AWS Identity and Access Management (IAM) made easy with Terraform

Kala Maturi, Technology Services
Yoon Lee, Technology Services

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS
IT PROFESSIONALS FORUM
AT URBANA - CHAMPAIGN

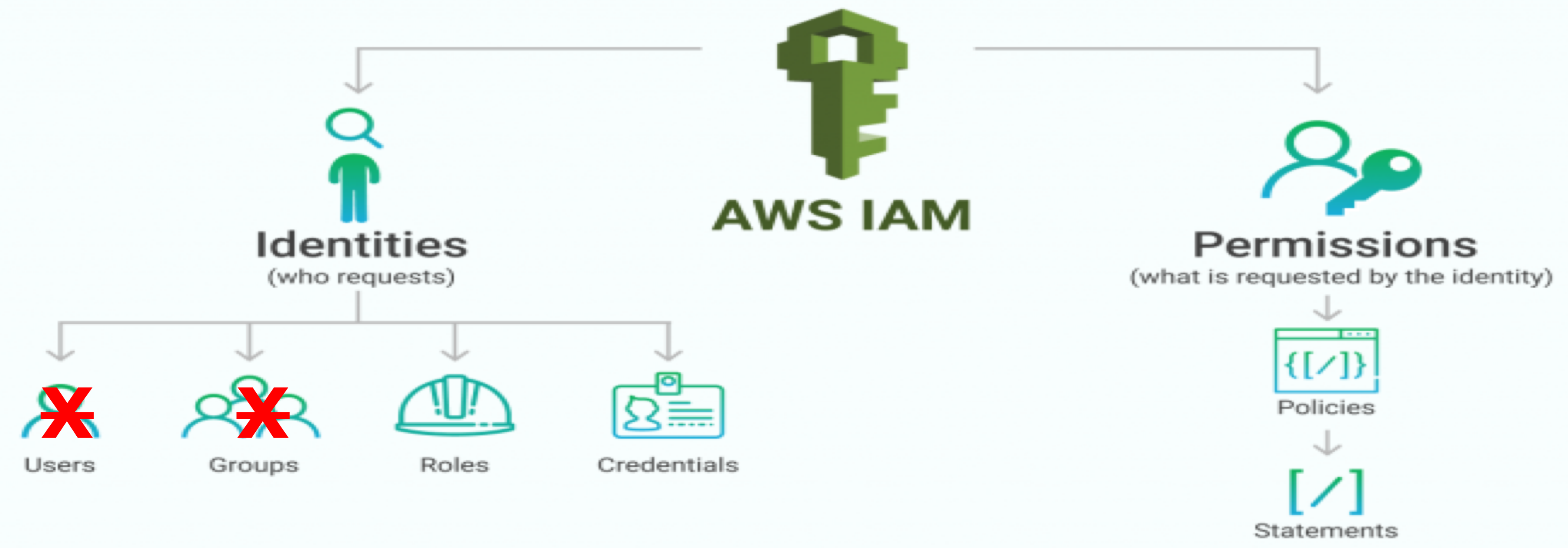
Topics

- AWS Authentication
- AWS Authorization
- About Roles & Policies
- Best practices
- Terraform code for IAM policy and role
- AWS IAM demo

AWS IAM (Identity and Access Management)

- AWS IAM is a web service that can be used to securely control access to AWS resources
- IAM can be used to control who can use AWS resources (authentication)
- IAM lets you manage which AWS resources can be accessed in what ways (authorization)

AWS IAM (Identity and Access Management)



Authentication

- What is an IAM role?
  - IAM Role is an IAM identity that you can create in your account that has specific permissions
- AD (Active Directory) and Shibboleth attributes are used in granting access to AWS accounts

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS
IT PROFESSIONALS FORUM
AT URBANA - CHAMPAIGN

Naming convention for IAM roles

- Role names in AD (Active Directory)
  - AWS-<Account ID>-<RoleName>
  - Example: AWS-XXXXXXXXXXXX-KalturaAdmin

