**Prices Paid Portal (P3) Installation Plan**

**Version 1.7**

**02/05/2016**



**ACCEPTANCE / APPROVAL PAGE**

|  |  |  |
| --- | --- | --- |
| // //\_  Gary W Barber Architect | Prepared by | Date |
| // // \_  Name  Quality Assurance | Reviewed by | Date |
| // // \_  Name  Margaret Nguyen | Approved by | Date |

**DOCUMENT CHANGE HISTORY**

|  |  |  |
| --- | --- | --- |
| **Version Number** | **Date** | **Description** |
| 1.0 | March 6, 2015 | Baseline release |
| 1.1 | March 12, 2015 | Added reference to service account creation |
| 1.2 | April 16, 2015 | Stubbed out section 5.1 to include headers for Solr, Banana and Drupal Updates. |
| 1.3 | April 16, 2015 | Added Git update procedures in section 6.1.1 |
| 1.4 | July 1, 2015 | Added step to rebuild Solr Index from Drupal |
| 1.5 | August 5, 2015 | Added Scaling Section |
| 1.6 | February 1, 2016 | Updated Solr section to include update to Solr, ZK and Fusion installation instructions and AWS instance type information. |
| 1.7 | February 5, 2016 | Changed CAP to P3, removed references to Hallways |

**Table of Contents**

1. Introduction 5

1.1 Overview Of Software/System 5

2. System environment 5

2.1 Instance Type REQUIREMENTS 6

2.2 Site Requirements 8

3. Security considerations 8

4. Software/System Inventory 8

5. Installation Instructions 9

5.1 Github/ Jenkins 9

5.1.1 Manual GitHub Installation 11

5.1.2 Trigger Build 11

6. Installation Instructions For Upgrades 13

6.1 Lucidworks Solr server 13

1.1.1 Contents of the Fusion Home directory 14

1.1.1 Lucidworks with zookeeper 15

1.1.1.2 Update heap sizes in Fusion 18

3. Running Fusion 20

1.1.1.3 Fusion Configuration Settings 21

1.1.1.4 Create system\_banana-default pipelines 26

6.1.1 Web server 28

6.1.2 Building P3 code 28

6.1.3 Automation with Jenkins 28

6.1.4 Enable FIPS in Tomcat 29

6.1.5 Password 31

7. SSL Certificates 32

8. DNS Names 32

8.1 De-Installation Instructions For Upgrades 33

8.1.1 Git Upgrade 34

9. Frequently Asked Questions 34

10. References 34

**Tables and Figures**

**Table 1: Development VPC** 6

**Table 2: Test VPC** 7

**Table 3: Staging VPC** 7

**Table 4: Production VPC** 8

**Table 9: GitHub / Jenkins Plugin Installation** 10

**Table 10: GitHub/ Jenkins Plugin Manual Installation** 11

**Table 11: Trigger to Build when GitHub Changes** 12

# Introduction

## Overview Of Software/System

The System Installation Plan describes how the Prices Paid Portal (P3) will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort (such as hardware, software. facilities, materials, and personnel), and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase; the final version is provided in the Integration and Test Phase and is used for guidance during the Implementation Phase.

GSA’s P3 team manages the the Solr environment.  Using the practice of frequent, iterative development we let customer feedback inform how and what we build.

* GSA’s P3 team is building the Prices Paid Portal Tool is our effort to provide greater visibility on the prices paid by government agencies for commonly purchased goods and services and related purchasing behaviors to the acquisition community, in order to support efforts to reduce total cost of ownership for these goods and services.
* We are a working example of the government’s adoption of Agile. The P3 team follows the principles of frequent iterative feedback from across government as we build the Prices Paid Portal using a collaborative and transparent process.    
  ​

# System environment

The P3 Environment is hosted in Amazon AWS EC2. There are currently 3 Virtual Private Clouds (VPC) that host the Development, Test, Staging, and Production environment. The Development VPC consists of an Apache Webserver, Jetty, Apache Tomcat, and Apache Solr. The Test environment consists of an Apache Webserver, Jetty, Apache Tomcat, and Apache Solr. The Production server consists of an Apache Webserver, Jetty, Apache Tomcat, and Apache Solr.

## Instance Type REQUIREMENTS

**Table 1: Development VPC**

|  |  |
| --- | --- |
| **Banana Web Server – AWS type: m3. large** | |
| ***Processor Type*** | 2 vCPU |
| ***System Memory*** | 7.5 gb |
| ***Hard Disk Space*** | 1 x 32 gb SSD |
| **Solr Server– AWS type: m3. xlarge** | |
| ***Processor Type*** | 4 vCPU |
| ***System Memory*** | 15 gb memory |
| ***Hard Disk Space*** | 2x40 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Solr Server– AWS type: m3. xlarge** | |
| ***Processor Type*** | 4 vCPU |
| ***System Memory*** | 15 gb memory |
| ***Hard Disk Space*** | 2x40 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Parts*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |



**Table 2: Test VPC**

|  |  |
| --- | --- |
| **Banana Web Server – AWS type: m3. Large** | |
| ***Processor Type*** | 2 vCPU |
| ***System Memory*** | 7.5 gb |
| ***Hard Disk Space*** | 1 x 32 gb SSD |
| **Solr Server– AWS type: m3. xlarge** | |
| ***Processor Type*** | 4 vCPU |
| ***System Memory*** | 15 gb memory |
| ***Hard Disk Space*** | 2x40 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Solr Server– AWS type: m3. xlarge** | |
| ***Processor Type*** | 4 vCPU |
| ***System Memory*** | 15 gb memory |
| ***Hard Disk Space*** | 2x40 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Parts*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |



**Table 3: Staging VPC**

|  |  |
| --- | --- |
| **Banana Web Server – AWS type: m3. Large** | |
| ***Processor Type*** | 2 vCPU |
| ***System Memory*** | 7.5 gb |
| ***Hard Disk Space*** | 1 x 32 gb SSD |
| **Solr Server—**AWS type: c3.8xlarge | |
| ***Processor Type*** | 32 vCPU |
| ***System Memory*** | 60 gb memory |
| ***Hard Disk Space*** | 2x320 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Solr Server—**AWS type: c3.8xlarge | |
| ***Processor Type*** | 32 vCPU |
| ***System Memory*** | 60 gb memory |
| ***Hard Disk Space*** | 2x3200 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Parts*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |



**Table 4: Production VPC**

|  |  |
| --- | --- |
| **Banana Web Server – AWS type: m3. large** | |
| ***Processor Type*** | 2 vCPU |
| ***System Memory*** | 7.5 gb |
| ***Hard Disk Space*** | 1 x 32 gb SSD |
| **Solr Server** —AWS type: c3.8xlarge | |
| ***Processor Type*** | 32 vCPU |
| ***System Memory*** | 60 gb memory |
| ***Hard Disk Space*** | 2x3200 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Solr Server—AWS type: c3.8xlarge** | |
| ***Processor Type*** | 32 vCPU |
| ***System Memory*** | 60 gb memory |
| ***Hard Disk Space*** | 2x320 gb SSD |
| ***Open Ports*** | *22, 8764, 8765, 8983, 8984, 7984, 7983, 8764, 7764* |
| **Zookeeper – AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Parts*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |
| **Zookeeper– AWS type: m3. large** | |
| ***Processor Type*** | *2 vCPU* |
| ***System Memory*** | 7.5 gb memory |
| ***Hard Disk Space*** | 1x32 gb SSD |
| ***Open Ports*** | 2181, 2888, 3888 |

## Site Requirements

P3 will be deployed on the Amazon Elastic Compute Cloud (Amazon EC2). EC2 is a web service that provides resizable computing capacity in the cloud. It is designed to make web-scale [cloud computing](http://aws.amazon.com/what-is-cloud-computing/) easier for developers. Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

# Security considerations

The P3 site uses the OMB Max service that works with Amazon Virtual Private Cloud (VPC). No authorized privileges; read-only access to the contents of P3. User must be a Government employee, based on registration credentials in OMB MAX.

# Software/System Inventory

The software stack on the Prices Paid Portal consists of the following;

|  |  |
| --- | --- |
| **Software** | **Description** |
| Redhat Enterprise Linux (RHEL) | Linux Operating System |
| Lucidworks Fusion Manager | 1.2.3 |
| LucidSolr Solr | 4.10.4 |
| Lucidworks Zookeeper | 3.4.6 |
| Java | Java Standard SE open1.8.0\_25 |
| Apache Tomcat | 7.0.61 |
| Apache Http Webserver |  |
| Jetty | Jetty 8.1.10.v20130312 |
| AngularJS |  |
| Banana |  |

**Table 3: Software Inventory**

# Installation Instructions

This section provides step-by-step procedures for accomplishing the installation. References may be made to other documents, such as operator manuals. Safety precautions, marked by WARNING or CAUTION, will be included where applicable Media Boots Procedures

## Github/ Jenkins

Listed below are the instructions for installing the GitHub Plugin on Jenkins on the GSA on premise servers by Using the Interface.

First determine if service accounts have been created for GitHub and Jenkins. They will be used to check out code from GitHub and for Jenkins to kick off the deployment process to the target server.

