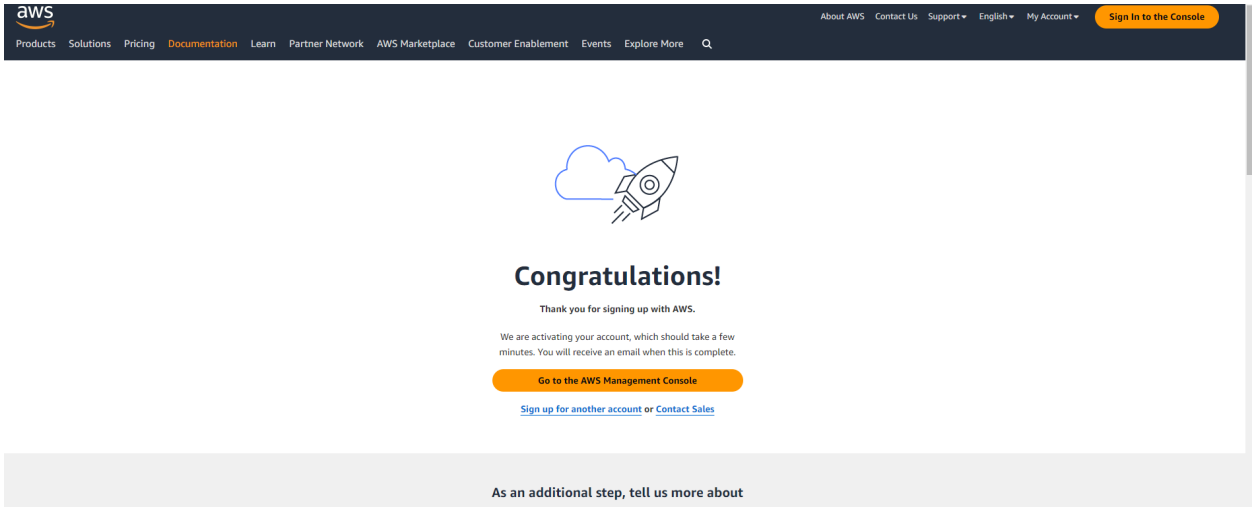
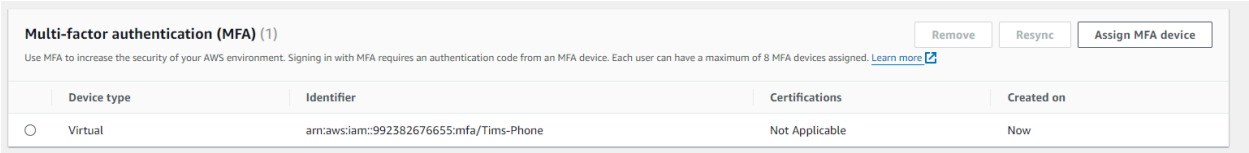
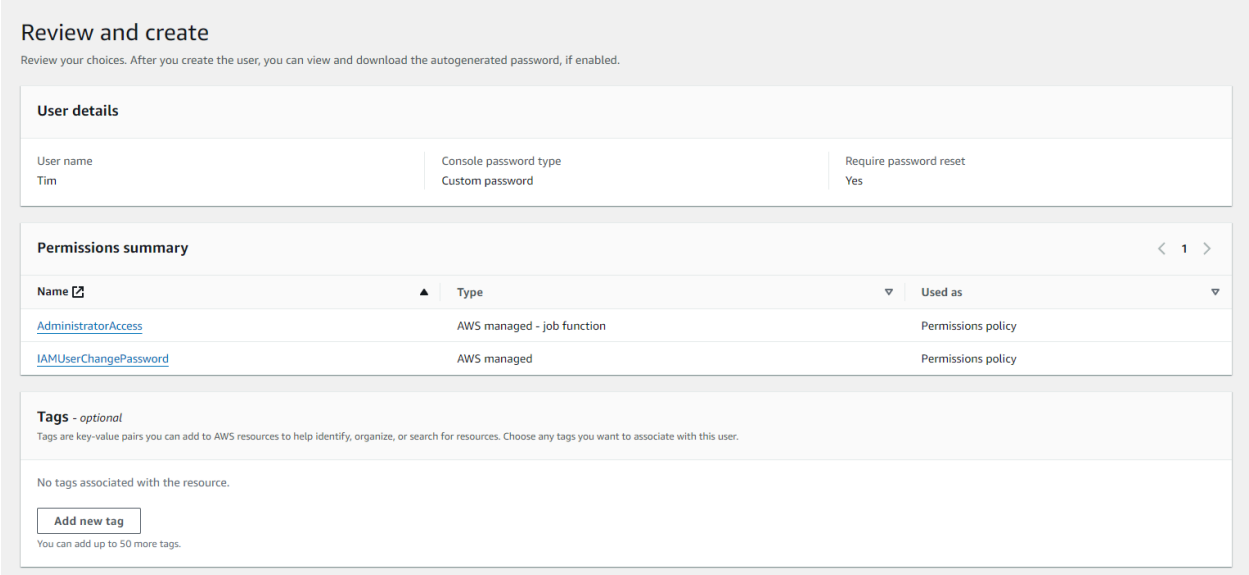
Task 1:



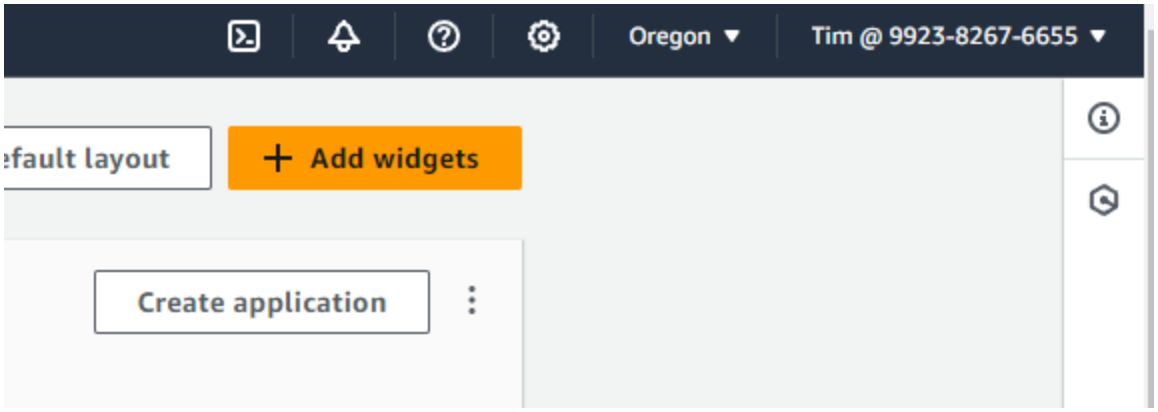
In this step, I have created my AWS account.



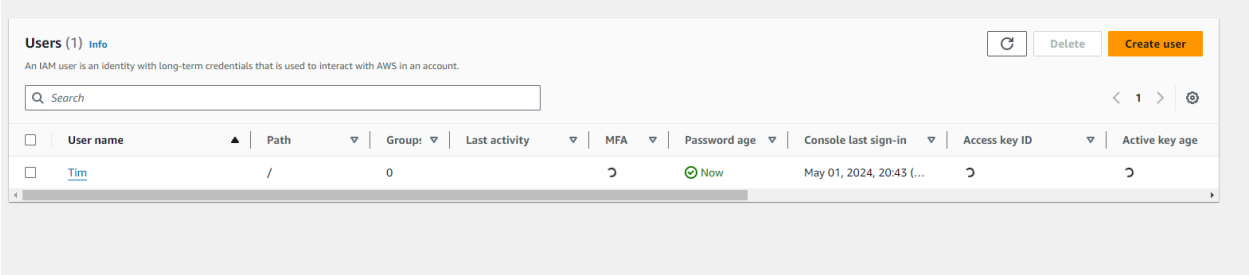
In this step, I have set up my phone for MFA.



