Expose Kibana and Logstash with Ingrass

Install Ingress Packages

Install NGINX Ingress Controller

```
helm repo add ingress-nginx https://kubernetes.github.io/ingr
ess-nginx/
helm install ingress-nginx ingress-nginx/ingress-nginx --name
space elasticsearch --create-namespace
```

Verify Installation

```
kubectl get pods -n elasticsearch
kubectl get svc -n elasticsearch
```

Kibana Setup

- Update the Kibana Service to type `NodePort` on he kibana-developement.yaml

```
spec:
  selector:
    app: kibana
  ports:
  - port: 5601
    targetPort: 5601
    nodePort: 30001  # You can specify a port in the NodePort
range
  type: NodePort
```

- Apply the changes

```
kubectl apply -f kibana-deployment.yaml
```

- Create Ingress Resource file

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: kibana-ingress
  namespace: elasticsearch
spec:
  ingressClassName: nginx
  rules:
  - host: 172.23.37.31  # Replace with your IP address
    http:
      paths:
      - path: /kibana
        pathType: Prefix
        backend:
          service:
            name: kibana
            port:
              number: 5601
```

- Apply the file with

```
kubectl apply -f kibana-ingress.yaml
```

Access Kibana

http://<node-ip>:30001

Logstash

Since logstash doesn't require external traffic routing we simply add NodePort to the logstash values.yaml file and no need to create logstash-ingress-resource yaml file

```
service:
    type: NodePort
    ports:
      - name: beats
        port: 5044
        protocol: TCP
        targetPort: 5044
      - name: monitoring
        port: 9600
        protocol: TCP
        targetPort: 9600
      - name: syslog
        port: 5140
        protocol: TCP
        targetPort: 5140
        nodePort: 30006
```

Reinstall logstash with

```
helm uninstall logstash  -n elasticsearch
helm dep build logstash-parent/ -n elasticsearch
helm install logstash logstash-parent/ -n elasticsearch
kubectl get pod -n elasticsearch -w
```

Troubleshooting

Issue

kubectl apply -f kibana-ingress.yaml

```
Warning: annotation "kubernetes.io/ingress.class" is deprecat
ed, please use 'spec.ingressClassName' instead
Error from server (InternalError): error when creating "kiban
a-ingress.yaml": Internal error occurred: failed calling webh
ook "validate.nginx.ingress.kubernetes.io": failed to call we
```

```
bhook: Post "https://ingress-nginx-controller-admission.elast
icsearch.svc:443/networking/v1/ingresses?timeout=10s": tls: f
ailed to verify certificate: x509: certificate is valid for i
ngress-nginx-ingress-nginx-admission, ingress-nginx-ingress-n
ginx-admission.elasticsearch.svc, not ingress-nginx-controlle
r-admission.elasticsearch.svc
```

Solution

Uninstall then install again

```
helm uninstall ingress-nginx -n elasticsearch
```

List all webhook configurations:

```
kubectl get validatingwebhookconfigurations
```

Delete the Ingress-related webhook configuration:

```
kubectl delete validatingwebhookconfiguration ingress-nginx-a
dmission
```

Delete Services, Secrets, and ConfigMaps

```
kubectl get svc -n elasticsearch
kubectl delete svc ingress-nginx-controller -n elasticsearch
```

List and delete secrets:

```
kubectl get secrets -n elasticsearch
kubectl delete secret ingress-nginx-ingress-nginx-admission -
n elasticsearch
```

List and delete configmaps:

```
kubectl get configmaps -n elasticsearch
kubectl delete configmap ingress-nginx-controller -n elastics
earch
```

Check for Remaining Pods

```
kubectl get pods -n elasticsearch
kubectl delete pod <ingress-pod-name> -n elasticsearch
```

Check all resources in the namespace:

```
kubectl get all -n elasticsearch
```

Delete any remaining resources:

```
kubectl delete deployment ingress-nginx-controller -n elastic
search
kubectl delete service ingress-nginx-controller -n elasticsea
rch
```

Reinstall the NGINX Ingress Controller and apply ingress-resources.yaml