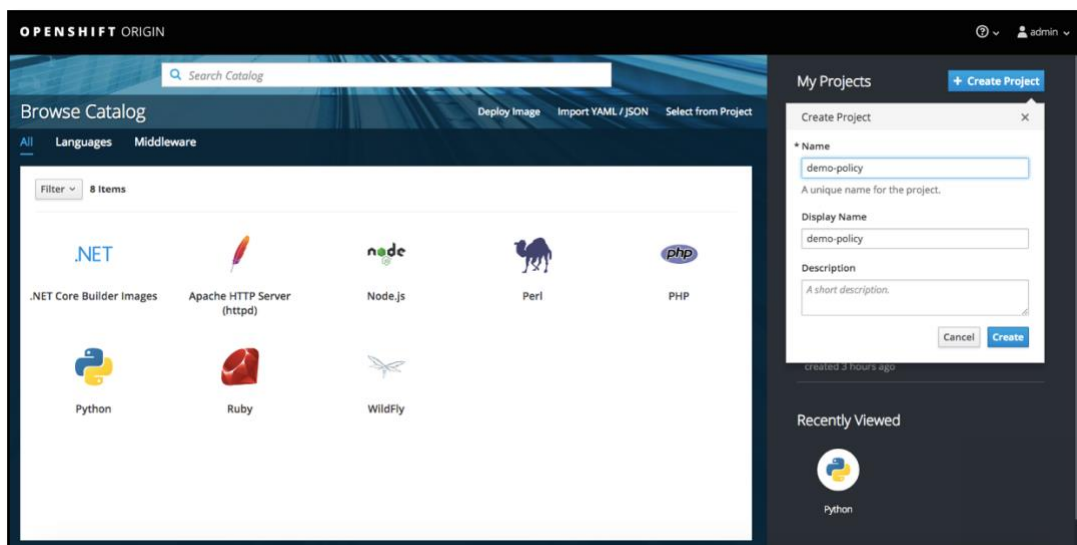


Use Case - 3

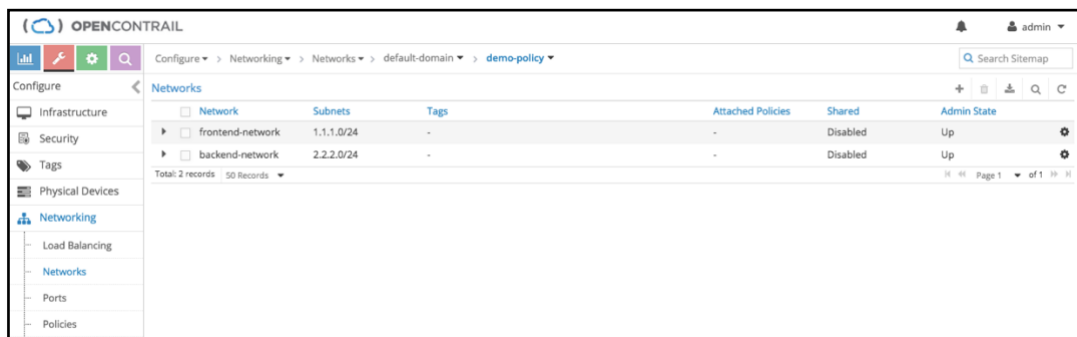
- Secure highly scalable applications at microservices level using OpenShift

1. Login to the OpenShift dashboard & **create** a new project



2. Login to the Contrail dashboard & **create** two new Virtual Networks in the above created project

Contrail Dashboard > Configure > Networking > Networks (demo-policy) > (+)



