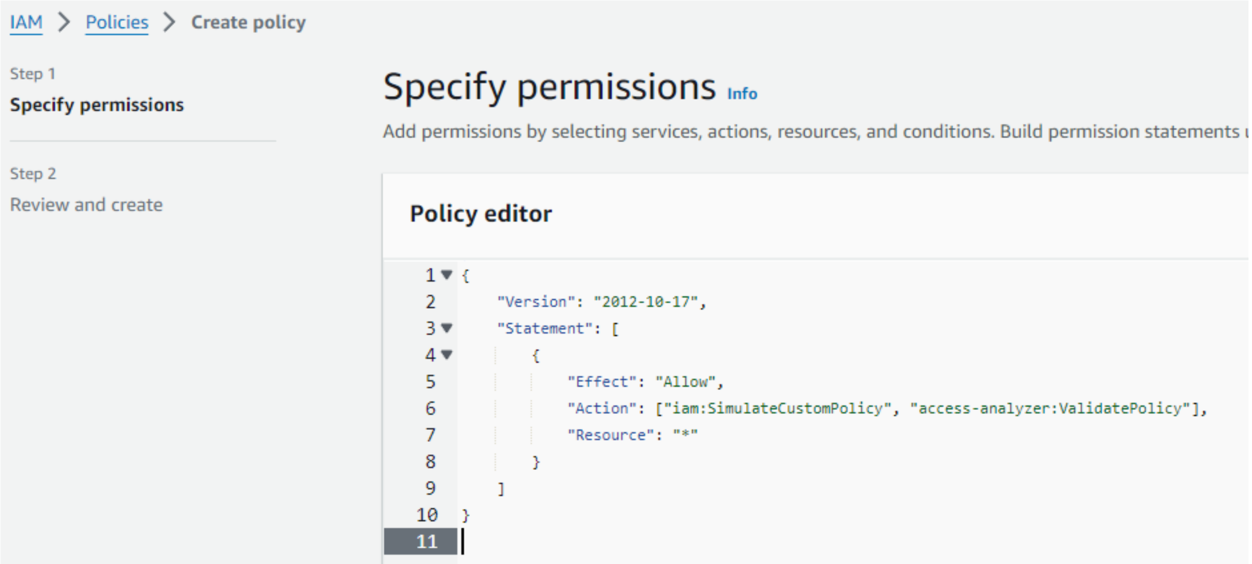
Setup the environment

In my journey as an AWS-certified professional, I've embarked on several exciting projects that really showcase my dedication to enhancing cloud security and making sure our operations run smoothly. Here's a quick look at four key policies I put together, each designed to tackle specific security and operational challenges:

1. Control who has permissions to access a resource through a resource-based policy, and scoping down access using a global condition context key
2. Control access to sensitive resources using tags through a SCP
3. Restrict the types of EC2 instances that a principal is allowed to launch through an identity-based policy
4. Ensure AWS services access your resources only on your behalf by adding conditions to a resource-based policy

