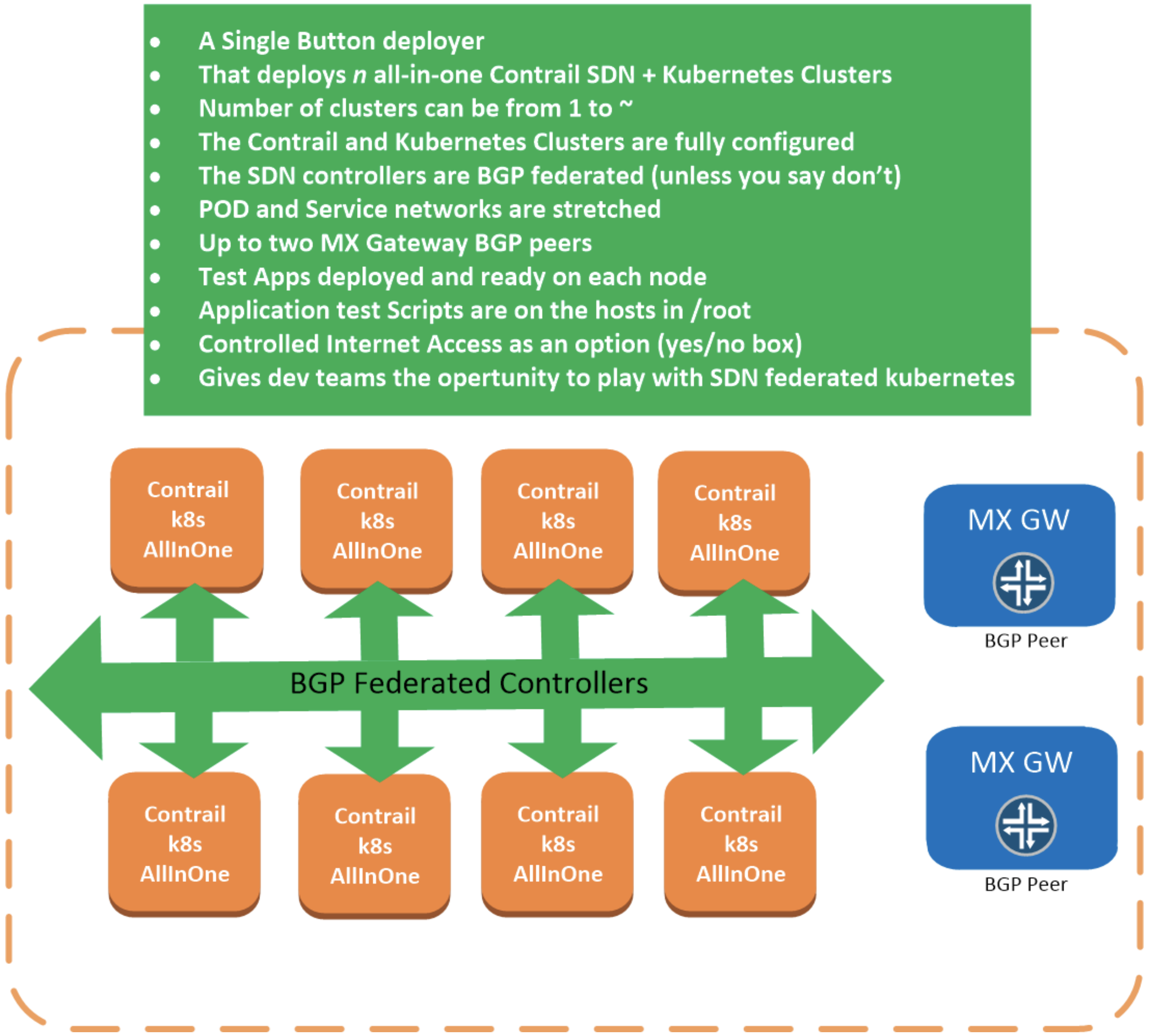
Federation: Moving beyond one Kubernetes cluster with Contrail SDN

Last edited by **Simon Green** 1 month ago

Here we explore Kubernetes federation, using AWS, an auto scaling group and a great deal of automation



