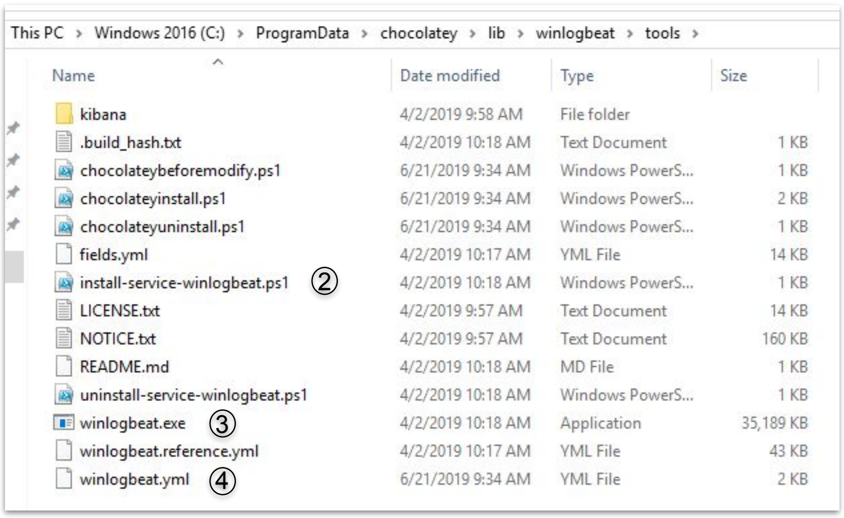
Configuring Winlogbeat


This PC › Windows 2016 (C:) › ProgramData › chocolatey › lib › winlogbeat › tools ›

Name	Date modified	Type	Size
kibana	4/2/2019 9:58 AM	File folder	
.build_hash.txt	4/2/2019 10:18 AM	Text Document	1 KB
chocolateybeforemodify.ps1	6/21/2019 9:34 AM	Windows PowerS...	1 KB
chocolateyinstall.ps1	6/21/2019 9:34 AM	Windows PowerS...	2 KB
chocolateyuninstall.ps1	6/21/2019 9:34 AM	Windows PowerS...	1 KB
fields.yml	4/2/2019 10:17 AM	YML File	14 KB
install-service-winlogbeat.ps1 ②	4/2/2019 10:18 AM	Windows PowerS...	1 KB
LICENSE.txt	4/2/2019 9:57 AM	Text Document	14 KB
NOTICE.txt	4/2/2019 9:57 AM	Text Document	160 KB
README.md	4/2/2019 10:18 AM	MD File	1 KB
uninstall-service-winlogbeat.ps1	4/2/2019 10:18 AM	Windows PowerS...	1 KB
winlogbeat.exe ③	4/2/2019 10:18 AM	Application	35,189 KB
winlogbeat.reference.yml	4/2/2019 10:17 AM	YML File	43 KB
winlogbeat.yml ④	6/21/2019 9:34 AM	YML File	2 KB

In this lab we'll trace through the workflow of shipping logs with Beats. There are many kinds of Beats. You'll find a list of Beats officially supported by Elastic here: https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html.

There are also Community Beats, which are contributed by third parties. Elastic maintains a list of Community Beats here: https://www.elastic.co/guide/en/beats/libbeat/current/community-beats.html.

We'll be looking at Winlogbeat, which is a light-weight data shipper that ships Windows Event Logs to either Logstash or directly into Elasticsearch. We'll trace through how to configure Winlogbeat on the Windows host to collect specific logs, how to ship them to Logstash, and finally how to configure Logstash to receive, parse and output the logs to Elasticsearch. We'll begin in the Windows VM.

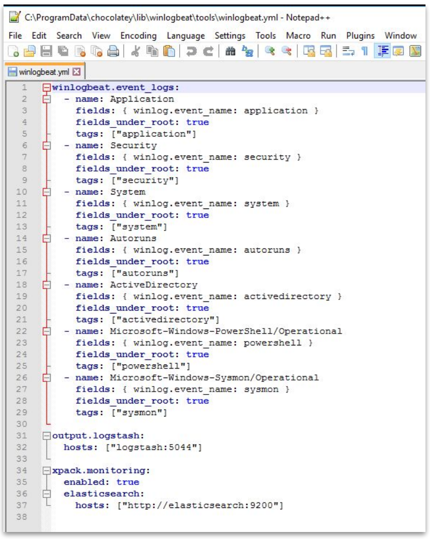
1) In the Windows VM, open the **winlogbeat** folder shortcut on the desktop. Alternatively, you can navigate to **C:\ProgramData\chocolatey\lib\winlogbeat\tools**. If you aren't installing with Chocolatey, the path would obviously be different.

