XL Deploy (released in 2009), used for application deployment automation and environment management;

XL Release (released in 2013), used for orchestrating, automating and providing visibility into the release pipeline

XebiaLabs' products have clean, easy-to-use interfaces

WORK FLOW BASED

MODEL-BASED(CL DEPLOY) 5 UNIQUE FEATURES

[XL Deploy](https://xebialabs.com/products/xl-deploy/) is the only application release automation solution that is agentless across all target platforms. Connect to Windows and Unix target systems using proven, industry-standard remote protocols; no agent installation and maintenance, no overhead on the target systems, no firewall ports to be opened, and no security reviews.

**System Requirements:**

**Operating system:** Microsoft Windows (32-bit or 64-bit) or a Unix-family operating system running Java

* **Oracle, IBM, or Apple Java Development Kit (JDK):**
  + For XL Deploy 5.1.0 and earlier: JDK 7
  + For XL Deploy 5.1.1 and later: JDK 7 or 8
  + For XL Deploy 5.5.0 and later: JDK 8

**Warning:** It is a known issue that using XL Deploy 5.1.1, 5.1.2, or 5.1.3 with JDK 8 may prevent pending tasks from being recovered after the XL Deploy server is restarted. To prevent this issue, use JDK 7 with these versions. Note that XL Deploy 5.1.0 and earlier should not be used with JDK 8.

* **RAM:** At least 2 GB of RAM available for XL Deploy
* **Hard disk space:** Sufficient hard disk space to store the XL Deploy repository

The XL Deploy server itself only uses about 70 MB of disk space. The main hard disk space usage comes from the repository which stores your deployment packages and deployment history. The size of the repository will vary from installation to installation, but depends mainly on the:

* Size and storage mechanism used for artifacts
* Number of packages in the system
* Number of deployments performed (specifically, the amount of logging information stored)

While it is possible to store the repository in an [external database](https://docs.xebialabs.com/xl-deploy/how-to/configure-the-xl-deploy-repository.html), it should not be stored on NFS. Also, XL Deploy always requires that the disk space for the server be persistent. This is important for several reasons:

* Lucene indexes are stored in the repository directory; if the disk space is not persistent, these indexes will be rebuilt each time the server starts, which is very time-consuming
* Configuration files such as deployit.conf and deployit-defaults.properties are updated by the running system
* Log files are also updated by the running system (unless configured otherwise)

Estimating required disk space

Follow this procedure to obtain an estimate of the total required disk space:

1. Install and configure XL Deploy for your environment as described in this document. Make sure you correctly set up the database- or file-based repository.
2. Estimate the number of packages to be imported, either the total number or the number per unit of time (NumPackages)
3. Estimate the number of deployments to be performed, either the total number or the number per unit of time (NumDeployments)
4. Record the amount of disk space used by XL Deploy (InitialSize)
5. Import a few packages using the GUI or CLI
6. Record the amount of disk space used by XL Deploy (SizeAfterImport)
7. Perform a few deployments
8. Record the amount of disk space used by XL Deploy (SizeAfterDeployments)

The needed amount of disk space in total is equal to:

Space Needed = ((SizeAfterImport - InitialSize) \* NumPackages) +

((SizeAfterDeployments - SizeAfterImport) \* NumDeployments)

If NumPackages and NumDeployments are expressed per time unit (that is, the number of packages to be imported per month), then the end result represents the space needed per time unit as well.

Depending on the environment, the following may also be required:

* **Database:** XL Deploy’s Jackrabbit repository supports a number of different databases; for more information, see [Configure the XL Deploy repository](https://docs.xebialabs.com/xl-deploy/how-to/configure-the-xl-deploy-repository.html)
* **LDAP:** To enable group-based security, an LDAP x.509 compliant registry is needed; for more information, see [Configure the XL Deploy security file](https://docs.xebialabs.com/xl-deploy/how-to/configure-the-xl-deploy-security-file.html)

XL Deploy uses a repository to store all of its data such as configuration items (CIs), deployment packages, logging, etc. XL Deploy can use the filesystem or a database for binary artifacts (deployment packages) and CIs and CI history. By default, XL Deploy uses the filesystem to store all data in the repository.

