DevOps笔记1  
  
To create an IAM user with AdministratorAccess using AWS CLI and save the Access Key ID and Secret Access Key, follow these steps:

**Step 1: Set Up AWS CLI**

Ensure you have AWS CLI installed and configured on your machine. If not, you can install it by following the instructions from the [official AWS CLI documentation](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html).

**Step 2: Configure AWS CLI**

Configure AWS CLI with existing credentials (if not already configured) using the command:

bash

复制代码

aws configure

Provide your existing AWS Access Key ID, Secret Access Key, default region name, and output format.

**Step 3: Create an IAM User**

Run the following command to create a new IAM user (replace YourUserName with your desired username):

bash

复制代码

aws iam create-user --user-name YourUserName

**Step 4: Attach AdministratorAccess Policy to the User**

Attach the AdministratorAccess policy to the newly created user:

bash

复制代码

aws iam attach-user-policy --user-name YourUserName --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

**Step 5: Create Access Keys for the IAM User**

Generate access keys for the user, which will be used for programmatic access:

bash

复制代码

aws iam create-access-key --user-name YourUserName

This command will return the Access Key ID and Secret Access Key. It’s crucial to save these keys securely, as you won’t be able to retrieve the Secret Access Key again once you navigate away.

**Step 6: Save the Access Key ID and Secret Access Key**

Here’s an example output from Step 5:

json

复制代码

{

"AccessKey": {

"UserName": "YourUserName",

"AccessKeyId": "AKIAEXAMPLE12345",

"Status": "Active",

"SecretAccessKey": "abc123SECRETKEYxyz789",

"CreateDate": "2024-09-26T12:34:56Z"

}

}

Make sure to save the AccessKeyId and SecretAccessKey securely, such as in a password manager or a secure notes application.

**Step 7: Verify Access**

You can verify your access by running a simple AWS CLI command, such as listing your current IAM users:

bash

复制代码

aws iam list-users

If you see the list of users including the one you just created, your new credentials and permissions are working correctly.

These steps will help you create an IAM user, provide administrator access, and securely save the access keys for future use.

4o

**您说：**

运行到了第二步，他问我要AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OLWE]: 我用的是root user登录的 如何获取Access key id？

**ChatGPT 说：**

ChatGPT

If you are logged in as the root user of your AWS account and need to obtain or create an Access Key ID and Secret Access Key, follow these steps. However, it's important to note that using root credentials for day-to-day activities, especially programmatic access, is highly discouraged due to security reasons. It’s safer to use IAM users with appropriate permissions.

**Steps to Obtain Access Key ID and Secret Access Key for the Root User:**

1. **Log in to AWS Management Console**:
   * Go to the [AWS Management Console](https://aws.amazon.com/console/) and log in using your root account credentials.
2. **Access Your Security Credentials**:
   * Click on your account name or the **account icon** at the top right corner of the console.
   * Select **Security Credentials** from the dropdown menu. This will direct you to the Security Credentials page.
3. **Manage Access Keys**:
   * Scroll down to the **Access keys for CLI, SDK, & API access** section.
   * You might already see an existing Access Key ID, but if you don't or want to create a new one, click on **Create New Access Key**.
   * AWS will generate a new Access Key ID and Secret Access Key.
4. **Download and Save the Keys**:
   * After creating a new access key, AWS will display the Access Key ID and Secret Access Key once.
   * Click **Download .csv file** to save these keys securely. Alternatively, copy them manually and store them in a safe location, such as a password manager.

**Important**: You won’t be able to see the Secret Access Key again after this point, so make sure to save it immediately.

1. **Configure AWS CLI**:
   * Now that you have your root user Access Key ID and Secret Access Key, you can use them to configure the AWS CLI:

bash

复制代码

aws configure

* + Enter the Access Key ID and Secret Access Key when prompted.

**Important Security Notice:**

While you can use root credentials for initial setup, it is highly recommended to create IAM users with specific permissions for regular use. Always avoid using the root account for programmatic access and daily tasks due to security risks.

Would you like to proceed with creating an IAM user instead to ensure safer practices?

4o

**您说：**

Default region name [us-east-1]: 回答什么？

**ChatGPT 说：**

ChatGPT

The **Default region name** refers to the AWS region you want to set as your default when using AWS CLI commands. This determines where your resources will be created if you don't specify a region explicitly in your commands.

