IAM

Thursday, June 16, 2016 9:33 AM

INTRODUCTION

The purpose of this page is to introduce the central terms and ideas within the AWS Identity and Access Management service, specifically Groups, Policies, Users and Roles. It will be helpful to begin with the core AWS model: Build a Virtual Private Cloud (VPC) and build out Elastic Compute (EC2) instances within that VPC. We also hasten to add that not everything is built on EC2 as services are released that provide functionality without the underlying burden of a computer.

TERMINOLOGY

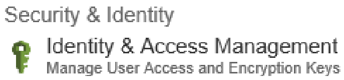
Groups
Roles: See https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html
Policies
Users

Managing Users for a Hackathon

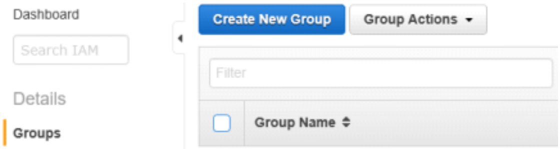
Suppose I have an account on AWS and my friend Ariel is going to use it for a Neuro-Hackathon, and he in turn will want to manage participant access to that account. We can make everybody a super user but it is considered better form to create a group and a set of users and allow each of them to belong to that group. Let's try that. But be warned: The procedure we give here ***does not work*** the way we wanted; so I will describe what happens briefly here and we will fix the problem down at the end of this section.

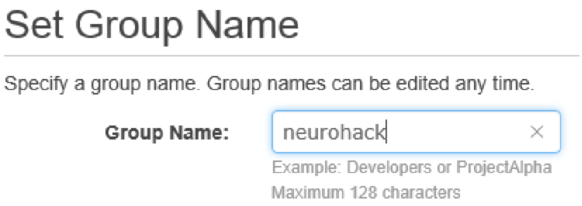
What we want on AWS is to create an entity called a 'Group' that has some space to create computations; and then we create some temporary Users who can all use that Group space. The person who does this creating is an Administrator. The original idea here is to make Ariel a low-level administrator or 'Power User' by assigning him to the Group. Then he would add the others so it would bootstrap the hackathon. Unfortunately this approach does not give him enough leverage in the AWS system; so at the very end we also give him an Administrator Policy. Once that is done he can go ahead and create the other users using the procedure we give here.

I log in as my 'admin' self to the AWS console and find the IAM link:


Security & Identity
Identity & Access Management
Manage User Access and Encryption Keys

I'm going to create a group called neurohack, all lower case.





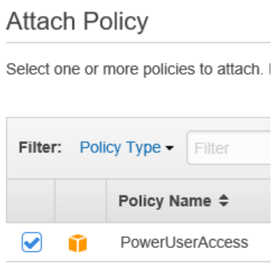
Set Group Name

Specify a group name. Group names can be edited any time.

Group Name: neurohack

Example: Developers or ProjectAlpha
Maximum 128 characters

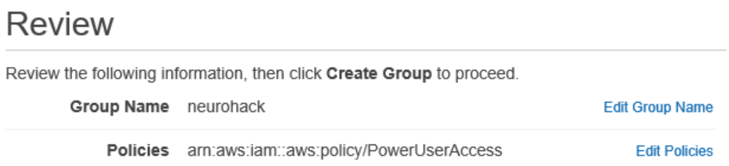
I give it a PowerUserAccess policy. That lets Ariel and everyone else pretty much run the show inside this group.


Attach Policy

Select one or more policies to attach. I

☑ PowerUserAccess

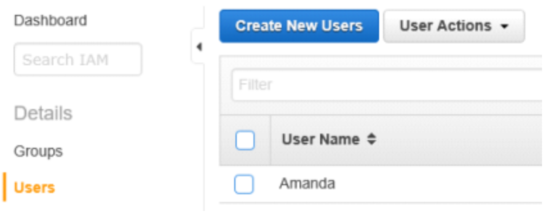
So to review:


Review

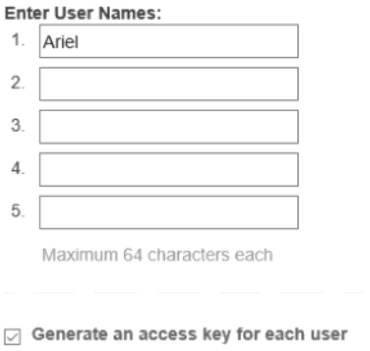
Review the following information, then click **Create Group** to proceed.

