# Create VPC in AWS.

VPC creation catalogue is used to provision VPC where we can launch our AWS resources in a virtual network that we define and also it will do the below functionalities.

* Enable DNS Host name and Enable DNS Support
* Tenancy can be specified either as Default or Dedicated.

## Service Now Workflow:

## Sample Payload

{

"ScriptFiles": [

{

"Name": "VPCCreation.py",

"Type": "PY",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/VPC\_Provisioning/VPCCreation\_V2.py"

},

{

"Name": "VPCCreation.yml",

"Type": "CFT",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/VPC\_Provisioning/VPCCreation\_V1.yml"

}

],

"ScriptPayload": {

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

"VPCName": "CEDevFinVPC064",

"CIDRblock": "10.2.0.0/16",

"EnableDnsHostnames": "false",

"EnableDnsSupport": "false",

"InstanceTenancy": "default",

"StackName": "RITM0010787-VPCCreation",

"Tags": [

{

"Key": "Owner",

"Value": "System Administrator"

},

{

"Key": "Project",

"Value": "Customer Support"

},

{

"Key": "BusinessUnit",

"Value": "Finance"

},

{

"Key": "Application",

"Value": "apache linux den 200"

},

{

"Key": "Environment",

"Value": "Development"

},

{

"Key": "CreatedBy",

"Value": "CloudExponence"

},

{

"Key": "RequestID",

"Value": "RITM0010787"

}

]

}

}

### Developer Notes:

IAM User Creation – Form Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Validation if any** |
| VPC CIDR | Single Line text | To input VPC CIDR block | CIDR type validation(The format should be in n.n.n.n) |
| Tenancy | Select Box | Default/Dedicated/Host | None |
| Enable DNS hostnames | Check box | If to enable DNS host name for the VPC | None |
| Enable DNS Support | Check box | If to enable DNS Support for the VPC | None |

Internal Mapping in Form

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Mapping** |
| VPC name | String | Name of VPC | Got Name from Name Pattern 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 VPCCreation.py and VPCCreation.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.