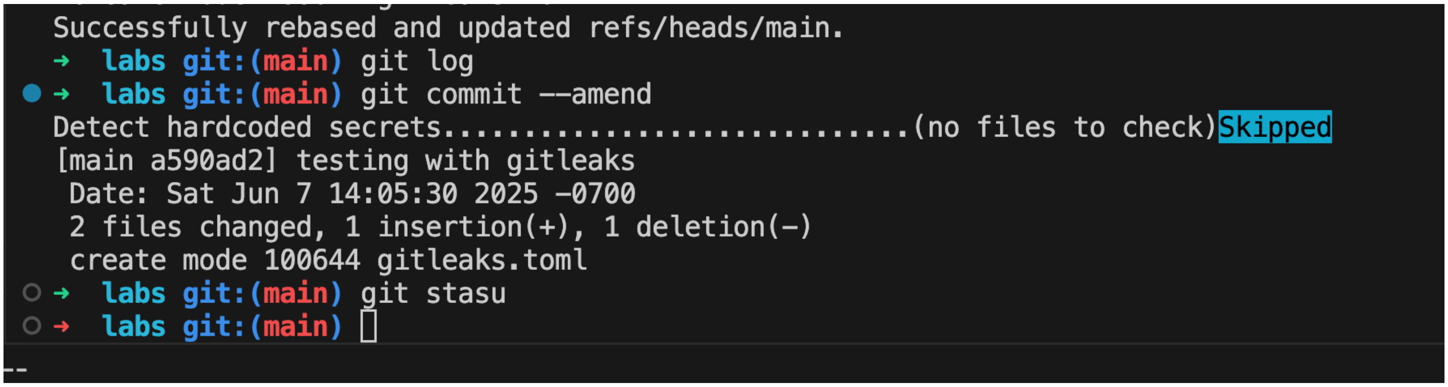
Lab 3

Prep

- ☒ Gitea set up
- ☒ MFA set up
- ☒ Add git ignore
- ☒ Secrets/Token Management
    - ☒ Consider secret-scanning
        - ☒ Added git-leaks on pre-commit hook
- ☒ Create & Connect to a Git repository
    - ☒ https://code.wizards.cafe
- ☒ Modify and make a second commit



```
 Successfully rebased and updated refs/heads/main.
 → labs git:(main) git log
● → labs git:(main) git commit --amend
 Detect hardcoded secrets...............................(no files to check)Skipped
 [main a590ad2] testing with gitleaks
  Date: Sat Jun 7 14:05:30 2025 -0700
  2 files changed, 1 insertion(+), 1 deletion(-)
  create mode 100644 gitleaks.toml
○ → labs git:(main) git stasu
○ → labs git:(main)
--
```

Figure 1: image of terminal

- ☒ Test to see if gitea actions works
- ☒ Have an existing s3 bucket

Resources

- ☒ Capital One Data Breach
- ☒ Grant IAM User Access to Only One S3 Bucket
- ☐ IAM Bucket Policies
- ☐ Dumping S3 Buckets!

