Air Force
Distributed Common Ground System (AF DCGS)

Elastic Logging and Aggregation Cluster (ELAC) 8.11.3 Upgrade Instructions

13 February 2024

AFLCMC/HBGEB
750 Third Street
Robins AFB, GA 31098

Controlled by: USAF
Controlled by: AFLCMC /HBGB/AFRL/RIEB
CUI Category: Controlled Technical Information
Distribution/Dissemination Controls: Distribution D
POC: AFRL.RIE.OADCGS@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.

The revision numbers within the Change/Revision Record should also coincide with the AF DCGS document control number (DCN) given to this document, i.e., for DCGS DCN, DCGS-TECH-GT1-0025_Rev-1, the last line in this table should be Rev-1.

CHANGE / REVISION RECORD

Revision	Date	Page/Paragraph	Description of Change	Made By
Initial	11 Oct 2023		Initial Release	S. Truxal/AFLCMC/ES
			DCN Assigned for: NRM_ESXXXXX	AFLCMC/HBGB/ C/DM
Rev-1	13 Feb 2024		MTE Redlines	S. Truxal/AFLCMC/ES

Table of Contents

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

1. INTRODUCTION

This document is intended to be used to:

- Upgrade Enterprise Elasticsearch from version 8.6.2 to version 8.11.3.
- Install 2nd Elastic Cluster at WCH
- Configure Cross Cluster Replication between Clusters
- Transition to Searchable Snapshots for long term storage
- Add Site Spaces to Kibana

If Enterprise Elastic is not currently installed in the environment, please refer to *DCGS-TECH-ENTSVCS-2435_Upgr-Instr_ELAC.pdf* to install the initial version before executing these procedures. If Elasticsearch is installed but running a version older than 8.6.2 please consult with an Elastic Subject Matter Expert (SME) before executing this upgrade.

1.1 Location of Installation

The installation will be conducted in NOFORN and REL environments.

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 [8.11] | Elastic
  - Kibana - Release notes | Kibana Guide [8.11] | Elastic
  - Logstash – Release notes | Logstash Guide [8.11] | Elastic
  - Beats – Release notes | Beats Guide [8.11] | Elastic
- Searchable Snapshots
  - ILM Cold and Frozen Phases
  - New Baseline Space
  - Site Spaces
- **Updates for Production Enclave Only**
  - Addition of WCH on Production system (WCH)
  - Cross Cluster Replication
  - Automated Logstash Failover
- Addition of ART integrations
  - Unicorn
  - ECP

1.2.1 Detailed Change Summary:

1.2.1.1 Changed

- Updates all scripts to use Satellite/Capsule servers instead of repo servers

- Updates configuration of node roles during an upgrade, adding roles for cold, frozen, and transform
- Updates dcgs_default_policy to add cold phase and creates the repository for these indexes
- Updates Cleanup.py to be able to run multiple different delete queries
- Updates the elasticDataCollector to handle automatic Logstash failover should connection to the cluster be lost
- Updates the RequestHandler.py class to return better formatted and more detailed API calls
- Updates metricbeat.yml to add an app config file option
- Corrects some deprecations that arose from upgrading to Elasticsearch 8.X
- Updates Logstash log to squash Logstash Info messages
- Updates puppet to ensure kibana.yml in /etc/filebeat/modules.d when Kibana installed
- Updates get_ldap_hosts.sh.epp to only query hosts at the site it is running from
- Updates all.module.system.yml.epp to add udf to the array of filesystems to be ignored to prevent monitoring issues
- Updates to ensure the presence of the icmp_include.txt file
- Updates heartbeat.yml.epp to to use “logstash” instead of “logstash-xxx”
- Updates upgrade_logstash.sh script to communicate with both ECH and WCH clusters on prod.
- Updates the Logstash admin username and password in the logstash.yml.epp
- Bug Fixes
    - Updates AuditCheck to run every five minutes instead of every 24 hours
    - Updates SCCM Monitoring template to remove looking for "IISADMIN" and "W3SVC"
    - Corrects a decode function from updatepasswd that was causing errors
    - Updates CiscoSwitch to not calculate utilization for VLANs
    - Update ids used for documents in current-healthdata index
    - Updates all.module.system.yml to add udf to the array of filesystem.ignore_types to prevent showing CDs/DVDs causing disk full symptoms
    - Updates the index privileges to remove “manage” from Jr Kibana Admin
    - Corrects CiscoSwitch.sh script to fix reversed Catalyst inlet and CPU temps
    - Updates esp_linux_syslog pipeline to mark “wu” (Workstation Unix) boxes as “Linux”
    - Updates version determination for components during installation
    - Corrects FX2 symptoms showing “NONE symptom” when there are issues
    - Updates to only authenticate to domain controllers at cluster site
    - Updates Data Collector to correct reference before assignment and formatting bugs

1.2.1.2 Added

- Adds Cross Cluster Replication
- Adds transition to Searchable Snapshots
- Adds the Ashes ST integration (Partial)
- Adds UNICORN integration.
- Adds ECP Integration.
- Adds boostrap_site_specific.sh to allow running bootstrap only at sites
- Adds 8.11.3 Ingest Pipelines and Templates
- Adds the esw_current-healthdata-updater Watcher
- Adds ingest pipelines for Kibana auditing data
- Adds a new Baseline space that will hold latest baseline artifacts (dashboards, visuals, etc.)

- Adds the ability for threads running in the Elastic Data Collector to automatically restart
- Adds a new class ThreadInfo which holds the Data Collector threads and other information
- Adds new component and index templates for Kibana login data
- Adds a Cyber Ops space
- Adds site spaces
- Adds puppet.Reason for updated Puppet visuals

1.2.1.3 Removed

- Removes unused Ingest Pipelines and Templates from distribution (8.9.0 ingest pipelines, 8.6.1 ingest pipelines, est legacy templates)

2. SYSTEM ENVIRONMENT

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

2.1 Elastic Search Cluster

IMPORTANT: Cluster Configurations have changed for this upgrade and must be made before executing upgrade procedure. See tables below for resource allocation changes.

Optional Cluster Sizes – This install document is intended to be used for different Elastic Cluster sizes used in different environments For the Virtual Machine (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.
- **Cold Data node (c):** Cold data nodes are part of the cold tier. When you no longer need to search time series data regularly, it can move from the warm tier to the cold tier. While still searchable, this tier is typically optimized for lower storage costs rather than search speed.

- **Frozen Data node (f):** Frozen data nodes are part of the frozen tier. Once data is no longer being queried, or being queried rarely, it may move from the cold tier to the frozen tier where it stays for the rest of its life.
- **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.
- **Transform node (t):** A node that has the `transform` role. If you want to use transforms, there must be at least one transform node in your cluster.
- **Remote-eligible node (r):** A node that has the `remote_cluster_client` role, which makes it eligible to act as a remote client.

2.1.1 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	mr
elastic-node-2	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	mr
elastic-node-3	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	mr
elastic-node-4	Linux	8	64	*OS 90 GB*	Data - 100GB(SSD)	/ELK-local	lrst
elastic-node-5	Linux	8	64	*OS 90 GB*	Data - 100GB(SSD)	/ELK-local	lrst
elastic-node-6	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	hirs
elastic-node-7	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	hirs
elastic-node-8	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	hirs
elastic-node-9	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	hirs

VM Description	OS	VCPUs	RAM (GB)	Disk 0	Disk 1	Data Dir	Role
elastic-node-10	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-11	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-12	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-13	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-14	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-15	**Linux**	**8**	**64**	***OS 90 GB***	**Data – 100GB(SSD)**	**/ELK-local**	**f**

2.1.2 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	mr
elastic-node-2	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	mr
elastic-node-3	Linux	4	8	*OS 90 GB*	N/A	/ELK-local	mr
elastic-node-4	Linux	8	64	*OS 90 GB*	Data - 100GB(SSD)	/ELK-local	lrst
elastic-node-5	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	hirs
elastic-node-6	Linux	8	64	*OS 90 GB*	Data - 2TB(SSD)	/ELK-local	hirs
elastic-node-7	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-8	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-9	Linux	8	64	*OS 90 GB*	None(nfs)	/ELK-nfs	cirw
elastic-node-10	**Linux**	**8**	**64**	***OS 90 GB***	**Data – 100GB(SSD)**	**/ELK-local**	**f**

2.1.3 7 Node Cluster - Small (REL)

The clusters for Elastic will reside at the hubs. Each cluster will initially be made up of 7 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 7 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	ciw
elastic-node-5	Linux	4	64	*OS 90 GB*	None(nfs)	/ELK-nfs	ciw
elastic-node-6	Linux	4	64	*OS 90 GB*	None(nfs)	/ELK-nfs	ciw
elastic-node-7	Linux	4	64	*OS 90 GB*	Data – 100GB(SSD)	/ELK-local	f

2.1.4 Logstash Nodes

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

2.1.4.1 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

2.1.4.2 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 2.1.4 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 upgrade Elasticsearch and it's components from 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-8.11.3.tar.gz
  2. Elastic_Linux_Beats-8.11.3.tar.gz

- 02.zip file
  1. oadcgs-es-elastic-sccm-2.1.38.1.zip
  2. Elastic_Windows_Beats-8.11.3.zip

- 03.zip file
  1. oadcgs-es-elastic-reposerver- 2.1.38.1.tar.gz
  2. oadcgs-dsil_elastic_clients-2.1.1.tar.gz
  3. oadcgs-dsil_elastic_servers-1.2.18.tar.gz

There should be an Elastic repo on the OA DCGS repo server where the RPMs will be loaded for the installation. 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**. Starting with this release Satellite will be used to sync repositories between sites.**

The following RPMs should be 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-8.11.3.tar.gz
    - elasticsearch-8.11.3-x86_64.rpm
    - kibana-8.11.3-x86_64.rpm
    - logstash-8.11.3-x86_64.rpm
2. Elastic_Linux_Beats-8.11.3.tar.gz
    - metricbeat-8.11.3-x86_64.rpm
    - filebeat-8.11.3-x86_64.rpm
    - heartbeat-8.11.3-x86_64.rpm

NOTE:

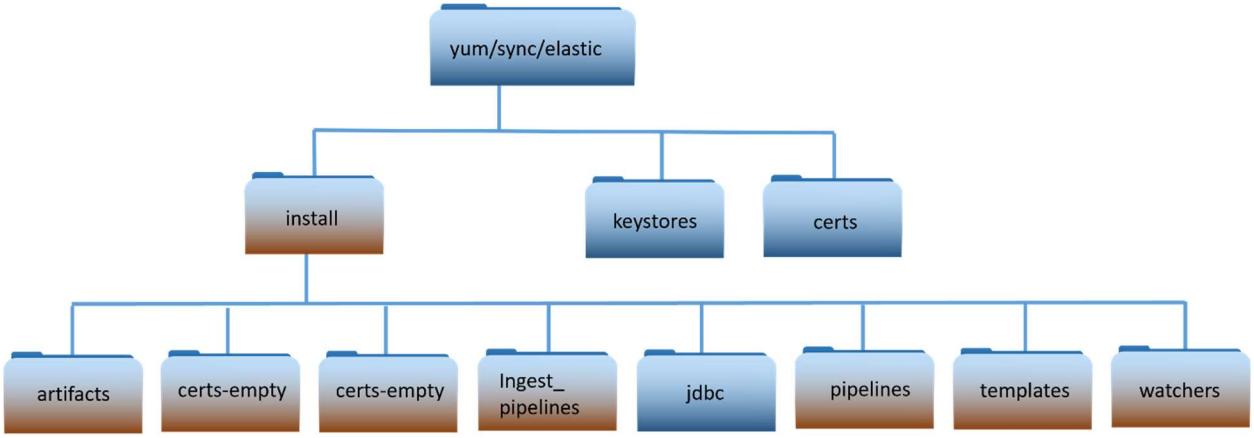
When installing Elasticsearch, Logstash and Kibana, only those 3 RPMs are placed on the repo server to install/upgrade the core components prior to adding any Beat RPMs. Beat collectors are automatically installed/upgraded so this should not happen until the Elastic cluster is upgraded, running, and 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 sync folder on the repo server. These directories will be populated when extracting the **oadcgs-es-elastic-reposerver- x.x.x.x.tar.gz** archive.

NOTE: This directory is moved to a new location starting in this upgrade "yum/sync/elastic"

NOTE for Upgrades:

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_unicorn - Configures and loads unicorn pipeline into Kibnaa
- bootstrap_indexes.sh – Ensures aliases for indexes are configured
- create_spaces.sh – Creates site spaces in Kibana
- installKibana.sh – installs Kibana
- installElasticNode.sh – Installs an Elasticsearch node
- installElasticDataCollector.sh – Installs elastic data collector on a Logstash instance
- installMetricbeatkeystore.sh – Creates Metricbeat user and updates the Metricbeat keystore
- installWatchers.sh – installs watchers into Kibana
- load_auditsettings.sh – Loads/Updates audit settings for Elastic
- load_CCR.sh – Sets up Cross Cluster Replication(CCR) for indexes
- load_objects.sh – Loads all objects that come with the release into baseline space
- load_ILM_Policies.sh – Loads Index Lifecycle Management Policies
- make_elastic_csrs.sh – Creates certificate requests for Elasticsearch nodes
- update_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
- load_role_mappings.sh – Loads mappings to Active Directory groups into Kibana
- update_default_space.sh – backs up default space and makes new “baseline” space
- update_ingest_pipelines.sh – Loads all ingest pipelines into Elastic
- update_kibana_settings.sh – Updates kibana spaces with security banner
- updateLicense.sh – updates the Elastic License into the cluster
- update_lifecycle.sh – updates existing indexes with updated lifecycle policies
- 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_node.sh
- verifyArchiveDir.sh – ensure archive directory is present on Isilon share

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 installs and configures Filebeat on hosts running mission applications to collect log files.
Version:	2.1.1
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/modules/docker.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:	1.2.18
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 upgrade from the 8.6.2 version of Elastic to 8.11.3. If you are running a version of Elastic prior to version 8.6.2 please refer to one of the following:

- *ES-018 - Elastic Logging and Aggregation Cluster (ELAC) – 8.6.2 Upgrade Instructions.docx*

5.1 Estimated Implementation Time

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

- Cluster Install (WCH Only): ~ 2 days (Includes time to create VMs)
- Cluster upgrade: ~ 8 hours
- Site upgrade: ~ 1 hour

5.2 Cleanup of Existing Versions and Files

N/A

5.3 Pre-installation updates

Any preparation that is needed before the installation begins will be done in this section.

For this and future versions Elastic will be using the Satellite server to make synchronization between site easier. To allow the use of the Satellite server the elastic repository and installation artifacts are being separated into two locations:

1. The “Elastic_Files” location that will contain:
    - The “install” directory with scripts and artifacts used for upgrades.
    - The “keystores” directory which holds all the existing system specific keystore information.
    - The “certs” directory which holds all the existing system certificate information.
2. The “Elastic” yum repository which holds all Elasticsearch rpms.

When using Satellite files are still initially added to the repository server at the hub (where sat01 is running) and then will be replicated to the sites by satellite.

5.3.1 Updates to Cluster Nodes (MTE/CTE Only)

Roles Required: Infrastructure/VM Administrator

Before beginning the upgrade updates required for the cluster nodes need to be made. Refer to the tables in section 2.1 for details on the cluster configurations.

1. MTE/CTE – 10 Node Cluster
    - Add 100GB SSD drive to Elastic Node 10

5.3.2 Add new DNS alias

Required Roles: Windows Administrator

To allow consistent access to installation files a new DNS alias "satrepo" is added at each site as part of this upgrade. This new alias must be added to each site before starting the installation. This new alias is needed to run the install scripts throughout this document.

Elastic RPMs: The rpm locations will remain the same on the repository server

Repository server location: yum/elastic

Accessing: https://satrepo/Pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic

Elastic installation files: The location of these files have changed

Repository server location: yum/sync/elastic

Accessing: https://satrepo/Pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files

Add a new "satrepo" DNS Alias at each site:

3. If site is location of Satellite server(ex: ECH) then "satrepo" should be an alias for that server.
    - Example: On Production Low "satrepo" would point to s00su01sat01
4. If site is location of Capsule Server then "satrepo" should be an alias for the cap01 server.
    - Example: On Production Low at site 01 "satrepo" should be an alias for s01su01cap01

NOTE: If you have questions on setting up this new alias contact an Elastic SME for guidance. The installation steps below will not work without these aliases being setup correctly.

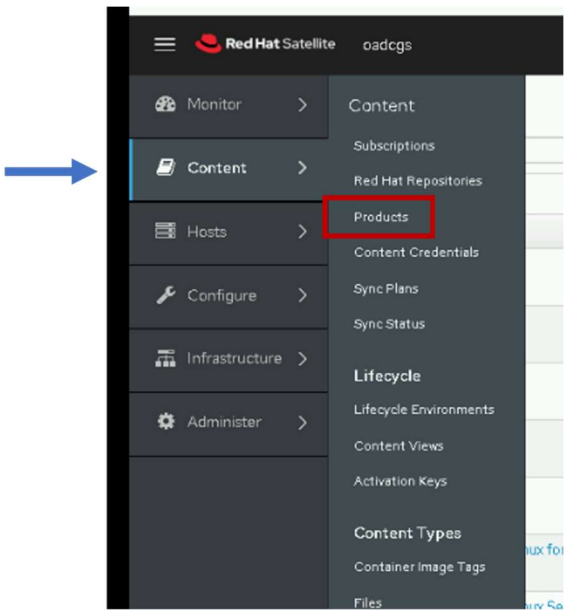
5.3.3 Ensure Satellite is configured properly for Elastic repositories.

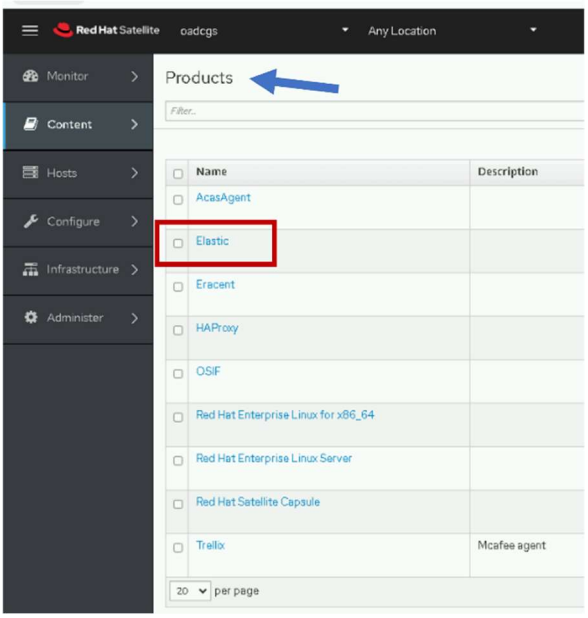
Required Roles: Satellite Administrator

Additional References: IAAS-024-Satellite Server – System Administrator Guide

This step is to ensure that the “Elastic” Product and its repositories exist on the Satellite Server. For this and all subsequent versions of Elastic the Satellite server will be used to ensure consistent delivery of all rpms and installation artifacts to all sites.

Ensure that the Elastic Product is created and configured correctly in satellite.





The Elastic Product should have two Repositories:

- Elastic – yum repository that holds all elastic rpms.
- Elastic Files – File repository for installation scripts/artifacts.

The "Elastic" (yum) repository should have the following settings:

Name: Elastic

Label: Elastic

Type: yum

Upstream URL: https://<fully qualified name of repo server>/yum/elastic

Example: https://u00su01ro01.ech.dcgs.mil/yum/elastic

Published At: https://<fully qualified name of satellite server>/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic/

Example: https://u00su01sat05.ech.dcgs.mil/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic/

The "Elastic Files" (file) repository should have the following settings:

Name: Elastic Files

Label: Elastic_Files

Type: file

Upstream URL: https://<fully qualified name of repo server>/yum/sync/elastic

Example: https://u00su01ro01.ech.dcgs.mil/yum/sync/elastic

Published At: https://<fully qualified name of satellite server>/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/

Example:
https://u00su01sat05.ech.dcgs.mil/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/

5.3.4 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 file repository at the site of the Satellite server. Once placed there the contents will need to be sync'd with Satellite and the new content deployed.

5.3.4.1 Update sync directory on Master Repo

This is the first release where we will be using the yum/sync/elastic directory for our Files Repository. We need to make sure it is configured properly.