## **Location of the repository**

By default, the repository is located in XL\_DEPLOY\_SERVER\_HOME/repository. To change the location, change the value of jcr.repository.path in XL\_DEPLOY\_SERVER\_HOME/conf/deployit.conf. For example:

jcr.repository.path=file://opt/xldeploy/repository

## **Using a database**

XL Deploy can also use a database to store its repository. To use a database, you must configure the built-in Jackrabbit JCR implementation, depending on what you want to store in the database:

| **Type of data to store in the database** | **Properties to configure** |
| --- | --- |
| Only binary artifacts | DataStore |
| Only CIs and CI history | PersistenceManager and FileSystem |
| All data (binary artifacts and CIs and CI history) | DataStore, PersistenceManager and FileSystem |

**Important:** XL Deploy must initialize the repository before it can be used. Run XL Deploy’s setup wizard and initialize the repository after making any changes to the repository configuration.

To use XL Deploy with [MySQL](http://www.mysql.com/), ensure that the [JDBC driver for MySQL](http://dev.mysql.com/downloads/connector/j/) JAR file is located in XL\_DEPLOY\_SERVER\_HOME/lib or on the Java classpath.

**LDAP:** To enable group-based security, an LDAP x.509 compliant registry is needed; for more information, see [Configure the XL Deploy security file](https://docs.xebialabs.com/xl-deploy/how-to/configure-the-xl-deploy-security-file.html)

### **Networking requirements**

Before installing XL Deploy, ensure that the network connection to the XL Deploy host name works. You should be able to successfully execute ping xl\_deploy\_hostname.

By default, the XL Deploy server uses port 4516. If, during installation, you choose to enable secure communication (SSL) between the server and the XL Deploy GUI, the server uses port 4517.

To enable secure communication and/or to change the port number during installation, choose the [manual setup option](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#manual-setup) in the command-line server setup wizard. The wizard will take you through the setup steps.

## **Client requirements**

### **GUI clients**

To use the XL Deploy GUI, you must meet the following requirements:

* **Web browser:**
  + Firefox
  + Chrome
  + Safari
  + For XL Deploy 5.0.0 or later: Internet Explorer 10 or later
  + For XL Deploy 4.5.x: Internet Explorer 8, 9, or later (Internet Explorer 10 or later if using the [Compare feature](https://docs.xebialabs.com/xl-deploy/how-to/compare-configuration-items.html))
* **Flash player:** Version 9 or later

**Note:** Internet Explorer Compatibility View is not supported.

### **CLI clients**

To use the XL Deploy CLI, you must meet the following requirements:

* **Operating system:** Microsoft Windows or Unix-family operating system running Java
* **Java Runtime Environment:** The same Java Development Kit (JDK) version as your version of XL Deploy

## **Middleware server requirements**

### **Windows middleware server requirements**

Windows-based middleware servers that XL Deploy interacts with must meet the following requirements:

* **File system access:** The target file system should be accessible via CIFS from the XL Deploy server
* **Host access:** The target host should be accessible from the XL Deploy server via WinRM or Windows Telnet server running in stream mode
* **Directory shares:** The account used to access a target system should have access to the host’s administrative shares such as C$
* **Ports:**
  + For CIFS connectivity, port 445 on the target system should be accessible from the XL Deploy server
  + For Telnet connectivity, port 23 should be accessible from the XL Deploy server
  + For WinRM connectivity, port 5985 (HTTP) or port 5986 (HTTPS) should be accessible from the XL Deploy server

### **Extending middleware support**

It is possible to connect XL Deploy to middleware servers that do not support SSH, Telnet, or WinRM. This requires you to use the [Overthere](https://github.com/xebialabs/overthere) remote execution framework to create a custom access method that connects to the server.