Lab

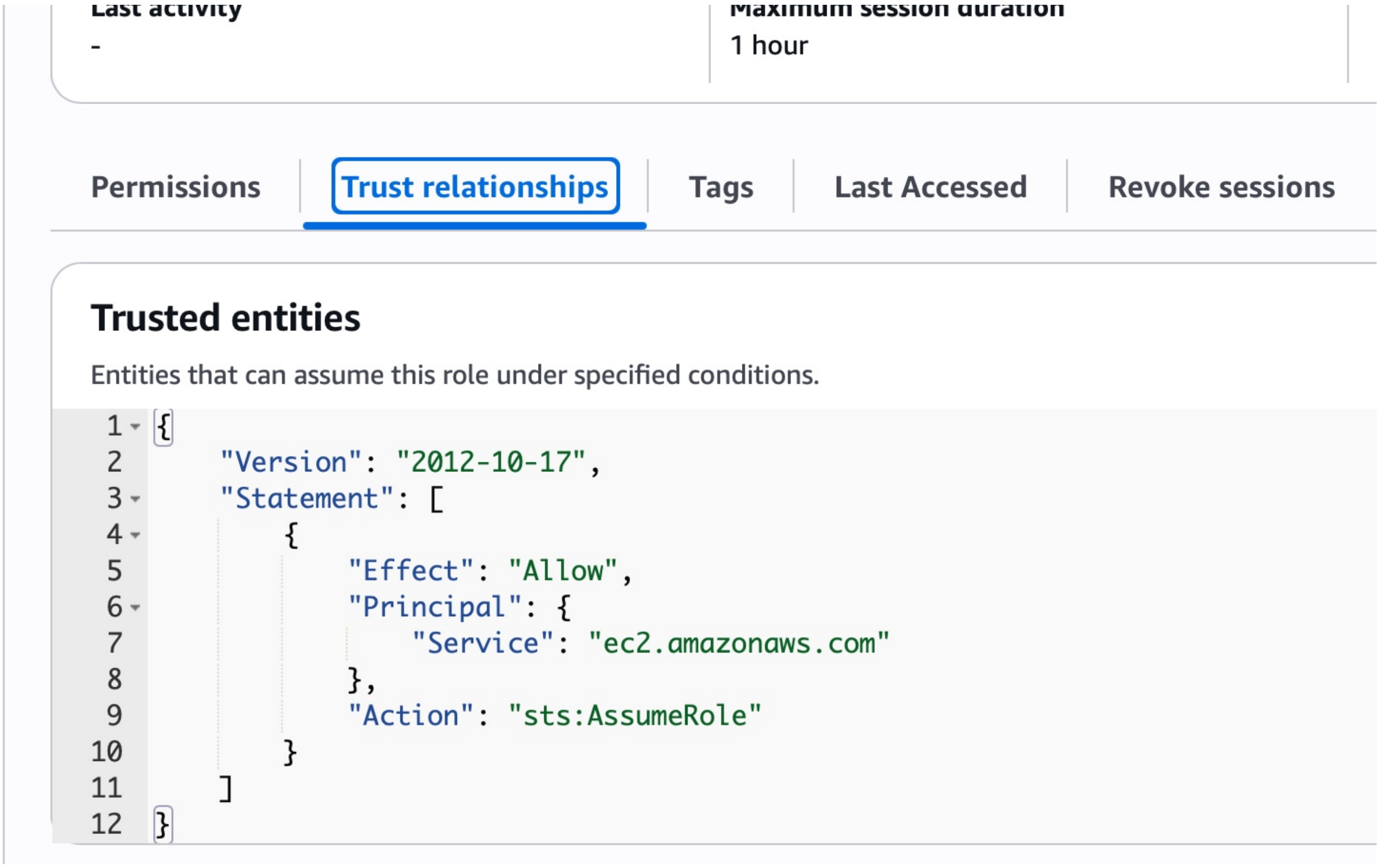
- ☒ create a custom IAM Policy
- ☒ create an IAM Role for EC2

- ☒ Attach the Role to your EC2 Instance
- ☒ Verify is3 access from the EC2 Instance
    - HTTPS outbound was not set up
        - I did not check outbound rules (even when the lab explicitly called this out) because it mentioned lab 2, so my assumption was that it had already been set up (it was not). When connection to s3 failed I double checked lab 3 instructions

Stretch

- ☒ Create a bucket policy that blocks all public access but allows your IAM role
    - ☐ Implmented: guide

- ☒ **Experiment** with requiring MFA or VPC conditions.
    - ☒ MFA conditions
        - MFA did not work out of the box after setting it in the s3 bucket policy. The ways I found you can configure MFA:
        - stackoverflow
        - official guide



