IAM AUTOMATION WITH EASE

ANSIBLE OR HashiCorp Terraform

THE CLOUD BOOTCAMP

Andre Torres / May 2021

Real case
PROBLEM

A company is migrating to AWS. It needs to migrate the users, set their passwords, and add them to their respective groups. However, they only provided a .csv file with the date.

Approach #1

Using Ansible to migrate users and groups to AWS

Why using Ansible in automation

Simple

- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Usable by every team
- Get productive quickly

Powerful

- App deployment
- Configuration management
- Workflow orchestration
- Network automation
- Orchestrate the app lifecycle

Agentless

- Agentless architecture
- Uses OpenSSH & WinRM
- No agents to exploit or update
- Get started immediately
- More efficient & more secure

Overview

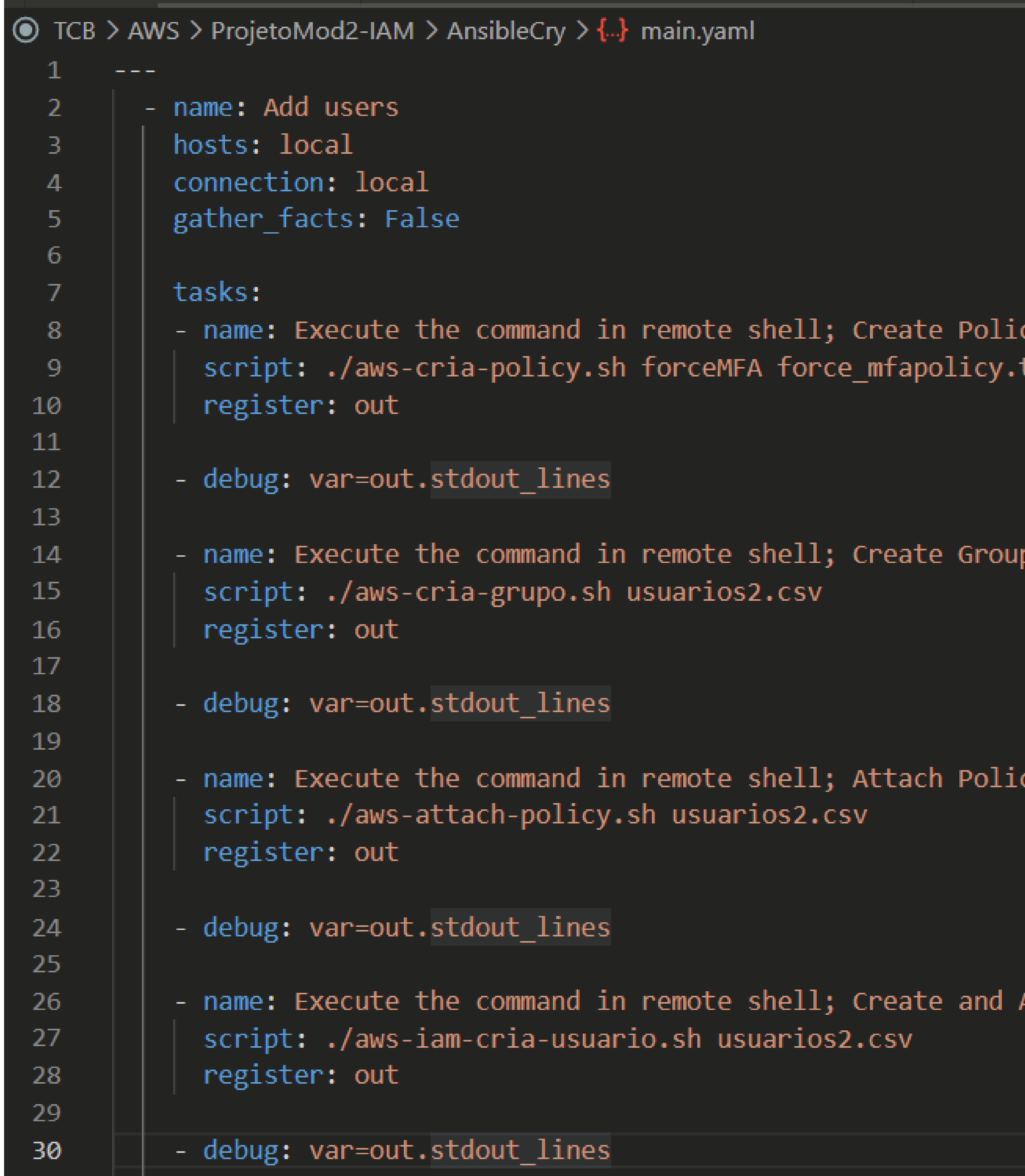
Used tools:

Local computer with AWS-CLI installed

Ansible

Steps:

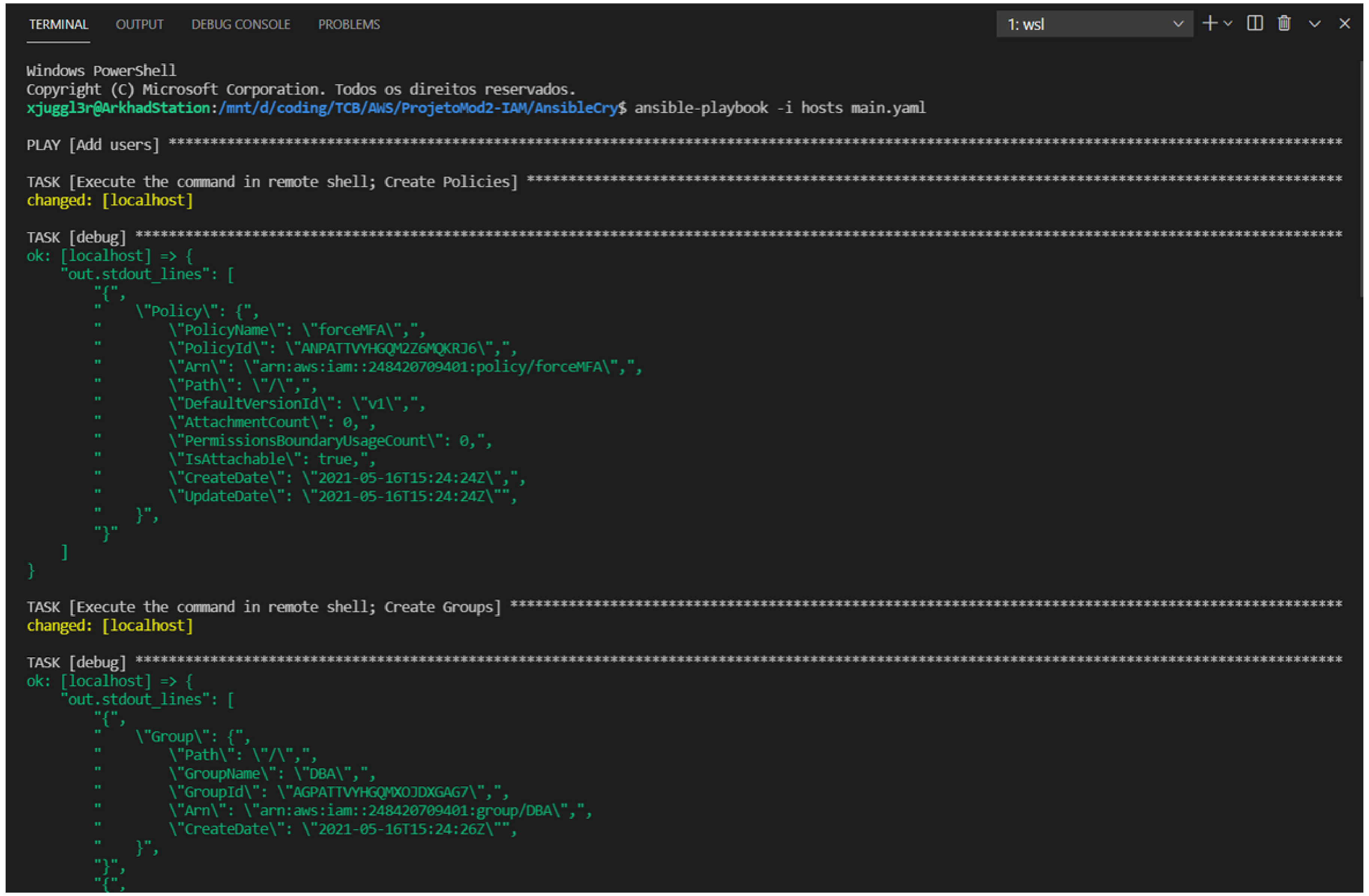
- Create Ansible's hosts file
- Create Ansible's main.yaml file
- Define order
  - Create Enforce MFA Policy to Groups
  - Find arn for other AWS Policies that will be used
  - Create Groups from the company's .csv file
  - Attach AWS-managed and Enforce MFA Policies to groups
  - Create Users
  - Add users to groups
- Create Bash scripts for each step
- Write main.yaml to connect to AWS and run the scripts



