ElasticSearch：

下载 ElasticSearch 安装包：
Linux：https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.0.0-linux-x86_64.tar.gz
Windows：https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.0.0-windows-x86_64.zip

解压：tar -xzf elasticsearch-7.0.0-linux-x86_64.tar.gz
编辑 ElasticSearch 配置文件 elasticsearch-7.0.0/config/elasticsearch.yml，修改以下参数内容（示例）：

```
cluster.name: ck-elasticsearch
node.name: node-01
network.host: 10.0.1.4
discovery.seed_hosts: ["127.0.0.1"]
cluster.initial_master_nodes: ["node-01"]
```

其中 network.host 的 ip 地址为 VM 的 ifconfig 得到的 eth0 地址（azure VM 对外访问的 nic 内网地址）
切换到 root，修改系统 limit 参数：
运行：vi /etc/sysctl.conf，添加：

```
vm.max_map_count=655350
```

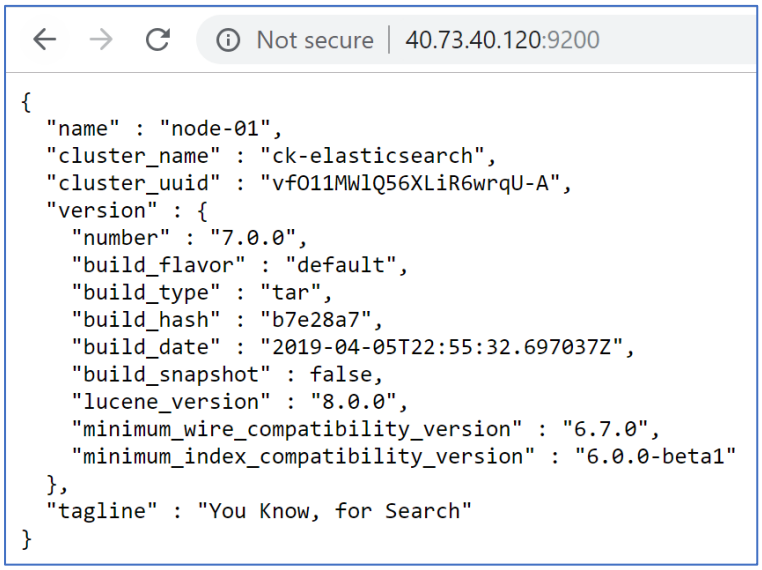
保存后运行 **sysctl -p**

运行：vi /etc/security/limits.conf，加入：

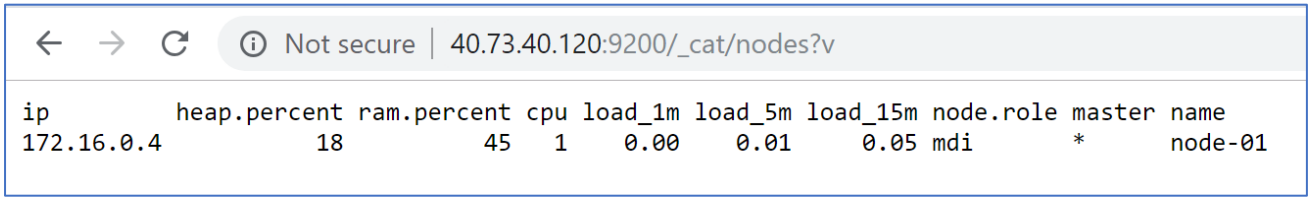
```
* soft nofile 655350
* hard nofile 655350
```

重新打开一个 ssh 终端，到 elasticsearch-7.0.0/bin 目录下，运行：**./elasticsearch -d**

到 azure portal 上，放开 ElasticSearch 所在虚机的 9200 端口的入站允许
使用浏览器或 curl 访问 <VM_ip>:9200，应看到如下内容：



访问接口 http://40.73.40.120:9200/_cat/nodes?v，看到如下内容则 ElasticSearch 安装成功：



Kibana：

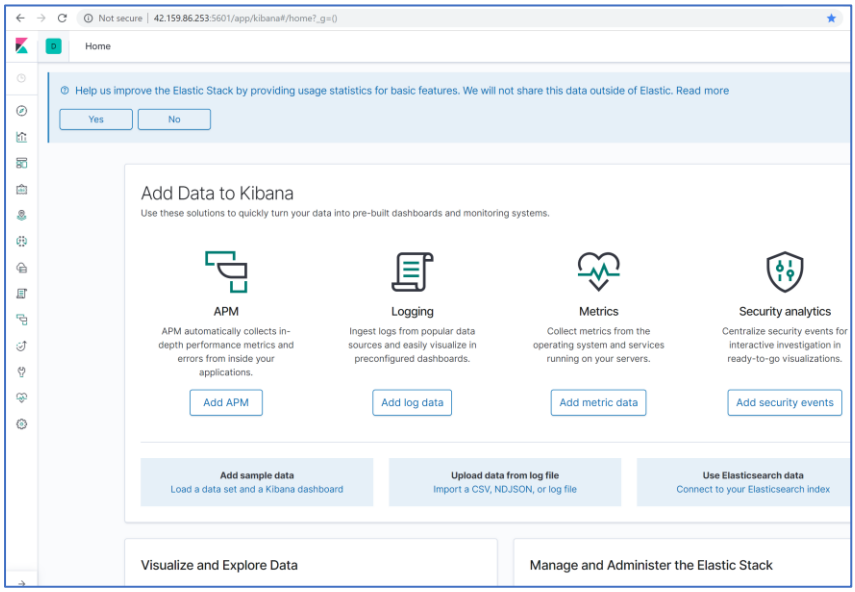
下载 Kibana 安装包：
Linux：https://artifacts.elastic.co/downloads/kibana/kibana-7.0.0-linux-x86_64.tar.gz
Windows：https://artifacts.elastic.co/downloads/kibana/kibana-7.0.0-windows-x86_64.zip