2) Winlogbeat ships with a Powershell script called **install-service-winlogbeat.ps1**. This can be used to install Winlogbeat as a service. It's a simple script. If you're familiar with Powershell, feel free to look at the script and see how it's installed.

3) The **winlogbeat.exe** binary is run from this directory and is the only executable required to run Winlogbeat.

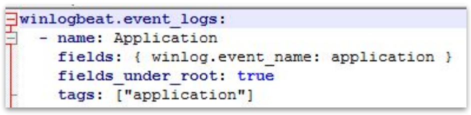
4) Winlogbeat's configuration is contained in the **winlogbeat.yml** file.

Configuring Winlogbeat


C:\ProgramData\chocolatey\lib\winlogbeat\tools\winlogbeat.yml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
winlogbeat.yml

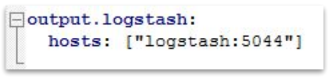
```yaml
winlogbeat.event_logs:
  - name: Application
    fields: { winlog.event_name: application }
    fields_under_root: true
    tags: ["application"]
  - name: Security
    fields: { winlog.event_name: security }
    fields_under_root: true
    tags: ["security"]
  - name: System
    fields: { winlog.event_name: system }
    fields_under_root: true
    tags: ["system"]
  - name: Autoruns
    fields: { winlog.event_name: autoruns }
    fields_under_root: true
    tags: ["autoruns"]
  - name: ActiveDirectory
    fields: { winlog.event_name: activedirectory }
    fields_under_root: true
    tags: ["activedirectory"]
  - name: Microsoft-Windows-PowerShell/Operational
    fields: { winlog.event_name: powershell }
    fields_under_root: true
    tags: ["powershell"]
  - name: Microsoft-Windows-Sysmon/Operational
    fields: { winlog.event_name: sysmon }
    fields_under_root: true
    tags: ["sysmon"]

output.logstash:
  hosts: ["logstash:5044"]

xpack.monitoring:
  enabled: true
  elasticsearch:
    hosts: ["http://elasticsearch:9200"]
```

②



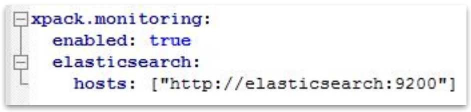
```yaml
winlogbeat.event_logs:
  - name: Application
    fields: { winlog.event_name: application }
    fields_under_root: true
    tags: ["application"]
```

③



```yaml
output.logstash:
  hosts: ["logstash:5044"]
```

④



```yaml
xpack.monitoring:
  enabled: true
  elasticsearch:
    hosts: ["http://elasticsearch:9200"]
```

Let's look at the different parts of the Winlogbeat configuration file **winlogbeat.yml**.

1) Right-click the file and select **Edit with Notepad++**

2) You probably noticed by the file extension this file is written in YAML format. We won't go into YAML syntax, but if you're having trouble understanding how the file is organized, it's worth a Google search. The first section in the file is the **winlogbeat.event_logs** section. Each Event Log we're shipping has an entry in this section. The first section is for the **Application** event log, indicated in the **name** field. This field is required.

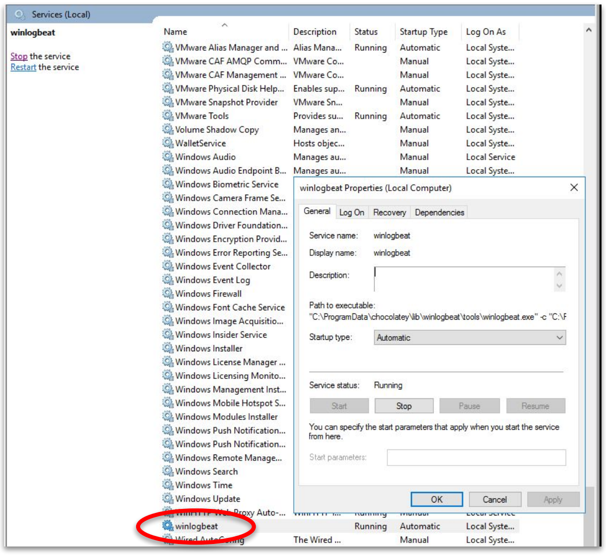
   Can you tell what other logs are being shipping?

   The other parameters are option, and explained in **winlogbeat.reference.yml** in the same directory.

3) The next section is the **output.logstash** section. This is where Winlogbeat ships the logs to. In this case, the hostname is logstash and the port is 5044. When you initially set-up the lab, an Ansible playbook configured DNS on the Windows Server and created an entry for "logstash" to point to your CentOS VM. An IP address would also work in this setting.