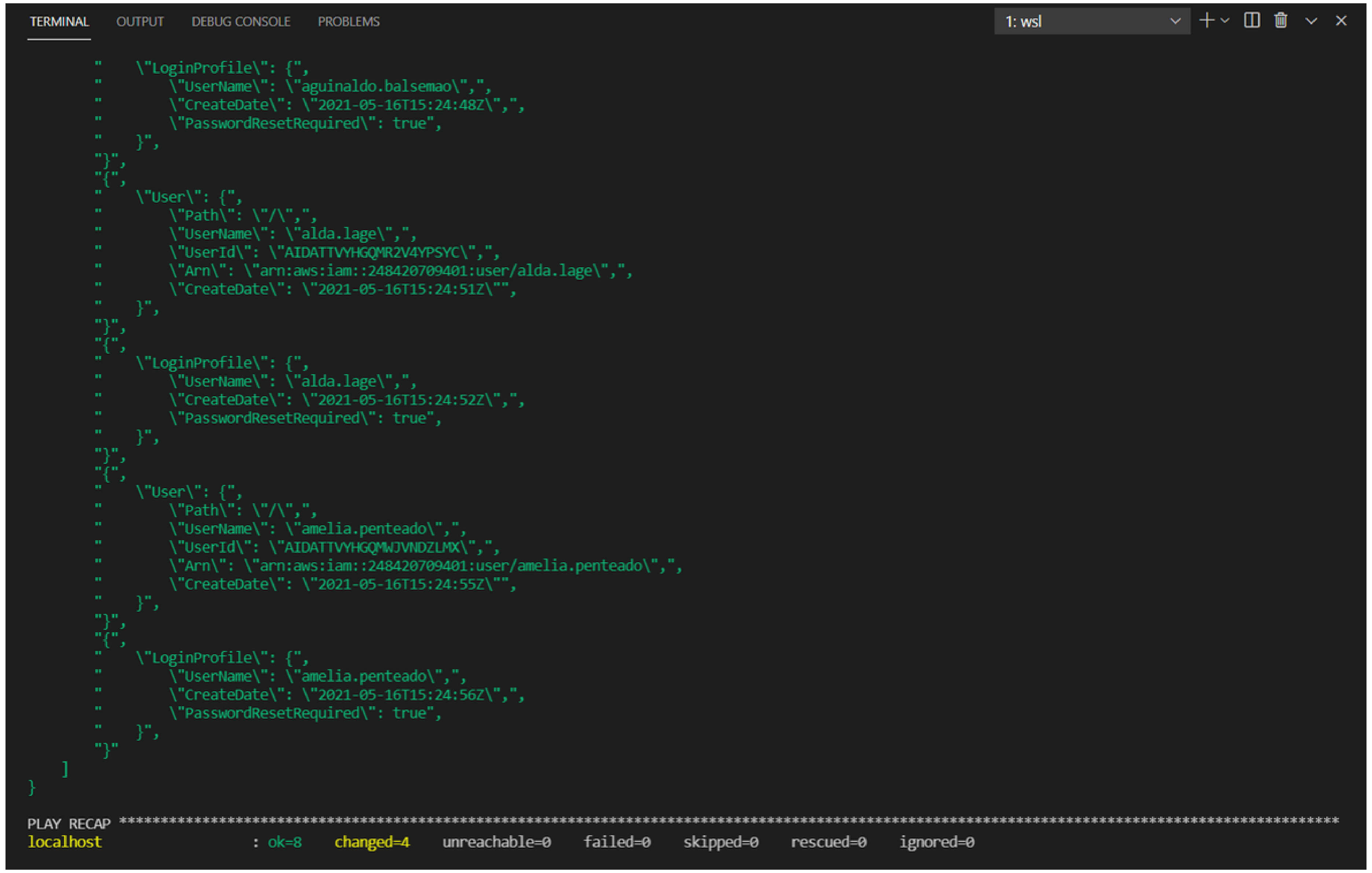
```yaml
---
- name: Add users
  hosts: local
  connection: local
  gather_facts: False

  tasks:
  - name: Execute the command in remote shell; Create Polic
    script: ./aws-cria-policy.sh forceMFA force_mfapolicy.t
    register: out

  - debug: var=out.stdout_lines

  - name: Execute the command in remote shell; Create Group
    script: ./aws-cria-grupo.sh usuarios2.csv
    register: out

  - debug: var=out.stdout_lines

  - name: Execute the command in remote shell; Attach Polic
    script: ./aws-attach-policy.sh usuarios2.csv
    register: out

  - debug: var=out.stdout_lines

  - name: Execute the command in remote shell; Create and A
    script: ./aws-iam-cria-usuario.sh usuarios2.csv
    register: out

  - debug: var=out.stdout_lines
```

