Barracuda Load Balancer ADC CloudFormation Template (CFT)

https://campus.barracuda.com/doc/51188861/

{

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

"Description" : "Barracuda Load Balancer ADC - Sample CFT showing how to launch two instances in Active/Passive HA pair",

"Metadata" : {

"AWS::CloudFormation::Interface": {

"ParameterGroups" : [

{

"Label" : { "default" : "Network Configuration" },

"Parameters" : ["VpcId", "SubnetID", "ADCAdditionalPort"]

},

{

"Label" : { "default":"Amazon EC2 Configuration" },

"Parameters" : ["InstanceType", "ConfigureHA", "AssignElasticIp"]

},

{

"Label" : { "default":"Barracuda ADC BootStrap configuration" },

"Parameters" : ["ADCServiceName", "ADCServiceType", "ADCServicePort",

"ADCHTTPRedirectPort", "ADCInstantSSLDomain", "ADCServiceNetmask", "ADCServers"]

}

],

"ParameterLabels" : {

"VpcId" : { "default" : "Which VPC should this be deployed to?" },

"SubnetID" : { "default" : "Select the subnet of the VPC where you want to create the instance" },

"InstanceType" : { "default" : "Instance Type" },

"AssignElasticIp" : { "default" : "Assign Elastic IP ?" },

"ConfigureHA" : { "default" : "Configure instances in High Availability Mode ?" },

"ADCServiceName" : { "default" : "Service Name" },

"ADCServiceType" : { "default" : "Service Type" },

"ADCServicePort" : { "default" : "Service Port" },

"ADCAdditionalPort" : { "default" : "Additional Port" },

"ADCHTTPRedirectPort" : { "default" : "HTTP Redirect Port" },

"ADCInstantSSLDomain" : { "default" : "Secure Site Domain" },

"ADCServiceNetmask" : { "default" : "Service Netmask" },

"ADCServers" : { "default" : "Servers" }

}

}

```
},
"Parameters" : {
   "VpcId": {
      "Description": "Select the VPC chosen for this deployment",
      "Type": "AWS::EC2::VPC::Id"
   },
   "SubnetID": {
      "ConstraintDescription": "Enter valid Subnet Id's associated to the VPC (subnet-*)",
      "Type": "AWS::EC2::Subnet::Id",
      "Description": "Select subnet id which has been already assigned to the VPC used."
   },
   "InstanceType": {
      "Default": "m3.medium",
      "ConstraintDescription": "Choose from the following EC2 instance types: T2, M3, M4, C4",
      "Type": "String",
      "Description": "Choose the instance type to use for this deployment",
      "AllowedValues": [
         "m3.medium",
         "m3.large",
         "m3.xlarge",
```

```
            "m3.2xlarge",
            "m4.large",
            "m4.xlarge"
        ]
    },
    "AssignElasticIp": {
        "Description": "Associate Elastic Ip for accessing management interfaces and service that will be configured",
        "Type": "String",
        "Default" : "No",
        "AllowedValues": [
            "Yes",
            "No"
        ]
    },
    "ConfigureHA": {
        "Description": "Configure instances in Active/Passive HA pair",
        "Type": "String",
        "Default" : "No",
        "AllowedValues": [
            "Yes",
```

```
            "No"
        ]
    },
    "ADCServiceName": {
        "Description": "Specify the Service Name to be configured on the Barracuda ADC",
        "AllowedPattern": "[0-9a-zA-Z-_]*",
        "MinLength": "2",
        "MaxLength": "64",
        "Type": "String"
    },
    "ADCServiceType": {
        "Description": "Specify the Service Type to be configured on the Barracuda ADC",
        "Type": "String",
        "Default" : "HTTP",
        "AllowedValues": [
            "Layer-4-TCP",
            "Barracuda-Web-Filter",
            "TCP-Proxy",
            "Secure-TCP-Proxy",
            "HTTP",
```

```
            "HTTPS",
            "Instant-SSL",
            "FTP",
            "FTP-SSL",
            "Layer-7-RDP",
            "Layer-4-UDP",
            "UDP-Proxy"
          ]
      },
      "ADCServicePort": {
          "Description": "Specify the Service Port to be configured on the Barracuda ADC. This port is exposed to the outside world. Default is 80.",
          "Default": "80",
          "ConstraintDescription": "Must be a valid port number (1-65535).",
          "Type": "Number",
          "MaxValue": "65535",
          "MinValue": "1"
      },
      "ADCAdditionalPort": {
          "Description": "(OPTIONAL) Specify any additional port to be opened in security group for dataplane interface.  Default value -1 means no additional port will be opened. This CFT by default will open 'Service Port' in security group for data plane interface. The following ports will be opened in security group for managament interface(eth0): 8000, 443, 8002, 22, icmp(for ping test), VRRP(112)
```

