

|  |  |
| --- | --- |
| Document name | Nexus Guide |
| Version no. | 1.0 |
| Release date |  |



|  |
| --- |
| Guidelines of Lunch Committee  (HR-GLC) |

|  |
| --- |
| Guidelines of Lunch Committee  (HR-GLC) |

NEXUS

##### Document History

| Version No. | Authored / Modified by | Reviewed by, Date | Approved by, Date | Remark / Change History |
| --- | --- | --- | --- | --- |
| 1.0 | Sakshi Kathuria  2-June-15 |  |  | Initial Document |

NEXUS REPOSITORY MANAGER

Nexus manages software components required for development, deployment, and provisioning**. If you develop software, Nexus can help you share those components with other developers and end users**.  Nexus greatly simplifies the maintenance of your own internal repositories and access to external repositories. With Nexus you can completely control access to, and deployment of, every component in your organization from a single location.

Nexus is available in three editions:

* Nexus OSS
* Nexus Pro
* Nexus Pro+

Nexus OSS - It provides infrastructure and services for organizations that use repository managers to obtain and deliver software. If you create software libraries or applications for your end users, you can use Nexus OSS to distribute your software.

Nexus PRO - It acts as a central point of access to external repositories and a central distribution point with the intelligence required to support the decision that go into making quality software.

Nexus PRO+ - The Nexus Pro+ includes a Nexus Lifecycle server that can be used to define component usage policies and automate the enforcement during the release process with the Nexus Staging Suite.

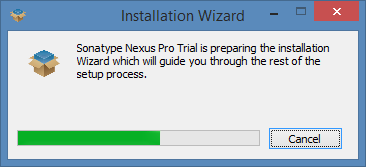
Nexus PRO FEATURES:

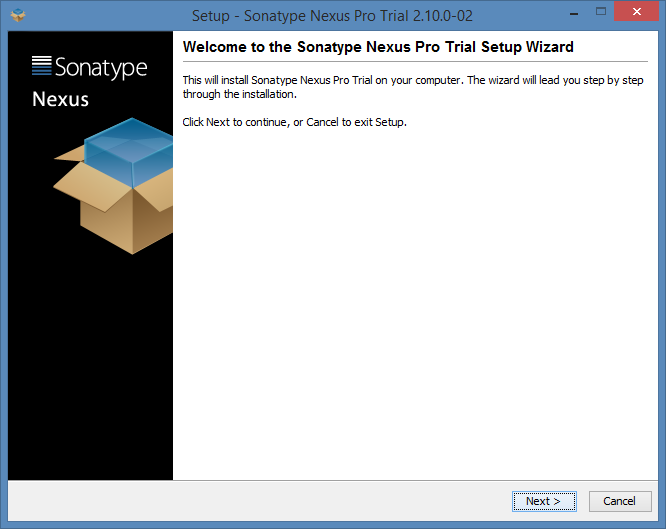
* Nexus Staging Suite: It provides workflow support for the release process of binary software components. If you need to create a release component and deploy it to a hosted repository, you can use the Staging Suite to post a collection of related, staged components which can be tested, promoted, or discarded as a unit. Nexus keeps track of the individuals who are involved in a staged, managed release and can be used to support the decisions that go into producing quality software.
* Enterprise LDAP Support: Nexus Pro offers [LDAP support](http://books.sonatype.com/nexus-book/reference/ldap.html) features for enterprise LDAP deployments, including detailed configuration of cache parameters, support for multiple LDAP servers and backup mirrors, the ability to test user logins, support for common user/group mapping templates, and the ability to support more than one schema across multiple servers.
* Support for Atlassian Crowd: If your organization uses Atlassian Crowd, Nexus Pro can [delegate authentication and access control to a Crowd server](http://books.sonatype.com/nexus-book/reference/crowd.html) and map Crowd groups to the appropriate Nexus roles.
* Maven Settings Management: Nexus Pro along with the Nexus M2Settings Maven Plugin allows you to [manage Maven settings](http://books.sonatype.com/nexus-book/reference/maven-settings.html). Once you have developed a Maven Settings template, developers can then connect to Nexus Pro using the Nexus M2Settings Maven plugin which will take responsibility for downloading a Maven settings file from Nexus and replacing the existing Maven settings on a local workstation.