Group Name	neurohack	Edit Group Name
Policies	arn:aws:iam::aws:policy/PowerUserAccess	Edit Policies

Now let's add Ariel as a User. He will not get any Policy but he will get login credentials. Furthermore before I send those to him I will add him to the neurohack group.



□ Amanda

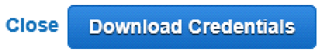

Enter User Names:

1. Ariel
2.
3.
4.
5.

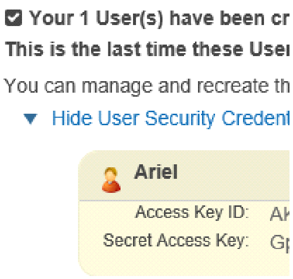
Maximum 64 characters each

☑ Generate an access key for each user

And I will download the credentials although I don't think I wind up using them. I also rename them so I know they are associated with Ariel.



Here is what the credentials "look like" (I have cut the strings off after the first couple of characters):


☑ Your 1 User(s) have been cr
This is the last time these Usei

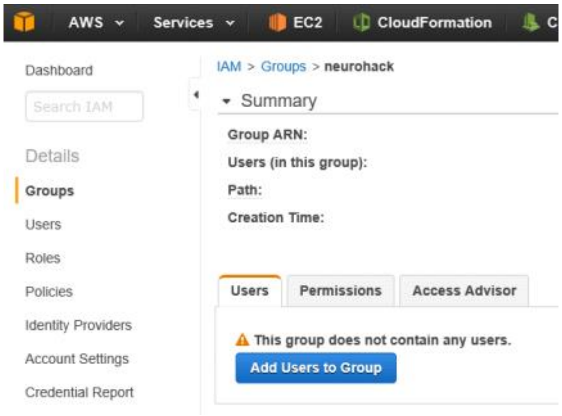
You can manage and recreate th

▼ Hide User Security Credent

Ariel

Access Key ID: Ak
Secret Access Key: Gı

Now I close the Credentials page and proceed to add Ariel to the neurohack group.



IAM > Groups > neurohack

Summary

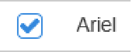
Group ARN:

Users (in this group):

Path:

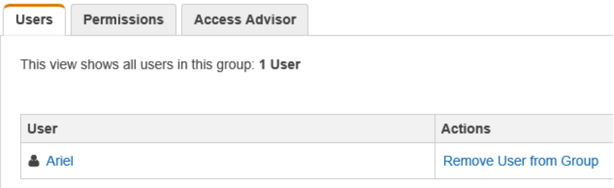
Creation Time:

This group does not contain any users.



☑ Ariel

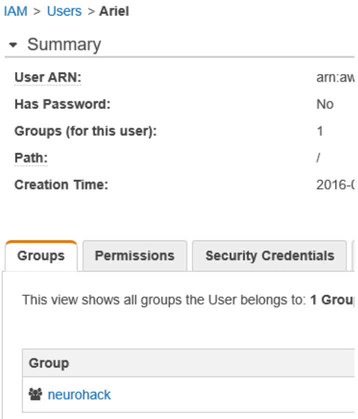
…and here he shows up now:



This view shows all users in this group: **1 User**

User	Actions
Ariel	Remove User from Group

So now let's generate login credentials and send them to him. Here is his entry among the account Users:



IAM > Users > Ariel

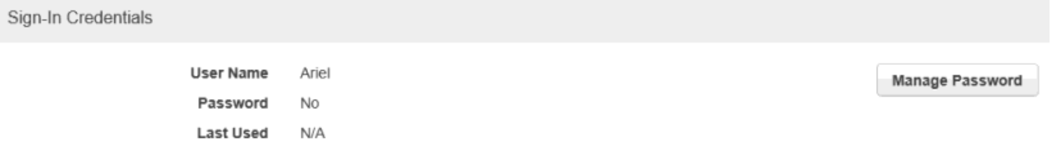
Summary

User ARN:	arn:aw
Has Password:	No
Groups (for this user):	1
Path:	/
Creation Time:	2016-(

This view shows all groups the User belongs to: **1 Grou**

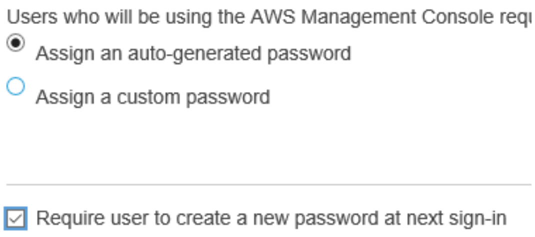
Group
neurohack

So under the Security Credentials tab we just create a login:



Sign-In Credentials

User Name	Ariel
Password	No
Last Used	N/A



Users who will be using the AWS Management Console requ

● Assign an auto-generated password

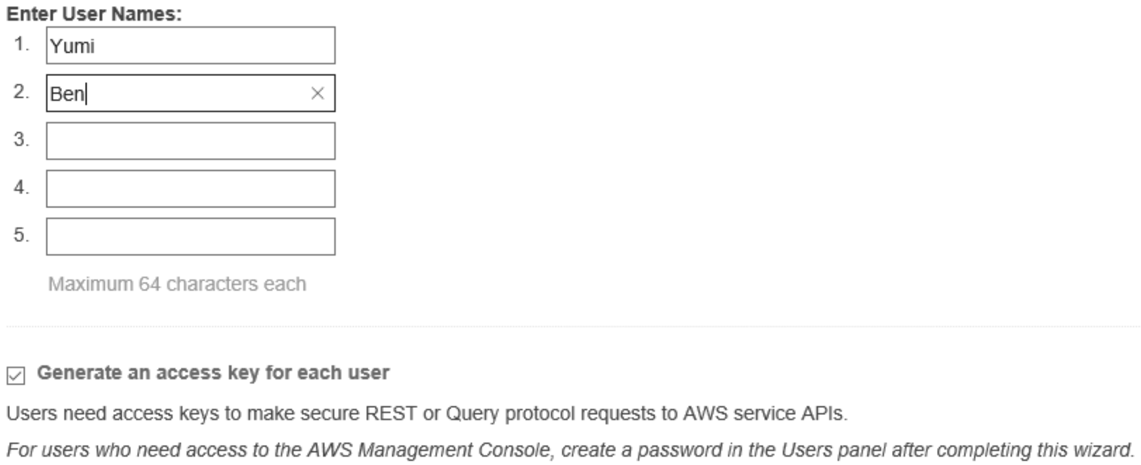
○ Assign a custom password

☑ Require user to create a new password at next sign-in

And again we download these credentials. Two things we did not do are: Require Multi Factor Authorization (MFA) and print the credentials on a piece of paper to hand to Ariel. Those are necessary steps but I don't show them here.

Adding a person

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html



Enter User Names:

1. Yumi
2. Ben
3.
4.
5.

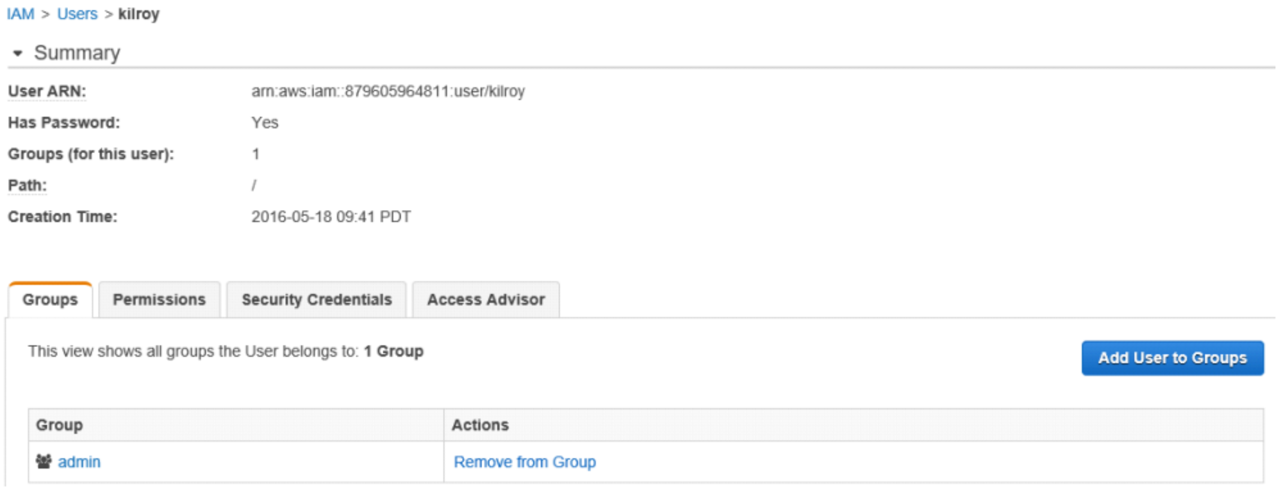
Maximum 64 characters each

☑ **Generate an access key for each user**

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.