protocol. For details regarding these ports please refer to Barracuda ADC AWS deployment techlib",

"Default": "-1",

"ConstraintDescription": "Must be a valid port number (1-65535).",

"Type": "Number",

"MaxValue": "65535",

"MinValue": "-1"

},

"ADCHTTPRedirectPort": {

"Description": "(OPTIONAL) Specify the HTTP redirect port for an Instant SSL service. Default is 80",

"Default": "80",

"ConstraintDescription": "Must be a valid port number (1-65535).",

"Type": "Number",

"MaxValue": "65535",

"MinValue": "1"

},

"ADCInstantSSLDomain": {

"Description": "(OPTIONAL) Specify the secure side domain for an Instant SSL service. To include all domains, enter an asterisk (*). ",

"Default": "*",

"ConstraintDescription": "Must be a valid domain as per the certificate. Use ADC Management

UI to upload certificate",

"Type": "String"

},

"ADCServiceNetmask": {

"Description": "The netmask for the service.",

"Default": "255.255.255.0",

"Type": "String"

},

"ADCServers": {

"Description": "Specify the Server IP:Server Port combination in comma separated format e.g. 10.10.1.1:80, 10.10.2.1:80. This will be configured as backend servers on the Barracuda ADC. Alternatively, you can also enter the FQDN of the instance or a downstream ELB to connect to.",

"ConstraintDescription": "Must be a valid IP address or FQDN and Port separated by colon(:) in csv format",

"Type": "String"

}

},

"Mappings": {

"RegionMap": {

"us-east-1": {

"ImageID": "NOT_SUPPORTED"