Last activity
-

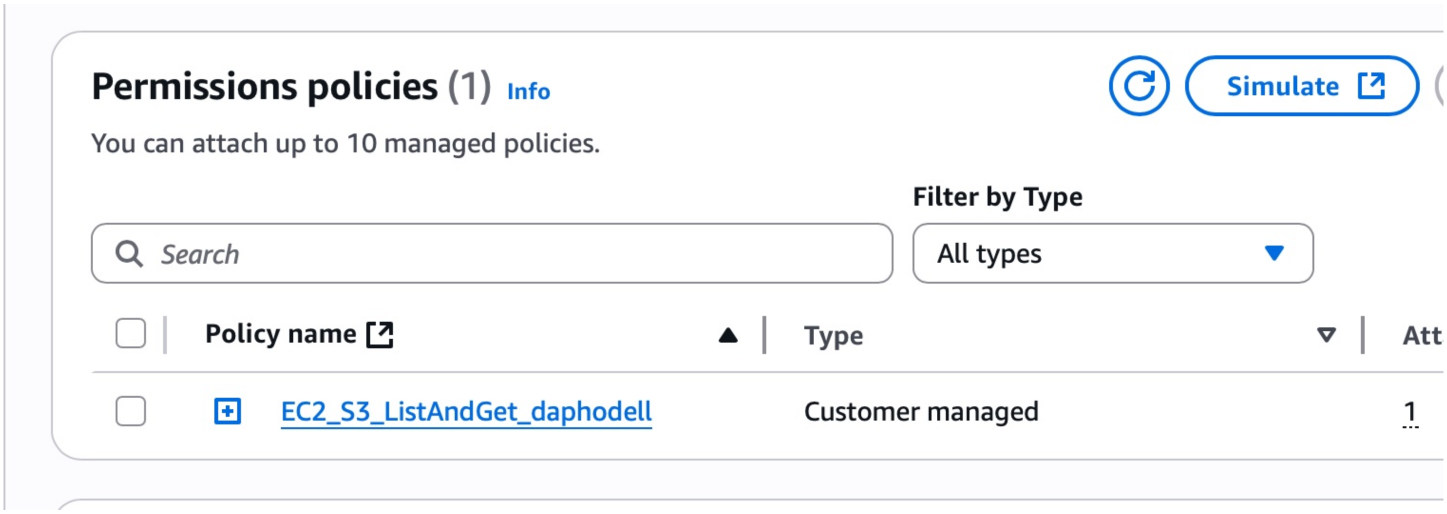
Maximum session duration
1 hour

Trusted entities

Entities that can assume this role under specified conditions.

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

Figure 2: trust relationships



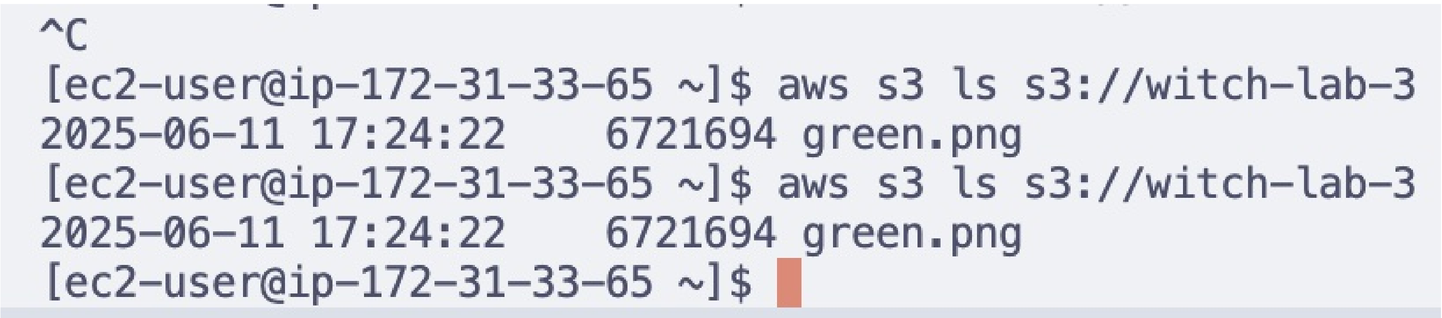
Permissions policies (1) Info

You can attach up to 10 managed policies.

Filter by Type: All types

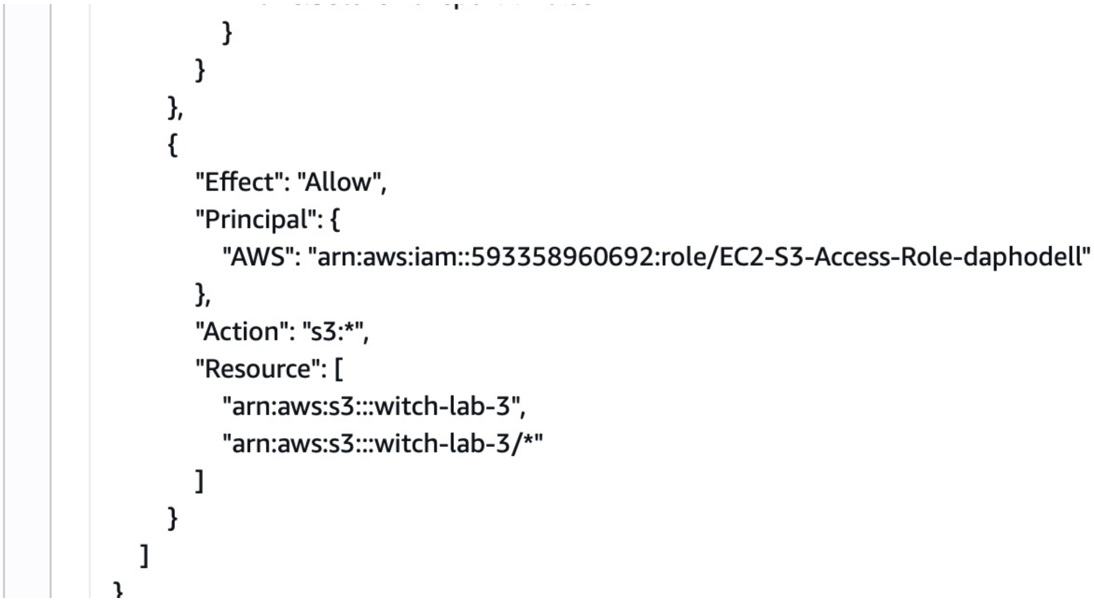
□	Policy name	Type	Att
□	EC2_S3_ListAndGet_daphodell	Customer managed	1