1. Login to the Master Repository and ensure there is a "yum/sync/elastic" directory. If not create it.
2. Move the keystores and certs directory from "yum/elastic" to "yum/sync/elastic"
    a. cd /var/www/html/yum/sync/elastic
    b. mv ../../elastic/keystores .
    c. mv ../../elastic/certs .
3. Before updating the "yum/sync/elastic" with the new install directory for 8.11.3 it should only contain 2 directories. These directories were moved from the previous location in step 2.
    a. keystores
    b. certs

5.3.4.2 Update install directory with 8.11 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.

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

    # sudo su
    # cd /var/www/html/yum/sync/elastic

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 are now ready to be deployed by the Satellite administrator. Check with the Satellite admin to ensure the files are synced and the new content is deployed.

NOTE: You can update the repo with the new core rpms before deploying this update as they can both be deployed together.

Ensure the following is done by the Satellite administrator:

1. The PULP_MANIFEST must be regenerated by running pulp-manifest on the updated directory. The pulp-manifest command is only available on the Satellite server. To run it on the updated "yum/sync/elastic" directory the Satellite administrator will need to mount the yum directory from the repo server onto the satellite server. Once mounted change into that directory and run:

   ```
   # pulp-manifest ./
   ```

2. The updated file repository is "synced" from the satellite server.
3. The updated content view is published from the satellite server.

5.3.4.3 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
   ```

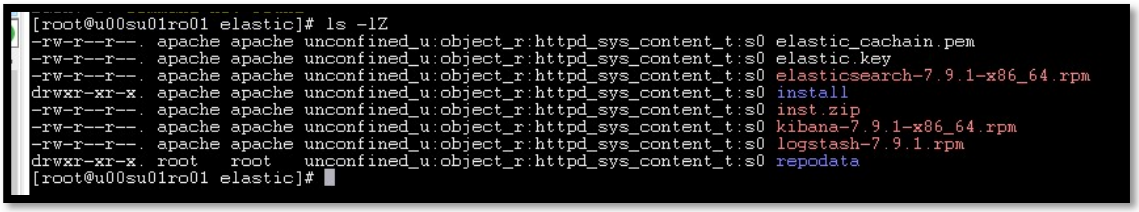


Figure 1. Example of correct file permission in the Elastic repo

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.

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

8. Ask Satellite administrator to re-sync yum repository and publish updates.
   a. The updated yum repository is "synced" from the satellite server.
   b. The updated content view is published from the satellite server.

5.4 Addition of second cluster at WCH

The following section should only be performed on the production enclaves and will add a second Elasticsearch Cluster at WCH. If you are working on CTE, MTE or REL please proceed to section 5.6 Installation Instructions for Upgrades.

5.4.1 Pre-Installation Checks

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

- The 15 Elasticsearch VMs are allocated; refer to section 2.1.1

5.4.1.1 Accounts and Passwords

- You have the password for the 0a_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**

5.4.1.2 Storage

5.4.1.2.1 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 table 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

5.4.1.2.2 NFS

- The /ELK-nfs directory on all Elastic nodes should be an NFS mount to the **elac** share on the WCH Isilon. This mount point is set up by the **dsil_elastic_servers** puppet module (section 5.8.2).
- The **0a_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

5.4.1.3 DNS Aliases

- DNS Aliases are set:
  - elastic-node-1
  - elastic-node-2
  - elastic-node-3
  - elastic-node-x
  - kibana-wch (WCH NSX Load Balancer should be used for this IP)

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

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

  After this upgrade the following URLs will be valid:

  https://kibana – Access to primary cluster - ECH Kibana instances

  https://kibana-wch – Access to secondary cluster – WCH kibana instances

5.4.1.4 Obtain PKI Certificates

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

5.4.1.4.1 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-wch
- kibana-wch.wch
- kibana-wch.{fully qualified} (ex: kibana-wch.wch.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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/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: u0asu01el01.crt, u0asu01el02.crt
Private Keys: {hostname}.key	examples: u0asu01el01.key, u0asu01el02.key

5. Ensure the elastic_cachain.pem file exists in the certs directory and contains all the root and sub-ca certificates for the system being installed. This file will be copied over to the /etc/elasticsearch/certs directory and renamed to cachain.pem when Elasticsearch is installed on each node.

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

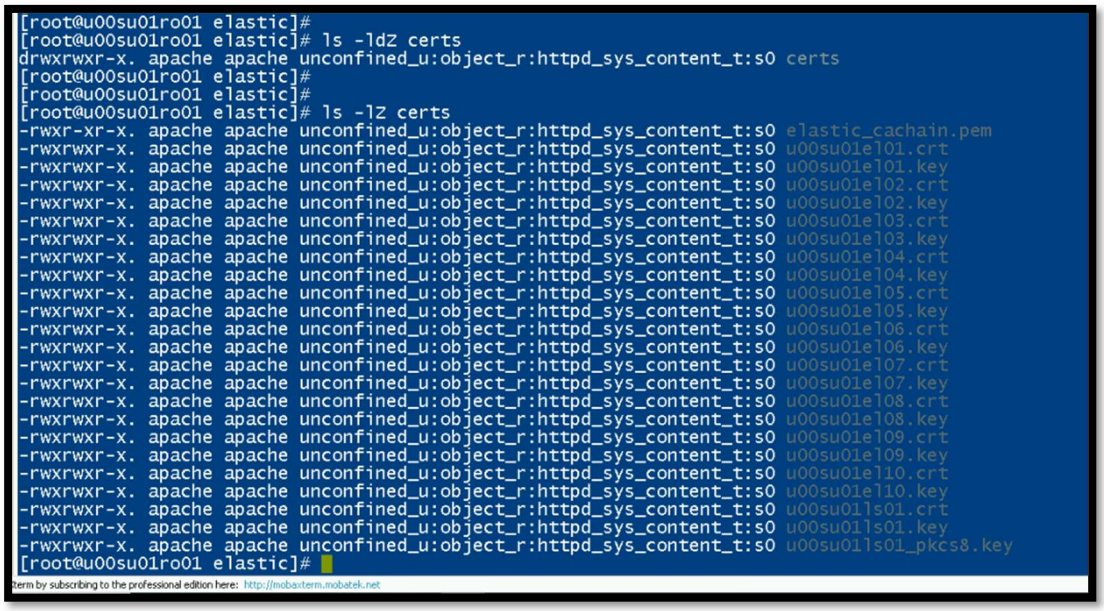


Figure 2. ls -lZ on certs directory

NOTE: The image above only shows the ECH certs, you will see both ECH(00) and WCH(0a) certs in this directory after updating.

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

   # restorecon *

9. Ask satellite administrator to regenerate PULP_MANIFEST, re-sync and publish the updates

5.4.1.5 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 at WCH, can be accessed by navigating to https://kibana-wch on the DCGS system. This URL directs the user to a load balancer that will forward the requests to the appropriate Kibana instance at WCH. 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.*

Kibana instances will be running on nodes 10 and 15 on the WCH Cluster.

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-wch.wch.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-wch.wch.dcgs.mil`**
   `nslookup` **`kibana-wch.wch.dcgs.mil`**

IMPORTANT: In the 8.6 upgrade there was 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-wch url 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

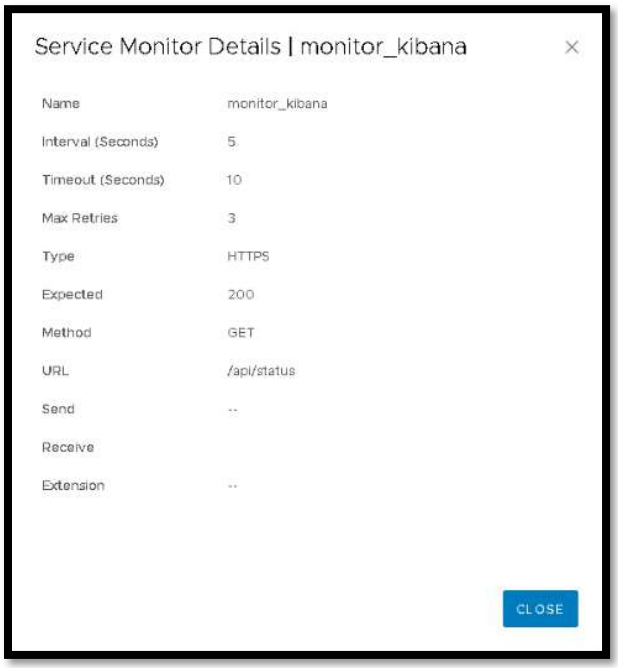


Figure 3. Updated Service Monitor configuration

5.4.1.6 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.

5.4.1.7 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
```

5.4.1.7.1 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 refreshit 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.

5.4.1.7.2 Install/Configure SAKM

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

5.4.1.7.2.1 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
```

5.4.1.7.2.2 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

5.4.1.7.2.3 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
```

5.4.2 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.

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 repository and accessible via the new satrepo alias:

https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/certs

Validate access and visibility of one of the new certs from one of the new nodes by running the following:

curl -k https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/certs/{x}0asu01el01.crt

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. Elasticsearch, Logstash, and Kibana RPMs copied Elastic repository and rebuilt.
2. Satellite admin ensured RPMs and installation artifacts are synchronized to all sites
3. Install Elasticsearch; ensure cluster is at 100% health.
4. Install all Kibana instances.
5. Load Roles.
6. Load Role Mappings.
7. Validate Active Directory Login
8. Load Audit Settings.
9. Add License
10. Adjust Concurrent Recovery settings.
11. Memory Lock Check.
12. Secure Elastic with Break-Glass Password

13. Verify Roles.
14. Remove Unneeded Accounts
15. Update Ingest Pipelines in Elasticsearch for previous version.
16. Continue to Upgrade Instructions

STOP - DO NOT PROCEED WITHOUT THE INSTALL DIRECTORY ON THE REPO UPDATED AND PUBLISHED BY SATELLITE

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

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 WCH Cluster

The following section is for installing a 2nd cluster on the production system at the West Coast Hub (WCH). This section only applies to the production system and should be skipped for all other upgrades.

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 Verify Service Account (From Each VM)

WARNING: Elasticsearch is modified to run as the **0a_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 section for more details.

1. Log in to each Elastic node and sudo to root.
2. su to the Elastic service account for WCH

```
# su 0a_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 ".."

5.4.3.2 Ensure Archive Directory exists

Before installing Elasticsearch the Archive directory on the Isilon share must be created. Follow the steps below to ensure that the directory is created.

1. Login to any Elasticsearch Node
   # sudo su

2. Run verify script

```
# curl -s -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/verifyArchiveDir.sh | bash
```

The script will print one of the two messages:

- elasticArchive directory already exists.

or

- Created elasticArchive directory.

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. If it will be, you must verify that there is an **ELK-local** directory already created for the data.
   a. Is this node 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 0a_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.

      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:

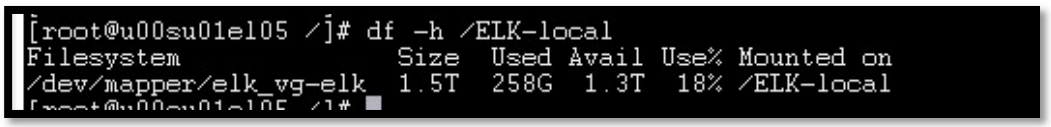


Figure 4. df command

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.

   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). The Elastic repo may also be contained in the redhat.repo file. Check with the Satellite administrator to get the correct name of the Elasticsearch repository.

5.4.3.5 Install Elasticsearch

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