},

```
"us-west-1": {
   "ImageID": "NOT_SUPPORTED"
},
"us-west-2": {
   "ImageID": "ami-d8b577b8"
},
"sa-east-1": {
   "ImageID": "NOT_SUPPORTED"
},
"eu-central-1": {
   "ImageID": "NOT_SUPPORTED"
},
"eu-west-1": {
   "ImageID": "NOT_SUPPORTED"
},
"ap-southeast-1": {
   "ImageID": "NOT_SUPPORTED"
},
"ap-southeast-2": {
   "ImageID": "NOT_SUPPORTED"
```

```
    },
    "ap-northeast-1": {
       "ImageID": "NOT_SUPPORTED"
    },
    "ap-northeast-2": {
       "ImageID": "NOT_SUPPORTED"
    }
  },
  "ServiceTypeMap": {
    "Layer-4-TCP": {
       "ServiceType" : "L4"
    },
    "Barracuda-Web-Filter": {
       "ServiceType" : "INLINE"
    },
    "TCP-Proxy": {
       "ServiceType" : "L7Tcp"
    },
    "Secure-TCP-Proxy": {
       "ServiceType" : "SSL"
```

```
},
"HTTP": {
   "ServiceType" : "HTTP"
},
"HTTPS": {
   "ServiceType" : "HTTPS"
},
"Instant-SSL": {
   "ServiceType" : "INSTANTSSL"
},
"FTP": {
   "ServiceType" : "FTP"
},
"FTP-SSL": {
   "ServiceType" : "FTPSSL"
},
"Layer-7-RDP": {
   "ServiceType" : "RDP"
},
"Layer-4-UDP": {
```

```
            "ServiceType" : "UDP"
        },
        "UDP-Proxy": {
            "ServiceType" : "L7UDP"
        }
    }
},
"Conditions" : {
        "AttachElasticIp" : {"Fn::Equals" : [{"Ref" : "AssignElasticIp"}, "Yes"]},
        "HAPair" : {"Fn::Equals" : [{"Ref" : "ConfigureHA"}, "Yes"]},
        "HAwithElasticIp" : {
                        "Fn::And": [
                            {"Fn::Equals" : [{"Ref" : "AssignElasticIp"}, "Yes"]},
                            {"Fn::Equals" : [{"Ref" : "ConfigureHA"}, "Yes"]}
                        ]
        },
        "OpenAdditionalPort" : {
            "Fn::Not" : [{
                "Fn::Equals" : [
                    {"Ref" : "ADCAdditionalPort"},
```

```
                -1
            ]
        }]
    },
    "ServiceTypeUDP" : {
                "Fn::Or": [
                    {"Fn::Equals" : [{"Ref" : "ADCServiceType"}, "Layer-4-UDP"]},
                    {"Fn::Equals" : [{"Ref" : "ADCServiceType"}, "UDP-Proxy"]}
                ]
    }
},
"Resources": {
  "HARole": {
    "Type": "AWS::IAM::Role",
    "Condition" : "HAPair",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [ {
          "Effect": "Allow",
          "Principal": {
```

```
            "Service": [ "ec2.amazonaws.com" ]
         },
         "Action": [ "sts:AssumeRole" ]
      } ]
   },
   "Path": "/",
   "Policies": [ {
      "PolicyName": "HA_Takeover",
      "PolicyDocument": {
         "Statement": [ {
            "Effect": "Allow",
            "Action": [
               "ec2:AssignPrivateIpAddresses",
               "ec2:DescribeInstances",
               "ec2:DetachNetworkInterface",
               "ec2:AttachNetworkInterface"
            ],
            "Resource": "*"
         } ]
      }
```

```
        } ]
      }
    },
    "HARoleProfile": {
      "Type": "AWS::IAM::InstanceProfile",
      "Condition" : "HAPair",
      "Properties": {
          "Path": "/",
          "Roles": [ {
            "Ref": "HARole"
          } ]
      }
    },
    "mgmtENISG": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Security Group for MGMT ENI",
        "VpcId": { "Ref": "VpcId" },
        "SecurityGroupIngress": [
          { "IpProtocol": "tcp", "FromPort": "443", "ToPort": "443", "CidrIp": "0.0.0.0/0" },
```

```
            { "IpProtocol": "tcp", "FromPort": "8000", "ToPort": "8000", "CidrIp": "0.0.0.0/0" },
            { "IpProtocol": "tcp", "FromPort": "8002", "ToPort": "8002", "CidrIp": "0.0.0.0/0" },
            { "IpProtocol": "tcp", "FromPort": "22", "ToPort": "22", "CidrIp": "0.0.0.0/0" },
            { "IpProtocol": "icmp", "FromPort": "8", "ToPort": "-1", "CidrIp": "0.0.0.0/0" },