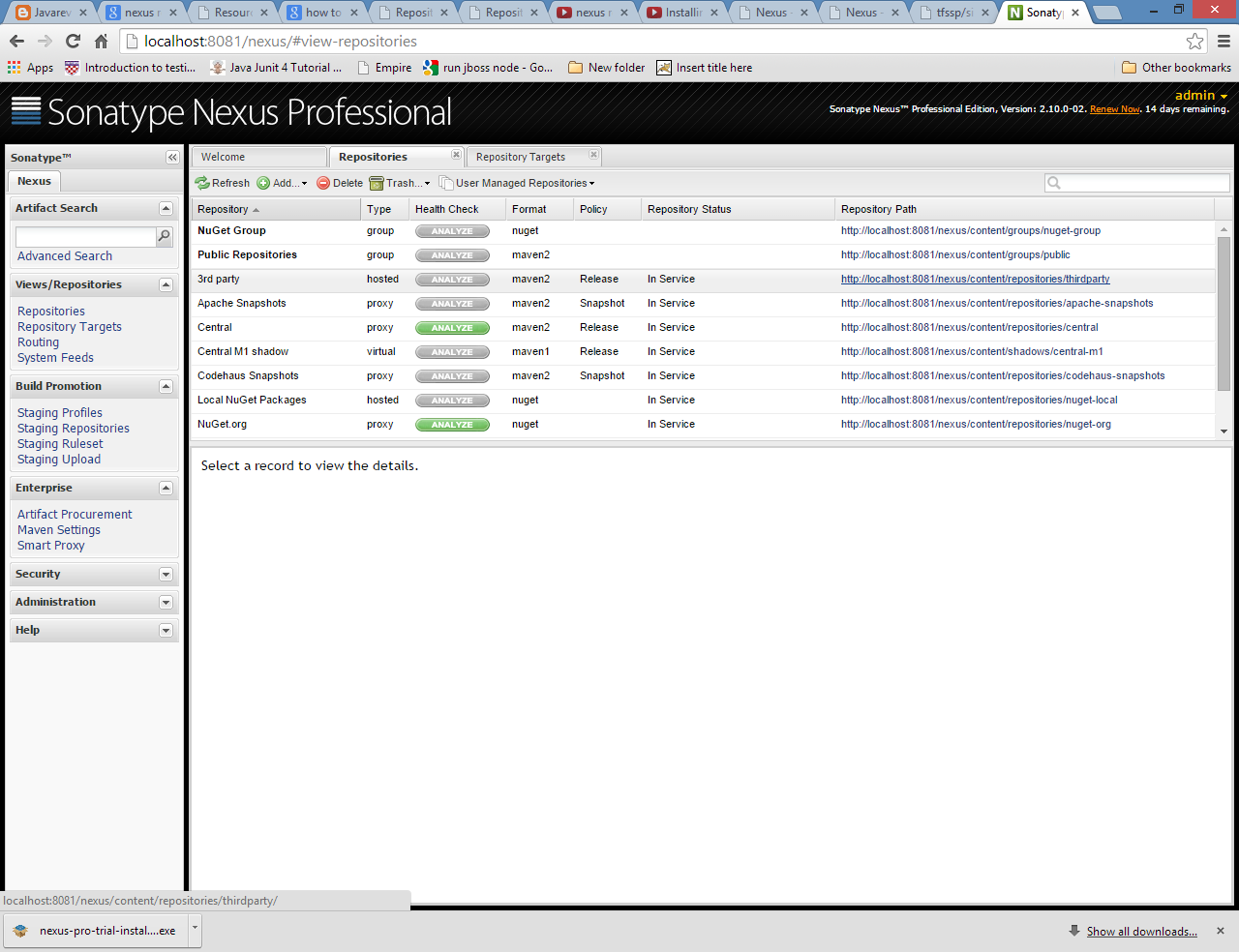
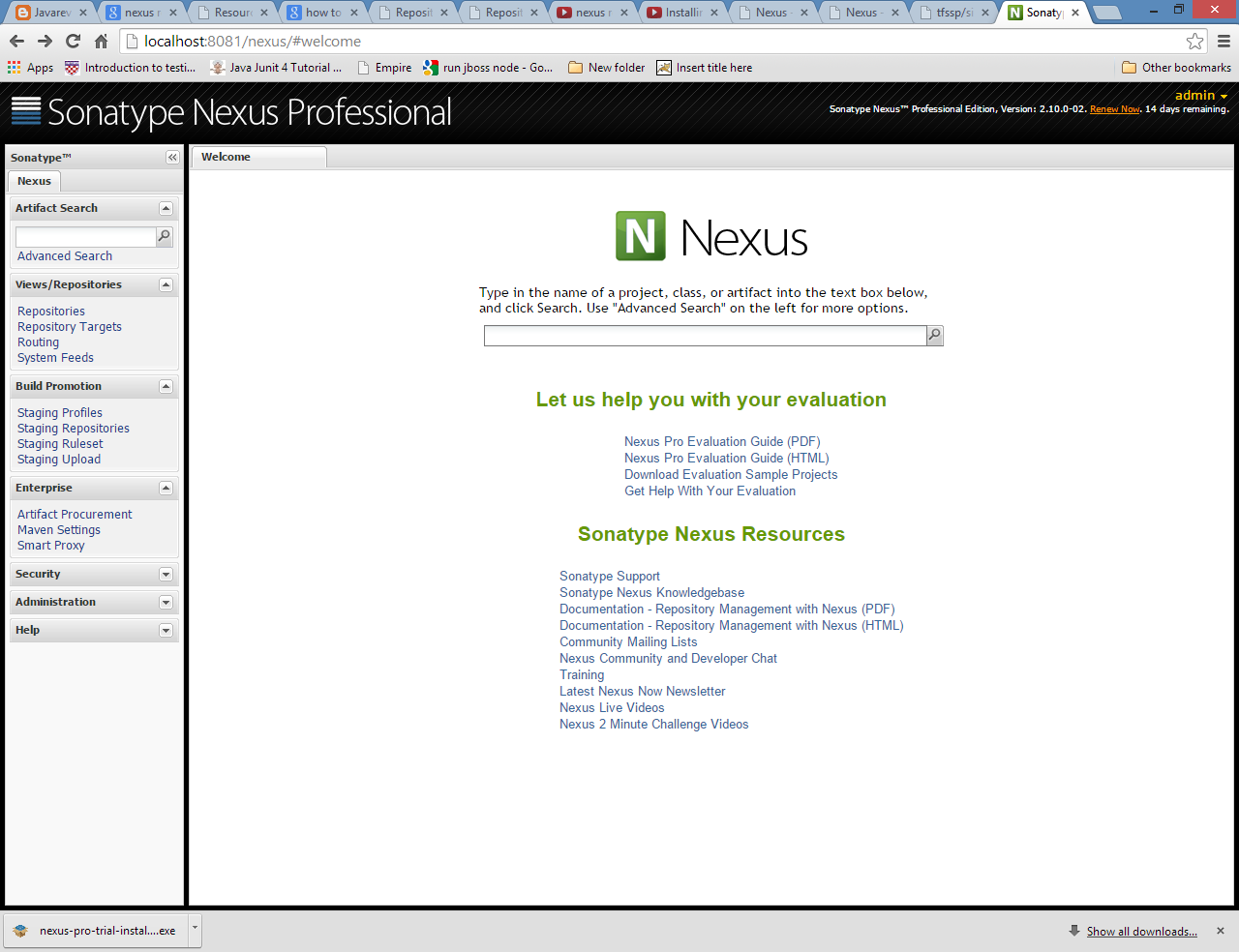