Ansible in Action

Ansible starts the playbook, runs it, and creates the desired resources

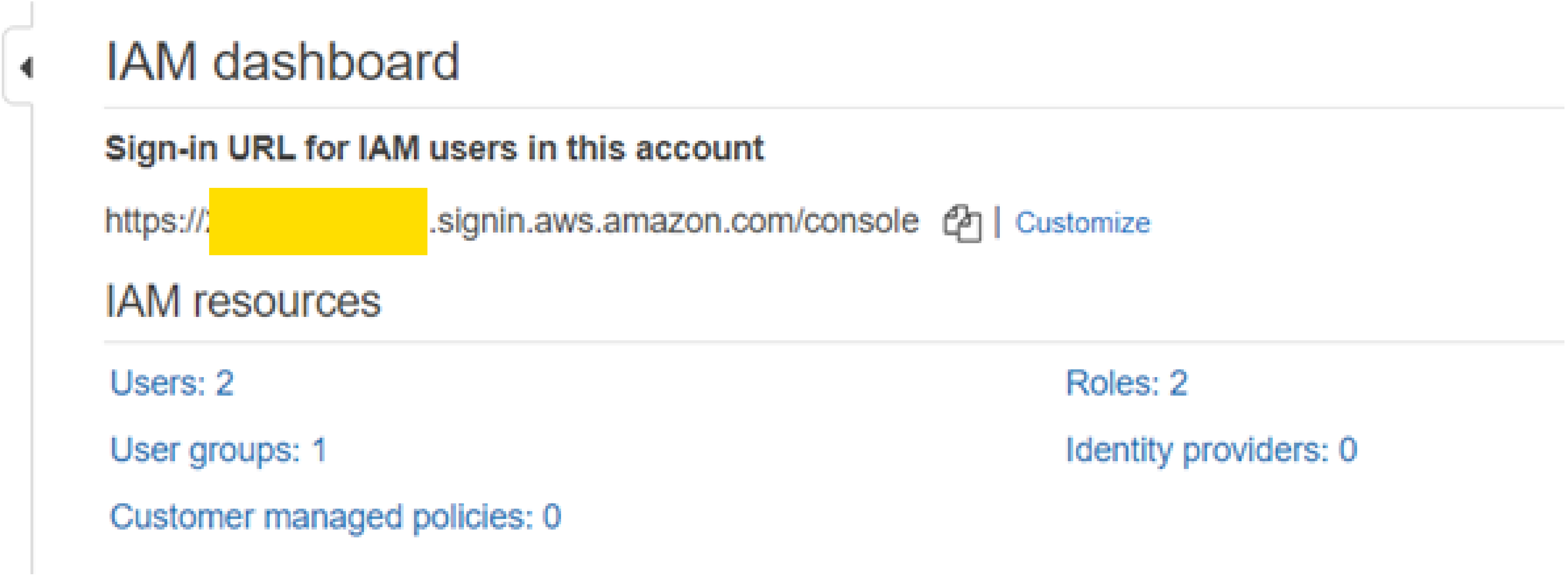


```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.
xjuggl3r@ArkhadStation:/mnt/d/coding/TCB/AWS/ProjetoMod2-IAM/AnsibleCry$ ansible-playbook -i hosts main.yaml

PLAY [Add users] *********************************************************************************************************

TASK [Execute the command in remote shell; Create Policies] **************************************************************
changed: [localhost]

TASK [debug] *************************************************************************************************************
ok: [localhost] => {
    "out.stdout_lines": [
        "{",
        "    \"Policy\": {",
        "        \"PolicyName\": \"forceMFA\",",
        "        \"PolicyId\": \"ANPATTVYHGQM2Z6MQKRJ6\",",
        "        \"Arn\": \"arn:aws:iam::248420709401:policy/forceMFA\",",
        "        \"Path\": \"/\",",
        "        \"DefaultVersionId\": \"v1\",",
        "        \"AttachmentCount\": 0,",
        "        \"PermissionsBoundaryUsageCount\": 0,",
        "        \"IsAttachable\": true,",
        "        \"CreateDate\": \"2021-05-16T15:24:24Z\",",
        "        \"UpdateDate\": \"2021-05-16T15:24:24Z\"",
        "    }",
        "}"
    ]
}

TASK [Execute the command in remote shell; Create Groups] ****************************************************************
changed: [localhost]

TASK [debug] *************************************************************************************************************
ok: [localhost] => {
    "out.stdout_lines": [
        "{",
        "    \"Group\": {",
        "        \"Path\": \"/\",",
        "        \"GroupName\": \"DBA\",",
        "        \"GroupId\": \"AGPATTVYHGQMXOJDXGAG7\",",
        "        \"Arn\": \"arn:aws:iam::248420709401:group/DBA\",",
        "        \"CreateDate\": \"2021-05-16T15:24:26Z\"",
        "    }",
        "}",
        "{",
```



```
        "    \"LoginProfile\": {",
        "        \"UserName\": \"aguinaldo.balsemao\",",
        "        \"CreateDate\": \"2021-05-16T15:24:48Z\",",
        "        \"PasswordResetRequired\": true",
        "    }",
        "}",
        "{",
        "    \"User\": {",