**Common AWS Regions:**

Here are some common regions you might choose based on your location or preferences:

* **us-east-1** (N. Virginia) - Default for many services and widely used.
* **us-west-2** (Oregon) - Commonly used for services in the western U.S.
* **eu-west-1** (Ireland) - Popular European region.
* **ap-southeast-1** (Singapore) - Good for Southeast Asia.
* **ap-northeast-1** (Tokyo) - Popular for Japan and East Asia.
* **ap-southeast-2** (Sydney) - Suitable for Australia and New Zealand.
* **cn-north-1** (Beijing) - For AWS China region (requires specific permissions).

**How to Choose:**

* If you're not sure, you can leave it as **us-east-1**, which is widely used and has broad service availability.
* If you have specific regional requirements (e.g., for data residency), choose the region closest to your needs.

You can always change the region later by specifying it in your CLI commands or reconfiguring AWS CLI using aws configure.

Let me know if you have a specific region in mind or need help choosing one!

4o

**您说：**

{ "User": { "Path": "/", "UserName": "xiaofangzong", "UserId": "AIDAS2VS4W5Nixxxxxxx", "Arn": "arn:aws:iam::194722445146:user/xiaofangzong", "CreateDate": "2024-09-26T02:51:41+00:00" } } (END) 当屏幕上出现这些信息时，怎么返回terminal信息？

**ChatGPT 说：**

ChatGPT

When you see output like this in the terminal and you want to return to the command prompt, you are likely viewing the output using a pager (like less), which is the default behavior for some commands in the terminal. To exit this view and return to the terminal prompt, follow these steps:

**How to Exit the Pager and Return to the Terminal Prompt:**

1. **Press q:**  
   Simply press the q key on your keyboard. This command exits the pager and takes you back to the command line prompt.

This happens because some AWS CLI commands automatically pipe output to a pager if the output is long or structured, allowing you to scroll through the information easily. Exiting the pager with q will not close your terminal, just the current view.

4o

**您说：**

❯ aws iam attach-user-policy --user-name xiaofangzong --policy-arm arn:aws:iam::aws:policy/AdministratorAccess usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters] To see help text, you can run: aws help aws <command> help aws <command> <subcommand> help aws: error: the following arguments are required: --policy-arn 哪里出错了？

**ChatGPT 说：**

ChatGPT

The error occurred because of a typo in the command; specifically, you used --policy-arm instead of --policy-arn. Here's the corrected command:

**Correct Command:**

bash

复制代码

aws iam attach-user-policy --user-name xiaofangzong --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

**Explanation:**

* --policy-arn is the correct flag to specify the Amazon Resource Name (ARN) of the policy you want to attach to the user.
* Make sure the ARN is correct; it should be exactly as shown above for the AdministratorAccess policy.

After running the corrected command, you should not encounter the error.

4o

**您说：**

如何用CLI来check本地的configuration？

**ChatGPT 说：**

ChatGPT

To check your local AWS CLI configuration, you can use the following command:

bash

复制代码

aws configure list

**Explanation of the Output:**

This command will display the current configuration settings for your AWS CLI, including:

* **Access Key ID**: Shows which Access Key ID is currently in use.
* **Secret Access Key**: Displays if the Secret Access Key is configured (hidden for security reasons).
* **Region**: Displays the default AWS region configured.
* **Output format**: Shows the default output format (e.g., JSON, text, table).

**Additional Commands:**

1. **View Detailed Configuration Files**:
   * To see the configuration files being used, run:

bash

复制代码

aws configure list-profiles

* + This command shows the list of profiles configured in your AWS CLI.

1. **Check Specific Profile**:
   * If you have multiple profiles and want to check a specific one, use:

bash

复制代码

aws configure get <setting> --profile <profile-name>

* + Replace <setting> with region, aws\_access\_key\_id, aws\_secret\_access\_key, etc., and <profile-name> with the name of your profile.

1. **View Configuration Files Directly**:
   * AWS CLI configuration is stored in files located at:
     + **Credentials File**: ~/.aws/credentials
     + **Config File**: ~/.aws/config
   * You can view these files directly using a text editor or by using cat command:

bash

复制代码

cat ~/.aws/config

cat ~/.aws/credentials

These commands will help you verify and inspect your AWS CLI configuration settings.