4) The final section is **xpack.monitoring**. Remember seeing the **Beats** section when we looked at the monitoring view in Kibana? This is how this was configured. When we ran the initial set-up, we also created a DNS entry for "elasticsearch," which also pointed to your CentOS VM.

The Winlogbeat Service



Services (Local)

winlogbeat

Stop the service
Restart the service

Name	Description	Status	Startup Type	Log On As
VMware Alias Manager and ...	Alias Mana...	Running	Automatic	Local Syste...
VMware CAF AMQP Comm...	VMware Co...		Manual	Local Syste...
VMware CAF Management ...	VMware Co...		Manual	Local Syste...
VMware Physical Disk Help...	Enables sup...	Running	Automatic	Local Syste...
VMware Snapshot Provider	VMware Sn...		Manual	Local Syste...
VMware Tools	Provides su...	Running	Automatic	Local Syste...
Volume Shadow Copy	Manages an...		Manual	Local Syste...
WalletService	Hosts objec...		Manual	Local Syste...
Windows Audio	Manages au...		Manual	Local Service
Windows Audio Endpoint B...	Manages au...		Manual	Local Syste...
Windows Biometric Service				
Windows Camera Frame Se...				
Windows Connection Mana...				
Windows Driver Foundation...				
Windows Encryption Provid...				
Windows Error Reporting Se...				
Windows Event Collector				
Windows Event Log				
Windows Firewall				
Windows Font Cache Service				
Windows Image Acquisitio...				
Windows Insider Service				
Windows Installer				
Windows License Manager ...				
Windows Licensing Monito...				
Windows Management Inst...				
Windows Mobile Hotspot S...				
Windows Modules Installer				
Windows Push Notification...				
Windows Push Notification...				
Windows Remote Manage...				
Windows Search				
Windows Time				
Windows Update				
WinHTTP Web Proxy Auto-...	WinHTTP ...	Running	Manual	Local Service
winlogbeat		Running	Automatic	Local Syste...
Wired AutoConfig	The Wired ...		Manual	Local Syste...

winlogbeat Properties (Local Computer)

General | Log On | Recovery | Dependencies

Service name: winlogbeat
Display name: winlogbeat
Description:
Path to executable:
"C:\ProgramData\chocolatey\lib\winlogbeat\tools\winlogbeat.exe" -c "C:\F
Startup type: Automatic

Service status: Running

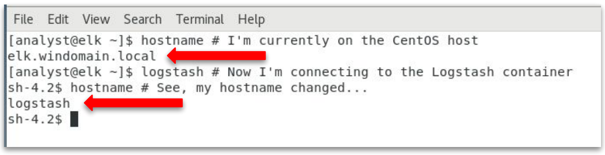
You can specify the start parameters that apply when you start the service from here.

Start parameters:

Any changes to **winlogbeat.yml** will require the winlogbeat service to restart before taking affect. It doesn't automatically reload the configuration. You can restart the winlogbeat service through services.msc (**win + r** > **services.msc** > **Enter**), from a Powershell terminal as admin (**restart-service winlogbeat**), or from an administrative command prompt (**sc stop winlogbeat && sc start winlogbeat**).

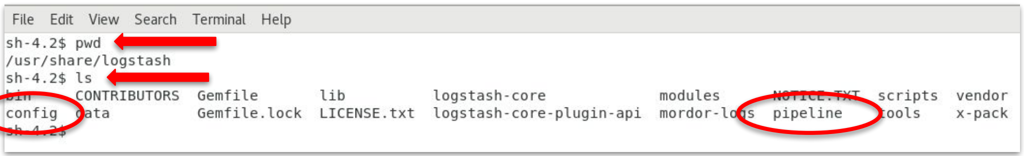
The Winlogbeat Pipeline - Logstash

①



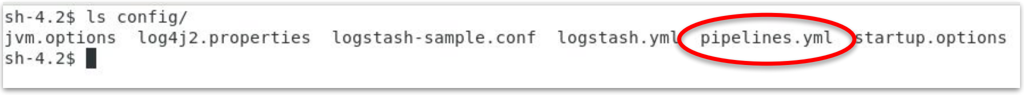
```
[analyst@elk ~]$ hostname # I'm currently on the CentOS host
elk.windomain.local
[analyst@elk ~]$ logstash # Now I'm connecting to the Logstash container
sh-4.2$ hostname # See, my hostname changed...
logstash
sh-4.2$
```

②



```
sh-4.2$ pwd
/usr/share/logstash
sh-4.2$ ls
bin     CONTRIBUTORS  Gemfile       lib          logstash-core             modules      NOTICE.TXT  scripts  vendor
config  data          Gemfile.lock  LICENSE.txt  logstash-core-plugin-api  mordor-logs  pipeline    tools    x-pack
sh-4.2$
```

③



```
sh-4.2$ ls config/
jvm.options  log4j2.properties  logstash-sample.conf  logstash.yml  pipelines.yml  startup.options
sh-4.2$
```

Next, we'll move over to the CentOS VM. Go ahead and open a terminal (there's a shortcut on the desktop).