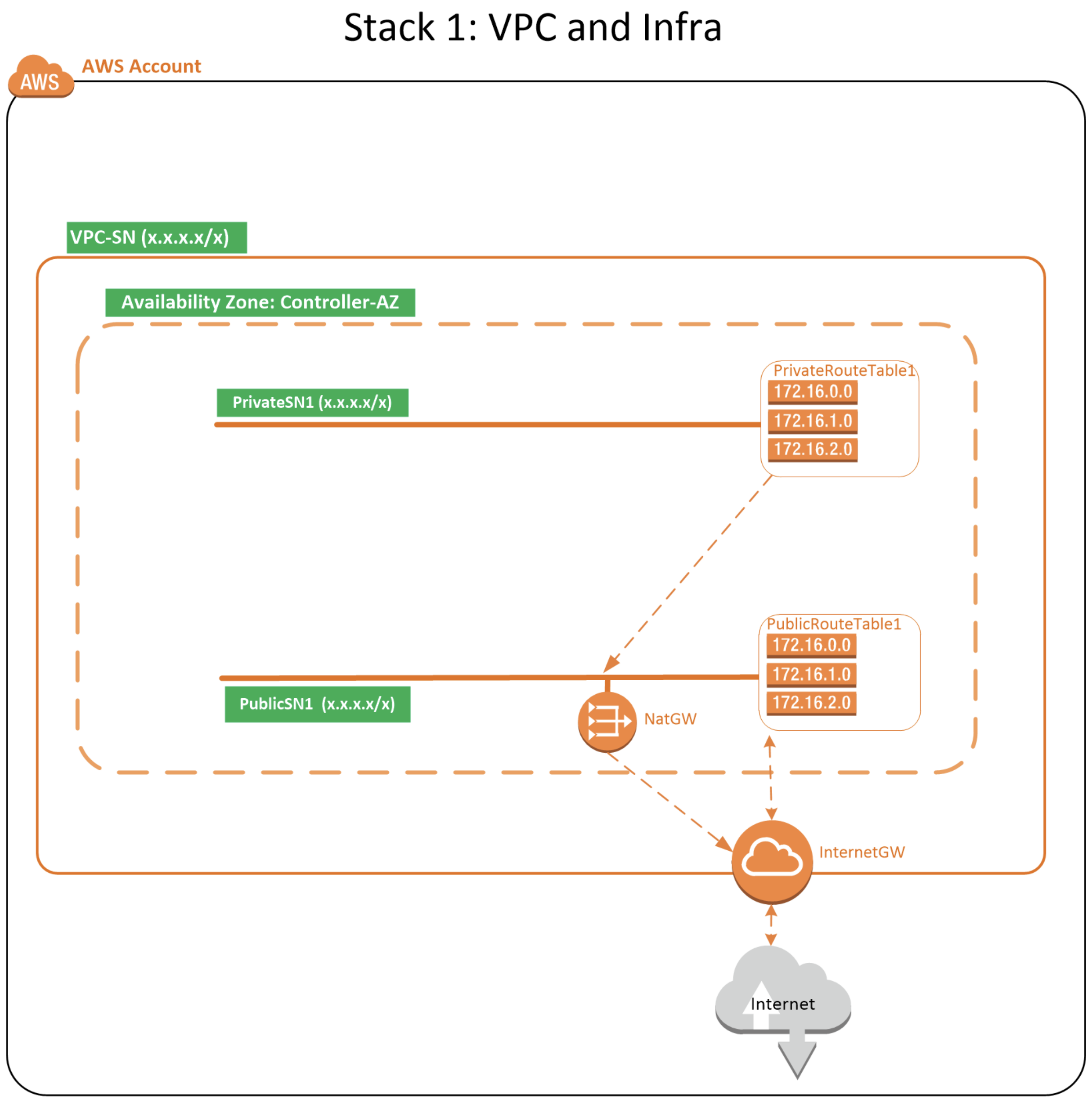
- A Single Button deployer
- That deploys *n* all-in-one Contrail SDN + Kubernetes Clusters
- Number of clusters can be from 1 to ~
- The Contrail and Kubernetes Clusters are fully configured
- The SDN controllers are BGP federated (unless you say don't)
- POD and Service networks are stretched
- Up to two MX Gateway BGP peers
- Test Apps deployed and ready on each node
- Application test Scripts are on the hosts in /root
- Controlled Internet Access as an option (yes/no box)
- Gives dev teams the opertunity to play with SDN federated kubernetes

One Time Setup:

1. You need an AWS account. Login to the AWS console with your browser.
2. In your AWS account console you need to an EC2 SSH key pair named "ContrailKey" within the region Frankfurt
3. You might be asked to accept the commercial Ts and Cs for CentosOS free ami first time you deploy, the link to click is in the stack error.
4. You will need an account on the juniper contrail docker registry hub.juniper.net/contrail (user and password) in order to use our contrail SDN software.
5. you should be able to deploy this stack in most regions of AWS, I'm using Ireland here.

