安装EFK插件

通过在每台node上部署一个以DaemonSet方式运行的fluentd来收集每台node上的日志。Fluentd将docker日志目录/var/lib/docker/containers和/var/log目录挂载到Pod中，然后Pod会在node节点的/var/log/pods目录中创建新的目录，可以区别不同的容器日志输出，该目录下有一个日志文件链接到/var/lib/docker/contianers目录下的容器日志输出。
官方文件目录：https://github.com/kubernetes/kubernetes/tree/master/cluster/addons/fluentd-elasticsearch

给 Node 设置标签

定义 DaemonSet fluentd-es-v1.22 时设置了 nodeSelector beta.kubernetes.io/fluentd-ds-ready=true ，所以需要在期望运行 fluentd 的 Node 上设置该标签；

```
kubectl label nodes 192.168.251.101 beta.kubernetes.io/fluentd-ds-ready=true
kubectl label nodes 192.168.251.102 beta.kubernetes.io/fluentd-ds-ready=true
kubectl label nodes 192.168.251.103 beta.kubernetes.io/fluentd-ds-ready=true
```

执行定义文件

```
# kubectl create -f .

service/elasticsearch-logging created
serviceaccount/elasticsearch-logging created
clusterrole.rbac.authorization.k8s.io/elasticsearch-logging created
clusterrolebinding.rbac.authorization.k8s.io/elasticsearch-logging created
statefulset.apps/elasticsearch-logging created
configmap/fluentd-es-config-v0.1.4 created
serviceaccount/fluentd-es created
clusterrole.rbac.authorization.k8s.io/fluentd-es created
clusterrolebinding.rbac.authorization.k8s.io/fluentd-es created
daemonset.apps/fluentd-es-v2.2.0 created
deployment.apps/kibana-logging created
service/kibana-logging created
```

检查执行结果

```
# kubectl get deployment -n kube-system | grep kibana

kibana-logging         1         1         1            1           24m
```

```
# kubectl get pods -n kube-system | grep -E 'elasticsearch|fluentd|kibana'

elasticsearch-logging-0                 1/1       Running   0          23m
elasticsearch-logging-1                 1/1       Running   0          23m
fluentd-es-v2.2.0-mqvd9                 1/1       Running   0          23m
fluentd-es-v2.2.0-vrp9x                 1/1       Running   0          23m
fluentd-es-v2.2.0-w7gbj                 1/1       Running   0          23m
kibana-logging-56c4d58dcd-c2dtk         1/1       Running   0          23m
```

```
# kubectl get service  -n kube-system | grep -E 'elasticsearch|kibana'

elasticsearch-logging   ClusterIP   10.254.7.17      <none>
9200/TCP                        24m
kibana-logging          ClusterIP   10.254.249.78    <none>
5601/TCP                        24m
```

kibana Pod 第一次启动时会用较长时间(10-20分钟)来优化和 Cache 状态页面，可以 tailf 该 Pod 的日志观察进度：

```
# kubectl logs kibana-logging-56c4d58dcd-c2dtk -n kube-system

{"type":"log","@timestamp":"2018-08-18T00:02:01Z","tags":
["status","plugin:kibana@6.2.4","info"],"pid":1,"state":"green","message":"Status
changed from uninitialized to green -
Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
{"type":"log","@timestamp":"2018-08-18T00:02:01Z","tags":
["status","plugin:elasticsearch@6.2.4","info"],"pid":1,"state":"yellow","message":"Status
 changed from uninitialized to yellow - Waiting for
Elasticsearch","prevState":"uninitialized","prevMsg":"uninitialized"}
{"type":"log","@timestamp":"2018-08-18T00:02:01Z","tags":
["status","plugin:timelion@6.2.4","info"],"pid":1,"state":"green","message":"Status
changed from uninitialized to green -
Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
{"type":"log","@timestamp":"2018-08-18T00:02:01Z","tags":
["status","plugin:console@6.2.4","info"],"pid":1,"state":"green","message":"Status
changed from uninitialized to green -
Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
{"type":"log","@timestamp":"2018-08-18T00:02:01Z","tags":
["status","plugin:metrics@6.2.4","info"],"pid":1,"state":"green","message":"Status
```

```
changed from uninitialized to green -
Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
{"type":"log","@timestamp":"2018-08-18T00:02:01Z","tags":
["listening","info"],"pid":1,"message":"Server running at http://0:5601"}
{"type":"log","@timestamp":"2018-08-18T00:02:04Z","tags":
["status","plugin:elasticsearch@6.2.4","error"],"pid":1,"state":"red","message":"Status
changed from yellow to red - Request Timeout after
3000ms","prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}
```

访问 kibana

通过 kube-apiserver 访问：

```
# kubectl cluster-info

Kubernetes master is running at https://192.168.251.101:6443
Elasticsearch is running at https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/elasticsearch-logging/proxy
Heapster is running at https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/heapster/proxy
Kibana is running at https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/kibana-logging/proxy
KubeDNS is running at https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
monitoring-grafana is running at https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/monitoring-grafana/proxy
monitoring-influxdb is running at https://192.168.251.101:6443/api/v1/namespaces/kube-system/services/monitoring-influxdb:http/proxy
```

浏览器访问 URL：

http://192.168.251.101:8080/api/v1/namespaces/kube-system/services/kibana-logging/proxy/

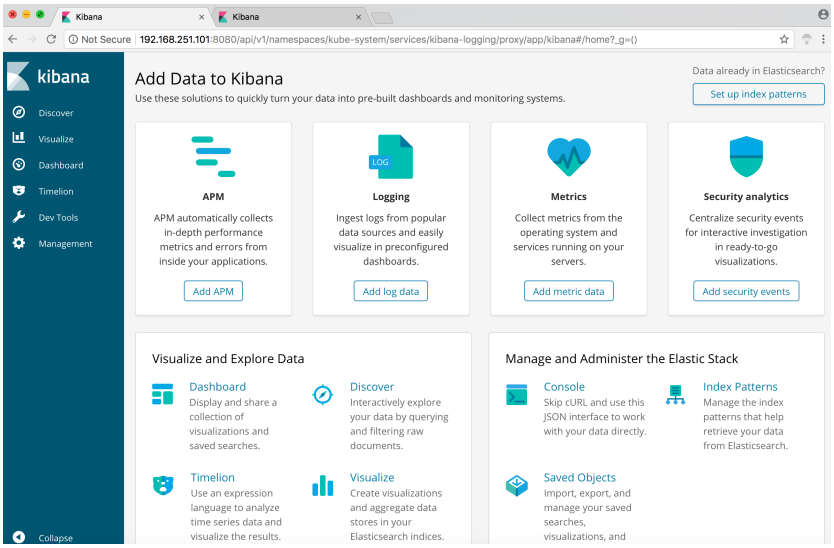
通过 kubectl proxy 访问：

创建代理

```
# kubectl proxy --address='192.168.251.101' --port=8086 --accept-hosts='^*$'
Starting to serve on 192.168.251.101:8086
```

浏览器访问 URL：

http://192.168.251.101:8086/api/v1/namespaces/kube-system/services/kibana-logging/proxy/



点击 右上角 set up index patterns **页面创建一个 index（相当于 mysql 中的一个 database）**