For users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.

Ok I created starsynth and am adding kilroy to that group:



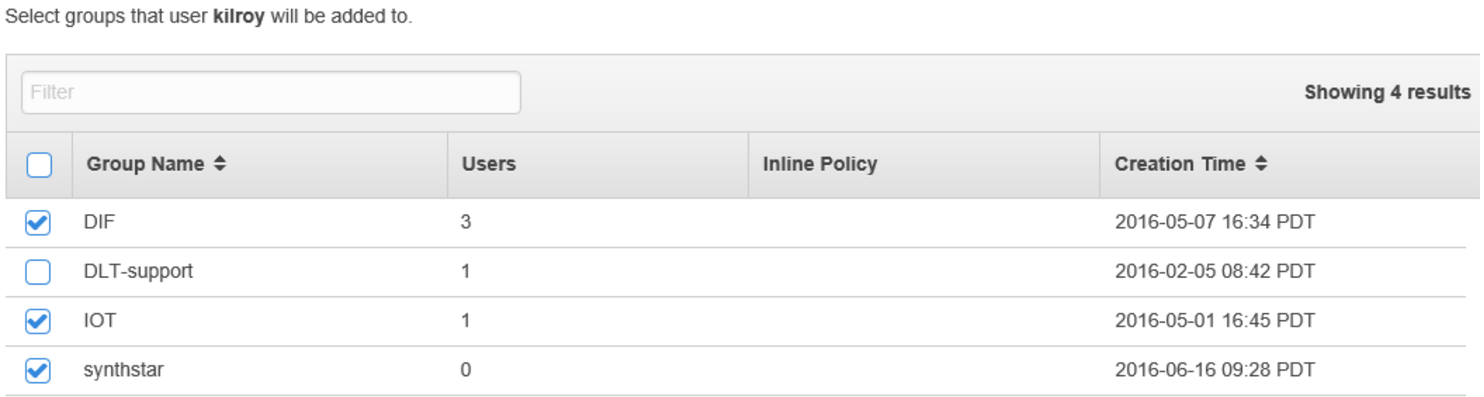
IAM > Users > **kilroy**

▾ Summary

User ARN:	arn:aws:iam::879605964811:user/kilroy
Has Password:	Yes
Groups (for this user):	1
Path:	/
Creation Time:	2016-05-18 09:41 PDT

This view shows all groups the User belongs to: **1 Group**

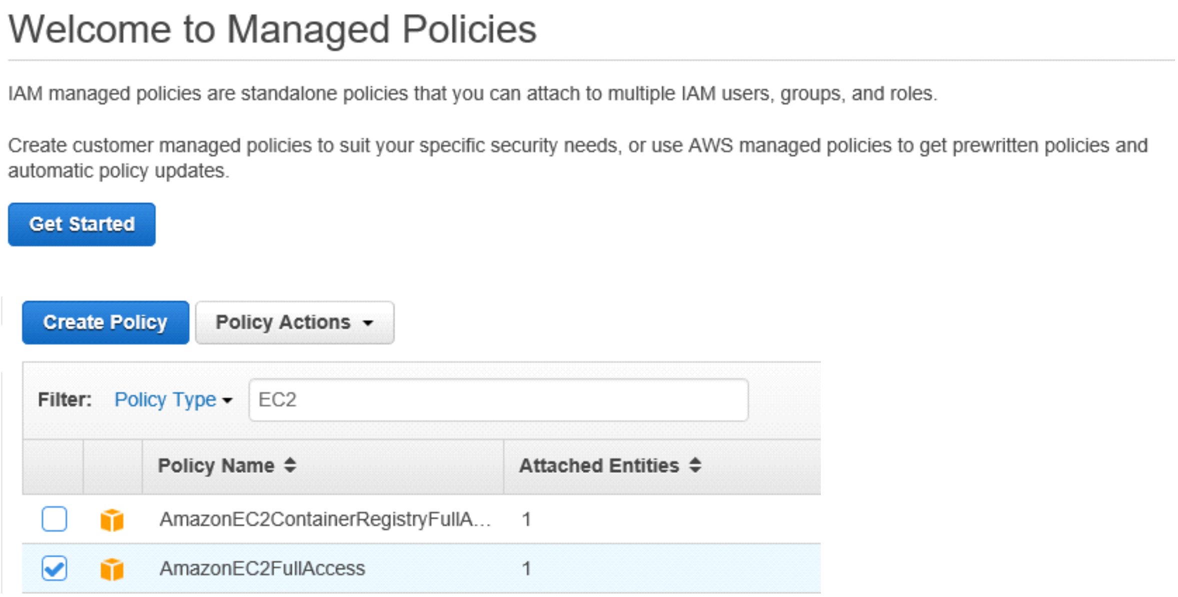
Group	Actions
admin	Remove from Group



Select groups that user **kilroy** will be added to.

