

Yeshwanth chauhan

Creating vpc and Ec2 instances using json script

This screenshot shows the AWS CloudFormation console for the 'enstance001' stack. The left sidebar displays a list of stacks, including 'enstance001' and 'vpc111', both with a status of 'CREATE_COMPLETE'. The main panel shows the 'enstance001' stack details, including its ID, description, status, and creation time.

Stacks (2)

- enstance001
2023-02-01 22:14:43 UTC+0530
CREATE_COMPLETE
- vpc111
2023-02-01 22:00:20 UTC+0530
CREATE_COMPLETE

enstance001

Stack ID: am:aws:cloudformation:us-east-1:1471963814578:stack/enstance001/ba94bf60-a24f-11ed-8919-0a1ad0f0f839

Description: Template Creates a single EC2 instance with a single ENI which has multiple private and public IPs

Status: CREATE_COMPLETE

Status reason: -

Root stack: -

Parent stack: -

Created time: 2023-02-01 22:14:43 UTC+0530

Deleted time: -

Updated time: -

This screenshot shows the AWS CloudFormation console for the 'vpc111' stack. The left sidebar displays a list of stacks, including 'enstance001' and 'vpc111', both with a status of 'CREATE_COMPLETE'. The main panel shows the 'vpc111' stack details, including its ID, description, status, and creation time.

Stacks (2)

- enstance001
2023-02-01 22:14:43 UTC+0530
CREATE_COMPLETE
- vpc111
2023-02-01 22:00:20 UTC+0530
CREATE_COMPLETE

vpc111

Stack ID: am:aws:cloudformation:us-east-1:1471963814578:stack/vpc111/b847fb20-a24d-11ed-8bce-0e49cf81c389

Description: -

Status: CREATE_COMPLETE

Status reason: -

Root stack: -

Parent stack: -

Created time: 2023-02-01 22:00:20 UTC+0530

Deleted time: -

Updated time: -

Yeshwanth chauhan

Creating vpc and Ec2 instances using json script

us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#vpcs:

Virtual private cloud

Your VPCs

Name	env	VPC ID	State	IPv4 CIDR	IPv6 CIDR	Main route table
VPCXX...	-	vpc-079d4f644883dd0fd	Available	172.18.0.0/16	-	rtb-0e6f50565e90a3d65

vpc-079d4f644883dd0fd / VPCXX-2023

Activate Windows
Go to Settings to activate Windows.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances?v=3&case=tag:true%5C,client:false;regex=tags:false%5C,client:false

Instances

Name	Instance ID	Instance state	Availability Zone	Private IP address	Subnet IDs
-	i-0ae5853093a1be4e8	Running	us-east-1a	172.18.0.80	subnet-0e182873b02557945

