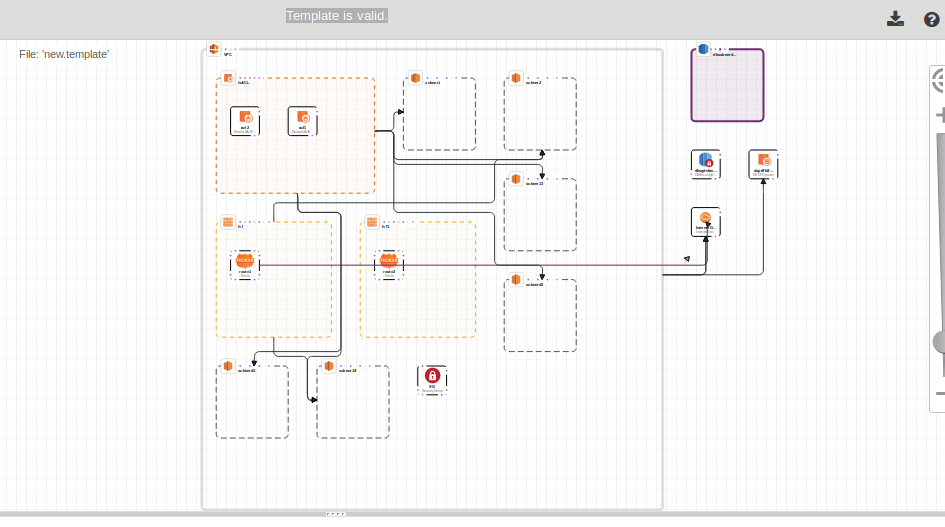
**Cloud Formation Exercise**

**1. Create a template that**

* **Creates a VPC, Route tables, 6 Subnets (2 each for public, private and database)**
* **Launches a bastion server in public subnet with a public ip attached and corresponding security group to allow access to only port 22 from internet.**
* **Launches an ALB in public subnet, with its own seperate security group, listener etc.**
* **Export the VPC id, bastion sever security group id, subnet ids, ALB listener ARN and other things required by next question.**



{

"AWSTemplateFormatVersion": "2010-09-09",

"Resources": {

"VPC": {

"Type": "AWS::EC2::VPC",

"Properties": {

"CidrBlock": "172.16.0.0/23",

"InstanceTenancy": "default",

"EnableDnsSupport": "true",

"EnableDnsHostnames": "true",

"Tags": [

{

"Key": "Name",

"Value": "CloudFormation\_VPC"

}

]

}

},

"subnet1": {

"Type": "AWS::EC2::Subnet",

"Properties": {

"CidrBlock": "172.16.0.0/24",

"AvailabilityZone": "us-east-1a",

"VpcId": {

"Ref": "VPC"

},

"Tags": [

{

"Key": "Name",

"Value": "Public Subnet1"

}

]

}

},

"subnet2": {

"Type": "AWS::EC2::Subnet",

"Properties": {

"CidrBlock": "172.16.2.0/24",

"AvailabilityZone": "us-east-1b",

"VpcId": {

"Ref": "VPC"

},

"Tags": [

{

"Key": "Name",

"Value": "Public subnet 2"

}

]

}

},

"subnet13": {

"Type": "AWS::EC2::Subnet",

"Properties": {

"CidrBlock": "176.16.1.0/24",

"AvailabilityZone": "us-east-1b",

"VpcId": {

"Ref": "VPC"

},

"Tags": [

{

"Key": "Name",

"Value": "Private subnet1"

}

]

}

},

"subnetd2": {

"Type": "AWS::EC2::Subnet",

"Properties": {

"CidrBlock": "172.16.5.0/24",

"AvailabilityZone": "us-east-1a",

"VpcId": {

"Ref": "VPC"

},

"Tags": [

{

"Key": "Name",

"Value": "database subnet 2"

}

]

}

},

"subnetd1": {

"Type": "AWS::EC2::Subnet",

"Properties": {

"CidrBlock": "172.16.4.0/24",

"AvailabilityZone": "us-east-1b",

"VpcId": {

"Ref": "VPC"

},

"Tags": [

{

"Key": "Name",

"Value": "database subnet 1"

}

]

}

},

"subnet14": {

"Type": "AWS::EC2::Subnet",

"Properties": {

"CidrBlock": "172.16.3.0/24",

"AvailabilityZone": "us-east-1a",

"VpcId": {

"Ref": "VPC"

},

"Tags": [

{

"Key": "Name",

"Value": "Private subnet2"

}

]

}

},

"InternetGateway": {

"Type": "AWS::EC2::InternetGateway",

"Properties": {

"Tags": [

{

"Key": "Application",

"Value": "Internet Gateway Created"

}

]

}

},

"AttachGateway": {

"Type": "AWS::EC2::VPCGatewayAttachment",

"Properties": {

"VpcId": {

"Ref": "VPC"

},

"InternetGatewayId": {

"Ref": "InternetGateway"

}

}

},

"doptfffdf697": {

"Type": "AWS::EC2::DHCPOptions",

"Properties": {

"DomainName": "ap-east-1.compute.internal",

"DomainNameServers": [

"AmazonProvidedDNS"

]

}

},

"NACL": {

"Type": "AWS::EC2::NetworkAcl",

"Properties": {

"VpcId": {

"Ref": "VPC"

}

}

},

"RT": {

"Type": "AWS::EC2::RouteTable",

"Properties": {

"VpcId": {

"Ref": "VPC"

}

}

},

"RT1": {

"Type": "AWS::EC2::RouteTable",

"Properties": {

"VpcId": {

"Ref": "VPC"

}

}

},

"dbsubnetdefault": {

"Type": "AWS::RDS::DBSubnetGroup",

"Properties": {

"DBSubnetGroupDescription": "default",

"SubnetIds": [

"subnet-6c4a1f04",

"subnet-de0db692"

]

}

},

"SG": {

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

"Properties": {

"GroupDescription": "Enable HTTPS access via port 443",

"VpcId": "VPC"

}

},

"dbsgdefault": {

"Type": "AWS::RDS::DBSecurityGroup",

"Properties": {

"GroupDescription": "default"

}

},

"acl1": {

"Type": "AWS::EC2::NetworkAclEntry",

"Properties": {

"CidrBlock": "0.0.0.0/0",

"Egress": "true",

"Protocol": "-1",

"RuleAction": "allow",

"RuleNumber": "100",

"NetworkAclId": {

"Ref": "NACL"

}

}

},

"acl2": {

"Type": "AWS::EC2::NetworkAclEntry",

"Properties": {

"CidrBlock": "0.0.0.0/0",

"Protocol": "-1",

"RuleAction": "allow",

"RuleNumber": "100",

"NetworkAclId": {

"Ref": "NACL"

}

}

},

"subnetacl1": {

"Type": "AWS::EC2::SubnetNetworkAclAssociation",

"Properties": {

"NetworkAclId": {

"Ref": "NACL"

},

"SubnetId": {

"Ref": "subnetd1"

}

}

},

"subnetacl2": {

"Type": "AWS::EC2::SubnetNetworkAclAssociation",

"Properties": {

"NetworkAclId": {

"Ref": "NACL"

},

"SubnetId": {

"Ref": "subnet2"

}

}

},

"subnetacl3": {

"Type": "AWS::EC2::SubnetNetworkAclAssociation",

"Properties": {

"NetworkAclId": {

"Ref": "NACL"

},

"SubnetId": {

"Ref": "subnetd2"

}

}

},

"subnetacl4": {

"Type": "AWS::EC2::SubnetNetworkAclAssociation",

"Properties": {

"NetworkAclId": {

"Ref": "NACL"

},

"SubnetId": {

"Ref": "subnet14"

}

}

},

"subnetacl5": {

"Type": "AWS::EC2::SubnetNetworkAclAssociation",

"Properties": {

"NetworkAclId": {

"Ref": "NACL"

},

"SubnetId": {

"Ref": "subnet1"

}

}

},

"subnetacl6": {

"Type": "AWS::EC2::SubnetNetworkAclAssociation",

"Properties": {

"NetworkAclId": {

"Ref": "NACL"

},

"SubnetId": {

"Ref": "subnet13"

}

}

},

"gw1": {

"Type": "AWS::EC2::VPCGatewayAttachment",

"Properties": {

"VpcId": {

"Ref": "VPC"

},

"InternetGatewayId": {

"Ref": "InternetGateway"

}

}

},

"subnetroute1": {

"Type": "AWS::EC2::SubnetRouteTableAssociation",

"Properties": {

"RouteTableId": {

"Ref": "RT"

},

"SubnetId": {

"Ref": "subnet2"

}

}

},

"subnetroute2": {

"Type": "AWS::EC2::SubnetRouteTableAssociation",

"Properties": {

"RouteTableId": {

"Ref": "RT"

},

"SubnetId": {

"Ref": "subnet14"

}

}

},

"route1": {

"Type": "AWS::EC2::Route",

"Properties": {

"DestinationCidrBlock": "0.0.0.0/0",

"RouteTableId": {

"Ref": "RT"

},

"GatewayId": {

"Ref": "InternetGateway"

}

},

"DependsOn": "gw1"

},

"route2": {

"Type": "AWS::EC2::Route",

"Properties": {

"DestinationCidrBlock": "0.0.0.0/0",

"RouteTableId": {

"Ref": "RT1"

},

"InstanceId": "01234aditya"

}

},

"dchpassoc1": {

"Type": "AWS::EC2::VPCDHCPOptionsAssociation",

"Properties": {

"VpcId": {

"Ref": "VPC"

},

"DhcpOptionsId": {

"Ref": "doptfffdf697"

}

}

},

"ingress1": {

"Type": "AWS::EC2::SecurityGroupIngress",

"Properties": {

"GroupId": {

"Ref": "SG"

},

"IpProtocol": "tcp",

"FromPort": "443",

"ToPort": "443",

"CidrIp": "0.0.0.0/0"

}

},

"egress1": {

"Type": "AWS::EC2::SecurityGroupEgress",

"Properties": {

"GroupId": {

"Ref": "SG"

},

"IpProtocol": "-1",

"CidrIp": "0.0.0.0/0"

}

}

},

"Description": ""

}