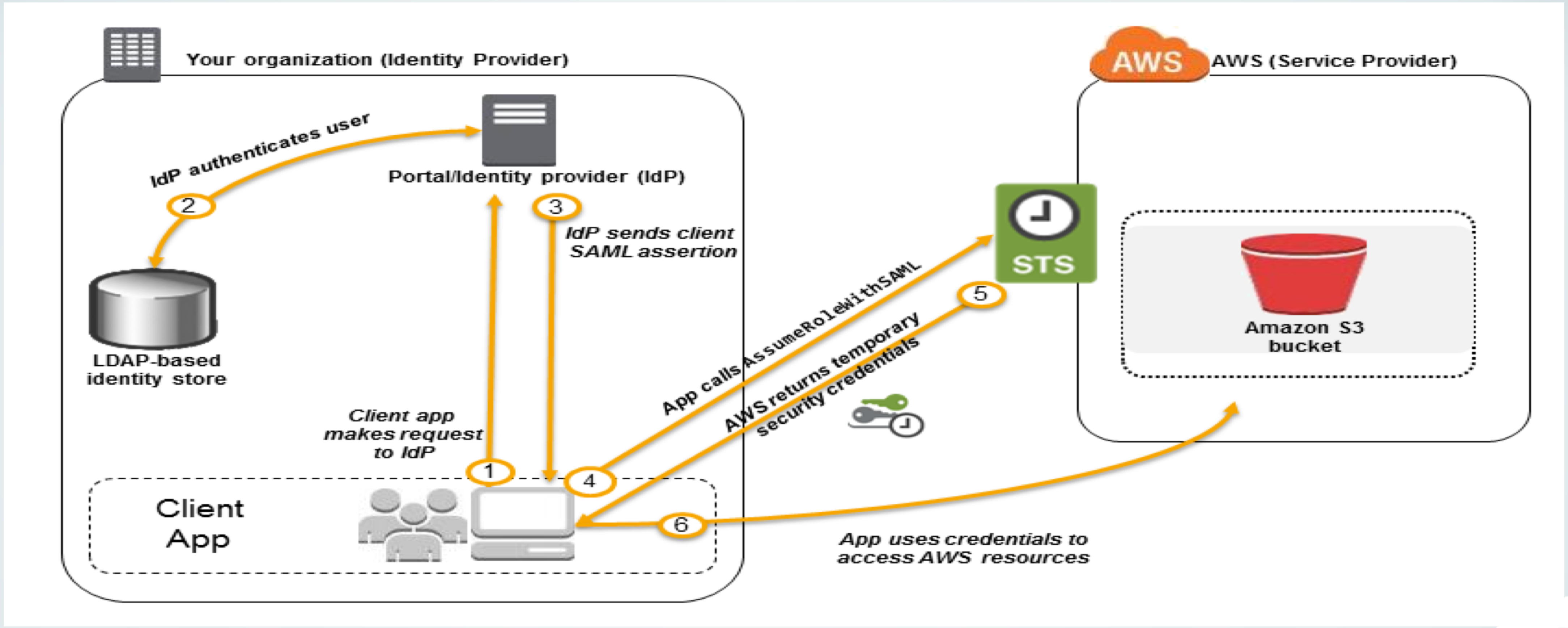
BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS
IT PROFESSIONALS FORUM
AT URBANA - CHAMPAIGN

Naming convention for IAM roles

- Role names in AWS
  - ServiceNameAdmin
  - Example: KalturaAdmin
  - AccountAdmins (devops group)
  - Example:ApplicationServicesAdmins

AuthN & AuthZ


Your organization (Identity Provider)
IdP authenticates user
2
Portal/Identity provider (IdP)
3
IdP sends client SAML assertion
LDAP-based identity store
Client app makes request to IdP
1
4
Client App
App calls AssumeRoleWithSAML
AWS returns temporary security credentials
5
STS
AWS
AWS (Service Provider)
Amazon S3 bucket
6
App uses credentials to access AWS resources

AuthN & AuthZ

- Client application makes a sign-in request to organizations IdP to log in
- IdP authenticates the user and generates a SAML authentication response which includes assertions that identify the user and include attributes about the user

AuthN & AuthZ

- Application then makes an unsigned call to STS (Security Token Service) with the AssumeRoleWithSAML action to request temporary security credentials
- Application passes the ARN of the SAML provider, the ARN of the role to assume, the SAML assertion about the current user returned by IdP

AuthN & AuthZ

- AWS verifies the SAML assertion is trusted and valid, if so returns temporary security credentials that have the permissions for the role named in the request
- Using the temporary security credentials the application makes signed requests to AWS to access the services

About Roles

- AWS permissions are granted to a user by associating the user with a role
- A user can be associated with multiple roles
- Each role has one or more policies attached

What is an IAM Policy ?

