# Create ALB in AWS.

This catalogue is used to provision Application Load Balancer in AWS Environment and also it will do the below functionalities during the provision.

* This works at the Seventh layer of OSI Layer.
* This load balancer distributes the incoming application traffic across multiple targets such as EC2 instances.
* We can add maximum number of five listeners to this catalogue.
* Based on number of listener we select , it will display listener details accordingly through Catalog UI Policies.
* These listeners checks for request from clients using the protocol and port which we configure.
* Each port should be unique.
* Listener rules route requests to Existing / Newly created target groups.
* For the newly created target groups , we can configure the health check so that listener routes only to the healthy targets.

## Service Now Workflow:

## Sample Payload

{

"ScriptFiles": [

{

"Name": "ALBCreation.py",

"Type": "PY",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/ALB/ALB.py"

},

{

"Name": "ALBCreation.yml",

"Type": "CFT",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/ALB/ALB.yml"

}

],

"ScriptPayload": {

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

"ALBName": "CEnpANSBLITALB032",

"Subnets": "subnet-992bcbe1,subnet-e6aac0cd",

"VPCName": "vpc-4204933a",

"LoadBalancerType": "internet-facing",

"SecurityGroups": "",

"StackName": "SCTASK0013267",

"Listener1Details": "HTTP,148,,ELBSecurityPolicy-2016-08",

"Listener2Details": "HTTP,145,,ELBSecurityPolicy-2016-08",

"Listener3Details": ",,,",

"Listener4Details": ",,,",

"Listener5Details": ",,,",

"TargetGroupType": "new\_target\_group",

"ExistingTargetGroupArn": "",

"TargetGroupName": "CEnpANSBLITALB-TG031",

"HealthCheckPath": "/",

"HealthCheckPort": "",

"HealthCheckProtocol": "HTTP",

"HealthCheckIntervalSeconds": "30",

"HealthCheckTimeoutSeconds": "5",

"PortNo": "80",

"ProtocolType": "HTTP",

"UnhealthyThresholdCount": "2",

"HealthyThresholdCount": "5",

"SuccessCode": "200",

"Target1": "i-0f4ddbaa7cf3394d6",

"Target2": "",

"Target3": "",

"Target4": "",

"Target5": "",

"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": "RITM0012465"

}

]

}

}

### Developer Notes:

ALB Creation – Form Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Validation if any** |
| Scheme Internet Facing/Internal None | Select Box | Internet Facing/Internal | None |
| VPC Name | Lookup Select Box | VPC ID | None |
| Subnets | List Collector | Name of the Subnet | Validate Subnet(Atleast two subnets should be selected) |
| Security Group | List Collector | Security Group Name | None |
| How many Listeners do you want to Add | Select Box | No. of Listeners[1-5] | None |
| Load Protocol 1 | Select Box | HTTP/HTTPS | None |
| Load Port 1 | Single Line Text | Port in which request will be configured | Validate Port1 |
| Load Protocol 2 | Select Box | HTTP/HTTPS | None |
| Load Port 2 | Single Line Text | Port in which request will be configured | Validate Port2 |
| Load Protocol 3 | Select Box | HTTP/HTTPS | None |
| Load Port 3 | Single Line Text | Port in which request will be configured | Validate Port3 |
| Load Protocol 4 | Select Box | HTTP/HTTPS | None |
| Load Port 4 | Single Line Text | Port in which request will be configured | Validate Port4 |
| Load Protocol 5 | Select Box | HTTP/HTTPS | None |
| Load Port 5 | Single Line Text | Port in which request will be configured | Validate Port5 |
| Certificate ARN | Single Line Text | Certificate for the HTTPS Protocol selected | None |
| SSL Policy | Select Box | Security policy | None |
| Target Group | Select Box | Existing / New Target group. | None |
| Target Group Name(Existing) | Select Box | Target group to which client should route the request | Populate Target Group(Target group is populated with api call) |
| Protocol | Select Box | Protocol of the target | None |
| Port | Single Line Text | Port of the target | Validate Port |
| Health Check Protocol | Select Box | Protocol in which health check is configured | None |
| Health Check Path | Single Line Text | Path for health check | None |
| Override Port | Single Line Text | Override port if given port Isn’t available | None |
| Success Code | Single Line Text | Success code for health check | None |
| Healthy Threshold | Single Line Text | Number of health check successes | Validate Healthy Threshold – Must be integer(2-10) |
| Unhealthy Threshold | Single Line Text | Number of health check failures | Validate Unhealthy Threshold Must be integer(2-10) |
| Timeout | Single Line Text | Amount of time In which no response means failure(in seconds) | Validate Timeout – Must be Integer(2-60) and less than interval |
| Interval | Single Line Text | The time between health check | Validate Interval |
| Targets | List Collector | Instances | Validate Targets(Maximum selected target must be 5) |
| Alb\_check | Check Box | To check if all validations are satisfied | OnSubmit |

*\*Validate Port : The port number should be integer between 1 and 65535.*

*And Each port number should be unique for each protocol.*

Internal Mapping

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Table** |
| Subnets Internet Facing/Internal None | List Collector | Subnet ID | CE Subnets |
| Security Group | List Collector | Security Group ID | CE Network Security Group |
| TargetGroup Name | String | Name of the Target Group | Got from Namepattern Table |
| ALB Name | String | Name of the Application Load Balancer | Got from Namepattern Table |

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 .py and EC2Creation.yml 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.