```
# curl -s -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/installElasticNode.sh | bash
```

Repeat Steps 5.5.3.1 thru 5.5.3.4 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 cluster 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-wch 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 u0asu01el10 then it can be accessed directly at https://u0asu01el10:5601 or https://elastic-node-10:5601. 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.

Kibana will be installed on node 10 and node 15 in the WCH Cluster

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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/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-wch 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://u0asu01el10.wch:5601 or https://u0asu01el15.wch: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.

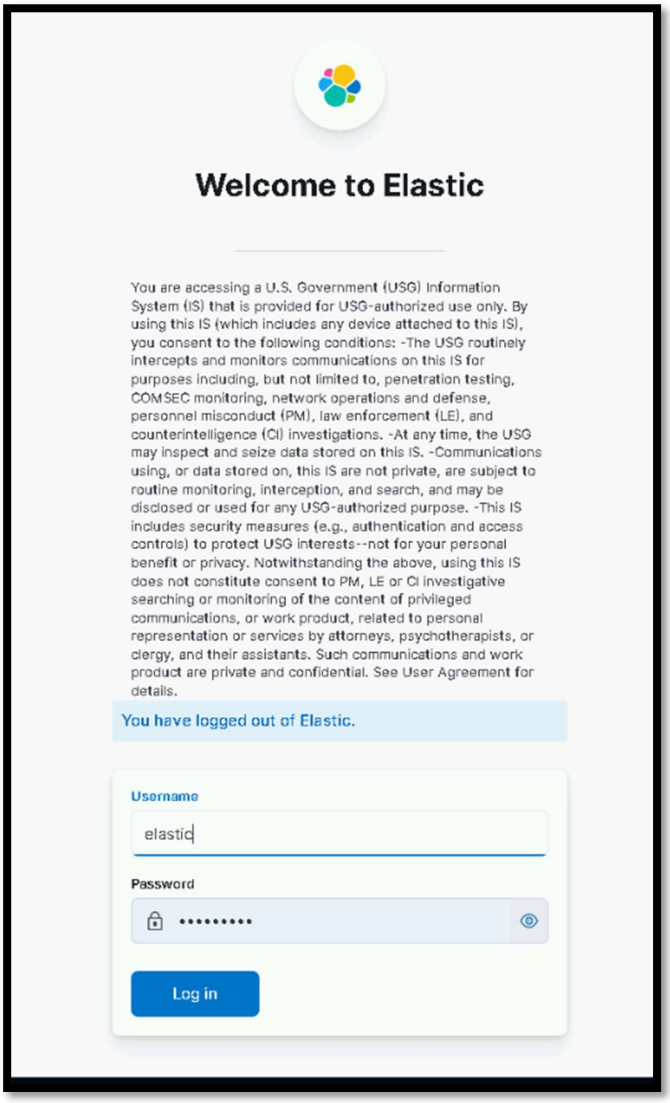


Figure 5. Login Screen

NOTE: Multiple configuration scripts executed later in this document communicate 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-wch in your browser, revisit the NSX configuration instructions before proceeding with the installation. If you cannot configure NSX, consult with an OADCGS Elastic SME for guidance.

5.4.4.3 Disable Usage Collection (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://kibana-wch
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**.

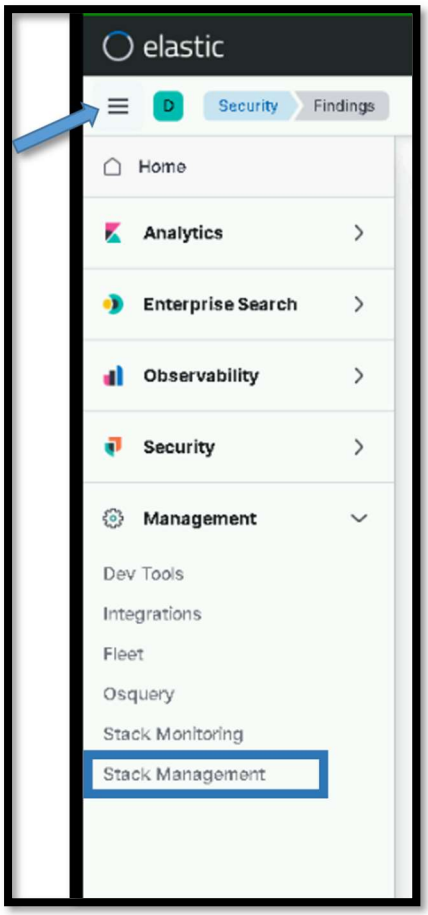
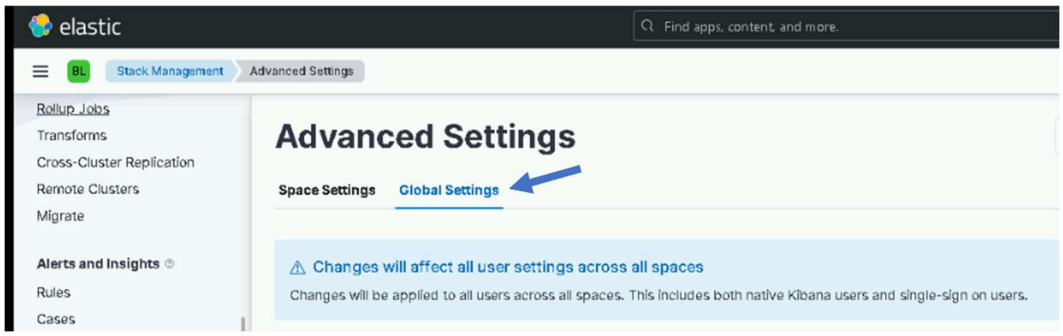


Figure 6. Stack Management

5. The **Stack Management** page displays. Under Kibana select **Advanced Settings** on the left side.
6. Select "Global Settings" at the top of the page



7. Enter **usage** in the search bar to filter.
8. If **usage collection** is **On**, click the slider to turn it **Off**.

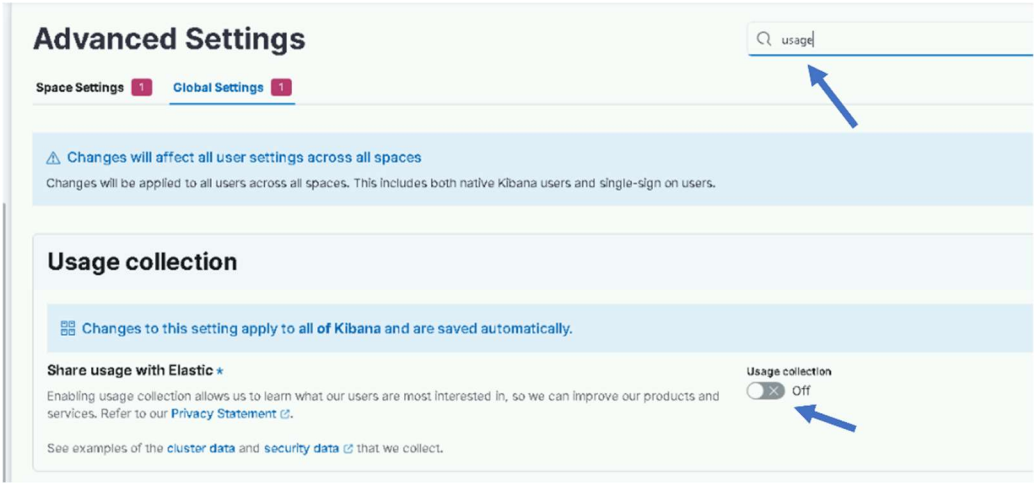


Figure 7. Disable Usage Collection

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 WCH Elastic nodes.

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/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 6 roles that should be loaded.

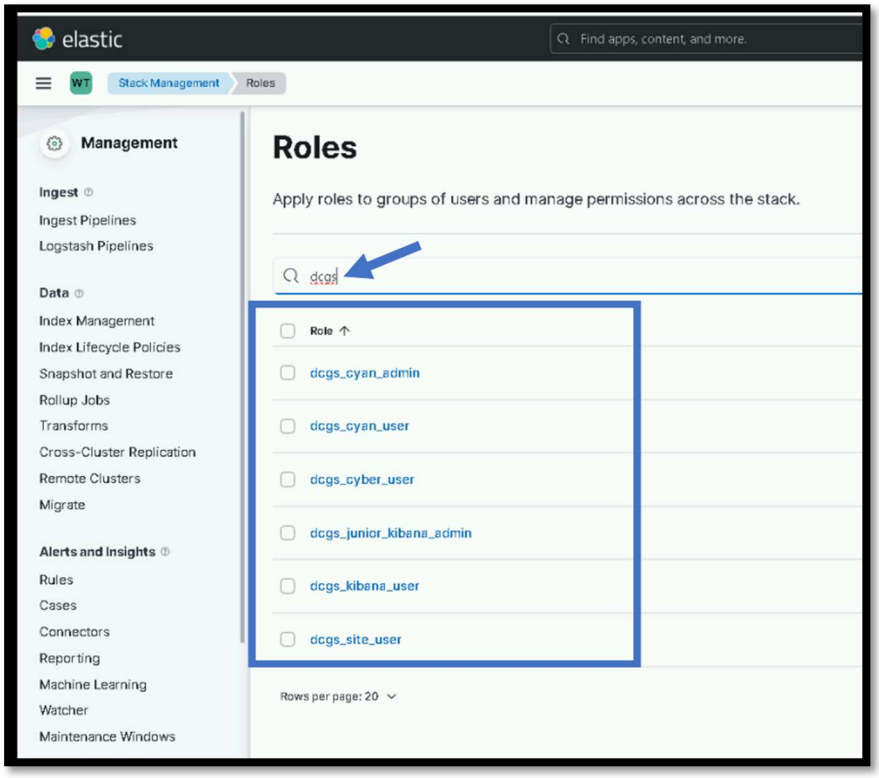


Figure 8. Roles

6. Erase **dcgs** and type **stale** in the search bar to see the stale-delete role that should be loaded

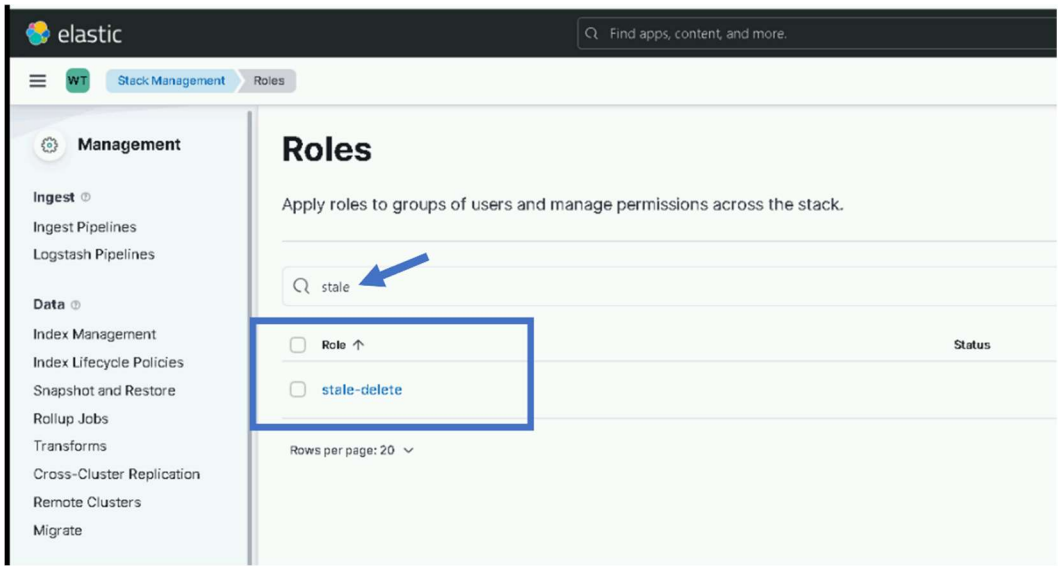


Figure 9 stale-delete role

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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/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 (https://kibana-wch) with the “elastic” user
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 **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.

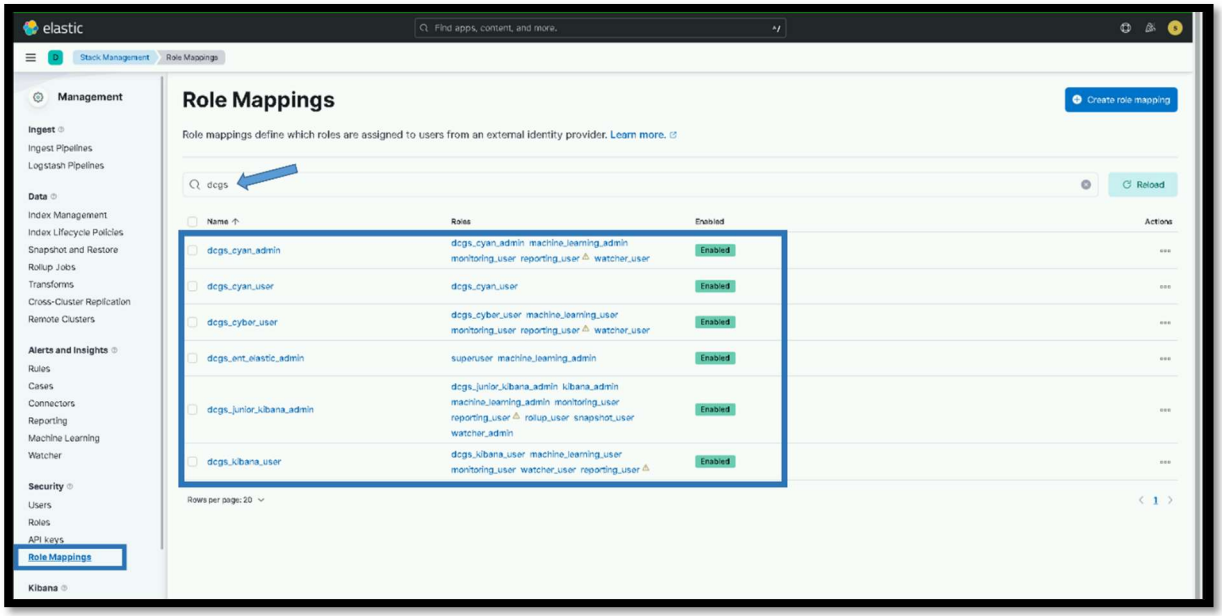


Figure 10. Role Mappings

5.4.5.2.3 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.

5.4.5.3 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-wch (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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/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**.

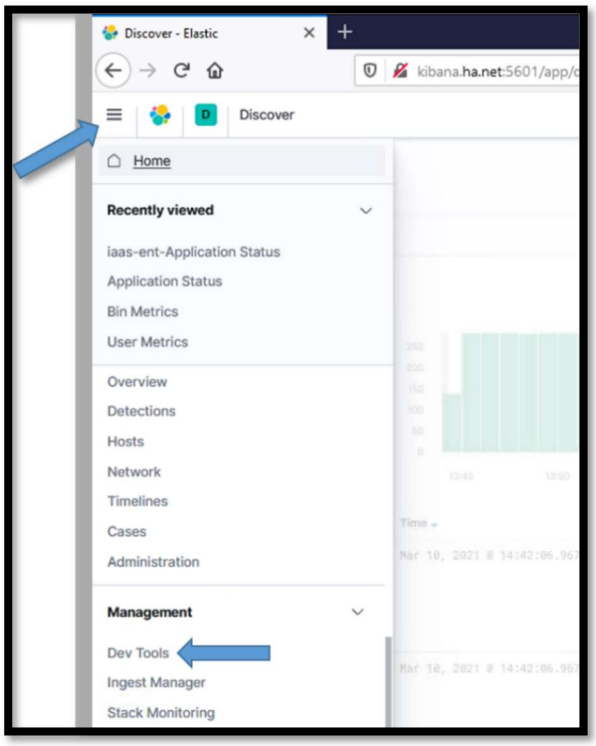


Figure 11. Dev Tools

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:

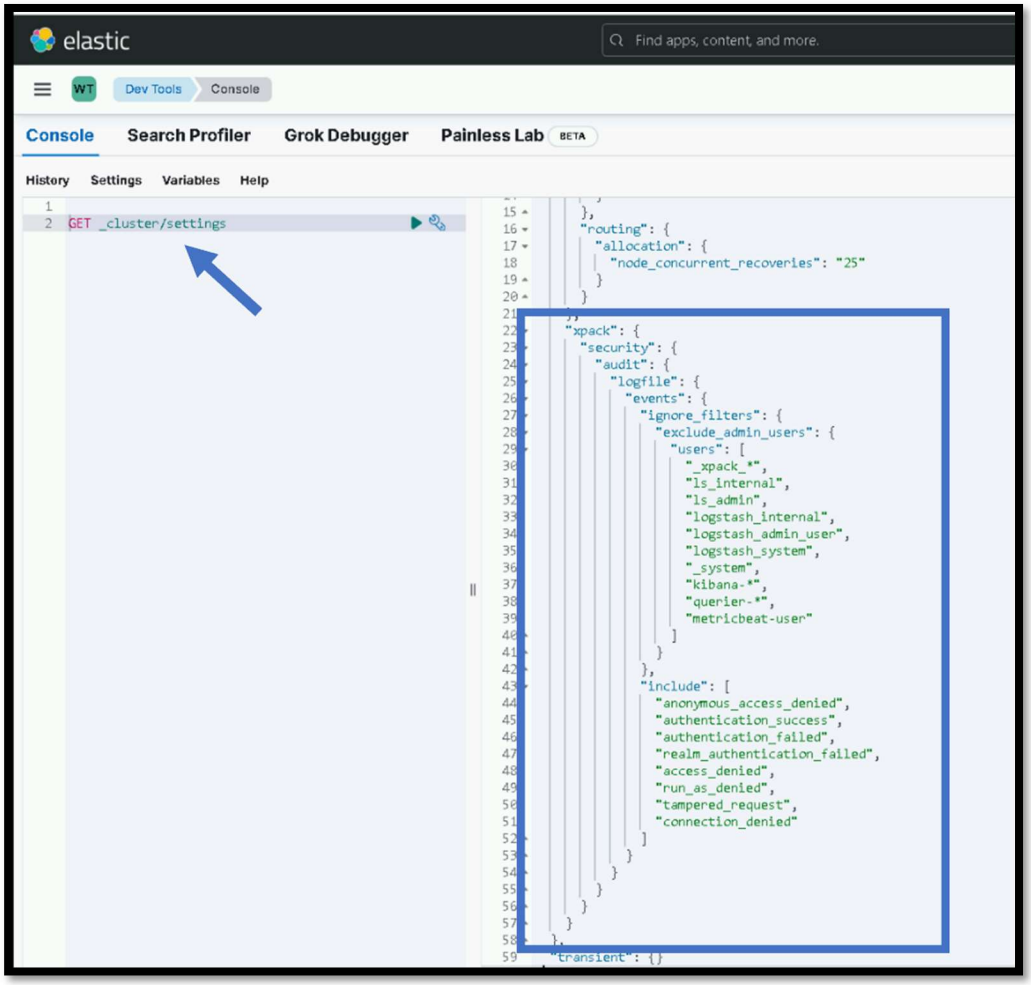


Figure 12. Expected audit settings

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 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
   }
}
```

5.4.5.7 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.

5.4.6 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

5.4.7 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: If they exist do not remove the following accounts, the **kibana_xx**, **logstash_admin_user**, or **logstash_internal** accounts.

5.4.8 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 13. 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.

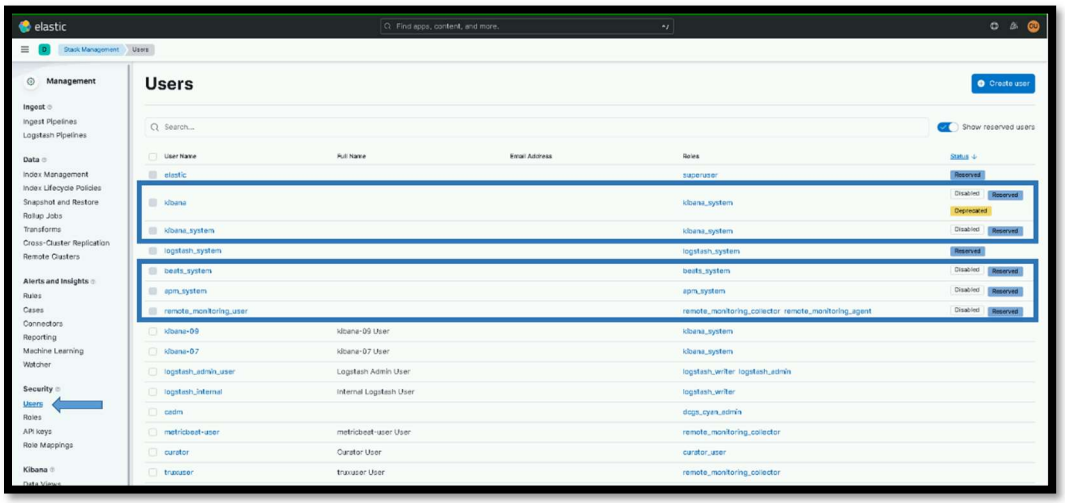


Figure 14. Disabled accounts

5.4.9 Update Ingest Pipelines in Elasticsearch for previous version

This step is to add ingest pipelines for the 8.6.2 versions into the WCH cluster. These may be needed if any data from that version of Filebeat or Winlogbeat comes into the cluster.

Run the following command from any of the WCH elastic nodes to add the 8.6.2 ingest pipelines to the WCH Cluster:

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/update_ingest_pipelines.sh | bash -s 8.6.2
```

5.4.10 Setup Metricbeat monitoring of cluster.

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

Ensure the Elasticsearch Cluster is “Green” before continuing.

1. Log in to elastic-node-1 of the new Elasticsearch Cluster on WCH and become root.

   ```
   # sudo su
   ```

2. Run the installMetricbeatKeystore.sh script.

   ```
   # curl -k
   https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
   ic_Files/install/installMetricbeatKeystore.sh | bash -s wch
   ```

3. **When the script is complete you will see an important notice, similar to one shown here:**

   The metricbeat-user did not exist in the new cluster so it is created and the metricbeat.keystore is updated with its password. Follow the directions and copy the new metricbeat.keystore to the elastic keystores directory on the Master repo server. Once the metricbeat-wch.keystore file exists on the Master repository ask the Satellite administrator to republish the Elastic Files repository so the file is available on the Capsule Server at WCH. If this is not completed running this script on the remaining nodes will not work correctly. Once complete, proceed onto the next step of the upgrade.

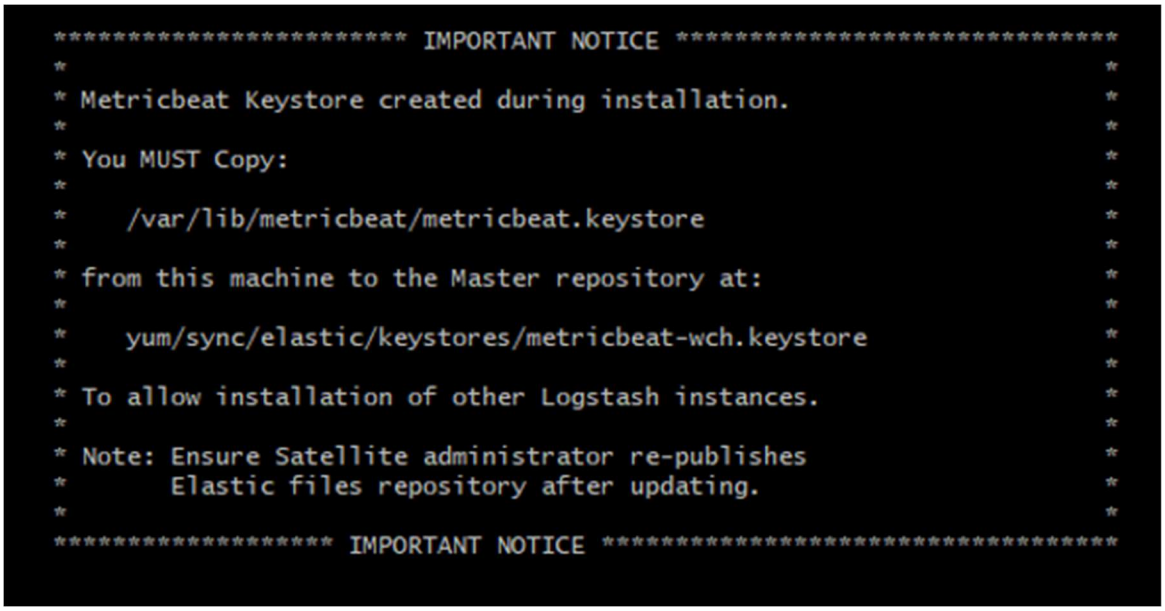


Figure 15- Metricbeat keystore copy notice

1. From the path above scp metricbeat.keystore <username>@x00su01ro01:/tmp(Or the Master Repository for the system)
2. login to the Master repo server and sudo to root
3. cp /tmp/metricbeat.keystore /var/www/html/yum/sync/elastic/keystores/metricbeat-wch.keystore
4. chown apache:apache /var/www/html/yum/elastic/install/metricbeat-wch.keystore

   IMPORTANT: Make sure when you copy the new keystore it is named "metricbeat-wch.keystore" on the repository.

5. Ask the Satellite administrator to re-publish the Elastic Files repository.

4. **DO NOT PROCEED UNTIL THE UPDATED FILE REPOSITORY HAS BEEN PUBLISHED BY THE SATELLITE ADMINISTRATOR.**
5. Run the script on the reset of the elastic nodes in the new WCH Cluster (node2 – node15)
6. Login to the node sudo to root and execute the same script as above:

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/installMetricbeatKeystore.sh | bash -s wch
```

7. This time the script will detect that the metricbeat-user has already been created in the WCH cluster and will look for the metricbeat-wch.keystore file on the Satellite Capsule Server in the Elastic Files repository. You should see the following message if the keystore is updated successfully:

   Metricbeat keystore installation complete…

8. Repeat steps 6 & 7 on all remaining nodes of WCH cluster.

5.4.11 Continue to Upgrade Instruction

At this point the new WCH cluster has been installed and should be up and running. Now continue onto the upgrade of the existing cluster. Note that during the upgrade instructions some scripts will now be run on both the ECH and WCH clusters.

5.5 Installation Instructions for Upgrades

The instructions in this section are for use when upgrading the DCGS Enterprise Service Elasticsearch from version 8.6.2 to version 8.11.3.

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. Prepare new Frozen Node
2. Upgrade Elasticsearch Cluster (All Nodes)
3. Upgrade Kibana Instances
4. Update Roles
5. Update Kibana Settings
6. Update Filebeat Ingest Pipelines
7. Update Templates
8. Bootstrap Indexes
9. Configure Index Lifecycle Policies
10. Prevent Disk Usage Alerts for Frozen nodes
11. Create Site Spaces

12. Create Baseline Space
13. Update Kibana Settings
14. Load Kibana Saved Objects into baseline space (Update dashboards/visuals)
15. Upgrade Logstash Instances (All Sites)
16. Update Enterprise Services Centralized Logstash Pipelines
17. Add Unicorn Pipeline
18. Cleanup Current Health Data Index
19. Setup Cross Cluster Replication (Production Only)
20. Upgrade Data Collector (All Sites)
21. Install Watchers
22. Upgrade Beats (Linux, Windows and Domain Controllers)
23. Update puppet modules (dsil_elastic_clients and dsil_elastic_servers)
24. Remove "Run and Remove" scripts from system when upgrade is completed.

5.5.1 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: Before proceeding ensure that the new install directory has been copied to the Elastic repo. The directory should be in the **/var/www/html/yum/sync/elastic** repo directory and should be named **install**. The contents of the directory should match the description in Section 4.3. The Satellite administrator should have published the updates.

IMPORTANT: Also ensure the repo server has been updated with the new Core RPMs and the Satellite administrator has published the updates.

5.5.1.1 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).