解压：tax -xzf kibana-7.0.0-linux-x86_64.tar.gz
编辑 ElasticSearch 配置文件 kibana-7.0.0-linux-x86_64/config/kibana.yml，修改以下参数内容（示例）：

```
server.port: 5601
server.host: "10.0.1.4"
elasticsearch.hosts: ["http://40.73.40.120:9200"]
kibana.index: ".kibana"
server.ssl.enabled: false
```

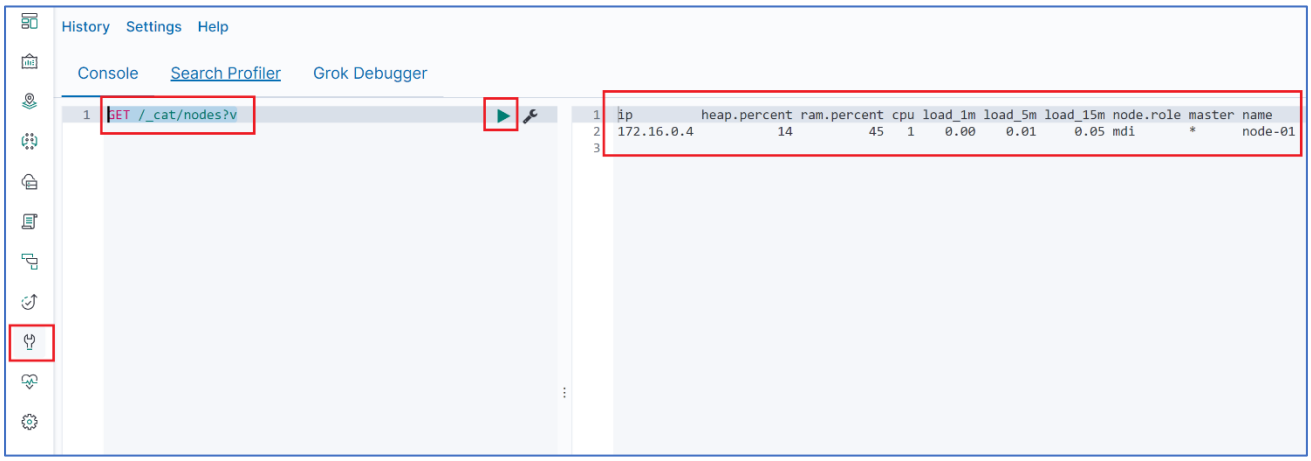
其中 server.host 的 ip 地址为 VM 的 ifconfig 得到的 eth0 地址（azure VM 对外访问的 nic 内网地址）
elasticsearch.hosts 为 kibana 可以访问到的 ElasticSearch 服务的地址和 9200 端口

到 azure portal 上，放开 Kibana 所在虚机的 5601 端口的入站允许
到 kibana-7.0.0-linux-x86_64/bin 目录下，运行 **nohup ./kibana &**
使用浏览器或 curl 访问 <VM_ip>:5601，看到如下内容则 Kibana 安装成功：



点击左侧工具栏的“Dev Tools”，执行 API “GET /_cat/nodes?v”，看到如下信息和以上 ElasticSearch 安装后执

行 API 的内容一致，则 Kibana 和 ElasticSearch 集成完毕：



Fluent-bit

执行命令创建 dev namespace （namespace 按实际需要命名）
kubectl create namespace dev

Git clone https://github.com/fluent/fluent-bit-kubernetes-logging 到本地
进入 fluent-bit-kubernetes-logging\ 目录，修改以下文件中的 namespace 值为 dev：
fluent-bit-role-binding.yaml
fluent-bit-service-account.yaml

执行：
kubectl apply -f fluent-bit-service-account.yaml
kubectl apply -f fluent-bit-role.yaml
kubectl apply -f fluent-bit-role-binding.yaml

进入 fluent-bit-kubernetes-logging\output\elasticsearch\ 目录，根据实际情况修改文件 fluent-bit-configmap.yaml 中以下 highlight 部分：

```
  filter-kubernetes.conf: |
    [FILTER]
        Name                kubernetes
        Match               kube.*
        Kube_URL https://ptfm-aks-u-ptfm-rsgp-uat-071a5d-3d6cbed2.hcp.chinaeast2.cx.prod.service.azk8s.cn:443
        Kube_CA_File        /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        Kube_Token_File     /var/run/secrets/kubernetes.io/serviceaccount/token
        Merge_Log           On
        K8S-Logging.Parser  On
        K8S-Logging.Exclude Off

  output-elasticsearch.conf: |
```

```
[OUTPUT]
    Name            es
    Match           *
    Host            42.159.86.253
    Port            9200
    Logstash_Format On
    Logstash_Prefix aks-fluentbit
    Replace_Dots    On
    Retry_Limit     False
```