Second determine if all SSH keys for the target servers have been installed in Jenkins so Jenkins can deploy the necessary information (e.g. code, configurations, etc...) to the target server.

|  |  |  |
| --- | --- | --- |
| **Step** | **Task** | **Results** |
| Note 1 | Instructions were obtained from <https://wiki.jenkins-ci.org/display/JENKINS/GitHub+Plugin> |  |
| Note 2 | This plugin requires that you have an HTTP URL reachable from GitHub, which means it's reachable from the whole internet. So it is implemented carefully with the possible malicious fake post-receive POSTS in mind. To cope with this, upon receiving a POST, Jenkins will talk to GitHub to ensure the push was actually made. |  |
| Note 3 | In case your Jenkins run inside the firewall and not directly reachable from the internet, this plugin lets you specify an arbitrary endpoint URL as an override in the automatic mode. The plugin will assume that you've set up reverse proxy or some other means so that the POST from GitHub will be routed to the Jenkins. |  |
| 1 | Go to your installation's management screen and clicking Manage Plugins (http://yourhost/jenkins/pluginManager/) and go to the Available tab. You'll find the Git plugin |  |
| 2 | Select the checkbox, and then either attempt to Install without restart or Download now and install after restart. The web interface will then download \*.hpi files from here. If you *Install without restart* the interface will show you progress and provide the results of the install.  Sometimes when you install, you will notice that the list of available plugins is empty. If that is the case, from *Advanced* tab on the *Manage Plugins* page, click on *Check now* (button available in the bottom right of the page) to forcefully check for new updates. Once that is done, you should see the list of plugins. |  |

**Table 9: GitHub / Jenkins Plugin Installation**

### Trigger Build

Trigger a Build when a change is pushed to Github

|  |  |  |
| --- | --- | --- |
| **Step** | **Task** | **Results** |
| Note 1 | Instructions were obtained from <https://wiki.jenkins-ci.org/display/JENKINS/GitHub+Plugin> |  |
| 1 | Go to the global configuration and choose the mode in which Jenkins manages [post-receive hooks in your GitHub repositories](http://help.github.com/post-receive-hooks/).  Automatic Mode: in this mode, Jenkins will automatically add/remove hook URLs to GitHub based on the project configuration in the background. You'll specify GitHub user names and OAuth token so that Jenkins can login as you to do this.  Manual Mode: in this mode, you'll be responsible for registering the hook URLs to GitHub. Click the https://wiki.jenkins-ci.org/images/icons/emoticons/help_16.gif icon to see the URL in Jenkins that receives the post-commit POSTs. |  |
| 2 | Open "Manage Jenkins > Configure Global Security" page and make sure that "Grant READ permissions for /github-webhook" is enabled in the "GitHub Authorization Settings" section. (If you are using the latest version of the plugin, you may not see the Github option on "Configure Global Security" page. Just go to the project configure page, in section "Build Triggers", check "Build when a change is pushed to GitHub") |  |
| 3 | Once that configuration is done, go to the project config of each job you want triggered automatically and simply check "Build when a change is pushed to GitHub" under "Build Triggers". With this, every new push to the repository automatically triggers a new build.  Note that there's only one URL and it receives all post-receive POSTs for all your repositories. The server side of this URL is smart enough to figure out which projects need to be triggered, based on the submission. |  |

**Table 11: Trigger to Build when GitHub Changes**

# Installation Instructions For Upgrades

## Lucidworks Solr server

* Download tar file from http://download.lucidworks.com/fusion-1.2.3.tar.gz
* Check <https://doc.lucidworks.com/index.html> for system requirements for updated information.
* Create directory /opt/lucidworks
* Install Fusion (if you would like to use Upstart for process management, you must install Fusion in /opt/lucidworks).
* Unpack the archive with "tar xzf fusion.tar.gz" (or "tar xzvf fusion.tar.gz").
* The resulting directory will be named "fusion". You can rename this if you wish.

### Contents of the Fusion Home directory

The fusion directory is considered your Fusion Home, and is referenced as $FUSION. It contains:

* KNOWN\_ISSUES.txt — a list of any known issues for this version of the software
* README.txt — an introduction file
* apps — directory, contains Fusion components 3rd-party distributions used by Fusion, including jarfiles and plugins.
* bin — directory, contains master script to run Fusion, per-component run scripts.
* conf — directory, configuration files for Fusion and ZooKeeper which contain parameters settings tuned for common use cases.
* data — directory, contains resources for Fusion components.
* docs — directory, contains Fusion license.
* examples — directory, Fusion signals example.
* fusion.build — text file containing version build information
* fusion.signature — text file containing version sig
* init — directory, contains systemd and upstart scripts for Linux
* scripts — directory, Fusion migration and upgrade scripts.
* var — directory, contains logfiles and system files created by Fusion components as well as .pid files for each running process.

*As distributed, Fusion uses the instance of Solr and Zookeeper included in the Fusion distribution both as the data store for Fusion’s own system and configuration information, as well as for all user collections. This is suitable for development purposes. For production deployments, ZooKeeper must be installed and run as a 3+ node ensemble.*

#### Running Fusion

All Fusion start scripts must be executed by a user who has permissions to read and write to the directories where Fusion is installed. These scripts don’t need to be run as root (or sudo), nor should they be. Use a suitable id, or create a new one, and then ensure that it owns the directory where fusion resides, (e.g. /opt/lucidworks).

##### Starting All Required Services

To run all required services from your Fusion Home :

* ./bin/fusion start (Linux/Mac OS)
* bin\fusion.cmd start (Windows)
* From the web:
  + SOLR: http://localhost:8983/solr/#
  + FUSION Admin: http://localhost:8764

This will start the bundled Solr and Zookeeper instance, the Fusion API, the Admin UI, and the Connectors. They each run in their own Jetty instances and on their own ports (see the section [Changing the Default Ports](https://doc.lucidworks.com/fusion/2.1/Installation_and_Configuration/Installing_Lucidworks_Fusion/Changing-the-Default-Ports.html) for more on ports

### Lucidworks with zookeeper

***Lucidworks with Zookeeper***

1. Create 3 zookeeper instances m3.large
2. yum install java-1.8.0-openjdk\_71
3. downloaded zookeeper: wget http://www.apache.org/dist/zookeeper/zookeeper-3.4.6/zookeeper-3.4.6.tar.gz
4. extract zookeper tar file to /opt/lucidworks/zookeeper-3.4.6 on zookeeper server

***Zookeeper changes***

1. create “data” directory under zookeeper root folder
2. create zoo.cfg file
3. copy zoo\_sample.cfg into zoo.cfg and change data path and server ip addresses

# The number of milliseconds of each tick

tickTime=2000

# The number of ticks that the initial

# synchronization phase can take

initLimit=10

# The number of ticks that can pass between

# sending a request and getting an acknowledgement

syncLimit=5

# the directory where the snapshot is stored.

# do not use /tmp for storage, /tmp here is just

# example sakes.

dataDir=/opt/lucidworks/zookeeper-3.4.6/data

# the port at which the clients will connect

clientPort=2181

# the maximum number of client connections.

# increase this if you need to handle more clients

#maxClientCnxns=60 – don’t have

#

# Be sure to read the maintenance section of the

# administrator guide before turning on autopurge.

#

# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc\_maintenance

#

# The number of snapshots to retain in dataDir

#autopurge.snapRetainCount=3- not added

# Purge task interval in hours

# Set to "0" to disable auto purge feature

#autopurge.purgeInterval=1

Development ZK

server.1=10.172.0.84:2888:3888

server.2=10.172.0.31:2888:3888

server.3=10.172.0.32:2888:3888

3. Create ./data/myid on each zookeeper server. Put a server number, such as 1,2,3 following order from zoo.cfg file

Zookeeper configuration changes in Lucidworks SOLR for Fusion

* 1. /opt/lucidworks/fusion/bin/config.sh (add following 2 lines to point to internal address of 3 zookeeper nodes)

FUSION\_ZK=10.172.0.84:2181, 10.172.0.31:2181, 10.172.0.32:2181 (use correct

FUSION\_SOLR\_ZK=10.172.0.84:2181,10.172.0.31:2181, 10.172.0.32:2181

* + 1. from /opt/lucidworks/fusion/bin
       1. edit solr file remove following line:

-DzkRun=$FUSION\_SOLR\_ZK \

#### Update heap sizes in Fusion

1. from /opt/lucidworks/fusion/bin
   1. edit config.sh
   2. Allocated JVM memory in **DEV** and **TEST** environments in both solr servers:

* API = 1g
* Connectors= 2g
* Solr= 4g
* UI= 512m

# this bash script file is included by common.sh

API\_PORT=8765

API\_STOP\_PORT=7765

API\_STOP\_KEY=fusion

API\_JAVA\_OPTIONS=(-Xmx1g -Xss256k -XX:MaxPermSize=256m -Dapple.awt.UIElement=true)

CONNECTORS\_PORT=8984

CONNECTORS\_STOP\_PORT=7984

CONNECTORS\_STOP\_KEY=fusion

CONNECTORS\_JAVA\_OPTIONS=(-Xmx2g -Xss256k -XX:MaxPermSize=256m -Dapple.awt.UIElement=true)

SOLR\_PORT=8983

SOLR\_STOP\_PORT=7983

SOLR\_STOP\_KEY=fusion

SOLR\_JAVA\_OPTIONS=(-Xmx4g -Xss256k -Dapple.awt.UIElement=true)

UI\_PORT=8764

UI\_STOP\_PORT=7764

UI\_STOP\_KEY=fusion

UI\_JAVA\_OPTIONS=(-Xmx512m -XX:MaxPermSize=256m -Dapple.awt.UIElement=true)

2.3 Allocation of JVM memory in **STAGING** and **PROD** in both Solr servers:

* + API = 2g
  + Connectors= 2g
  + Solr= 6g
  + UI= 512m

# this bash script file is included by common.sh

API\_PORT=8765

API\_STOP\_PORT=7765

API\_STOP\_KEY=fusion

API\_JAVA\_OPTIONS=(-Xmx2g -Xss256k -XX:MaxPermSize=256m -Dapple.awt.UIElement=true)

CONNECTORS\_PORT=8984

CONNECTORS\_STOP\_PORT=7984

CONNECTORS\_STOP\_KEY=fusion

CONNECTORS\_JAVA\_OPTIONS=(-Xmx2g -Xss256k -XX:MaxPermSize=256m -Dapple.awt.UIElement=true)

SOLR\_PORT=8983

SOLR\_STOP\_PORT=7983

SOLR\_STOP\_KEY=fusion

SOLR\_JAVA\_OPTIONS=(-Xmx6g -Xss256k -Dapple.awt.UIElement=true)

UI\_PORT=8764

UI\_STOP\_PORT=7764

UI\_STOP\_KEY=fusion

UI\_JAVA\_OPTIONS=(-Xmx512m -XX:MaxPermSize=256m -Dapple.awt.UIElement=true)

##### Starting All Required Services

To run all required services from your Fusion Home :

* ./bin/fusion start (Linux/Mac OS)
* bin\fusion.cmd start (Windows)

This will start the bundled Solr and Zookeeper instance, the Fusion API, the Admin UI, and the Connectors. They each run in their own Jetty instances and on their own ports (see the section [Changing the Default Ports](https://doc.lucidworks.com/fusion/2.1/Installation_and_Configuration/Installing_Lucidworks_Fusion/Changing-the-Default-Ports.html) for more on ports).