□	Group Name ⇕	Users	Inline Policy	Creation Time ⇕
☑	DIF	3		2016-05-07 16:34 PDT
□	DLT-support	1		2016-02-05 08:42 PDT
☑	IOT	1		2016-05-01 16:45 PDT
☑	synthstar	0		2016-06-16 09:28 PDT

Now the goal will be to create a Policy for synthstar (custom) that restricts IAM Users from doing things beyond the scope of that group.



Welcome to Managed Policies

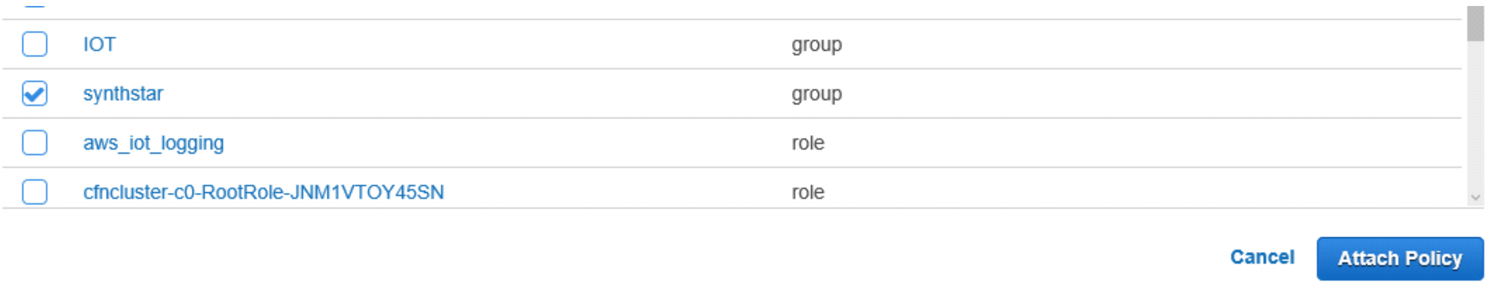
IAM managed policies are standalone policies that you can attach to multiple IAM users, groups, and roles.

Create customer managed policies to suit your specific security needs, or use AWS managed policies to get prewritten policies and automatic policy updates.