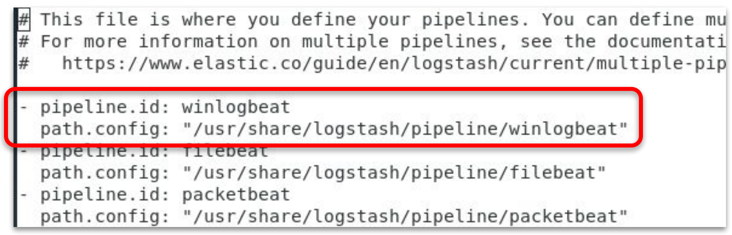
1) In the terminal, type **logstash** and press **Enter**. This command is aliased to **docker exec -it logstash /bin/sh**. This allows you to 'exec' into the Logstash container and get a shell. If you type **hostname** and press **Enter**, you'll see you're now on the host "logstash" and not on "elk.windomain.local," which is the hostname of your CentOS VM.

2) When you 'exec' into a Docker container, your current working directory is whatever was defined as the current working directory when the container was built. In this case, you're in /usr/share/logstash, which is the base path of the Logstash installation in the container. You can see this by running **pwd** (print working directory). If you run **ls** (list) the directory you'll see two directories we're going to be looking at: **config** and **pipeline**.

3) From the terminal, list the contents of the config directory: **ls config/**.

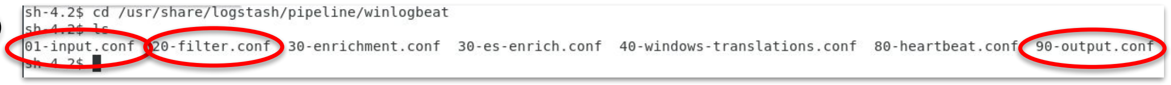
The Winlogbeat Pipeline - Logstash

①



```
# This file is where you define your pipelines. You can define mu
# For more information on multiple pipelines, see the documentati
#   https://www.elastic.co/guide/en/logstash/current/multiple-pip

- pipeline.id: winlogbeat
  path.config: "/usr/share/logstash/pipeline/winlogbeat"
- pipeline.id: filebeat
  path.config: "/usr/share/logstash/pipeline/filebeat"
- pipeline.id: packetbeat
  path.config: "/usr/share/logstash/pipeline/packetbeat"
```

②



```
sh-4.2$ cd /usr/share/logstash/pipeline/winlogbeat
sh-4.2$ ls
01-input.conf  20-filter.conf  30-enrichment.conf  30-es-enrich.conf  40-windows-translations.conf  80-heartbeat.conf  90-output.conf
sh-4.2$ 
```

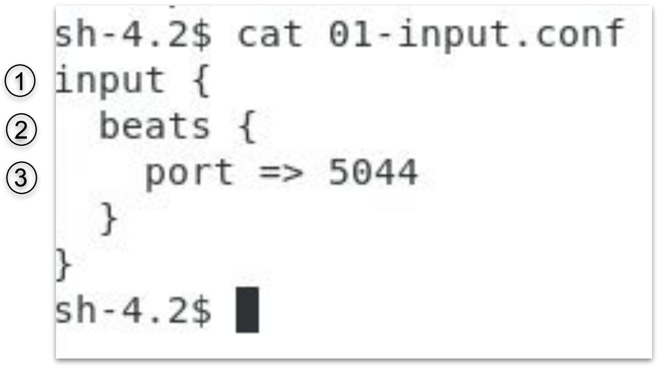
Let's look at how pipelines work:

1) Let's look at the contents of the **pipeline.yml** file. From the terminal run: **vim config/pipeline.yml**. You will see Logstash pipeline definitions which contain both a pipeline ID and the path to the configuration files used in the pipeline. As you can see, the path to the Winlogbeat configuration files is **/usr/share/logstash/pipeline/winlogbeat**.

2) Next, let's look in that directory and see what files are used for the Winlogbeat configuration in Logstash. From the terminal run **cd /usr/share/logstash/pipeline/winlogbeat** and then **ls**. You should see several ".conf" files, but we're most concerned with three right now: **01-input.conf**, **20-filter.conf**, and **90-output.conf**.