        "        \"Path\": \"/\",",
        "        \"UserName\": \"alda.lage\",",
        "        \"UserId\": \"AIDATTVYHGQMR2V4YPSYC\",",
        "        \"Arn\": \"arn:aws:iam::248420709401:user/alda.lage\",",
        "        \"CreateDate\": \"2021-05-16T15:24:51Z\"",
        "    }",
        "}",
        "{",
        "    \"LoginProfile\": {",
        "        \"UserName\": \"alda.lage\",",
        "        \"CreateDate\": \"2021-05-16T15:24:52Z\",",
        "        \"PasswordResetRequired\": true",
        "    }",
        "}",
        "{",
        "    \"User\": {",
        "        \"Path\": \"/\",",
        "        \"UserName\": \"amelia.penteado\",",
        "        \"UserId\": \"AIDATTVYHGQMWJVNDZLMX\",",
        "        \"Arn\": \"arn:aws:iam::248420709401:user/amelia.penteado\",",
        "        \"CreateDate\": \"2021-05-16T15:24:55Z\"",
        "    }",
        "}",
        "{",
        "    \"LoginProfile\": {",
        "        \"UserName\": \"amelia.penteado\",",
        "        \"CreateDate\": \"2021-05-16T15:24:56Z\",",
        "        \"PasswordResetRequired\": true",
        "    }",
        "}"
    ]
}

PLAY RECAP ***************************************************************************************************************
localhost                  : ok=8    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Results



IAM dashboard

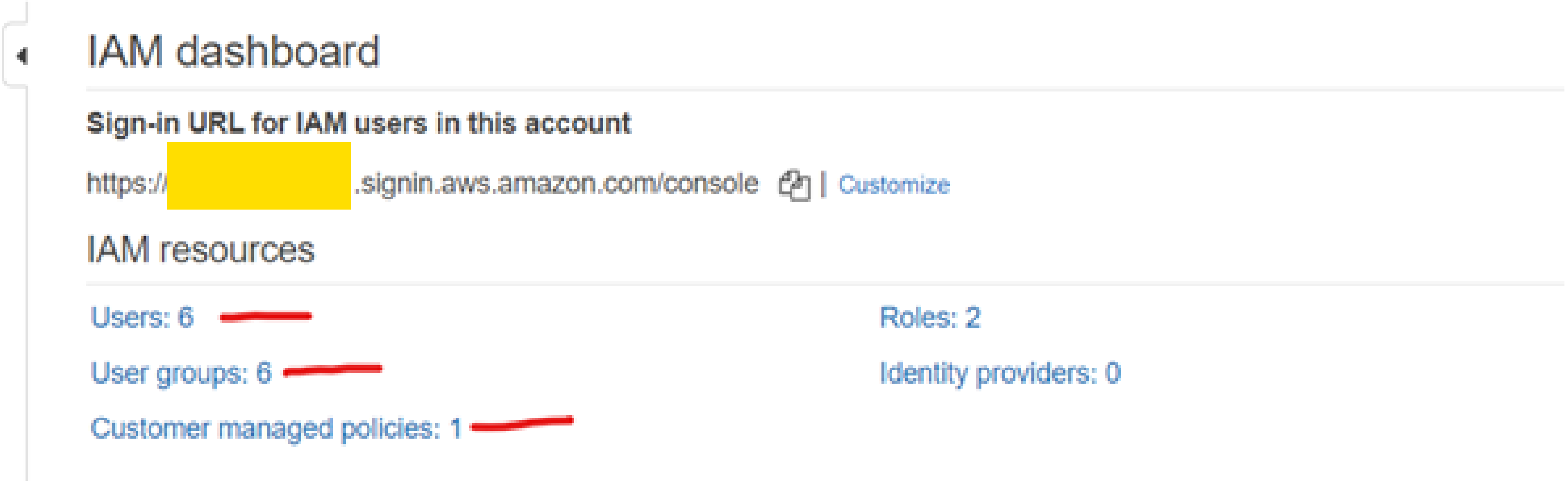
Sign-in URL for IAM users in this account

https://[redacted].signin.aws.amazon.com/console | Customize

IAM resources

Users: 2	Roles: 2
User groups: 1	Identity providers: 0
Customer managed policies: 0	

Before



IAM dashboard

Sign-in URL for IAM users in this account

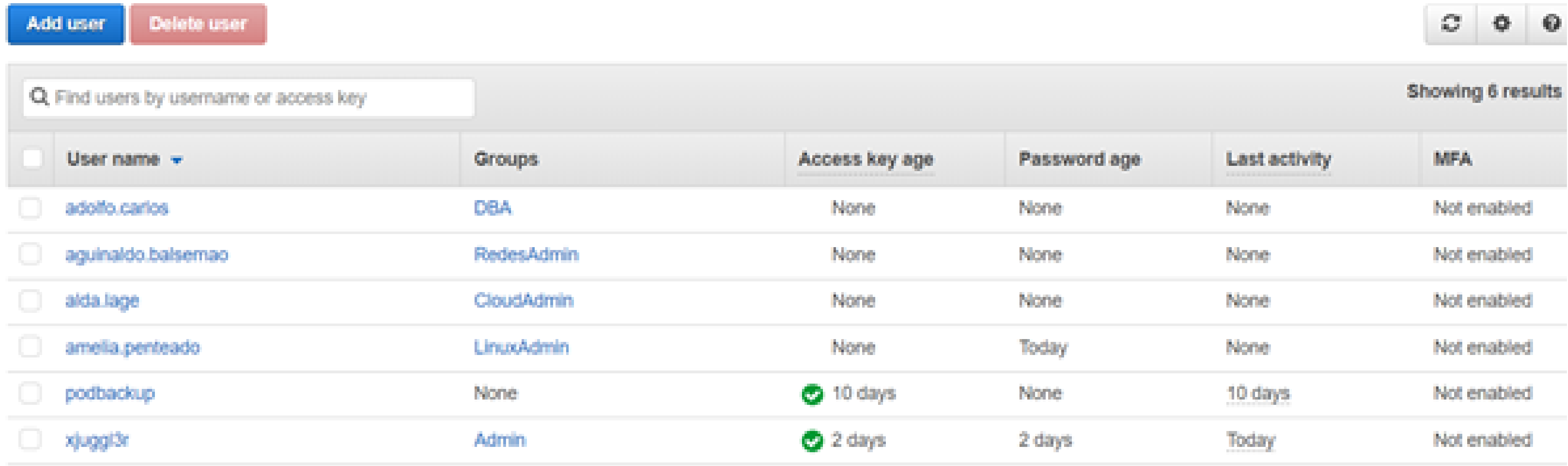
https://[redacted].signin.aws.amazon.com/console | Customize

IAM resources

Users: 6	Roles: 2
User groups: 6	Identity providers: 0
Customer managed policies: 1	

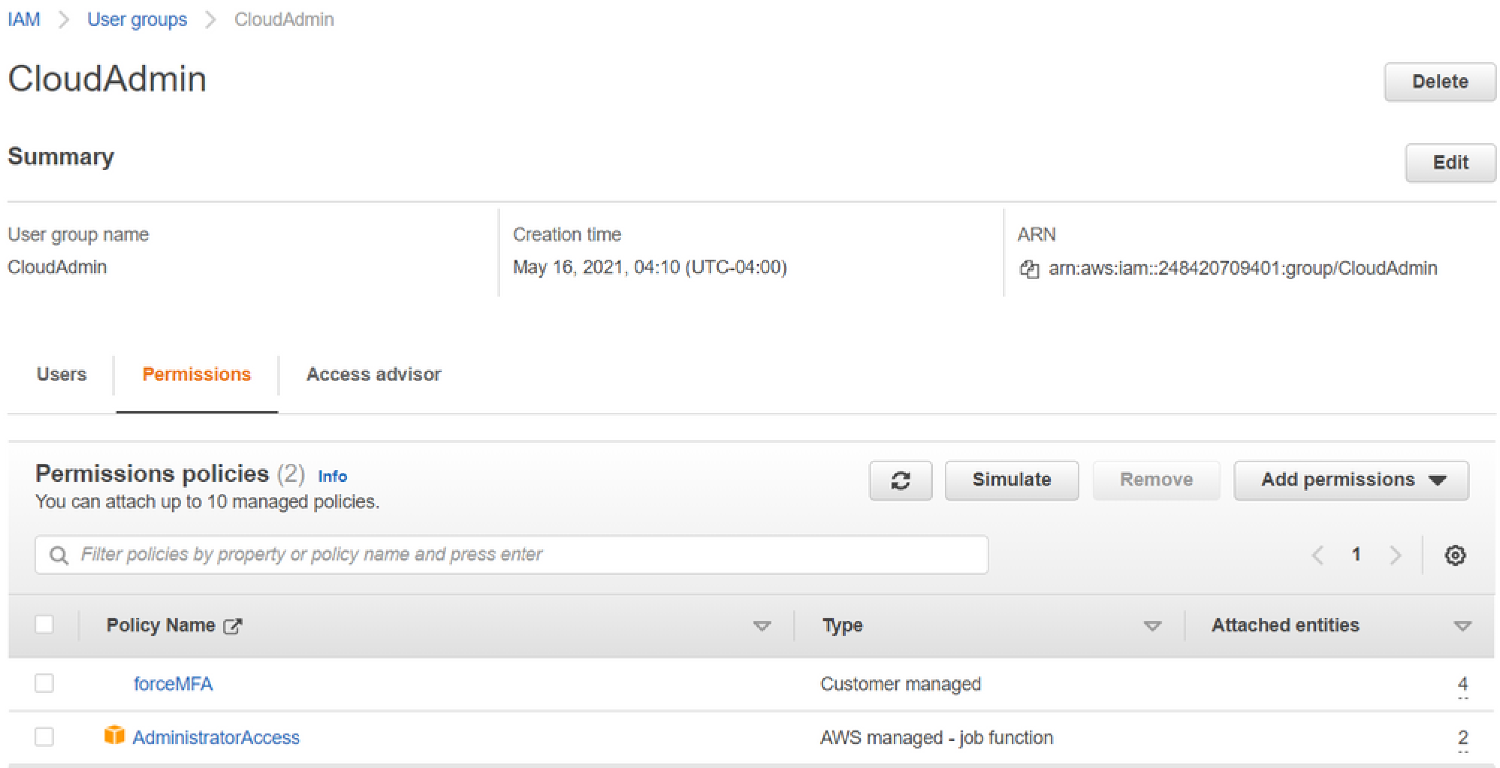
After

Results



□	User name	Groups	Access key age	Password age	Last activity	MFA
□	adolfo.carlos	DBA	None	None	None	Not enabled
□	aguinaldo.balsemao	RedesAdmin	None	None	None	Not enabled
□	alda.lage	CloudAdmin	None	None	None	Not enabled
□	amelia.penteado	LinuxAdmin	None	Today	None	Not enabled
□	podbackup	None	✓ 10 days	None	10 days	Not enabled
□	xjuggl3r	Admin	✓ 2 days	2 days	Today	Not enabled

Users in groups


IAM > User groups > CloudAdmin
Delete
Edit
Users
Permissions
Access advisor
Simulate
Remove
Add permissions
Filter policies by property or policy name and press enter
1

CloudAdmin

Summary

User group name
CloudAdmin

Creation time
May 16, 2021, 04:10 (UTC-04:00)

ARN
arn:aws:iam::248420709401:group/CloudAdmin

Permissions policies (2) Info
You can attach up to 10 managed policies.

□	Policy Name	Type	Attached entities
□	forceMFA	Customer managed	4
□	AdministratorAccess	AWS managed - job function	2

Groups with policies

Approach #2

Using Terraform to migrate users and groups to AWS

Why using Terraform in automation

Simple

- Human readable automation
- Cloud Agnostic
- Management is easy
- History of infrastructure changes
- Open-source with community support

Powerful

- Resource deployment
- Easy rollbacks
- Large ecosystem of modules
- Variables can be created to make generic templates so they can be reused

Agentless

- Agentless architecture
- Masterless - no extra pushes
- No agents to exploit or update
- Get started immediately
- More efficient & more secure

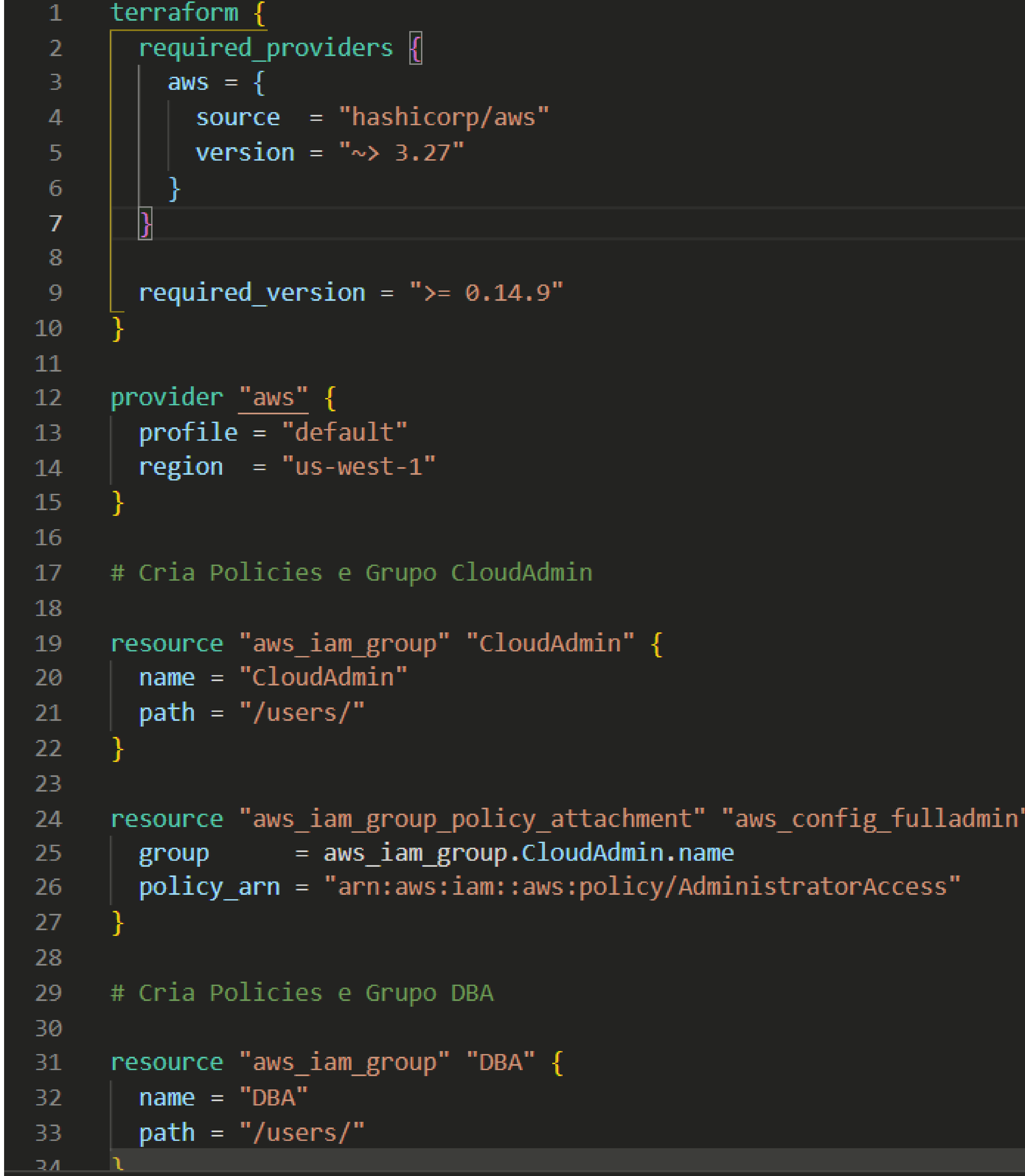
Overview

Used tools:

Local computer with Terraform Installed

Steps:

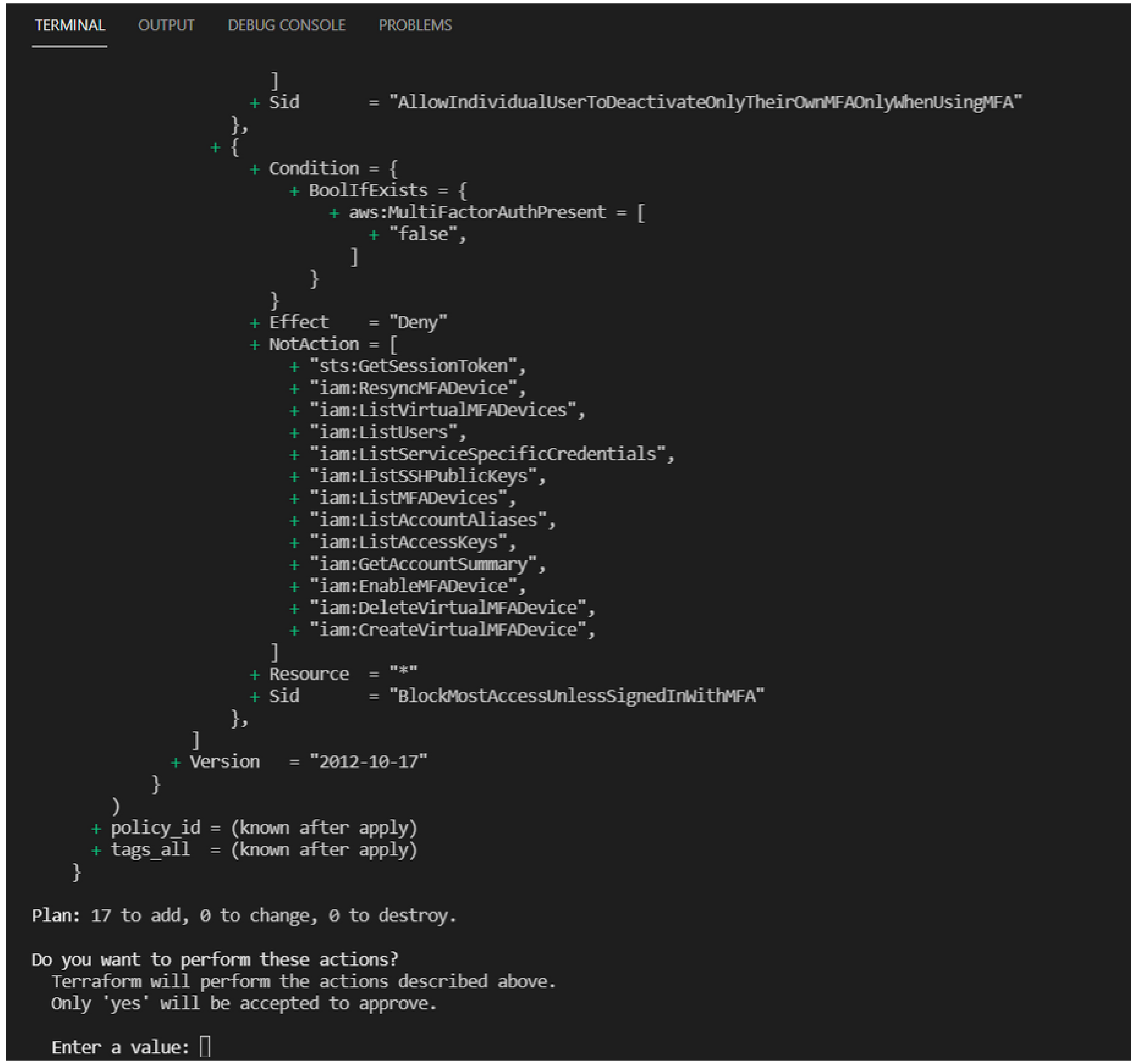
- Create Terraform's main.tf file with all config or modularize each process
- Define order
  - Create Enforce MFA Policy to Groups
  - Find arn for other AWS Policies that will be used
  - Create Groups from the company's .csv file
  - Attach AWS-managed and Enforce MFA Policies to groups
  - Create Users
  - Add users to groups
- Create Bash scripts for each step
- Write main.tf to connect to AWS and execute tasks



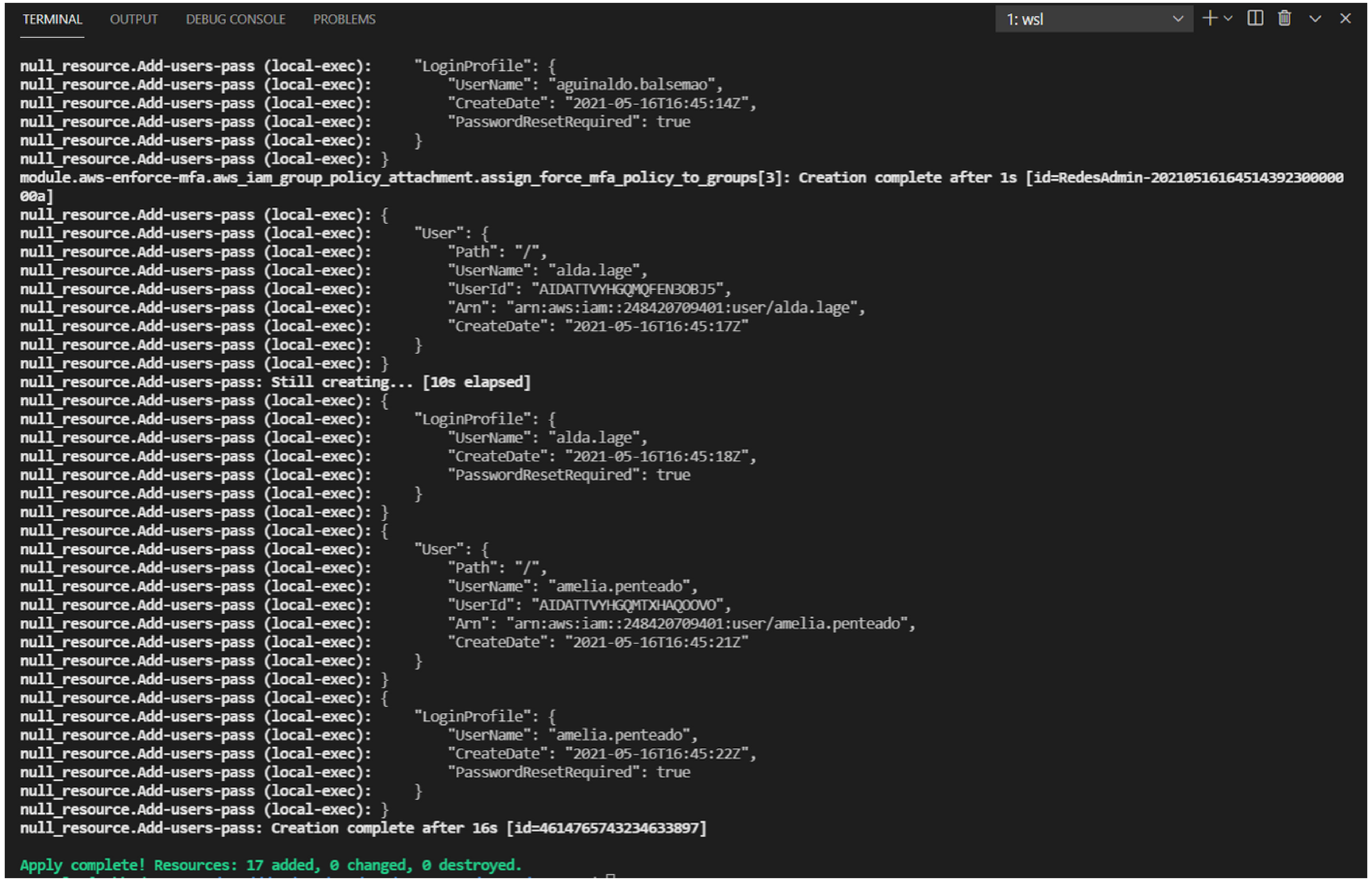
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }

  required_version = ">= 0.14.9"
}

provider "aws" {
  profile = "default"
  region  = "us-west-1"
}

# Cria Policies e Grupo CloudAdmin

resource "aws_iam_group" "CloudAdmin" {
  name = "CloudAdmin"
  path = "/users/"
}

resource "aws_iam_group_policy_attachment" "aws_config_fulladmin"
  group      = aws_iam_group.CloudAdmin.name
  policy_arn = "arn:aws:iam::aws:policy/AdministratorAccess"
}

# Cria Policies e Grupo DBA

resource "aws_iam_group" "DBA" {
  name = "DBA"
  path = "/users/"
}
```