3. From OpenShift dashboard, **launch** two pods in the above project

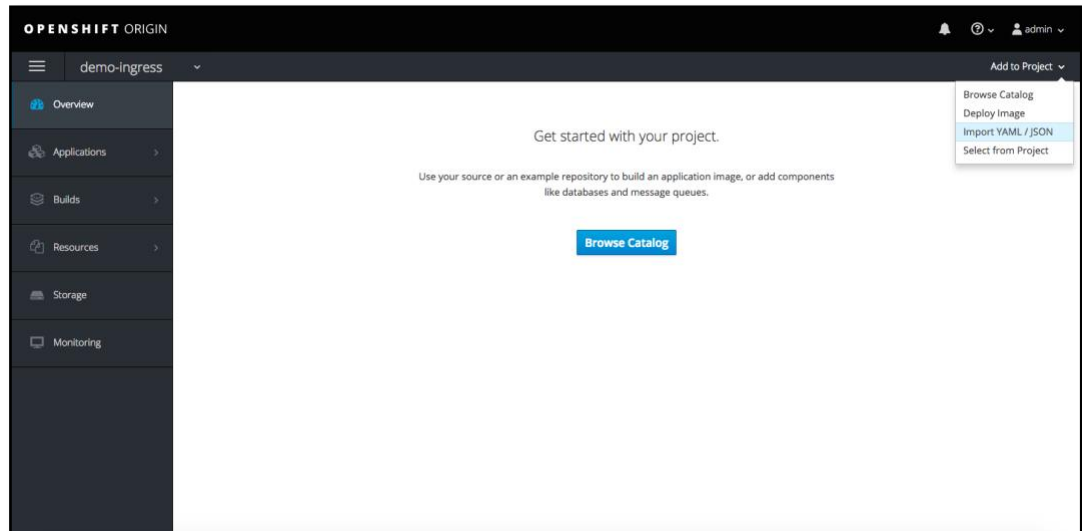
OpenShift Dashboard > Add to Project > Import YAML/JSON

There are two pods used for this demo

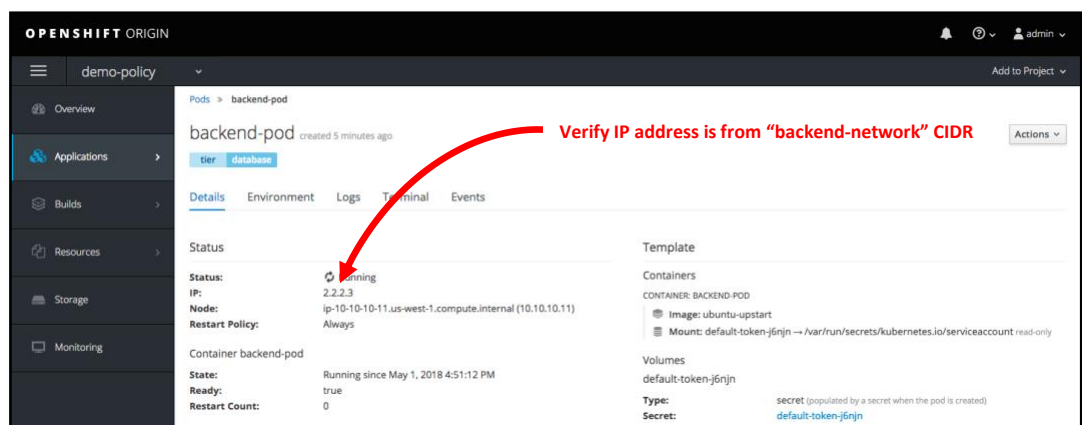
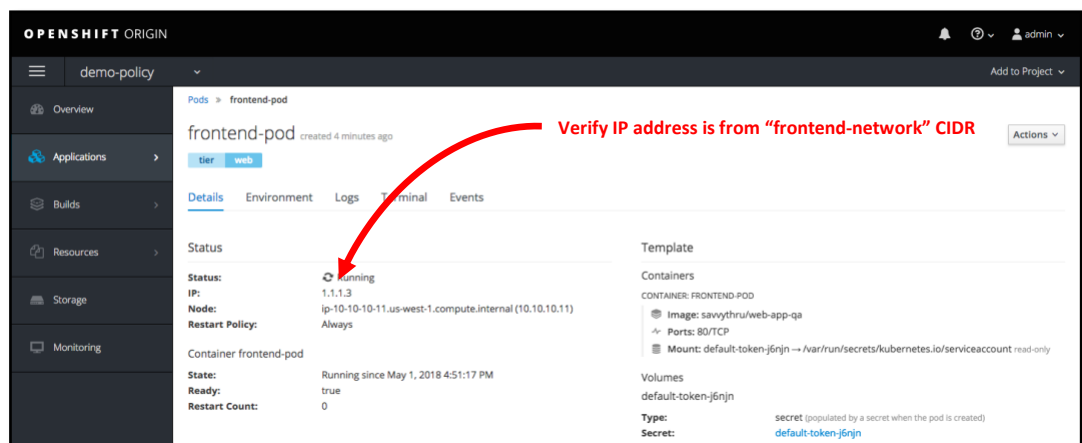
- backend-pod** launched in “**backend-network**” with label “**tier=database**”
- frontend-pod** launched in “**frontend-network**” with label “**tier=web**”

