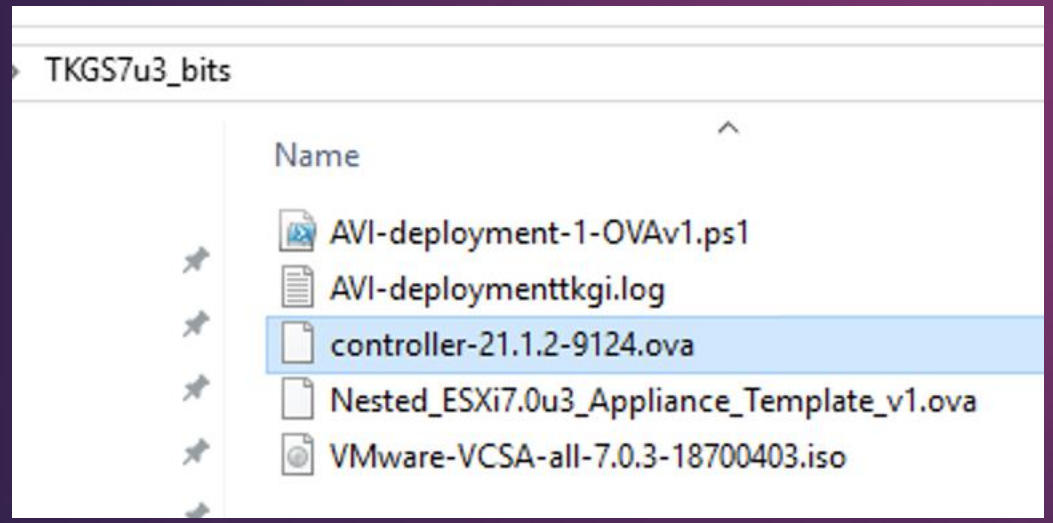
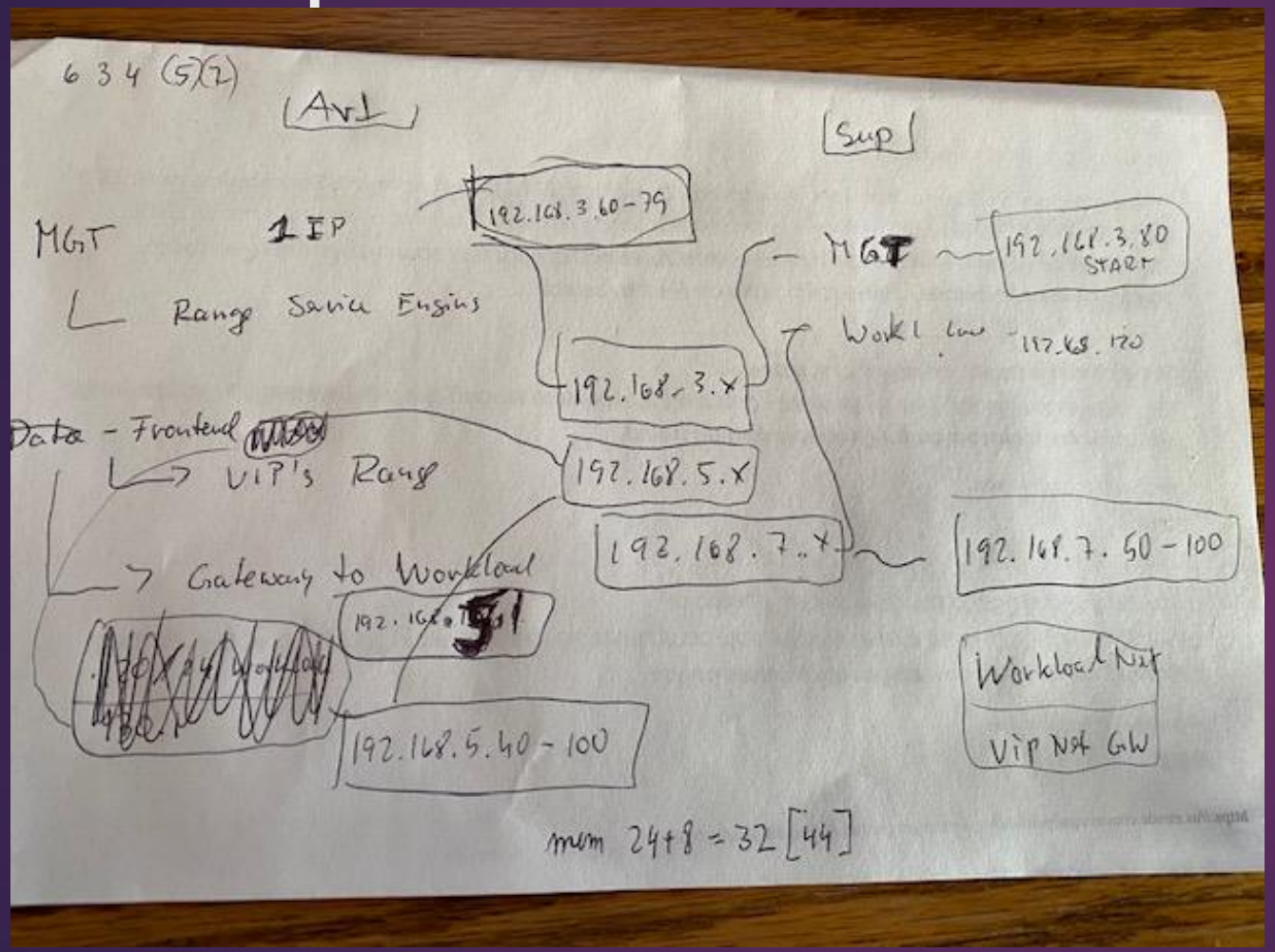
# Deploy with AVI

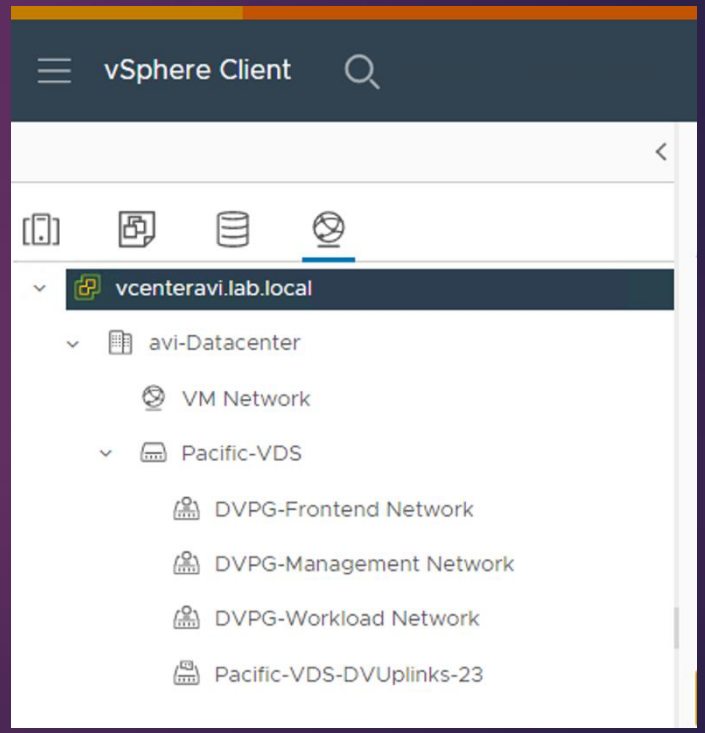
VERSION 2

# Versions used



# Set up


6 3 4 (5)(2)
Avi
Sup
MGT
1 IP
192.168.3.60-79
Range Sanic Ensins
MGT
192.168.3.80
START
Workl
192.168.3.x
Data - Frontend
VIP's Rang
192.168.5.x
192.168.7.x
192.168.7.50-100
Gateway to Workload
Workload Net
VIP Net GW
192.168.5.40-100
mem 24+8=32[44]


vSphere Client
vcenteravi.lab.local
avi-Datacenter
VM Network
Pacific-VDS
DVPG-Frontend Network
DVPG-Management Network
DVPG-Workload Network
Pacific-VDS-DVUplinks-23

# My Network layout (.3)(.5)(.7)

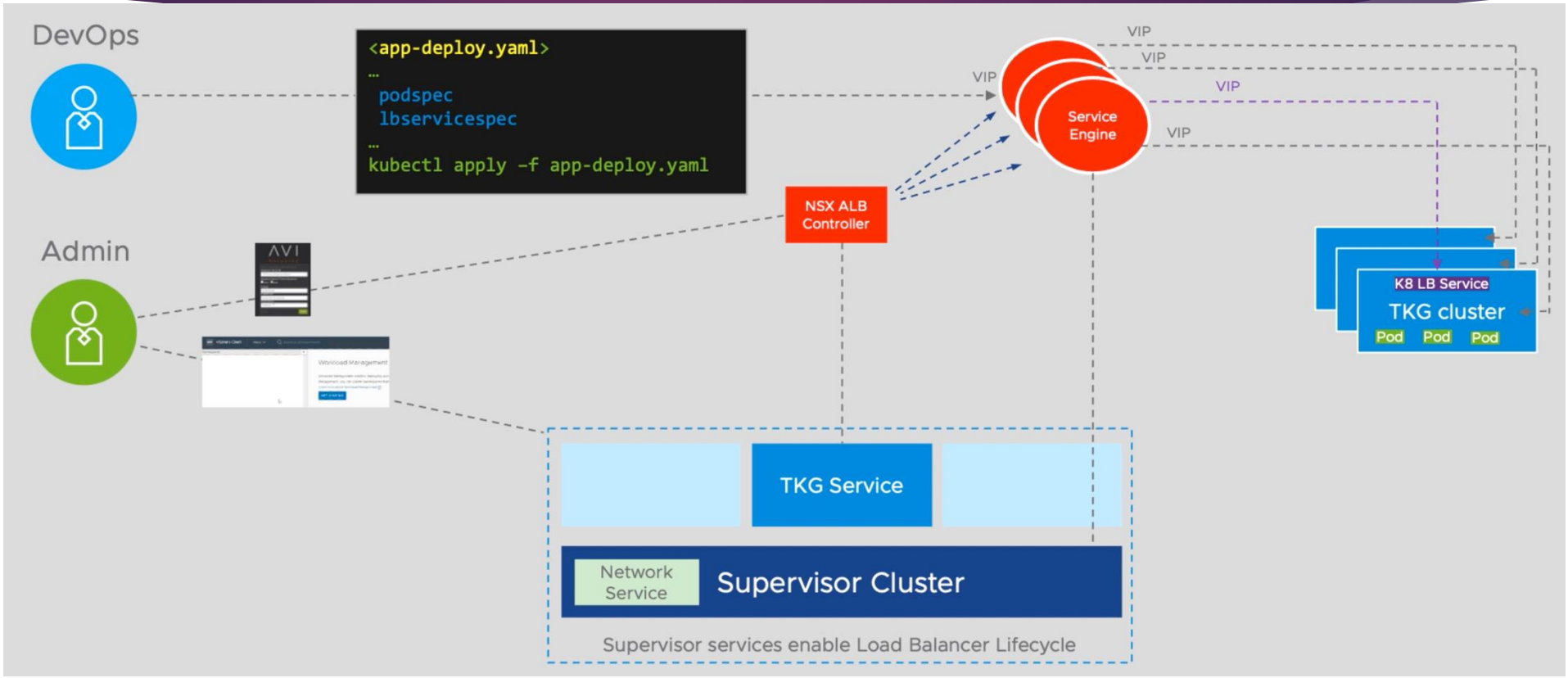
- **AVI** **Supervisor**
- 192.168.3.40 (AVI) 192.168.3.x MGT .50 vCenter .51-3 ESXi
- 192.168.3.60-79 MGT 192.168.3.80 (Start)
- 192.168.5.x Front
- 192.168.5.40-100 Front
- 192.168.7.x Workload
- Route 192.168.5.1 (192.168.7.0/24) Workload to frontend 192.168.7.50-100

# Default AVI license after install

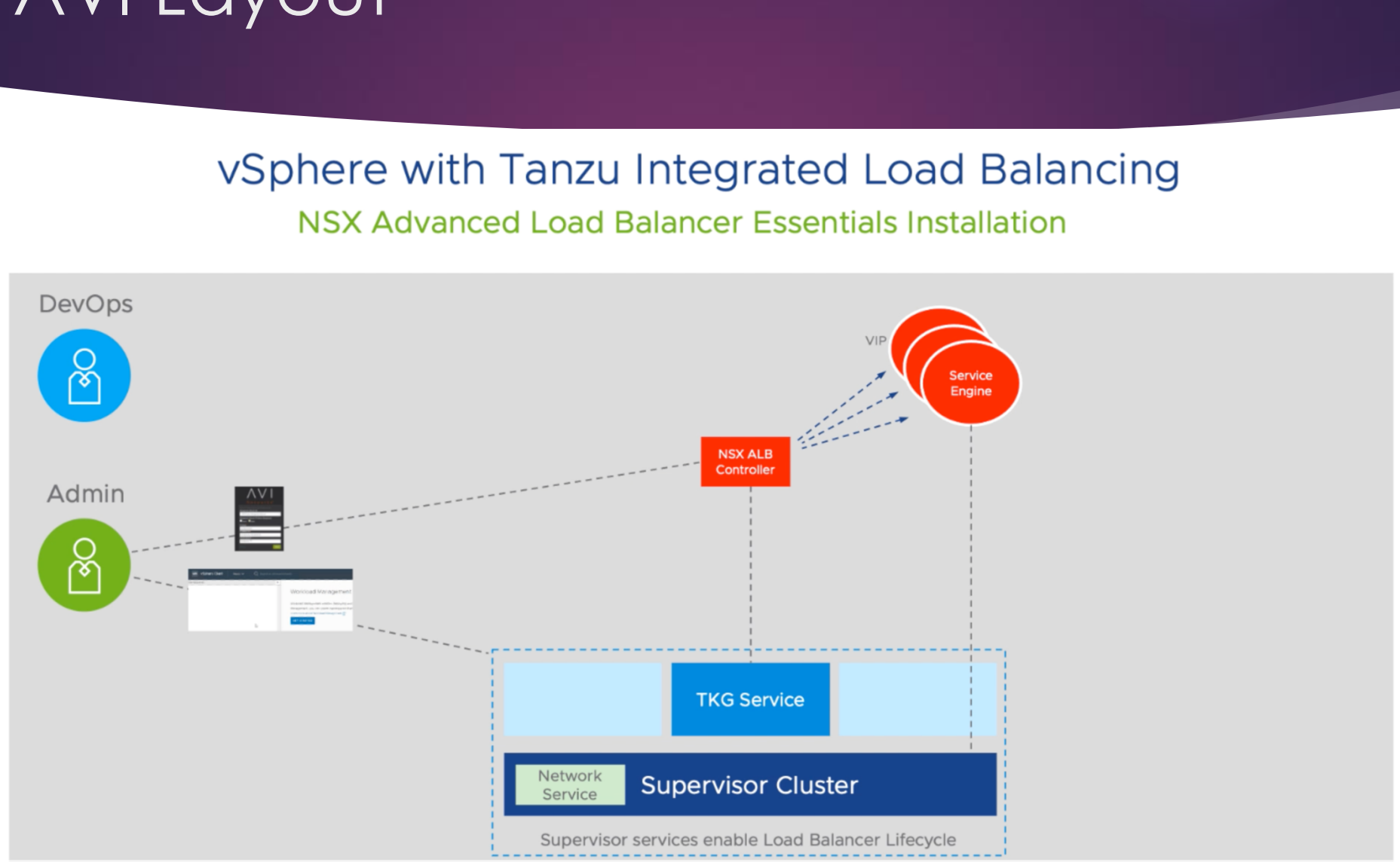
Displaying 2 items

| Description | License ID | Tier | Service Cores | SE vCPUs | SE Sockets | Start Date | Expiry |
|---|---|---|---|---|---|---|---|
| Eval | Eval | ENTERPRISE | 20 | 20 | 0 | Oct 25, 2021 | Nov 24, 2021 |
| Trial | Trial | ENTERPRISE | 2 | 2 | 0 | Oct 25, 2021 | Dec 31, 2034 |

# AVI Layout


DevOps
<app-deploy.yaml>
…
podspec
lbservicespec
…
kubectl apply -f app-deploy.yaml
Admin
AVI
NSX ALB Controller
VIP
VIP
VIP
VIP
VIP
Service Engine
K8 LB Service
TKG cluster
Pod
Pod
Pod
TKG Service
Network Service
Supervisor Cluster
Supervisor services enable Load Balancer Lifecycle

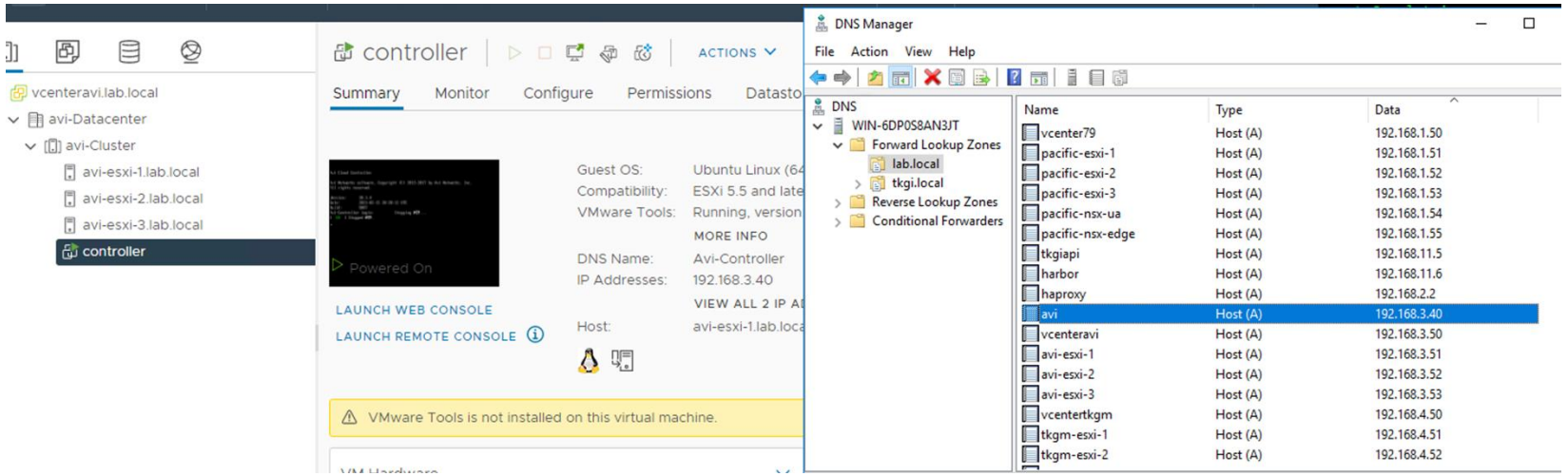
# AVI Layout


vSphere with Tanzu Integrated Load Balancing
NSX Advanced Load Balancer Essentials Installation
DevOps
Admin
VIP
Service Engine
NSX ALB Controller
TKG Service
Network Service
Supervisor Cluster
Supervisor services enable Load Balancer Lifecycle