1. Stack1 deploys the infra
2. Stack2 is a brownfield and deploys Contrail+Kubernetes clusters into that infra
3. If you want to use your own AWS infra simply skip stack1 and go straight to stack2

Stack1



How to deploy Stack1: Infra for the Federated AllInOne clusters

1. Click this link to launch the CloudFormation stack into your account (Ireland) Launch Stack

   Note: if you right click and copy this link, then share it with your customers via email. It will work for them as well

   ->NEXT

   ->NEXT

   ->NEXT

   ->Deploy Stack

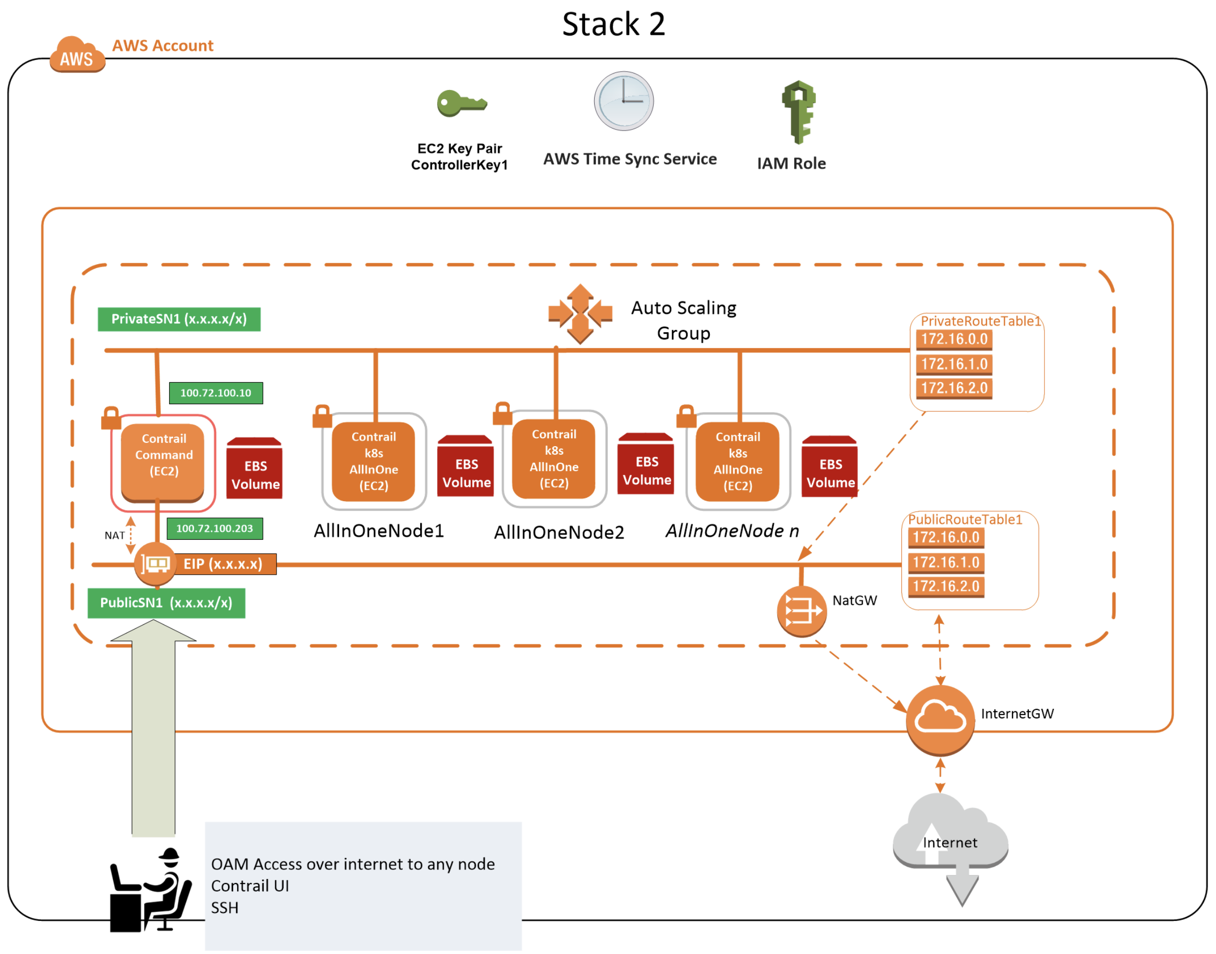
if you prefer to deploy using AWS cli then the command is provided below, with example values.

```
echo "all parameters are optional in this stack"
aws cloudformation create-stack \
--stack-name Kubernetes-Federated-Stack1-VPC \
--disable-rollback \
--template-url https://s3-eu-central-1.amazonaws.com/contrail-one-click-deployers/Contrail-k8s-Federated-Stack1.json \
--parameters \
ParameterKey=VPCCIDR1,ParameterValue="100.72.100.0/22" \
ParameterKey=PrivateSubnetCIDR1,ParameterValue="100.72.100.0/24" \
ParameterKey=PrivateSubnetCIDR2,ParameterValue="100.72.101.0/24" \
ParameterKey=PrivateSubnetCIDR3,ParameterValue="100.72.102.0/24" \
ParameterKey=PublicSubnetCIDR1,ParameterValue="100.72.103.0/24" \
ParameterKey=AvailabilityZone1,ParameterValue="eu-west-1a" \
ParameterKey=AvailabilityZone2,ParameterValue="eu-west-1b" \
ParameterKey=AvailabilityZone3,ParameterValue="eu-west-1c" \
ParameterKey=SiteName,ParameterValue="k8sFederation"
```