NOTE: All pod specs are located [here](#). Copy paste the pod YAMLs into the OpenShift window & hit **Create**

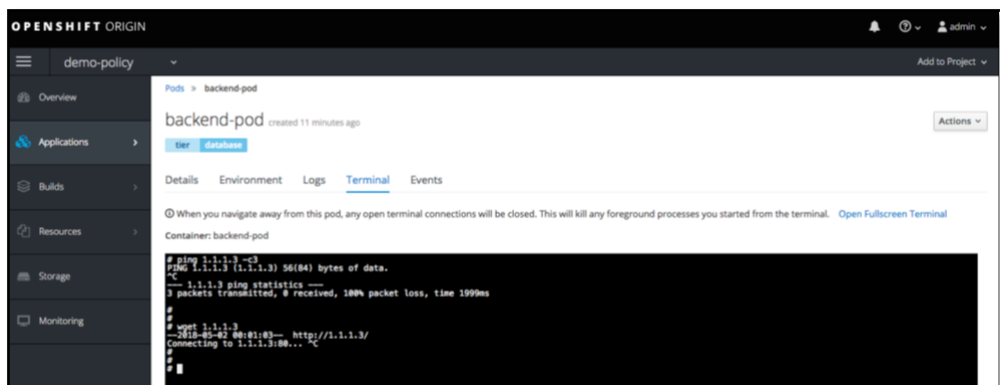
Let's create the frontend & backend pods



Navigate to *OpenShift Dashboard > Applications > Pods* & verify pods came up fine



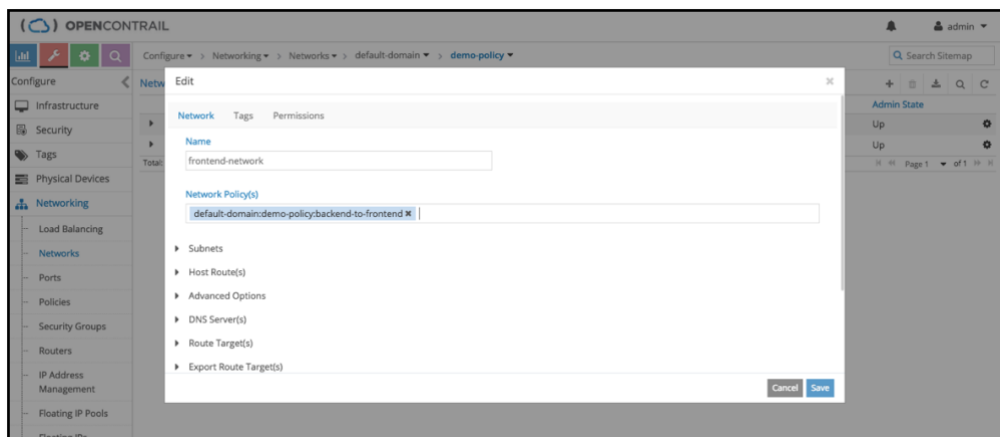
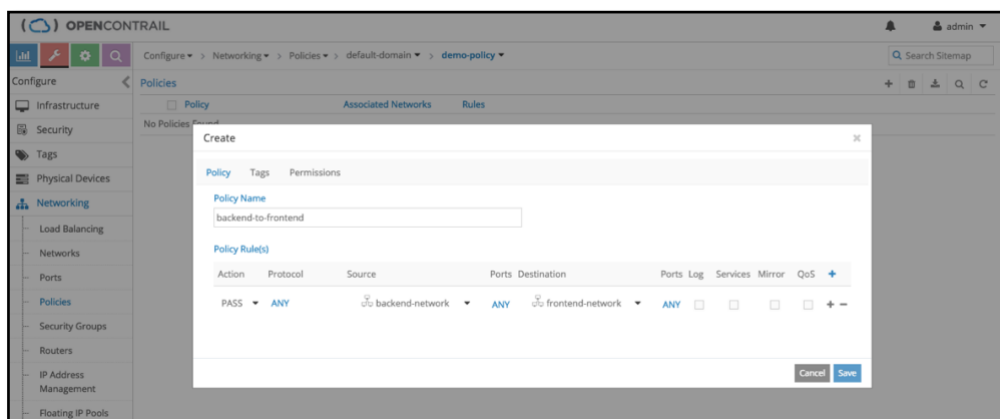
4. Login to the “**backend-pod**”, initiate ping & wget towards the “**frontend-pod**”



