How to use SSO with AWS CLI

DevX environment uses AWS SSO as the primary authentication method and it is more secure because of the MFA. Use of IAM Users with AWS ACCESS KEYS is not permitted in DevX environment. In that context, this guide shows how to use SSO with AWS CLI (or any AWS SDK).

**Prerequisites**

1. [Install AWS Cli](#)

Step-by-step guide

1. On a Mac/Linux terminal or Windows Powershell/ComandPrompt run the ' `aws configure sso` ' command to [configure CLI with SSO](#)

2. Enter the SSO start URL 'https://devx-systems.awsapps.com/start '

3. Enter the SSO region 'us-east-2'

4. At this point, a browser will be opened with the SSO start URL for you login and ask you to 'allow' the sign-in to CLI.

5. Once allowed, the CLI will prompt you to select one AWS account from the list of AWS Accounts authorized for your SSO login. Please select the account that you need to connect to from AWS CLI.

6. CLI will next prompt you to select a IAM role. Please select the one that is appropriate for the AWS CLI.

7. Set the client region and output format as you desire and set a aws profile name (preferably, <account-name>-<role-name>, say ion-dev-admin) in the next prompt(s).

8. The terminal screen might look like below for the above steps. Please do not provide SSO session name if you intend to use IntelliJ or Terraform with s3 backend. We noticed that the IntelliJ or Terraform do not recognize the SSO session names yet.

   a.
   ```
   1  › aws configure sso
   2  SSO session name (Recommended): devx
   3  SSO start URL [None]: https://devx-systems.awsapps.com/start
   4  SSO region [None]: us-east-2
   5  SSO registration scopes [sso:account:access]:
   6  Attempting to automatically open the SSO authorization page in your default browser.
   7  If the browser does not open, open the following URL:
   8
   9  https://oidc.us-east-2.amazonaws.com/authorize?
      response_type=code&client_id=SqasHaqYHEn2wmkz1oEHQ3VzLWVhc3QtMg&redirect_uri=http%3A%2
   ```

```
        F%2F127.0.0.1%3A62663%2Foauth%2Fcallback&state=232e5ead-7377-4f75-96d0-
        cef3d81c3625&code_challenge_method=S256&scopes=sso%3Aaccount%3Aaccess&code_challenge=q
        q_0DG1PdCCVztaYaRAJ-RLbVWj7avU_6uVCdPY97qs
 10  There are 58 AWS accounts available to you.
 11  Using the account ID 019847571497
 12  There are 3 roles available to you.
 13  Using the role name "devx-administrator-access"
 14  Default client Region [None]:
 15  CLI default output format (json if not specified) [None]:
 16  To use this profile, specify the profile name using --profile, as shown:
 17
 18  aws sts get-caller-identity --profile tri11
```

If you have already logged onto https://myapps.microsoft.com with your DevX credentials, your browser must sign you in on AWS Access Portal automatically. It might ask for permission like below to complete the AWS CLI SSO configuration.
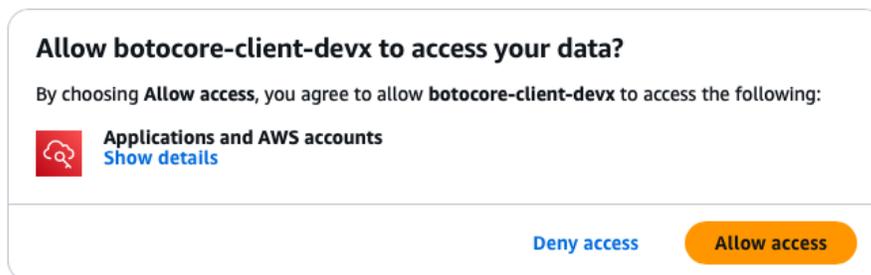
9. After this step your ~/.aws/config file should have the following profile configured

   a. > cat ~/.aws/config

```
 1  [profile tri11]
 2  sso_session = devx
 3  sso_account_id = 507566090046
 4  sso_role_name = devx-administrator-access
 5  [sso-session devx]
 6  sso_start_url = https://devx-systems.awsapps.com/start
 7  sso_region = us-east-2
 8  sso_registration_scopes = sso:account:access
```

10. To use this profile, specify the profile name using `--profile`

   a. Example:

```
 1  aws s3 ls --profile tri11
```

11. To login and logout:

```
 1  aws sso login --sso-session devx
```

```
2   aws sso logout
```

Troubleshooting

1. If AWS CLI is not authenticated to an AWS account with appropriate role, it is easier to delete the profile(s) and recreate. In some observed cases, the profiles had case-mismatch but the errors were misleading.

ⓘ

Related articles

- 📄 Organization wide backup solution
- 📄 Backup Policy Implementation
- 📄 Upgrading Control Tower
- 📄 Restoring Data from the Backups
- 📄 AWS S3 Backup Management