You'll notice these files are all prefixed with a number. Logstash loads all of the ".conf" files in this directory and every applicable log is processed by each of these files. Logstash uses the name of the file to determine the order in which it is used. Files are sorted alphabetically (and numerically) so we can use the numbered prefixes to indicate an order in which we want them to be applied to our messages.

The Winlogbeat Pipeline - Logstash Input



```
sh-4.2$ cat 01-input.conf
① input {
②   beats {
③     port => 5044
    }
  }
sh-4.2$
```

Logstash pipelines are made up of three stages: **input**, **filter**, and **output**. Let's first look at the input stage. From a terminal run **cat 01-input.conf**.

You should quickly notice the syntax resembles JSON. You can read more about the configuration file syntax here: https://www.elastic.co/guide/en/logstash/6.7/configuration-file-structure.html.

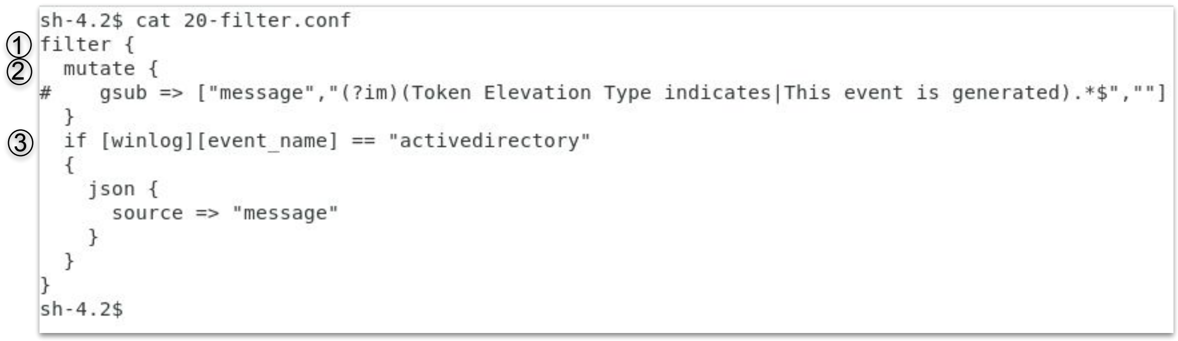
In the file, we can see a few things:

1) This is an input file. It defines what plugin(s) will be used to receive logs and what port they'll be listening on.

2) This input is using the **beats** plugin. This plugin is an official plugin that allows Logstash to receive traffic from Beats like Winlogbeat, Filebeat, and Packetbeat. This traffic can require authentication and TLS depending on your configuration.

3) This input is listening on port 5044. If you recall the output configuration in the **winlogbeat.yml** file, it was pointing to "logstash:5044." This is how we configuration Logstash to listen for and understand that traffic.

The Winlogbeat Pipeline - Logstash Filter



```
sh-4.2$ cat 20-filter.conf
① filter {
②   mutate {
#     gsub => ["message","(?im)(Token Elevation Type indicates|This event is generated).*$",""]
  }
③   if [winlog][event_name] == "activedirectory"
  {
    json {
      source => "message"
    }
  }
}
sh-4.2$
```

Next, let's look at a filter. There are several filters here, as everything between the input and output is a filter. For the sake of simplicity, let's look at **20-filter.conf**. Just like before, let's cat the file. Run **cat 20-filter.conf** from the terminal.

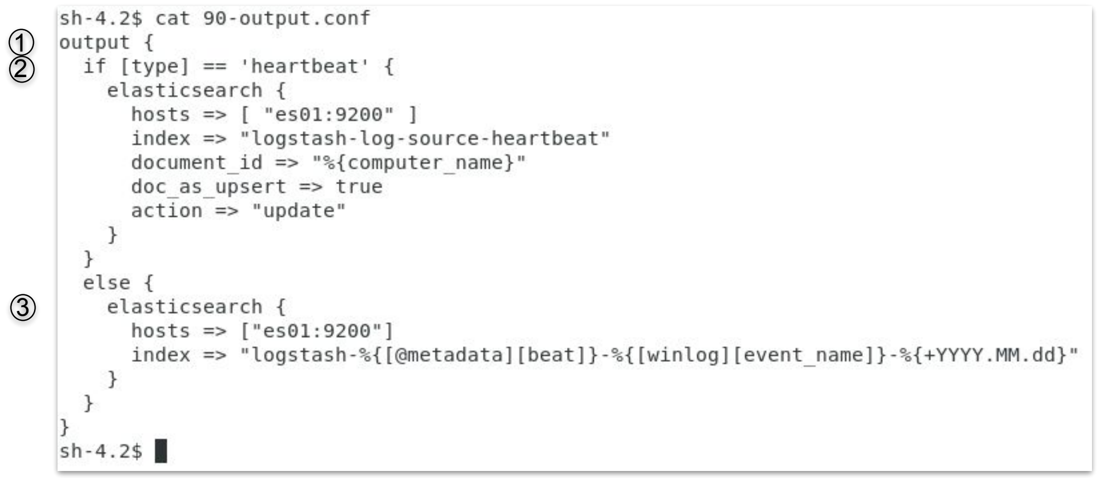
1) Looking at the file, we can see this is a filter.