进入 fluent-bit-kubernetes-logging\output\elasticsearch\ 目录，根据实际情况修改文件 fluent-bit-ds.yaml 中以下 highlight 部分：

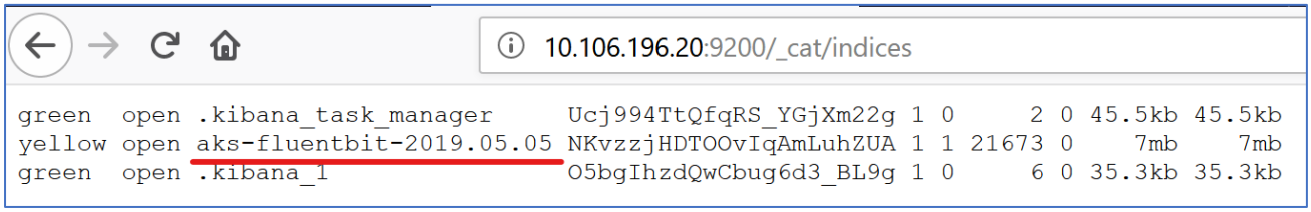
```
spec:
  containers:
  - name: fluent-bit
    image: fluent/fluent-bit:1.0.6
    imagePullPolicy: Always
    ports:
      - containerPort: 2020
    env:
    - name: FLUENT_ELASTICSEARCH_HOST
      value: "10.106.196.20"
    - name: FLUENT_ELASTICSEARCH_PORT
      value: "9200"
    volumeMounts:
    - name: varlog
      mountPath: /var/log
    - name: varlibdockercontainers
      mountPath: /var/lib/docker/containers
      readOnly: true
    - name: fluent-bit-config
      mountPath: /fluent-bit/etc/
  terminationGracePeriodSeconds: 10
```

执行命令：
kubectl apply -f fluent-bit-configmap.yaml
kubectl apply -f fluent-bit-ds.yaml

使用以下命令检查 fluent-bit 正常运行：

```
F:\AKS-env-setup>kubectl get daemonset -n dev
NAME         DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
fluent-bit   3         3         3       3            3           <none>          52m
```

稍等片刻，到浏览器访问 http://10.106.196.20:9200/_cat/indices，得到以下输出：



其中 aks-fluentbit-2019.05.05 是 fluentbit 收集 AKS 节点上的容器日志，在 ElasticSearch 中创建索引并上传日志内容。

至此 fluentbit 配置完毕。

Filebeat

注意 filebeat 和 fluent-bit 都是用来收集日志，并发送到 ElasticSearch，因此运行两者之一就可以。

参考 yaml 文件:
https://github.com/kylercai/kylercRepo/blob/master/aks/efk-logging/filebeat-kubernetes.yaml

此 yaml 文件定义以 daemonset 的方式运行 filebeat，同时创建运行 filebeat daemonset 所需的 service account、cluster roe、cluster role binding、configmap 等。