- A policy is a document which defines the actions that a user can perform on an Amazon resource
  - Actions example: GetObject/PutObject in S3 or RestartAppServer in Elastic Beanstalk
- A Terraform policy document contains statement, actions, resources and a condition

Designing Policies

- How to determine access needs for Service Admins?
    - Meet with Service Admins to gather requirements
        - Example: Few Authman Admin requirements
    - Able to pull and push images to ECR
    - Ability to kill tasks in ECS instance
    - Ability to do the snapshots of the RDS database

Designing Policies

- Design and create custom IAM policies
  - Able to pull and push images to ECR

- Created custom policy called -- ecr-authman-rw
  - Restricted access to repository -- authman

- Attach policies to the roles

BREAKING BARRIERS SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

Best Practices

- Principle of least privilege
- Use “Access Advisor” in the AWS Console to track permissions
- Enable multi-factor authentication
- Do regular audits of roles and members
- Use STS(Security Token Service) instead of storing access keys

Scenario:Amazon S3 access

- A user needs to access to S3 bucket called ‘itpro-demo’
- User should be able to download, upload and delete files within that bucket

Terraform IAM policy code

Data source block

```
data “aws_iam_policy_document” “default” {
  statement {
    actions = [ “S3:ListBucket”,
                “S3:GetBucketLocation”, ]
    resources  = [“arn:aws:s3:::itpro-demo”]
  }
}
```

Terraform IAM policy code

```
statement {
   actions = [“S3:GetObject”,
              “S3:PutObject”,
              “S3:DeleteOject”, ]
   resources  = [“arn:aws:s3:::itpro-demo/*”]
}
```

Terraform IAM policy code

```
statement {
   actions = [“S3:ListAllMyBuckets”, ]
   resources  = [“arn:aws:s3:::*”]
}
```

Terraform IAM policy code

Resource block

```
resource “aws_iam_policy” “default” {
   name = “S3BucketAccess”
   path = “/”
   description = “Policy that allows access to S3
   bucket”
   policy =
   “${data.aws_iam_policy_document.default.json}”
 }
```

Terraform IAM role code

Resource block

```
resource “aws_iam_role” “default” {
   name = “testrole”
   description = “Test role for ITPF demo”
      assume_role_policy =
      “${data.aws_iam_policy_document.saml.json}”
}
```

BREAKING BARRIERS SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

Terraform IAM role code

Data source block

```
data “aws_iam_policy_document” “saml” {
   statement {
   actions = [“sts:AssumeRolewithSAML”]
   principals {
      type = “Federated”
   identifiers =
   [“arn:aws:iam::XXXXXXXXXXXX:saml-
provider/shibboleth.illinois.edu”]
}
```

BREAKING BARRIERS SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

Terraform IAM role code

```
        condition {
          test = “StringEquals”
          variable = “SAML:aud”
          values =
        [“https://signin.aws.amazon.com/saml”]
        }
      }
    }
```