NOTE: Since Virtual Networks are isolated by default, the “**frontend-pod**” & “**backend-pod**” will not be able to talk to each other

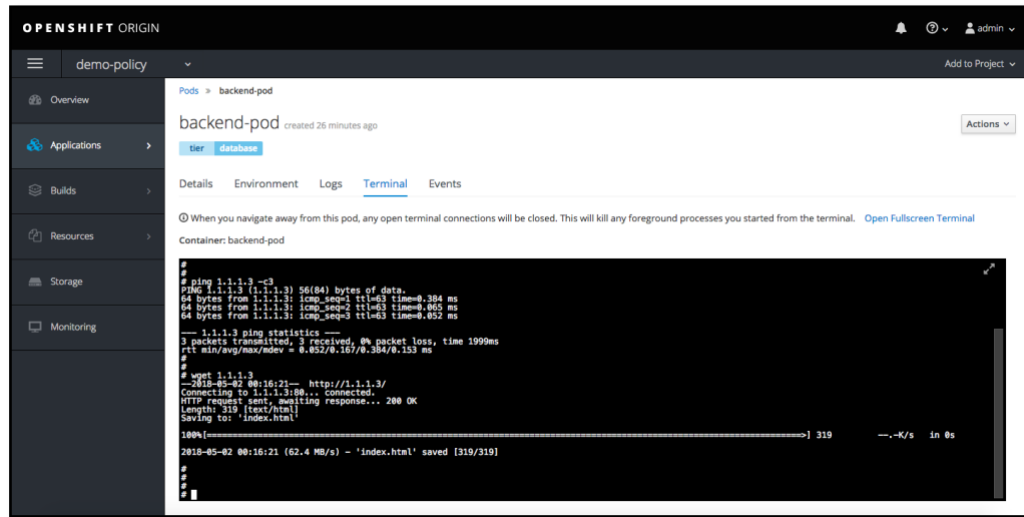
5. Create a **Contrail Network Policy** & attach them to the two Virtual Networks, in order to leak routes between the them

Contrail Dashboard > Configure > Networking > Policies (demo-policy) > (+)
Contrail Dashboard > Configure > Networking > Networks (demo-policy) > Edit (attach)



6. Repeat **STEP 4**

Verify from the **“backend-pod”** that you can ping & get content from **“frontend-pod”**



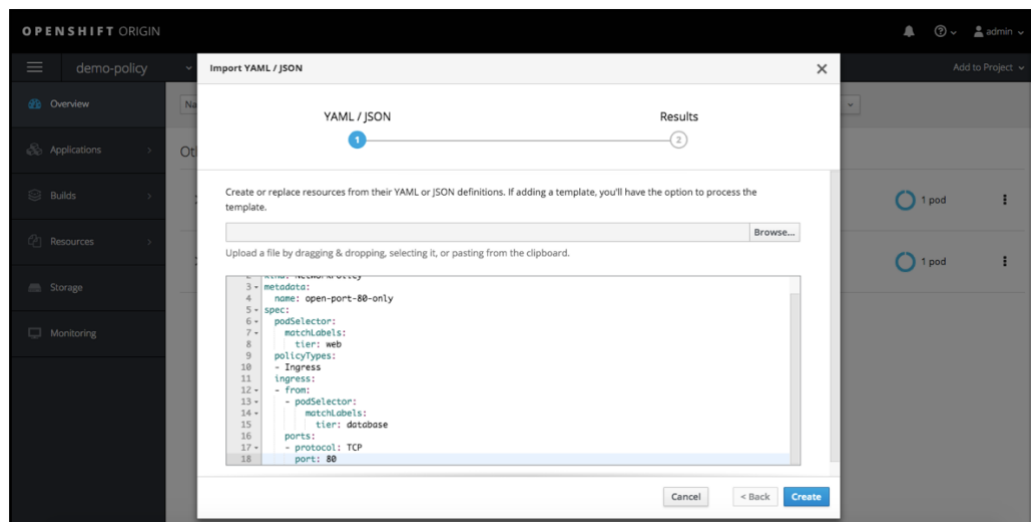
7. Now, let's define a more granular network-policy which will allow **“frontend-pod”** to accept traffic only on **port 80** & from **“backend-pod”**