Terraform in Action

Terraform starts the workspace, runs it, and creates the desired resources

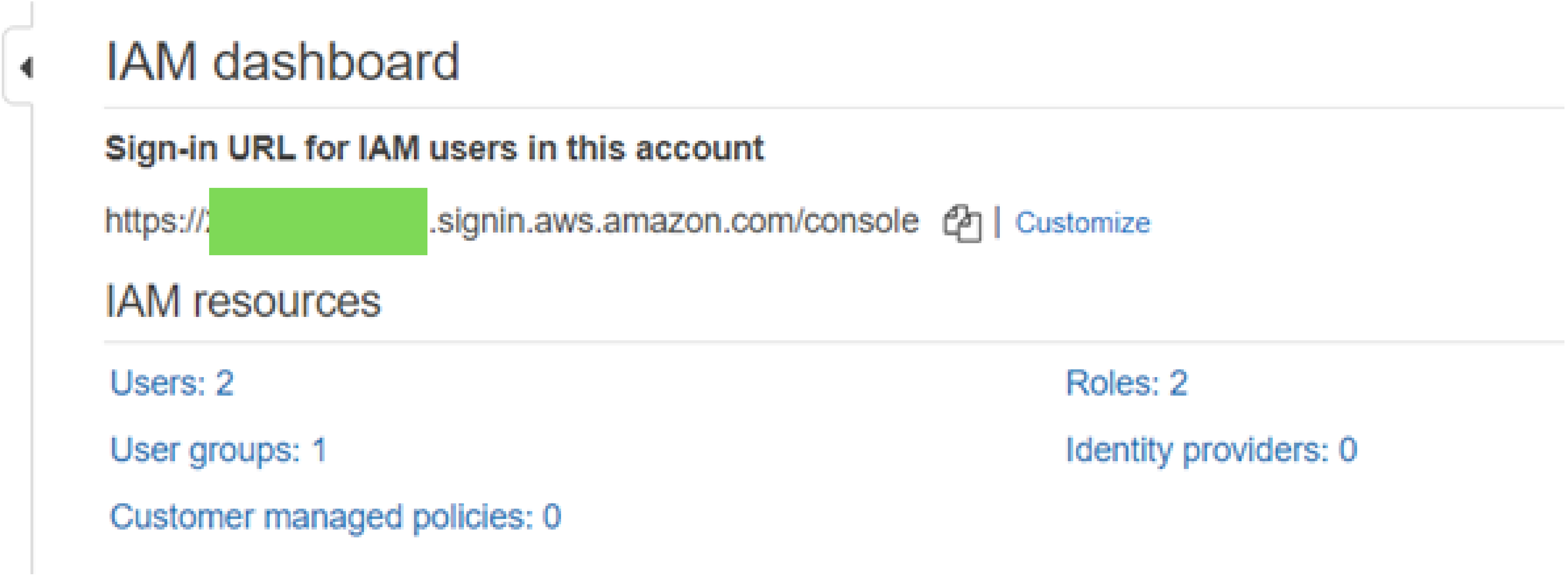


```
                       ]
                     + Sid       = "AllowIndividualUserToDeactivateOnlyTheirOwnMFAOnlyWhenUsingMFA"
                   },
                 + {
                     + Condition = {
                         + BoolIfExists = {
                             + aws:MultiFactorAuthPresent = [
                                 + "false",
                               ]
                           }
                       }
                     + Effect    = "Deny"
                     + NotAction = [
                         + "sts:GetSessionToken",
                         + "iam:ResyncMFADevice",
                         + "iam:ListVirtualMFADevices",
                         + "iam:ListUsers",
                         + "iam:ListServiceSpecificCredentials",
                         + "iam:ListSSHPublicKeys",
                         + "iam:ListMFADevices",
                         + "iam:ListAccountAliases",
                         + "iam:ListAccessKeys",
                         + "iam:GetAccountSummary",
                         + "iam:EnableMFADevice",
                         + "iam:DeleteVirtualMFADevice",
                         + "iam:CreateVirtualMFADevice",
                       ]
                     + Resource  = "*"
                     + Sid       = "BlockMostAccessUnlessSignedInWithMFA"
                   },
                ]
              + Version   = "2012-10-17"
            }
        )
      + policy_id = (known after apply)
      + tags_all  = (known after apply)
    }

Plan: 17 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: 
```



```
null_resource.Add-users-pass (local-exec):     "LoginProfile": {
null_resource.Add-users-pass (local-exec):         "UserName": "aguinaldo.balsemao",
null_resource.Add-users-pass (local-exec):         "CreateDate": "2021-05-16T16:45:14Z",
null_resource.Add-users-pass (local-exec):         "PasswordResetRequired": true
null_resource.Add-users-pass (local-exec):     }
null_resource.Add-users-pass (local-exec): }
module.aws-enforce-mfa.aws_iam_group_policy_attachment.assign_force_mfa_policy_to_groups[3]: Creation complete after 1s [id=RedesAdmin-2021051616451439230000
00a]
null_resource.Add-users-pass (local-exec): {
null_resource.Add-users-pass (local-exec):     "User": {
null_resource.Add-users-pass (local-exec):         "Path": "/",
null_resource.Add-users-pass (local-exec):         "UserName": "alda.lage",
null_resource.Add-users-pass (local-exec):         "UserId": "AIDATTVYHGQMQFEN3OBJ5",
null_resource.Add-users-pass (local-exec):         "Arn": "arn:aws:iam::248420709401:user/alda.lage",
null_resource.Add-users-pass (local-exec):         "CreateDate": "2021-05-16T16:45:17Z"
null_resource.Add-users-pass (local-exec):     }
null_resource.Add-users-pass (local-exec): }
null_resource.Add-users-pass: Still creating... [10s elapsed]
null_resource.Add-users-pass (local-exec): {
null_resource.Add-users-pass (local-exec):     "LoginProfile": {
null_resource.Add-users-pass (local-exec):         "UserName": "alda.lage",
null_resource.Add-users-pass (local-exec):         "CreateDate": "2021-05-16T16:45:18Z",
null_resource.Add-users-pass (local-exec):         "PasswordResetRequired": true
null_resource.Add-users-pass (local-exec):     }
null_resource.Add-users-pass (local-exec): }
null_resource.Add-users-pass (local-exec): {
null_resource.Add-users-pass (local-exec):     "User": {
null_resource.Add-users-pass (local-exec):         "Path": "/",
null_resource.Add-users-pass (local-exec):         "UserName": "amelia.penteado",
null_resource.Add-users-pass (local-exec):         "UserId": "AIDATTVYHGQMTXHAQOOVO",
null_resource.Add-users-pass (local-exec):         "Arn": "arn:aws:iam::248420709401:user/amelia.penteado",
null_resource.Add-users-pass (local-exec):         "CreateDate": "2021-05-16T16:45:21Z"
null_resource.Add-users-pass (local-exec):     }
null_resource.Add-users-pass (local-exec): }
null_resource.Add-users-pass (local-exec): {
null_resource.Add-users-pass (local-exec):     "LoginProfile": {
null_resource.Add-users-pass (local-exec):         "UserName": "amelia.penteado",
null_resource.Add-users-pass (local-exec):         "CreateDate": "2021-05-16T16:45:22Z",
null_resource.Add-users-pass (local-exec):         "PasswordResetRequired": true
null_resource.Add-users-pass (local-exec):     }
null_resource.Add-users-pass (local-exec): }
null_resource.Add-users-pass: Creation complete after 16s [id=4614765743234633897]

Apply complete! Resources: 17 added, 0 changed, 0 destroyed.
```

