SECURING YOUR INFRASTRUCTURE WITH IAM ROLES

ADRIAN DRUMMOND & DYLAN VAUGHN

Anatomy of an ARN

arn:partition:service:region:account-id:resource

arn:partition:service:region:account-id:resourcetype/resource

arn:partition:service:region:account-id:resourcetype:resource

Example:
arn:aws:lambda:us-west-2:101836606311:function:TestPoo

http://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html#genref-arns

Anatomy of a Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FirstStatement",
      "Effect": "Allow",
      "Action": ["iam:ChangePassword"],
      "Resource": "*"
    },
    {
      "Sid": "SecondStatement",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Sid": "ThirdStatement",
      "Effect": "Allow",
      "Action": [
        "s3:List*",
        "s3:Get*"
      ],
      "Resource": [
        "arn:aws:s3:::confidential-data",
        "arn:aws:s3:::confidential-data/*"
      ],
      "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
    }
  ]
}
```

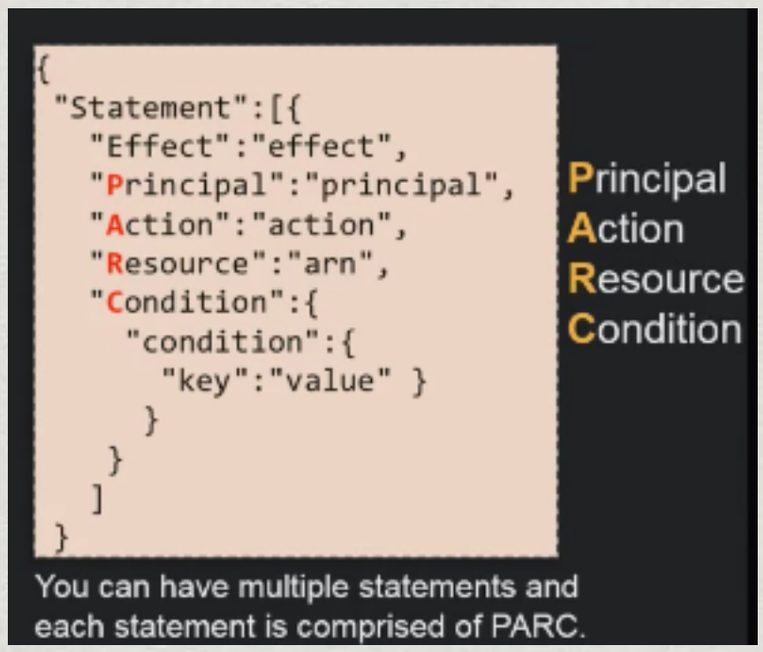
```
{
  "Version": "2012-10-17",
  "Id": "S3-Account-Permissions",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:root"]},
    "Action": "s3:*",
    "Resource":
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
  }]
}
```

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

Anatomy of an Policy


```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "FirstStatement",
   "Effect": "Allow",
   "Action": ["iam:ChangePassword"],
   "Resource": "*"
  },
  {
   "Sid": "SecondStatement",
   "Effect": "Allow",
   "Action": "s3:ListAllMyBuckets",
   "Resource": "*"
  },
  {
   "Sid": "ThirdStatement",
   "Effect": "Allow",
   "Action": [
    "s3:List*",
    "s3:Get*"
   ],
   "Resource": [
    "arn:aws:s3:::confidential-data",
    "arn:aws:s3:::confidential-data/*"
   ],
   "Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
  }
 ]
}
```

