# Create Security Group in AWS.

Security group creation catalogue is used to provision security group(virtual firewall) in AWS Environment and also it will do the below functionalities during the provision.

* Add the ingress and Egress rules to the security group
* VPC Name will be populated based on Region and account ID we selected .
* Security Group Ingress – Adds an Inbound Rule to a security group. It will allow instance to receive traffic from specified protocol and CIDR IP range.
* Security Group Egress – Adds Outbound Rule to a Security Group. It will allow instance to send traffic to destination port or CIDR IP range.
* In this catalogue, we can add maximum three number of rules for each inbound and outbound.
* When we select the number of rules required, It will display Rule details accordingly through Catalog UI Policies.
* We can also create security group without rules.
* We are passing the egress and ingress rule values dynamically through lambda function.

## Service Now Workflow:

## Sample Payload

{

"ScriptPayload": {

"Parameters": {

"sgGroupName": "CEnpANSBLITSecurityGroup001",

"GroupDescription": "This is for testing purpose",

"VpcId": "vpc-4204933a"

},

"templateData": {

"AWS::EC2::SecurityGroup": {

"SecurityGroupIngress": [

{

"IpProtocol": "tcp",

"CidrIp": "10.20.30.40/16",

"FromPort": "443",

"ToPort": "443"

},

{

"IpProtocol": "tcp",

"CidrIp": "10.20.30.41/16",

"FromPort": "3389",

"ToPort": "3389"

}

],

"SecurityGroupEgress": [

{

"IpProtocol": "tcp",

"CidrIp": "10.20.30.42/17",

"FromPort": "80",

"ToPort": "80"

}

]

}

},

"Tags": [

{

"Key": "Owner",

"Value": "Cloud\_Exponence"

},

{

"Key": "Project",

"Value": "IT"

},

{

"Key": "BusinessUnit",

"Value": "IT IS"

},

{

"Key": "Application",

"Value": "Ansible"

},

{

"Key": "Environment",

"Value": "Non Production"

},

{

"Key": "CreatedBy",

"Value": "CloudExponence"

},

{

"Key": "RequestID",

"Value": "RITM0012437"

}

],

"StackName": "SCTASK0013236",

"Region": "us-west-2"

}

}

### Developer Notes:

Security Group Creation – Form Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Validation if any** |
| Group Description | Single Line text | Description of the Security Group | None |
| VPC Name | Lookup Select Box | The ID of the VPC for Security Group | None |
| How Many Rules Do you want to Add?(Inbound Rule) | Select Box | Number of Inbound Rules[0-3] | None |
| IP Protocol(ip\_protocol1) | Select Box | IP Protocol Name | Port Mapping for Protocol 1 |
| CIDR IP(cidr\_ip1) | Single Line text | IP Address in CIDR Format | Validate IP1 |
| FromPort(fromport1) | Single Line text | Start of Port for protocol selected | Validate FromPort1 (if protocol is UDP or TCP) |
| ToPort(toport1) | Single Line text | End of port for protocol selected | Validate ToPort1 (if protocol is UDP or TCP) |
| IP Protocol(ip\_protocol2) | Select Box | IP Protocol Name | Port Mapping for Protocol 2 |
| CIDR IP(cidr\_ip2) | Single Line text | IP Address in CIDR Format | Validate IP2 |
| FromPort(fromport2) | Single Line text | Start of Port for protocol selected | Validate FromPort2 (if protocol is UDP or TCP) |
| ToPort(toport2) | Single Line text | End of port for protocol selected | Validate ToPort2 (if protocol is UDP or TCP) |
| IP Protocol(ip\_protocol3) | Select Box | IP Protocol Name | Port Mapping for Protocol 3 |
| CIDR IP(cidr\_ip3) | Single Line text | IP Address in CIDR Format | Validate IP3 |
| FromPort(fromport3) | Single Line text | Start of Port for protocol selected | Validate FromPort3 (if protocol is UDP or TCP) |
| ToPort(toport3) | Single Line text | End of port for protocol selected | Validate ToPort3 (if protocol is UDP or TCP) |
| How Many Rules Do you want to Add?(Outbound Rule) | Select Box | Number of Outbound Rules[0-3] | None |
| IP Protocol(ip\_protocol4) | Select Box | IP Protocol Name | Port Mapping for Protocol 4 |
| CIDR IP(cidr\_ip4) | Single Line text | IP Address in CIDR Format | Validate IP4 |
| FromPort(fromport4) | Single Line text | Start of Port for protocol selected | Validate FromPort4 (if protocol is UDP or TCP) |
| ToPort(toport4) | Single Line text | End of port for protocol selected | Validate ToPort4 (if protocol is UDP or TCP) |
| IP Protocol(ip\_protocol5) | Select Box | IP Protocol Name | Port Mapping for Protocol 5 |
| CIDR IP(cidr\_ip5) | Single Line text | IP Address in CIDR Format | Validate IP5 |
| FromPort(fromport5) | Single Line text | Start of Port for protocol selected | Validate FromPort5 (if protocol is UDP or TCP) |
| ToPort(toport5) | Single Line text | End of port for protocol selected | Validate ToPort5 (if protocol is UDP or TCP) |
| IP Protocol(ip\_protocol6) | Select Box | IP Protocol Name | Port Mapping for Protocol 6 |
| CIDR IP(cidr\_ip6) | Single Line text | IP Address in CIDR Format | Validate IP6 |
| FromPort(fromport6) | Single Line text | Start of Port for protocol selected | Validate FromPort6 (if protocol is UDP or TCP) |
| ToPort(toport6) | Single Line text | End of port for protocol selected | Validate ToPort6 (if protocol is UDP or TCP) |

Internal Mapping in Form

|  |  |  |  |
| --- | --- | --- | --- |
| **Key Name** | **Type** | **Table** |  |
| VPC Name | Lookup Select Box | CE Virtual Network[u\_cmdb\_ci\_ce\_virtual\_network] |  |
| Security Group Name | String | Got from Namepattern Table |  |

*Note : VPC Name will be populated based on account id and location*

The following IP Protocols are listed in select box

|  |
| --- |
| * Custom TCP |
| * Custom UDP |
| * All ICMP - IPv4 |
| * All ICMP - IPv6 |
| * SSH |
| * SMTP |
| * DNS(UDP) |
| * DNS(TCP) |
| * HTTP |
| * POP3 |
| * IMAP |
| * LDAP |
| * HTTPS |
| * SMB |
| * SMTPS |
| * IMAPS |
| * POP3S |
| * MSSQL |
| * NFS |
| * MYSQL/Aurora |
| * RDP |
| * RedShift |
| * PostgreSQL |
| * Oracle-RDS |
| * WinRM-HTTP |
| * WinRM-HTTPS |
| * Elastic Graphics |

\*Port Mapping for Protocol[n]:

This Script is to map the fromport and toport of the ip when a particular protocol is chosen.For example , when HTTP is selected , their fromport and toport will be 80.

\*Validate IP[n] :

The script is validate to the given ip address matches the regex pattern.The IP address should be in format of m.m.m.m/[a] where m can be 0-255 and a can be 16-30.

\*Validate Fromport[n] :

If the protocol is Custom TCP and Custom UDP , the fromport should be between 0 to 65535.

\*Validate ToPort[n]:

If the protocol is Custom TCP and Custom UDP , toport should be between 0 to 65535.

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 lambda\_function.py and template.json 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.