Elastic Search – EFK

helm install elasticsearch elastic/elasticsearch \

--set replicas=1 \

--set resources.requests.memory="512Mi" \

--set resources.requests.cpu="500m" \

--set persistence.enabled=false \

--set service.type=LoadBalancer \

--namespace logging

kubectl get secrets --namespace=logging elasticsearch-master-credentials -ojsonpath='{.data.username}' | base64 -d

elastic

kubectl get secrets --namespace=logging elasticsearch-master-credentials -ojsonpath='{.data.password}' | base64 -d

U40saeUHGTzrdacK

helm install kibana --set service.type=LoadBalancer elastic/kibana -n logging

kubectl get pods -n logging

kubectl get pods --namespace=logging -l release=kibana -w

kubectl get svc -n logging

helm repo add fluent <https://fluent.github.io/helm-charts>

helm repo update

--copy the password for ElasticSearch and update it in fluentbit manifest---

kubectl get secrets --namespace=logging elasticsearch-master-credentials -ojsonpath='{.data.password}' | base64 -d

vi fluentbit-values.yaml



helm install fluent-bit fluent/fluent-bit -f fluentbit-values.yaml -n logging

kubectl get pods -n logging

kubectl get svc -n logging

A diagram of a company

Description automatically generated

Uninstall:

helm list -n logging

helm uninstall fluent-bit -n logging

helm uninstall kibana -n logging

helm uninstall elasticsearch -n logging

kubectl get all -n logging

kubectl delete namespace logging

kubectl get pvc -n logging

kubectl delete pvc <pvc-name> -n logging

kubectl get pv

kubectl delete pv <pv-name>

Manuall Setup

<https://github.com/Bhoopesh123/efk-setup>

To Generate some logs – run counter.yaml

apiVersion: v1

kind: Pod

metadata:

name: counter

spec:

containers:

- name: count

image: busybox

args: [/bin/sh, -c,

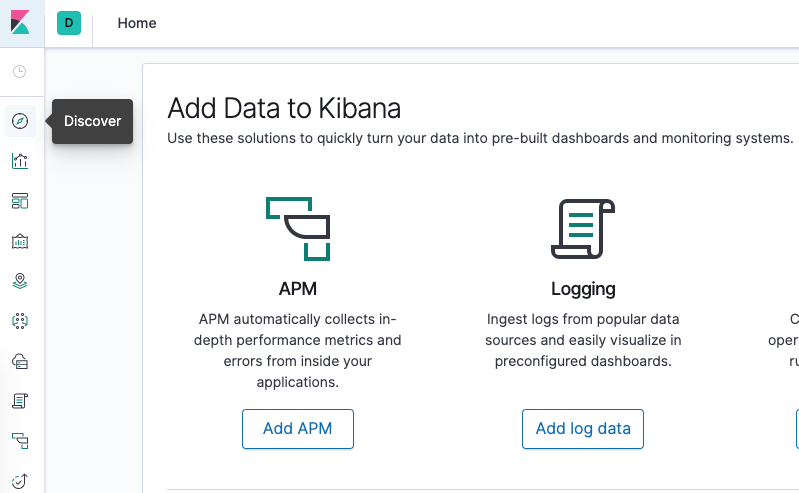
'i=0; while true; do echo "$i: $(date)"; i=$((i+1)); sleep 1; done']

