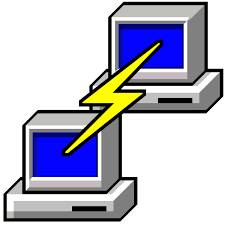
Cloud Computing Overview (AWS CLI)

HOS05: Identity and Access Management (IAM)

Developed by Marvin Gold 12/19/2020

Reviewed by Kim Nguyen class of 2021

School of Technology and Computing



**Learning Outcomes**

* Create IAM users
* Create IAM groups
* Add IAM users to groups
* List groups
* Attach a managed policy to a group
* Verify a users attached policies

**Background**

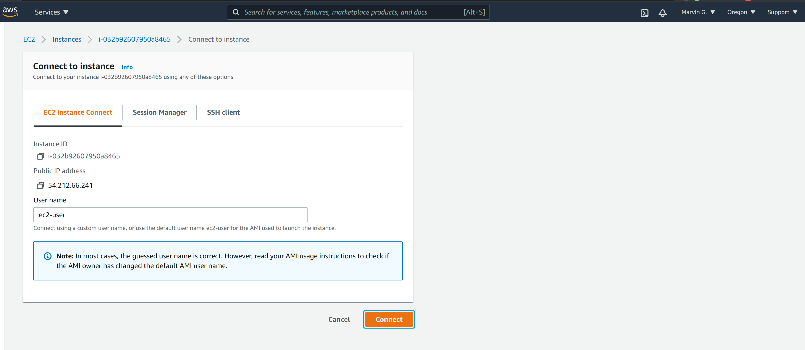
AWS Identity and Access Management (IAM) is a tool for controlling permissions in your AWS infrastructure. There are three main components. *Users, Groups* and *Roles*. The permissions are controlled with *policies* that we attach to *users, groups* or *roles.* A *managed policy* is a policy that AWS controls. In this HOS, we are going to learn how to create IAM users and groups. Then we are going to assign policies to these users and groups.

**References**

*Using AWS Identity and Access Management from the AWS CLI - AWS Command Line Interface*. (n.d.). Retrieved December 19, 2020, from <https://docs.aws.amazon.com/cli/latest/userguide/cli-services-iam.html>

**Step 1: Log onto your Amazon EC2 Instance**

Log onto your Amazon EC2 instance using either OpenSSH, Putty, or from the AWS Management Console using ‘EC2 Instance Connect’ (shown below). Note all IP addresses used in this series of Hands on Skills will be released by the time we start class.

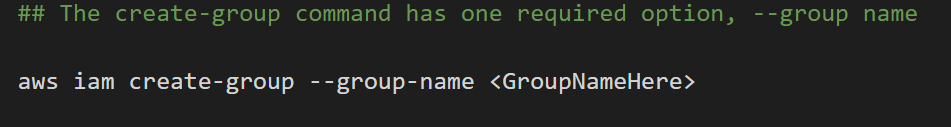


If you receive an error that you must first set your AWS Region and you must configure your AWS CLI, please follow these instructions.

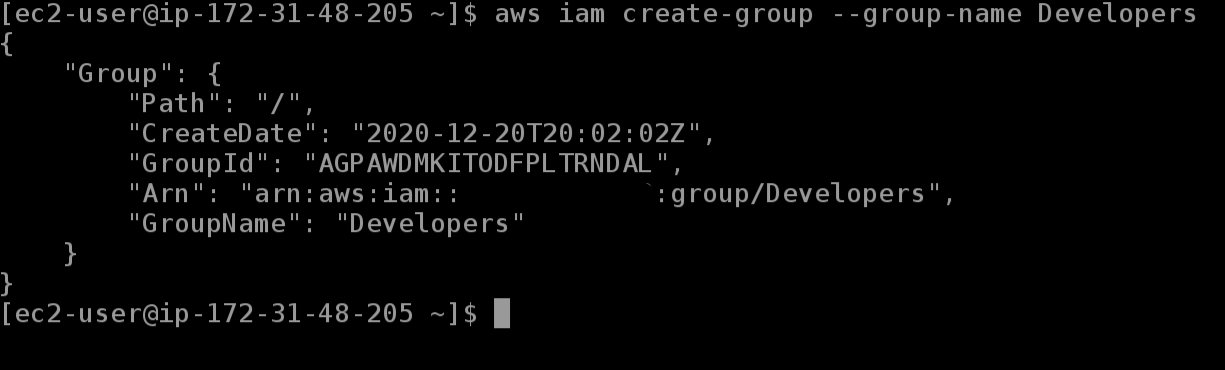
<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>  
  