# Deploy

- Deploy Avi Controller OVA
- Deploy vCenter OVA
- Deploy ESXi hosts (vSAN)
- In my case all automated via William Lam's Powershell scripts

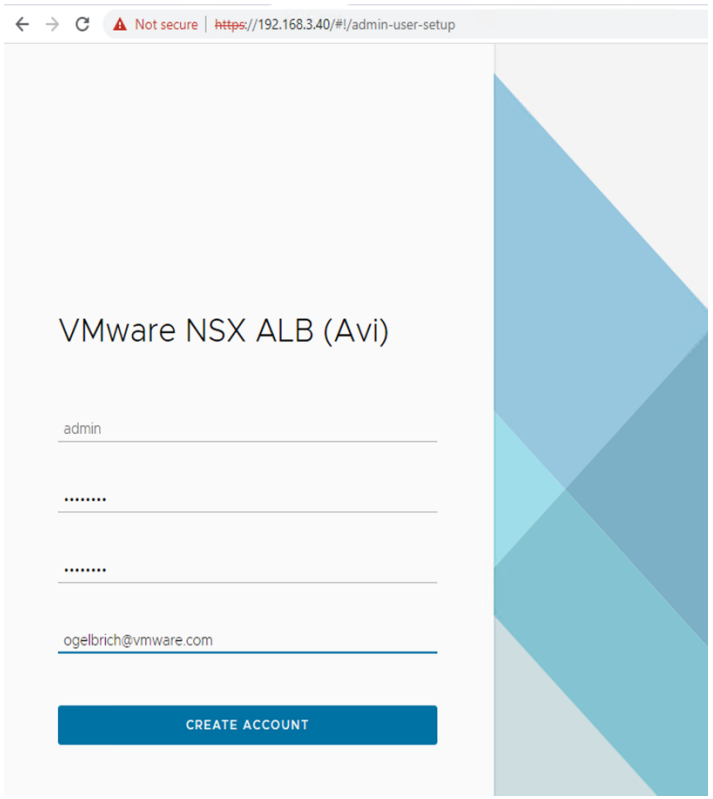
# AVI 192.168.3.40 in DNS forward and reverse

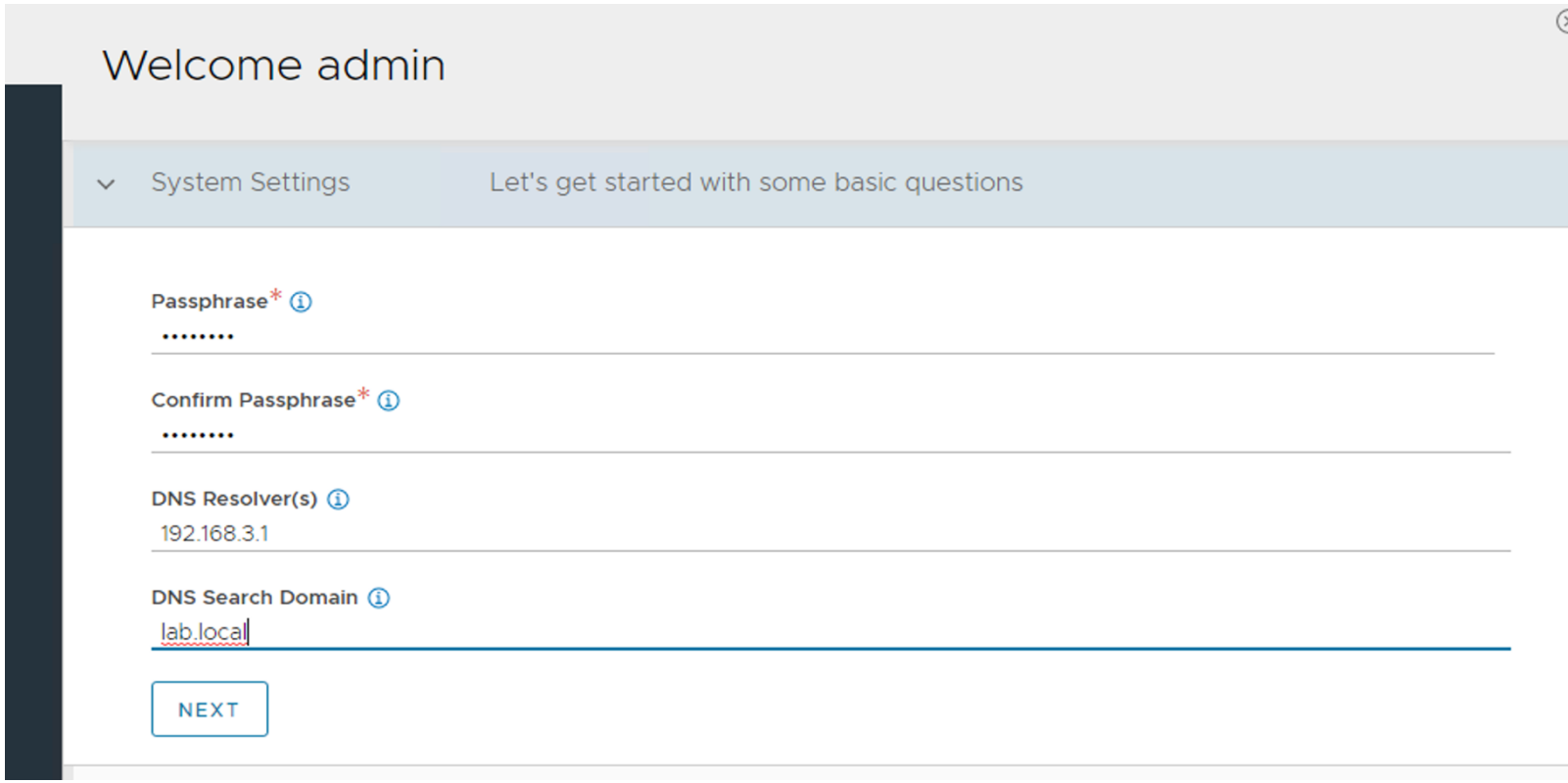


# AVI via Browser for final setup

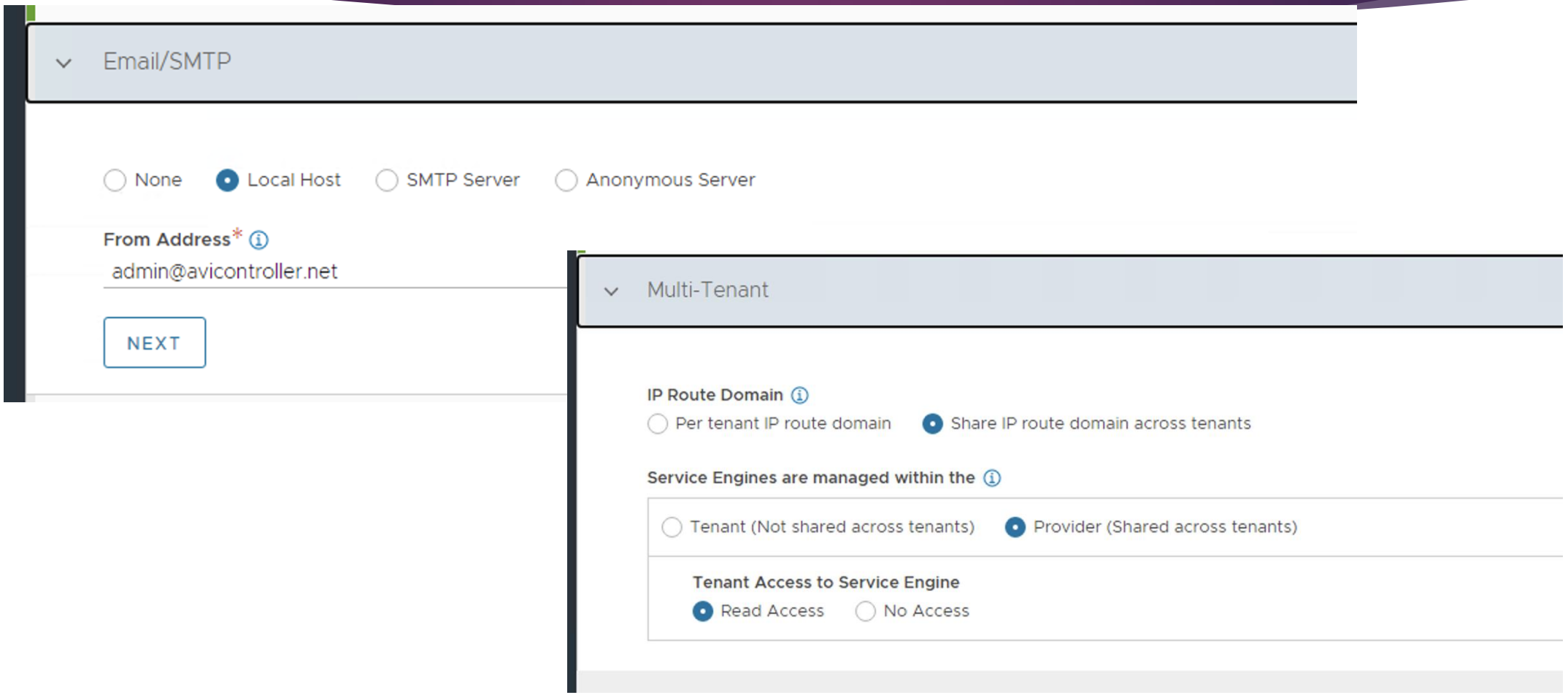
- Open browser to http://192.168.3.40
- It may take a little bit before the GUI is up!

