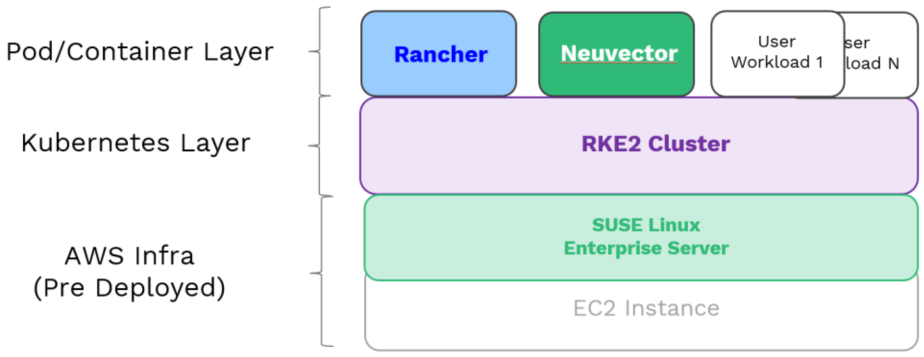
DELL – SUSE Lab Instructions

1 High Level Overview



2 Log into EC2 Instances

Using either PEM key (for Linux clients) or PPK key (for Windows clients) that was provided, log into your preassigned EC2 instances.
All the EC2 instances has been preconfigured with the username: ec2-user

For Linux clients, you can use the following command

```
ssh -i /path/to/your/key.pem ec2-user@ip_address
```

For Windows clients, you can use the application putty

3 Deploy RKE2

Download the installation script for RKE2 and run the script to allow RKE2 to be deployed.

```
sudo bash -c 'curl -sfL https://get.rke2.io | INSTALL_RKE2_CHANNEL="v1.32" sh -'
```

Run the following commands to configure the config file for RKE2

```
sudo mkdir -p /etc/rancher/rke2
sudo bash -c 'echo "write-kubeconfig-mode: \"0644\"" > /etc/rancher/rke2/config.yaml'
```

Start and enable rke2-server.service

```
sudo systemctl enable rke2-server.service
sudo systemctl start rke2-server.service
```

Download and install kubectl which will be the CLI client to communicate with the RKE2 cluster

```
curl -LO https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl
```

```
-   sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

Kubectl uses a file called kubeconfig file that has been created by RKE2 during the starting of the service. The file must be placed into a specific folder.

```
-   mkdir -p ~/.kube
-   ln -s /etc/rancher/rke2/rke2.yaml ~/.kube/config
```

Run kubectl commands to ensure the cluster is working as expected.

```
-   Kubectl get pods -A
-   Kubectl get nodes -A
```

4 Deploy Rancher

Install the k8s package manager Helm which manages the deployment of applications.

```
-   curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
```

Deploy application cert-manager using helm. Cert-manager is responsible for managing the TLS certificates for Rancher

```
-   helm repo add jetstack https://charts.jetstack.io
-   helm install \
 cert-manager jetstack/cert-manager \
 --namespace cert-manager \
 --set installCRDs=true \
 --create-namespace
```

Deploy Rancher. Do take note to include the IP address that has been assigned to you under the hostname parameter so as to ensure a publicly routable DNS based on sslip.io is configured.

```
-   helm repo add rancher-stable https://releases.rancher.com/server-charts/stable
-   kubectl create ns cattle-system
-   helm install rancher rancher-stable/rancher \
--namespace cattle-system \
--set hostname=rancher.ip_address.sslip.io \
--set bootstrapPassword=admin \
--set replicas=1 \
--version v2.11.2
```

You can also monitor the deployment status of Rancher.

```
kubectl -n cattle-system rollout status deployment/rancher
```

Once Rancher is ready, you can access Rancher web console on any internet explorer by using the hostname that you have configured earlier. The password is “admin” and you would need to set a new password with a minimum of 12 characters.

5 Explore Rancher

Now that Rancher web console can be assessed, you can explore the web console and look at some of the features that are available out of the box.

There are some key tasks that we need to do before we can continue with the lab work.

Add new Helm Repository

From the global menu, select the “local cluster”. Then, from the “local cluster”, go to “Apps -> Repositories -> Create” and input the following.

Name: rodeo

Index URL: https://rancher.github.io/rodeo

Install Wordpress Application

Following that, a new repo would be available. Next, go to “Apps -> Charts -> Select “rodeo” from the pull down bar -> select “Wordpress” -> Install” and input the following.

Namespace: default

Name: wordpress

Next, from the forms, go to “Services and Load Balancing” and input the following. Again, ensure the correct ip address is input.

Hostname: wordpress.ip_address.sslip.io

Wordpress would be successfully deployed shortly.

6 Deploy Neuvector

Ensure that the “local cluster” is selected. Go to “Apps -> Charts -> Select “Rancher” from the pull down bar -> select “Neuvector” -> Install”. There is no need to change any settings from the helm chart.

Neuvector would be successfully deployed shortly.

Note that a new tab named “Neuvector” would appear under the local cluster and you can now access Neuvector web console.

7 Explore Neuvector

Note: Neuvector's demo can be hard to follow as the sequence and logic involved requires quite a fair bit of hands on and understanding. The instructions here are the high level overview. For complete and detail steps, we can refer to the full Neuvector Lab called Neuvector Rodeo.

Now that Neuvector web console can be assessed, you can explore the web console and look at some of the features that are available out of the box.

Turn on "Auto Scan" by toggling on the "Auto Scan" toggle on the top right.

This enables the scanning feature for all containers currently running on the system, as well as any subsequent deployments. CIS and compliance scans are also run on the nodes themselves, as well as kubernetes.

8 Admission Controller

Go to "Policy -> Admission Control -> Status -> Create Rule" and input the following and then click on the + icon to add the rule
Type: Deny
Comment: Deny default namespace
Criterion: Namespace
Operator: Is one of
Value: default

Go to Rancher web console and create a new deployment.
Ensure that the "local cluster" is selected. Go to "Workloads -> Deployments -> Create" and input the following and create a new deployment
Name: hello-world
Container Image: rancher/hello-world:latest

Now we might need to toggle between both the web consoles and play around with the settings. You will notice that as we toggle the settings from between "Disabled", "Monitor" and "Protect", how the "Hello-world" application will be affected and how reporting would be seen on Neuvector if you go to "Notifications -> Risk Reports".

9 Show learned rules of Wordpress

Neuvector can automatically learn the standard behaviour of your applications and build up a ruleset to use in a zero trust manner. Each Pod can be in one of 3 modes.

- **Discover** to learn the behaviour and automatically built up a ruleset
- **Monitor** to alert, but not block behaviour that is not covered by rules
- **Protect** to alert and block behaviour that is not covered by rules

Go to Policy > Groups and search for the "wordpress" group and explore the different tabs. Again, we might need to toggle between the web consoles and play around with the settings from here onwards. You will notice that as we toggle the multiple settings that are available, how the "Wordpress" application will be affected and how reporting would be seen on Neuector if you go to "Notifications -> Security Events".

Do take note that the command below would be use frequently from the Rancher web console to showcase this part of the demo. Please ensure that the right namespace is used. You can execute the command below by using the “kubectl” feature.

```
kubectl -n default exec $(kubectl get pods -n default -l app.kubernetes.io/name=wordpress -o name) -- curl -v https://suse.com
```