4o

**您说：**

Authenticate to your IAM user on your command line. Below are commands to type on your terminal: export AWS\_REGION=ap-southeast-2 (or other regions you want to authenticate into) export AWS\_DEFAULT\_REGION=ap-southeast-2 (need to match with the one above) export AWS\_PROFILE=default (profile name need to match the one in your ~/.aws/config file) Verify your authentication using “aws s3 ls”. If you don't see any error message, you’re good to go! AWS looks for above environment variables in your shell to match your identity. Therefore, you need to do above steps (3 export commands) every time you opens a new terminal tab to authenticate with AWS. To make this easier, you can put these commands in your ~/.bashrc or ~/.zshrc config file. 这段是什么意思？

**ChatGPT 说：**

ChatGPT

This set of instructions explains how to authenticate your AWS CLI session using environment variables, specifically with your IAM user credentials, and how to make the process easier for future terminal sessions.

**Explanation of the Commands and Instructions:**

1. **Set Environment Variables**:
   * The commands provided set environment variables in your terminal session to specify the AWS region and profile you want to use:

bash

复制代码

export AWS\_REGION=ap-southeast-2

export AWS\_DEFAULT\_REGION=ap-southeast-2

export AWS\_PROFILE=default

* + **export AWS\_REGION=ap-southeast-2**: This sets the AWS region where your commands will be directed. You can change ap-southeast-2 (Sydney) to any region you prefer.
  + **export AWS\_DEFAULT\_REGION=ap-southeast-2**: Similar to the previous command, this sets the default region for AWS CLI operations. It should match the region set in the previous command.
  + **export AWS\_PROFILE=default**: This sets the AWS profile you want to use. Profiles are defined in your ~/.aws/config file, and default is the standard profile if no others are set.

1. **Verify Authentication**:
   * To ensure your authentication is correct, use the following command:

bash

复制代码

aws s3 ls

* + This command attempts to list your S3 buckets. If you do not receive any error messages, it indicates that your AWS CLI is correctly authenticated with your IAM user credentials.

1. **Environment Variables Persistence**:
   * AWS CLI uses these environment variables (AWS\_REGION, AWS\_DEFAULT\_REGION, and AWS\_PROFILE) to match your identity and configuration settings.
   * **Persistence Note**: Since environment variables are session-specific, you need to set them each time you open a new terminal. To make this easier, you can add the export commands to your shell configuration file, such as ~/.bashrc or ~/.zshrc.
2. **Automating with Shell Config Files**:
   * Add the three export commands to your ~/.bashrc (for Bash users) or ~/.zshrc (for Zsh users) file to automate the process. This way, every time you open a terminal, these variables will be set automatically:

bash

复制代码

echo 'export AWS\_REGION=ap-southeast-2' >> ~/.bashrc

echo 'export AWS\_DEFAULT\_REGION=ap-southeast-2' >> ~/.bashrc

echo 'export AWS\_PROFILE=default' >> ~/.bashrc

# For Zsh users, replace ~/.bashrc with ~/.zshrc

* + After adding these lines, either restart your terminal or run source ~/.bashrc (or source ~/.zshrc for Zsh) to apply the changes.

**Purpose:**

This approach makes it easier to authenticate and set up your AWS CLI environment consistently without needing to manually set these variables every time you start a new terminal session.

4o

**您说：**

echo 'export AWS\_REGION=ap-southeast-2' >> ~/.bashrc echo 'export AWS\_DEFAULT\_REGION=ap-southeast-2' >> ~/.bashrc echo 'export AWS\_PROFILE=default' >> ~/.bashrc 这是什么意思 用中文回答

**ChatGPT 说：**

ChatGPT

这些命令的意思是将 AWS CLI 的环境变量设置命令添加到你的 ~/.bashrc 文件中，这样每次打开一个新的终端时，这些环境变量就会自动设置，无需每次手动输入。具体解释如下：