Starting Individual Services

To run the bundled ZooKeeper and Solr instances, execute:

* ./bin/solr start (Linux/Mac OS)
* bin\solr.cmd start (Windows)

which should make Solr available on <http://localhost:8983/solr/#/>

To run the Fusion administration UI, execute:

* ./bin/ui start (Linux/Mac OS)
* bin\ui.cmd start (Windows)

which will be available on <http://localhost:8764/>

To have the Fusion API served by Jetty, execute:

* ./bin/api start (Linux/Mac OS)
* bin\api.cmd start (Windows)

which will then be available on <http://localhost:8765/api>

To run the Connectors, execute:

* ./bin/connectors start (Linux/Mac OS)
* bin\connectors.cmd start (Windows)

which will be available on <http://localhost:8984/connectors/>

##### Running Services In The Foreground

To run the above service in the foreground, use the "run" command-line argument in place of "start".

##### Stopping Services

Each of the applications listed above has a corresponding "stop" invocation. For example, to stop all services that have been previously started:

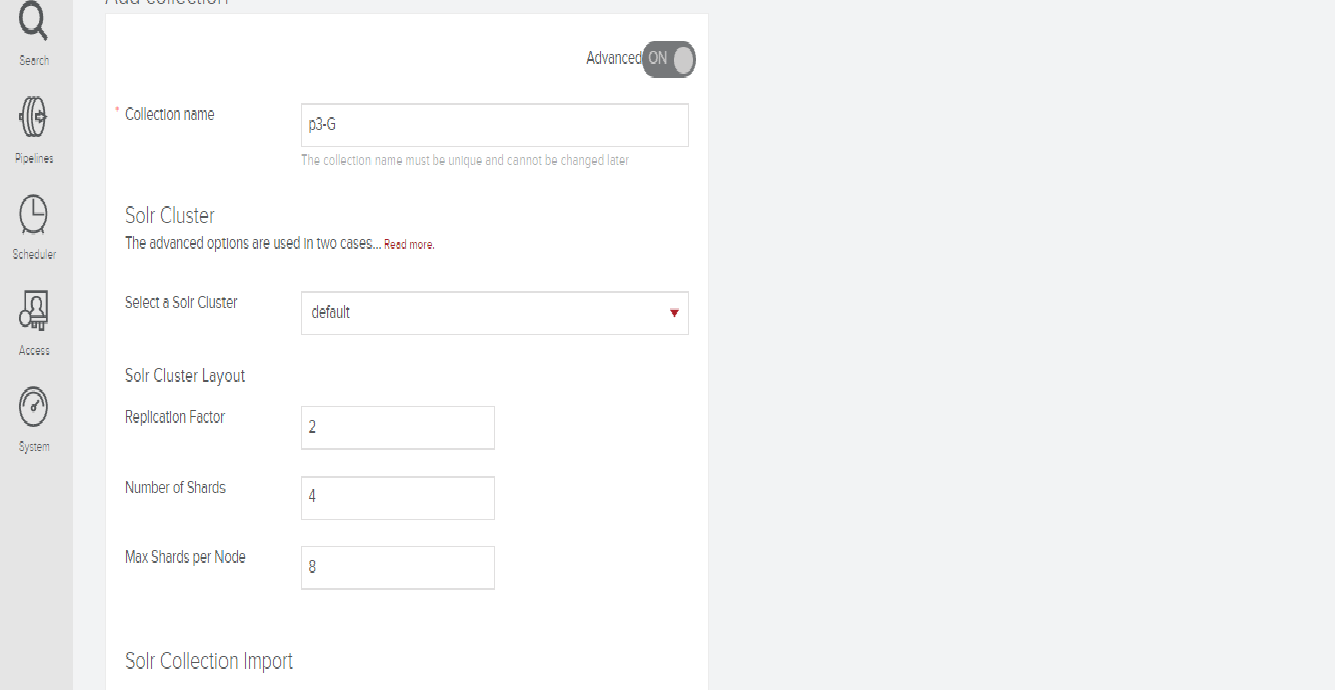
* -best practice –
  + stop Fusion
    - ./bin/fusion stop
  + stop Zookeeper
    - ./bin/zkServer.sh stop
  + start Zookeeper
    - ./bin/zkServer.sh start
  + start Fusion
    - ./bin/fusion start

#### Fusion Configuration Settings

* **Check if both Solr servers are running before creating the collection.**

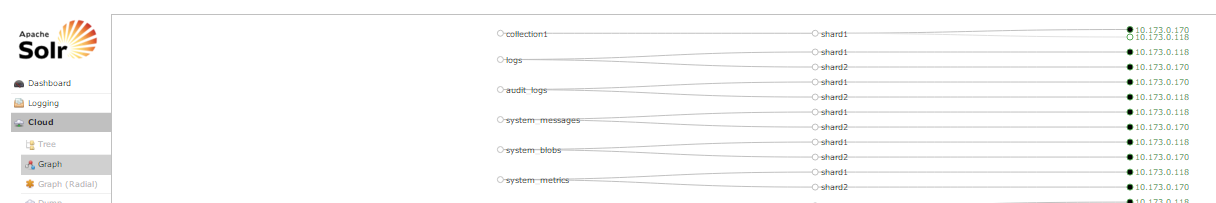
Initial collection set up below:

3.1 Create a collection called p3-G with 4 shards and 2 replicas

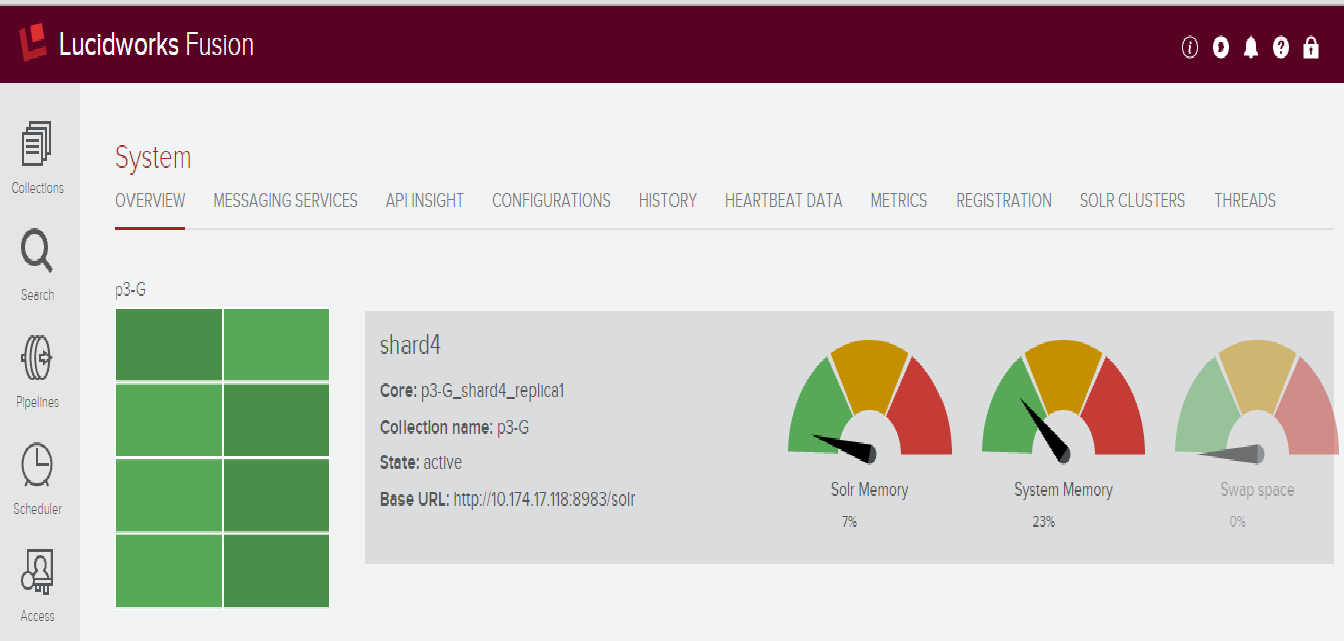


## Check the system if the collection is created appropriately.

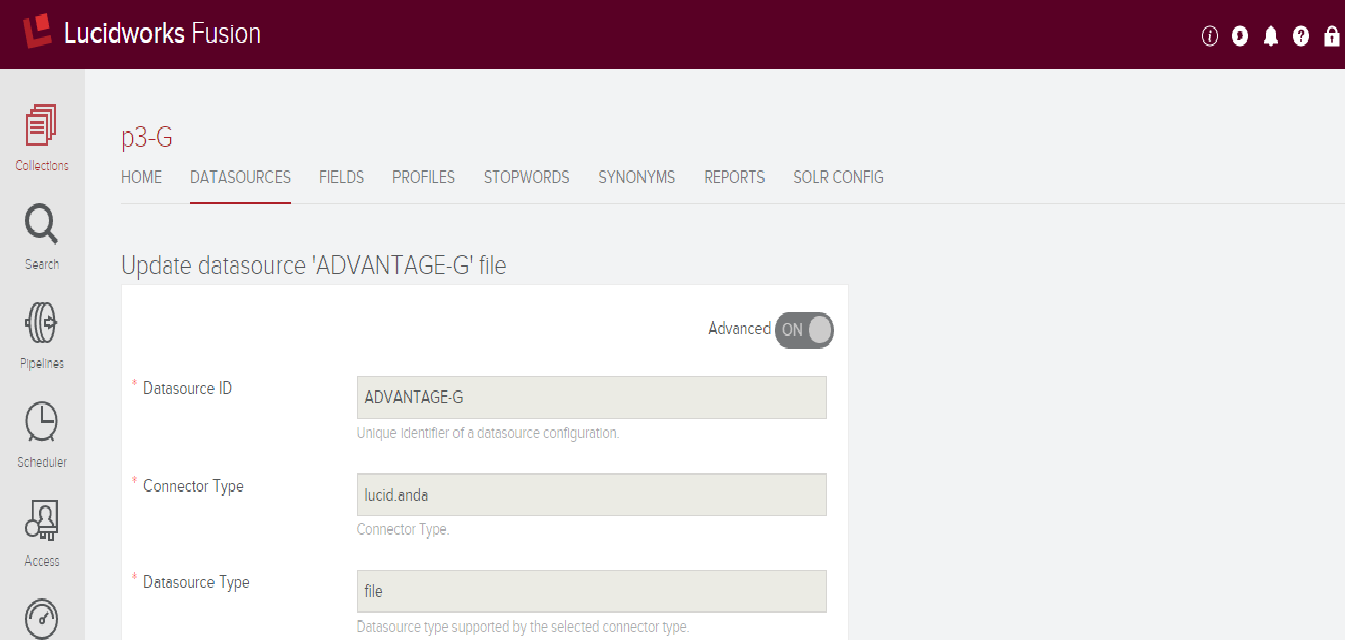
* Audit.log has 2 shards. (Check in Solr consul)