```
{
 "Version": "2012-10-17",
 "Id": "S3-Account-Permissions",
 "Statement": [{
  "Sid": "1",
  "Effect": "Allow",
  "Principal": {"AWS": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:root"]},
  "Action": "s3:*",
  "Resource":
   "arn:aws:s3:::mybucket",
   "arn:aws:s3:::mybucket/*"
 }]
}
```



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

Versions of Policy

You may have noticed a version number in most policies.
Here is what you need to know:

- **The version specifies the policy language version used. It is not a version you create.**
- **Versions allow IAM to enhance the policy language while continuing to support existing policies.**
- **It is best practice to always specify the current version in your policies.**
- **At the time of publishing, this version is "2012-10-17".**
  - **This will enable you to use the most recent features in the policy language.**
- **If you have an older policy that uses version "2008-10-17," it will continue to work.**
  - **However, you will not be able to use newer features like policy variables in this policy.**
  - **If you do not specify a version then it defaults to "2008-10-17."**

The Non-Crappy Documentation

http://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html
https://aws.amazon.com/documentation/
https://aws.amazon.com/documentation/iam/?icmpid=docs_menu
https://aws.amazon.com/iam/developer-resources/
https://aws.amazon.com/iam/
http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html
http://docs.aws.amazon.com/IAM/latest/APIReference/Welcome.html

https://docs.aws.amazon.com
redirects to:
https://aws.amazon.com/documentation/

Have you ever been denied?

- decode-authorization-message

- http://docs.aws.amazon.com/cli/latest/reference/sts/decode-authorization-message.html

- https://youtu.be/y7-fAT3z8Lo?t=3144\

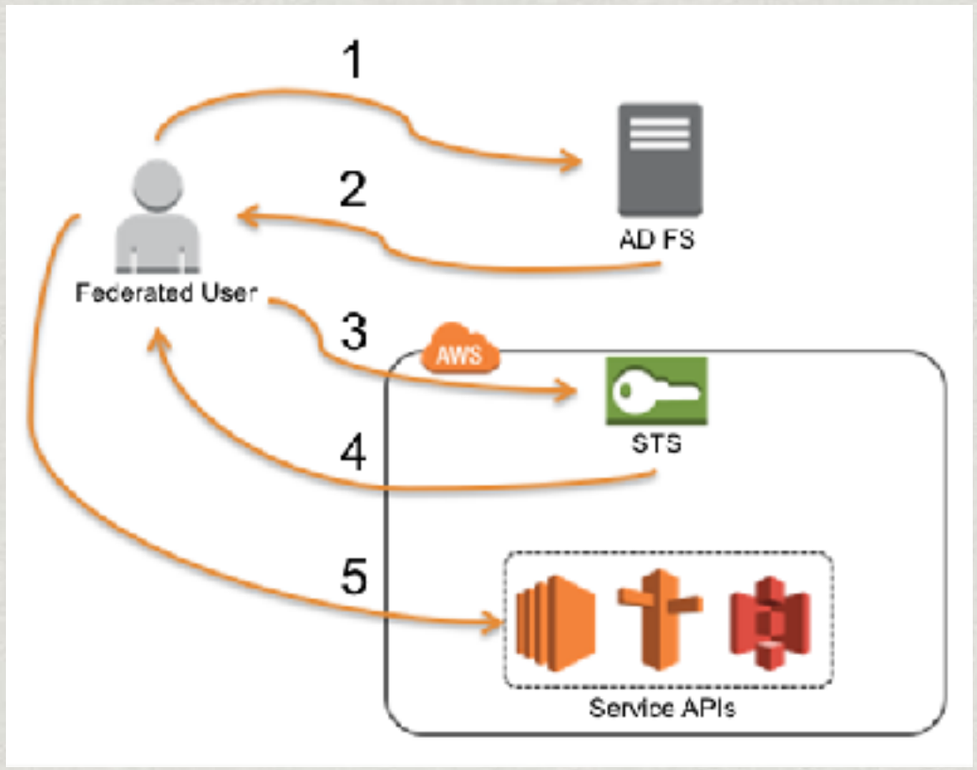
What is an Instance Profile

“An instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.”

IAM : Assume Role

- Returns a set of temporary security credentials


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service":
"ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



YOU CAN'T SIMPLY SET THE ACTION TO EC2:*
AND ALSO USE A RESOURCE OTHER THAN "*"

INSTEAD:
TO GRANT PERMISSION TO A SPECIFIC RESOURCE, THE POLICY MUST EXPLICITLY LIST THE ACTIONS THAT ARE BEING GRANTED OR DENIED, AND AS NOTED, ONLY SOME EC2 ACTIONS LET YOU SPECIFY A RESOURCE.

HTTPS://AWS.AMAZON.COM/BLOGS/SECURITY/DEMYSTIFYING-EC2-RESOURCE-LEVEL-PERMISSIONS/

FOR EXAMPLE:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TheseActionsSupportResourceLevelPermissions",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:StopInstances",
                "ec2:StartInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:accountid:instance/*"
        }
    ]
}
```

RESOURCE-LEVEL PERMISSOINS

WITHOUT THIS,
THE RESOURCE WOULD HAVE
TO BE: “*”

TIPS:

- Create a No Privileges Group
- Learn how to view Cloud Trail Activity
- Can your policy be expressed using ‘NotAction’
- Wildcards:
  - * = multi-character wildcard
  - ? = single-character wildcard
    - EXAMPLE: “Action”:”iam”:*AccessKey*
- …..more
- Can you apply your permissions using tags?
- Tag EVERYTHING
  - Use AWS Resource Groups
    - https://resources.console.aws.amazon.com/r/group
    - https://resources.console.aws.amazon.com/r/tags

The Challenge

The Challenge

1. Create a brand new AWS user named ‘bob’
2. Use the AWS CLI to launch a CloudFormation stack using ‘bob’ credentials
3. Enter team information into the SNS topic and subscribe the team
4. Click on the homepage link for the EC2 server launched.
5. Enter the URL for another team’s EC2 instance home page into a new SNS topic
6. Change the new SNS topics subscription from a URL to a Lambda function in the other team’s account
7. Use cross account permissions to allow the team to call your lambda function
8. Each team will call another team’s lambda function to create a massive Rube Goldberg machine in the cloud
9. Develop a lambda function to deny access to the cross account role. Demonstrate this Event-Based Security action
10. Dylan’s Challenge Steps……

RULES TO REMEMBER

“You can’t used Managed Policies with Resource-based policies.”

–Jeff Wierer-

Senior IAM Manager

“It’s key that you understand the nuances of what is supported and what is Not, so you can create good policies”

–Jeff Wierer-

Senior IAM Manager

“Create separate policy statements: One for your bucket, and then a separate one for your objects in the bucket.”

–Adrian Drummond-

DevOps Dude