It just takes a few minutes to deploy stack1.

Note: All fields can be changed, but for the first run we suggest you go with the defaults

Stack2



How to deploy Stack2: Auto Scaling group if Contrail+Kubernetes nodes fully configured and ready to use

1. Click this link to launch the CloudFormation stack into your account (Ireland) Launch Stack

   Note: if you right click and copy this link, then share it with your customers via email. It will work for them as well
   ->NEXT
   complete the field ContainerRegistryUserName (this is your juniper docker registry user name)
   complete the field ContainerRegistryPassword (this is your juniper docker registry password)
   idVPC, This is the id of the VPC you will deploy into Console->VPC->VPCs->k8s-Federated
   idPublicSubnet1, This is the public subnet in the VPC.
   idPrivateSubnet1, This is the private subnet id to deploy the instances into. Console->VPC->Subnets->k8s-Federated-Private
   NumberOfClusters. This is the number of all in one clusters to deploy 1-250. 2+ and I'll federate them. The Default is 2.
   KeyName. This is the ec2 ssh key pair you want to use to access the instances.
   **Note: some browser password managers will wipe out the ContrailPassword if you see its bank then enter contrail123 or another memorable password
   ->NEXT
   Tick the box: I acknowledge that AWS CloudFormation might create IAM resources.
   ->NEXT
   ->Deploy Stack

If you plan to deploy lots of clusters then you will have to raise you limits for m5 instances in your AWS account, raise a support ticket and wait 5 minutes.

if you prefer to deploy using AWS cli then the command is provided below, with example values.

```
aws cloudformation create-stack \
--capabilities CAPABILITY_IAM \
--stack-name Kubernetes-Federated-Stack2-AllInOne \
--disable-rollback \
--template-url https://s3-eu-central-1.amazonaws.com/contrail-one-click-deployers/Contrail-k8s-Federated-Stack2.json \
--parameters \
ParameterKey=ContainerRegistryUserName,ParameterValue="JNPR-FieldUserxxx" \
ParameterKey=ContainerRegistryPassword,ParameterValue="EwRUWcLKbzTe2Nnxxxxx" \
ParameterKey=idPublicSubnet1,ParameterValue="subnet-0cc12622d5170bc8a" \
ParameterKey=idPrivateSubnet1,ParameterValue="subnet-00aa23253cebaeff3" \
ParameterKey=idVPC,ParameterValue="vpc-01083cb87dc1c7ff6" \
ParameterKey=NumberOfClusters,ParameterValue="10"
```

Stack2 will complete once one of the nodes reaches the end of its user-data. Typically around 15 minutes.
Whether you deploy 1 or 100 nodes its still 15 minutes as they deploy in parallel.