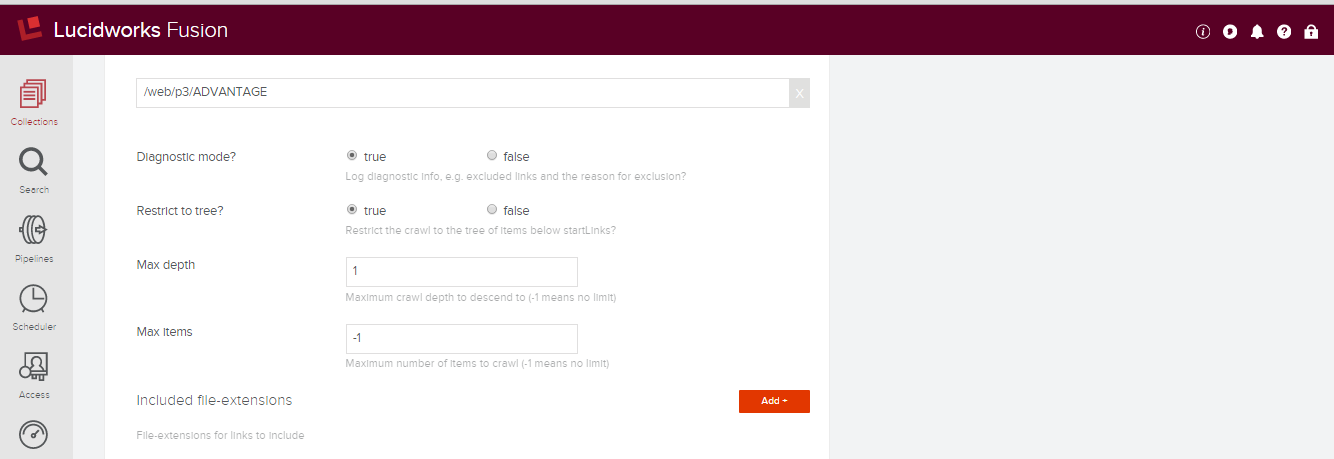


* Check if collection nodes are both up in fusion (see below)



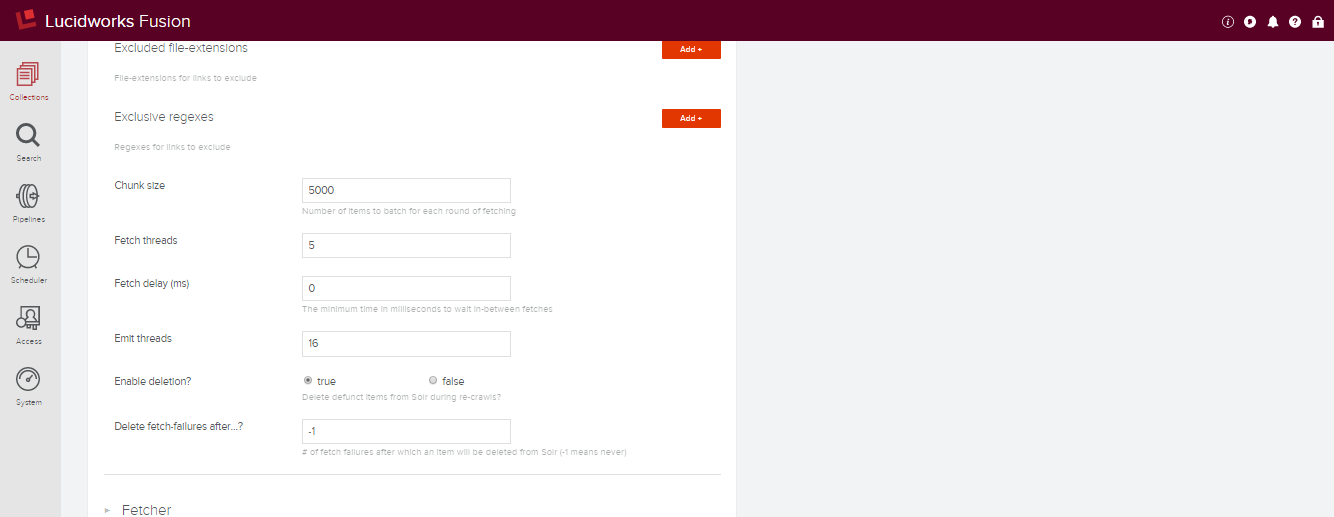
3.3 In datasource section, add a new datasource and configure the setup as below:



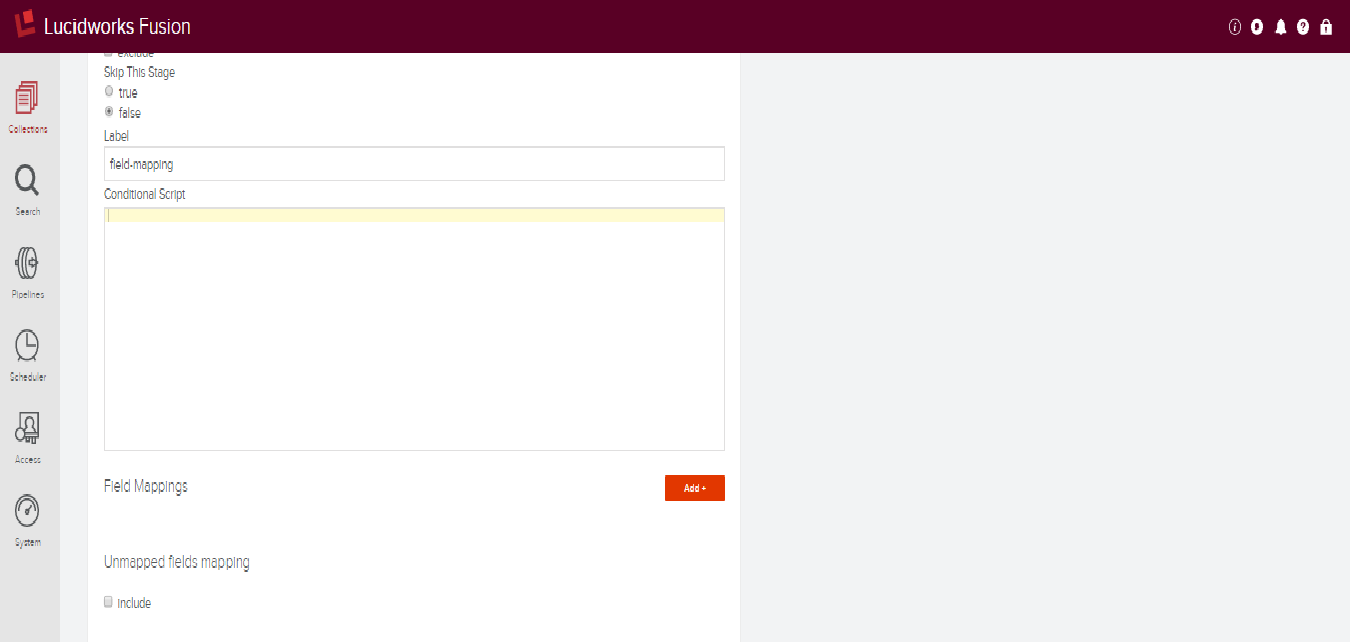


3.4 Based on the size of datasources, **chunk size** numbers can be modified:

* Less than 5mil documents in one datasource chunk size was setup to 5000
* More than 5mil documents in one datasource chunk size was setup to 15000

