LexPredict ContraxSuite Documentation

Administration Guide for Linux

Release 1.0.1 - September 1, 2017

Getting Started

Licensing Information

ContraxSuite is available under a dual-license open-source model. Unless otherwise released by ContraxSuite, LLC, you can use and modify this software under the terms of the GNU AFFERO GENERAL PUBLIC LICENSE.

If you have question about licensing or would like to request a release from the AGPL license, please email license@contraxsuite.com.

ContraxSuite also relies on a software and data dependencies that are independently licensed. For more information about these dependencies and their respective licensing models, please refer to the Software and Data Dependency documentation in this release.

Support, Customization, Hosting, or Training

We can help! If you need assistance customizing, hosting, training, or supporting your ContraxSuite instance, please reach out to us to discuss options. You can always email us support@contraxsuite.com or create issue on Github.

Prerequisite Skills

This document assumes basic familiarity with the installation and configuration enterprise multi-tier applications. In particular, personnel with Linux, Java, and Python experience are best suited to successfully support ContraxSuite installations.

How to Get Support

For support or help in setting up the application, please contact support@contraxsuite.com.

Scope

This Administration Guide is designed to meet the needs of system administrators, IT operations personnel, and developers. This Guide is not meant for use by end-users.

CAUTION: If you are planning to use ContraxSuite in production or for real projects, you should carefully evaluate whether the default architecture meets your technical and security requirements.

Administration Guide for Linux

Managing the RDBMS

Postgresql

Task	Command
Starting the server	**`sudo systemctl start postgresql`**
Stopping the server	**`sudo systemctl stop postgresql`**
Checking service status	**`sudo systemctl status postgresql`**
Log file path	**`/var/log/postgresql/`**

Managing the Message Broker

Redis

Task	Command
Starting the server	**`sudo systemctl start redis_6379`**
Stopping the server	**`sudo systemctl stop redis_6379`**
Checking service status	**`sudo systemctl status redis_6379`**
Log file path	● **`/var/log/redis_6379.log`** ● **`/var/log/redis/`**

Managing the Enterprise Search Index

ElasticSearch 2.x

Task	Command
Starting the server	**`sudo systemctl start elasticsearch`**
Stopping the server	**`sudo systemctl stop elasticsearch`**
Checking service status	**`sudo systemctl status elasticsearch`**
Log file path	**`/var/log/elasticsearch/`**

Managing the Application Server

uwsgi

The uwsgi service name depends on the **templates_prefix** setting in fabricrc file used for installation. The default name is **contrax** unless otherwise changed.

Task	Command
Starting the server	**`sudo systemctl start contrax_uwsgi`**
Stopping the server	**`sudo systemctl stop contrax_uwsgi`**
Checking service status	**`sudo systemctl status contrax_uwsgi`**
Log file path	**`/var/log/uwsgi/`**

Managing the Distributed Task Engine

Celery

Managing the Celery task engine should be done from within the application's python virtual environment (virtualenv). To activate the application's virtual environment, follow the steps below:

```
1. cd /opt/lexpredict-contraxsuite/
2. source ve/bin/activate
3. cd contraxsuite-services/
```

Note: The path to the project root in step #1 above is defined in fabric deployment automation. The example above uses the default paths, but your path may vary.

Celery can be configured to launch as many workers as desired. The examples below demonstrate how to launch celery with **two (2)** workers. For more information about optimizing Celery configuration, please review the Celery documentation or reach out to support.

Task	Command
Starting the server	`celery multi start 2 -A apps.task -B -f celery.log`
Stopping the server	`celery multi stop 2 -A apps.task`
Checking **registered** tasks	`celery inspect registered -A apps.task`
Checking **active** tasks	`celery inspect active -A apps.task`
Log file path	`CELERY_LOG_FILE_NAME (settings.py)`

Managing the Web Server

Nginx

Task	Command
Starting the server	`sudo systemctl start nginx`
Stopping the server	`sudo systemctl stop nginx`
Checking service status	`sudo systemctl status nginx`
Log file path	`/var/log/nginx/`

More Documentation

This Administration Guide focuses on simple single-server deployments for development, research, or testing. As a result, it does not cover many of the architecture, installation, or configuration topics that most organizations will encounter.

For more information, organizations should refer to these documents below:

- Installation and Configuration Guide
- Software and Data Dependencies
- Architecture Diagram
- Data Model Diagram