IAAS-018 – Elastic Logging and Aggregation Cluster (ELAC) 8.6 Upgrade Instructions

July 17, 2023
OA DCGS

Deleted: June
Deleted: 19

AFLCMC/HBGEB
AF DCGS Engineering
235 Byron Street, Suite 19A Robins AFB, GA 31098

Controlled by: AFLCMC/HBGEB
Controlled by: AFRL/RIEB
CUI Category: Defense, CTI
Distribution/Dissemination Controls: Dist. E
POC: AFRL.RIE.OADCGSCM@us.af.mil

THIS PAGE INTENTIONALLY LEFT BLANK

CHANGE LOG

This record shall be maintained throughout the life of the document. Each published update shall be recorded. Revisions are a complete re-issue of the entire document. A revision shall be made annually or when applicable. Refer questions concerning this document to: robert.heddleson@us.af.mil

CHANGE / REVISION RECORD

Revision	Date	Description of Change	Made By
1.0	05/08/23	Initial Release CR-YEAR-OADCGS-XXX	S. Truxal/AFLCMC/ES
1.1	06/01/23	Updates for Installation	S. Truxal/AFLCMC/ES
1.2	06/19/23	Updates for template configs	S. Truxal/AFLCMC/ES

TABLE OF CONTENTS

List of Figures

List of Tables

Table 1 Cluster Hardware Requirements at Hubs (15 Nodes) 4
Table 2 Cluster Hardware Requirements at Hubs (10 Nodes) 4
Table 3 Cluster Hardware Requirements at Hubs (3 Nodes) 5
Table 4 Logstash Hardware Requirements at Sites in Production and Test Enviornments 5
Table 5 Logstash Hardware Requirements for REL 5
Table 6 Puppet Modules Required 11
Table 7 Supported Devices and Access Types 33
Table 8 Access test command per device 34
Table 9 Elastic nodes to install Kibana on 40
Table 10 Beat Installation Location 76
Table 11 Types for Ingested HBSS Metrics 95
Table 12 Upgrade Order 121
Table 13 Elastic nodes to upgrade Kibana on 126

THIS PAGE INTENTIONALLY LEFT BLANK

1 Introduction

This document is intended to be used to upgrade Enterprise Elasticsearch from version 7.17.6 to version 8.6.2. If Enterprise Elastic is not currently installed in the environment, please refer to *ES-018 – Elastic Logging and Aggregation Cluster (ELAC) – System Installation Instructions* to install the initial version before executing these procedures. If Elasticsearch is installed but running a version older than 7.17.6, please consult with an Elastic SME before executing this upgrade.

1.1 Location of Installation

The installation will be conducted in NOFORN.

1.2 Overview

This upgrade includes the following: (For complete details see the CHANGELOG included with each component)

- The version of Elasticsearch and its components will be upgraded.
  - Elastic - Release notes | Elasticsearch Guide [8.6] | Elastic
  - Kibana - Release notes | Kibana Guide [8.6] | Elastic
  - Logstash – Release notes | Logstash Guide [8.6] | Elastic
  - Beats – Release notes | Beats Guide [8.6] | Elastic
- New non-timeseries dcgs-current-healthdata-iaas-ent index holding state of all hosts, applications, and groups on system.
- Ability to detect when hosts, applications or group status has not been updated for a period of time making the data “stale”
- Addition of new symptom, “No_Audit” when hosts are not sending audit data (windows events/Linux syslog).
- Elastic Data Collector updates
  - Fixes issues detected in previous versions.
  - Improved monitoring and logging.
  - Symptoms for DellIdrac and FX2 issues
  - CiscoSwitch class update to ensure query of entity list is successful before allowing other queries.
  - FX2 class updated logic to look for the drsServerModel when removing empty values
  - New API to query data collector status on port 9601.
  - Versioning of data collector added – Available via new API
  - Fix to vSphere class to handle reconnection on restart of Vcenter
  - Addition of timeout on spawned SNMP workers to ensure subprocess are terminated properly
- Elimination of monitoring system processes on both Linux and Windows hosts by Metricbeat casing a major reduction in volume of Metricbeat data
- Addition of Single Worker Pipeline to handle processing of logs when order of ingest must be maintained.
- Addition of new ingest pipelines and datatypes:
  - ACAS data
  - Serena data – Was also available in 7.17
- Addition of default fields for search in all indexes

- Addition of ingest pipelines for Winlogbeat data. Javascript processors that did processing were removed in version 8 and replace with ingest pipelines.
- Updates to install_beats_windows SCCM installation script.
    - Processing added to remove any old scheduled tasks that were left on windows boxes from older installs.
    - Ability to configure Filebeat based on presence of a service.
    - Addition of automated configuration for Single worker pipeline when needed.
- Additional processing in Winlogbeat pipeline to parse impersonation data.
- Updates to allow data to be processed correctly by the auditd and iptables ingest pipelines
- Addition of existing ART integrations
    - Render
    - Guardian
    - SOAESB
    - Socet
    - MAAS
    - Xplorer

2 System Environment

The environment required to run Elastic will be described in this section.

2.1.1 Elastic Search Cluster

Optional Cluster Sizes – This install document is intended to be used for different Elastic Cluster sizes used in different environments For the VM requirements, review the tables in this section specific to the cluster size being installed for your environment.

As Elastic clusters grow, nodes become specialized. The following list describes the roles nodes may have. These abbreviations are used to define node roles in the Cluster configurations that follow. See https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html for more information.

- **Master-eligible node (m):** A node that has **node.master** set to **true** (default), which makes it eligible to be elected as the **master** node, which controls the cluster.
- **Data Content**(s): System indices and other indices that aren't part of a data stream are automatically allocated to the content tier.
- **Hot Data node (h):** Hot data nodes are part of the hot tier. The hot tier is the Elasticsearch entry point for time series data and holds your most-recent, most-frequently-searched time series data. Nodes in the hot tier need to be fast for both reads and writes, which requires more hardware resources and faster storage (SSDs). For resiliency, indices in the hot tier should be configured to use one or more replicas.
- **Warm Data node (w):** Warm data nodes are part of the warm tier. Time series data can move to the warm tier once it is being queried less frequently than the recently-indexed data in the hot tier. The warm tier typically holds data from recent weeks. Updates are still allowed, but likely infrequent. Nodes in the warm tier generally don't need to be as fast as those in the hot tier. For resiliency, indices in the warm tier should be configured to use one or more replicas.
- **Ingest node (i):** A node that has **node.ingest** set to **true** (default). Ingest nodes are able to apply an ingest pipeline to a document in order to transform and enrich the document before indexing.
- **Machine learning node (l):** A node that has **xpack.ml.enabled** and **node.ml** set to **true**, which is the default behavior in the Elasticsearch default distribution. If you want to use machine learning features, there must be at least one machine learning node in your cluster. For more information about machine learning features, see Machine learning in the Elastic Stack.

2.1.1.1 15 Node Cluster - Large (Production High & Low)

The clusters for Elastic will reside at the hubs. Each cluster will initially be made up of 15 nodes with the minimum configuration described in the following table. Kibana will be installed on elastic-nodes 10 and 15 in this configuration.

NOTE: The Elastic license will change over time; there are currently 15 nodes in the Elastic cluster but there can be more or fewer. The size of the cluster may change based on the amount of data ingested and stored.

Table 1 Cluster Hardware Requirements at Hubs (15 Nodes)

VM Description	OS	VCPUs	RAM(GB)	Disk 0	Disk 1	Data Dir	Role
elastic-node-1	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	m
elastic-node-2	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	m
elastic-node-3	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	m
elastic-node-4	Linux	8	64	*OS 90 GB*	Data - 100GB(SSD)	/ELK-local	ls
elastic-node-5	Linux	8	64	*OS 90 GB*	Data - 100GB(SSD)	/ELK-local	ls
elastic-node-6	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	his
elastic-node-7	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	his
elastic-node-8	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	his
elastic-node-9	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	his
elastic-node-10	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-11	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-12	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-13	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-14	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-15	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw

2.1.1.2 10 Node Cluster – Medium (CTE & MTE)

The clusters for Elastic will reside at the hubs. Each cluster will initially be made up of 10 nodes with the minimum configuration described in the following table. Kibana will be installed on elastic-nodes 7 and 10 in this configuration.

NOTE: The Elastic license will change over time; there are currently 10 nodes in the Elastic cluster but there can be more or fewer. The size of the cluster may change based on the amount of data ingested and stored.

Table 2 Cluster Hardware Requirements at Hubs (10 Nodes)

VM Description	OS	VCPUs	RAM(GB)	Disk 0	Disk 1	Data Dir	Role
elastic-node-1	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	m
elastic-node-2	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	m
elastic-node-3	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	m
elastic-node-4	Linux	8	64	*OS 90 GB*	Data - 100GB(SSD)	/ELK-local	ls
elastic-node-5	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	his
elastic-node-6	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	his
elastic-node-7	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-8	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-9	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-10	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw

2.1.1.3 6 Node Cluster - Small (REL)

The clusters for Elastic will reside at the hubs. Each cluster will initially be made up of 6 nodes with the minimum configuration described in the following table. Kibana will be installed on elastic-nodes 5 and 6 in this configuration.

NOTE: The Elastic license will change over time; there are currently 6 nodes in the Elastic cluster but there can be more or fewer. The size of the cluster may change based on the amount of data ingested and stored.

Table 3 Cluster Hardware Requirements at Hubs (3 Nodes)

VM Description	OS	VCPUs	RAM(GB)	Disk 0	Disk 1	Data Dir	Role
elastic-node-1	Linux	8	64	*OS 90 GB*	Data - 1TB(SSD)	/ELK-local	mlhis
elastic-node-2	Linux	8	64	*OS 90 GB*	Data - 1TB(SSD)	/ELK-local	mhis
elastic-node-3	Linux	8	64	*OS 90 GB*	Data - 1TB(SSD)	/ELK-local	mhis
elastic-node-4	Linux	4	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-5	Linux	4	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw
elastic-node-6	Linux	4	64	*OS 90 GB*	None(nfs)	/ELK-nfs	iw

2.1.2 Logstash Nodes

Each site will have one Logstash instance with the minimum configuration described in the following table.

2.1.2.1 Production (High & Low), CTE and MTE

Table 4 Logstash Hardware Requirements at Sites in Production and Test Enviornments

VM Description	OS	VCPUs	RAM(GB)	Disk 0	Disk 1	Data Dir
Logstash	Linux	8	32	*OS 90 GB*	500GB	ELK-local

2.1.2.2 REL

Table 5 Logstash Hardware Requirements for REL

VM Description	OS	VCPUs	RAM(GB)	Disk 0	Disk 1	Data Dir
Logstash	Linux	4	32	*OS 90 GB*	500GB	ELK-local

2.2 Minimum Software Requirements

Each Elastic and Logstash VM should be installed with RedHat Linux version 7.3 or greater. After the installation of the Linux operating system, each node should be joined to Puppet to become a Puppet client, at which time Puppet will manage the configuration of each system.

Service Account Kerberos Management (SAKM) must then be installed on each VM and configured to sustain the Kerberos ticket for the Elastic service account for the site where the VM resides. Example (00_elastic.svc).

2.3 Site Requirements

Prior to installation, a Logstash VM must be provisioned at each site data is to be ingested from; see Section 2.1.2 for hardware requirements. The installation of all the Logstash software and all collection components at the site can be done remotely so no actual site presence is necessary.

3 Security Considerations

Several types of administrators and privileges will be needed during this installation. Ensure that the installer has proper access rights to execute these installation procedures. Scan load Scan as directed by normal operations.

4 Prerequisites

This document provides instructions to do an initial installation of Enterprise Elasticsearch at version 8.6.2or upgrade Elasticsearch version 7.17.6 to version 8.6.2.

4.1 Additional Documents Required for Installation

- *ES-018 – SCCM – Instructions for Building an SCCM Package to Install Beats for Windows*.

4.2 Roles Required

This document assumes the user is a Linux administrator and that the OA System is up and available. Subject Matter Experts/Administrators in the following areas will also be necessary for portions of the installation:

- Puppet
- System Center Configuration Manager (SCCM)
- Domain Administrator
- xx_elastic.svc account password for each site may be used for:
  - XtremIO storage device
  - Isilon storage device
- SNMP v3 credentials for each of the following:
  - Cisco Nexus7k switch
  - Cisco Nexus5k switch
  - Cisco 3850 switch
  - Data Domain storage device
  - FX2 Chassis
  - FC630/640 Blade Servers
  - R630/640 Servers

The installer has the "Elastic Administrator" OneIM Role, which puts them in the **ent Elastic Admins** Active Directory (AD) Group.

4.3 Installation Artifacts

The following artifacts are required to perform the installation.

Verify a folder named **elastic** exists on the fileserver under the admin\ess directory called **elastic**. This folder will contain a local copy of the artifacts needed during the installation process. If the folder does not exist, it can be populated by obtaining the artifacts from DCGS Configuration Management.

- 01.zip file
  1. Elastic_Core_Components-8.6.2.tar.gz
  2. Elastic_Linux_Beats-8.6.2.tar.gz

- 02.zip file
  1. oadcgs-es-elastic-sccm-2.0.40.1.zip

2. Elastic_Window_Beats-8.6.2.zip

- 03.zip file
  1. oadcgs-es-elastic-reposerver- 2.0.40.1.tar.gz
  2. oadcgs-dsil_elastic_clients-2.0.3.1.tar.gz
  3. oadcgs-dsil_elastic_servers-1.2.13.1.tar.gz

There should be an Elastic repo on the OA DCGS repo server where the RPMs will be loaded for the installation. If this is an initial install the repo will be created in section 5.4. The RPMs will be placed on the repo server in stages to prevent components from being installed/upgraded out of order. The RPMs will be extracted from the previously referenced archives and placed in the **elastic** directory on the fileserver for use when needed.

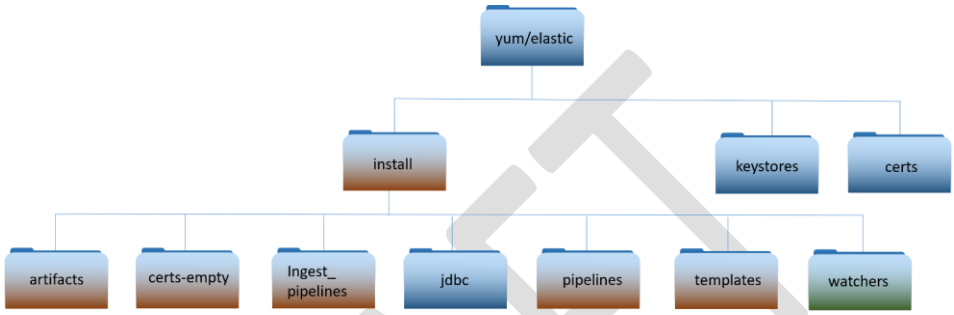
The following RPMs should extracted from the delivered artifacts and be available on the fileserver **but not added to the repo server until instructed** during the installation process.

1. Elastic_Core_Components-8.6.2.tar.gz
   - elasticsearch-8.6.2-x86_64.rpm
   - kibana-8.6.2-x86_64.rpm
   - logstash-8.6.2-x86_64.rpm
2. Elastic_Linux_Beats-8.6.2.tar.gz
   - metricbeat-8.6.2-x86_64.rpm
   - filebeat-8.6.2-x86_64.rpm
   - heartbeat-8.6.2-x86_64.rpm

NOTE: When installing Elasticsearch, Logstash and Kibana, only those 3 RPMs are placed on the repo server to install/upgrade the core components prior to adding any Beat RPMs. Beat collectors are automatically installed/upgraded so this should not happen until the Elastic cluster is upgraded, running, and ready to receive data. Be sure to only place the RPMs in the Elastic repo when directed during the installation process.

The following diagram shows the directory layout of the Elastic repository on the repo server. These directories will be populated when extracting the **oadcgs-es-elastic-reposerver- x.x.x.x.tar.gz** archive.

NOTE for Upgrades: The green highlighting indicates a new directory added after the 7.17.6 upgrade. The orange highlighting indicates contents in the folder have changed for this upgrade.



Important: The "certs" and "keystores" directories are not part of the delivery. These directories hold system specific information.

NOTE: This document only highlights the scripts/files needed to perform this upgrade. For details on existing files, refer to previous installation/upgrade documents.

install – installation scripts used for upgrade.

- activate_acas.sh – Used to add ACAS ingest datatype
- activate_serena.sh – Used to add Serena ingest datatype
- bootstrap_indexes.sh – Ensures aliases for indexes are configured
- insertIntoPipeline.sh – Used by ART integration scripts to update Filebeat pipeline
- installElasticDataCollector.sh – Installs elastic data collector on a Logstash instance
- load_auditsettings.sh – Loads/Updates audit settings for Elastic
- load_objects.sh – Loads all objects that come with the release
- load_ILM_Policy.sh – Loads Index Lifecycle Management Policies
- load_SLM_Policy.sh – Loads Snapshot Lifecycle Management Polices
- load_logstash_pipelines.sh – Updates Logstash pipelines for new version into Kibana
- load_templates.sh – updates templates into Elastic
- load_roles.sh – Loads roles into Kibana
- upgrade_logstash.sh – python script used to upgrade Logstash instance
- reindex_renamed_indices.sh – reindexing script
- load_ingest_pipelines.sh – Loads all ingest pipelines into Elastic
- update_kibana_settings.sh – Updates kibana advanced settings with security banner

- upgrade_logstash.sh – Script used to upgrade logstash instance
- upgrade_node.sh – Script used to upgrade an Elasticsearch node
- upgrade.py – python script used by upgrade_nodes.sh

4.4 Puppet Modules Required

Table 6 Puppet Modules Required

Module:	dsil_elastic_clients
Description:	The **dsil_elastic_clients** module is used to automatically install Metricbeat and Filebeat on Linux hosts. Metricbeat is installed on all Linux hosts in the OA DCGS system. Filebeat is installed on all Elasticsearch and Logstash hosts to collect log files from the Elastic applications. This module also supplies a generic utility function **setup_beat** that can be used by any other Puppet modules to request that Filebeat be installed on hosts controlled by that module. OA DCGS ARTs and other Puppet clients that wish to have data gathered using Filebeat will use this function to install Filebeat on their hosts. The basic use of this module is described here; knowledge of Puppet is expected to understand the implementation. Please contact an Enterprise Services Elastic SME for questions or issues using the function.
Version:	2.0.3.1
Parameters:	install (Boolean) – If true beat components will be installed/upgraded. If false beat components will be removed. **default**: true restart_beats (Boolean) – If true restart beats on every puppet run. If false only restart beats on configuration change. **default**: false
Resources:	**Files** /etc/metricbeat/metricbeat.yml /etc/metricbeat/modules/logstash-xpack.yml /etc/metricbeat/modules/elasticsearch-xpack.yml /etc/metricbeat/modules/kibana-xpack.yml /etc/metricbeat/modules/system.yml

	/etc/metricbeat/appmonitor_linux.js /etc/filebeat/filebeat.yml /etc/filebeat/modules/elasticsearch.yml etc/filebeat/modules/logstash.yml **Packages** Metricbeat, Filebeat

Module:	dsil_elastic_servers
Description:	The **dsil_elastic_servers** module is used to configure Elastic, Kibana, and Logstash servers. The module opens the necessary ports, creates mounts, and performs other configuration tasks necessary to allow each component to run properly. Elasticsearch, Kibana, and Logstash are all upgraded following the procedures in this document, but they will not be able to run successfully unless the hosts are configured with this module. The Heartbeat component of Elastic is also automatically upgraded by this Puppet module. This module then ensures that Heartbeat is running on each Logstash host.
Version:	1.2.12.1
Parameters:	None
Resources:	**Files** /etc/heartbeat/monitors.d/ess.http.hub.yml /etc/heartbeat/monitors.d/ess.http.site.yml /etc/heartbeat/monitors.d/ess.icmp.yml /etc/heartbeat/monitors.d/ess.tcp.hub.yml /etc/heartbeat/monitors.d/ess.tcp.site.yml /etc/heartbeat/heartbeat.yml **Packages** Heartbeat

5 Installation Instructions

The instructions in this section are used to install Enterprise Elastic for the first time as version 8.6.2 or to upgrade from the 7.17.6 version of Elastic to 8.6.2. If you are running a version of Elastic prior to version 7.17.6 please refer to one of the following:

- *ES-018 – Elastic Logging and Aggregation Cluster (ELAC) – System Installation Instructions*
- *ES-018 - Elastic Logging and Aggregation Cluster (ELAC) - 7.16.3 Upgrade Instructions.docx*
- *ES-018 - Elastic Logging and Aggregation Cluster (ELAC) - 7.17.6 Upgrade Instructions.docx*

5.1 Estimated Implementation Time

The time to upgrade depends on multiple factors; the following estimates are given.

- Cluster Install: ~ 2 days
- Cluster upgrade: ~ 6 hours
- Site upgrade: ~ 1 hour

5.2 Cleanup of Existing Versions and Files

N/A

5.3 Media Boot Procedures

N/A

5.4 Software Installation Instructions

This section is only to be used for initial installs of Elasticsearch into an environment. **See Section 5.5 for upgrades.**

5.4.1 Pre-Installation Instructions

The following items must be completed before proceeding with the installation of **Enterprise Elastic**.

5.4.1.1 Create Elastic Repository

NOTE: A Linux administrator will be needed to execute this section.

In this section you will create a yum repository for Elasticsearch on the repo server. If one already exists, then you should still validate that it is configured correctly in puppet to be distributed to all Linux hosts.

1. Login to repo server "XXXsu01ro01 and create the directory for the elastic repo

   # sudo su

   # cd /var/www/html/yum

   # mkdir elastic

2. Create directories to hold keystores and certificates for environment.

```
# cd elastic

# mkdir keystores

# mkdir certs
```

Note: The above steps only create the location for the elastic repository, it will be populated during the installation process.

5.4.1.2 Update Repo Server

NOTE: A Linux administrator will be needed to execute this section.

The installation scripts and artifacts are delivered and placed in the "install" directory of the elastic repo on the repo server at each site. This is done so the scripts and artifacts are accessible for use during the installation at each site.

5.4.1.3 Create install directory with 8.6 install package.

Follow the steps in this section to create the "install" folder in the elastic repository with the scripts and artifacts needed for the 8.6 install.

1. Login to repo server, sudo to root and change directory to the elastic repo

   ```
   # sudo su
   # cd /var/www/html/yum/elastic (This is an example; the actual repo name may be different)
   ```

2. Copy oadcgs-es-elastic-reposerver- X.X.X.X.tar.gz to repo server

3. Uncompress the new install directory

   ```
   # tar -zxf oadcgs-es-elastic-reposerver- X.X.X.X.tar.gz --strip-components=1
   ```

4. Correct permissions

   You should be in the /var/www/html/yum/elastic directory before executing the following.

   ```
   # chown -R apache:apache install
   # chmod -R ugo+rx install
   # restorecon -R *
   # ls -ltrZ
   ```

The new install directory and its contents should now be ready for use.

5.4.1.4 Accounts and Passwords

- **xx_elastic.svc** service accounts exist for each site and have been given the correct privileges.

  Service account should be a member of the following groups:
  - fs – xx app-elac full control (xx = site number)
  - domain users
  - ent infrastructure read only
  - dcgs service accounts
- You have the password for the **xx_elastic.svc** account.
- You have the password for the Elastic bootstrap account **elastic.**
- **The installer is a member of the "ent elastic admins" group in Active Directory**

5.4.1.5 Storage

5.4.1.5.1 Local

- If local storage is being used for any boxes, the 2nd drive must be partitioned and mounted as /ELK-local on the respective VMs. See tables in section 2.1.1 to determine which nodes will be using local storage.
- The 2nd drive should be configured and the /etc/fstab file should be updated so it is mounted to /ELK-local
- GPT partitioning is necessary for drives larger than 2TB, suggest using parted for disk partitioning
- Logical Volume Manager should be used
  - pvcreate – Initialize physical volume
  - vgcreate - Create a volume group
  - lvcreate – Create a logical volume

5.4.1.5.2 NFS

- The /ELK-nfs directory on all Elastic nodes should be an NFS mount to the **elac** share on the Isilon. This mount point is set up by the **dsil_elastic_servers** puppet module (section 5.8.2).
- The **xx_elastic.svc** service account should have read/write access to this share.
- A storage admin will be required to configure the elac share on the Isilon if it does not already exist
  - Only Elastic and Logstash nodes should have access to the share
  - Share size – 150TB or larger

5.4.1.6 DNS Aliases

- DNS Aliases are set:
  - logstash (There should be a Logstash alias for each sites logstash host)
  - elastic-node-1
  - elastic-node-2
  - elastic-node-3
  - elastic-node-x

- kibana (NSX Load Balancer should be used for this IP)

Note: the kibana alias should be made in the base domain (ex: dcgs.mil) not the ECH(or Cluster install location) site domain to allow access to https://kibana from all sites.

Take note of these aliases as you will need them throughout this document.

5.4.1.7 Obtain PKI Certificates

Before proceeding with the installation of Elasticsearch or of any of its components, PKI Certificates must be obtained for the Elastic and Logstash Servers. Once obtained, the certificates must be placed in the **certs** directory on the repo server so they are available during the installation process.

5.4.1.7.1 Elastic Certificates (includes Kibana)

Certificates are needed for each Elasticsearch node. Elastic Certificates contain the following:

CN: hostname of Elastic VM (ex: u00su01el01.ech.dcgs.mil)

Aliases:

- fully qualified hostname (ex: u00su01el01.ech.dcgs.mil)
- hostname (ex: u00su01el01)
- hostname.{first segment of domain} (ex: u00su01el01.ech)
- elastic-node-{x} (ex: elastic-node-1)
- elastic-node-{x}.{first segment of domain} (ex: elastic-node-1.ech)

Additional Aliases if Kibana runs on the VM:

- kibana
- kibana.{first segment of domain} (ex: kibana.ech)
- kibana.{fully qualified} (ex: kibana.ech.dcgs.mil)

A convenience script is provided to make the creation of the Elastic Server Certificate requests easy for the installer. To create PKI certificate requests for Elastic to run on the system:

1. Log in to any existing Linux server at the site where the Elastic Cluster will be installed and do the following from your home directory:

```
# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/make_elastic_csrs.sh | bash
```

2. When the script completes, 3 directories will be present in the location where it was run.
   - Reqs: This directory holds the CSR Request information in text format.
   - Keys: The private key associated with the certificate request for each Elastic node.
   - CSRs – The actual PKI Certificate Request for each Elastic node.

3. The *.csr files should be submitted to the certificate authority for the system the Elastic Cluster is being installed on to obtain public certificates for each node. For systems submitted to the JWICS

certificate authority (i.e. CTE High or Enterprise High), also include a text file named SANS.txt listing the SubjectAlternativeNames listed in the CSR (for each CSR).

NOTE: In this installation, Kibana will use the certificate for the elastic node where it runs.

4. Once the Elasticsearch node certificates have been obtained, both the new certs and the keys for each node must be copied to the **certs** directory of the Elastic repo on the repo server ({xxx}u01ro01).

   PKI Certificates must be in the following format for the installation scripts to work properly:

Public Cert: {hostname}.crt	examples: u00su01el01.crt, u00su01el02.crt
Private Keys: {hostname}.key	examples: u00su01el01.key, u00su01el02.key

5. Create an elastic_cachain.pem file in the certs directory with the cachain from the certificate authority for the system being installed.

   NOTE: There is also an empty cachain.pem file delivered with the oadcgs-es-elastic-sccm package that must be updated to hold the correct root and sub-ca certificates during installation.

6. After placing all files in the certs directory you must ensure they have the correct owner/group:

   # chown -R apache:apache certs

7. The **certs** directory and all the certificate files must also have selinux context **httpd_sys_content_t** set. If you copy the certificates into the directory, they will automatically get this context set. If you moved them, they won't. Ensure all files have the correct context set by executing:

   # ls -ldZ to list the directory and ls -lZ to list its contents

```
[root@u00su01ro01 elastic]#
[root@u00su01ro01 elastic]# ls -ldZ certs
drwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 certs
[root@u00su01ro01 elastic]#
[root@u00su01ro01 elastic]#
[root@u00su01ro01 elastic]# ls -lZ certs
-rwxr-xr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elastic_cachain.pem
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el01.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el01.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el02.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el02.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el03.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el03.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el04.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el04.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el05.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el05.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el06.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el06.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el07.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el07.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el08.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el08.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el09.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el09.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el10.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01el10.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01ls01.crt
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01ls01.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 u00su01ls01_pkcs8.key
[root@u00su01ro01 elastic]#
tterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net
```

Figure 1- ls -lZ on certs directory

8. If all files do not have **httpd_sys_context_t** set, execute the following from the elastic repo directory:

   # restorecon *

5.4.1.7.2 Logstash Certificates

A Logstash instance runs at each DCGS site to collect data for that site. A PKI certificate is needed for each Logstash Instance. Do this to obtain a Logstash certificate request for each site.

Logstash Certificates contain the following:

CN: hostname of Logstash VM (ex: u00su01ls01.ech.dcgs.mil)

Aliases:

- fully qualified hostname (ex: u00su01ls01.ech.dcgs.mil)
- hostname (ex: u00su01ls01)
- hostname.{first segment of domain} (ex: u00su01ls01.ech)
- logstash-{site} (ex: logstash-u00)
- logstash-{site}.{first segment of domain} (ex: logstash-u00.ech)
- logstash – Generic alias that allows hosts to be relocated

A convenience script is provided to make the creation of the Logstash Server Certificate requests easy for the installer. To create PKI certificate requests for Logstash to run on the system:

1. Log in to any existing Linux server at the site where the Logstash Server is to be installed and do the following from your home directory:

```
# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/make_logstash_csr.sh | bash
```

2. When the script completes, 3 directories will be present in the location it was run
   - Reqs: This directory holds the CSR Request information in text format.
   - Keys: The private key associated with the certificate request for the Logstash instance
   - CSRs: The actual PKI Certificate Request for the Logstash instance.

3. The *.csr files should be submitted to the certificate authority for the system the Elastic Cluster is being installed on to obtain public certificates for each node. For systems submitted to the JWICS certificate authority (i.e., CTE High or Enterprise High), also include a text file named SANS.txt listing the SubjectAlternativeNames listed in the CSR (for each CSR).

4. Once the Logstash certificate(s) has been obtained, both the new cert and the key for each Logstash instance must be copied to the **certs** directory of the Elastic repo on the repo server ({xxx}su01ro01) for the associated site.

PKI Certificates must be in the following format for the installation scripts to work properly:

Public Cert: {hostname}.crt	examples: u00su01ls01.crt
Private Keys: {hostname}.key	examples: u00su01ls01.key

5.4.1.7.3 Root Certificates

As mentioned previously, the **elastic_cachain.pem** file containing the Root Certificate Authority (CA) and Sub CA used to issue all Elastic certificates must exist in the **certs** directory in the Elastic repository. If this file is not present in the **certs** directory on the repo server, you must create it:

1. Log in to the repo server in your environment ({xxx}su01ro01 box in your environment) and become root:
   ```
   # cd /etc/pki/ca-trust/source/anchors
   ```

2. You should see the following 2 files in this directory:
   - rootca.pem
   - subca.pem

   If they are not there, stop and ask for guidance from a Linux Admin or Elasticsearch SME.

   It is possible that the cachain.pem is already there. In any case, ensure that the source(s) signing all of the certs (there may be different sources for different certs obtained from a single request) are present in the cachain.

3. Create the elastic_cachain.pem file:

```
# cat rootca.pem subca.pem > / var/www/html/yum/elastic/install/certs
/elastic_cachain.pem
```

4. Copy the elastic_cachain.pem file into the extracted SCCM install; e.g., on the fileserver in **<install>\oadcgs-es-elastic-sccm-x.x.x.x\oadcgs-es-elastic-sccm\sccm\shareDir** replacing the dummy cachain.pem.

Note: You can run the following command to examine all the certs in the cachain.pem file.

```
# openssl crl2pkcs7 -nocrl -certfile cachain.pem | openssl pkcs7 -print_certs -text -noout
```

5.4.1.8 Elastic Puppet Modules

NOTE: A Puppet administrator is required to execute this section.

Puppet modules are used to automate some of the configuration on Linux hosts and the installation of some Elastic components. There are currently two modules added for Elastic, but minor modifications to the base OSIF configuration is also necessary during installation. A Puppet SME should be involved in adding these modules and ensuring Puppet is configured properly for Elastic.

NOTE: The Puppet modules must be installed before attempting installation or backout.

5.4.1.8.1 Adding Elastic profiles

Execution of the Elastic modules is controlled through the Elastic profiles that need to be added to each puppet branch. Add the following two files and contents to the <branch>/site-modules/profile/manifests directory.

elastic_clients.pp

```
class profile::elastic_clients (Boolean $install_beats=true, Boolean $restart_beats=false) {

  class { '::dsil_elastic_clients':
      install => $install_beats,
      restart_beats => $restart_beats,
   }
}
```

elastic_server.pp

```
class profile::elastic_servers {

  # call the module
  include dsil_elastic_servers

}
```

NOTE: These files are delivered with the modules and can be found in the "profile" directory in each module. As described above the files should be copied to the environments site-modules/profile/manifests directory to enable the modules.

5.4.1.8.2 Elastic Servers – dsil_elastic_servers Module

The **dsil_elastic_servers** module is used to configure Elastic, Kibana, and Logstash servers. The module opens the necessary ports, creates mounts, and performs other configuration tasks necessary to allow each component to run properly. Elasticsearch, Kibana, and Logstash are all installed following the procedures in this document, but they will not be able to run successfully unless the hosts are configured with this module.

The Heartbeat component of Elastic is also automatically installed with an initial configuration on each Logstash instance by this Puppet module. This module then ensures that Heartbeat is running on each Logstash host.

This module also controls the logstash.yml file for each Logstash instance.

IMPORTANT: Before deploying these updates the "node_specific" directory must be populated with the correct Logstash pipeline configuration for each site. Upon activation of these changes, puppet will take control of the logstash.yml file on each Logstash instance (All Sites).

Before creating an updated tag and updating the Puppetfile in the pe-control-repo to start using the dsil_elastic_servers module you must first create a node specific configuration file for any site that needs to run additional pipelines that are not contained in the base set specified in the default configuration.

If a specific configuration is not given for a site, it will use the default configuration supplied in the "data/logstash.yml" file supplied with the baseline.

Default pipelines: That default configuration will run the following pipelines that are expected to be run at each site.

- esp_filebeat
- esp_filebeat-singleworker
- esp_filebeat-logstash
- esp_heartbeat
- esp_linux_syslog
- esp_loginsight
- esp_metricbeat
- esp_winlogbeat

The "data/logstash.yml file" contains the puppet variable "dsil_elastic_servers::logstash::pipelines" containing the above default list of pipelines. The easiest and recommended way to create a node specific configuration file is to copy this file to use as a template.

Example of variable definition from file:

```
dsil_elastic_servers::logstash::pipelines: '["esp_filebeat", "esp_filebeat-
singleworker", "esp_filebeat-logstash", "esp_heartbeat", "esp_linux_syslog",
"esp_loginsight", "esp_metricbeat", "esp_winlogbeat"]'
```

Additional pipelines: The following pipelines should only be run at sites where the datatype is available for ingest:

- esp_eracent_database
- esp_hbss_epo
- esp_hbss_metrics
- esp_idm_database
- esp_postgres
- esp_puppet_database
- esp_sccm_database
- esp_serena_database
- esp_sqlServer_stats

Directory structure of dsil_elastic_servers repository:

dsil_elastic_servers/data:
logstash.yml – contains pipelines that should run at all sites

dsil_elastic_servers/data/node_specific:
CXXsu01ls01.yml (Add one for each site with additional pipelines)

NOTE: When setting up the node specific configuration files for the first time, it is recommended that you use the existing logstash.yml file at each site as a guide to populate the node specific file.

Steps to create a custom node specific configuration for each site on the enclave

1. Login to the Logstash instance at the site CXXsu01ls01
   a. C = Classifier (‘u’, ‘s’ or ‘t’)
   b. XX = site number

2. cd /etc/logstash and view the current logstash.yml file

   # cd /etc/logstash
   # cat logstash.yml

3. Examine the current list of pipelines being run at the site by looking at the xpack.management.pipeline.id array.
4. If the list contains only the default pipelines than a node specific configuration file is not needed for this site; continue onto the next site

5. You have identified a site that needs a node specific configuration. Copy "data/logstash.yml" to "node_specific/CXXsu01ls01.yml" configuration file

   Example: cp logstash.yml node_specific/s00su01ls01.yml

6. Update the "dsil_elastic_servers::logstash::pipelines" array in the newly created node specific file to contain the same pipelines that are currently running at the site.

   NOTE: The site should be running the default configuration with the possibility of additional pipelines. If any of the default pipelines were not in the original xpack.management.pipeline.id array, then they should be added.
7. Continue onto the next site

Once you have created node specific configuration files for any sites that are running additional pipelines, you can create a new tag for the dsil_elastic_servers repo and update the Puppetfile in the pe-control-repo to start using the updated dsil_elastic_servers module.

5.4.1.8.3 Elastic Clients – dsil_elastic_clients Module

The **dsil_elastic_clients** module is used to automatically install Metricbeat and Filebeat on Linux hosts. Metricbeat is installed on all Linux hosts in the OA DCGS system. Filebeat is installed on all Elasticsearch and Logstash hosts to collect log files from the Elastic applications. Filebeat is also installed on hosts to collect application logs.

Configurations for Metricbeat and Filebeat collection are distributed with this module.

5.4.1.8.3.1 Filebeat Configurations

Each host running Filebeat will receive a filebeat.yml configuration file which uses two directories for configuring what data to collect.

modules.d – holds configuration files supplied by Elastic. When a file is used from this directory it is updated for use on the DCGS system and a configuration is added to puppet for deployment.

The following Filebeat modules are part of this distribution.

- es.module.elasticsearch.yml.epp – Module to collect Elasticsearch logs
- ls.module.logstash.yml.epp – Module to collect logstash logs
- ls01.module.netflow.yml.epp – Netflow module (not currently used)

inputs.d – holds custom configuration files created for use on DCGS.

The following Filebeat inputs are part of this distribution.

- datacollector.input.yml.epp – input to collect all data from elastic data collector

A specific filebeat.yml configuration may override the default if necessary for a host. This is done when the default output port or some other base configuration differs for a specific hosts. The follow specific Filebeat configurations are part of the distribution:

- ls01.filebeat.yml.epp - filebeat.yml configuration for Logstash hosts
- soaesb.filebeat.yml.epp – filebeat.yml configuration for hosts running soaesb

5.4.1.8.3.2 Metricbeat Configurations

Metricbeat uses the metricbeat.yml and configurations from the modules.d directory if enabled to determine what data to collect.

The following Metricbeat configurations are part of the distribution.

- Module configurations:
  - all.module.docker.yml.epp – docker module configuration
  - all.module.system.yml.epp – System module configuration
  - es.module.elasticsearch-xpack.yml.epp – Elasticsearch module configuration
  - es.module.kibana-xpack.yml.epp – Kibana module configuration
  - ls.module.logstash-xpack.yml.epp – Logstash module configuration

- Metricbeat yml file configurations:

  A hostname in DCGS has a standard format and this module pulls sections of the hostname that it's running on to determine if a specific configuration file exists for the hosts. Two sections of the hostname are used for this:

  A) Site number - characters 2 and 3 of the hosts name
  B) Host designator – characters 8-15 (Most hosts only have 4 characters for this)

  Example: hostname – u00su01mp01

  Site number = "00"
  Host designator = "mp01"

  This module will determine the configuration to use for Metricbeat in the following manor:

  If a "<Host designator>-<Site number>-metricbeat.yml.epp" configuration file exists then

  Use that as the metricbeat.yml file for this host.

  Else if a <Host designator>-metricbeat.yml.epp configuration file exists then,

  Use that as the metricbeat.yml file for this host.

  Else

  Use the metricbeat.yml.epp file for the metricbeat.yml file for this host.

  Configuration files deployed with this baseline include:

  - metricbeat.yml – Generic configuration

- etcd01-00-metricbeat.yml.epp – Postgres ETCD configuration
- etcd02-00-metricbeat.yml.epp – Postgres ETCD configuration
- etcd03-00.metricbeat.yml.epp – Postgres ETCD configuration
- mp01-00.metricbeat.yml.epp – Arcsight ESM configuration
- mp02-00.metricbeat.yml.epp - Arcsight ESM configuration (if clustered)
- mp03-00.metricbeat.yml.epp - Arcsight ESM configuration (if clustered)
- mp04-00.metricbeat.yml.epp - Arcsight ESM configuration (if clustered)
- mp05-00.metricbeat.yml.epp - Arcsight ESM configuration (if clustered)
- mq01.metricbeat.yml.epp – ArcSight ARMC configuration
- mq02-00.metricbeat.yml.epp – ArcSight ARMC configuration
- mr01.metricbeat.yml.epp – ArcSight Logger configuration
- mr02-00.metricbeat.yml.epp – ArcSight Logger configuration
- mt01-00.metricbeat.yml.epp – ArcSight Syslog configuration
- pg01-00.metricbeat.yml.epp – Postgres database configuration
- pg02-00.metricbeat.yml.epp – Postgres database configuration
- pg03-00.metricbeat.yml.epp – Postgres database configuration
- pup1-00.metricbeat.yml.epp – OSIF Puppet Sever configuration
- pup1-0a.metricbeat.yml.epp – OSIF Puppet Sever configuration
- pup1.metricbeat.yml.epp – OSIF Puppet Compiler configuration

IMPORTANT: The configurations are designed for an installation at ECH (Site 00) and will need some adjustments if this install is being done on a different enclave.

5.4.1.8.4 Elastic Node Groups

Node Groups (previously Classifications) for Elastic also need to be configured in the Puppet web console. The rules in each node group should be set up properly to ensure that the correct Elastic Puppet module is run on the correct VMs.

There is an Elastic Servers node group for Logstash and Elastic Nodes. There is an Elastic Clients node group for all other hosts.

Following is an example of what the node groups for Elastic should look like:

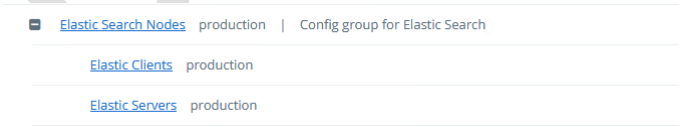


Figure 2 Elastic node groups

The node matching rules for Elastic clients should include all Linux hosts. The following rule allows the Elastic Clients Puppet module to be run on all RedHat boxes.

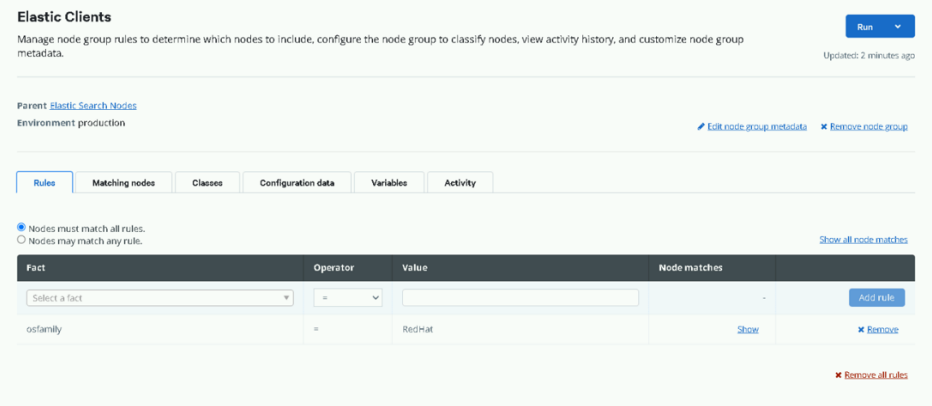


Figure 3 Node Matching Rules

The configuration for Elastic clients should include the **profile::elastic_clients** class, as shown:

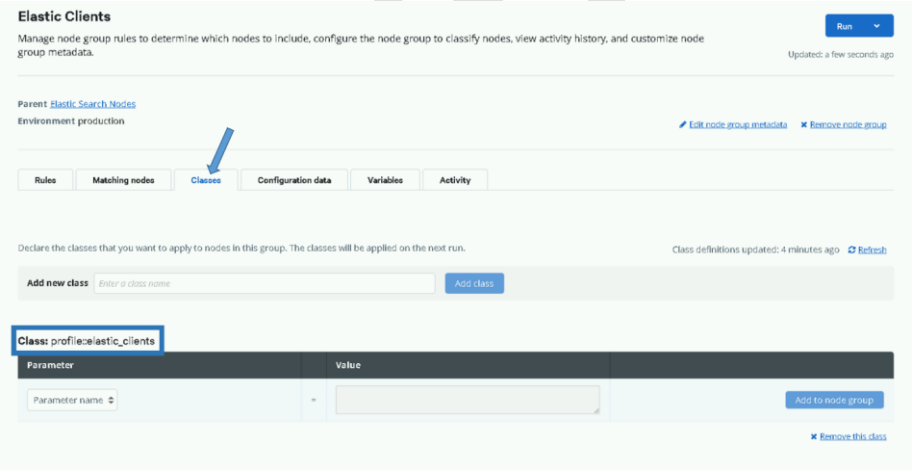


Figure 4 profile::elastic_clients

The node matching rules for Elastic servers should include all Logstash and Elastic VMs. The following rules allow the Elastic Servers Puppet module to be run on all Elastic and Logstash Servers.

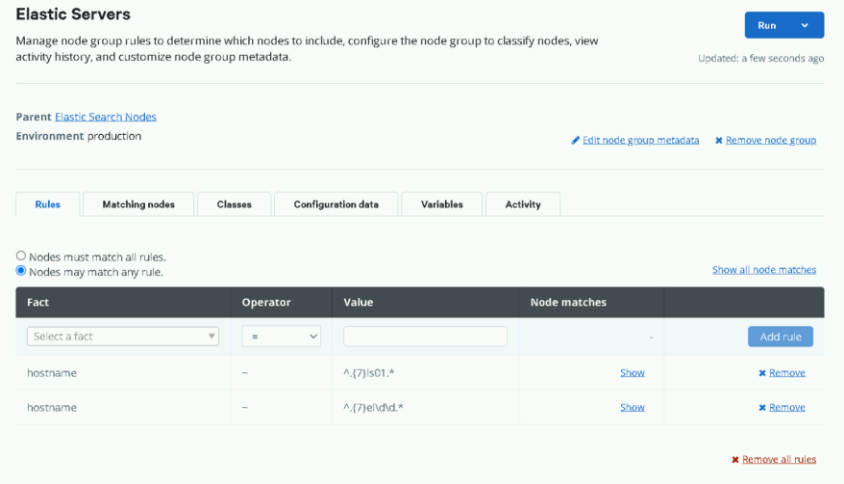


Figure 5 Allow Elastic Servers Puppet module to run on all Elastic and Logstash Servers

The configuration for Elastic servers should include the **profile::elastic_servers** class, as shown:

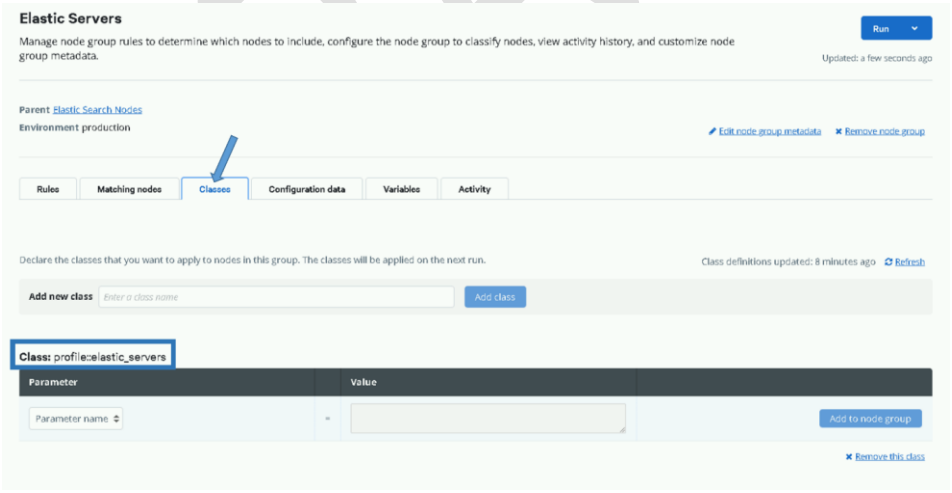


Figure 6 profile::elastic_servers

5.4.1.9 Configure NSX Load Balancer

NOTE: The Network Administrator role on the NSX manager is required to execute this section.