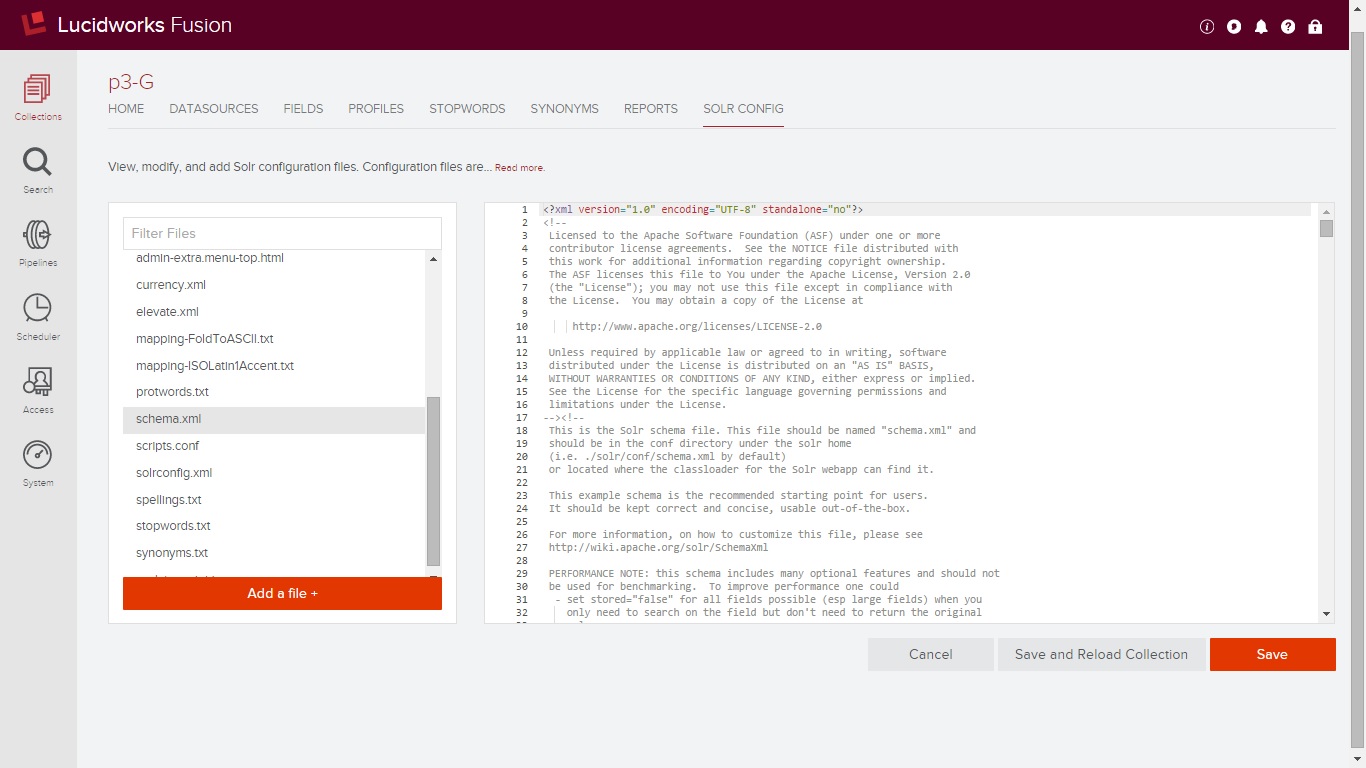


3.5 Remove all field mapping parameters:

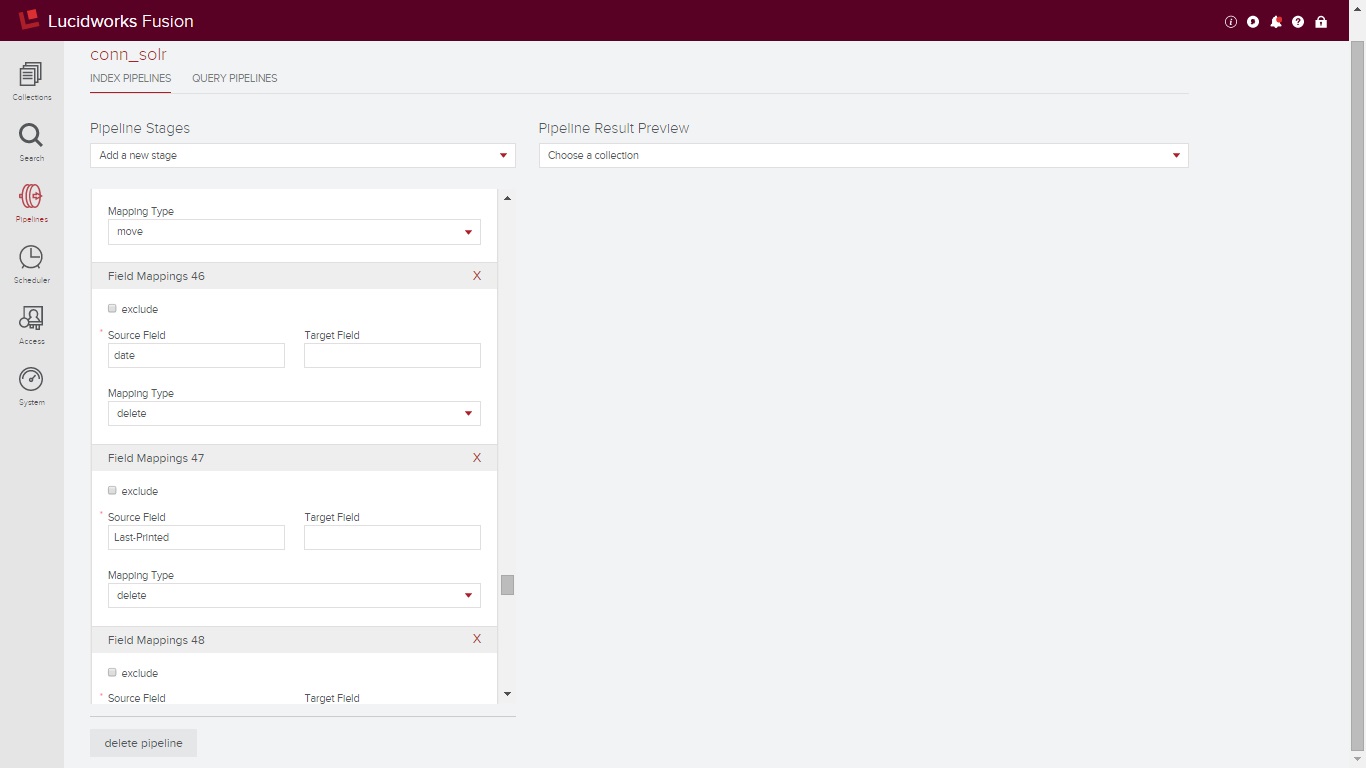
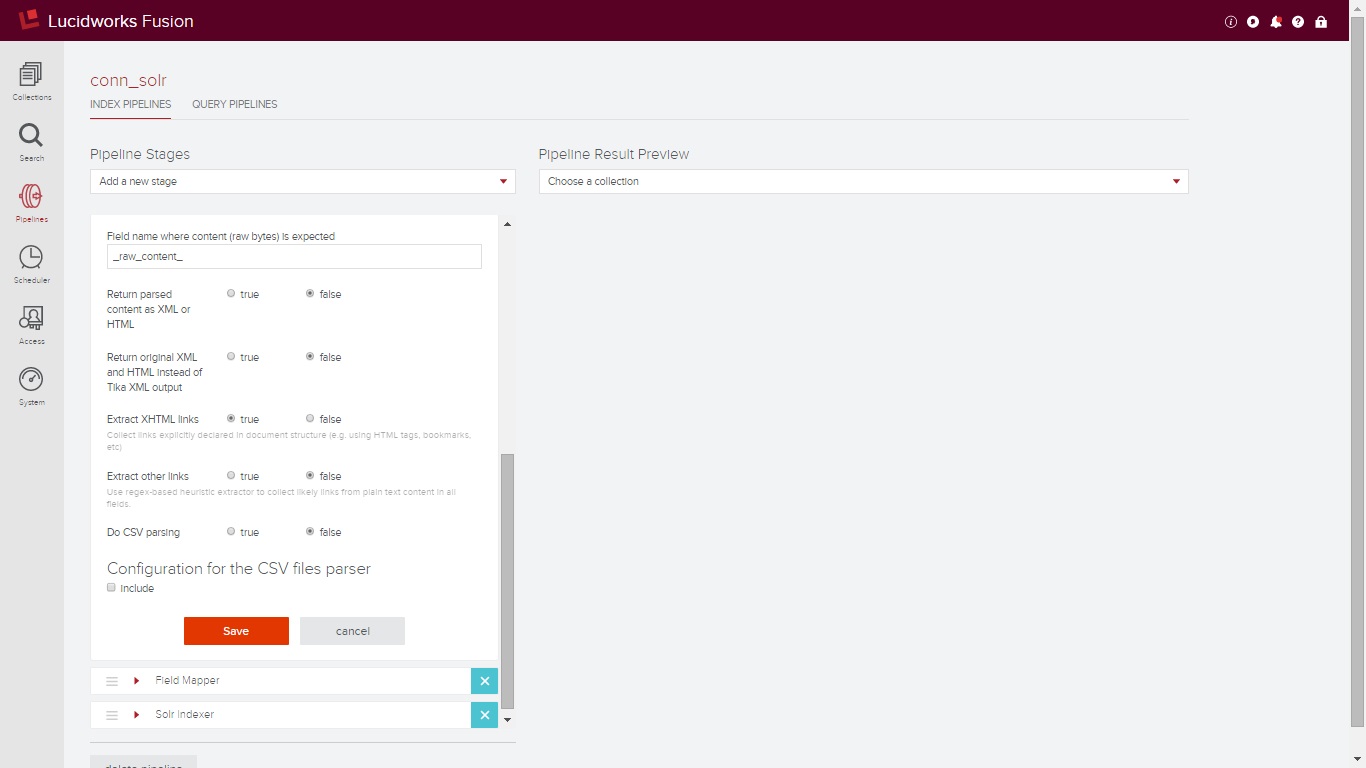
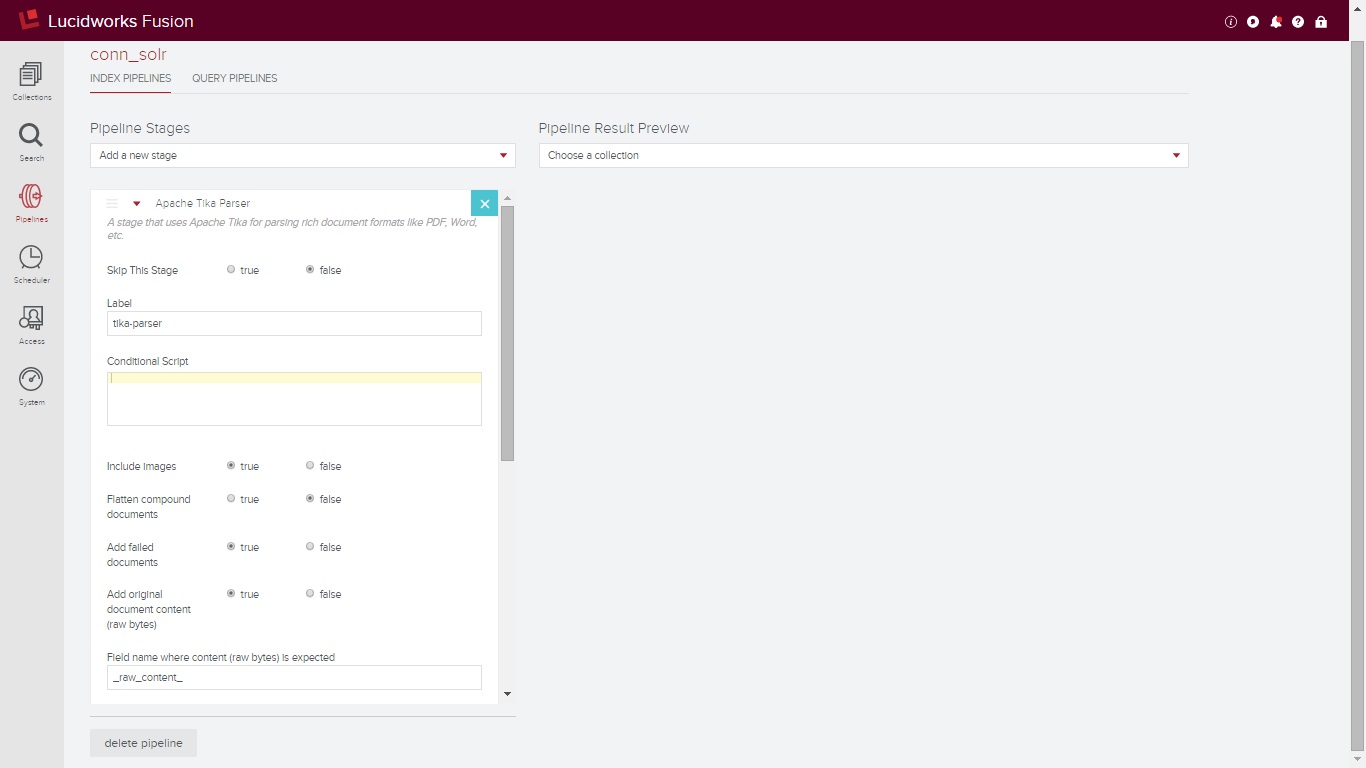


