Identity Access Management (IAM) & SSM

Least Privilege Rule

- Users
- Groups
- Roles

Today we're creating an IAM role for an EC2 to access Session Manager – this way you don't need to handle AWS credentials saved to your laptop

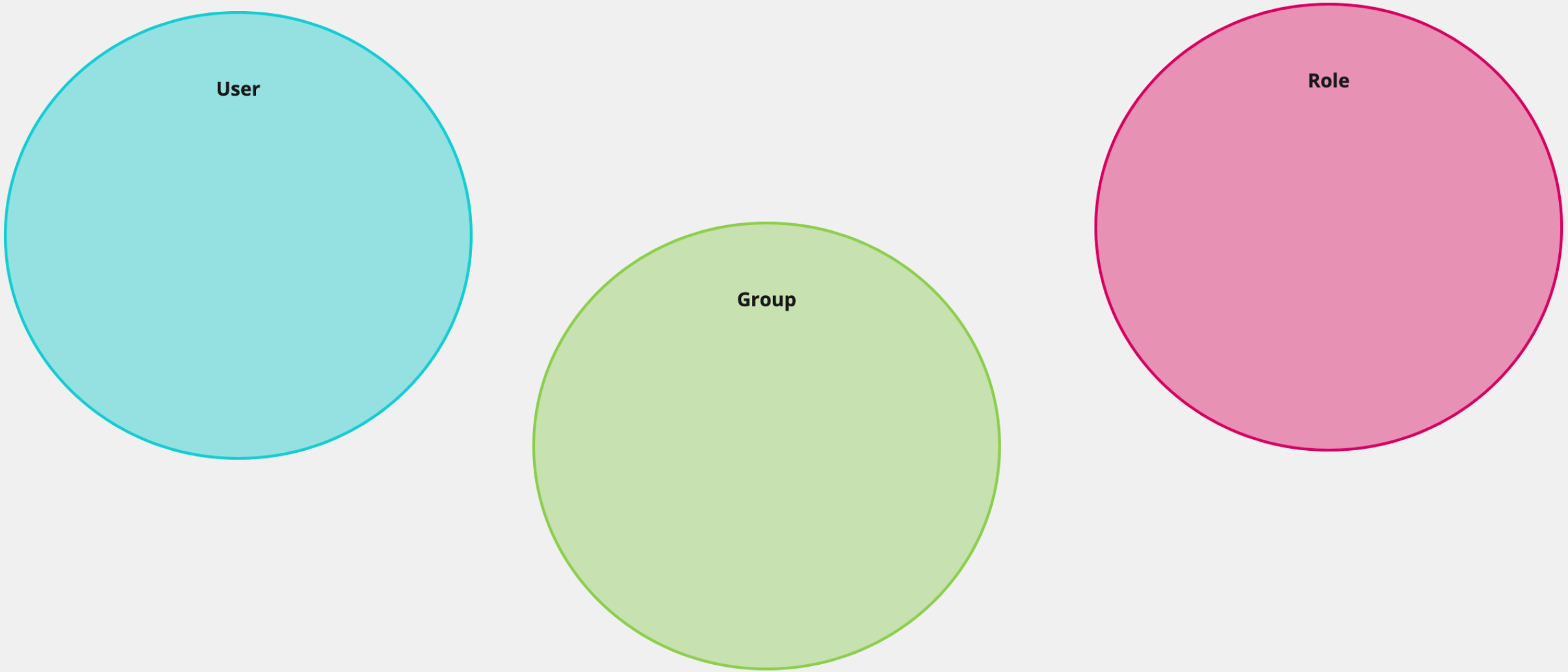
IAM Policies

- Policies define the specifics of who/what can access AWS resource/s and how
- AWS has predefined policies or you can make your own

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
                    "192.0.2.0/24",
                    "203.0.113.0/24"
                ]
            },
            "Bool": {"aws:ViaAWSService": "false"}
        }
    }
}
```

Firstly, let's refresh ourselves on IAM with an activity!!

What is the difference between these 3 different IAM resource types?
When would you use each one? Give an example



SSH (Secure Shell)

Using an AWS key pair you can log onto an EC2 instance from the command line

- Requires you to have the AWS CLI installed
- You'll be handling a private key saved to your computer
- There are 'best practices' around managing SSH usage including secure storage and rotation of keys – we won't be going into this

SSM's Session Manager

SSM manager is not SSH, it is making requests using HTTPS.

Can be used via console or AWS CLI.

Here's some tips I've found in this handy Hackernoon article for people who want a deep dive:

- Enable WHO can use this feature via ***ssm:enabled*** in IAM policies
- Ensure people can only terminate their own sessions using:

```
"Effect": "Allow",
"Action": [
    "ssm:TerminateSession"
],
"Resource": [
    "arn:aws:ssm:*:*:session/${aws:username}-*"
]
```

Hackernoon Article: https://hackernoon.com/ditch-your-ssh-keys-and-enable-aws-ssm-ec1c2b27350c

We're going to use SSM Session Manager

- Most people have experience with SSH, so this is something new
- We don't have to troubleshoot machine-specific CLI issues – we'll be using the console
- As we move towards '*servers as cattle, not pets*' means we have less reasons to log onto individual servers
- I haven't worked directly with an SSH key management process, don't want to give wrong advice