1. **echo 'export AWS\_REGION=ap-southeast-2' >> ~/.bashrc**  
   这个命令将 export AWS\_REGION=ap-southeast-2 添加到 ~/.bashrc 文件的末尾。AWS\_REGION 是 AWS CLI 的一个环境变量，用于指定默认的 AWS 区域，这里设置为亚太地区的悉尼区域 (ap-southeast-2)。
2. **echo 'export AWS\_DEFAULT\_REGION=ap-southeast-2' >> ~/.bashrc**  
   这个命令将 export AWS\_DEFAULT\_REGION=ap-southeast-2 添加到 ~/.bashrc 文件的末尾。AWS\_DEFAULT\_REGION 也是用于指定默认的 AWS 区域，和上一个变量相同，确保默认区域设置为 ap-southeast-2。
3. **echo 'export AWS\_PROFILE=default' >> ~/.bashrc**  
   这个命令将 export AWS\_PROFILE=default 添加到 ~/.bashrc 文件的末尾。AWS\_PROFILE 用于指定 AWS CLI 使用的配置文件名称，这里设置为 default，即使用默认的配置文件。

**目的：**

这些命令将以上的环境变量写入 ~/.bashrc 文件中，当你启动一个新的 Bash 终端时，这些设置会自动加载，使得 AWS CLI 运行时自动使用这些配置。你不需要每次打开终端时手动设置这些变量，提高了操作的便捷性。

如果你使用的是 Zsh，可以将 ~/.bashrc 替换为 ~/.zshrc。配置完成后，你可以运行 source ~/.bashrc 来立即生效。

4o

**您说：**

如何检查环境变量已经设置成功

**ChatGPT 说：**

ChatGPT

要检查环境变量是否已经成功设置，你可以使用以下方法在终端中查看当前设置的环境变量：

**方法 1: 使用 echo 命令查看特定环境变量**

你可以使用 echo 命令来查看每个环境变量的值。例如：

bash

复制代码

echo $AWS\_REGION

echo $AWS\_DEFAULT\_REGION

echo $AWS\_PROFILE

**方法 2: 使用 env 或 printenv 命令查看所有环境变量**

使用 env 或 printenv 命令可以查看当前所有设置的环境变量。你可以直接运行：

bash

复制代码

env

或者：

bash

复制代码

printenv

这些命令会列出所有的环境变量，你可以在输出中查找 AWS\_REGION、AWS\_DEFAULT\_REGION 和 AWS\_PROFILE，并确认它们的值是否与你设置的一致。

**方法 3: 重新加载并检查 .bashrc**

如果你刚刚编辑了 ~/.bashrc 文件，你需要重新加载它，然后再检查环境变量是否成功设置：

bash

复制代码

source ~/.bashrc

然后再使用方法 1 (echo) 查看这些变量的值，确保它们已经正确设置。

**总结**

通过以上方法，你可以确认 AWS CLI 所需的环境变量已经正确设置。确保这些值与预期相符，这样就可以在每次打开终端时自动加载你所需要的 AWS 配置。

4o

**您说：**

❯ source ~/.bashrc source: no such file or directory: /Users/fangweidong/.bashrc

**ChatGPT 说：**

ChatGPT

在 macOS 系统上，默认使用的是 Zsh 作为终端的默认 shell，因此配置文件通常是 ~/.zshrc 而不是 ~/.bashrc。你收到的错误信息是因为系统找不到 ~/.bashrc 文件。以下是如何修复并检查环境变量的步骤：

**解决方法：**

1. **修改命令，把配置添加到 ~/.zshrc 文件：**

运行以下命令，将环境变量添加到 ~/.zshrc：

bash

复制代码

echo 'export AWS\_REGION=ap-southeast-2' >> ~/.zshrc

echo 'export AWS\_DEFAULT\_REGION=ap-southeast-2' >> ~/.zshrc

echo 'export AWS\_PROFILE=default' >> ~/.zshrc

1. **重新加载 ~/.zshrc 文件：**

运行以下命令重新加载配置文件，使新设置的环境变量生效：

bash

复制代码

source ~/.zshrc

1. **检查环境变量是否已成功设置：**

使用以下命令检查每个环境变量：

bash

复制代码

echo $AWS\_REGION

echo $AWS\_DEFAULT\_REGION

echo $AWS\_PROFILE

如果这些命令输出与你设置的值相符（ap-southeast-2 和 default），那么环境变量已成功设置并生效。

4o

**您说：**

我要移除export AWS\_REGION=ap-southeast-2 export AWS\_DEFAULT\_REGION=ap-southeast-2 export AWS\_PROFILE=default 这三条 该怎么做

