



## Sample deployment templates

---

This appendix contains the following topics:



---

**Note** The templates in this appendix are samples for your reference and do not contain real values.

---

- [Sample manifest template for VMware vCenter, on page 1](#)
- [Set seed node explicitly, on page 3](#)
- [Sample CloudFormation template for installing Crosswork Cluster VMs on AWS EC2, on page 3](#)
- [Sample CloudFormation Template for installing Crosswork Data Gateway on EC2, on page 18](#)

## Sample manifest template for VMware vCenter

The following example deploys a Crosswork cluster containing 3 Hybrid nodes and 2 worker nodes.



---

**Note** In case you are using resource pools, please note that individual ESXi host targeting is not allowed and vCenter is responsible for assigning the VM to a host in the resource pool. If vCenter is not configured with resource pools, then the exact ESXi host path must be passed.

---

```
*****
vCenter Example
*****

ClusterIPStack = "IPv4"
ManagementVIP = "172.25.87.94"
ManagementIPNetmask = "255.255.255.192"
ManagementIPGateway = "172.25.87.65"
DataVIP = "192.168.123.94"
DataIPNetmask = "255.255.255.0"
DataIPGateway = "0.0.0.0"
DNS = "171.70.168.183"
DomainName = "cisco.com"
CWPASSWORD = "****"
VMSize = "Large"
NTP = "ntp.cisco.com"
CloneTimeOut = 90
ManagerDataFsSize = 450
ThinProvisioned = true
```

```

BackupMinPercent = 50
EnableHardReservations = false
ManagerDataFsSize = 450
WorkerDataFsSize = 450

CwVMs = {
  "0" = {
    VMName = "vm0",
    ManagementIPAddress = "172.25.87.82",
    DataIPAddress = "0.0.0.0",
    DataIPAddress = "192.168.123.82",
    NodeType = "Hybrid"
  },
  "1" = {
    VMName = "vm1",
    ManagementIPAddress = "172.25.87.83",
    DataIPAddress = "0.0.0.0",
    DataIPAddress = "192.168.123.83",
    NodeType = "Hybrid"
  },
  "2" = {
    VMName = "vm2",
    ManagementIPAddress = "172.25.87.84",
    DataIPAddress = "0.0.0.0",
    DataIPAddress = "192.168.123.84",
    NodeType = "Hybrid"
  },
  "3" = {
    VMName = "vmworker",
    ManagementIPAddress = "172.25.87.85",
    DataIPAddress = "0.0.0.0",
    DataIPAddress = "192.168.123.84",
    NodeType = "Worker"
  },
  "4" = {
    VMName = "vmworker2",
    ManagementIPAddress = "172.25.87.86",
    DataIPAddress = "0.0.0.0",
    DataIPAddress = "192.168.123.86",
    NodeType = "Worker"
  },
}

/***** vCentre Resource Data with Cw VM assignment *****/

VCentreDC = {
  VCentreAddress = "172.25.87.90",
  VCentreUser = administrator@vsphere.local,
  VCentrePassword = "*****",
  DCname = "dc-cr",
  MgmtNetworkName = "VM Network",
  DataNetworkName = "DPortGroup10",
  VMs = [
    {
      HostedCwVMs = [
        "0",
        "1",
        "2",
        "3", "4"
      ]
    }
  ],
}

```

```

    Host = "172.25.87.93",
    Datastore = "datastore3"
  HSDatastore = "datastore3",
},]
}

```

## Set seed node explicitly

The cluster installer tool, by default, selects the first VM (VM 0) as the seed node. You can set the seed node explicitly by adding the following section to the manifest template (.tfvars file) indicating the unique key of the seed node.



**Note** You are recommended not to modify the default seed node value unless advised to do so by the Cisco Customer Experience team.

```

cluster_settings = {
#Default Minimum number of nodes in inventory
  min_inventory = 3
#Default Max number of nodes in inventory
  max_inventory = 9
#Default Min number of manager nodes
  min_mgr_nodes = 2
#Default Max number of manager nodes
  max_mgr_nodes = 3
#Default seed node key name
  default_seed_node = "0"
}

```

## Sample CloudFormation template for installing Crosswork Cluster VMs on AWS EC2



**Attention** The following CF template (.yaml file) contains the details to install a Crosswork cluster with 3 VMs. Please note that it is only a sample, and you can always create a different CF template according to your production preferences and execute it as per the steps mentioned in this section. This document assumes that you are familiar with AWS and the CloudFormation concepts, and as such, the CF template creation is out of the scope of this document.

```

Description: "Sample CF Template for deploying Cisco Crosswork cluster VMs, with single hybrid, on EC2"

```

```

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:
          default: "Cw Network Configuration"
        Parameters:
          - VpcId
          - SecGroup

```

```

- CwSSHPassword
- CwAmiId
- CwMgmtSubnetId
- CwMgmtSubnetNetmask
- CwMgmtSubnetGateway
- CwMgmtVIP
- InterfaceDeploymentMode
- CwDataSubnetId
- CwDataSubnetNetmask
- CwDataSubnetGateway
- CwDataVIP
- Label:
  default: "Cw VM customization"
  Parameters:
    - InstanceType
    - DataDiskSize
    - K8sServiceNetwork
    - K8sPodNetwork
- Label:
  default: "OPTIONAL - VM IP addressing"
  Parameters:
    - Cw1MgmtIP
    - Cw1DataIP
    - Cw2MgmtIP
    - Cw2DataIP
    - Cw3MgmtIP
    - Cw3DataIP

Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
    ConstraintDescription: Must be the VPC Id of an existing Virtual Private Cloud.

  SecGroup:
    Type: AWS::EC2::SecurityGroup::Id
    Description: Pre-created security group to be applied. Must allow ingress access for
ports 22, 30160:31560

  CwMgmtSubnetId:
    Type: AWS::EC2::Subnet::Id
    Description: Select the management subnet for the Crosswork VMs

  CwMgmtSubnetNetmask:
    Type: String
    Description: Enter the management subnet netmask in dotted decimal form, eg 255.255.255.0

    Default: "255.255.255.0"
    AllowedPattern: (\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}

  CwMgmtSubnetGateway:
    Type: String
    Description: Enter the management default gateway on the selected management subnet.
This is typically the first address on the subnet.
    AllowedPattern: (\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}

  CwMgmtVIP:
    Type: String
    Description: OPTIONAL - Specify a free address on the management subnet to be used as
the VIP. If not specified an address will be assigned automatically.
    AllowedPattern: ((\d{1,3})\.\d{1,3})\.\d{1,3}\.\d{1,3})|^$
    Default: ""

  CwDataSubnetId:

```

```

    Type: AWS::EC2::Subnet::Id
    Description: Select the data subnet for the Crosswork VMs. In single interface deployments
select the same subnet as for the management interface.

InterfaceDeploymentMode:
  Type: String
  Description: Select 1 (Management only) or 2 (Management + Data) interface deployment
mode.
  AllowedValues:
    - 1
    - 2

CwDataSubnetNetmask:
  Type: String
  Description: Enter the data subnet netmask in dotted decimal form, eg 255.255.255.0.
Ignored when deploying in single interface mode.
  Default: "255.255.255.0"
  AllowedPattern: (\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3})

CwDataSubnetGateway:
  Type: String
  Description: Enter the management default gateway on the selectec data subnet. This is
typically the first address on the subnet. Ignored when deploying in single interface mode.

  AllowedPattern: (\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3})
  Default: '0.0.0.0'

CwDataVIP:
  Type: String
  Description: OPTIONAL - Specify a free address on the data subnet to be used as the
VIP. If not specified an address will be assigned automatically.
  AllowedPattern: ((\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3}))|^$
  Default: ""

CwAmiId:
  Type: AWS::EC2::Image::Id
  Description: Provide Crosswork AMI ID.

# MgmtPublicIP:
#   Type: String
#   Description: Enter your public IP. Will be use to restrict CNC SSH and UI access to
this IP only
#   Default: 0.0.0.0/0

CwSSHPassword:
  Type: String
  Description: Enter CNC SSH Password. NOTE; Use of external secret store is recommended.

  NoEcho: True

InstanceType:
  Description: Enter EC2 instance type for the node instances.Default is m5.4xlarge.
  Type: String
  AllowedValues:
    - m5.4xlarge
    - m5.8xlarge
    - m5.2xlarge
    - m5.12xlarge
    - m5d.4xlarge
    - m5d.8xlarge
    - m5d.2xlarge
    - m5d.12xlarge
    - m5n.4xlarge
    - m5n.8xlarge

```

```

- m5n.2xlarge
- m5n.12xlarge
- r5.4xlarge
- r5.8xlarge
- r5.2xlarge
- r5.12xlarge
- c5.4xlarge
- c5.8xlarge
- c5.2xlarge
- c5.12xlarge
- m5zn.2xlarge
- m5zn.3xlarge
- m5zn.4xlarge
Default: m5.4xlarge

DataDiskSize:
  Description: Cw data disk size.
  Type: Number
  MinValue: 450
  Default: 450

K8sServiceNetwork:
  Type: String
  Description: "OPTIONAL - Enter the network address for the k8s service network. The
CIDR range is fixed to '/16'."
  AllowedPattern: (\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3})
  Default: '10.96.0.0'

K8sPodNetwork:
  Type: String
  Description: "OPTIONAL - Enter the network address for the k8s pod network. The CIDR
range is fixed to '/16'."
  AllowedPattern: (\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3})
  Default: '10.244.0.0'

Cw1MgmtIP:
  Type: String
  Description: OPTIONAL - Specify a free address on the management subnet. If not specified
an address will be assigned automatically.
  AllowedPattern: ((\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3}))|^$
  Default: ""

Cw1DataIP:
  Type: String
  Description: OPTIONAL - Specify a free address on the data subnet. If not specified an
address will be assigned automatically.
  AllowedPattern: ((\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3}))|^$
  Default: ""

Cw2MgmtIP:
  Type: String
  Description: OPTIONAL - Specify a free address on the management subnet. If not specified
an address will be assigned automatically.
  AllowedPattern: ((\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3}))|^$
  Default: ""

Cw2DataIP:
  Type: String
  Description: OPTIONAL - Specify a free address on the data subnet. If not specified an
address will be assigned automatically.
  AllowedPattern: ((\d{1,3})\.\(\d{1,3})\.\(\d{1,3})\.\(\d{1,3}))|^$
  Default: ""

Cw3MgmtIP:

```

```

    Type: String
    Description: OPTIONAL - Specify a free address on the management subnet. If not specified
    an address will be assigned automatically.
    AllowedPattern: ((\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})|^$
    Default: ""

    Cw3DataIP:
    Type: String
    Description: OPTIONAL - Specify a free address on the data subnet. If not specified an
    address will be assigned automatically.
    AllowedPattern: ((\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})|^$
    Default: ""

    CwClusterPlacementStrategy:
    Type: String
    Description: Specify the EC2 instance placement strategy. Default 'cluster' ensures
    maximum throughput.
    Default: cluster
    AllowedValues:
    - cluster
    - partition
    - spread

    Conditions:
    DeployDataInterface: !Not
    - !Equals
    - !Ref InterfaceDeploymentMode
    - "1"

    SetMgmtVIP: !Not
    - !Equals
    - !Ref CwMgmtVIP
    - ""

    SetDataVIP: !Not
    - !Equals
    - !Ref CwDataVIP
    - ""

    SetCw1IP0: !Not
    - !Equals
    - !Ref Cw1MgmtIP
    - ""

    SetCw1IP1: !Not
    - !Equals
    - !Ref Cw1DataIP
    - ""

    SetCw2IP0: !Not
    - !Equals
    - !Ref Cw2MgmtIP
    - ""

    SetCw2IP1: !Not
    - !Equals
    - !Ref Cw2DataIP
    - ""

    SetCw3IP0: !Not
    - !Equals
    - !Ref Cw3MgmtIP
    - ""

```

```

SetCw3IP1: !Not
- !Equals
- !Ref Cw3DataIP
- ""

Resources:
  EC2ENIRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Policies:
        - PolicyName: eni-modification
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - ec2:DescribeNetworkInterfaces
                  - ec2:AssignPrivateIpAddresses
                  - ec2:UnassignPrivateIpAddresses
                Resource: "*"

  CwPlacementGroup:
    Type: AWS::EC2::PlacementGroup
    Properties:
      Strategy: !Sub ${CwClusterPlacementStrategy}

  CwEC2IamInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      InstanceProfileName: !Sub ${AWS::StackName}-CwEC2IamInstanceProfile
      Path: "/cw/"
      Roles:
        - !Ref EC2ENIRole

  CwInstance1MgmtInterface:
    Type: AWS::EC2::NetworkInterface
    Properties:
      Description: "VM1-Mgmt-eth0"
      GroupSet:
        #- !Ref 'SSHSecurityGroup'
        - !Ref SecGroup
      PrivateIpAddresses:
        !If
        - SetCw1IP0
        - !If
          - SetMgmtVIP
          - - Primary: false
            PrivateIpAddress: !Ref CwMgmtVIP
          - Primary: true
            PrivateIpAddress: !Ref Cw1MgmtIP
        - - Primary: true
          PrivateIpAddress: !Ref Cw1MgmtIP
        - !If
          - SetMgmtVIP
          - - Primary: false

