3 bucket name, Lambda function names, and WebSocket URL.

11) Prepare Input Files (S3)

Upload your Excel template and CSV (example):

```
aws s3 cp ./nasmatch-portfolio.xlsx
s3://<BucketFromOutputs>/source/nasmatch-portfolio.xlsx
```

```
aws s3 cp ./Portfolio_Positions_Aug-21-2025.csv
s3://<BucketFromOutputs>/source/Portfolio_Positions_Aug-21-2025.csv
```

12) Invoke & Validate

Invoke

```
aws lambda invoke --function-name <ExcelProcessorFnName> --cli-binary-format
raw-in-base64-out --payload '{"source_key":"source/Portfolio_Positions_Aug-21-2025.csv",
"target_key":"source/nasmatch-portfolio.xlsx","output_key":"output/nasmatch-portfolio-up
dated.xlsx"}' /tmp/out.json && cat /tmp/out.json
```

Fetch run report

```
BUCKET=$(jq -r .bucket /tmp/out.json)
REPORT=$(jq -r .report_key /tmp/out.json)
aws s3 cp "s3://$BUCKET/$REPORT" ./run-report.json && cat ./run-report.json
```

Download updated workbook

```
OUT=$(jq -r .output_key /tmp/out.json)
aws s3 cp "s3://$BUCKET/$OUT" ./nasmatch-portfolio-updated.xlsx
```

13) MCP Tool Test (WebSocket)

```
npm i -g wscat
wscat -c wss://<id>.execute-api.us-east-1.amazonaws.com/prod -x '{"type":"call_tool","to
ol":"process_excel","args":{"source_key":"source/Portfolio_Positions_Aug-21-2025.csv","t
arget_key":"source/nasmatch-portfolio.xlsx","output_key":"output/nasmatch-portfolio-upda
ted.xlsx"},"request_id":"p1"}'
```

14) Troubleshooting Cheatsheet

- **Docker bundling fails**: Open Docker Desktop and retry `npm run build`.
- **AccessDenied on Bedrock**: Ensure Model access is enabled in us-east-1 and IAM policy allows the exact model ARNs.
- **per_ticker_writes = 0**: Check run-report.json → missing_accounts; fix mapping via DDB or env *ACCOUNT_NAME_MAP_JSON*.
- **Wrong rows**: Set env *ROW_QTY*/*ROW_COST* in CDK env for the Lambda.
- **Private VPC**: Add Interface VPC Endpoint for *com.amazonaws.us-east-1.bedrock-runtime*.

15) Teardown

```
npx cdk destroy BaseInfra
```

Note: S3 bucket is set to RETAIN for safety; empty it before destroy if needed.

16) Security Notes

- Prefer IAM Identity Center (SSO) for human access over static keys.
- Store no secrets in git. Use AWS-managed encryption (KMS) in this stack.
- Grant least privilege for Bedrock models; restrict to needed model ARNs.