Note: All fields can be changed, but for the first run we suggest you go with the defaults
Note: I strongly suggest changing the stack parameter UserLocation which determines who can connect to it. If you leave it blank then we will allow access from 0.0.0.0/0, the whole of the internet.

1. Get a coffee, it typically takes 15 minutes to deploy. It will complete once one of the nodes reaches the end of its user-data.
2. Once the stack completes the CloudFormation outputs tab will show you how to access the hosts, the Contrail Command UI and the OpenStack UI
3. Whether you deploy 1 or 100 nodes its still 15 minutes as they deploy in parallel.

What does it build:

1. A Command instance configured for SSH and Web UI assess to the nodes in the cluster.
2. An auto scaling group
3. N x AllInOne Kubernetes+Contrail instances fully configured, peered, federated with test applications up and running.

---

1. How to access the nodes

```
#The stack outputs show you how to access the nodes. I have a helper script that configured the nat rules.
#for Example:
#Lets assume the public ip assigned to my command instance was 81.1.2.3
#The output of the stack will say something like.
ssh -i [your ContrailKey1 private key file] centos@34.254.121.19 sudo /usr/bin/add-nat.sh -n [node ip address]
#The console->ec2->instances view will show the available nodes. Click one and node the ip. Lets assume its 100.72.10
```

```
#Your command would be something like:
ssh -i ~/ContrailKey-ireland.pem centos@34.254.121.19 sudo /usr/bin/add-nat.sh -n 100.72.100.146
#After running it port 222 gets you ssh access to node 100.72.100.146
ssh -i ~/Downloads/ContrailKey-ireland.pem centos@34.254.121.19 -p 222
#Point your web browser to http://34.254.121.19:8143 and you will get the contrail ui
#You can rerun the add-nat.sh script to access a different node.
#If you switch to a different node, clear your browser history in order to see the new UI. For example:
ssh -i ~/ContrailKey-ireland.pem centos@34.254.121.19 sudo /usr/bin/add-nat.sh -n 100.72.100.159
ssh -i ~/Downloads/ContrailKey-ireland.pem centos@34.254.121.19 -p 222
#Clear your browser history and Point to http://34.254.121.19:8143 and you will get the contrail ui
```

2. Checking Kubernetes

```

#ssh into the node of interrest using port 222 as described above
sudo bash
#to see the pods and ip addresses
kubectl get pods --all-namespaces -o wide
#to connect to a test pod
kubectl -it exec [pod id] /bin/bash
kubectl -it exec [pod id] -n testing /bin/bash
```

3. Testing Hints

```
#All pod networks and service networks in the default name space have the same route target assigned (asn 64513 route
#so you will be able to ping between all pods and services in default namespace in any cluster.
#All testing namespaces (isolated namespace) have the same route target assigned (asn 64513 route-target 3000)
#so you will be able to ping between all pods and services in the testing namespace in any cluster
#how to performance test pods in separate federated clusters
#iperf3 -s on one pod - cluster A
#iperf3 -c [other pods ip] - cluster B
#You are now performance testing between pods in seperate federated clusters.
#In AWS the bigger the instance the faster the network. These are small instances, but they have amazing Contrail vro
```

For more testing hints I have a YouTube video.

All Done

How to secure OAM access:

1. The stack parameter UserLocation determines who can connect to it.
2. If you leave it blank we will allow access from 0.0.0.0/0, the whole of the internet.
3. If you set it to your laptop ip address x.x.x.x/32 only you can connect to it, its secured within the public SG.

4. You can also manually change the rules in the contrail command AWS security group.

The stack is very flexible, and as you see above almost all parameters have a workable default value. Details follow.