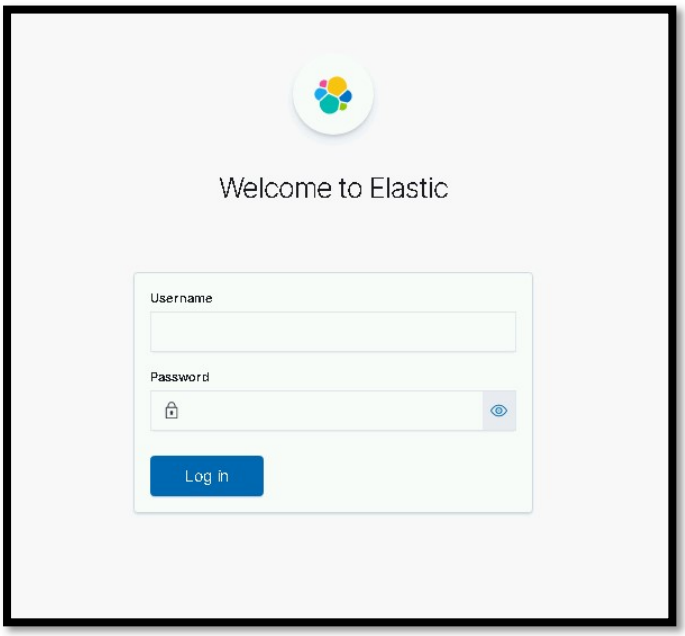


Figure 16. Login Screen

3. After successful login, you will be asked to select a workspace. Select **Default**.

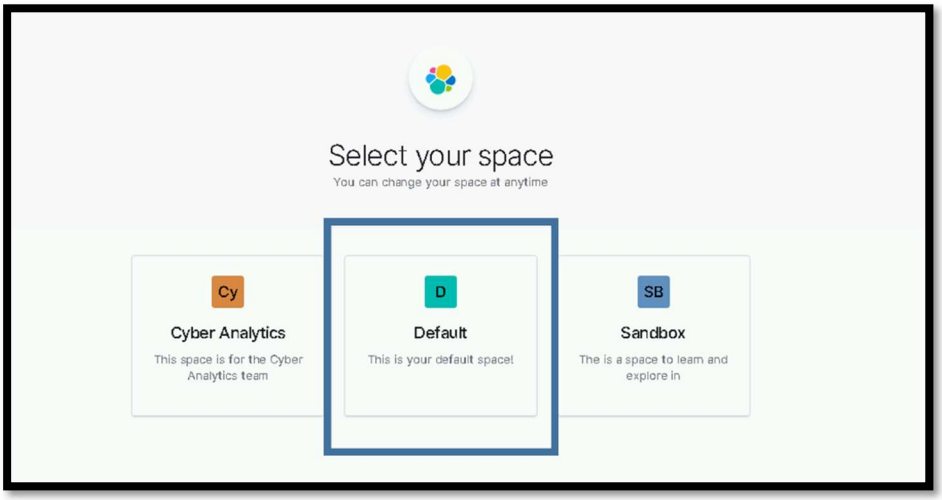


Figure 17. Select Default Workspace

4. Using the Kibana hamburger menu, select **Stack Monitoring**.

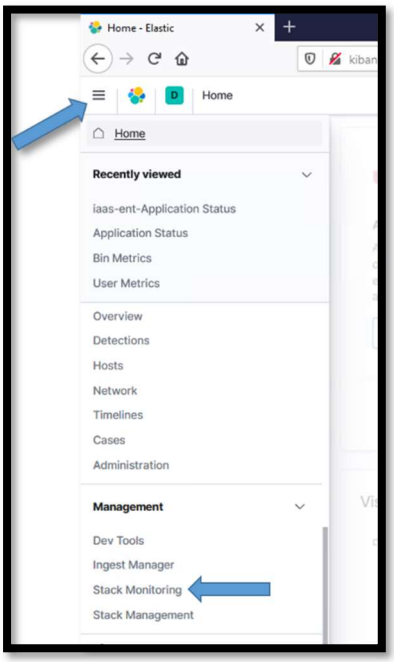


Figure 18. Navigate to Stack Monitoring

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.

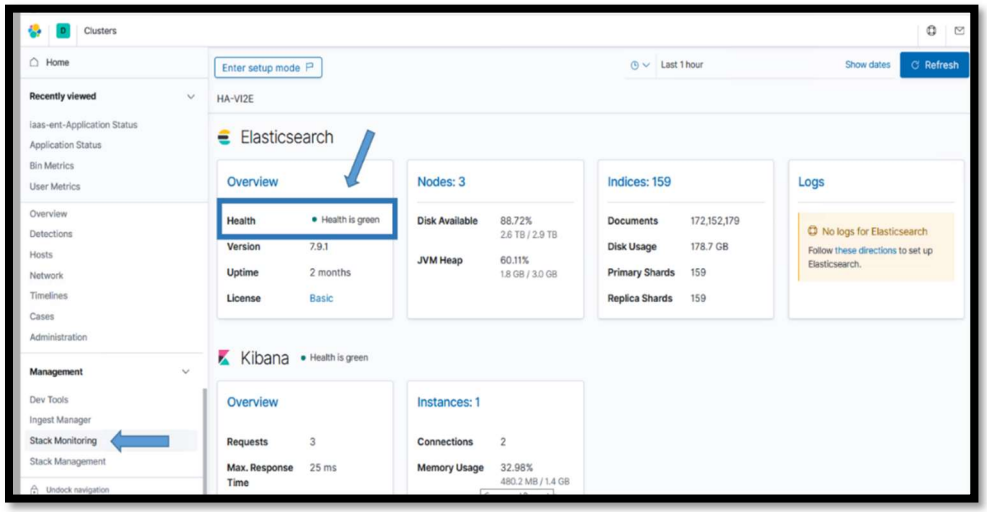


Figure 19. Health Status Should Be Green

5.5.1.2 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.

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 7. Upgrade Order

Order	7 Node Cluster	10 Node Cluster	15 Node Cluster
1	elastic-node-4	elastic-node-4	elastic-node-4
2	elastic-node-5	elastic-node-5	elastic-node-5
3	elastic-node-6	elastic-node-6	elastic-node-6
4	elastic-node-7	elastic-node-7	elastic-node-7
5	elastic-node-1	elastic-node-8	elastic-node-8
6	elastic-node-2	elastic-node-9	elastic-node-9

Order	7 Node Cluster	10 Node Cluster	15 Node Cluster
7	elastic-node-3	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

5.5.1.2.1 Prepare new Frozen Node

5.5.1.2.1.1 MTE, CTE and Production Frozen Node

IF you are upgrading a 10 or 15 node cluster you must first move all the data off of the node that will become the Frozen tier. **This step may take hours to complete so execute it as soon as possible.** The cluster should remain up until all data is transitioned off the node.

1. Determine the IP address of the node that is becoming the Frozen node
    a. CTE/MTE – elastic-node-10
    b. Production High/Low – elastic-node-15
2. Ensure the Infrastructure team has added the new 100GB SSD 2nd drive to this node
    a. Validate the the 2nd drive is now mounted on /ELK-local

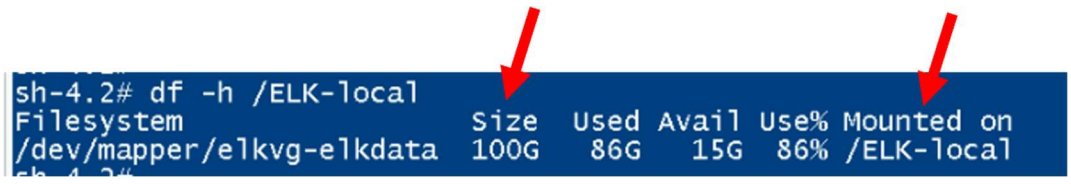


Figure 20- Validate new 100GB 2nd disk

3. Move all current shards off future frozen node

Run this command from the console of kibana:

```
PUT _cluster/settings
{
  “persistent” :
  {
    “cluster.routing.allocation.exclude._ip” : “X.X.X.X”
  }
}
```

Example:

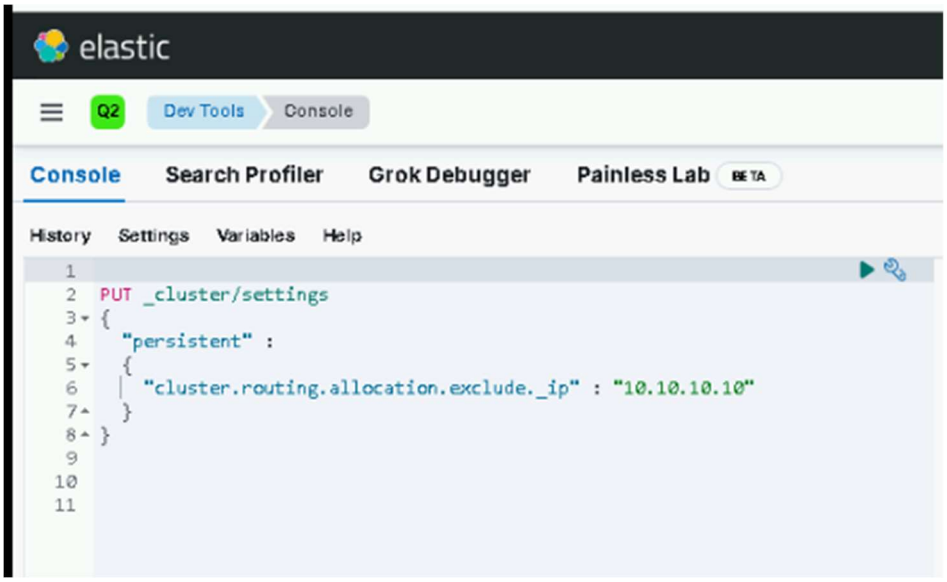


Figure 21- exclude._ip example

4. Monitor cluster for data migration off the node. You can use the “Nodes” view on Stack Monitoring to do this or run the following command:

   GET _cat/shards?v&s=node:desc

5. **DO NOT PROCEED WITH THE UPGRADE UNTIL ALL DATA IS OFF OF THE FUTURE FROZEN NODE**

5.5.1.2.1.2 REL Frozen Node

Before proceeding with the upgrade on REL you must verify the infrastructure team has added a new node for the cluster (elastic-node-7) and add it to the cluster.

Ensure the following on the new node:

- Node has 100GB SSD 2nd Drive
- 2nd Drive is mounted as /ELK-local
- Node is added to puppet and iptables shows ports 9200 and 9300 available.
- SAKM is installed and running on node managing the elastic service account.
- Validate /ELK-nfs is mounted and accessible with service account.
- If all looks good let's install Elasticsearch on this node and add it to the cluster

Run the following:

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

```
# curl -s -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/installElasticNode.sh | bash
```

```
Select 1 for 7 Node cluster.
Select 1 for data locally stored in /ELK-local.
```

Start new node:

Run the following command on the new node 7:

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

Wait at least 5 minutes to give the node a chance to start, then validate it's running.

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

Validate new node is part of cluster. Run the following command to list the nodes in order by name:

```
# GET _cat/nodes?v&s=name
```

NOTE: If node 7 is not in the list STOP and contact and elastic SME for guidance.

5.5.1.2.2 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 and has been published by Satellite. 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). **NOTE**: The repo name may be redhat.repo after the move to Satellite.

Ensure the **Elasticsearch.x86_64** package is the version you are upgrading to.

After ensuring the node can read the upgrade package, execute the following script to perform the upgrade on the node:

```
# curl -s -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/upgrade_node.sh | bash
```

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. (**NOTE**: If you are upgrading REL node 7 does not need to be upgraded as that node was a fresh install and is already running the upgrade version)

5.5.1.2.3 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**.

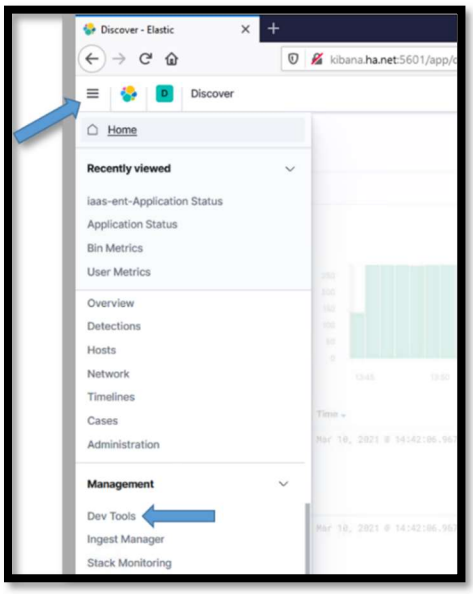


Figure 22. Select Dev Tools

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:

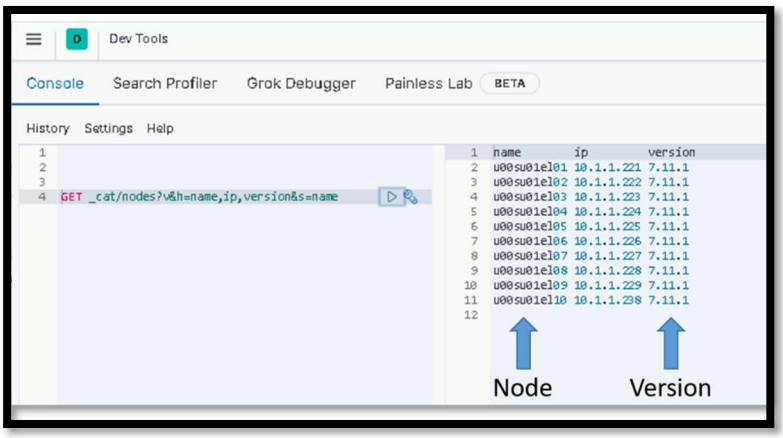


Figure 23. Check node versions example

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.

5.5.1.2.4 Complete Cluster Upgrade

During the execution of the upgrade_node.sh in section 5.5.1.2.1, 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**.

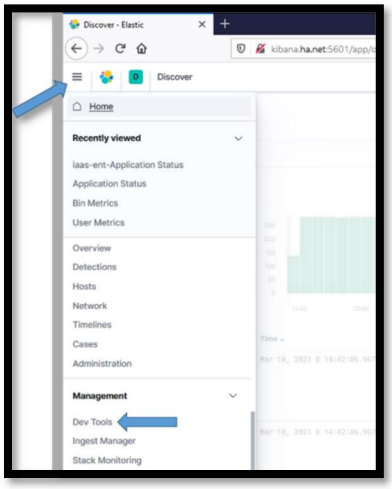


Figure 24. Select Dev Tools

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.

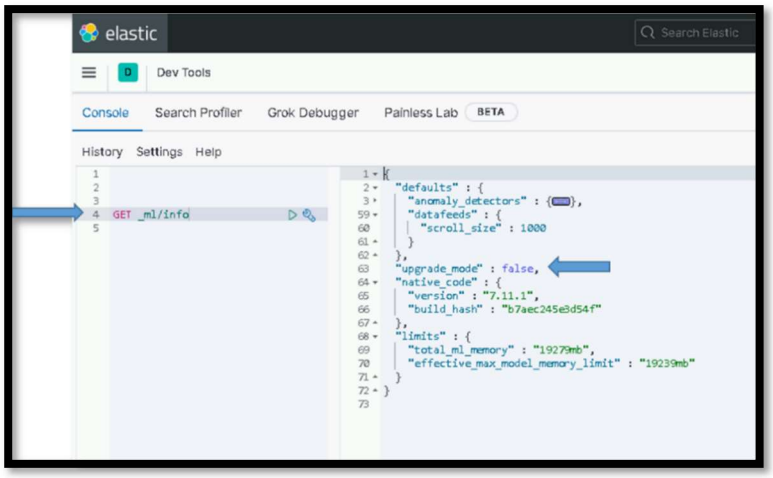


Figure 25. Verify ml upgrade_mode is false

5.5.1.2.5 Remove IP Exclude on Frozen Node (MTE/CTE and Production Only)

Prior to upgrading you moved all the data off on the new Frozen node. This node has now been configured to only hold Frozen data and needs to be removed from the "exclude._ips" list.

Run the following command to do this:

```
PUT _cluster/settings
{
  "persistent" :
  {
    "cluster.routing.allocation.exclude._ip" : null
  }
}
```

5.5.1.3 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 8. Elastic nodes to upgrade Kibana on

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

5.5.1.3.1 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.**

5.5.1.3.2 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.

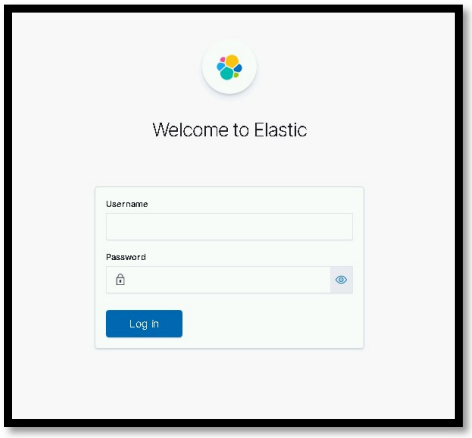


Figure 26. Login Screen example

7. Select the **Default Space** and then select **Discover** from the hamburger menu as described in section 5.5.1.2.3
8. Select any index pattern to verify you can view data.

5.5.2 Update Elastic Search Configurations

NOTE:

An Elasticsearch administrator will be needed to execute this section.

5.5.2.1 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.

5.5.2.1.1 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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/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.

5.5.2.1.2 Verify Kibana Roles are Loaded

To verify the Kibana roles were successfully loaded:

7. Click the menu (three horizontal lines) button at the top left of the screen. The navigation menu displays.
8. Scroll to **Management** at the bottom and select **Stack Management**.
9. The **Stack Management** page displays. Under **Security** select **Roles** on the left side.
10. Enter **dcgs** in the search bar to see the 6 roles that should be loaded.

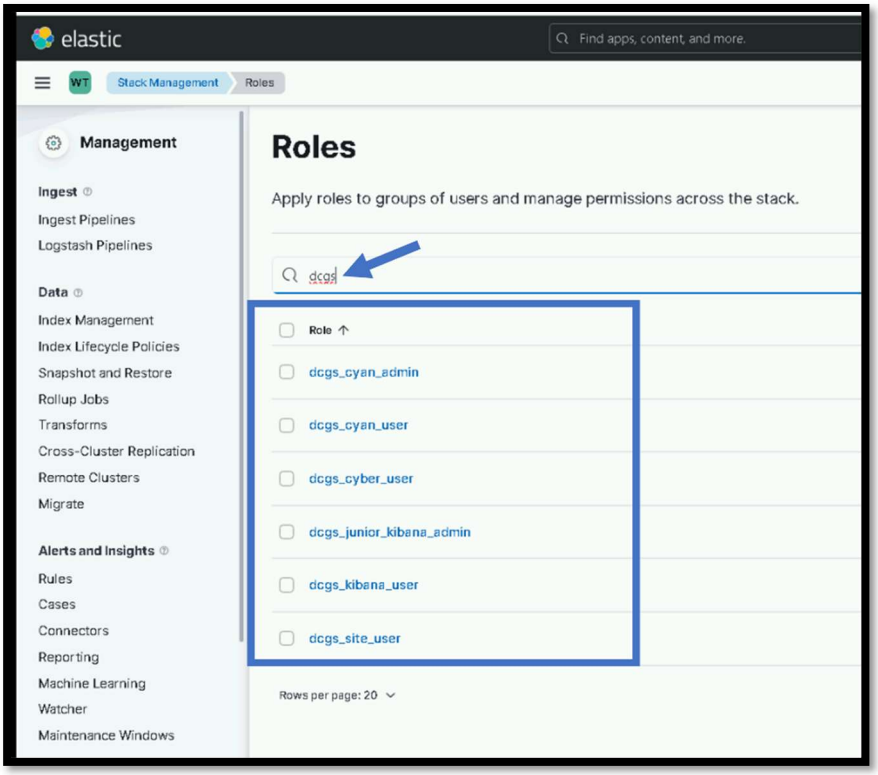


Figure 27. Roles

5.5.2.2 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:

IMPORTANT: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps from on both Clusters https://kibana and https://kibana-wch

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/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.