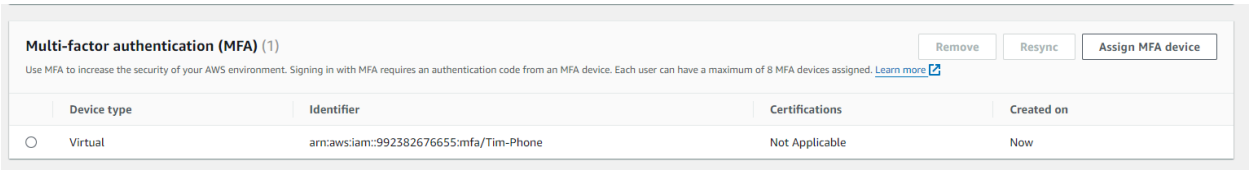
In this step, I have created a new user. https://992382676655.signin.aws.amazon.com/console



I am now signed in as my IAM admin.

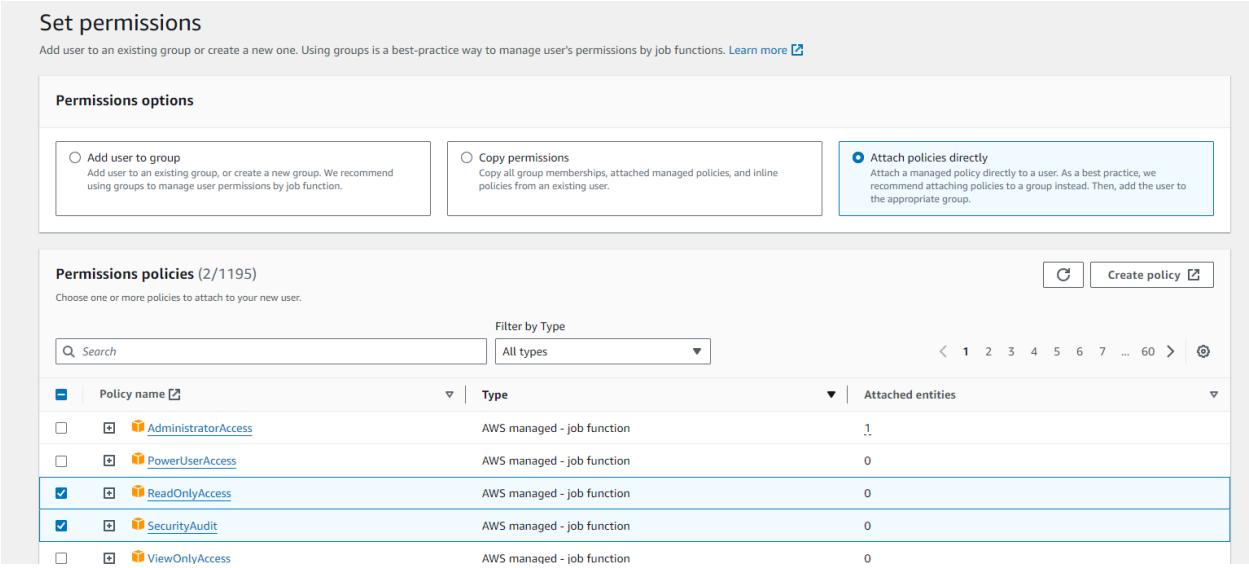


Here I can see the user that I created.