2) There are multiple plugins that can be used by a filter. Here, we see the **mutate** filter plugin being used. This plugin contains an option called **gsub**. In this configuration gsub is commented out, so the mutate filter plugin essentially does nothing. If you'd like to read more about the mutate filter plugin and the gsub configuration option though, you can find more here: https://www.elastic.co/guide/en/logstash/6.7/plugins-filters-mutate.html

3) Last, a conditional is used. The conditional **if** is saying, if the object **winlog.event_name** is equal to the string **activedirectory**, the apply the following configuration. When dotted objects like "winlog.event_name" are used in Elasticsearch, they're often referenced in Logstash using square brackets like in the configuration above.

   If the condition is met, the **json** filter plugin is applied to the **message** field in the log that's received. The json filter plugin tells Logstash to automatically parse out key value pairs from JSON blobs that can be found in whichever field **source** is pointing to. This makes parsing JSON logs extremely simple!

The Winlogbeat Pipeline - Logstash Output



```
sh-4.2$ cat 90-output.conf
① output {
②   if [type] == 'heartbeat' {
      elasticsearch {
        hosts => [ "es01:9200" ]
        index => "logstash-log-source-heartbeat"
        document_id => "%{computer_name}"
        doc_as_upsert => true
        action => "update"
      }
    }
    else {
③     elasticsearch {
        hosts => ["es01:9200"]
        index => "logstash-%{[@metadata][beat]}-%{[winlog][event_name]}-%{+YYYY.MM.dd}"
      }
    }
  }
sh-4.2$ █
```

Finally, let's run **cat 90-output.conf** from the terminal.

1) The first line of this file indicates this is an output configuration.

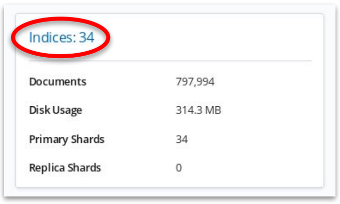
2) Just like the previous filter, we can see a conditional is being used. Ignore this for now as the first portion of the conditional is unrelated to what we're looking at.

3) After the 'else,' we can see the **elasticsearch** output plugin is being used. It's configured to output logs that come through this pipeline to the host **es01** on port **9200**. The host "es01" is the name of our Elasticsearch node. By default, Elasticsearch's REST API listens on port 9200.

   Additionally, this configuration defines the name of the index these logs will be written into. In this case, variables are being used to automatically assign the name, depending on the name of the Beat, the Windows Event Log source, and the date. Next we'll see how those variables are resolved and what the data looks like once it's in Elasticsearch.

Winlogbeat - Kibana

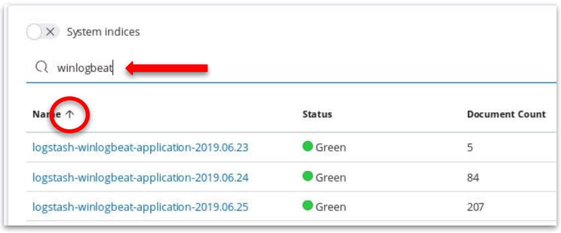
①



Indices: 34

Documents	797,994
Disk Usage	314.3 MB
Primary Shards	34
Replica Shards	0

②

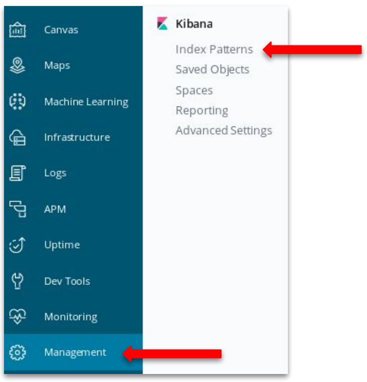


System indices

winlogbeat

Name ↑	Status	Document Count
logstash-winlogbeat-application-2019.06.23	● Green	5
logstash-winlogbeat-application-2019.06.24	● Green	84
logstash-winlogbeat-application-2019.06.25	● Green	207

③



In your CentOS VM first exit out of the Logstash container by simultaneously pressing **ctrl + d**.

Navigate to Kibana in a browser using the shortcut on the desktop. Alternatively, you can also navigate to http://<centos_vm_ip>:5601. You can get the IP using the shortcut on the desktop, or by running **ip a | grep ens33 | grep inet | awk '{print $2}'** and ignoring the "/24" at the end of the result. Click on **Monitoring**.

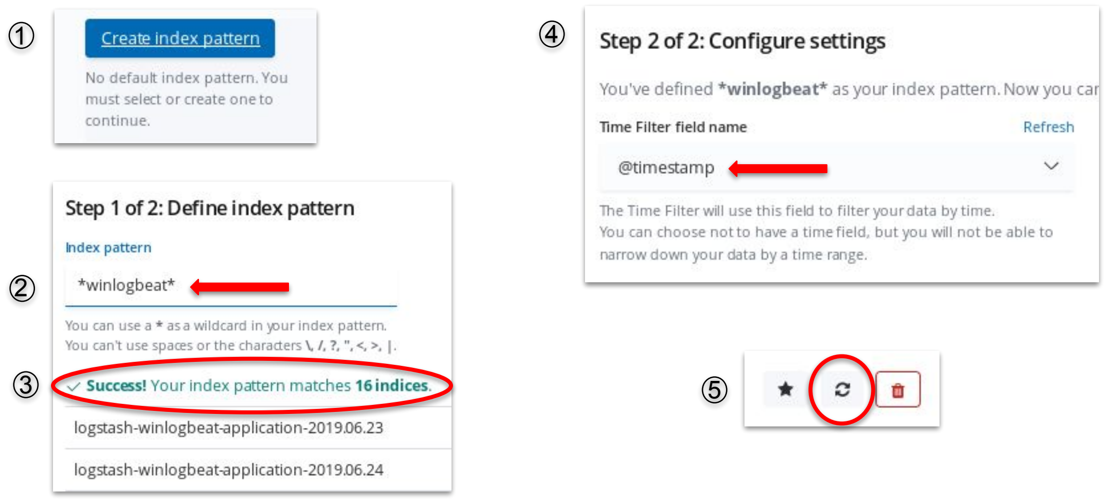
1) Click the **Indices** link to view all the indices stored in your Elasticsearch cluster.

2) In the search bar, type **winlogbeat**. You'll see Kibana performs real-time search as you type. You should quickly see results matching your query. You may also notice an arrow. This indicates how the results are sorted. In the example above, they're assorted in ascending order by **Name**.

   The variables we saw in the Logstash output file are now resolved. The variable %**{[@metadata][beat]}** is now resolved to **winlogbeat**, %**{[winlog][event_name]}** is resolved to **application** in the top result, and you can see the date has been appended to the end of the index name.

3) Click on **Management** and then **Index Patterns**.

Winlogbeat - Kibana



Create index pattern

No default index pattern. You must select or create one to continue.

Step 1 of 2: Define index pattern

Index pattern

winlogbeat

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

✓ Success! Your index pattern matches 16 indices.

logstash-winlogbeat-application-2019.06.23

logstash-winlogbeat-application-2019.06.24

Step 2 of 2: Configure settings

You've defined *winlogbeat* as your index pattern. Now you car

Time Filter field name — Refresh

@timestamp

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

You'll need to create an index pattern to search Winlogbeat logs in Kibana. Index patterns are easy to create:

1) Click **Create index pattern**.

2) Create your index pattern. The index pattern you create can either match an index exactly, or you can use a wildcard (*) to match only a substring within the index name. In the example above, we are using ***winlogbeat*** to match all Winlogbeat logs. Instead, we could use **logstash-winlogbeat-*** or we could use ***winlogbeat-application-*** if we only wanted to match on Windows Application logs.

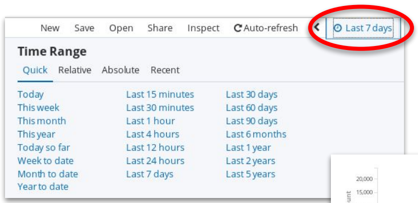
3) If you're satisfied with your pattern and you see the **Success! Your index pattern matches…** message, click the **Next step** button.

4) Select the Time Filter field name, which in this case will be **@timestamp** and click the **Create index pattern** button.