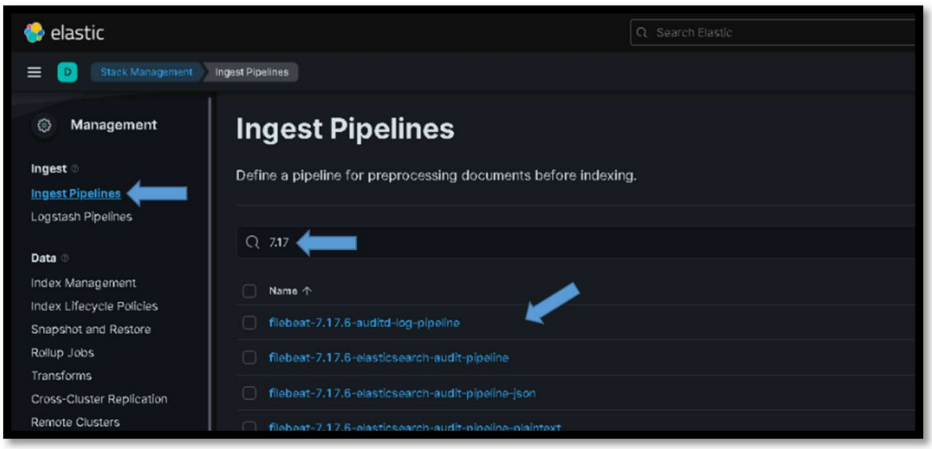


Figure 28. Example of Ingest Pipelines for version 7.17

For version 8.11.3 you should see the following pipelines:

- filebeat-8.11.3-auditd-log-pipeline
- filebeat-8.11.3-elasticsearch-audit-pipeline
- filebeat-8.11.3-elasticsearch-deprecation-pipeline
- filebeat-8.11.3-elasticsearch-gc-pipeline
- filebeat-8.11.3-elasticsearch-server-pipeline
- filebeat-8.11.3-elasticsearch-slowlog-pipeline
- filebeat-8.11.3-iptables-log-pipeline
- filebeat-8.11.3-logstash-log-pipeline
- filebeat-8.11.3-logstash-slowlog-pipeline
- filebeat-8.11.3-system-auth-pipeline
- filebeat-8.11.3-system-syslog-pipeline
- filebeat-8.11.3-elasticsearch-audit-pipeline-json
- filebeat-8.11.3-elasticsearch-deprecation-pipeline-json
- filebeat-8.11.3-elasticsearch-server-pipeline-json
- filebeat-8.11.3-elasticsearch-slowlog-pipeline-json
- filebeat-8.11.3-logstash-log-pipeline-json
- filebeat-8.11.3-logstash-slowlog-pipeline-json
- filebeat-8.11.3-elasticsearch-audit-pipeline-plaintext
- filebeat-8.11.3-elasticsearch-deprecation-pipeline-plaintext
- filebeat-8.11.3-elasticsearch-server-pipeline-plaintext
- filebeat-8.11.3-elasticsearch-slowlog-pipeline-plaintext
- filebeat-8.11.3-logstash-log-pipeline-plaintext
- filebeat-8.11.3-logstash-slowlog-pipeline-plaintext

- filebeat-8.11.3-kibana-audit-pipeline
- filebeat-8.11.3-kibana-audit-pipeline-json
- filebeat-8.11.3-kibana-log-pipeline
- filebeat-8.11.3-kibana-log-pipeline-7
- filebeat-8.11.3-kibana-log-pipeline-ecs
- winlogbeat-8.11.3-sysmon
- winlogbeat-8.11.3-security
- winlogbeat-8.11.3-routing
- winlogbeat-8.11.3-powershell
- winlogbeat-8.11.3-powershell_operational

5.5.2.3 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: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps from on both Clusters https://kibana and https://kibana-wch

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

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

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/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_fc6xx
   - esti_fx2
   - esti_isilon
   - esti_nexus5k

- esti_nexus7k
- esti_r6xx
- esti_xtremio
- esti_healthdata
- esti_current-healthdata
- esti_idm
- esti_sccmdb
- esti_hbss-epo
- esti_hbss-metrics
- esti_hbss-dlp
- esti_sqlserver
- esti_eracent
- esti_puppet
- esti_vsphere
- esti_filebeat-"${esver}"
- esti_db_postgres
- esti_heartbeat-"${esver}"
- esti_winlogbeat-"${esver}"
- esti_iptables
- esti_serena
- esti_render
- esti_soaesb
- esti_acas
- esti_socetgxp
- esti_gxpxplorer
- esti_maas_logs
- esti_loginsight_syslog
- esti_logindata
- esti_ecp
- esti_ashes
- esti_unparsable-syslog

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.

5.5.2.4 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: Currently 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.

IMPORTANT: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps from on both Clusters https://kibana and https://kibana-wch

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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
   ic_Files/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 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
   ```

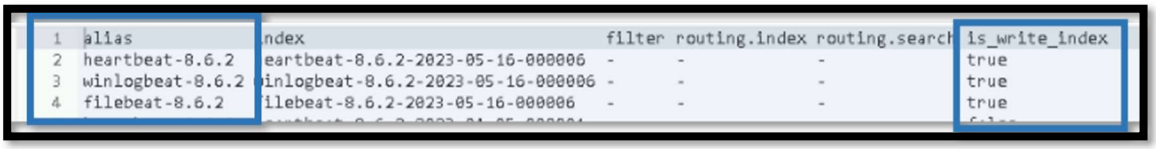


Figure 29. GET _cat/aliases/*beat-{version}>*?v&s=alias output

3. 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.

5.5.2.5 Configure Index Lifecycle Management (ILM) [Needs updated]

The Index Lifecycle Management (ILM) Policies are used to automatically manage the indices in Elasticsearch. ILM Policies in Elasticsearch ensures Elastic does not become overwhelmed with data over time. In the current implementation each index has one of the following **lifecycles:**

- Non-Security related indexes (Metrics, non-security related logs, etc):
  Hot → Warm → Cold → Deleted
  ILM Policy: dcgs_default_policy

- Security Related indexes:

  Hot → Warm → Cold → Frozen → Deleted

  ILM Policies:

  - dcgs_audits_syslog_policy
  - dcgs_db_policy
  - dcgs_hbss_epo_policy
  - dcgs_syslog_policy
  - dcgs_vsphere_policy
  - winlogbeat_policy

  NOTE: In these lifecycles the index is deleted after 5 years

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.

5.5.2.5.1 Load DCGS Default ILM Policy

The DCGS ILM policies control the lifetime of all DCGS indexes. These polices must be loaded to ensure indices are managed so the amount of data in the cluster does not grow too large over time.

IMPORTANT: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps from on both Clusters https://kibana and https://kibana-wch

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

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/load_ILM_Policies.sh | bash
```

2. Upon successful loading, the policies will appear in the list of **Index Lifecycle Policies** on the Kibana **Stack Management** page (under Data). Type "dcgs" in the search bar to filter and you should see the following policies:

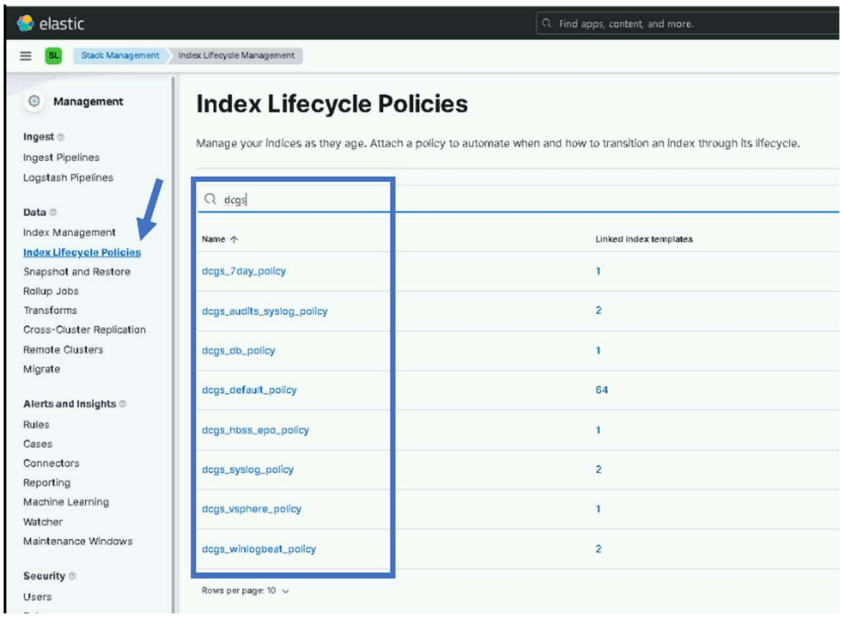


Figure 30 Index Lifecycle Policies

5.5.2.5.2 Update Lifecycle Policies

This step is to now update any existing security indexes to use the newly loaded ILM polices in the previous step.

Login to any node on the existing Elasticsearch Cluster and run the following:

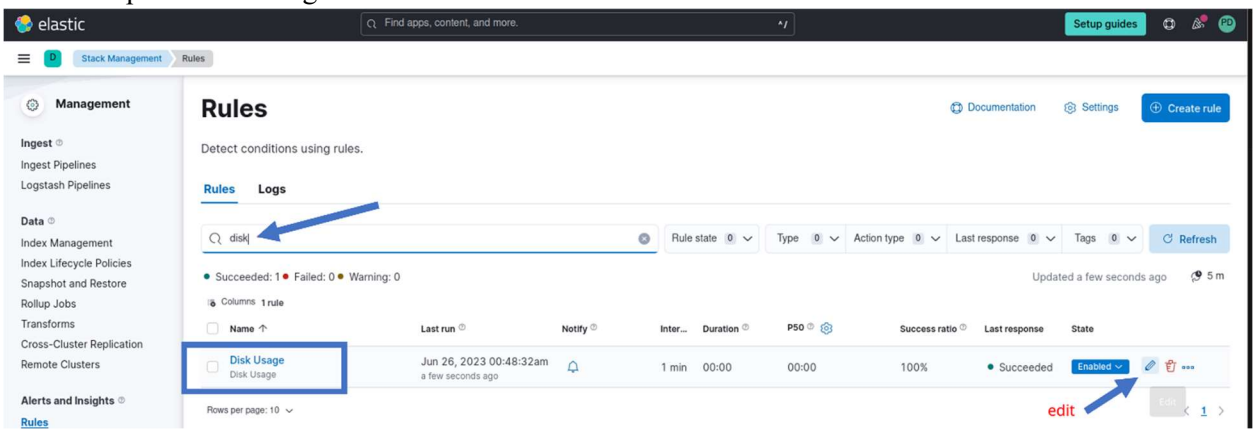
NOTE: This script does not have to be run on the new cluster on the production systems.

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/update_lifecycle.sh | bash
```

5.5.2.6 Prevent Disk Usage Alerts for Frozen Tier

On the new Frozen node we are using the default value for *xpack.searchable.snapshot.shared_cache.size* which is 90% of the disk. This conflicts with the default disk usage threshold for Elasticsearch nodes in Kibana Stack Monitoring of 80%. This will lead to false alerts because the frozen cache is allocated upfront (frozen nodes are expected to have 90% disk utilization with default settings). To resolve this issue an exclusion will be added to the Disk Usage Alerting rule for the Frozen node. Follow the following steps to add this exclusion.

1. Open Kibana and navigate to **Stack Management** -> Alerts and Insights -> **Rules**
2. Type “Disk” into the search bar
3. Select the pencil on the right to edit the alert



4. Add filter to exclude frozen tier from alerting:

   NOT elasticsearch.node.roles : “data_frozen”

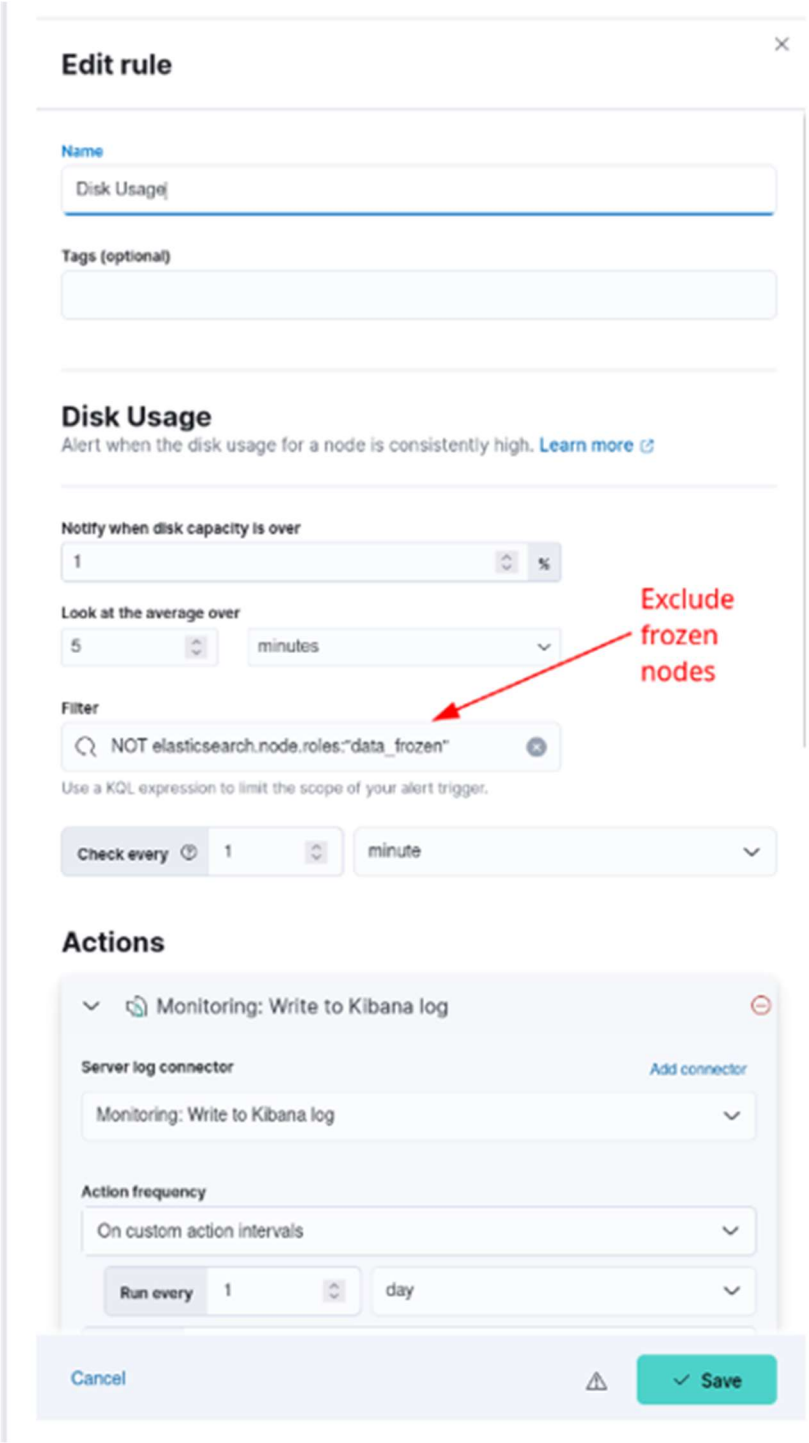


Figure 31- Exclude frozen nodes from Disk Usage rule.

5. Select “Save” to save the updated rule

5.5.3 Update Kibana

5.5.3.1 Create Spaces

This section is to create new spaces in Kibana.

5.5.3.1.1 Create Site Spaces

In this version "Site Spaces" will be created in all environments. Site spaces are being created to allow each site to have more control over what they would like to use in their environment.

IMPORTANT: To have edit privileges in a particular site space a user at the site would need request membership to the Kibana administrator group for that site ("xx Kibana Administrator").

IMPORTANT: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps from on both Clusters https://kibana and https://kibana-wch

Login to any of the elastic nodes and run the following command to create the new spaces:

```
# curl -k https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/create_spaces.sh | bash
```

After running this script validate that the following spaces now exist in Kibana:

- ALL Enclaves - cyberops
- MTE - 24
- CTE - 24,70,r0
- REL - d1,d2
- Production Low - 00,0a,01,02,03,04,05,13,14,15,16,17,19,50,69
- Production High - 00,0a,01,02,03,04,05,16,17,18,19,21,22,23,45,49,69

5.5.3.1.2 Create Baseline Space

This version brings the new "baseline" space which will replace the "default" space as the location for all delivered Kibana artifacts such as visuals, dashboards, data views, saved searches, etc. The script will move all existing items in the current "default" space to a new "default-deprecated" space. Ensure that you retrieve anything needed from this space as it will be removed in future versions.

IMPORTANT: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps from on both Clusters https://kibana and https://kibana-wch

Login to any of the elastic nodes and run the following command to create the new baseline space:

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/update_default_space.sh | bash
```

After running this script validate that the new baseline space exists in Kibana. If the space has not been made **STOP** and contact an Elastic SME for guidance. This space will be needed in later step of this upgrade.

5.5.3.2 Update Kibana Settings

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

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/update_kibana_settings.sh | bash
```

This will set the 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 32. Example showing security banner and dark mode

5.5.3.3 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.

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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/load_objects.sh | bash
```

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

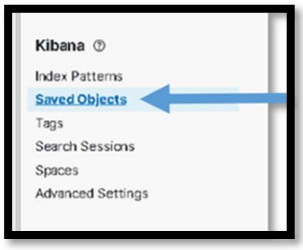
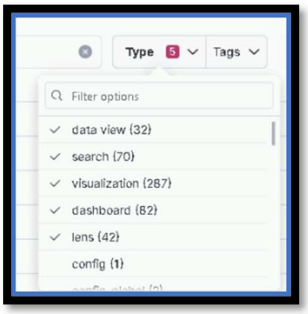


Figure 33. Select Saved Objects

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 (32)
    - search (70)
    - visualization (287)
    - dashboard (82)
    - lens (42)

    Example:



The following dashboards will be loaded into the baseline space:

Dashboard Name	Description	Drilldowns
IAAS-ES-KIBANA-User-Login-Dashboard	This dashboard tracks the login information and status by users in Kibana	
IAAS-ES-SYSTEM-Dashboard	Displays the health status and symptoms of devices across the system; if a host is displayed on this dashboard, it has an issue	Data Collector Status: Links to the IAAS-ES-Data-Collector-Details Dashboard Syslog Details: Links to IAAS-ES-SYSLOG-Dashboard Windows Event Details: Links to IAAS-ES-WINDOWS-Events-Dashboard Infrastructure Dashboard: Links to IAAS-ES-Infrastructure Overall Status Dashboard Puppet Dashboard: Links to IAAS-ES-Puppet-Dashboard
IAAS-ES-NETWORK-Detailed Status Dashboard	Displays the health and status of the different switches in the network; displays health, status, fan status, input/output utilization, CPU/memory usage, and temperatures; user can input specified switches and/or DCGS sites to filter results	

IAAS-ES-Puppet-Dashboard	Displays the status of Puppet; shows if it is disabled on a host and the reason why; shows Failures and activity feeds	
IAAS-ES-VSPHERE-Details by Blade Dashboard	Displays the metrics from the Blade Servers; displays the CPU and Memory Utilization by Blade in Production and Management VMs; user can input specified data between U00 and U0A with filter	
IAAS-ES-WINDOWS-Events-Dashboard	Displays event information from Windows logs; filters for ERROR and WARNING log level information; displays breakdown of log levels, providers of events, number of events, events with corresponding IDs, log levels by host, and detailed breakdown of the log level	
IAAS-ES-FX2-Detailed Status Dashboard	Displays the health and metric status of the FX2 switches; the overall status, temperatures, and power metrics are displayed	Link to IAAS-ES-Infrastructure Overall Status Dashboard FC6XX Detailed View: links to IAAS-ES-FC6XX-Detailed Status Dashboard
IAAS-ES-SYSLOG-Dashboard	Displays SYSLOG information from monitored hosts; displays log levels, number of logs by log level, the type of SYSLOG facility, critical messages, the most used commands, log level severity over time, and detailed messages of the SYSLOG	
IAAS-ES-HEARTBEAT-Website/TCP Port Monitor Statistics Dashboard	Displays the metrics given by Heartbeat; displays certification expiration, TCP port status, top access times by host for Windows, and top TCP connection times by host	
IAAS-ES-METRICBEAT-Health & Status	Displays the metrics given by Metricbeat; displays the top CPU usage users, filesystem usage, top CPU usage by process, and CPU usage by Cores allocated to hosts; user can filter data between sites	
IAAS-ES-FC6XX-Detailed Status Dashboard	Displays the metric data of the FC6XX hardware; the overall status, traffic, disk, virtual disk, cooling devices, interface statistics, and system information are displayed	Link to IAAS-ES-Infrastructure Overall Status Dashboard Link to IAAS-ES-FX2-Detailed Status Dashboard
IAAS-ES-HBSS-Host-Intrusion-Dashboard	Displays security information for Windows hosts; displays event history, events by hosts, network intrusion information, top threats, and blocked intrusion events	
IAAS-ES-ACAS-Dashboard	Displays vulnerabilities monitored by ACAS; displays the total vulnerabilities, current systems that are vulnerable, scanner status, and vulnerabilities scoring information	