[Edit this page](https://github.com/xebialabs/online-docs-jekyll/edit/master/xl-deploy/concept/requirements-for-installing-xl-deploy.markdown)

**Installation of XL deploy:**

To install XL Deploy:

1. [Ensure that you meet the system requirements.](https://docs.xebialabs.com/xl-deploy/concept/requirements-for-installing-xl-deploy.html)
2. Download the XL Deploy server archive (ZIP file) from the [XebiaLabs Software Distribution site](https://dist.xebialabs.com/) (requires customer log-in) or from the link provided when you sign up for a [free trial](https://xebialabs.com/products/xl-deploy/trial/).
3. [Extract the XL Deploy server archive.](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#extract-the-xl-deploy-server-archive)
4. [Install the license.](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#install-the-license)
5. [Run the server setup wizard](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#run-the-server-setup-wizard) and select whether to perform a [simple setup](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#simple-setup) or [manual setup](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#manual-setup).
6. [Finish the setup process and log in to XL Deploy.](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#finish-the-setup-process-and-log-in)
7. [Optionally install the XL Deploy command-line interface (CLI).](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#install-the-xl-deploy-cli)

**Tip:** For information about installing XL Deploy as a daemon or service, refer to [Install XL Deploy as a service](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy-as-a-service.html). If you are upgrading a previously installed version of XL Deploy, refer to [Upgrade XL Deploy](https://docs.xebialabs.com/xl-deploy/how-to/upgrade-xl-deploy.html).

## **Extract the XL Deploy server archive**

First, extract the XL Deploy server archive:

1. Log in to the computer where you want to install XL Deploy.
2. Create an installation directory such as /opt/xebialabs/xl-deploy or C:\Program Files\XL Deploy.
3. Copy the XL Deploy server archive to the directory.
4. Extract the archive in the directory.

**Tip:** It is recommended to install the XL Deploy server as a non-root user such as xldeploy.

## **Install the license**

If you received an XL Deploy license key by email, you will be prompted to enter it after you install and start XL Deploy.

If you do not have an XL Deploy license key, you can download a license file (deployit-license.lic) from the [XebiaLabs Software Distribution site](https://dist.xebialabs.com/) (requires customer log-in). Place the license file in the conf directory before you install XL Deploy. Be sure that you do not modify the license file in any way.

Refer to [XL Deploy licensing](https://docs.xebialabs.com/xl-deploy/concept/xl-deploy-licensing.html) for information about how the XL Deploy license works.

## **Run the server setup wizard**

To install the XL Deploy server, go to a command line or terminal window and execute the appropriate command to start the setup wizard:

| **Operating system** | **XL Deploy version** | **Command** |
| --- | --- | --- |
| Microsoft Windows | XL Deploy 4.5.x or earlier | server.cmd -setup |
| Microsoft Windows | XL Deploy 5.0.0 or later | run.cmd -setup |
| Unix-based systems | XL Deploy 4.5.x or earlier | server.sh -setup |
| Unix-based systems | XL Deploy 5.0.0 or later | run.sh -setup |

To stop the setup wizard at any time, enter exitsetup. All changes to the configuration will be discarded.

### **Select a setup method**

The setup wizard shows the welcome message. You can:

* Answer yes (or press ENTER) to install XL Deploy with a default configuration. This method makes it easy for you to quickly get started with XL Deploy. See [Simple setup](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#simple-setup) for more information.
* Answer no to install XL Deploy manually. This method gives you explicit control over all XL Deploy settings. See [Manual setup](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#manual-setup) for more information.

**Note**: If you have installed XL Deploy in the same location before, the setup wizard will ask you whether you want to edit the existing configuration or create a new one. Answer yes (or press ENTER) to edit the existing configuration. The setup wizard will load all settings from the existing configuration and allow you to choose simple or manual setup. Answer no to start over with an empty configuration.

### **Simple setup**

If you choose the simple setup, XL Deploy will be installed with these settings:

* The server will run with security enabled.
* The server will not use secure communication between the XL Deploy graphical user interface (GUI) and the XL Deploy server.
* The server will listen on XL Deploy’s standard HTTP port (4516).
* The server will use / as the context root.
* The server will use a minimum of 3 and a maximum of 24 threads.
* Applications can be imported from the importablePackages directory.

#### Step 1 Provide a password for the admin user

The setup wizard will promote you to provide a password for the admin user. The admin user has all permissions and is used to connect to XL Deploy’s JCR repository.

To use the default password of admin, press ENTER twice. If you plan to connect to an existing repository, enter the password that you already use to connect to that repository.

