3. Limit the types of EC2 instances that can be launched with an identity-based policy

Use Case

The objective was to configure a policy that restricts the types of Amazon EC2 instances that can be launched within our organization, particularly focusing on limiting instances to type t3.small. This policy is aimed at controlling costs and ensuring that only smaller, less expensive instances are used in non-production environments.

Policy type - Identity-based policy

Values to use in policy:

Allowed instance type: t3.small
Action: ec2:RunInstances

The Approach

1. **Identifying the Right Condition Key:**

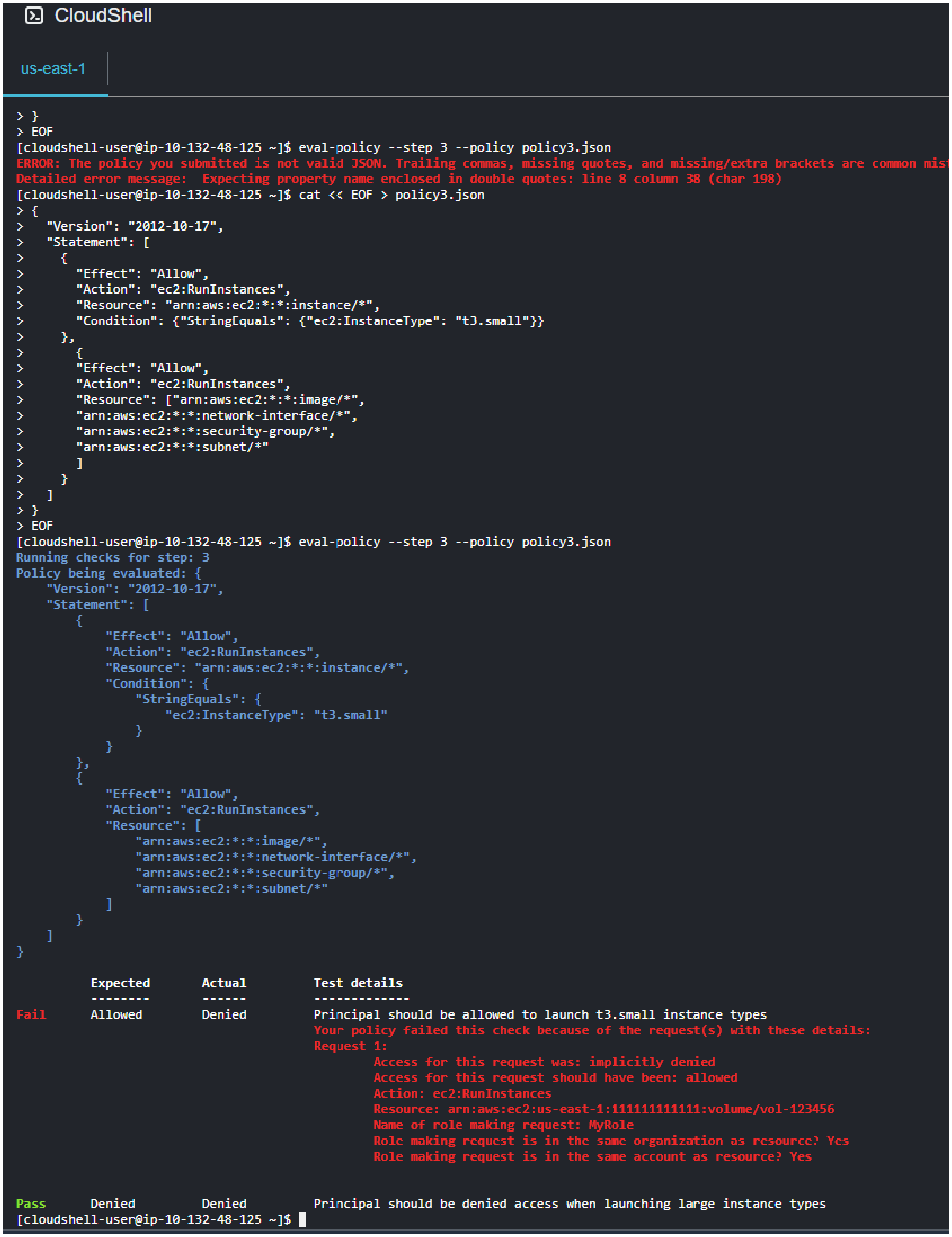
The key to enforcing this policy was the ec2:InstanceType condition key. This key allows control over which EC2 instance types a principal can launch, making it ideal for implementing cost control measures across development and testing environments.

2. **Understanding Policy Elements:**

Principal: Applies to all IAM roles/users within our AWS account due to its absence in the statement, thereby using the default inclusive setting.
Action: ec2:RunInstances, specifically targeting the action that launches EC2 instances.
Resource: Multiple resources are involved in launching an instance, such as the instance itself, associated images, network interfaces, security groups, subnets, and volumes. The resources that come along with ec2:RunInstances can be found here.
Condition: "StringEquals" - restricts the launch to only t3.small instances.

The Execution

Addressing Missing Resource ARNs:

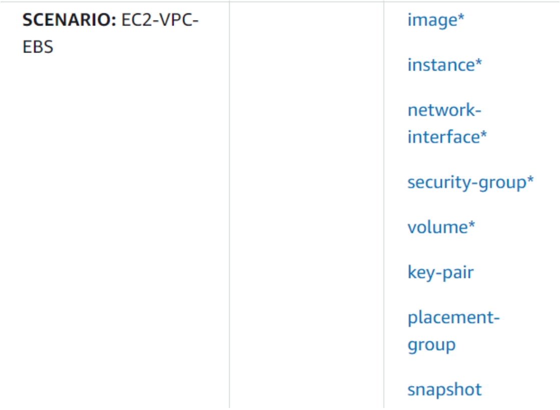


```
> }
> EOF
[cloudshell-user@ip-10-132-48-125 ~]$ eval-policy --step 3 --policy policy3.json
ERROR: The policy you submitted is not valid JSON. Trailing commas, missing quotes, and missing/extra brackets are common mist
Detailed error message:  Expecting property name enclosed in double quotes: line 8 column 38 (char 198)
[cloudshell-user@ip-10-132-48-125 ~]$ cat << EOF > policy3.json
> {
>   "Version": "2012-10-17",
>   "Statement": [
>     {
>       "Effect": "Allow",
>       "Action": "ec2:RunInstances",
>       "Resource": "arn:aws:ec2:*:*:instance/*",
>       "Condition": {"StringEquals": {"ec2:InstanceType": "t3.small"}}
>     },
>       {
>       "Effect": "Allow",
>       "Action": "ec2:RunInstances",
>       "Resource": ["arn:aws:ec2:*:*:image/*",
>       "arn:aws:ec2:*:*:network-interface/*",
>       "arn:aws:ec2:*:*:security-group/*",
>       "arn:aws:ec2:*:*:subnet/*"
>       ]
>     }
>   ]
> }
> EOF
[cloudshell-user@ip-10-132-48-125 ~]$ eval-policy --step 3 --policy policy3.json
Running checks for step: 3
Policy being evaluated: {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:InstanceType": "t3.small"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:image/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:subnet/*"
            ]
        }
    ]
}

          Expected       Actual         Test details
          --------       ------         ------------
Fail      Allowed        Denied         Principal should be allowed to launch t3.small instance types
                                        Your policy failed this check because of the request(s) with these details:
                                        Request 1:
                                                Access for this request was: implicitly denied
                                                Access for this request should have been: allowed
                                                Action: ec2:RunInstances
                                                Resource: arn:aws:ec2:us-east-1:111111111111:volume/vol-123456
                                                Name of role making request: MyRole
                                                Role making request is in the same organization as resource? Yes
                                                Role making request is in the same account as resource? Yes


Pass      Denied         Denied         Principal should be denied access when launching large instance types
[cloudshell-user@ip-10-132-48-125 ~]$ 
```

During the initial draft, the ARN for EBS volumes was omitted, which could have led to issues with launching instances since specifying all necessary resource ARNs is crucial for the ec2:RunInstances action. I omitted the volume's ARN because the mandatory ARN for the first scenario were those for image, network interface, security group, subnet. However, I understood that the scenario needed for this case were:

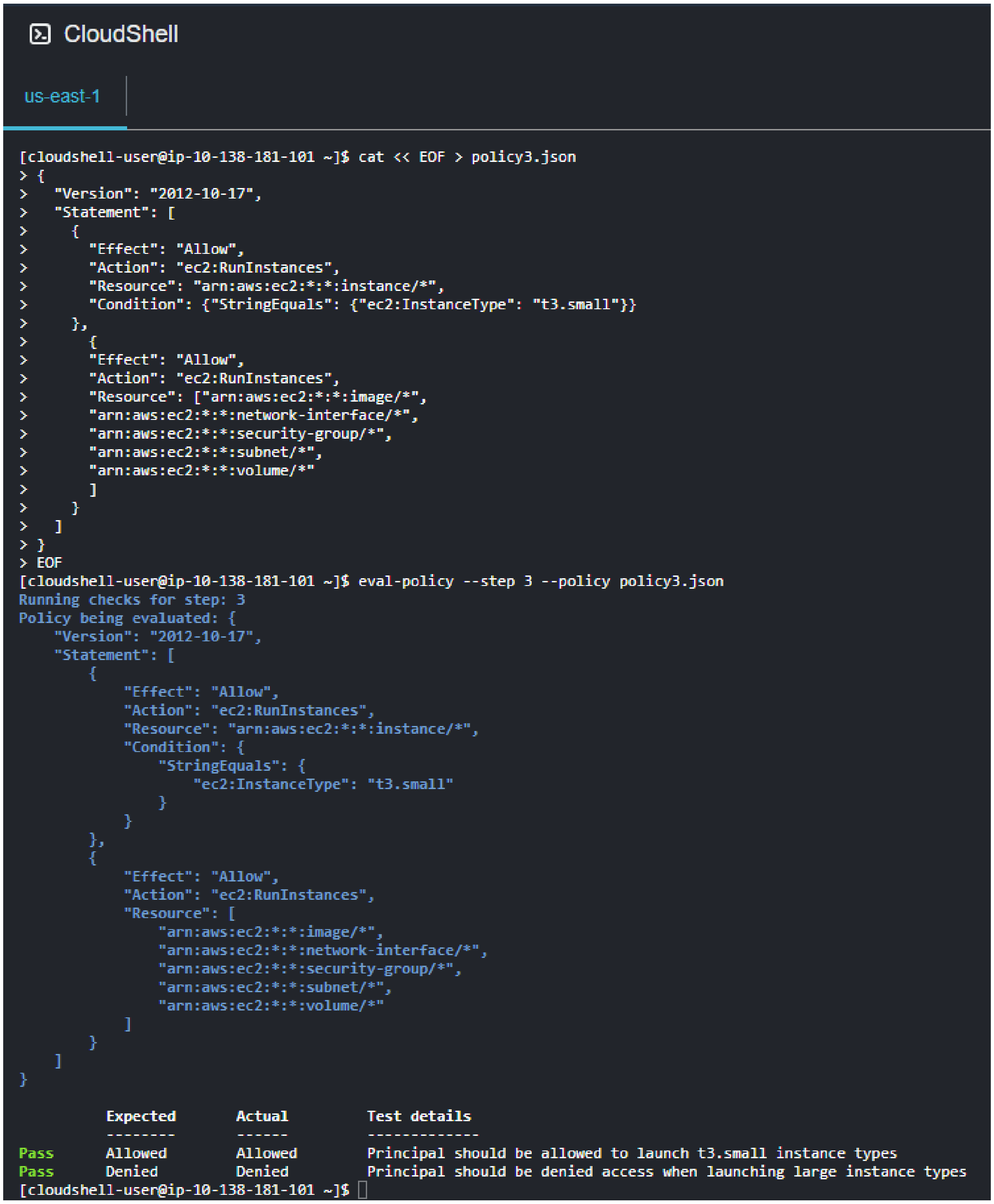


SCENARIO: EC2-VPC-EBS		image* instance* network-interface* security-group* volume* key-pair placement-group snapshot

Correcting the Policy:
After identifying the omission, the policy was corrected by adding the ARN for EBS volumes, ensuring that all related resources required for launching an instance were properly included. This ensures that permissions are correctly aligned with AWS's requirements for launching instances.

```
cat << EOF > policy3.json
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "ec2:RunInstances",
   "Resource": "arn:aws:ec2:*:*:instance/*",
   "Condition": {"StringEquals": {"ec2:InstanceType": "t3.small"}}
  },
   {
   "Effect": "Allow",
   "Action": "ec2:RunInstances",
   "Resource": ["arn:aws:ec2:*:*:image/*",
   "arn:aws:ec2:*:*:network-interface/*",
   "arn:aws:ec2:*:*:security-group/*",
   "arn:aws:ec2:*:*:subnet/*",
   "arn:aws:ec2:*:*:volume/*" // Added the ARN for EBS volumes
   ]
  }
 ]
}
EOF
```

Testing and Validation



```
[cloudshell-user@ip-10-138-181-101 ~]$ cat << EOF > policy3.json
> {
>   "Version": "2012-10-17",
>   "Statement": [
>     {
>       "Effect": "Allow",
>       "Action": "ec2:RunInstances",
>       "Resource": "arn:aws:ec2:*:*:instance/*",
>       "Condition": {"StringEquals": {"ec2:InstanceType": "t3.small"}}
>     },
>       {
>       "Effect": "Allow",
>       "Action": "ec2:RunInstances",
>       "Resource": ["arn:aws:ec2:*:*:image/*",
>       "arn:aws:ec2:*:*:network-interface/*",
>       "arn:aws:ec2:*:*:security-group/*",
>       "arn:aws:ec2:*:*:subnet/*",
>       "arn:aws:ec2:*:*:volume/*"
>       ]
>     }
>   ]
> }
> EOF
[cloudshell-user@ip-10-138-181-101 ~]$ eval-policy --step 3 --policy policy3.json
Running checks for step: 3
Policy being evaluated: {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:InstanceType": "t3.small"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*:*:image/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:volume/*"
            ]
        }
    ]
}

          Expected       Actual         Test details
          --------       ------         ------------
Pass      Allowed        Allowed        Principal should be allowed to launch t3.small instance types
Pass      Denied         Denied         Principal should be denied access when launching large instance types
[cloudshell-user@ip-10-138-181-101 ~]$ 
```

Evaluation Command: Ran eval-policy --step 3 --policy policy3.json to rigorously test the policy’s effectiveness in a controlled environment.

The Outcome

The policy was successfully implemented and met the organizational goal of restricting instance types to control costs in non-production environments. It allowed for the launch of t3.small instances only, effectively preventing the use of larger, more expensive instance types without the necessary approvals, thereby aligning with the company's cost-management strategies and operational efficiencies.