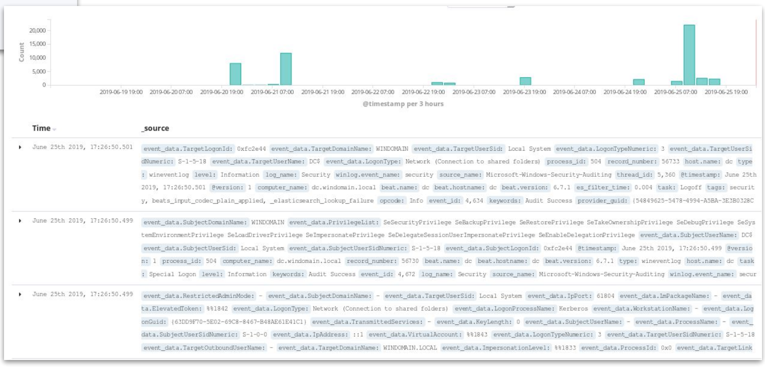
5) Click the refresh button at the top right of the browser window.

Viewing and Searching Winlogbeat Logs

①



②



```
Time      _source
▸ June 25th 2019, 17:26:50.501   event_data.TargetLogonId: 0xfc2e44 event_data.TargetDomainName: WINDOMAIN event_data.TargetUserSid: Local System event_data.LogonTypeNumeric: 3 event_data.TargetUserSi
dNumeric: S-1-5-18 event_data.TargetUserName: DC$ event_data.LogonType: Network (Connection to shared folders) process_id: 504 record_number: 56733 host.name: dc type
: wineventlog level: Information log_name: Security winlog.event_name: security source_name: Microsoft-Windows-Security-Auditing thread_id: 5,360 @timestamp: June 25th
2019, 17:26:50.501 @version: 1 computer_name: dc.windomain.local beat.name: dc beat.hostname: dc beat.version: 6.7.1 es_filter_time: 0.004 task: Logoff tags: securit
y, beats_input_codec_plain_applied, _elasticsearch_lookup_failure opcode: Info event_id: 4,634 keywords: Audit Success provider_guid: {54849625-5478-4994-A5BA-3E3B0328C

▸ June 25th 2019, 17:26:50.499   event_data.SubjectDomainName: WINDOMAIN event_data.PrivilegeList: SeSecurityPrivilege SeBackupPrivilege SeRestorePrivilege SeTakeOwnershipPrivilege SeDebugPrivilege SeSys
temEnvironmentPrivilege SeLoadDriverPrivilege SeImpersonatePrivilege SeDelegateSessionUserImpersonatePrivilege SeEnableDelegationPrivilege event_data.SubjectUserName: DC$
event_data.SubjectUserSid: Local System event_data.SubjectUserSidNumeric: S-1-5-18 event_data.SubjectLogonId: 0xfc2e44 @timestamp: June 25th 2019, 17:26:50.499 @versio
n: 1 process_id: 504 computer_name: dc.windomain.local record_number: 56730 beat.name: dc beat.hostname: dc beat.version: 6.7.1 type: wineventlog host.name: dc task
: Special Logon level: Information keywords: Audit Success event_id: 4,672 log_name: Security source_name: Microsoft-Windows-Security-Auditing winlog.event_name: secur

▸ June 25th 2019, 17:26:50.499   event_data.RestrictedAdminMode: - event_data.SubjectDomainName: - event_data.TargetUserSid: Local System event_data.IpPort: 61804 event_data.LmPackageName: - event_da
ta.ElevatedToken: %%1842 event_data.LogonType: Network (Connection to shared folders) event_data.LogonProcessName: Kerberos event_data.WorkstationName: - event_data.Log
onGuid: {63DD9F70-5E02-69C8-8467-B48AE61E41C1} event_data.TransmittedServices: - event_data.KeyLength: 0 event_data.SubjectUserName: - event_data.ProcessName: - event_
data.SubjectUserSidNumeric: S-1-0-0 event_data.IpAddress: ::1 event_data.VirtualAccount: %%1843 event_data.LogonTypeNumeric: 3 event_data.TargetUserSidNumeric: S-1-5-18
event_data.TargetOutboundUserName: - event_data.TargetDomainName: WINDOMAIN.LOCAL event_data.ImpersonationLevel: %%1833 event_data.ProcessId: 0x0 event_data.TargetLink
```

Now we'll look at the log messages that are coming in to ELK from Winlogbeat. Click the **Discover** link at the top-left of the browser window.

1) In the top-right corner, you should see a time filter button. It may say **Last 15 minutes**. If it does, expand the search range to something much larger. The example above shows results for the last seven days.

2) You should now see the first 500 results returned to Kibana by Elasticsearch.

Feel free to click around and explore Kibana if you've finished.