```



```

        PrivateIpAddress: !Ref CwMgmtVIP
        - !Ref 'AWS::NoValue'
    SecondaryPrivateIpAddressCount:
    !If
    - SetMgmtVIP
    - !Ref 'AWS::NoValue'
    - !If
    - SetCw1IP0
    - !Ref 'AWS::NoValue'
    - 1
    SubnetId: !Ref CwMgmtSubnetId
    Tags:
    - Key: Name
      Value: Cw-VM1-eth0

CwInstance1DataInterface:
Type: AWS::EC2::NetworkInterface
Properties:
  Description: "VM1-Data-eth1"
  GroupSet:
    #- !Ref 'SSHSecurityGroup'
    - !Ref SecGroup
  PrivateIpAddresses:
    !If
    - SetCw1IP1
    - !If
    - SetDataVIP
    - - Primary: false
      PrivateIpAddress: !Ref CwDataVIP
    - Primary: true
      PrivateIpAddress: !Ref Cw1DataIP
    - - Primary: true
      PrivateIpAddress: !Ref Cw1DataIP
    - !If
    - SetDataVIP
    - - Primary: false
      PrivateIpAddress: !Ref CwDataVIP
    - !Ref 'AWS::NoValue'
  SecondaryPrivateIpAddressCount:
    !If
    - SetDataVIP
    - !Ref 'AWS::NoValue'
    - !If
    - SetCw1IP1
    - !Ref 'AWS::NoValue'
    - 1
  SubnetId: !Ref CwDataSubnetId
  Tags:
  - Key: Name
    Value: Cw-VM1-eth1
  Condition: DeployDataInterface

CwInstance2MgmtInterface:
Type: AWS::EC2::NetworkInterface
Properties:
  Description: "VM2-Mgmt-eth0"
  GroupSet:
    #- !Ref 'SSHSecurityGroup'
    - !Ref SecGroup
  PrivateIpAddresses:
    !If
    - SetCw2IP0
    - - Primary: true
      PrivateIpAddress: !Ref Cw2MgmtIP

```

```

    - !Ref 'AWS::NoValue'
  SubnetId: !Ref CwMgmtSubnetId
  Tags:
    - Key: Name
      Value: Cw-VM2-eth0

CwInstance2DataInterface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    Description: "VM2-Data-eth1"
    GroupSet:
      #- !Ref 'SSHSecurityGroup'
      - !Ref SecGroup
    PrivateIpAddresses:
      !If
        - SetCw2IP1
        - - Primary: true
          PrivateIpAddress: !Ref Cw2DataIP
      - !Ref 'AWS::NoValue'
    SubnetId: !Ref CwDataSubnetId
  Tags:
    - Key: Name
      Value: VM2-eth1
  Condition: DeployDataInterface

CwInstance3MgmtInterface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    Description: "VM3-Mgmt-eth0"
    GroupSet:
      #- !Ref 'SSHSecurityGroup'
      - !Ref SecGroup
    PrivateIpAddresses:
      !If
        - SetCw3IP0
        - - Primary: true
          PrivateIpAddress: !Ref Cw3MgmtIP
      - !Ref 'AWS::NoValue'
    SubnetId: !Ref CwMgmtSubnetId
  Tags:
    - Key: Name
      Value: VM3-eth0

CwInstance3DataInterface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    Description: "VM3-Data-eth1"
    GroupSet:
      #- !Ref 'SSHSecurityGroup'
      - !Ref SecGroup
    PrivateIpAddresses:
      !If
        - SetCw3IP1
        - - Primary: true
          PrivateIpAddress: !Ref Cw3DataIP
      - !Ref 'AWS::NoValue'
    SubnetId: !Ref CwDataSubnetId
  Tags:
    - Key: Name
      Value: VM3-eth1
  Condition: DeployDataInterface

# SSHSecurityGroup:
# #

```

```

http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-security-group.html
#   Type: AWS::EC2::SecurityGroup
#   Properties:
#     VpcId: !Ref "VpcId"
#     GroupDescription: Enable access to CNC VMs
#     Tags:
#       - Key: Name
#         Value: "Cw-SG-1"
#     SecurityGroupIngress:
#       # Must allow 22 and all of the service port range 30160:31560
#       #       - CidrIp: !Ref CwMgmtSubnetId
#         #       FromPort: 22
#         #       IpProtocol: tcp
#         #       ToPort: 22
#       #       - CidrIp: !Ref CwMgmtSubnetId
#         #       FromPort: 30603
#         #       IpProtocol: tcp
#         #       ToPort: 30603
#     - CidrIp: 10.0.0.0/8
#       FromPort: -1
#       IpProtocol: -1
#       ToPort: -1

#EC2 Launch Template Creation
CommonCwLaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: !Sub CommonCwLaunchTemplate-${AWS::StackName}
    LaunchTemplateData:
      InstanceType: !Ref 'InstanceType'
      ImageId: !Ref 'CwAmiId'
      IamInstanceProfile:
        Name: !Ref CwEC2IamInstanceProfile
      EbsOptimized: True
#     InstanceMarketOptions:
#     MarketType: spot
      Placement:
        GroupName: !Ref CwPlacementGroup
      BlockDeviceMappings:
        - Ebs:
            VolumeSize: 50
            VolumeType: standard
            DeleteOnTermination: True
            Encrypted: False
            #Iops: 1000
            DeviceName: /dev/sda1
        - Ebs:
            VolumeSize: 10
            DeleteOnTermination: True
            VolumeType: gp3
            DeviceName: /dev/sdc
        - Ebs:
            VolumeSize: !Ref DataDiskSize
            DeleteOnTermination: True
            VolumeType: gp3
            Iops: 6000
            DeviceName: /dev/sdd
        - Ebs:
            VolumeSize: 10
            VolumeType: gp3
            DeleteOnTermination: True
            #Iops: 6000
            DeviceName: /dev/sdm
        - Ebs:

```

```

        VolumeSize: 156
        DeleteOnTermination: True
        VolumeType: gp3
        Iops: 6000
        DeviceName: /dev/sdf
    - Ebs:
        VolumeSize: 250
        DeleteOnTermination: True
        VolumeType: gp3
        DeviceName: /dev/sdg
    MetadataOptions:
        HttpPutResponseHopLimit: 2
    PrivateDnsNameOptions:
        EnableResourceNameDnsARecord: True
    TagSpecifications:
    - ResourceType: instance
      Tags:
    - Key: cisco-bu-group
      Value: "spnaa"
    - Key: cisco-bu-owner
      Value: ""
    - Key: cisco-bu-project-name
      Value: "Crosswork"
    - Key: cisco-bu-release
      Value: "440"
    - Key: cisco-bu-role
      Value: "test"
    - Key: cisco-ops-runtime-optin
      Value: "in"
    - Key: cisco-ops-runtime-policy
      Value: "mon-fri"
    - Key: cisco-ops-timezone
      Value: "PST"
    - Key: cisco-sec-internetfacing
      Value: "false"

    CwInstancel:
    #
    http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html

    Type: AWS::EC2::Instance
    DependsOn:
    - CommonCwLaunchTemplate
    Properties:
    LaunchTemplate:
    Version: 1
    LaunchTemplateId: !Ref CommonCwLaunchTemplate
    NetworkInterfaces: !If
    - DeployDataInterface
    - - NetworkInterfaceId: !Ref CwInstancelMgmntInterface
      DeviceIndex: "0"
    - NetworkInterfaceId: !Ref CwInstancelDataInterface
      DeviceIndex: "1"
    - - NetworkInterfaceId: !Ref CwInstancelMgmntInterface
      DeviceIndex: "0"
    Tags:
    - Key: Name
      Value: Cw-EC2-VM1
    UserData: !Base64
    Fn::Join:
    - ''
    - - !Sub |
      <?xml version="1.0" encoding="UTF-8"?>
      <Environment

```

```

    <PlatformSection>
      <Kind>EC2</Kind>
    </PlatformSection>
    <PropertySection>
      <Property oe:key="CWPassword" oe:value="\${CwSSHPassword}"/>
      <Property oe:key="CWUsername" oe:value="cw-admin"/>
    - Fn::Join:
      - ""
      - - '<Property oe:key="AwsIamRole" oe:value="'
        - '!Ref EC2ENIRole'
        - '"/>'
        - "\n"
    - !Sub |
      <Property oe:key="IsSeed" oe:value="True"/>
      <Property oe:key="VMType" oe:value="Hybrid"/>
      <Property oe:key="ManagementIPv4Address"
oe:value="\${CwInstance1MgmtInterface.PrimaryPrivateIpAddress}"/>
      <Property oe:key="ManagementIPv4Gateway" oe:value="\${CwMgmtSubnetGateway}"/>

      <Property oe:key="ManagementIPv4Netmask" oe:value="\${CwMgmtSubnetNetmask}"/>

      <Property oe:key="ManagementIPv6Address" oe:value="::0"/>
      <Property oe:key="ManagementIPv6Gateway" oe:value="::1"/>
      <Property oe:key="ManagementIPv6Netmask" oe:value="64"/>
      <Property oe:key="ManagerPeerIPs"
oe:value="\${CwInstance1MgmtInterface.PrimaryPrivateIpAddress}
\${CwInstance2MgmtInterface.PrimaryPrivateIpAddress}
\${CwInstance3MgmtInterface.PrimaryPrivateIpAddress}"/>
      - Fn::Join:
        - ""
        - - '<Property oe:key="ManagementVIP" oe:value="'
          - Fn::Select: [0, Fn::GetAtt: [CwInstance1MgmtInterface,
SecondaryPrivateIpAddresses]]
          - '"/>'
          - "\n"
    - !If
      - DeployDataInterface
      # Join statement to construct the Data Interface configs
      - Fn::Join:
        - "\n"
        - - Fn::Sub: |
          <Property oe:key="DataIPv4Address"
oe:value="\${CwInstance1DataInterface.PrimaryPrivateIpAddress}"/>
          <Property oe:key="DataIPv4Netmask" oe:value="\${CwDataSubnetNetmask}"/>

          <Property oe:key="DataIPv4Gateway" oe:value="\${CwDataSubnetGateway}"/>

          <Property oe:key="DataPeerIPs"
oe:value="\${CwInstance1DataInterface.PrimaryPrivateIpAddress}
\${CwInstance2DataInterface.PrimaryPrivateIpAddress}
\${CwInstance3DataInterface.PrimaryPrivateIpAddress}"/>
          - Fn::Join:
            - ""
            - - '<Property oe:key="DataVIP" oe:value="'
              - Fn::Select: [0, Fn::GetAtt: [CwInstance1DataInterface,
SecondaryPrivateIpAddresses]]
              - '"/>'
              - "\n"
      # Default settings when no data interface is present
      - |
        <Property oe:key="DataIPv4Address" oe:value="0.0.0.0"/>
        <Property oe:key="DataIPv4Netmask" oe:value="255.255.255.0"/>
        <Property oe:key="DataIPv4Gateway" oe:value="0.0.0.0"/>
        <Property oe:key="DataVIP" oe:value="0.0.0.0"/>

```

```

        <Property oe:key="DataPeerIPs" oe:value=""/>
- !Sub |
  <Property oe:key="NTP" oe:value="169.254.169.123"/>
  <Property oe:key="DNsv4" oe:value="169.254.169.253"/>
  <Property oe:key="DNsv6" oe:value="::0"/>
  <Property oe:key="Domain" oe:value=""/>
  <Property oe:key="InitMasterCount" oe:value="3"/>
  <Property oe:key="InitNodeCount" oe:value="3"/>
  <Property oe:key="VMLocation" oe:value="AWS"/>
  <Property oe:key="DataIPv6Address" oe:value="::0"/>
  <Property oe:key="DataIPv6Gateway" oe:value="::1"/>
  <Property oe:key="DataIPv6Netmask" oe:value="64"/>
  <Property oe:key="Deployment" oe:value="cw_ipv4"/>
  <Property oe:key="Disclaimer" oe:value="Cisco Crosswork"/>
  <Property oe:key="K8Orch" oe:value=""/>
  <Property oe:key="CwInstaller" oe:value="False"/>
  <Property oe:key="corefs" oe:value="20"/>
  <Property oe:key="ddatafs" oe:value="\${DataDiskSize}"/>
  <Property oe:key="logfs" oe:value="10"/>
  <Property oe:key="ramdisk" oe:value="0"/>
  <Property oe:key="ssd" oe:value="50"/>
  <Property oe:key="K8sServiceNetworkV4" oe:value="\${K8sServiceNetwork}"/>
  <Property oe:key="K8sPodNetworkV4" oe:value="\${K8sPodNetwork}"/>
</PropertySection>
</Environment>

CwInstance2:
#
http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html

Type: AWS::EC2::Instance
DependsOn:
- CommonCwLaunchTemplate
Properties:
LaunchTemplate:
  Version: 1
  LaunchTemplateId: !Ref CommonCwLaunchTemplate
NetworkInterfaces: !If
- DeployDataInterface
- - NetworkInterfaceId: !Ref CwInstance2MgmtInterface
  DeviceIndex: "0"
- - NetworkInterfaceId: !Ref CwInstance2DataInterface
  DeviceIndex: "1"
- - NetworkInterfaceId: !Ref CwInstance2MgmtInterface
  DeviceIndex: "0"
Tags:
- Key: Name
  Value: Cw-EC2-VM2
UserData: !Base64
  'Fn::Join':
  - ''
  - - !Sub |
    <?xml version="1.0" encoding="UTF-8"?>
    <Environment
      <PlatformSection>
        <Kind>EC2</Kind>
      </PlatformSection>
      <PropertySection>
        <Property oe:key="CWPASSWORD" oe:value="\${CwSSHPASSWORD}"/>
        <Property oe:key="CWUsername" oe:value="cw-admin"/>
      </PropertySection>
    </Environment>
  - Fn::Join:
    - ''
    - - <Property oe:key="AwsIamRole" oe:value=""
      - !Ref EC2ENIRole