Instance: i-0ae5853093a1be4e8

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

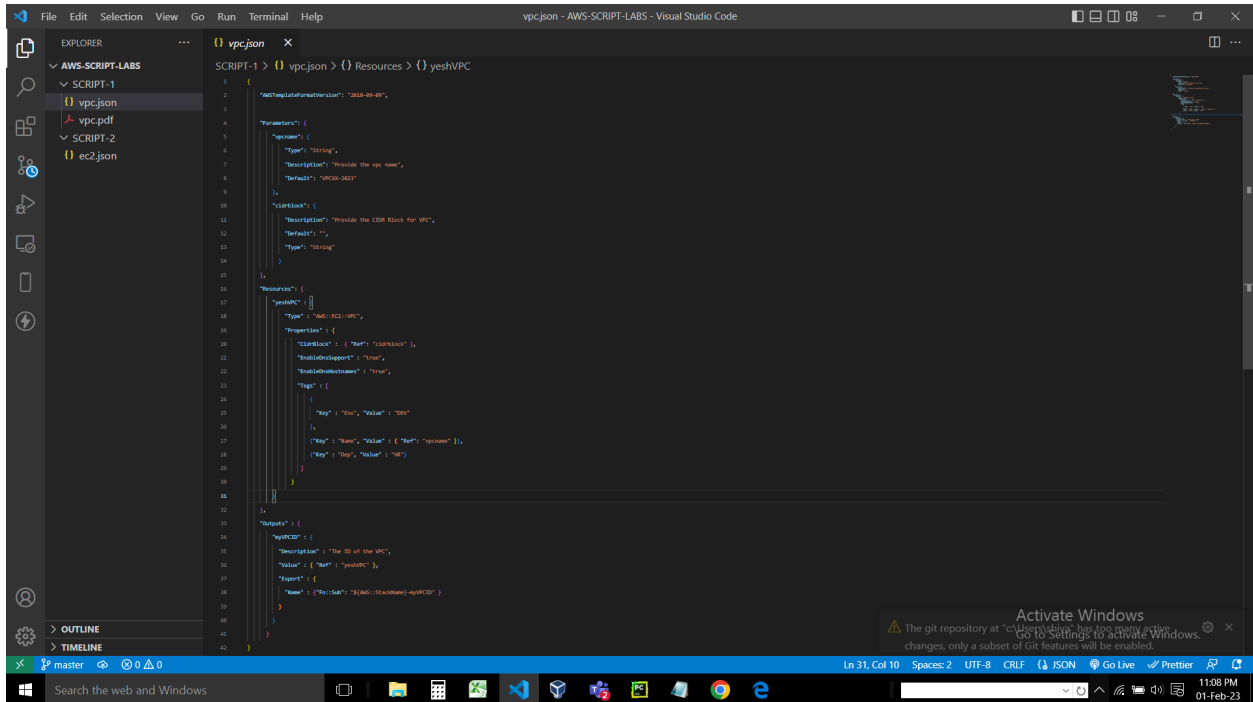
Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ae5853093a1be4e8	-	172.18.0.80
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-18-0-80.ec2.internal	ip-172-18-0-80.ec2.internal	-
Answer private resource DNS name	Instance type	
-	t2.micro	
Auto-assigned IP address	VPC ID	

Activate Windows
Go to Settings to activate Windows.