3.5 Import schema.xml

* click save and reload collection

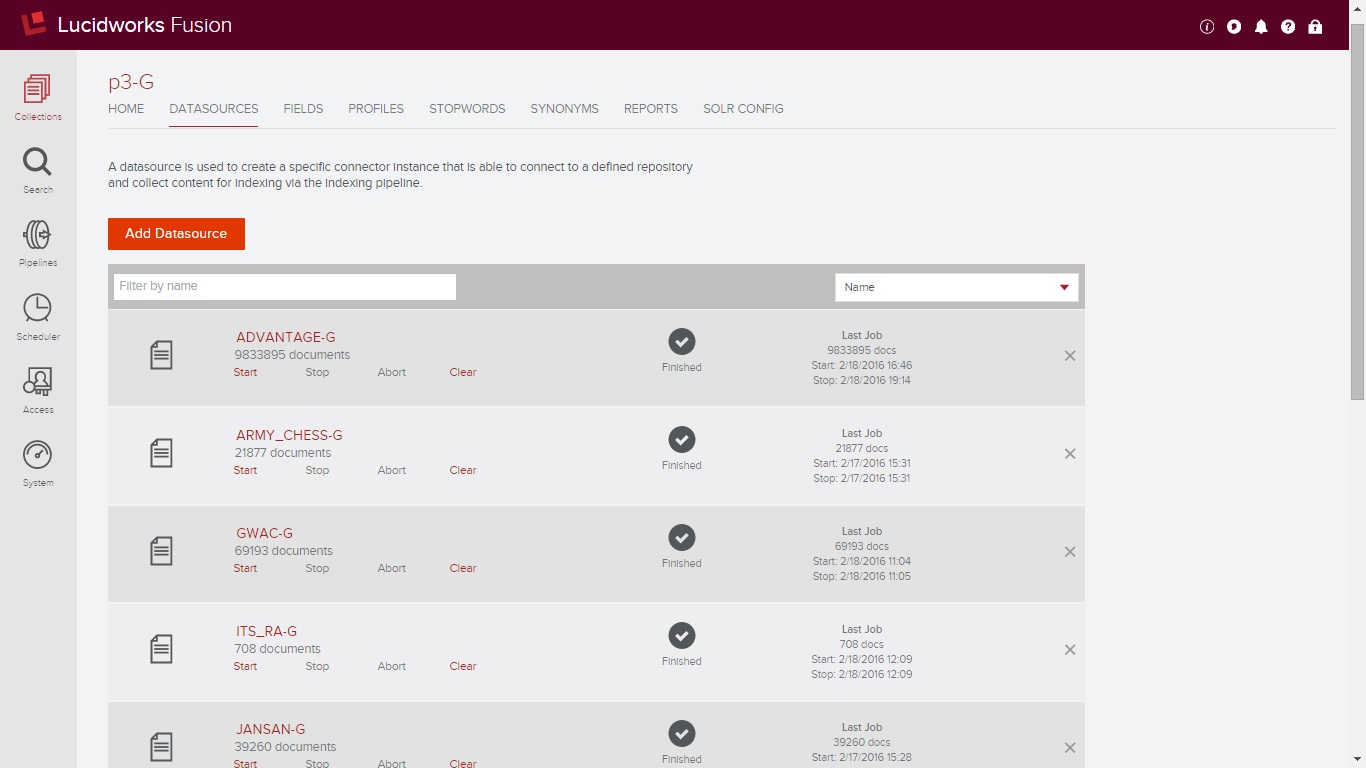


3.6. Configuring Indexing pipeline:



3.6 Start indexing, by click on start botton.

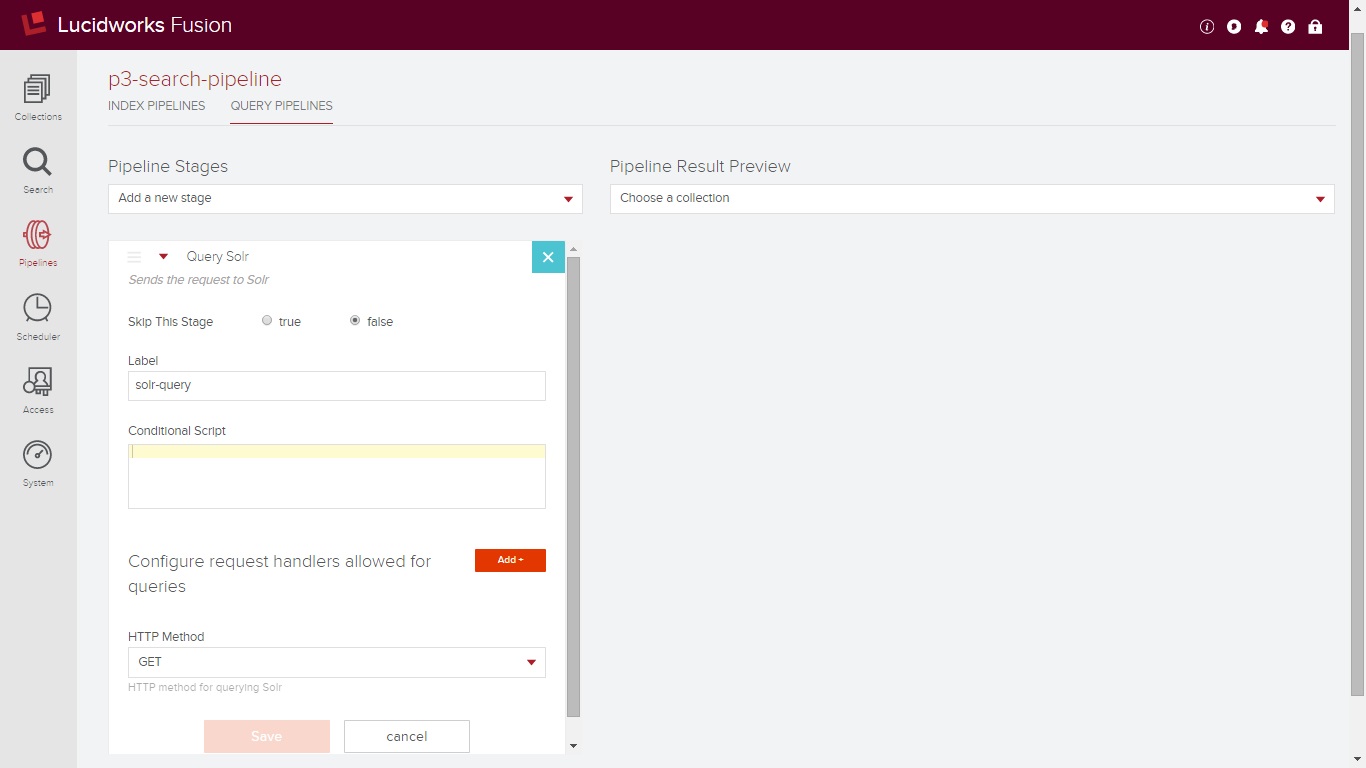
The best practice to reindex the datasource is to clear (click the clear botton) old-indexed first and then start new indexing.