Remember, do not use your root *account* for these operations. You should be using another IAM user. You should not have access keys to your root account.

**Step 2: Create an IAM Group**

An IAM group is a collection of users that have common permissions. We are going to make a group to put our Developers in and give them a set of baseline permissions. Then we are going to give the lead developer some extra permissions. But first, lets create the group our developers will be a part of.

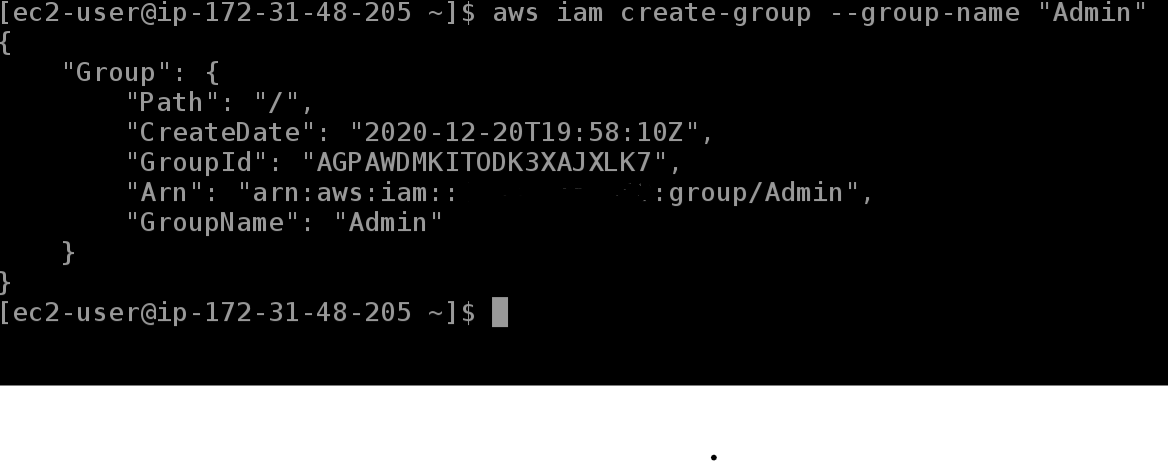


Create a group called “Developers”



Notice your group has an “Arn” or Amazon Resource Name, this is a way to systematically reference a resource you own. (I have redacted my AWS Account number out of the Arn). More info about ARN can be found [here](https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html).

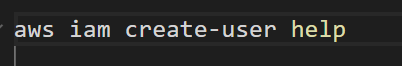
Now that you have a group for your Developers, let’s create a group for the Administrators, let’s call that group Admin.



**Step 3: Create Users**

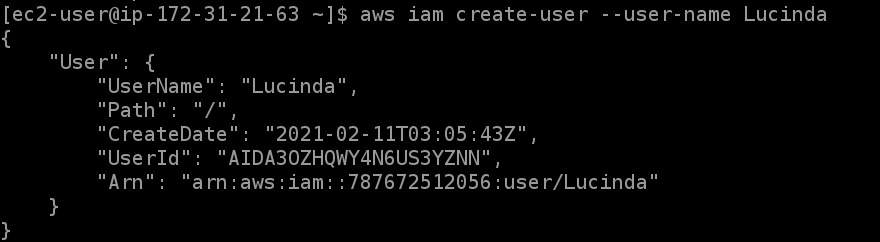
Now that we have groups to put users in, it’s time to create users!