Breakdown of all stack parameters:

1. AllInOneInstanceType - Instance Type for the AllInOne Nodes - Default=m5.2xlarge
2. AvailabilityZone1 - Availability Zone for cluster - Default=eu-west-1a
3. CommandInstanceType - Instance Type for Contrail Command used for OAM access - Default m4.2xlarge
4. ContainerRegistryUserName - Your Juniper DockerHub Login - No Default must be completed
5. ContainerRegistryPassword - Your Juniper DockerHub Password - No Default must be completed
6. ContainerRegistryTag - The contrail version to download - Default=2005.1.66
7. ContrailCommandAZ1PrivateIP - The private subnet ip to use for Contrail command - Default=100.72.100.10
8. ContrailCommandAZ1PublicIP - The public subnet ip to use for Contrail Command - Default=100.72.103.10
9. ContrailCommandPrivateSNGatewayIP - The AWS GW IP on the private subnet, used by Contrail Command - Default=100.72.100.1
10. ContrailPassword - The Contrail UI Password - Default=contrail123 (user is admin)
11. DebugLogs - true enabled bug logs on all containers and will impact performance - Default=false
12. DeployContrailCommand - false and contrail command will not be deployed - Default=true (used if you have VPN or direct connect OAM access)
13. FullMesh - true and the stack will configure a full bgp mesh to federate all of the Clusters SDN controllers with each other - Default=true
14. InstallerAGITRepoOrS3 - whether to fetch the contrail yaml from s3 or GitHub - Default S3
15. InstallerBGitHubInstallerLocation - GitHub location (unused currently)
16. InstallerCS3Location - S3 location for the contrail cli yaml - Default=https://s3-eu-central-1.amazonaws.com/contrail-one-click-deployers/contrail-k8s-allinone.yaml
17. KeyName - your ssh key to use for the aws EC2 instances - Default=ContrailKey (either create a key named ContrailKey or select your ssh key)
18. MX1ASN - BGP ASN to use for the peer to Gateway MX1 (optional)
19. MX1IPAddress - BGP peer address for MX1 (optional)
20. MX1Name - name for MX1 (optional)
21. MX1Password - password for MX1 if you are running netconf (optional)
22. MX1User - username for MX1 if you are using netconf (optional)
23. MX2ASN - BGP ASN to use for the peer to Gateway MX2 (optional)
24. MX2IPAddress - BGP peer address for MX2 (optional)
25. MX2Name - name for MX2 (optional)
26. MX2Password - password for MX2 if you are running netconf (optional)
27. MX2User - username for MX2 if you are using netconf (optional)
28. NodeMTU - Host MTU for the AllInOneNodes (physical nic MTU) - Default=9000
29. NumberOfClusters - Number of AllInOne Contrail+Kubernetes clusters to deploy - Default=2
30. SGSubnet1 - Subnet to open up inbound traffic, in the instance security groups - Typically set for in premise federation, OAM access, or peering to other VPCs
31. SGSubnet2 - Subnet to open up inbound traffic, in the instance security groups - Typically set for in premise federation, OAM access, or peering to other VPCs
32. SGSubnet3 - Subnet to open up inbound traffic, in the instance security groups - Typically set for in premise federation, OAM access, or peering to other VPCs

33. SGSubnet4 - Subnet to open up inbound traffic, in the instance security groups - Typically set for in premise federation, OAM access, or peering to other VPCs
34. SSHPassword - back door password to use for instance access without an ssh key.
35. SiteName - Name used in the meta data for services - Default=k8sFederation
36. TestPod - if true we deploy test pods in default and testing namespace, as we las test scripts, for all clusters - Default=true
37. UserLocation - The subnet that can access Contrail Commands public IP for OAM access - Default=0.0.0.0/0
38. idVPC - The id of the VPC to deploy into - No Default, Must be set
39. idPrivateSubnet1 - The id of the Private Subnet to deploy into - No Default, Must be set
40. idPublicSubnet1 - The id of the Public Subnet to deploy into - Used by command. If you are not deploying Contrail command use a random value.
41. k8sBaseASN - The base number used for ASN. Each host we add the last octet if the host IP address to get unique ASN's per cluster.
42. k8sPodBaseIP - The base number used for the Pod IPAM. Each host we add the last octet if the host IP address into the third octet of this subnet address, to get unique Pod subnets per cluster.
43. k8sServiceBaseIP - The base number used for the Service IPAM. Each host we add the last octet if the host IP address into the third octet of this subnet address, to get unique Service subnets per cluster.
44. k8sFabricBaseIP - The base number used for the fabric IPAM. Each host we add the last octet if the host IP address into the third octet of this subnet address, to get unique fabric subnets per cluster.
45. k8sVersion - The Kubernetes version to install - Default=1.14.8