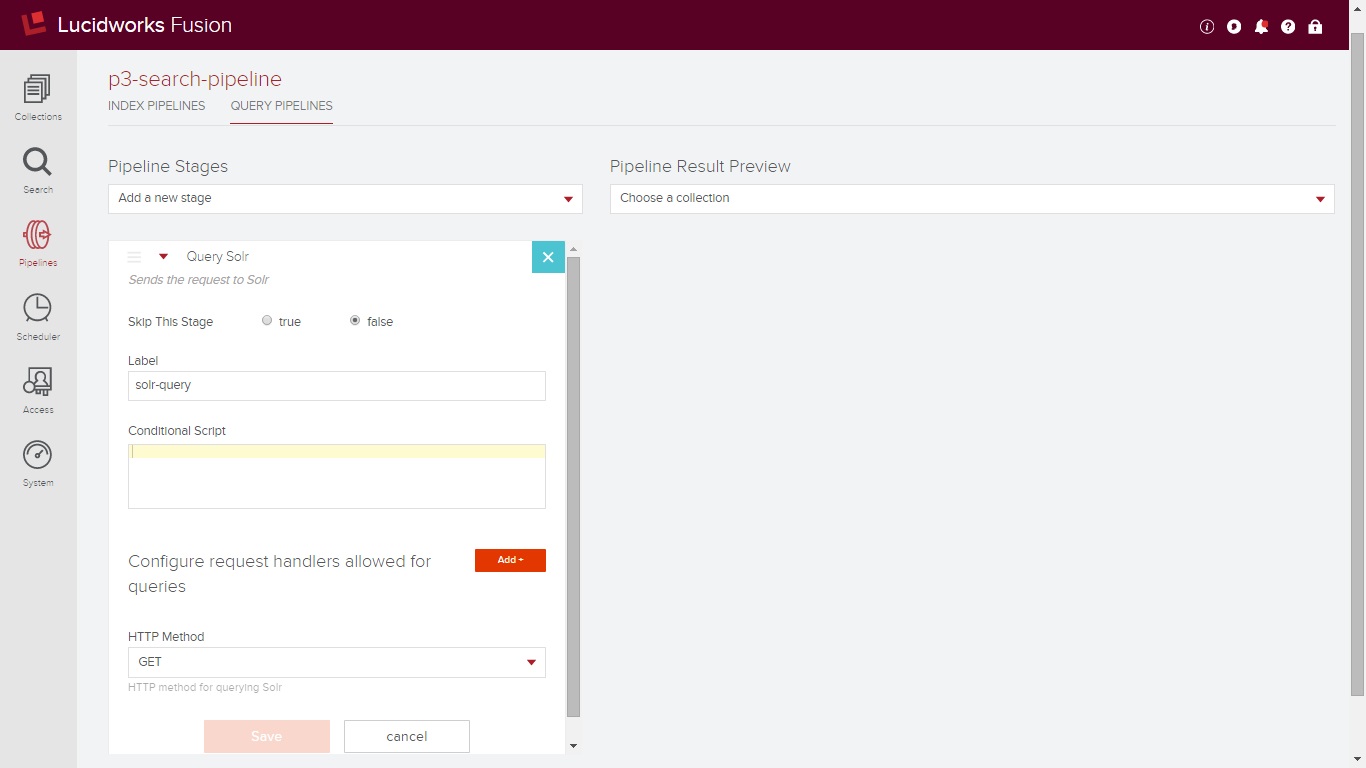


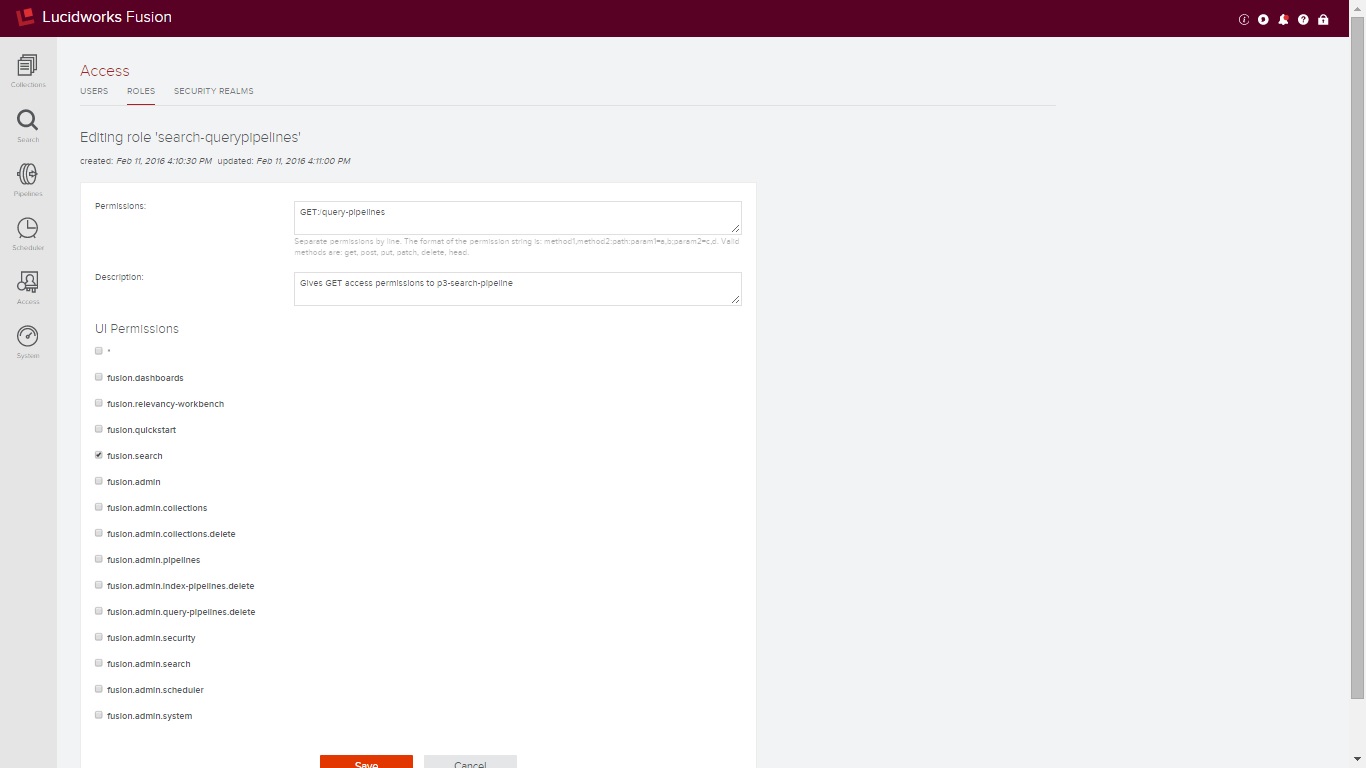
##### Onfiguring query pipeline: Add P3-user

\* **before connecting banana this must be created**

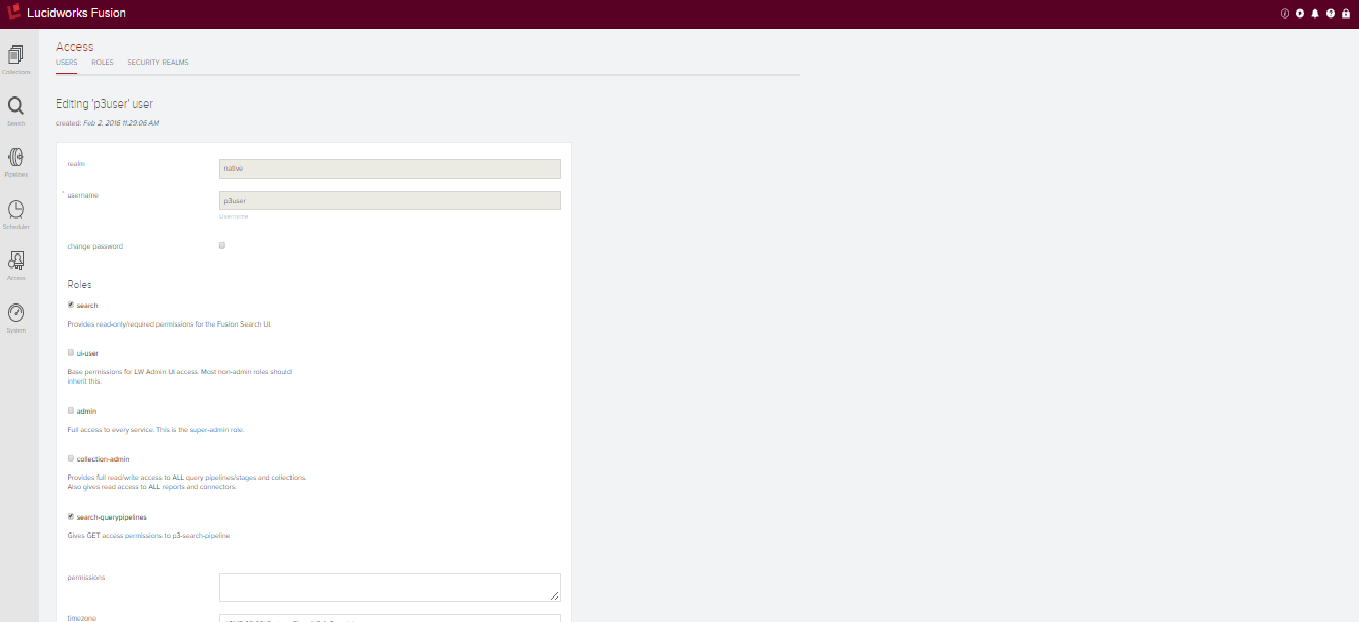
3.1.1 Create p3-search-pipeline



3.1.2 Add [search-querypipelines](http://96.127.66.114:8764/admin/roles/edit/9bb3c6e1-1c2f-4298-81ae-c724fd48167b) role in access section



3.1.3 Add p3user



### Install Banana

Installation and Quick Start

Requirements

A modern web browser. The latest version of Chrome and Firefox have been tested to work. Safari also works, except for the "Export to File" feature for saving dashboards. We recommend that you use Chrome or Firefox while building dashboards.

Solr 5 or 4.4+ (Solr server's endpoint must be open, or a proxy configured to allow access to it).

A webserver (optional).

Installation Options

Run Banana Web App within your existing Solr instance

Solr 5 Instructions

Run Solr at least once to create the webapp directory:

cd $SOLR\_HOME/bin/

./solr start

Copy banana folder to $SOLR\_HOME/server/solr-webapp/webapp/

Browse to http://localhost:8983/solr/banana/src/index.html

Solr 4 Instructions

Run Solr at least once to create the webapp directories:

cd $SOLR\_HOME/example

java -jar start.jar

Copy banana folder to $SOLR\_HOME/example/solr-webapp/webapp/

Browse to http://localhost:8983/solr/banana/src/index.html

NOTES: If your Solr server/port is different from localhost:8983, edit banana/src/config.js and banana/src/app/dashboards/default.json to enter the hostname and port that you are using. Remember that banana runs within the client browser, so provide a fully qualified domain name (FQDN), because the hostname and port number you provide should be resolvable from the client machines.

If you have not created the data collections and ingested data into Solr, you will see an error message saying "Collection not found at .." You can use any connector to get data into Solr. If you want to use LogStash, please go to the Solr Output Plug-in for LogStash Page for code, documentation and examples.

Complete SLK Stack

Lucidworks has packaged Solr, LogStash (with a Solr Output Plug-in), and Banana (the Solr port of Kibana), along with example collections and dashboards in order to rapidly enable proof-of-concepts and initial development/testing. See http://www.lucidworks.com/lucidworks-silk/.

Building and installing from a WAR file

Pull the source code of Banana version that you want from the release branch in the repo; For example, version 1.5.0 will be tagged as v1.5.0.

Run a command line "ant" from within the banana directory to build the war file:

cd $BANANA\_REPO\_HOME

ant

The war file will be called banana-<buildnumber>.war and will be located in $BANANA\_REPO\_HOME/build. Copy the war file and banana's jetty context file to Solr directories:

For Solr 5:

cp $BANANA\_REPO\_HOME/build/banana-<buildnumber>.war $SOLR\_HOME/server/webapps/banana.war

cp $BANANA\_REPO\_HOME/jetty-contexts/banana-context.xml $SOLR\_HOME/server/contexts/

For Solr 4:

cp $BANANA\_REPO\_HOME/build/banana-<buildnumber>.war $SOLR\_HOME/example/webapps/banana.war

cp $BANANA\_REPO\_HOME/jetty-contexts/banana-context.xml $SOLR\_HOME/example/contexts/

Run Solr:

For Solr 5:

cd $SOLR\_HOME/bin/

./solr start

For Solr 4:

cd $SOLR\_HOME/example/

java -jar start.jar

Browse to http://localhost:8983/banana (or the FQDN of your Solr server).

Banana Web App run in a WebServer

Banana is an AngularJS app and can be run in any webserver that has access to Solr. You will need to enable CORS on the Solr instances that you query, or configure a proxy that makes requests to banana and Solr as same-origin. We typically recommend the latter approach.

Storing Dashboards in Solr

If you want to save and load dashboards from Solr, create a collection using the configuration files provided in either the resources/banana-int-solr-4.4 (for Solr 4.4) directory or the resources/banana-int-solr-4.5 directory (for Solr 4.5 and above). If you are using SolrCloud, you will need to upload the configuration into ZooKeeper and then create the collection using that configuration.