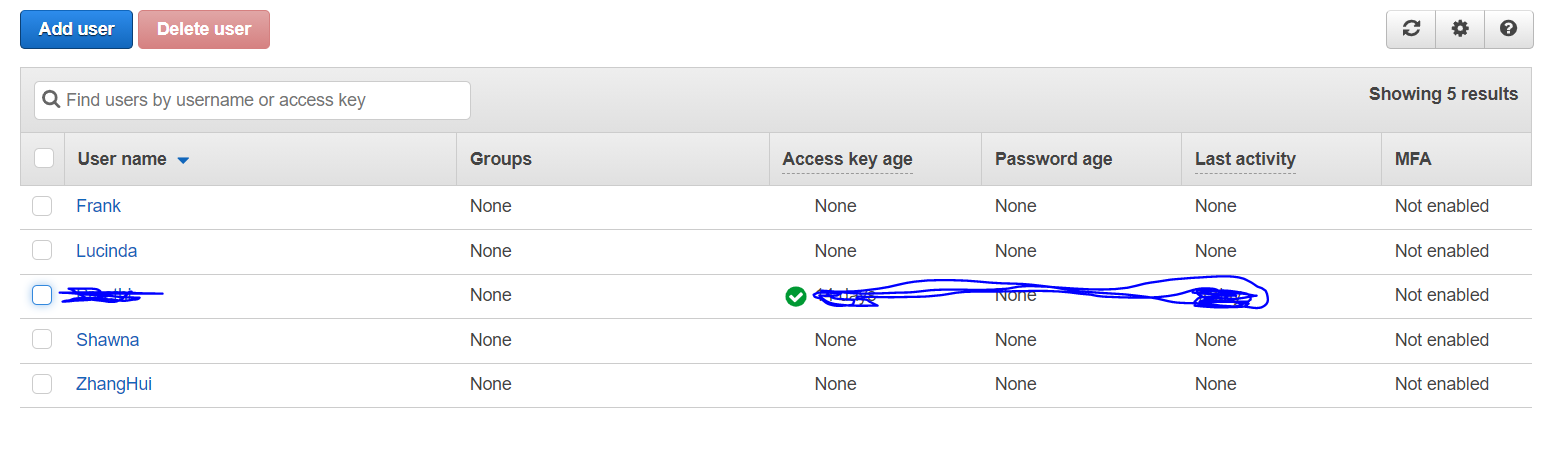
First, let’s take a look at what commands are available to us to create users.



**Question:** What parameters do you have to pass in to create a user?

**Answer:**





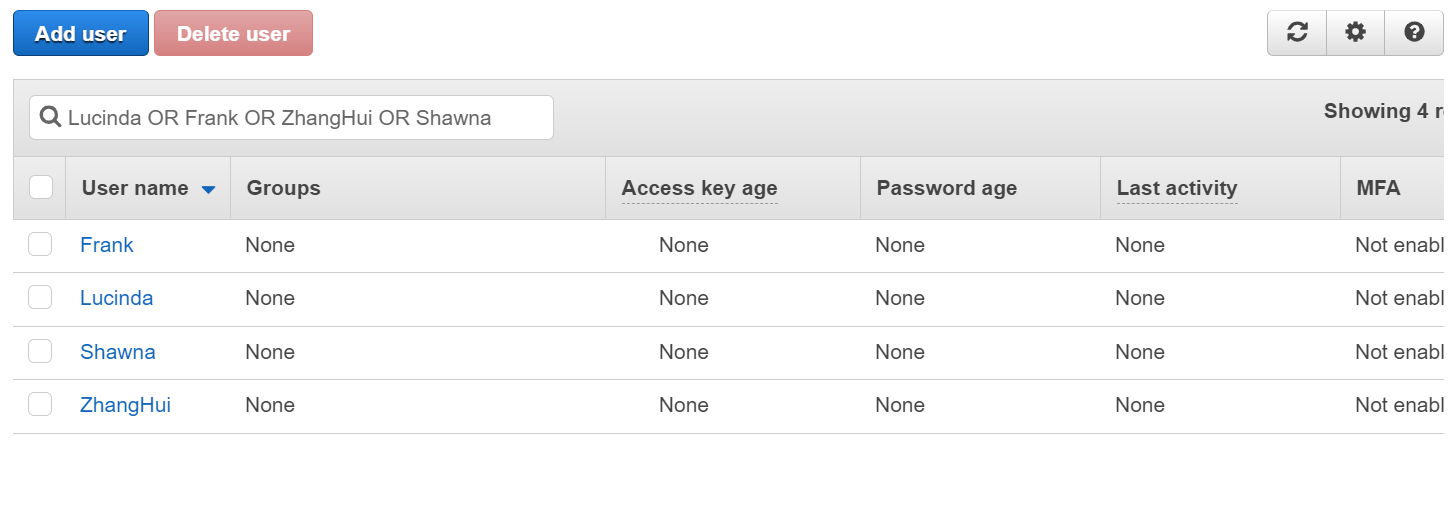
Create the following user(s)