# First time set up

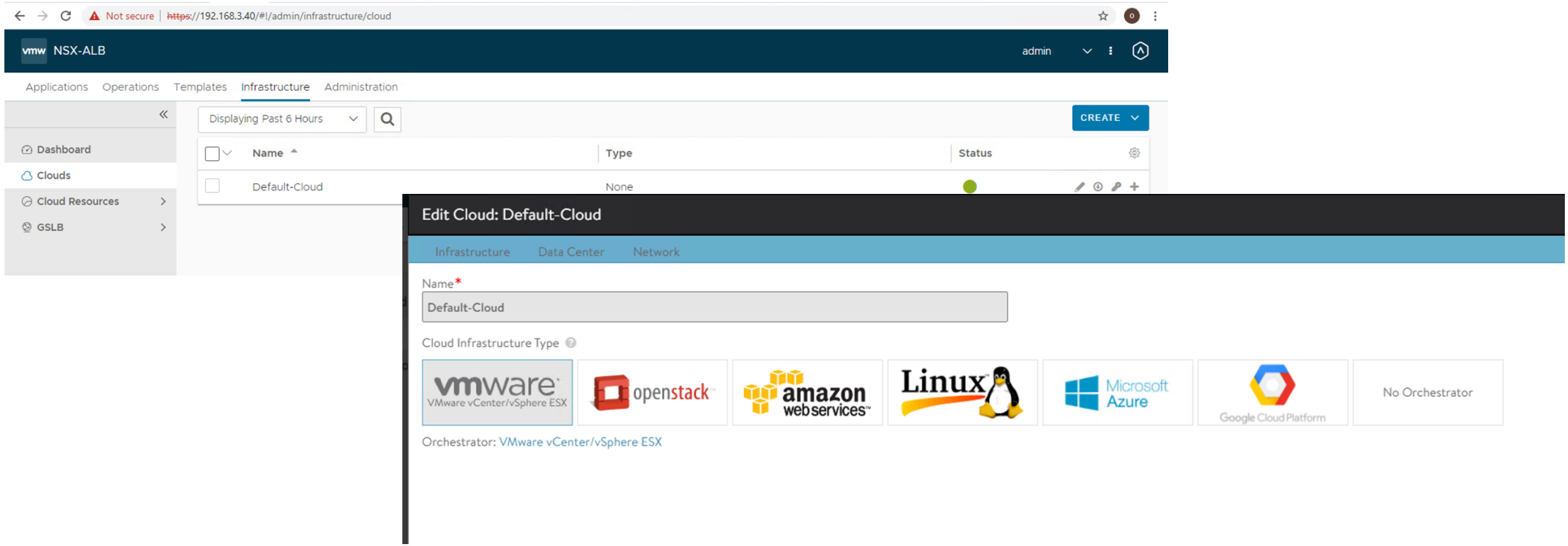


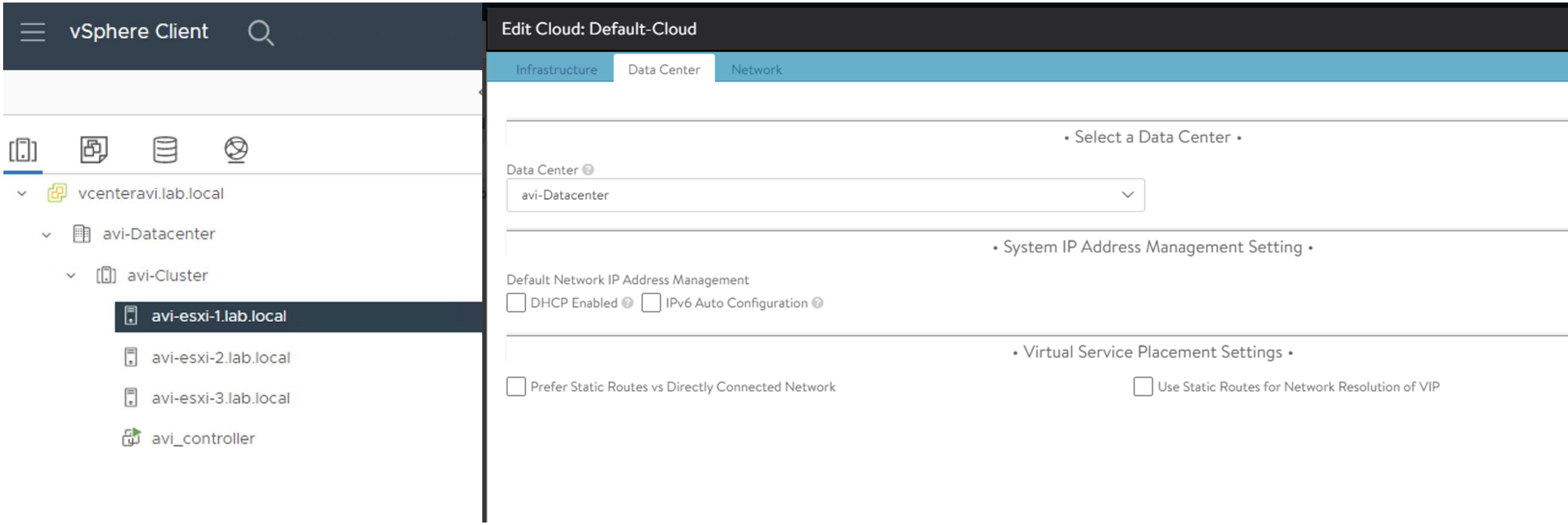
# Frist time set up



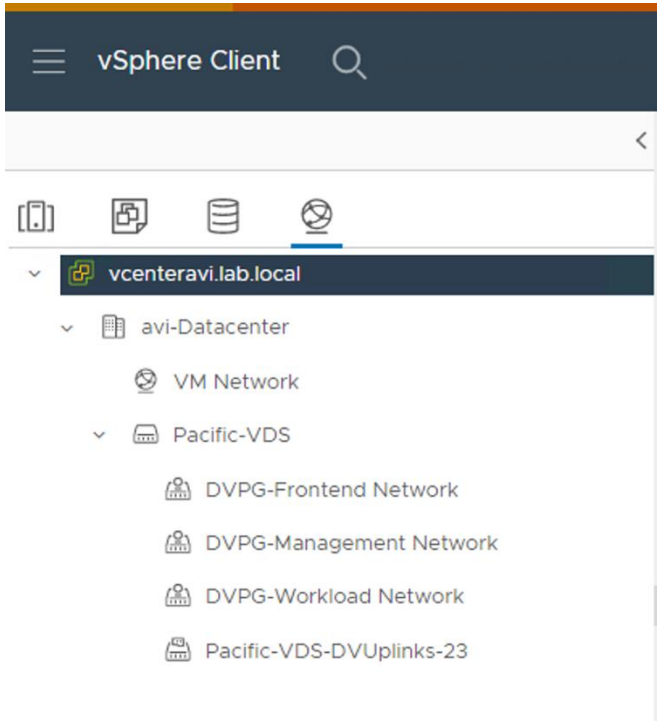
# Create Default Cloud

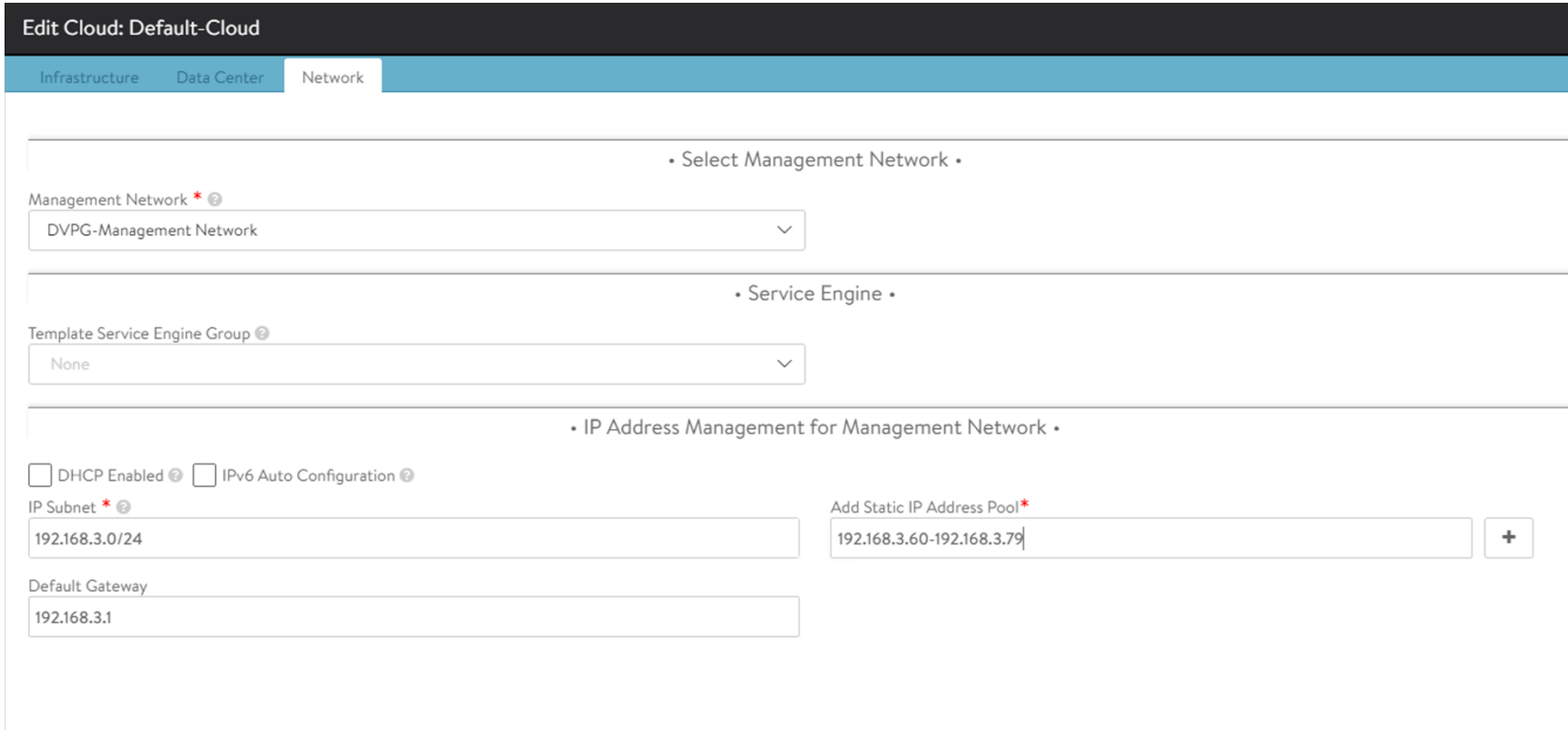


# Default Cloud Data Center

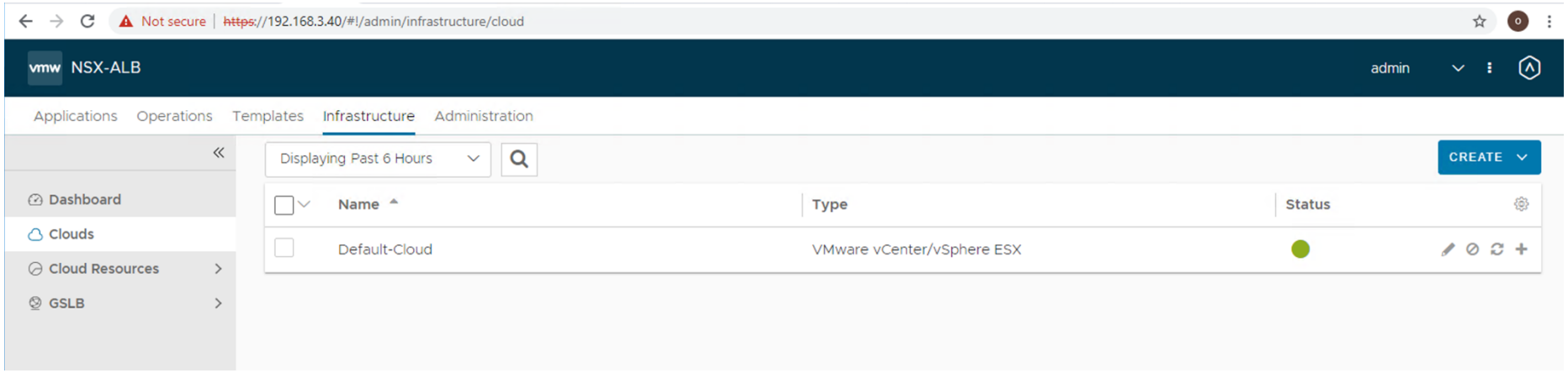


# Default Cloud Networking

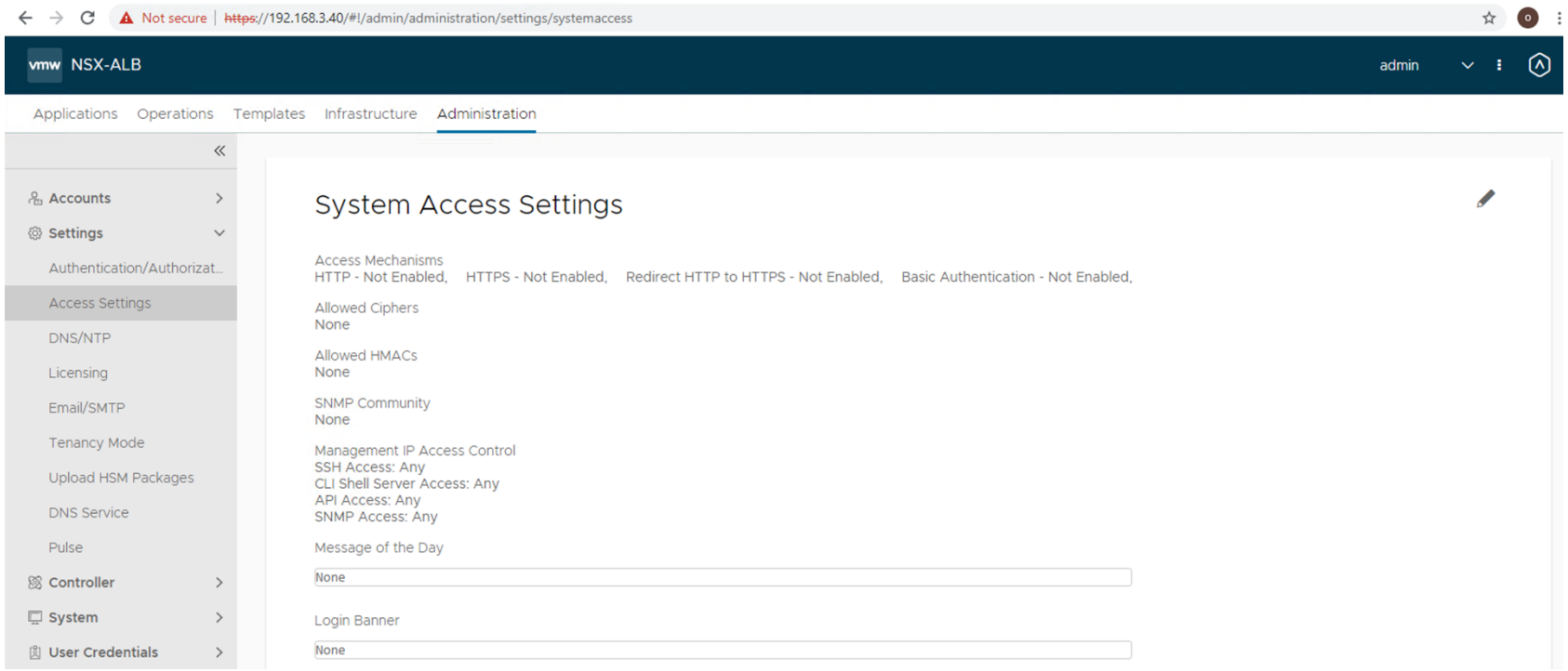


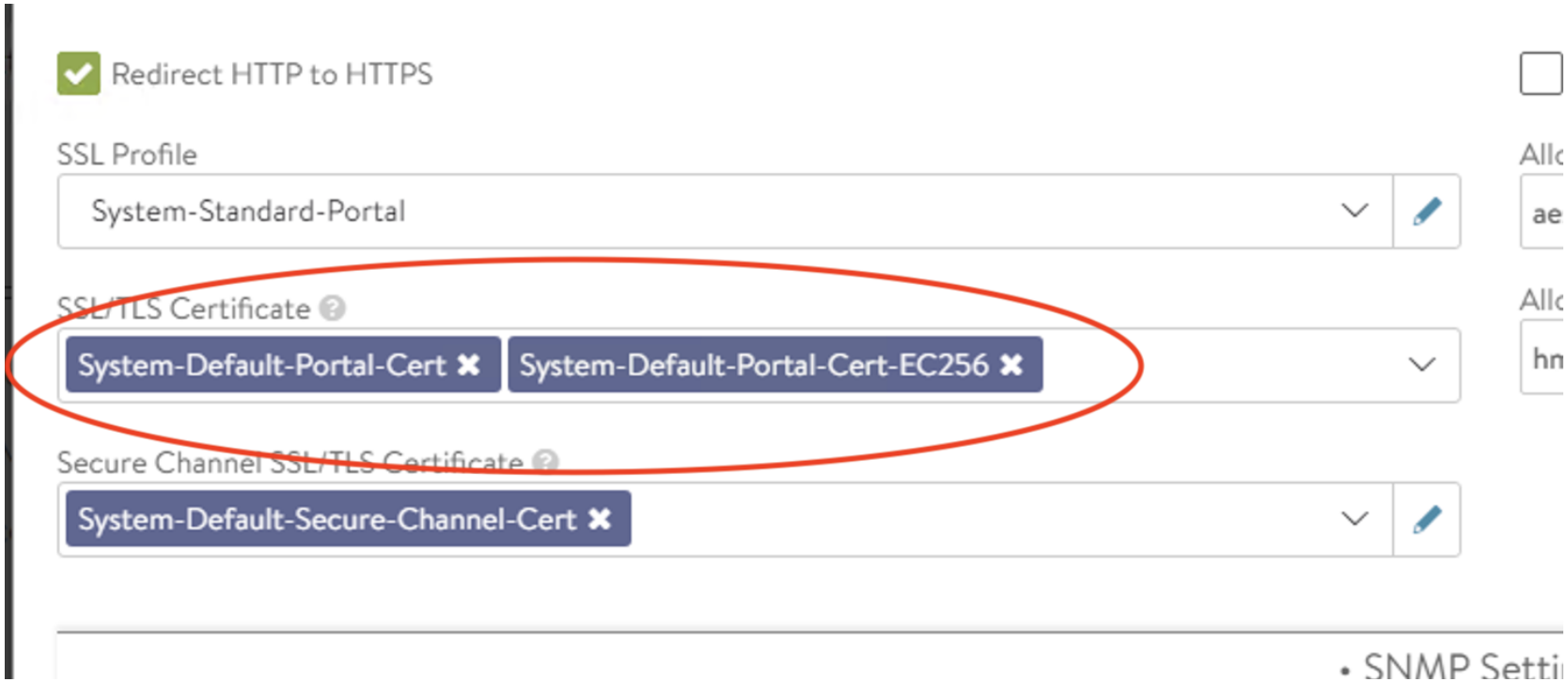
# Default Cloud is green



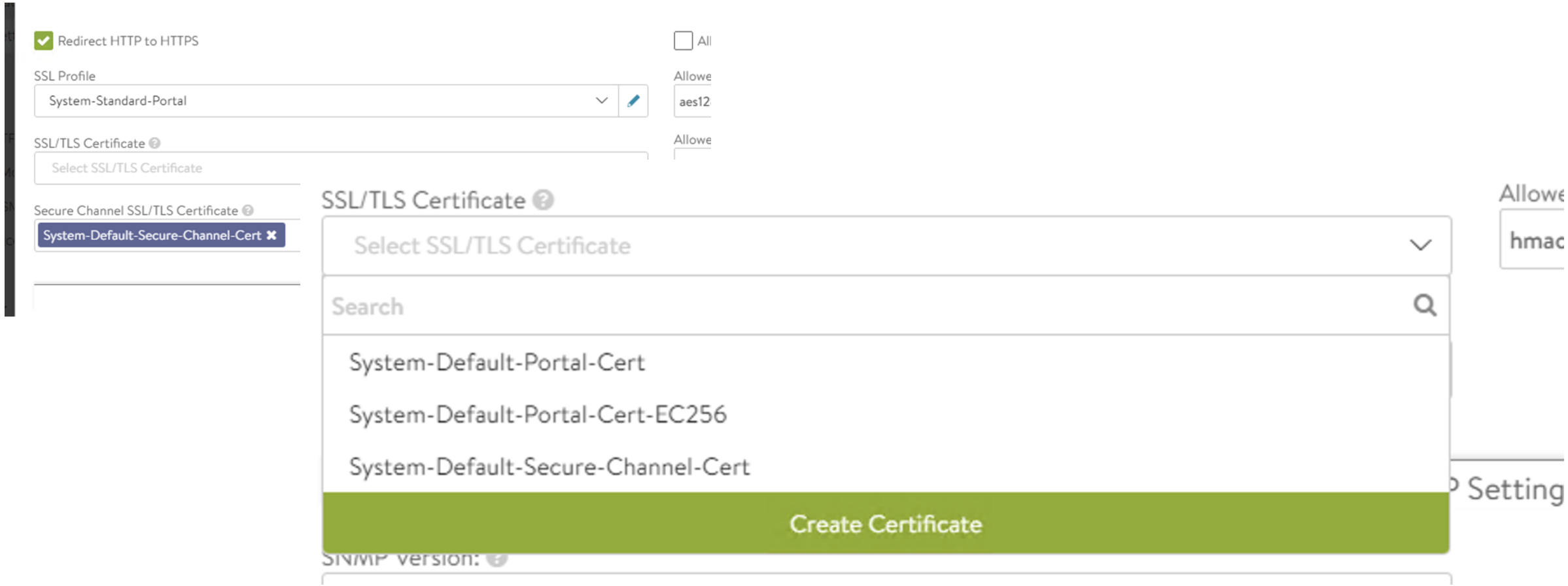
# Edit Cert (Pencil)



