lgtm enterprise

LGTM Enterprise
Installation and Upgrade Guide

Release 1.23.1, February 2020

Semmle™

Semmle Inc.
44 Montgomery Street
3rd Floor
San Francisco, CA 94104

LGTM Enterprise release 1.23.1
Document published February 11, 2020

Contents

Introduction **7**

What is LGTM? 7

About LGTM Enterprise 7

About this document 7

Upgrading LGTM to the latest version *8*

Time required 8

Related documentation 9

Prerequisites for installation **10**

Worker prerequisites 11

SSL certificates 11

Third-party software installed by LGTM **13**

Access to third-party packages during installation 13

Installation and upgrade methods **14**

Interactive deployment **15**

Installation overview 15

Deployment phases 15

Preparing to install or upgrade 16

Upload and unpack the LGTM distribution file(s) *16*

Upload the license file *18*

Install or upgrade? 18

Installing LGTM (interactive method) 19

Run the installation script *19*

Upgrading LGTM (interactive method) 24

Run the upgrade script *24*

Post-upgrade tasks *26*

Programmatic deployment **27**

Installation or upgrade? 27

Procedural overview 27
Creating and provisioning machines 27
Check for Java 28
Create a location for temporary LGTM files 28
Create a location for version-non-specific files 28
Locate the certificate, license, and integrations files 29
Uploading and unpacking the LGTM distribution file(s) 30
Deployment procedures 31
Single-machine programmatic deployment 32
Generate a cluster configuration file 32
Deploying a version of LGTM Enterprise 33
Programmatically adding integrations 37
Start LGTM 37
Install the license 38
Place reusable files under version control 38
Create a user account and log in 38
Distributed cluster programmatic deployment 39
Introduction 39
Prerequisites 40
Design and generate the cluster configuration 40
Install LGTM files on each machine in the cluster 44
Initialize the LGTM database and start all components 46
Including integrations in a programmatic installation 50
Saving the integration settings 50
Adding the integration settings to the programmatic installation 52
AWS deployment 53
Installing LGTM on AWS 53
Preparation 53
Launch an EC2 instance containing LGTM Enterprise 55

Basic details about the LGTM instance 60
Upgrading LGTM on AWS 61
Dockerized deployment **65**
Assumptions 65
Distribution file for Docker 65
Deployment using Helm 66
Configuration options 66
Worked example 67
Prerequisites 67
Procedure 68
Post-installation procedures **73**
Accessing the administration interface 73
Modifying a Dockerized deployment 74
Useful help topics 75
Post-upgrade procedures **77**
Post-upgrade housekeeping 77
Appendix A—Notes for upgrading LGTM Enterprise **78**
Upgrading from 1.22 to 1.23 78
After you upgrade to 1.23 78
Upgrading from 1.21 to 1.22 79
Before you upgrade to 1.22 79
After you upgrade to 1.22 79
Upgrading from 1.20 to 1.21 79
Before you upgrade to 1.21 79
Upgrading from 1.19 to 1.20 80
Before you upgrade to 1.20 80
After you upgrade to 1.20 80
Upgrading from 1.18 to 1.19 80
After you upgrade to 1.19 80

Appendix B—Amazon EC2 instance types **82**

Introduction

What is LGTM?

LGTM is a variant analysis platform that combines deep semantic code search with data science insights to help developers ship secure code.

Note
"LGTM" is the most commonly posted comment on code review platforms (such as GitHub). It means: "looks good to me."

About LGTM Enterprise

LGTM Enterprise is built on Semmle's tried and tested analysis technology. The product is the on-premises twin of Semmle's LGTM.com platform—a publicly available, free to use, cloud-based service that continuously analyzes over 135,000 open source projects worked on by over 700,000 developers, and containing data for more than 39 million commits.

Most notably, LGTM Enterprise offers the full power of LGTM variant analysis to customers in an on-premises version. LGTM allows developers to efficiently browse alerts in their source code (using either a web browser or a plugin for their IDE), while at the same time incorporating broader insights. It seamlessly integrates with platforms like GitHub Enterprise and Bitbucket Server for code review, authorization, and authentication.

About this document

This guide provides instructions for installing and upgrading LGTM Enterprise.

Four methods are described:

- **Interactive deployment**

  This involves running the supplied installation script, or upgrade script, and entering information in response to prompts.

By default, the installation script installs LGTM Enterprise on a single machine and runs just one general worker daemon to perform project analysis. It's therefore not suitable for analyzing many or large projects. It can, however, be used for demonstration or simple evaluation purposes. It also provides the basis for a larger deployment, by giving you a simple system to which you can add worker host machines, to provide additional processing resources. You can find information on expanding your system in LGTM's administrator help.

- **Programmatic deployment**

  This method is designed for system administrators who want to be able to install or upgrade LGTM Enterprise on machines in an easily repeatable way, without having to step through prompts for information, entering details manually.

- **AWS deployment**

  Deploying on Amazon Web Services (AWS) is a great way to get an instance of LGTM Enterprise up and running very quickly on a single virtual machine, on a robust and easy-to-maintain virtualization platform. To install LGTM Enterprise on AWS, you select an Amazon Machine Image (AMI) and use this to launch an Elastic Compute Cloud (EC2) instance, with a separate Elastic Block Store (EBS) volume for persistent data. You then log in to LGTM Enterprise and configure the system as required in the administration interface.

- **Dockerized deployment**

  If you already use a container management system such as Kubernetes to deploy applications within Docker containers, this method offers a familiar workflow. Out of the box, the supplied Docker package allows you to very quickly install a basic LGTM Enterprise instance, which you can then modify as required.

Upgrading LGTM to the latest version

The upgrade process is very similar to the installation process. On a distributed LGTM cluster, upgrades must be performed on each machine in the cluster. The instructions in this guide cover upgrading your system.

Before upgrading, check the "Appendix A—Notes for upgrading LGTM Enterprise" on page 78 for any pre-upgrade steps you need to take.

Time required

You should allow an hour to perform an initial test installation of a basic, single-machine system. Installing and setting up LGTM Enterprise on a single virtual machine is a good way to start if you have not performed an installation before. Installing a larger, multi-machine system for live use will take proportionally longer, depending on the architecture you choose.

After installing LGTM Enterprise you should check the system by adding some projects for analysis (you can delete these later, if required). You need to allow the application time to perform the analysis. If you're installing an initial, temporary instance of LGTM Enterprise, it's a good idea to add just a few projects at first, and come back in three or four hours to check that results are available on the site. Alternatively, you can add more projects and leave the application to run overnight before looking at the analysis results.

Upgrade is a simple process and, in most cases, should not take more than an hour (excluding any pre- and post-upgrade steps that may be required).

Related documentation

- *LGTM Enterprise System Architecture* (PDF)
- *LGTM Enterprise System Requirements* (PDF)
- LGTM Enterprise administrator help

  This is available at help.semmle.com/lgtm-enterprise/admin, or by clicking **Admin help** at the top of the administration pages in LGTM Enterprise. You may wish to read the guidelines on securing LGTM before you plan your installation.

Prerequisites for installation

The LGTM Enterprise distribution is designed to make installation and configuration as easy as possible. It includes almost all of the components you need. For example, the packages that are installed include the nginx HTTP and reverse proxy server for serving pages from the LGTM Enterprise web application. Some prerequisites are, however, assumed and it is important that you make sure these are all in place before proceeding with installation.

The instructions in this document assume that all of the following requirements have been met:

- You have downloaded the appropriate distribution file(s) from the LGTM Enterprise releases page.

  Three distribution files are available: one for running LGTM on Debian or Ubuntu (`lgtm-<version>-deb.tar.gz`), one for Red Hat or CentOS (`lgtm-<version>-rpm.tar.gz`), plus an MSI file for Windows worker host machines (`lgtm-worker_<version>.msi`). The two Linux distribution files contain everything you need to install or upgrade LGTM Enterprise on the specific Linux distros. To run LGTM worker daemons on a platform other than that used by the non-worker machines in your system you will need to download more than one LGTM distribution file.

- You have a valid license for LGTM Enterprise.

- You have familiarized yourself with the architecture of an LGTM Enterprise system and decided on a specific architecture for your installation. For information, see the *LGTM Enterprise System Architecture* PDF, or the topics under "System architecture overview" in the administrator help.

  For example, a demo system can be contained on a single machine. A trial system might host all of the control pool components on one machine and use one or more additional machines for the work pool. A live deployment of a large system might use several more machines for a fully distributed cluster design.

- The machines that are going to host LGTM Enterprise are running a supported operating system. See the details in the *LGTM Enterprise System Requirements* PDF, or the administrator help.

  We recommend that you start with a fresh Linux installation (for example, on a virtual machine).

- The machines that are going to host LGTM Enterprise are appropriately resourced.

  For example, for a single-machine installation for demo or simple evaluation purposes, the machine should have a minimum of 8 cores, 16 GB of RAM, and 100 GB of free disk space.

Important
For a live deployment of LGTM Enterprise see the separate *LGTM Enterprise System Requirements* PDF, or the "System requirements" topic in the administrator help.

- On the machine that is going to host the LGTM web application, port 443 is free and accessible remotely. For more information about port usage, see the *LGTM Enterprise System Architecture* PDF, or Network connections in the administrator help.

Note
Alternatively, you can configure the bundled third-party nginx web proxy to use a different port for SSL connections. However, this is beyond the scope of this document. For more information, see the nginx documentation.

- The installing user must be able to run commands using sudo on all LGTM host machines.

Worker prerequisites

To analyze projects with LGTM, the machines on which worker daemons will build and analyze projects must have all of the environment prerequisites to allow this to be done. These prerequisites are likely to vary from project to project.

You can install the prerequisites for the projects you plan to build at any time, however, it’s usually simplest to install LGTM Enterprise first, then check the environment (see the topic "Configuring hosts to analyze code" in the administrator help), and then add any missing prerequisites. On Linux computers, this ensures that EPEL (Extra Packages for Enterprise Linux) is present and can be used to install the missing packages.

For full system requirement details for a live deployment of LGTM Enterprise, see the separate *LGTM Enterprise System Requirements* PDF (or you can find the same information in the administrator help).

SSL certificates

Note
This section is not relevant to Dockerized deployments of LGTM Enterprise, where SSL termination is handled by the container management system. See "Dockerized

deployment" on page 65.

To ensure that data transferred to and from servers is protected, LGTM Enterprise relies on SSL/TLS encryption. When you install LGTM Enterprise, self-signed certificates are automatically generated to protect communication between internal components, for example, the web application and the file store.

By default, the web application is configured to use a self-signed certificate to communicate externally. Using this certificate can be useful initially for testing purposes. However, while you can establish trust using the self-signed certificate, this requires you to disable SSL verification in your repository management system. This is not recommended for production systems and corporate policies usually prevent this. Consequently, you require a true, trusted certificate for the domain at which the LGTM Enterprise web application will be located.

If you *do* decide to use the default self-signed certificate, be aware that:

- Pull request integration with will fail unless you carry out additional configuration on the repository system to permit self-signed certificates.
- Users will see a browser security message when they attempt to access LGTM Enterprise. They may need to add a security exception for the site before they can view it (behavior varies for different browsers).
- Users will be unable to use the LGTM plugins to view alerts in their IDE.

Important

Before installing LGTM Enterprise, obtain an SSL certificate and private key for the domain at which the LGTM Enterprise web application will be located. These should be in separate files, in PEM format. The certificate must be issued by a trusted certification authority and must *not* require a password to be supplied.

For more information about SSL certificates for LGTM Enterprise see "Certificate details" in the LGTM administrator help on help.semmle.com.

Third-party software installed by LGTM

LGTM Enterprise uses the following third-party software. If these resources are not found on the relevant machines during installation of LGTM Enterprise, they are automatically installed.

- Java JRE 8 (if necessary, OpenJDK 8 will be installed)
- Python 2.7
- PostgreSQL 9.5 or later
- RabbitMQ Server
- Nginx
- Git
- Subversion
- Solr
- Minio

Solr and Minio are supplied within the LGTM distribution. If the other software listed above needs to be installed it is downloaded from an external repository. Except where the version number is shown above, the latest available version is downloaded. Any dependencies are also installed.

Access to third-party packages during installation

The simplest way to install LGTM Enterprise is to enable internet access and to run the installation script supplied with the application. This enables the installer to download and install the required packages from their official online package repositories using the standard tools available in your Linux distribution. If you're using Red Hat Enterprise or CentOS, this will require access to the EPEL (Extra Packages for Enterprise Linux) repository. Once LGTM has been deployed, internet access can be closed down.

If internet access is not available, you can do either of the following:

- Configure the machine so that it can access an internally hosted mirror that contains these package repositories and their dependencies. The installer will download and install the required packages from your internal mirror.

- *For installations on Red Hat or CentOS:* Extract the third-party dependencies pack before you run the installer. Details for doing this are included in the installation instructions.

Installation and upgrade methods

This guide describes four methods of installing or upgrading LGTM Enterprise:

- **Interactive deployment**

  This is perhaps the simplest way to install or upgrade LGTM Enterprise. Just unpack the distribution and run the supplied installation script or upgrade script. You will be prompted to enter information about your system.

  To use this method, see: "Interactive deployment" on the next page.

- **Programmatic deployment**

  This method provides a way to store information about the system you want to create, so that you can install or upgrade LGTM Enterprise repeatedly, without having to manually enter replies to prompts. More initial preparation is required—to parameterize information and set up the means of issuing commands (either using a configuration management system, such as Ansible, or as a shell script)—but having done this, installing and upgrading LGTM Enterprise on virtual machines becomes a quick and easy to repeat process.