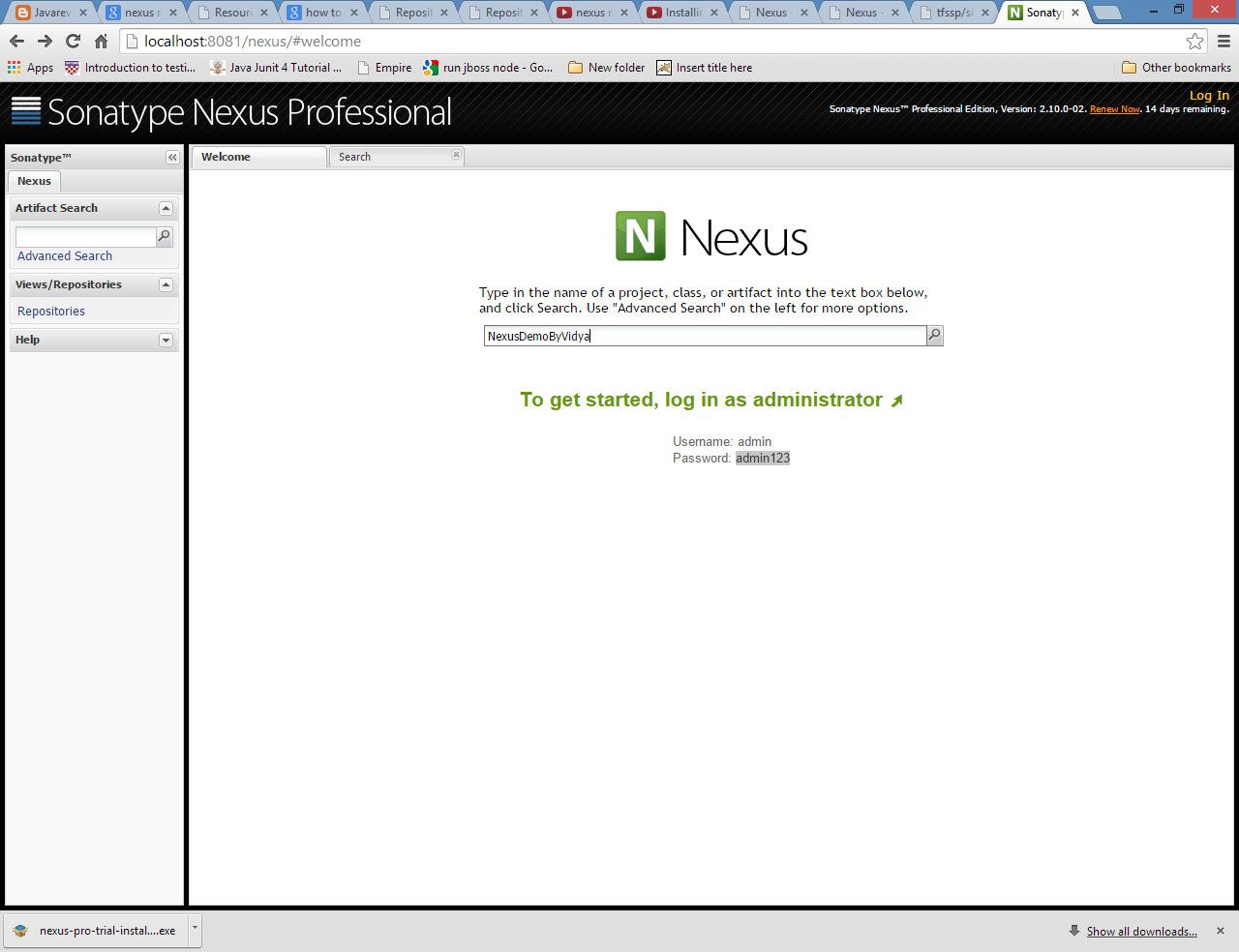
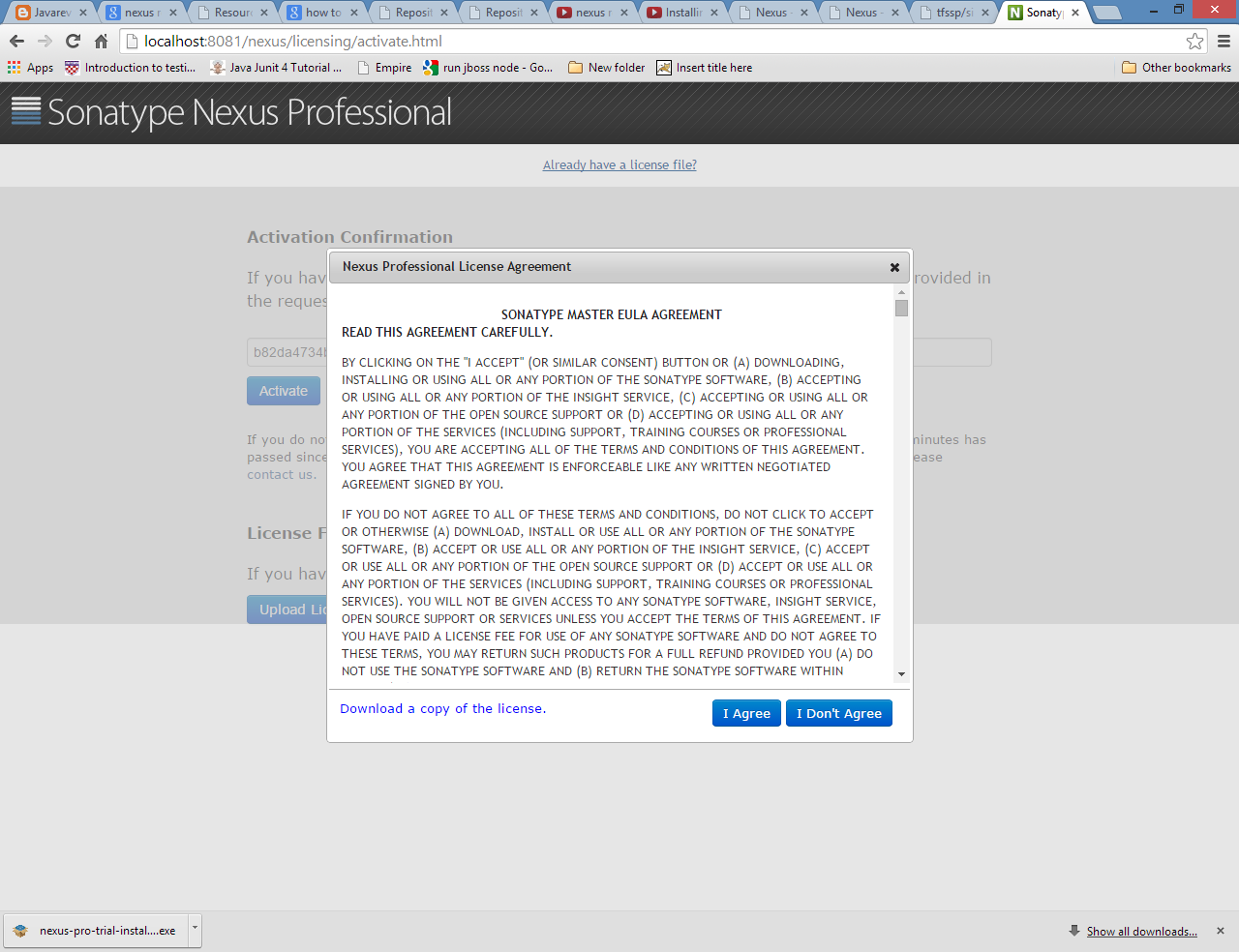
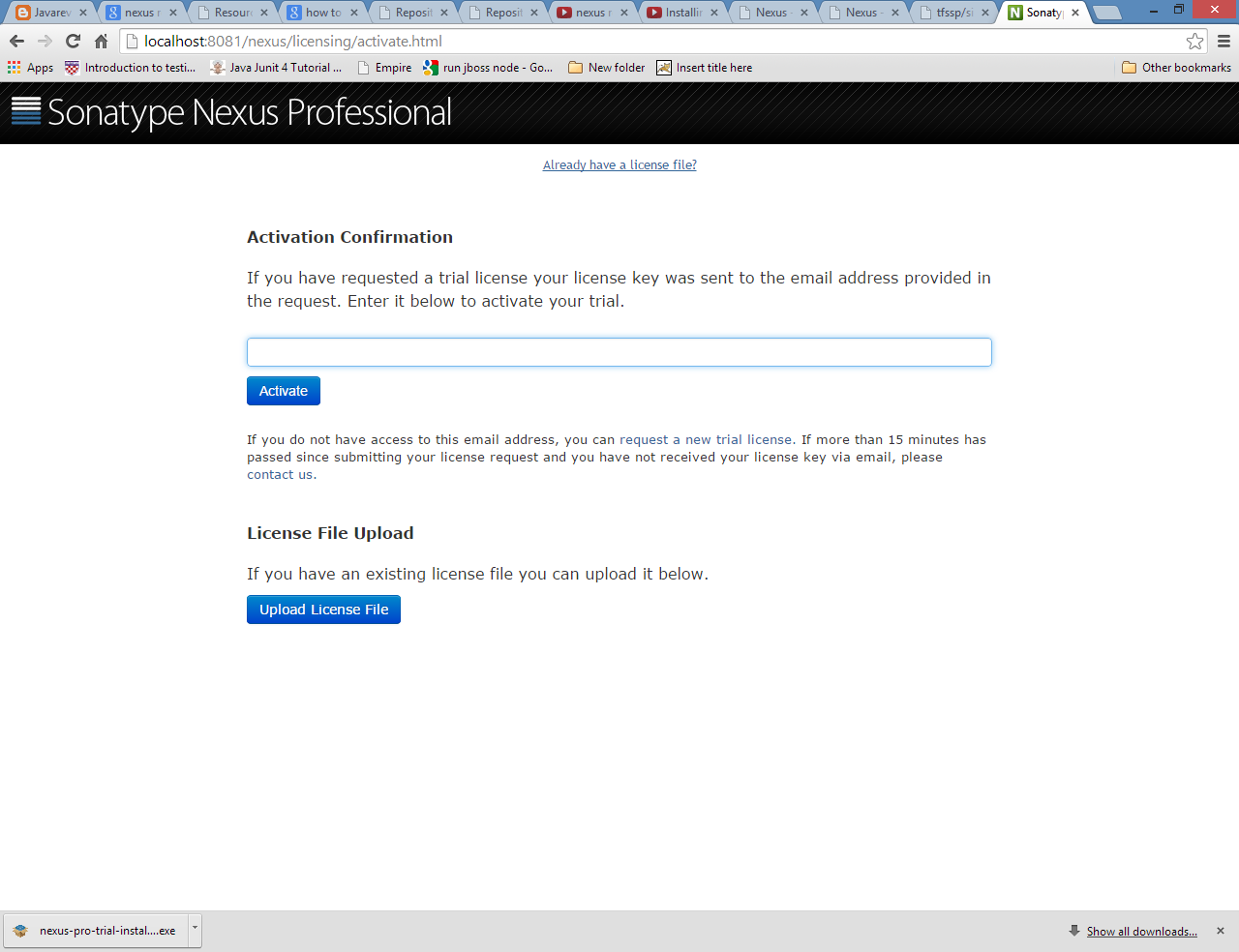
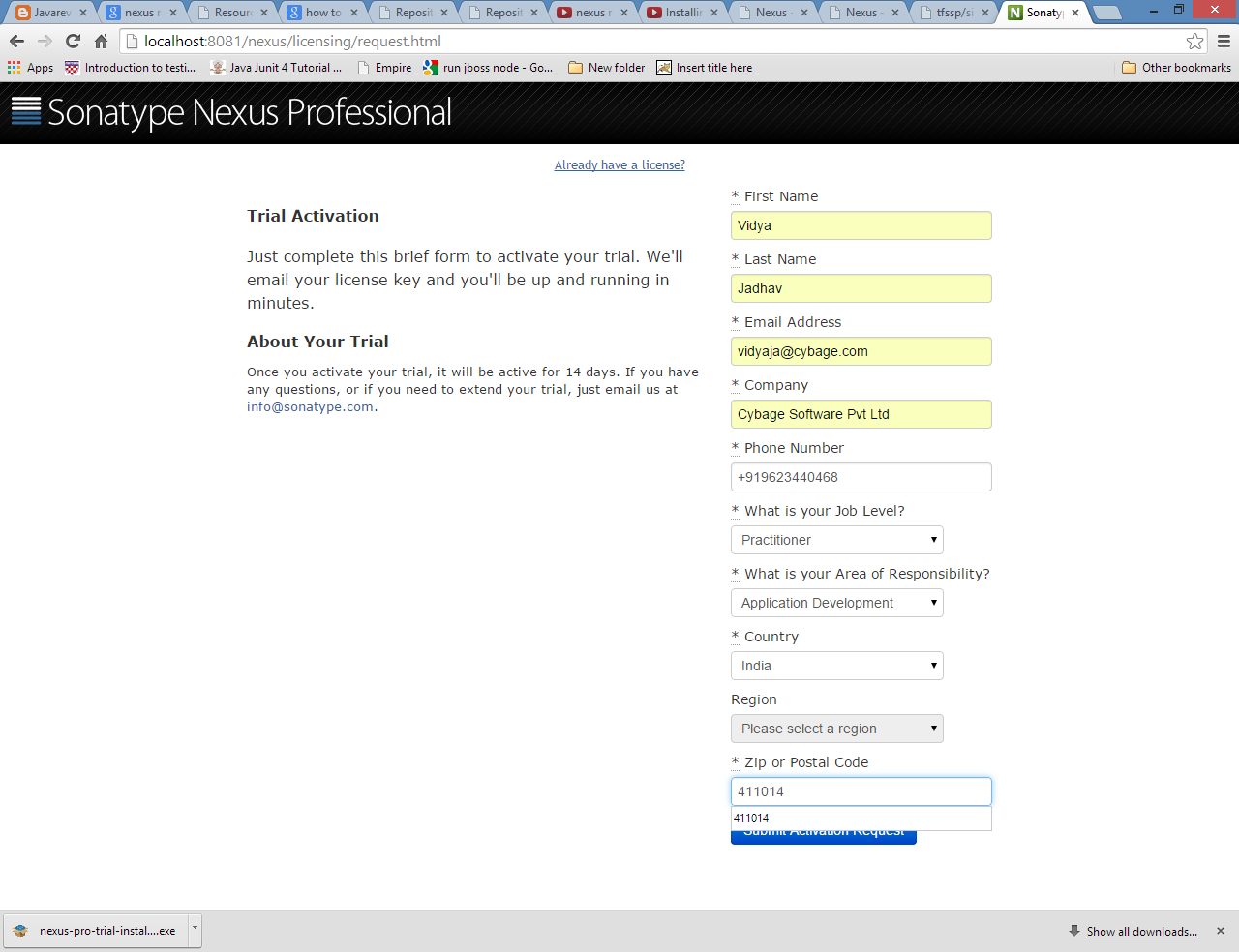
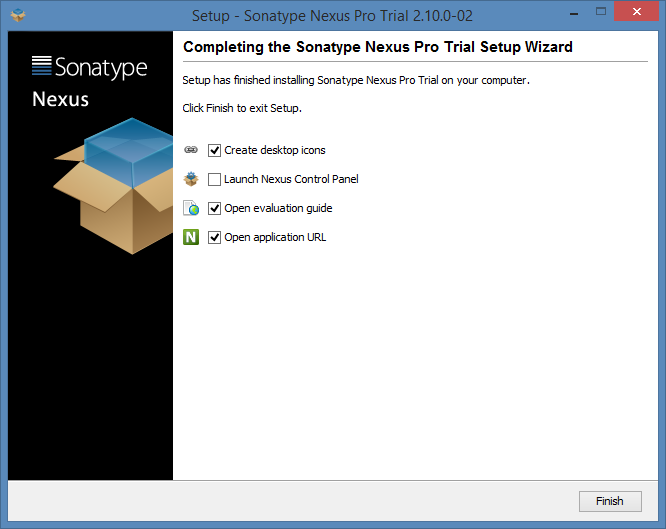
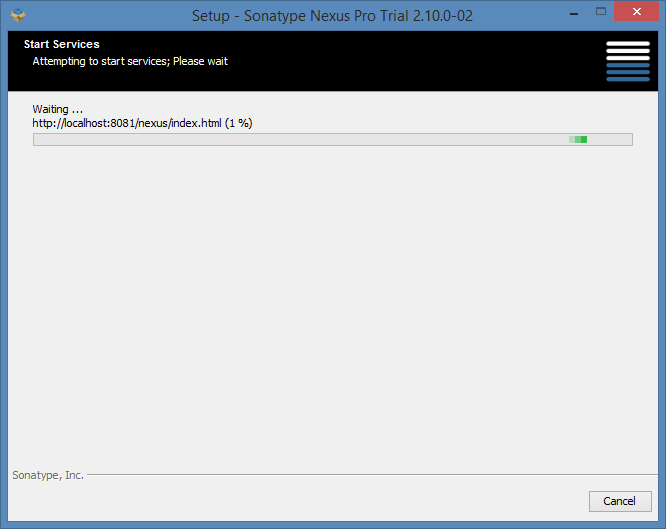
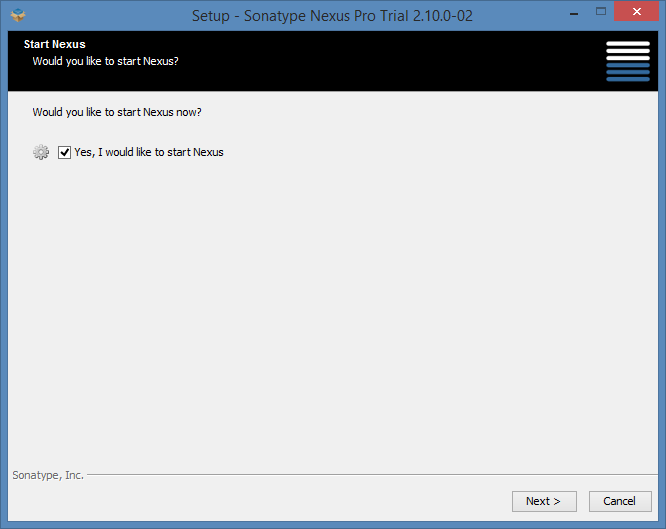