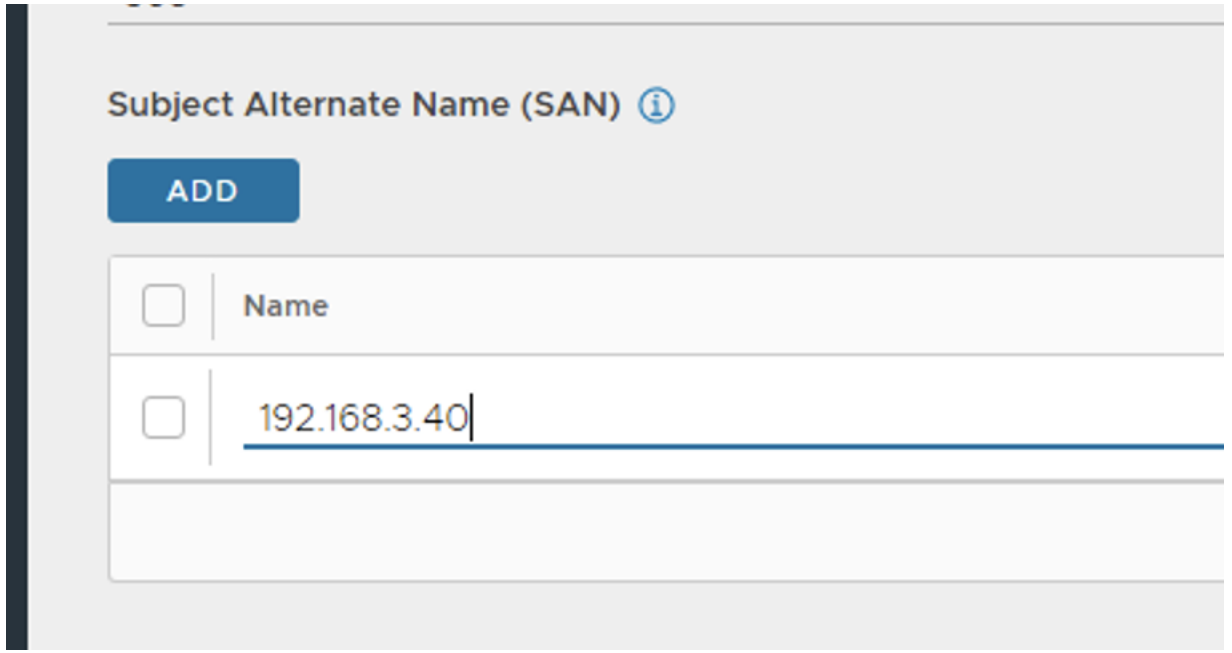
# Delete these two certs



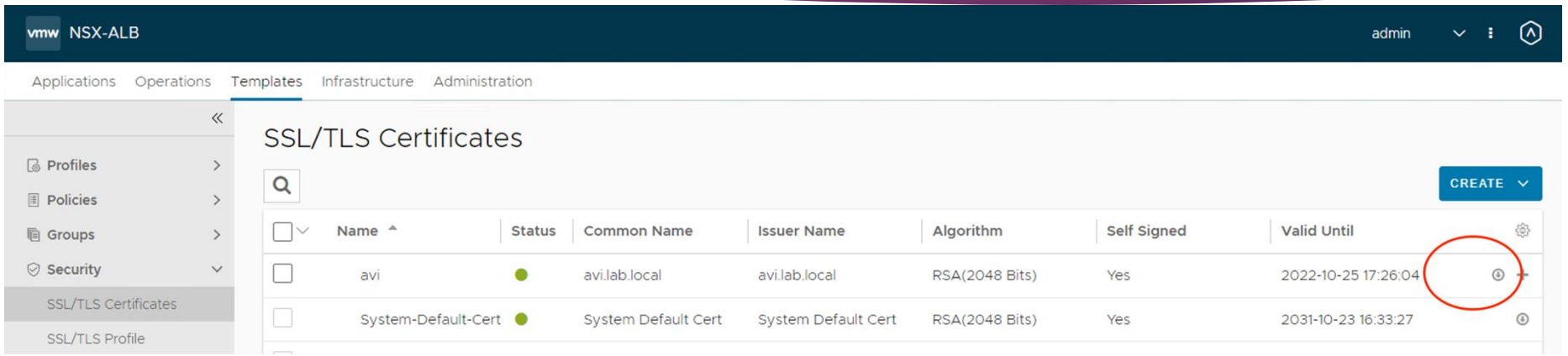
# Create new Cert

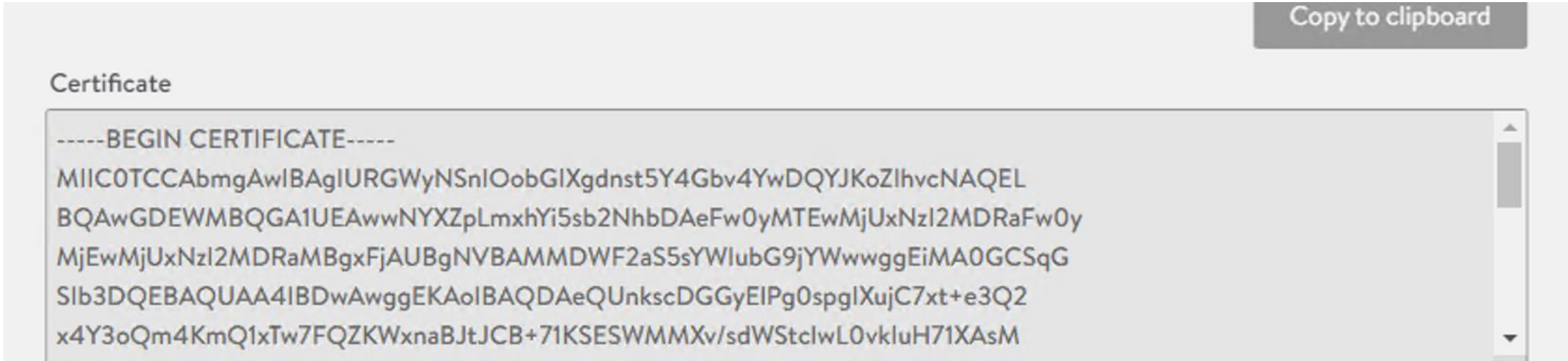


# New cert with SAN

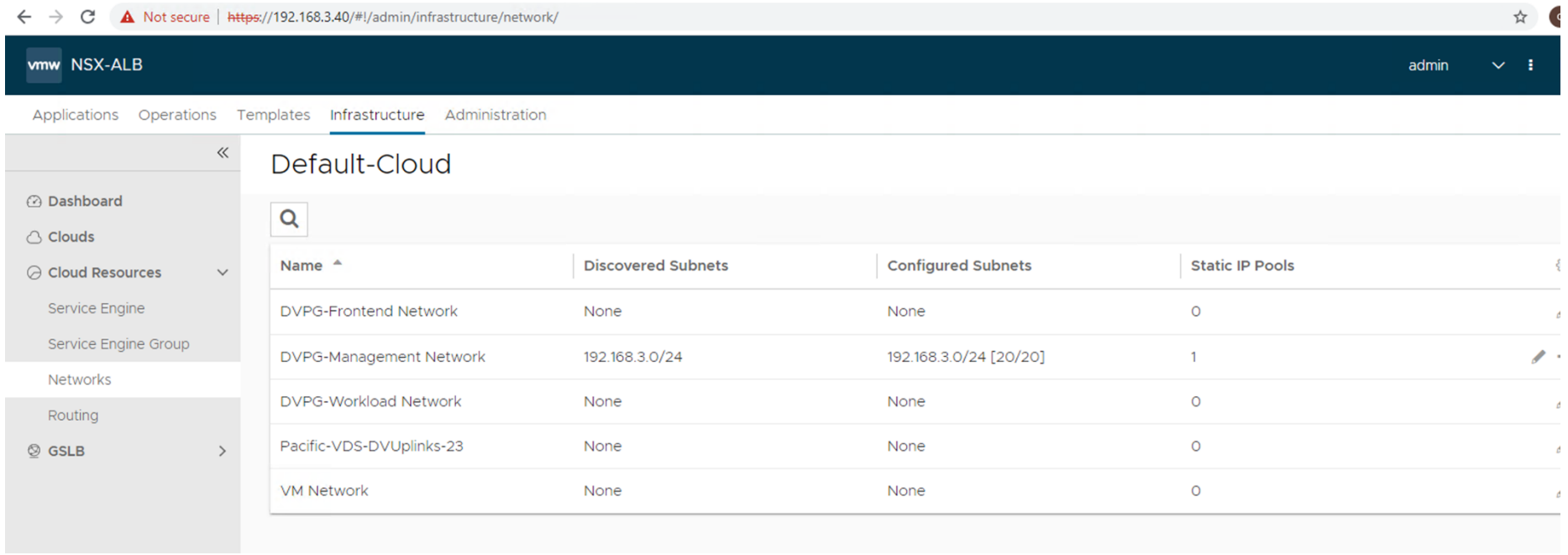




# Save of the cert for vCenter WCP enablement

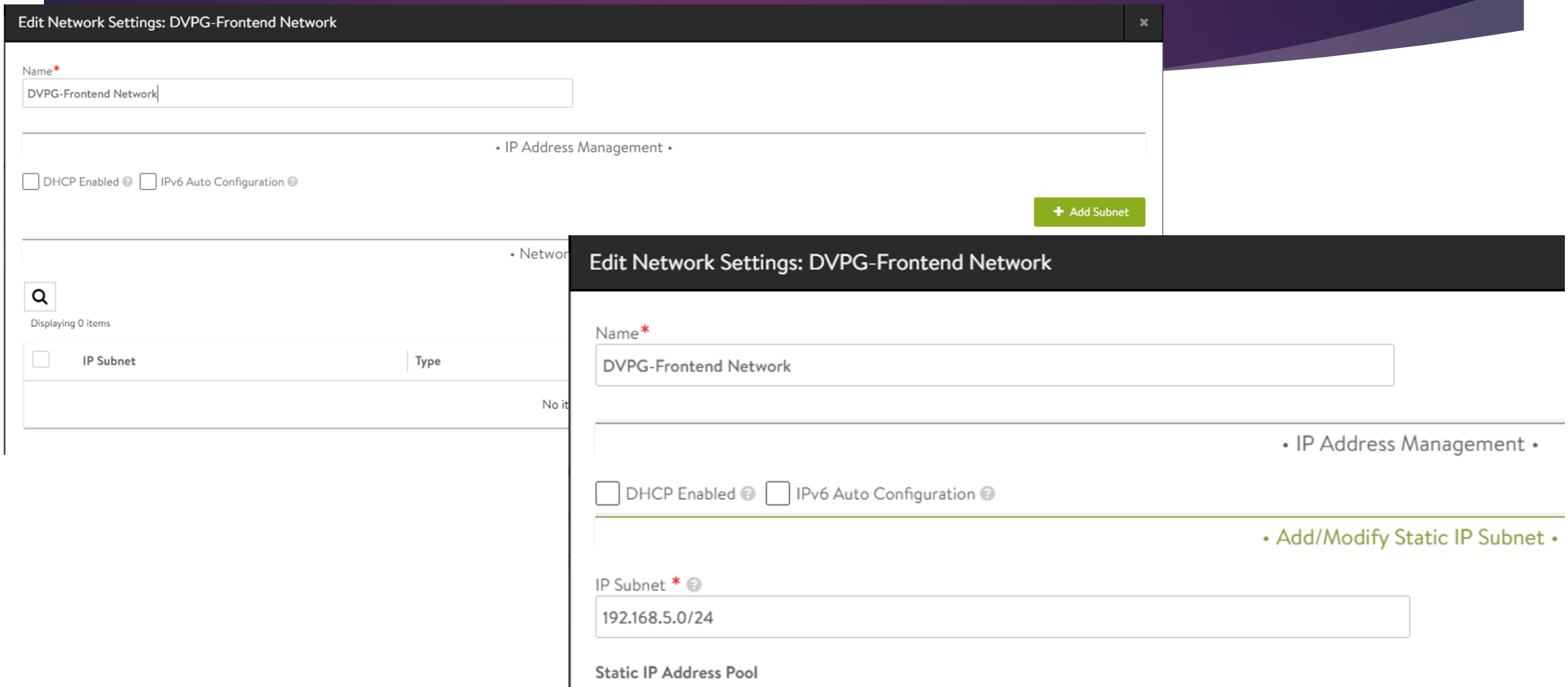


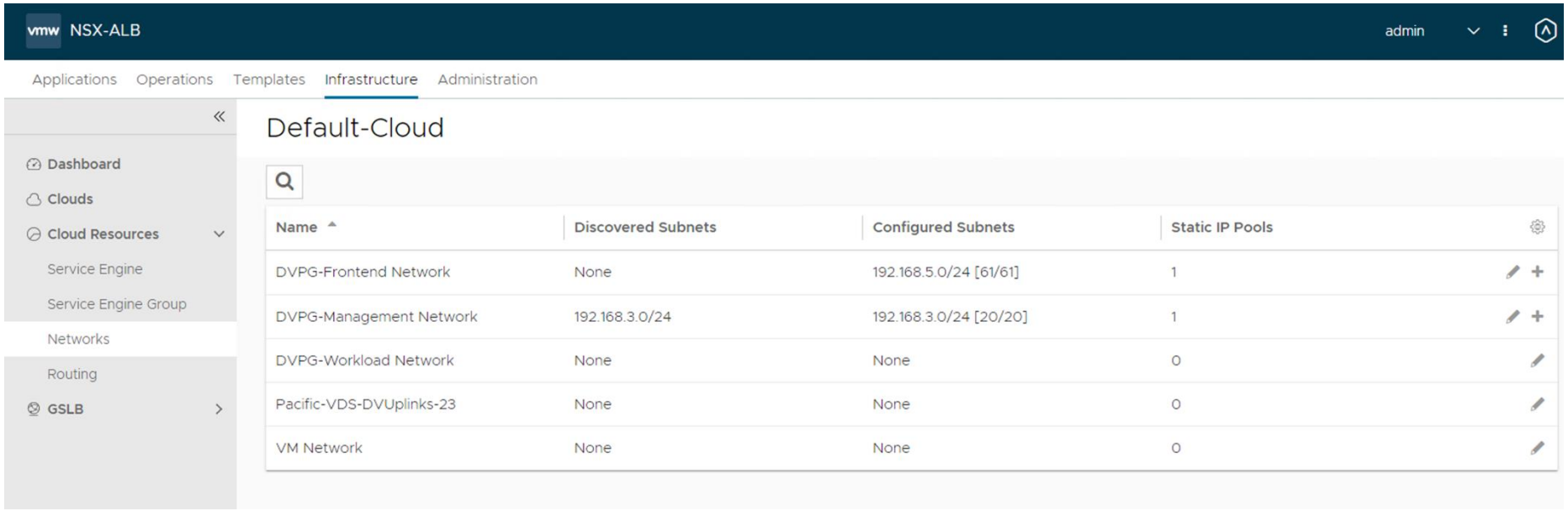
# Front end network (edit)



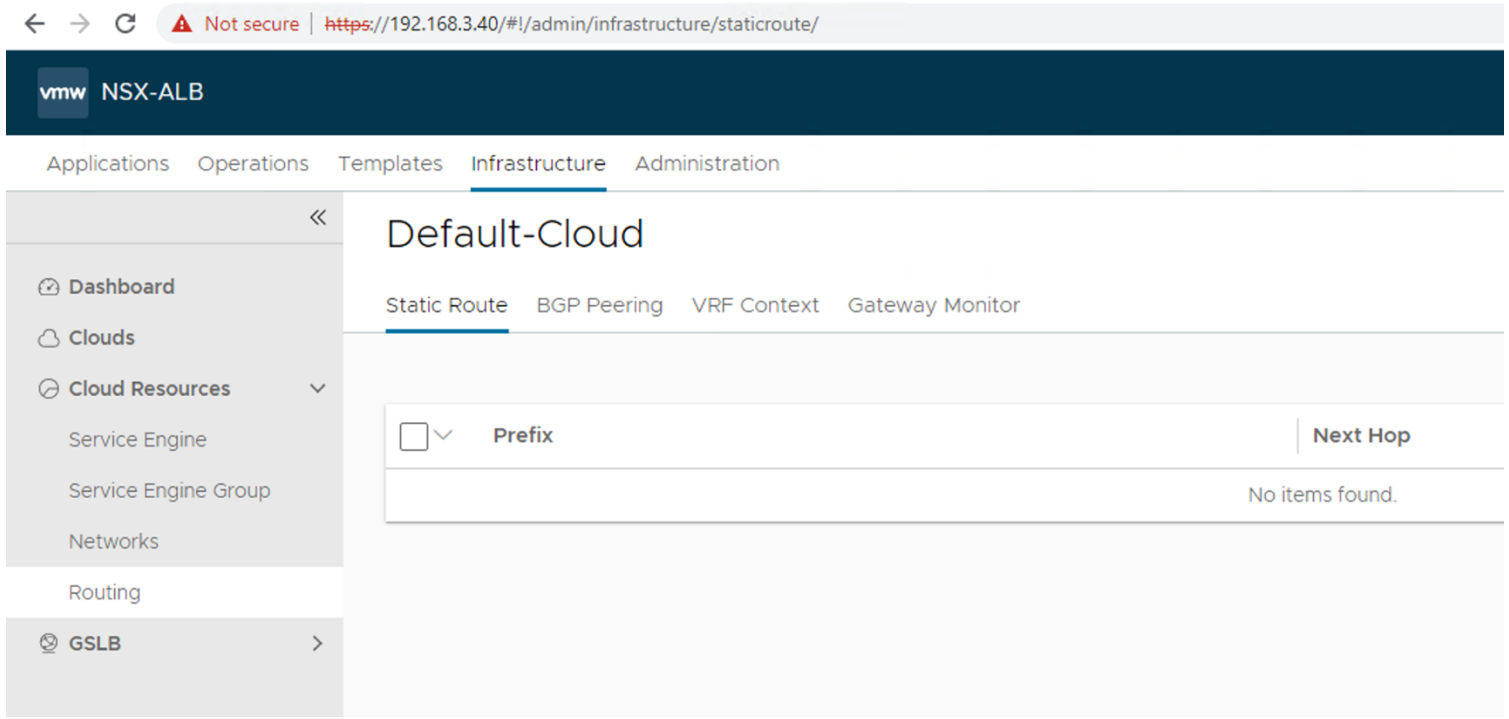
# Front end network add subnet



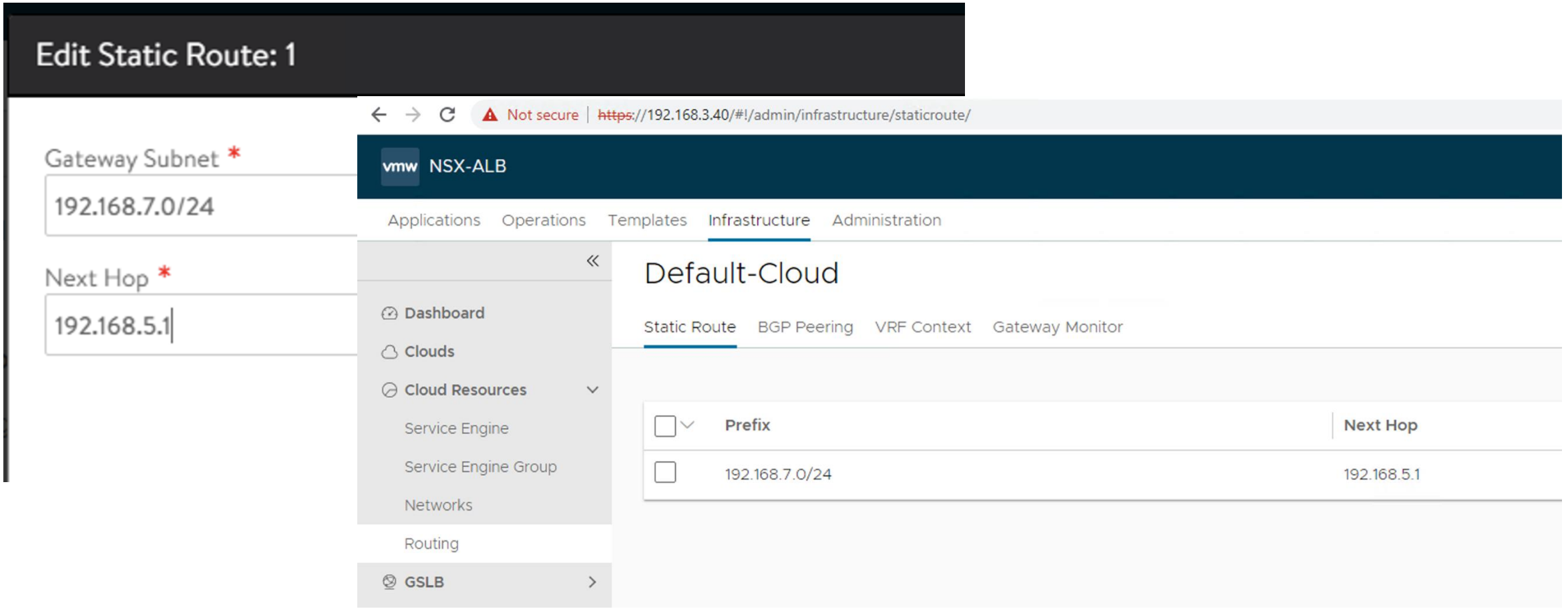
# Result


vmw NSX-ALB
admin
Applications
Operations
Templates
Infrastructure
Administration
Default-Cloud
Dashboard
Clouds
Cloud Resources
Service Engine
Service Engine Group
Networks
Routing
GSLB
Name
Discovered Subnets
Configured Subnets
Static IP Pools
DVPG-Frontend Network
None
192.168.5.0/24 [61/61]
1
DVPG-Management Network
192.168.3.0/24
192.168.3.0/24 [20/20]
1
DVPG-Workload Network
None
None
0
Pacific-VDS-DVUplinks-23
None
None
0
VM Network
None
None
0

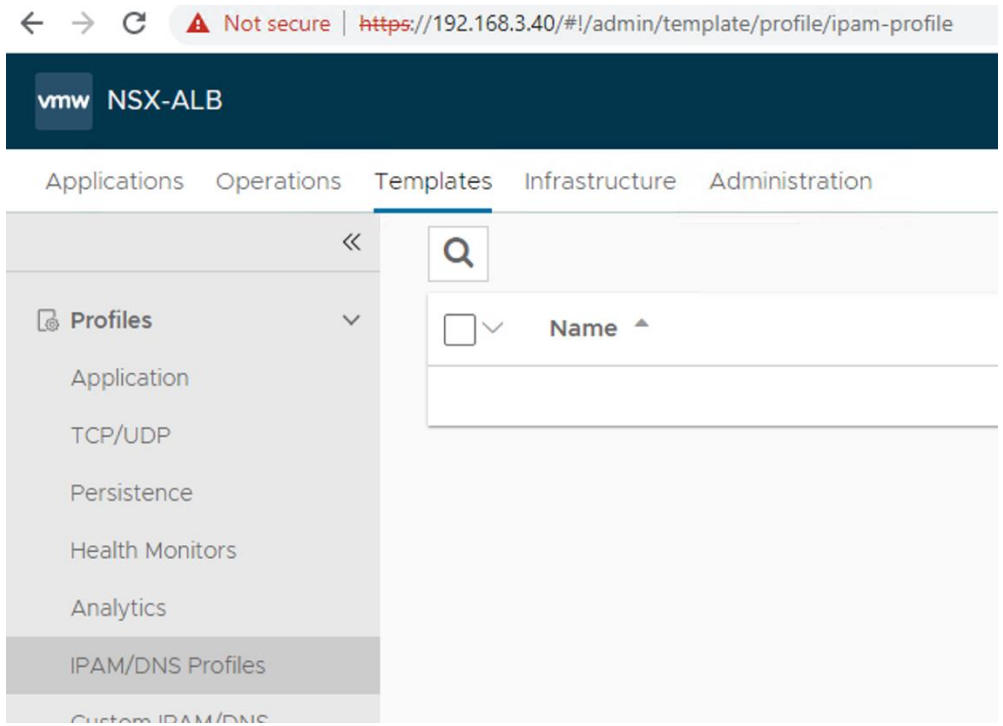
# Routing

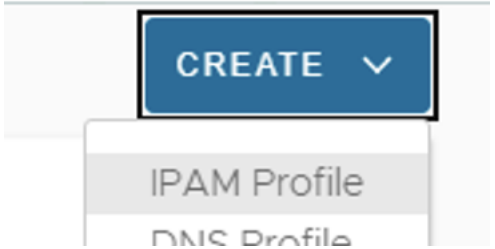

Not secure | https://192.168.3.40/#!/admin/infrastructure/staticroute/
vmw NSX-ALB
Applications
Operations
Templates
Infrastructure
Administration
Dashboard
Clouds
Cloud Resources
Service Engine
Service Engine Group
Networks
Routing
GSLB
Default-Cloud
Static Route
BGP Peering
VRF Context
Gateway Monitor
Prefix
Next Hop
No items found.