ZhangHui

Lucinda

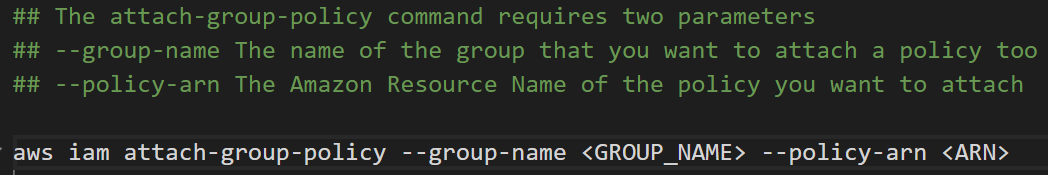
Frank  
Shawna

You can verify in the IAM Console that they were created.



**Step 4: Add AWS Managed Policies to Groups**

Now we have our 4 developers and your regular user in your AWS Account. The new four developers by default do not have any permissions. And the groups we created “Developers” and “Admin” also do not have any permissions either. Let’s add some permissions to the two groups we created earlier.



**Add the following Policies to the Developer Group**

arn:aws:iam::aws:policy/AWSCodeCommitPowerUser

arn:aws:iam::aws:policy/AmazonSQSFullAccess

arn:aws:iam::aws:policy/AmazonS3FullAccess

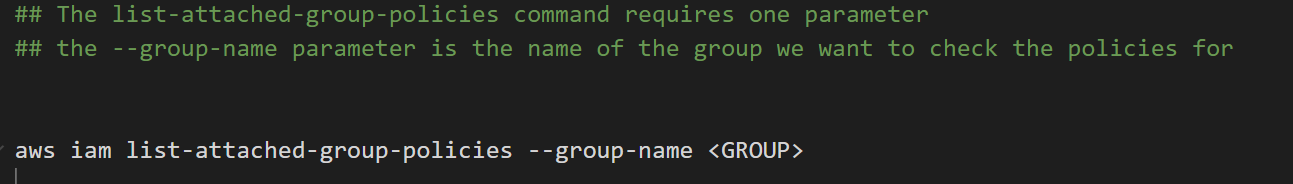
arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess

arn:aws:iam::aws:policy/AmazonSNSFullAccess

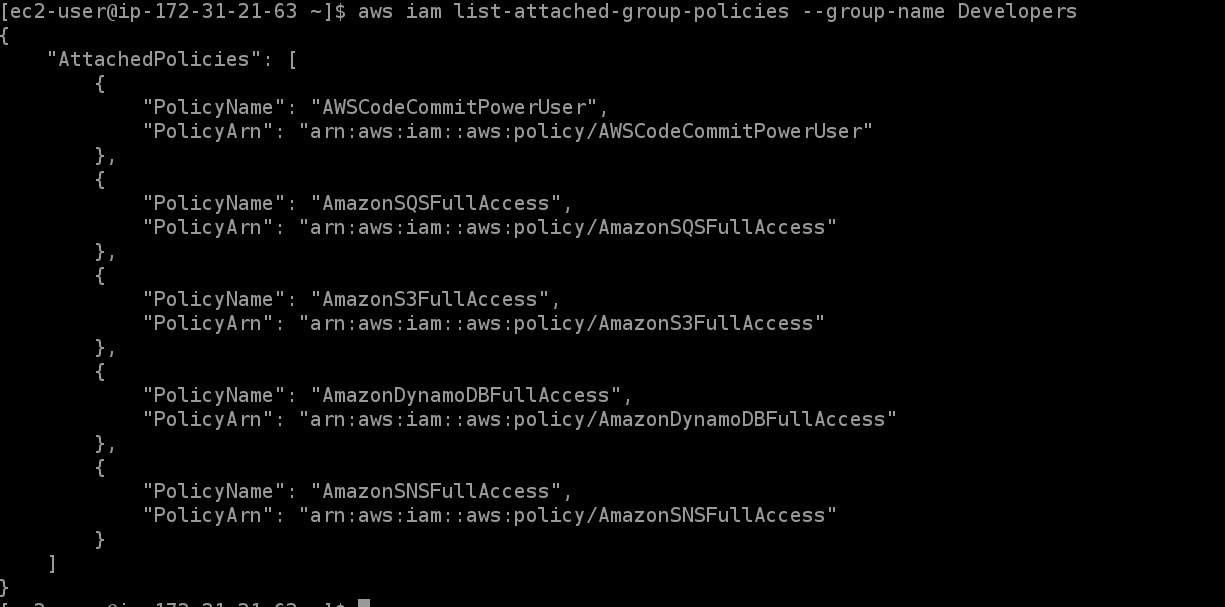
**Add the following Policy to the Admin group**

arn:aws:iam::aws:policy/AdministratorAccess

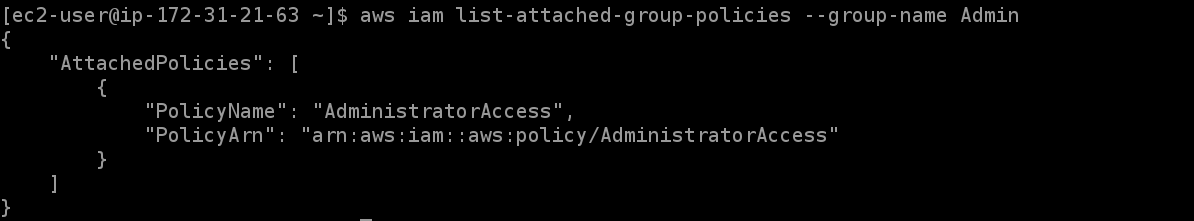
Let’s verify your groups have the correct policies



Check that your **Developers** have the correct policies and put a **SCREENSHOT** here.



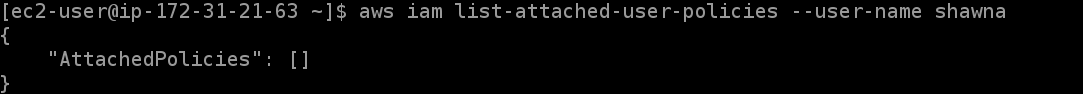
Check that your **Admin** group has the correct policies and put a **SCREENSHOT** here.



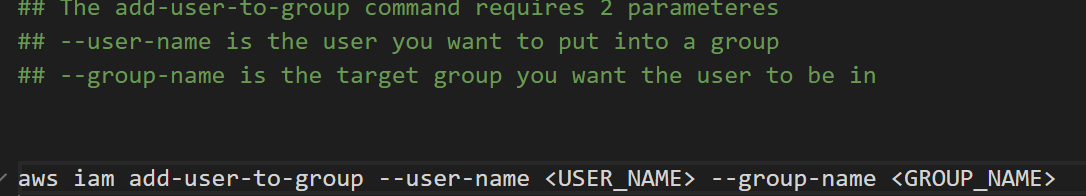
We can do the same thing with users as well to check policies by using the list-attached-users-policies. Use that command and check Shawna’s permissions, write one or two sentences on what is different about Shawna’s permissions when running the list-attached-user-permissions command.