  To use this method, see: "Programmatic deployment" on page 27.

- **AWS deployment**

  This process involves provisioning a new Amazon EC2 instance from an Amazon Machine Image (AMI) on Amazon Web Services (AWS), and adding a separate Elastic Block Store (EBS) volume for persistent data. The complete LGTM Enterprise instance runs on a single machine, making it very easy to configure and maintain.

  To use this method, see: "AWS deployment" on page 53.

- **Dockerized deployment**

  Use the supplied bundle of Docker images to create a set of Docker containers, using a container management system such as Kubernetes. Each of the main LGTM Enterprise services runs in a separate Docker container. If you already deploy applications in Docker, this is a very convenient way to install LGTM Enterprise.

  To use this method, see: "Dockerized deployment" on page 65.

Interactive deployment

LGTM Enterprise comes with scripts that step you through the installation and upgrade processes. These scripts provide the easiest way to perform a one-off installation or upgrade of LGTM Enterprise. This section of the guide tells you how to use this interactive method of installing or upgrading LGTM Enterprise.

If, however, you need to perform repeated installations or upgrades with parameterized input that does not require you to type in answers to prompts for information, then you'll probably want to consider configuring programmatic installation or upgrade for LGTM Enterprise—see "Programmatic deployment" on page 27.

Installation overview

If you are performing an installation, the instructions given here assume that you have already set up one or more machines on which to install LGTM Enterprise See the "Prerequisites for installation" on page 10.

Tip

Typically you will be installing onto virtual machines (VMs) and it's strongly recommended that you take a snapshot of the VMs before starting the installation, so that you can easily start the process over if required.

Deployment phases

The interactive deployment method comprises two main phases:

1. **Preparing to install or upgrade**

   Uploading the distribution file(s) and your license file, then unpackaging the distribution file(s).

   See "Preparing to install or upgrade" on the next page.

2. **Running the installer or upgrader**

   See:

- "Installing LGTM (interactive method)" on page 19
- "Upgrading LGTM (interactive method)" on page 24

After the installation or upgrade script finishes you will log into LGTM Enterprise and perform some checks. For an installation, you can then start to configure your system as required. Links to useful help topics on these subjects are provided later in this guide.

Preparing to install or upgrade

Before using the interactive scripts to install or upgrade LGTM Enterprise, you need to perform the following tasks.

Upload and unpack the LGTM distribution file(s)

1. Open a command console for the machine that will host the LGTM coordinator (part of the control pool).

   For example, connect to the server using an SSH client.

2. Create a directory to store installation and configuration files used during an installation or upgrade.

   In this guide, the commands use an example directory of `lgtm-releases`, at the `$HOME` location, but you can use any directory for this purpose. For example:

   ```
   mkdir $HOME/lgtm-releases
   ```

   > **Note**
   > The files in this location are only used during the installation and upgrade processes, or when you deploy changes to your LGTM cluster.

3. Upload the appropriate distribution file(s) for LGTM Enterprise (`lgtm-`*`<version>`*`-`*`<platform>`*`.tar.gz`) to the `lgtm-releases` directory.

   As of version 1.21 of LGTM Enterprise, there are three distribution files: one for Debian or Ubuntu, one for Red Hat or CentOS, and an MSI file for worker host machines running Windows. Upload the appropriate files for your deployment.

> **Important**
>
> For versions of LGTM Enterprise prior to 1.21 there is just one distribution file: `lgtm-<version>.tar.gz`.

4. *For Red Hat and CentOS only:* If internet access is not available and you want to perform an offline installation or upgrade, upload the supplied `lgtm-third-party-rpms-<yyyy-mm-dd>.tar.gz` file into the same directory.

5. In the command console, change to the `lgtm-releases` directory.

6. Extract the contents of the LGTM distribution package(s).

   - *For Debian and Ubuntu only*

     a. Untar the "deb" package:

     ```
     tar -xvf lgtm-<version>-deb.tar.gz
     ```

     b. If you have (or intend to have) worker host machines running RedHat, untar the "rpm" package:

     ```
     tar -xvf lgtm-<version>-rpm.tar.gz lgtm-<version>/lgtm/lgtm-worker-<version>.noarch.rpm
     ```

     c. If you have (or intend to have) worker host machines running Windows, move the `lgtm-worker_<version>.msi` file into the `lgtm-<version>/lgtm` directory.

   - *For Red Hat and CentOS only*

     a. Untar the "rpm" package:

     ```
     tar -xvf lgtm-<version>-rpm.tar.gz
     ```

     b. If you have (or intend to have) worker host machines running Debian, untar the "deb" package:

     ```
     tar -xvf lgtm-<version>-deb.tar.gz lgtm-<version>/lgtm/lgtm-worker-<version>_all.deb
     ```

c. If you have (or intend to have) worker host machines running Windows, move the `lgtm-worker_<version>.msi` file into the `lgtm-<version>/lgtm` directory.

> **Important**
>
> For versions of LGTM Enterprise prior to 1.21, untar the single distribution file:
>
> ```
> tar -xvf lgtm-<version>.tar.gz
> ```

The `lgtm-releases` directory now contains an `lgtm-<version>` subdirectory, within which are subdirectories called `generated` and `lgtm`. If you have (or intend to have) workers running on a different platform to the main part of the system, you've now added the appropriate worker files to the `lgtm` directory.

7. *For Red Hat and CentOS only:* If you're performing an offline installation or upgrade, extract the contents of the dependencies package into the `lgtm-<version>/lgtm` directory:

   ```
   tar -xvf lgtm-third-party-rpms-<yyyy-mm-dd>.tar.gz -C lgtm-<version>/lgtm
   ```

   This adds the directories: `database`, `queue`, and `web`, the presence of which causes LGTM to perform an offline installation.

Upload the license file

1. Upload your LGTM license file to the `lgtm-releases/lgtm-<version>/lgtm` directory which you created previously.

2. Change the file name to `license.dat` if it is not already called this.

3. Within the `lgtm-releases` directory, create a subdirectory called `state`.

Install or upgrade?

You're now ready to run the installation or upgrade script and answer the prompts.

Continue with the appropriate section for an installation or an upgrade:

- "Installing LGTM (interactive method)" on the next page
- "Upgrading LGTM (interactive method)" on page 24

Installing LGTM (interactive method)

The interactive installation script is stored in the `lgtm` subdirectory of the `lgtm-<version>` directory that you extracted earlier, for example: `lgtm-releases/lgtm-<version>/lgtm/install.sh`. Typically, you will need to use the sudo command to run the script—this grants the elevated permissions that are needed to make the required file system changes.

The script displays a series of prompts for input. These prompts, and the details you enter (with the exception of passwords), are saved to an `install-debug.bin` file, located alongside the `install.sh` script. This log file can be useful for debugging installation issues.

Run the installation script

From the `lgtm-releases/lgtm-<version>/lgtm` directory enter:

```
sudo ./install.sh
```

Reply to the prompts, as follows:

1. `What is the fully qualified domain name of this machine (this will be used for internal communication between different cluster machines)?`

   The text in square brackets gives the fully qualified domain name (FQDN) of the current machine. Generally this value is correct and you can just press Enter to use this name. Alternatively, enter the correct FQDN here.

   > **Important**
   >
   > If the server has both private and public FQDNs, specify the private FQDN here.

2. `A cluster configuration for a single machine installation has been created. Do you want to edit it manually before continuing?`

   The cluster configuration file specifies the hostnames of machines on which the various cluster components will run. It also specifies the number of worker daemons that should be run to build and analyze projects. For more information about editing the configuration file, see the topic "Editing and deploying the cluster configuration" in the administrator help.

   If you want to configure a particular setup at this stage—rather than using the simple, single-machine setup provided by the installer—type y and press Enter to edit the file. Alternatively, press Enter to continue to the next step.

> **Note**
> Editing the cluster configuration at this stage is optional. You can modify the system later if required. If you have not installed LGTM Enterprise before, you may want to use the default configuration for now, and then modify it later—for example, increasing the resources for analysis by adding more worker daemons.

If you entered y:

If a local text editor is available, the cluster configuration file is opened for editing. Otherwise, you're prompted to choose an editor.

After you have edited and saved the file you are returned to the installation process.

3. `Please provide a password for encrypting the installation manifest:`

   The LGTM manifest file contains sensitive information (for example, authentication details) which is encrypted. You are prompted to enter a password for encrypting/decrypting this data. You will need this password whenever the manifest is changed, for example, when you upgrade LGTM.

   > **Note**
   > Defining a password is optional. If control of the credentials stored in the manifest is not a concern, or if the file is protected by some other means, then you do not need to enter a password.
   >
   > If you press Enter without giving a password, the manifest is encrypted with a fixed password which provides only a superficial degree of security.
   >
   > If you define a password, make sure you store it securely as you will need it to upgrade the system in future.

   Enter a password for encrypting/decrypting the file. This is not displayed on the screen. You're asked to confirm this password before the script continues.

   The installer now installs a number of packages and then starts LGTM's core services. The output ends with messages such as:

```
[info] Starting rabbitmq-server-lgtm...
Created symlink /etc/systemd/system/multi-user.target.wants/rabbitmq-server-
lgtm.service → /lib/systemd/system/rabbitmq-server-lgtm.service.
[success] Started rabbitmq-server-lgtm.
[info] Starting solr-lgtm...
Created symlink /etc/systemd/system/multi-user.target.wants/solr-lgtm.service →
/lib/systemd/system/solr-lgtm.service.
[success] Started solr-lgtm.
[success] Core LGTM services on this machine have been started.
[info] Creating database...
Oct 24, 2019 1:10:28 PM com.semmle.cloud.commands.UpgradeTool runApi
INFO: Creating tables...
Oct 24, 2019 1:10:35 PM com.semmle.cloud.commands.UpgradeTool runApi
INFO: Applying initial migrations...
Oct 24, 2019 1:10:35 PM com.semmle.cloud.commands.UpgradeTool patchConfig
INFO: Patching configuration...
Oct 24, 2019 1:10:36 PM com.semmle.cloud.commands.UpgradeTool patchConfig
INFO: Finished patching
Oct 24, 2019 1:12:25 PM com.semmle.cloud.commands.UpgradeTool runApi
INFO: Default distribution updated successfully, archiving previous distributions
...
Oct 24, 2019 1:12:25 PM com.semmle.cloud.model.commands.AnalysisSchedulerCommands
schedule
INFO: 0 jobs in progress. Scheduling up to 1 Jobs.
Oct 24, 2019 1:12:25 PM com.semmle.cloud.model.commands.AnalysisSchedulerCommands
schedule
INFO:  Priority-Attributions:0 Priority-Revision-Builds: 0 Default-Attributions:0
Default-Revision-Builds: 0
[info] Verifying LGTM database integrity...
ADDED TABLES (0):
DELETED TABLES (0):
MODIFIED TABLES (0):
[question] If you want to create a new admin account, please enter the email
address to be used for logging in now. Otherwise, leave blank.
```

4. If you want to create a new admin account, please enter the email address to be used for logging in now. Otherwise, leave blank.

   Enter a valid email address.

Important

You should create an LGTM Enterprise administrator now, to allow you to log in to LGTM when the installation is complete. If you do not do so now you will need to create a user from the command line after installation. For details of how to do this, see the reference topics for the create-user and grant-admin actions in the LGTM Enterprise administrator help.

If required, you can delete this initial administrator account later, replacing it with an externally managed administrator account.

5. `And what should the password be?`

   Enter a password. This is not displayed on the screen. You're asked to confirm this password before the script continues.

The installer continues with the installation, displaying messages about various services being started. By default, the LGTM work pool has two workers.

The script outputs messages similar to those shown below, and ends by prompting you to deploy LGTM Enterprise to any other machines in the LGTM cluster:

```
[info] Starting LGTM...
[info] Starting postgres-lgtm...
[success] Started postgres-lgtm.
[info] Starting minio...
[success] Started minio.
[info] Starting rabbitmq-server-lgtm...
[success] Started rabbitmq-server-lgtm.
[info] Starting solr-lgtm...
[success] Started solr-lgtm.
[info] Starting lgtm-scheduler...
Created symlink from /etc/systemd/system/multi-user.target.wants/lgtm-
scheduler.service to /usr/lib/systemd/system/lgtm-scheduler.service.
[success] Started lgtm-scheduler.
[info] Starting lgtm-jobdealer...
Created symlink from /etc/systemd/system/multi-user.target.wants/lgtm-
jobdealer.service to /usr/lib/systemd/system/lgtm-jobdealer.service.
[success] Started lgtm-jobdealer.
[info] Starting nginx-lgtm...
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx-lgtm.service
to /usr/lib/systemd/system/nginx-lgtm.service.
[success] Started nginx-lgtm.
[info] Starting lgtm-webapp...
Created symlink from /etc/systemd/system/multi-user.target.wants/lgtm-webapp.service
to /usr/lib/systemd/system/lgtm-webapp.service.
```

```
[success] Started lgtm-webapp.
[info] Starting lgtm-worker@1...
Created symlink from /etc/systemd/system/multi-user.target.wants/lgtm-
worker@1.service to /usr/lib/systemd/system/lgtm-worker@.service.
[success] Started lgtm-worker@1.
[info] Starting lgtm-worker@2...
Created symlink from /etc/systemd/system/multi-user.target.wants/lgtm-
worker@2.service to /usr/lib/systemd/system/lgtm-worker@.service.
[success] Started lgtm-worker@2.
[success] All LGTM services on this machine have been started.
[prompt] If you have manually edited the cluster configuration file to introduce
additional machines, complete the installation by copying $HOME/lgtm-releases/lgtm-
<version> to each machine in the cluster and running the
'generated/<hostname>/install-machine.sh' script. Press Enter when complete...
```

If you edited the default cluster configuration to specify other hosts, in addition to the current machine, deploy LGTM Enterprise to these machines following the instructions reported by the script. When you've finished deploying LGTM to all the hosts in the cluster, press Enter.

```
[success] You can now access the LGTM administration interface in your browser by
logging in with the email and password you supplied earlier.
```

You're ready to log in to LGTM Enterprise and start configuring the system—see "Accessing the administration interface" on page 73.