kubectl apply counter.yaml

A diagram of a software company

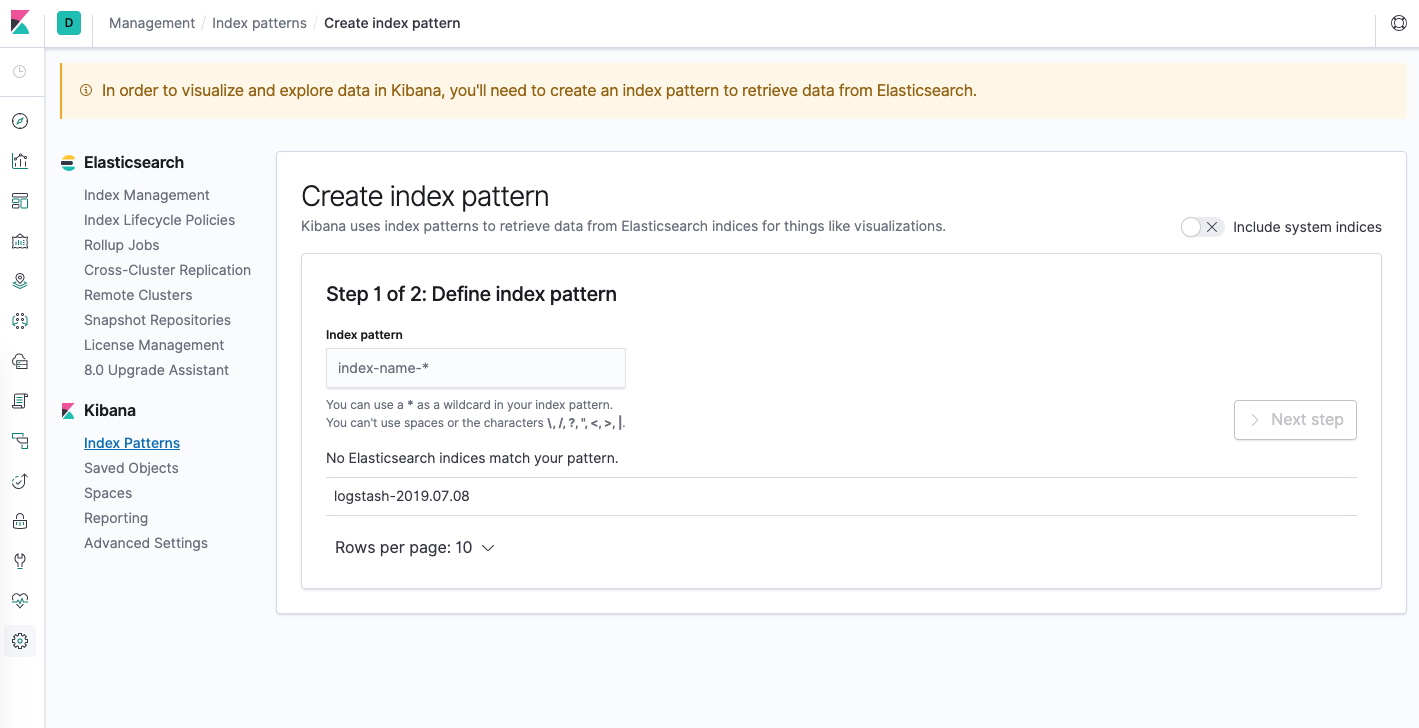
Description automatically generated

Click on **Discover** in the left-hand navigation menu:



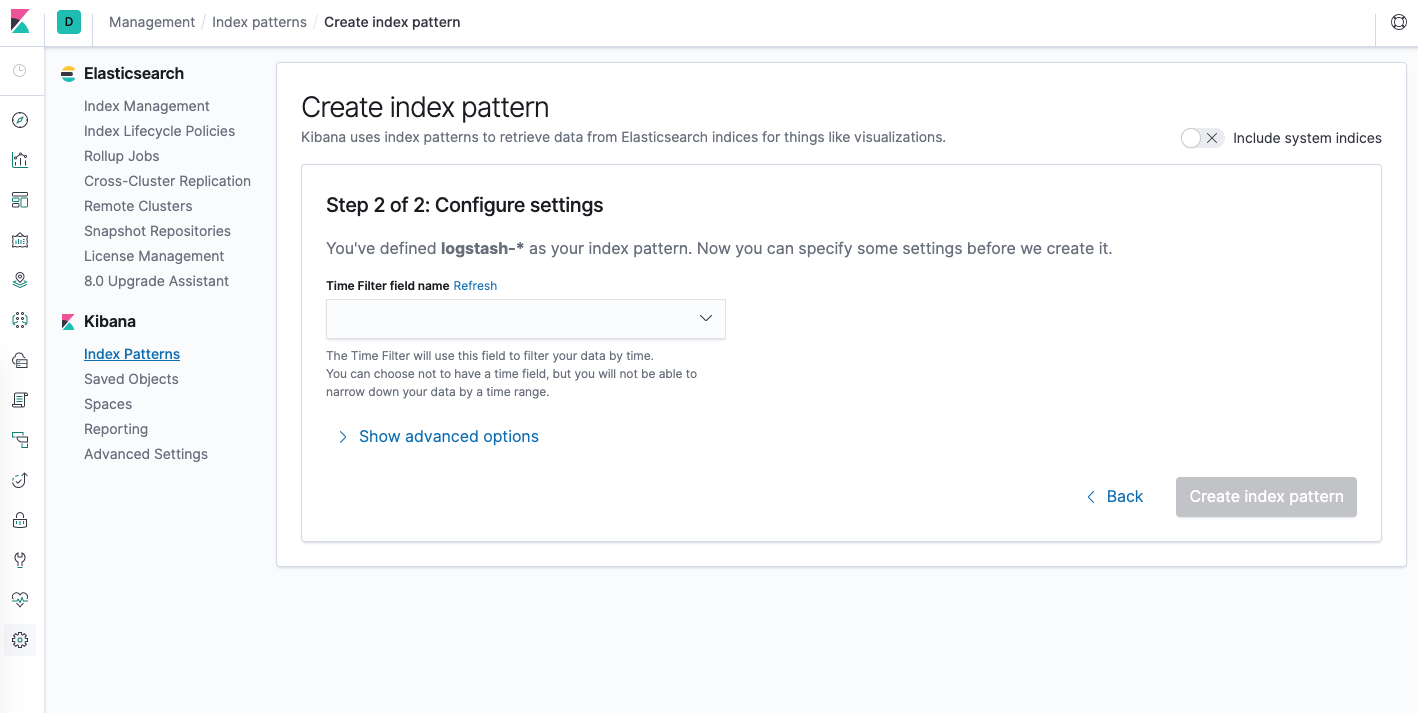
You should see the following configuration window:

Stack Management -🡪 DataViews🡪Create new



This allows you to define the Elasticsearch indices you’d like to explore in Kibana. To learn more, consult [Defining your index patterns](https://www.elastic.co/guide/en/kibana/current/tutorial-define-index.html) in the official Kibana docs. For now, we’ll just use the logstash-\* wildcard pattern to capture all the log data in our Elasticsearch cluster. Enter logstash-\* in the text box and click on **Next step**.