            { "IpProtocol": 112, "FromPort": "0", "ToPort": "-1", "CidrIp": "0.0.0.0/0" }
        ]
    }
},
"dpENISG": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Security Group for Data Plane ENI",
        "VpcId": { "Ref": "VpcId" },
        "SecurityGroupIngress": [
            { "IpProtocol": { "Fn::If" : [ "ServiceTypeUDP", "udp", "tcp" ] },
              "FromPort": {"Ref": "ADCServicePort"},
              "ToPort": {"Ref": "ADCServicePort"}, "CidrIp": "0.0.0.0/0"
            },
            {
                "Fn::If" : [
```

```
                    "OpenAdditionalPort",
                      { "IpProtocol":  { "Fn::If" : [ "ServiceTypeUDP", "udp", "tcp" ] } ,
                        "FromPort": {"Ref": "ADCAdditionalPort"},
                        "ToPort": {"Ref": "ADCAdditionalPort"}, "CidrIp": "0.0.0.0/0"
                      },
                      {"Ref" : "AWS::NoValue"}
                ]
            }
        ]
    }
},
"dpENI": {
    "Type": "AWS::EC2::NetworkInterface",
        "Properties": {
            "SubnetId": { "Ref": "SubnetID" },
            "Description": "Dataplane Interface (ge-1-1)",
            "GroupSet": [ { "Ref": "dpENISG" } ],
            "SecondaryPrivateIpAddressCount" : 1,
            "SourceDestCheck": "false"
    }
```

```
    },
    "dpEIP" :  {
       "Type" : "AWS::EC2::EIP",
       "Condition" : "AttachElasticIp",
       "Properties" : {
          "Domain" : "vpc"
       }
    },
    "dpEIPAssoc" : {
       "Type" : "AWS::EC2::EIPAssociation",
       "Condition" : "AttachElasticIp",
       "Properties" : {
          "NetworkInterfaceId" : { "Ref" : "dpENI" },
          "AllocationId" : { "Fn::GetAtt" : ["dpEIP", "AllocationId"] },
          "PrivateIpAddress" : { "Fn::Select" : ["0", { "Fn::GetAtt" : ["dpENI",
"SecondaryPrivateIpAddresses"] } ] }
       }
    },
    "mgmtENI": {
       "Type": "AWS::EC2::NetworkInterface",
       "Properties": {
```

```
        "SubnetId": { "Ref" : "SubnetID" },
        "Description": "Management Interface (eth0)",
        "GroupSet": [ { "Ref": "mgmtENISG" } ],
        "SourceDestCheck": "true"
      }
    },
    "mgmtEIP" :  {
      "Type" : "AWS::EC2::EIP",
      "Condition" : "AttachElasticIp",
      "Properties" : {
        "Domain" : "vpc"
      }
    },
    "mgmtEIPAssoc" : {
      "Type" : "AWS::EC2::EIPAssociation",
      "Condition" : "AttachElasticIp",
      "Properties" : {
        "NetworkInterfaceId" : { "Ref" : "mgmtENI" },
        "AllocationId" : { "Fn::GetAtt" : ["mgmtEIP", "AllocationId"] },
        "PrivateIpAddress" : { "Fn::GetAtt" : ["mgmtENI", "PrimaryPrivateIpAddress" ] }
```

```
        }
      },
      "AdcInstance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
          "ImageId": { "Fn::FindInMap": [ "RegionMap", { "Ref": "AWS::Region" }, "ImageID" ] }  ,
          "InstanceType": { "Ref": "InstanceType" },
          "IamInstanceProfile" : { "Fn::If" : [ "HAPair" , {"Ref" : "HARoleProfile"}, {"Ref" :
"AWS::NoValue"} ] },
          "Tags": [{ "Key" : "Name", "Value" : "ADC-1-CFT"}],
          "NetworkInterfaces": [
            { "NetworkInterfaceId": { "Ref": "mgmtENI" }, "DeviceIndex": "0" },
            { "NetworkInterfaceId": { "Ref": "dpENI" }, "DeviceIndex": "1" }
          ],
          "UserData": { "Fn::Base64" : { "Fn::Join" : ["", [
            "#!/bin/bash\n",
            "/opt/aws/bwaf/aws_bootstrap.pl ",
            "    --command init-config ",
            "    --clustered ", { "Ref" : "ConfigureHA" },
            "    --service_type ", { "Fn::FindInMap": [ "ServiceTypeMap", { "Ref": "ADCServiceType"
}, "ServiceType" ] },
```

```
                "    --service_name ", { "Ref" : "ADCServiceName" },

                "    --service_ip ", { "Fn::Join" : [" ", { "Fn::GetAtt" : ["dpENI",
