Bonus:  
EC2-Instance with user data using CloudFormation:

Use this code :

Graphical user interface, application

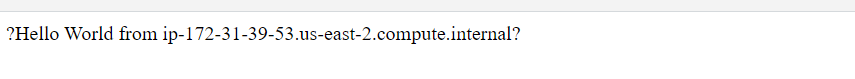
Description automatically generated

Graphical user interface, application, Word

Description automatically generated

EC2-Creted Successfully.

Open the public IPV4 address of EC2



Use this code :

{

"Description": "Template to Create an EC2 instance in a VPC",

"Parameters": {

"VpcId": {

"Type": "String",

"Description": "VPC id",

"Default": "vpc-0c629755aa87e1d94"

},

"ImageId": {

"Type": "String",

"Description": "Linux 2 AMI",

"Default": "ami-02d1e544b84bf7502 "

},

"InstanceType": {

"Type": "String",

"Description": "Choosing t2 micro because it is free",

"Default": "t2.micro"

},

"KeyName": {

"Description": "SSH Keypair to login to the instance",

"Type": "AWS::EC2::KeyPair::KeyName",

"Default": "june9keypair"

}

},

"Resources": {

"DemoInstance": {

"Type": "AWS::EC2::Instance",

"Properties": {

"ImageId": {

"Ref": "ImageId"

},

"InstanceType": {

"Ref": "InstanceType"

},

"KeyName": {

"Ref": "KeyName"

},

"SecurityGroupIds": [

{

"Ref": "DemoSecurityGroup"

}

],

"UserData": {

"Fn::Base64": {

"Fn::Sub": "#!/bin/bash\nyum update -y\nyum install -y httpd.x86\_64\nsystemctl start httpd.service\nsystemctl enable httpd.service\necho ?Hello World from $(hostname -f)? > /var/www/html/index.html\n"

}

}

}

},

"DemoSecurityGroup": {

"Type": "AWS::EC2::SecurityGroup",

"Properties": {

"VpcId": {

"Ref": "VpcId"

},

"GroupDescription": "SG to allow SSH access via port 22",

"SecurityGroupIngress": [

{

"IpProtocol": "tcp",

"FromPort": "22",

"ToPort": "22",

"CidrIp": "0.0.0.0/0"

},

{

"IpProtocol": "tcp",

"FromPort": 80,

"ToPort": 80,

"CidrIp": "0.0.0.0/0"

},

{

"IpProtocol": "tcp",

"FromPort": 443,

"ToPort": 443,

"CidrIp": "0.0.0.0/0"

}

],

"Tags": [

{

"Key": "Name",

"Value": "EC2-SG"

}

]

}

}

},

"Outputs": {

"DemoInstanceId": {

"Description": "Instance Id",

"Value": {

"Ref": "DemoInstance"

}

}

}

}