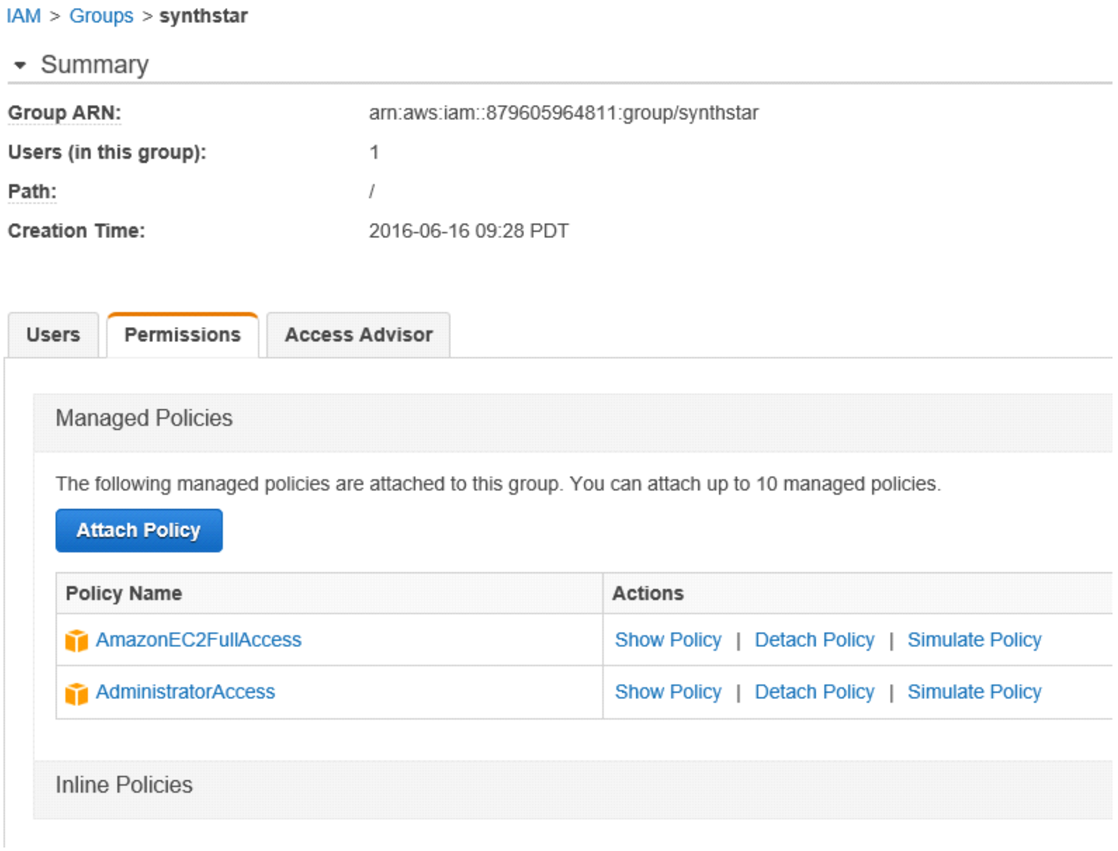
Filter: Policy Type ▾ EC2

		Policy Name ⬍	Attached Entities ⬍
□	📦	AmazonEC2ContainerRegistryFullA...	1
☑	📦	AmazonEC2FullAccess	1

Go to Policy Actions drop-down and select Attach Policy:



□	IOT	group
☑	synthstar	group
□	aws_iot_logging	role
□	cfncluster-c0-RootRole-JNM1VTOY45SN	role



IAM > Groups > **synthstar**

▾ Summary

Group ARN:	arn:aws:iam::879605964811:group/synthstar
Users (in this group):	1
Path:	/
Creation Time:	2016-06-16 09:28 PDT

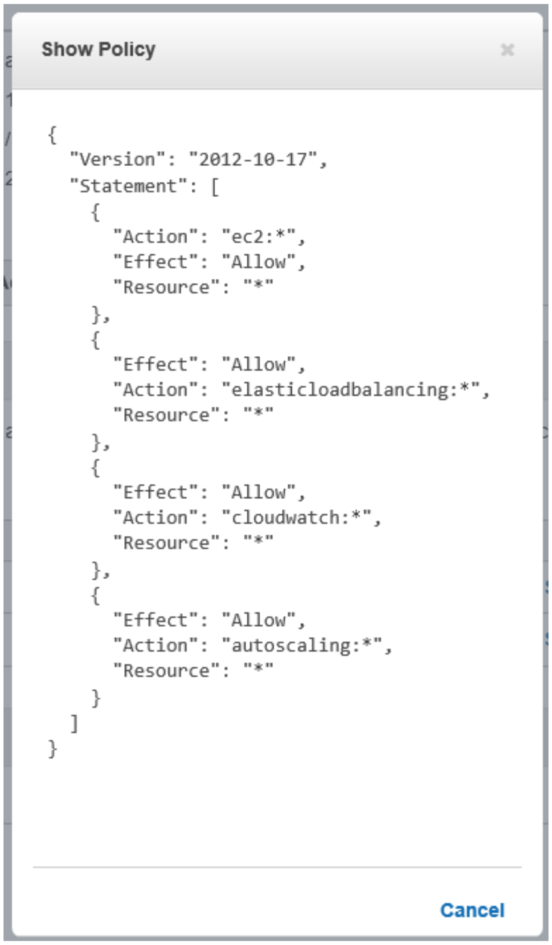
Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Policy Name	Actions
📦 AmazonEC2FullAccess	Show Policy | Detach Policy | Simulate Policy
📦 AdministratorAccess	Show Policy | Detach Policy | Simulate Policy