参考以下 highlight 部分对 yaml 文件内容进行定制，其中 multiline.*内容定义 filebeat 对多行日志进行合并显示。

```
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: filebeat
  namespace: kube-system
  labels:
    k8s-app: filebeat
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: filebeat
  labels:
    k8s-app: filebeat
rules:
- apiGroups: [""] # "" indicates the core API group
  resources:
  - namespaces
```

```
  - pods
  verbs:
  - get
  - watch
  - list
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: filebeat
subjects:
- kind: ServiceAccount
  name: filebeat
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: filebeat
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: filebeat-config
  namespace: kube-system
  labels:
    k8s-app: filebeat
data:
  filebeat.yml: |-
    filebeat.inputs:
    - type: log
      enabled: true
      paths:
        - /var/log/containers/*.log

    filebeat.config:
      inputs:
        # Mounted `filebeat-inputs` configmap:
        path: ${path.config}/inputs.d/*.yml
        # Reload inputs configs as they change:
        reload.enabled: false
      modules:
        path: ${path.config}/modules.d/*.yml
        # Reload module configs as they change:
        reload.enabled: false
```

```
    # To enable hints based autodiscover, remove `filebeat.config.inputs` configuration and uncomment this:
    #filebeat.autodiscover:
    #  providers:
    #    - type: kubernetes
    #      hints.enabled: true

    processors:
      - add_cloud_metadata:

    output.elasticsearch:
      hosts: ["42.159.86.253:9200"]
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: filebeat-inputs
  namespace: kube-system
  labels:
    k8s-app: filebeat
data:
  kubernetes.yml: |-
    - type: docker
      containers.ids:
      - "*"
      multiline.pattern: '^[[:space:]]|^Caused by:'
      multiline.negate: false
      multiline.match: after
      processors:
        - add_kubernetes_metadata:
            in_cluster: true
---
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: filebeat
  namespace: kube-system
  labels:
    k8s-app: filebeat
spec:
  template:
    metadata:
      labels:
        k8s-app: filebeat
```

```
spec:
  serviceAccountName: filebeat
  terminationGracePeriodSeconds: 30
  containers:
  - name: filebeat
    image: docker.elastic.co/beats/filebeat:7.1.0
    args: [
      "-c", "/etc/filebeat.yml",
      "-e",
    ]
    env:
    - name: ELASTICSEARCH_HOST
      value: "42.159.86.253"
    - name: ELASTICSEARCH_PORT
      value: "9200"
    securityContext:
      runAsUser: 0
      # If using Red Hat OpenShift uncomment this:
      #privileged: true
    resources:
      limits:
        memory: 200Mi
      requests:
        cpu: 100m
        memory: 100Mi
    volumeMounts:
    - name: config
      mountPath: /etc/filebeat.yml
      readOnly: true
      subPath: filebeat.yml
    - name: inputs
      mountPath: /usr/share/filebeat/inputs.d
      readOnly: true
    - name: data
      mountPath: /usr/share/filebeat/data
    - name: varlibdockercontainers
      mountPath: /var/lib/docker/containers
      readOnly: true
  volumes:
  - name: config
    configMap:
      defaultMode: 0600
      name: filebeat-config
  - name: varlibdockercontainers
```

```
      hostPath:
        path: /var/lib/docker/containers
    - name: inputs
      configMap:
        defaultMode: 0600
        name: filebeat-inputs
    # data folder stores a registry of read status for all files, so we don't send everything again on a
Filebeat pod restart
    - name: data
      hostPath:
        path: /var/lib/filebeat-data
        type: DirectoryOrCreate


---
```

其中 multiline 设置:

multiline.pattern: '^[[:space:]]|^Caused by:'
multiline.negate: false
multiline.match: after

含义为:

符合 pattern 的行，即以空格和"Caused by:"开头的行，将被合并到上一行之后。

多行日志合并设置应根据自身的日志合并需要调整。multiline 还有以下 option 可供调整:

- multiline.flush_pattern
  指定正则表达式去匹配指定的行作为 multline-message 的结束，刷新的内存，开始匹配新的多行
- multiline.max_lines
  指定合并最大行数
- multiline.timeout
  设定一个超时时间，在时间结束后，即使没有匹配到新 pattern 来启动新事件，Filebeat 也会发送多行事件。默认值是 5 秒

执行命令:

```
λ kubectl apply -f "filebeat-kubernetes.yaml"
serviceaccount/filebeat created
clusterrole.rbac.authorization.k8s.io/filebeat created
clusterrolebinding.rbac.authorization.k8s.io/filebeat created
configmap/filebeat-config created
configmap/filebeat-inputs created
daemonset.extensions/filebeat created
```

使用以下命令检查 filebeat 正常运行:

```
λ kubectl get daemonset -n kube-system
NAME                      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR                 AGE
azure-cni-networkmonitor  3         3         3       3            3           beta.kubernetes.io/os=linux   29d
azure-ip-masq-agent       3         3         3       3            3           beta.kubernetes.io/os=linux   29d
filebeat                  3         3         3       3            3           <none>                        12h
kube-proxy                3         3         3       3            3           beta.kubernetes.io/os=linux   29d
kube-svc-redirect         3         3         3       3            3           beta.kubernetes.io/os=linux   29d
omsagent                  3         3         3       3            3           beta.kubernetes.io/os=linux   29d
```

稍等片刻，到浏览器访问 http://42.159.86.253:9200/_cat/indices，得到以下输出（highlight filebeat 在 ElasticSearch 中创建的 index，并用于上传日志）:

Not secure | 42.159.86.253:9200/_cat/indices

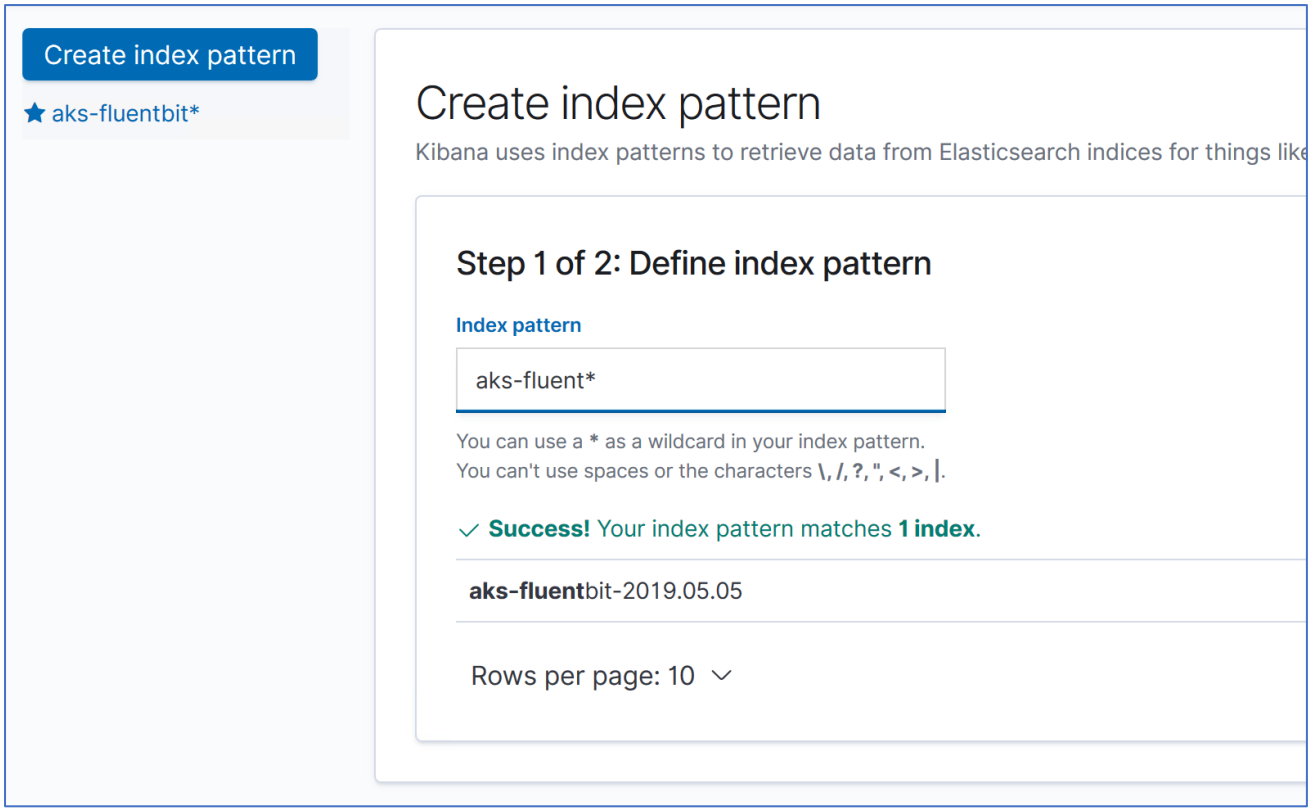
Apps system devops aks sf study AI info dev go

```
yellow open aks-fluentbit-2019.05.20         aGZkItrhRDKbDxXF1Z-JdA 1 1  914916 0 173.5mb 173.5mb
yellow open aks-fluentbit-2019.05.21         00giDzwxRBadqIf8kmmRng 1 1  415048 0  80.1mb  80.1mb
yellow open aks-fluentbit-2019.05.13         bKH-XG6NSauUIJuZmvMO7w 1 1  915242 0 174.7mb 174.7mb
yellow open aks-fluentbit-2019.05.08         ya2HY8bMRU2W8R-ykhlNOQ 1 1  915047 0 173.4mb 173.4mb
yellow open aks-fluentbit-2019.05.18         yI7HrqdKQ4iuypfVFiiNKA 1 1  915854 0 174.4mb 174.4mb
yellow open aks-fluentbit-2019.05.06         jvE20VkCTUmMGe1LNrQQAA 1 1  914914 0 173.9mb 173.9mb
yellow open aks-fluentbit-2019.05.10         08VtEU5bTN-4AV0ob63xuQ 1 1  914462 0 172.4mb 172.4mb
yellow open aks-fluentbit-2019.05.19         o-XPtmK0STSEi8AJ8SrBgg 1 1  914903 0 174.1mb 174.1mb
yellow open aks-fluentbit-2019.05.14         BMNL74mtS0OhYLyzlskGFA 1 1  914347 0 172.9mb 172.9mb
green  open .kibana_1                        wE4S--JuRiyylEiqmOZZbw 1 0       6 0  91.1kb  91.1kb
yellow open aks-fluentbit-2019.05.11         kwMNNdpxTKqnBro-UlSPrQ 1 1  914825 0 173.8mb 173.8mb
yellow open aks-fluentbit-2019.05.15         LEubQZRbQGeXTIwfYA_MYw 1 1  914102 0 173.9mb 173.9mb
yellow open aks-fluentbit-2019.05.05         bpv6MaZrTHmQiMgSFlFzvA 1 1  370304 0  70.1mb  70.1mb
yellow open aks-fluentbit-2019.05.12         CZNCR3C7T4mwElN32xuMFQ 1 1  914356 0 173.5mb 173.5mb
yellow open aks-fluentbit-2019.05.07         PvNq0DfbTEC18MSU5_4Wyg 1 1  916966 0 175.1mb 175.1mb
green  open .kibana_task_manager             WYZPf_WIQ7evnSRg6ZUCWg 1 0       2 0  21.4kb  21.4kb
yellow open aks-fluentbit-2019.05.16         IXPUA-DARE2DerxjNvFizg 1 1  916239 0 175.2mb 175.2mb
yellow open aks-fluentbit-2019.05.09         NGeKiWWcTpm7XiVi0w22mw 1 1  914482 0 173.4mb 173.4mb
yellow open aks-fluentbit-2019.05.17         JXBiugMmSCSycmrr-nb7zA 1 1  914555 0 173.3mb 173.3mb
yellow open filebeat-7.1.0-2019.05.21-000001 OUrsv5NsQ7aygBjbycynlA 1 1 1245225 0 390.8mb 390.8mb
```