NOTE: We accomplish this using the native Kubernetes construct called as labels

The policy definition creates a Contrail Security firewall rule that will allow traffic to workloads with label (**tier=web**) from workloads with label (**tier=database**) only on **port 80**

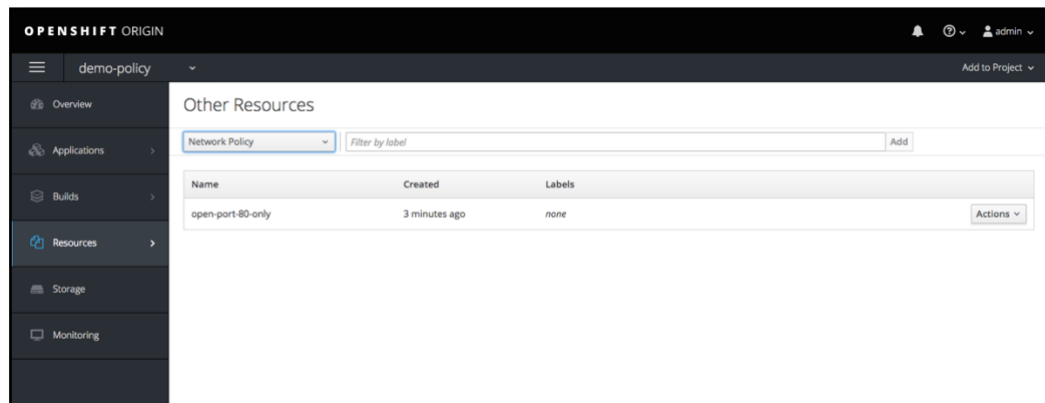
NOTE: All network-policy specs are located [here](#). Copy paste the **web-to-db** YAMLs into the OpenShift window & hit **Create**