#### Step 2 Initialize the repository

The setup wizard will ask if you want to initialize the repository. Answer yes to create the repository, or no to connect to an existing repository.

**Warning**: If you choose to initialize the repository and you have installed XL Deploy in the same location before, any information stored in the repository will be lost.

#### Step 3 Generate an encryption key

If you choose to initialize the repository, the setup wizard will ask if you want to generate an encryption key to protect passwords that you store in the repository. Answer yes to generate a new key.

Answering yes will create an encryption key in XL\_DEPLOY\_SERVER\_HOME/conf/repository-keystore.jceks. XL Deploy will use this key to encrypt and decrypt all password properties; it will also be used for [encrypted entries in dictionaries](https://docs.xebialabs.com/xl-deploy/how-to/create-a-dictionary.html) and passwords that are set in the XL\_DEPLOY\_SERVER\_HOME/conf/deployit.conf file.

If you answer no, then you agree to use XL Deploy’s own encryption key, or to use a key that you have previously generated.

#### Step 4 Provide a password for the encryption key

If you choose to generate an encryption key, you can also provide a password to secure the key. You will be required to enter this password when XL Deploy starts, either:

* Interactively via a prompt
* Non-interactively via a command-line parameter
* Non-interactively via a configuration file

If you do not want to provide a password, press ENTER twice.

See [Finish the setup process and log in](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#finish-the-setup-process-and-log-in) to complete the setup process.

### **Manual setup**

Manual setup gives you control over all of XL Deploy’s installation settings.

#### Step 1 Provide a password for the admin user

The setup wizard will promote you to provide a password for the admin user. The admin user has all permissions and is used to connect to XL Deploy’s JCR repository.

To use the default password of admin, press ENTER twice. If you plan to connect to an existing repository, enter the password that you already use to connect to that repository.

#### Step 2 Configure secure communication

The setup wizard will prompt you to set up secure communication (SSL) between the XL Deploy graphical user interface (GUI) and the XL Deploy server.

#### Step 3 Generate a self-signed certificate

A digital certificate is required for secure communication; normally, certificates are signed by a Certificate Authority (CA). However, if you choose secure communication between the GUI and the server, the setup wizard will ask if you want XL Deploy to generate a self-signed digital certificate.

**Important:** For security reasons, a self-signed certificate is not recommended for production environments. It may trigger security warnings in some browsers and Flash players. A self-signed certificate can only be used when you access the XL Deploy GUI at https://localhost:4516.

#### Step 4 Use your own keystore

Instead of using a self-signed digital certificate, you can use your own keystore for secure communication between the GUI and the server. XL Deploy’s built-in Jetty server requires a certificate with the name jetty to be present in the keystore.

The setup wizard will ask you for the keystore path (for example, mykeystore.jks), the keystore password, and the password of the jetty certificate in the keystore.

#### Step 5 Enable mutual SSL

The setup wizard will ask if you want to enable mutual SSL. If you answer yes, it will prompt you for the location and password to your truststore.

#### Step 6 Set up the HTTP configuration

The setup wizard will ask the HTTP bind address, HTTP port number, and web context root where you would like XL Deploy to run.

**Note**: If you chose to enable secure communication, the default port will be 4517 instead of 4516.

#### Step 7 Set up the thread configuration

The setup wizard will prompt you for the minimum and maximum number of threads that the XL Deploy server should use to handle incoming connections.

#### Step 8 Configure the repository

The setup wizard will ask where you want to store the JCR repository. If the directory does not exist, XL Deploy will create it. The setup wizard will also ask if you want to initialize the repository.

**Warning**: If you choose to initialize the repository and you have installed XL Deploy in the same location before, any information stored in the repository will be lost.

#### Step 9 Generate an encryption key

If you choose to initialize the repository, the setup wizard will ask if you want to generate an encryption key to protect passwords that you store in the repository. Answer yes to generate a new key.