Inline Policies

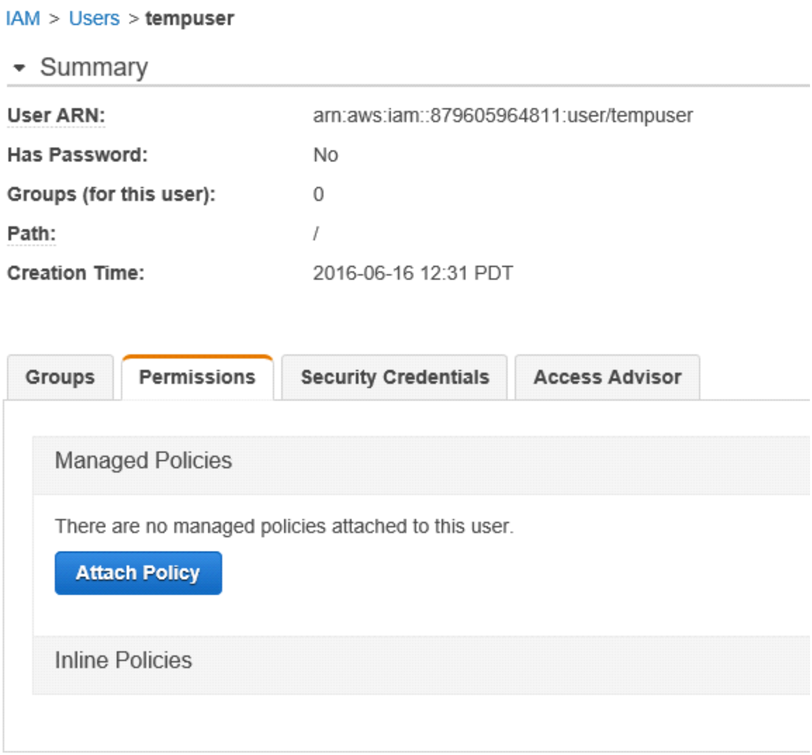
Now Show Policy on EC2FullAccess:



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ec2:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "cloudwatch:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:*",
      "Resource": "*"
    }
  ]
}
```

And we can cut and paste this into a blank Policy and work from there.

Now let's make a new user 'tempuser'. Notice that they do not have any de novo Policies:



User ARN: arn:aws:iam::879605964811:user/tempuser

Has Password: No

Groups (for this user): 0

Path: /

Creation Time: 2016-06-16 12:31 PDT

There are no managed policies attached to this user.

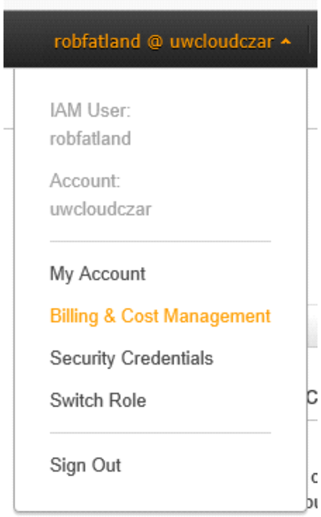
Why would I give them a Policy? I won't! I shant! Rather I will attach them to a Group and give the Group a policy so that they can proceed as group members to do those things in that group.

Incidentally if you do want to create a super user of sorts: Power User is the Policy; it is "everything but IAM access".

Policies can be attached to Users, Groups and Roles. Groups is the way to go to control how we do stuff.

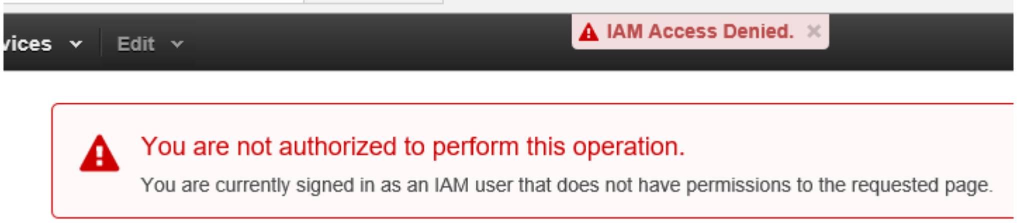
Now let's talk about billing. I want to know how much damage tempuser does to my bottom line each month. Tagging!

On the console go to my account dropdown...



IAM User:
robfatland

Account:
uwcloudczar



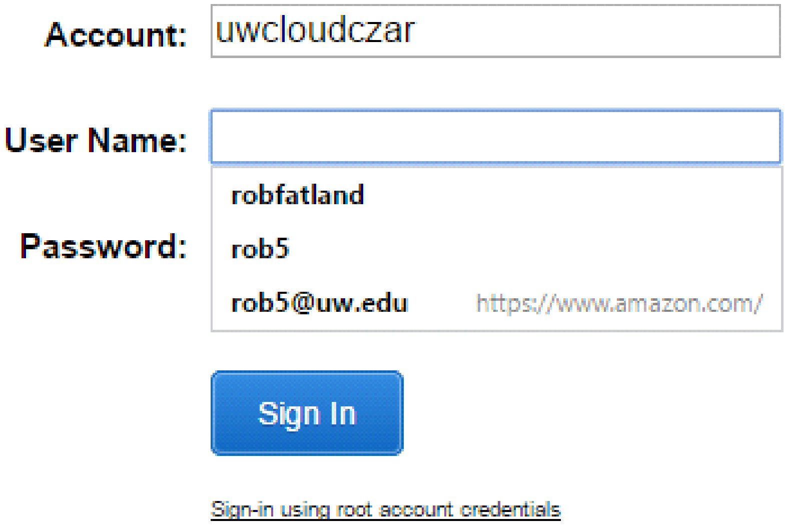
IAM Access Denied.

You are not authorized to perform this operation.

You are currently signed in as an IAM user that does not have permissions to the requested page.

Oops... better open a different browser.

And sign in using root credentials (small type):

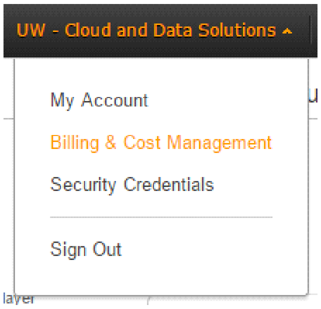


Account: uwcloudczar

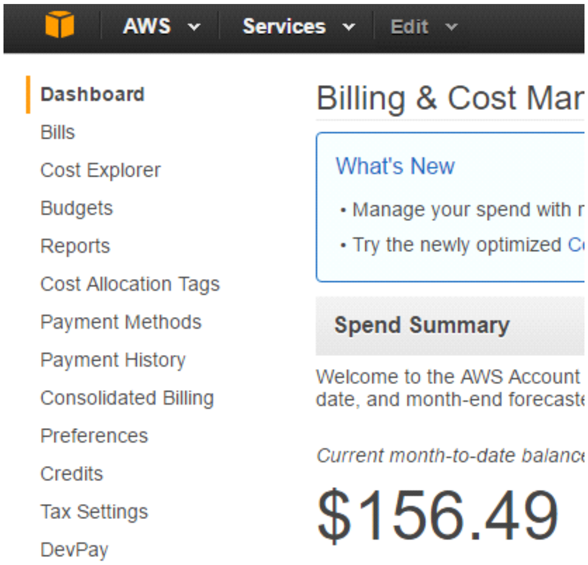
User Name:

Password:

robfatland
rob5
rob5@uw.edu https://www.amazon.com/



And now notice on the left sidebar "Cost Allocation Tags"



Billing & Cost Mar

What's New

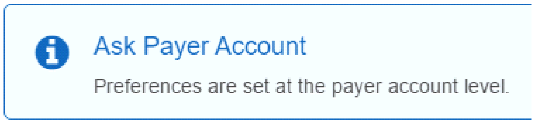
- Manage your spend with r
- Try the newly optimized C

Spend Summary

Welcome to the AWS Account
date, and month-end forecast

Current month-to-date balance

$156.49



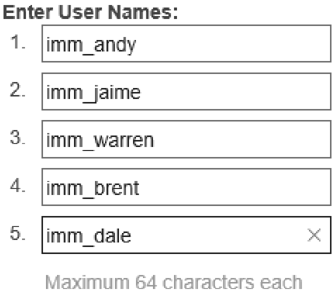
Ask Payer Account

Preferences are set at the payer account level.

So this is DLT funny business... and the Orbitera application is mentioned in this context.

Group: IAM_Immersion_for_IT

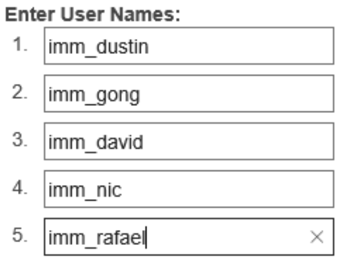
Added some users:



Enter User Names:

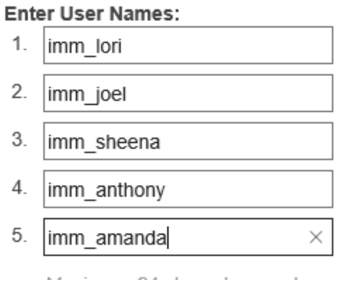
1. imm_andy
2. imm_jaime
3. imm_warren
4. imm_brent
5. imm_dale

Maximum 64 characters each



Enter User Names:

1. imm_dustin
2. imm_gong
3. imm_david
4. imm_nic
5. imm_rafael



Enter User Names:

1. imm_lori
2. imm_joel
3. imm_sheena
4. imm_anthony
5. imm_amanda