Tip

For information about deploying LGTM to other machines in the cluster, see the following topics in the administrator help:

- Adding and removing workers
- Moving control pool components

Upgrading LGTM (interactive method)

The interactive upgrade script is stored in the `lgtm` subdirectory of the `lgtm-<version>` directory that you extracted earlier, for example: `lgtm-releases/lgtm-<version>/lgtm/upgrade.sh`. Typically, you will need to use the sudo command to run the script—this grants the elevated permissions that are needed to make the required file system changes.

Before proceeding with a interactive upgrade, make sure you have:

- Performed the preparatory tasks described in "Preparing to install or upgrade" on page 16.
- Read the relevant notes for the upgrade you're performing—see "Appendix A—Notes for upgrading LGTM Enterprise" on page 78.

Each LGTM Enterprise release has its own version-specific `upgrade.sh` script. When you're ready to upgrade, run the `upgrade.sh` script for the version you are upgrading to. Run this on the coordinator server, as described below. When you run the script, you will be prompted to perform any required actions on other servers in the LGTM cluster. For information about the servers in an LGTM cluster, see System architecture overview.

The prompts displayed by the upgrader, and the details you enter, are saved to an `upgrade-debug.bin` file, located alongside the `upgrade.sh` script. This log file can be useful for debugging any issues you have completing the upgrade successfully.

Important

- LGTM Enterprise will be unavailable during the upgrade. However, you don't need to take the system down yourself, the upgrade script does this for you.
- If LGTM's `manifest.xml` file was encrypted with a custom password when the system was installed, you need to know the password to upgrade the system (if you've lost the password, see Changing or resetting a manifest password).

Run the upgrade script

1. Open a command console for the coordinator server.

   Note
   The host name of the coordinator server is specified in the `<LGTM releases directory>/state/lgtm-cluster-config.yml` file—where *<LGTM releases directory>*

> is the directory into which the installation package for the current release was unpacked (for example, `$HOME/lgtm-releases/`).

2. Display the version of the currently running LGTM system by entering:

   ```
   sudo lgtm-cli version
   ```

3. Make a note of the version number. You will use this later to confirm that the system was upgraded.

4. Change to the `lgtm-`*`<version>`*`/lgtm` directory, created when you unpacked the distribution file.

5. Run the `upgrade.sh` script in this directory:

   ```
   sudo ./upgrade.sh
   ```

   > **Note**
   >
   > - The script expects to find a cluster configuration file called `lgtm-cluster-config.yml` within a `state` directory located alongside the `lgtm-`*`<version>`* directory—for example, `$HOME/lgtm-releases/state/lgtm-cluster-config.yml`. If the cluster configuration file is located elsewhere, give the path to this file as an argument to the script:
   >
   >   ```
   >   sudo ./upgrade.sh <alternative-path-to>/<cluster-configuration-file>
   >   ```
   >
   > - The script must be run using sudo, or as the root user.

6. Enter the password that was used to encrypt the manifest file when LGTM Enterprise was originally installed.

7. Follow the remaining on-screen prompts.

8. When the upgrade is complete, check the expiration date of your license by entering:

   ```
   sudo lgtm-cli license
   ```

   If there are less than three months left before expiry, we recommend that you make a reminder to upload a new license before this date to avoid disruption to users. If the license is due to expire soon and you have questions about renewing your license, contact GitHub's sales and account management team. New licenses can be uploaded via the administration pages, or using the `license` CLI action, at any time.

Post-upgrade tasks

There are just a few final tasks to complete: see "Post-upgrade procedures" on page 77.

Programmatic deployment

Programmatic deployment is a method of installing or upgrading which, after it has been started, requires no human interaction. It will simply run through the entire process using details that have previously been stored in a script or a configuration management system. This allows you, for example, to run commands that will commission virtual machine instances, install LGTM Enterprise, and start the application in a ready-to-use state.

Installation or upgrade?

The installation and upgrade processes are very similar. The procedures described in the following sections cover both installing LGTM Enterprise from scratch, and upgrading an existing instance of LGTM Enterprise to a newer version. Sections, or individual steps, labeled *For installation only* or *For upgrade only* can be omitted depending on whether you are creating the command sequence for an installation or for an upgrade.

Procedural overview

Deployment consists of the following main stages:

1. *For installation only:* Create and provision virtual machines
2. Upload and unpack the distribution file(s)
3. Design and generate the cluster configuration
4. Install LGTM packages and generated files on each machine in the cluster
5. Initialize/upgrade the LGTM database and start all components

Creating and provisioning machines

For installation only

The "Prerequisites for installation" on page 10 require you to have commissioned the machines on which you are going to install LGTM Enterprise. If you are using a configuration management system however, you could add the creation of virtual machines as the first step in your programmatic installation procedure.

Having commissioned appropriately resourced machines, configure your Ansible playbook (or other configuration management command sequence) to perform the following provisioning tasks.

Check for Java

The interactive installer installs Java automatically if it is not found. However, for a programmatic installation, you need to make sure Java is available on the machines on which you're installing LGTM Enterprise

On each machine that will make up your LGTM cluster, check for Java, and install it if it is not found.

For example, run:

```
java -version
```

If the return value does not show that either OpenJDK or Oracle Java is installed (IBM Java is not supported), you must install it now—for example:

- *On Debian/Ubuntu:*

  ```
  sudo apt-get update && sudo apt-get install openjdk-11-jre-headless
  ```

- *RHEL/CentOS:*

  ```
  sudo yum install java-11-openjdk-headless
  ```

Create a location for temporary LGTM files

On each machine that will make up your LGTM cluster, create a directory to store installation and configuration files used during an installation or upgrade.

In this guide, the commands use an example directory of `lgtm-releases`, at the `$HOME` location, but you can use any directory for this purpose. For example:

```
mkdir $HOME/lgtm-releases
```

> **Note**
> The files in this location are only used during the installation and upgrade processes, or when you deploy changes to your LGTM cluster.

Create a location for version-non-specific files

On the coordinator host machine, create a `state` directory to store files that are not supplied with LGTM, are not version-specific, and relate to the instance of LGTM Enterprise you are installing or upgrading:

```
mkdir $HOME/lgtm-releases/state
```

Note

- The `state` directory is intended as a location in which to store copies of files that you will use in future deployments, such as the cluster configuration file, and the system's manifest file. We recommend that you put the contents of this directory under version control, with the exception of any private key files. You can then checkout the `state` directory for each deployment of this system.
- The location of these files is not prescribed, and you can change it if required. However, locating the files in a directory called `state` is recommended for ease in following the documentation and discussing issues with GitHub Support.

Locate the certificate, license, and integrations files

Do the following on the coordinator host machine:

1. Copy the following files to the `$HOME/lgtm-releases/state` directory:

   - The SSL certificate file for the domain of the URL at which users will access LGTM.

     In the examples given here this file is called `lgtm-web.crt`.

   - The corresponding SSL private key file.

     In the examples given here this file is called `lgtm-web.key`.

   Note

   For more information, see "SSL certificates" on page 11.

2. Copy your license file for LGTM Enterprise to any directory from where it can be loaded into LGTM during installation or upgrade (for example, your home directory on the LGTM coordinator host machine).

3. *Optional—for installation only:* If you have previously saved the integrations settings for an LGTM Enterprise installation, put this file in a secure location that is accessible from the coordinator machine.

> **Important**
> The integrations file contains secrets—for example, the passwords of accounts on third-party systems—so you should take appropriate care to protect this file.

Uploading and unpacking the LGTM distribution file(s)

1. Upload the appropriate distribution file(s) for LGTM Enterprise (`lgtm-<version>-<platform>.tar.gz`) to the `lgtm-releases` directory.

   As of version 1.21 of LGTM Enterprise, there are three distribution files: one for Debian or Ubuntu, one for Red Hat or CentOS, and an MSI file for worker host machines running Windows. Upload the appropriate files for your deployment.

   > **Important**
   > For versions of LGTM Enterprise prior to 1.21 there is just one distribution file: `lgtm-<version>.tar.gz`.

2. *For Red Hat and CentOS only:* If internet access is not available and you want to perform an offline installation or upgrade, upload the supplied `lgtm-third-party-rpms-<yyyy-mm-dd>.tar.gz` file into the same directory.

3. In the command console, change to the `lgtm-releases` directory.

4. Extract the contents of the LGTM distribution package(s).

   - *For Debian and Ubuntu only*

     a. Untar the "deb" package:

     ```
     tar -xvf lgtm-<version>-deb.tar.gz
     ```

     b. If you have (or intend to have) worker host machines running RedHat, untar the "rpm" package:

     ```
     tar -xvf lgtm-<version>-rpm.tar.gz lgtm-<version>/lgtm/lgtm-worker-<version>.noarch.rpm
     ```

c. If you have (or intend to have) worker host machines running Windows, move the `lgtm-worker_<version>.msi` file into the `lgtm-<version>/lgtm` directory.

- *For Red Hat and CentOS only*

a. Untar the "rpm" package:

```
tar -xvf lgtm-<version>-rpm.tar.gz
```

b. If you have (or intend to have) worker host machines running Debian, untar the "deb" package:

```
tar -xvf lgtm-<version>-deb.tar.gz lgtm-<version>/lgtm/lgtm-worker-
<version>_all.deb
```

c. If you have (or intend to have) worker host machines running Windows, move the `lgtm-worker_<version>.msi` file into the `lgtm-<version>/lgtm` directory.

Important
For versions of LGTM Enterprise prior to 1.21, untar the single distribution file:

```
tar -xvf lgtm-<version>.tar.gz
```

The `lgtm-releases` directory now contains an `lgtm-<version>` subdirectory, within which are subdirectories called generated and `lgtm`. If you have (or intend to have) workers running on a different platform to the main part of the system, you've now added the appropriate worker files to the `lgtm` directory.

5. *For Red Hat and CentOS only*: If you're performing an offline installation or upgrade, extract the contents of the dependencies package into the `lgtm-<version>/lgtm` directory:

```
tar -xvf lgtm-third-party-rpms-<yyyy-mm-dd>.tar.gz -C lgtm-<version>/lgtm
```

This adds the directories: database, queue, and web, the presence of which causes LGTM to perform an offline installation.

Deployment procedures

After "Creating and provisioning machines" on page 27 (installation only) and "Uploading and unpacking the LGTM distribution file(s)" on the previous page you can continue the programmatic deployment process. This guide describes procedures for two different cluster topologies. Choose which of these works best for you:

- A simple, **single-machine system** suitable for demo purposes

  Install or upgrade LGTM Enterprise entirely on one machine. We provide an example bash script that you can modify and run to perform a single-machine deployment. If required, you can save the same commands used in the script in a configuration management system—but running a script provides an easy way of performing a simple programmatic deployment.

  See: "Single-machine programmatic deployment" below.

- A **distributed system** where cluster components are hosted on multiple virtual machines

  Typically, a distributed cluster might host the core components (database, file storage, message queue, and search service) on one machine, together with the web application, and use a number of other machines to host worker daemons. Alternatively, you can also host the web application on one or more separate machines. The distribution of cluster components is very flexible. For more information, see the *LGTM Enterprise System Architecture* guide (PDF).

  The instructions for installing or upgrading a distributed cluster assume that you're using a configuration management system such as Ansible to store and replay commands.

  See: "Distributed cluster programmatic deployment" on page 39.

Single-machine programmatic deployment

All of the components of LGTM Enterprise can be run on a single machine. This can be useful to provide a very simple setup for demo purposes but the resulting system will lack the necessary resources to serve as a performant, multi-user system.

If you want to install or upgrade a multi-machine system, see "Distributed cluster programmatic deployment" on page 39.

Generate a cluster configuration file

For installation only

Prior to running a programmatic deployment of LGTM Enterprise for the first time, you need to create a file describing your single-machine "cluster."

1. Use the supplied cluster configuration generation tool to create a simple cluster configuration file.

   From the `lgtm-releases` directory, run:

   ```
   java -jar lgtm-<version>/lgtm/lgtm-config-gen.jar init
   ```

You will now have a file called lgtm-cluster-config.yml in the state directory you created earlier.

2. If you have an SSL certificate—which is highly recommended for a production deployment (see "SSL certificates" on page 11)—open the newly created cluster configuration file in an editor and find the ssl block.

   Within this block, add entries for certificate_path and key_path, specifying the names of the files.

```
ssl:
  hsts: false
  certificate_path: <certificate file>
  key_path: <key file>
```

> **Note**
>
> The paths to these files are relative to the cluster configuration file. If you have followed the instructions given here, this file is located in the lgtm-releases/state directory. This is the same place that you saved the certificate file and the key file, so you can specify their location by just giving the file names, without a path.

Deploying a version of LGTM Enterprise

The following steps cover an initial installation, a redeployment, or an upgrade to a new version of LGTM Enterprise. A bash script containing these steps is available, as a worked example, on GitHub.com.