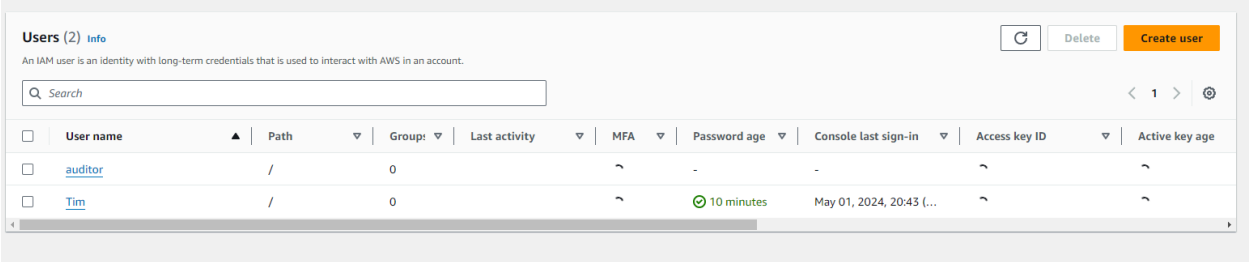


I have set up my phone as an MFA device for this user as well.

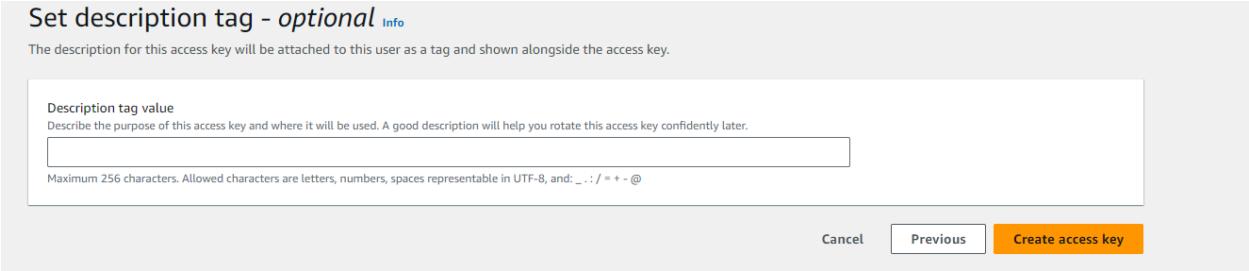
Task 2:



In this step, I have created a new user named “auditor” and am assigning it less permissions. Instead of giving it admin permissions, I only give them the 2 that are checked in the picture.



Auditor has been created.



In this step, I am creating an access key for the auditor user.

```
timothyd@ubuntu:~$ sudo apt install awscli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docutils-common groff gsfonts imagemagick imagemagick-6-common
  imagemagick-6.q16 libaom3 libdav1d5 libde265-0 libfftw3-double3 libheif1
  libilmbase25 libjxr-tools libjxr0 liblqr-1-0 libmagickcore-6.q16-6
```

I am now in my Ubuntu VM and am installing the AWS cli.

```
timothyd@ubuntu:~$ aws --version
aws-cli/1.22.34 Python/3.10.12 Linux/6.5.0-28-generic botocore/1.23.34
```

Here we can see it has been successfully installed.

```
timothyd@ubuntu:~$ aws configure --profile auditor
AWS Access Key ID [None]: AKIA6ODU552X2A6XZSPL
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-1
Default output format [None]: json
```

I have configured the tool to use the IAM auditor credentials.

```
timothyd@ubuntu:~$ sudo apt install python3-virtualenv -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-distlib python3-filelock python3-pip-whl python3-platformdirs python3-s
```

Here I have installed a python virtual environment.

```
timothyd@ubuntu:~$ virtualenv -p python3 venv
created virtual environment CPython3.10.12.final.0-64 in 254ms
  creator CPython3Posix(dest=/home/timothyd/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/timothyd/.local/share/virtualenv)
    added seed packages: pip==22.0.2, setuptools==59.6.0, wheel==0.37.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
timothyd@ubuntu:~$ source venv/bin/activate
(venv) timothyd@ubuntu:~$ pip install scoutsuite
```

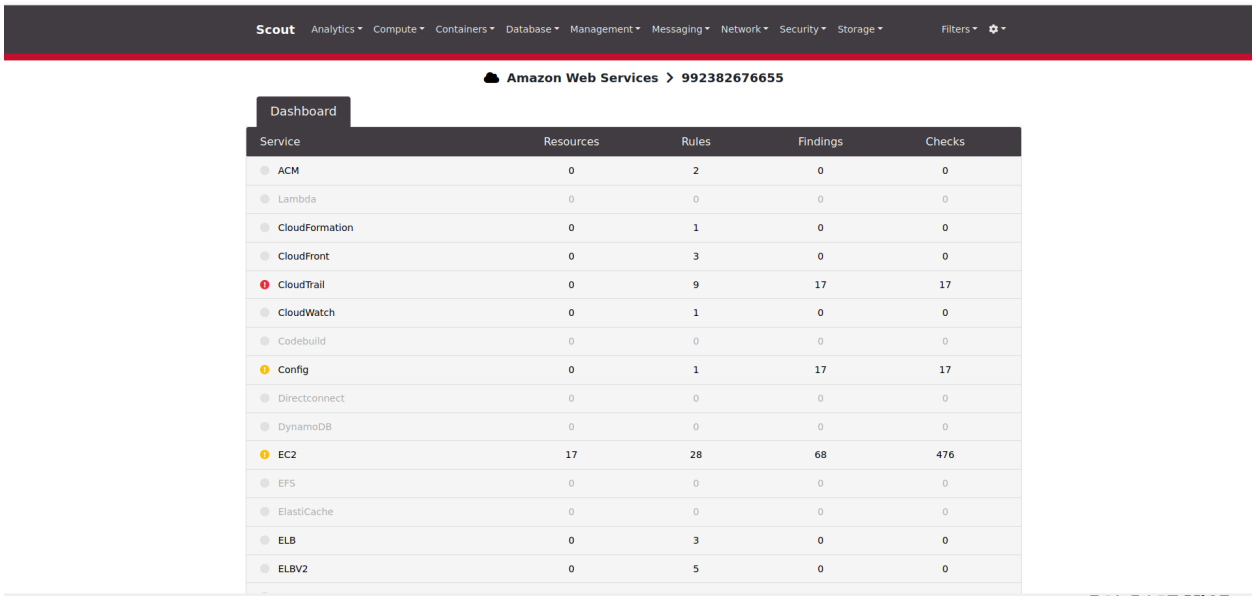
In this step, I have created the virtual environment and then downloaded scoutsuite.

```
(venv) timothyd@ubuntu:~$ scout --help
usage: scout [-h] [-v] {aws,gcp,azure,aliyun,oci,kubernetes} ...

options:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit

The provider you want to run scout against:
  {aws,gcp,azure,aliyun,oci,kubernetes}
    aws                 Run Scout against an Amazon Web Services account
    gcp                 Run Scout against a Google Cloud Platform account
    azure               Run Scout against a Microsoft Azure account
    aliyun              Run Scout against an Alibaba Cloud account
    oci                 Run Scout against an Oracle Cloud Infrastructure account
    kubernetes          Run Scout against a Kubernetes cluster

To get addtional help on a specific provider run: scout.py {provider} -h
(venv) timothyd@ubuntu:~$
```

Scout has been successfully installed.

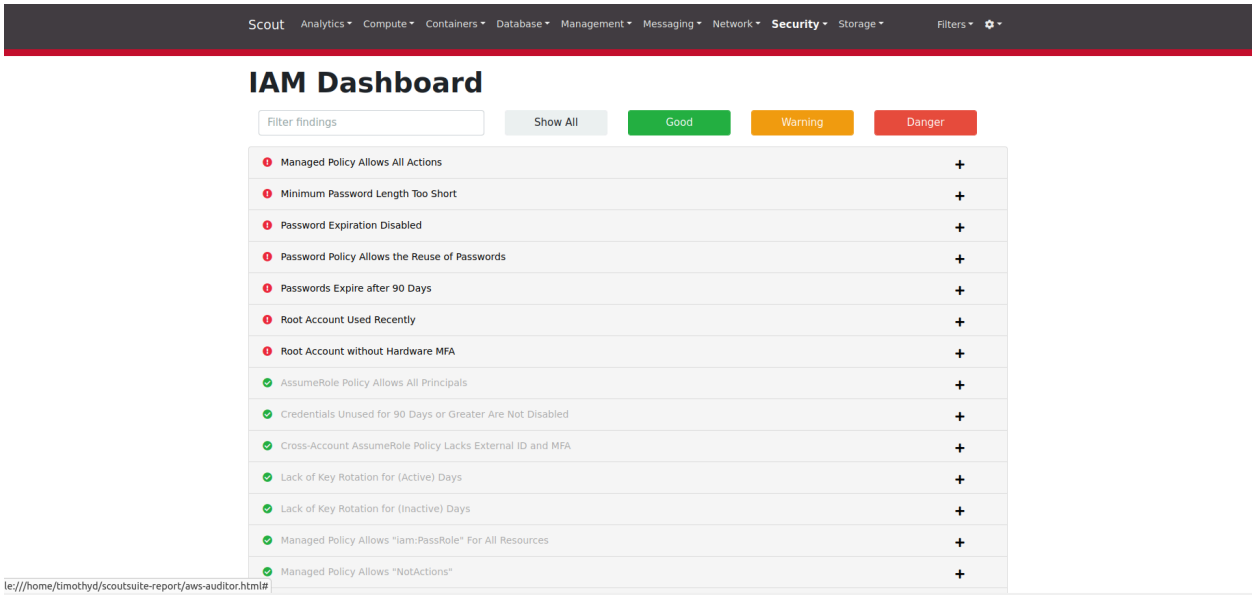
```
(venv) timothyd@ubuntu:~$ scout aws --profile auditor
2024-05-01 21:35:32 ubuntu scout[5605] INFO Launching Scout
2024-05-01 21:35:32 ubuntu scout[5605] INFO Authenticating to cloud provider
2024-05-01 21:35:34 ubuntu scout[5605] INFO Gathering data from APIs
2024-05-01 21:35:34 ubuntu scout[5605] INFO Fetching resources for the ACM service
2024-05-01 21:35:35 ubuntu scout[5605] INFO Fetching resources for the Lambda service
2024-05-01 21:35:35 ubuntu scout[5605] INFO Fetching resources for the CloudFormation service
2024-05-01 21:35:36 ubuntu scout[5605] INFO Fetching resources for the CloudTrail service
2024-05-01 21:35:36 ubuntu scout[5605] INFO Fetching resources for the CloudWatch service
2024-05-01 21:35:37 ubuntu scout[5605] INFO Fetching resources for the CloudFront service
2024-05-01 21:35:37 ubuntu scout[5605] INFO Fetching resources for the CodeBuild service
2024-05-01 21:35:38 ubuntu scout[5605] INFO Fetching resources for the Config service
2024-05-01 21:35:38 ubuntu scout[5605] INFO Fetching resources for the Direct Connect service
2024-05-01 21:35:39 ubuntu scout[5605] INFO Fetching resources for the DynamoDB service
2024-05-01 21:35:39 ubuntu scout[5605] INFO Fetching resources for the EC2 service
2024-05-01 21:35:40 ubuntu scout[5605] INFO Fetching resources for the EFS service
2024-05-01 21:35:40 ubuntu scout[5605] INFO Fetching resources for the ElastiCache service
2024-05-01 21:35:40 ubuntu scout[5605] INFO Fetching resources for the ELB service
2024-05-01 21:35:41 ubuntu scout[5605] INFO Fetching resources for the ELBv2 service
2024-05-01 21:35:41 ubuntu scout[5605] INFO Fetching resources for the EMR service
2024-05-01 21:35:42 ubuntu scout[5605] INFO Fetching resources for the IAM service
```

In this step, I have started the scan.



Service	Resources	Rules	Findings	Checks
ACM	0	2	0	0
Lambda	0	0	0	0
CloudFormation	0	1	0	0
CloudFront	0	3	0	0
CloudTrail	0	9	17	17
CloudWatch	0	1	0	0
Codebuild	0	0	0	0
Config	0	1	17	17
Directconnect	0	0	0	0
DynamoDB	0	0	0	0
EC2	17	28	68	476
EFS	0	0	0	0
ElastiCache	0	0	0	0
ELB	0	3	0	0
ELBV2	0	5	0	0

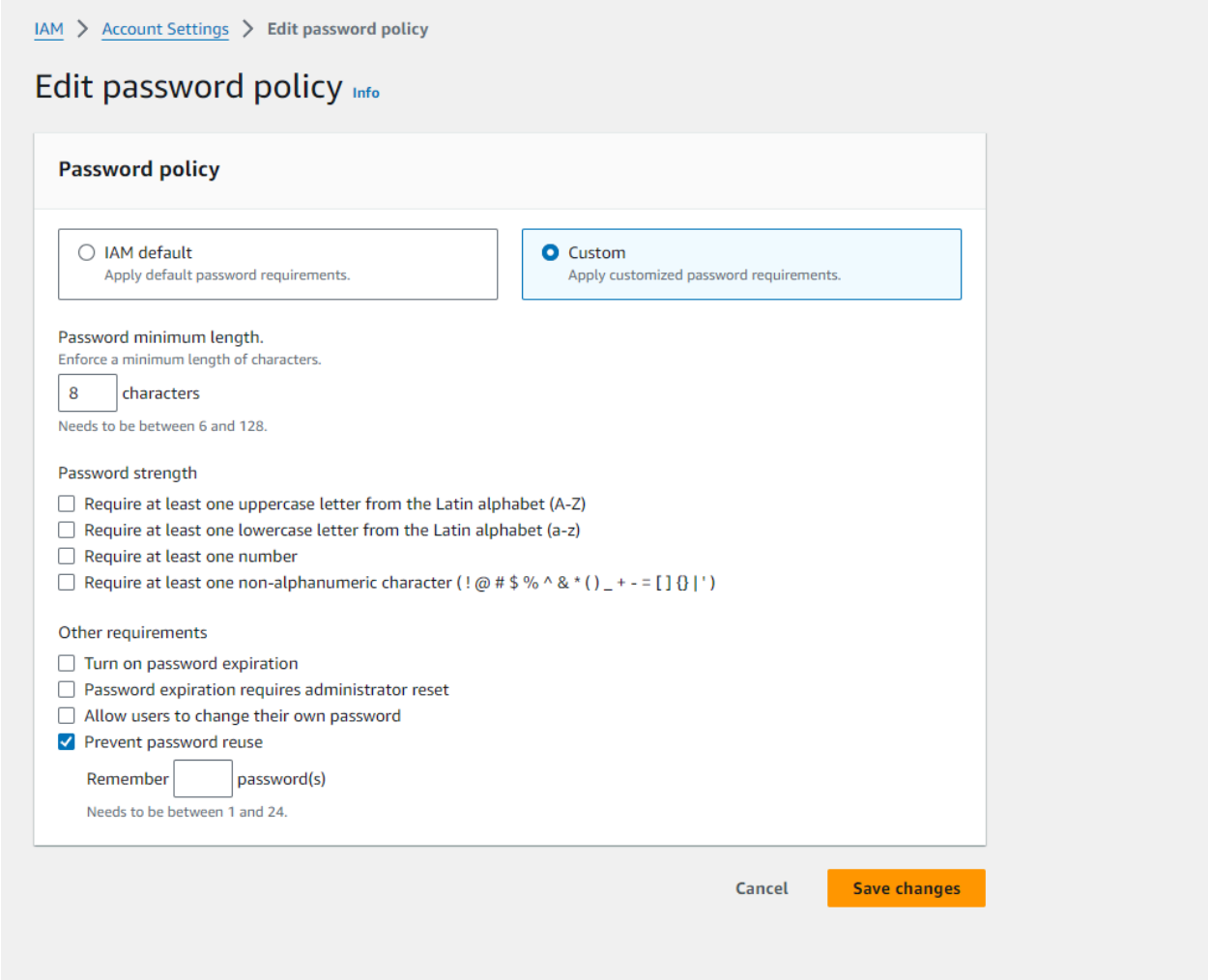
The scan has been completed and the report has been made.