Kibana, the web interface to Elastic, can be accessed by navigating to https://kibana on the DCGS system. This URL directs the user to a load balancer that will forward the requests to the appropriate Kibana instance. Depending on the configuration there can be one or more Kibana instances available to handle user requests. Before proceeding with the installation ensure that the NSX Load Balancer is configured to handle user requests.

To configure NSX for Elasticsearch, refer to *ES-018 - VMware - NSX-V Load Balancer Deployment Guide.*

Refer to Section XXX Kibana for information on servers where Kibana instances will be running, which is based on the size of the cluster being installed.

The load balanced name for each service must be created in DNS, if this was not completed during the deployment of the load balancer execute the following steps.

1. Create a **DNS A** record for the Kibana load balancer address which points to the virtual IP for the Kibana service.
   Example: FQDN for target host: **`kibana.ech.dcgs.mil`**
2. Perform testing and validation of the Kibana DNS A record for the ElasticSearch Kibana load balancer portal FQDN and IP address.
3. Open a command window and run the following commands:
   `ping` **`kibana.ech.dcgs.mil`**
   `nslookup` **`kibana.ech.dcgs.mil`**

IMPORTANT: In the 8.6 upgrade there is a change to the Kibana status API that affects the configuration of the service monitor of the load balancer. The following update must be made to allow the https://kibana url to continue to function properly. This update may not be present in the *ES-018 - VMware - NSX-V Load Balancer Deployment Guide.*

Edit the Service monitor and make the following changes:

- Modify “Expected” from 204 to 200
- Remove “green” from the “Receive” field

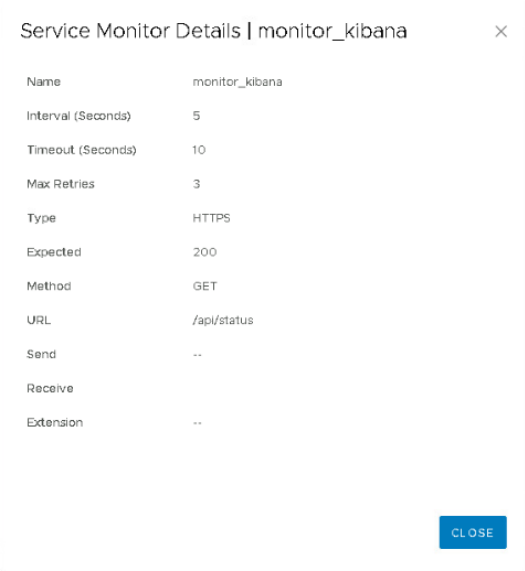


Figure 7- Updated Service Monitor configuration

5.4.1.10 Elasticsearch and Logstash VM Creation

All the VMs needed for this installation should have already been provisioned according to the requirements tables shown previously. The IP Addresses and DNS Aliases should have already been assigned. If 2nd drives are allocated, they have been configured and are ready for use.

5.4.1.11 Service Account Kerberos Management (SAKM)

During installation, Elasticsearch is modified to be run by the Elastic service account. Elastic relies on the Kerberos ticket for the Elastic service account to be automatically updated by the SAKM scripts. Before deploying Elastic or Logstash on any VMs, ensure that SAKM has been installed. If it hasn't been installed, you can utilize the following steps to configure it. The installed SAKM must be at least **refreshticket-1.0.5-1.1.noarch**.

> **NOTE:** To check the installed SAKM version run the following command on the system to check:

```
# rpm -qa | grep refreshticket
```

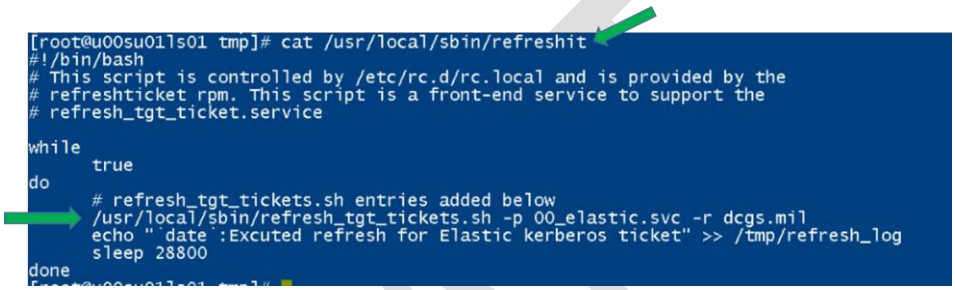
5.4.1.11.1 Verify SAKM Installed

The SAKM package may be installed but not configured for the Elastic service account. To verify that SAKM has been configured to refresh the Kerberos ticket for the Elastic service account do the following on each Elastic VM:

1) Show the contents of the rerfrehshit script:

   # cat /usr/local/sbin/refreshit

   The output should look very similar to this:



2) Validate that there is a line for the Elastic service account used for the cluster. The example above shows a line for the "00_elastic.svc" account.
3) Verify that this line is configured correctly by running it manually from the command line. Just cut and paste it.

   Example:

   # /usr/local/sbin/refresh_tgt_tickets.sh -p 00_elastic.svc -r dcgs.mil
   success

4) Validate that "success" is returned.

If **success** is returned, then SAKM is configured on the VM.

If access is denied, ensure the permission on the file /usr/local/sbin/refresh_tgt_tickets.sh are set to read and execute for all users (i.e. 755). Incorrect permission may be caused by an improper SAKM version; again ensure you have at least refreshticket-1.0.5-1.1.noarch installed.

If SAKM is installed and working correctly on this VM move onto the next one to test, if it's not working or not installed move onto the next section to Install/Configure SAKM.

5.4.1.11.2 Install/Configure SAKM

This section should only be executed if SAKM is not installed or working correctly on an Elastic or Logstash Instance.

5.4.1.11.2.1 SAKM Install

```
# yum install refreshticket
```

NOTE: If repo for refreshticket is not available you can copy the refreshticket RPM to the host and run `rpm –i refreshticket-1.0.5-1.1.noarch`

5.4.1.11.2.2 Create SAKM Keytab

You can create a keytab file using the ktutil command.

1. First, as root, create the directory where the keytab will reside.

   ```
   # mkdir /usr/local/etc/sakm/XX_elastic.svc
   # chown XX_elastic.svc /usr/local/etc/sakm/XX_elastic.svc
   # chmod 700 /usr/local/etc/sakm/XX_elastic.svc
   ```

2. su to the AD account and run the ktutil command to create the keytab.

   ```
   # su – XX_elastic.svc

   $ ktutil
   > addent -password -p XX_elastic.svc@<REALM> -k 1 -e aes256-cts-hmac-
   sha1-96
     Enter password for XX_elastic.svc@<REALM>:
   > wkt /usr/local/etc/sakm/XX_elastic.svc/XX_elastic.svc.keytab
   > exit

   $
   ```

3. Change permissions.

   ```
   $ chmod 700 /usr/local/etc/sakm/XX_elastic.svc/XX_elastic.svc.keytab
   ```

4. Test the keytab is working.

   ```
   $ kinit XX_elastic.svc -k -t
   /usr/local/etc/sakm/XX_elastic.svc/XX_elastic.svc.keytab
   ```
   # No errors should be returned

5.4.1.11.2.3 Set Up Background Management

The Elastic service account must be added to the script that continually updates the Kerberos tickets. Run the following command:

```
# refresh_tgt_tickets.sh –m –p XX_elastic.svc –r <REALM>
```

Added " /usr/local/sbin/refresh_tgt_tickets.sh -p XX_elastic.svc -r <REALM>" to /usr/local/sbin/refreshit

NOTE: <REALM> will be replaced with the machine's REALM. If the entry already exists for the Elastic service account, running the command will not do anything and the script will return **Entry exists in refreshit script – No Update Necessary!**

NOTE: If the script was updated, it must be stopped and restarted because the refreshit script is already running using the previous file content.

```
# pkill refreshit

# /etc/rc.local
```

5.4.1.12 Device Monitoring

NOTE: This prerequisite is not for the installation of Elastic but for the configuration of devices to be monitored. This information should be available before setting up devices to be monitored in section 5.4.6.1.6. An Infrastructure SME will be required to obtain or configure usernames and passwords on devices.

This implementation of Elastic provides the ability to monitor infrastructure devices on DCGS. During the install process, the installer will be asked to provide either SNMP or Rest usernames and credentials to allow querying of data from specific devices. It is a prerequisite of configuring each device that is to be monitored that it has the correct access configured. Some devices are accessed using SNMP and others via a Rest API; the following table shows the devices that are currently supported and the means of access.

Table 7 Supported Devices and Access Types

Device	Access Type
XtremIO	Rest API
Isilon	Rest API
Data Domain	SNMP
Cisco Switches	SNMP
Fx2 Chassis	SNMP
Fc6xx Blades	SNMP
R6xx Servers	SNMP

Device Information needed to monitor XtremIO, Isilon and Data Domain devices:

- URL – This is the URL to access the device's web interface.
- Username – Username to access the device via REST API.
- Password – Password for Username.
- Display Name – Short unique description of device (ex: ech-isilon, ech-xtremio).
- Host – DNS resolvable name or IP address of host (ex: u00av01xms1).

Device Information needed to monitor Cisco switches, fx2 chassis, fc6xx and r6xx blades:

- URL – This is the URL to access the device's web interface (CMC, IDRAC, Cisco Prime).
- Username – Username to access the device via SNMP.
- Password – Authentication Password for Username.

- Priv Password – Privacy Password for Username.
- Display Name – Short unique description of device (ex: ech-fx2, ech-5k).
- Host – DNS resolvable name or IP address of host (ex: u00av01xms1).

NOTE: Because we are using SNMP v3 there are two passwords required.

SNMP and Rest Access are not Elastic-specific and can easily be tested using the following commands for each device. The following table provides test commands that can be used to verify access to each device type.

Table 8 Access test command per device

Device	Command Used to Test Access
Data Domain	snmpwalk -v3 -l authPriv -u <username> -a SHA -A <auth password> -x AES -X <priv password> <IP of Datadomain>:161 1.3.6.1.4.1.19746.1.13.1
Cisco Switches	snmpwalk -v3 -l authPriv -u <username> -a SHA -A <auth password> -x AES -X <priv password> <IP of Switch>:161 1.3.6.1.4.1.9.9.109.1.1.1.1
Fx2/FC630/R630	snmpwalk -v3 -l authPriv -u <username> -a SHA -A <auth password> -x AES -X <priv password> <IP of device>:161 1.3.6.1.4.1.674.10892.2.2.1
Isilon	curl -v -XGET -k -u '<username>:<password' 'https://<Isilon IP>:8080/platform/8/protocols/nfs/exports' | python -m json.tool
XtremIO	curl -v -XGET -k -u '<username>:<password>' 'https://<xtremIO hostname or IP>/api/json/v2/types/events'

5.4.2 Final pre checks

Assumptions: At this point all VMs for Elastic have been allocated, joined to Puppet, and SAKM has been installed.

NOTE: The NFS share requires a Kerberos ticket to access. This ticket is currently maintained by the refreshit script that was put in place by SAKM.

These install instructions are for installing **Enterprise Elasticsearch**. If a previous version of Elastic is running in this environment, the cluster and all Elastic components must be removed before proceeding.

STOP – If you are installing new nodes, do you have PKI certificates for them?

Prior to performing the installation, all PKI certs must be available in the **certs** directory in the Elastic repo.

IMPORTANT: THE INSTALL ORDER MATTERS

When installing Elastic and its components, the order of installation matters. The following sections give instructions on installing the different components of Elastic. Use the following as a guide on the order to execute each section.

1. Copy Elasticsearch, Logstash, and Kibana RPMs to Elastic repository and rebuild.
2. Install Elasticsearch; ensure cluster is at 100% health.

3. Install all Kibana instances.
4. Load Roles.
5. Load Role Mappings.
6. Validate Active Directory Login
7. Load Audit Settings.
8. Add License
9. Update ingest pipelines
10. Load Templates.
11. Load Spaces & Saved Objects.
12. Load Index Lifecycle Management Policy.
13. Bootstrap the initial write indexes.
14. Setup snapshot lifecycle management
15. Load Enterprise Services Centralized pipelines into Elastic.
16. Install health data watcher
17. Install all Logstash instances (All Sites) and verify they are running.
18. Verify Role Mappings and Roles
19. Remove unneeded accounts
20. Install beats
21. Setup forwarding of Linux syslog
22. Setup site specific Ingest

STOP - DO NOT PROCEED WITHOUT THE INSTALL DIRECTORY ON THE REPO

Before proceeding ensure that the install directory has been copied to the Elastic repo. The directory should be in the **/var/www/html/yum/elastic** repo directory and should be named **install**. The contents of the directory should match the description in Section 4.3.

If possible, use MobaXterm to log in to all the Elastic VMs for this installation. This will allow you to move easily between the VMs without logging out during the installation.

5.4.3 Elasticsearch

The following steps require that the admin have **root** permissions to perform the install. The **#** at the beginning of a command signifies that if should be run as root. If you don't know how to become root on a Linux machine, you should not be performing this installation.

5.4.3.1 Setup Repo with Core RPMs

NOTE: A Linux administrator will be needed to execute this section.

1. Before running the installation, the RPMs for the core Elastic components must be copied to the Elastic repo on the DCGS repo server (ex: u00su01ro0) and the repo must be recreated. Copy these components from wherever you decided to stage the Elastic-Elastic Core Components in section 4.3.
2. **Copy** the following RPMs to the Elastic repo (/var/www/html/yum/elastic):
   - elasticsearch-X.X.X-x86_64.rpm
   - kibana-X.X.X-x86_64.rpm

- logstash-X.X.X-x86_64.rpm

3. Ensure RPMs have correct owner/group:

```
# chown -R apache:apache *
```

4. Repo files must have selinux context **httpd_sys_content_t** set. If you copied the RPMs into the directory, they will automatically get this context set. If you moved them, they won't. Ensure all files have the correct context set by executing:

```
# ls -lZ
```

```
[root@u00su01ro01 elastic]# ls -lZ
drwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 certs
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elastic_cachain.pem
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elastic.key
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elasticsearch-8.6.2-x86_64.rpm
drwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 install
drwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 keystores
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 kibana-8.6.2-x86_64.rpm
-rwxrwxr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 logstash-8.6.2-x86_64.rpm
drwxr-xr-x. root   root   unconfined_u:object_r:httpd_sys_content_t:s0 repodata
[root@u00su01ro01 elastic]#
```

Figure 8 Set httpd_sys_content_t

5. If all files do not have **httpd_sys_context_t** set, execute the following:

```
# restorecon  *
```

6. Recreate the Elastic repo so it's ready for use:

```
# createrepo ./
# gpg --detach-sign --armor ./repodata/repomd.xml
```

NOTE: There are 2 dashes in front of **detach-sign** and **armor** in the above command.

7. Confirm overwriting the file (if it already exists)
   Enter **y** to overwrite

5.4.3.2 Verify Service Account (From Each VM)

WARNING: Elasticsearch is modified to run as the **xx_elastic.svc** account during installation. Elastic must run as the service account, or it will not be able to write data over the NFS to the Isilon. See prerequisites in the **Error! Reference source not found.** Section for details.

1. Log in to each Elastic node and sudo to root.
2. su to the Elastic service account for this site (*{sitecode}*_elastic.svc):

```
# su 00_elastic.svc
```

3. Verify the service account can access the Isilon share on this node.

```
# cd /ELK-nfs
# ls -la
```

Verify the location is not empty; it should have at a minimum "." and ".."

5.4.3.3 Elasticsearch Install – Adding a Node

This step will be done on each VM that will become an Elasticsearch node.

1. Log in to the VM and sudo to root
2. Verify iptables are set up correctly. This should be handled by Puppet. All Elastic nodes should be included in the "Elastic Servers" Classification on the Puppet server.

   `# iptables --list -n` (there are 2 dashes in front of list)

3. **Verify ports 9200 and 9300 are open. You should see a line for *multiport dports 9200, 9300 /* 106 allow input from other Elastic nodes and clients */.***
   NOTE: If these ports are not open stop and verify hosts are set up correctly in Puppet.
4. To determine if data is stored locally on this node, check the table in Section 2.1.1 for the cluster size to be installed. If it will be, you must verify that there is an **ELK-local** directory already created for the data.
   a. Is this a 10-node or 15-node cluster and this node is a **Master** node? The ELK-local directory for a master node in these clusters is located at root level on the OS disk. If the directory does not exist, create it.

      ```
      # mkdir /ELK-local
      # chown XX_elastic.svc /ELK-local
      ```

   b. Elastic nodes that store "hot" data or machine learning nodes usually have a 2nd local disk for storage. The ELK-local directory should be the mount point for the volume group allocated from that 2nd drive for this type of node. This is also true for **Master** nodes in a 6-node cluster configuration.

      The files system should be set up to mount the **ELK-local** drive automatically in the **/etc/fstab** for this node. Check the **/etc/fstab** file and verify a line similar to the following exists:

      ```
      /dev/mapper/elk_vg-elk /ELK-local  xfs defaults 0 0
      ```

      Doing a "df -h" command on the ELK-local file system should show something like this:

      ```
      [root@u00su01el05 /]# df -h /ELK-local
      Filesystem              Size  Used Avail Use% Mounted on
      /dev/mapper/elk_vg-elk  1.5T  258G  1.3T  18% /ELK-local
      ```

      Figure 9 df command

      If this mount point does not exist, please consult with a Linux administrator or the OA DCGS Elastic SME on how to create it before proceeding.

   NOTE: All nodes should have an **ELK-nfs** mount point to the Elastic share on the Isilon even if they store data locally.

5.4.3.4 Verify VM can see Elastic Repo

1. Identify the Elastic Repo by listing all the repos. It should now contain 3 items.

   ```
   # yum repolist all
   ```

2. List the contents of the Elastic repo to verify that it has the Elasticsearch RPM using the following command.

   ```
   yum repo-pkgs {elastic repo name} list
   ```

   example: `# yum repo-pkgs elastic list`

   NOTE: If you do not see Elasticsearch in the repository **do not proceed**. You can attempt a **yum clean all** and try the previous steps again. If you still don't see the Elastic repo, you may need to verify it is set up properly.

5.4.3.5 Install Elasticsearch

Assumption: Elastic RPMs and installation scripts have been added to Elastic repo.

```
# curl -s -k https://{xxx}su01ro01.`hostname -
d`/yum/elastic/install/installElasticNode.sh | bash
```

NOTE: The back quote characters (`) used in the above command are the on the key with the tilde (~).

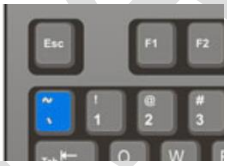


Figure 10 back quote characters (`)

NOTE: You can verify the path to the Elastic repository by checking the repo definition found in **/etc/yum.repos.d/elastic-search-rpms.repo** (the name of the repo may differ).

Repeat for all nodes.

5.4.3.6 Verify SSL Settings for ElasticSearch

The installation script executed previously should have populated the **/etc/elasticsearch/certs** directory with the PKI certificates for this node, along with the root certificate authority's public certificate. Verify these certs have been installed correctly.

Change the certificate directory for the node:

```
# cd /etc/elasticsearch/certs
```

Verify certificates are present.

NOTE: If the certificates are not present, stop. This must be resolved before the Elasticsearch service can run.

Verify that /etc/hostname and the hostname command use the same name as the certificate files (i.e server name without domain). If there is a mismatch, hostname and /etc/hostname should be changed to match. (Alternately, if there is a need, links to the certificate files with the used name can be created in the same location.)

Repeat these steps on all nodes before continuing.

5.4.3.7 Start & Test Elasticsearch

Once installation and configuration are complete on all nodes, you can start Elasticsearch. Start the nodes from smallest to largest in the node numbering scheme. All master nodes will be started first.

1. Run the following command on each node (starting with Node 1):

   ```
   # systemctl start elasticsearch.service
   ```

   (Repeat for all nodes.)

2. Wait at least 5 minutes to give the nodes a chance to start, then proceed.

   ```
   # systemctl status elasticsearch.service
   ```

3. Once all nodes have been started, you must configure the initial passwords for reserved Elasticsearch users.
   NOTE: This is only done once on any node.
4. Log in to any of the Elastic VMs and become root.
5. Run the following command:

   ```
   # /usr/share/elasticsearch/bin/elasticsearch-setup-passwords
   interactive
   ```

   Set all these passwords to **elastic** to allow the rest of this install to be successful.

   You will be prompted for passwords for the following reserved users:
   - elastic
   - apm_system
   - kibana_system
   - logstash_system
   - beats_system
   - remote_monitoring_user

6. After setting the initial passwords, run the following commands to check the status of the clustet using the local “elastic” account. Remember, do not include the {}.

```
curl -k -u elastic:elastic https://elastic-node-{x}:9200/_cluster/health?pretty
```

NOTE: This is a good test to make sure you set the Elastic user password to elastic properly.

IMPORTANT: If the above query do not work, STOP and contact an OADCGS SME for guidance.

5.4.4 Kibana

Kibana is the web interface used to visualize data in Elasticsearch. Kibana currently runs in conjunction with Elasticsearch on an Elastic VM. There may be one or multiple instances of Kibana running to support user access to the Elastic cluster. All Kibana instances are accessible via https://kibana from any site. An NSX load balancer is set up to route the traffic to the Kibana to allow consistent navigation for the users.

NOTE: It is possible to still access Kibana instances, directly bypassing the NSX Load balancer, by using the hostname they are running on in the URL. For example, if Kibana is running on u00su01el03 then it can be accessed directly at https://u00su01el03:5601 or https://elastic-node-3:5601. This direct access should only be used during this procedure for installation checks. It can also be used for debugging issues or maintenance but should not be used by general users.

Use this table to determine which Elastic nodes to install Kibana on.

Table 9 Elastic nodes to install Kibana on

# of Nodes in Cluster	Elastic Nodes where Kibana is installed
6	Node 3 and Node 4
10	Nodes 7 and Node 10
15	Node 10 and Node 15

NOTE: You must be root to install Kibana.

5.4.4.1 Install Kibana

```
# sudo su
```

NOTE: Kibana can take up to 45 minutes to install. To avoid interruption of the installation, the screen command will be used to create a session to run the install command. For more information about the screen command consult the *Linux man page* for **screen**.

```
# screen -S install-session

# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/installKibana.sh | bash
```

If you are installing multiple Kibana instances, start the next one as well.

NOTES

- **If your SSH session times out while waiting for Kibana to be installed, return to your install-session by typing the following after re-establishing an SSH session to the computer.**

  ```
  # screen -d -r install-session
  ```

- **To detach from a running screen session type ctrl+a ctrl+d.**
- **If the Kibana installation is terminated for any reason, STOP and contact an OADCGS SME for guidance.**

Run Puppet on the host after Kibana is installed to open necessary ports and update the kibana yml configuration file:

```
# puppet agent -t
```

Verify iptables are set up correctly. This should be handled by Puppet. Kibana normally runs on an Elastic node. That node (or nodes if multiple instances) should be included in the **Elastic Servers** Classification on the Puppet server.

```
# iptables --list -n
```

Verify port 5601 is open. If this port is not open, stop and verify hosts are set up correctly in Puppet.

5.4.4.2 Start & Test Kibana (For Each Kibana Node, If Applicable)

This test will access this instance of Kibana by explicitly specifying the name of the VM where it is installed in the URL. General user access to this instance should be controlled by the NSX load balancer using the https://kibana URL once it's configured.

1. Start Kibana:

   # systemctl start kibana

   NOTE: The puppet run in the previous section may have already started Kibana, this step is just to ensure it's running. If it's already running this will not do anything.

2. Give Kibana a few minutes to come up and connect to Elastic, then verify that it started correctly.

   ```
   # sytemctl status kibana
   ```

3. If Kibana starts with no issue, test Kibana in any browser from any computer that has network access to the Kibana node. Type the following URL into the browser:

   ```
   https://elastic-node-{x}:5601
   ```

   If it loads to a Kibana login window, success!

4. You can now log in to Kibana using the **elastic** reserved user account for checking things during the remainder of this installation. You should have set the password to this account to **elastic** in section 5.4.3.7.

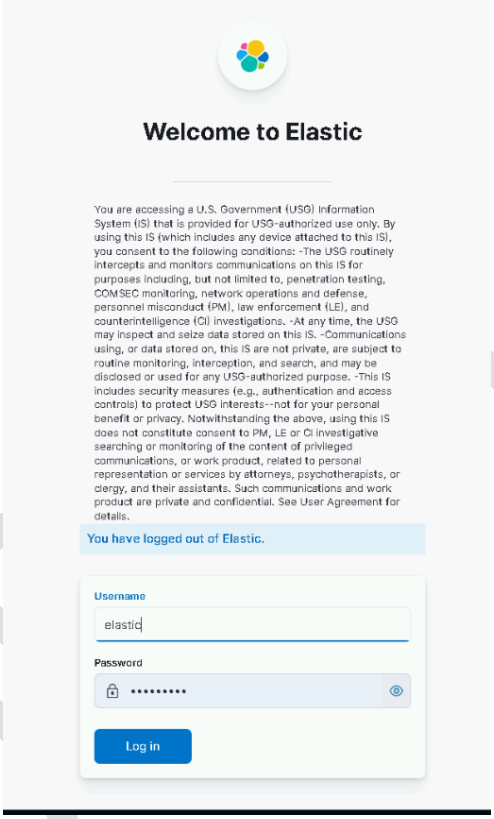


Figure 11 Login Screen

5.4.4.3 Disable Telemetry (On One Kibana Node)

By default, Usage Collection (also known as Telemetry) is enabled. This must be disabled for DCGS.

1. Navigate to Kibana in your browser: https://elastic-node-{x}: 5601
2. Log in to Kibana with username **elastic** and password **elastic**.

3. Click the menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.
4. Scroll to **Management** at the bottom and select **Stack Management**.

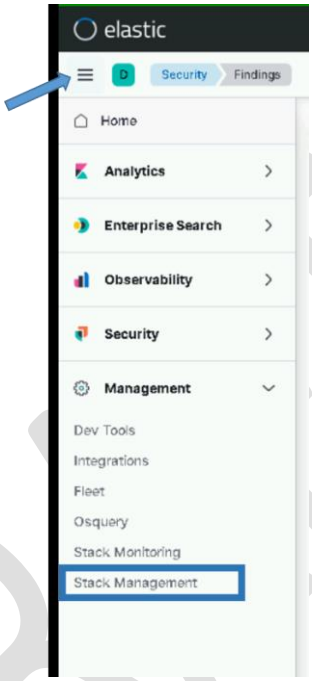


Figure 12 Stack Management

5. The **Stack Management** page displays. Under Kibana select **Advanced Settings** on the left side.
6. Enter **telemetry** in the search bar to filter.

7. If **telemetry:enabled** is **On**, click the slider to turn it **Off**.

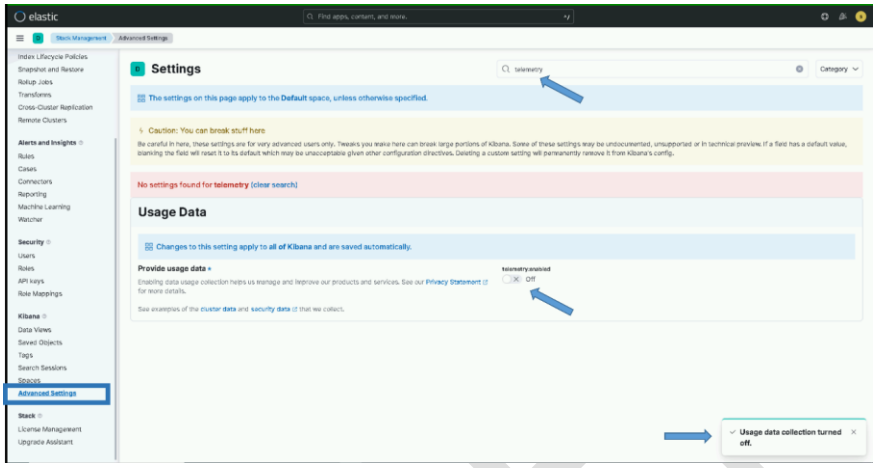


Figure 13 Disable Telemetry

5.4.5 Elastic Search Configuration

IMPORTANT: The installer must be a member of the "ent elastic admins" group to complete some of the following sections.

5.4.5.1 Kibana Roles

With role-based access control (RBAC), you can provide users access to data, tools, and Kibana spaces. On DCGS we have custom roles setup that are mapped to active directory groups.

5.4.5.1.1 Load Kibana Roles

To load Kibana roles for the cluster run the following command as root from any of the running Elastic nodes.

```
# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/load_roles.sh | bash -s install
```

NOTE: These settings are dynamically applied, and no restarts are necessary. Running this script multiple times will continue to update the roles and is not harmful.

5.4.5.1.2 Verify Kibana Roles are Loaded

To verify the Kibana roles were successfully loaded:

1. Login to Kibana with the "elastic" user
2. Click the menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.

3. Scroll to **Management** at the bottom and select **Stack Management**.
4. The **Stack Management** page displays. Under Security select **Roles** on the left side.
5. Enter **dcgs** in the search bar to see the 5 roles that should be loaded.

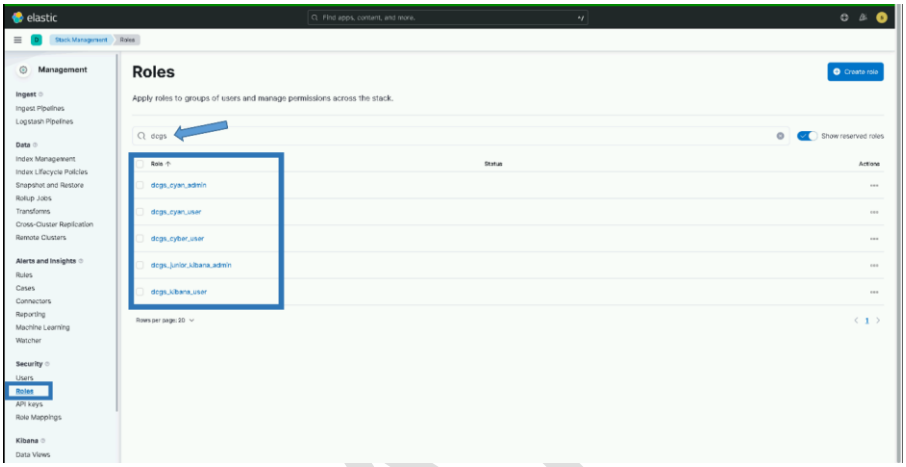


Figure 14 Roles

5.4.5.2 Role Mappings

Role mappings are used to map active directory groups to Kibana roles. This allows privileges in Kibana to be assigned using DCGS active directory groups.

5.4.5.2.1 Load Role Mappings

To load role mappings for the cluster, run the following command as root from any of the running Elastic nodes. This script uses the ldapsearch binary contained in the openldap-clients package. If the package is not installed, the script will attempt to install it. If the package is unavailable, the script will fail, and you will need to install it manually to proceed.

```
# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/load_role_mappings.sh | bash -s install
```

NOTE: These settings are dynamically applied, and no restarts are necessary. Running this script multiple times will continue to update the role mappings and is not harmful.

5.4.5.2.2 Verify Role Mappings are Loaded

To verify the Kibana roles were successfully loaded:

1. Login to Kibana with the “elastic” user
2. Click the menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.
3. Scroll to **Management** at the bottom and select **Stack Management**.

4. The **Stack Management** page displays. Under **Security** select **Role Mappings** on the left side.
5. Enter **dcgs** in the search bar to see the 6 DCGS role mappings that were loaded.

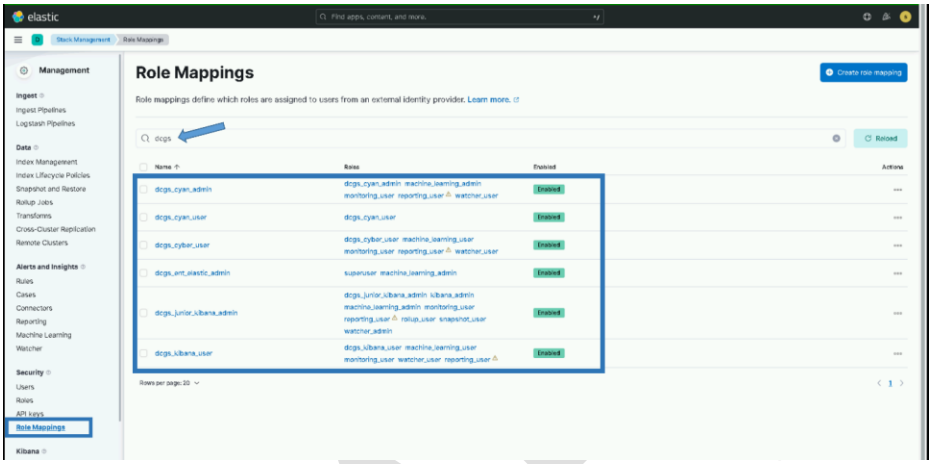


Figure 15 Role Mappings

5.4.5.3 Validate Active Directory Login

After loading the roles and role mappings users should be able to login to Kibana using any DCGS privileged active directory account (.adm, .wks, .dba, etc). This section is to validate this is working before continuing with the installation.

1. Go to the Kibana login page: https://kibana (If you are already logged in as "elastic" from a previous step logout)
2. Verify you can login to Kibana using your <firstname.lastname>.adm account
3. If the login is successful, then proceed with the installation. If the login fails then **STOP and contact an OADCGS SME for guidance.**

NOTE: From this point forward you will be using your own user name and password when executing the installation scripts.

5.4.5.4 Audit Settings

The logging of security-related events such as authentication failures and refused connections is enabled when installing an Elasticsearch node. The audit information will be written to the **/var/log/elasticsearch/<clustername>_audit.json** file, for example, **ECH_Cluster_audit.json**.

5.4.5.4.1 Load Audit Settings

To set the audit settings to control the number of events logged for the cluster, run the following command as root from any of the running Elastic nodes:

```
# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/load_auditsettings.sh | bash
```

NOTE: These settings are dynamically applied, and no restarts are necessary.

5.4.5.4.2 Verify Audit Settings

Verification of the audit settings and other settings in Elastic can be done from the Kibana Dev Tools console. To access the console:

1. Click the menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.
2. Scroll to **Management** at the bottom and select **Dev Tools**.

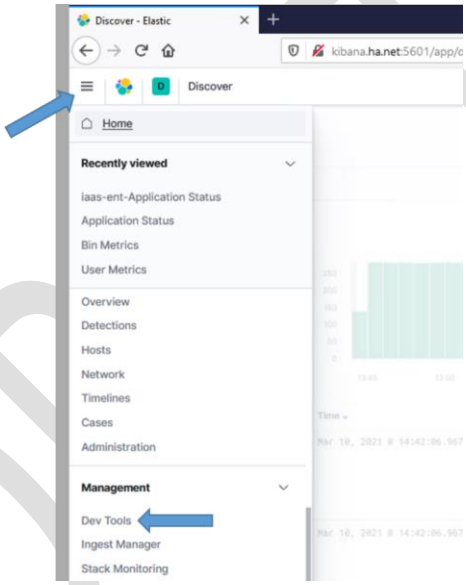


Figure 16 Dev Tools

3. The **Dev Tools** console displays.
4. To verify the settings are in place, run the following command from the Kibana Dev Tools console:

   ```
   GET _cluster/settings
   ```

5. Verify the following audit settings are contained in the output:

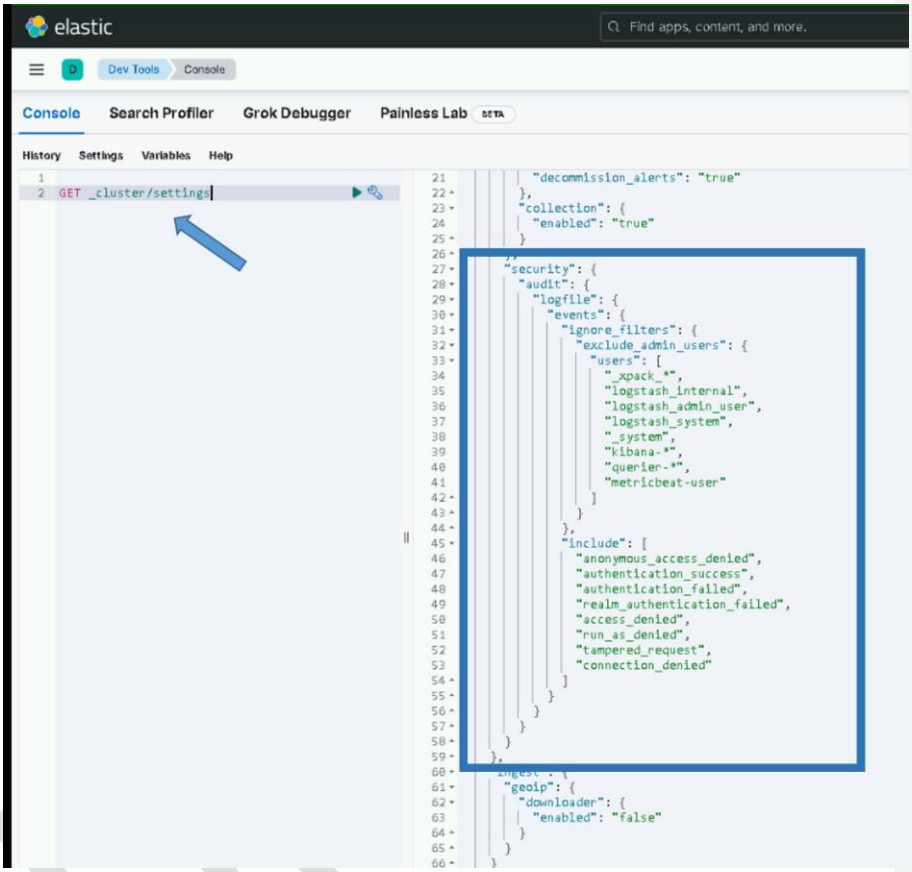


Figure 17- Expected audit settings

5.4.5.5 Add License for Elasticsearch

If the cluster is running and is 100% healthy, add the license key by executing the following.

NOTE: This only needs to be done on the initial cluster install or whenever a license update is required.

```
# curl -k https://xxxsu01ro01/yum/elastic/install/updateLicense.sh | bash
```

5.4.5.6 Update Ingest Pipelines in Elasticsearch

Elasticsearch ingest pipelines are used to aid ingest of data into Elasticsearch. Many Filebeat and Winlogbeat modules have associated ingest pipelines. These pipelines are not loaded into Elasticsearch automatically; they must be loaded each time you install or upgrade beats. Ingest pipelines only need to be loaded one time for use with all beat instances. To make the loading of the ingest pipelines easy, a convenience script has been written to load the pipelines. This script **MUST** be run each time beats are upgraded on the system.

To load the ingest pipelines, run the following command as root from any of the running Elastic nodes:

```
# curl –k https://{site code}su01ro01.`hostname –d`/yum/elastic/install/update_ingest_pipelines.sh | bash
```

1. Verify the ingest pipelines are loaded in Elastic. Select the **Ingest Pipelines** page under **Stack Management** to view all the ingest pipelines loaded into Elasticsearch. Filter the page with the version number you are installing to see the ingest pipelines for that specific version.

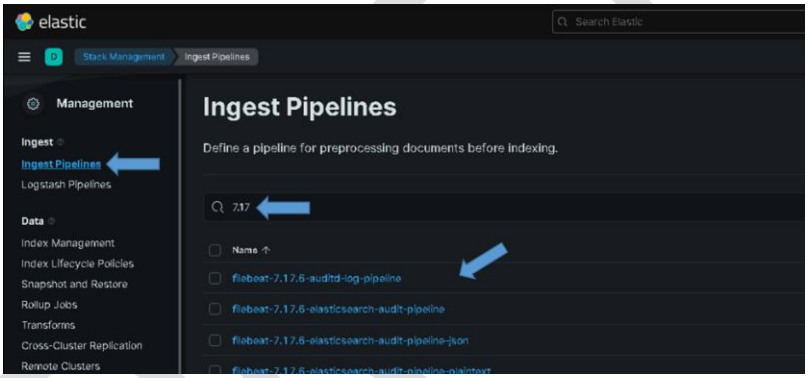


Figure 18 Example of Ingest Pipelines for version 7.17

For version 8.6.2 you should see the following pipelines:

- `filebeat-8.6.2-auditd-log-pipeline`
- `filebeat-8.6.2-elasticsearch-audit-pipeline`
- `filebeat-8.6.2-elasticsearch-deprecation-pipeline`
- `filebeat-8.6.2-elasticsearch-gc-pipeline`
- `filebeat-8.6.2-elasticsearch-server-pipeline`
- `filebeat-8.6.2-elasticsearch-slowlog-pipeline`
- `filebeat-8.6.2-iptables-log-pipeline`
- `filebeat-8.6.2-logstash-log-pipeline`
- `filebeat-8.6.2-logstash-slowlog-pipeline`
- `filebeat-8.6.2-system-auth-pipeline`
- `filebeat-8.6.2-system-syslog-pipeline`
- `filebeat-8.6.2-elasticsearch-audit-pipeline-json`
- `filebeat-8.6.2-elasticsearch-deprecation-pipeline-json`
- `filebeat-8.6.2-elasticsearch-server-pipeline-json`
- `filebeat-8.6.2-elasticsearch-slowlog-pipeline-json`
- `filebeat-8.6.2-logstash-log-pipeline-json`

- filebeat-8.6.2-logstash-slowlog-pipeline-json
- filebeat-8.6.2-elasticsearch-audit-pipeline-plaintext
- filebeat-8.6.2-elasticsearch-deprecation-pipeline-plaintext
- filebeat-8.6.2-elasticsearch-server-pipeline-plaintext
- filebeat-8.6.2-elasticsearch-slowlog-pipeline-plaintext
- filebeat-8.6.2-logstash-log-pipeline-plaintext
- filebeat-8.6.2-logstash-slowlog-pipeline-plaintextw
- Winlogbeat-8.6.2-powershell
- winlogbeat-8.6.2-powershell_operational
- winlogbeat-8.6.2-routing
- winlogbeat-8.6.2-security
- winlogbeat-8.6.2-sysmon

5.4.5.7 Load Templates

After the cluster has been installed/upgraded and is running, the templates needed to ingest data properly must be updated. The templates are located in the **templates** folder of the **install** directory of the Elastic Repo.

In this section all index and component templates will be added to Elasticsearch. The following naming conventions are used for Enterprise Service templates in Elasticsearch.

Index templates – esti_<template name>
Component templates – estc_<template name>

IMPORTANT: DO THIS BEFORE INSTALLING ANY BEATS COLLECTORS OR ANY LOGSTASH INSTANCES.

1. Run the following command as root from any of the running Elastic nodes to update the templates.

   # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/load_templates.sh | bash

2. After loading the templates, they can be verified (sorted by name) by executing the following command from the Kibana Dev Tools console.

   ```
   GET _cat/templates/esti*?v&s=name
   ```

3. The following index templates should be loaded by this script:
   - esti_catalyst
   - esti_datadomain
   - esti_db_postgres
   - esti_eracent
   - esti_fc6xx
   - esti_filebeat-{version}
   - esti_fx2
   - esti_hbss-epo

- esti_hbss-metrics
- esti_healthdata
- esti_current-healthdata
- esti_heartbeat-{version}
- esti_idm
- esti_iptables
- esti_isilon
- esti_nexus5k
- esti_nexus7k
- esti_puppet
- esti_r6xx
- esti_sccmdb
- esti_serena
- esti_render
- esti_soaesb
- esti_sqlserver
- esti_vsphere
- esti_winlogbeat-{version}
- esti_xtreamio
- esti_acas
- esti_socketgxp
- esti_gxpxplorer
- esti_maas_logs

NOTE: esti_metricbeat-{version}-{site}, esti_audits_syslog-{site} and esti_syslog-{site} index templates are generated dynamically later in the installation process.

NOTE: There may be other index templates, but the above templates should all exist after running the load_templates script above.

NOTES:

- All Enterprise Service index templates prefixed with “esti_” and the {version} in the previously listed names will be replaced with the current version of the beat being installed.
- If the templates are not loaded, **STOP**, and contact an OADCGS Elastic SME for guidance.

You can also use the **Index Management** interface in Kibana to manage Index Templates, Component Templates, and Legacy Templates.

The index templates for site specific indexes will be loaded during each Logstash upgrade.

5.4.5.8 Load Kibana Saved Objects

NOTE: You must be root and a member of the **ent elastic admins** AD group to load saved objects into Kibana. Having the **Elastic Administrator** OneIM Role will place the user in this group.

Kibana will be configured with 3 spaces (Default, Cyber Analytics, Sandbox) during this installation. The Default space will be loaded with a set of initial visuals and dashboards along with the index patterns for all data types in this version. For the remainder of these instructions, when logging into Kibana you must select the Default space.

1. Run the following command as root from any of the running Elastic nodes to install objects; red text in the output can be ignored:

   ```
   # curl -k https://{site code}su01ro01.`hostname -
   d`/yum/elastic/install/load_objects.sh | bash -s install
   ```

2. After running the script, verify the objects are loaded. Navigate to the **Stack Management** screen. Select **Saved Objects** under the **Kibana** section.

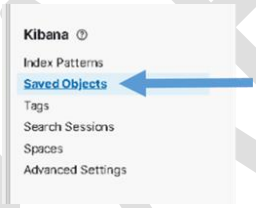


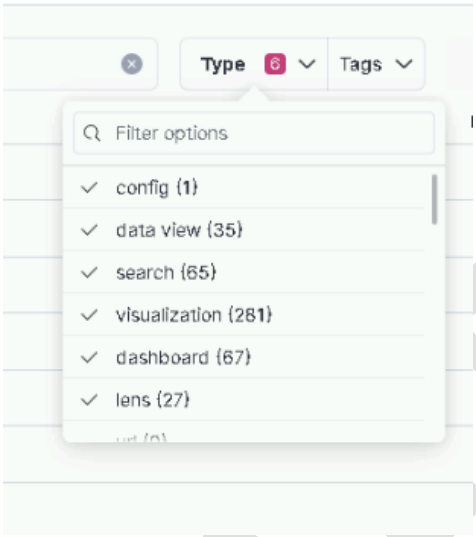
Figure 19 Select Saved Objects

3. The **Saved Objects** page displays. There should be at least 476 Objects loaded.

4. Select the **Type** drop-down, scroll down and examine each type. The following shows the minimum number you should see for each type. There may be more if additions were added that are not delivered with the baseline.

   - data view (35)
   - search (65)
   - visualization (281)
   - dashboard (67)
   - lens (27

   Example:



Note: As part of this install objects for exiting ART integrations are added. This include GXPXplorer, SOAESB, SocetGXP and MAAS data views, lens, visualization, dashboard and search objects.

There is currently no data to display in these loaded visuals. You will be setting up data ingest in the following sections.

5.4.5.9 Update Kibana Settings (On one Kibana node only)

Login into a Kibana node and run the following script to update global Kibana settings:

 # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/update_kibana_settings | bash

This will set the new security banner at the top of each page in Kibana and enable dark mode. The security banner should be appropriate for the classification of the system Kibana is running on. If the banner does not look correct, contact an Elastic SME for guidance.

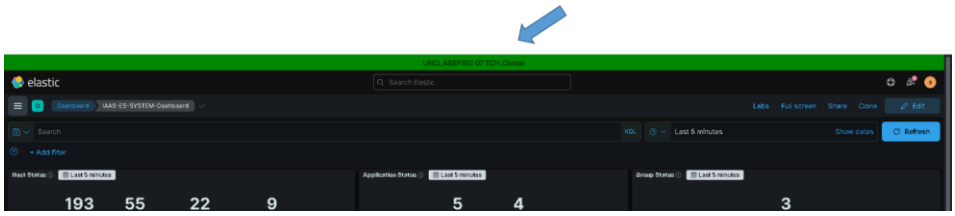


Figure 20 Example showing security banner and dark mode

5.4.5.10 Configure Index Lifecycle Management (ILM)

The Index Lifecycle Management (ILM) Policies are used to automatically manage the indices in Elasticsearch. In the current implementation each index has a **lifecycle** that starts as hot, moves to warm, and then is eventually deleted. This ensures Elastic does not become overwhelmed with data over time. In this version of Elastic, all indexes will be assigned to the same lifecycle policy, the **dcgs_default_policy**. As requirements change for indexes this may change in the future.

For ILM to be functional, the following must be in place

1. An ILM policy must be loaded into Elastic.
2. A template for each index must exist to assign it to the ILM policy.
3. Each index must be bootstrapped as the initial write index.
4. Verification that each index is moving through the lifecycle phases should be done.

