Setting up logstash:-

1. sudo su
2. yum update -y
3. cd /root
4. wget https://download.elasticsearch.org/logstash/logstash/packages/centos/logstash-1.4.2-1_2c0f5a1.noarch.rpm
5. yum install logstash-1.4.2-1_2c0f5a1.noarch.rpm -y
6. rm -f logstash-1.4.2-1_2c0f5a1.noarch.rpm
7. nano /etc/logstash/conf.d/logstash.conf

Logstash config file:-

```
input {
  syslog {
    type => "syslog"
    port => 5544
 }
}

filter {
  # Remove the useless fields sent by syslog
  mutate {
    remove_field => ["timestamp", "host", "facility_label", "severity_label", "severity", "facility",
"priority"]
  }
}

output {
 stdout { }
 elasticsearch_http {
  host => "elasticsearchdata01"
  port => 9200
 }
}
```

Restart logstash.

Service logstash restart

Verify that the port is opened up in firewall and you can access the same in instance using telnet.

Setting up Elasticsearch:-
FYI:- The below documentation was used in Amazon cloud and thus cloud.aws and discovery can be ignored.

```
sudo su
yum update -y
cd /root
wget https://download.elasticsearch.org/elasticsearch/elasticsearch/elasticsearch-1.4.4.noarch.rpm
yum install elasticsearch-1.4.4.noarch.rpm -y
rm -f elasticsearch-1.4.4.noarch.rpm
cd /etc/elasticsearch
nano elasticsearch.yml
```

Config

```
cluster.name: niraj
cloud.aws.access_key: ACCESS_KEY_HERE
cloud.aws.secret_key: SECRET_KEY_HERE
cloud.aws.region: us-east-1
discovery.type: ec2
discovery.ec2.tag.Name: "niraj-elasticsearch"
http.cors.enabled: true
http.cors.allow-origin: "*"
```

Commands

```
service elasticsearch start
```

Attach the message logging policy to the API Proxy(Either request or response flow).

And use the below sample configs:-

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MessageLogging async="false" continueOnError="false" enabled="true" name="rsyslog">
   <DisplayName>rsyslog</DisplayName>
   <Syslog>
      <Message>[Request Message ="OrganizationName=  {organization.name}, ProxyName=
{apiproxy.name}, EnvironmentName=  {environment.name}, ProxyURL=  {proxy.url}, APIProxyRevision=
{apiproxy.revision}, ClientIP=  {target.ip}, ProxyBasePath=  {proxy.basepath}, URIPort= {virtualhost.port},
Call= {request.verb}, ProxyBasePath = {proxy.pathsuffix},{request.header.X-Forwarded-Proto}, Status
Code= {message.status.code}. "] Weather request for WOEID {request.queryparam.w}.</Message>
```

```
      <Host>xx.xx.xx.xx</Host>
      <Port>5544</Port>
      <Protocol>TCP</Protocol>
   </Syslog>
</MessageLogging>
```

All Done!!

PS: - You can use tcpdump -v port 5544 to verify that packets are being received on that port.