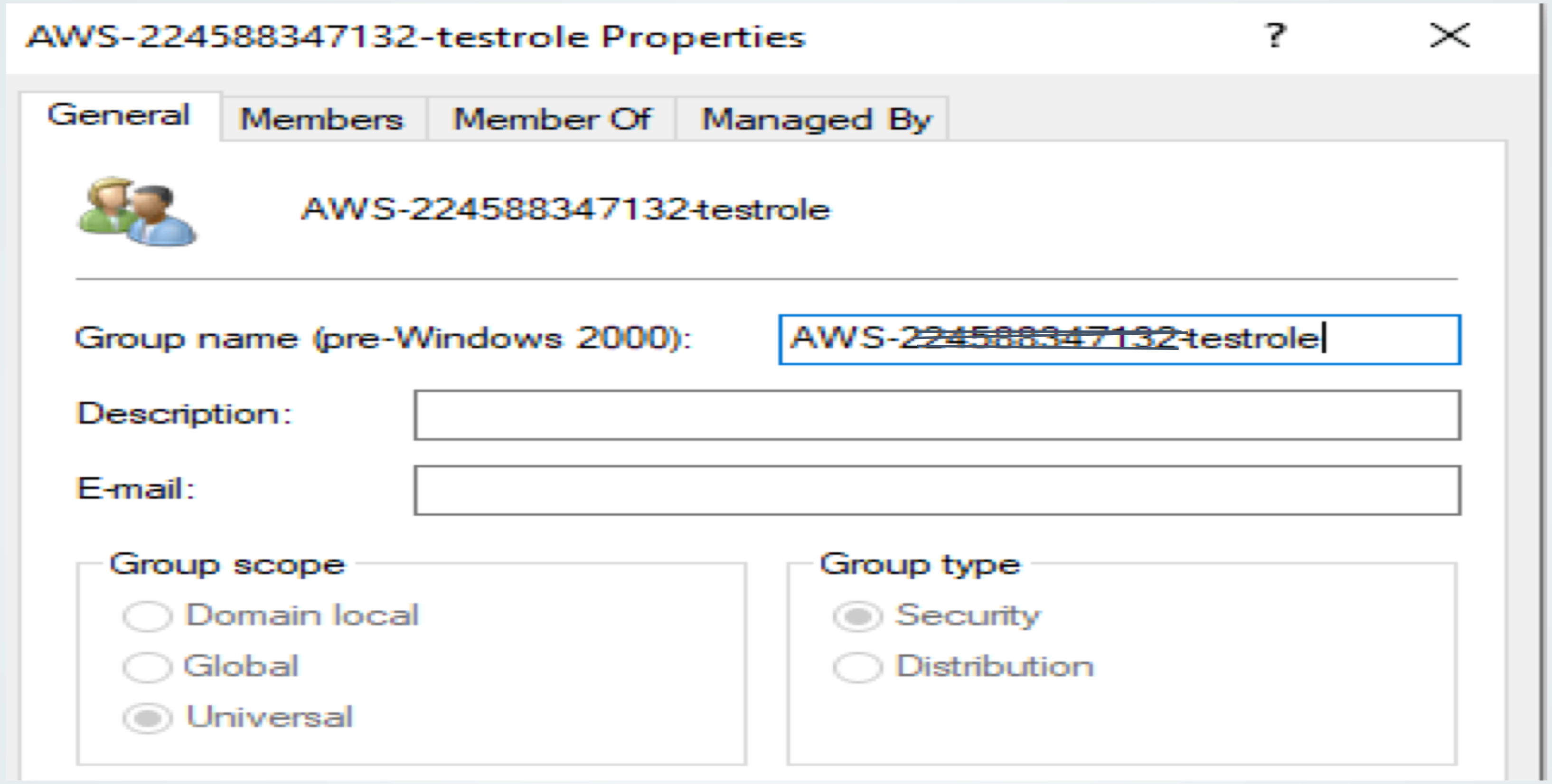
BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

Attaching policy to the role

```
resource “aws_iam_policy_attachment” “test-attach” {
   name = “S3BucketAccess”
   roles = [“${aws_iam_role.default.name}”]
   policy_arn =
   “arn:aws:iam::XXXXXXXXXXXX:policy/S3BucketAccess”
}
```

Role in AD group



AWS-224588347132-testrole Properties

AWS-224588347132-testrole

Group name (pre-Windows 2000): AWS-~~224588347132~~-testrole

Description:

E-mail:

Group scope: Domain local / Global / [✓ Universal]

Group type: [✓ Security] / Distribution

BREAKING BARRIERS SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

Demo

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS
IT PROFESSIONALS FORUM
AT URBANA - CHAMPAIGN

References

- AWS IAM Documentation
  https://aws.amazon.com/documentation/iam/
- IAM Best Practices to Live By
  https://youtu.be/_wiGpBQGCjU (52:49)
- How to Become an IAM Policy Ninja
  https://youtu.be/y7-fAT3z8Lo (55:38)

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

References

- IAM Role
  http://jayendrapatil.com/tag/iam-role/
- Granting access to the AWS Console
  https://tinyurl.com/yyzb3a4q
- Introduction to Terraform
  https://www.terraform.io/intro/index.html

References

- GitHub Repo for example Terraform code
  https://tinyurl.com/yy53f33b

Questions ?

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS
IT PROFESSIONALS FORUM
AT URBANA - CHAMPAIGN

Contact

- Kala Maturi – cmaturi@Illinois.edu
- Yoon Lee – yoonlees@Illinois.edu

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS IT PROFESSIONALS FORUM AT URBANA - CHAMPAIGN

Thank you!

BREAKING BARRIERS
SPRING 2019

UNIVERSITY OF ILLINOIS
IT PROFESSIONALS FORUM
AT URBANA - CHAMPAIGN