**Scenario 4:**

* Remove all the policies for QA group
* Create an iam policy for QA group which allows to read all ec2 and start and stop the ec2 instance if the instance type is t2.\*
* Also try evaluating with policy simulator.
* Solution:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "ec2:Describe\*",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": "elasticloadbalancing:Describe\*",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"cloudwatch:ListMetrics",

"cloudwatch:GetMetricStatistics",

"cloudwatch:Describe\*"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": "autoscaling:Describe\*",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"ec2:StartInstances",

"ec2:StopInstances",

"ec2:TerminateInstances"

],

"Resource": "\*",

"Condition": {

"StringLike": {

"ec2:InstanceType": "t2.\*"

}

}

}

]

}

**Scenario 5:**

* Create an IAM policy for mysql admins which will give read permission on all RDS activities but gives the access to
  + CREATE, DELETE databases only if they are mysql
* Try to create postgres admins which will give read permission on all RDS activites but gives access to
  + CREATE, DELETE databases only if they are postgres
* For creating the IAM Policies refer to actions and resources <https://docs.aws.amazon.com/service-authorization/latest/reference/list_amazonrds.html>
* For creating the cluster we can use condition key rds:Database engine

{

"Version": "2012-10-17",

"Statement": [

{

"Action": [

"rds:Describe\*",

"rds:ListTagsForResource",

"ec2:DescribeAccountAttributes",

"ec2:DescribeAvailabilityZones",

"ec2:DescribeInternetGateways",

"ec2:DescribeSecurityGroups",

"ec2:DescribeSubnets",

"ec2:DescribeVpcAttribute",

"ec2:DescribeVpcs"

],

"Effect": "Allow",

"Resource":