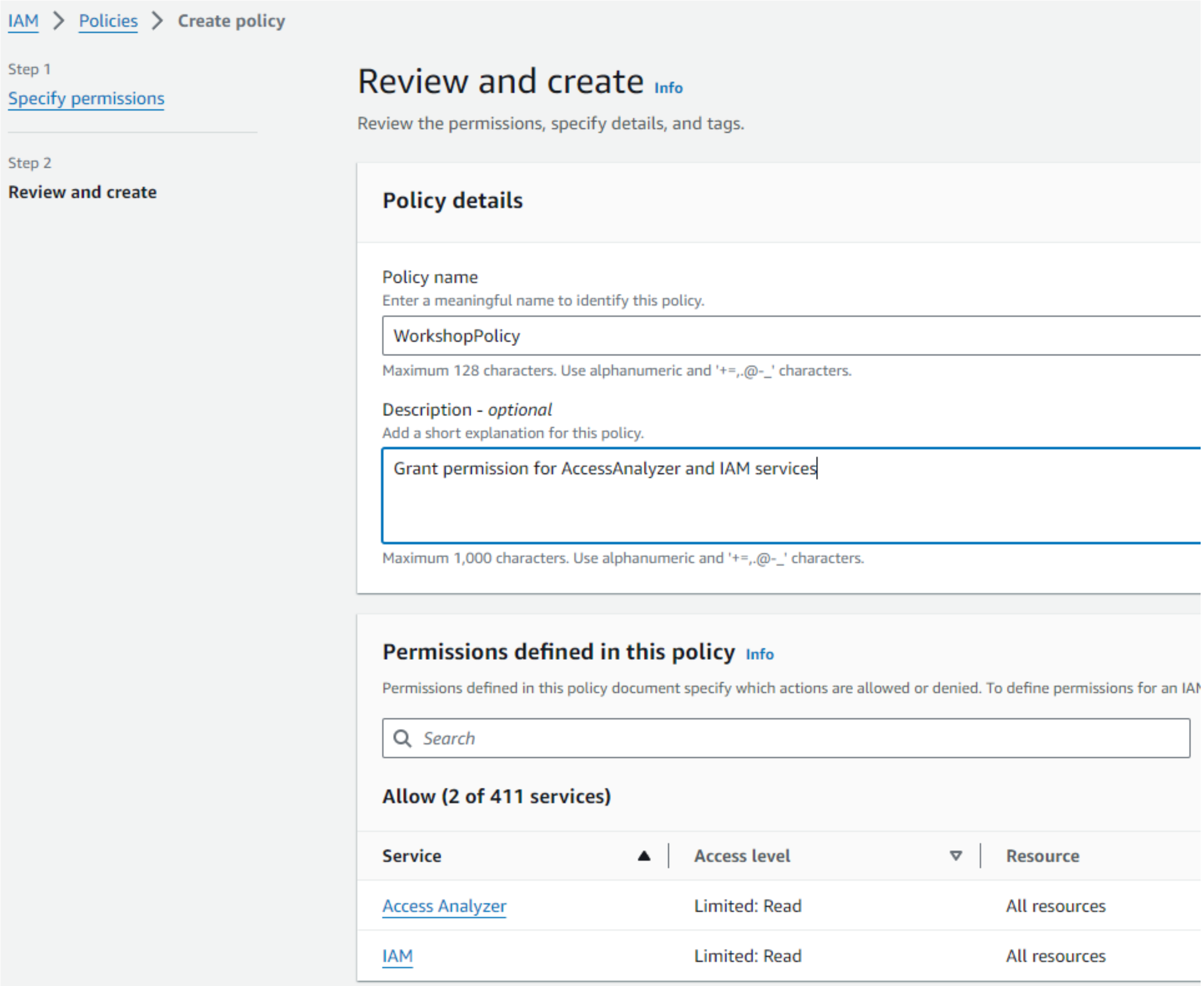
Services used: IAM, Access Analyzer, S3, SNS

1. I created a policy to grant permissions for those required for this labs: iam:SimulateCustomPolicy and access-analyzer:ValidatePolicy; and added it to my IAM user:



Add permissions by selecting services, actions, resources, and conditions. Build permission statements

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iam:SimulateCustomPolicy", "access-analyzer:ValidatePolicy"],
            "Resource": "*"
        }
    ]
}
```



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
WorkshopPolicy
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - *optional*
Add a short explanation for this policy.
Grant permission for AccessAnalyzer and IAM services
Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

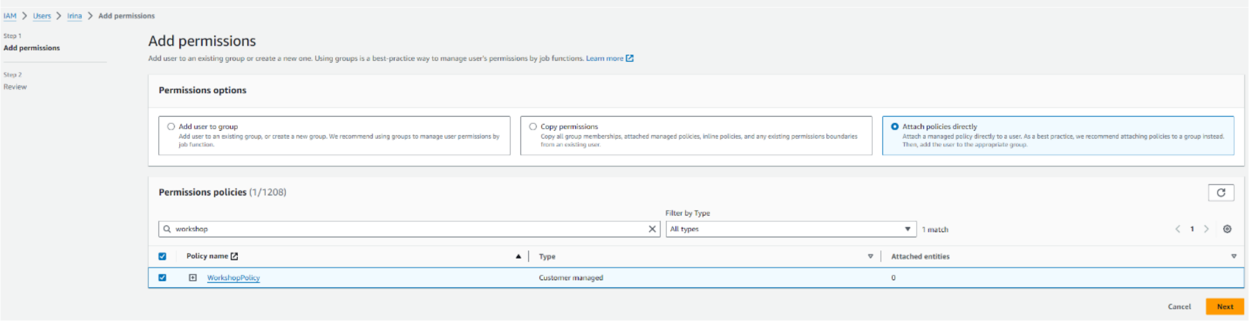
Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM

Search

Allow (2 of 411 services)

Service ▲	Access level ▽	Resource
Access Analyzer	Limited: Read	All resources
IAM	Limited: Read	All resources



IAM > Users > Irina > Add permissions

Step 1
Add permissions

Step 2
Review

Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1208)

workshop

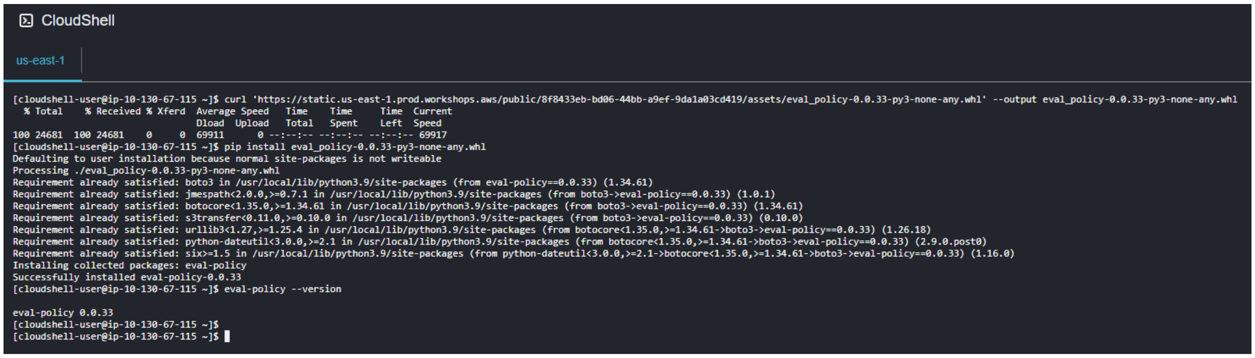
Filter by Type
All types
1 match

☑	Policy name ▲	Type ▽	Attached entities ▽
☑	WorkshopPolicy	Customer managed	0

I used CloudShell to check the policies validity.

2. I downloaded and installed the tool using the below commands:
   - curl 'https://static.us-east-1.prod.workshops.aws/public/8f8433eb-bd06-44bb-a9ef-9da1a03cd419/assets/eval_policy-0.0.33-py3-none-any.whl' --output eval_policy-0.0.33-py3-none-any.whl
   - pip install eval_policy-0.0.33-py3-none-any.whl

3. Verify the tool was correctly installed by running this command:
   - eval-policy --version



```
[cloudshell-user@ip-10-130-67-115 ~]$ curl 'https://static.us-east-1.prod.workshops.aws/public/8f8433eb-bd06-44bb-a9ef-9da1a03cd419/assets/eval_policy-0.0.33-py3-none-any.whl' --output eval_policy-0.0.33-py3-none-any.whl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 24681  100 24681    0     0  69911      0 --:--:-- --:--:-- --:--:-- 69917
[cloudshell-user@ip-10-130-67-115 ~]$ pip install eval_policy-0.0.33-py3-none-any.whl
Defaulting to user installation because normal site-packages is not writeable
Processing ./eval_policy-0.0.33-py3-none-any.whl
Requirement already satisfied: boto3 in /usr/local/lib/python3.9/site-packages (from eval-policy==0.0.33) (1.34.61)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.9/site-packages (from boto3->eval-policy==0.0.33) (1.0.1)
Requirement already satisfied: botocore<1.35.0,>=1.34.61 in /usr/local/lib/python3.9/site-packages (from boto3->eval-policy==0.0.33) (1.34.61)
Requirement already satisfied: s3transfer<0.11.0,>=0.10.0 in /usr/local/lib/python3.9/site-packages (from boto3->eval-policy==0.0.33) (0.10.0)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/local/lib/python3.9/site-packages (from botocore<1.35.0,>=1.34.61->boto3->eval-policy==0.0.33) (1.26.18)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.9/site-packages (from botocore<1.35.0,>=1.34.61->boto3->eval-policy==0.0.33) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.35.0,>=1.34.61->boto3->eval-policy==0.0.33) (1.16.0)
Installing collected packages: eval-policy
Successfully installed eval-policy-0.0.33
[cloudshell-user@ip-10-130-67-115 ~]$ eval-policy --version

eval-policy 0.0.33
[cloudshell-user@ip-10-130-67-115 ~]$
[cloudshell-user@ip-10-130-67-115 ~]$ 
```

All done, now we can create and validate the policies.