# Accessing S3 with AWS IAM Roles from EC2 instance

IAM Policy

An IAM (Identity and access management) policy is an entity in AWS, that enables you to manage access to AWS services and resources in a secure fashion.

Policy Types

There are two important types of policies:

**Identity-Based-Policies: User**

**Resource-Based-Policies: AWS resource**

### Identity-Based-Policy

1. Identity-based policies are policies that you can attach to an AWS identity (such as a user, group of users, or role).
2. Identity-based policies are further classified as:

* AWS Managed Policies
* Custom Managed Policies

#### AWS Managed Policies

1. AWS Managed policies are those policies that are created and managed by AWS itself.
2. If you are new to IAM policies, you can start with AWS managed policies before managing your own.

#### Custom Managed Policies

1. Custom managed policies are policies that are created and managed by you in your AWS account.
2. Customer managed policies provide us with more precise control than AWS managed policies.
3. You can create and edit an IAM policy in the visual editor or by creating the JSON policy document directly.
4. You can create your own IAM policy using the following link: <https://awspolicygen.s3.amazonaws.com/policygen.html>

### Resource-Based-Policy

1. Resource-based policies are policies that we attach to a resource such as an Amazon S3 bucket.
2. Resource-based policies grant the specified permission to perform specific actions on particular resources and define under what conditions these policies apply to them.
3. Resource-based policies are in line with other policies.
4. There are currently no AWS-managed resource-based policies.
5. There is only one type of resource-based policy called a *trust policy*, which is attached to an IAM role.
6. An IAM role is both an identity and a resource that supports resource-based policies.

IAM Role

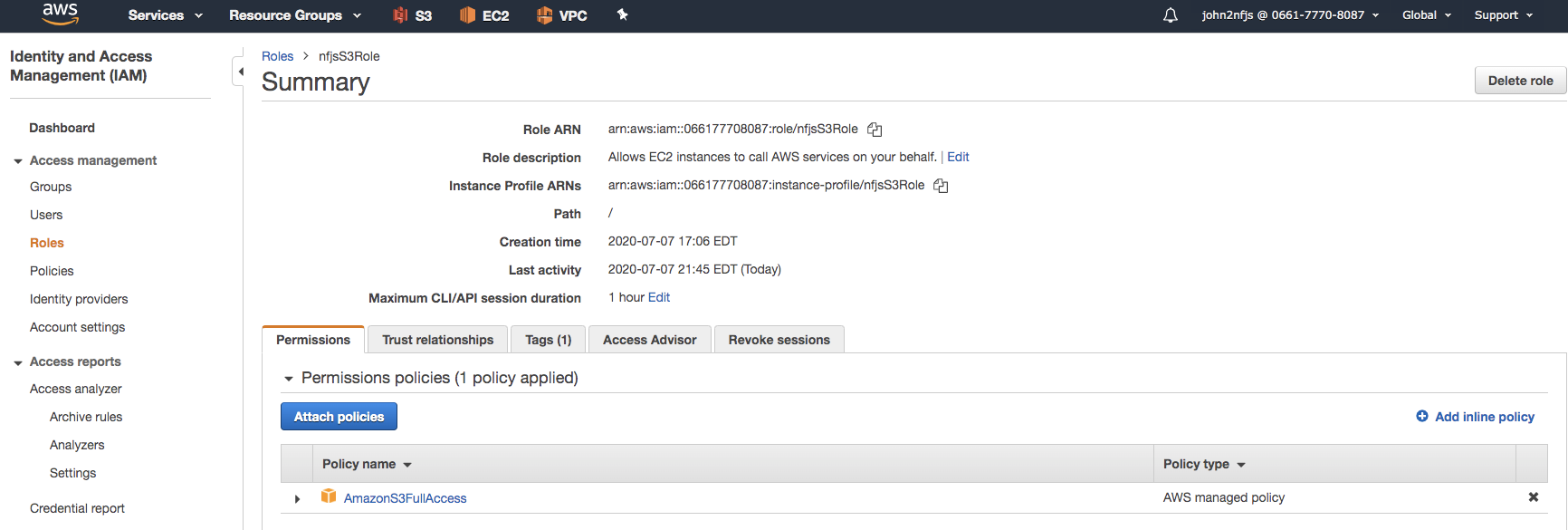
1. An IAM *role* is an AWS IAM identity (that we can create in our AWS account) that has specific permissions.
2. It is similar to an IAM user, which determines what the identity can and cannot do in AWS.
3. Instead of attaching a role to a particular user or group, it can be attached to anyone who needs it.
4. The advantage of having a role is that we do not have standard long-term credentials such as a password or access keys associated with it.