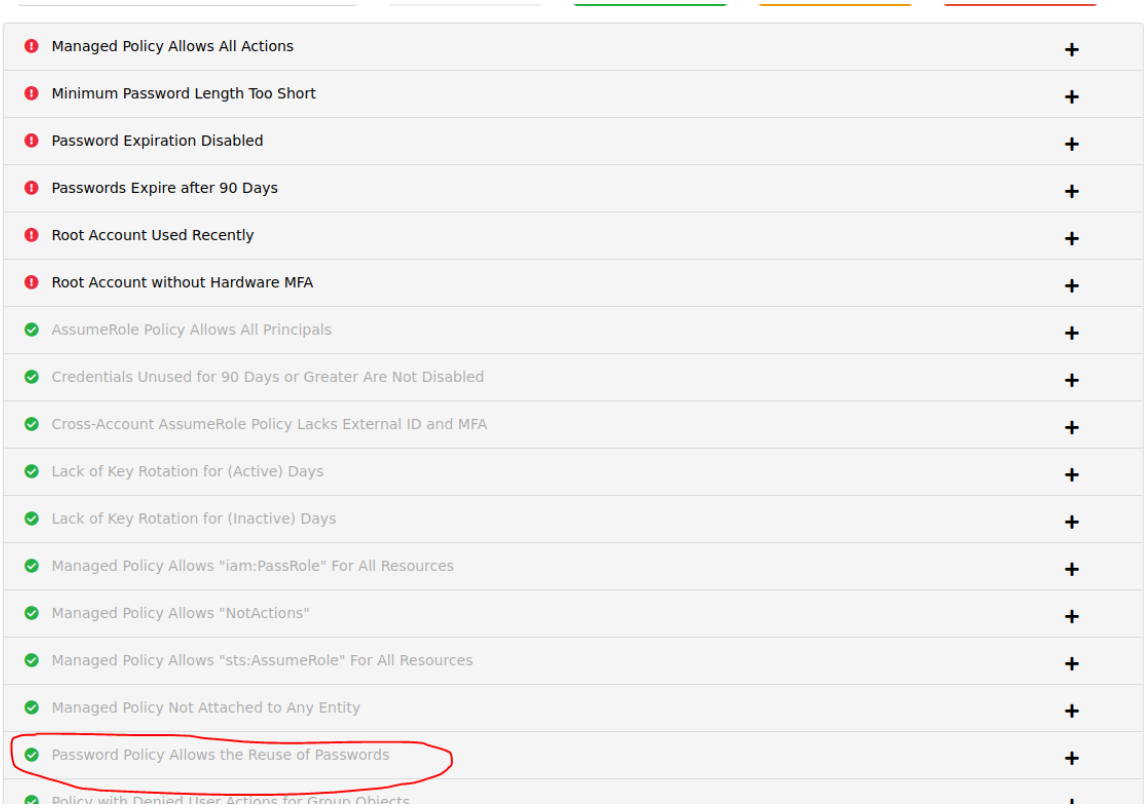


The vulnerability that I have picked is the “Password Policy Allows the Reuse of Passwords” in the IAM service. This vulnerability pretty much just means that when setting or resetting a password, it should not be a password that has already been used on the account. This is a vulnerability because if your account has been compromised and you need to reset your password, you should not reset it to a password that has been already exposed to a hacker. This will just allow for your account to be easily compromised again. Instead, you should pick a new unique password that will make it more difficult for an attacker to get into your account.

The steps that are needed to fix this is to first go into the “Account Settings” and then go into “Edit Password Policy”. From here , you check the “Custom” box and it will allow you to change the password policies. Check the box that says “Prevent password reuse” and then input the number of

previous passwords that you want to be remembered. Then hit “Save Changes”.





After rerunning the scan, we can now see that “Password Policy Allows the Reuse of Passwords” is no longer a vulnerability. This fix does not cost any money to implement.