"SecondaryPrivateIpAddresses"] } ] },

                "    --service_port ", { "Ref" : "ADCServicePort" },

                "    --interface ge-1-1 ",

                "    --http_redirect_port ", { "Ref" : "ADCHTTPRedirectPort" },

                "    --domain ", "'", { "Ref" : "ADCInstantSSLDomain" }, "'",

                "    --service_netmask ", { "Ref" : "ADCServiceNetmask" },

                "    --servers ", { "Ref" : "ADCServers" }, "\n"

            ]]}}

        }

    },

    "dpENI2": {

        "Type": "AWS::EC2::NetworkInterface",

        "Condition" : "HAPair",

            "Properties": {

                "SubnetId": { "Ref": "SubnetID" },

                "Description": "Dataplane Interface (ge-1-1)",

                "GroupSet": [ { "Ref": "dpENISG" } ],

                "SourceDestCheck": "false"

        }
```

```
},
"mgmtENI2": {
   "Type": "AWS::EC2::NetworkInterface",
   "Condition" : "HAPair",
   "Properties": {
      "SubnetId": { "Ref" : "SubnetID" },
      "Description": "Management Interface (eth0)",
      "GroupSet": [ { "Ref": "mgmtENISG" } ],
      "SourceDestCheck": "true"
   }
},
"mgmtEIP2" :  {
   "Type" : "AWS::EC2::EIP",
   "Condition" : "HAwithElasticIp",
   "Properties" : {
      "Domain" : "vpc"
   }
},
"mgmtEIPAssoc2" : {
   "Type" : "AWS::EC2::EIPAssociation",
```

```
        "Condition" : "HAwithElasticIp",
        "Properties" : {
           "NetworkInterfaceId" : { "Ref" : "mgmtENI2" },
           "AllocationId" : { "Fn::GetAtt" : ["mgmtEIP2", "AllocationId"] },
           "PrivateIpAddress" : { "Fn::GetAtt" : ["mgmtENI2", "PrimaryPrivateIpAddress" ] }
        }
     },
     "AdcInstance2": {
        "Type": "AWS::EC2::Instance",
        "Condition" : "HAPair",
        "DependsOn" : "AdcInstance",
        "Properties": {
           "ImageId": { "Fn::FindInMap": [ "RegionMap", { "Ref": "AWS::Region" }, "ImageID" ] }  ,
           "InstanceType": { "Ref": "InstanceType" },
           "IamInstanceProfile" : { "Fn::If" : [ "HAPair" , {"Ref" : "HARoleProfile"}, {"Ref" :
"AWS::NoValue"} ] },
           "Tags": [{ "Key" : "Name", "Value" : "ADC-2-CFT"}],
           "NetworkInterfaces": [
              { "NetworkInterfaceId": { "Ref": "mgmtENI2" }, "DeviceIndex": "0" },
              { "NetworkInterfaceId": { "Ref": "dpENI2" }, "DeviceIndex": "1" }
           ],
```

```
            "UserData": { "Fn::Base64" : { "Fn::Join" : ["", [
                "#!/bin/bash\n",
                "/opt/aws/bwaf/aws_bootstrap.pl ",
                "    --command init-config ",
                "    --peer_node ", { "Fn::GetAtt" : ["mgmtENI", "PrimaryPrivateIpAddress" ] },
                "    --clustered ", { "Ref" : "ConfigureHA" }, "\n"
            ]]}}
        }
    }
},
"Outputs" : {
"InstanceId" : {
    "Value" : { "Ref" : "AdcInstance" },
    "Description" : "Instance ID of ADC"
},
"InstanceIdSecondary" : {
    "Value" : { "Ref" : "AdcInstance2" },
    "Condition" : "HAPair",
    "Description" : "Instance ID of ADC(Secondary)"
},
```

```
  "ManagementURL" : {
     "Condition" : "AttachElasticIp",
     "Value" : { "Fn::Join" : ["", [ "http://", { "Ref" : "mgmtEIP" }, ":8000"]]},
     "Description" : "URL for accessing ADC management GUI"
  },
  "ManagementURLSecondary" : {
     "Condition" : "HAwithElasticIp",
     "Value" : { "Fn::Join" : ["", [ "http://", { "Ref" : "mgmtEIP2" }, ":8000"]]},
     "Description" : "URL for accessing ADC management GUI(Secondary)"
  },
  "ServiceIPPort" : {
     "Condition" : "AttachElasticIp",
     "Value" : { "Fn::Join" : ["", [ { "Ref": "dpEIP" }, ":", {"Ref" : "ADCServicePort"}]]},
     "Description" : "Service IP and Port for accessing the virtual service"
  }
 }
}
```

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.