Results



IAM dashboard

Sign-in URL for IAM users in this account

https://[redacted].signin.aws.amazon.com/console | Customize

IAM resources

Users: 2

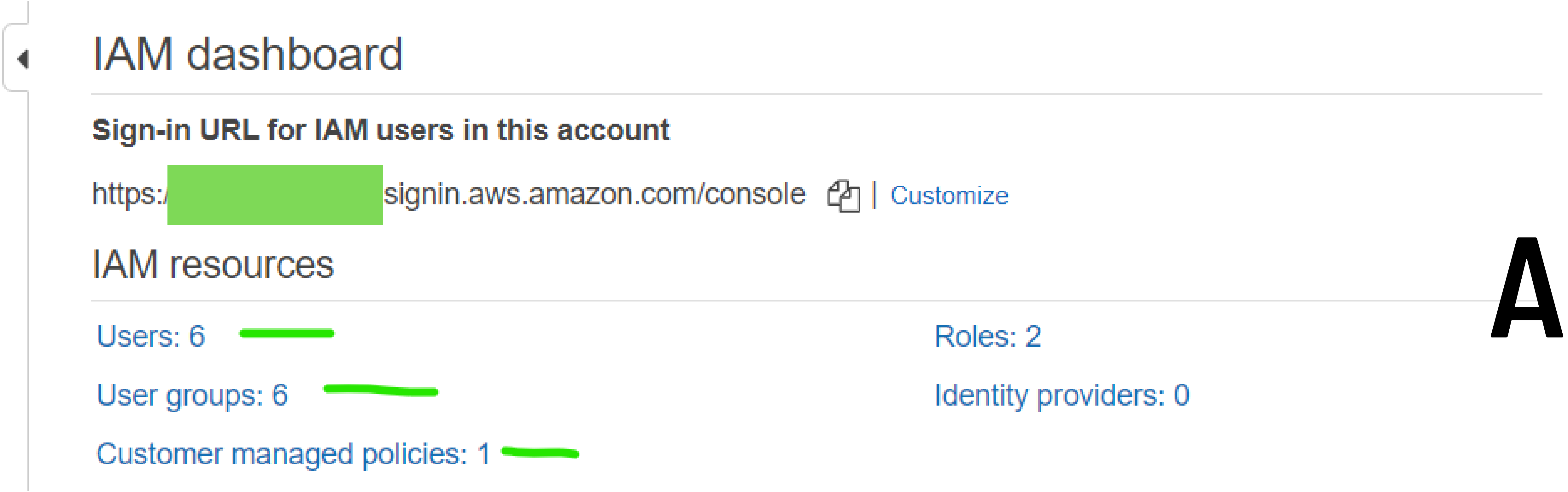
Roles: 2

User groups: 1

Identity providers: 0

Customer managed policies: 0

Before



IAM dashboard

Sign-in URL for IAM users in this account

https:/[redacted]signin.aws.amazon.com/console | Customize

IAM resources

Users: 6

Roles: 2

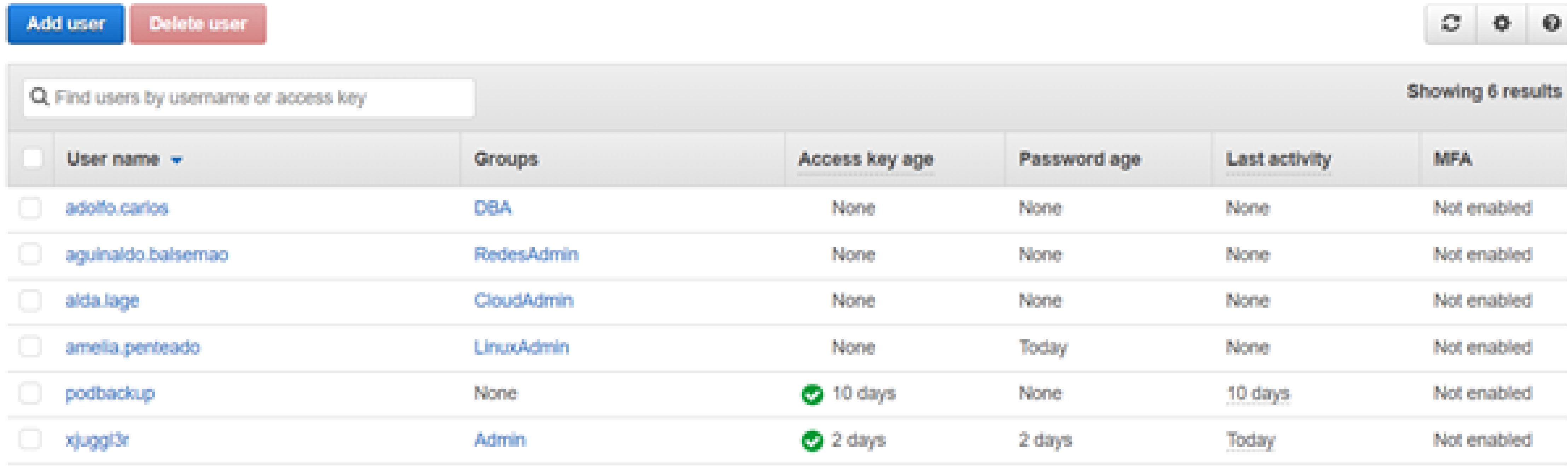
User groups: 6

Identity providers: 0

Customer managed policies: 1

After

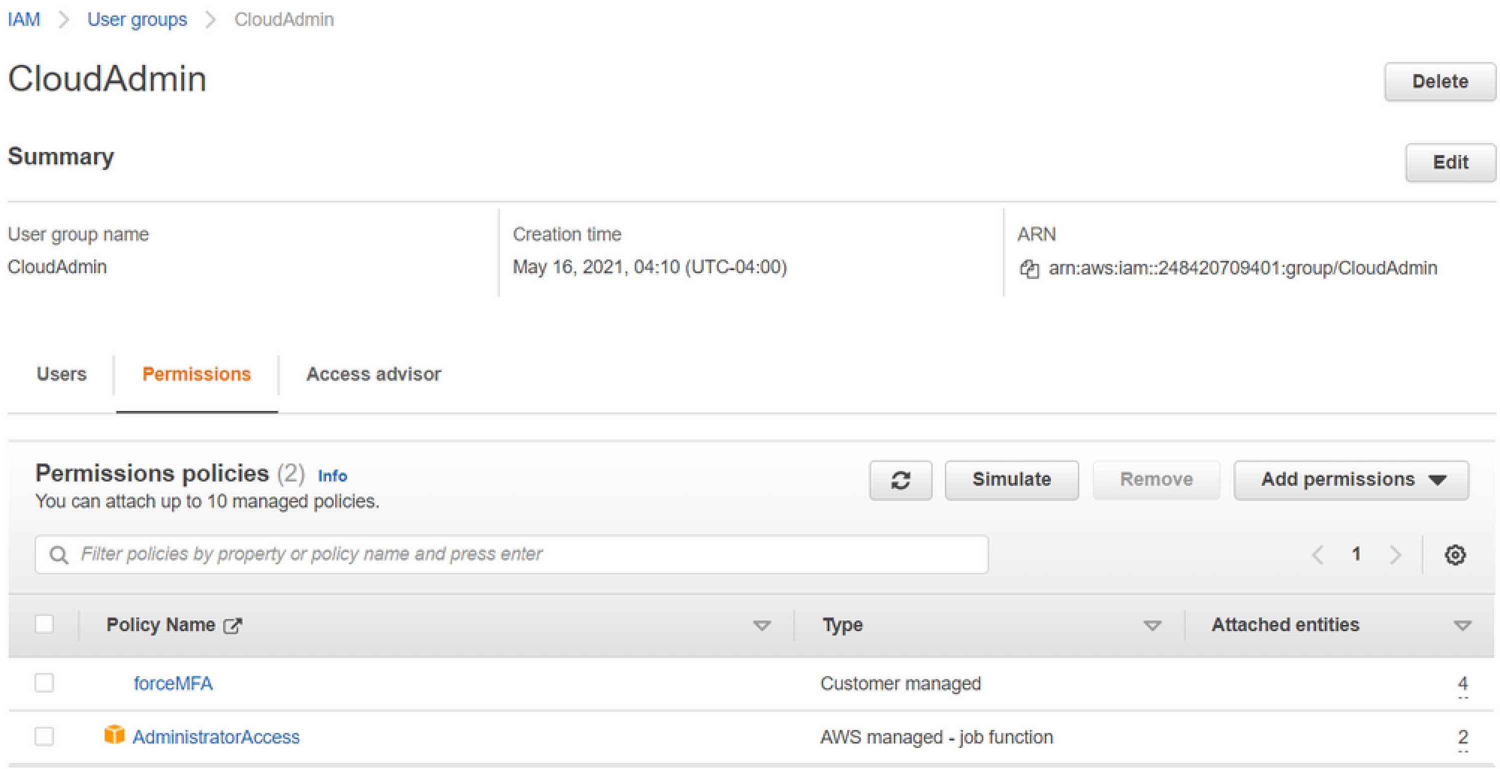
Results



Showing 6 results

☐	User name	Groups	Access key age	Password age	Last activity	MFA
☐	adolfo.carlos	DBA	None	None	None	Not enabled
☐	aguinaldo.balsemao	RedesAdmin	None	None	None	Not enabled
☐	alda.lage	CloudAdmin	None	None	None	Not enabled
☐	amelia.penteado	LinuxAdmin	None	Today	None	Not enabled
☐	podbackup	None	✓ 10 days	None	10 days	Not enabled
☐	xjuggl3r	Admin	✓ 2 days	2 days	Today	Not enabled

Users in groups


IAM > User groups > CloudAdmin
Delete
Edit
Users
Permissions
Access advisor
Simulate
Remove
Add permissions
Filter policies by property or policy name and press enter
1

CloudAdmin

Summary

User group name	Creation time	ARN
CloudAdmin	May 16, 2021, 04:10 (UTC-04:00)	arn:aws:iam::248420709401:group/CloudAdmin

Permissions policies (2) Info

You can attach up to 10 managed policies.