5.4.5.10.1 Load DCGS Default ILM Policy

The DCGS default ILM policy controls the lifetime of all DCGS indexes. This policy must be loaded to ensure indices are managed so the amount of data in the cluster does not grow too large over time.

1. Run the following command as root from any of the running Elastic nodes to install the policy:

```
# curl -k https://xxxsu01ro01.`hostname -
d`/yum/elastic/install/load_ILMPolicy.sh | bash
```

2. Upon successful loading, the **dcgs_default_policy** appears in the list of **Index Lifecycle Policies** on the Kibana **Stack Management** page (under Data).

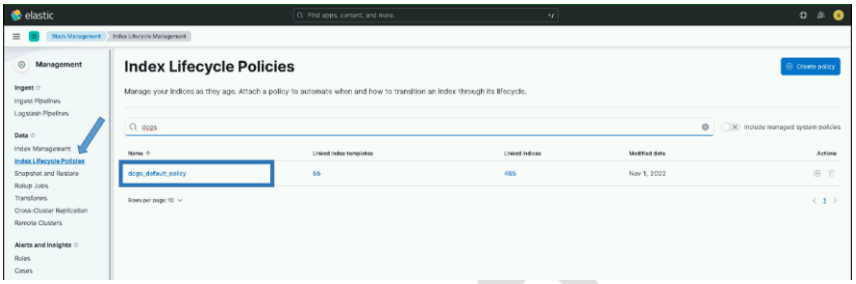


Figure 21 Index Lifecycle Policies

5.4.5.10.2 Default Component Template for ILM

This section is informational, no installation steps are executed.

Index templates can contain a collection of component templates, as well as directly specify settings, mappings, and aliases. All index templates for DCGS include the estc_dcgs_defaults component template which configures ILM with the dcgs_default_policy. The previous section loaded the dcgs_default_policy which is used by all templates to define the life cycle for data.

The estc_dcgs_defaults component template contains the following settings:

```
{
  "index": {
    "lifecycle": {
      "name": "dcgs_default_policy"
    },
    "routing": {
      "allocation": {
        "include": {
          "_tier_preference": "data_hot"
        }
      }
    },
    "mapping": {
      "total_fields": {
        "limit": "10000"
      }
    },
    "refresh_interval": "5s",
    "number_of_shards": "1",
    "query": {
      "default_field": [
        "event.original"
      ]
    },
    "number_of_routing_shards": "30",
    "number_of_replicas": "1"
  }
}
```

Figure 22- estc_dcgs_defaults settings

This assigns the **dcgs_default_policy** as the lifecycle policy and specifies that all ingested data will initially be writing to nodes in the data_hot tier.

You can retrieve the contents of this template by executing the following command in the Kibana Dev Tools console:

```
GET _component_template/estc_dcgs_defaults
```

5.4.5.11 Bootstrap Indexes

To make sure Elastic is ready to receive data from the upgraded beats and any new indexes, you need to bootstrap an initial index and designate it as the write index for the rollover alias specified in the new index templates. The name of this index must match the template's index pattern and end with a number. Each index template has a rollover_alias specified for this purpose. On rollover, this value is incremented to generate a name for the new index.

IMPORTANT: There are three site-base indexes; "metricbeat", "dcgs-syslog-iaas-ent" and "dcgs-audits_syslog-iaas-ent". This means that there will be one alias per site for these indexes. These aliases are bootstrapped during the upgrade to Logstash at each site later in the process. You will not see an alias these 3 indexes after this step is complete.

1. Run the following command as root from any of the running Elastic nodes to bootstrap the initial write indexes for the Elastic data types:

   # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/bootstrap_indexes.sh | bash

   NOTE: This script will only bootstrap indexes that do not currently have an alias configured. Running this script more than one time causes no harm.

2. To verify spot check beats indexes have bootstrapped and have a write index, execute the following command from the Kibana Dev Tools console, which sorts them by name:

   ```
   GET _cat/aliases/*beat-{version}*?v&s=is_write_alias:desc
   ```

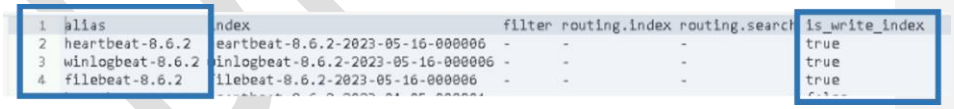
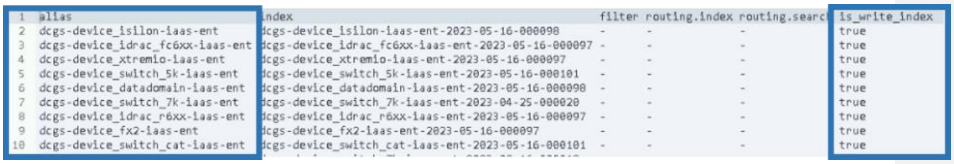


*Figure 23 GET _cat/aliases/*beat-{version}>*?v&s=is_write_alias:desc output*

3. Now spot check the device indexes

   ```
   GET _cat/aliases/dcgs-device*?v&s=is_write_alias:desc
   ```



	alias	index	filter	routing.index	routing.search	is_write_index
2	dcgs-device_isilon-iaas-ent	dcgs-device_isilon-iaas-ent-2023-05-16-000098	-	-	-	true
3	dcgs-device_idrac_fc6xx-iaas-ent	dcgs-device_idrac_fc6xx-iaas-ent-2023-05-16-000097	-	-	-	true
4	dcgs-device_xtremio-iaas-ent	dcgs-device_xtremio-iaas-ent-2023-05-16-000097	-	-	-	true
5	dcgs-device_switch_5k-iaas-ent	dcgs-device_switch_5k-iaas-ent-2023-05-16-000101	-	-	-	true
6	dcgs-device_datadomain-iaas-ent	dcgs-device_datadomain-iaas-ent-2023-05-16-000098	-	-	-	true
7	dcgs-device_switch_7k-iaas-ent	dcgs-device_switch_7k-iaas-ent-2023-04-25-000020	-	-	-	true
8	dcgs-device_idrac_r6xx-iaas-ent	dcgs-device_idrac_r6xx-iaas-ent-2023-05-16-000097	-	-	-	true
9	dcgs-device_fx2-iaas-ent	dcgs-device_fx2-iaas-ent-2023-05-16-000097	-	-	-	true
10	dcgs-device_switch_cat-iaas-ent	dcgs-device_switch_cat-iaas-ent-2023-05-16-000101	-	-	-	true

Figure 24-GET _cat/aliases/dcgs-device?v&s=is_write_alias:desc*

4. All data types have been bootstrapped successfully. If there are no aliases listed for the version you are installing, or none have the **is_write_index** set to **true**, consult with an OADCGS Elastic SME for guidance.

5.4.5.12 Setup Snapshot Lifecycle Management Policies

NOTE: You must be root and a member of the **ent elastic admins** AD group to load saved objects into Kibana. Having the **Elastic Administrator** OneIM Role will place the user in this group.

Snapshot Lifecycle Management is used to automatically archive selected indexes for long term storage.

The following indexes have been identified as having data that needs to be retained long term and are included in this versions SLM Policy configuration.

- dcgs-syslog
- dcgs-audits_syslog
- winlogbeat
- dcgs-hbss_epo
- dcgs-db*
- dcgs-vsphere
- .siem-signals

1. Login to any Elasticsearch Node
   # sudo su

2. Run load SLM Policy script

```
# curl -k https://{site code}su01ro01.`hostname -
d`/yum/elastic/install/load_SLM_Policy.sh | bash
```

5.4.5.13 Adjust Concurrent Incoming/Outgoing Recoveries (optional)

How many concurrent outgoing shard recoveries are allowed to happen on a node? Outgoing recoveries are the recoveries where the source shard (most likely the primary unless a shard is relocating) is allocated on the node. The default is 2. Incoming recoveries are the recoveries where the target shard

(most likely the replica unless a shard is relocating) is allocated on the node. The default is 2. We are setting this value to 20 to allow faster cluster startups.

```
PUT _cluster/settings {
    "persistent" : {
       "cluster.routing.node_concurrent_recoveries" : 20
   }
}
```

5.4.5.14 Memory Lock Check

When the JVM does a major garbage collection it touches every page of the heap. If any of those pages are swapped out to disk they will have to be swapped back into memory. That causes lots of disk thrashing that Elasticsearch would much rather use to service requests.

There are several ways to configure a system to disallow swapping. One way is by requesting the JVM lock the heap in memory through **mlockall** (Unix) or **virtual lock** (Windows). This is done via the Elasticsearch setting `bootstrap.memory_lock`. However, there are cases where this setting can be passed to Elasticsearch, but Elasticsearch is not able to lock the heap (e.g., if the Elasticsearch user does not have **memlock unlimited**). The memory lock check verifies that if the **bootstrap.memory_lock** setting is enabled, the JVM was successfully able to lock the heap. To pass the memory lock check, you might have to configure `bootstrap.memory_lock`.

NOTE: If swapping is not enabled on the machine, memory locking is not needed. If swapping is turned on you can check to see if Elastic was able to prevent memory from being swapped by checking the value of mlockall on each host.

```
GET _nodes?filter_path=**.mlockall
```

Remember, if this returns false, things still may be okay if swapping on the system is disabled. See https://www.elastic.co/guide/en/elasticsearch/reference/6.5/setup-configuration-memory.html for more information.

5.4.5.15 Centralized Pipelines

The centralized pipeline management feature for Logstash provides management and automated orchestration of the Logstash deployments. All Logstash ingest pipelines are now stored in the Elasticsearch cluster and managed by Kibana. Before any of the Logstash instances on DCGS can run the Logstash pipelines must be loaded and available in Elasticsearch. Pipelines can be viewed/edited from the Pipelines management UI provided in Kibana. Centralizing the pipelines allows the management of all ingest feeds from Kibana. The pipelines loaded into Elasticsearch are available for any Logstash instance to use. The Logstash configuration file **logstash.yml** on each Logstash instance dictates which pipelines are used by that instance. The logstasy.yml file for each Logstash instance is controlled by the dsil_elastic_servers puppet module. Configuration of the Logstash instances to use the pipelines is covered in the Logstash section of this document. This section is dedicated to the loading of pipelines into Kibana/Elasticsearch to make them available for use.

5.4.5.15.1 Load Enterprise Services Centralized Pipelines

The deployment of Elasticsearch as a service includes the collection of multiple datatypes. The ingest pipelines for these datatypes must be loaded before configuring any Logstash instances to ingest data. Perform the following to load the Enterprise Services ingest pipelines.

NOTE: The load_pipelines script communicates with Kibana using the kibana alias, which is set up to route traffic to the NSX load balancer. If you cannot connect to Kibana by typing https://kibana **in your browser, revisit the NSX configuration instructions before proceeding with the installation of Centralized Pipelines. If you cannot configure NSX, consult with an OADCGS Elastic SME for guidance.**

```
# sudo su

# curl -k https://{site code}su01ro01.`hostname -
d`/yum/elastic/install/update_logstash_pipelines.sh | bash
```

After running the script, verify the pipelines have been loaded. Once the pipelines are loaded, the Logstash instances can be configured to use them.

NOTE: The pipelines only need to be loaded one time, not once for each Logstash instance.

To verify the pipelines have been loaded, go to **Stack Management in Kibana** and look in the **Pipelines** section.

1. From the hamburger menu, select **Stack Management**.
2. Select **Logstash Pipelines** in the **Ingest** area.
3. The **Pipelines** page displays. (Below is an example of the page)

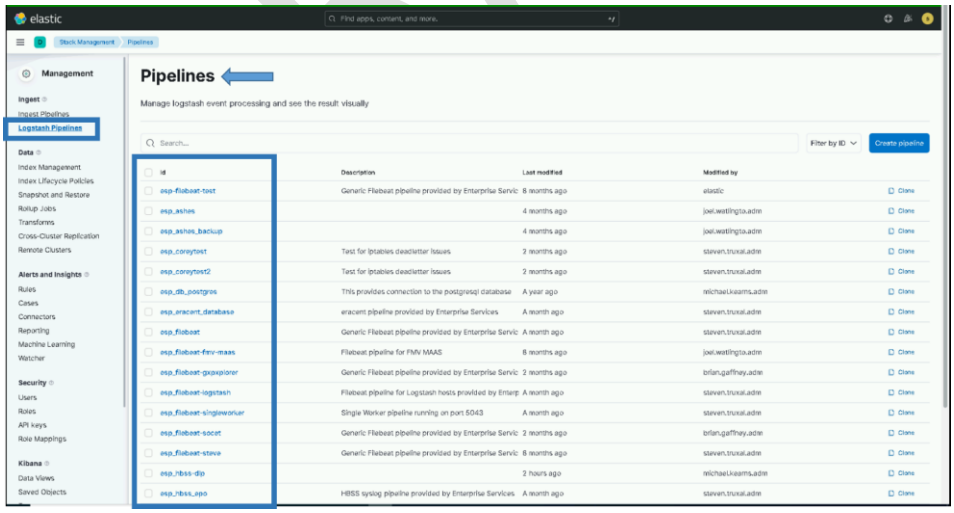


Figure 25 Example of Pipelines page in Kibana

IMPORTANT: Loading the pipelines into Elastic makes them available for use by any Logstash Instance but does not automatically add them to any Logstash configuration files. When installing Logstash it is important to verify/configure what pipelines are active on each Logstash Instance. Logstash pipeline configurations are controlled by puppet, see section 5.4.1.8.2 for details.

5.4.5.16 Install health data watcher

This version brings the ability to detect when a host's status has not been updated for a period of time. When this happens the hosts, status will become "Stale". The "esw_current-healthdata-stale-state" watcher is used to monitor the update times for host data in the "dcgs-current-healthdata-iaas-ent" index.

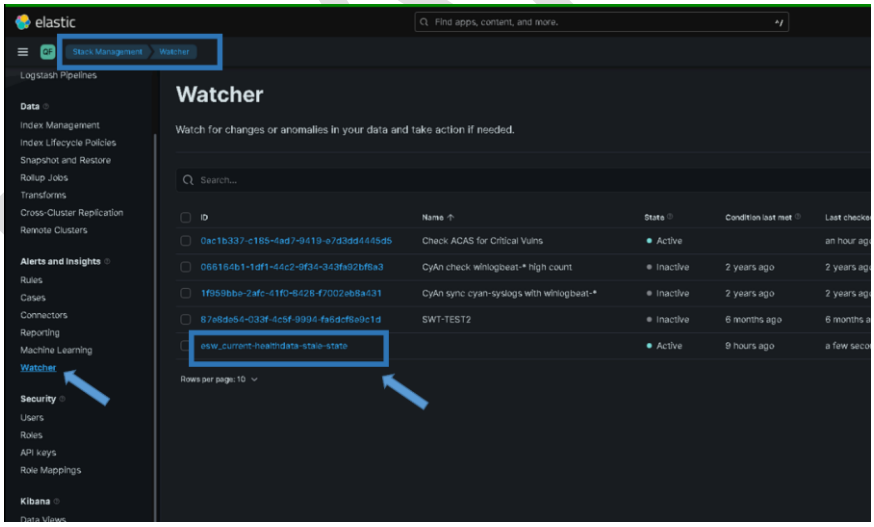
Follow the instructions below to install the watcher:

1. Run the following command as root from any of the running Elastic nodes install the watcher.

   # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/installWatchers.sh | bash

2. Verify the watcher was loaded correctly.

   - From the hamburger menu, select **Stack Management**.
   - Select **Watcher** in the "Alerts and Insights" section.
   - Validate the "esw_current-healthdata-stale-state" watcher is listed.



5.4.6 Logstash

Logstash ingests, enhances, and ships all data to the Elastic Cluster. There is currently one Logstash VM at every site. All data from Beats and other collectors at each site is collected by the Logstash for the site. The **DCGS_Site** field in each document received is set to the site the data is coming from and then the data is sent from that Logstash instance into the Elastic cluster.

The following users/roles are used by each Logstash instance to interact with the Elastic cluster:

- Users:
  - logstash_system – Used for monitoring the Logstash instance
  - logstash_admin_user – Used to manage centralized pipelines
  - logstash_internal – Used by Logstash pipelines to write data into Elasticsearch
- Roles:
  - logstash_writer – Used by Logstash to create/write indexes

The required users/roles for Logstash are automatically set up during the installation of the first Logstash instance for the cluster. The passwords for the users are dynamically generated and set in Elastic, and stored in the **logstash.keystore** on the initial Logstash instance. These passwords are never seen during creation and are used to deploy all additional instances of Logstash throughout the system. After the completion of the installation of the initial Logstash instance, the **/etc/logstash/logstash.keystore** file **MUST** be copied to the repo server and placed in the installation directory of the Elastic Repo (elastic/install). The logstash.keystore **MUST** be present for additional Logstash installations or the install will fail.

IMPORTANT: The logstash.keystore must be placed on the repo server to be available for each site.

NOTE: You must be root to install Logstash.

5.4.6.1.1 Install Logstash on the Dedicated Logstash VM

1. Log in to the Logstash VM for each site and become root. This is done on the xxxsu01ls01 VM.

   ```
   # sudo su
   ```

2. Verify iptables are set up correctly. This should be handled by Puppet. All Elastic nodes should be included in the **Elastic Servers** Classification on the Puppet server.

   ```
   # iptables --list -n
   ```

3. **Verify ports 5040-5060, and 9200 are open.** If these ports are not open, stop and verify hosts are set up correctly in Puppet.
4. Install Logstash:

   ```
   # curl -k https://{site code}su01ro01.`hostname -d
   `/yum/elastic/install/installLogstash.sh | bash
   ```

   NOTE: This command cannot be run multiple times. If running fails, you must delete the logstash* users/roles in Kibana to run again.

5. If this is the first Logstash instance, copy the **/etc/logstash/logstash.keystore** file to the repo server in the installation directory of the Elastic Repo (elastic/install).

5.4.6.1.2 Configure Logstash

Logstash configurations are managed using centralized pipeline management. The Logstash configuration file **logstash.yml** is set up to allow this. The initial installation will place the default pipelines in the configuration file. This may need to be updated depending on the data types that each Logstash instance will ingest.

IMPORTANT: The logstash.yml file and the configuration for which pipelines to run on each logstash box is managed by puppet. See Section 5.4.1.8.2 for details on setting up pipelines to run on each Logstash instance.

5.4.6.1.3 Start & Verify Logstash

1. To start Logstash:

   ```
   # systemctl start logstash
   ```

2. Wait for logstash to start; it may take a few minutes. You can monitor the progress by tailing the **/var/log/logstash/logstash-plain.log**.

5.4.6.1.4 Install Data Collector

To collect data from Infrastructure devices on DCGS, the elasticDataCollector must be installed and configured at every site on each Logstash instance.

NOTE: You must be root and a member of the **ent elastic admins** AD group to install the Elastic Data Collector. Having the **Elastic Administrator** OneIM Role will place the user in this group.

1. Log in to the **Logstash VM** for each site and become root. This is done on the *{site code}*su01ls01 VM.

   ```
   # sudo su
   ```

2. Install the Elastic Data Collector.

   ```
   # curl -k https://{site code}su01ro01.`hostname -
   d`/yum/elastic/install/installElasticDataCollector.sh | bash
   ```

   NOTE: The script will prompt for your password as your user account will be used to communicate with the Elasticsearch cluster. You will also be asked to enter the site of the Elastic cluster; this defaults to ech but may be different on test enclaves.

   If you are unsure of the cluster site, you can test access to the cluster by using ping:

   ```
   # ping elastic-node-1.{site}
   ```

5.4.6.1.5 Update Groups of Servers to Monitor

During the installation, if not already present, a file named groups.ini was placed in the /ELK-local/elasticDataCollector directory. This is the configuration file used to set up groups of hosts to monitor. To add groups, follow the format defined in the header of the file.

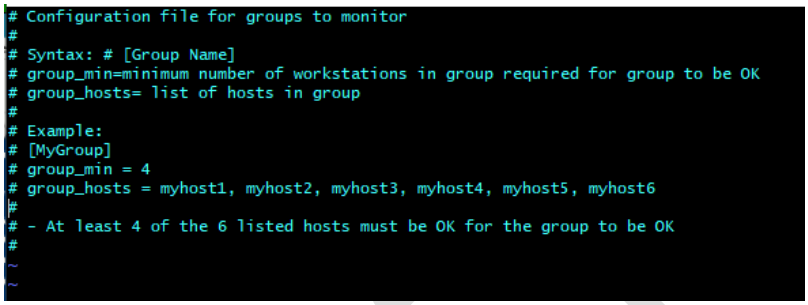


Figure 26 Configuration File

NOTE: The elasticDataCollector service must be restarted after updating this file.

```
# systemctl restart elasticDataCollector
```

NOTE: If this file already exists it is not overwritten during the upgrade.

5.4.6.1.6 Set Up Devices to Be Monitored

After installing the data collector, if not done on a previous install, a configuration file must be created that contains the devices to monitor for each site. A configurator tool is delivered as part of this upgrade to set up the devices to monitor. The tool was installed with the data collector. Use the tool to configure the devices to monitor for each site.

NOTE: The devices to monitor may be updated any time device information changes. During upgrades is a good time to verify that each site is configured to monitor the correct devices. The configurator can be used to verify the monitoring configuration of a site. Making updates is not required if the current monitoring configuration is satisfactory.

Before proceeding with this section, you will need to gather the following information for all infrastructure devices that will be monitored for each site. You must create a configuration file for each site (Logstash instance) where you want to monitor devices.

Device Information needed to monitor XtremIO and Isilon devices:

- URL – This is the URL to access the device's web interface.
- Username – Username to access the device via REST API.
- Password – Password for Username.
- Display Name – Short unique description of device (ex: ech-isilon, ech-xtremio).
- Host – DNS resolvable name or IP address of host (ex: u00av01xms1).

Device Information needed to monitor Cisco switches, fx2 chassis, fc6xx blades, r6xx servers and Data Domain storage devices:

- URL – This is the URL to access the device's web interface (CMC, IDRAC, Cisco Prime).
- Username – Username to access the device via SNMP.
- Password – Authentication Password for Username.
- Priv Password – Privacy Password for Username.
- Display Name – Short unique description of device (ex: ech-fx2, ech-5k).
- Host – DNS resolvable name or IP address of host (ex: u00av01xms1).

5.4.6.1.6.1 Verify X11-Forwarding Enabled

To use this tool you must be able to open a window from the Logstash box. The easiest way to do this is by using mobaXterm which has an embedded X Server. When the data collector was installed, it set up the SSH daemon to allow X11 connections. Puppet was also disabled on the Logstash host to allow this change to remain during device configuration. To have the change automatically reverted, Puppet must be re-enabled upon completion of section 5.5.5.2.3.2. To verify X11 is set up correctly, do the following:

1. Connect to the Logstash VM with a new mobaXterm session.
2. Verify X11-forwarding is enabled. There will be a green check box next to **X11-forwarding**, as shown in the following figure.

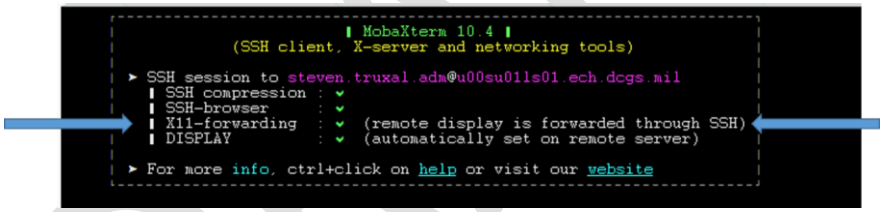


Figure 27 X11 Forwarding Setup Successfully

NOTE: When Puppet is re-enabled, X11-forwarding will be automatically disabled. To run the configurator if this occurs you can manually enable X11-forwarding by executing the following steps.

1. Execute the following command:
   `# sed -i 's/X11Forwarding no/X11Forwarding yes/g' /etc/ssh/sshd_config`
2. `# systemctl restart sshd`
3. Retry the previous test to verify X11 forwarding is set up correctly.

NOTE: if the steps above are still not getting X11-fowarding enabled one last thing to try is to ensure the AdressFamily setting is correct.

1. # cd /etc/ssh
2. # vi sshd_config
3. look for the line setting for "AddressFamily". If it is comment out, then uncomment and verify it is set to "inet"

Should be: AddressFamily inet

4. Save changes (:wq)
5. # systemctl restart sshd
6. Retry the previous test to verify X11 forwarding is setup correctly.

IMPORTANT: If you cannot get X11-forwarding enabled, **STOP**, and consult a Linux SME for help.

5.4.6.1.6.2 Create Collector Configuration

NOTE: The previous step must be successful to continue. The GUI will not display if X11-forwarding is not enabled on the Logstash VM.

1. Log in to the Logstash VM with your .adm account and list the xauth cookies.

```
xauth list
```

You may see multiple cookies if X11-forwarding is enabled on other hosts. Take note of the cookie for this host; you will see the ls01 host name at the beginning.

NOTE: If there are multiple entries for the ls01 host the last entry is most likely the one you will use. To make sure, execute **echo $DISPLAY** to get the correct number of the display for your SSH session. Use the cookie line that has both **ls01** and the display number.

Figure 28 xauth list example with logstash host

2. Sudo to become root.
   ```
   # sudo su
   ```
3. Copy the xauth cookie and add it to the roots xauth cookies.
   ```
   # xauth add <cookie>
   ```
   Example:

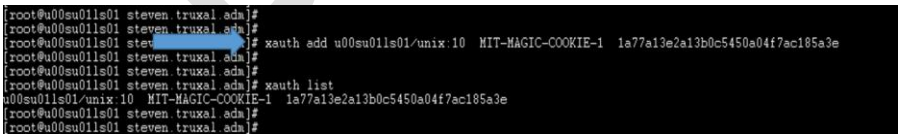


Figure 29 xauth add <cookie>

4. Run the configurator GUI.
   ```
   # cd /etc/logstash/scripts
   # . ./venv/bin/activate
   # python ./configurator.py
   ```

5. The device configuration window, **The Configurator**, displays.
   NOTE: If the window does not come to the foreground, look for an icon in the taskbar.

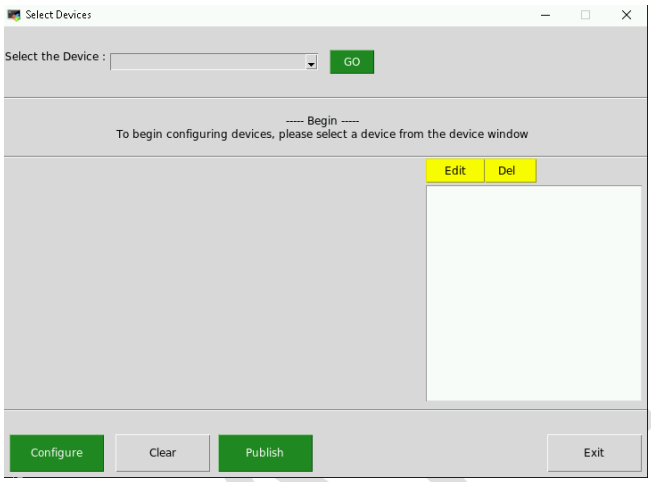


Figure 30 Device Configuration GUI

   NOTE: If the data collector was already installed on a previous version the devices that are currently set up for monitoring will be displayed. If there are no updates to be made, verify that all devices to be monitored are listed and exit. If this is the initial configuration or edits need to be made, continue with this section.

6. Configure devices one at a time until you have added all devices to monitor for the site.

7. Select a device from the **Select the Device** menu and click **Go**.
   NOTE: You must select the device type again for every new device.

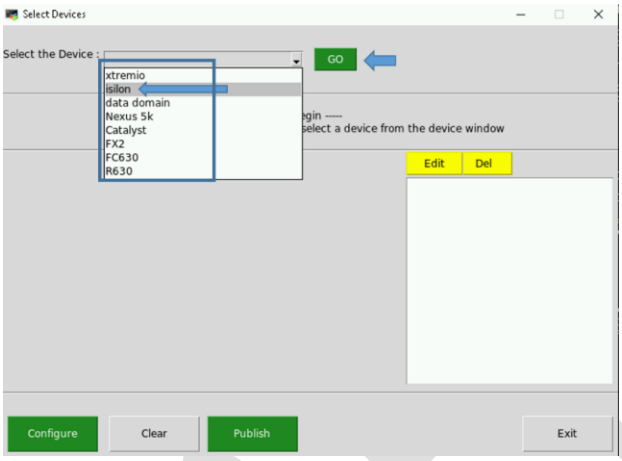


Figure 31 Select Device to Configure

8. Fill in the fields with the required information to monitor the device. Click **Configure**.

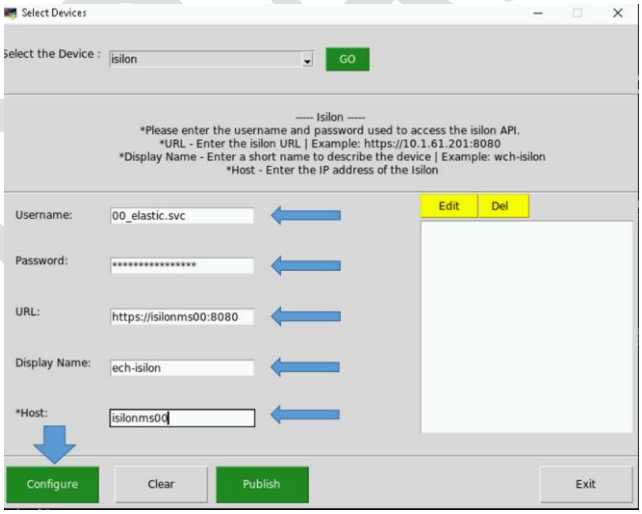


Figure 32 Configure Device

9. The list on the right will populate.

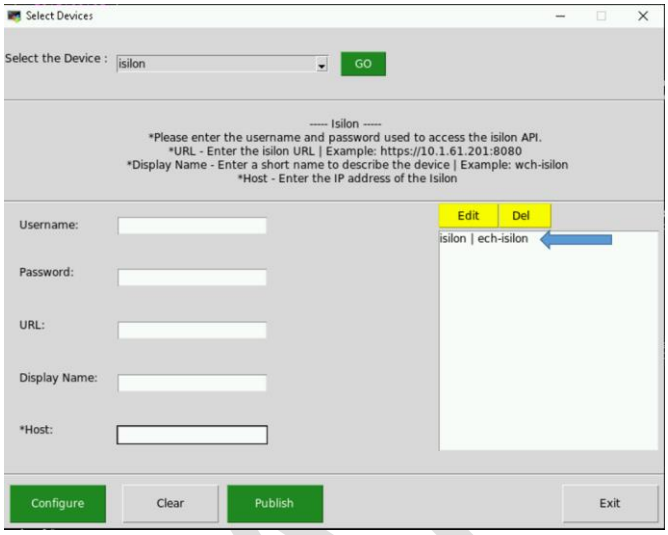


Figure 33 Device Added to List

10. Enter another device or press **Publish** to test your configuration.

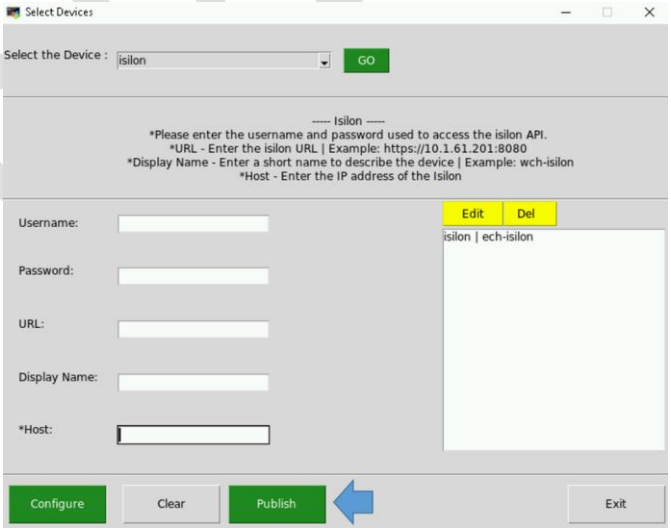


Figure 34 Publish Device Configuration

11. The Configurator will verify the values you have entered by trying to access the device(s). If all goes well your configuration will be updated and a success message will be displayed.

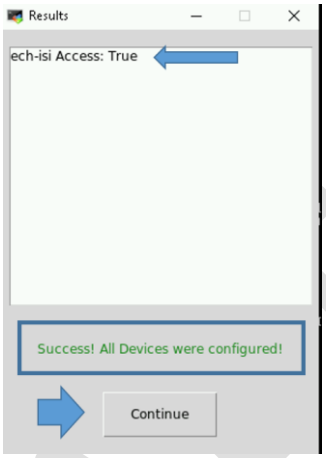


Figure 35 Successful Publish

a. If any of your access parameters are incorrect the **Publish** will fail, and the configuration file will not be updated.

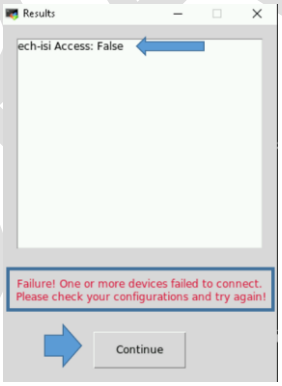


Figure 36 Unsuccessful Publish

12. Select the **Continue** button in the **Results** window to continue. If you entered information incorrectly and the **Publish** failed, you can either **Edit** or **Del** the device that failed.

NOTES:

- If any of the devices fail, the configuration file is NOT updated. Access to all devices must be successful to create/update a device configuration file.
- Every successful publish will restart the elasticDataCollector service so the current configuration is read in and collection from the updated devices is started.

13. If the **Publish** failed, you can use the **Edit** feature to correct the issue. For security reasons the password is not shown, so if everything else looks correct try re-entering the password. To review this feature, select the device you'd like to modify and press the **Edit** button.

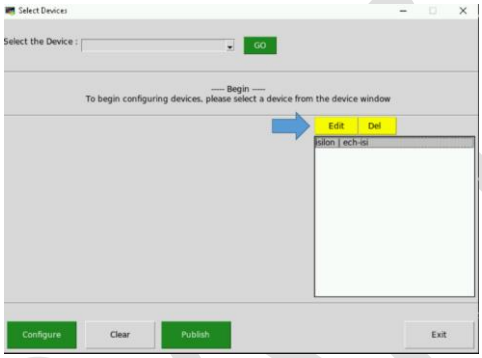


Figure 37 Select Edit to modify device information

14. The **Edit Item** dialog appears. The information that was entered for the device you selected is displayed. You can make changes to any of the items and select **Save** or go back to the main screen without making any modifications by selecting **Cancel**.

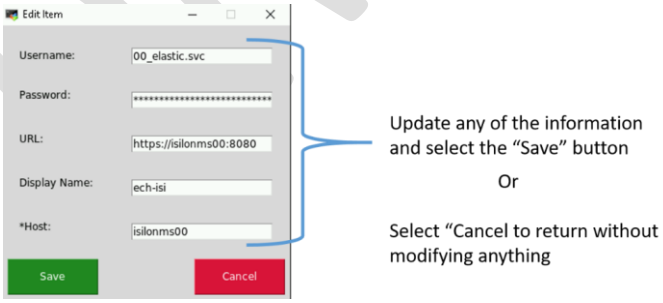


Figure 38 Edit Item

15. Once back at the main screen you can select **Publish** again, verify your device configurations, or continue entering devices by selecting a new device from the **Select the Device** menu.

16. Once you have configured all your devices and the **Publish** is successful, you can exit the Configurator. Your device configuration is automatically saved on each successful Publish.
17. Re-enable Puppet on the Logstash host by executing the following command:
    ```
    # puppet agent –enable
    ```

    IMPORTANT: Don't skip this step.

NOTES:

- You can run the configurator GUI at any time to either view, modify, or add devices for this Logstash instance to monitor.
- Every successful publish will restart the elasticDataCollector service so the current configuration is read in and collection from the devices is started.

5.4.6.1.7 Verify Device Data is Being Collected

Once you have a good configuration in place and the elasticDataCollector service is running we can verify that Elastic is receiving data for the devices. The easiest way to do this is viewing the **IAAS-ES-Infrastructure Overall Status Dashboard**. This dashboard shows the overall status of each of the devices you have configured.

1. Select the **Dashboard** option from the hamburger menu.

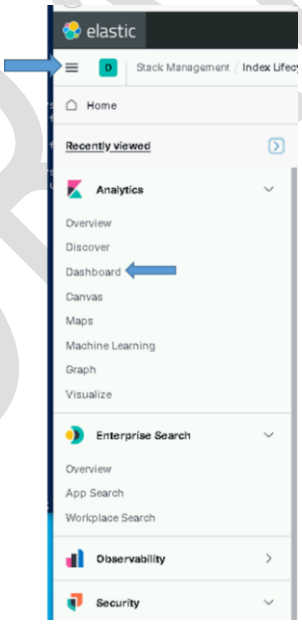


Figure 39 Select Dashboards Option

2. Type **IAAS-ES-Infrastructure Overall Status Dashboard** in the search bar.

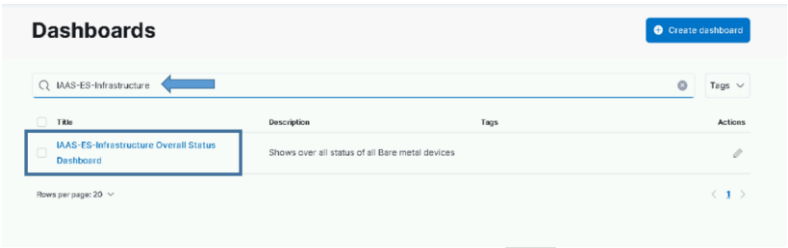


Figure 40 Select IAAS-ES-Infrastructure Status Dashboard

3. Verify that the devices you entered into the configuration are listed in the dashboard.

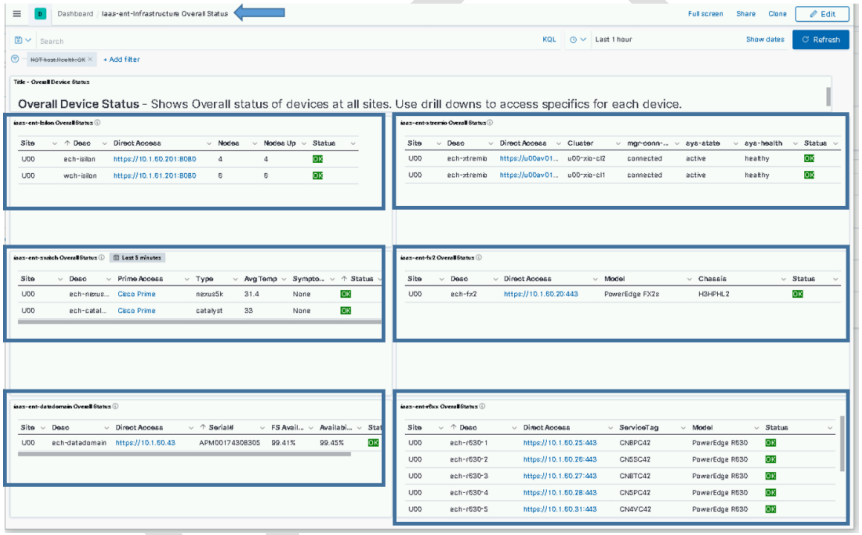


Figure 41 Devices Listed in Dashboard

There should be 6 areas on the dashboard:

- Isilon
- Switches
- Data Domain
- XtremIO
- Fx2
- Rc6xx

Important: Repeat sections in 5.4.6 for each Logstash instance.

5.4.7 Secure Elastic with Break-Glass Password

This section is to be completed after the successful install and checkout of Elasticsearch, Kibana, and all Logstash instances on the entire system.

Now that Elasticsearch has been successfully installed and integrated into Active Directory, all access to Elastic should be done using DCGS accounts. There are two Active Directory groups that are used for cluster administration.

- **ent elastic admins** – Members of this group will oversee installations/upgrades and all configuration aspects of Elasticsearch and its components. This group should be limited to a very small group of people as it gives all privileges in Elastic.
- **ent kibana admins** – Members of this group will oversee day-to-day operations with Elastic, which includes but is not limited to:
    - Ensuring the cluster is running correctly
    - Recovering from any cluster or ingest issues
    - Creating visuals/dashboards

5.4.8 Verify Role Mappings

Prior to checking Active Directory access, check the Kibana Role Mapping (which map AD groups to Kibana Roles):

1. Log into Kibana using the Elastic account.
2. Go to **Stack Management** and under **Security**, select **Role Mappings**.
3. Select the name of one of the Mappings and open the switch to JSON Editor link at the bottom.
4. Verify the string(s) contain(s) a valid AD group (it is possible that the string content got cut off, so the end may be missing).
5. Save, if necessary.
6. Repeat for all Mappings.

5.4.9 Verify Roles

As the installer of Elasticsearch, you should be a member of the **ent elastic admins** group. Verify that you are a member of this group and use your AD account to log in to Kibana.

NOTE: If you are not a member of **ent elastic admins**, you must find someone who is to verify they are able to log in to Kibana.

After logging into Kibana as an **ent elastic admin**, verify your privileges to ensure the role mappings have been created successfully. Execute the following from the Kibana console:

```
GET _security/user/_privileges
```

Verify the privileges contain the following (remember this is for a member of the **ent elastic admins** group):

- `"cluster" : [ "all" ]`
- `"indices" : [{"names" : ["*"], "privileges" : ["all"]}`
- `"applications": [{"application" : "*","privileges" : ["*" ],"resources" : ["*"]}`

If everything looks good, remove unneeded accounts.

> **NOTE: Do not** remove the following accounts, the **kibana_xx**, **logstash_admin_user**, or **logstash_internal** accounts.

5.4.10 Remove Unneeded Accounts

When Elastic was initially set up in section 5.4.3.7 Start & Test Elasticsearch, passwords were set for the following accounts:

- elastic
- apm_system
- kibana_system
- logstash_system
- beats_system
- remote_monitoring_user

During the installation of Logstash, the password for the **logstash_system** was modified but the rest have not been changed. These user accounts are **Reserved** accounts and cannot be deleted. To protect access to Elastic you must change the passwords to each of these accounts, record the passwords, and store them in a safe as **break-glass** passwords. As indicated previously, access to Elastic should now be accomplished using Active Directory accounts. As an extra measure, we will also disable the accounts that are not needed.

1. Log in to Kibana using your Active Directory account (member of **ent elastic admins**).
2. Navigate to **Management** > **Stack Management** on the side navigation menu.
3. Select **Users** under the **Security** section.
4. Set Break-Glass passwords for each of the following accounts:
   - elastic
   - apm_system
   - kibana
   - kibana_system
   - beats_system
   - remote_monitoring_user

   NOTE: Do not change the **Logstash_system** account password.

5. Select the account and change the password using the **Edit User** interface.

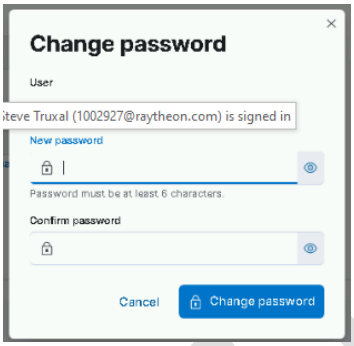


Figure 42 Change password

6. Now that the passwords have been changed, disable the accounts we don't use for extra protection. From the Kibana console execute the following commands:

```
PUT _security/user/apm_system/_disable
PUT _security/user/kibana/_disable
PUT _security/user/kibana_system/_disable
PUT _security/user/beats_system/_disable
PUT _security/user/remote_monitoring_user/_disable
```

NOTE: Do not disable the **elastic** accounts as this will be the only way to log in to Elastic if you have issues with Active Directory authentication.

7. These user accounts will now show **Disabled** on the **Users** page in Kibana.

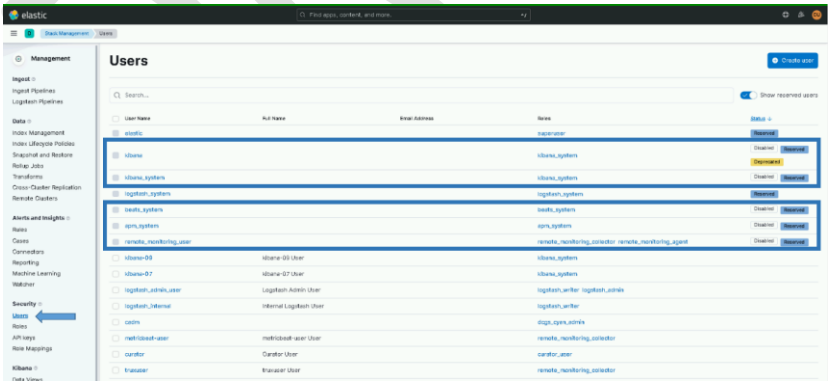


Figure 43 Disabled accounts

5.4.11 Install Beats

Beats are the agents that sit on all machines to collect and send data to Elastic. Beats agents are installed automatically on both windows and Linux systems. Puppet is used on Linux to distribute Beats agents and SCCM is used on Windows.

The following types of Beats are used at this time:

Filebeat: A lightweight data shipper for forwarding and centralizing log data. Installed as an agent on servers where log collection is desired, Filebeat monitors the log files at locations specified, collects log events, and forwards them to Logstash. Filebeat is automatically installed and upgraded on hosts that are designated as needing Filebeat for data collection.

For Linux hosts, Puppet is used for the automated install. See section **Error! Reference source not found.** for more details.

For windows, SCCM is used. Filebeat will automatically be installed on all windows boxes that have configuration files in place on the SCCM installation folder. ARTs and other entities that need Filebeat installed on hosts should work with the Elastic team to have the configurations for each host integrated into the OADCGS Elastic configuration.

Heartbeat: A lightweight daemon that you install on a remote server to periodically check the status of your services to determine whether they are available. Heartbeat is used to determine if your servers/services are reachable. One instance of Heartbeat is installed at each site on the Logstash instance for that site.

Metricbeat: A lightweight data shipper installed on each host to periodically collect metrics from the operating system and from services running on the server. Metricbeat takes the metrics and statistics that it collects and ships them to Logstash.

Winlogbeat: A lightweight data shipper installed on the Windows Event Collector of each site. Winlogbeat forwards all windows events received by the Event Collector to Logstash for ingestion into Elastic.

The following table outlines where each type of beat may be installed.

Table 10 Beat Installation Location

Beat	Installation Method	Destination Servers
Metricbeat	Puppet/SCCM	All Servers
Filebeat	Puppet/SCCM	Designated Servers
Winlogbeat	SCCM	ec01 at each site
Heartbeat	Puppet	Logstash at each site

The templates for each beat MUST be loaded into Elasticsearch before allowing the automatic install/upgrade to take place. If the templates are not loaded prior to the start of data ingest, indexes with improper data mappings may be created causing some baseline visuals to work incorrectly. Before updating the OA Repo server or SCCM with the beat rpm/zip files for installation, follow the steps in

section 5.4.5.7 to load all the templates needed for ingest. Do not place the new Beats on the repo server before upgrading all Elastic nodes, Kibana, and Logstash instances.

5.4.11.1 Verify Beat Templates are Loaded

Verify all beats templates were loaded by running the following command from the Kibana console:

```
GET _cat/templates/esti*?v&s=name
```

This will list the templates, sorted by **name**. Verify the templates loaded for the following Beats:

- est_filebeat-{version}
- est_hearbeat-{version}
- est_metricbeat-{version}
- est_winlogbeat-{version}

5.4.11.2 SCCM Configuration

NOTE: An SCCM administrator is required to execute this section.

SCCM is used to deploy all beat collectors for Windows systems. Currently SCCM deploys Winlogbeat, Metricbeat, and Filebeat on OA DCGS systems.

IMPORTANT: Before creating the SCCM installation package, ensure that the following files are in the shareDir that is used to create the package.

1. The zip files for the beats to be installed should be present in the **zipfiles** folder (see section 4.3).
    - filebeat-x.x.x.windows-x86_64.zip
    - metricbeat-x.x.x.windows-x86_64.zip
    - winlogbeat-x.x.x.windows-x86_64.zip
2. The **cachain.pem** file that will be used for the package contains the ROOT and SUB certificates for the system you are installing on. The cachain.pem file that is shipped is empty.

To configure SCCM for Elasticsearch, refer to *ES-018 – SCCM – Instructions for Building an SCCM Package to Install Beats for Windows*.

NOTE: SCCM updates for Beats deployments should always occur after installs/upgrades of the Elastic core components.

5.4.11.3 ART Integration on Windows

NOTE: This section is informational and will not be executed during the initial installation.

When ARTs or other clients request that Filebeat be installed on their VM to collect logs, the configuration file specific to their application should be placed in the **configs/filebeat** directory of the SCCM share for Elastic. The config file should be named using the hostname minus the first 7 characters of the host where the Filebeat is to be installed. Filebeat modules can also be specified with the same naming convention.

