# Attach IAM Policy To IAM Role in AWS.

This catalogue is used for attaching IAM policy to IAM role. Upon selecting the role , it will list down the current policies of the specific role.

## Service Now Workflow:

## Sample Payload

{

"ScriptFiles": [

{

"Name": "IAM\_Role\_Policy\_attachment.py",

"Type": "PY",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/IAM Management/IAM\_Role\_Policy\_attachment.py"

}

],

"ScriptPayload": {

"Region": "us-west-2",

"RoleName": "test4",

"PolicyArn": [

"arn:aws:iam::957654751236:policy/InstanceConnect",

"arn:aws:iam::aws:policy/AlexaForBusinessFullAccess",

"arn:aws:iam::aws:policy/CloudFrontFullAccess",

"arn:aws:iam::aws:policy/AmazonAthenaFullAccess",

"arn:aws:iam::aws:policy/CloudWatchSyntheticsFullAccess",

"arn:aws:iam::aws:policy/job-function/Billing",

"arn:aws:iam::aws:policy/IAMFullAccess",

"arn:aws:iam::aws:policy/AmazonAppStreamFullAccess"

],

"StackName": "SCTASK0013231",

"Tags": [

{

"Key": "Owner",

"Value": "Cloud\_Exponence"

},

{

"Key": "Project",

"Value": "IT"

},

{

"Key": "BusinessUnit",

"Value": "IT IS"

},

{

"Key": "Application",

"Value": "Demo"

},

{

"Key": "Environment",

"Value": "Non Production"

},

{

"Key": "CreatedBy",

"Value": "CloudExponence"

},

{

"Key": "RequestID",

"Value": "RITM0012432"

}

]

}

}

### Developer Notes:

Attach IAM Policy To IAM Role – Form Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Validation if any** |
| Role Name | Lookup Select Box | Role for which policy to be attached | None |
| Current Policies of the role | MultiLine Text | Displays current policies of role | Validate Current Policies – Role , OnChangeof Role |
| Policy Details | List Collector | Policy to be attached to role | OnChangeOfPolicy |
| PolicyCheck | Check Box | To check if all validations are satisfied | OnSubmit |

Internal Mapping in Form

|  |  |  |
| --- | --- | --- |
| **Key Name** | **Type** | **Table** |
| RoleName | Lookup Select Box | CE IAM Role[u\_cmdb\_ci\_ce\_iam\_role] |
| Policy Name | List Collector | CE IAM Policy[u\_cmdb\_ci\_ce\_iam\_policy] |

*Note : Policy name are collected where path is not equal to service-role.*

\*Validate Current Policies – Role :

This Script validates if the current policies is equal to 10.Then it will pop up an error message to choose another role as maximum policy per role is 10.

\*On Change of Policy :

This script is to validate the number of policies per role must be 10.

\*On Change of Role:

This script is to fetch the current policies of the role selected. This will invoke backend script ‘ce\_count\_policy\_for\_role\_aws’ which in turn retrieve current policies of the role via REST API ‘AWSOAUTHENDPOINT’.

\*OnSubmit:

This script is to validate the variable policycheck. The request will be processed only if the variable returns true.

Automation Workflow:

The “ScriptFiles” section contains the scripts location in the github repo and ScriptPayload” contains the list of inputs to be passed.

The ResourceDeploymentStepapigateway invokes ResourceDeploymentSingleStepFunction.

The following operations happen in the “ResourceDeploymentSingle”stepfunction:

1. The“ResourceDeploymentStep” lambda is invoked .
2. Lambda will check the input format of payload and verifies it. It also get the credentials through the role associated to it for the below executions.
3. It will get the git repo credentials from the SSM parameter store and then download the IAM\_Role\_Policy\_attachment.py files in s3.
4. Then the inputs will be passed to the python file where it will trigger the CloudFormation Template file from S3 bucket if the stack has to be created.If there is no stack creation, python file itself will do the required activity.
5. The final activity of the step function is to notify success message to servicenow if the stack is successfully created.
6. If there is no stack creation involved, it will directly notify the servicenow after the python scripts successfully executes.

Error Handling:

1. Step function will check if the stack is successfully created. It will notify service now of the status of stack creation if error
2. If there is no stack creation, it will notify servicenow if there are any error from python scripts.
3. Any other runtime error from step function /lambda is also notified to servicenow.