If you have followed the previous sections of this guide, the machine on which you are deploying LGTM already contains the following, within a directory such as $HOME/lgtm-releases:

```
.
├── lgtm-<version>
├── lgtm-<version>-<platform>.tar.gz
├── [lgtm-<version>-<platform>.tar.gz]
├── [lgtm-third-party-rpms-<yyyy-mm-dd>.tar.gz]
└── state
    ├── lgtm-cluster-config.yml
    └── manifest.xml
```

1. Run the following command to generate the necessary files for your configuration of LGTM Enterprise:

   ```
   LGTM_CREDENTIALS_PASSWORD=<manifest-password> java -jar lgtm-
   <version>/lgtm/lgtm-config-gen.jar generate --overwrite
   ```

   Note
   If you are upgrading to a version of LGTM Enterprise older than 1.21.0 you must add `--input state/lgtm-cluster-config.yml --output lgtm-<version>/generated` after generate in the above command. The paths are relative to the working directory (in this case `$HOME/lgtm-releases`).

   The files are generated in an `lgtm-<version>/generated` directory. These include a script called `install-machine.sh` which you will use shortly.

   Important
   The manifest password defined here is used to protect secrets normally used for inter-machine cluster communication. If you are performing an initial installation, make sure to retain this password as you will need it when you upgrade LGTM, or when you deploy changes you've made to the cluster configuration.

2. If LGTM is already installed, stop it by running:

   ```
   sudo lgtm-down
   ```

3. The `lgtm-config-gen.jar` tool created various files in the `lgtm-<version>/generated` directory. Included in these files is a script called `install-machine.sh`. Run this now to install the necessary packages for LGTM Enterprise and to copy the files that were generated for your configuration of LGTM into the correct place on the local machine:

   ```
   LGTM_DONT_START=true DEBIAN_FRONTEND=noninteractive sudo --preserve-env lgtm-
   <version>/generated/localhost/install-machine.sh
   ```

   Note
   The `DEBIAN_FRONTEND` environment variable is only required for Debian and Ubuntu. It has no effect in other Linux distributions, so can be left in or omitted.

4. Start the core services that LGTM depends on, so that the database is available:

```
sudo lgtm-up --core-only
```

> **Note**
>
> *For a pre-1.20 upgrade only:* If you are upgrading to a version before 1.20.0 (for example, from 1.18.*x* to 1.19.*x*), replace the sudo `lgtm-up --core-only` command with these three commands:
>
> ```
> sudo systemctl start postgres-lgtm
> ```
>
> ```
> sudo systemctl start minio
> ```
>
> ```
> sudo systemctl start rabbitmq-server-lgtm
> ```

5. Run the following sequence of commands to create the database schema and perform any necessary upgrades:

> **Note**
>
> This sequence of commands is appropriate for any type of deployment (an installation, an upgrade, or a deployment of cluster changes). In all cases, run *all* of the following commands.

a. Create the schema (if necessary):

```
sudo lgtm-upgrade --action CREATE --if-not-exists --config
/etc/lgtm/config.json
```

> **Note**
>
> `config.json` is one of the files generated for your LGTM configuration when you run `lgtm-config-gen.jar`. It is copied to the `/etc/lgtm` directory when you run the `install-machine.sh` script.

b. Ensure that an existing database has an up-to-date schema:

```
sudo lgtm-upgrade --action FULL --schema-only
```

c. Initialize the system and CodeQL analysis engine:

```
sudo lgtm-upgrade --action INITIALIZE
```

Note

If you are upgrading to a version of LGTM Enterprise older than 1.21.0 you must add a `--core` flag pointing to the "odasa" distribution zip file. For example:

```
sudo lgtm-upgrade --action INITIALIZE --core lgtm-<version>/lgtm/odasa-*.zip
```

d. Validate the database schema:

```
sudo lgtm-upgrade --action VALIDATE
```

e. Run any necessary data migrations:

```
sudo lgtm-upgrade --action FULL
```

Important

For an upgrade, the data migrations carried out by the `FULL` action of the `lgtm-upgrade` tool may take time proportional to the size of your instance of LGTM. So, for a very large instance of LGTM, this may cause an undesirable amount of downtime. You can, in this situation, start LGTM before carrying out the data migrations. To do this, run sudo `lgtm-up` immediately prior to running the `FULL` action.

However, it's very important that these migrations are performed as you won't be able to upgrade a future version of LGTM Enterprise (other than a maintenance release) without having run them.

f. Check that the database is now fully up to date:

```
sudo lgtm-upgrade --action CHECK
```

6. Rename one of the files that was generated for your configuration, as follows:

```
sudo mv /etc/lgtm/config.json /etc/lgtm/config.initial.json
```

Note

The `config.json` file is never reused. Changing its name:

a. Preserves a copy of the initial system configuration for reference.

b. Prevents you from accidentally overwriting a current system configuration (which may contain changes made in the administration interface) with the initial configuration, if you were to run the `lgtm-upgrade` tool's `CREATE` action at a later date using the original JSON file.

Programmatically adding integrations

Optional

Note

Generally this step only applies when you are installing LGTM Enterprise, or where you have tested integrations with third-party systems on a test instance of LGTM and you want to deploy this configuration to a production instance.

If you have previously created an integrations file (see "Including integrations in a programmatic installation" on page 50), you can use this now to add preconfigured integrations to the new instance of LGTM Enterprise.

Run this command:

```
sudo lgtm-cli patch-config <path-to-integrations-file>
```

Important

The integrations file contains secrets—for example, the passwords of accounts on third-party systems—so you should take appropriate care to protect this file.

Start LGTM

Run this command:

```
sudo lgtm-up
```

Install the license

Before using a fresh installation, install the LGTM license that you uploaded to the LGTM host machine as one of the provisioning tasks:

```
sudo lgtm-cli license --install <path-to-license-file>
```

Note
For an upgrade, you can install a new license from the Settings page of LGTM's administration interface.

Place reusable files under version control

Optional, but recommended

If you have not already done so, we recommend that you put the contents of the `state` directory under version control, with the exception of any private key files.

The `state` directory contains files that describe your LGTM instance. You will use these files again when you upgrade, or if you want to install another instance of LGTM Enterprise with a similar configuration. Placing this directory under version control means that you can easily check it out onto the appropriate server in future when you do an installation or upgrade.

Create a user account and log in

For installation only

Create an administrator account:

1. `sudo lgtm-cli create-user --email admin@example.com --password <password> --display-name Administrator`

2. `sudo lgtm-cli grant-admin --name admin@example.com`

You can now browse to your new LGTM Enterprise instance, log in using the account you just created, and start configuring LGTM Enterprise—see "Accessing the administration interface" on page 73.

Important
If required, you can delete this initial administrator account later, replacing it with an externally managed administrator account.

Distributed cluster programmatic deployment

Introduction

This section describes how to set up programmatic installation or upgrade for an LGTM Enterprise cluster that is distributed between multiple machines.

For details of the various components of an LGTM Enterprise cluster, see the LGTM Enterprise System Architecture PDF, or the "System architecture overview," "Control pool," and "Work pool" topics in the administrator help.

For the purposes of explaining how this can be done we're going to assume the following cluster topology, where the system is distributed between five virtual machines:

- 1 coordinator VM

  Hosting these cluster components:

  - Coordinator
  - Database
  - File store
  - Message queue
  - Search service

- 2 web host VMs*
- 2 worker host VMs (each running 3 workers)

* Multiple web hosts assumes that the URL used to access LGTM Enterprise is served by two load-balanced web servers. The means by which you implement load balancing is beyond the scope of this documentation.

The system could be further distributed by putting some or all of the cluster components on separate machines. However, this is not usually necessary.

Resources on GitHub

If you are using Ansible to set up programmatic installation or upgrade, you may find it useful—as an alternative to creating a playbook from scratch—to download the following example of an Ansible playbook:

https://github.com/Semmle/lgtm-ansible-example

The instructions in the remainder of this section explain the commands used in a programmatic deployment and will help you to create the stored procedure you need for carrying out a programmatic deployment in any configuration management system.

Alternatively, if you prefer not to use a configuration management system, you may find it useful to download the following example bash script and accompanying the README file:

https://github.com/Semmle/lgtm-bash-deployment/tree/master/multi-machine

Note

The `deploy-multi.sh` file is provided purely as an example of a script that you could use to perform a distributed deployment.

For details of how to use this script to programmatically install or upgrade a distributed LGTM cluster see the README file that accompanies the script on GitHub.

Prerequisites

Your configuration management system playbook (or similar command sequence) should already include commands to create and provision the required virtual machines—as detailed on pages 27—29.

Before proceeding, make sure:

- *For installation only:* The "Prerequisites for installation" on page 10 have all been met.
- *For upgrade only:* You have read the relevant notes for the upgrade you're performing—see "Appendix A—Notes for upgrading LGTM Enterprise" on page 78.

You can now continue with the main deployment stages we outlined earlier.

Design and generate the cluster configuration

In this stage of the process you will:

- Create a cluster configuration file (if you are installing and don't already have one)
- Use the cluster configuration file to generate the files you need to deploy the system

Create the cluster configuration file

Installation only

The cluster topology of an LGTM Enterprise instance is defined in the cluster configuration file: `lgtm-cluster-config.yml`. Your first task is to create this file.

Important

Given that the cluster configuration file is an input to the programmatic installation process, it should be stored under version control so that it can easily be reused. Typically, the contents of the `state` directory, which you created when provisioning machines for LGTM, are version controlled (with the exception of any private key files that you saved there).

Create a file called `lgtm-cluster-config.yml` in the `state` directory (for example, `$HOME/lgtm-releases/state`) with the contents shown below, replacing the placeholder values shown in italics in the example below with real values.

Alternatively, to create multiple clusters with the same topology, you could create a templated version of `lgtm-cluster-config.yml` (saving the template file under version control) and use commands stored in your configuration management system to produce the actual `lgtm-cluster-config.yml` file, inserting appropriate hostnames for the different cluster components. This would streamline the process of deploying multiple clusters programmatically (for example, a development cluster, a UAT cluster, and a production cluster).

Note

- You can download, or copy and paste, the following file from https://help.semmle.com/lgtm-enterprise/admin/help/installation/scripts/lgtm-cluster-config.yml.TEMPLATE. Copying and pasting directly from this PDF is not recommended as this may change characters and spacing.

- The placeholder values shown below as `<...-`*`hostname`*`>` should all be replaced by the hostname by which the machine can be accessed by other machines in the cluster. The placeholder value *`<LGTM-website-URL>`* must include a URL scheme and a hostname (or IP address)—for example, `https://lgtm.internal.example.com`. This is the URL at which users will access LGTM Enterprise.

- Environment variable expansion within values in the cluster configuration file is only supported for Windows worker host machines. For more information, see "Using environment variables in the cluster configuration file" in the administrator help.

```
version: 8
coordinator:
  hostname: "<coordinator-hostname>"
database:
  hostname: "<coordinator-hostname>"
file_storage:
  hostname: "<coordinator-hostname>"
```

```
queue:
  hostname: "<coordinator-hostname>"
search:
  hostname: "<coordinator-hostname>"
task_workers:
  hosts:
  - hostname: "<coordinator-hostname>"
web:
  hosts:
  - hostname: "<webserver-1-hostname>"
  - hostname: "<webserver-2-hostname>"
  ssl:
    certificate_path: "lgtm-web.crt"
    key_path: "lgtm-web.key"
workers:
  hosts:
  - hostname: "<workerhost-1-hostname>"
    specs:
    - type: "GENERAL"
      copies: 1
    - type: "ON_DEMAND"
      copies: 1
    - type: "QUERY"
      copies: 1
    labels: []
  - hostname: "<workerhost-2-hostname>"
    specs:
    - type: "GENERAL"
      copies: 1
    labels: []
  temp_path: "/var/cache/lgtm/workers/"
data_path: "/var/lib/lgtm/"
```

Important

- The entries under `ssl:` in the configuration shown above assume that you have already generated SSL certificate and key files for the domain of the URL at which users will access LGTM, and you have saved these files on the coordinator host machine at the location suggested in the preliminary task "Locate the certificate, license, and integrations files" on page 29.

  If required, you can use a self-signed certificate that LGTM generates, rather than using your own. To do this, remove the entries under `ssl:` in the configuration file. This is not recommended.

  For more information, see "SSL certificates" on page 11.

- Make sure that the lines are indented as shown above.

  You can check that the contents of the file are valid YAML by using a validation tool, such as `yamllint`. For more information, see yamllint.readthedocs.io.

Generate the LGTM configuration

The following series of commands use the cluster configuration file to generate the files that are required to install or upgrade the LGTM Enterprise cluster that the configuration file defines.

In your configuration management system playbook (or similar command sequence) add the following commands:

1. Run the following command on the coordinator machine. This command provides the password used to encrypt and decrypt information in LGTM's manifest file, and generates the files for your configuration of LGTM Enterprise:

   ```
   LGTM_CREDENTIALS_PASSWORD=<manifest-password> \
     java -jar <version-directory>/lgtm/lgtm-config-gen.jar \
       generate --overwrite
   ```

   For example:

   ```
   LGTM_CREDENTIALS_PASSWORD=<password> \
     java -jar $HOME/lgtm-releases/lgtm-<version>/lgtm/lgtm-config-gen.jar \
       generate --overwrite
   ```

   Important
   If you are copying the above command from this PDF, make sure that the hyphens shown above are retained when you paste the copied text.

   Note
   - The password supplied in this command is used to encrypt and decrypt information in LGTM's manifest file.

     If you are installing LGTM Enterprise, make sure you retain the manifest password in a secure location. You will need it again when you upgrade LGTM, or when you deploy changes you've made to the cluster configuration file.