Navigate to *OpenShift Dashboard > Add to Project > Import YAML/JSON*

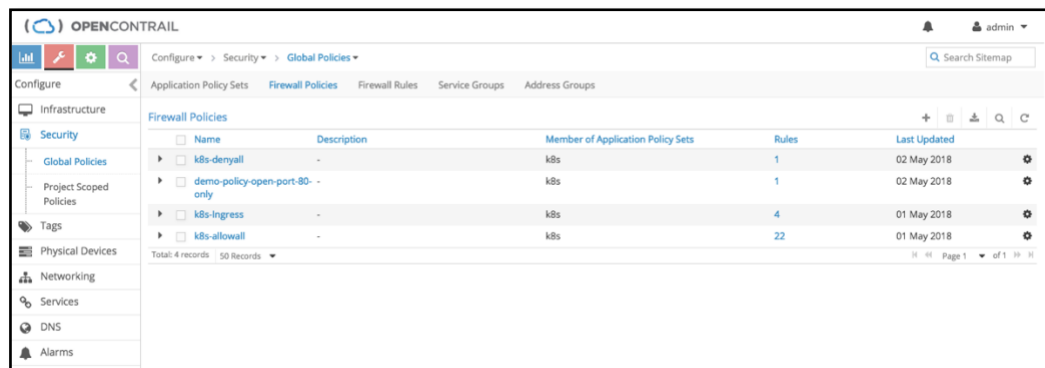


8. Verify that the **OpenShift Network Policy & Contrail Security Firewall Rule** are successfully created

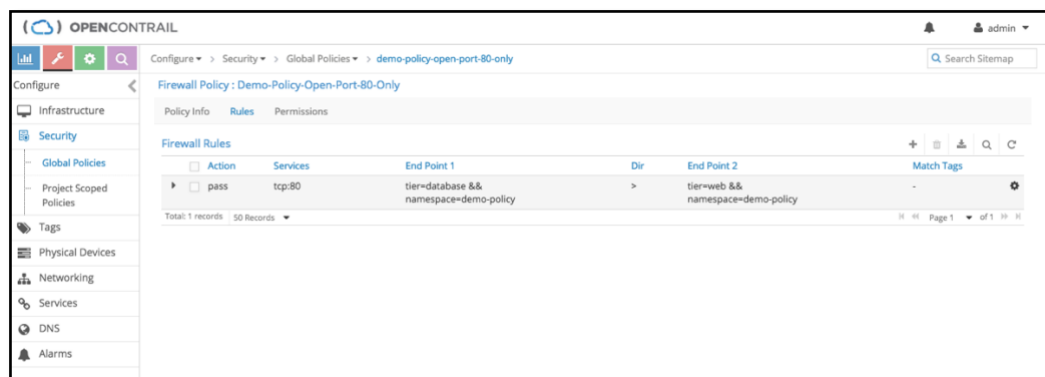
Navigate to *OpenShift Dashboard > Resources > Other Resources* & verify **OpenShift Network Policy** was successfully created



Navigate to *Contrail Dashboard > Resources > Other Resources* & verify **Contrail Security Firewall Rule** was successfully created



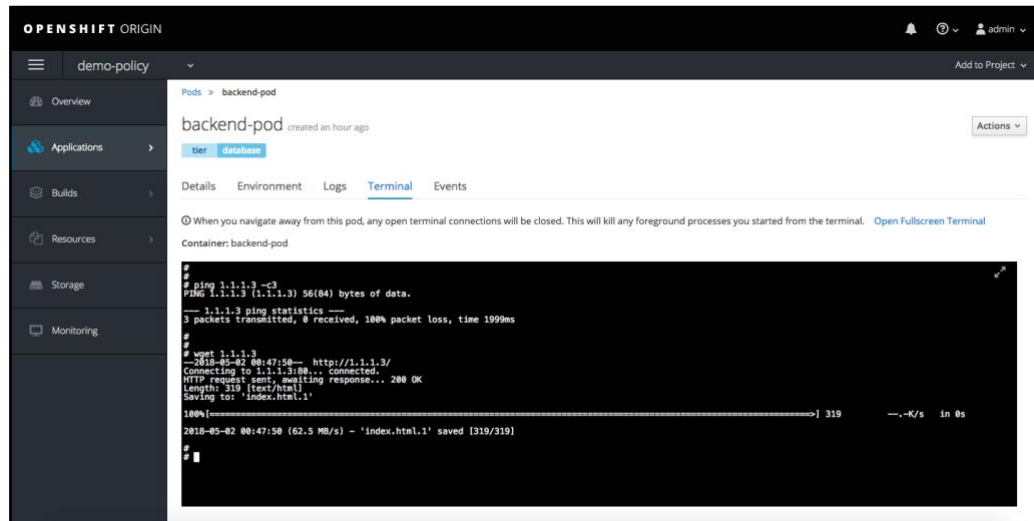
Click on the **Rules** tab to find the specific ACL rule



9. Repeat STEP 4

Login to the “**backend-pod**”, initiate ping & wget towards the “**frontend-pod**”

Since **Contrail Security Firewall Rule** is enforced on “**frontend-pod**” to allow only TCP protocol on port 80 from “**backend-pod**”, **ping (ICMP)** & **all other traffic** is blocked & only **wget (TCP)** is allowed



NOTE: The “**frontend-pod**” allows incoming **TCP** connections from “**backend-pod**” on **port 80** & drops all other traffic

NOTE: Contrail dashboard provides visualization & rich analytics such as packet counters, ACL hits counter, etc., to explain the same

Navigate to *Contrail dashboard > Monitor > Security > Traffic Groups (demo-policy)*