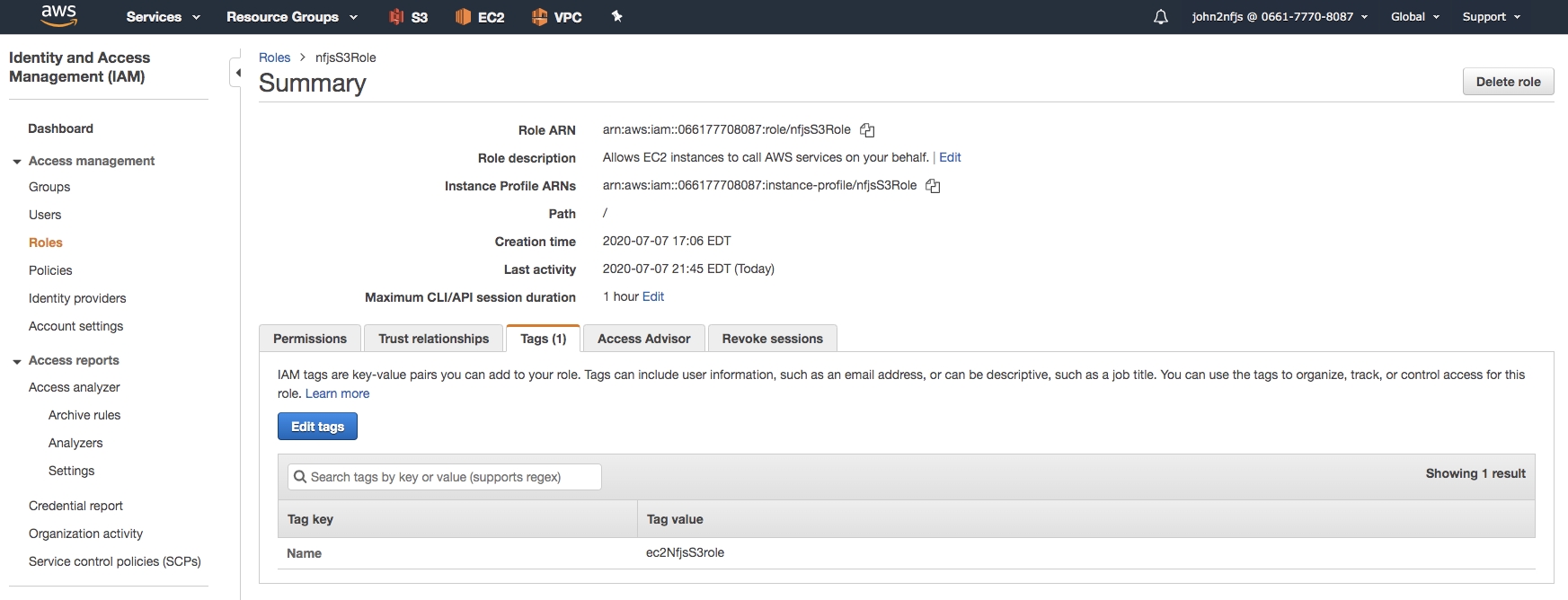
## 

## Steps

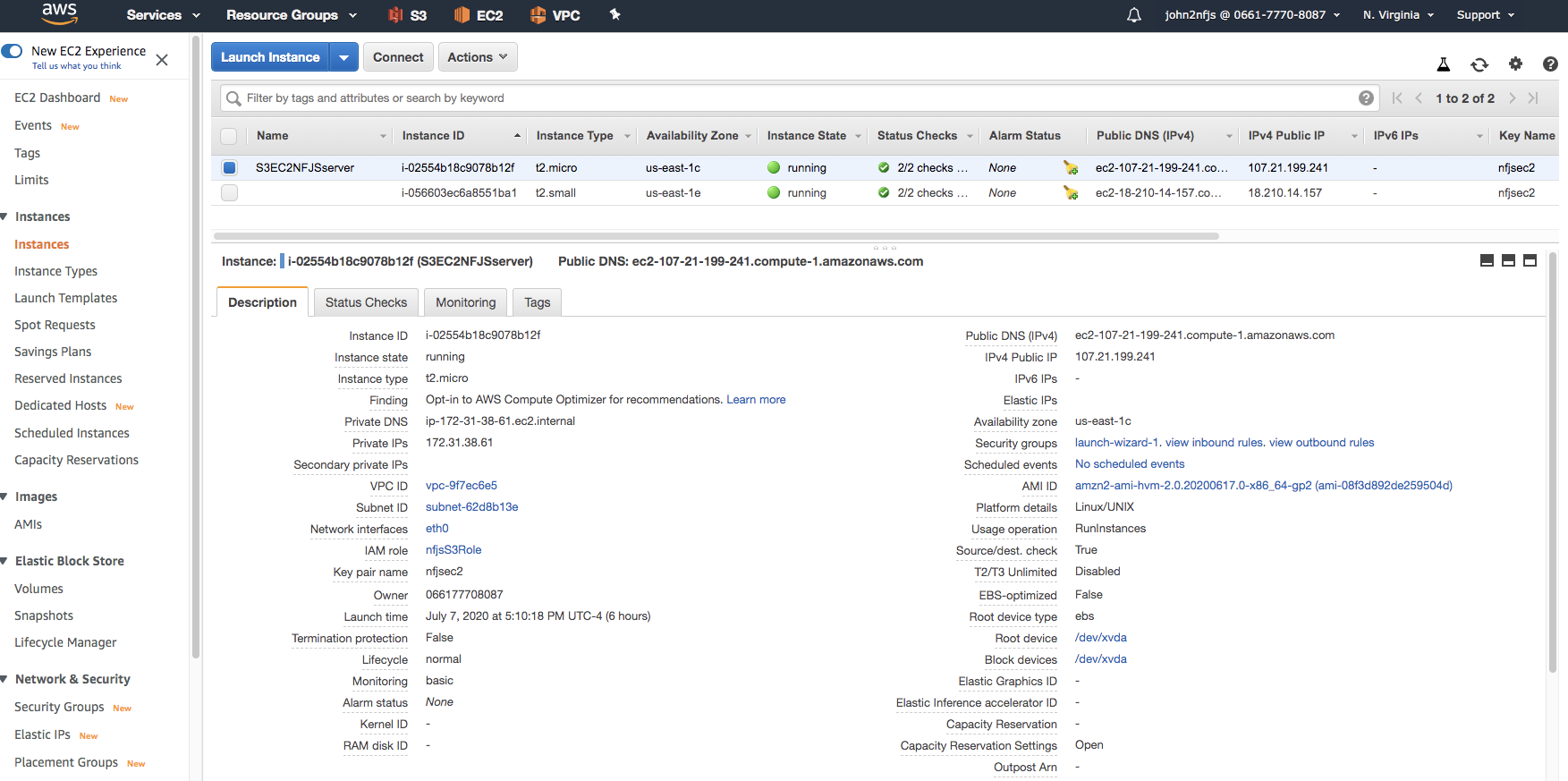
### 

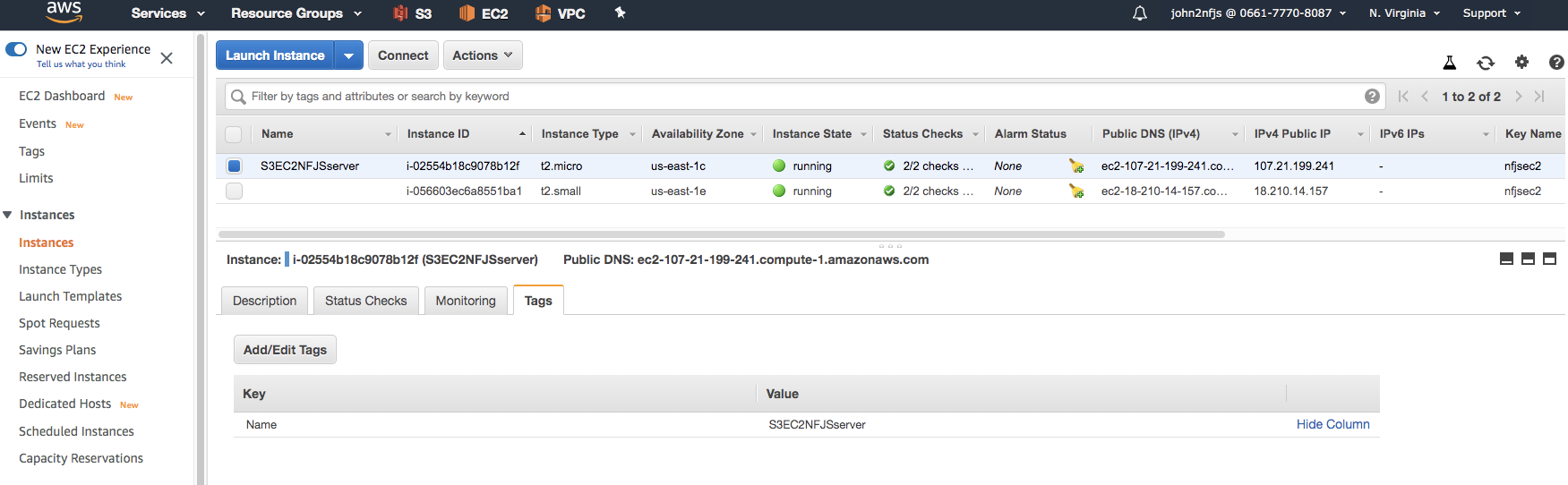
### Creating IAM Role





### Launching EC2 Instance





### Viewing the S3 Buckets in EC2

ssh -i nfjsec2.pem ec2-user@107.21.199.241

[ec2-user@ip-172-31-38-61 html]$ sudo su

[root@ip-172-31-38-61 html]# **aws s3 ls**

2020-07-07 18:57:06 nfjsbucketsnstest

2020-07-07 22:19:51 nfjsdestinationbucket1

2020-07-07 22:18:35 nfjssourcebucket1

2020-07-07 18:11:57 nfjstest

2020-06-23 03:58:45 test-bucketrohitbh

touch test22.txt

aws s3 mv test.txt s3://nfjstest