NGINX Ingress Controller

Deployment with the new Deployment Guides

TL;DR

- Install `following the instructions` in: [https://kubernetes.github.io/ingress-nginx/deploy/]

You need to modify service to expose to external IPs - check `SECTION 2.`

- Install with `ingress_install.yaml` modified to fit our Local K8s instance *via our GitLab*

RUN: `kubectl apply -f ingress_install.yaml`

Check Services and modify to expose to external IPs.

```
externalIPs:
    - 10.0.20.31
```

Here is an example of modified service:

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/version: 1.1.1
  name: ingress-nginx-controller
  namespace: ingress-nginx
spec:
  clusterIP: 10.100.186.233
  clusterIPs:
  - 10.100.186.233
  externalIPs:
  - 10.0.20.31
  externalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - name: http
    nodePort: 30905
    port: 80
    protocol: TCP
    targetPort: http
  - name: https
    nodePort: 31471
    port: 443
    protocol: TCP
    targetPort: https
  selector:
    app.kubernetes.io/component: controller
    app.kubernetes.io/instance: ingress-nginx
    app.kubernetes.io/name: ingress-nginx
  sessionAffinity: None
  type: LoadBalancer
```

Applying an Ingress Object ***AFTER*** Ingress controller is installed:

- Notes:
  - Ingress is namespaced.
  - Documentation: https://kubernetes.io/docs/concepts/services-networking/ingress/
- Check syntax of example here:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-name
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          service:
            name: test
            port:
              number: 80
```

Installing SSL Certificates for domains other than (Self Signed):

Most of the times this should not happen. Most of our certs are using Let’sEncrypt. Use different certs only in production!

- Notes:
  - This requires SSL certificates that are signed for the *.url.com (for example).
  - ⚠️ **Should not be SELF SIGNED**
  - TLS Secret is per namespace. This means that every namespace have its own ***Secret*** of TLS.
- Building a TLS Secret to a namespace:
  - Command:
    - `kubectl create secret tls <name_to_give_secret> --key <key_file.crt> --cert <certificate_file.crt> -n <namespace_name>`
  - Example
    - `kubectl create secret tls ingress-tls --key $HOME/certs/example.key --cert $HOME/certs/example/crt -n test_namespace`

- Adding TLS to ingress object as:

```
  tls:
    - hosts:
        - test.example.com
      secretName: ingress-tls
```

- Example Ingress Object with TLS:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-name
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  tls:
    - hosts:
        - test.example.com
      secretName: ingress-tls
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          service:
            name: test
            port:
              number: 80
```

Applying an ingress object as:

kubectl apply -f <filename.yaml> -n <namespace_name>

Troubleshooting

- Error not found 404 or 502 or other:
  - Check Check pod is up and running.
  - Check pod is exposing port.
  - Check service endpoints to match pod.
  - Check service port to match to ingress.
  - Check service port to match target port of pod exposed port.
  - Check ExternalIps are configured in service of ingress controller.
  - Check logs of ingress nginx controller pod.
  - Check for double or overlapping ingress objects.
- If none of the above works...
  - RTFM - Read the fucking manual - or Reinstall ingress.