For example, if a client has a filebeat.yml configuration file and an activemq module file for the host u00sm01cl01, the following files would be placed in the **configs/filebeat** directory:

- cl01.filebeat.yml (This is the filebeat.yml for the host.)
- cl01.module.activemq.yml (This is the activemq.yml file for the host.)

IMPORTANT NOTE: The output section of the configuration **output.logstash** in the filebeat.yml is automatically generated by the installation script and should not be included in the configuration file placed in the SCCM share.

After placing these files in the directory, SCCM will automatically install and configure Filebeat on the **cl01** host.

Consult an OA DCGS Elastic SME for more information on client configurations.

5.4.11.4 Puppet Configuration to Install Linux Hosts

NOTE: A Puppet administrator is required to execute this section.

As mentioned previously, Puppet is used to automatically install Metricbeat and Filebeat on Linux hosts. Follow these steps to ensure Puppet is configured properly for installation.

1. Copy the following RPMs to the Elastic repo (/var/www/html/yum/elastic) (see section 4.3)
   - filebeat-X.X.X-x86_64.rpm
   - heartbeat-X.X.X-x86_64.rpm
   - metricbeat-X.X.X-x86_64.rpm
2. Ensure RPMs have the correct owner/group:

   ```
   # chown -R apache:apache *
   ```

3. Repo files must have SELinux context **httpd_sys_content_t** set. If you copy the RPMs into the directory, they will automatically have this context set. If you move them, they won't. Check to ensure all files have the correct context set by executing:

   ```
   # ls -lZ
   ```

```
[root@u00su01ro01 elastic]# ls -lZ
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elastic_cachain.pem
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elastic.key
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 elasticsearch-7.9.1-x86_64.rpm
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 filebeat-7.9.1-x86_64.rpm
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 heartbeat-7.9.1-x86_64.rpm
drwxr-xr-x. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 install
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 inst.zip
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 kibana-7.9.1-x86_64.rpm
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 logstash-7.9.1.rpm
-rw-r--r--. apache apache unconfined_u:object_r:httpd_sys_content_t:s0 metricbeat-7.9.1-x86_64.rpm
drwxr-xr-x. root   root   unconfined_u:object_r:httpd_sys_content_t:s0 repodata
[root@u00su01ro01 elastic]#
```

Figure 44 httpd_sys_context_t

4. If all files do not have **httpd_sys_context_t** set, execute the following:

   ```
   # restorecon *
   ```

5. Recreate the Elastic repo so it's ready for use:

```
# createrepo ./
# gpg --detach-sign --armor ./repodata/repomd.xml
```

   NOTE: There are 2 dashes in front of **detach-sign** and **armor** in the previous command.

6. Confirm overwriting the file (if it already exists).
   Enter **y** to overwrite.

Once the RPMs are in place and the Puppet module is configured correctly, Beats will begin to appear on hosts that are running Puppet.

5.4.11.5 Configure Metricbeat monitoring for Elastic Cluster

NOTE: An Elasticsearch administrator will be needed to execute this section.

Metricbeat running on each Elastic node is also used to collect monitoring data about the cluster. To this point the Stack Monitoring area in Elastic will not show any data because Metricbeat has not been configured to collect monitoring data. In the previous section we enabled the automatic installation of Metricbeat onto all Linux hosts. When puppet runs on the elastic nodes it will install Metricbeat but it will not be able to run correctly because it is missing the user and password information needed by the Elasticsearch module that is automatically configured to collect cluster monitoring data. This section is to resolve this issue as Metricbeat must be installed before we can setup it's keystore.

5.4.11.5.1 Create Metricbeat user and keystore

Ensure the Elasticsearch Cluster is "Green" before continuing.

1. Log into elastic-node-1 and become root.
   ```
   # sudo su
   ```
2. Validate that the cluster is green before continuing.
   # curl -k -u <your user id> https://elastic-node-1:9200/_cluster/health?pretty

   Example: *curl -k -u joe.smith.adm https://elastic-node-1:9200/_cluster/health?pretty*
3. Validate the response show the cluster status as "green". See example below.

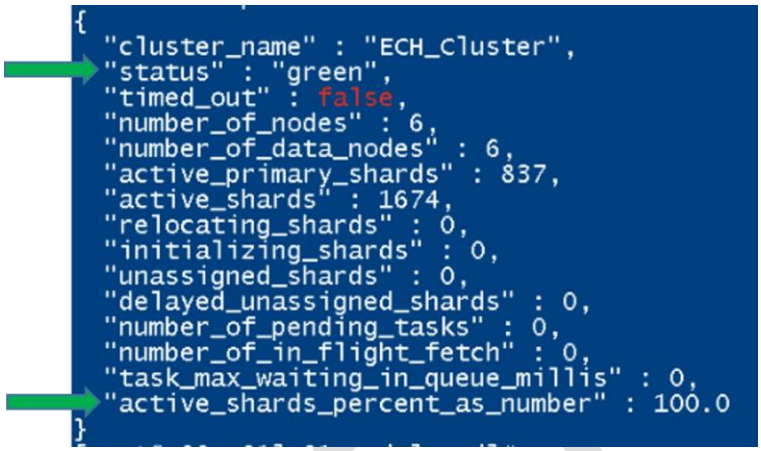


Figure 45- _cluster/health reponse example

If the cluster is not "green" stop and correct before continuing. Contact an Elastic SME if needed.isn

4. Ensure Metricbeat has already been installed on the host
   # rpm -qa | grep -i metricbeat

The command should print out the metricbeat version installed on the system. If Metricbeat is installed proceed to step 6.

5. If nothing was returned in step 4 then run "puppet agent -t" on the host to have puppet install Metricbeat.

   # puppet agent -t

   Return to step 4 again to validate Metricbeat was installed. If puppet shows an error when running or Metricbeat is still not installed, then consult with an Elastic SME for guidance.

6. Run the installMetricbeatKeystore.sh script.
   # curl –k https://{site code}su01ro01.`hostname –d`/yum/elastic/install/installMetricbeatkeystore.sh | bash

a. You should see the message below when the script finishes. The script will create the Metricbeat-user and place it's randomly generated password into the metricbeat.keystore on the host the script is run on. Follow the directions to copy the new metricbeat.keystore to the elastic keystores directory on the repo server for future use.

IMPORTANT: If this is not completed, Metricbeat will not run correctly on any of the Elastic nodes.

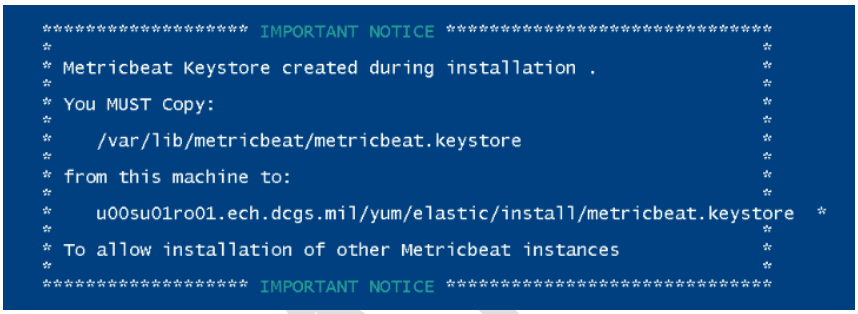


Figure 46 Metricbeat.keystore copy notice

7. Copy the metricbeat.keystore to the repo server for future use
   a. From the path above scp metricbeat.keystore <username>@x00su01ro01:/tmp
   b. login to the repo server and sudo to root
   c. cp /tmp/metricbeat.keystore /var/www/html/yum/elastic/keystores
   d. chown apache:apache /var/www/html/yum/elastic/install/metricbeat.keystore

8. Restart metricbeat on the host to ensure it uses the new keystore
   # systemctl restart metricbeat

5.4.11.5.2 Distribute keystore to all elastic nodes

Login to each remaining elastic node and run the installMetricbeatkeystore.sh script to copy over the keystore from the repo server. The script will validate the user has been created and that the keystore file is on the repo server when executed.

IMPORANT: This section needs to be completed on every remaining elastic node to ensure metricbeat is configured correctly to gather cluster monitoring data.

1. Ensure Metricbeat has already been installed on the host
   # rpm -qa | grep -i metricbeat

The command should print out the metricbeat version installed on the system. If Metricbeat is installed proceed to step 3.

2. If nothing was returned in step 1 then run “puppet agent -t” on the host to have puppet install Metricbeat.

   # puppet agent -t

   Return to step 1 again to validate Metricbeat was installed. If puppet shows an error when running or Metricbeat is still not installed, then consult with an Elastic SME for guidance.

3. Run the installMetricbeatKeystore.sh script.
   # curl –k https://{site code}su01ro01.`hostname –d`/yum/elastic/install/installMetricbeatkeystore.sh | bash

4. Restart metricbeat on the host to ensure it uses the new keystore
   # systemctl restart metricbeat

5.4.11.5.3 Validate Metricbeat monitoring in Kibana

Once Metricbeat has been configured properly on all nodes monitoring data should be showing in Kibana. This section gives steps to validate that monitoring data is flowing into the cluster.

1. Login to Kibana and select “Stack Monitoring” from the hamburger menu. (You may have to select the Default space first)
2. Validate that cluster overview information is displayed.

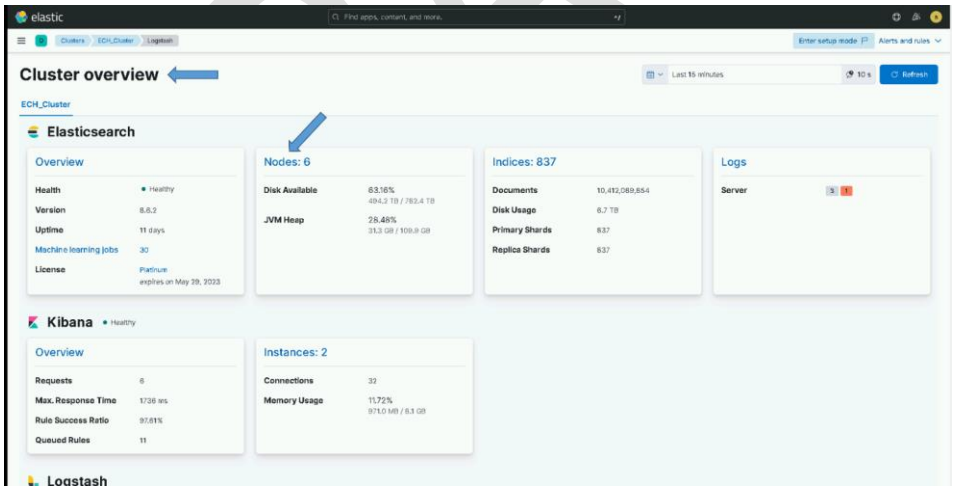


Figure 47- Example of Cluster overview

NOTE: If there is no monitoring information displayed consult with an Elastic SME for guidance

5.4.11.6 Metricbeat vSphere Data Collection

Once Metricbeat is automatically installed on the Logstash instance at each site it should be configured to collect vSphere statistics for that site. This is currently done manually by executing the following steps.

1. Log in to the Logstash host for each site and become root:

   ```
   # cd /etc/metricbeat
   ```

2. Create a keystore to hold the service account password:

   ```
   # metricbeat keystore create
   ```

3. Add the password for the {xx}_elastic.svc account to the keystore; {xx} is site number:

   ```
   # metricbeat keystore add V_PWD
   ```

4. Enter the Elastic service account password when prompted.

   NOTE: Ensure it is entered correctly; it does not prompt you twice.

Now that the service account password has been added to the Metricbeat keystore, set up the vSphere module to collect data.

1. Edit the vsphere.yml.disabled file in the modules.d directory of metricbeat:

   ```
   # vi modules.d/vsphere.yml.disabled
   ```

2. Update this file to look like the following example. Use the correct hostname for vc01 and the correct service account name for the site where this Logstash instance is running.

   Below the venter host is: u00av01vc01.ech.dcgs.mil
   The Elastic service account is: 00_elastic.svc

```
# Module: vsphere
# Docs: https://www.elastic.co/guide/en/beats/metricbeat/7.9/metricbeat-module-vsphere.html

- module: vsphere
  metricsets:
    - datastore
    - host
    - virtualmachine
  period: 30s
  hosts: ["https://u00av01vc01.ech.dcgs.mil/sdk"]

  username: "00_elastic.svc"
  password: "${V_PWD}"
  # If insecure is true, don't verify the server's certificate chain
  insecure: true
  # Get custom fields when using virtualmachine metric set. Default false.
  # get_custom_fields: false
~
~
~
```

Figure 48 Vsphere yml file example

NOTE: Spaces are important in YML files; ensure file's spacing matches the figure or it may not load correctly.

3. Enable the vSphere module:

```
# metricbeat modules enable vsphere
```

4. Test the configuration:

```
# metricbeat test modules vsphere | grep OK
```

5. The following should display:

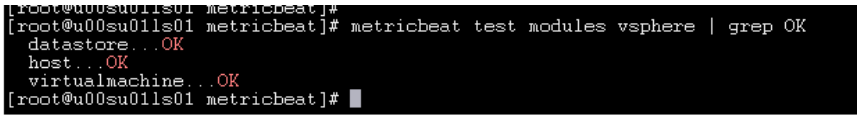


Figure 49 View configuration

6. Restart Metricbeat to have it begin sending vSphere data into Elasticsearch:

```
# systemctl restart metricbeat
```

7. Once configuration is complete, verify vSphere data is received by checking the **metricbeat- *** indexes under the **Discover tab on the** left side. Add a filter for the vSphere Module to ensure you are receiving data specifically for vSphere by using the **+Add Filter** tab and specifying **event.module: vsphere** for the filter.

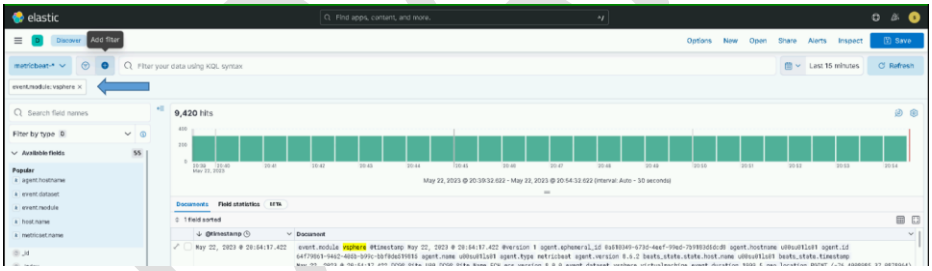


Figure 50 Verify vSphere data is received

5.4.11.7 Configure Heartbeat

Heartbeat is installed automatically by the **dsil_elastic_servers** Puppet module. Unlike Metricbeat and Filebeat, the configurations are not continually updated by Puppet. On the initial install the baseline monitor files are loaded. The computer names and port numbers in these files must be verified for correctness at each site where Heartbeat is installed. Any additional monitoring should also be added after the initial install. Note that Puppet will not overwrite any changes made to any of the monitors located in the **monitors.d** sub directory.

The initial configuration of Heartbeat will most likely need to be adjusted after its initial installation.

NOTE: Come back to this step after Heartbeat is installed. (Hearbeat is installed on each Logstash instance.)

5.4.11.7.1 Verify and Add Monitors

Heartbeat uses 3 different types of monitors to determine if your servers/services are reachable:

- **ICMP:** Uses an ICMP Echo Request to ping the configured hosts.
- **TCP:** Connects via TCP and optionally verifies the endpoint by sending and/or receiving a custom payload.
- **HTTP:** Connects via HTTP/s and optionally verifies that the host returns the expected response. PKI certificate start and end dates are also returned with HTTPS requests.

The following HTTPS and TCP monitor configurations are installed by default. The computer names and ports in these monitors must be verified for correctness after the initial install. Some servers/services are only available at the hub so there are extra monitors configured for that location.

- At ECH:
  - ess.http.hub.yml
  - ess.http.site.yml
  - ess.tcp.hub.yml
- At each Site:
  - ess.http.site.yml
  - ess.tcp.site.yml

The default ICMP monitor only pings the Logstash instance that Heartbeat is installed on. To get a list of hosts to ping for each site's Logstash instance, the **get_ldap_hosts.sh** script is executed hourly by the **cron.hourly** script **heartbeat.cron**. The get_ldap_hosts.sh queries Active Directory to get a list of all computers that start with the site's designator. This script can also be executed manually to verify the configurations. Run the following to modify the **/etc/heartbeat/monitors.d/ess.icmp.yml** file, updating the list of computers to ping for the Logstash instance:

```
# /etc/heartbeat/get_ldap_hosts.sh
```

View the /etc/heartbeat/monitors.d/ess.icmp.yml file to examine and verify the hosts to be pinged. If there are any hosts in the list that should not be pinged, they can be added to the exclusion list.

To add hosts that should not be pinged to the exclusion list, edit the **/etc/heartbeat/icmp_exclude.txt** file and hostnames (one per line) for them to be excluded. After adding hosts to the file, you can run the **get_ldap_hosts.sh** script manually or wait for it to be run automatically. The hosts will no longer be in the list contained in the ess.icmp.yml file.

Once configuration is complete, verify Heartbeat data is received by selecting the Kibana **Discover** tab and selecting the **Heartbeat-*** indexes. You should see "hits" populating on the selected graph to confirm Heartbeat data is being received.

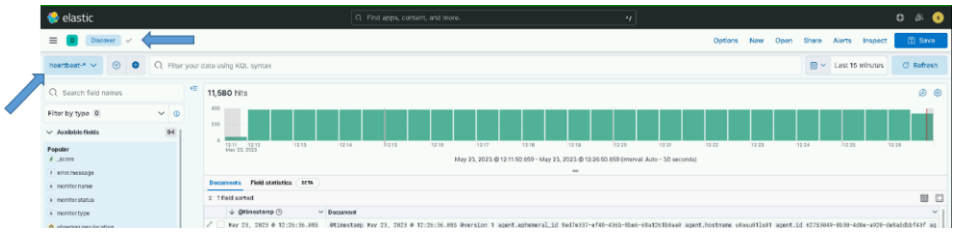


Figure 51 Verify Heartbeat data is received

5.4.12 Linux Syslogs

NOTE: A Puppet administrator is required to execute this section.

The syslog OSIF modules must be updated to forward syslog data to Logstash. Ingesting some of the data into Elasticsearch relies on the correct configuration of Linux hosts by some of the base OSIF modules. This section details any changes to OSIF needed for the Elasticsearch installation. A Puppet SME will be required to ensure this portion of the installation is done correctly.

NOTE: If this section does not match the currently installed version of OSIF consult both the Puppet and Elastic SME for guidance on ensuring updates are made to forward syslog data to Logstash hosts.

5.4.12.1 OSIF Common YAML

Changes to support Elastic configuration must be made in the OSIF common YAML file. Make your updates according to the version of OSIF running in the environment.

5.4.12.1.1 OSIF Version 1.1.16.1 or Later

The following changes should be made when the system where the installation is taking place is running version 1.1.16.1 or later of OSIF.

The rpmverify.sh script is run by cron regularly to ensure ownership and permissions of files are not modified from the RPM distribution. In some cases, file ownership and permissions are validly modified after installation. This is true for the components of Elastic. To ensure that the rpmverify.sh script does not undo changes made after installing an Elastic RPM, the component names must be added to the rpmverify exclusion list. To do this add the following to the osif.yaml file:

rpmverify::excluded_packages:

- ‘elasticsearch’
- ‘kibana’
- ‘logstash’

Ensure the following 2 variables are present and configured correctly in the osif.yaml to control sending of audit data to Logstash:

syslog::client::logstash_server: ‘logstash’

```
# Forward audit to Logstash
```

```
      syslog::send_to_logstash: true
```

5.4.12.1.2 OSIF Version 1.0.26 or Earlier

The following changes should be made when the system where the installation is taking place is running version 1.0.26 or earlier of OSIF.

Add Boolean in common.yaml to control sending of audit data to Logstash:

```
      # Forward audit to Logstash
      Syslog::send_to_logstash: true
```

5.4.12.2 osif_syslog Puppet Module

NOTE: This module is distributed with the OSIF baseline and may not match what is detailed in this section. If there is a discrepancy consult with both the Puppet and Elastic SME to ensure the module is configured properly.

The osif syslog module is used to set up the rsyslog configuration for all Linux hosts. Enterprise Elasticsearch should now be configured to receive syslog data from all Linux hosts as part of the default data types that are received. The osif_syslog Puppet module must be updated to have each Linux host send a copy of its syslog data to the Logstash instance at the site where the host resides. Puppet modules are controlled differently in version 1.0.26 and later. The updates to the module are basically the same in both cases but the way the module is deployed is different. This section is to ensure the module has the ability to configure rsyslog on hosts to forward data to Logstash.

Ensure the lookup of the **send_to_logstash** Boolean value to the class declaration of **ng_syslog**, between the open and close parentheses in the init.pp of the osif_syslog module.

```
class osif_syslog (
     …
     …
     Boolean $send_to_logstash = lookup('syslog::client::send_to_logstash',
Boolean, 'first', false),
     …
){
```

Ensure the following firewall rule exists in the module. (client.pp)

```
  # Check to see if we should send syslog to logstash
  $dest="logstash"
  if $osif_syslog::send_to_logstash {
    firewall { '052 allow rsyslog to logstash':
      chain       => 'OUTPUT',
      proto       => ['tcp'],
      dport       => $::logstash_port,
      destination => $dest,
      action      => 'accept',
    }
  }
```

Ensure the following is in the rsyslog client configuration template file (rsyslog.conf.client.epp)

```
<% if ($osif_syslog::send_to_logstash) { -%>
# Setup 2nd forwarded for Elastic
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
$ActionQueueFileName fwdRule2 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList   # run asynchronously
$ActionResumeRetryCount -1    # infinite retries if host is down
$DefaultNetstreamDriverCAFile       <%= $osif_syslog::ca_cert %>
$DefaultNetstreamDriver gtls        # use gtls netstream driver
$ActionSendStreamDriverMode 1       # require TLS for the connection
$ActionSendStreamDriverAuthMode anon # server is NOT authenticated
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
*.* @@logstash:<%= $osif_syslog::logstash_port %>
# ### end of the forwarding rule ###
<% } -%>
```

Once configuration is complete, verify syslog data is received by checking the **dcgs-syslog-*** indexes on the Kibana **Discover** tab, ensuring that we are currently receiving "hits" from syslog, and confirming the timestamp underneath the hits information.

Add a filter for "tags: Linux" to ensure your only looking at data forwarded via rsyslog

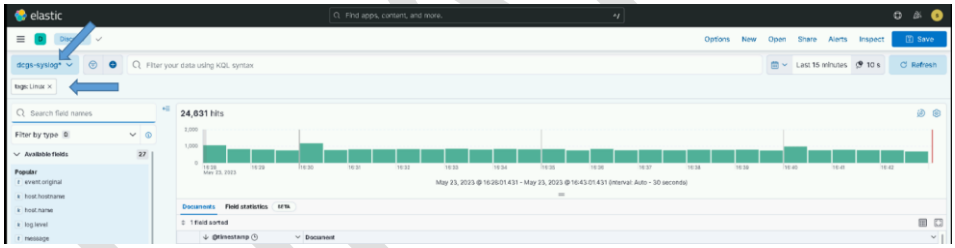


Figure 52 Verify Syslog data is received

5.4.13 Site Specific Ingest

Most of the data ingested into Elastic comes from all sites but there are some datatypes that only exist at specific sites. This section is to ensure that ingest for these datatypes is set up correctly and the Logstash instance at the specific site is configured to receive the data.

5.4.13.1 Service Account Used for Database Queries

Some of the data ingested into Elastic is queried from different SQL databases on DCGS. The Logstash instance at the site where the database is located is used to query any databases at that site. To execute queries against these databases, the Elastic service account **xx_elastic.svc** is used and must have read privileges for any database that is being queried. Before proceeding with enabling any database ingest, ensure the Elastic service account that will be used to query data is given privileges to read each database.

Follow the steps in *ES-018 - Microsoft SQL – Configuring SQL for Elastic Monitoring Instructions* to give permissions to the Elastic service account.

NOTE: Database Administration privileges are required.

The Logstash pipelines used to execute the database queries rely on the Elastic service account password being in the Logstash KeyStore. The key holding the password must be added to the Logstash KeyStore of any Logstash instance that will make database queries.

The key name is **es_svc_acct_password**.

Execute the following commands on the Logstash VM that will be using database pipelines to see if the key exists in the KeyStore:

```
# set -o allexport
# source /etc/sysconfig/logstash
# /usr/share/logstash/bin/logstash-keystore list --path.settings
/etc/logstash
```

```
[root@u00su01ls01 /]# /usr/share/logstash/bin/logstash-keystore list --path.settings /etc/logstash
Using bundled JDK: /usr/share/logstash/jdk
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties

es_man_password
es_mon_password
es_svc_acct_password
logstash_writer
mykey
puppet_postgres_password
ssl_passphrase
```

Figure 53 – logstash-keystore example output

If the key is not listed, execute the following to add it:

```
# /usr/share/logstash/bin/logstash-keystore add es_svc_acct_password --
path.settings /etc/logstash
```

Enter the service account password when prompted and press Enter. You should then see:

Added 'es_svc_acct_password' to the Logstash KeyStore.

5.4.13.2 SCCM Database

History – Added in 7.9.1 Release.

Dependencies – Permission given to the elastic service account to read from the SCCM database.

Use this section to verify that ingest for SCCM data is configured properly.

The Logstash pipeline used to read SCCM data is **esp_sccm_database**.

Before turning on ingest for this data type, examine the pipeline in Kibana to ensure that the **jdbc_connection_string** to access the SCCM database is correct for your environment. The SCCMDB DNS name is used to access the correct host and environment variables are used for other site-specific values. Below shows the how environment variables are used to make the pipeline generic. Environment variables that are used in Logstash pipelines are defined in the /etc/syslog/logstash file on each Logstash instance.

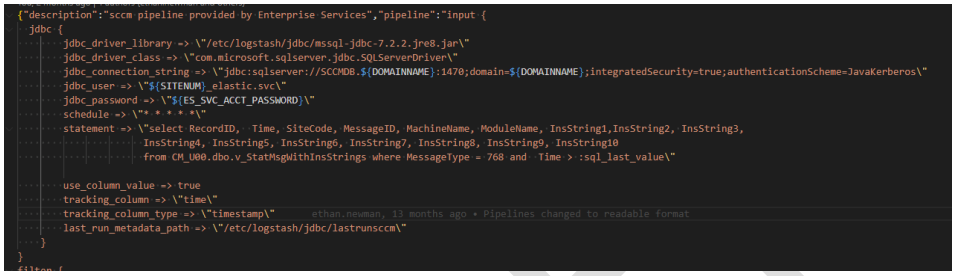


Figure 54 SCCM Pipeline Example

In this version, minimal data is queried from the SCCM database. This data source will be matured in future releases.

To set up querying of SCCM data add the **esp_sccm_database** pipeline to the node specific yml file for the Logstash instance that will query the data. This must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.

NOTE: There is only one instance of the SCCM database on the system; this pipeline should only be added to one instance of Logstash. On production it should be added to the ECH instance of Logstash. On CTE/MTE, choose the instance of Logstash that runs at the site where the Elastic Cluster is installed.

Once configuration is complete, verify SCCM data is received by going to Kibana **Discover** and selecting **dcgs-db_sccm*** data view. Verify that hits are being received and the number of hits displayed is increasing. Also confirm that the timestamp (underneath the hit count) is updating to reflect these changes.

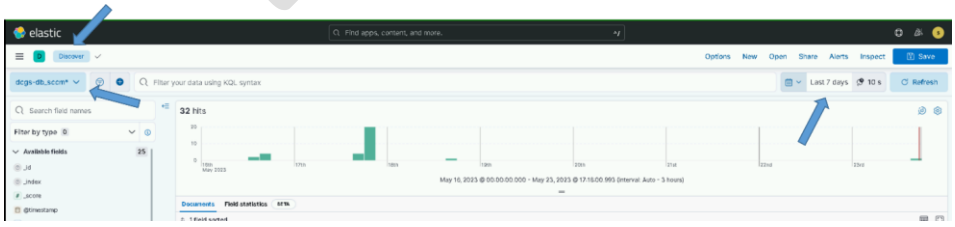


Figure 55 Verify SCCM data is received

NOTE: SCCM data can be scarce so if it's just being configured you may have to wait for a while to see any.

5.4.13.3 Ingest data from Puppet Postgres Database

History – Added in 7.16.3 Release.

Dependencies – Elastic user account exists in Puppet PostgreSQL database.

Use this section to verify that ingest for Puppet data is configured properly.

Audit information for Puppet will be ingested into Elasticsearch using the esp_puppet_database pipeline. The pipeline makes PostgreSQL queries on the Puppet database to ingest data into Elastic. This pipeline will run on the Logstash instance at the site that contains the Puppet Master-of-Masters. On the production system this is most likely ECH, but you can verify with a Puppet SME.

5.4.13.3.1 Create Elastic Postgres User

NOTE: A Puppet administrator is required to execute this section.

NOTE: This section should only be executed if the **elastic** Postgres user has not already been created. If you are not sure, you can view the list of users by doing the following:

1. Log into the Puppet Master-of-Masters and assume root privileges.
2. Enter the following command to access the database:
   `# sudo su - pe-postgres -s /bin/bash`
3. Connect to the pe-activity database by running the following command:
   `-bash-4.2$ /opt/puppetlabs/server/bin/psql -d pe-activity`
4. List existing users:
   `pe-activity=# \du`
5. Quit PSQL:
   pe-activity=# \q
6. If the list of users contains the **elastic** user, skip to section **Error! Reference source not found.**.

If the **elastic** user has not been created yet, for Elastic to have access to the Postgres database, the following setup must be completed.

The following steps will create a new user in the Puppet Enterprise Postgres Database. These steps must be followed to allow the esp_puppet_database Logstash pipeline access to the Puppet database.

1. Log into the Puppet Master-of-Masters and assume root privileges.
2. Enter the following command to access the database:
   `# sudo su - pe-postgres -s /bin/bash`
3. Execute the following command to create the **elastic** user:

```
-bash-4.2$ /opt/puppetlabs/server/bin/createuser -D -S -R –P -l
elastic
```

4. When prompted, enter a password for the user and then again to verify it. **Remember this password as you will need to add it to the Logstash keystore.**
5. Connect to the pe-activity database by running the following command:
   ```
   -bash-4.2$ /opt/puppetlabs/server/bin/psql -d pe-activity
   ```
6. Enter the following commands to grant SELECT permissions on the database for the user just created:
   ```
   pe-activity=# GRANT SELECT ON events TO elastic;
   pe-activity=# GRANT SELECT ON event_commits TO elastic;
   pe-activity=# GRANT CONNECT ON DATABASE "pe-activity" TO elastic;
   ```
7. Quit PSQL:
   ```
   pe-activity=# \q
   ```

5.4.13.3.2 Add Password for Elastic Postgres User to Logstash Keystore

After creating the Elastic user/password for accessing the Puppet Postgres database, the password must be added to the Logstash keystore at the site where the puppet Master-of-Masters is installed. The esp_puppet_database Logstash pipeline will need the password to query data from the database.

Log in to the Logstash VM and perform the following steps.

1. Become root:
   ```
   # sudo su
   ```
2. Add password to Logstash keystore:

   ```
   # set -o allexport
   # source /etc/sysconfig/logstash
   # /usr/share/logstash/bin/logstash-keystore --path.settings
   /etc/logstash add puppet_postgres_password
   ```

   NOTE: You can list the current keystore values by using the **list** option for the logstash-keystore command:
   ```
   # /usr/share/logstash/bin/logstash-keystore –path.settings
   /etc/logstash list
   ```
3. When prompted, enter the password created in the previous section. Enter value for puppet_postgres_password.

   NOTE: You will only enter the password once.

5.4.13.3.3 Activate esp_puppet_database Pipeline

To start receiving audit information from the Puppet database the esp_puppet_database pipeline must be activated on the Logstash VM at the site where the puppet Master-of-Masters resides.

To set up querying of puppet data add the **esp_puppet_database** pipeline to the node specific yml file for the Logstash instance that will query the data. This must be updated in the dsil_elastic_servers puppet

module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.

```
dsil_elastic_servers::logstash::pipelines:
  '["esp_postgres", "esp_metricbeat", "esp_hbss_epo", "esp_filebeat-singleworker",
  "esp_sqlserver_stats", "esp_winlogbeat", "esp_filebeat", "esp_linux_syslog",
  "esp_loginsight", "esp_arcsight_udp", "esp_heartbeat",
  "esp_filebeat-logstash", "esp_hbss_metrics", "esp_puppet_database",
  "esp_sccm_database", "esp_eracent_database"]
```

Figure 56 Add "esp_puppet_database"

Check /var/log/logstash/logstash-plain.log for any errors in the pipeline. If you see authentication errors you may have entered the password incorrectly into the Logstash keystore. Try the previous section again and restart Logstash to resolve.

Once configuration is complete, verify puppet data is received by going to Kibana **Discover** and selecting **dcgs-puppet-*** indexes in the drop-down under **+Add Filter**. Verify that hits are being received.

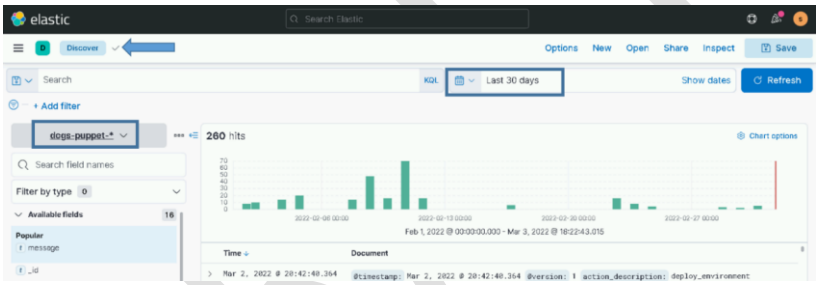


Figure 57 Example of puppet data in Discover tab

NOTE: Puppet data can be scarce so if it's just being configured you may have to wait for a while to see any.

5.4.13.4 HBSS Data

NOTE: HBSS is transitioning to Endpoint Security Solutions (ESS). Both may be referred to in this document.

There are two pipelines to receive data directly from the HBSS ePO server. There is only one ePO server in each environment, usually at the Hub. You can verify the location of the HBSS ePO with an HBSS SME to determine which Logstash instance should be ingesting the data.

Use this section to verify that ingest for HBSS syslog data is configured properly.

5.4.13.4.1 HBSS EPO data

History – Added in 7.9.1 Release.

Dependencies – HBSS is configured to forward using rsyslog to Logstash port 5049

After the pipelines are installed in section 5.4.5.15.1, Logstash is ready to ingest data from HBSS. The e**sp_hbss_epo** pipeline must be in the Logstash YAML file for the Logstash instance that is to ingest the HBSS data. There is only one HBSS ePO per enclave on DCGS and it is usually at the hub location. Check with an HBSS SME to determine which Logstash instance will be ingesting the data.

1. To set up Logstash to receive HBSS syslog data the node specific yml file for the Logstash instance must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.

2. Ensure the **esp_hbss_epo** pipeline ID is included in the **dsil_elastic_servers::logstash::pipelines: [ ]** array for the Logstash instance that will receive data from HBSS

   Example:

```
dsil_elastic_servers::logstash::pipelines:
  '["esp_postgres", "esp_metricbeat", "esp_hbss_epo", "esp_filebeat-singleworker",
  "esp_sqlserver_stats", "esp_winlogbeat", "esp_filebeat", "esp_linux_syslog",
  "esp_loginsight", "esp_arcsight_udp", "esp_heartbeat",
  "esp_filebeat-logstash", "esp_hbss_metrics", "esp_puppet_database",
  "esp_sccm_database", "esp_eracent_database"]'
```

Figure 58 Ensure the esp_hbss_epo pipeline ID is included

3. Once you are sure Logstash is ready to receive HBSS syslog data, ask an HBSS SME to set up the HBSS ePO to send its syslog to Logstash. See the HBSS *Syslog Publishing of ePolicy Orchestrator Events to Elastic* document for configuration instructions.
4. When configuration is complete, verify HBSS data is received by going to Kibana **Discover**, checking the **dcgs-hbss_epo*** indexes. Confirm that hits are being received (and the count is increasing). Additionally, confirm the timestamp has updated to reflect the receipt of data.

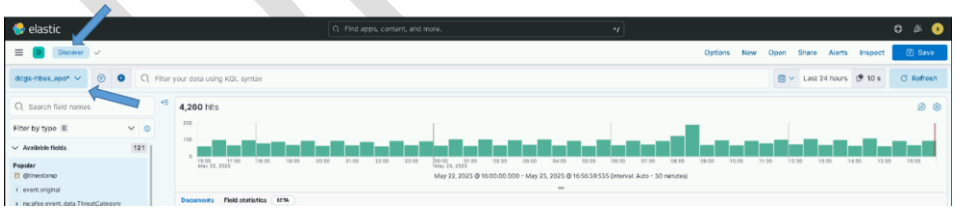


Figure 59 Verify HBSS data is received

5.4.13.4.2 HBSS Metrics

History – Added in 7.16.3 Release.

Dependencies – CR-2022-OADCGS-013: ESS - ePO-Monitor for Elastic data forwarded to Logstash port 5046

As of this version Metricbeat is not installed on the ePO server. To get health metrics, a custom PowerShell script was created to send the ePO metrics directly to Elastic. The esp_hbss_metrics pipeline receives data directly from the HBSS PowerShell script on port 5046. The types of messages that are ingested from the PowerShell script are shown in the following table.

Table 11 Types for Ingested HBSS Metrics

ID	Type	Description
1000	Information	ePO Monitor Script run started
1	Information	Services Report
101	Error	Service not Running or not installed Error
2	Information	Disk Use Report
52	Warning	Disk Warning Report
102	Error	Disk Error Report
3	Information	File sizes Report
4	Information	Web certificate information
54	Warning	Web certificate warning
104	Error	Web certificate Error
5	Information	SQL Backup Report (most recent)
6	Information	CPU Utilization
7	Information	Memory Utilization

The esp_hbss_metrics pipeline receives the PowerShell scripts data directly from the HBSS ePO server.

1. To set up Logstash to receive HBSS metrics the node specific yml file for the Logstash instance must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.
2. Ensure the **esp_hbss_metrics** pipeline ID is included in the **dsil_elastic_servers::logstash::pipelines: [ ]** array.

   Example:

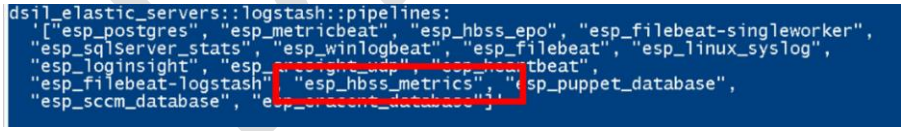


Figure 60 Ensure the esp_hbss_metrics pipeline ID is included

3. If you must add the pipeline ID, Logstash must be restarted before it will be ready to ingest the HBSS data.

   ```
   # systemctl restart logstash
   ```

   IMPORTANT: Only do this if you had to add **esp_hbss_metrics** to the array.

4. Once you are sure Logstash is ready to receive HBSS PowerShell data, ask an HBSS SME to set up the HBSS ePO to send metrics collected using the new PowerShell script to Logstash. See the *ESS – ePO-Monitor for Elastic Installation Instructions* document for configuration instructions.
5. When configuration is complete, verify HBSS data is received by going to Kibana **Discover**, checking the **dcgs-hbss-metrics-*** indexes in the drop-down under **Add Filter**, and confirming that hits are being received (and the count is increasing). Additionally, confirm the timestamp has updated to reflect the receipt of data.

NOTE: The HBSS PowerShell script sends data every 30 minutes so you will only see events on those intervals.

5.4.13.5 Eracent Audit Data

History – Added in 7.16.3 Release.

Dependencies – Database exists, and Elastic service account is given read access.

The Logstash pipeline used to read Eracent data is **esp_eracent_database**. The pipeline executes SQL queries to ingest data from the Eracent database. There is only one instance of the Eracent database on the system; this pipeline should only be added to one instance of Logstash. On production it should be added to the ECH instance of Logstash. On CTE/MTE, choose the instance of Logstash that runs at the site where the Elastic Cluster is installed.

Examine the pipeline in Kibana to ensure that the **jdbc_connection_string** to access the Eracent database is correct for your environment. (It should include the SQL instance and database names). The **jdbc_user** field will be populated with the service account name for the site of the Logstash instance running the pipeline. (xx_elastic.svc where XX = site number). Also validate the hostname for the Eracent database (OADCGS02) is correct for your environment.

Figure 61 Example of Eracent pipeline configuration

1. To set up Logstash to receive Eracent data the node specific yml file for the Logstash instance must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.

2. Ensure the **esp_eracent_database s** pipeline ID is included in the **dsil_elastic_servers::logstash::pipelines: [ ]** array.

   Example:

```
dsil_elastic_servers::logstash::pipelines:
  '["esp_postgres", "esp_metricbeat", "esp_hbss_epo", "esp_filebeat-singleworker",
  "esp_sqlServer_stats", "esp_winlogbeat", "esp_filebeat", "esp_linux_syslog",
  "esp_loginsight", "esp_arcsight_udp", "esp_heartbeat",
  "esp_filebeat-logsta                    esp_puppet_database",
  "esp_sccm_database", "esp_eracent_database"]'
```

Figure 62 Ensure the esp_eracent_database pipeline ID is included

1. If you must add the pipeline ID, Logstash must be restarted before it will be ready to ingest the Eracent data.

   ```
   # systemctl restart logstash
   ```

2. When configuration is complete, verify Eracent data is received by going to Kibana **Discover**, selecting the dcgs-db_eracent-***** data view, and confirming that hits are being received (and the count is increasing). Additionally, confirm the timestamp has updated to reflect the receipt of data.

Figure 63 Example of Eracent data in Discover tab

NOTE: Eracent data can be scarce so you may have to wait for a while to see any if it's just being configured.

5.4.13.5.1 SQL Server Statistics

The Logstash pipeline used to read SQL Server Statistics data is **esp_sqlServer_stats**.

Before turning on ingest for this data type, examine the pipeline in Kibana to ensure that the **jdbc_connection_string** for each query to access the SQL database is correct for your environment. There are multiple queries used to gather this information so check with an SQL ADMIN to ensure they are correct for each Logstash instance you enable this query on.

To set up querying of SQL Server Statistics, add the **esp_sqlServer_stats** pipeline to the node specific yml file for the Logstash instance that will query the data. This must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.

NOTE: There are multiple SQL database instances throughout the system. This pipeline should be added to Logstash instances at sites where databases are located. You must check with the SQL Administrator to ensure the connection strings are correct for each site.

Once configuration is complete, verify SQL server data is received by going to Kibana **Discover** and selecting the **dcgs-db_sqlserver*** data view. Verify that hits are being received and the number of hits displayed is increasing. Also confirm that the timestamp (underneath the hit count) is updating to reflect these changes.

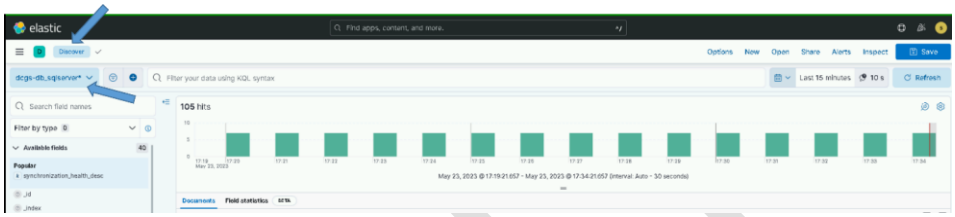


Figure 64 Verify SQL server data is received

5.4.13.6 SQL Server Statistics

History – Added in 7.9.1 Release.

Dependencies – Databases exist, and Elastic service account is given read access.

The Logstash pipeline used to read SQL server statistics is **esp_sqlServer_stats**. The pipeline executes SQL queries to ingest database status data from any SQL database. This pipeline should be configured on the Logstash instance at the site where the databases are located. The Logstash instance can be used to query any databases at any site but should only be configured to run on one Logstash instance. To execute queries against these databases, the Elastic service account **xx_elastic.svc** is used and must have read privileges for any database that is being queried. Before proceeding with enabling any database ingest, ensure the Elastic service account that will be used to query data is given privileges to read each database. Follow the steps in *ES-018 - Microsoft SQL – Configuring SQL for Elastic Monitoring Instructions* to give permissions to the Elastic service account.

On production, databases are located at the hub so the ECH instance of Logstash should be used. On CTE/MTE, choose the instance of Logstash that runs at the site where the Elastic Cluster is installed.

The query being done on each database is the same, but the installer must add a **jdbc** block for each SQL database that should be monitored. The xx_elastic.svc account must be given read access to each database to be able to execute the query.

The following image shows a jdbc block configured to query data from a database on u00sm01sq20 at port 1460.

Figure 65 Example SQL query to database

To configure a query to a database, copy the existing block in the esp_sqlserver.

Update the following:

jdbc_connection_string – Should have the host and port of the database you are querying.
last_run_metadata_path – Update the name of the file to be unique. This file holds the last query time to allow only asking for new data on each query.

The jdbc_user is automatically set to the elastic service account for the site of the Logstash instance that is running the pipeline(xx_elastic.svc). The jdbc_password is the service account password and should already be stored in the logstash.keystore for use by any pipeline.

1. To set up querying of SQL Server Statistics, add the **esp_sqlServer_stats** pipeline to the node specific yml file for the Logstash instance that will query the data. This must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.
2. Ensure the **esp_sqlServer_stats** pipeline ID is included in the **dsil_elastic_servers::logstash::pipelines: [ ]** array.

   Example:

Figure 66 Ensure the esp_sqlServer_stats pipeline ID is included

3. Once configuration is complete, verify SQL server data is received by going to Kibana **Discover** and selecting the **dcgs-db_sqlserver*** data view. Verify that hits are being received and the number of hits displayed is increasing. Also confirm that the timestamp (underneath the hit count) is updating to reflect these changes.

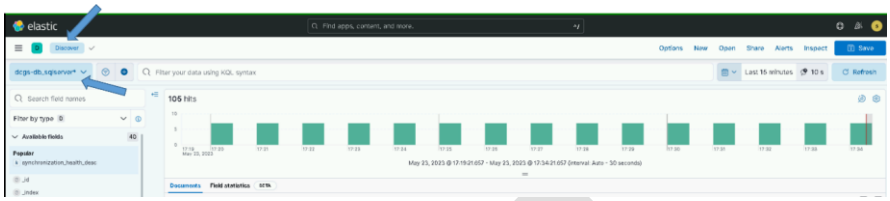


Figure 67 Verify SQL server data is received

Formatted: Indent: Left: 0"

Formatted: Normal

5.4.13.7 Re-Activate Log Insight data Ingest

Note: A Log Insight SME is required for this section

To re-enable log forwarding from Log Insight to Logstash the following updates must be done on Log Insight at each site. Two forwarders will be created, one sending data in raw format ant the other sending syslog format.

Note: replace {xxx} below with the site designator: examples: “s00”, “t01”

Create the syslog forwarder:

1) Login to Log Insight web console and select “Log Management” from the menu on the left side.
2) Select “Log Forwarding” and then “New Destination.”
3) Use the following for the options:

Name: Logstash-syslog
Host: logstash
Protocol: Syslog
Transport: TCP
Filter: “hostname” “does not match” “{xxx}sm*”
“agentgenerated matches agentgenerated”
Port: 5050
Worker Count: 8

Create the raw forwarder:

1) Login to Log Insight web console and select “Log Management” from the menu on the left side.
2) Select “Log Forwarding” and then “New Destination.”
3) Use the following for the options:

Name: Logstash-raw
Host: logstash
Protocol: Raw
Transport: TCP
Filter: "hostname" "does not match" "{xxx}sm*"
"agentgenerated does not match agentgenerated"
"text does not match */ifs/{xxx}/app/elac/*"
"product does not match prelude"
Port: 5050
Worker Count: 8

5.4.13.8 Setup DLP data ingest from ESS (HBSS)

Formatted: Heading 4

The following 2 sections describe two options for ingesting decoded DLP information from the EPO Server. The first option is the preferred method but cannot be used until ESS is running version 11.10.100 or later. Once ESS is upgraded to this level this option should be used. If ESS is running an older version, then the 2nd option must be used until ESS is upgraded.

5.4.13.8.1 Query using DLP API (Preferred option)

Formatted: Heading 5

To use this method simply activate the esp_hbss_dlp pipeline that is included with the release. This is done by adding it to the logstash.yml configuration for the site that talks to the ESS EPO. A puppet update is required to make this configuration change. See section 5.5.2.1 for details on how to update the logstash.yml configuration for a site.

5.4.13.8.2 Receive DLP data from existing ArcSight connector (Short term work around)

Formatted: Heading 5

As stated above this option is to be used only until ESS is upgraded to version 11.10.100 or higher. If this method is implemented, then once ESS is upgraded the ingest method should be changed to use the esp_hbss_dlp pipeline as described in the previous section.

To receive data from the Existing ArcSight connector do the following:

1) Activate the esp_hbss_dlp-via-connector ingest pipeline by adding it to the logstash.yml configuration. See section 5.5.2.1 for details on how to update the logstash.yml configuration for a site. This pipeline should only be added to the logstash configuration for the site where the EPO server resides.
2) Follow the install instruction provided for configuring the ESS ArcSight connector to forward data to Logstash: *"IAAS-018 – ESS – Temporary ArcSight Connector for Elastic Installation Instructions.docx"* (Document can be found in install/docs director on reposerver)
    a. Artifacts needed for the installation are distributed with the oadcgs-es-elastic-reposerver package in the install/artifacts/ess directory.
        - DeployArcSightMod.zip – Contains artifacts needed for configuring ArcSight connector.
        - DeployArcSightMod.text - Holds Hashes for Artifacts in zip

3) Validate the DLP events are received in the dcgs-hbss_epo_dlp-iaas-ent index

5.4.13.9 Switch Syslog data ingest

Formatted: Heading 4

This section is to configure the switches at each site to send syslog data directly to Logstash. Each switch should be setup to send syslog data directly to Logstash.

5.4.13.9.1 Prepare Logstash to receive switch data

Formatted: Heading 5

Switches may send syslog data via UDP or TCP to Logstash. The esp_syslog_tcp and esp_syslog_udp pipelines have been added to allow the ingestion of this data. These new pipelines must be activated on each Logstash instance by adding them to the logstash.yml configuration.

Validate that the esp_syslog_tcp and esp_syslog_upd pipelines are contained in all logstash configurations defined in puppet. See section 5.5.2.1 for details on puppet configurations for logstash.

5.4.13.9.2 Configure switches to forward syslog data

The network/switch SMEs are needed to perform this configuration. Ensure that all switches are configured to send syslog data to one of the following logstash endpoints at each site:

TCP (Unencrypted) – Logstash: port 5055

UDP – Logstash: 514 (port forwarding is enabled on logstash to send this data to port 5040)
(**Note**: UDP data can also be sent directly to Logstash port 5040)

Provide network team with instructions to configure switches to forward data to Logstash at each site located in *10Appendix A Prime Update Instructions* or *"Prime Updates.docx"* (Document can be found in install/docs director on reposerver)

Artifacts needed for the installation are distributed with the oadcgs-es-elastic-reposerver package in the install/artifacts/prime_templates directory:

- Cisco_Prime_Logstash_Update_Templates.zip

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

5.4.13.10 Serena data ingest

IMPORTANT: This section only applies to systems that have connectivity to Serena. If there is no network connection to Serena from the system you are installing on then skip this section

NOTE: You must be root and a member of the **ent elastic admins** AD group to load saved objects into Kibana. Having the **Elastic Administrator** OneIM Role will place the user in this group.

NOTE: An SQL Database SME will also be needed to give the elastic service account permissions to the Serena database.

IMPORTANT: Prior to executing this section the service account at the site where this feature will be activated must be given permissions to read data from the Serena database by an SQL Database SME.

This section will active the ingestion of data from the Serena database. The following information will be needed and prompted for by the active script.

- HOSTNAME - Hostname of SQL server holding Serena database
- PORT_NUM - Port on the SQL server to access the database
- DOMAIN_NAME - Domain that this activation is being performed on
- INSTANCE_NAME - Serena database instance name
- DATABASE_NAME - Serena database name

These values will be used to create the JDBC connection string as follows:
*jdbc:sqlserver://**HOSTNAME:PORT_NUM**;domain=**DOMAIN_NAME**;instanceName=**INSTANCE_NAME**;databaseName=**DATABASE_NAME**;integratedSecurity=true;authenticationScheme=JavaKerberos\"*

This is an example of a connection string created by the activate script and added to the esp_serena pipeline.

*jdbc:sqlserver://**u00sm01sq20.dcgs.mil**:**1460**;domain=**dcgs.mil**;instanceName=**ES01**;databaseName=**SBM_APP_2012**;integratedSecurity=true;authenticationScheme=JavaKerberos"*

Follow these steps to run the active serena script:

1. Login to the Logstash Instance that will be used to communicate with the Serena database. The best choice for this would be the Logstash instance at the same site that hosts the Serena database.

   # sudo su

2. Run the active serena script

   ```
   # curl -k https://{site code}su01ro01.`hostname -
   d`/yum/elastic/install/activate_serena.sh | bash
   ```

3. To set up querying of Serena data, add the **esp_serena_database** pipeline to the node specific yml file for the Logstash instance that will query the data. This must be updated in the dsil_elastic_servers puppet module. Refer to section 5.4.1.8.2 for information on controlling the pipelines that run on each Logstash instance.

5.4.13.11 Activate ACAS data ingest

NOTE: An ACAS SME will be needed to execute this section

IMPORTANT: This section depends on AD version 3.12: *RFC CR-2023-OADCGS-015 – Standard Change: Upgrade Enterprise Service Foundation Data Active Directory (ESFDAD) NOFORN from v3.3.11 to V.3.12.* If Active Directory is not running this version skip this section, it will be attempted again in the next Elastic upgrade.

The Elastic Data Collector can query Vulnerability, Scanner Status, and System Status data from ACAS to be ingested into the collector. It uses the TenableSC API to be able to display this data. By default, this capability is disabled. The following guide will show the user how to activate the ACAS portion of the Data Collector.

5.4.13.11.1 Configure ACAS to allow API Keys

1. Log in to Tenable.sc via the user interface as an **Admin** (**Error! Hyperlink reference not valid.**).

Figure 68- Example of tenable login page

2. In the top navigation bar, click **System** > **Configuration**. The **Configuration** page appears.

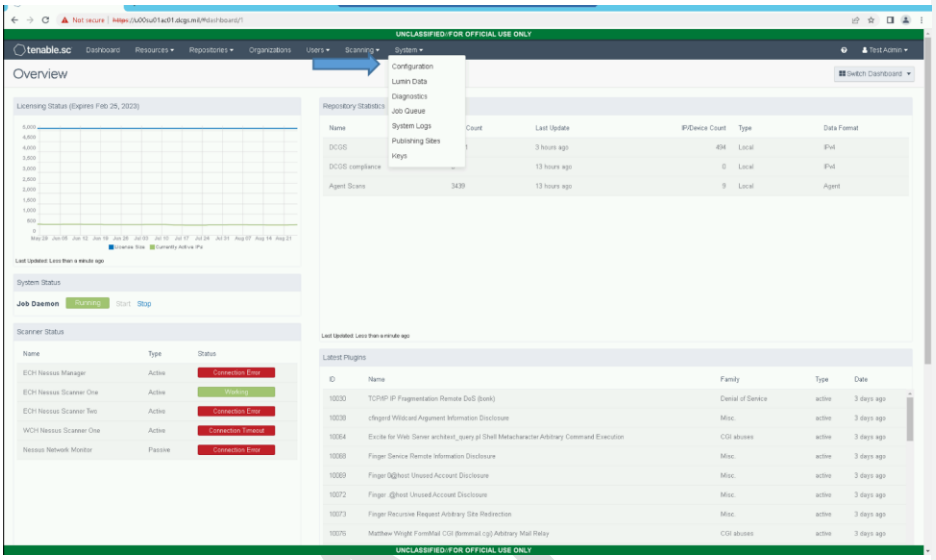


Figure 69- Example of selecting "Configuration" option.

3. Click the **Security** tile. The **Security Configuration** page appears.

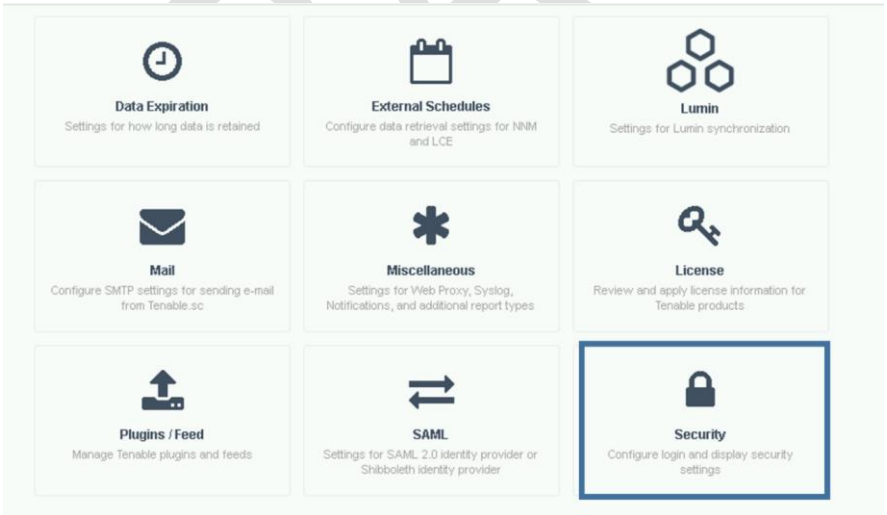


Figure 70- Example of selecting Security option.

4. In the **Authentication Settings** section, click **Allow API Keys** to enable the toggle. Click **Submit**.

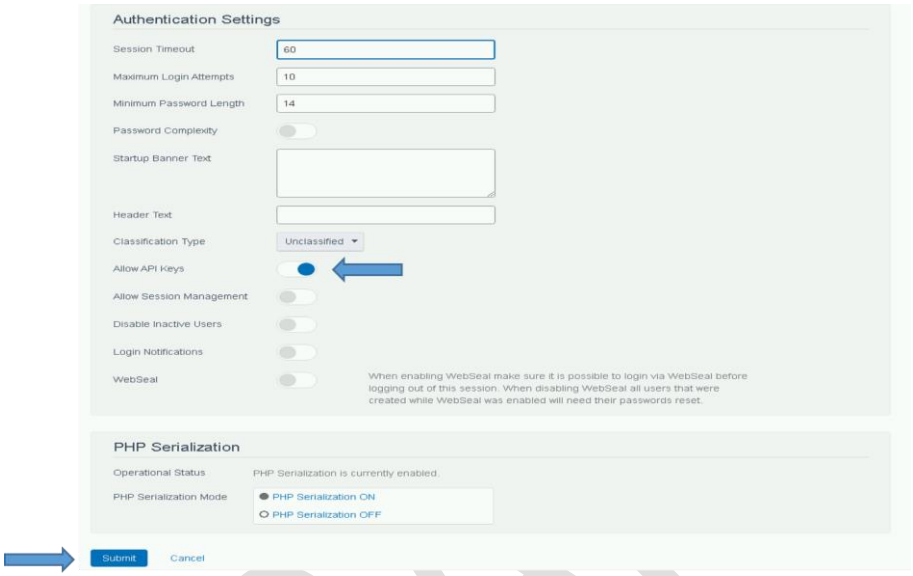


Figure 71- Example of turning on API Keys option.

5.4.13.11.2 Generate API Keys for the Elastic Service Account

API Keys for use by the elasticDataCollector should be generated from the "ent_elastic_acas.svc" account.

1. Log in to Tenable.sc via the user interface.

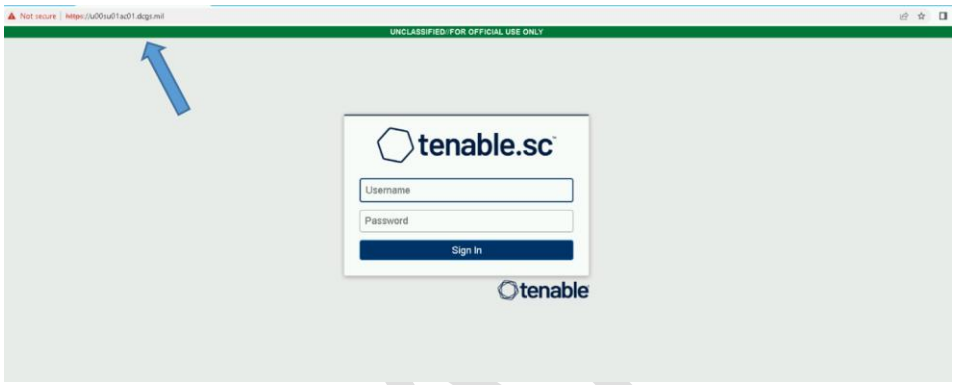


Figure 72- Example login page for tenable

2. Click **Users** > **Users**. The **Users** page appears.

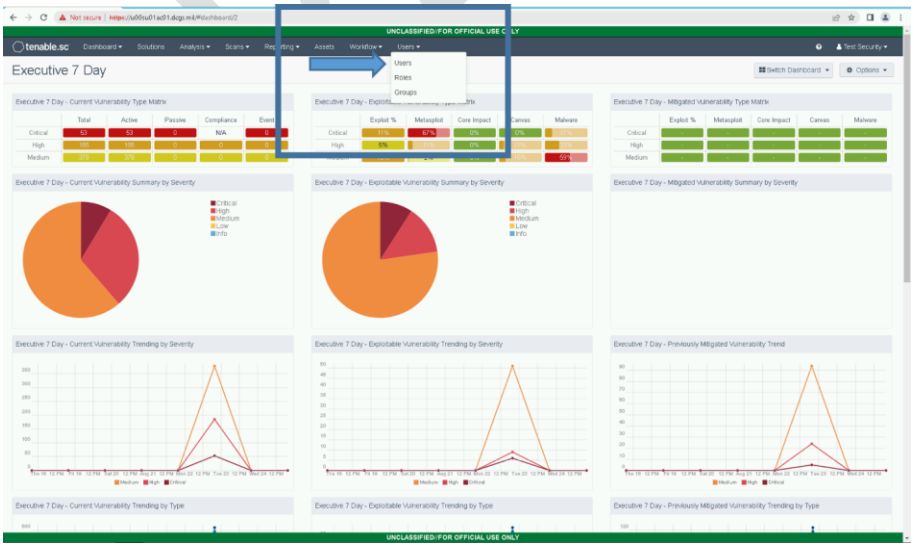


Figure 73- Example selecting "Users."

3. Click the **Settings Button** for the ent_elastic_acas.svc account to generate an API key, then **Generate API Key**

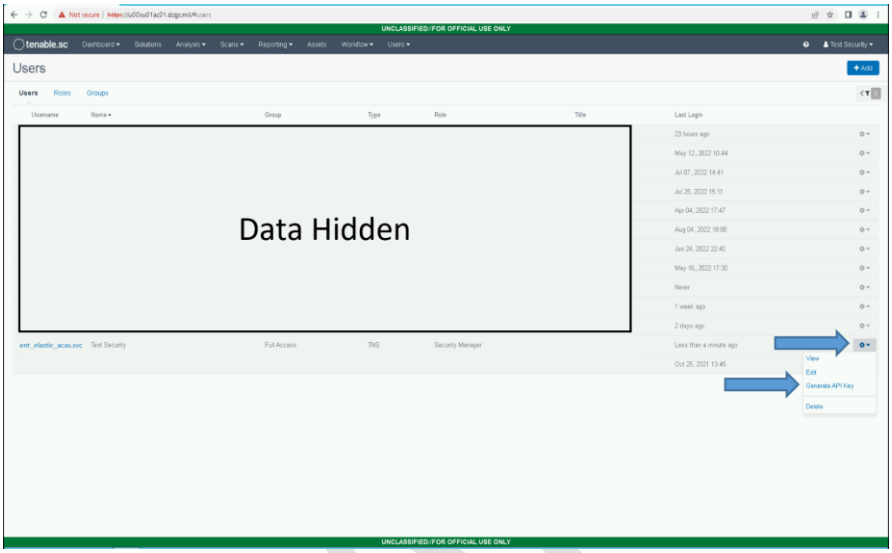


Figure 74- Example of selecting "Generate API Key" for a user.

4. Click **Generate**.

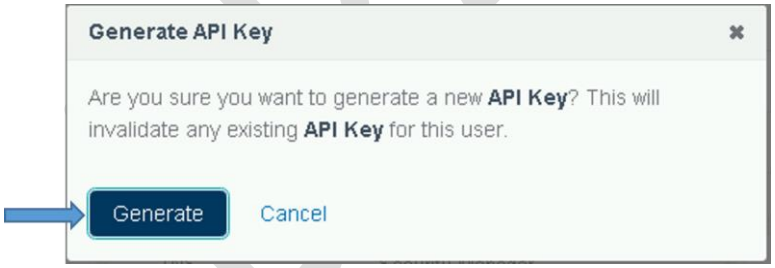


Figure 75- Example of Generate API Key confirmation.

5. The **Your API Key** window appears, displaying the access key and secret key for the user.

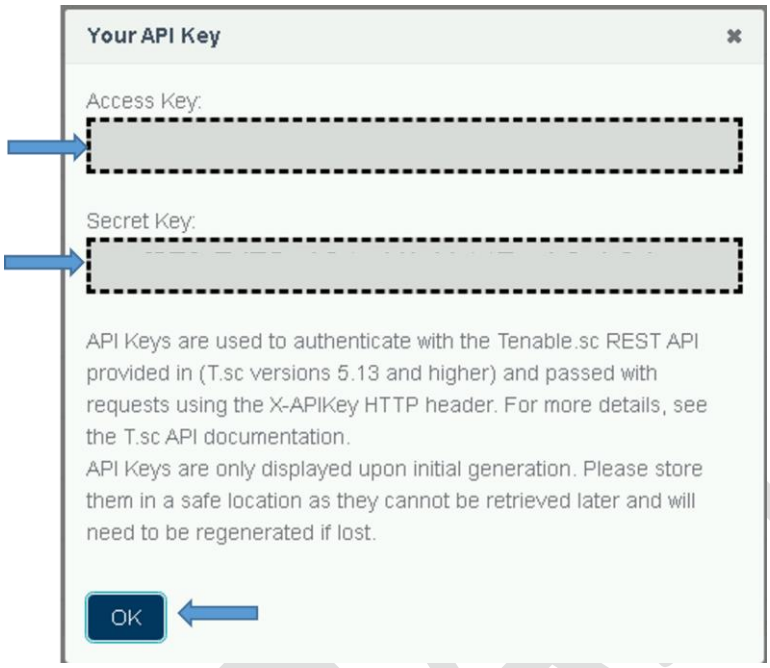


Figure 76- Example API Key Display

6. **Save the Access and Secret Keys for the Elastic Service Account**.

5.4.13.11.3 Activate ACAS for the ElasticDataCollector at the HUB

Log in to the Logstash VM at the site where the ACAS server resides (On production this is ECH) and perform the following steps.

1. Sudo to become root.
   #sudo su

   Run activate_acas.sh
   # curl –k https://{site code}su01ro01.`hostname –d`/yum/elastic/install/activate_acas.sh | bash

2. Input the Access and Secret Keys obtained above into the command prompt when prompted.

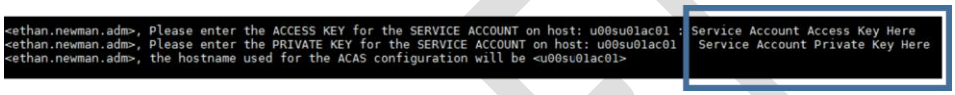


Figure 77- Example of prompts during install

3. The script will also create and assign the ACAS delete role to the querier user for the site where it is installed and create the acas.cron job. An example of the script running successfully:

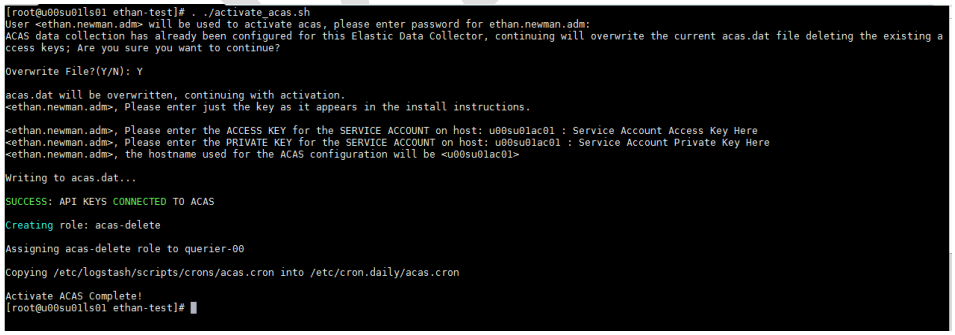


Figure 78- An example of the script running successfully

4. Restart the Elastic Data Collector
   # systemctl restart elasticDataCollector

5. To check if the script worked, look in /etc/logstash/scripts/data/acas.dat and see if the keys were created. It should look like:
   # cat /etc/logstash/scripts/data/acas.dat

6. Once configuration is complete, verify ACAS data is received by going to **Kibana Discover** and selecting **dcgs-acas-*.**

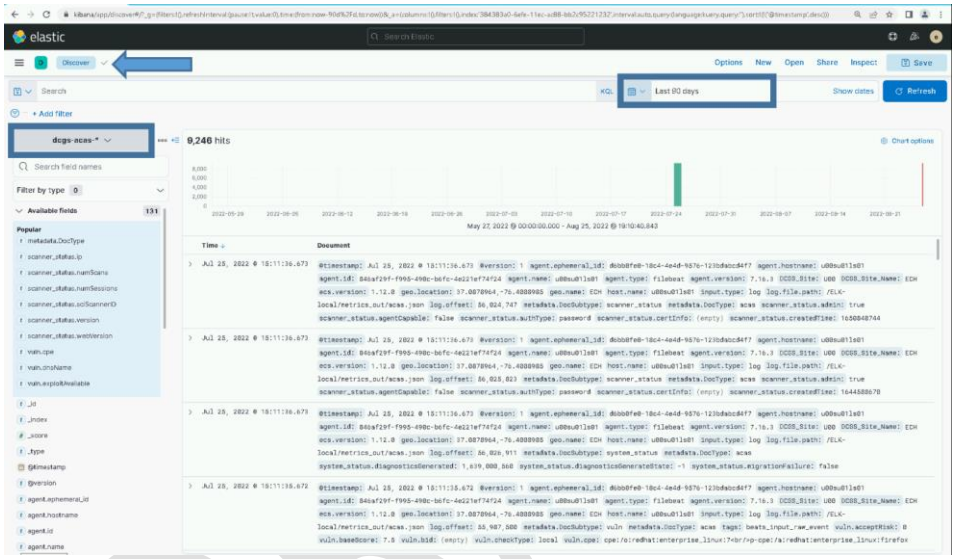


Figure 79- Discover showing example acas data

5.4.14 Remove "Run and Remove" scripts from system when upgrade is completed

The scripts that are delivered in the install directory on the repo servers are intended for use during this upgrade. Once the upgrade is completed this directory and these scripts may be removed from the system as they are not used for operational purposes.

5.5 Installation Instructions for Upgrades

The instructions in this section are for use when upgrading the DCGS Enterprise Service Elasticsearch from version 7.17.6 to version 8.6.2.

IMPORTANT: THE INSTALL ORDER MATTERS

The following steps will be accomplished during this install. The steps must be done in the order specified to ensure a successful upgrade. Use the following as a guide on the order to execute each section.

1. Update repo server at all sites
2. Update puppet modules (dsil_elastic_clients and dsil_elastic_servers)
3. Upgrade Elasticsearch Cluster (All Nodes)
4. Upgrade Kibana Instances
5. Update Roles
6. Update Kibana Settings
7. Update Filebeat Ingest Pipelines
8. Update Templates
9. Bootstrap Indexes
10. Update Enterprise Services Centralized Logstash Pipelines
11. Install Watchers
12. Upgrade Logstash Instances (All Sites)
13. Upgrade Data Collector (All Sites)
14. Upgrade Beats (Linux, Windows and Domain Controllers)
15. Update Dashboards/Visuals in Kibana
16. Reindex existing data
17. Activate Serena data ingest (Optional)
18. Activate ACAS data ingest (Depends on AD version 3.12)
19. Remove “Run and Remove” scripts from system when upgrade is completed.

5.5.1 Update Repo Server

NOTE: A Linux administrator will be needed to execute this section.

The installation scripts and artifacts are delivered and placed in the “install” directory of the elastic repo on the repo server at each site. This is done so the scripts and artifacts are accessible for use during the installation at each site.

5.5.1.1 Update install directory with 8.6 install package

Follow the steps in this section to update the “install” folder in the elastic repository with the scripts and artifacts needed for the 8.6 install.

2. Login to repo server, sudo to root and change directory to the elastic repo

   # sudo su
   # cd /var/www/html/yum/elastic (This is an example, the actual repo name may be different)

5. Backup current install directory

   # mv install install.7.17.6

6. Copy oadcgs-es-elastic-reposerver- X.X.X.X.tar.gz to repo server
7. Uncompress the new install directory

   # tar -zxf oadcgs-es-elastic-reposerver- X.X.X.X.tar.gz --strip-components=1

8. Correct permissions

   You should be in the /var/www/html/yum/elastic directory before executing the following.

   # chown -R apache:apache install
   # chmod -R ugo+rx install
   # restorecon -R *
   # ls -ltrZ

The new install directory and its contents should now be ready for use.

5.5.2 Update Elastic Puppet Modules

NOTE: A Puppet administrator is required to execute this section.

Puppet modules are used to automate some of the configuration on Linux hosts and the installation of some Elastic components. Both puppet modules used for Elastic must be updated for this upgrade. A Puppet SME should be involved in updating these modules and ensuring Puppet is configured properly for Elastic.

5.5.2.1 Elastic Servers – dsil_elastic_servers Module

NOTE: A Puppet administrator is required to execute this section.

A puppet administrator needs to update the dsil_elastic_servers module with the new module delivered in oadcgs-dsil_elastic_server.X.X.X.X.tar.gz and re-deploy the module to all puppet environments.

Updates included in upgrade include:

- Addition of esp_filebeat-singleworker pipeline to default logstash.yml configuration.
- Additions to windows ids dictionary
- Update to get_ldap_hosts.sh script to not update icmp list if nothing is returned from AD
- kibana.yml updated to use environment variables for Elastic hostnames

To update by just copying in the changed files use the following as a guide:

Files changed for this upgrade:

- metadata.json
- data/logstash.yaml
- files/dictionaries/windows_ids.yml
- templates/get_ldap_hosts.sh.epp

- templates/kibana.yml.epp

IMPORTANT: Before deploying these updates the "esp_filebeat-singleworker" pipeline should be added to any entries in the "node_specific" directory. These entries are not delivered as part of the baseline and must be copied from the existing repo as they are specific to each environment. The files in the "node_specific" directory must be populated with the correct Logstash pipeline configuration for each site.

Before creating an updated tag and updating the Puppetfile in the pe-control-repo to start using the updated dsil_elastic_servers module you must first create a node specific configuration file for any site that needs to run additional pipelines that are not contained in the base set specified in the default configuration.

If a specific configuration is not given for a site, it will use the default configuration supplied in the "data/logstash.yml" file supplied with the baseline.

Default pipelines: That default configuration will run the following pipelines that are expected to be run at each site.

- esp_filebeat
- esp_filebeat-logstash
- **esp_filebeat-singleworker (New in this version)**
- esp_heartbeat
- esp_linux_syslog
- esp_loginsight
- esp_metricbeat
- esp_winlogbeat

The "data/logstash.yml file" contains the puppet variable "dsil_elastic_servers::logstash::pipelines" containing the above default list of pipelines. The easiest and recommended way to create a node specific configuration file is to copy this file to use as a template.

Example of variable definition from file:

```
dsil_elastic_servers::logstash::pipelines: '["esp_filebeat", "esp_filebeat-
logstash", "esp_heartbeat", "esp_linux_syslog", "esp_loginsight",
"esp_metricbeat", "esp_winlogbeat", "esp_filebeat-singleworker"]'
```

Additional pipelines: The following pipelines should only be run at sites where the datatype is available for ingest:

- esp_eracent_database
- esp_hbss_epo
- esp_hbss_metrics
- esp_idm_database

- esp_postgres
- esp_puppet_database
- esp_sccm_database
- esp_serena_database
- esp_sqlServer_stats

Directory structure of dsil_elastic_servers repository:

dsil_elastic_servers/data:
logstash.yml – contains pipelines that should run at all sites

dsil_elastic_servers/data/node_specific:
CXXsu01ls01.yml (Add one for each site with additional pipelines)

NOTE: When setting up the node specific configuration files for the first time, it is recommended that you use the existing logstash.yml file at each site as a guide to populate the node specific file.

Steps to create a custom node specific configuration for each site on the enclave

8. Login to the Logstash instance at the site CXXsu01ls01
    a. C = Classifier ('u', 's' or 't')
    b. XX = site number

9. cd /etc/logstash and view the current logstash.yml file

    # cd /etc/logstash
    # cat logstash.yml

10. Examine the current list of pipelines being run at the site by looking at the xpack.management.pipeline.id array.
11. If the list contains only the default pipelines than a node specific configuration file is not needed for this site; continue onto the next site
12. You have identified a site that needs a node specific configuration. Copy "data/logstash.yml" to "node_specific/CXXsu01ls01.yml" configuration file

    Example: cp logstash.yml node_specific/s00su01ls01.yml

13. Update the "dsil_elastic_servers::logstash::pipelines" array in the newly created node specific file to contain the same pipelines that are currently running at the site.

    NOTE: The site should be running the default configuration with the possibility of additional pipelines. If any of the default pipelines were not in the original xpack.management.pipeline.id array, then they should be added.
14. Continue onto the next site

Once you have created node specific configuration files for any sites that are running additional pipelines, you can create a new tag for the dsil_elastic_servers repo and update the Puppetfile in the pe-control-repo to start using the updated dsil_elastic_servers module.

5.5.2.2 Elastic Clients – dsil_elastic_clients Module

NOTE: A Puppet administrator is required to execute this section.

A puppet administrator needs to update the dsil_elastic_clients module with the new module delivered in oadcgs-dsil_elastic_clients.X.X.X.X.tar.gz and re-deploy the module to all puppet environments.

Updates included in upgrade include:

- Metricbeat does not ingest process information for system processes
- Addition of Metricbeat docker module on hosts where docker is running
- Added rsyslog to default processes being monitored on all Linux hosts
- Module refactored to allow easier integration with ARTs
- Auto Filebeat install/configuration on any host running SOAESB
- Filebeat.yml has been updated to use generic configuration. Most inputs are now configured using files in the inputs.d directory

After installing new module be sure to create a new tag to update in Puppetfile of pe-control-repo to deploy and start using updated module.

IMPORTANT: Be sure to review the templates and update any system specific information for hosts. MTE and CTE may have additional changes as most configuration files are tailored for the production system. Also, if there were any additional templates added that are not part of the official baseline be sure to include those in the module before pushing.

5.5.2.3 Update Puppetfile

The Puppetfile must be updated with the Elastic modules for them to be included in the branch when it is pushed. After updating each module, a new tag must be assigned and then updated in each branch's Puppetfile.

On each puppet branch, edit the **<branch>/Puppetfile** and lines for each of the Elastic modules.

```
mod 'dsil_elastic_clients',
    :git => 'git@{site code}su01pup1.ech.dcgs.mil:dsil_elastic_clients.git',
    :tag => 'v1.1.XX'
mod 'dsil_elastic_servers',
    :git => 'git@{site code}su01pup1.ech.dcgs.mil:dsil_elastic_servers.git',
    :tag => 'v1.1.XX'
```

The value for **:git =>** is installation dependent; copy the value from another module in the file.

The **:tag** value is also dependent on the system and can be obtained by running the following command in the modules branch:

```
# git tag -l
```

5.5.3 Upgrade Elasticsearch components

Before continuing, back up any visuals/dashboards that have been developed that are not part of prior Enterprise Services releases. Use the **Saved Objects** interface in Kibana to export anything you would like to back up. Note that this upgrade will not delete anything from Elastic/Kibana but some dashboard/visuals will be updated; the backup is optional but may be done to ensure that any work done in Elastic is preserved in case there is an issue with the upgrade.

IMPORTANT: Do not use VM snapshot/restore with Elastic Nodes. This may cause issues with the Elastic Cluster. If you are unsure if or how to back up items in Elastic, please consult an Elasticsearch SME.

The components of your Elastic Stack will be upgraded in the following order:

1. Elasticsearch
2. Kibana
3. Logstash
4. Beats

The following steps require that the admin have **root** permissions to perform the install. The **#** at the beginning of a command signifies that it should be run as root. If you don't know how to become root on a Linux machine, you should not be performing this installation.

IMPORTANT: At this point the new oadcgs-es-elastic-reposerver-X.X.X.X.tar.gz for this upgrade must have already been installed on the repo server. If this has not been completed, please refer to section 4.3 for the correct version and destination.

5.5.3.1 Update Repo with New Core RPMs

NOTE: A Linux administrator will be needed to execute this section.

1. Before executing the upgrade instructions, the RPMs for the new Elastic_Core_ Components-8.x.x.gz must be copied to a tmp folder on the DCGS repo server (ex: u00su01ro0).
2. Extract the gzip using the following command:

   ```
   tar -zxf Elastic_Core_Components-8.6.2.tar.gz
   ```

3. **Copy** the new RPMs for the following to the Elastic repo (/var/www/html/yum/elastic):
   - elasticsearch-8.x.x-x86_64.rpm
   - kibana-8.x.x-x86_64.rpm
   - logstash-8.x.x-x86_64.rpm
4. Ensure RPMs have the correct owner/group:
   ```
   # chown -R apache:apache *
   ```

5. Repo files must have selinux context **httpd_sys_content_t** set. If you copied the RPMs into the directory, they will automatically have this context set. If you moved them, they won't. Ensure all files have the correct context set by executing:

```
# ls -lZ
```

Figure 80 Example of correct file permission in the Elastic repo

6. If all files do not have **httpd_sys_context_t** set, execute the following:
   ```
   # restorecon *
   ```

7. Recreate the Elastic repo so it's ready for use:
   ```
   # createrepo ./
   # gpg --detach-sign --armor ./repodata/repomd.xml
   ```

   NOTE: There are 2 dashes in front of **detach-sign** and **armor** in the previous command.

7. Confirm overwriting the file (if it already exists). Enter **y** to overwrite.

5.5.3.2 Prepare for Elasticsearch Node Upgrades

The Elastic cluster must be Healthy ("Health is green") before starting the cluster upgrade.

1. Open your favorite web browser and navigate to the following url: https://kibana.
2. Log in to Kibana using your privileged AD account (.wks, .adm, or .dba).

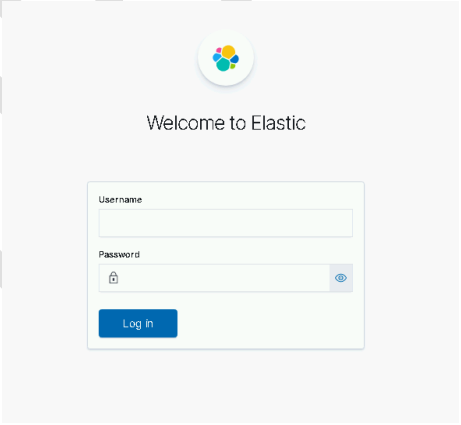


Figure 81 Login Screen

3. After successful login, you will be asked to select a workspace. Select **Default**.

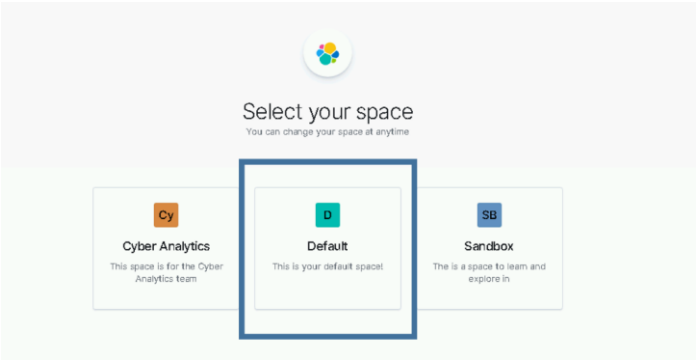


Figure 82 Select Default Workspace

4. Using the Kibana hamburger menu, select **Stack Monitoring**.

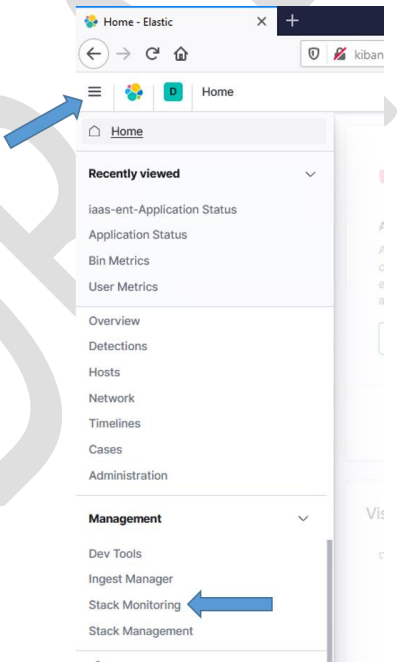


Figure 83 Navigate to Stack Monitoring

5. The **Stack Monitoring** dashboard displays. Verify the cluster is Healthy. If the Health is not "green", **STOP**, fix the issues with the cluster to bring it back to "green" before proceeding. If you do not know how to restore the cluster's health, please consult with an Elastic SME to return the cluster back to "green" status.

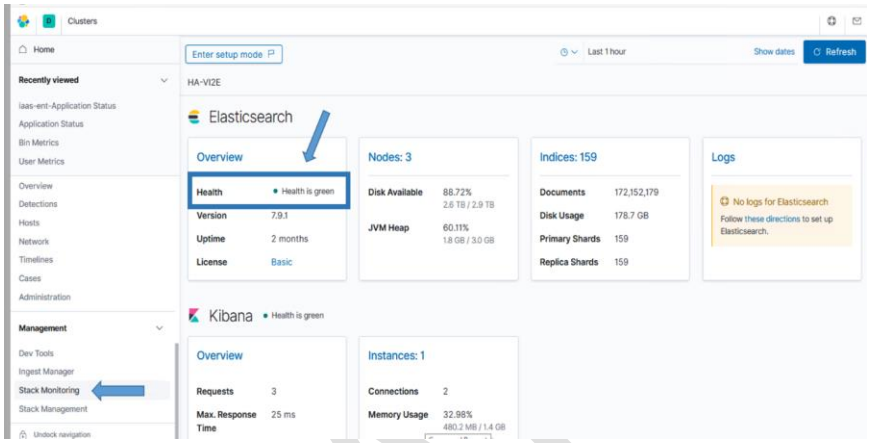


Figure 84 Health Status Should Be Green

5.5.3.3 Elasticsearch

NOTE: An Elasticsearch administrator will be needed to execute this section.

A Rolling Upgrade will be performed on the Elastic cluster. A rolling upgrade allows an Elasticsearch cluster to be upgraded one node at a time so upgrading does not interrupt service. Running multiple versions of Elasticsearch in the same cluster beyond the duration of an upgrade is not supported, as shards cannot be replicated from upgraded nodes to nodes running the older version.

The nodes of the cluster will be updated in the following order.

NOTE: In a 3-node cluster all nodes are Master-eligible.

1. Nodes that are not master-eligible. You can retrieve a list of these nodes with **GET /_nodes/_all,master:false** or by finding all the nodes configured with **node.master: false**.
2. Master-eligible nodes, which are the remaining nodes. You can retrieve a list of these nodes with **GET /_nodes/master:true**.

   NOTE: These commands can be executed from Dev Console in Kibana.

Upgrading the nodes in this order ensures that the master-ineligible nodes are always running a version at least as new as the master-eligible nodes. Newer nodes can always join a cluster with an older master, but older nodes cannot always join a cluster with a newer master. By upgrading the master-eligible nodes last,

you ensure that all the master-ineligible nodes will be able to join the cluster whether the master-eligible nodes have been upgraded or not. If you upgrade any master-eligible nodes before the master-ineligible nodes, then there is a risk that the older nodes will leave the cluster and will not be able to rejoin until they have been upgraded.

To make things easier, the following table is provided as a guide for upgrade order of the nodes in each DCGS cluster.

Ping each server's DNS alias to ensure it exists; this will mitigate future problems as scripts use the elastic-node-* and <server>.<fqdn> aliases.

Table 12 Upgrade Order

Order	3 Node Cluster	10 Node Cluster	15 Node Cluster
1	elastic-node-1	elastic-node-4	elastic-node-4
2	elastic-node-2	elastic-node-5	elastic-node-5
3	elastic-node-3	elastic-node-6	elastic-node-6
4		elastic-node-7	elastic-node-7
5		elastic-node-8	elastic-node-8
6		elastic-node-9	elastic-node-9
7		elastic-node-10	elastic-node-10
8		elastic-node-1	elastic-node-11
9		elastic-node-2	elastic-node-12
10		elastic-node-3	elastic-node-13
11			elastic-node-14
12			elastic-node-15
13			elastic-node-1
14			elastic-node-2
15			elastic-node-3

5.5.3.3.1 Upgrade Each Elasticsearch Node

Upgrade each node using the order in the previous table.

IMPORTANT: The user logging into each Elastic node doing the upgrade must be a member of the **ent elastic admins** AD group to have the correct permission in Elasticsearch to upgrade the node. Having the **Elastic Administrator** OneIM Role will place the user in this group. If the user is not a member of this group, **STOP** and either add them to the group or find a user who is already in the group to do the upgrade.

1 **Assumption**: Elastic RPMs and installation scripts have been added to Elastic repo at all applicable sites and each repository is functional. A good check before running the upgrade script on each node is to verify yum can see the Elasticsearch package for the version you are upgrading to.

Before upgrading the node ensure that the rpmverify.exclude file has the correct excludes:

```
# cd /etc
# cat rpmverify.exclude
```

verify that it contains the following lines:

elasticsearch
kibana
logstash

If the lines are not present, contact a puppet SME and have them add the lines to the **rpmverify:exclude_packages** section of the osif.yaml file.

Ensure the following command works properly before executing the upgrade_node.sh script:

```
# yum repo-pkgs <elastic repo name> list
```

Example: yum repo-pkgs *elastic* list

NOTE: The elastic repo name is the name given to the Elasticsearch repository on the repo server. In most cases it is just **elastic** but if you're not sure you can check by executing **yum repolist all** to show all the repositories available.

NOTE: You can verify the path to the Elastic repository by checking the repo definition found in **/etc/yum.repos.d/elastic-search-rpms.repo** (the name of the repo may differ).

Ensure the **Elasticsearch.x86_64** package is the version you are upgrading to.

After ensuring the node can read the upgrade package from the repo server, execute the following script to perform the upgrade on the node:

```
# curl -s -k https://{site code}su01ro01.`hostname -
d`/yum/elastic/install/upgrade_node.sh | bash
```

NOTE: The back quote characters (`) used in the previous command are on the keyboard key with the tilde (~).

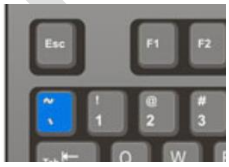


Figure 85 Back Quote Characters (`)

You will be prompted for your password so the script can use it, along with your user ID, to update settings in Elasticsearch during the upgrade. The password is only used during script execution and is NOT saved.

The script will:

- Disable allocation of replicas in Elastic.

- Halt machine learning jobs to allow for the upgrade.
- Stop the node you're upgrading.
- Upgrade the node then restart it.
- Wait for the cluster to return to 100% Healthy.

Once the script is complete, you will see the message **Upgrade for this node complete, continue with next node**. If you do not see this message, or you see any errors, you should stop and contact an Elastic SME for guidance.

Repeat for all nodes in the cluster.

5.5.3.3.2 Verify Upgrade Versions

After upgrading all nodes, you should verify the version of each node. The easiest way to do this is using the Kibana Dev Tools console.

1. Log in to Kibana with your .wks or .adm account and navigate to the **Dev Tools** panel.
2. Click the hamburger menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.
3. Scroll to **Management** at the bottom and select **Dev Tools**.

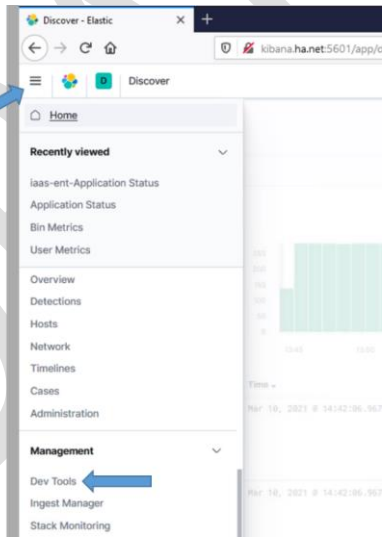


Figure 86 Select Dev Tools

4. The Dev Tools console displays. To check the node versions, execute the following command:

```
GET _cat/nodes?v&h=name,ip,version&s=name
```

Execute the command by pressing <ctrl><enter> or selecting the execute icon on the right of the entered command.

5. You will see all nodes listed with their versions, as in the following example:

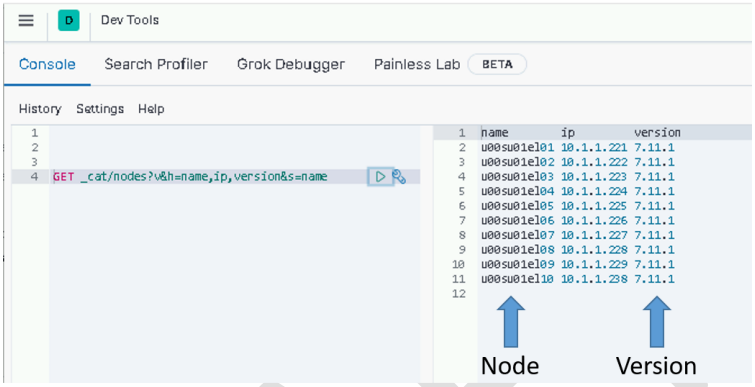


Figure 87 Check node versions example

NOTE: If all the nodes are not upgraded, go back, and upgrade the ones you missed. If you believe a node should be upgraded and it's not, consult with an Elastic SME before proceeding.

5.5.3.3.3 Complete Cluster Upgrade

During the execution of the upgrade_node.sh in section 5.5.3.3.1, all machine learning jobs were stopped for the upgrade to occur. Now that the cluster is upgraded, we need to re-enable machine learning jobs. The easiest way to do this is using the Kibana Dev Tools console. To access the console:

1. Log in to Kibana with your .wks or .adm account and navigate to the **Dev Tools** panel.
2. Click the hamburger menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.

3. Scroll to **Management** at the bottom and select **Dev Tools**.

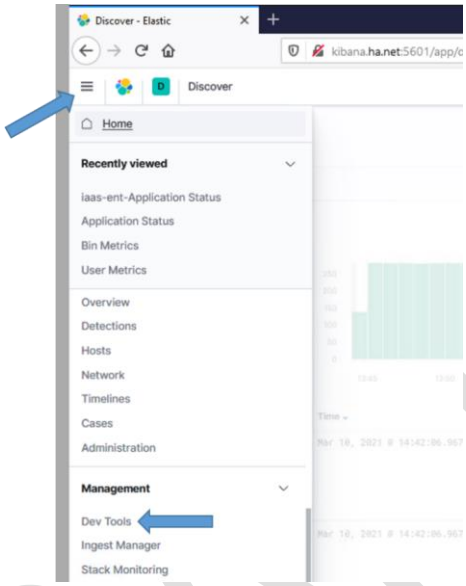


Figure 88 Select Dev Tools

4. The Dev Tools console displays.
5. To re-enable machine learning jobs, execute the following command:

```
POST _ml/set_upgrade_mode?enabled=false
```

6. You can verify that upgrade_mode is disabled by executing the following command and examining the output.

```
GET _ml/info
```

7. Scroll down or collapse fields, clicking on the down arrows to find the **upgrade_mode** field and verify it is set to false.

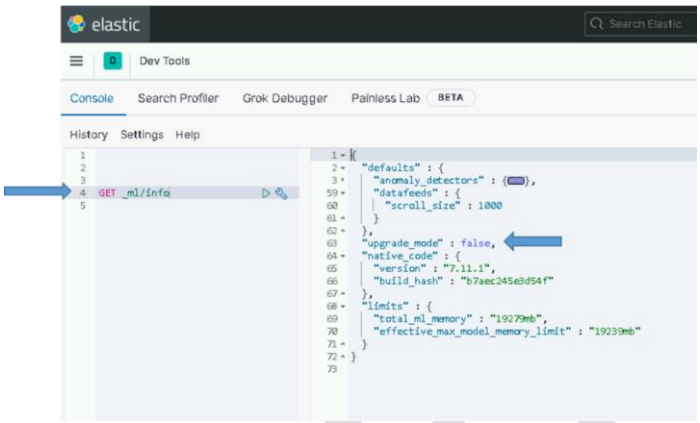


Figure 89 Verify ml upgrade_mode is false

5.5.3.4 Upgrade Kibana

The following steps require that the admin have **root** permissions to perform the install. The **#** at the beginning of a command signifies that it should be run as root. If you don't know how to become root on a Linux machine, you should not be performing this installation.

Use this table to determine which Elastic nodes to upgrade Kibana on.

Table 13 Elastic nodes to upgrade Kibana on

# of Nodes in Cluster	Elastic Nodes where Kibana is installed
3	Node 3
10	Node 7 and Node 10
15	Node 10 and Node 15

5.5.3.4.1 Upgrade Kibana Instance

NOTE: An Elasticsearch administrator will be needed to execute this section.

Different versions of Kibana running against the same Elasticsearch index, such as during a rolling upgrade, can cause data loss. This is because older instances will continue to write saved objects in a different format than the newer instances. To prevent this from happening ensure that all old Kibana instances are shutdown before starting up instances on a newer version.

Log into each Elastic node that is running Kibana and become root.

```
# sudo su
```

Puppet now controls the kibana.yml file so first disable puppet to ensure it doesn't interfere with the upgrade

puppet agent –disable

Now stop Kibana on the server

```
# systemctl stop kibana
```

Ensure Kibana is not running

```
# systemctl status kibana
```

NOTE: The following steps can be run on all Kibana nodes at the same time after you have verified that the Kibana service has stopped on all Kibana nodes.

NOTE: Kibana can take up to 45 minutes to upgrade. To avoid interruption of the upgrade, the screen command will be used to create a session to run the install command. For more information about the screen command, consult the Linux man page for **screen**.

```
# screen –S install-session
# yum upgrade kibana
```

*Ensure upgrade version is correct and type **Y**. Press **Enter**.*

NOTES:

- ***You can start the upgrade on both instances to* reduce the upgrade time.**
- **If your SSH session times out while waiting for Kibana to be installed, return to your install-session by typing the following after re-establishing an SSH session to the computer.**
  # screen –d –r install-session
- **To detach from a running screen session type ctrl+a ctrl+d.**
- **If the Kibana installation is terminated for any reason, STOP, and contact an OADCGS SME for guidance.**

5.5.3.4.2 Update Load Balancer Configuration

In the 8.6 upgrade there is a change to the Kibana status API that affects the currently configured service monitor of the load balancer. The following update must be made to allow the https://kibana url to continue to function properly.

Edit the Service monitor and make the following changes:

- Modify "Expected" from 204 to 200
- Remove "green" from the "Receive" field

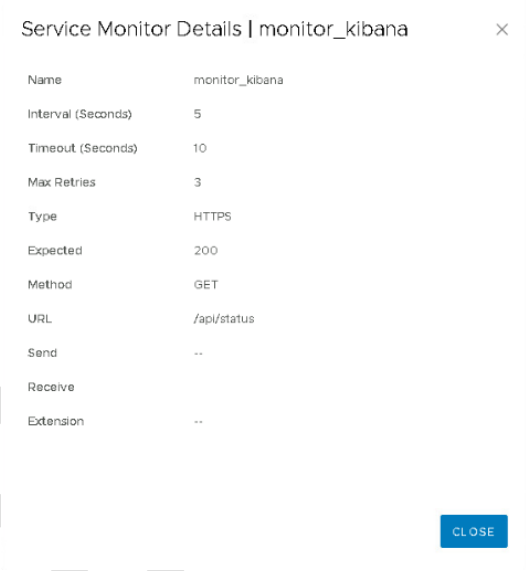


Figure 90- Updated Service Monitor configuration

5.5.3.4.3 Start & Test Kibana

This test will access this instance of Kibana by explicitly specifying the name of the VM where it is installed in the URL. General user access to this instance should be controlled by the NSX load balancer using the https://kibana URL once it's configured.

Do the following once all Kibana nodes have been upgraded:

1. Run daemon reload to ensure systemd is up to date:
   # systemctl daemon-reload
2. Re-enable puppet and run manually to ensure kibana.yml is up to date.
   # puppet agent –enable
   # puppet agent -t

3. Start Kibana:
   ```
   # systemctl start kibana
   ```

   Note: The puppet run should have already started Kibana, this step is to just ensure it's being started.

4. Give Kibana a few minutes to come up and connect to Elastic.
   NOTE: The time before you can access the Kibana web page can vary between versions. On the initial startup after an upgrade, Kibana may take additional time to be ready as it does housekeeping for the new version. We have seen this take up to an hour on some versions.
5. If Kibana starts with no issue, test Kibana from any computer that has network access to the Kibana node. Open your favorite web browser and navigate to the following URL:

   ```
   # https://elastic-node-{x}:5601   (example: https://elastic-node-10:5601)
   ```

   If it loads to a Kibana login window, success!

6. You can now log in to Kibana using **your privileged AD account (.wks, .adm)** for checking things during the remainder of this upgrade.

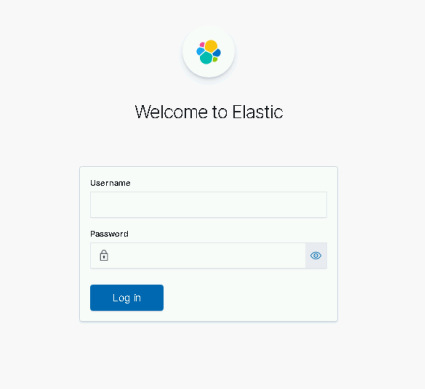


Figure 91 Login Screen example

7. Select the **Default Space** and then select **Discover** from the hamburger menu as described in section 5.5.3.3.2
8. Select any index pattern to verify you can view data.

5.5.4 Update Elastic Search Configurations

NOTE: An Elasticsearch administrator will be needed to execute this section.

5.5.4.1 Update Roles

The security features provide a role-based access control (RBAC) mechanism, which enables you to authorize users by assigning privileges to roles and assigning roles to users or groups These roles are mapped to DCGS Active Directory groups to provide access controls to data types.

5.5.4.1.1 Load Kibana Roles

To update Kibana roles for the cluster run the following command as root from any of the running Elastic nodes.

```
# curl –k https://xxxsu01ro01.`hostname –d`/yum/elastic/install/load_roles.sh | bash
```

NOTE: These settings are dynamically applied, and no restarts are necessary. Running this script multiple times will continue to update the roles and is not harmful.

5.5.4.1.2 Verify Kibana Roles are Loaded

To verify the Kibana roles were successfully loaded:

6. Click the menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.
7. Scroll to **Management** at the bottom and select **Stack Management**.
8. The **Stack Management** page displays. Under **Security** select **Roles** on the left side.
9. Enter **dcgs** in the search bar to see the 5 roles that should be loaded.

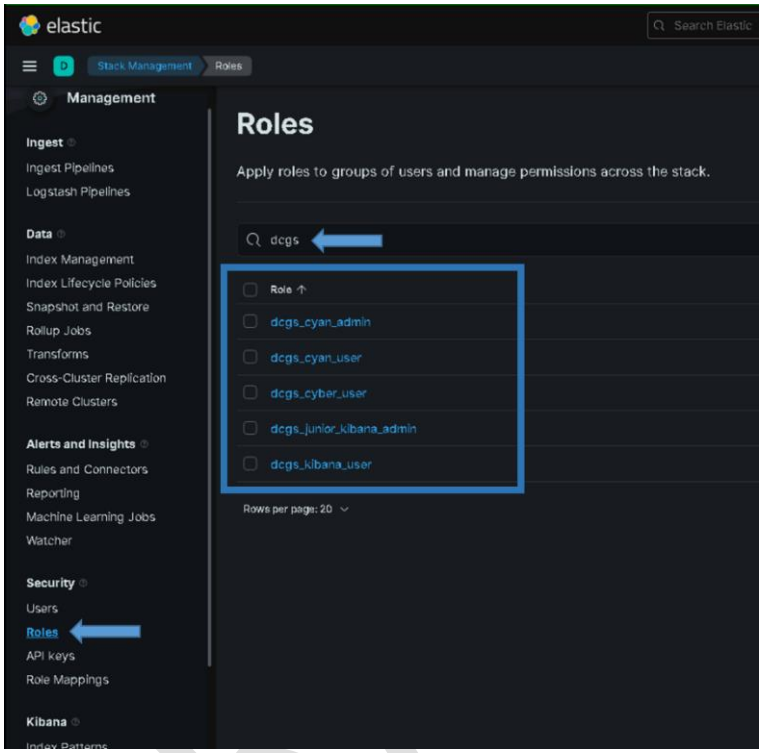


Figure 92 Roles

5.5.4.2 Update Ingest Pipelines in Elasticsearch

Elasticsearch ingest pipelines are used to aid ingest of data into Elasticsearch. Many Filebeat and Winlogbeat modules have associated ingest pipelines. These pipelines are not loaded into Elasticsearch automatically; they must be loaded each time you install or upgrade beats. Ingest pipelines only need to be loaded one time for use with all beat instances. To make the loading of the ingest pipelines easy, a convenience script has been written to load the pipelines. This script **MUST** be run each time beats are upgraded on the system.

To load the ingest pipelines, run the following command as root from any of the running Elastic nodes:

```
# curl –k https://{site code}su01ro01.`hostname –d`/yum/elastic/install/update_ingest_pipelines.sh | bash
```

2. Verify the ingest pipelines are loaded in Elastic. Select the **Ingest Pipelines** page under **Stack Management** to view all the ingest pipelines loaded into Elasticsearch. Filter the page with the version number you are installing to see the ingest pipelines for that specific version.

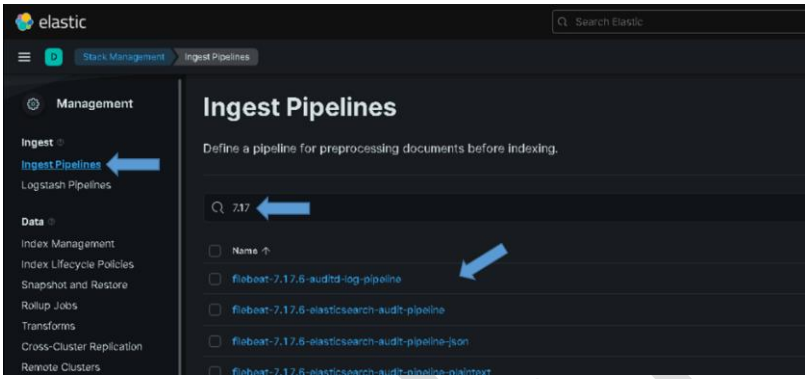


Figure 93 Example of Ingest Pipelines for version 7.17

For version 8.6.2 you should see the following pipelines:

- filebeat-8.6.2-auditd-log-pipeline
- filebeat-8.6.2-elasticsearch-audit-pipeline
- filebeat-8.6.2-elasticsearch-deprecation-pipeline
- filebeat-8.6.2-elasticsearch-gc-pipeline
- filebeat-8.6.2-elasticsearch-server-pipeline
- filebeat-8.6.2-elasticsearch-slowlog-pipeline
- filebeat-8.6.2-iptables-log-pipeline
- filebeat-8.6.2-logstash-log-pipeline
- filebeat-8.6.2-logstash-slowlog-pipeline
- filebeat-8.6.2-system-auth-pipeline
- filebeat-8.6.2-system-syslog-pipeline
- filebeat-8.6.2-elasticsearch-audit-pipeline-json
- filebeat-8.6.2-elasticsearch-deprecation-pipeline-json
- filebeat-8.6.2-elasticsearch-server-pipeline-json
- filebeat-8.6.2-elasticsearch-slowlog-pipeline-json
- filebeat-8.6.2-logstash-log-pipeline-json
- filebeat-8.6.2-logstash-slowlog-pipeline-json
- filebeat-8.6.2-elasticsearch-audit-pipeline-plaintext
- filebeat-8.6.2-elasticsearch-deprecation-pipeline-plaintext
- filebeat-8.6.2-elasticsearch-server-pipeline-plaintext
- filebeat-8.6.2-elasticsearch-slowlog-pipeline-plaintext
- filebeat-8.6.2-logstash-log-pipeline-plaintext
- filebeat-8.6.2-logstash-slowlog-pipeline-plaintext
- Winlogbeat-8.6.2-powershell
- winlogbeat-8.6.2-powershell_operational
- winlogbeat-8.6.2-routing
- winlogbeat-8.6.2-security
- winlogbeat-8.6.2-sysmon

5.5.4.3 Update Templates

After the cluster has been installed/upgraded and is running, the templates needed to ingest data properly must be updated. The templates are located in the **templates** folder of the **install** directory of the Elastic Repo.

In this section all index and component templates will be added to Elasticsearch. The following naming conventions are used for Enterprise Service templates in Elasticsearch.

Index templates – esti_<template name>
Component templates – estc_<template name>

IMPORTANT: DO THIS BEFORE UPGRADING ANY BEATS COLLECTORS OR UPGRADING ANY LOGSTASH INSTANCES.

4. Run the following command as root from any of the running Elastic nodes to update the templates.

   # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/load_templates.sh | bash

5. After loading the templates, they can be verified (sorted by name) by executing the following command from the Kibana Dev Tools console.

   ```
   GET _cat/templates/esti*?v&s=name
   ```

6. The following index templates should be loaded by this script:
   - esti_catalyst
   - esti_datadomain
   - esti_db_postgres
   - esti_eracent
   - esti_fc6xx
   - esti_filebeat-{version}
   - esti_fx2
   - esti_hbss-epo
   - esti_hbss-metrics
   - esti_healthdata
   - esti_current-healthdata
   - esti_heartbeat-{version}
   - esti_idm
   - esti_iptables
   - esti_isilon
   - esti_nexus5k

- esti_nexus7k
- esti_puppet
- esti_r6xx
- esti_sccmdb
- esti_serena
- esti_render
- esti_soaesb
- esti_sqlserver
- esti_vsphere
- esti_winlogbeat-{version}
- esti_xtreamio
- esti_acas
- esti_socketgxp
- esti_gxpxplorer
- esti_maas_logs

NOTE: esti_metricbeat-{version}-{site}, esti_audits_syslog-{site} and esti_syslog-{site} index templates are generated dynamically later in the installation process.

NOTE: There may be other index templates, but the above templates should all exist after running the load_templates script above.

NOTES:

- All Enterprise Service index templates prefixed with "esti_" and the {version} in the previously listed names will be replaced with the current version of the beat being installed.
- If the templates are not loaded, **STOP**, and contact an OADCGS Elastic SME for guidance.

You can also use the **Index Management** interface in Kibana to manage Index Templates, Component Templates, and Legacy Templates.

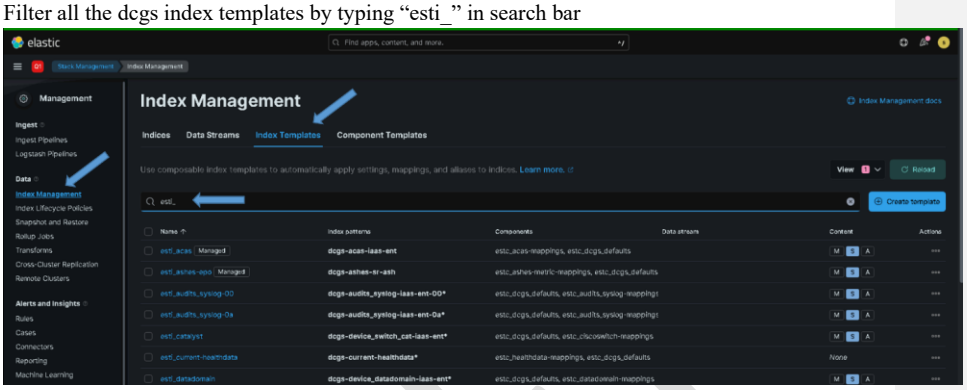
The index templates for site specific indexes will be loaded during each Logstash upgrade.

5.5.4.4 Update Component Template Ordering

This version updates the ordering of the component templates in all index templates to place the estc_dcgs_default template first. This has been done to allow the creation of a "default_field" used for searching for any index that does not have one. To allow this new field to be overridden by component templates that already define the "default_field" the initial definition by the estc_dcgs_default component template must come first. Beat templates that are supplied from Elastic define their own default search fields which should be used.

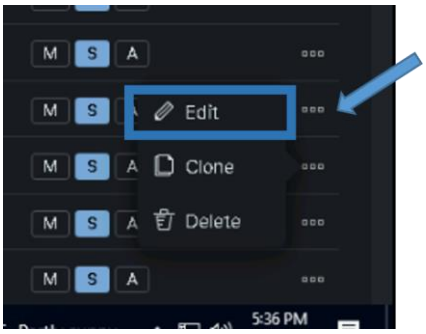
After the upgrade older index templates may not have been updated with this change and must be updated manually following the instructions below.

1. Navigate to the “Index Templates” tab of “Index Management” in Kibana
    a. “Hamburger Menu” -> “Stack Management” -> “Index Management” Then select the “Index Templates” tab
2. Filter all the dcgs index templates by typing “esti_” in search bar

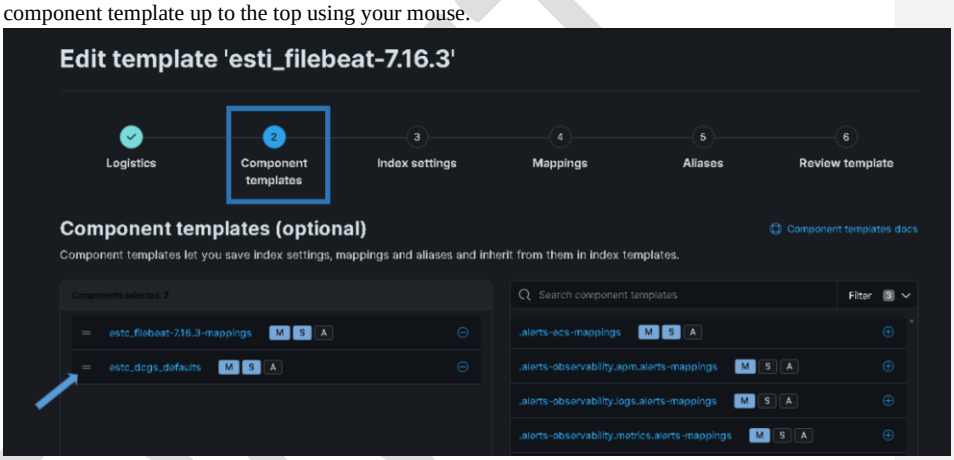


3. Look at the “Components” column and identify any index template that contains the “estc_dcgbs_defaults” component template not listed as the first component template.

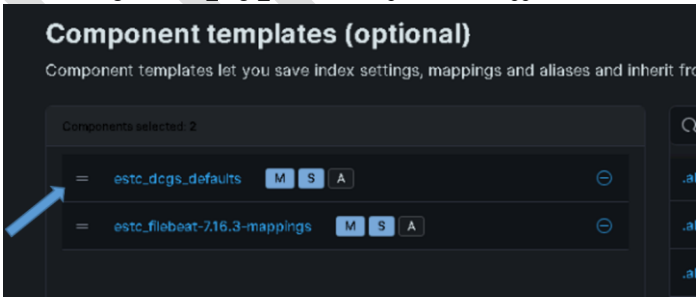
4. Edit each of the index templates that contain the estc_dcgs_defaults component template but don’t have it listed first.



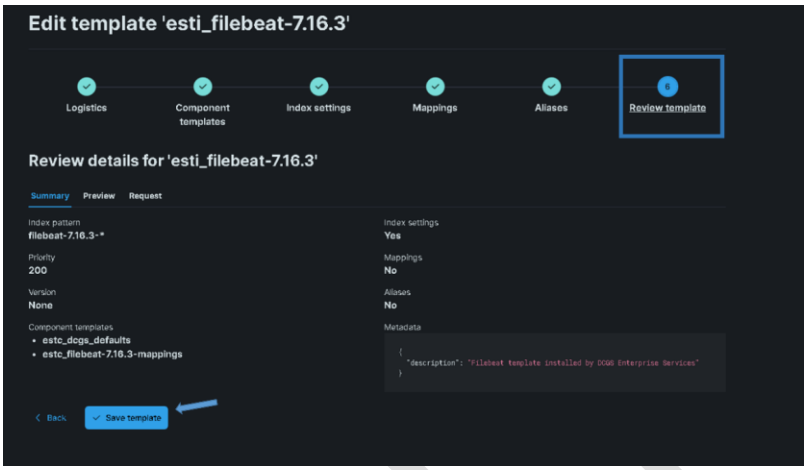
5. Select number 2 “Component templates” in the edit view and drag the “estc_dcgs_defaults” component template up to the top using your mouse.



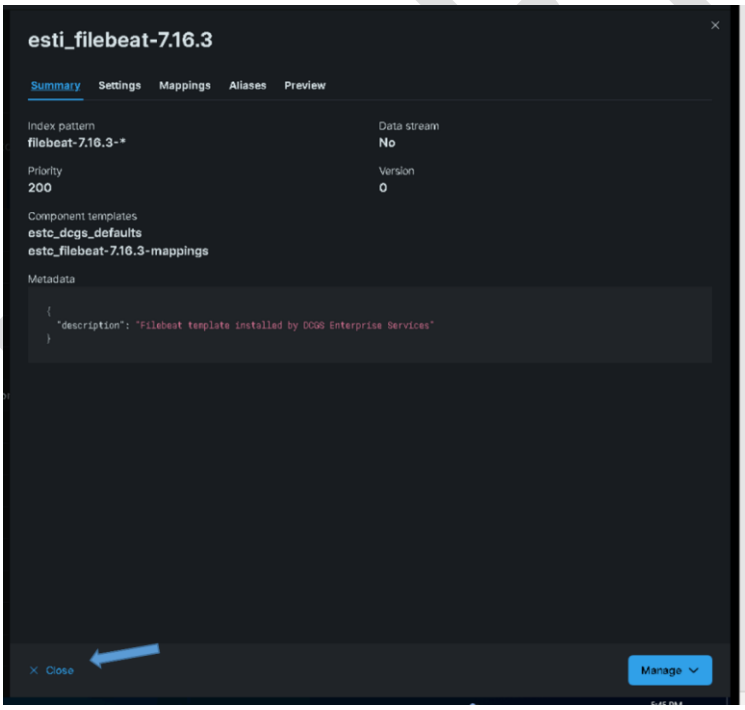
6. After moving the “estc_dcgs_defaults” template should appear first in the list.



7. Select step 6 “Review template” and then “Save template”



8. Select “Close”



9. Repeat from step 2 for all index templates that need updated

Deleted: ¶

Formatted: List Paragraph, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

5.5.4.5 Bootstrap Indexes

To make sure Elastic is ready to receive data from the upgraded beats and any new indexes, you need to bootstrap an initial index and designate it as the write index for the rollover alias specified in the new index templates. The name of this index must match the template's index pattern and end with a number. Each index template has a rollover_alias specified for this purpose. On rollover, this value is incremented to generate a name for the new index.

IMPORTANT: In version 7.17 there are three site-base indexes; "metricbeat", "dcgs-syslog-iaas-ent" and "dcgs-audits_syslog-iaas-ent". This means that there will be one alias per site for these indexes. These aliases are bootstrapped during the upgrade to Logstash at each site later in the process. You will not see an alias these 3 indexes after this step is complete.

5. Run the following command as root from any of the running Elastic nodes to bootstrap the initial write indexes for the Elastic data types:

   # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/bootstrap_indexes.sh | bash

   NOTE: This script will only bootstrap indexes that do not currently have an alias configured. Running this script more than one time causes no harm.

6. To verify beats indexes have bootstrapped and have a write index, execute the following command from the Kibana Dev Tools console, which sorts them by name:

   ```
   GET _cat/aliases/*beat-{version}*?v&s=alias
   ```

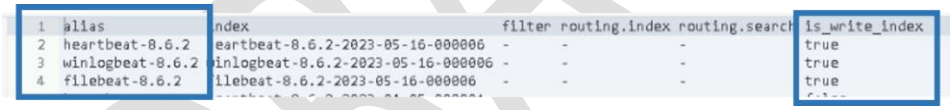


*Figure 94 GET _cat/aliases/*beat-{version}>*?v&s=alias output*

7. All data types have been bootstrapped successfully. If there are no aliases listed for the version you are installing, or none have the **is_write_index** set to **true**, consult with an OADCGS Elastic SME for guidance.

5.5.4.6 Update Enterprise Services Centralized Logstash Pipelines

The deployment of Elasticsearch as a service includes the collection of multiple datatypes. The ingest pipelines for these datatypes must be updated before configuring any Logstash instances to ingest any new data. Perform the following to update the Enterprise Services ingest pipelines.

WARNING: All pipelines will be overwritten; there have been minor changes in the filtering section of most. Clone any pipelines that have been changed since or updated since the 7.16.3 upgrade; you will have to merge the updates back into the baseline pipelines. The following are Pipelines you may want to clone (back up) before running the update script.

- esp_sccm_database
- esp_puppet_database

- esp_eracent_database
- esp_filebeat

These pipelines may have site-specific information in the **input** section.

NOTE: The update_pipelines script communicates with Kibana using the kibana alias, which is set up to route traffic to the NSX load balancer. If you cannot connect to Kibana by typing https://kibana **in your browser, revisit the NSX configuration instructions before proceeding with the installation of Centralized Pipelines. If you cannot configure NSX, consult with an OADCGS Elastic SME for guidance.**

 # sudo su

 # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/update_logstash_pipelines.sh | bash

After running the script, verify the pipelines have been loaded. Once the pipelines are loaded, the Logstash instances can be configured to use them.

NOTE: The pipelines only need to be loaded one time, not once for each Logstash instance.

To verify the pipelines have been loaded, go to **Stack Management in Kibana** and look in the **Pipelines** section.

4. From the hamburger menu, select **Stack Management**.
5. Select **Logstash Pipelines** in the **Ingest** area.
6. The **Pipelines** page displays.

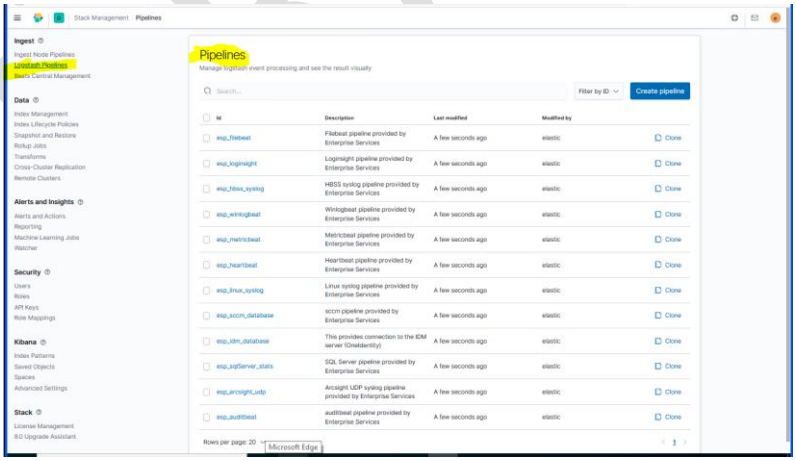


Figure 95 Pipelines

IMPORTANT: After the pipelines have been updated you must manually pull any changes needed from the pipelines you cloned with system-specific information. This information should be merged into the new pipelines and the clones deleted.

IMPORTANT: Loading the pipelines into Elastic makes them available for use by any Logstash Instance but does not automatically add them to any Logstash configuration files. When upgrading Logstash it is important to verify/configure what pipelines are active on each Logstash Instance. Logstash pipeline configurations are now controlled by puppet, see section 5.5.2.1 for details.

5.5.4.7 Install health data watcher

This version brings the ability to detect when a host's status has not been updated for a period of time. When this happens the hosts, status will become "Stale". The "esw_current-healthdata-stale-state" watcher is used to monitor the update times for host data in the "dcgs-current-healthdata-iaas-ent" index.

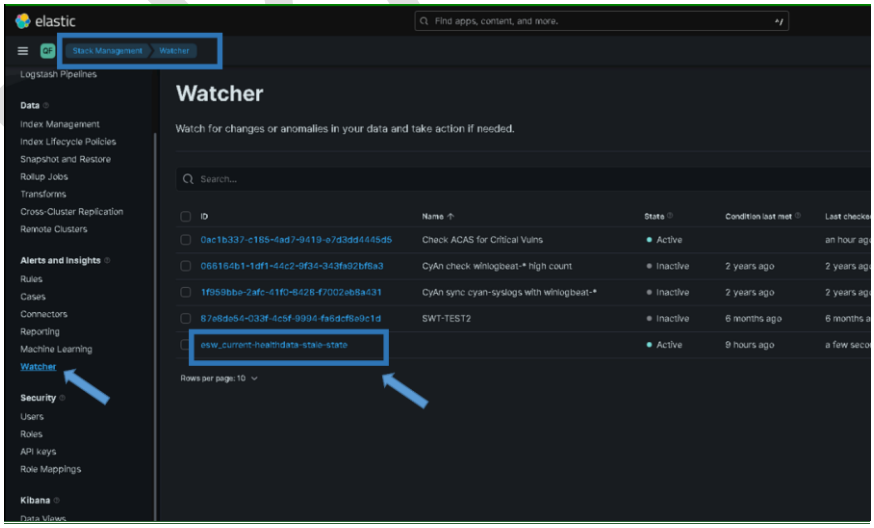
Follow the instructions below to install the watcher:

3. Run the following command as root from any of the running Elastic nodes install the watcher.

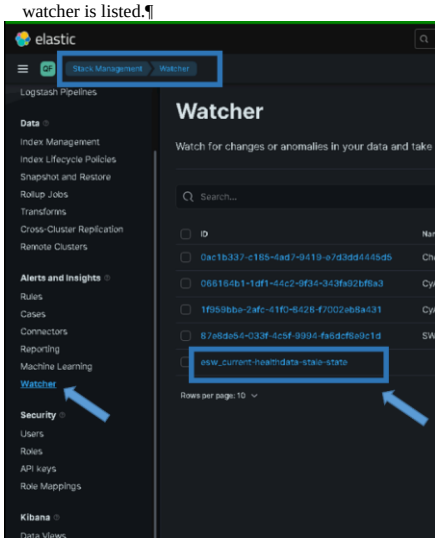
   # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/installWatchers.sh | bash

4. Verify the watcher was loaded correctly.

   - From the hamburger menu, select **Stack Management**.
   - Select **Watcher** in the "Alerts and Insights" section.
   - Validate the "esw_current-healthdata-stale-state" watcher is listed.



Moved down [1]: <#>Install health data watcher¶
This version brings the ability to detect when a host's status has not been updated for a period of time. When this happens the hosts, status will become "Stale". The "esw_current-healthdata-stale-state" watcher is used to monitor the update times for host data in the "dcgs-current-healthdata-iaas-ent" index.¶
Follow the instructions below to install the watcher:¶
<#>Run the following command as root from any of the running Elastic nodes install the watcher.¶
 # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/installWatchers.sh | bash¶
<#>Verify the watcher was loaded correctly.¶
<#>From the hamburger menu, select **Stack Management**. ¶
<#>Select **Watcher** in the "Alerts and Insights" section. '
<#>Validate the "esw_current-healthdata-stale-state" watcher is listed.¶



¶
¶
¶

Moved (insertion) [1]

5.5.5 Upgrade Logstash and ElasticDataCollector (All Instances)

The following sections are to upgrade both Logstash and the ElasticDataCollector at each site. It is preferred to do these upgrades together to reduce the number of logins to each Logstash Instance.

5.5.5.1 Upgrade Logstash

NOTE: You must be root and a member of the **ent elastic admins** AD group to upgrade Logstash. Having the **Elastic Administrator** OneIM Role will place the user in this group.

There is a Logstash server at each site responsible for forwarding the data collected at the site into Elasticsearch. **The following upgrade procedure must be executed on each Logstash instance**. Note that you can use the **Logstash nodes** screen in Kibana to identify and monitor the Logstash instances and versions they are running.

IMPORTANT: Each Logstash server should have a 2nd drive that is 500GB SSD in size (Allocated from the XtremIO), and is mounted as /ELK-local. If this is missing, stop here and correct. **Persistent queues will be enabled in this version and if the /ELK-local drive does not exist or have enough storage capacity ingest will fail.**

1. Select the **Nodes** link in the **Logstash** area of **Cluster overview** page.

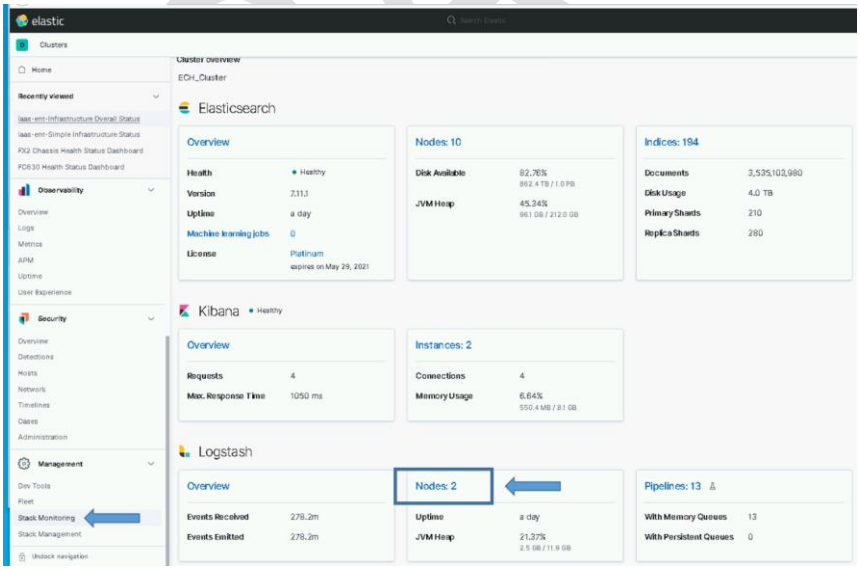


Figure 96- Logstash Node Monitoring Selection

2. The Logstash nodes page will show all the Logstash instances that are feeding data into Elastic and their version.

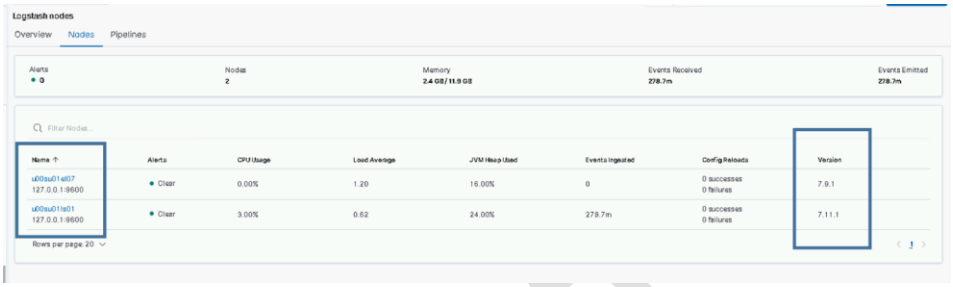


Figure 97 Logstash Nodes example (Note: el07 is running logstash to add a row for the example)

3. Upgrade Each Logstash Instance.
   a. Log in to the **Logstash VM** for each site and become root. This is done on the *{site code}*su01ls01 VM.

      ```
      # sudo su
      ```

   b. Execute upgrade_logstash.sh script.

      NOTE: The script will prompt for your password as your user account will be used to communicate with the Elasticsearch cluster. You will also be asked to enter the site of the Elastic cluster, this defaults to ech but may be different on test enclaves.

      If you are unsure of the cluster site, you can test access to the cluster by using ping:

      # ping elastic-node-1.<site>

      # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/upgrade_logstash.sh | bash

      You will be prompted for the site of the Elastic cluster; this is the site where the Elastic Cluster has been installed for this environment.

      "Enter site of Elastic cluster this Logstash will send to (ex: ech, isec). default [ech] :

   c. When the script finishes you should see the following printed to the screen:

```
Initial metricbeat indexes bootstrapped, log4j fixes, updated jdbc connector and /etc/sysconfig/logstash modified.
Logstash Upgrade Complete.
```

Figure 98 Upgrade Complete

NOTES:

- You can check for error messages in **/var/log/logstash/logstash-plain.log**.
- The pipelines running on each Logstash instance are now controlled by puppet. Ensure the array of pipelines shown for xpack.managment.pipeline.id listed in the **/etc/logstash/logstash.yml** file is correct. You will see a line similar to the following:

  Xpack.management.pipeline.id: ["esp_metricbeat", "esp_winlogbeat", "esp_filebeat", "esp_linux_syslog", "esp_loginsight", esp_heartbeat", esp_filebeat-logstash"]

  Some Logstash instances should be configured to use the following pipelines:

  - esp_eracent_database – Only at hub
  - esp_hbss_epo – Only at hub
  - esp_hbss_syslog – Only at hub
  - esp_sccm_database – Only at hub
  - esp_puppet_database - On the Logstash at the site of the Puppet Master, usually hub
  - esp_sqlServer_stats – Only at hub

5.5.5.2 Upgrade Data Collector

To collect data from Infrastructure devices on DCGS, the elasticDataCollector must be installed/upgraded and configured at every site on each Logstash instance.

NOTE: The Puppet agent will be disabled during the upgrade of the elasticDataCollector to facilitate X11 forwarding to allow running the configurator in 5.5.5.2.3.2.

NOTE: You must be root and a member of the **ent elastic admins** AD group to install the Elastic Data Collector. Having the **Elastic Administrator** OneIM Role will place the user in this group.

NOTE: The same script is used to install or upgrade the elasticDataCollector

3. Log in to the **Logstash VM** for each site and become root. This is done on the *{site code}*su01ls01 VM.

   ```
   # sudo su
   ```

4. Install the Elastic Data Collector.

   ```
   # curl -k https://{site code}su01ro01.`hostname -
   d`/yum/elastic/install/installElasticDataCollector.sh | bash
   ```

   NOTE: The script will prompt for your password as your user account will be used to communicate with the Elasticsearch cluster. You will also be asked to enter the site of the Elastic cluster; this defaults to ech but may be different on test enclaves.

   If you are unsure of the cluster site, you can test access to the cluster by using ping:

   ```
   # ping elastic-node-1.{site}
   ```

   NOTE: You may also be asked for the xx_elastic.svc account password; if vSphere monitoring was not previously configured the script will prompt for this password. If you do not get

prompted the vSphere monitoring was already configured and the password has already been encrypted and stored. If this password needs to be changed in the future consult the Elastic System Administrators Guide for instructions on what to update for Elastic when the service account password changes.

5.5.5.2.1 Ensure Application Monitoring Configuration is Correct for Site

During the installation, if not already present, a file named appsconfig.ini was placed in the /ELK-local/elasticDataCollector directory. This file holds the definitions of applications to monitor, which span multiple machines. Review this file after installing the elasticDataCollector at each site. Edit the file using vi to ensure that the correct applications and hostnames are being monitored for the site.

On the Logstash VM

cd /ELK-local/elasticDataCollector

vi appsconfig.ini

NOTE: The elasticDataCollector service must be restarted if this files is modified.

```
# systemctl restart elasticDataCollector
```

NOTE: If this file already exists it is not overwritten during the upgrade.

5.5.5.2.2 Update Groups of Servers to Monitor

During the installation, if not already present, a file named groups.ini was placed in the /ELK-local/elasticDataCollector directory. This is the configuration file used to set up groups of hosts to monitor. To add groups, follow the format defined in the header of the file.

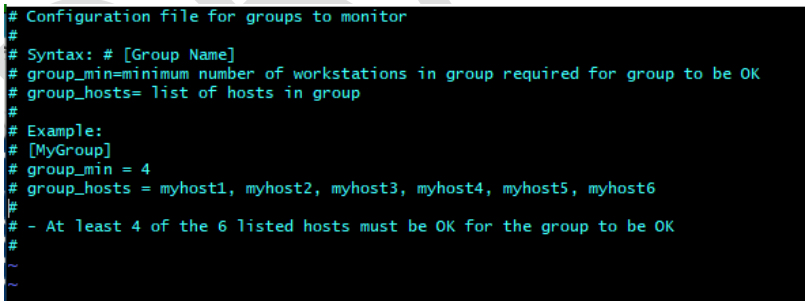


Figure 99 Configuration File

NOTE: The elasticDataCollector service must be restarted after updating this file.

```
# systemctl restart elasticDataCollector
```

NOTE: If this file already exists it is not overwritten during the upgrade.

5.5.5.2.3 Set Up Devices to Be Monitored (If Needed)

After installing the data collector, if not done on a previous install, a configuration file must be created that contains the devices to monitor for each site. The configurator tool is delivered as part of this upgrade to set up the devices to monitor. The tool was installed with the data collector. Use the tool to configure the devices to monitor for each site.

NOTE: The devices to monitor may be updated any time device information changes. During upgrades is a good time to verify that each site is configured to monitor the correct devices. The configurator can be used to verify the monitoring configuration of a site.

IMPORTANT: Making updates is not required if the current monitoring configuration is satisfactory. If no changes are desired re-enable puppet on this Logstash Instance and move onto the next section

```
# puppet agent --enable
```

Before proceeding with this section, you will need to gather the following information for all infrastructure devices that will be monitored for each site. You must create a configuration file for each site (Logstash instance) where you want to monitor devices.

Device Information needed to monitor XtremIO and Isilon devices:

- URL – This is the URL to access the device's web interface.
- Username – Username to access the device via REST API.
- Password – Password for Username.
- Display Name – Short unique description of device (ex: ech-isilon, ech-xtremio).
- Host – DNS resolvable name or IP address of host (ex: u00av01xms1).

Device Information needed to monitor Cisco switches, fx2 chassis, fc6xx blades, r6xx servers and Data Domain storage devices:

- URL – This is the URL to access the device's web interface (CMC, IDRAC, Cisco Prime).
- Username – Username to access the device via SNMP.
- Password – Authentication Password for Username.
- Priv Password – Privacy Password for Username.
- Display Name – Short unique description of device (ex: ech-fx2, ech-5k).
- Host – DNS resolvable name or IP address of host (ex: u00av01xms1).

5.5.5.2.3.1 Verify X11-Forwarding Enabled

To use this tool, you must be able to open a window from the Logstash box. The easiest way to do this is by using mobaXterm which has an embedded X Server. When the data collector was installed, it set up the SSH daemon to allow X11 connections. Puppet was also disabled on the Logstash host to allow this change to remain during device configuration. To have the change automatically reverted, Puppet must be re-enabled upon completion of section 5.5.5.2.3.2. To verify X11 is set up correctly, do the following:

3. Connect to the Logstash VM with a new mobaXterm session.
4. Verify X11-forwarding is enabled. There will be a green check box next to **X11-forwarding**, as shown in the following figure.

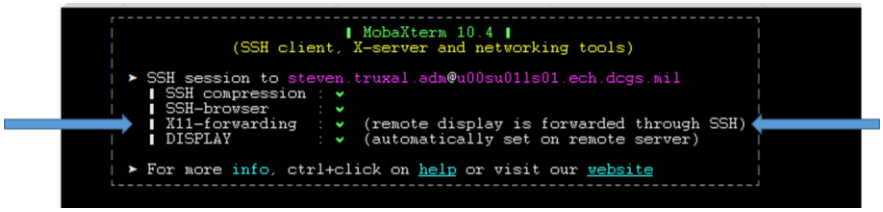


Figure 100 X11 Forwarding Setup Successfully

NOTE: When Puppet is re-enabled, X11-forwarding will be automatically disabled. To run the configurator if this occurs you can manually enable X11-forwarding by executing the following steps.

4. Execute the following command:
   `# sed -i 's/X11Forwarding no/X11Forwarding yes/g' /etc/ssh/sshd_config`
5. `# systemctl restart sshd`
6. Retry the previous test to verify X11 forwarding is set up correctly.

NOTE: If the previous steps are still not getting X11-fowarding enabled, ensure the AdressFamily setting is correct.

7. `# cd /etc/ssh`
8. `# vi sshd_config`
9. Look for the line setting for **AddressFamily**. If it is commented out, uncomment and verify it is set to **inet**.

   Should be: AddressFamily inet

10. Save changes (:wq)
11. # systemctl restart sshd
12. Retry the previous test to verify X11 forwarding is setup correctly.

IMPORTANT: If you cannot get X11-forwarding enabled, **STOP**, and consult a Linux SME for help.

5.5.5.2.3.2 Create Collector Configuration

NOTE: The previous step must be successful to continue. The GUI will not display if X11-forwarding is not enabled on the Logstash VM.

18. Log in to the Logstash VM with your .adm account and list the xauth cookies.

    ```
    xauth list
    ```

    You may see multiple cookies if X11-forwarding is enabled on other hosts. Take note of the cookie for this host; you will see the ls01 host name at the beginning.

    NOTE: If there are multiple entries for the ls01 host the last entry is most likely the one you will use. To make sure, execute **echo $DISPLAY** to get the correct number of the display for your SSH session. Use the cookie line that has both **ls01** and the display number.

Figure 101 xauth list example with logstash host

19. Sudo to become root.
    ```
    # sudo su
    ```
20. Copy the xauth cookie and add it to the roots xauth cookies.
    ```
    # xauth add <cookie>
    ```
    Example:

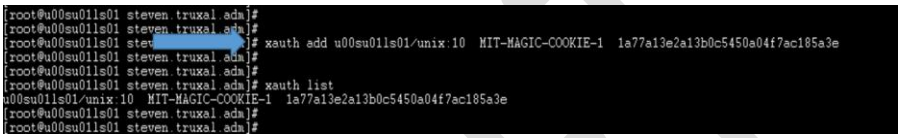


Figure 102 xauth add <cookie>

21. Run the configurator GUI.
    ```
    # cd /etc/logstash/scripts
    # . ./venv/bin/activate
    # python ./configurator.py
    ```

22. The device configuration window, **The Configurator**, displays.
    NOTE: If the window does not come to the foreground, look for another mobaXterm icon in the taskbar.

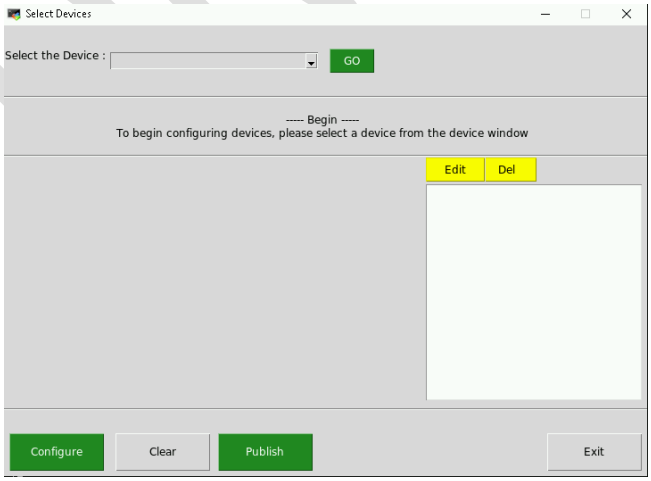


Figure 103 Device Configuration GUI

NOTE: If the data collector was already installed on a previous version the devices that are currently set up for monitoring will be displayed. If there are no updates to be made, verify that all devices to be monitored are listed and continue to section 5.5.5.2.3.3. If this is the initial configuration or edits need to be made, continue with this section.

23. Configure devices one at a time until you have added all devices to monitor for the site.

    NOTE: The elasticDataCollector uses standard SNMP and REST requests to query device information. If issues occur configuring any of the devices refer to SNMP troubleshooting commands in section **Error! Reference source not found.** to ensure the user/password(s) being used are correct. If the SNMP or REST commands do not work from the command line, they will not work from the data collector.
24. Select a device from the **Select the Device** menu and click **Go**.
    NOTE: You must select the device type again for every new device.

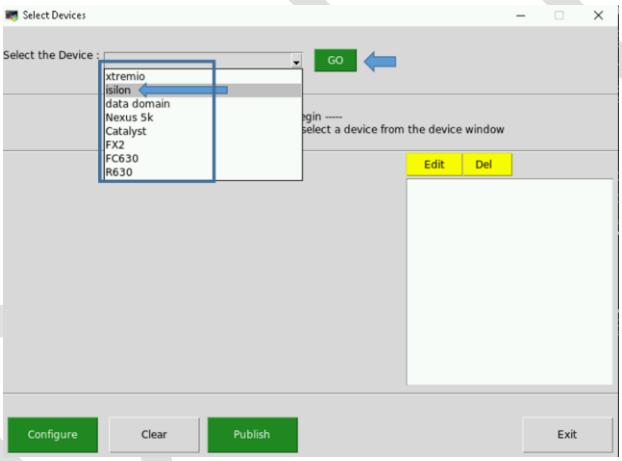


Figure 104 Select Device to Configure

25. Fill in the fields with the required information to monitor the device. Click **Configure**.

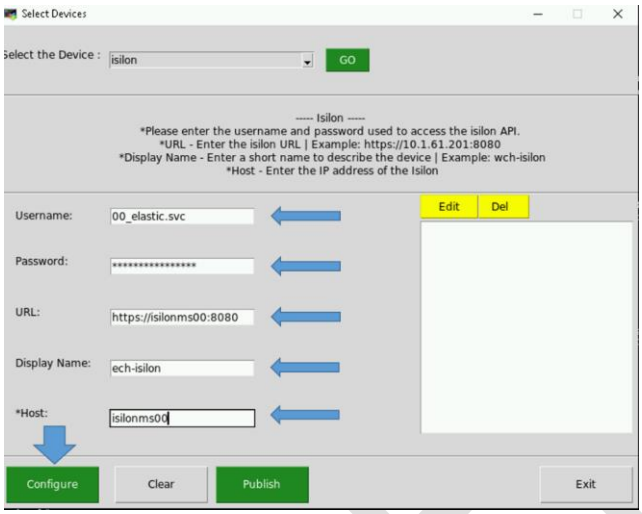


Figure 105 Configure Device

26. The list on the right will populate.

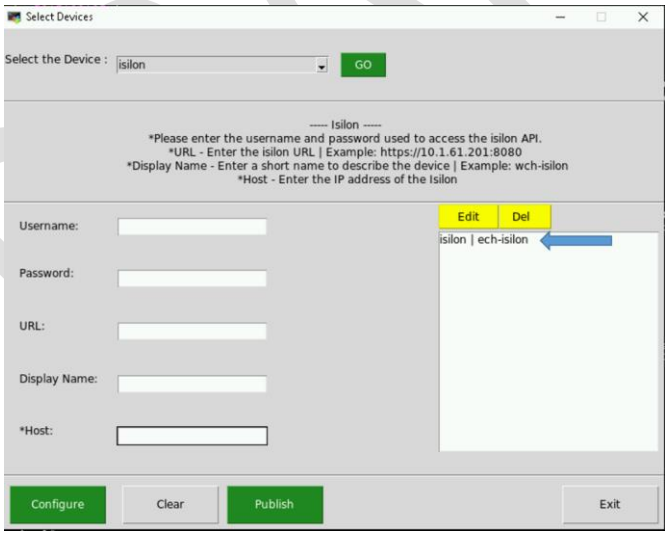


Figure 106 Device Added to List

27. Enter another device or press **Publish** to test your configuration.

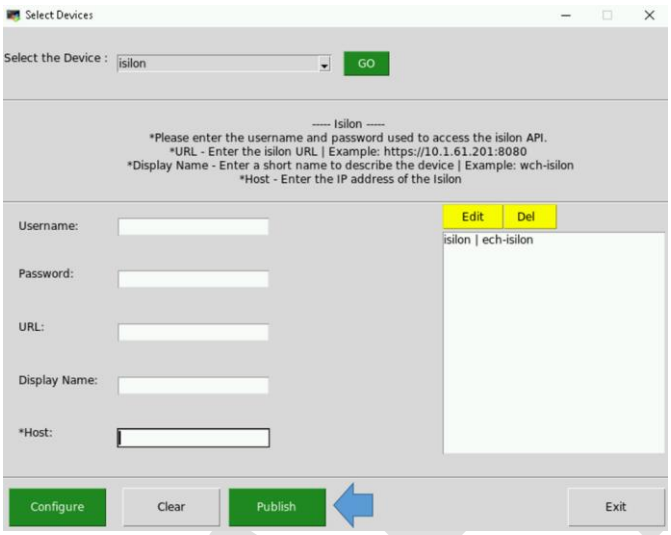


Figure 107 Publish Device Configuration

28. The Configurator will verify the values you have entered by trying to access the device(s). If all goes well your configuration will be updated and a success message will be displayed.

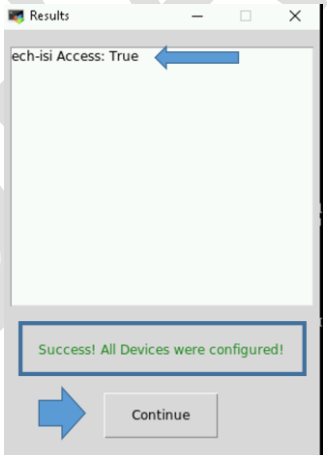


Figure 108 Successful Publish

a. If any of your access parameters are incorrect the **Publish** will fail, and the configuration file will not be updated.

Figure 109 Unsuccessful Publish

29. Select the **Continue** button in the **Results** window to continue. If you entered information incorrectly and the **Publish** failed, you can either **Edit** or **Del** the device that failed.

    NOTES:
    - If any of the devices fail, the configuration file is NOT updated. Access to all devices must be successful to create/update a device configuration file.
    - Every successful publish will restart the elasticDataCollector service so the current configuration is read in and collection from the updated devices is started.

30. If the **Publish** failed, you can use the **Edit** feature to correct the issue. For security reasons the password is not shown, so if everything else looks correct try re-entering the password. To review this feature, select the device you'd like to modify and press the **Edit** button.

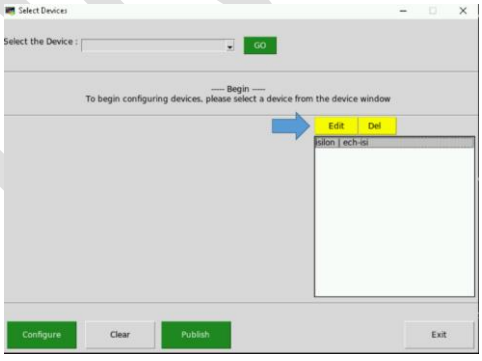


Figure 110 Select Edit to modify device information

31. The **Edit Item** dialog appears. The information that was entered for the device you selected is displayed. You can make changes to any of the items and select **Save** or go back to the main screen without making any modifications by selecting **Cancel**.

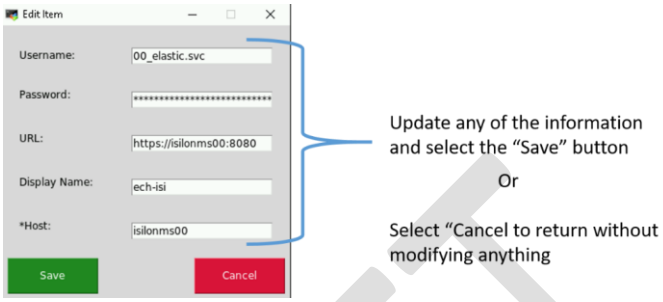


Figure 111 Edit Item

32. Once back at the main screen you can select **Publish** again, verify your device configurations, or continue entering devices by selecting a new device from the **Select the Device** menu.
33. Once you have configured all your devices and the **Publish** is successful, you can exit the Configurator. Your device configuration is automatically saved on each successful Publish.
34. Re-enable Puppet on the Logstash host by executing the following command:

```
# puppet agent --enable
```

IMPORTANT: Don't skip the previous step.

NOTES:

- You can run the configurator GUI at any time to either view, modify, or add devices for this Logstash instance to monitor.
- Every successful publish will restart the elasticDataCollector service so the current configuration is read in and collection from the devices is started.

5.5.5.2.3.3 Verify Device Data is Being Collected

Once you have a good configuration in place and the elasticDataCollector service is running we can verify that Elastic is receiving data for the devices. The easiest way to do this is viewing the **IAAS-ES-Infrastructure Overall Status Dashboard**. This dashboard shows the overall status of each of the devices you have configured.

4. Select the **Dashboard** option from the hamburger menu.

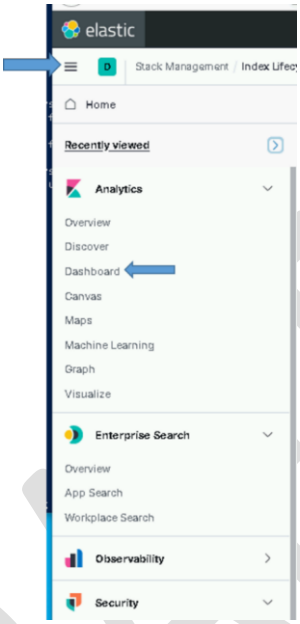


Figure 112 Select Dashboards Option

5. Type **IAAS-ES-Infrastructure Overall Status Dashboard** in the search bar.

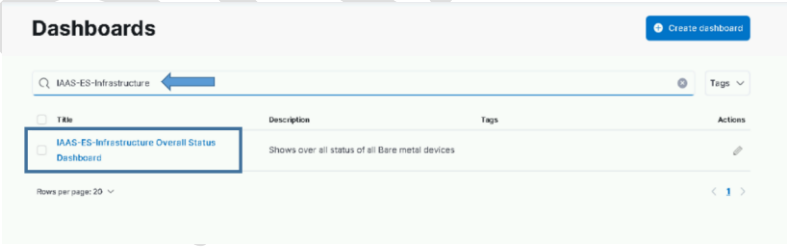


Figure 113 Select IAAS-ES-Infrastructure Status Dashboard

6. Verify that the devices you entered into the configuration are listed in the dashboard.

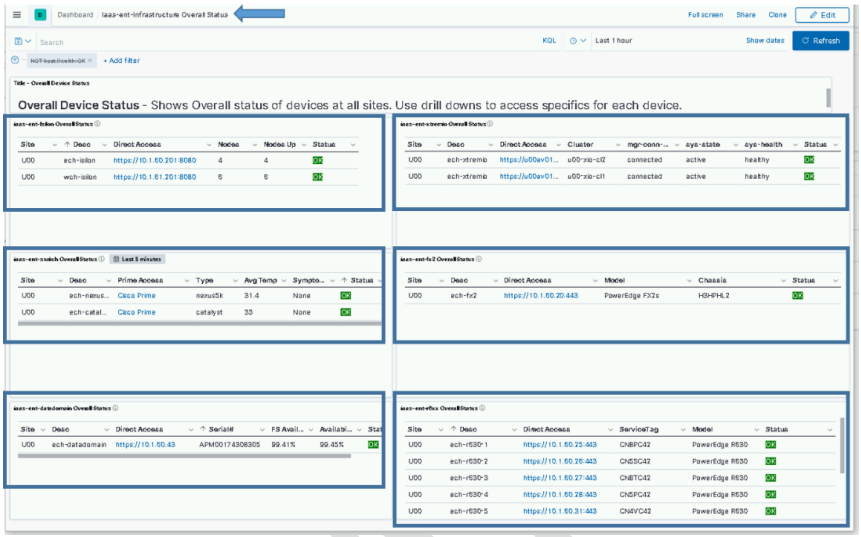


Figure 114 Devices Listed in Dashboard

There should be 6 areas on the dashboard:

- Isilon
- Switches
- Data Domain
- XtremIO
- Fx2
- Rc6xx

IMPORTANT: All Logstash and elasticDataCollector instances must be upgraded before moving on to the next step.

5.5.5.2.3.4 Update 7k switch configuration for Data Collector (If needed)

When setting up switches for monitoring in the previous installation of the data collector there was not an option for 7k switches. Because of this the 7k switches were configured as 5k switches or possibly catalyst switches. This version brings the ability to configure 7k switches for monitoring but also requires that any 7k switches that were previously setup as 5k switches must be changed to the correct configuration to ensure proper monitoring.

Follow these steps to modify an existing 7k switches configuration from being identified as a 5k or catalyst to the proper identification as a 7k.

Open the configurator GUI as described in Section 5.5.5.2.3 above to correct any of the 7k switches for the site that have been previously configured as 5k or catalyst switches by executing the following steps for each incorrectly configured switch.

1. Select the 7k switch that is currently defined as a 5k or catalyst switch
2. Select "Edit"

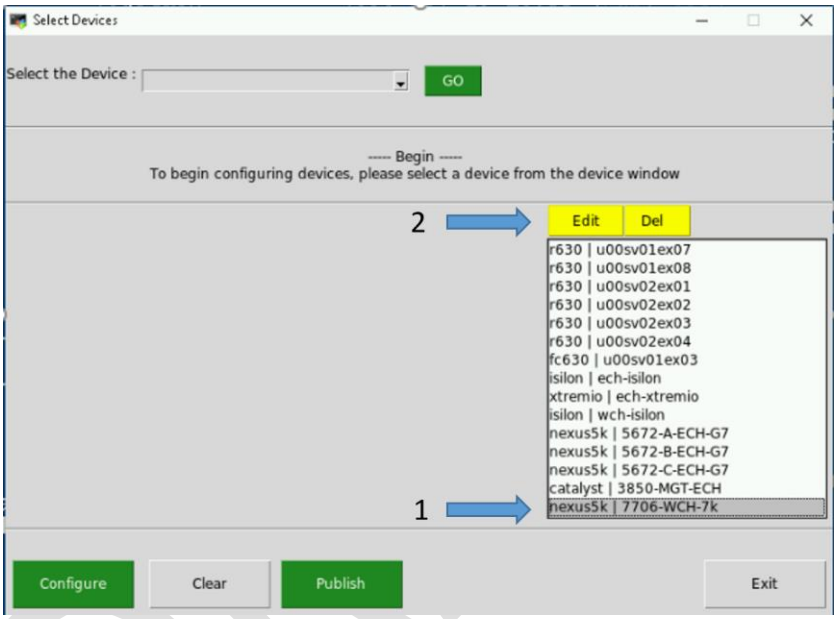


Figure 115- Example of selecting incorrectly configured 7k switch

3. From the "Device Name" pull down menu select "nexus7k" and then "Save" (No need to update anything else unless necessary)

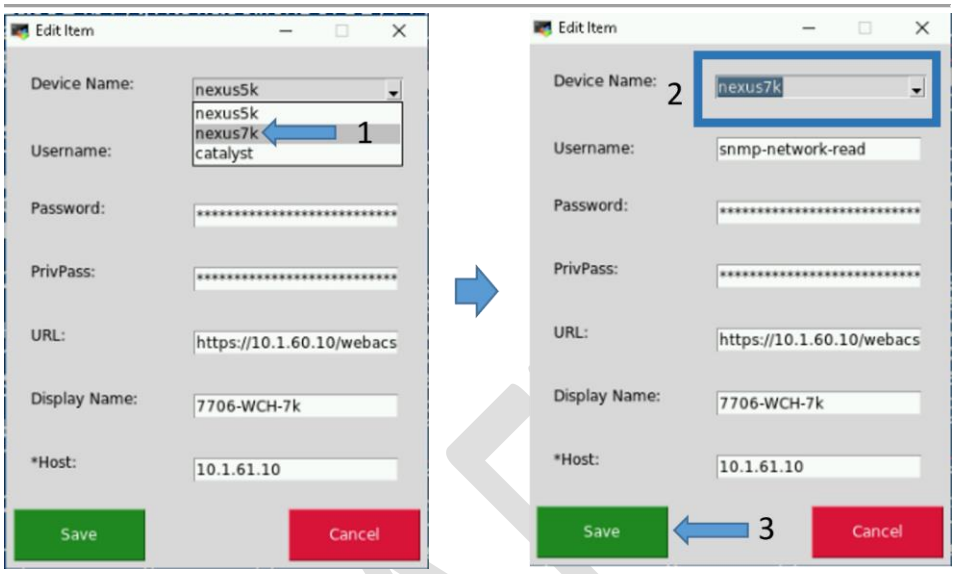


Figure 116- Change 5k device to 7k device

4. Repeat steps 1 thru 3 for all 7k switches that were previously misconfigured
5. Select the “Publish” button to update the stored device values

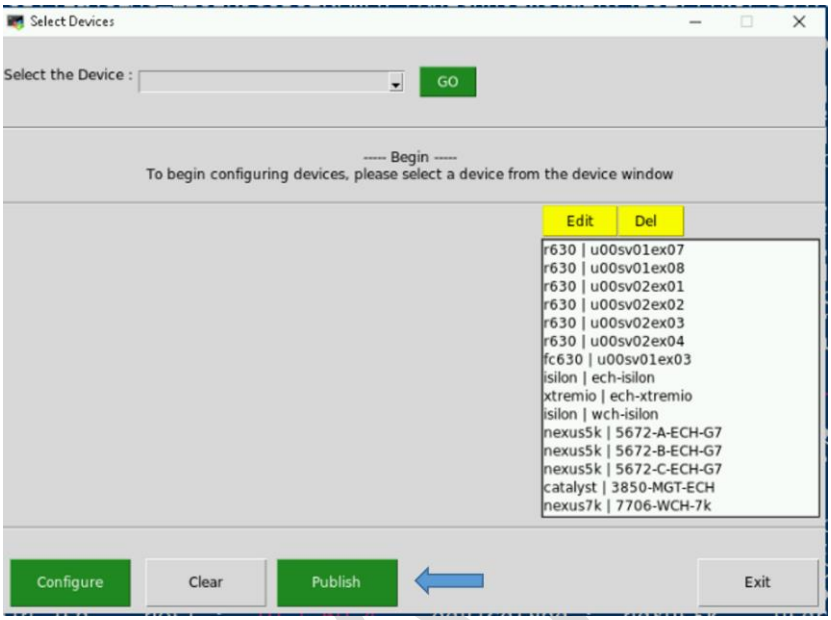


Figure 117- Publish Device Configurations

6. The Configurator will verify the values you have entered by trying to access the device(s). If all goes well your configuration will be updated and a success message will be displayed

Figure 118- Example of successful publish of device information

7. Select “Continue” to return to main window and then select “Exit”

Note: The ElasticDataCollector will be restarted on a successful publish

5.5.6 Upgrade Beats

5.5.6.1 Verify Beat Templates are Loaded

All templates, including beats, have been converted to composable index templates; these templates were loaded in section 5.5.4.3. These steps are to verify the templates for the new version of beats have been loaded correctly.

Verify all beats component templates were loaded properly by ensuring they are present in the **Component Templates** section of the **Index Management** screen in Kibana.

1. From the hamburger menu, select **Stack Management**.
2. Select **Index Management** in the **Data** area.
3. Select the **Component Templates** tab.
4. Type **beat-{version}** in the **Search** bar.

The following is an example showing the Index Management page with the Component Templates for the 7.16.3 beats.

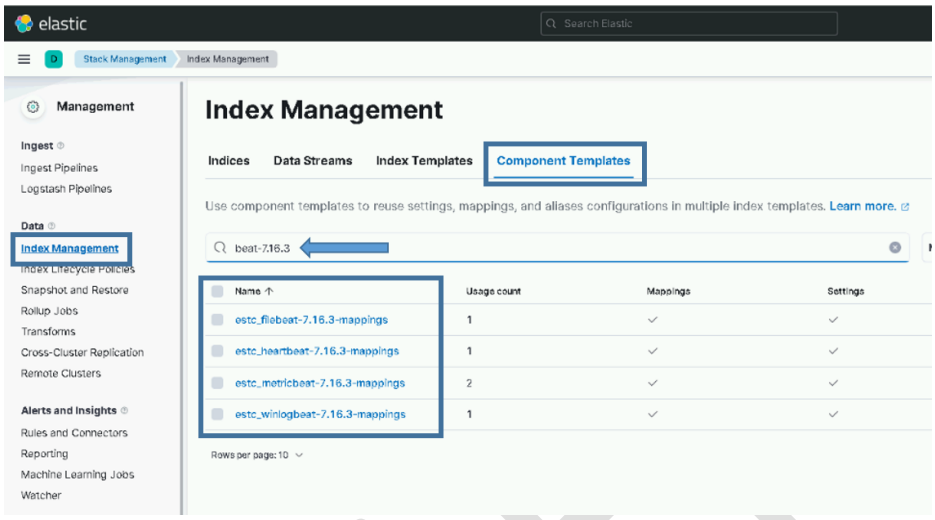


Figure 119 Example of beats component templates for version 7.16.3

You should see a component template for each type of beat:

- estc_filebeat-{version}-mappings
- estc_heartbeat-{version}-mappings
- estc_metricbeat-{version}-mappings
- estc_winlogbeat-{version}-mappings

If you do not see the component templates for all 4 types of beats, you should stop here and consult with an Elastic SME.

Verify all beats index templates were loaded properly by ensuring they are present in the **Index Templates** section of the **Index Management** screen in Kibana.

Go to **Stack Management in Kibana** and look on the **Index Templates** tab of the **Index Management** screen in the **Data** section.

1. From the hamburger menu, select **Stack Management**.
2. Select **Index Management** in the **Data** area.
3. Select the **Index Templates** tab.
4. Type **beat-{version}** in the Search bar.

The following is an example showing the Index Management page with the Index Templates for the 7.16.3 beats.

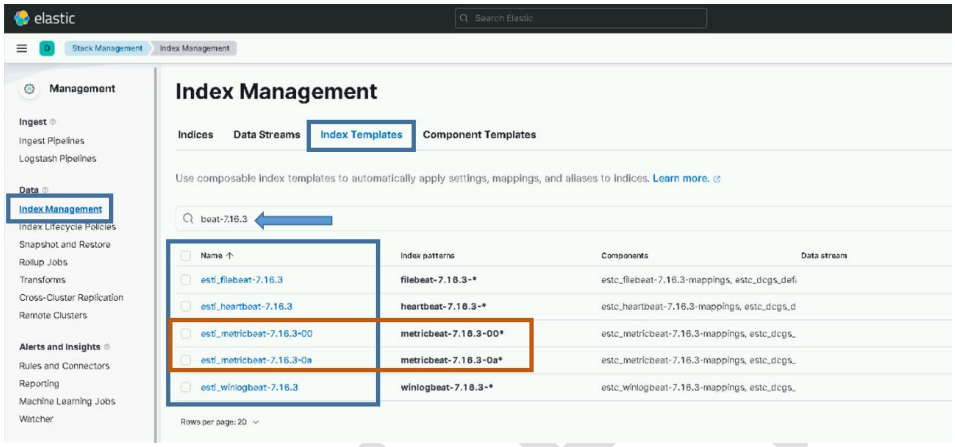


Figure 120 Example of beats index templates for version 7.16.3

You should see an index template for filebeat, heartbeat, and winlogbeat.

- esti_filebeat-{version}
- esti_heartbeat-{version}
- esti_winlogbeat-{version}

If you do not see an index template for each beat shown, you should stop here and consult with an Elastic SME.

You should also see site-based index templates for Metricbeat. There should be a Metricbeat index template for each site that is sending data into Elastic.

- esti_metricbeat--{version}-{site}

If you do not see index templates for every site, you should stop here and consult with an Elastic SME.

5.5.6.2 SCCM Configuration to deploy beats on Windows

NOTE: An SCCM administrator is required to execute this section.

SCCM is used to deploy all beat collectors for Windows systems. Currently SCCM deploys Winlogbeat, Metricbeat, and Filebeat on OA DCGS systems.

1. Extract oadcgs-es-elastic-sccm- X.X.X.X.zip onto the SCCM share (staging area).
2. Copy the elastic_cachain.pem file from the existing Elastic share into the extracted SCCM install, replacing the dummy cachain.pem (e.g., on the fileserver in <install>\oadcgs-es-elastic-sccm-1.3.24.19\oadcgs-es-elastic-sccm\sccm\shareDir\).

To upgrade beats collectors used on Windows system in DCGS, refer to Section 5.3 Installation Instructions for Upgrades in *ES-018 – SCCM – Instructions for Building an SCCM Package to Install Beats for Windows*.

IMPORTANT: The cachain.pem file delivered with the upgrade is empty and must be replaced with the correct cachain.pem for the system.

NOTE: Make sure the following files are present on the SCCM share after installing the new SCCM package:

NOTE: Also ensure any system specific configurations are not lost by copying them into the new package. This would include configuration files not delivered or configuration files modified after delivery.

- install_beats_windows.ps1
- remove_Beats.ps1
- cachain.pem (Updated with system cachain.pem)
- configs/Metricbeat
    - all.module.windows.yml
    - all.module.system.yml
    - appmonitor_win.js
    - dc01.metricbeat.yml
    - dc02.metricbeat.yml
    - hb10.metricbeat.yml
    - hb11.metricbeat.yml
    - jb01.metricbeat.yml
    - metricbeat.yml
    - sc01.metricbeat.yml
    - sc02.metricbeat.yml
    - sc03.metricbeat.yml
    - sc04.metricbeat.yml
    - sc05.metricbeat.yml
    - sc06.metricbeat.yml
    - wb01.metricbeat.yml
- configs/filebeat
    - filebeat.yml
    - inputs.txt
- configs/filebeat/inputs.d
    - example_config.yml
- configs/winlogbeat
    - ec01.winlogbeat.yml

IMPORTANT: You will need the following zip files to execute the instructions.

1. The zip files for the beats to be upgraded should now be added to the **zipfiles** directory:

- filebeat-{version}.windows-x86_64.zip
- metricbeat-{version}.windows-x86_64.zip
- winlogbeat-{version}.windows-x86_64.zip

2. Extract Elastic-Elastic_Window_Beats-7.X.X.zip onto the SCCM share.

NOTES:

- SCCM updates for Beats deployments should always occur after installs/upgrades of the Elastic core components.
- You will not need to redeploy this package. The Distribution Points will update on a daily schedule. For quicker results, you can right click the package in the SCCM console and select **Update Distribution Points**.

5.5.6.3 Metricbeat upgrade for Domain Controllers

NOTES:

- A user with the Domain Admin Role is required to execute this section. An SCCM SME may be needed to verify the correct location of the Elastic Share for SCCM.
- The DCMedia Folder is replicated among all domain controllers so this update only needs to be made from one domain controller and the others should also receive the update.
- The DCMedia DFS was configured in a previous Elastic upgrade. If it is not configured consult an Elastic SME for guidance.

Follow these steps to upgrade Metricbeat on all domain controllers:

1. Determine location of new Metricbeat zip file (Metricbeat-X.X.X-windows-x86_64.zip) from install team.
2. Login to any Domain Controller
3. Copy new Metricbeat to **Error! Hyperlink reference not valid.**
4. Remove zip files for older versions

The scheduled task will execute once a day, ensuring that Metricbeat is installed and running on each domain controller. After the scheduled execution time (4 am in the current DFS installation instructions) verify that Metricbeat has been upgraded by checking the version in Elasticsearch.

NOTE: If after 24 Metricbeat has not been upgraded to the correct version reach out to an Elastic SME for guidance.

5.5.6.4 Linux Beats

NOTE: A Puppet administrator is required to execute this section.

Puppet is used to automatically install Metricbeat and Filebeat on Linux hosts. Details on this can be found in *ES-018 – Elastic Logging and Aggregation Cluster (ELAC) – System Installation Instructions*. Puppet should already be set up for installing beats automatically so in this section you will only be updating the repo with the new RPMs.

6. Copy the following RPMs to the Elastic repo (/var/www/html/yum/elastic):
   - filebeat-{version}-x86_64.rpm
   - heartbeat-{version}-x86_64.rpm
   - metricbeat-{version}-x86_64.rpm
7. Ensure RPMs have the correct owner/group:

   ```
   # chown -R apache:apache *
   ```

8. Repo files must have SELinux context **httpd_sys_content_t** set. If you copy the RPMs into the directory, they will automatically have this context set. If you move them, they won't. Check to ensure all files have the correct context set by executing:

   ```
   # ls -lZ
   ```

Figure 121 httpd_sys_context_t

9. If all files do not have **httpd_sys_context_t** set, execute the following:

   ```
   # restorecon *
   ```

10. Recreate the Elastic repo so it's ready for use:

    ```
    # createrepo ./
    # gpg --detach-sign --armor ./repodata/repomd.xml
    ```

    NOTE: Ensure there are 2 dashes before "detach-sign" and "armor" when running this command.

Now that the RPMs are available, Puppet will automatically install the new version when it is run on each host.

5.5.6.4.1 Update Puppet to Restart Beats on Every Puppet Run

NOTE: A Puppet administrator is required to execute this section.

Puppet must be executing correctly on each Linux box for the beats upgrade/restart to work properly.

To ensure the beats are restarted after being upgraded by Puppet set the new **restart_beats** parameter to **true** in the Puppet web interface.

1. Under **Node Groups** find the group for Elastic Clients then select the **restart_beats** parameter on the **Classes** tab (**Configuration** tab in older Puppet versions) of the **Elastic Clients** classification.

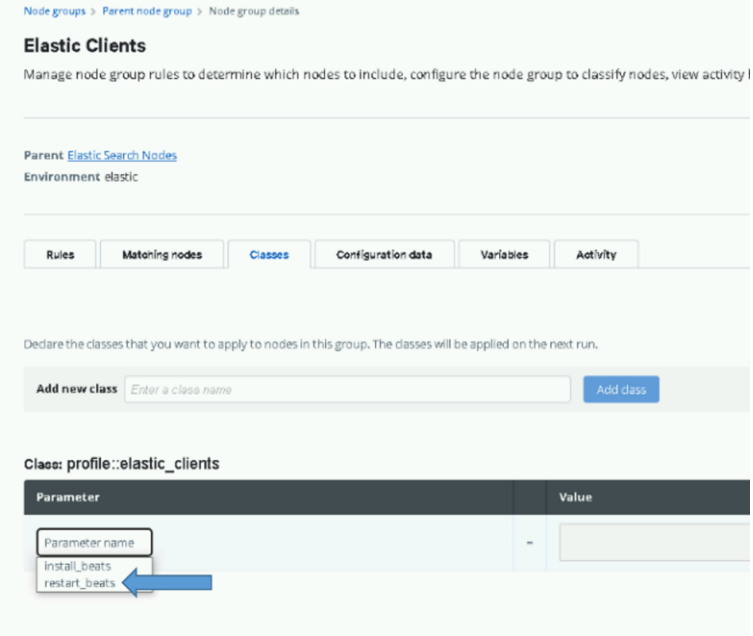


Figure 122 Select restart_beats

2. Once selected, change the value from the default (**false**) to **true** and click **Add to Node Group** (**Add Parameter** in older Puppet versions).

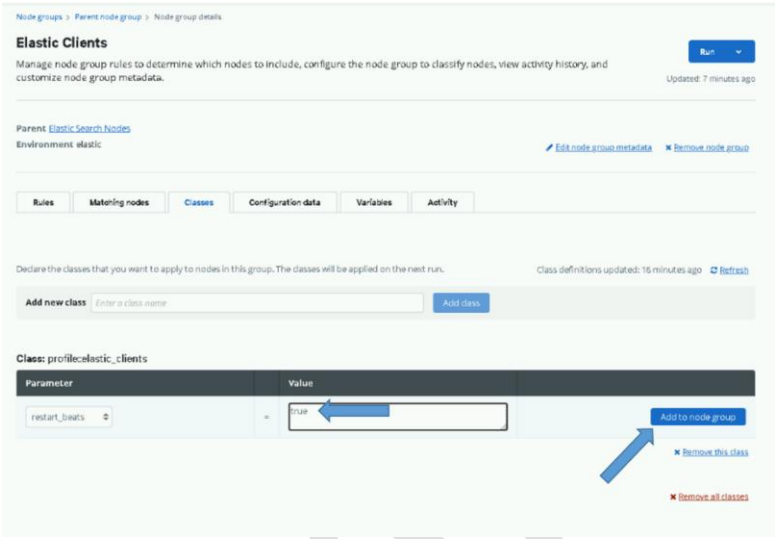
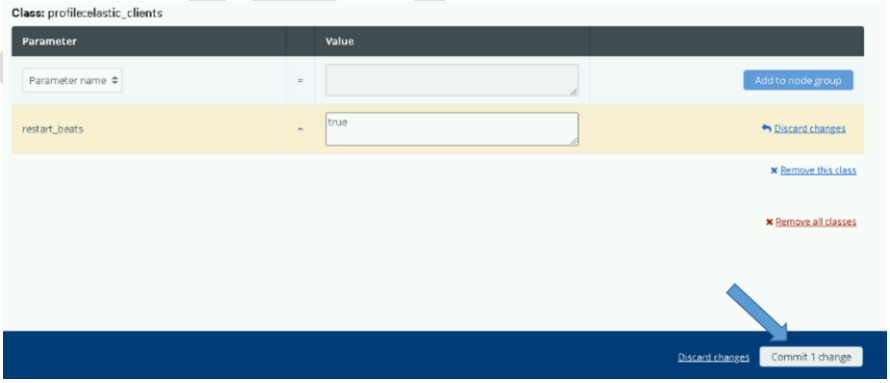


Figure 123 Add to Node Group

3. Click the **Commit 1 change** button to commit the change.



4. After setting this option you can monitor the beats being upgraded using the **Beats Versions Dashboard**, which is loaded in the next section.
5. Once the job finishes or in about an hour all beats should have been upgraded to 8.6.2, return to this section and remove this parameter. Upon removal, the setting will return to the default value

(**false)** and the beats will stop being restarted on every Puppet run. (**DO NOT FORGET TO DO THIS.**)

Figure 124 Remove Parameter

5.5.7 Load Kibana Saved Objects

NOTE: You must be root and a member of the **ent elastic admins** AD group to load saved objects into Kibana. Having the **Elastic Administrator** OneIM Role will place the user in this group.

5. Run the following command as root from any of the running Elastic nodes to install objects; red text in the output can be ignored:

   ```
   # curl -k https://{site code}su01ro01.`hostname -
   d`/yum/elastic/install/load_objects.sh | bash
   ```

6. After running the script, verify the objects are loaded. Navigate to the **Stack Management** screen. Select **Saved Objects** under the **Kibana** section.

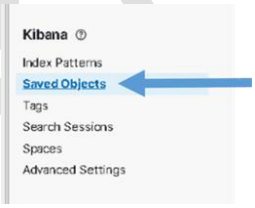


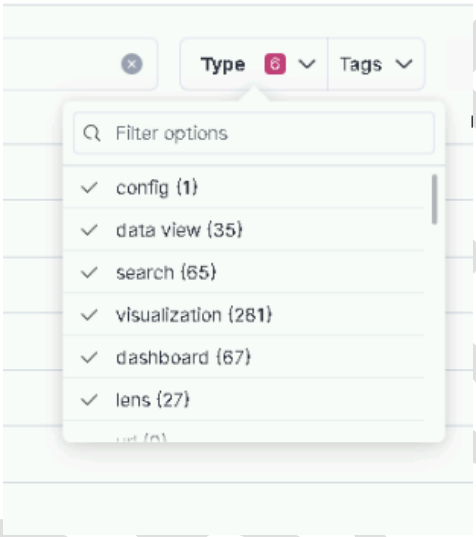
Figure 125 Select Saved Objects

7. The **Saved Objects** page displays. There should be at least 476 Objects loaded.

8. Select the **Type** drop-down, scroll down and examine each type. The following shows the minimum number you should see for each type. There may be more if additions were added that are not delivered with the baseline.

- data view (35)
- search (65)
- visualization (281)
- dashboard (67)
- lens (27

Example:



Note: As part of this upgrade object for exiting ART integrations are added. This include GXPXplorer, SOAESB, SocetGXP and MAAS data views, lens, visualization, dashboard and search objects.

5.5.8 Validate URL Links

Elastic has modified the way it handles object IDs in version 8 which brings a breaking change for embedded URL links. We are using a few of these that may need to be manually updated if new ids are assigned to the dashboards they are pointing to during the previous import step. This section gives instruction on updating these URL links if they are not working. This breaking change is projected to be fixed in Elastic version 8.8 so this should only be an issue for this release.

NOTE: This issue will also occur when copying dashboards between spaces.

IMPORTANT: When a dashboard is copied into a space the UUID (dashboard id) for the dashboard is updated. Therefore, any URLs pointing to other dashboards will fail (Will contain old UUID). To fix this a manual intervention is needed. This is fixed in future Elastic releases (Expected in version 8.8)

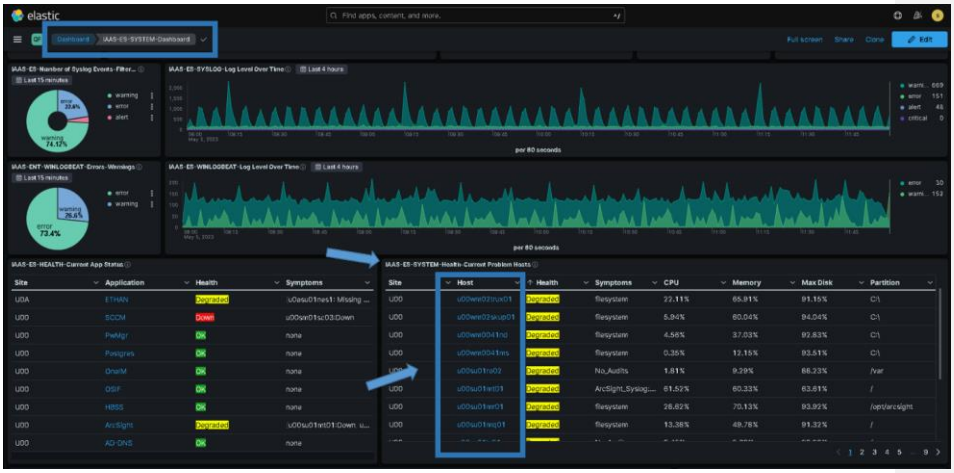
There are 2 URL links that need to be checked. If either does not work, follow the steps to correct the link. Both links are contained in the "dcgs-current-healthdata*" Data View for the dcgs-current-healthdata-iaas-ent index.

The following fields have a custom URL field format which can be affected.

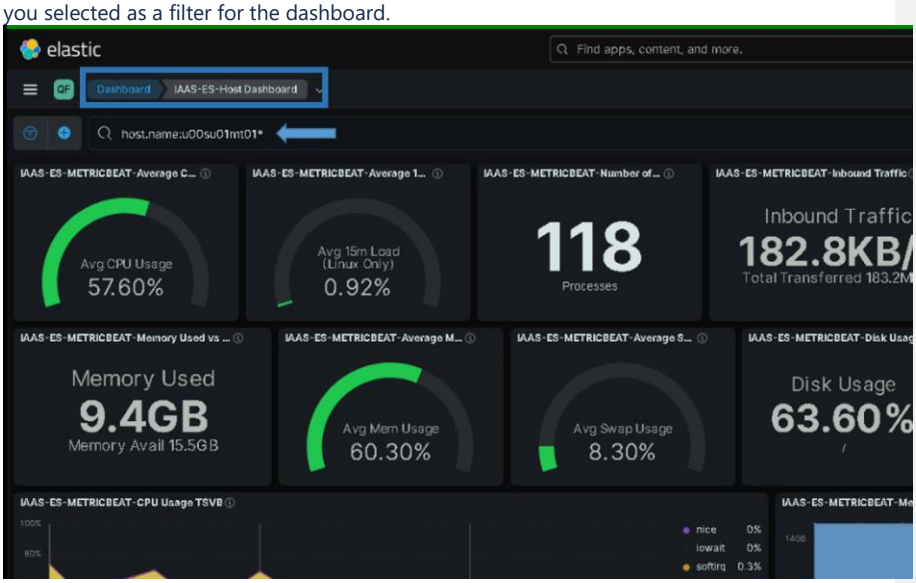
- host.name – URL link to the "IAAS-ES-Host Dashboard" passing the hostname clicked on as the filter for the dashboard.
- app.Name – URL link to the "IAAS-ES-SYSTEM-Application-Info" dashboard passing the application name as the filter for the dashboard

5.5.8.1 Validate URL link for IAAS-ES-Host Dashboard

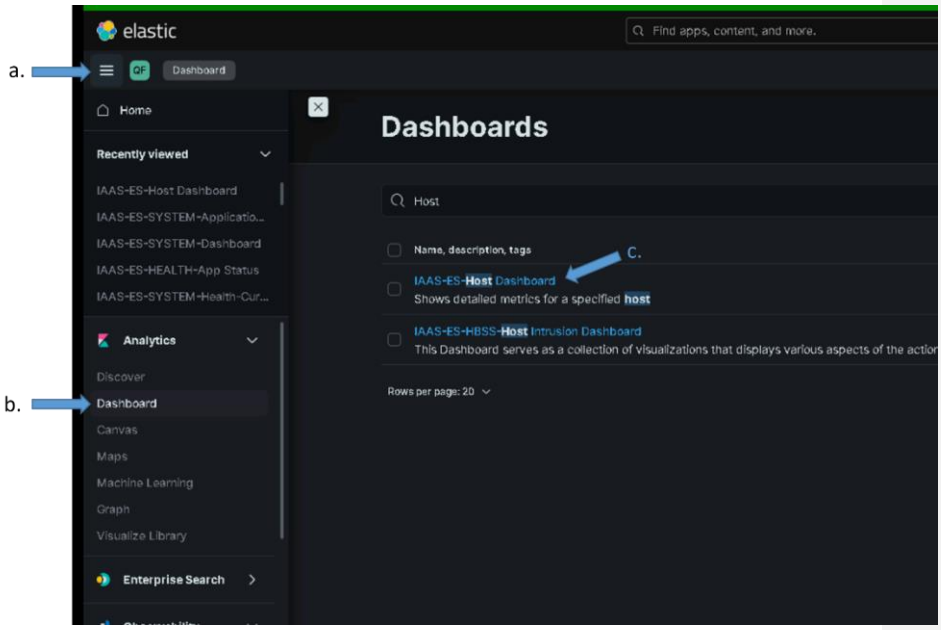
1) Bring up the IAAS-ES-SYSTEM-Dashboard and click on any hostname in the "IAAS-ES-SYSTEM-Health-Current Problem Hosts" visual.



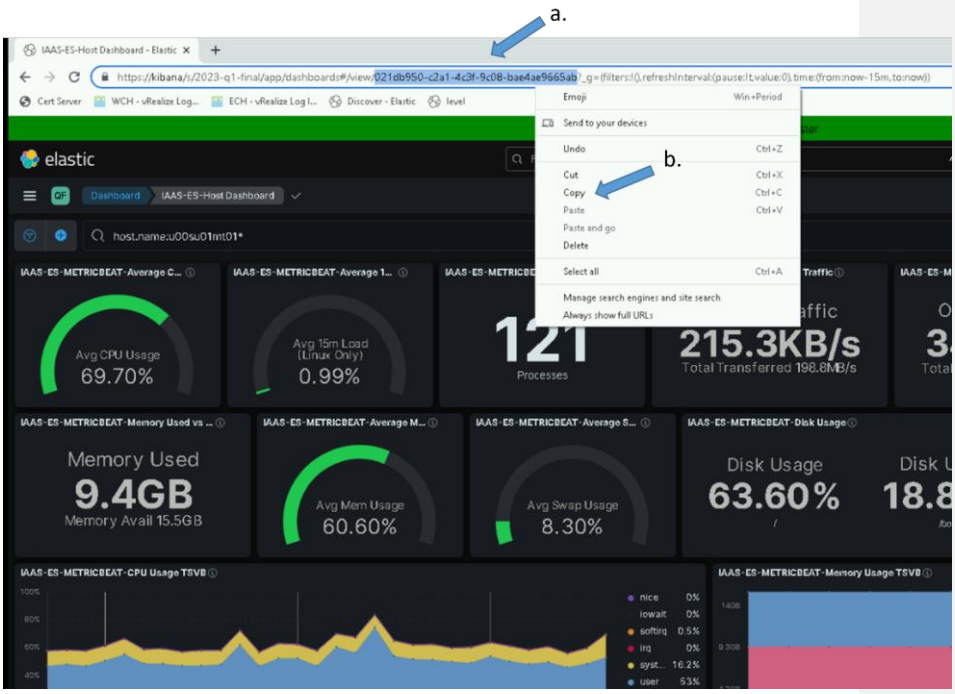
2) Validate that the link opens the "IAAS-ES-Host Dashboard" and passes the hostname that you selected as a filter for the dashboard.



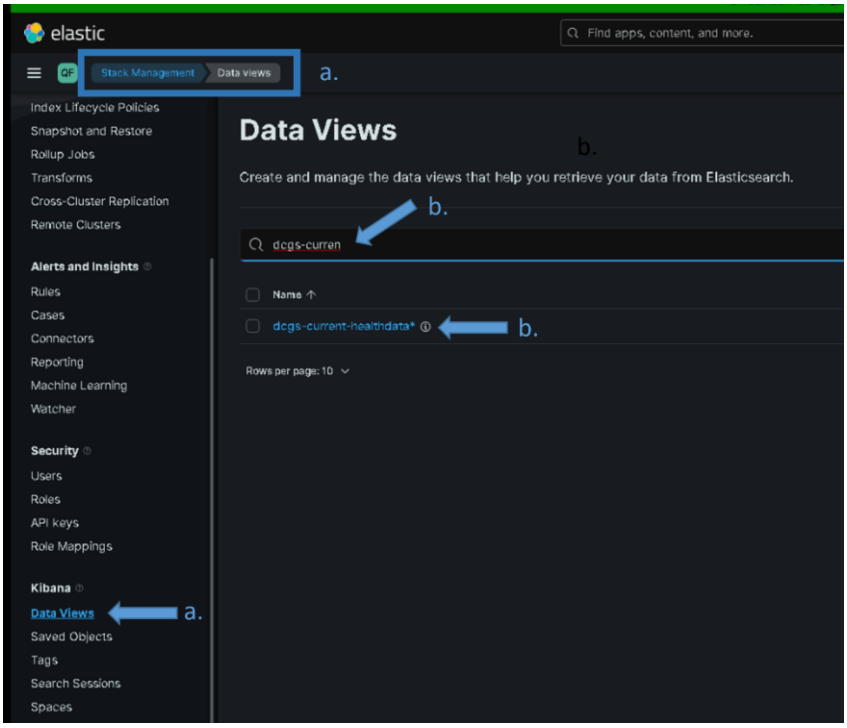
3) If the dashboard appears and the filter is correct the link is working, continue to the next URL link validation - 5.5.8.2. If the dashboard does not appear or it is not filtered for the host you've selected continue to step 4.
4) Navigate to the "IAAS-ES-Host Dashboard"
    a. Select "Dashboard from the hamburger menu
    b. Type "Host" in the search field
    c. Select the "IAAS-ES-Host Dashboard"



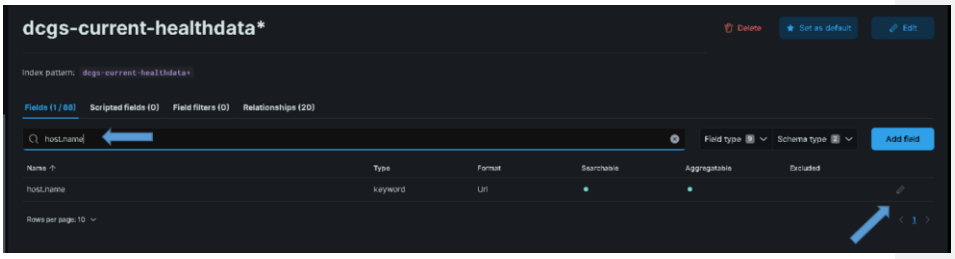
5) Copy the 36-character dashboard id (UUID) in the URL for the dashboard from the Address bar of the browser into your clipboard by highlighting it, right clicking and selecting copy.
    a. Highlight the dashboard id in the Address Bar
    b. Right Click and select copy



6) Navigate to the "dcgs-current-healthdata*" Data View and select edit for the host.name field
   a. From Hamburger Menu select "Stack Management" -> "Data Views"
   b. Type "dcgs-current" in search bar and select the data view



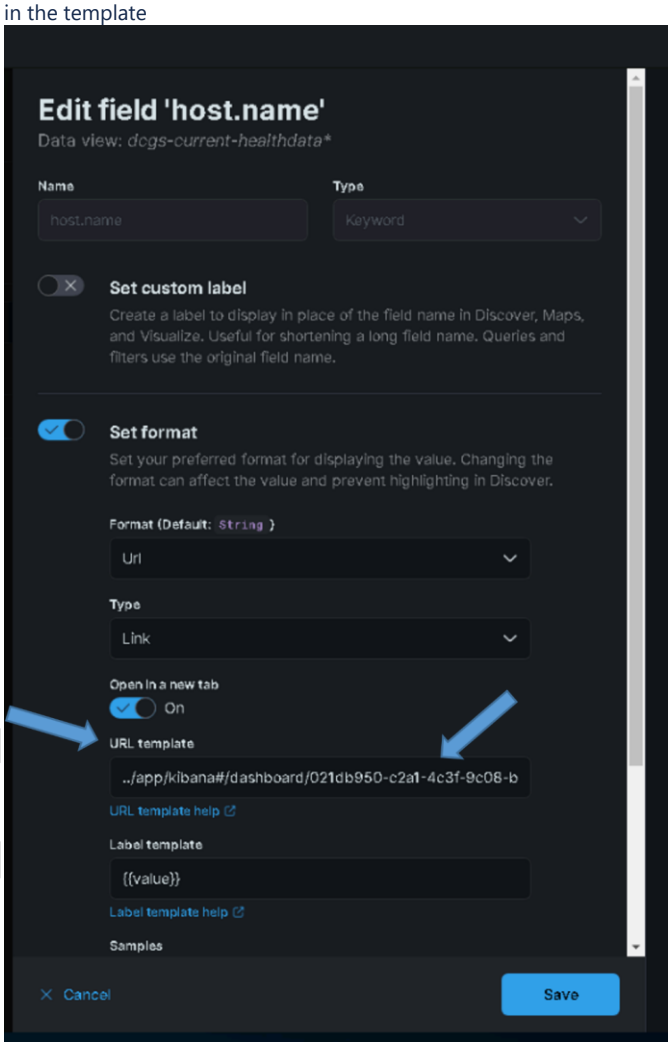
7) Type “host.name” in the search field and select the pencil icon to edit the field



8) Replace the existing dashboard ID in the “URL template” field with the one you are holding in your clipboard (From the copy)
   a. Delete the existing id that is shown after the “dashboard/” in the template

      Example of what URL template looks like before removing the id:
      ../app/kibana#/dashboard/**b4acc984-701a-4d7b-94cd-08da2a60ee85**?_a=(query...

b. Right click and paste the id from your clipboard as the new id after the "dashboard/" in the template



c. Ensure the new URL link is correct with no extra spaces. Below is an example of the new URL link using the UUID from the example above:

../app/kibana#/dashboard/**021db950-c2a1-4c3f-9c08-bae4ae9665ab**?_a=(query:(language:kuery,query:'host.name:{{value}}*'))

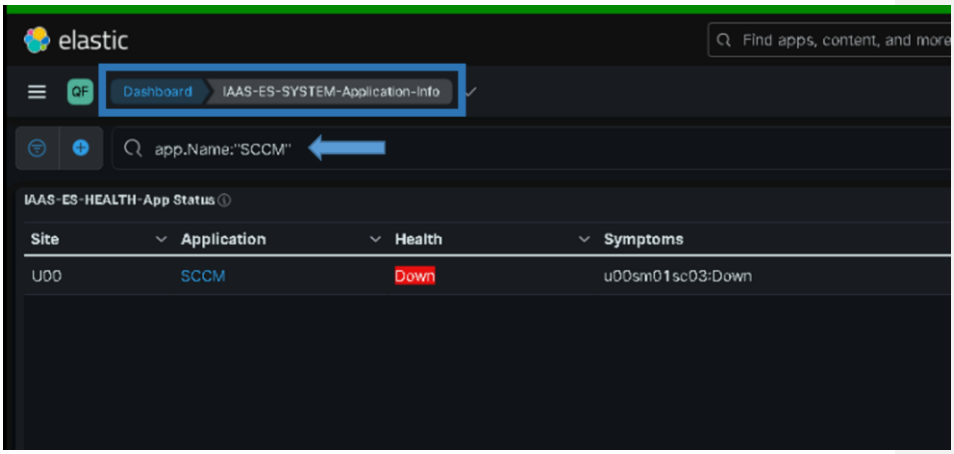
Note: There is not spaces in the id (**021db950-c2a1-4c3f-9c08-bae4ae9665ab**)

9) Select "Save" on the Edit field 'host.name' pop-up.
10) Return to step 1 and validate the link now works.