**Answer:**

This command returns the names and ARNs of the managed policies for the IAM user named shawna in the Aws account. list-attached-user-policies is a paginated operation. Multiple API calls may be issued to retrieve the entire data set of results.



**Step 5: Add our USERS to the groups**



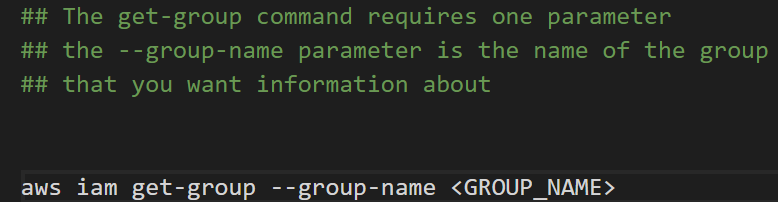
Add the users to the appropriate groups according to the table below

|  |  |
| --- | --- |
| Admin | Developer |
| Shawna | ZhangHui |
|  | Frank |
|  | Lucinda |
|  | Shawna |

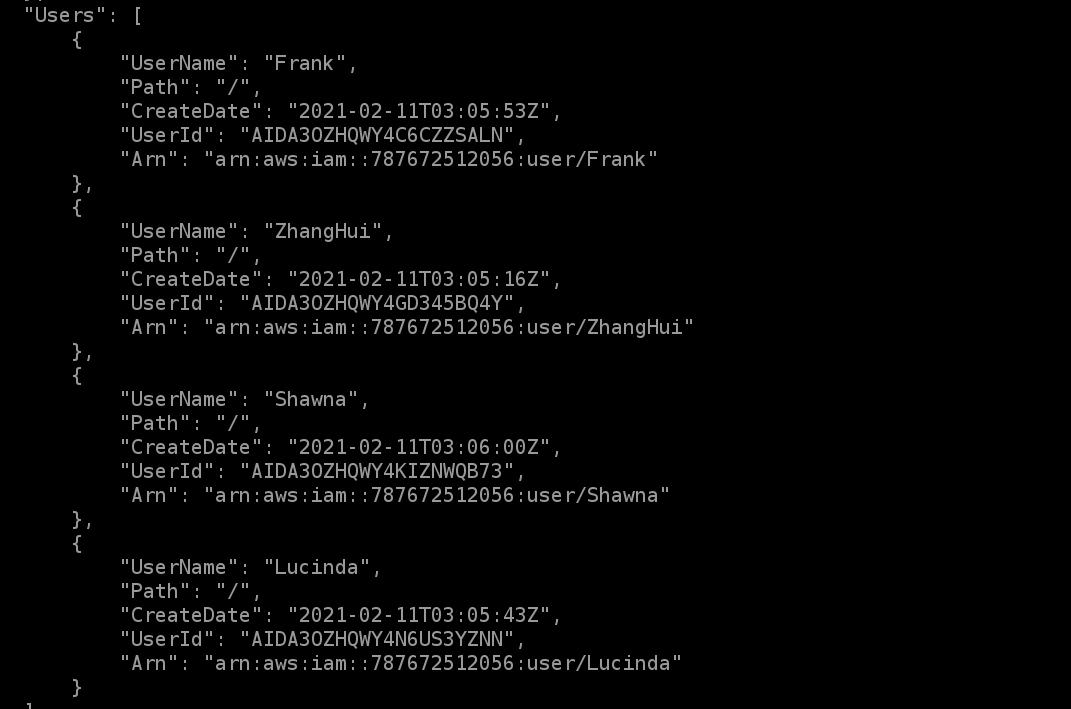
Notice how Shawna is in two groups? This means she will have access to the permissions of the Developer group and the Admin group.

**Step 6: List group members**

After you’ve added the users to the groups, lets verify the right people are in the right group using the command

‘get-group’

**Take a SCREEN SHOT of your Developer group**



**Take a SCREEN SHOT of your Admin group**