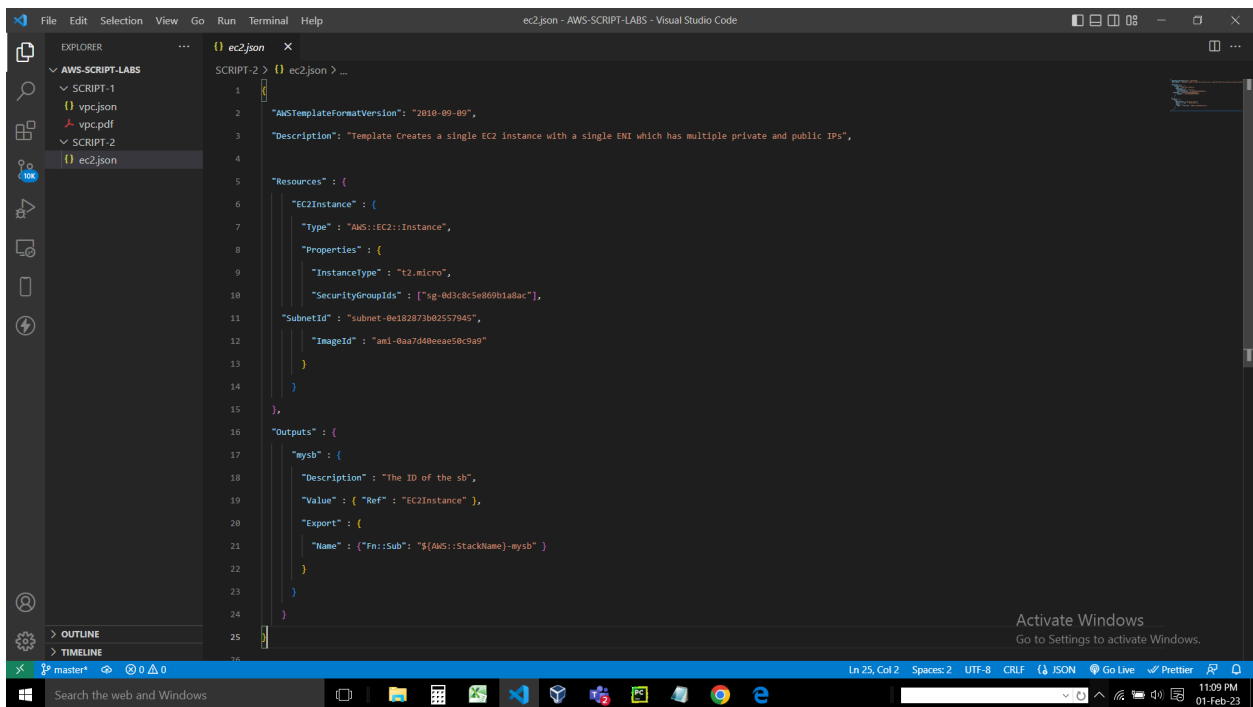
Yeshwanth chauhan

Creating vpc and Ec2 instances using json script



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'AWS-SCRIPT-LABS'. Under 'SCRIPT-1', the file 'vpc.json' is selected. The main editor window shows the content of 'vpc.json', which is an AWS CloudFormation template for creating a VPC. The terminal at the bottom shows the command 'vpcjson > () Resources > () yeshVPC'. The status bar at the bottom indicates 'Ln 31, Col 10', 'Spaces: 2', 'UTF-8', 'CRLF', 'JSON', 'Go Live', 'Prettier', and the time '11:08 PM 01-Feb-23'.

```
1 {
2   "AWSTemplateFormatVersion": "2010-09-09",
3   "Parameters": {
4     "VpcName": {
5       "Type": "String",
6       "Description": "Provide the vpc name",
7       "Default": "VPC-2023"
8     }
9   },
10  "Resources": {
11    "VPC": {
12      "Type": "AWS::EC2::VPC",
13      "Properties": {
14        "CidrBlock": { "Ref": "cidrblock" },
15        "EnableDnsSupport": "true",
16        "EnableDnsHostnames": "true",
17        "Tags": [
18          {
19            "Key": "Name", "Value": "VPC"
20          },
21          {
22            "Key": "Name", "Value": { "Ref": "VpcName" }
23          },
24          {
25            "Key": "Tag", "Value": "VPC"
26          }
27        ]
28      }
29    },
30    "cidrblock": {
31      "Type": "AWS::EC2::VPC",
32      "Properties": {
33        "CidrBlock": { "Ref": "cidrblock" },
34        "EnableDnsSupport": "true",
35        "EnableDnsHostnames": "true",
36        "Tags": [
37          {
38            "Key": "Name", "Value": "VPC"
39          },
40          {
41            "Key": "Tag", "Value": "VPC"
42          }
43        ]
44      }
45    }
46  },
47  "Outputs": {
48    "VPCID": {
49      "Description": "The ID of the VPC",
50      "Value": { "Ref": "VPC" },
51      "Export": {
52        "Name": { "Fn::Sub": "${AWS::StackName}-VPCID" }
53      }
54    }
55  }
56 }
```



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'AWS-SCRIPT-LABS'. Under 'SCRIPT-2', the file 'ec2.json' is selected. The main editor window shows the content of 'ec2.json', which is an AWS CloudFormation template for creating an EC2 instance. The terminal at the bottom shows the command 'ec2json > ...'. The status bar at the bottom indicates 'Ln 25, Col 2', 'Spaces: 2', 'UTF-8', 'CRLF', 'JSON', 'Go Live', 'Prettier', and the time '11:09 PM 01-Feb-23'.

```
1 {
2   "AWSTemplateFormatVersion": "2010-09-09",
3   "Description": "Template Creates a single EC2 instance with a single ENI which has multiple private and public IPs",
4   "Resources": {
5     "EC2Instance": {
6       "Type": "AWS::EC2::Instance",
7       "Properties": {
8         "InstanceType": "t2.micro",
9         "SecurityGroupIds": ["sg-0d3c8c5e869b1a8ac"],
10        "SubnetId": "subnet-0e182873b02557945",
11        "ImageId": "ami-9aa7d40ee05b09a9"
12      }
13    },
14    "mysb": {
15      "Type": "AWS::EC2::Subnet",
16      "Properties": {
17        "CidrBlock": "10.0.0.0/24",
18        "VpcId": "vpc-0d3c8c5e869b1a8ac",
19        "Tags": [
20          {
21            "Key": "Name", "Value": "Subnet"
22          },
23          {
24            "Key": "Tag", "Value": "Subnet"
25          }
26        ]
27      }
28    }
29  },
30  "Outputs": {
31    "mysb": {
32      "Description": "The ID of the sb",
33      "Value": { "Ref": "EC2Instance" },
34      "Export": {
35        "Name": { "Fn::Sub": "${AWS::StackName}-mysb" }
36      }
37    }
38  }
39 }
```