# Routing (work load to front)



# IPAM


Not secure | https://192.168.3.40/#!/admin/template/profile/ipam-profile
vmw NSX-ALB
Applications
Operations
Templates
Infrastructure
Administration
Profiles
Application
TCP/UDP
Persistence
Health Monitors
Analytics
IPAM/DNS Profiles
Name


CREATE
IPAM Profile

# IPAM

Edit IPAM/DNS Profile: my-ipam

Name *
my-ipam

Type *
Avi Vantage IPAM

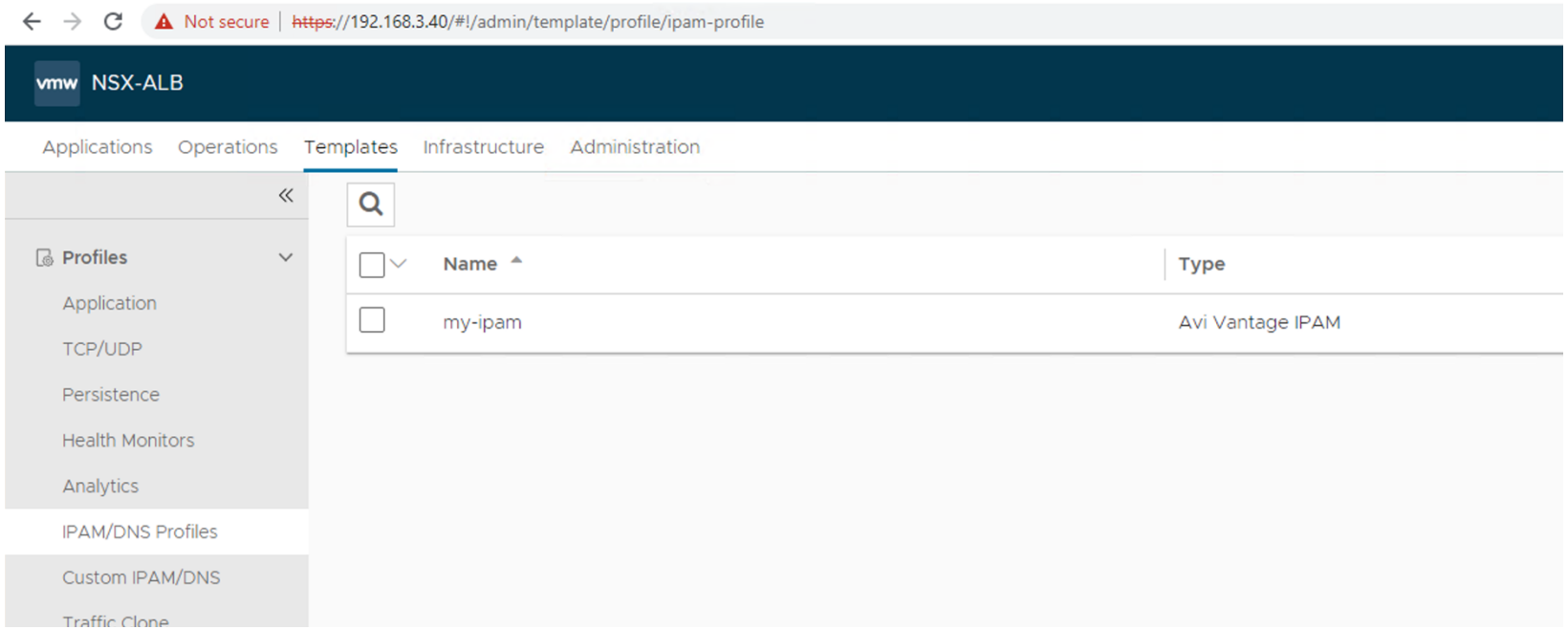
Allocate IP in VRF

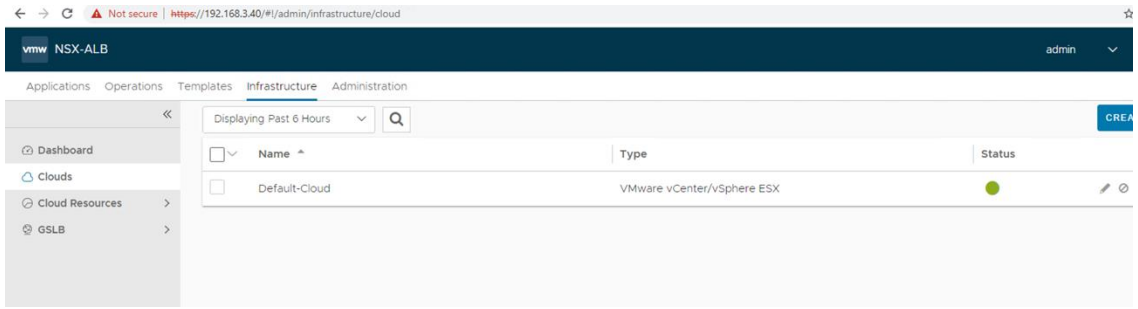
Avi Vantage IPAM Configuration

Cloud for Usable Network
Default-Cloud

Usable Network *
DVPG-Frontend Network

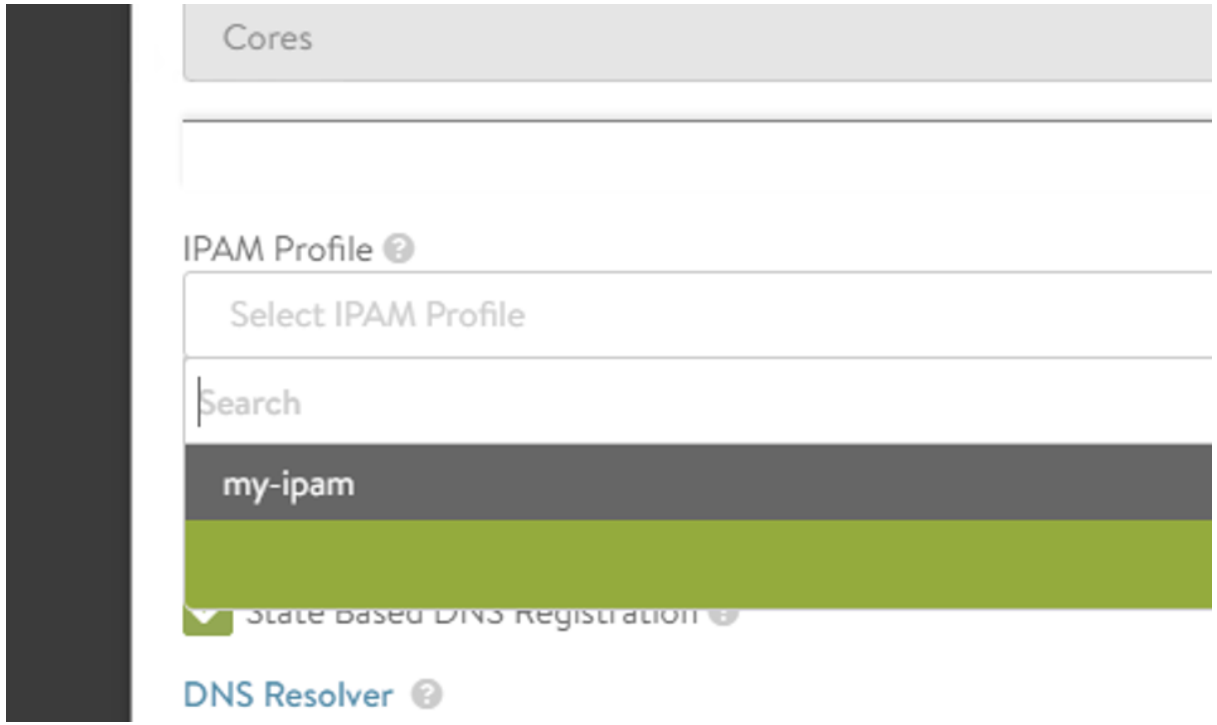
+ Add Usable Network

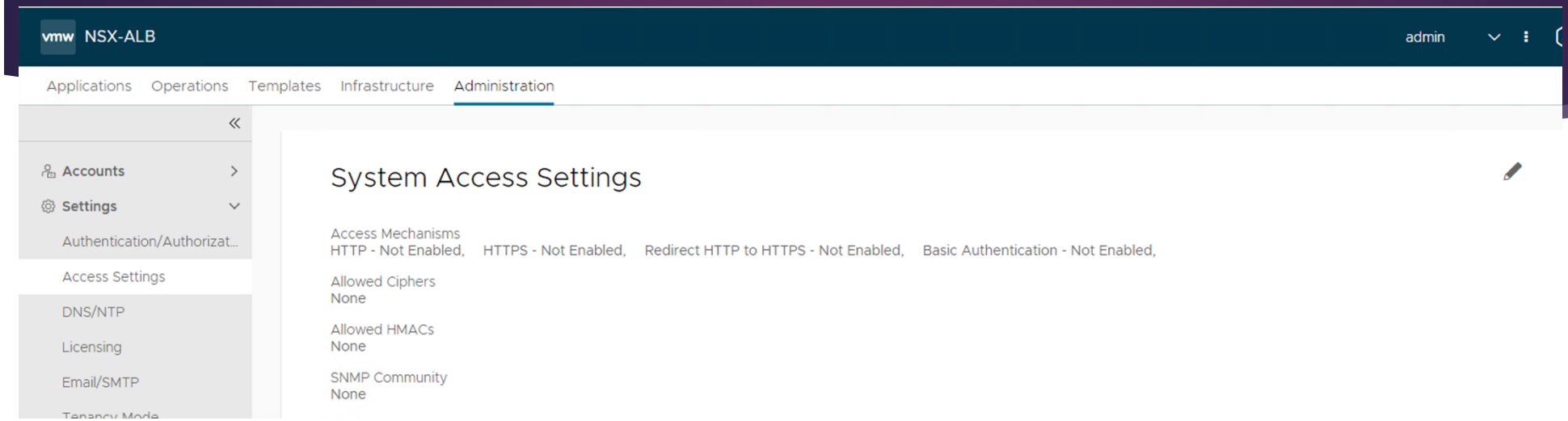





# Default Cloud edit and add IPAM

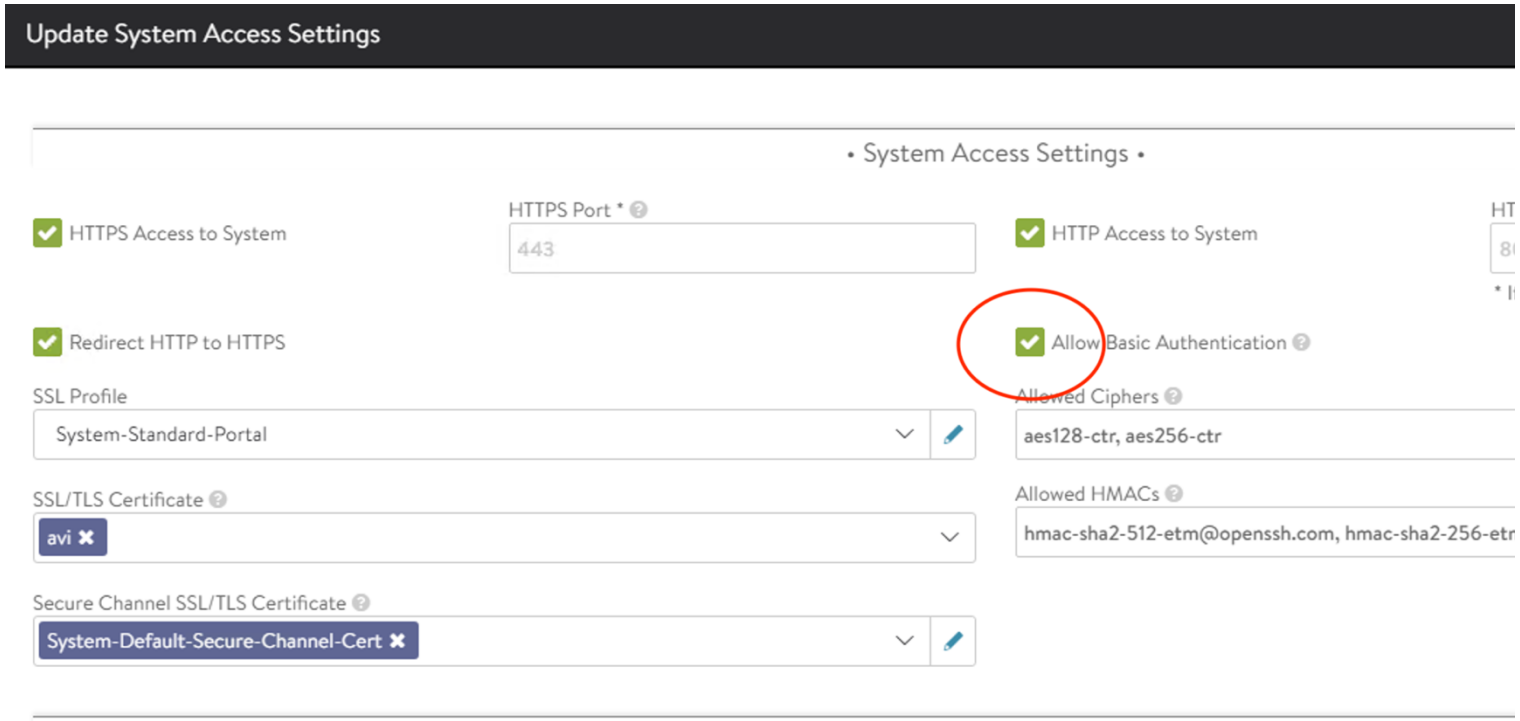
# Default Cloud edit and add IPAM



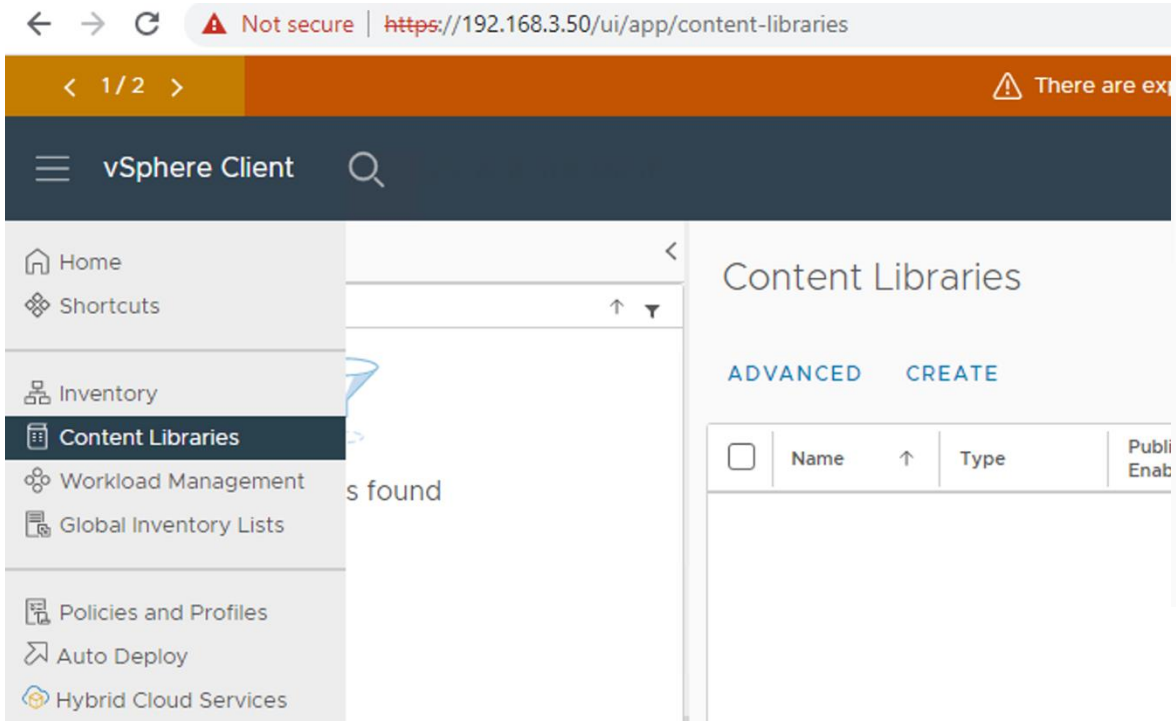


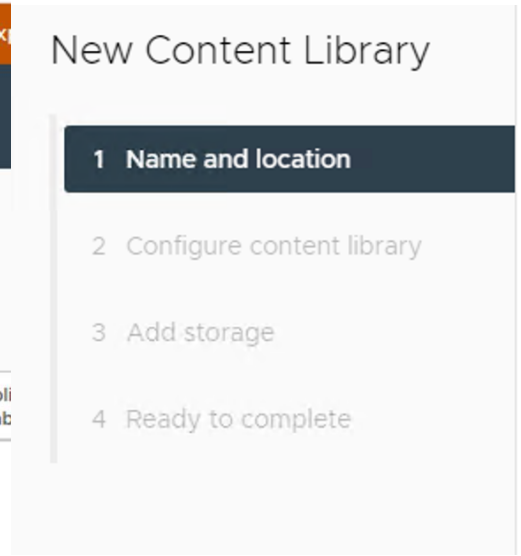

# Default Cloud edit and add IPAM

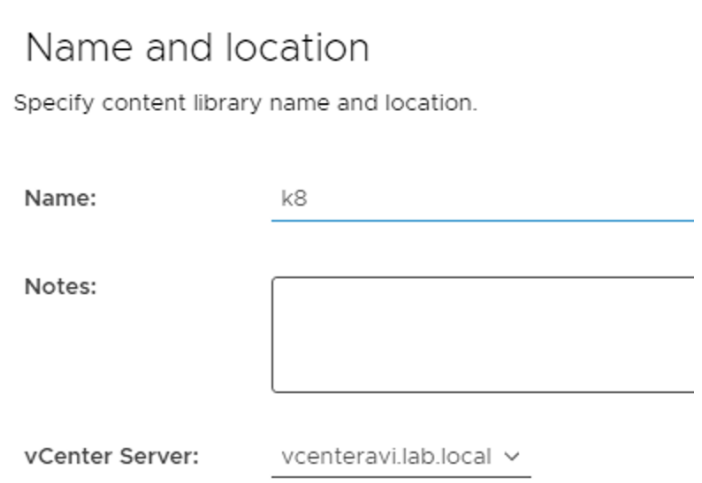
# Due to a bug in u3 make sure basic auth is enabled