**ChatGPT 说：**

ChatGPT

要移除 ~/.zshrc 文件中的以下三条环境变量设置：

bash

复制代码

export AWS\_REGION=ap-southeast-2

export AWS\_DEFAULT\_REGION=ap-southeast-2

export AWS\_PROFILE=default

请按照以下步骤操作：

**步骤：**

1. **打开 ~/.zshrc 文件进行编辑**：

你可以使用任何文本编辑器来编辑 ~/.zshrc 文件，例如使用 nano、vim 或者 code（VS Code）。

使用 nano 编辑器的命令如下：

bash

复制代码

nano ~/.zshrc

或者使用 vim：

bash

复制代码

vim ~/.zshrc

如果你使用 VS Code，可以用以下命令：

bash

复制代码

code ~/.zshrc

1. **找到并删除环境变量设置**：

在打开的编辑器中，找到以下行：

bash

复制代码

export AWS\_REGION=ap-southeast-2

export AWS\_DEFAULT\_REGION=ap-southeast-2

export AWS\_PROFILE=default

删除这些行。

1. **保存并退出编辑器**：
   * 如果你使用 nano，按 Ctrl + X，然后按 Y 确认保存并按 Enter 退出。
   * 如果你使用 vim，输入 :wq，然后按 Enter 保存并退出。
   * 如果你使用 code，点击保存（或者按 Ctrl + S），然后关闭文件。
2. **重新加载 ~/.zshrc 文件**：

运行以下命令来重新加载配置文件，使修改生效：

bash

复制代码

source ~/.zshrc

完成以上步骤后，这些环境变量将从你的终端配置中移除。

4o

**您说：**

.zshrc 是什么的简称

**ChatGPT 说：**

ChatGPT

.zshrc 是 “Z shell run commands”的简称，专用于 Zsh（Z Shell）这个 shell 程序的配置文件。

**详细解释：**

* **Zsh**：Zsh 是 Unix shell 的一种，是 Bash（Bourne Again Shell）的一种增强版本，提供了更多的功能、更强的可定制性和更丰富的插件系统。Zsh 被许多开发者和用户喜爱，因为它支持自动补全、语法高亮、自动纠错等功能。
* **.zshrc 文件**：这个文件是 Zsh 的配置文件。每当你打开一个新的 Zsh 终端或启动一个新的 Zsh 会话时，.zshrc 文件中的命令就会被自动执行。它通常用于设置环境变量、别名、自定义 shell 提示符以及加载其他插件或配置。
* **.zshrc 的作用**：你可以在 .zshrc 中添加各种配置来个性化你的终端体验，比如设置默认路径、定义命令别名、配置 shell 提示符样式，以及加载 Zsh 插件（如 Oh My Zsh）。

在 macOS 中，Zsh 是默认的终端 shell，因此 .zshrc 文件通常是用来调整终端行为和外观的主要配置文件。

4o

**您说：**

echo 'export AWS\_REGION=ap-southeast-2' >> ~/.zshrc echo是什么意思

**ChatGPT 说：**

ChatGPT

echo 是一个常见的命令行工具，用于在终端中输出文本或变量的值。它的作用是将指定的内容显示在屏幕上或写入文件中。

**echo 命令的详细解释：**

* **基本功能**：echo 命令会将你指定的文本内容输出到终端或将内容重定向到其他位置（如文件）。
* **语法**：

bash

复制代码

echo [选项] [内容]

* **示例解释**：  
  在你给出的命令中：

bash

复制代码

echo 'export AWS\_REGION=ap-southeast-2' >> ~/.zshrc

这个命令的作用是：

* 1. **echo 'export AWS\_REGION=ap-southeast-2'**：将 export AWS\_REGION=ap-southeast-2 这段文本输出到标准输出（即终端）。
  2. **>> ~/.zshrc**：将输出的内容追加到 ~/.zshrc 文件的末尾。>> 是追加重定向符号，它会将 echo 的输出内容附加到指定文件中，而不会覆盖文件中已有的内容。

**常见用途：**

* **输出文本**：在屏幕上显示指定的文本，如 echo "Hello, World!"。
* **显示变量值**：如 echo $PATH，用于查看当前的环境变量 PATH 的值。
* **写入文件**：将内容追加或重定向到文件，如 echo "alias ll='ls -la'" >> ~/.zshrc，将命令别名追加到配置文件中。

使用 echo，你可以方便地将文本或变量内容输出到屏幕或写入文件中进行配置和调试。