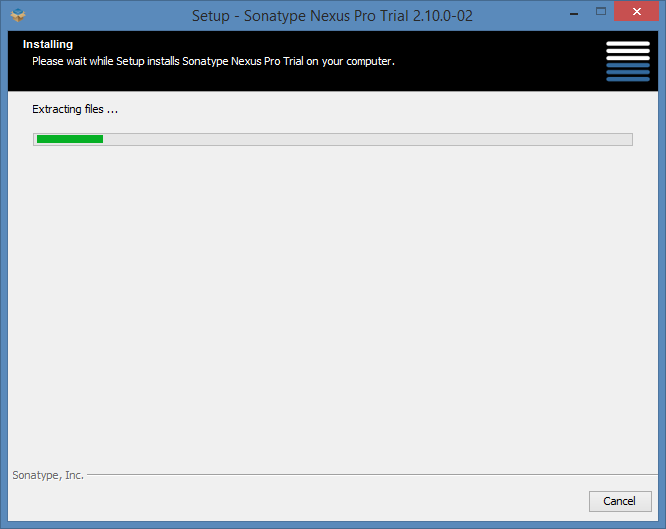
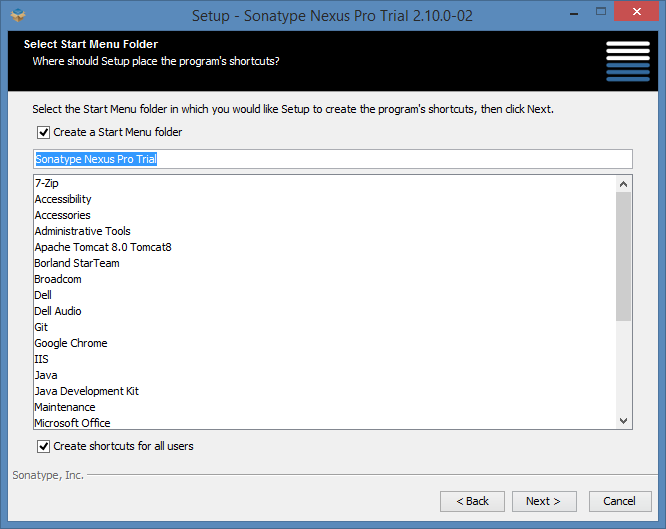