The Solr server configured in config.js will serve as the default node for each dashboard; you can configure each dashboard to point to a different Solr endpoint as long as your webserver and Solr put out the correct CORS headers. See the README file under the resources/enable-cors directory for a guide.

### Need info

### Web server

* Install Apache Tomcat
  + prerequisite: java 1.8 to be installed
  + wget http://mirrors.sonic.net/apache/tomcat/tomcat-7/v7.0.61/bin/apache-tomcat-7.0.61.tar.gz
  + tar -xzpvf apache-tomcat-7.0.61.tar.gz
  + mv apache-tomcat-7.0.61 /usr/local/tomcat7
  + wget http://www.slf4j.org/dist/slf4j-1.7.12.tar.gz
  + cp slf4j-1.7.12/integration/lib/slf4j-\* /usr/local/tomcat7/lib
  + configure server.xml, web.xml, tomcat-users.xml (files to be versioned in Git)
  + cd /usr/local/tomcat7
  + ./startup.sh
  + http://localhost:80
  + copy p3.war into /usr/local/tomcat7/webapps and restart tomcat

### Building P3 code

mkdir /tmp/p3

cd /tmp/p3

git clone https://github.com/GSA/PricesPaidGui-AWS.git

cd PricesPaidGui-AWS

ant -Ddeploy.to="test" <target environment>

cp build/p3.war into /usr/local/tomcat7/webapps/p3.war

service tomcat restart

rm -rf /tmp/p3

### Automation with Jenkins

* Install latest git
* open all ports on the web server, for the Jenkins server: 96.127.63.172/32
* add user id jenkins on the web servers, and copy same keys on all servers
* set group ownership to jenkins on /usr/local/tomcat/webapps and chmod to 775
* set group ownership to jenkins on /usr/local/tomcat/webapps/p3.war and chmod to 660
* yum install ant
* yum install java-1.7.0-openjdk-devel.x86\_64
* setup jenkins slave node
* setup jenkins job

### Enable FIPS in Tomcat

Enable FIPS for Tomcat

sudo su -

cd /var/tmp

1. Change the following line in server.xml adding FIPSMode="on"

<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" FIPSMode="on"/>

2. install apr (Apache portable runtime)

wget http://www.interior-dsgn.com/apache/apr/apr-1.5.2.tar.gz

tar -xzpvf apr-1.5.2.tar.gz

cd apr-1.5.2

./configure

make

make install

3. install apr-util

wget http://apache.mirrors.pair.com//apr/apr-util-1.5.4.tar.gz

tar -xzpvf apr-util-1.5.4.tar.gz

cd apr-util-1.5.4

./configure --with-apr=/usr/local/apr

make

make install

4. Add this line to /etc/profile

export LD\_LIBRARY\_PATH='$LD\_LIBRARY\_PATH:/usr/local/apr/lib'​

5. install tomcat-native

wget http://mirror.sdunix.com/apache/tomcat/tomcat-connectors/native/1.1.33/source/tomcat-native-1.1.33-src.tar.gz

tar -xzpvf tomcat-native-1.1.33-src.tar.gz

cd tomcat-native-1.1.33-src/jni/native

JAVA\_HOME=/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86\_64

sudo ./configure --with-apr=/usr/local/apr --with-java-home=$JAVA\_HOME

make

make install

6. Adding the following line to catalina.sh

CATALINA\_OPTS="-Djava.library.path=/usr/local/apr/lib"

7. Intall openssl-fips

wget https://www.openssl.org/source/openssl-fips-2.0.9.tar.gz

tar -xzpvf openssl-fips-2.0.9.tar.gz

cd openssl-fips-2.0.9

./config

./config fips

make

make install

8. Install openssl-fips-ecp

wget https://www.openssl.org/source/openssl-fips-ecp-2.0.9.tar.gz

tar -xzpvf openssl-fips-ecp-2.0.9.tar.gz

cd openssl-fipps-ecp-2.0.0

./config

make

make install

9. install openssl

wget https://www.openssl.org/source/openssl-1.0.2-latest.tar.gz

cd openssl-1.0.2c/

./config fips --prefix=/usr/local --with-fipsdir=/usr/local/ssl/fips-2.0

make

make install

10. restart tomcat

### Password

All passwords used are created following below guidelines:

"An authentication scheme using passwords as a credential must implement the following security requirements:

(1) Passwords must contain a minimum of eight (8) characters which include a combination of letters, numbers, and special characters. Accounts used to access Federal Desktop Core Configurtion (USGB) compliant workstations (i.e. Windows XP and Windows Vista) must contain a minimum of sixteen (16) characters but do not have to contain a combination of letters, numbers, and special characters.

(2) Information systems must be designed to require passwords to be changed every 90 days.

(3) Information systems must automatically lockout users after not more than ten (10) failed access attempts during a 30-minute time period. Accounts must remain locked for a minimum of 30 minutes for the next login prompt.

(4) Passwords for Blackberry devices must be a minimum of 4 characters, but do not have to be a combination of letters, numbers, and special characters. Passwords for all other mobile devices such as GSA approved smart phones, iPads, and tablets must be a minimum of 6 characters but do not have to be a combination of letters, numbers, and special characters. The six-character password requirement also applies to personal mobile devices accessing GSA data or systems.

(5) Passwords must not be stored in forms (i.e. Windows dialog boxes, web forms, etc.).

(6) All default passwords on network devices, databases, operating systems, etc. must be changed.

(7) Other than default or one time use passwords, passwords must never be sent via email, regular mail, or interoffice mail.

(8) User IDs and passwords must never be distributed together (i.e. same e-mail, regular mail, interoffice mail, etc.).

(9) Users must be authenticated before resetting or distributing a password.

# SSL Certificates

* Make a certificate request (send it to Carter Brian)
* Upload certificate and certificate key into load balancer
* Wild card certificates (\*.cap.gsa.gov) were used for dev & test. Prod certificate will be moved from P3 over to GovCloud

# DNS Names

Submit ServiceNow ticket to Jerod Weaver to create CNAME to following DNS names:

Must state if the dns will be external or internal accessible.

<https://p3-dev.cap.gsa.gov> p3-dev-gsaE-elb-ext-1597058048.us-gov-west-1.elb.amazonaws.com

[https://p3-test.cap.gsa.gov](https://p3-dev.cap.gsa.gov) p3-test-gsaE-elb-ext-272371029.us-gov-west-1.elb.amazonaws.com

[https://p3.cap.gsa.gov](https://p3-dev.cap.gsa.gov) p3-prod-gsaE-elb-ext-963029994.us-gov-west-1.elb.amazonaws.com

**De-Installation Instructions**

**T**his section will provide **step-by-step** procedures for accomplishing the de-installation. References may be made to other documents, such as operator manuals. Safety precautions, marked by WARNING or CAUTION, shall be included where applicable.

*For Example:*

Listed below are the instructions for de-installing <the software on the client machine.> <the following software onto the server>

Make sure you use the correct de-installation procedures or the software may not be de-installed correctly. Most, if not all, of these procedures will be conducted by the site ISSO or ISM. If you have questions about the procedures in this chapter, contact your site system administrator.

Click Start > Settings > Control Panel. The Control Panel will appear.

Click the Add/Remove Programs icon. The Add/Remove Programs dialog will appear (below).

**Table 8: De-installation Instructions**

|  |  |  |
| --- | --- | --- |
| **Step** | **Task** | **Results** |
| Note 1 | These procedures assume that the Kernel has already been installed on your system. |  |
| Note 2 | You will be required to log in with administrator privileges in order to de-install |  |
| 1 | Log on to the system as the system administrator. |  |
| 2 | Open the Installer. This is done by clicking Start > Programs > System Administration > Installer. |  |
| 3 | Select from the list of Installed Segments **so** that it is highlighted. |  |
| **4** | Select the De-Install button. After a few moments, the program will be removed from your system. |  |
| 5 | Exit the Installer. |  |

## De-Installation Instructions For Upgrades

This section will provide step-by-step procedures for accomplishing the de-installation of upgrades. References may be made to other documents, such as operator manuals. Safety precautions, marked by WARNING or CAUTION, shall be included where applicable.

*For Example:*

Listed below are the instructions for de-installing the following upgrades to the P3 System

**Table 9: De-installation for Upgrades**

|  |  |  |
| --- | --- | --- |
| **Step** | **Task** | **Results** |
| Note 1 | These procedures assume that the Kernel has already been installed on your system. |  |
| Note 2 | You will be required to log in with administrator privileges in order to de-install |  |
| 1 | Log on to the system as the system administrator. |  |
| 2 | Open the Installer. This is done by clicking Start > Programs > System Administration > Installer. |  |
| 3 | Select from the list of Installed Segments so that it is highlighted. |  |
| 4 | Select the De-Install button. After a few moments, the program will be removed from your system. |  |
| 5 | Exit the Installer. |  |

### Git Upgrade

Upgraded default RedHat version of Git 1.7.1 to Git 1.8.2.3 on test-02 server.

|  |  |  |
| --- | --- | --- |
| **Step** | **Task** | **Results** |
| 1 | Sudo yum remove git |  |
| 2 | cd /usr/local/src |  |
| 3 | wget https://www.kernel.org/pub/software/scm/git/git-1.8.2.3.tar.gz |  |
| 4 | tar xzvf git-1.8.2.3.tar.gz |  |
| 5 | cd git-1.8.2.3 |  |
| 6 | make prefix=/usr/local all |  |
| 7 | make prefix=/usr/local install |  |
| 8 | git --version |  |
| 9 | git version 1.8.2.3 |  |

# Frequently Asked Questions

This section presents to the user frequents questions about the installation and troubleshooting of the system/program and the common answers to those questions.

# References

This section contains all reference material used in the creation of the system installation plan or required for the operation of the system or program. It includes websites, system documents, references to other systems and their documents and any other document that was either used in the creation of the manual, referenced in the manual or provides addition information useful to the user. A sample reference table is included below for your use or if you prefer you can delete the table and use your own format.

**Table 10: References**

|  |  |  |
| --- | --- | --- |
| ***Document Number*** | ***Rev.*** | ***Document Title*** |
| *LS-71062* | *C* | *Software Requirements Specification (SRS)* |
| *LS-71062-8* | *D* | *Interface Design Document* |

###### Acronyms and Definitions

**Table 12: Acronyms and Definitions**

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| P3 | Prices Paid Portal |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |