Test Networking

Launch pods into Secondary CIDR network

Let's launch few pods and test networking

```
kubectl create deployment nginx --image=$ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com/aws/nginx/nginx
kubectl scale --replicas=3 deployments/nginx
kubectl expose deployment/nginx --type=NodePort --port 80
kubectl get pods -o wide
```

```
NAME              READY  STATUS   RESTARTS  AGE   IP          NODE                            NOMINATED
NODE   READINESS GATES
nginx-9c796bbf9-f4hhq  1/1    Running  0       11s    100.64.33.201   ip-10-0-1-117.eu-west-
1.compute.internal   <none>        <none>
nginx-9c796bbf9-sx7gn  1/1    Running  0       11s    100.64.139.157   ip-10-0-3-121.eu-west-
1.compute.internal   <none>        <none>
nginx-9c796bbf9-t7s7k  1/1    Running  0       2m21s  100.64.135.55   ip-10-0-3-121.eu-west-
1.compute.internal   <none>        <none>
```

If after 10-20 seconds have elapsed you see the containers are not Running - but in status ContainerCreating instead run this script to re-annotate the worker nodes:

```
CLUSTER=$(aws ssm get-parameter --name /workshop/tf-eks/cluster-name --query Parameter.Value --output text)
./reannotate-nodes.sh $CLUSTER
```

And then re-check the pods status with `kubectl get pods -o wide`]

You can use busybox pod and ping pods within same host or across hosts using IP address

```
kubectl run -i --rm --tty debug --image=$ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com/aws/docker/library/busybox -- sh
If you don't see a command prompt, try pressing enter.
```

```
/ #
```

Test access to internet and to nginx service

Internet connectivity - should fail (hang) as we build our EKS cluster in a private VPC !

```
wget google.com -O -
Connecting to google.com (172.217.5.238:80)
```

Type ctrl-c to escape:

```
<ctrl-c>
#
```

Test internal connectivity:

```
wget nginx -O -
Connecting to nginx (10.100.170.156:80)
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>

***TRUNCATED**

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
-                    100% |***********************************************************************************|
612  0:00:00 ETA
written to stdout
```

Finally exit from the busybox container:

```
exit
```

Proceed to the next step