> You may want to save the password to a vault and include commands in your Ansible playbook (or similar) to extract the password for use in this step of the installation process.
>
> - If you are upgrading to a version of LGTM Enterprise older than 1.21.0 you must add `--input state/lgtm-cluster-config.yml --output lgtm-<version>/generated` after generate in the above command. The paths are relative to the working directory (in this case `$HOME/lgtm-releases`).

Optional: snapshot the VMs

Upgrade only

Before proceeding with the upgrade, this is a good point to take a snapshot of each of the VMs in your LGTM cluster. This will allow you to redo the upgrade from this point if any of the commands you run afterward contain errors.

Install LGTM files on each machine in the cluster

The next main deployment stage (for upgrading as well as for installation) is to copy the version-specific release directory to each of the other machines in the cluster, and run a script to install the required software on that machine.

Add commands to your configuration management system playbook (or similar command sequence) to iterate through *each non-coordinator machine in the cluster* (for example, the worker host and web host machines) carrying out each of the following steps:

1. Copy the `lgtm-<version>` directory from the coordinator host machine to one of the other machines in the cluster (for example, a web application host machine or a worker host machine).

   > **Note**
   > The subdirectories of `lgtm-<version>/generated` that have the hostnames of machines other than the machine you are copying to can be omitted.

   For example, after adding a private key for the current user on the local machine (the coordinator) and the corresponding public key on the remote machine, you could use rsync to copy the directory. The following rsync command—run from the `$HOME/lgtm-releases/lgtm-<version>` directory on the coordinator machine—copies the `lgtm-<version>` directory to a remote machine, excluding the directories that relate to other cluster host machines. This ensures that unneeded files are not copied to the remote machine. The command assumes that you have created a `$HOME/lgtm-releases` directory on the remote machine, as per the instructions on page 28.

```
rsync --recursive --archive --compress --progress \
 --include "generated/<remote-machine-hostname>/" \
 --exclude "generated/*/" . \
 <user>@<remote-machine-hostname>:$HOME/lgtm-releases/lgtm-<version>
```

2. *For upgrade only:* Stop LGTM by running the following command:

   - Linux machines:

     ```
     sudo lgtm-down
     ```

   - Windows worker host machines (as an administrator):

     ```
     %PROGRAMDATA%\LGTM\lgtm-worker\bin\lgtm-down.bat
     ```

   > **Note**
   > The order in which you bring down LGTM on the various machines is not significant.

3. *Linux only:* Set the `LGTM_DONT_START` environment variable to `true`, to avoid LGTM Enterprise attempting to start up when you run the next command:

   ```
   export LGTM_DONT_START=true
   ```

4. The `lgtm-<version>/generated` directory contains a subdirectory with the name of the current host. We now need to run an installation script located within this host-specific directory.

   Do the following on each non-coordinator machine in the cluster.

   - For Linux machines:

     ```
     sudo -E <version-directory>/generated/<local-hostname>/install-machine.sh
     ```

   - For Windows worker host machines:

     a. Copy the `lgtm-worker_<version>.msi` file from the `$HOME/lgtm-releases/lgtm-<version>/lgtm` directory to the `$HOME/lgtm-releases/lgtm-<version>/generated/<hostname>` directory:

```
cp <version-directory>/lgtm/lgtm-worker_*.msi <version-
directory>/generated/<local-hostname>/
```

b. Retrieve the logon details (domain\username and password) for the user that will be used to run worker daemons on this machine—for example, extract these from a vault in preparation for the next step.

> **Important**
>
> This user account must have full access to the working directory specified for this machine in the cluster configuration (`temp_path` property). The account details are passed directly to `sc.exe` and used to set up services for LGTM.

c. Run the `install-machine.bat` script as the user account whose logon details you retrieved, using the syntax:

```
install-machine.bat <domain>\<username><logon-password>
```

For example:

```
<version-directory>/generated/<local-hostname>/install-machine.bat
.\lgtmuser <lgtmuser-password>
```

Once this is done, do the following on the coordinator machine:

1. *For upgrading only:* `sudo lgtm-down`

2. `export LGTM_DONT_START=true`

3. `sudo -E <version-directory>/generated/<local-hostname>/install-machine.sh`

Initialize the LGTM database and start all components

The final stage of the deployment involves starting the database, message queue, and file storage components, initializing or upgrading the database, and then bringing up the system.

Add commands to your configuration management system playbook (or similar command sequence) to perform the following actions on the appropriate machines:

1. Bring up the core services for LGTM by running:

   ```
   sudo lgtm-up --core-only
   ```

   Run this command on machines that host any of these cluster components:

- LGTM database
- File storage service
- Message queue

In our example, these components are all hosted on the same machine as the coordinator, so this command only needs to be run on the coordinator machine.

> **Note**
>
> *For a pre-1.20 upgrade only:* If you are upgrading to a version before 1.20.0 (for example, from 1.18.*x* to 1.19.*x*), replace the sudo `lgtm-up --core-only` command with these three commands:
>
> ```
> sudo systemctl start postgres-lgtm
>
> sudo systemctl start minio
>
> sudo systemctl start rabbitmq-server-lgtm
> ```

2. On the coordinator machine, initialize or upgrade the LGTM database by running the following commands.

> **Note**
>
> Run the following sequence of commands whether you are performing an installation or an upgrade.

a.
```
sudo lgtm-upgrade --action CREATE \

  --if-not-exists --config /etc/lgtm/config.json
```

> **Note**
>
> `config.json` is one of the files generated for your LGTM configuration when you run `lgtm-config-gen.jar`. It is copied to the `/etc/lgtm` directory when you run the `install-machine.sh` script.

b. `sudo lgtm-upgrade --action FULL --schema-only`

c. `sudo lgtm-upgrade --action INITIALIZE`

> **Note**
>
> If you are upgrading to a version of LGTM Enterprise older than 1.21.0 you must add a `--core` flag pointing to the "odasa" distribution zip file. For example:
>
> ```
> sudo lgtm-upgrade --action INITIALIZE --core lgtm-
> <version>/lgtm/odasa-*.zip
> ```

d. `sudo lgtm-upgrade --action VALIDATE`

e. `sudo lgtm-upgrade --action FULL`

f. `sudo lgtm-upgrade --action CHECK`

3. On the coordinator machine, rename the system configuration file as follows:

   ```
   sudo mv /etc/lgtm/config.json /etc/lgtm/config.initial.json
   ```

   > **Note**
   >
   > The `config.json` file is never reused. Changing its name:
   >
   > a. Preserves a copy of the initial system configuration for reference.
   >
   > b. Prevents you from accidentally overwriting a current system configuration (which may contain changes made in the administration interface) with the initial configuration, if you were to run the `lgtm-upgrade` tool's `CREATE` action at a later date using the original JSON file.

4. On each machine in the cluster, bring up LGTM Enterprise by running the following command :

   - Linux machines:

     ```
     sudo lgtm-up
     ```

   - Windows worker host machines (as an administrator):

     ```
     %PROGRAMDATA%\LGTM\lgtm-worker\bin\lgtm-up.bat
     ```

> **Note**
>
> The order in which you bring up LGTM on the various machines is not significant.

5. *For installation only:* On the coordinator machine, create a user account that you will use to log on to LGTM Enterprise and grant the account administrator privileges so that it has access to the administration interface:

   ```
   sudo lgtm-cli create-user --email <admin-user-email-address> \

     --password <admin-user-email-address> \

     --display-name Administrator

   sudo lgtm-cli grant-admin --name <admin-user-email-address>
   ```

   > **Note**
   >
   > The user can change the password specified here (their email address) and the display name ("Administrator") after logging in.
   >
   > If required, you can delete this initial administrator account later, replacing it with an externally managed administrator account.

6. On the coordinator machine, install your LGTM Enterprise license:

   ```
   sudo lgtm-cli license --install <version-directory>/lgtm/license.dat
   ```

7. *Optional—for installation only:* If you have previously created an integrations file (see "Including integrations in a programmatic installation" on the next page), use this now to add preconfigured integrations to the new instance of LGTM Enterprise.

   On the coordinator machine, run:

   ```
   sudo lgtm-cli patch-config <path-to-integrations-file>
   ```

   > **Important**
   >
   > The integrations file contains secrets—for example, the passwords of accounts on third-party systems—so you should take appropriate care to protect this file.

If you are performing an installation, you can now go to the administration interface and start configuring LGTM Enterprise—see "Accessing the administration interface" on page 73.

If you are upgrading LGTM Enterprise, check the relevant After you upgrade to... section of the appendix, starting on page 78, for any additional steps you need to take to configure the new version.

Including integrations in a programmatic installation

LGTM "integrations" specify the integration between LGTM and third-party systems. These integrations enable LGTM to access authentication and authorization data for users. They also allow two-way communication with repositories and issue tracking systems. For more information about LGTM integrations, see the topic on "Defining integrations" in the administrator help.

Typically, you will want a programmatic installation to add integrations to the resulting system, so that you don't have to add them manually after installation, each time you install LGTM Enterprise.

Before you can include integrations in a programmatic installation, you need to have first configured the integrations in the administration interface of an existing instance of LGTM Enterprise.

Including integrations in the commands for a programmatic installation therefore involves:

1. Installing LGTM Enterprise—for example, using the interactive installation method.
2. Adding integrations as required (instructions are provided in the administrator help).
3. Saving the integration settings as a JSON file (see below).
4. Passing the integrations JSON file to the `lgtm-cli patch-config` command at the end of a programmatic installation.

Saving the integration settings

If you have already configured integrations, in an existing LGTM Enterprise instance, you can capture the settings as follows:

1. Working in a console window, on the LGTM coordinator machine, change to your user's home directory.

> **Tip**
> If the command-line JSON processor `jq` is installed on the host machine, you can use the following commands as a shortcut, instead of steps 2—5 below. Run these commands and then skip to step 6:

```
sudo lgtm-cli get-config > running-config.json
jq "{ access: .access }" running-config.json > integrations.json
rm running-config.json
```

2. Use the `get-config` action of the `lgtm-cli` command to output the current running configuration to a file:

   ```
   sudo lgtm-cli get-config > integrations.json
   ```

   This creates a file called `integrations.json` in the current directory containing details of the running configuration.

   > ### Terminology
   >
   > **Running configuration**: data within LGTM's internal database that describes the currently running system. Not to be confused with the cluster configuration file, which defines the topology of an LGTM Enterprise cluster.

3. Open the file for editing.

   The JSON data in this file consists of a pair of braces: `{ ... }` which enclose a set of properties, many of which contain nested properties.

4. Leaving the outer braces in place, delete all top-level properties apart from `"access"`, which describes your existing integrations.

   The file should now comprise:

   ```
   {
       "access" : {
           <details of various types of integrations>
       }
   }
   ```

5. *Optional:* Check that the file is valid JSON by using a tool such as `jq`.

Running `jq . <path-to-file>` will display the file's contents if it's valid JSON, and if not will display information about errors in the file. For more information, see stedolan.github.io/jq/manual.

6. Move the `integrations.json` file to a secure location for use when you perform a programmatic installation.

> **Important**
> The integrations file contains secrets—for example, the passwords of accounts on third-party systems—so you should take appropriate care to protect this file.

Adding the integration settings to the programmatic installation

Now that you have saved the integration settings to a file you can use the `lgtm-cli patch-config` command to load these settings, as the final step in a programmatic installation (see the instructions on page 37 for a single-machine programmatic deployment, or page 49 for a distributed cluster programmatic deployment).

AWS deployment

Installing LGTM on AWS

Deploying on AWS involves launching an EC2 instance that already contains LGTM Enterprise. The entire LGTM system (including workers) runs within a single EC2 instance. You can specify some basic system properties (such as the number of workers) before you launch the EC2 instance, and then fine-tune the configuration within the LGTM Enterprise administration interface.

Preparation

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. In the navigation pane, under **Network & Security**, click **Key Pairs**.

3. On the Key Pairs page, click either **Import key pair** (to upload an existing public key file, or paste in the text of a public key), or **Create key pair** (to create a new key pair and download the private key).