Figure 3: permissions



```
^C
[ec2-user@ip-172-31-33-65 ~]$ aws s3 ls s3://witch-lab-3
2025-06-11 17:24:22    6721694 green.png
[ec2-user@ip-172-31-33-65 ~]$ aws s3 ls s3://witch-lab-3
2025-06-11 17:24:22    6721694 green.png
[ec2-user@ip-172-31-33-65 ~]$ 
```

Figure 4: screenshot of listing s3 contents



```
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::593358960692:role/EC2-S3-Access-Role-daphodell"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::witch-lab-3",
                "arn:aws:s3:::witch-lab-3/*"
            ]
        }
    ]
}
```

Figure 5: restrict to role

        - ☒ via cli roles - I set up a new set of role-trust relationships. Update s3 Role: Update action: sts:assumerole Update principle (for user – could not target group) Add condition (MFA bool must be true)
        - Commands referenced: I set up a script that looks like this

```
MFA_TOKEN=$1

if [ -z "$1" ]; then
  echo "Error: Run with MFA token!"
  exit 1
fi

if [ -z $BW_AWS_ACCOUNT_SECRET_ID ]; then
  echo "env var BW_AWS_ACCOUNT_SECRET_ID must be set!"
  exit 1
fi

AWS_SECRETS=$(bw get item $BW_AWS_ACCOUNT_SECRET_ID)

export AWS_ACCESS_KEY_ID=$(echo "$AWS_SECRETS" | jq -r '.fields[0].value')
export AWS_SECRET_ACCESS_KEY=$(echo "$AWS_SECRETS" | jq '.fields[1].value' | tr -d '"')

SESSION_OUTPUT=$(aws sts assume-role --role-arn $S3_ROLE --role-session-name $SESSION_TYPE --serial-number $MF
#echo $SESSION_OUTPUT
export AWS_SESSION_TOKEN=$(echo "$SESSION_OUTPUT" | jq '.Credentials.SessionToken' | tr -d '"')
export AWS_ACCESS_KEY_ID=$(echo "$SESSION_OUTPUT" | jq '.Credentials.AccessKeyId' | tr -d '"')
export AWS_SECRET_ACCESS_KEY=$(echo "$SESSION_OUTPUT" | jq '.Credentials.SecretAccessKey' | tr -d '"')
#echo $AWS_SESSION_TOKEN
#echo $AWS_ACCESS_KEY_ID
#echo $AWS_SECRET_ACCESS_KEY
aws s3 ls s3://witch-lab-3
```

```
* configuration via ~/.aws/credentials
* 1Password CLI with AWS Plugin
  * I use bitwarden, which also has an AWS Plugin
  * I've seen a lot more recommendations (TBH it's more like 2 vs 0)
    for 1password for password credential setup. Wonder why?
```

- ☒ **Host a static site**
    - ☒ Enable a static website hosting (`index.html`)
    - ☒ Configure route 53 alias or CNAME for `resume.<yourdomain>` to the bucket endpoint.
    - ☒ Deploy CloudFront with ACM certificate for HTTPS
        - see: resume
- ☐ **Private "Invite-Only" Resume Hosting**
    - ☐ **Pre-signed URLs** `aws s3 presign s3://<YOUR_BUCKET_NAME>/resume.pdf --expires-in 3600`

Further Exploration

- ☐ Snapshots & AMIs

- ☐ Create an EBS snapshot of `/dev/xvda`
- ☐ Register/create an AMI from that snapshot
    - ☐ How do you "version" a server with snapshots? Why is this useful?
- ☐ Launch a new instance from your AMI

- ☐ Linux & Security Tooling
- ☐ Scripting & Automation

- ☐ Bash: report world-writable files
- ☐ Python with boto3: list snapshots, start/stop instances

Further Reading

- [ ]

- [ ]
- [ ]

Reflection

- What I built
- Challenges
  - Groups cannot be used as the principal in a trust relationship
  - The stretch goal for setting up s3 + mfa was a bit of a pain:
    - The earlier lab had me set up a trust relationship on the role to allow EC2 as a principal on the role When I later updated IAM permissions to include MFA, I promptly forgot about this detail and had chatgpt help me with troubleshooting. It was pretty good at helping me figure out the issue
- Security concerns On scale and security at scale

Terms

Identity Access Management

```
graph LR
  IAMPolicy -- attaches to --> IAMIdentity
  ExplainIAMIdentity[users, groups of users, roles, AWS resources]:::aside
  ExplainIAMIdentity -.-> IAMIdentity
classDef aside stroke-dasharray: 5 5, stroke-width:2px;
```

End lab

- □ On June 20, 2025, do the following:
  - □ Clean up
    - □ Custom roles
    - □ Custom policies
  - □ Shut down ec2 Instance