☐	Policy Name	Type	Attached entities
☐	forceMFA	Customer managed	4
☐	AdministratorAccess	AWS managed - job function	2

Groups with policies

The Repositories

GitLab

xJuggl3r / Ansible_AWS_IAM_automation

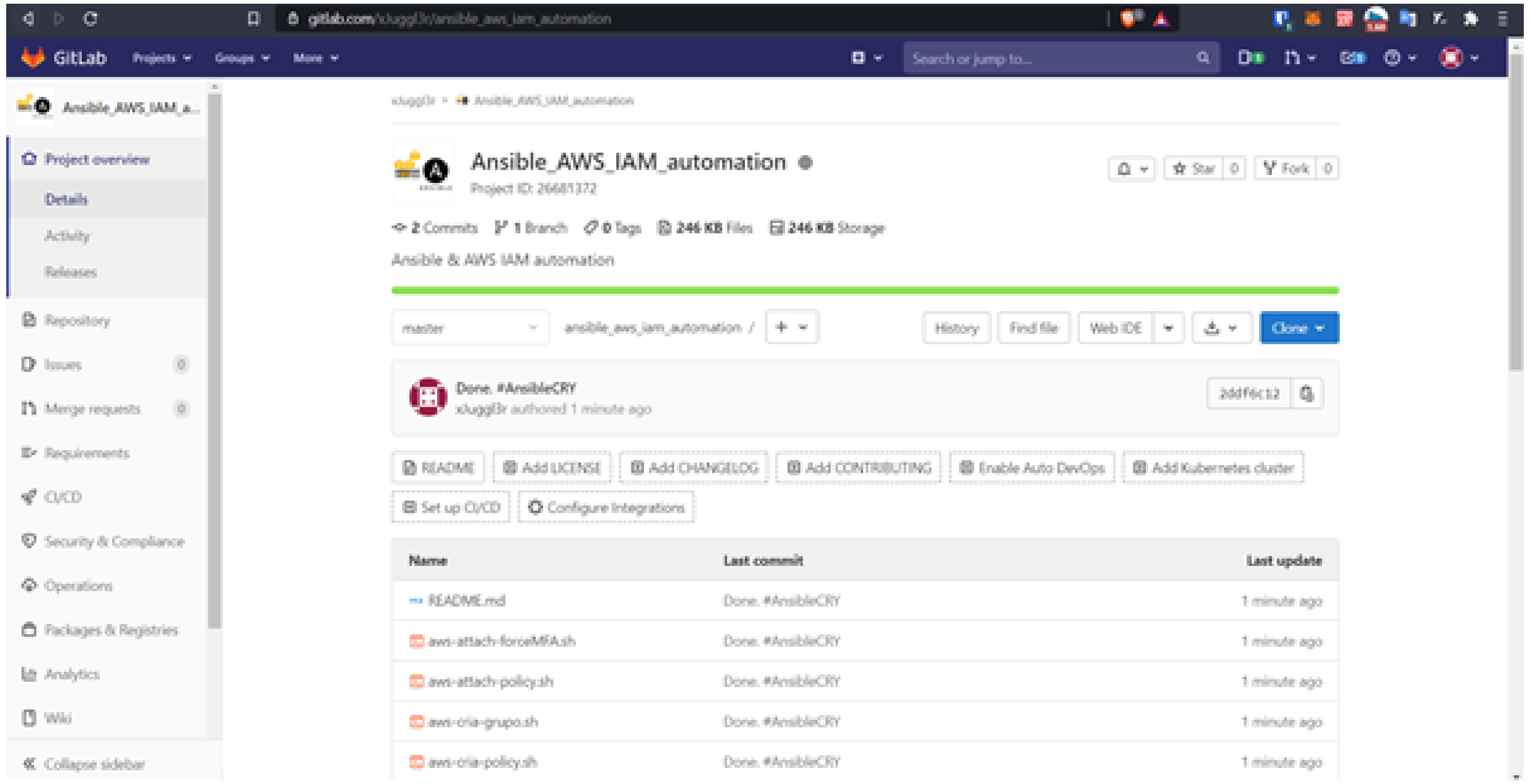
Ansible & AWS IAM automation

GitLab

xJuggl3r / Terraform AWS IAM automation

A small IaaC to migrate on-prem users to AWS using Terraform

GitLab



gitlab.com/xJuggl3r/ansible_aws_iam_automation

xJuggl3r > Ansible_AWS_IAM_automation

Ansible_AWS_IAM_automation

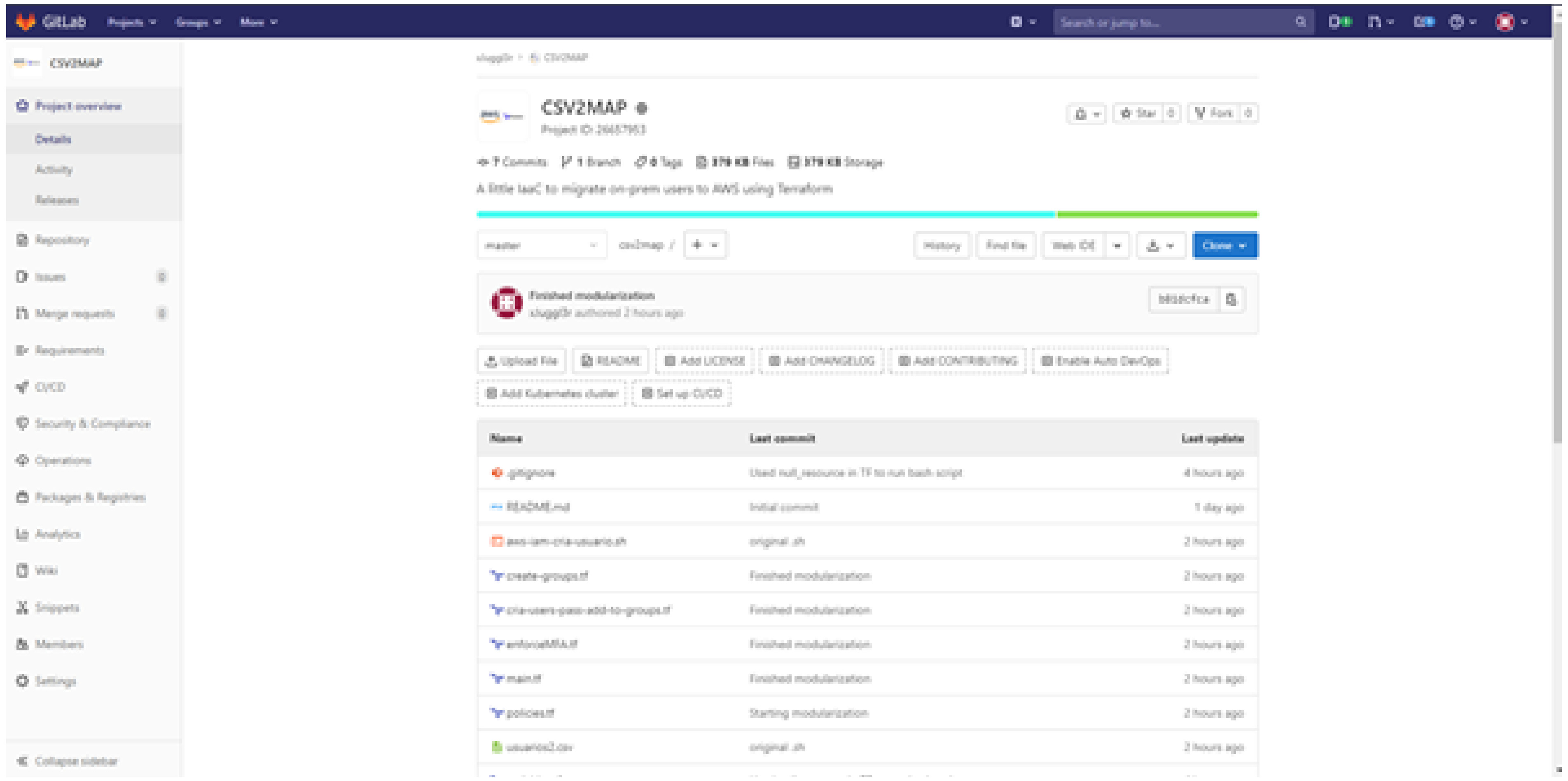
Project ID: 26681372

2 Commits · 1 Branch · 0 Tags · 246 KB Files · 246 KB Storage

Ansible & AWS IAM automation

Done. #AnsibleCRY
xJuggl3r authored 1 minute ago
2ddf6c12

Name	Last commit	Last update
README.md	Done. #AnsibleCRY	1 minute ago
aws-attach-forceMFA.sh	Done. #AnsibleCRY	1 minute ago
aws-attach-policy.sh	Done. #AnsibleCRY	1 minute ago
aws-cria-grupo.sh	Done. #AnsibleCRY	1 minute ago
aws-cria-policy.sh	Done. #AnsibleCRY	1 minute ago



xJuggl3r > CSV2MAP

CSV2MAP

Project ID: 26657953

7 Commits · 1 Branch · 0 Tags · 379 KB Files · 379 KB Storage

A little IaaC to migrate on-prem users to AWS using Terraform

Finished modularization
xJuggl3r authored 2 hours ago
b81dcfca

Name	Last commit	Last update
.gitignore	Used null_resource in TF to run bash script	4 hours ago
README.md	Initial commit	1 day ago
aws-iam-cria-usuario.sh	original .sh	2 hours ago
create-groups.tf	Finished modularization	2 hours ago
cria-users-pass-add-to-groups.tf	Finished modularization	2 hours ago
enforceMFA.tf	Finished modularization	2 hours ago
main.tf	Finished modularization	2 hours ago
policies.tf	Starting modularization	2 hours ago
usuarios2.csv	original .sh	2 hours ago

https://gitlab.com/xJuggl3r/ansible_aws_iam_automation

https://gitlab.com/xJuggl3r/csv2map