至此 filebeat 配置完毕。

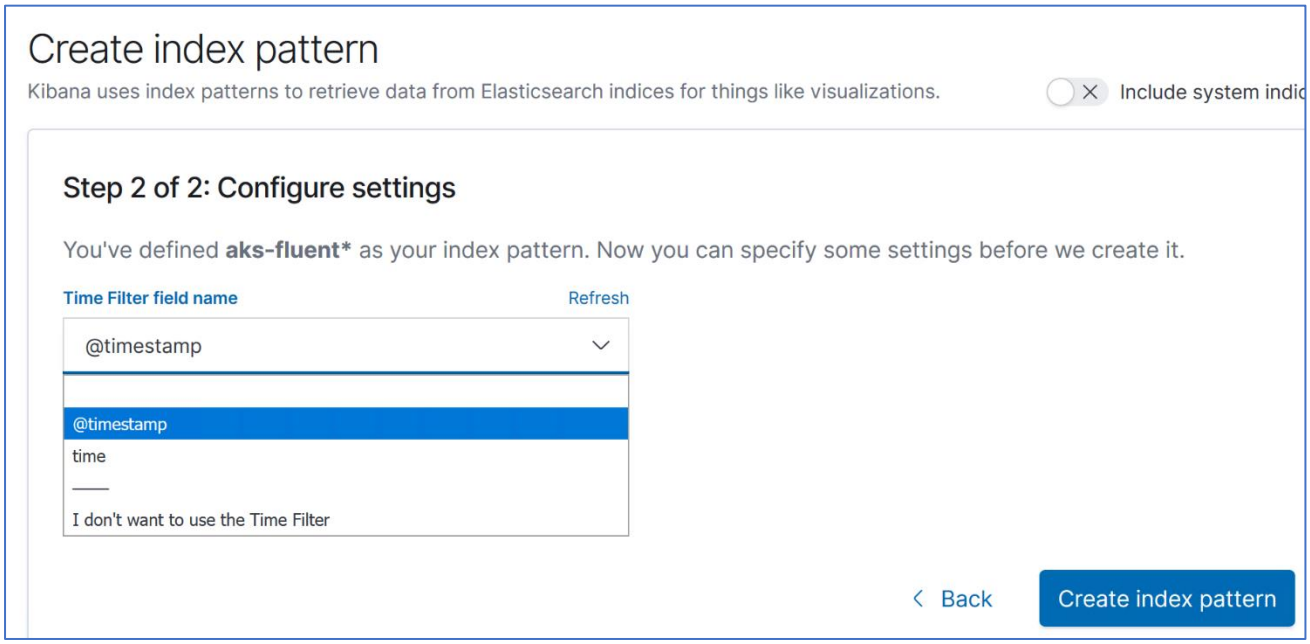
配置 Kibana 查询日志

进入 Kibana 首页面板，选择左侧工具栏最下方 Management（齿轮图标）
点击 ElasticSearch -> Index Management，应能看到 ElasticSearch 中已有的索引
点击 Kibana -> Index Patterns -> Create index pattern，创建索引模式，例如:

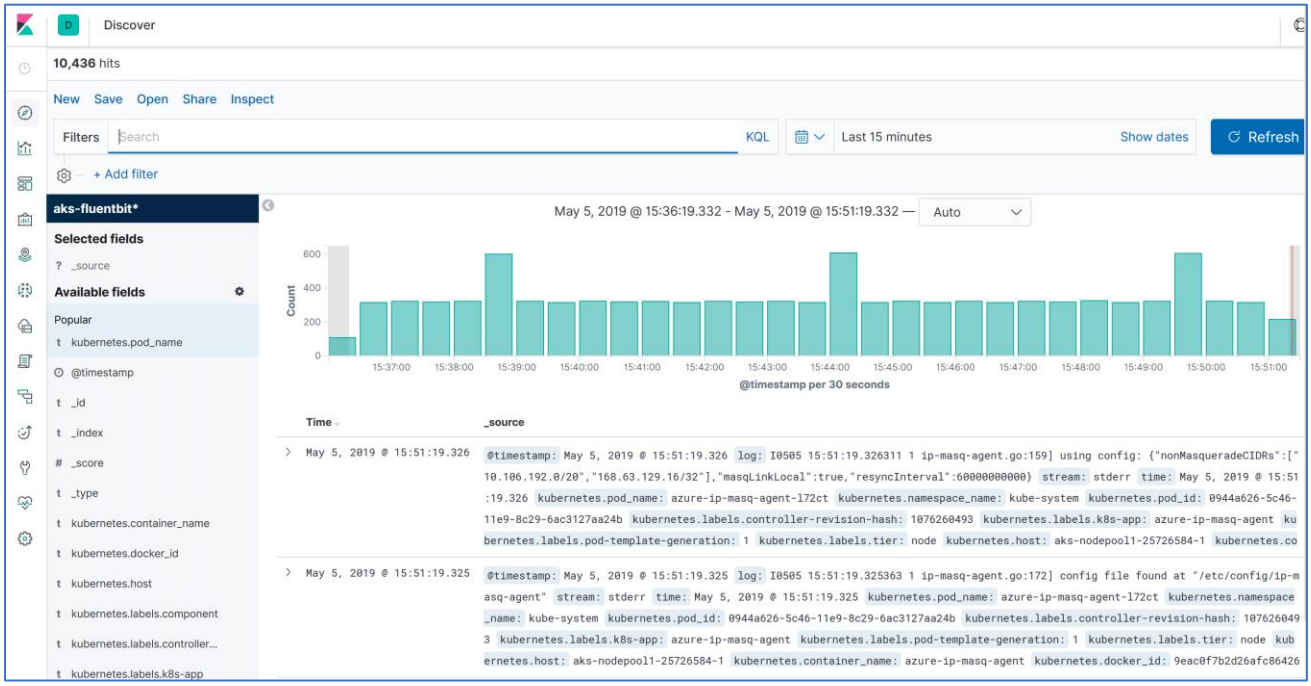


如果日志收集端选择 filebeat，则在此处的 index pattern 可以选择“filebeat-*”。

点击 Next step，进行配置后点击 Create index pattern，例如：



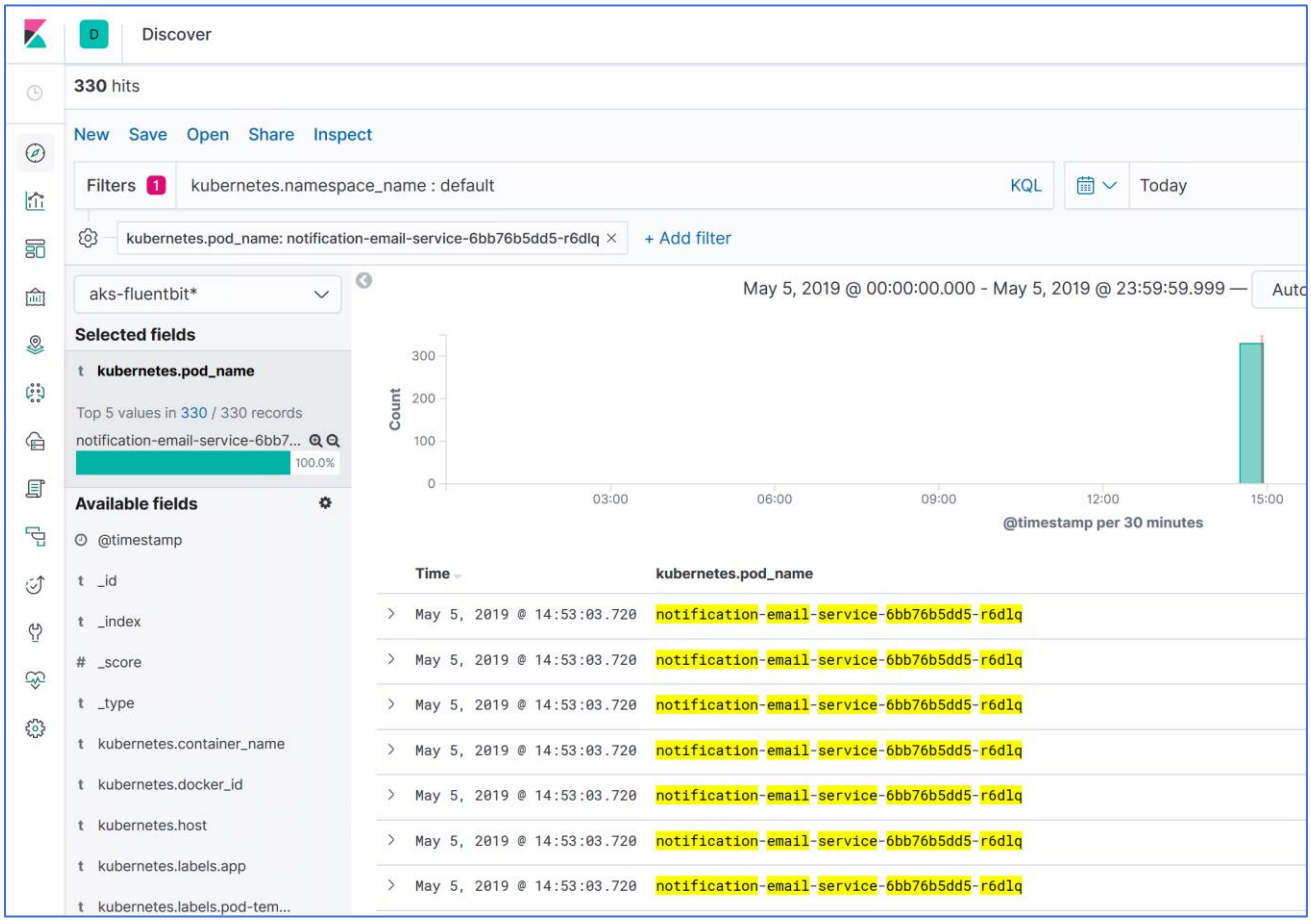
之后，到工具栏左侧选择 Discover （最上方图标），选择对应的 index pattern，Kibana 将自动获取相应的日志记录，例如：