Answering yes will create an encryption key in XL\_DEPLOY\_SERVER\_HOME/conf/repository-keystore.jceks. XL Deploy will use this key to encrypt and decrypt all password properties; it will also be used for [encrypted entries in dictionaries](https://docs.xebialabs.com/xl-deploy/how-to/create-a-dictionary.html) and passwords that are set in the XL\_DEPLOY\_SERVER\_HOME/conf/deployit.conf file.

If you answer no, then you agree to use XL Deploy’s own encryption key, or to use a key that you have previously generated.

#### Step 10 Provide a password for the encryption key

If you choose to generate an encryption key, you can also provide a password to secure the key. You will be required to enter this password when XL Deploy starts, either:

* Interactively via a prompt
* Non-interactively via a command-line parameter
* Non-interactively via a configuration file

If you do not want to provide a password, press ENTER twice.

See [Finish the setup process and log in](https://docs.xebialabs.com/xl-deploy/how-to/install-xl-deploy.html#finish-the-setup-process-and-log-in) to complete the setup process.

#### Step 11 Configure the location for importable packages

By default, you can import deployment packages from the importablePackages directory. If you would like to change this, enter a path when the setup wizard prompts you. If the directory does not exist, XL Deploy will create it.

## **Finish the setup process and log in**

After you have configured all options, the setup wizard shows a summary of the configuration that you have selected. Answer yes to finish the setup process. Answer no to exit setup.

If you answer yes, the setup wizard will start XL Deploy and show the URL where you can access it; for example, http://localhost:4516. Open this URL in a browser and log in with the username admin and the password that you provided during the setup process.

**Tip:** For information about starting XL Deploy in the future, refer to [Start XL Deploy](https://docs.xebialabs.com/xl-deploy/how-to/start-xl-deploy.html).

## **XL Deploy server directory structure**

After you install XL Deploy, it will have the following directory structure:

* bin: Contains the server binaries
* conf: Contains server configuration files and the XL Deploy license
* doc: Contains the XL Deploy product documentation
* ext: Contains server extensions
* hotfix: Contains hotfixes (XL Deploy 4.5.x and earlier only)
* hotfix/lib: Contains hotfixes that fix issues with the server software (XL Deploy 5.0.0 and later)
* hotfix/plugins: Contains hotfixes that fix issues with the plugin software (XL Deploy 5.0.0 and later)
* importablePackages: Default location for importable deployment packages
* lib: Contains libraries that the server needs
* log: Contains server log files (this directory is only present once you have started XL Deploy server)
* plugins: Contains XL Deploy middleware plugins
* recovery.dat: Stores tasks that are in progress for recovery purposes (this file is only present after you have started XL Deploy server)
* samples: Contains sample plugins and configuration snippets

## **Automatically install XL Deploy with default values**

You can automate the installation of XL Deploy with a set of default values that you save in a file. This is useful, for example, when setting up XL Deploy using a tool such as Puppet or Ansible. To install XL Deploy, use the following command:

bin/run.sh -setup -reinitialize -force -setup-defaults /path/to/deployit.conf

Where the deployit.conf file contains the installation values that you want to use.

## **Install the XL Deploy CLI**

For information about installing the XL Deploy command-line interface (CLI), refer to [Install the XL Deploy CLI](https://docs.xebialabs.com/xl-deploy/how-to/install-the-xl-deploy-cli.html).

## **Failover configuration**

If you store the XL Deploy repository in a [database](https://docs.xebialabs.com/xl-deploy/how-to/configure-the-xl-deploy-repository.html#using-a-database), you can set up a failover configuration as described in [Configure failover for XL Deploy](https://docs.xebialabs.com/xl-deploy/how-to/configure-failover.html).

# **Install the XL Deploy CLI**

You can install the XL Deploy command-line interface (CLI) on any remote computer and connect to the XL Deploy server.

1. Ensure that XL Deploy is running.
2. Open a terminal window or command prompt and go to the XL\_DEPLOY\_CLI\_HOME/bin directory.
3. Execute the start command:
   1. Unix-based operating systems: ./cli.sh
   2. Microsoft Windows: cli.cmd
4. Enter your user name and password.
5. The CLI attempts to connect to the server on localhost, running on XL Deploy’s standard port of 4516.

## **Set environment variables**

After you install the XL Deploy CLI, it is recommended that you set the DEPLOYIT\_CLI\_HOME environment variable to the root directory where the CLI is installed.

A second environment variable, DEPLOYIT\_CLI\_OPTS, can be used to provide Java Virtual Machine (JVM) options for the XL Deploy CLI process. For example, to set the initial Java heap size to 512 MB and the maximum Java heap size to 2 GB, set the environment variable as follows:

* Unix-based operating systems: export DEPLOYIT\_CLI\_OPTIONS="-Xms512m -Xmx2g"
* Microsoft Windows: set DEPLOYIT\_CLI\_OPTIONS="-Xms512m -Xmx2g"

If DEPLOYIT\_CLI\_OPTS is not set, the CLI startup scripts provide sensible defaults.

## **Sample CLI scripts**

# Load package

package = repository.read('Applications/Sample Apps/BookStore/1.0.0')

# Load environment

environment = repository.read('Environments/Testing/TEST01')

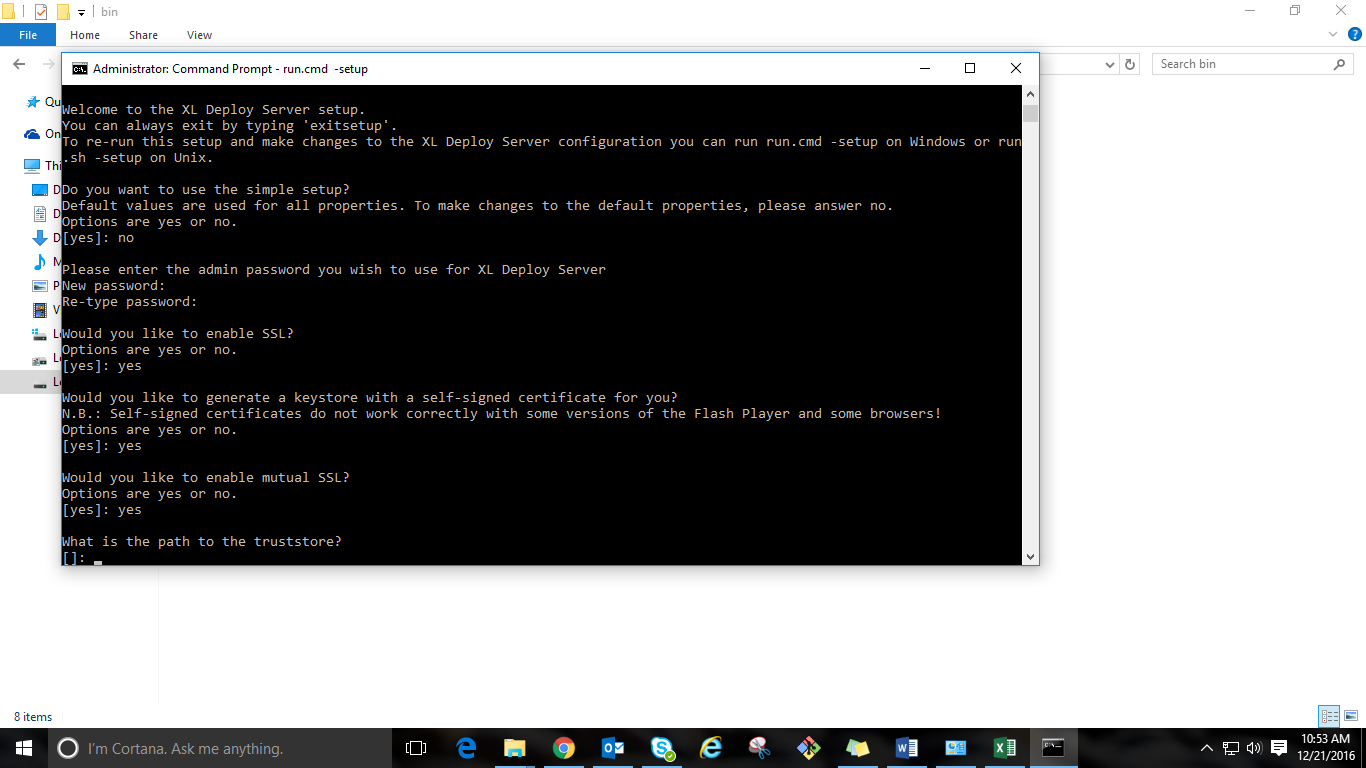
# Start deployment

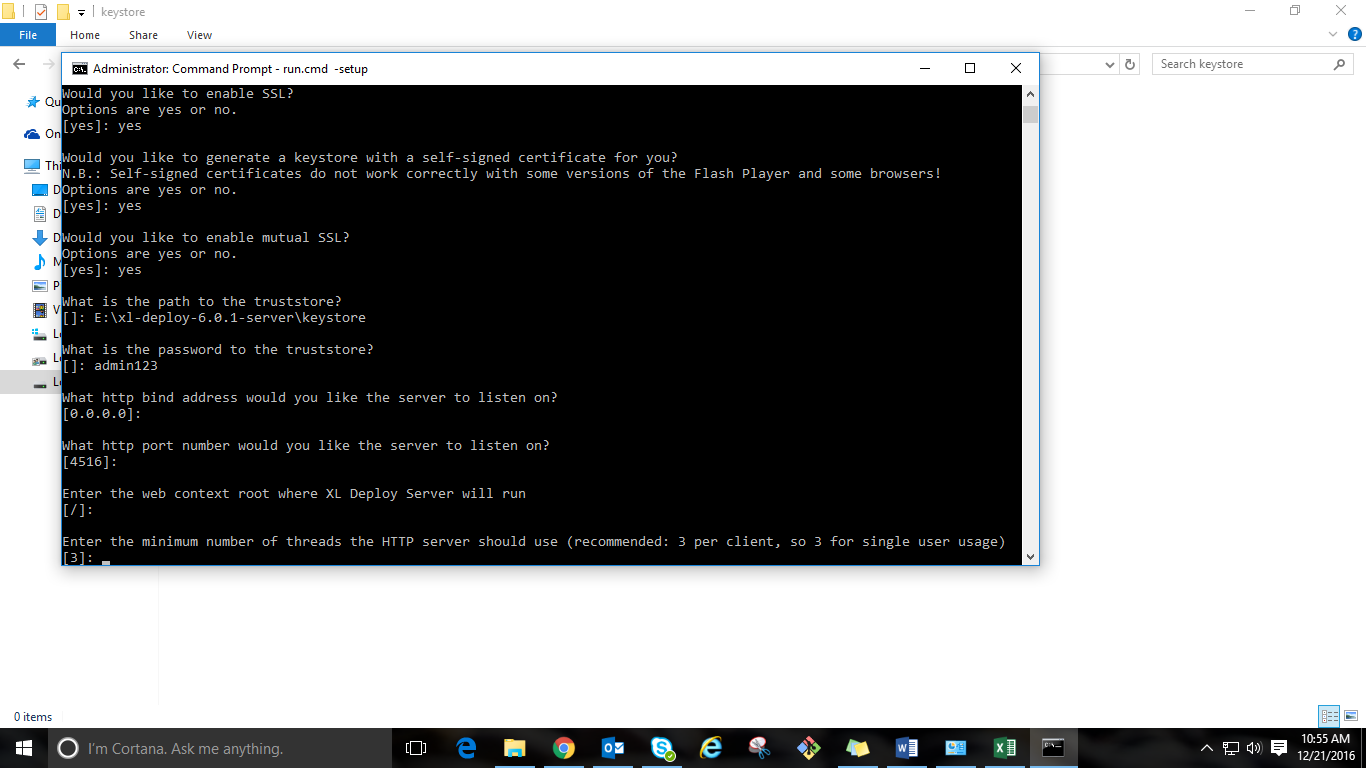
deploymentRef = deployment.prepareInitial(package.id, environment.id)

depl = deployment.prepareAutoDeployeds(deploymentRef)

task = deployment.createDeployTask(depl)

deployit.startTaskAndWait(task.id





deployer = security.createUser("john", "secret")

security.assignRole("developers", ["john"])

adminUser = security.readUser('admin') adminUser.password = 'newpassword\_1' security.modifyUser(adminUser)

admin.password=newpassword\_1