```

```

- '"/>'
- "\n"
- !Sub |
  <Property oe:key="IsSeed" oe:value="False"/>
  <Property oe:key="VMType" oe:value="Hybrid"/>
  <Property oe:key="ManagementIPv4Address"
oe:value="\${CwInstance2MgmtInterface.PrimaryPrivateIpAddress}"/>
  <Property oe:key="ManagementIPv4Gateway" oe:value="\${CwMgmtSubnetGateway}"/>

  <Property oe:key="ManagementIPv4Netmask" oe:value="\${CwMgmtSubnetNetmask}"/>

  <Property oe:key="ManagementIPv6Address" oe:value="::0"/>
  <Property oe:key="ManagementIPv6Gateway" oe:value="::1"/>
  <Property oe:key="ManagementIPv6Netmask" oe:value="64"/>
  <Property oe:key="ManagerPeerIPs"
oe:value="\${CwInstance1MgmtInterface.PrimaryPrivateIpAddress}
\${CwInstance2MgmtInterface.PrimaryPrivateIpAddress}
\${CwInstance3MgmtInterface.PrimaryPrivateIpAddress}"/>
  - Fn::Join:
    - ""
    - - '<Property oe:key="ManagementVIP" oe:value="'
      - Fn::Select: [0, Fn::GetAtt: [CwInstance1MgmtInterface,
SecondaryPrivateIpAddresses]]
      - '"/>'
      - "\n"
  - !If
    - DeployDataInterface
    # Join statement to construct the Data Interface configs
    - Fn::Join:
      - "\n"
      - - Fn::Sub: |
          <Property oe:key="DataIPv4Address"
oe:value="\${CwInstance2DataInterface.PrimaryPrivateIpAddress}"/>
          <Property oe:key="DataIPv4Netmask" oe:value="\${CwDataSubnetNetmask}"/>

          <Property oe:key="DataIPv4Gateway" oe:value="\${CwDataSubnetGateway}"/>

          <Property oe:key="DataPeerIPs"
oe:value="\${CwInstance1DataInterface.PrimaryPrivateIpAddress}
\${CwInstance2DataInterface.PrimaryPrivateIpAddress}
\${CwInstance3DataInterface.PrimaryPrivateIpAddress}"/>
          - Fn::Join:
            - ""
            - - '<Property oe:key="DataVIP" oe\value="'
              - Fn::Select: [0, Fn::GetAtt: [CwInstance1DataInterface,
SecondaryPrivateIpAddresses]]
              - '"/>'
              - "\n"
          # Default settings when no data interface is present
          - |
            <Property oe:key="DataIPv4Address" oe:value="0.0.0.0"/>
            <Property oe:key="DataIPv4Netmask" oe:value="255.255.255.0"/>
            <Property oe:key="DataIPv4Gateway" oe:value="0.0.0.0"/>
            <Property oe:key="DataVIP" oe:value="0.0.0.0"/>
            <Property oe:key="DataPeerIPs" oe:value=""/>
      - !Sub |
        <Property oe:key="NTP" oe:value="169.254.169.123"/>
        <Property oe:key="DNSv4" oe:value="169.254.169.253"/>
        <Property oe:key="DNSv6" oe:value="::0"/>
        <Property oe:key="Domain" oe:value=""/>
        <Property oe:key="InitMasterCount" oe:value="3"/>
        <Property oe:key="InitNodeCount" oe:value="3"/>
        <Property oe:key="VMLocation" oe:value="AWS"/>
        <Property oe:key="DataIPv6Address" oe:value="::0"/>

```

```

    <Property oe:key="DataIPv6Gateway" oe:value="::1"/>
    <Property oe:key="DataIPv6Netmask" oe:value="64"/>
    <Property oe:key="Deployment" oe:value="cw_ipv4"/>
    <Property oe:key="Disclaimer" oe:value="Cisco Crosswork"/>
    <Property oe:key="K8Orch" oe:value=""/>
    <Property oe:key="CwInstaller" oe:value="False"/>
    <Property oe:key="corefs" oe:value="20"/>
    <Property oe:key="ddatafs" oe:value="\${DataDiskSize}"/>
    <Property oe:key="logfs" oe:value="10"/>
    <Property oe:key="ramdisk" oe:value="0"/>
    <Property oe:key="ssd" oe:value="50"/>
    <Property oe:key="K8sServiceNetworkV4" oe:value="\${K8sServiceNetwork}"/>
    <Property oe:key="K8sPodNetworkV4" oe:value="\${K8sPodNetwork}"/>
  </PropertySection>
</Environment>

CwInstance3:
#
http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html

Type: AWS::EC2::Instance
Properties:
  LaunchTemplate:
    Version: 1
    LaunchTemplateId: !Ref CommonCwLaunchTemplate
  NetworkInterfaces: !If
    - DeployDataInterface
    - - NetworkInterfaceId: !Ref CwInstance3MgmtInterface
      DeviceIndex: "0"
    - - NetworkInterfaceId: !Ref CwInstance3DataInterface
      DeviceIndex: "1"
    - - NetworkInterfaceId: !Ref CwInstance3MgmtInterface
      DeviceIndex: "0"
  Tags:
    - Key: Name
      Value: Cw-EC2-VM3
  UserData: !Base64
    'Fn::Join':
      - ''
      - - !Sub |
          <?xml version="1.0" encoding="UTF-8"?>
          <Environment
            <PlatformSection>
              <Kind>EC2</Kind>
            </PlatformSection>
            <PropertySection>
              <Property oe:key="CWPassword" oe:value="\${CwSSHPassWord}"/>
              <Property oe:key="CWUsername" oe:value="cw-admin"/>
            </PropertySection>
          </Environment>
      - Fn::Join:
          - ""
          - - '<Property oe:key="AwsIamRole" oe:value="'
            - !Ref EC2ENIRole
            - '"/>'
            - "\n"
      - !Sub |
          <Property oe:key="IsSeed" oe:value="False"/>
          <Property oe:key="VMType" oe:value="Hybrid"/>
          <Property oe:key="ManagementIPv4Address"
            oe:value="\${CwInstance3MgmtInterface.PrimaryPrivateIpAddress}"/>
          <Property oe:key="ManagementIPv4Gateway" oe:value="\${CwMgmtSubnetGateway}"/>
          <Property oe:key="ManagementIPv4Netmask" oe:value="\${CwMgmtSubnetNetmask}"/>
          <Property oe:key="ManagementIPv6Address" oe:value="::0"/>

