Project Documentation

EC2 Security Group Ingress Restriction using Lambda

# Objective

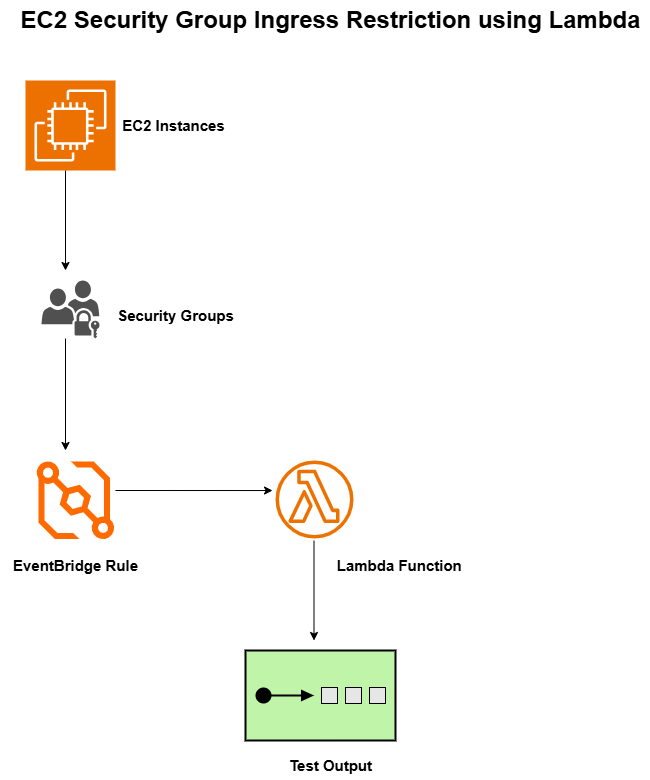
* Identify EC2 security groups that allow inbound access (ingress) from 0.0.0.0/0.
* Check for access to remote administration ports like:
* - SSH (22)
* - RDP (3389)
* Trigger a Lambda function when such rules are found.
* Identify EC2 instances associated with those security groups.
* Apply a tag (e.g., InsecureAccess=True) to the flagged instances.
* Display the flagged instances in the Lambda test output.

# Architecture

The architecture includes the following components:

* EC2 Instances with associated Security Groups.
* Security Groups that may have insecure rules allowing access from 0.0.0.0/0.
* A Lambda function that is triggered (either manually or via EventBridge).
* Lambda checks security groups, identifies affected EC2 instances, and tags them.
* Lambda test output or logs show the flagged instances.

# Architecture Diagram:



# Components Used

* AWS Lambda
* Amazon EC2
* Security Groups
* IAM Role for Lambda
* EventBridge Rule (optional for automation)
* AWS SDK (Boto3 - Python)

## Expected Output in Test Block

When you run/test the function, the output should look like this:



These are the EC2 instances that were found to be associated with security groups allowing risky ingress from 0.0.0.0/0 on ports 22 or 3389.

# Testing

* Deploy the Lambda function.
* Attach the required IAM role.
* Create a test event (empty JSON is fine).
* Manually add a security group with an insecure rule.
* Associate it with an EC2 instance.
* Run the test and check if the instance gets tagged and listed.

# Improvements (Optional)

* Send alerts via **SNS or email** when risky instances are found.
* Automatically **revoke** the insecure rule.
* Integrate with **AWS Config** for continuous monitoring.
* Create a **CloudWatch Alarm** if flagged instances exceed a threshold.