IAAS-ES-R6XX-Detailed Status Dashboard	Displays the metric data of the R6XX hardware; the comb status, network traffic, interface status, disk status, and system information are displayed	Link to IAAS-ES-Infrastructure Overall Status Dashboard
IAAS-ES-XTREMIO-Detailed Status Dashboard	Displays the metric data of the XTREMIO hardware; displays the cluster health, general information, free space, volume usage, latency over time, and event information	Link to IAAS-ES-Infrastructure Overall Status Dashboard
IAAS-ES-ISLION-Detailed Status Dashboard	Displays the metric data of the ISILON hardware; displays the hardware information, general overview, capacity used, node information, temperatures, SMB sharing information, client information, and exporting information	Link to IAAS-ES-Infrastructure Overall Status Dashboard
IAAS-ES-SWITCH-Detailed Status Dashboard	Displays the health and metric data of the switches; displays the overall status, general information, memory usage, up/down status of switches, fan statistics, and port information	Link to IAAS-ES-Infrastructure Overall Status Dashboard
IAAS-ES-DATADOMAIN-Detailed Status Dashboard	Displays the metric data of the data domains; displays overview, sensor temperatures, filesystem usage, and disk information	Link to IAAS-ES-Infrastructure Overall Status Dashboard
IAAS-ES-DEVICE-Simple Infrastructure Status Dashboard	Displays the overall status of all of the devices on the sites	Data Domain Details: Links to IAAS-ES-DATADOMAIN-Detailed Status Dashboard FX2 Details: Links to IAAS-ES-FX2-Detailed Status Dashboard Isilon Details: Links to IAAS-ES-ISLION-Detailed Status Dashboard Switch Details: Links to IAAS-ES-SWITCH-Detailed Status Dashboard Xtremio Details: Links to IAAS-ES-XTREMIO-Detailed Dashboard r630 Details: Links to IAAS-ES-R6XX-Detailed Status Dashboard
IAAS-ES-BEATS-Version	Displays the version of the beats that are installed on the hosts	
IAAS-ES-VSPHERE-CPU-Memory Gauges Dashboard	Displays the CPU and Memory usage by hosts in Production and Management VMs; can be filtered by sites	
IAAS-ES-LOGINSIGHT-Data Dashboard	Displays the LOGINSIGHT data; displays VBAR events by process for severity and hosts, VBAR process events and hosts, and the LOGINSIGHT activity feeds	

IAAS-ES-LINUX-User Activity Dashboard	Displays Linux information for the system; displays logons/logoffs by host with corresponding timestamps and process usage and information	
IAAS-ES-SYSTEM-Application-Info	Displays the health status of the different applications; displays by application and by host, with a host information table	
IAAS-ES-Overview Dashboard	Displays the overviews of the system information for Windows and Linux; displays event information and activity feed information by host; can filter by sites	
IAAS-ES-WINLOGBEAT-Dashboard	Displays Winlogbeat information; displays event information, IDs, sources, log levels, and VBAR events	
IAAS-ES-SQL-Server Stats Dashboard	Displays the SQL database status and SQL server health for the hosts	
IAAS-ES-Data-Collector-Details	Displays the status of the Data Collector on each site	
IAAS-ES-Infrastructure Overall Status Dashboard	Displays the overall status of the devices at all sites	Isilon Details: Links to IAAS-ES-ISLION-Detailed Status Dashboard Detailed XtremIO Status: Links to IAAS-ES-XTREMIO-Detailed Dashboard Detailed Switch Status: Links to IAAS-ES-SWITCH-Detailed Status Dashboard Detailed FC6xx Status: Links to IAAS-ES-FC6XX-Detailed Status Dashboard Detailed Fx2 Status: Links to IAAS-ES-FX2-Detailed Status Dashboard Data Domain Detailed Status: Links to IAAS-ES-DATADOMAIN-Detailed Status Dashboard Detailed R6xx Status: Links to IAAS-ES-R6XX-Detailed Status Dashboard
IAAS-ES-IDM-Dashboard	Displays account and password information from the IDM	
IAAS-ES-SCCM-Dashboard	Displays general information from SCCM	
IAAS-ES-Host Dashboard	Displays host information provided by Metricbeat; displays CPU/memory usage, traffic information, and disk usage	

[Elastic Security] Detection rule monitoring	Displays the detection rule information of the system; displays number of rules and executions, statuses for rules, delays in rules, alert response times, and top rules by various criteria	

ART Applications Dashboards:

Application	Dashboard Name	Description
SOCET GXP		
	geo-ha-socet-PluginExecutionOverview	SOCET GXP plugin operation; pass/fail, duration, users, events
	geo-ha-socet-PublishDetails	Internal details of the SOCET GXP publish process; stages, duration, events
	geo-ha-socet-PublishOverview	SOCET GXP publish process; pass/fail, duration, users, events
	geo-ha-socet-Raw-SOCET-GXP-Log	SOCET GXP log records; errors, hosts, raw log entries
	geo-ha-socet-ScreenerOverview	SOCET GXP screener function; pass/fail, duration, users, events
GXP Xplorer		
	geo-ha-xplorer-AutotagOverview	GXP Xplorer Autotag aggregations/events; actions/tasks, events
	geo-ha-xplorer-BinOverview	GXP Xplorer Bin aggregations/events; actions/tasks/compartments/subcompartments, events
	geo-ha-xplorer-BrowseMetrics	GXP Xplorer Browse Metrics; files processes, generate times, ave/max/min duration, events
	geo-ha-xplorer-CatalogSearchMetrics	GXP Xplorer Catalog Search Metrics; types/durations
	geo-ha-xplorer-ConversionMetrics	GXP Xplorer Conversion aggregations/events; operations/outcomes, events
	geo-ha-xplorer-CrawlMetrics	GXP Xplorer Crawl Metrics; directories scanned, durations, events
	geo-ha-xplorer-DataManagementOverview	GXP Xplorer Data Management; actions/tasks, actions/products affected, tasks
	geo-ha-xplorer-DeleteOverview	GXP Xplorer Delete Overview; actions/tasks/products affected, events
	geo-ha-xplorer-ExportMetrics	GXP Xplorer Export Metrics; timeline, distribution, events
	geo-ha-xplorer-IngestDetails	GXP Xplorer Ingest Details; processing time timeline/percent, flux in, flux out, ave ingest time, event details
	geo-ha-xplorer-IngestMetrics	GXP Xplorer Ingest Metrics; products ingested timeline, processing time timeline, ave/fastest/slowest ingest time, products ingested/failed, events, pass/fail/errors
	geo-ha-xplorer-MoveOverview	GXP Xplorer Move aggregations/events; actions/tasks, events

	geo-ha-xplorer-Notifications	GXP Xplorer Notifications; type/status/description
	geo-ha-xplorer-OperationMetrics	GXP Xplorer Operation Metrics; number of operations/catalog searches/logins, operation distribution, operation counts
	geo-ha-xplorer-Overview	GXP Xplorer overview; user IPs, user operations, operations, catalog searches
	geo-ha-xplorer-ParseMetrics	GXP Xplorer Parse Metrics; files/time timelines, events
	geo-ha-xplorer-PurgeOverview	GXP Xplorer Purge aggregations/events; actions/tasks/on ingest, events
	geo-ha-xplorer-StandingOrderMetrics	GXP Xplorer Standing Order Metrics; timeline/status, events
	geo-ha-xplorer-StreamingMetrics	GXP Xplorer Streaming Metrics; streaming users, streams open, active streams, streams timeline
	geo-ha-xplorer-SynchronizationMetrics	GXP Xplorer Synchronization Metrics; sync failures, state change events, sync transfer events, sync task events
	geo-ha-xplorer-TilePyramidOverview	GXP Xplorer Tile Pyramid Overview; files processed/duration timelines, ave/max/min duration, events
	geo-ha-xplorer-UserMetrics	GXP Xplorer User Metrics; user timeline, user operations, user IPs, authentication failures, users
	geo-ha-xplorer-WTMOverview	GXP Xplorer WTM aggregations/events; actions/tasks/on ingest, events
MAAS		
	geo-fmv-maas-Applications	MAAS Applications; Capture Manager, DVAx, Map Coordinator, Realtime Stream Viewer
	geo-fmv-maas-Data-Loss	MAAS Data Loss; Lost Link, Missing KLV, Missing Packets, Discontinuities
	geo-fmv-maas-Records	MAAS Log Records; file count, record count, host/files, messages, doc subtypes, log levels, threads, locations, users, original message lengths/stats
	geo-fmv-maas-Services	MAAS Services; MCCS, LCCS, AOI, Wildfly
	geo-fmv-maas-System	MAAS System; MCCS, Lost Link, LCCS, Capture Failover, AOI, Storage, Missing KLV, Wildfly
SOAESB		
	dcgs-soaesb-Image Notification Dashboard	SOAESB Image Re-Ingest GUI; timeline, image ingest process records / link to tell SOAESB to reingest an image

	dcgs-soaesb-Core Logs Dashboard	SOAESB Log Records; log.level timeline/distribution, log records
ECP		
	geo-sensors-ecp-Logs Dashboard	ECP Log Records; log.level timeline, dashboard links, hosts/files, log.levels, log records
	geo-sensors-ecp-MissionConnection	ECP Mission Connection Messages; message timeline, dashboard links, messages/namespaces, log levels, request messages, status messages
	geo-sensors-ecp-PositionReport	ECP Position Report Messages; message timeline, dashboard links, messages/namespaces, log levels, uci namespace messages, uas namespace messages
	geo-sensors-ecp-QueryRequest	ECP Query Request Messages; message timeline, dashboard links, messages/namespaces, messages/capabilities, log levels, request messages, status messages, mission plans
	geo-sensors-ecp-QueryRequest-Action-Asars	ECP QueryRequestStatus Action ASARS Messages per-document details; dashboard links, document selector, mission plans, routes, planned from, actions, SAR subcapabilities, collection constraints, product output, image collection element, footprint, capability command commons, action targets
	geo-sensors-ecp-QueryRequest-Action-Syers	ECP QueryRequestStatus Action SYERS Messages per-document details; dashboard links, document selector, mission plans, routes, planned from, actions, action targets, action collection, footprint, action counts
	geo-sensors-ecp-QueryRequest-System/Subsystem	ECP QueryRequest System/Subsystem Messages; message timeline, log levels, dashboard links, system messages, subsystem messages
	geo-sensors-ecp-Services	ECP Services Messages; message timeline, dashboard links, messages/namespaces, log levels, service error messages, service messages, uci/uas service lifecycle requests, service lifecycle statuses
	geo-sensors-ecp-Xml Messages	ECP Content of log XML bodies; message timeline, dashboard links, messages/namespaces, log levels, XML messages
Unicorn		
	IAAS-ES-UNICORN	UNICORN dashboard; job-class logs, job-classless logs, service logs, quartz logs, AMPQ logs, action logs, error logs, system logs, log4net logs
Render		
	IAAS-ES-RENDER	Render dashboard; site/host/template/caveat/country selector, rest types, log level counts, total hosts, dependencies, log level timeline, JSON messages, payload messages
Guardian		

	IAAS-ES-Guardian-Dashboard	Guardian Dashboard; logs over time, syslog events, facilities, critical messages, top exec commands, severity over time, messages

5.5.4 Upgrade Logstash (All Sites)

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**.

In this version the following new users are being added for Logstash. These users are replacing the existing logstash users which will become deprecated in this version and will be removed in future versions.

- ls_internal is replacing logstash_internal
- ls_admin is replacing logstash_admin_user

IMPORTANT: On the production system the passwords for these users need to be the same on both clusters to allow failover to work properly. This will be taken care of during the course of this upgrade. The upgrade_logstash.sh script will communicate with both the ECH and WCH clusters when run in this environment. If you are upgrading the production system ensure both clusters are "Green" before continuing.

To ensure the new users' passwords are added to the Logstash keystore at each site a file (ls_users.dat) holding the encrypted passwords will be created during the initial Logstash upgrade. This file must be copied to the Master Repository Server and placed in the "yum/sync/elastic/keystores" directory. All other Logstash upgrades that occur after the initial will pull the passwords from this file and update each Logstash keystore.

IMPORTANT: Remember with the move to using Satellite any updates to the Master Repository Server must be published to be available at all sites.

You can use the **Logstash nodes** screen in Kibana to identify and monitor the Logstash instances and versions they are running.

1. Select the **Nodes** link in the **Logstash** area of **Cluster overview** page.

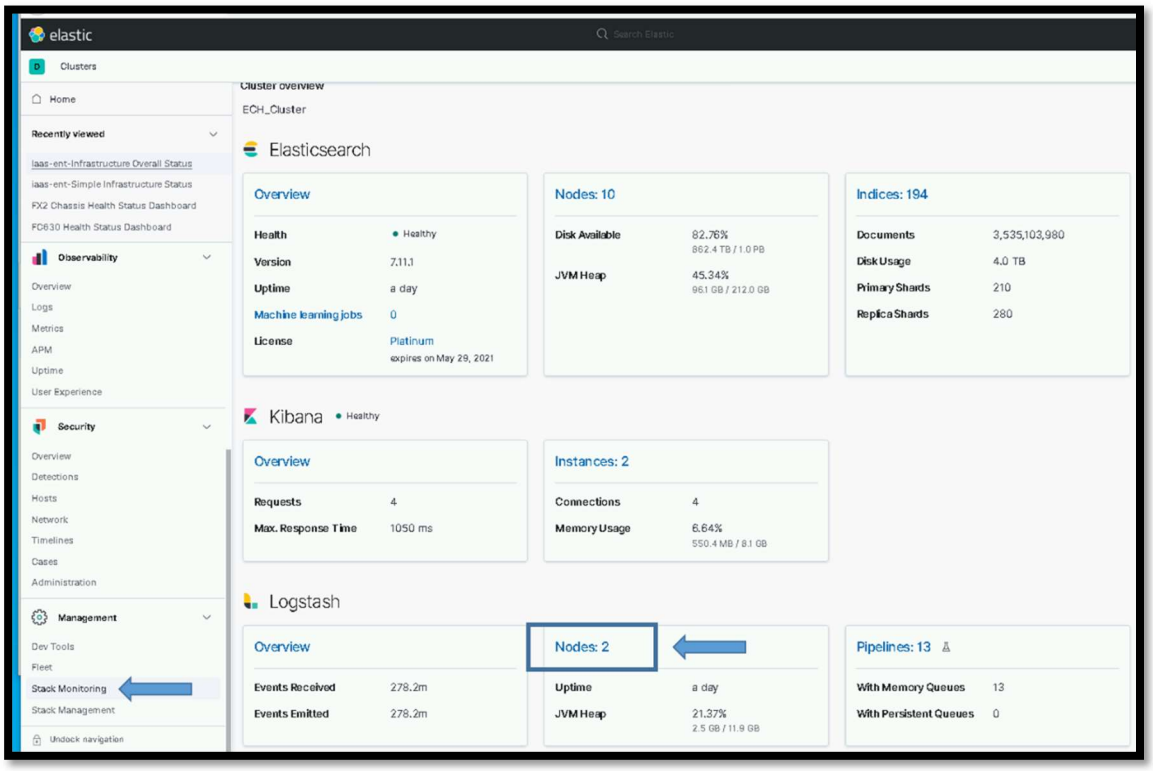


Figure 34. Logstash Node Monitoring Selection

2. The Logstash nodes page will show all the Logstash instances that are feeding data into Elastic and their version.

Figure 35. 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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_
Client/Elastic_Files/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. If this is the first Logstash being upgraded for this system, you will need to do a little extra work before proceeding to the other Logstash instances. You will see the following message printed to the screen:

```
********************* IMPORTANT NOTICE ***********************************
*                                                                          *
* Logstash Keystore created during installation.                           *
*                                                                          *
* You MUST Copy:                                                           *
*                                                                          *
*    /tmp/ls_users.dat                                                     *
*                                                                          *
* from this machine to the Master repository at:                           *
*                                                                          *
*    yum/sync/elastic/keystores/ls_users.dat                               *
*                                                                          *
* To allow installation of other Logstash instances.                       *
*                                                                          *
* Note: Ensure Satellite administrator re-publishes                        *
*       Elastic files repository after updating.                           *
*                                                                          *
********************* IMPORTANT NOTICE ***********************************
```

Figure 36- Message after First Logstash Upgrade

You **MUST** copy the ls_users.dat file to the Master Repository and place it in the elastic/keystores directory. After placing this file on the Repository, you will need to have the Satellite administrator re-publish the Elastic Files repository to ensure the ls_users.dat file is available for the rest of the Logstash Upgrades.

d. When the script finishes you should see the following message printed to the screen: **NOTE**: If this is the first Logstash being upgraded this message will follow the message shown in step c.

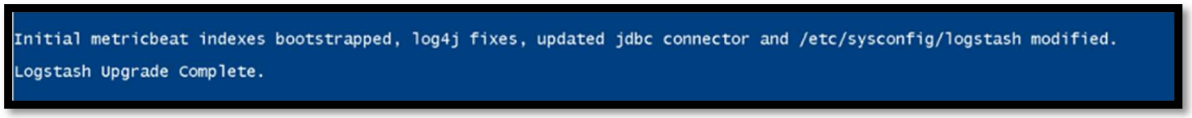


Figure 37. Upgrade Complete

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

Before proceeding to the next section ensure that all newly upgraded Logstash instances are working properly and communicating with Elasticsearch. Check The Logstash Node monitoring page in Kibana to validate the versions and ensure all instances have been upgraded.

5.5.5 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.

In this version every pipeline has been updated to use the new ls_admin user for Logtash to write data into Elasticsearch. The password for this user was added to the Logstash keystore on every Logstash instance as part of the upgrade_logstash.sh script. After loading the new pipelines you must validate that all Logstash instances are communicating correctly with Elastic.

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 8.6.2 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.

IMPORTANT: On the Production system this script MUST be run on one node from both the ECH and WCH clusters. Also execute the verification steps on both Clusters https://kibana and https://kibana-wch

 # sudo su

#

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/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.

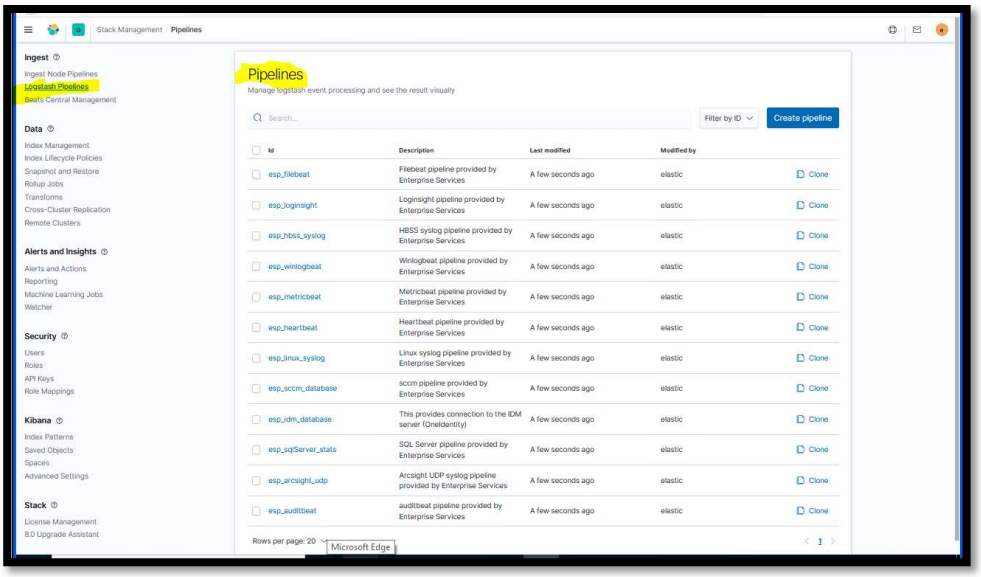


Figure 38. Pipelines

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.

After the pipelines have been loaded all Logstash instance will pull the new copy and start using it. You must now validate that all sites are still sending data into Elastic to ensure that the ls_admin password was correctly added to the keystores on each Logstash instance.

STOP: DO NOT CONTINUE UNTIL VALIDATING THAT YOU ARE STILL RECEIVING DATA FROM ALL LOGSTASH INSTANCES. This can be easily done in the Logstash section of Stack Monitoring.

5.5.6 Add Unicorn Pipeline

This section is to install the Logstash pipeline for collecting data from the Unicorn database.

Prior to executing this section, the unicorn database administrator will need to give the elastic service account at the site of the Unicorn database read access to the database. DO NOT PROCEED with this section or activate the esp_unicorn_database pipeline until this is completed.