根据查询需要，设置过滤条件，缩小日志查询范围，可以使用的方式有：

- 在 Filters 中输入查询条件，例如 kubernetes.namespace_name：default 为查询 kubernetes 中 default namespace 中的内容 （请查阅 KQL 查询语法）
- 在 Available fields 中移动鼠标到相应 fields，点击 add，相应的 field 会被添加到 Selected fields，点击此 field 会显示所有的 value，选择相应的取值进行过滤

例如：



过滤条件选择了：
default namespace
Kubernetes pod 名字为 notification-email-service-xxxxx

如果日志收集端选择 filebeat，并且配置了 multiline.pattern 等多行日志合并选项，则可在 Kibana 中看到相关多行日志合并显示的效果：

标准输出 stdout：

```
t  input.type                          docker
t  kubernetes.container.name           javademo
t  kubernetes.labels.app               javademo
t  kubernetes.labels.pod-template-hash 3527011171
t  kubernetes.namespace                default
t  kubernetes.node.name                aks-agentpool-32361114-2
t  kubernetes.pod.name                 javademo-796c4555c5-xmptr
t  kubernetes.pod.uid                  11358850-7bdf-11e9-9ffe-b226bbbe90d4
t  kubernetes.replicaset.name          javademo-796c4555c5
t  log.file.path                       /var/lib/docker/containers/ee0f6c822333a2e5e76d51119a7d5b01d2e0eb0006b7ddb0361efbc8c8a813e4/ee0f6c822333a2e5e76d51119
                                       1d2e0eb0006b7ddb0361efbc8c8a813e4-json.log
t  log.flags                           multiline
#  log.offset                          24,601
t  message                             This log simulates the multiple line log info collected&combined by filebeats..
                                        line 1 begins with space..
                                        line 2 begins with space..
                                        line 3 begins with space..
                                          at line 4 begins with space followed by word at..
                                          ... line 5 begins with space followed by ...
                                       Caused by: line 6 begins with words Caused by:..
t  stream                              stdout
   suricata.eve.timestamp              May 22, 2019 @ 00:33:06.464

May 22, 2019 @ 00:33:06.464   kubernetes.namespace: default @timestamp: May 22, 2019 @ 00:33:06.464 ecs.version: 1.0.0 agent.version: 7.1.0 agent.type: filebeat
agent.ephemeral_id: b4c3ebda-b01a-4030-9396-99126732a151 agent.hostname: filebeat-9d5b4 agent.id: bfdbbe87-9651-4c52-9011-f0a170abdf7b
message: java.io.IOException: This log simulates an exception thrown.. at
com.microsoft.azure.webapp.HelloWorldServlet.doGet(HelloWorldServlet.java:51) at javax.servlet.http.HttpServlet.service(HttpServlet.java:6
javax.servlet.http.HttpServlet.service(HttpServlet.java:742) at

Expanded document                                   View surrounding documents   View single do
```

标准错误 stderr:

```
t  kubernetes.replicaset.name          javademo-796c4555c5
t  log.file.path                       /var/lib/docker/containers/ee0f6c822333a2e5e76d51119a7d5b01d2e0eb0006b7ddb0361efbc8c8a813e4/ee0f6c822333a2e5e76d51119
                                       1d2e0eb0006b7ddb0361efbc8c8a813e4-json.log
t  log.flags                           multiline
#  log.offset                          25,410
t  message                             java.io.IOException: This log simulates an exception thrown..
                                               at com.microsoft.azure.webapp.HelloWorldServlet.doGet(HelloWorldServlet.java:51)
                                               at javax.servlet.http.HttpServlet.service(HttpServlet.java:635)
                                               at javax.servlet.http.HttpServlet.service(HttpServlet.java:742)
                                               at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)
                                               at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
                                               at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
                                               at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
                                               at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
                                               at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
                                               at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
                                               at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:493)
                                               at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:137)
                                               at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:81)
                                               at org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:660)
                                               at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:87)
                                               at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343)
                                               at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:798)
                                               at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:66)
                                               at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:808)
                                               at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1498)
                                               at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
                                               at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
                                               at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
                                               at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
                                               at java.lang.Thread.run(Thread.java:748)
t  stream                              stderr
   suricata.eve.timestamp              May 22, 2019 @ 00:33:06.464

May 22, 2019 @ 00:33:05.311   kubernetes.namespace: default @timestamp: May 22, 2019 @ 00:33:05.311 stream: stdout message: This log simulates the multiple line log
collected&combined by filebeats.. line 1 begins with space.. line 2 begins with space.. line 3 begins with space.. at line 4 begins with s
followed by word at.. ... line 5 begins with space followed by ... Caused by: line 6 begins with words Caused by:.. ecs.version: 1.0.0
agent.version: 7.1.0 agent.type: filebeat agent.ephemeral_id: b4c3ebda-b01a-4030-9396-99126732a151 agent.hostname: filebeat-9d5b4
agent.id: bfdbbe87-9651-4c52-9011-f0a170abdf7b log.offset: 19,693
```