

User Auth via Script and CSV - AWS CloudShell

Introduction

This Troubleshooting Guide (TSG) walks you through the process of configuring AWS authentication in CloudShell using an Access Key CSV and an automated script. Follow the steps carefully to avoid common authentication issues.

Step 1: Create IAM User & Access Key

1. Sign into the AWS Management Console.
2. Navigate to IAM -> Users -> Create user.
3. Assign the user a meaningful name (e.g., 'terraform-user').
4. ~~Under 'Access type,' enable 'Programmatic access'.~~ This is actually done elsewhere now.
5. Click 'Create access key' and download the .CSV file.

If your CSV file does not contain an 'Access key ID' and 'Secret access key', you may have downloaded the wrong file (such as an IAM login credentials CSV).

Step 2: Upload CSV & Script to CloudShell

1. Open AWS CloudShell.
2. Click 'Actions' -> 'Upload file'.
3. Upload '(your CSV filename).csv' and '(your script filename).sh'.
4. Verify the files are present using:

```
ls -la
```

Step 3: Ensure Correct File Format (LF Instead of CRLF)

If you created your script on Windows (e.g., in VS Code), it may have Windows-style CRLF line endings, which cause execution failures.

To check for incorrect line endings, run:

```
cat -A (your script filename).sh
```

If you see '^M' at the end of lines, convert the file to Unix format:

```
sed -i 's/$//' (your script filename).sh
```

Then verify again.

Step 4: Make the Script Executable & Extract Credentials

1. Give your script execution permissions:

```
chmod +x (your script filename).sh
```

2. Open your script and ensure it contains:

```
#!/bin/bash
```

```
CSV_FILE="/home/cloudshell-user/(your CSV filename).csv"
```

```
ACCESS_KEY=$(awk -F',' 'NR==2 {print $1}' "$CSV_FILE")
```

```
SECRET_KEY=$(awk -F',' 'NR==2 {print $2}' "$CSV_FILE")
```

```
aws configure set aws_access_key_id "$ACCESS_KEY"
```

```
aws configure set aws_secret_access_key "$SECRET_KEY"
```

```
aws configure set region "us-east-2"
```

```
aws configure set output "json"
```

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

3. Run the script:

```
./(your script filename).sh
```

Step 5: Verify Authentication & Troubleshooting

Run:

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

You don't have to do this it's the last line in your script.

It'll automatically give you the below output after the script is finished running.

Expected output:

```
{
```

```
"UserId": "(your user ID)",
```

```
"Account": "(your AWS account ID)",
```

```
"Arn": "arn:aws:iam::(your account ID):user/terraform-user"
```

```
}
```

Common Errors & Fixes

InvalidClientTokenId: Credentials are incorrect.

- Ensure the correct CSV file was used.
- Manually check with `cat (your CSV filename).csv``.

Permission Denied when running script:

```
chmod +x (your script filename).sh
```

If aws configure is still failing, delete old credentials and rerun the script:

```
rm -rf ~/.aws/credentials
```

```
./(your script filename).sh
```

Conclusion

You have successfully set up AWS CLI authentication in CloudShell using an automated script and CSV file. This method ensures efficiency and avoids manual entry errors. Keep this guide for future reference!

Crafted and Edited:

Joseph Luster