- Determine the site of the Unicorn database that data will be extracted from
- Request that the xx_elastic.svc service account for that site be given read access to the database
- Obtain the **HOSTNAME** where the database resides from the Unicorn database admin; You will need this to run the activate script below.
- Obtain the **DATABASE NAME** from the Unicorn database admin; This name will be used in the JDBC connection string and will be needed to run the activate script below.
- DO NOT PROCEED UNTIL THIS IS COMPLETED

Execute the following script to configure and load the esp_unicorn_database pipeline

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/activate_unicorn.sh | bash
```

You will be prompted for the information obtained above:

- `Enter the data source for the UNICORN database: <Enter HOSTNAME>`
- `Enter the initial catalog for the UNICORN database: <Enter DATABASE NAME>`

Ensure script completes with no errors.

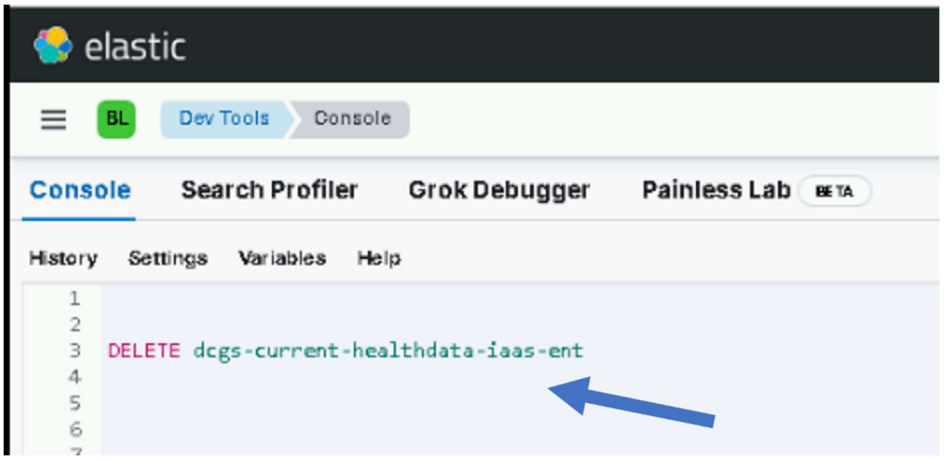
IMPORTANT: The esp_unicorn_database pipeline must be added to the Logstash node specific yaml file on puppet to be executed. Work with the puppet administrators to ensure this is completed.

5.5.7 Cleanup Current Health Data Index

There were changes made to the document ID's in the dcgs-current-healthdata-iaas-ent index for document subtypes "group", app-overall" and "datacollector". The prefix for these subtypes in the 8.6.2 version was "hostname", this has been changed to "site number". The loading and distribution of the new esp_filebeat-logstash pipeline in section 5.5.5 brought this update. To remove the old documents from the dcgs-current-healthdata-iaas-ent index we will simply delete the index and allow it to be regenerated.

To do this run the following command from the Kibana console in Dev Tools. NOTE: on the production system this should only be done on the ECH cluster.

- DELETE dcgs-current-healthdata-iaas-ent



NOTE: The index should be fully repopulated in about 5 minutes.

5.5.8 Setup Cross Cluster Replication

IMPORTANT: This section should only be executed on the production system after the WCH cluster has been installed. If you are installing on MTE/CTE or REL continue to the next section.

The following script will turn on Bi-directional Cross Cluster Replication (CCR) for the indexes in Elasticsearch. Once this is complete data that is ingested in the ECH will be replicated to the WCH Cluster and data that is ingested into the WCH cluster will be replicated to the ECH Cluster.

1. Login to any node on the ECH cluster and run the following script:

```
# curl -k
https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elast
ic_Files/install/load_CCR.sh | bash
```

2. Login to any node on the WCH cluster and run the same script.

After CCR setup is complete you will start seeing follower indexes on each cluster with a suffix of "-xxx" where xxx is either "ech" or "wch".

- Indexes on the WCH cluster that are following ECH indexes will have the "-ech" suffix
  Examples:
  - dcgs-syslog-iaas-ent-00-2023-12-13-000432-ech is the follower for dcgs-syslog-iaas-ent-00-2023-12-13-000435 on ECH
  - winlogbeat-8.11.3-2023-12-04-000014-ech is the follower for winlogbeat-8.11.3-2023-12-04-000014 on ECH
- Indexes on the ECH cluster that are following ECH indexes will have the "-wch" suffix
  Examples:
  - dcgs-syslog-iaas-ent-0a-2023-12-13-000154-wch is the follower for dcgs-syslog-iaas-ent-0a-2023-12-13-000154 on WCH
  - winlogbeat-8.11.3-2023-12-04-000011-wch is the follower for winlogbeat-8.11.3-2023-12-04-000011 on WCH

5.5.9 Upgrade Elastic 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.

IMPORTANT: No Data Collectors should be upgraded until all Logstash instances are upgraded and the Logstash pipelines have been upgraded. This is because this new version of the data collector brings automated failover between clusters and the new Logstash users are needed before this can be enabled.

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

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://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/installElasticDataCollector.sh | bash
   ```

   NOTE: The script will prompt for:

   a. Your password as your user account will be used to communicate with the Elasticsearch cluster(s).

b. The site of the Primary 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}
   ```

   If the Primary Cluster is “ech” the script assumes there is a 2nd cluster at “wch” and will make updates in both clusters.

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.

5.5.10 Update DLP ingest to use API

An HBSS SME will be needed to assist with this update.

A new DLP class has been added to the Elastic Data Collector to query the DLP data from HBSS. Before activating the new DLP data collector class the following must be done.

1. The HBSS SME must create a local user and password and assign it the correct privileges to read the DLP data using the new API. Have them contact the AFRL HBSS SME if guidance is needed.
2. The existing DLP ingest via the ArcSight connector should be disabled. To do this simply remove the esp_hbss_dlp-via-connector ingest pipeline from the logstash.yml at the site that is currently ingesting the data. This will require an update to the puppet configuration for that sitest logstash.yml file.

Once the above items are completed use the following sections to enable DLP collection from the Elastic Data Collector.

5.5.10.1 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 **Error! Reference source not found.**. 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.

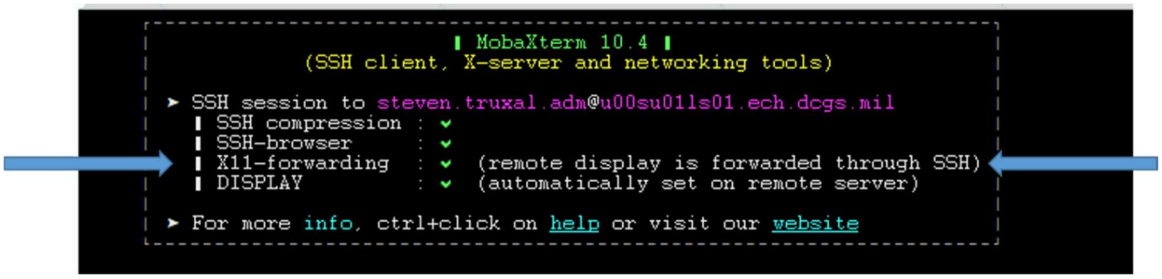


Figure 39 X11 Forwarding Setup Successfully

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 previous steps are still not getting X11-fowarding enabled, 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 commented out, 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.

5.5.10.2 Open the configurator GUI

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 40 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:

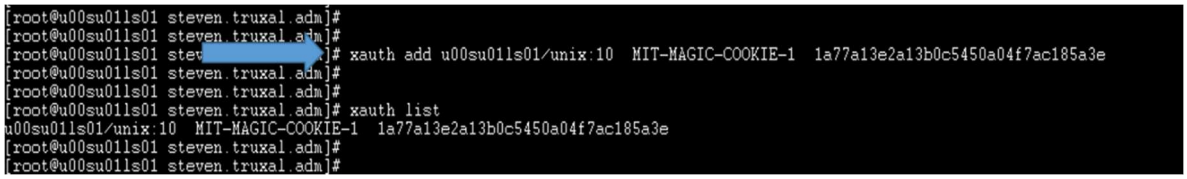


Figure 41 xauth add <cookie>

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 another mobaXterm icon in the taskbar.

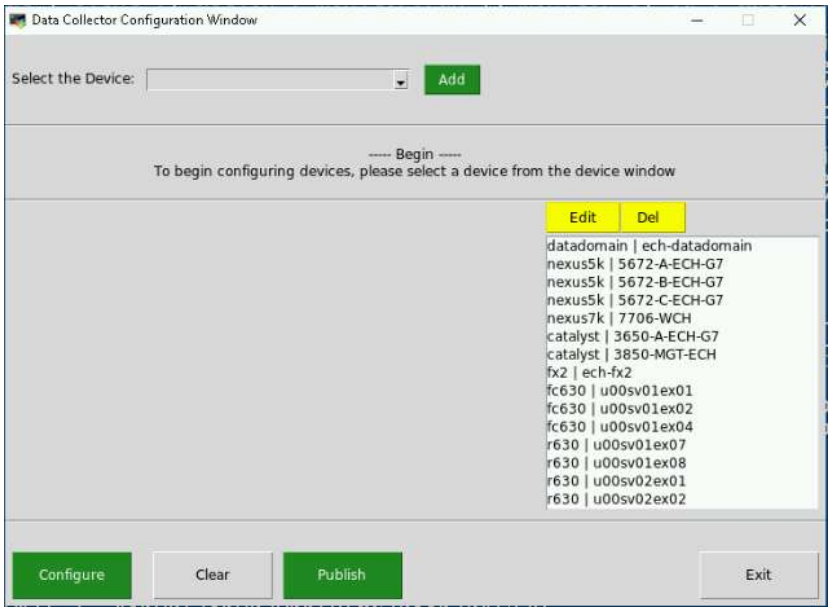


Figure 42 Device Configuration GUI

NOTE: This list of devices above is an example, what you will see is specific for the site where you bring up the configurator.

6. Select the "Select the Device" pull down and find the "hbss_dlp" option

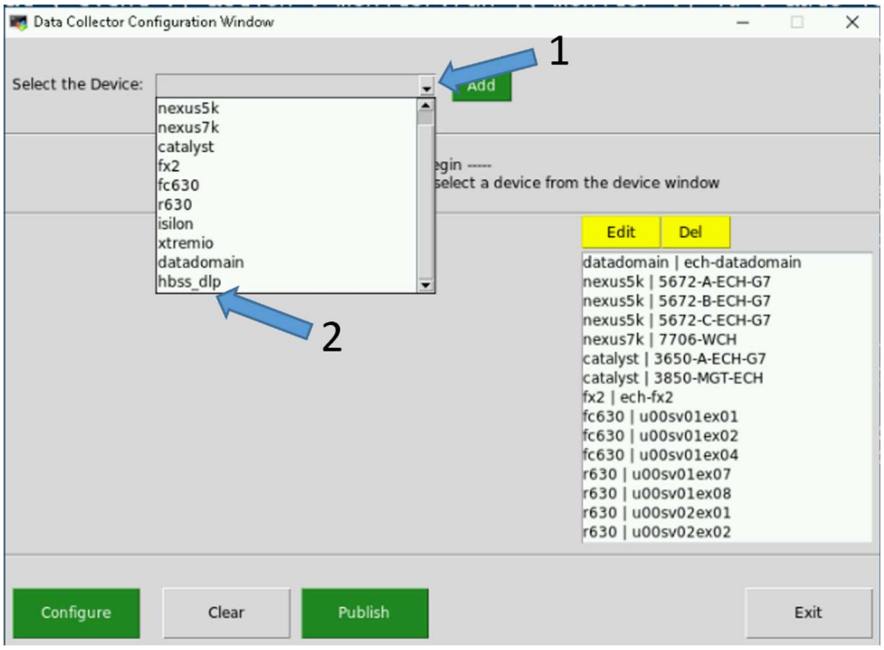


Figure 43-Example showing hbss_dlp option in menu.

7. Select the "Add" button

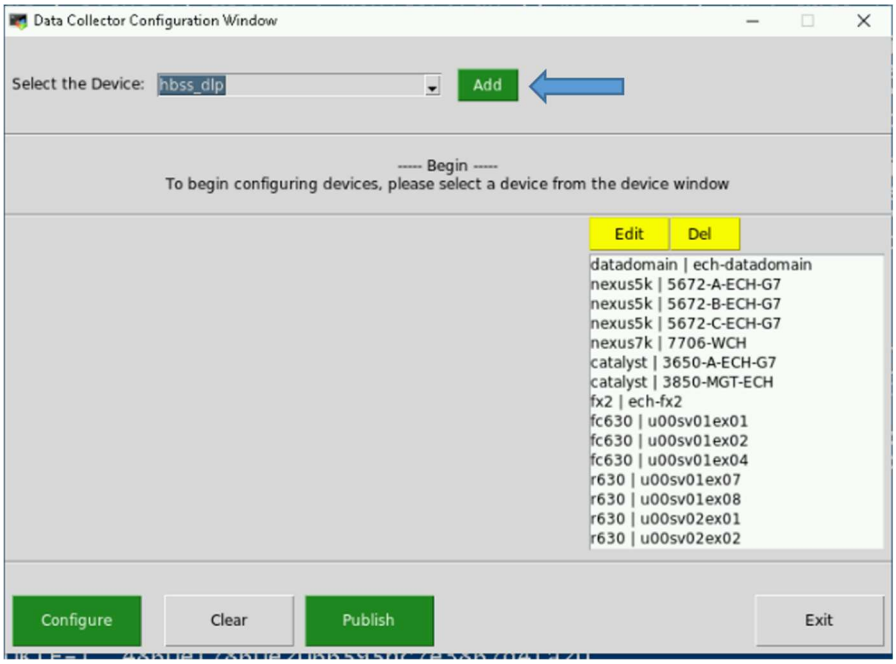


Figure 44- Example of selecting hbss_dlp option

8. Fill in all the fields and select the "Configure" button.

   NOTE: The username and password should be provided by the HBSS SME.

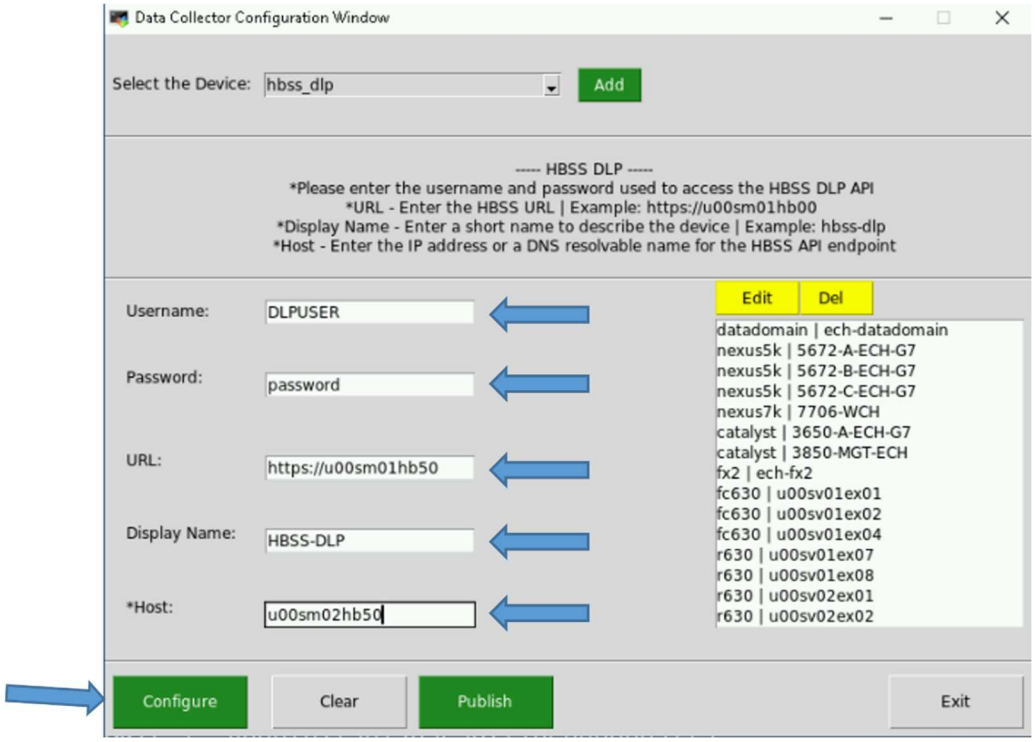


Figure 45- Example of configuration options for DLP

NOTE: The values above are only examples

9. Scroll the devices and validate your new entry appears in the list then select "Publish".

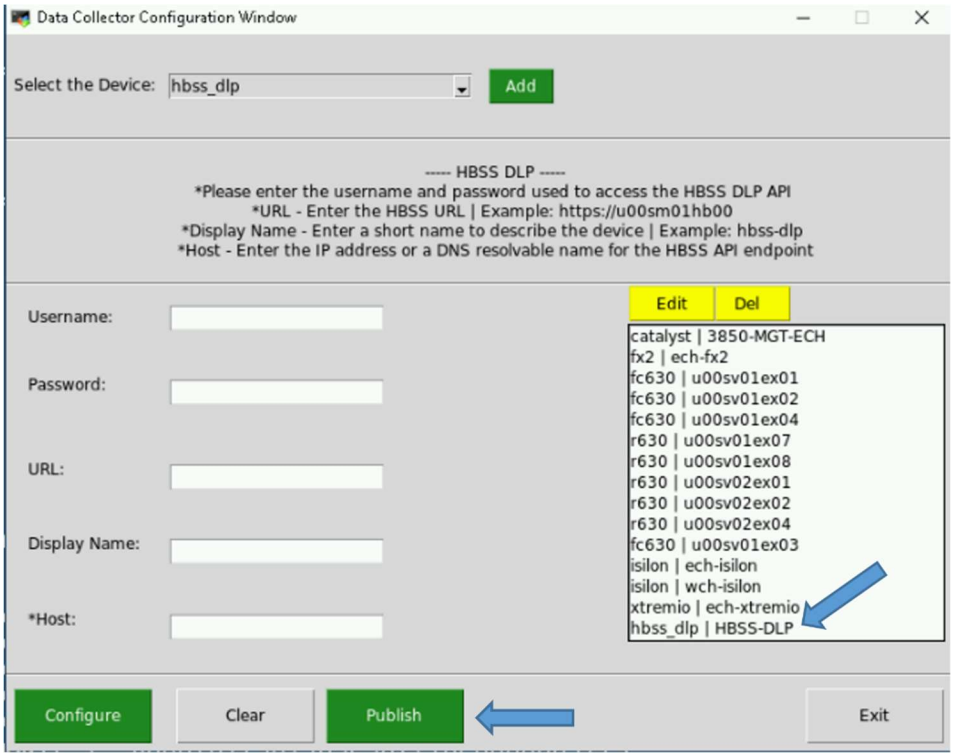


Figure 46-Example of configured DLP device showing in list

10. The following message will be displayed on the screen while access to each device is validated.

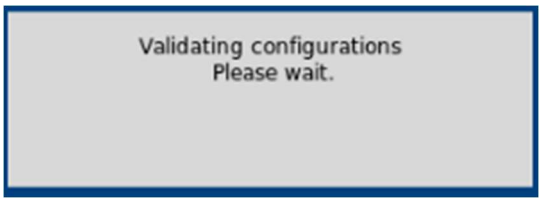


Figure 47- Validating notification

11. When validation is completed the "Results" window will appear. Ensure it says "Succss! All Devices were Configured!" Then hit "Continue"

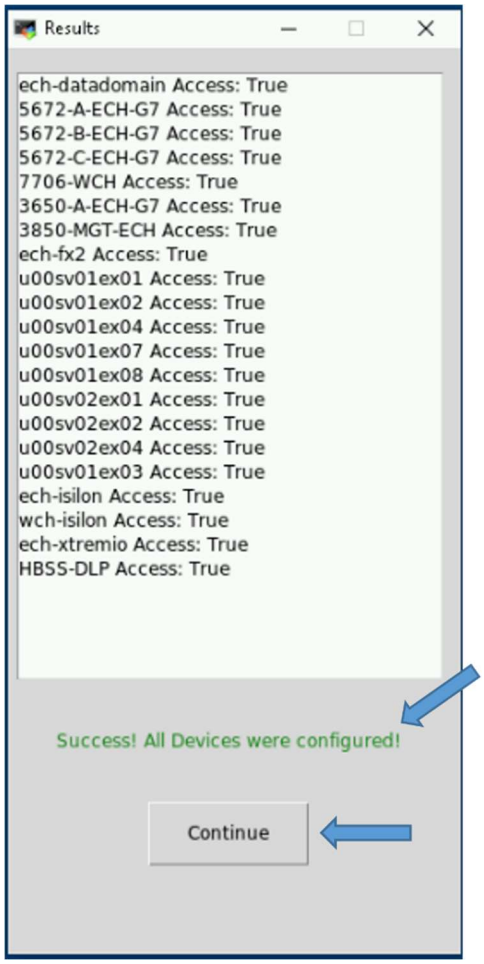


Figure 48- Example of successful results

12. The data collector should be automatically restarted after saving the new configuration. DLP events will now appear in the "dcgs-hbss_epo-iaas-ent" index and will be tagged with 'dlp' to allow easy filtering.

To validate the DLP thread is running in the data collector you can use the Elastic Data Collector API.

On the same logstash instance execute the following:

```
# curl -k localhost:9601
```

The following is an example of what the output will look like. You can see the HBSS-DLP thread is in the list of "ok" threads.

Figure 49 – Elastic DataCollector API output example showing DLP thread

5.5.11 Install watchers

There are two watchers for the production system with the WCH Cluster installed. All other systems will only have the stale state watcher.

Follow the instructions below to install the watchers:

1. Run the following command as root from any of the running Elastic nodes install the watcher.
2. `# curl -k https://satrepo/pulp/content/oadcgs/Library/custom/Elastic_Client/Elastic_Files/install/installWatchers.sh | bash`

3. Verify the watcher(s) loaded correctly.

- From the hamburger menu, select **Stack Management**.
- Select **Watcher** in the "Alerts and Insights" section.
- Validate the correct watcher(s) loaded.

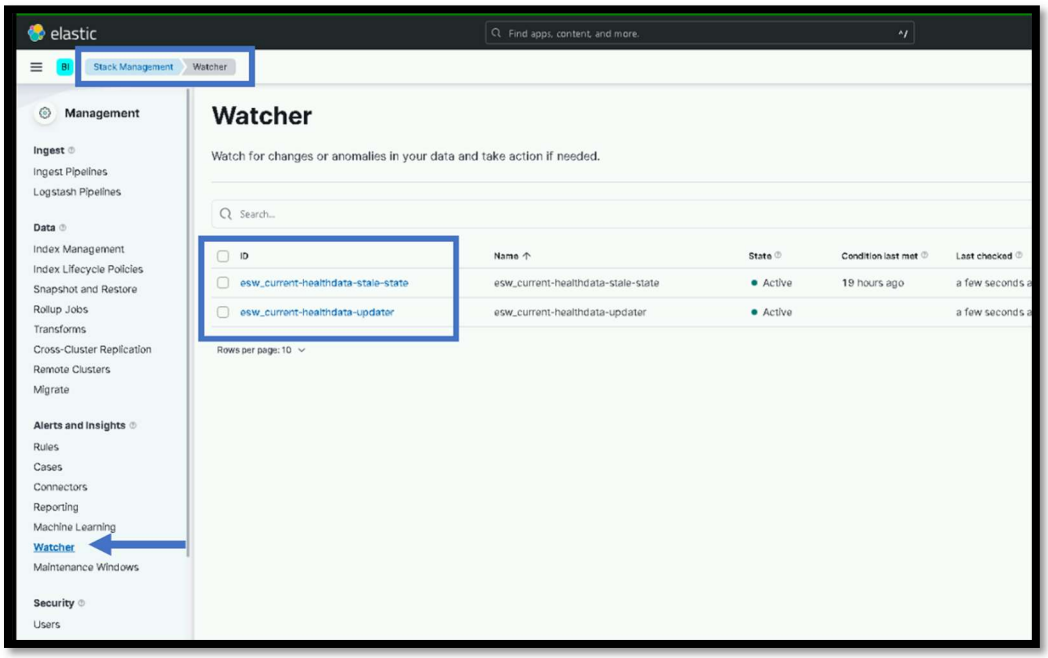


Figure 50 – Watchers loaded for production system

NOTE: For CTE/MTE and REL only the esw_current-healthdata-stale-state watcher should be loaded.

5.5.12 Upgrade Beats

5.5.12.1 Verify Beat Templates are Loaded

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.

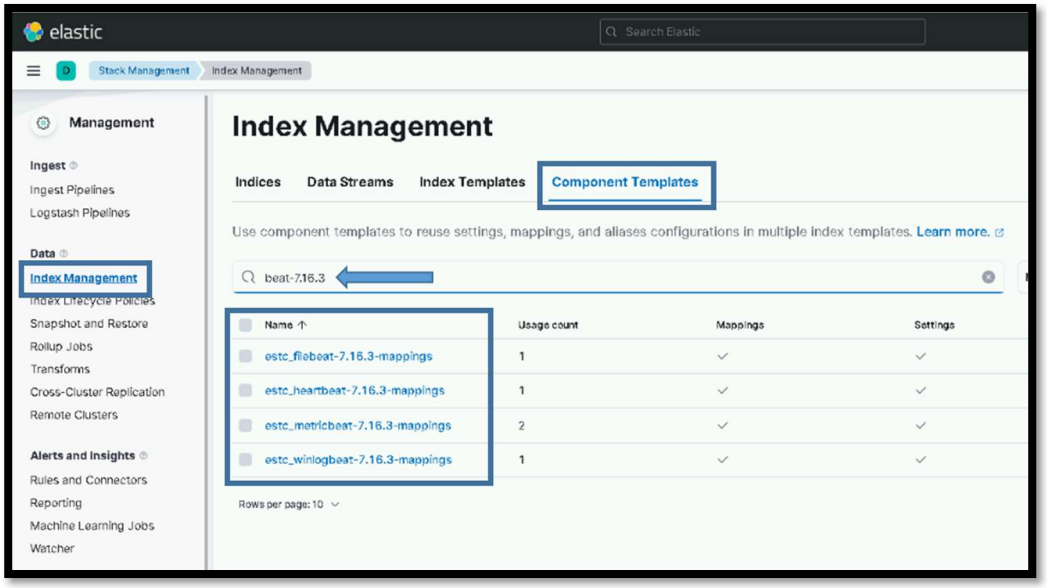


Figure 51. Example of beats component templates for version 7.16.3

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.

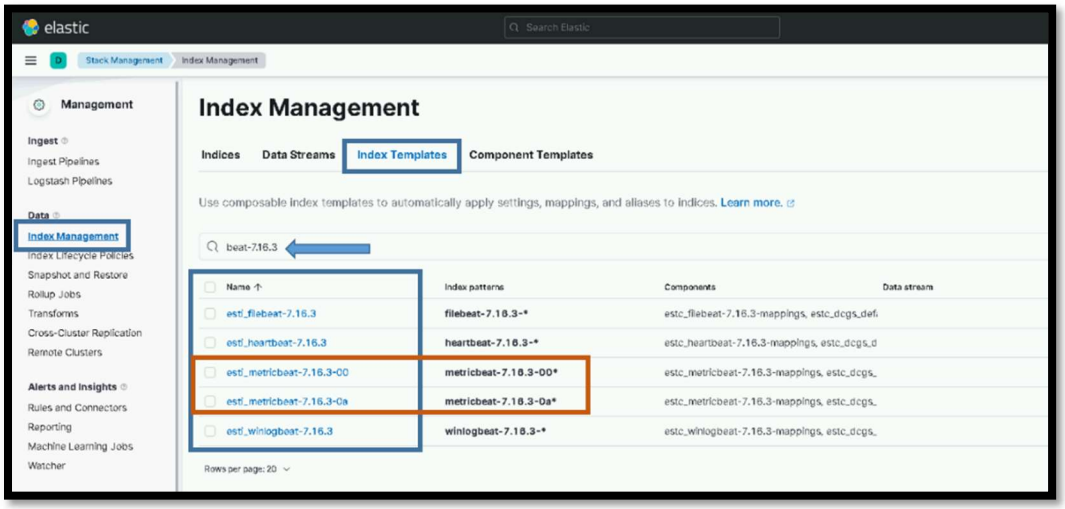


Figure 52. Example of beats index templates for version 7.16.3

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.

5.5.12.2 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-X.X.X.X\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
    - inputs.txt
- configs/metricbeat/inputs.d
    - example_config.app
- configs/filebeat
    - filebeat.yml
    - inputs.txt
    - inputsByService.txt
- configs/filebeat/inputs.d
    - ECP_AutoRouter.yml
    - ECP_Services.yml
    - ECP_Workstation.yml
    - example_config.yml

    - fmv_maas.yml
    - fmv_maas_server.yml
    - gxp_xplorer.yml
    - render.yml
    - socket_gxp.yml
    - unicorn.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

    NOTE: Older versions of the beats can also be removed at this time. It won't cause issues but there are no reasons to keep the old beat zip files.

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**.

5.5.12.3 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 zip file to DCMedia\Elastic\zipfiles directory.
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 hours Metricbeat has not been upgraded to the correct version reach out to an Elastic SME for guidance.

5.5.12.4 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.

1. Copy the following RPMs to the Master Elastic repo (/var/www/html/yum/elastic):
   - filebeat-{version}-x86_64.rpm
   - heartbeat-{version}-x86_64.rpm
   - metricbeat-{version}-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
   ```

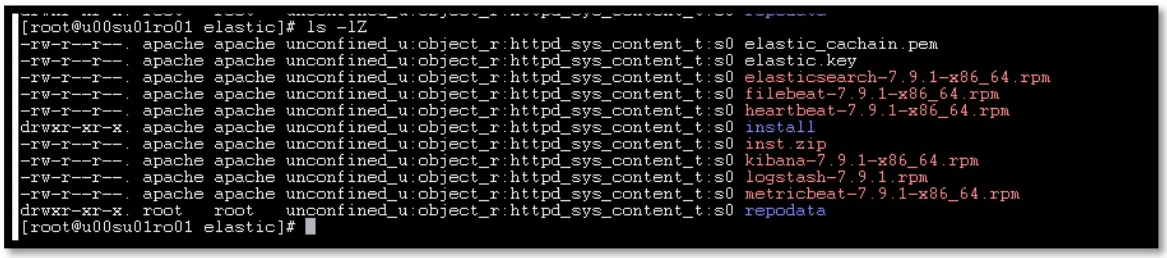


Figure 53. httpd_sys_context_t

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: Ensure there are 2 dashes before "detach-sign" and "armor" when running this command.

6. **IMPORTANT**: Notify Satellite administrator that updates have been made to the Master Repo and that the Elastic yum repository needs to be republished.

Once the Satellite administrator publishes the updated repository the RPMs will be available at all sites, Puppet will automatically install the new version when it is run on each host.

5.5.12.4.1 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.

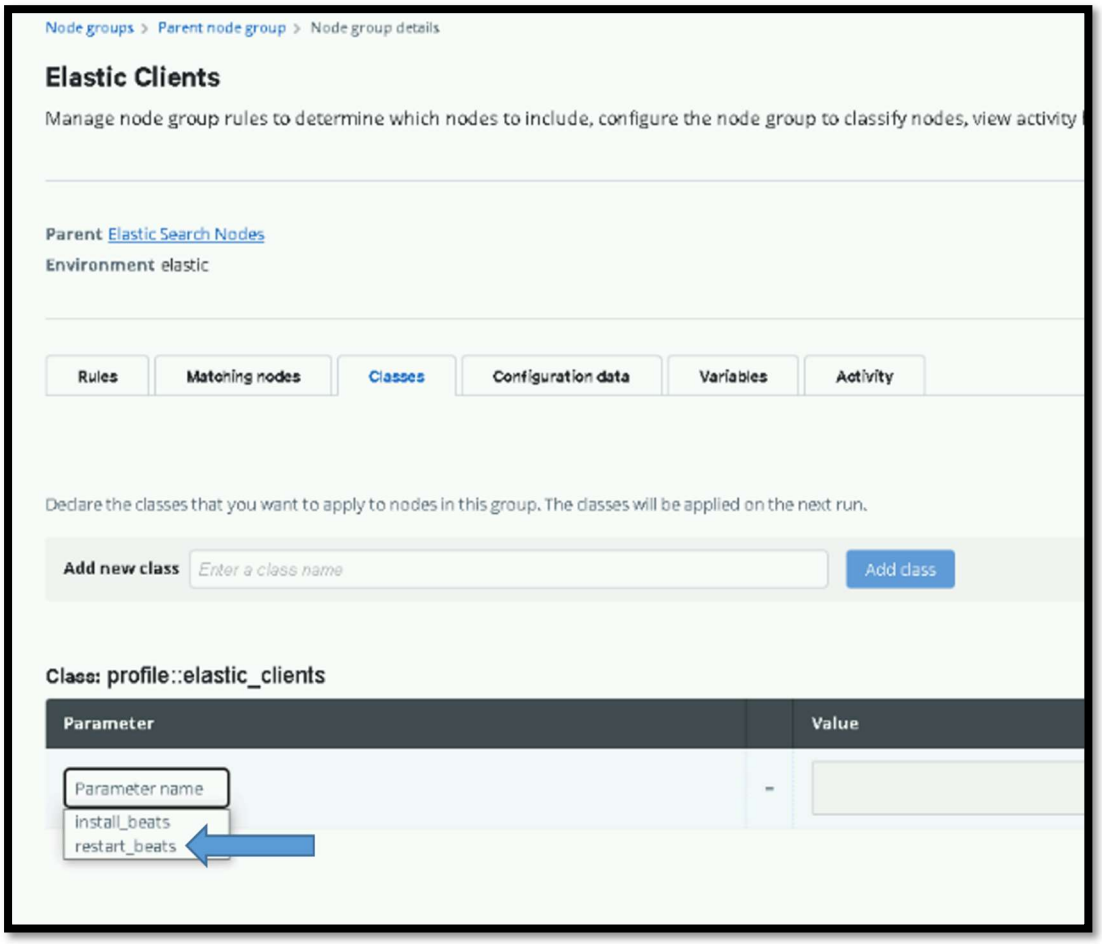


Figure 54. Select restart_beats

2. Once selected, change the value from the default (**false**) to **true** and click **Add to Node Group** (**Add Parameter** in older Puppet versions).

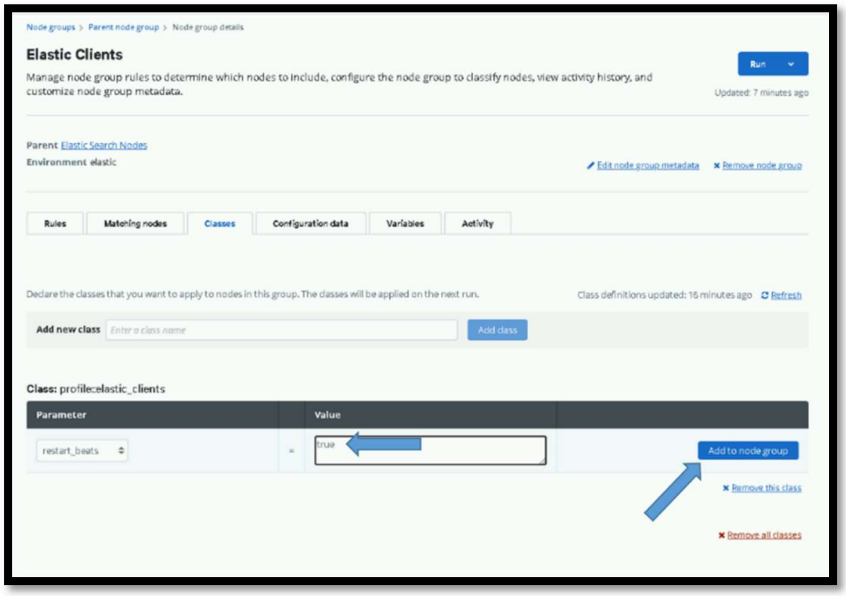
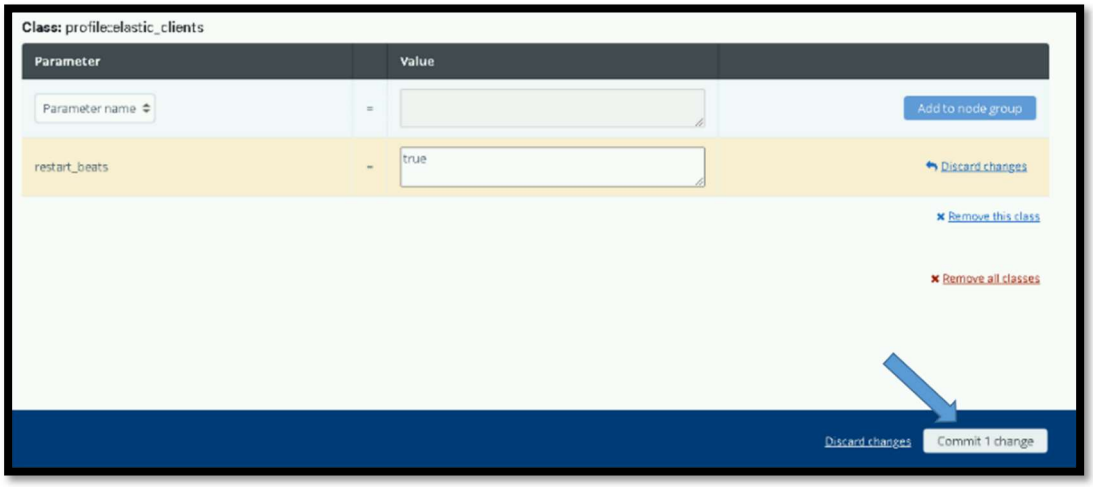


Figure 55. Add to Node Group

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 56. Remove Parameter

5.5.13 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.

5.5.13.1 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:

- Updates to get_ldap_hosts.sh script to only query data for site script is running from and force delete of temporary file to avoid warning
- Addition in init.pp to ensure the icmp_include.txt file exists in /etc/heartbeat directory

IMPORTANT: Do not overwrite your entire sandbox directory with the contents of this tar file. There are system specific configurations that need to be maintained. To update this module untar the file in /tmp or another directory and copy the following updated files into your sandbox:

Files changed for this upgrade:

- init.pp
- metadata.json
- get_ldap_hosts.sh.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.

New pipelines in this release should be added to the correct node specific entry:

- esp_unicorn_database – This pipeline should run at the site of the unicorn database.

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

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.

5.5.13.2 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.

IMPORTANT: backup your existing sandbox directory to compare with the new baseline to ensure any system specific updates are not lost.

Updates included in upgrade include:

- Kibana.yml is now controlled by puppet
- updated module for gathering kibana audits and logs
- Exclusion of udf filesystems for Metricbeat monitoring

Files changed in this version:

- init.pp
- es.module.kibana.yml.epp
- all.module.system.yml.epp

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.

5.5.13.3 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
```

a.

5.5.14 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.

5.6 List of Changes

See highlights of new features in Elastic, starting with the 8.6 release for each product.

Elastic 8.11: https://www.elastic.co/guide/en/elasticsearch/reference/8.11/release-highlights.html

Kibana 8.11: https://www.elastic.co/guide/en/kibana/8.11/whats-new.html

Logstash 8.11: https://www.elastic.co/guide/en/logstash/8.11/releasenotes.html

Beats 8.11: https://www.elastic.co/guide/en/beats/libbeat/8.11/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: ACRONYMS

Acronym	Definition
ACAS	Assured Compliance Assessment Solution
AD	Active Directory
AF DCGS	Air Force Distributed Common Ground Systems
API	Application Program Interface
ART	Agile Release Train
CA	Certificate Authority
CCR	Cross Cluster Replication
CSR	Certificate Signing Request
CTE	Controlled Test Environment
DNS	Domain Name Server
ECH	East Coast Hub
ELAC	Elastic Logging and Aggregation Cluster
ESS	Enterprise Service Segment
ePO	ePolicy Orchestrator
FQDN	Fully Qualified Domain Name
HBSS	Host Based Security System
IAAS	Identity Authentication and Authorization System
ID	Identification
ILM	Index Lifecycle Management
JVM	Java Virtual Machine
JWICS	Joint Worldwide Intelligence Communications System
MTE	Managed Test Environment

Acronym	Definition
NFS	Network File System
NOFORN	Not Releasable to Foreign Nationals
OA	Open Architecture
OneIM	One Identity Manager
OSIF	OA DCGS Secure Infrastructure Framework
PKI	Private Key Infrastructure
RBAC	Role-Based Access Control
REL	Releasable
RPM	RPM Package Manager
SAKM	Service Account Kerberos Management
SCCM	System Center Configuration Manager
SME	Subject Matter Expert
SNMP	Simple Network Management Protocol
SOAESB	Service Oriented Architecture Enterprise Service Bus
SQL	Structured Query Language
SSD	Solid State Drive
TCP	Transmission Control Protocol
URL	Universal Resource Locator
UUID	Unique User Identification
VM	Virtual Machine

APPENDIX B: 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
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 C: KNOWN ISSUES

You can delete this if there are no known issues.