4. Give the key pair an appropriate name and complete the form to import or create the key pair.

5. In the navigation pane, under **Network & Security**, click **Security Groups**.

6. Click **Create Security Group**.

7. Give the security group a name and description of your choice.

8. Click **Add Rule** and add appropriate rules.

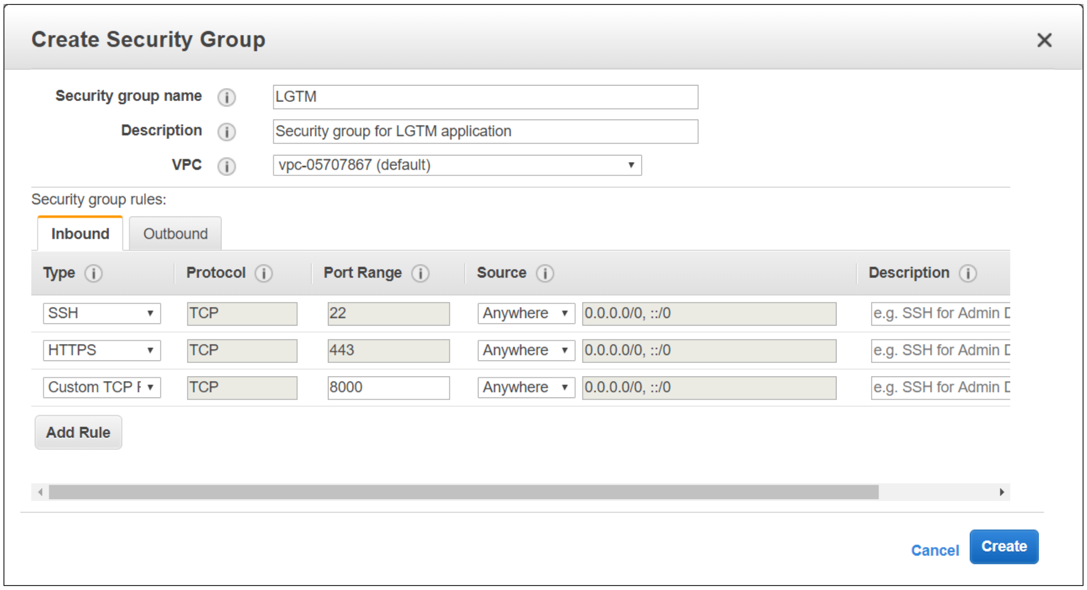
   Suggested rules for a publicly-accessible LGTM instance:

   a. **Inbound** (see Authorizing Inbound Traffic for Your Linux Instances)

      i. **Type**: `SSH`, **Source**: `Anywhere`

      ii. **Type**: `HTTPS`, **Source**: `Anywhere`

iii. **Type**: `Custom TCP`, **Port**: `8000`, **Source**: `Anywhere`



Create Security Group

Security group name	LGTM
Description	Security group for LGTM application
VPC	vpc-05707867 (default)

Security group rules:

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin D
HTTPS	TCP	443	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin D
Custom TCP F	TCP	8000	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin D

Caution

The above suggested rules leave LGTM Enterprise open for inbound traffic. If users will be connecting from a particular network you should configure that here in accordance with your company policies. For more information on system security, see the topic "Securing LGTM Enterprise" in the administrator help.

b. **Outbound**

(The default rule) **Type**: `All traffic`, **Protocol**: `All`, **Port Range**: `All (0-65535)`, **Destination**: `Custom`, `0.0.0.0/0`

9. After you've created the security group, edit it and add an Inbound rule that allows traffic from instances using the same security group:

   Type: `All traffic`, **Protocol**: `All`, **Port Range**: `All (0-65535)`, **Source**: `Custom`, *`<this security group*>`*

   * Security group IDs to choose from are suggested as you start typing the group name.

Launch an EC2 instance containing LGTM Enterprise

1. Go to https://github.com/Semmle/lgtm-enterprise/releases, scroll down to the "Amazon Web Services images" section for the current release and copy the ID of the Amazon Machine Image (AMI) for your region.

2. In the Amazon EC2 console (https://console.aws.amazon.com/ec2/), click **Services** > **EC2**.

3. In the navigation pane, click **Instances** > **Launch Instance** > **Community AMIs**.

4. Complete the sequence of steps in AWS to configure a new EC2 instance:

Step 1: Choose AMI

a. In the search box, paste the AMI ID you copied in a previous step and press Enter.

b. Select the required AMI from the search results.

Step 2: Choose Instance Type

a. Scroll down the list to the "Memory optimized" instance types and select an appropriate type. For example:

r5.xlarge (4 vCPUs, 32 GiB memory, EBS only)

This is the minimum specification. A production instance will need more resources. For help on choosing an appropriate instance type, see "Appendix B—Amazon EC2 instance types " on page 82.

> **Important**
> Make sure to choose an instance type with EBS storage only, not including a local NVMe-based SSD (usually indicated by a "d" in the instance type name). For example, you cannot use **r5d.xlarge**.

b. Click **Next: Configure Instance Details**.

Step 3: Configure Instance

a. Leave all of the default settings in the first part of the form (unless you specifically want to change these), and scroll down to the Advanced Details section.

b. Leave the **User data** radio buttons set to `As text`.

c. In the **User data** text field, enter the following configuration details using YAML syntax (changing the values appropriately):

```
admin-email: <email-address>
admin-password: '<password>'
```

To create an LGTM administrator account—to allow you to log in—you must specify both `admin-email` and `admin-password`. You should use a temporary password here and then change it in the administration interface when you first log in to LGTM.

Additional optional properties:

- `manifest-password`—if you don't specify a manifest password, one will be randomly generated and stored in `/data/lgtm-releases/.manifest-password`.
- `n-general`—number of general worker daemons (default: `1`)
- `n-query`—number of query worker daemons (default: `1`)
- `n-on-demand`—number of on-demand worker daemons (default: `0`)
- `environment`—you can set environment variables for workers using the same environment property as in an `lgtm-cluster-config.yml` file. See the topic "Setting environment variables for workers" in the administrator help.

  For example:

  ```
  environment:
    ENV_VAR1: "value-of-ENV_VAR1"
    ENV_VAR2: "value-of-ENV_VAR2"
  ```

Example entry in the **User data** text field:

```
admin-email: johndoe@example.com
admin-password: 'iRxh33&%j89!'
manifest-password: 'Wlo8^xj-w22k'
n-general: 3
environment:
  SOMEVAR: "some-value-for-all-workers-to-use"
  ANOTHERVAR: "another-value-for-all-workers-to-use"
```

> **Note**
> You can easily change these settings later, if required, by stopping the EC2 instance, editing this field, and restarting the instance.

d. Click **Next: Add Storage**.

Step 4: Add Storage

a. Set the **Root** volume to `50` GiB, and select **Delete on Termination**.

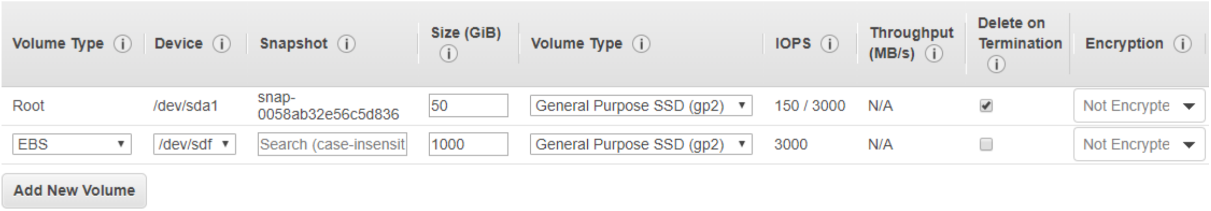
b. Click **Add New Volume**.

c. Configure the new volume:

- **Volume Type** (first column): EBS
- **Device**: `/dev/sdf`

  Make sure to change this from the default setting.
- **Size (GiB)**: `1000`
- **Delete on Termination**: not selected

  This volume is where analysis data will be stored. It is therefore important that this check box is *not* selected.
- Leave the other settings at their default values.



Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0058ab32e56c5d836	50	General Purpose SSD (gp2)	150 / 3000	N/A	☑	Not Encrypte
EBS	/dev/sdf	Search (case-insensit	1000	General Purpose SSD (gp2)	3000	N/A	□	Not Encrypte

d. Click **Next: Add Tags**.

Step 5: Add Tags

a. Click **Add Tag**.

b. **Key**: `Name`

c. **Value**: `LGTM` (or an alternative name of your choice)

   You'll use this name later in the process to find this instance.

d. Select **Instances** and **Volumes**.

e. Click **Next: Configure Security Group**.

Step 6: Configure Security Group

a. Click **Select an existing security group**.

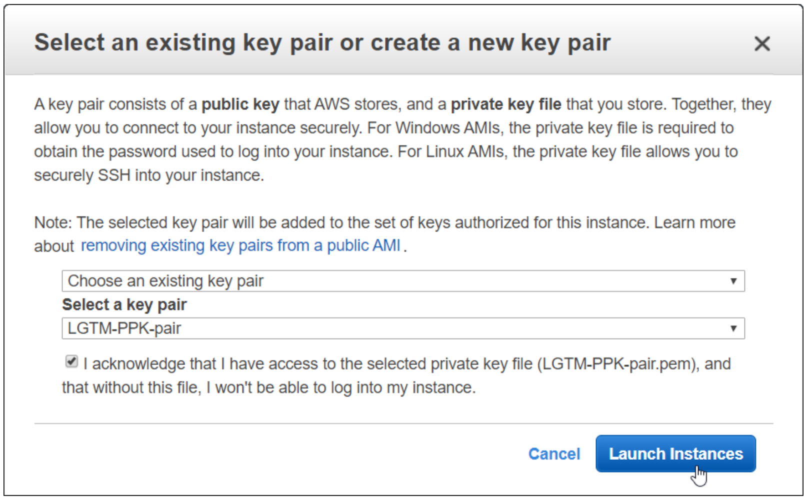
b. Select the security group you created in the Preparation section.

c. Click **Review and Launch**.

Step 7: Review Instance Launch

Review the details, then click **Launch**.

5. In the dialog box that's displayed, select the key pair you created or imported in the Preparation section.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Choose an existing key pair

Select a key pair

LGTM-PPK-pair

☑ I acknowledge that I have access to the selected private key file (LGTM-PPK-pair.pem), and that without this file, I won't be able to log into my instance.

6. Click **Launch Instances**.

7. On the Launch Status page, click **View Instances**.

8. Wait for the **Instance State** to change to "running."

9. In the navigation pane, under **Network & Security**, click **Elastic IPs**.

10. Click **Allocate Elastic IP** address.

11. On the Allocate Elastic IP address page choose whether you want the IP address to come from Amazon's pool of addresses or your own, and click **Allocate**.

12. Click the IP address you just added, to edit its properties.

13. Click **Associate Elastic IP address**.

14. On the Associate Elastic IP address page, click in the **Instance** field and choose "LGTM" (or your alternative name if you chose to use a different value for the Name tag).

15. Click **Associate**.

16. *Optional:* Click **Manage tags**, add a `Name` tag with an appropriate value and click **Save**.

The new LGTM Enterprise instance will probably take about 10 minutes to be initialized.

While the instance is initializing, you can view its output in a web browser at port 8000 of the instance's external hostname (or the elastic IP address)—that is, http://<*external-hostname-or-address*>:8000/. (Note: HTTP not HTTPS.) Refresh this page to see new log entries. If an error occurs, this web page will remain available. The web page becomes unavailable when LGTM has successfully initialized itself.

After LGTM has initialized itself, browse to https://*<external-hostname-or-address>*/admin. (Note: HTTPS.)

Log in using the account name and password you entered in the **User data** field of the instance properties.

At this point LGTM Enterprise is using a self-signed certificate, so your browser will display a security warning. There should be an advanced option you can choose to allow you to proceed to the site. For more information, and some next steps, see "Post-installation procedures" on page 73.

Basic details about the LGTM instance

- EC2 instances created from the LGTM AMI run Ubuntu 18.04 LTS.
- You can SSH into the instance as the user `lgtm-admin`.

  Note
  You can find connection information in AWS by selecting the EC2 instance and clicking **Actions** > **Connect**. However, note that the example connection string that's given in the Connect To Your Instance dialog box uses the root user. You must change this to `lgtm-admin`.

  Important
  The LGTM administrative shell is provided for troubleshooting and performing documented operations procedures only. Modifying system and application files, running programs, or installing unsupported software packages may break your installation.

- Data for the LGTM Enterprise instance is located at `/data`.
- The `/data/lgtm-releases` directory is owned by the user `lgtm-admin`.
- To view the output of the initialization process, use:

  ```
  sudo journalctl --unit lgtm-bootstrap.service
  ```

- If you need to change the number of workers, or environment variable values, you can stop the EC2 instance, change the user data (see "Step 3: Configure Instance" on page 55), and then start the instance again.

- If you need to change the EC2 instance type, perform an upgrade, using the same AMI as before, but choosing the required instance type, as described below.

Upgrading LGTM on AWS

This section describes how to upgrade an instance of LGTM Enterprise running on Amazon Web Services.

You can use the following procedure to:

- Upgrade the Amazon EC2 instance type on which LGTM is running.
- Upgrade the version of LGTM.

Important
Before upgrading to a new version of LGTM, check the "Appendix A—Notes for upgrading LGTM Enterprise" on page 78 for any pre-upgrade steps you need to take.

These steps provision a new EC2 instance, take a snapshot of the old EC2 instance's data volume, and mount the snapshot on the new EC2 instance.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Stop the old AWS instance:
   a. In the navigation pane, click **Instances**.
   b. Select the EC2 instance for LGTM Enterprise.
   c. Click **Actions** > **Instance State** > **Stop**.
3. Take a snapshot of the data volume:
   a. In the navigation pane, under **Elastic Block Store**, click **Volumes**.
   b. Select the large volume that serves as the persistent data store for LGTM.

Important
Make sure to select the data volume and not the volume that contains LGTM Enterprise itself.

c. Click **Actions** > **Create Snapshot**.

d. Enter a **Description**.

   For example, "LGTM Enterprise persistent data volume."

e. Click **Add Tag**.

f. **Key**: `Name`

g. **Value**: *`<give the snapshot a name>`*

   For example, "<*your name*> LGTM snapshot."

h. Click **Create Snapshot**.

i. On the "Create Snapshot Request Succeeded" message, copy the snapshot ID, then click **Close**.

4. To upgrade the version of LGTM Enterprise, launch a new EC2 instance from an AMI for the new version of LGTM.

   To upgrade the virtual machine, launch a new EC2 instance from the same AMI as the old instance.

   In both cases, repeat the instructions for launching a new instance (see "Launch an EC2 instance containing LGTM Enterprise" on page 55), except:

   a. If desired, select a larger **Instance Type**.

   b. In the **Add Storage** step, for the LGTM data volume, specify the snapshot created in the previous step, as follows:

- **Volume Type**: `EBS`
- **Device**: `/dev/sdf`
- **Snapshot**: Paste in the ID of the snapshot you created in step 3.
- **Size**: Use the same size as the data volume in the old EC2 instance.
- **Delete on Termination**: not selected

5. After launching the instance:

   a. Wait for the **Instance State** to change to "running."

   b. Change the public IP address of the new EC2 instance to the Elastic IP address of the old EC2 instance, as follows:

      i. In the navigation pane, click **Elastic IPs**.

      ii. Select the Elastic IP address of the old EC2 instance.

      iii. Click **Actions** > **Disassociate Elastic IP address**.

      iv. In the dialog box, click **Disassociate**.

      v. With the IP address still selected, click **Actions** > **Associate Elastic IP address**.

      vi. On the Associate Elastic IP address page, click in the **Instance** field and choose "LGTM" (or your alternative name if you chose to use a different value for the Name tag).

      vii. Click **Associate**.