```



```

        <Property oe:key="ManagementIPv6Gateway" oe:value="::1"/>
        <Property oe:key="ManagementIPv6Netmask" oe:value="64"/>
        <Property oe:key="ManagerPeerIPs"
oe:value="${CwInstance1MgmtInterface.PrimaryPrivateIpAddress}
${CwInstance2MgmtInterface.PrimaryPrivateIpAddress}
${CwInstance3MgmtInterface.PrimaryPrivateIpAddress}"/>
    - Fn::Join:
      - ""
      - - '<Property oe:key="ManagementVIP" oe:value="'
        - Fn::Select: [0, Fn::GetAtt: [CwInstance1MgmtInterface,
SecondaryPrivateIpAddresses]]
        - '"/>'
        - "\n"
    - !If
      - DeployDataInterface
      # Join statement to construct the Data Interface configs
      - Fn::Join:
        - "\n"
        - - Fn::Sub: |
            <Property oe:key="DataIPv4Address"
oe:value="${CwInstance3DataInterface.PrimaryPrivateIpAddress}"/>
            <Property oe:key="DataIPv4Netmask" oe:value="${CwDataSubnetNetmask}"/>

            <Property oe:key="DataIPv4Gateway" oe:value="${CwDataSubnetGateway}"/>

            <Property oe:key="DataPeerIPs"
oe:value="${CwInstance1DataInterface.PrimaryPrivateIpAddress}
${CwInstance2DataInterface.PrimaryPrivateIpAddress}
${CwInstance3DataInterface.PrimaryPrivateIpAddress}"/>
            - Fn::Join:
              - ""
              - - '<Property oe:key="DataVIP" oe:value="'
                - Fn::Select: [0, Fn::GetAtt: [CwInstance1DataInterface,
SecondaryPrivateIpAddresses]]
                - '"/>'
                - "\n"
            # Default settings when no data interface is present
            - |
                <Property oe:key="DataIPv4Address" oe:value="0.0.0.0"/>
                <Property oe:key="DataIPv4Netmask" oe:value="255.255.255.0"/>
                <Property oe:key="DataIPv4Gateway" oe:value="0.0.0.0"/>
                <Property oe:key="DataVIP" oe:value="0.0.0.0"/>
                <Property oe:key="DataPeerIPs" oe:value=""/>
            - !Sub |
                <Property oe:key="NTP" oe:value="169.254.169.123"/>
                <Property oe:key="DNSv4" oe:value="169.254.169.253"/>
                <Property oe:key="DNSv6" oe:value="::0"/>
                <Property oe:key="Domain" oe:value=""/>
                <Property oe:key="InitMasterCount" oe:value="3"/>
                <Property oe:key="InitNodeCount" oe:value="3"/>
                <Property oe:key="VMLocation" oe:value="AWS"/>
                <Property oe:key="DataIPv6Address" oe:value="::0"/>
                <Property oe:key="DataIPv6Gateway" oe:value="::1"/>
                <Property oe:key="DataIPv6Netmask" oe:value="64"/>
                <Property oe:key="Deployment" oe:value="cw_ipv4"/>
                <Property oe:key="Disclaimer" oe:value="Cisco Crosswork"/>
                <Property oe:key="K8Orch" oe:value=""/>
                <Property oe:key="CwInstaller" oe:value="False"/>
                <Property oe:key="corefs" oe:value="20"/>
                <Property oe:key="ddatafs" oe:value="${DataDiskSize}"/>
                <Property oe:key="logfs" oe:value="10"/>
                <Property oe:key="ramdisk" oe:value="0"/>
                <Property oe:key="ssd" oe:value="50"/>
                <Property oe:key="K8sServiceNetworkV4" oe:value="${K8sServiceNetwork}"/>

```

```

    <Property oe:key="K8sPodNetworkV4" oe:value="\${K8sPodNetwork}"/>
  </PropertySection>
</Environment>

```

## Outputs:

```

CrossworkClusterStack:
  Description: The Name of the Cw cluster stack
  Value: !Sub ${AWS::StackName}
CrossworkManagementVIP:
  Value:
    Fn::Select: [0, Fn::GetAtt: [CwInstance1MgmntInterface, SecondaryPrivateIpAddresses]]

Export:
  Name: !Sub ${AWS::StackName}-Cw-MgmntVIP
CrossworkManagementIP1:
  Value: !Sub ${CwInstance1MgmntInterface.PrimaryPrivateIpAddress}
CrossworkManagementIP2:
  Value: !Sub ${CwInstance2MgmntInterface.PrimaryPrivateIpAddress}
CrossworkManagementIP3:
  Value: !Sub ${CwInstance3MgmntInterface.PrimaryPrivateIpAddress}
CrossworkDataVIP:
  Value:
    Fn::Select: [0, Fn::GetAtt: [CwInstance1DataInterface, SecondaryPrivateIpAddresses]]
Export:
  Name: !Sub ${AWS::StackName}-Cw-DataVIP
  Condition: DeployDataInterface
CrossworkDataIP1:
  Value: !Sub ${CwInstance1DataInterface.PrimaryPrivateIpAddress}
  Condition: DeployDataInterface
CrossworkDataIP2:
  Value: !Sub ${CwInstance2DataInterface.PrimaryPrivateIpAddress}
  Condition: DeployDataInterface
CrossworkDataIP3:
  Value: !Sub ${CwInstance3DataInterface.PrimaryPrivateIpAddress}
  Condition: DeployDataInterface

```

## Sample CloudFormation Template for installing Crosswork Data Gateway on EC2




---

**Attention** The CF template (.yaml file) displayed in this section contains the details to install a Standard Crosswork Data Gateway with a single interface. Please note that it is only a sample, and you can always create a different CF template according to your production preferences and execute it as per the steps mentioned in this section. This document assumes that a user of this procedure is familiar with AWS and the CloudFormation concepts, and as such, the CF template creation is out of the scope of this document.

---

```
Description: "Sample template for deploying CDG4.1 VMs - v4.4"
```

## Metadata:

```

AWS::CloudFormation::Interface:
  ParameterGroups:
    -
      Label:
        default: "Cw Network Configuration"
      Parameters:
        - VpcId

```

- SecGroup
- CDGSSHPassword
- CDGAmiId
- CNCControllerIP
- CNCControllerPassword
- InterfaceDeploymentMode
- CDGInterface0SubnetId
- CDGInterface0Gateway
- CDGInterface0SubnetNetmask
- CDGInterface1SubnetId
- CDGInterface1Gateway
- CDGInterface1SubnetNetmask
- CDGInterface2SubnetId
- CDGInterface2Gateway
- CDGInterface2SubnetNetmask

**Parameters:****VpcId:**

Type: AWS::EC2::VPC::Id  
Description: VpcId of your existing Virtual Private Cloud (VPC)  
ConstraintDescription: Must be the VPC Id of an existing Virtual Private Cloud.

