# Create Subnet in AWS.

Subnet Creation catalogue is used to provision subnet with user provided subnet address in AWS Environment .

* VPC will be populated based on account id and region we select.
* When the VPC is selected , it automatically input the VPC address ,accordingly subnet address can be provisioned with user preferred availability zone.

## Service Now Workflow:

## Sample Payload

{

"ScriptFiles": [

{

"Name": "SubnetCreation.py",

"Type": "PY",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/Subnet\_Provisioning/SubnetCreation\_V2.py"

},

{

"Name": "SubnetCreation.yml",

"Type": "CFT",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/Subnet\_Provisioning/SubnetCreation\_V1.yml"

}

],

"ScriptPayload": {

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

"SubnetName": "CEnpANSBLITSubnet040",

"AvailabilityZone": "us-west-2a",

"SubnetCIDRblock": "172.31.115.0/24",

"MapPublicIpOnLaunch": "true",

"StackName": "SCTASK0013264",

"VpcId": "vpc-4204933a",

"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": "RITM0012463"

}

]

}

}

### Developer Notes:

Subnet Creation – Form Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Validation if any** |
| VPC Name | Lookup Select Box | The ID of the VPC the subnet is in. | None |
| VPC Address | Single Line Text | The IP Address of the VPC | None |
| CIDR Block | Single Line Text | The IPv4 CIDR block assigned to the subnet. | Validation- Subnet addres |
| Availability Zone | Lookup Select Box | Availability zone of the subnet | None |
| Map Public IP On Launch | Check Box | Indicates whether instances launched receive a public IPv4 address | None |
| Assign IPv6 Address on Creation | Check Box | Indicates whether instances launched receive an IPv6 address | None |
| IPv6 CIDR Block | Single Line Text | The IPv6 CIDR block | None |

Internal Mapping in Form

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Mapping** |
| SubnetName | String | Name of the Subnet | 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 SubnetCreation\_V2.py and SubnetCreation\_V1.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.