6. The new LGTM Enterprise instance may take about 10 minutes to initialize itself. When this has completed, you can log into the web interface as before and check that the system has upgraded successfully.

7. After you have established that the system upgraded successfully, you can clean up the resources used by the old EC2 instance:

a. Terminate the old EC2 instance:

   i. In the navigation pane, click **Instances**.

   ii. Select the old, stopped EC2 instance.

   iii. Click **Actions** > **Instance State** > **Terminate**.

   iv. Confirm that you want to terminate the instance.

b. Delete the root volume from the old AWS instance

   i. In the navigation pane, click **Volumes**.

   ii. Select the root volume for the old EC2 instance.

   iii. Click **Actions** > **Delete Volume**.

   iv. Confirm that you want to delete the volume.

c. Delete the snapshot you took of the old data volume:

   i. In the navigation pane, click **Snapshots**.

   ii. Select the root volume for the old EC2 instance.

   iii. Click **Actions** > **Delete**.

   iv. Confirm that you want to delete the snapshot.

Dockerized deployment

This section explains how to use Kubernetes to deploy LGTM Enterprise within Docker containers. The process described is the same whether you are carrying out an initial installation or an upgrade to a new version of LGTM Enterprise.

After deployment, each of the main LGTM services runs within its own Docker container. Your Dockerized LGTM Enterprise will have one container for each worker daemon (you may have many of these), one for each instance of the LGTM web application (typically one is sufficient), and one each for other services in LGTM's control pool.

Important
The Docker containers in which LGTM workers are deployed are based on Ubuntu. Currently no Windows-based worker containers are created. Therefore, you can only use a Dockerized deployment of LGTM Enterprise to analyze projects that can be built on Linux.

Assumptions

This topic assumes the following:

- You are familiar with the principles of Docker containerization and with Docker tools.
- You have access to a deployment of Kubernetes, or a similar container management system.
- You are familiar with the process of using Kubernetes, or similar, to install containerized applications.

Distribution file for Docker

The following compressed archive file is available for download:

`lgtm-`*`<release-number>`*`-helm.tar.gz`

The package includes an `lgtm-enterprise` directory, containing a Helm chart.

Deployment using Helm

You can use the supplied Helm chart in a number of ways, depending on how you want to deploy LGTM Enterprise.

- `helm template ...`

  Use the Helm client to generate plain Kubernetes objects locally. This is useful if Tiller (Helm's in-cluster component) is not installed within the Kubernetes cluster.

  The worked example in this topic uses this method.

- `helm install ...`

  If you are already using Helm to deploy containerized applications to your cluster, and you have the Tiller component installed within your Kubernetes cluster, this is the preferred command to use.

  See the Helm documentation for details of the `helm install` command.

- Custom deployments

  The supplied Helm chart can be used as the starting point for writing a completely custom container orchestration.

  Caution
  A custom configuration is likely to be a significant maintenance burden as it will have to be updated for each new LGTM Enterprise release.

Configuration options

The Helm chart contains a `values.yaml` file that shows the default values for all available configuration options when deploying using either `helm install` or `helm template`. For example, this extract from the file shows the configuration settings for the worker container:

```
worker:
  replicas:
    general: 1
    query: 1
    on-demand: 0
  resources:
    requests:
      memory: 8192Mi
      cpu: 125m
    limits:
      memory: 8192Mi
```

You can override these values by using the `-f` flag to point to a file containing replacement values. This is demonstrated in the worked example below.

Worked example

The procedure below steps you through one way of installing or upgrading LGTM Enterprise on a Google Kubernetes Engine (GKE) cluster—that is, a Kubernetes cluster running on Google Cloud Platform. The steps assume that you already have a machine set up with access to a GKE cluster as well as the Helm client installed locally (Tiller does not need to be installed in the cluster).

The resulting Dockerized LGTM Enterprise has the most basic of setups, containing one general worker daemon and one query worker daemon. After an initial basic installation in Docker, you can extend the system as required.

Prerequisites

Make sure following are true before you start this procedure:

- You have created a Kubernetes cluster and it is currently running.

- You have installed the Kubernetes CLI (`kubectl`) on your local computer.

- The current context set for the Kubernetes CLI is the cluster to which you want to deploy LGTM Enterprise.

  You can check the current context by running:

  ```
  kubectl config get-contexts
  ```

  And you can set it by running:

  ```
  kubectl config use-context <cluster name>
  ```

- You have installed the Helm client (`helm`) on your local computer.

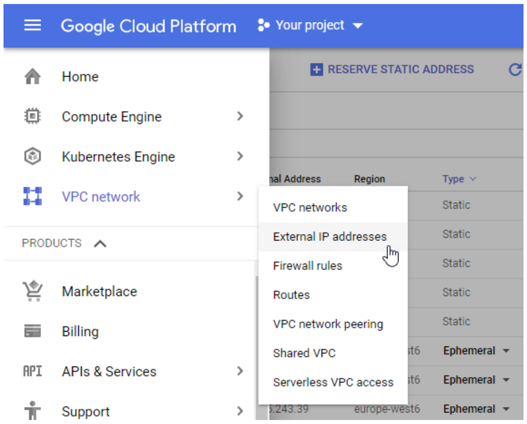
- You have downloaded the `lgtm-`*`<release-number>`*`-helm.tar.gz` package to your local computer.

- You have an LGTM Enterprise license.

Procedure

1. Create a static URL for the ingress to the LGTM Enterprise web application:

   a. In Google Cloud Platform, go to **VPC network** > **External IP addresses**:

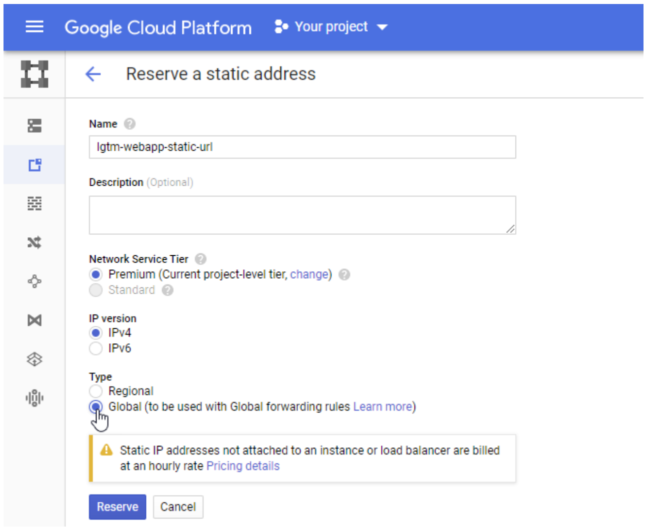
   

   b. Click **RESERVE STATIC ADDRESS**.

   c. Give the setting a name.

      For example, `lgtm-webapp-static-url`.

   d. Set the Type to Global.



Reserve a static address

Name: lgtm-webapp-static-url

Description (Optional)

Network Service Tier
● Premium (Current project-level tier, change)
○ Standard

IP version
● IPv4
○ IPv6

Type
○ Regional
● Global (to be used with Global forwarding rules Learn more)

Static IP addresses not attached to an instance or load balancer are billed at an hourly rate Pricing details

e. Click **Reserve**.

f. On the External IP addresses page, make a note of the name you chose and the static URL that has been assigned.

2. Create a file locally, alongside the downloaded `lgtm-<release-number>-helm.tar.gz` file, called `values-override.yaml` (or an alternative name if you prefer).

   Populate the file with the contents shown below (changing the values as indicated).

```
webapp:
  externalURLOverride: "http://<static IP address of web app>/"
  service:
    type: NodePort # This is needed because GKE uses load balancers
                   # that are external to the cluster so ClusterIP
                   # services are not visible to them.
  ingress:
    host: null # We don't have a domain name for this instance yet.
    annotations:
      kubernetes.io/ingress.global-static-ip-name: "<name of the web
app static IP entry>"
```

> **Important**
> You must change the `externalURLOverride` and `kubernetes.io/ingress.global-static-ip-name` values, replacing the text in angle brackets with the static IP address and the name that you chose for it in GCP.

We'll use this file to override default values that are provided in a `values.yaml` file, which is included within the package file.

If you want to change other default settings (for example, to give you more worker daemons) you could do so by examining the `values.yaml` file and adding the appropriate settings to the `values-override.yaml` file.

3. In the same location in which you created the `values-override.yaml` file, make a directory to store Kubernetes object templates:

   ```
   mkdir lgtm-kubernetes
   ```

   The current working directory should now include an `lgtm-kubernetes` directory, the `lgtm-`*`<release-number>`*`-helm.tar.gz` file and the `values-override.yaml` file.

4. Run the command:

   ```
   helm template --name my-lgtm --output-dir lgtm-kubernetes -f values-override.yaml lgtm-<release-number>-helm.tar.gz
   ```

   This creates a set of template files in the directory you created.

5. Apply these templates to create Kubernetes objects in your cluster:

   ```
   kubectl apply -f lgtm-kubernetes/lgtm-enterprise/templates
   ```

6. Check whether the Kubernetes pods are running by entering:

   ```
   kubectl get all
   ```

   The first table in the output shows which pods are running. If some are listed as initializing, run the command again until they are all either running or completed. For example:

   ```
   NAME                                        READY   STATUS     RESTARTS   AGE
   pod/my-lgtm-cli-5fcfc58dff-sxckm            1/1     Running    0          8m35s
   pod/my-lgtm-config-gen-576895c9b7-fvjv7     1/1     Running    0          8m34s
   ```

```
pod/my-lgtm-database-6495c4dbc-njmvw                    1/1     Running     0          8m18s
pod/my-lgtm-filestorage-6cf7d74c56-8vzqb                1/1     Running     0          8m14s
pod/my-lgtm-jobdealer-7654b7b8fc-ssmfq                  1/1     Running     0          8m29s
pod/my-lgtm-scheduler-664474b85d-77r2r                  1/1     Running     0          8m27s
pod/my-lgtm-search-8d7476767-2f78m                      1/1     Running     0          8m11s
pod/my-lgtm-task-worker-5446d7f9f7-bn9tk                1/1     Running     0          8m26s
pod/my-lgtm-taskqueue-594dcc8578-jbgwq                  1/1     Running     0          8m8s
pod/my-lgtm-upgrade-1.22.3-dev.20191030022938-pgk9n     0/1     Completed   0          8m25s
pod/my-lgtm-upgrade-79bc5cb6f6-vjtcd                    1/1     Running     0          8m24s
pod/my-lgtm-webapp-7bbf8fb758-hv9bq                     2/2     Running     0          8m23s
pod/my-lgtm-worker-general-b9b4568c4-8xrm7              1/1     Running     0          8m20s
pod/my-lgtm-worker-query-5d89995ddb-n8s5k               1/1     Running     0          8m19s
```

7. Create a new LGTM-maintained user account:

```
kubectl exec --stdin --tty "$(kubectl get pod --selector
"app.kubernetes.io/instance=my-lgtm,app.kubernetes.io/component=cli" --output
jsonpath="{.items[0].metadata.name}")" -- lgtm-cli create-user --email <email
address> --password <password> --display-name <name of choice>
```

Note

The above command is the same as the following, but with the pod name automatically inserted into the command:

```
kubectl exec "<name of LGTM CLI pod>" -- lgtm-cli create-user --email
<email address> --password <password> --display-name <name of choice>
```

For example:

```
kubectl exec "my-lgtm-cli-5fcfc58dff-sxckm" -- lgtm-cli create-user --
email jdoe@example.com --password JDs^P@$$VVurD --display-name "John
(admin)"
```

8. Grant this account administrator rights:

```
kubectl exec --stdin --tty "$(kubectl get pod --selector
"app.kubernetes.io/instance=my-lgtm,app.kubernetes.io/component=cli" --output
jsonpath="{.items[0].metadata.name}")" -- lgtm-cli grant-admin --name <email
address>
```

Note

The above command is the same as the following, but with the pod name automatically inserted into the command:

```
kubectl exec "<name of LGTM CLI pod>" -- lgtm-cli grant-admin --name
<email address>
```

For example:

```
kubectl exec "my-lgtm-cli-5fcfc58dff-sxckm" -- lgtm-cli grant-admin --
name jdoe@example.com
```

9. Using the static IP address for the web application, browse to:

   `http://<IP-address>/admin/`

10. Log in to LGTM Enterprise using the newly created account.

11. On the Settings page, update the server configuration by entering the address at which users will access LGTM in the **External URL** field.

    The address you enter must resolve to the static IP address of the ingress to the web application. You can change this address at any time, so, if required, you can use the static IP address initially and then change this to a domain name later.

12. Also on the Settings page, upload your LGTM Enterprise license.

Note

You must set the external URL before you can upload files to LGTM Enterprise.

Post-installation procedures

After installing LGTM Enterprise you can log into the application, go to the administration pages, perform some basic configuration, and test the system.

Accessing the administration interface

1. In a browser, go to https://<*LGTM-domain*>/admin.

   Initially, without a trusted SSL certificate, your browser will display a security warning. The exact message depends on the browser you are using.

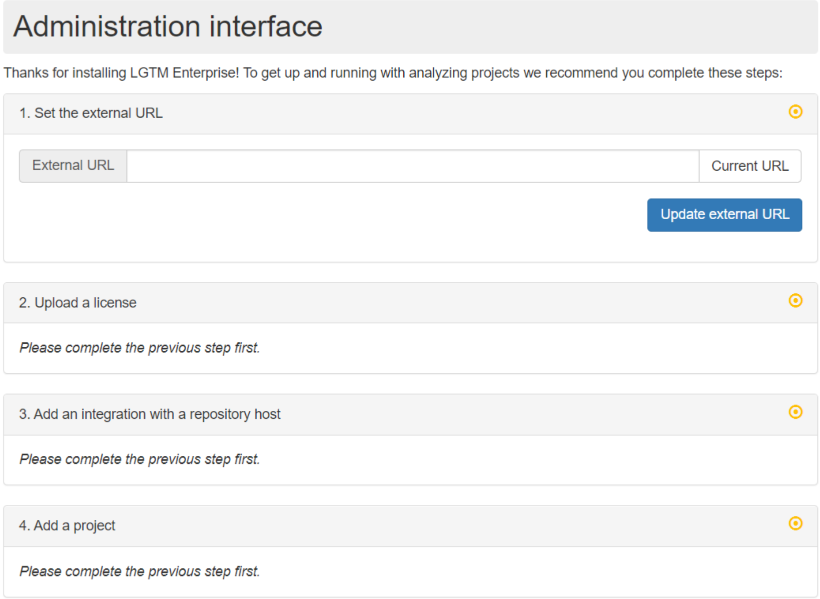
   There should be an advanced option you can choose to allow you to proceed to the site. Choose this option for now, so that you can access LGTM Enterprise and configure SSL. If you do not see the option to continue, do a hard refresh of the page.

   The LGTM Enterprise log in page is displayed.

2. Log in to LGTM Enterprise using the administrator account credentials you entered during the installation process.

3. If prompted to do so, choose a user name, or accept the default, and click **Let's go**.

   LGTM's admin home page is displayed, with some initial tasks you should perform:



Administration interface

Thanks for installing LGTM Enterprise! To get up and running with analyzing projects we recommend you complete these steps:

1. Set the external URL

2. Upload a license

Please complete the previous step first.

3. Add an integration with a repository host

Please complete the previous step first.

4. Add a project

Please complete the previous step first.

4. To see information about configuring and administering LGTM Enterprise, click **Admin help** in the menu bar. This displays LGTM's built-in administrator help.

Tip

A searchable version of the administrator help is available at help.semmle.com.

Modifying a Dockerized deployment

If you've deployed LGTM Enterprise in Kubernetes, using the helm template method described in "Dockerized deployment" on page 65, you can modify your configuration by redeploying the supplied Helm chart with new overrides. See the supplied `helm/lgtm-enterprise/values.yaml` file for details of the settings you can override.

1. Make the required changes to your existing values overrides file (in this example, this file is called `values-override.yaml`, and is saved alongside the `helm` and `lgtm` directories, and the `load-all-images.py` file—for more details, see "Dockerized deployment" on page 65).

   For example, you might add the following to change the default number of workers to 4 general workers, 1 query worker, and 1 on-demand worker:

   ```
   worker:
     replicas:
       general: 4
       query: 1
       on-demand: 1
     resources:
       requests:
         memory: 8192Mi
         cpu: 125m
       limits:
         memory: 8192Mi
   ```

2. From the directory where this file is saved, run the command:

   ```
   helm template --name my-lgtm --output-dir lgtm-kubernetes -f values-override.yaml ./helm/lgtm-enterprise
   ```

   This creates a set of template files in the `lgtm-kubernetes` directory.

3. Apply these templates to create Kubernetes objects in your cluster:

```
kubectl apply -f lgtm-kubernetes/lgtm-enterprise/templates
```

4. If your changes have affected a currently running Kubernetes pod (rather than adding new pods):

   a. Identify the name of the affected pod by running:

   ```
   kubectl get pods
   ```

   b. Delete the affected pod:

   ```
   kubectl delete pods <pod name>
   ```

   For example, this command deletes a web application pod:

   ```
   kubectl delete pods my-lgtm-webapp-6d7469c558-pshdv
   ```

   The deleted pod is immediately regenerated with the new configuration.

Useful help topics

The following topics on help.semmle.com (also available within the application's administrator help) provide useful information on administrative tasks you might want to carry out:

- Securing LGTM Enterprise—guidelines for setting up and maintaining a secure system
- Configuring the environment on the worker host machines to allow workers to analyze code
- Defining integrations with external systems to allow LGTM to access source code repositories, and to allow users appropriate access to project results
- Replacing the initial administrator account—recommended if you are using accounts managed by external systems
- Adding projects for initial analysis
- Removing projects added for testing
- Checking the progress of analysis for projects you have recently added
- Assessing worker resourcing: do you have too few, or too many, worker daemons?
- Adding and removing workers

- Editing and deploying the cluster configuration—for example, to add more worker host machines
- Creating user accounts
- Viewing analysis results as a standard LGTM user
- Making LGTM's plugins available for people to install in their software development applications

Post-upgrade procedures

Check the relevant After you upgrade to... section in the appendix, starting on page 78, for any additional steps you need to take to configure the new version.

Post-upgrade housekeeping

After upgrading LGTM Enterprise, if you have used the directory structure suggested in this documentation, you'll have the following directories:

- `<path-to>/lgtm-releases/lgtm-<old-version>`
- `<path-to>/lgtm-releases/lgtm-<current-version>`

You can delete the `lgtm-<old-version>` directory. It's not used after you've upgraded. For example, you might delete the `lgtm-1.20.2` directory after upgrading to LGTM Enterprise 1.21.

You can also delete the `lgtm-*.tar.gz` distribution file(s) in the `lgtm-releases` directory.

Appendix A—Notes for upgrading LGTM Enterprise

Important

- You must upgrade to each version in turn. This ensures that any data migration required by each upgrade is completed successfully before you begin the upgrade to the next version. This does not include maintenance releases so, for example, you can upgrade from any 1.22.x version to any 1.23.x version.
- If you are stepping through more than one upgrade, you must wait for data to be upgraded before you start the next upgrade. Go to the home page of the administration interface (<*LGTM-URL*>/admin/) and make sure there is not a message at the top of the page saying that "database migrations still remain to be run."
- If you have made any manual changes to the configuration of LGTM, beyond editing the cluster configuration file, upgrading may overwrite these changes. We recommend that you back up your system before upgrading.

Before proceeding with an upgrade, read the relevant section below.

You can then proceed, and perform either an interactive upgrade or a programmatic upgrade.

Upgrading from 1.22 to 1.23

After you upgrade to 1.23

Wait for databases to be upgraded—In previous releases, uploaded CodeQL databases (previously known as snapshots) became incompatible after upgrading to the new release. This prevented queries being successfully run against projects added in upload analysis mode. In this release, uploaded CodeQL databases are automatically upgraded so that queries can be run without requiring a new database to be generated and uploaded. The automatic upgrade happens as a number of scheduled jobs, one for each project. These jobs are spread out across a day, to reduce load on the system. You will have to wait a maximum of 24 hours, therefore, for all projects to be available for querying again after upgrading to this release.

Upgrading from 1.21 to 1.22

Before you upgrade to 1.22

Updated example programmatic deployment resources—The example resources at https://github.com/Semmle/lgtm-bash-deployment and https://github.com/Semmle/lgtm-ansible-example are usually updated for each release. If you previously used any of these resources for deploying LGTM 1.21, please download the new versions of the files for deploying LGTM 1.22. If you require any assistance with preparing or conducting a programmatic deployment, please contact GitHub Support.

After you upgrade to 1.22

Watch out for unexpected job failures—LGTM 1.22 introduces time limits for some additional jobs, including poll and attribution jobs, to improve security. If any jobs start to time out after the upgrade, you can increase the value in the Code analysis page.

Task worker is created on every web app host—This new type of worker processes some of the data that was previously processed by the web app. No action is required, new workers are automatically created on relevant machines during the upgrade.

Installations with projects in upload analysis mode—When you upgrade to a new version of LGTM, CodeQL databases (previously called snapshots) for projects in full and sparse analysis modes are automatically regenerated. This ensures that they are available for download and for use in the query console.

CodeQL databases for projects in upload analysis mode cannot be regenerated within LGTM, so you need to replace them with compatible databases. That is, upload either a new database created with version 1.22.x of the command-line tools, or an existing database upgraded to version 1.22.x. Until you do this, these projects will not be available for analysis. For more information about upload analysis, see Using upload analysis.

Upgrading from 1.20 to 1.21

Before you upgrade to 1.21

Updated example programmatic deployment resources—The example resources at https://github.com/Semmle/lgtm-bash-deployment and https://github.com/Semmle/lgtm-ansible-example have been modified for this release. If you previously used any of these resources for deploying LGTM 1.20, please download the new versions of the files for deploying LGTM 1.21. If you require any assistance with preparing or conducting a programmatic deployment, please contact GitHub Support.

Upgrading from 1.19 to 1.20

Before you upgrade to 1.20

Example programmatic deployment resources—If you want to use the example resources for setting up a programmatic deployment (at https://github.com/Semmle/lgtm-bash-deployment and https://github.com/Semmle/lgtm-ansible-example) use the version of these files tagged `v1.20.0`. If you require any assistance with preparing or conducting a programmatic deployment, please contact GitHub Support.

After you upgrade to 1.20

Contact GitHub Support if you have been running workers on Windows—Version 1.20 of LGTM Enterprise introduces a new, streamlined system of installing, upgrading and maintaining worker daemons on Windows. If you're already running workers on Windows, you can continue using these after upgrading. However, you should contact GitHub Support about migrating these workers to the new system to ensure compatibility with future updates.

Note

If you have not yet used Windows machines to run worker daemons and you now wish to do so, there is no need to contact GitHub Support. Instructions for adding Windows worker hosts are now included the administrator help—see Adding new hosts to the work pool.

Upgrading from 1.18 to 1.19

After you upgrade to 1.19

Caution

All integrations with repository access configured, other than those for Subversion and simple Git, must now specify a user name and password. This was strongly recommended in version 1.18 of LGTM Enterprise, but is now mandatory in 1.19. The reasons behind this change are described in security advisory SSA-2018-004.

After upgrading you must ensure that the settings for all repository access (except Subversion and simple Git) include the name and credentials for a valid account. Check the Administration interface page for the error message: "No new commits will be analyzed for projects hosted on repository provider <*name*>. Please add credentials to its configuration page."

The account is used by LGTM Enterprise:

- To determine a variety of metadata about repositories—for example, whether the repository name has changed
- To post comments and set statuses on the pull requests of projects for which automated code review integration has been set up

For more information about integrations, see Defining integrations with external systems.

Enabling custom queries—To enable users to add custom queries to your LGTM instance, make sure to set an organization identifier on the Settings page of the administration interface. See Defining your organization identifier.

Appendix B—Amazon EC2 instance types

Before launching an LGTM Enterprise instance on Amazon Web Services (AWS), you need to determine the Amazon EC2 instance type that best fits the needs of your organization. For information about the differences between instance types, see the reference page on the Amazon help site.

Based on your user license count—and a typical initial number of general worker daemons for that number of users—we recommend the following instance types.

User license	General workers	Recommended EC2 instance type
Trial, demo, or 10 light users	1	**r5.xlarge**
10–500 users	5	**r5.2xlarge**
500–1000 users	12	**r5.4xlarge**
1000–5000 users	30	**r5.8xlarge**
5000+ users	60	**r5.16xlarge**

The table above shows a good starting point for a deployment of LGTM Enterprise. If you have a particularly large number of projects—or particularly active projects—you may need to increase the number of worker daemons. You can do this after the initial deployment by stopping the instance and choosing an alternative EC2 instance type (with vCPUs and memory appropriately scaled up for the increased number of workers). At the same time, change the number of workers—by changing `n-general`, `n-query`, or `n-on-demand`, as required—in the **User data** field of the EC2 instance settings. Then start the instance again.

The instance type choices shown above assume that worker multi-threading is not enabled (the default configuration). If you enable workers to use multi-threading—by setting the `LGTM_THREADS` environment variable in the **User data** field—you'll need to choose instance types with accordingly increased vCPU. Switch to an **m5.*** or **c5.*** machine with a similar amount of memory, ensuring that (`<vCPUs in the EC2 instance> / <LGTM_THREADS>`) is greater than the number of general workers. The following table gives a couple of examples:

User license	General workers	Recommended EC2 instance type
500–1000 users with LGTM_THREADS: 2	12	**m5.8xlarge**
500–1000 users with LGTM_THREADS: 4	16	**c5.18xlarge**

Note

- Setting the `LGTM_THREADS` environment variable often requires setting the `LGTM_RAM` environment variable, to increase the memory available to workers above the default of 6 GB. For more information, see the topic "LGTM-specific environment variables" in the administrator help.

- If you need to set the `LGTM_RAM` environment variable—either because you have increased the `LGTM_THREADS` environment variable, or because the projects you are analyzing are large—you must make sure that the instance still has enough memory to cover the new demands. The number of general workers should not exceed `(`*`<instance RAM in GB>`*` - 24) / `*`<LGTM_RAM in GB>`*`)`. In this calculation, 24 GB is the minimum RAM needed for non-worker parts of LGTM Enterprise.

- Make sure to choose an instance type with EBS storage only, not including a local NVMe-based SSD (usually indicated by a "d" in the instance type name). For example, you cannot use **r5d.xlarge**.