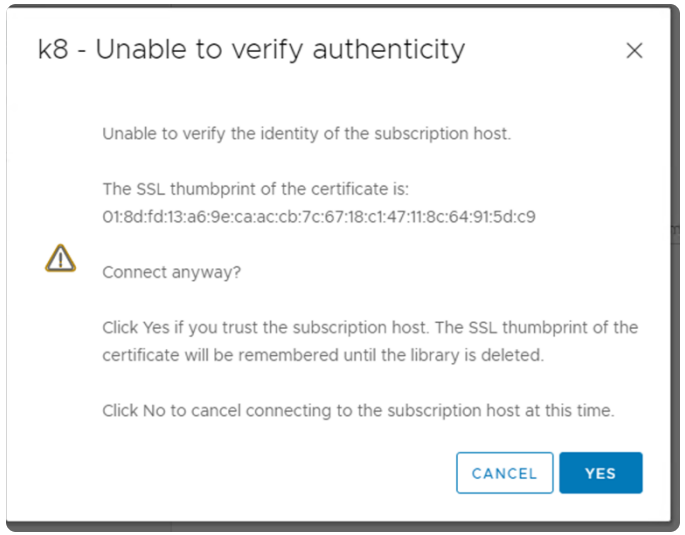


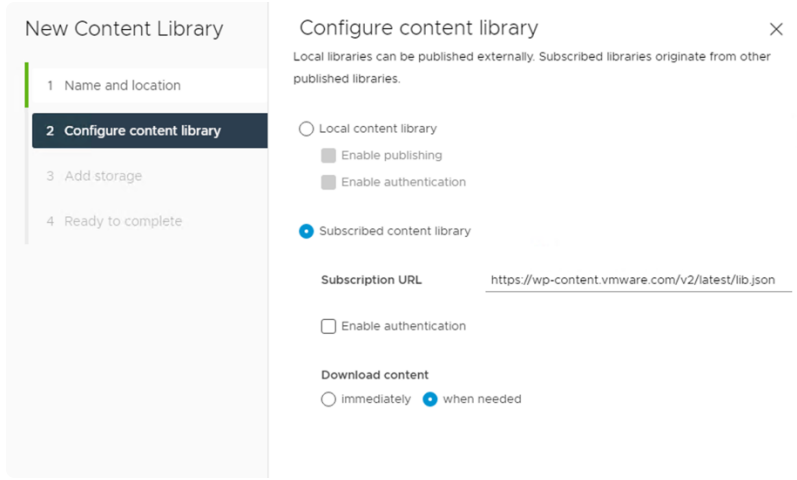
# WCP enablement add content Lib



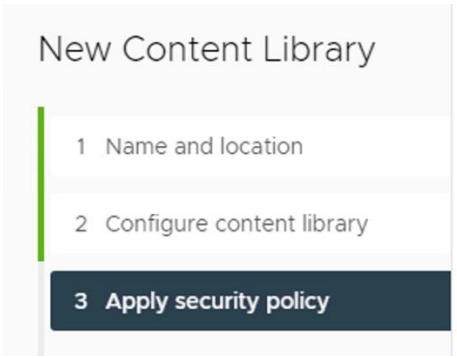



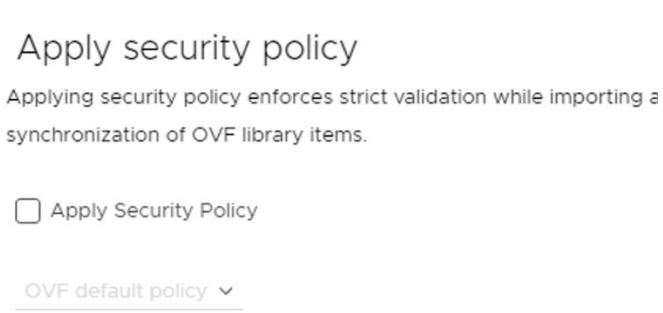



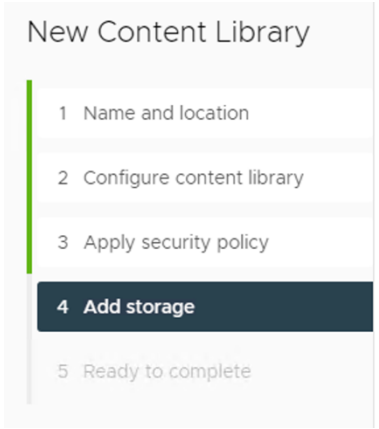


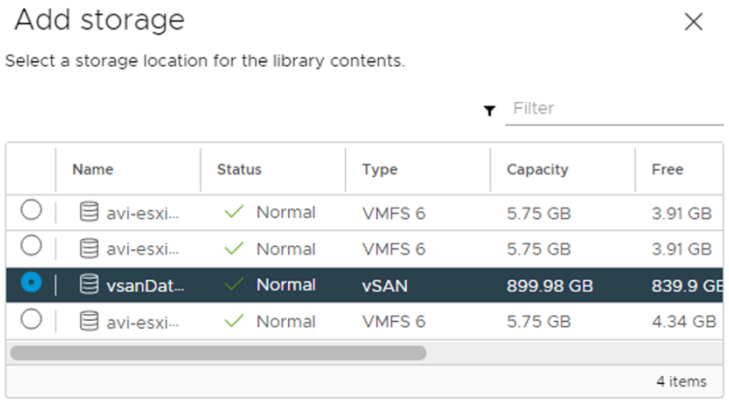




# Content lib subscription

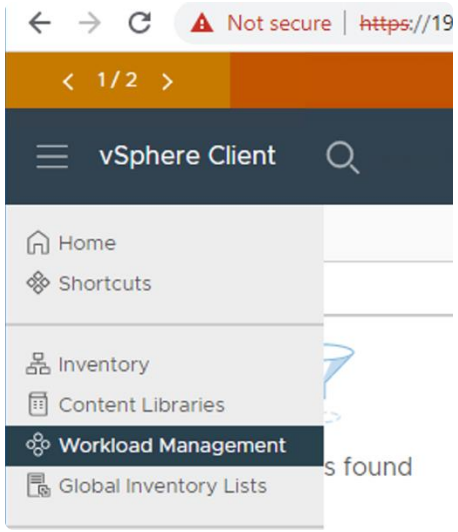


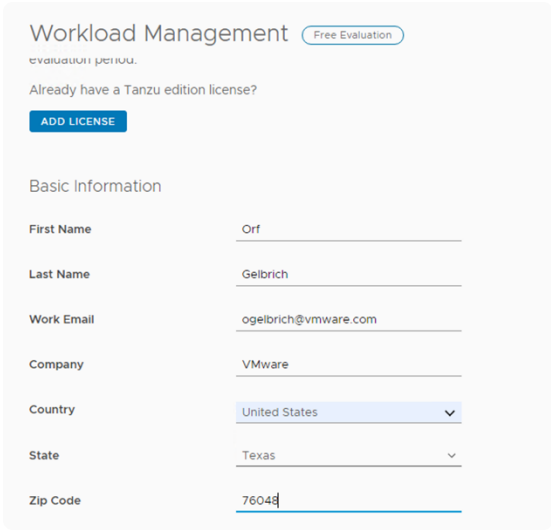



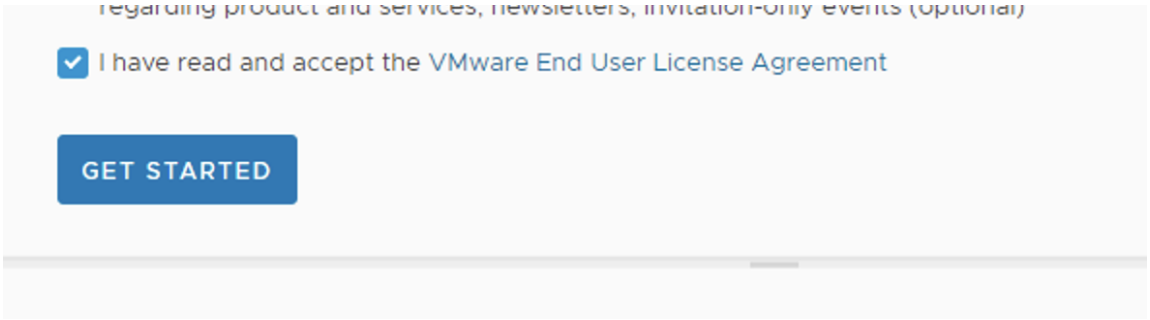



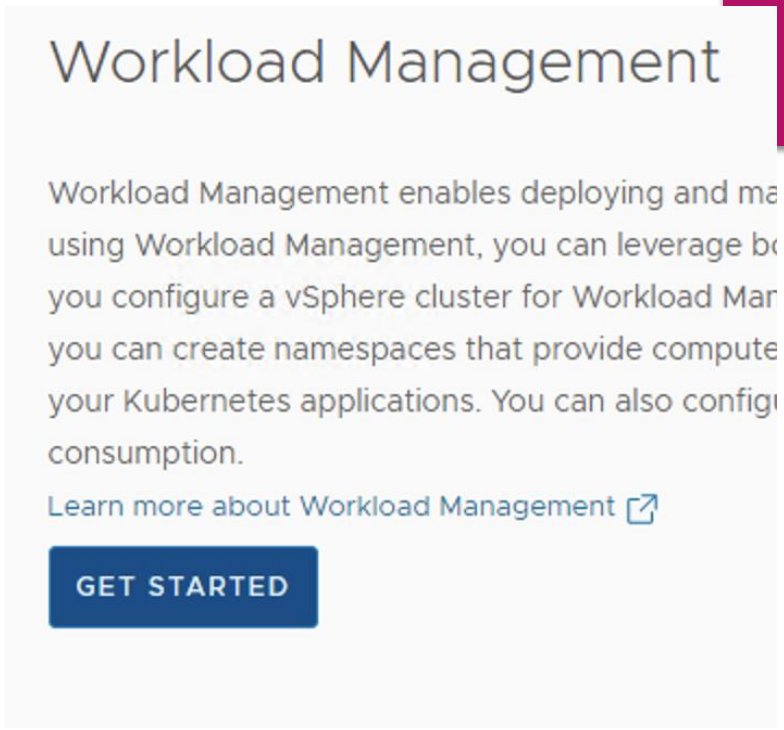

# Content lib cont…



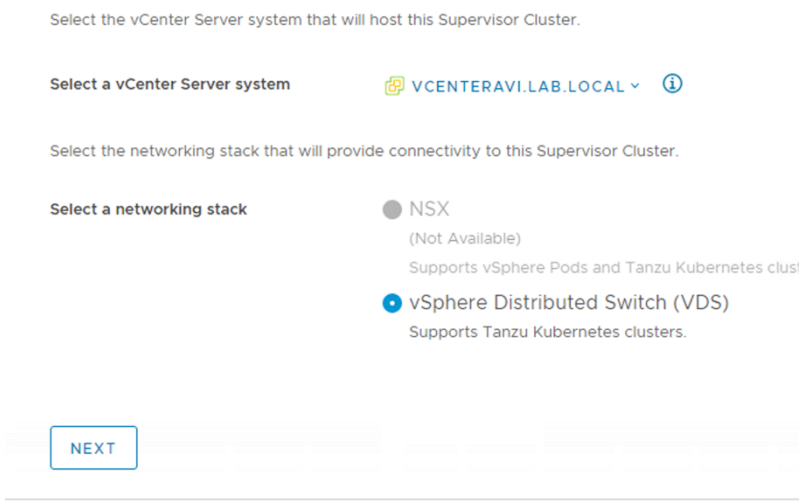



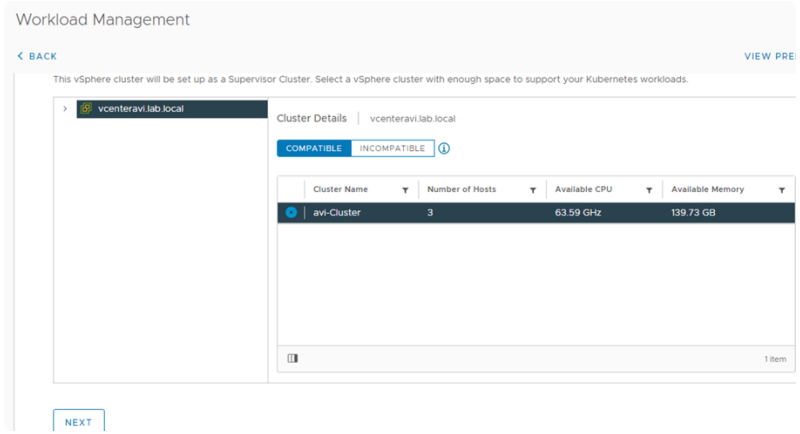

# WCP enablement



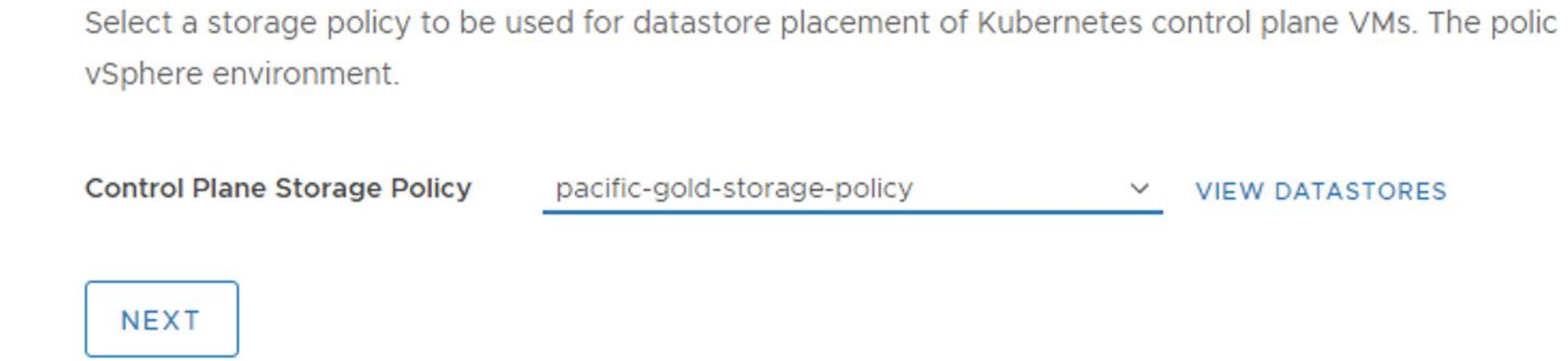

# WCP Enablement cont…



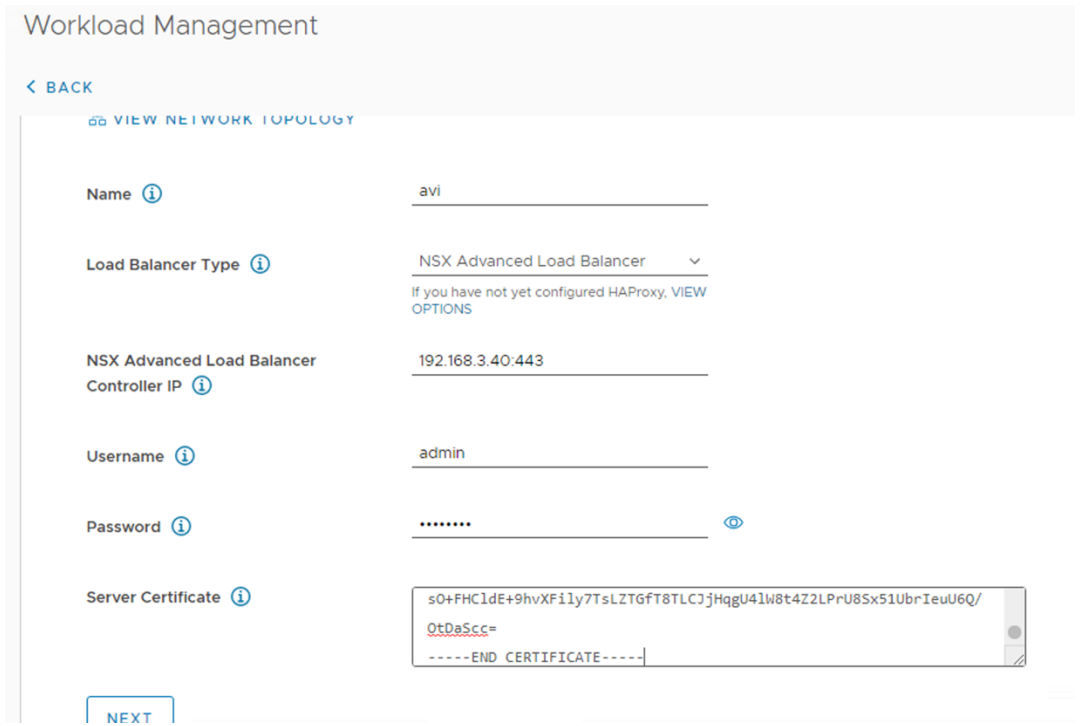




# WCP Enablement cont…

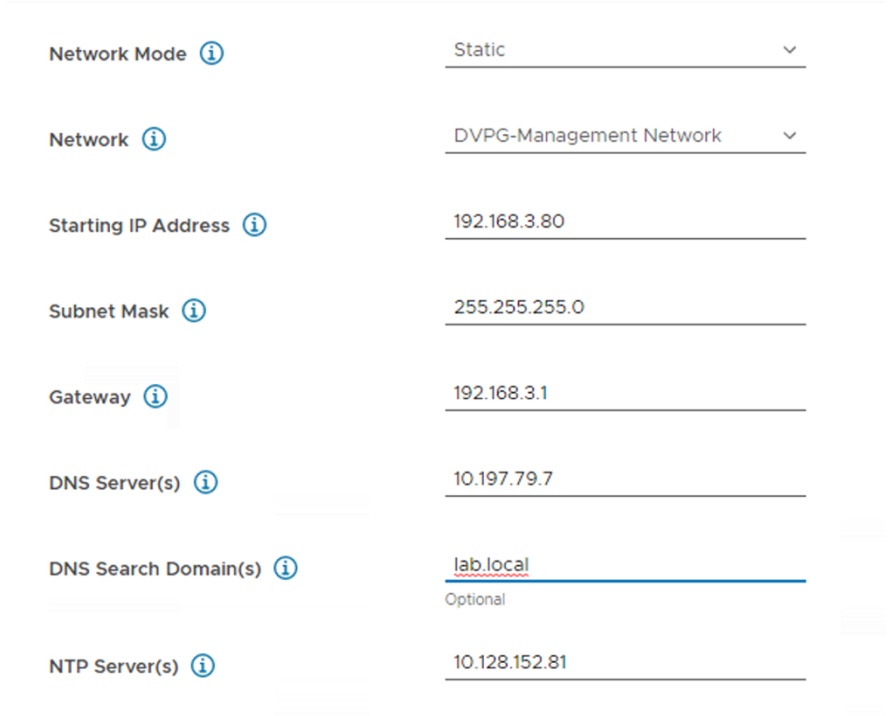
# WCP Enablement cont…

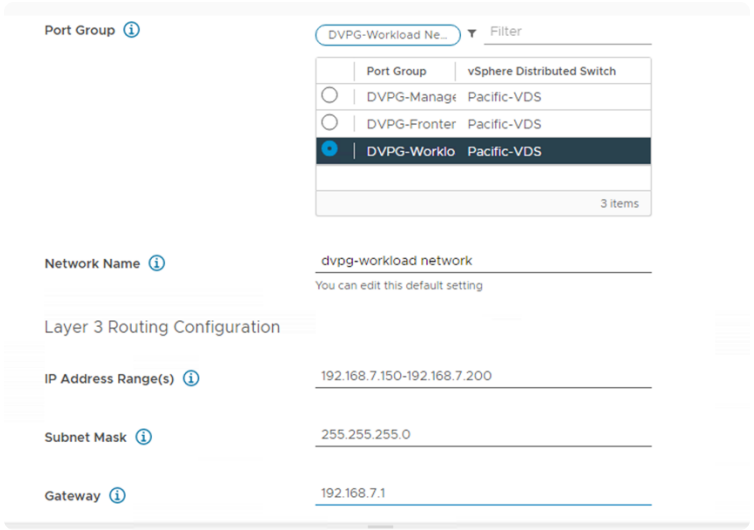


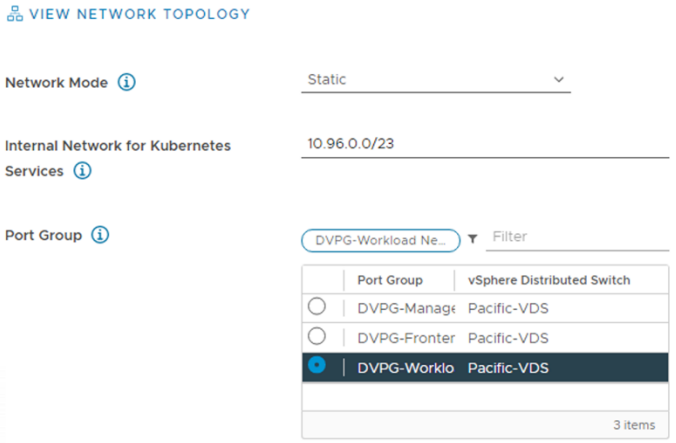
# WCP Enablement cont…



# WCP Enablement cont…

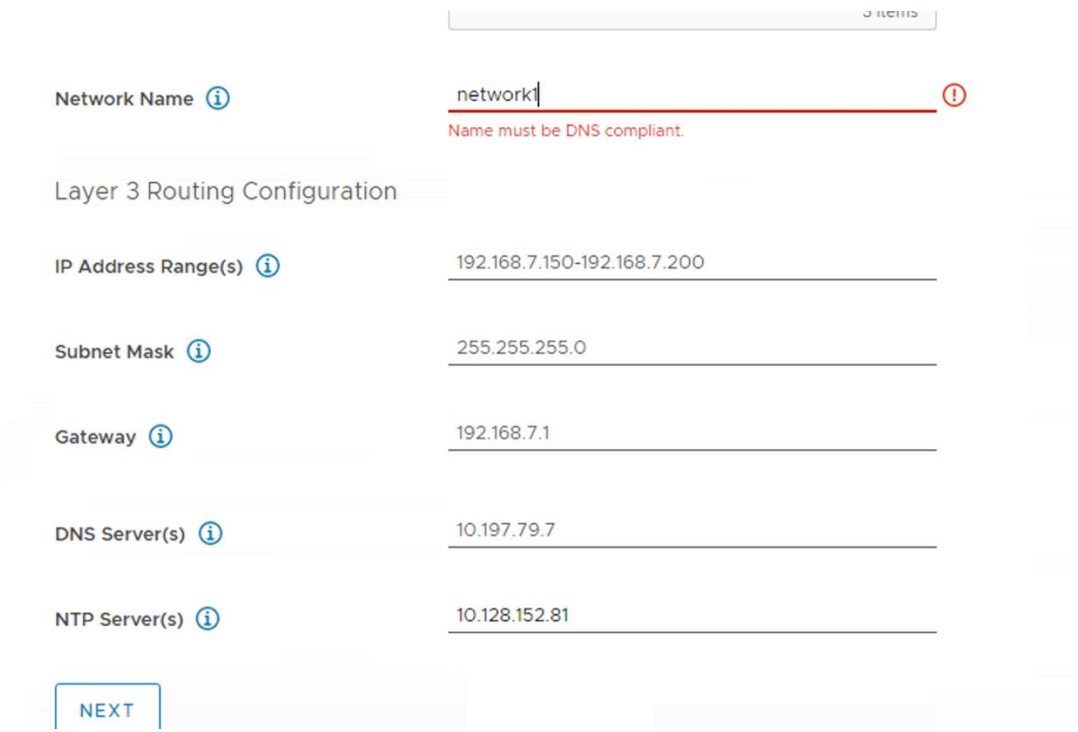




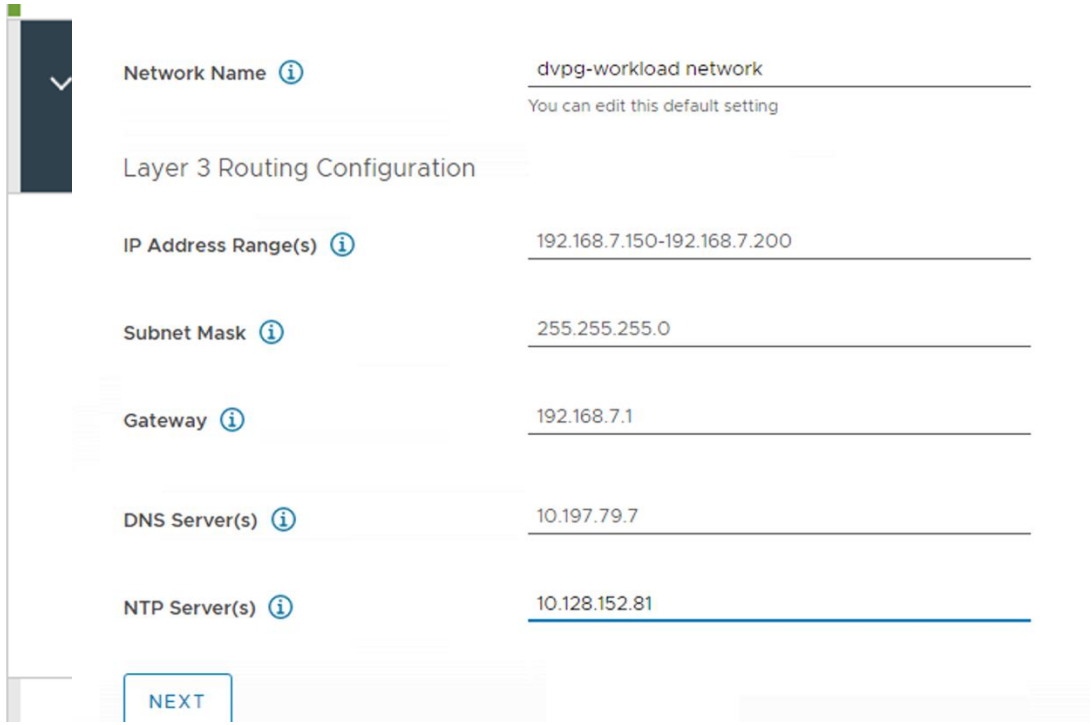




# WCP Enablement cont…

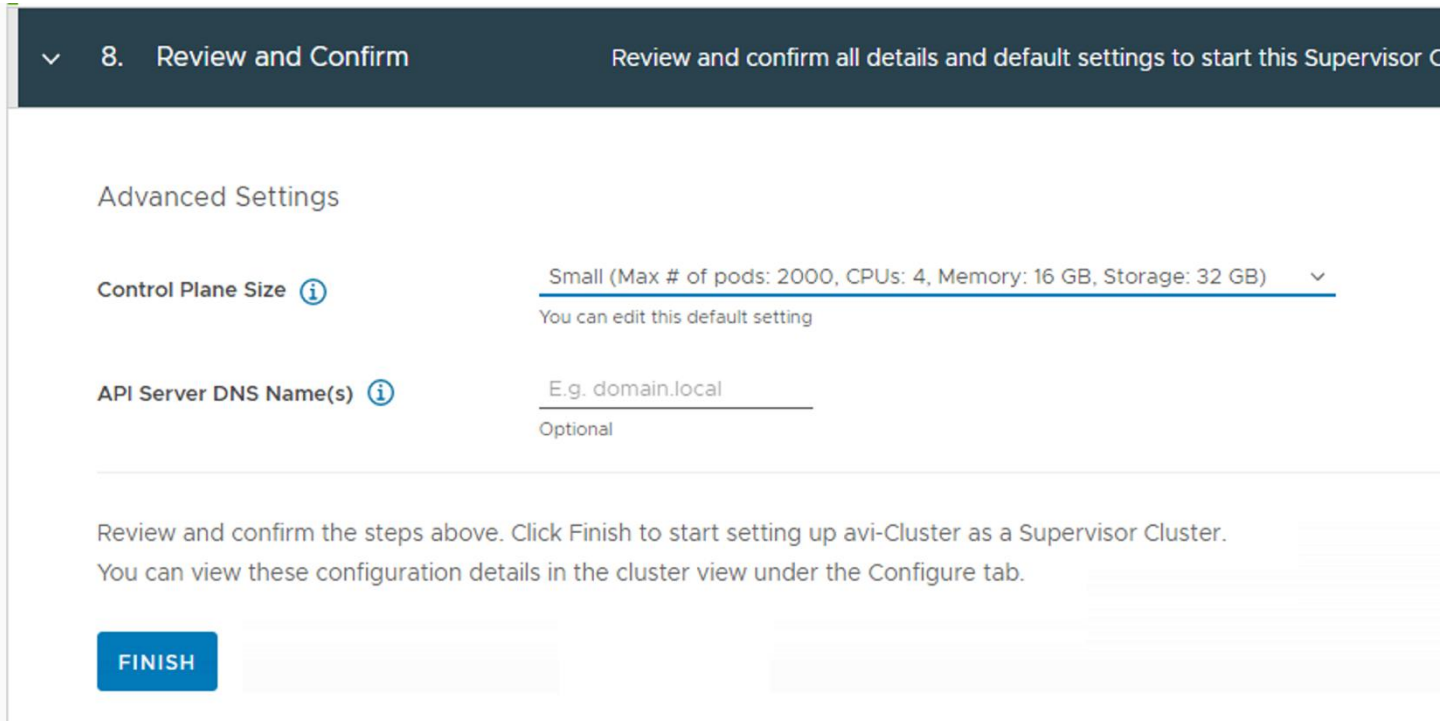
# WCP Enablement cont…



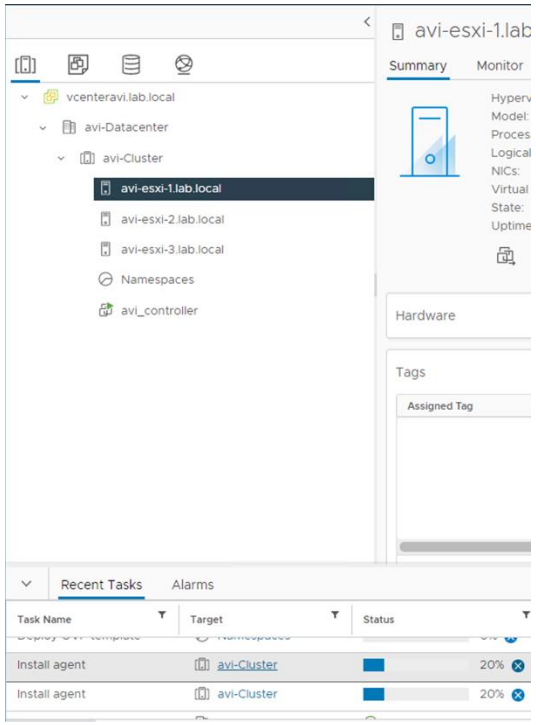
# WCP Enablement cont…



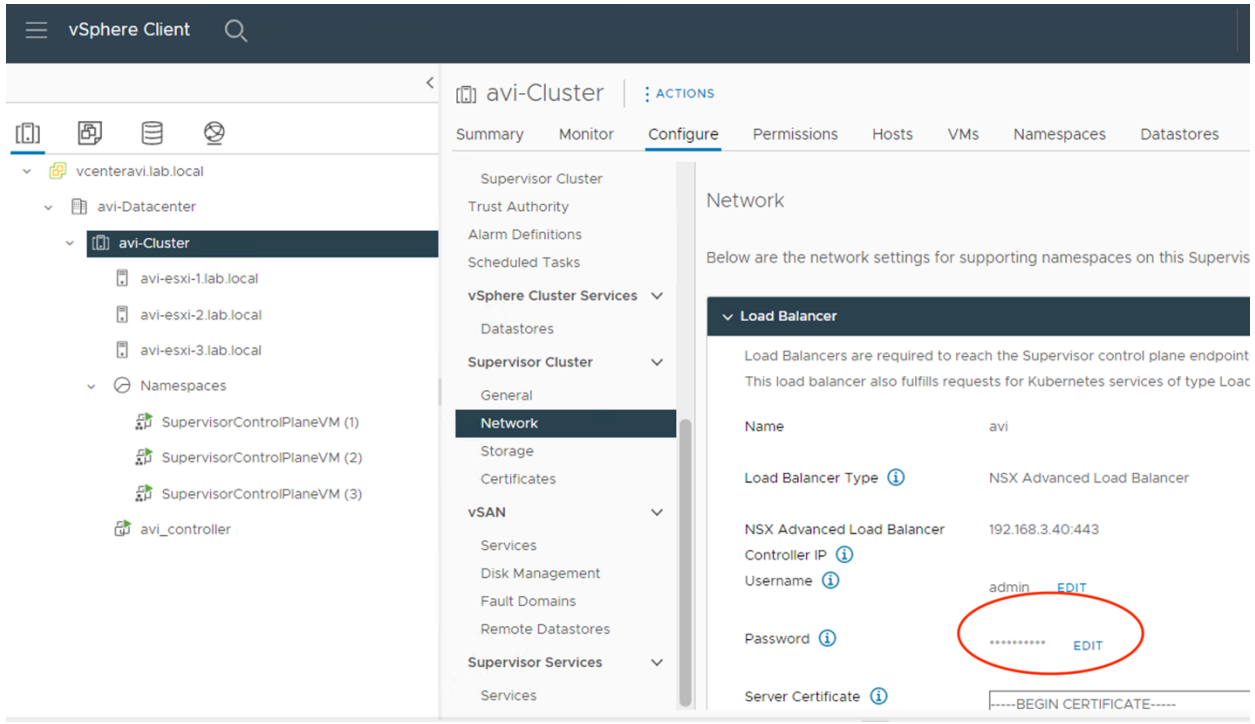
# WCP Enablement cont…



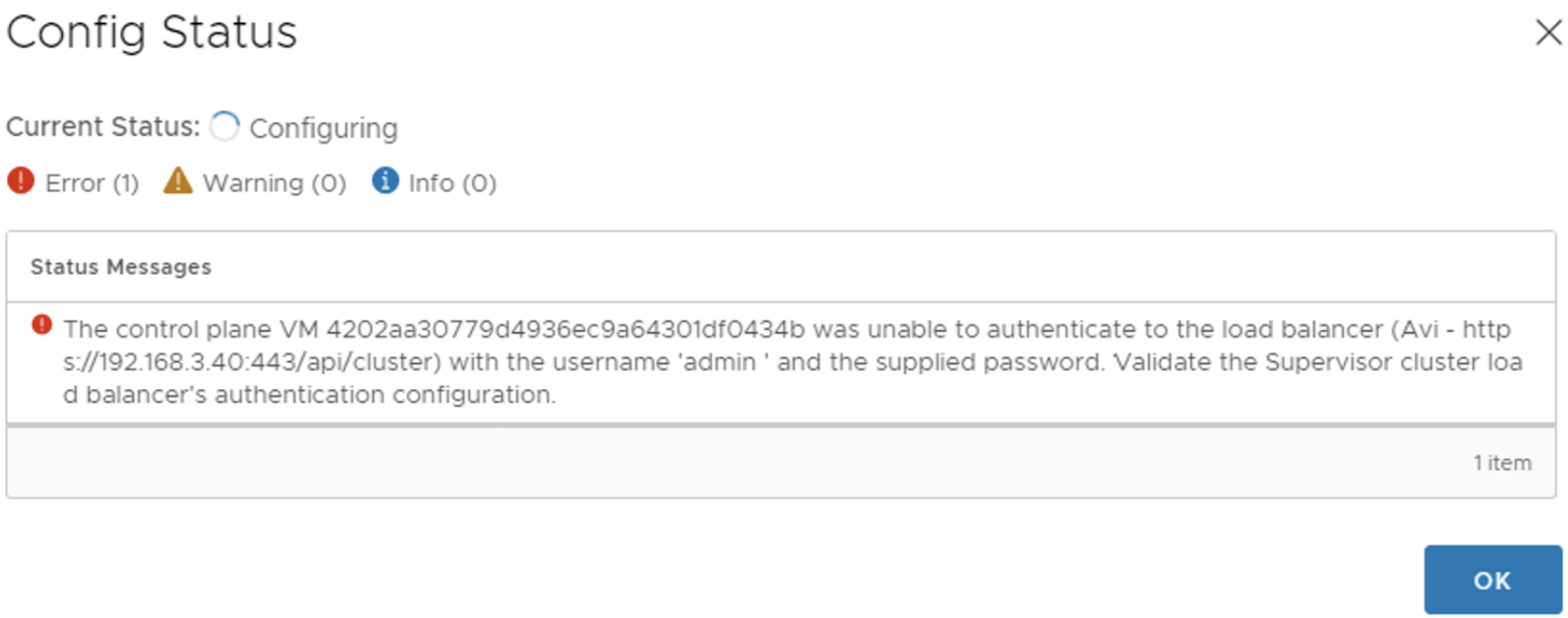
# WCP Enablement cont…



# Did I fat finger the AVI password? Edit here:



# Did I fat finger the AVI password?



# Still an issue… ssh to vCenter…

- ssh to vCenter ssh root@192.168.3.50
- shell
- /usr/lib/vmware-wcp/decryptK8Pwd.py
- ssh to IP and use password from above command
- kubectl get events -n vmware-system-ako -w
- kubectl get pods -A | grep ako
- kubectl logs -n vmware-system-ako pod/vmware-system-ako-ako-controller-manager-85bb7dd4-wnhrg
- kubectl get secrets -A | grep -i ako
- kubectl describe secrets/avi-secret -n vmware-system-ako
- kubectl get secret avi-secret -n vmware-system-ako -n vmware-system-ako -o jsonpath='{.data}'
- echo "Vk13YXJlMSE=" | base64 --decode

# Well look what we have here user ID with a training space…

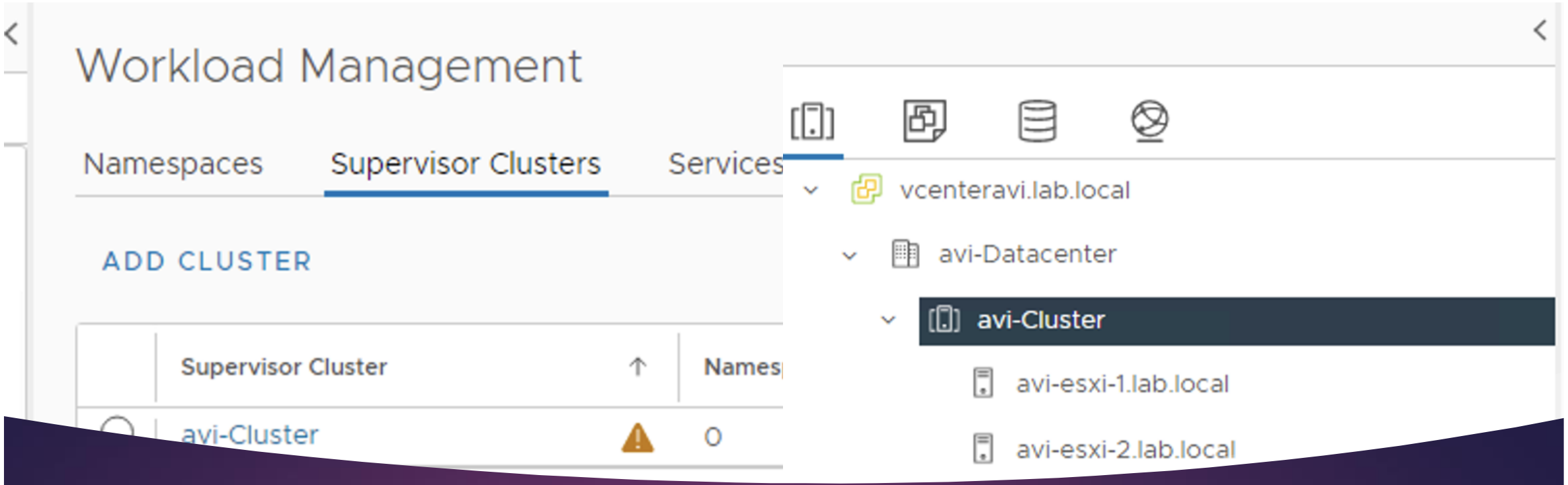
```
root@4202c1b4753322614d82535f07f425bd [ ~ ]# ^C
root@4202c1b4753322614d82535f07f425bd [ ~ ]# kubectl get secrets -A | grep -i ako
vmware-system-ako                       avi-secret                                       Opaque                                  3      122m