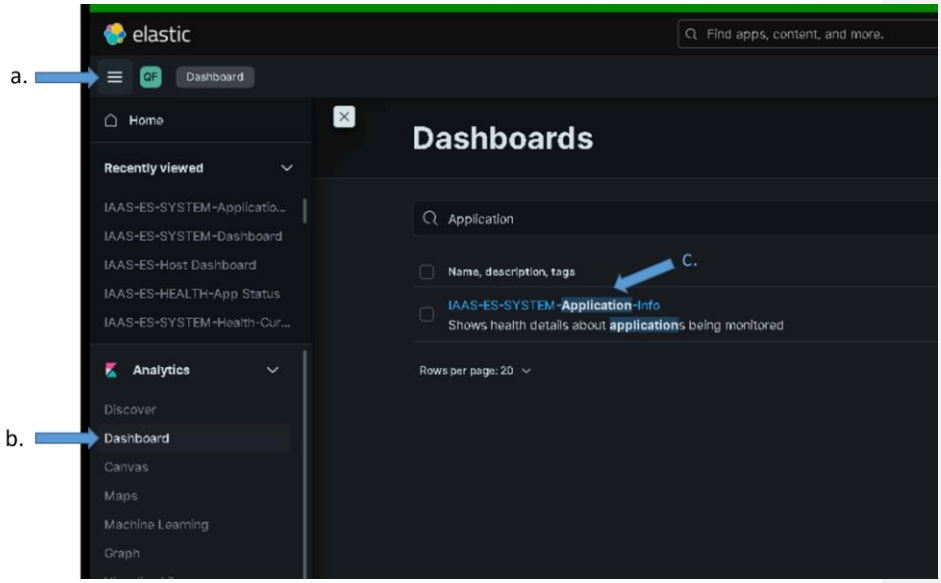
5.5.8.2 Validate URL link for IAAS-ES-SYSTEM-Application-Info dashboard

1) Bring up the IAAS-ES-SYSTEM-Dashboard and click on any Application in the "IAAS-ES-HEALTH-Current App Status" visual.

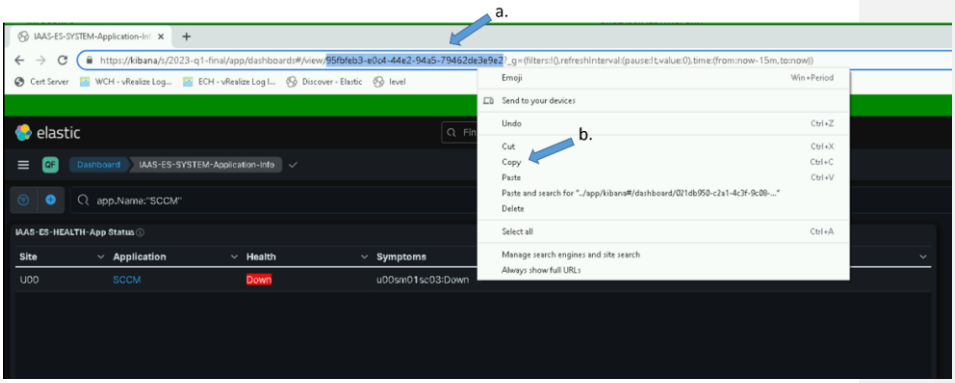
2) Validate that the link opens the "IAAS-ES-SYSTEM-Application-Info" dashboard and passes the application name that you selected as a filter for the dashboard.



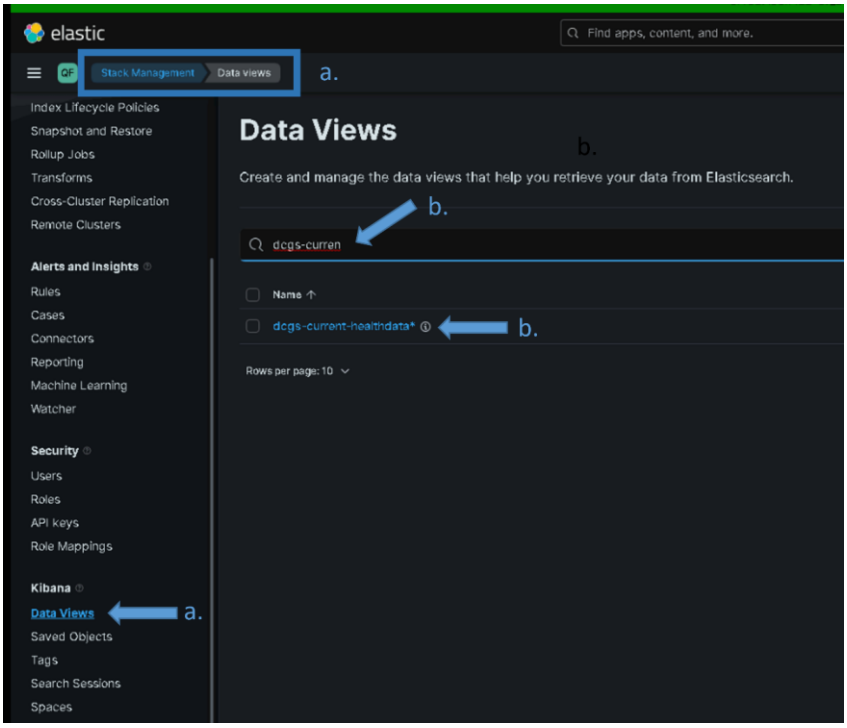
3) If the dashboard appears and the filter is correct the link is working, continue to the next section of the installation. If the dashboard does not appear or it is not filtered for the application, you selected continue to step 4.
4) Navigate to the "IAAS-ES-SYSTEM-Application-Info"
    a. Select "Dashboard from the hamburger menu
    b. Type "Application" in the search field
    c. Select the "IAAS-ES-SYSTEM-Application-Info"



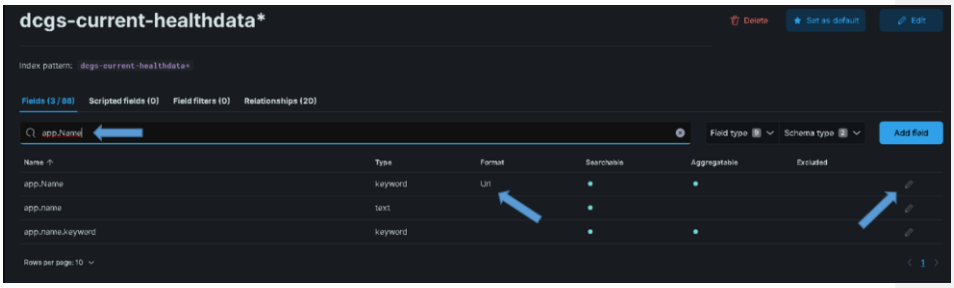
5) Copy the 36-character dashboard id (UUID) in the URL for the dashboard from the Address bar of the browser into your clipboard by highlighting it, right clicking and selecting copy.
    a. Highlight the dashboard id in the Address Bar
    b. Right Click and select copy



6) Navigate to the "dcgs-current-healthdata*" Data View and select edit for the app.Name field
    a. From Hamburger Menu select "Stack Management" -> "Data Views"
    b. Type "dcgs-current" in search bar and select the data view



7) Type “app.Name” in the search field and select the pencil icon to edit the field. Ensure you select the correct item with the “Url” Format.

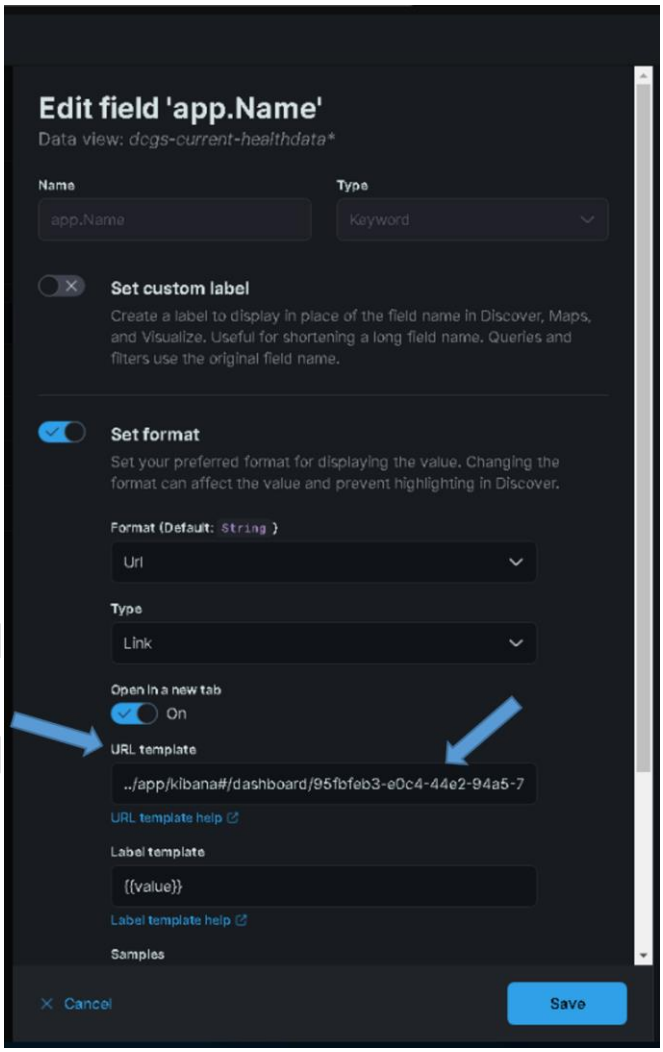


8) Replace the existing dashboard ID in the “URL template” field with the one you are holding in your clipboard (From the copy)
   a. Delete the existing id that is shown after the “dashboard/” in the template

      Example of what URL template looks like before removing the id:

../app/kibana#/dashboard/**b4acc984-701a-4d7b-94cd-08da2a60ee85**?_a=(query...

b. Right click and paste the id from your clipboard as the new id after the "dashboard/" in the template



c. Ensure the new URL link is correct with no extra spaces. Below is an example of the new URL link using the UUID from the example above:

../app/kibana#/dashboard/95fbfeb3-e0c4-44e2-94a5-79462de3e9e2?_a=(query:(language:kuery,query:'app.Name:"{{value}}"'))

Note: There is not spaces in the id (**95fbfeb3-e0c4-44e2-94a5-79462de3e9e2**)

9) Select "Save" on the Edit field 'host.name' pop-up.
10) Return to step 1 and validate the link now works.

5.5.9 Update Kibana Settings

Login into a Kibana node and run the following script to update global Kibana settings:

 # curl –k https://*{site code}*su01ro01.`hostname –d`/yum/elastic/install/update_kibana_settings | bash

This will set the new security banner at the top of each page in Kibana and enable dark mode. The security banner should be appropriate for the classification of the system Kibana is running on. If the banner does not look correct, contact an Elastic SME for guidance.

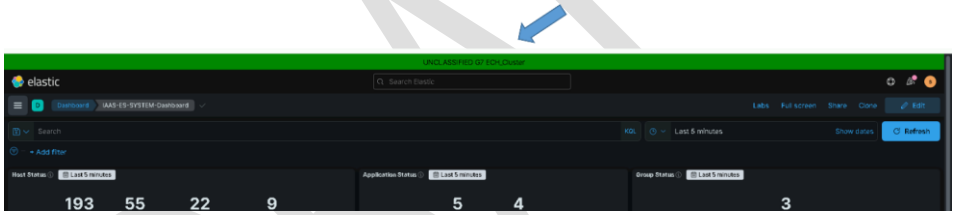


Figure 126 Example showing security banner and dark mode

5.5.10 Reindex existing data

NOTE: You must be root and a member of the **ent elastic admins** AD group to load saved objects into Kibana. Having the **Elastic Administrator** OneIM Role will place the user in this group.

In the 8.6.2 upgrade several fields have been updated in the ha and fmv indexes. Data that was previously indexed into elastic needs to be reindexed so it will be in the correct format. Running the script below automates this process.

If they exist, the following indexes will be re-indexed

```
"dcgs-filebeat-geo-ha-socet-socetevent-log"
"dcgs-filebeat-geo-ha-socet-socetraw-log"
"dcgs-filebeat-geo-ha-xplorer-ecs-log"
"dcgs-filebeat-geo-ha-xplorer-event-log"
"dcgs-filebeat-geo-ha-xplorer-notification-log"
"dcgs-filebeat-geo-fmv-maas-logs"
```

NOTE: Re-indexing may take a very long time. To avoid interruption of the re-index, the screen command will be used to create a session to run the install command.

1. Login to any Elasticsearch Node
   # sudo su

   # *screen -S install-session*

2. Run load `reindex_renamed_indices` script

```
# curl -k https://{site code}su01ro01.`hostname -
d`/yum/elastic/install/reindex_renamed_indices.sh | bash
```

- **If your SSH session times out while waiting for this script to run, return to your install-session by typing the following after re-establishing an SSH session to the computer.**
  # screen –d –r install-session
- **To detach from a running screen session type ctrl+a ctrl+d.**

5.5.11 Re-Activate Log Insight data Ingest

Note: A Log Insight SME is required for this section

To re-enable log forwarding from Log Insight to Logstash the following updates must be done on Log Insight at each site. Two forwarders will be created, one sending data in raw format ant the other sending syslog format.

Note: replace {xxx} below with the site designator: examples: “s00”, “t01”

Create the syslog forwarder:

4) Login to Log Insight web console and select “Log Management” from the menu on the left side.
5) Select “Log Forwarding” and then “New Destination.”
6) Use the following for the options:

Name:	Logstash-syslog
Host:	logstash
Protocol:	Syslog
Transport:	TCP
Filter:	“hostname” “does not match” “{xxx}sm*” “agentgenerated matches agentgenerated”
Port:	5050
Worker Count:	8

Create the raw forwarder:

Formatted: Normal

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Normal, Indent: Left: 1", No bullets or numbering

Formatted: Normal, Indent: Left: 1", No bullets or numbering

4) Login to Log Insight web console and select "Log Management" from the menu on the left side.
5) Select "Log Forwarding" and then "New Destination."
6) Use the following for the options:

Name:	Logstash-raw
Host:	logstash
Protocol:	Raw
Transport:	TCP
Filter:	"hostname" "does not match" "{xxx}sm*"
	"agentgenerated does not match agentgenerated"
	"text does not match */ifs/{xxx}/app/elac/*"
	"product does not match prelude"
Port:	5050
Worker Count:	8

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Normal, No bullets or numbering

Formatted: List Paragraph

5.5.12 Setup DLP data ingest from ESS (HBSS)

The following 2 sections describe two options for ingesting decoded DLP information from the EPO Server. The first option is the preferred method but cannot be used until ESS is running version 11.10.100 or later. Once ESS is upgraded to this level this option should be used. If ESS is running an older version, then the 2nd option must be used until ESS is upgraded.

Formatted: Superscript

5.5.12.1 Query using DLP API (Preferred option)

To use this method simply activate the esp_hbss_dlp pipeline that is included with the release. This is done by adding it to the logstash.yml configuration for the site that talks to the ESS EPO. A puppet update is required to make this configuration change. See section 5.5.2.1 for details on how to update the logstash.yml configuration for a site.

5.5.12.2 Receive DLP data from existing ArcSight connector (Short term work around)

As stated above this option is to be used only until ESS is upgraded to version 11.10.100 or higher. If this method is implemented, then once ESS is upgraded the ingest method should be changed to use the esp_hbss_dlp pipeline as described in the previous section.

To receive data from the Existing ArcSight connector do the following:

3) Activate the esp_hbss_dlp-via-connector ingest pipeline by adding it to the logstash.yml configuration. See section 5.5.2.1 for details on how to update the logstash.yml configuration for a site. This pipeline should only be added to the logstash configuration for the site where the EPO server resides.
4) Follow the install instruction provided for configuring the ESS ArcSight connector to forward data to Logstash: *"IAAS-018 – ESS – Temporary ArcSight Connector for Elastic Installation Instructions.docx"* (Document can be found in install/docs director on reposerver)
    a. Artifacts needed for the installation are distributed with the oadcgs-es-elastic-reposerver package in the install/artifacts/ess directory.

Formatted: Font: Italic

Formatted: Font: Not Italic

Formatted: Font: Italic

- DeployArcSightMod.zip – Contains artifacts needed for configuring ArcSight connector.
- DeployArcSightMod.text - Holds Hashes for Artifacts in zip.

3) Validate the DLP events are received in the dcgs-hbss_epo_dlp-iaas-ent index

5.5.13 Switch Syslog data ingest

This section is to configure the switches at each site to send syslog data directly to Logstash. Each switch should be setup to send syslog data directly to Logstash.

5.5.13.1 Prepare Logstash to receive switch data

Switch may send syslog data via UDP or TCP to Logstash. The ecp_syslog_tcp and ecp_syslog_udp pipelines have been added to allow the ingestion of this data. These new pipelines must be activated on each Logstash instance by adding them to the logstash.yml configuration.

Validate that the ecp_syslog_tcp and ecp_syslog_upd pipelines are contained in all logstash configurations defined in puppet. See section 5.5.2.1 for details on puppet configurations for logstash.

5.5.13.2 Configure switches to forward syslog data

The network/switch SMEs are needed to perform this configuration. Ensure that all switches are configured to send syslog data to one of the following logstash endpoints at each site:

TCP (Unencrypted) – Logstash: port 5055

UDP – Logstash: 514 (port forwarding is enabled on logstash to send this data to port 5040)
(**Note**: UDP data can also be sent directly to Logstash port 5040)

Provide network team with instructions to configure switches to forward data to Logstash at each site located in *10Appendix A Prime Update Instructions* or *"Prime Updates.docx"* (Document can be found in install/docs director on reposerver)

Artifacts needed for the installation are distributed with the oadcgs-es-elastic-reposerver package in the install/artifacts/prime_templates directory:

- Cisco_Prime_Logstash_Update_Templates.zip

5.5.14 Activate Serena data ingest

NOTE: You must be root and a member of the **ent elastic admins** AD group to load saved objects into Kibana. Having the **Elastic Administrator** OneIM Role will place the user in this group.

IMPORTANT: If Serena data was activated in a previous version, you can skip this step:

Formatted: Font: Italic
Formatted: Font: Italic
Formatted: List Paragraph, Bulleted + Level: 3 + Aligned at: 1.25" + Indent at: 1.5"
Formatted: Normal, Indent: Left: 0.25", No bullets or numbering
Formatted: List Paragraph
Formatted: Space After: 0 pt
Formatted: Font: Bold
Formatted: Normal, No bullets or numbering
Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"
Formatted: Normal, Indent: Left: 0.25"
Deleted: .

NOTE: An SQL Database SME will also be needed to give the elastic service account read permissions to the Serena database.

IMPORTANT: Prior to executing this section the service account at the site where this feature will be activated must be given permissions to read data from the Serena database by an SQL Database SME.

This section will active the ingestion of data from the Serena database. The following information will be needed and prompted for by the active script.

- HOSTNAME - Hostname of SQL server holding Serena database
- PORT_NUM - Port on the SQL server to access the database
- DOMAIN_NAME - Domain that this activation is being performed on
- INSTANCE_NAME - Serena database instance name
- DATABASE_NAME - Serena database name

These values will be used to create the JDBC connection string as follows:
*jdbc:sqlserver://**HOSTNAME:PORT_NUM**;domain=**DOMAIN_NAME**;instanceName=**INSTANCE_NAME**;databaseName=**DATABASE_NAME**;integratedSecurity=true;authenticationScheme=JavaKerberos\"*

This is an example of a connection string created by the activate script and added to the esp_serena pipeline.

*jdbc:sqlserver://**u00sm01sq20.dcgs.mil**:**1460**;domain=**dcgs.mil**;instanceName=**ES01**;databaseName=**SBM_APP_2012**;integratedSecurity=true;authenticationScheme=JavaKerberos"*

Follow these steps to run the active serena script:

4. Login to the Logstash Instance that will be used to communicate with the Serena database. The best choice for this would be the Logstash instance at the same site that hosts the Serena database.

   # sudo su

5. Run the active serena script

   ```
   # curl -k https://{site code}su01ro01.`hostname -
   d`/yum/elastic/install/activate_serena.sh | bash
   ```

5.5.15 Activate ACAS data ingest

NOTE: An ACAS SME will be needed to execute this section

IMPORTANT: This section depends on AD version 3.12: *RFC CR-2023-OADCGS-015 – Standard Change: Upgrade Enterprise Service Foundation Data Active Directory (ESFDAD) NOFORN from*

v3.3.11 to V.3.12. If Active Directory is not running this version skip this section, it will be attempted again in the next Elastic upgrade.

The Elastic Data Collector can query Vulnerability, Scanner Status, and System Status data from ACAS to be ingested into the collector. It uses the TenableSC API to be able to display this data. By default, this capability is disabled. The following guide will show the user how to activate the ACAS portion of the Data Collector.

5.5.15.1 Configure ACAS to allow API Keys

5. Log in to Tenable.sc via the user interface as an **Admin** (**Error! Hyperlink reference not valid.**).

Figure 127- Example of tenable login page

6. In the top navigation bar, click **System** > **Configuration**. The **Configuration** page appears.

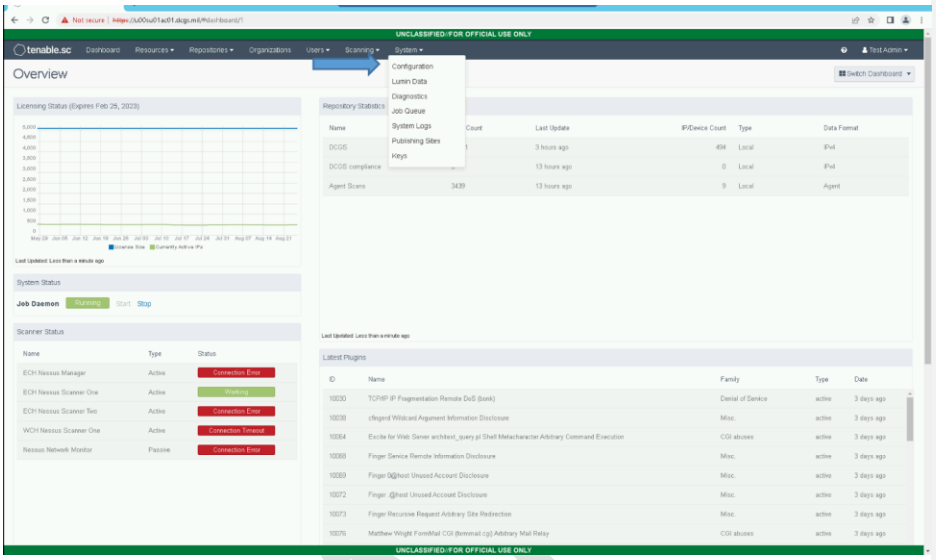


Figure 128- Example of selecting "Configuration" option.

7. Click the **Security** tile. The **Security Configuration** page appears.

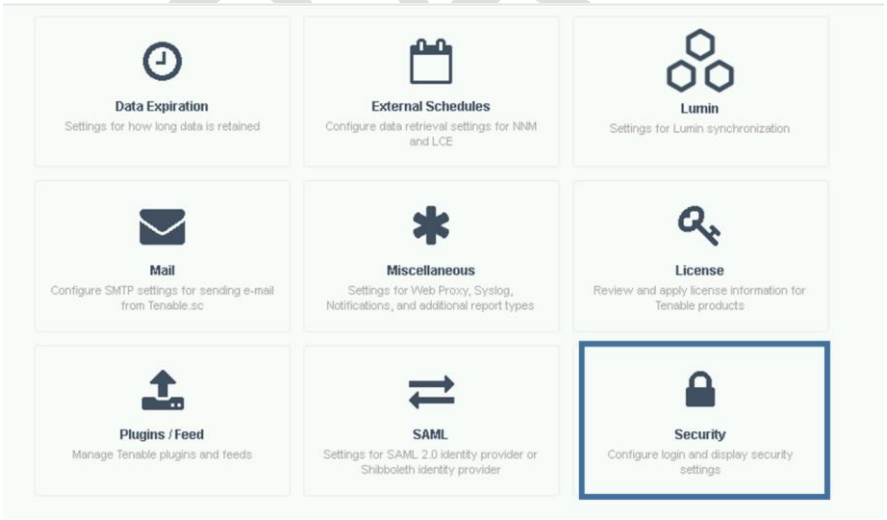


Figure 129- Example of selecting Security option.

8. In the **Authentication Settings** section, click **Allow API Keys** to enable the toggle. Click **Submit**.

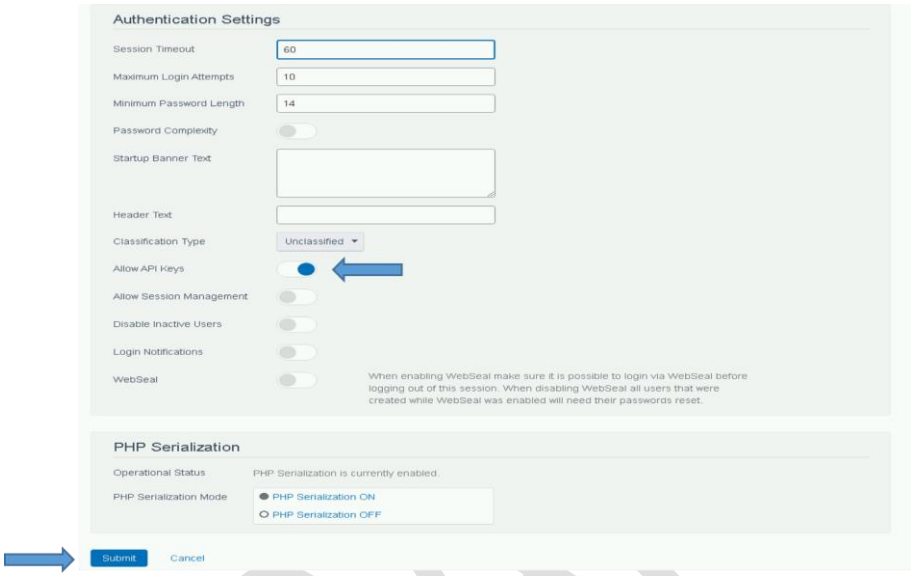


Figure 130- Example of turning on API Keys option.

5.5.15.2 Generate API Keys for the Elastic Service Account

API Keys for use by the elasticDataCollector should be generated from the “ent_elastic_acas.svc” account.

7. Log in to Tenable.sc via the user interface.

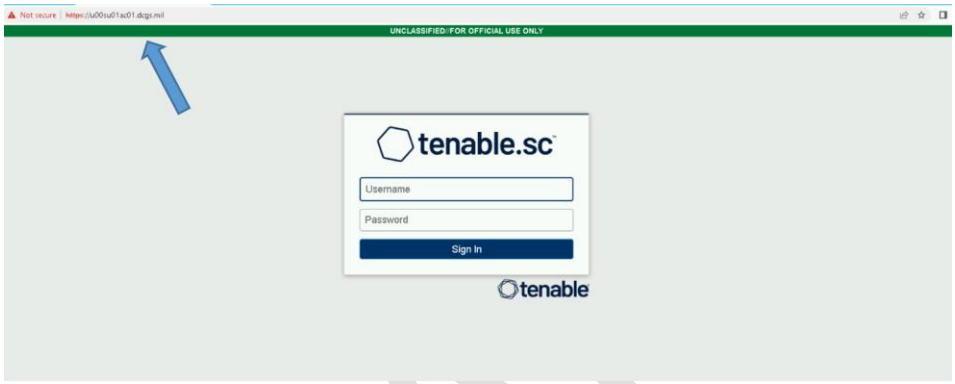


Figure 131- Example login page for tenable

8. Click **Users** > **Users**. The **Users** page appears.

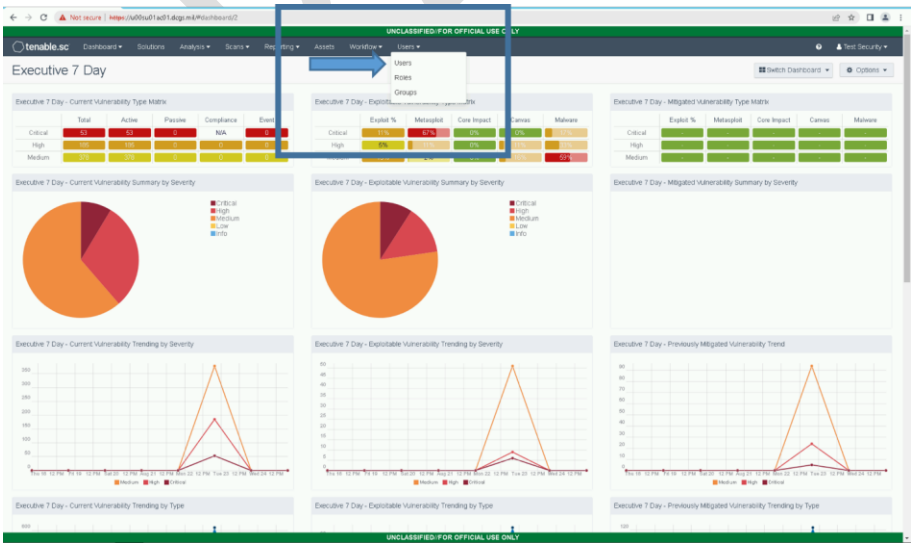


Figure 132- Example selecting "Users."

9. Click the **Settings Button** for the ent_elastic_acas.svc account to generate an API key, then **Generate API Key**

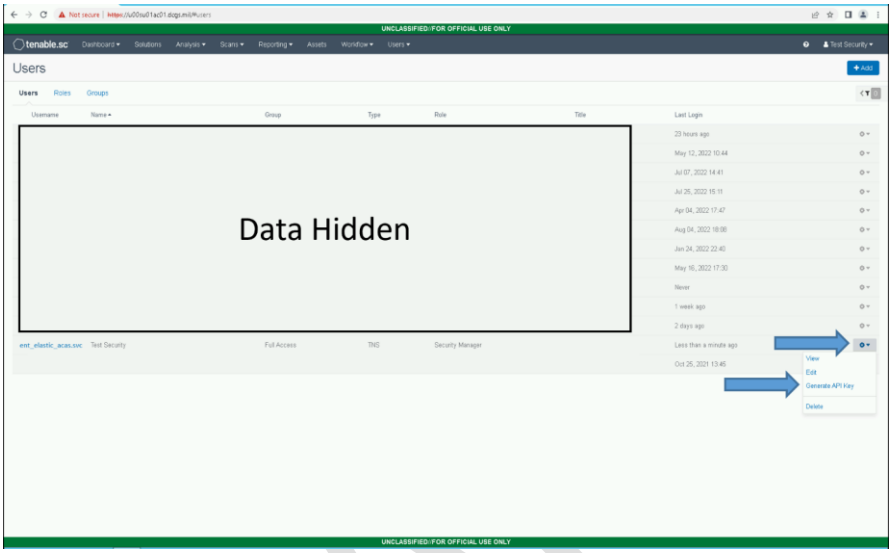


Figure 133- Example of selecting "Generate API Key" for a user.

10. Click **Generate**.

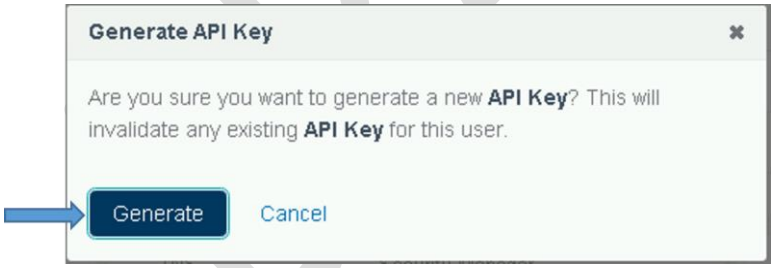


Figure 134- Example of Generate API Key confirmation.

11. The **Your API Key** window appears, displaying the access key and secret key for the user.

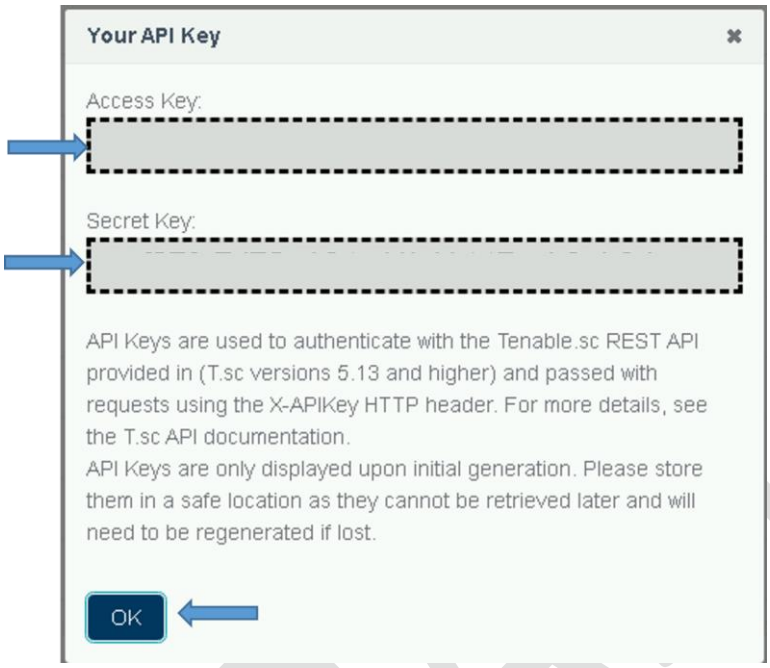


Figure 135- Example API Key Display

12. **Save the Access and Secret Keys for the Elastic Service Account**.

5.5.15.3 Activate ACAS for the ElasticDataCollector at the HUB

Log in to the Logstash VM at the site where the ACAS server resides (On production this is ECH) and perform the following steps.

7. Sudo to become root.
   #sudo su

   Run activate_acas.sh
   # curl -k https://{site code}su01ro01.`hostname -d`/yum/elastic/install/activate_acas.sh | bash

8. Input the Access and Secret Keys obtained above into the command prompt when prompted.

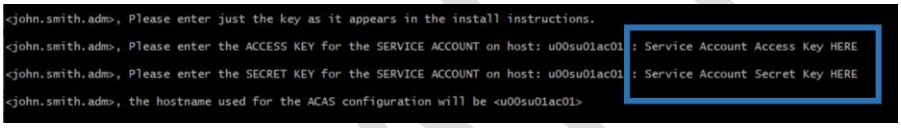


Figure 136- Example of prompts during install

9. The script will also create and assign the ACAS delete role to the querier user for the site where it is installed and create the acas.cron job. An example of the script running successfully:

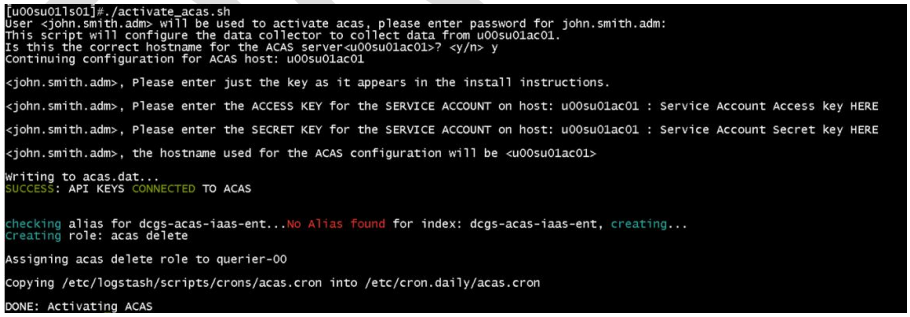


Figure 137- An example of the script running successfully

10. Restart the Elastic Data Collector
    # systemctl restart elasticDataCollector

11. To check if the script worked, look in /etc/logstash/scripts/data/acas.dat and see if the keys were created. It should look like:
    # cat /etc/logstash/scripts/data/acas.dat

12. Once configuration is complete, verify ACAS data is received by going to **Kibana Discover** and selecting **dcgs-acas-*.**

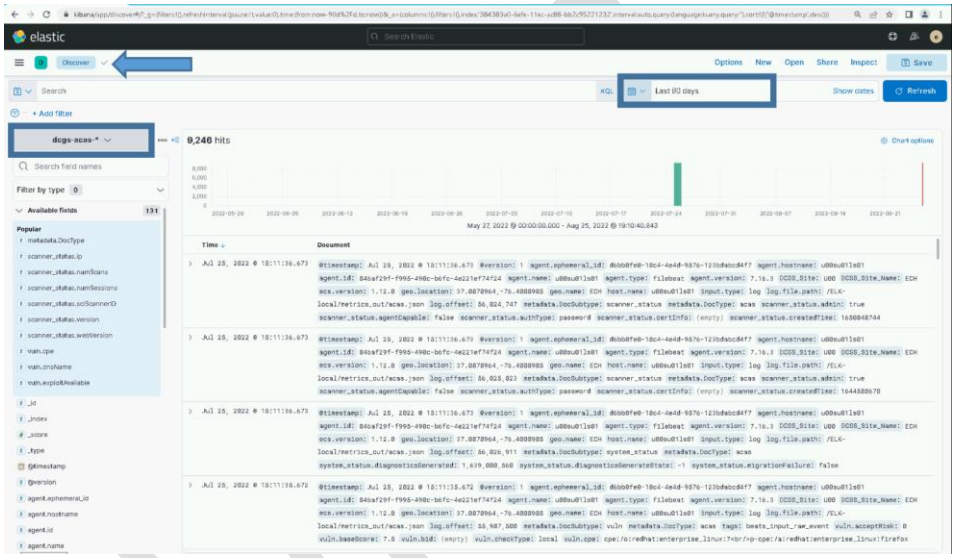


Figure 138- Discover showing example acas data

5.5.16 Remove "Run and Remove" scripts from system when upgrade is completed

The scripts that are delivered in the install directory on the repo servers are intended for use during this upgrade. Once the upgrade is completed this directory and these scripts may be removed from the system as they are not used for operational purposes.

5.6 List of Changes

See highlights of new features in Elastic, starting with the 8.6 release for each product.

Elastic 8.6: https://www.elastic.co/guide/en/elasticsearch/reference/8.6/release-highlights.html

Kibana 8.6: https://www.elastic.co/guide/en/kibana/8.6/whats-new.html

Logstash 8.6: https://www.elastic.co/guide/en/logstash/8.6/releasenotes.html

Beats 8.6: https://www.elastic.co/guide/en/beats/libbeat/8.6/release-notes.html

6 De-Installation (Back Out) Instructions

After upgrading the Elasticsearch version, it cannot be downgraded. If you wish to run an older version, the entire cluster must be removed and re-installed. See the De-Installation (Back Out) Instructions in *ES-018 – Elastic Logging and Aggregation Cluster (ELAC) – System Installation Instructions* for details.

6.1 Elastic Data Collector Back Out Procedure

The Elastic Data Collector may be removed from each Logstash instance by:

1. `# Systemctl stop elasticDataCollector`
2. `# rm –f /etc/systemd/system/elasticDataCollector.service`
3. `# systemctl daemon-reload`
4. `# rm –rf /etc/logstash/scripts`

6.2 Domain Controller GPO Back Out Procedure

The GPO that is used to install beats on the domain controllers can be removed by following backout instruction in the *ES-018 - Active Directory - DCMedia DFS for Elastic MetricBeat Installation Instructions*.

7 Frequently Asked Questions

1. What does a standalone cluster in the monitoring section of Kibana indicate concerning pipelines?

   If you see **standalone cluster** when you select **Stack Monitoring** it's an indication that there is an issue with one of the pipelines on a Logstash Instance. Look at the pipeline that is listed in the standalone cluster and view the Logstash log file on the problem host to determine the issue. After resolving make sure to restart Logstash as this issue will not resolve on a pipeline reload.

2. What are some useful GET commands to use in Dev Tools?

   GET _cat/aliases/*?v&s=is_write_index
   GET _cat/shards?v&s=state
   GET _cluster/settings
   GET _cluster/health
   GET _cat/nodes?v&h=name,ip,version&s=name
   GET _ml/info
   GET _cluster/settings
   GET _cat/templates/esti*?v&s=name
   GET _cat/aliases/*beat-7.16.3*?v&s=is_write_index
   GET _cat/indices/*?v

3. Why does **/var/lib/logstash** keep having its ownership permissions changed?

   This is an RPM verify issue. Section 6.6.2.1.1 of the 7.9.1 install instructions details how Puppet installs a script called rpmverify.sh, which is run by cron. **crontab -l** will show it. If the exclusions for Logstash, Elastic, and Kibana are not added, it will put the permissions back to what is set in the RPM. Please check with the OSIF team to ensure the exclusions are in place or you will also have issue on the Elastic nodes. To verify the status of this fix cat the **/etc/rpm-verify-exclusions** file and look for Logstash, Elastic, and Kibana from any Logstash Server.

4. Why can't I get databases to ingest into Elastic?

   AOA group names may NOT be what was originally defined. There is likely a wrong connect string for the AOA groups on the elastic side. It also likely has to do with the SPNS not existing in AD as covered before in https://jira.di2e.net/browse/DCGSCM-4190.

   You must have the correct SPNs for the AOA groups for any chance of Kerberos working. In G7 we use an AD account to connect to SQL from their systems and that uses Kerberos, which needs the correct SPNS. IDAM would likely be the same issue if permissions were given to the account you need to separate out the app from the database. If the SPNS are not there for the AOA group that the connect string is calling correctly there is ZERO chance it will work.

8 References

1. ES-018 – Elastic Logging and Aggregation Cluster (ELAC) – System Installation Instructions
2. ES-018 – SCCM – Instructions for Building an SCCM Package to Install Beats for Windows
3. ES-018 - Active Directory - DCMedia DFS for Elastic Metricbeat Installation Instructions
4. ES-018 - Microsoft SQL – Configuring SQL for Elastic Monitoring Instructions
5. ES-018 - ESS - Syslog Publishing of ePolicy Orchestrator Events to Elastic

9 Test Results

LEAVE THIS BLANK BUT DO NOT DELETE.

FILL OUT THE IAAS-023 TEST REPORT TEMPLATE.

10 Test Procedures

DO NOT FILL THIS OUT. DO NOT DELETE.
FILL OUT THE IAAS-023 TEST REPORT TEMPLATE.

Sprint:	Sprint it is tested in	**Epic:**	Epic as found on JIRA dev task
User Story:	Main task # (link)	**Test Procedure Name:**	
Test Procedure #:	Test Task Jira Ticket # (link)	**Release:**	1
System / Component:	System/Component as found on JIRA dev task	**Test Purpose:**	DT
Test Engineer:	Engineer Name	**Execution Date:**	dd-month-yy
Test Environment:		**Test Description:**	
Prerequisites:			
Estimated Implementation Time:			
Test Location:	Indicate if the test/install will be conducted in NOFORN, REL, or both NOFORN and REL.		
Notes:			

Acceptance Criteria:	
Overall Comments:	N/A
Overall Test Result:	P

Step	Action	Expected Result	Pass/Fail	Comments
1	I click X	Y Displays	P	
2			PWE	
3			F	

Appendix A Prime Update instructions

Note: The Cisco_Prime_Logstash_Update_Templates.zip file contains the following 2 templates needed below:

- IOS-XE Add DNS and Logstash
- NX-OS Add DNS and Logstash

1. Log into Cisco Prime
2. Navigate to Menu > Configuration > Global Variables
3. Click **Add**
4. Create the following 3 global variables

Name	Description	Type	Value	Display Label
gv.dns_1	First DNS Server	IPv4 Address	<enter dns 1 IP>	DNS-1
gv.dns_2	Second DNS Server	IPv4 Address	<enter dns 2 IP>	DNS-2
gv.siteSpecificDomain	Site Specific Domain	String	<enter site domain>	Site Specific Domain

5. Navigate to Menu > Configuration > Templates > Features & Technologies
6. Navigate to Templates > My Templates > CLI Templates (User Defined)
7. Click Import
    a. Change folder to CLI Templates (User Defined)
    b. Click Select Templates
    c. Select the following templates:
        i. IOS-XE Add DNS and Logstash
        ii. NX-OS Add DNS and Logstash
    d. Click OK
8. Select IOS-XE Add DNS and Logstash
9. Ensure all IOS-XE switches are selected under Devices
10. Click Next
11. Ensure Work Flow is selected
12. Click Next
13. Ensure all variables are correct. These should be pulled from the global variables set earlier
14. Click Next
15. Schedule Job

Job Name	Logstash IOS-XE <ENTER DATE>
Start Time	Now
Recurrence	None
Failure Policy	Stop on Failure
Copy Running Config	Checked

Deleted: Acronyms
Formatted: Space After: 0 pt
Formatted: Space After: 0 pt
Formatted
Formatted: List Paragraph, Indent: Left: 1"
Deleted: Acronym

Archive Config after Deploy	Checked

16. Click Next
17. Click Finish
18. Click Job Status
19. Navigate to User Jobs > Config Deploy
20. On failure click the job Name
    a. You should be able to see any issues populated here. Fix these then retry.
21. On success repeat for NX-OS from step 8 but selecting the NX-OS job and devices.
22. Check with Elastic to ensure they are receiving required logs.

Appendix B Known Issues

You can delete this if there are no known issues.