**CDGAmiId:**

Type: AWS::EC2::Image::Id  
Description: Provide CDG AMI ID

**CDGSSHPassword:**

Type: String  
NoEcho: True  
Description: Enter the SSH password to be configured on the CDG

**SecGroup:**

Type: AWS::EC2::SecurityGroup::Id  
Description: Pre-created security group to be applied. Must allow ingress access for ports 22, 30160:31560

**CNCControllerPassword:**

Type: String  
NoEcho: True  
Description: Enter the cw-admin user password used to access CNC/Cw Controller

**DataDiskSize:**

Description: Cw data disk size.  
Type: Number  
MinValue: 20  
Default: 50

**CDGProfile:**

Type: String  
Description: Deployment profile of the CDG  
AllowedValues:

- Standard
- Extended

Default: Standard

**InstanceType:**

Description: Enter EC2 instance type for the node instances. Default is m5zn.3xlarge.  
Type: String  
AllowedValues:

- m5.4xlarge
- m5.8xlarge
- m5.12xlarge
- m5d.4xlarge
- m5d.8xlarge

```

- m5d.12xlarge
- r5.4xlarge
- r5.8xlarge
- r5.12xlarge
- c5.4xlarge
- c5.8xlarge
- c5.12xlarge
- m5zn.3xlarge
Default: m5zn.3xlarge

InterfaceDeploymentMode:
  Type: String
  Description: Select the single (all traffic), dual (Management + Data) or triple
(Management + Data + Control) interface deployment mode.
  AllowedValues:
    - 1
    - 2
    - 3

CDGInterface0SubnetId:
  Type: AWS::EC2::Subnet::Id
  Description: Select the first interface subnet for the CDG VM.

CDGInterface0Gateway:
  Type: String
  Description: Enter the default gateway on the selected subnet. This is typically the
first address on the subnet.
  AllowedPattern: (\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}

CDGInterface1SubnetId:
  Type: AWS::EC2::Subnet::Id
  Description: Select the first interface subnet for the CDG VM. Ignored if not using
dual interface mode.

CDGInterface1Gateway:
  Type: String
  Description: Enter the default gateway on the selected subnet. This is typically the
first address on the subnet.
  AllowedPattern: (\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}
  Default: "0.0.0.1"

CDGInterface2SubnetId:
  Type: AWS::EC2::Subnet::Id
  Description: Select the first interface subnet for the CDG VM. Ignored if not using
triple interface mode.

CDGInterface2Gateway:
  Type: String
  Description: Enter the default gateway on the selected subnet. This is typically the
first address on the subnet.
  AllowedPattern: (\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}
  Default: "0.0.0.1"

CDGInterface0IPAddress:
  Type: String
  Description: OPTIONAL - Enter a *free* IP address on the 1st subnet. If set to "0.0.0.0",
an IP address will be allocated automatically .
  Default: "0.0.0.0"
  AllowedPattern: (\d{1,3})\.\d{1,3}\.\d{1,3}\.\d{1,3}|^$

CDGInterface0SubnetNetmask:
  Type: String
  Description: Enter the subnet netmask in dotted decimal form, eg 255.255.255.0.
  Default: "255.255.255.0"

```

```

    AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})

CDGInterface1IPAddress:
  Type: String
  Description: OPTIONAL - Enter a *free* IP address on the 2nd subnet. If set to 0.0.0.0,
an IP address will be allocated automatically.
  Default: "0.0.0.0"
  AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})|^$

CDGInterface1SubnetNetmask:
  Type: String
  Description: Enter the subnet netmask in dotted decimal form, eg 255.255.255.0. Ignored
if not using dual interface mode.
  Default: "255.255.255.0"
  AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})

CDGInterface2IPAddress:
  Type: String
  Description: OPTIONAL - Enter a *free* IP address on the 3rd subnet. If set to 0.0.0.0,
an IP address will be allocated automatically.
  Default: "0.0.0.0"
  AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})|^$

CDGInterface2SubnetNetmask:
  Type: String
  Description: Enter the subnet netmask in dotted decimal form, eg 255.255.255.0. Ignored
if not using triple interface mode.
  Default: "255.255.255.0"
  AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})

CNCControllerIP:
  Type: String
  Description: Specify the address of the Crosswork CDG controller
  AllowedPattern: ((\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3}))|^$
  Default: ""

Conditions:
  DeployInterface0: !Equals
    - !Ref InterfaceDeploymentMode
    - "1"

  DeployInterface1: !Or
    - Fn::Equals:
      - !Ref InterfaceDeploymentMode
      - "2"
    - Fn::Equals:
      - !Ref InterfaceDeploymentMode
      - "3"

  DeployInterface2: !Equals
    - !Ref InterfaceDeploymentMode
    - "3"

  Setif0IP: !Not
    - !Equals
      - !Ref CDGInterface0IPAddress
      - "0.0.0.0"

  Setif1IP: !And
    - !Not
      - !Equals
        - !Ref CDGInterface1IPAddress
        - "0.0.0.0"
    - !Not

```

```

    - !Condition DeployInterface0

Setif2IP: !And
  - !Not
    - !Equals
      - !Ref CDGInterface2IPAddress
      - "0.0.0.0"
  - !Not
    - !Condition DeployInterface0
  - !Not
    - !Condition DeployInterface1

Resources:
  EC2ENIRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Policies:
        - PolicyName: eni-modification
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - ec2:DescribeNetworkInterfaces
                  - ec2:AssignPrivateIpAddresses
                  - ec2:UnassignPrivateIpAddresses
                Resource: "*"

  CDGEC2IamInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      InstanceProfileName: !Sub ${AWS::StackName}-CDG-EC2IamInstanceProfile
      Path: "/cdg/"
      Roles:
        - !Ref EC2ENIRole

  CDG1VNIC0:
    Type: AWS::EC2::NetworkInterface
    Properties:
      Description: "CDG1-VNIC0"
      GroupSet:
        - !Ref SecGroup
      PrivateIpAddresses: !If
        - Setif0IP
          - Primary: true
            PrivateIpAddress: !Ref CDGInterface0IPAddress
        - !Ref 'AWS::NoValue'
      SubnetId: !Ref CDGInterface0SubnetId
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-CDG1-VNIC0

  CDG1VNIC1:
    Type: AWS::EC2::NetworkInterface
    Properties:

```

```

Description: "CDG1-VNIC1"
GroupSet:
  - !Ref SecGroup
PrivateIpAddresses: !If
  - Setif1IP
  - - Primary: true
    PrivateIpAddress: !Ref CDGInterface1IPAddress
  - !Ref 'AWS::NoValue'
SubnetId: !Ref CDGInterface1SubnetId
Tags:
  - Key: Name
    Value: !Sub ${AWS::StackName}-CDG1-VNIC1
Condition: DeployInterface1

CDG1VNIC2:
Type: AWS::EC2::NetworkInterface
Properties:
  Description: "CDG1-VNIC2"
  GroupSet:
    - !Ref SecGroup
  PrivateIpAddresses: !If
    - Setif2IP
    - - Primary: true
      PrivateIpAddress: !Ref CDGInterface2IPAddress
    - !Ref 'AWS::NoValue'
  SubnetId: !Ref CDGInterface2SubnetId
  Tags:
    - Key: Name
      Value: !Sub ${AWS::StackName}-CDG1-VNIC2
  Condition: DeployInterface2

CommonLaunchTemplateCDG4:
Type: AWS::EC2::LaunchTemplate
Properties:
  LaunchTemplateName: !Sub ${AWS::StackName}-CommonLaunchTemplateCDG4
  LaunchTemplateData:
    InstanceType: !Ref InstanceType
    ImageId: !Ref "CDGAmiId"
    BlockDeviceMappings:
      - Ebs:
          VolumeSize: !Ref DataDiskSize
          DeleteOnTermination: True
          VolumeType: standard
          DeviceName: /dev/sdb
    MetadataOptions:
      HttpPutResponseHopLimit: 2
  IamInstanceProfile:
    Arn: !GetAtt
      - CDGEC2IamInstanceProfile
      - Arn

CDGInstance:
Type: AWS::EC2::Instance
Properties:
  LaunchTemplate:
    Version: 1
    LaunchTemplateId: !Ref CommonLaunchTemplateCDG4
  NetworkInterfaces: !If
    - DeployInterface2
    - - NetworkInterfaceId: !Ref CDG1VNIC0
      DeviceIndex: "0"
    - NetworkInterfaceId: !Ref CDG1VNIC1
      DeviceIndex: "1"

```

```

    - NetworkInterfaceId: !Ref CDG1VNIC2
      DeviceIndex: "2"
  - !If
    - DeployInterface1
      - NetworkInterfaceId: !Ref CDG1VNIC0
        DeviceIndex: "0"
      - NetworkInterfaceId: !Ref CDG1VNIC1
        DeviceIndex: "1"
      - NetworkInterfaceId: !Ref CDG1VNIC0
        DeviceIndex: "0"
Tags:
  - Key: Name
    Value: !Sub ${AWS::StackName}-CDG4.0
UserData: !Base64
Fn::Join:
  - ''
  - - !Sub |
      AwsIamRole=${EC2ENIRole}
      ActiveVnics=${InterfaceDeploymentMode}
      AllowRFC8190=Yes
      AuditdAddress=
      AuditdPort=60
      ControllerCertChainPwd=${CNCControllerPassword}
      ControllerIP=${CNCControllerIP}
      ControllerPort=30607

ControllerSignCertChain=cw-admin@${CNCControllerIP}:/home/cw-admin/controller.pem
ControllerTlsCertChain=
Deployment=Crosswork On-Premise
Description=${AWS::StackName}-CDG4.1-1
DGAppdataDisk=5
DGCertChain=
DGCertChainPwd=
DGCertKey=
DNS=169.254.169.253
DNSSEC=False
DNSTLS=False
Domain=
EnrollmentPassphrase=
EnrollmentURI=
Hostname=${AWS::StackName}-CDG4.1
Label=
LLMNR=False
mDNS=False
NTP=169.254.169.123
NTPAuth=False
NTPKey=
NTPKeyFile=
NTPKeyFilePwd=
PortSNMPTrap=1062
PortSyslogUDP=9514
PortSyslogTCP=9898
PortSyslogTLS=6514
Profile=${CDGProfile}
ProxyBypass=
ProxyCertChain=
ProxyCertChainPwd=
ProxyPassphrase=
ProxyURL=
ProxyUsername=
SyslogAddress=
SyslogCertChain=
SyslogCertChainPwd=
SyslogPeerName=

```



```

SyslogPort=514
SyslogProtocol=UDP
SyslogTLS=False
UseRemoteAuditd=False
UseRemoteSyslog=False
Vnic0IPv4Address=${CDG1VNIC0.PrimaryPrivateIpAddress}
Vnic0IPv4Gateway=${CDGInterface0Gateway}
Vnic0IPv4Method=Static
Vnic0IPv4Netmask=${CDGInterface0SubnetNetmask}
Vnic0IPv4SkipGateway=False
Vnic0IPv6Address=:0
Vnic0IPv6Gateway=:1
Vnic0IPv6Method=None
Vnic0IPv6Netmask=64
Vnic0IPv6SkipGateway=False
- !If
- DeployInterface1
- !Sub |
  Vnic1IPv4Address=${CDG1VNIC1.PrimaryPrivateIpAddress}
  Vnic1IPv4Gateway=${CDGInterface1Gateway}
- |
  Vnic1IPv4Address=0.0.0.0
  Vnic1IPv4Gateway=0.0.0.1
- !Sub |
  Vnic1IPv4Method=Static
  Vnic1IPv4Netmask=${CDGInterface1SubnetNetmask}
  Vnic1IPv4SkipGateway=False
  Vnic1IPv6Address=:0
  Vnic1IPv6Gateway=:1
  Vnic1IPv6Method=None
  Vnic1IPv6Netmask=64
  Vnic1IPv6SkipGateway=False
- !If
- DeployInterface2
- !Sub |
  Vnic2IPv4Address=${CDG1VNIC2.PrimaryPrivateIpAddress}
  Vnic2IPv4Gateway=${CDGInterface2Gateway}
- |
  Vnic2IPv4Address=0.0.0.0
  Vnic2IPv4Gateway=0.0.0.1
- !Sub |
  Vnic2IPv4Method=None
  Vnic2IPv4Netmask=${CDGInterface2SubnetNetmask}
  Vnic2IPv4SkipGateway=False
  Vnic2IPv6Address=:0
  Vnic2IPv6Gateway=:1
  Vnic2IPv6Method=None
  Vnic2IPv6Netmask=64
  Vnic2IPv6SkipGateway=False
  dg-adminPassword=${CDGSSHPassword}
  dg-operPassword=${CDGSSHPassword}

```

## Outputs:

```

CDGStack:
  Description: The Name of the CDG cluster stack
  Value: !Sub ${AWS::StackName}
CDGInterface0IPAddress:
  Value: !Sub ${CDG1VNIC0.PrimaryPrivateIpAddress}
CDGInterface1IPAddress:
  Value: !Sub ${CDG1VNIC1.PrimaryPrivateIpAddress}
  Condition: DeployInterface1
CDGInterface2IPAddress:
  Value: !Sub ${CDG1VNIC2.PrimaryPrivateIpAddress}
  Condition: DeployInterface2

```