vmware-system-ako                       default-token-m5bbr                              kubernetes.io/service-account-token     3      122m
root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]# kubectl get secret avi-secret -n vmware-system-ako -n vmware-system-ako -o jsonpath='{.data}'
{"certificateAuthorityData":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMwVENDQWJtZ0F3SUJBZ01VUkdXeU5Tbk1Pb2JHSVhnZG5zdDVZNEdidjRZd0RRWUpLb1pJaHZjTkFRRUwKQlFBd0dERVdNQlFHQTFVRUF3d05ZWFpwTG14aFlpNXNiMk5oYkRBZUZ3MHlNVEV3TWpVeE56STJNRFJhRncweQpNakV3TWpVeE56STJNRFJhTUJneEZqQVVCZ05WQkFNTURXRjJhUzVzWVdJdWJHOWpZV3d3Z2dFaU1BMEdDU3FHClNJYjNEUUVCQVFVQUE0SUJEd0F3Z2dFS0FvSUJBUURBZVFVbmtzY0RHR31FbFBnMHNwZ01YdWpDN3h0K2UzUTIKeDRZM29RbTRLbVExeFR3N0ZRWktXeG5hQkp0SkNCKzcxS1NFU1dNTVh2L3NkV1N0Y2x3TDB2a011SDcxWEFzTQpXeFZNVys1NmdraVR2cDhWN1Q2akNXRlFRQnNIZ1o3VkxFVT1OaHJPN0szR3NTcDJLQjJzdWFnL1pEVjdsZkhKCk9hZlUraTJoL2UxYUowMG53R3F5QXFySWRkUVBuaHRmV09vREVIUkhuUDdIcmV6UE14N3ZhS2dRVnY5bVE0WkwKQjBKdXJQUi9zUGU0aWJuaFl3UVdVYkNuS2NHVjNrKytudHRUUVF3WlhPNVA1WUh2ZFlLNTh2eW5hd2JCS2sxbApjT0xJMU4rdmsrUDJid0ZpMFVDTEkycGcybm5aaERNTkZVTUptNHBtRzMwVEVKRmtsdzZkQWdNQkFBR2pFekFSCk1BOEdBMVVkRVFRSU1BYUhCTUNvQX1nd0RRWUpLb1pJaHZjTkFRUxCUUFEZ2dFQkFKc1dhTmVVZGJDKy8wV1oKYko1aXZ5M0FVV3d6K11Re1R4cXRNS2xseldOYmJMNy9zU1V6bEhYN0tTbEFZc3ppMC90T1lFpZU8xd25GWGFEWApwRzg1UmY3bTBZUk5TeE5RbD1CcXhXVytjdEVhd0tKU3kzOW93ckdweExROW11LzZyVjhiQzBWeWRuL0dLQTkwCnRuUE9BN09WcEpWU3JuRH16SmRaRmJWbVd0em9MdHYyZ25HREN0WjRPUTJFb01LbHE1Q113RXkrSUJ1Sk1QOFYKZEhSQjg5YXlnNysxSnpPSnN2c1NLL1BxU3lEU1BIRjhOc2Q5SFJVZzlqN3Y0cUxGcWNIQ1dpWkp1aXpQbnVObgpzTytGSENsZEUrOWh2WEZpbHk3VHNMWlRHZlQ4VExDSmpIcWdVNGxXOHQ0WjJMUHJVOFN4NTFVYnJJZXVVNlEvCk90RGFTY2M9Ci0tLS0tRU5EIENFU1RJRklDQVRFLS0tLS0=","password":"Vk13YXJlMSE=","username":"YWRtaW4g"}root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]# echo "Vk13YXJlMSE=" | base64 --decode
VMware1!root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]#
root@4202c1b4753322614d82535f07f425bd [ ~ ]# echo "YWRtaW4g" | base64 --decode
admin root@4202c1b4753322614d82535f07f425bd [ ~ ]#
```

# The system should update this user field but I have no time…

- echo -n '-----BEGIN CERTIFICATE-----
- My cert here….OtDaScc=
- -----END CERTIFICATE-----' > ./certificateAuthorityData.txt
- echo -n 'VMware1!' > ./password.txt
- echo -n 'admin' > ./username.txt
- kubectl delete secret avi-secret -n vmware-system-ako