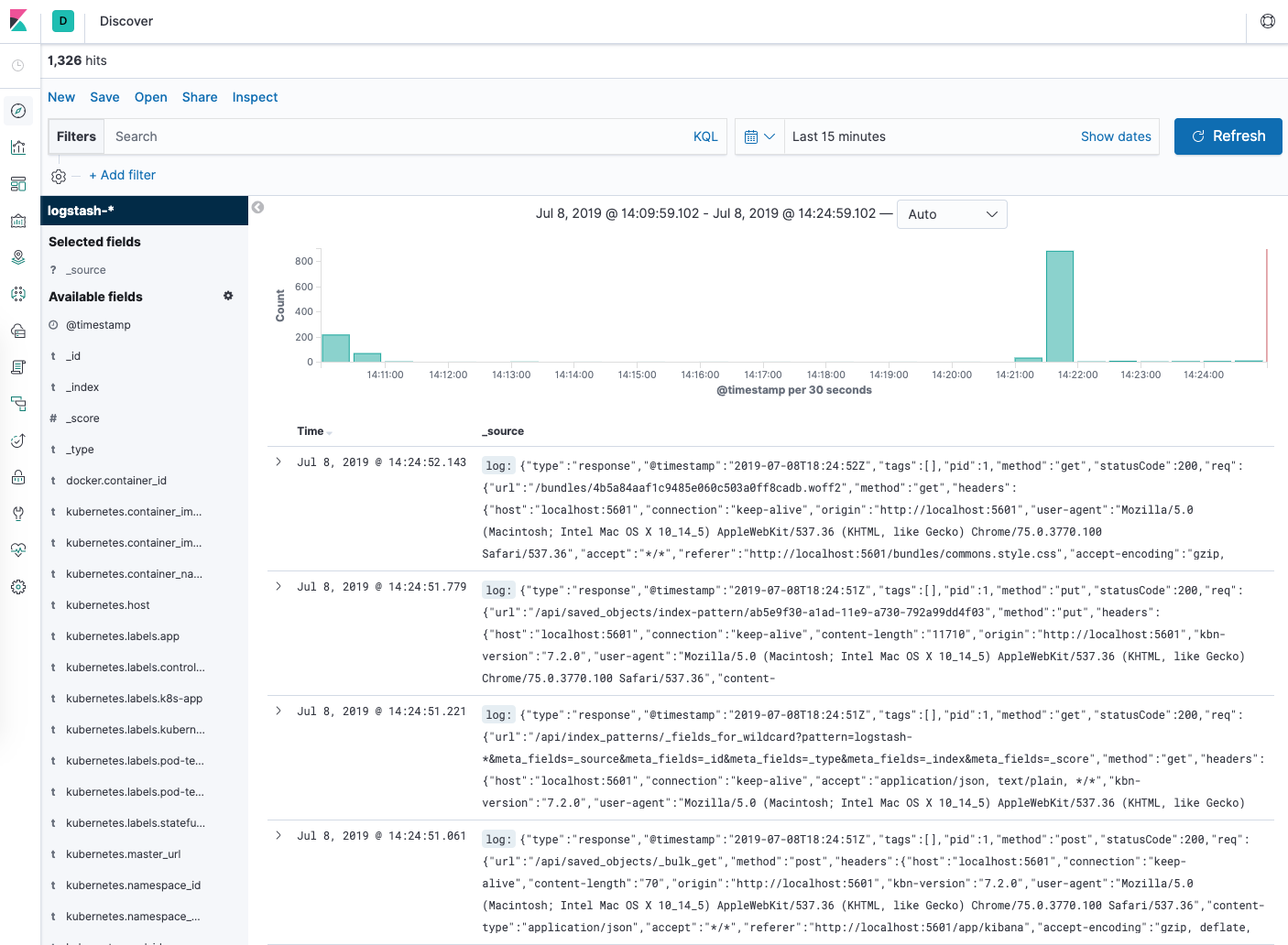
You’ll then be brought to the following page:



This allows you to configure which field Kibana will use to filter log data by time. In the dropdown, select the [**@timestamp**](https://www.digitalocean.com/community/users/timestamp) field, and hit **Create index pattern**.

Now, hit **Discover** in the left hand navigation menu.

You should see a histogram graph and some recent log entries:



At this point you’ve successfully configured and rolled out the EFK stack on your Kubernetes cluster. To learn how to use Kibana to analyze your log data, consult the [Kibana User Guide](https://www.elastic.co/guide/en/kibana/current/index.html).

In the next optional section, we’ll deploy a simple counter Pod that prints numbers to stdout, and find its logs in Kibana.