- kubectl create secret generic avi-secret -n vmware-system-ako --from-file=./certificateAuthorityData.txt --from-file=./password.txt --from-file=./username.txt
- kubectl edit secret avi-secret -n vmware-system-ako

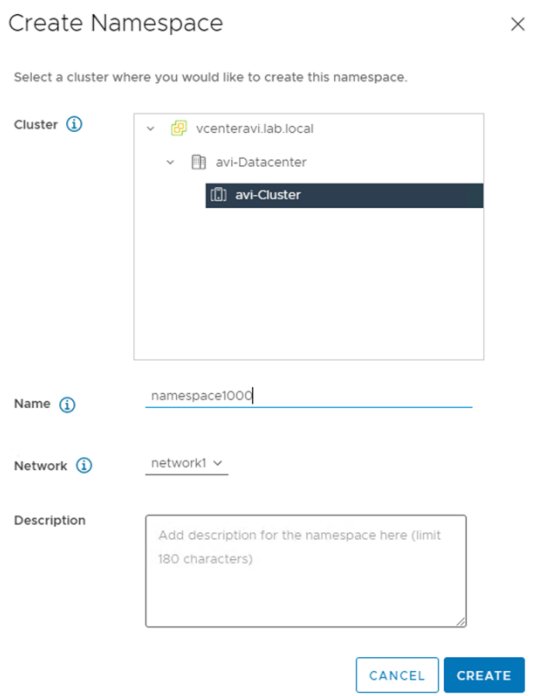
# Pod is running now…

- root@4202c1b4753322614d82535f07f425bd [ ~ ]# kubectl get pods -A | grep -i ako
- vmware-system-ako vmware-system-ako-ako-controller-manager-684cc57587-k4pvv 1/1 Running 28 139m
- root@4202c1b4753322614d82535f07f425bd [ ~ ]#

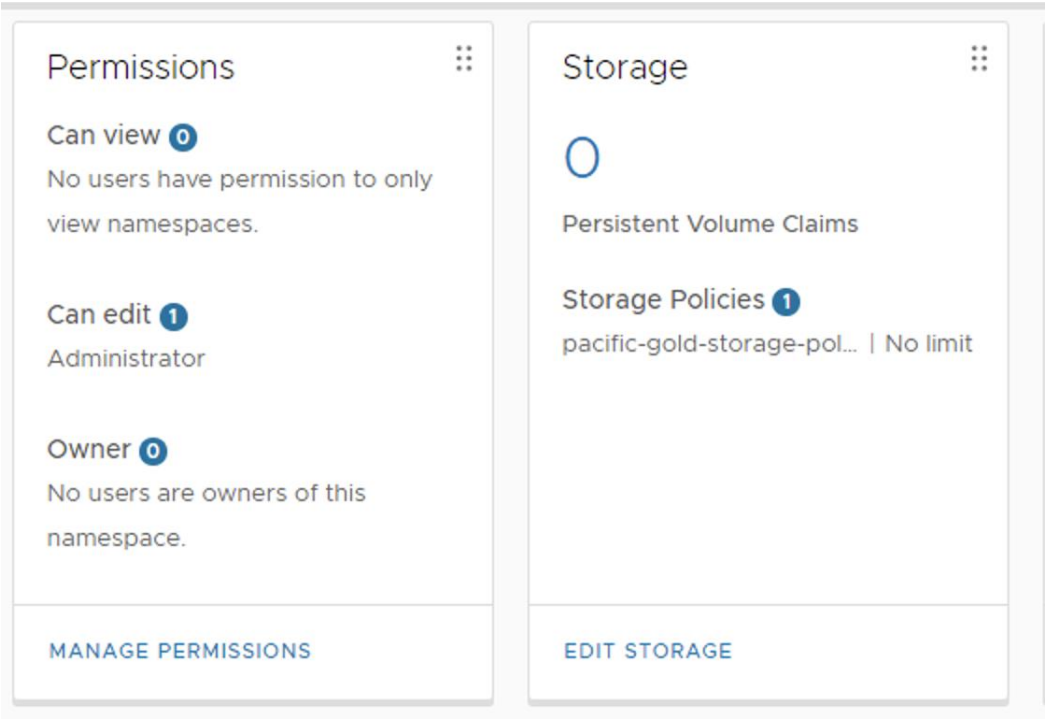



# SE are running now…
# and WCP green

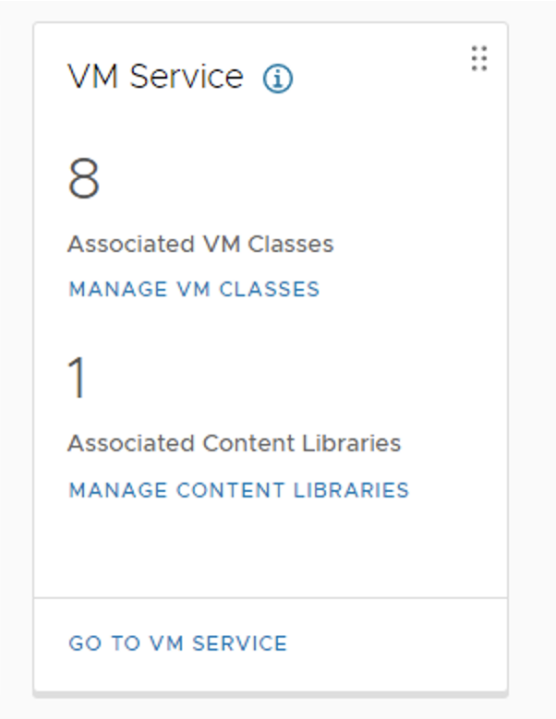
# Create namespace



# Add permissions and storage



# VM class and Content lib




```
[root@orfdns ~]# kubectl get virtualmachineclassbindings
NAME                    VIRTUALMACHINECLASS   AGE
best-effort-2xlarge     best-effort-2xlarge   20m
best-effort-4xlarge     best-effort-4xlarge   20m
best-effort-8xlarge     best-effort-8xlarge   20m
best-effort-large       best-effort-large     20m
best-effort-medium      best-effort-medium    20m
best-effort-small       best-effort-small     20m
best-effort-xlarge      best-effort-xlarge    20m
best-effort-xsmall      best-effort-xsmall    20m
[root@orfdns ~]#
```

# Loging on

- /usr/local/bin/kubectl-vsphere login --vsphere-username administrator@vsphere.local --server=https://192.168.5.40 --insecure-skip-tls-verify
- kubectl config use-context namespace1000
- [root@orfdns ~]# k get nodes
- NAME STATUS ROLES AGE VERSION
- 42025674cc72a310700121faddd8890a Ready control-plane,master 158m v1.21.0+vmware.1-wcp
- 4202aa30779d4936ec9a64301df0434b Ready control-plane,master 158m v1.21.0+vmware.1-wcp
- 4202c1b4753322614d82535f07f425bd Ready control-plane,master 164m v1.21.0+vmware.1-wcp

# Create Guest cluster

- kubectl-vsphere login --vsphere-username administrator@vsphere.local --server=https://192.168.5.40 --insecure-skip-tls-verify
- kubectl config use-context namespace1000
- k apply -f ./7u3cluster_photon.yaml

- kubectl vsphere login --server 192.168.5.40 --vsphere-username administrator@vsphere.local --tanzu-kubernetes-cluster-namespace namespace1000 --tanzu-kubernetes-cluster-name workload-vsphere-tkg2 --insecure-skip-tls-verify
- k config use-context workload-vsphere-tkg2
- kubectl apply -f https://github.com/ogelbric/YAML/raw/master/authorize-psp-for-gc-service-accounts.yaml
- k apply -f /root/7u2a/google-nginx2.yaml

# Overall Help

- https://github.com/ogelbric/YAML/blob/master/DemoScript.txt