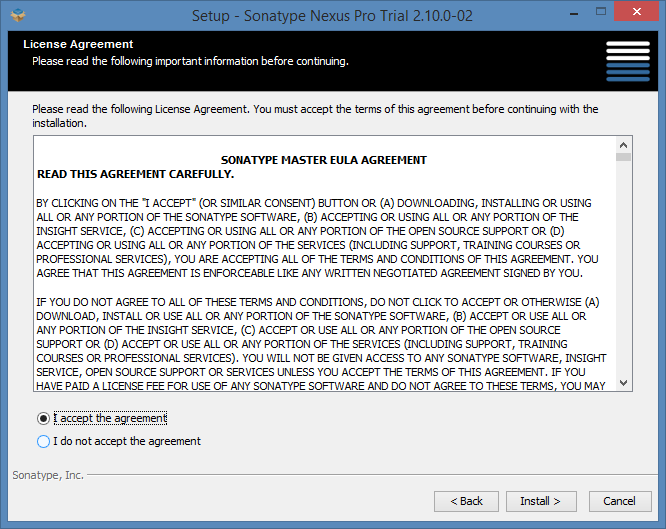
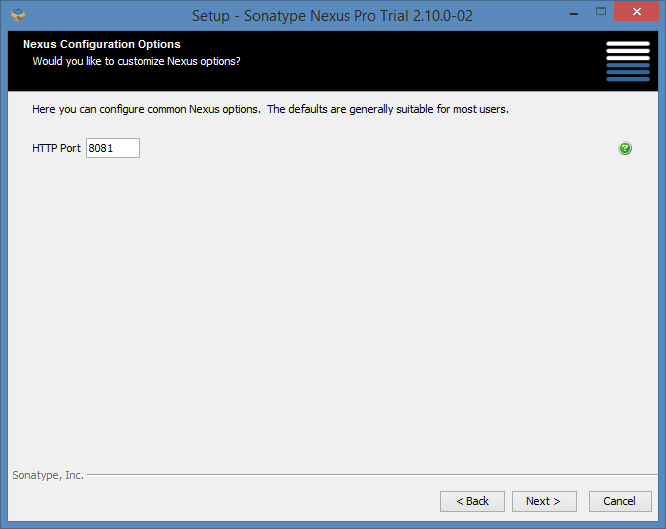
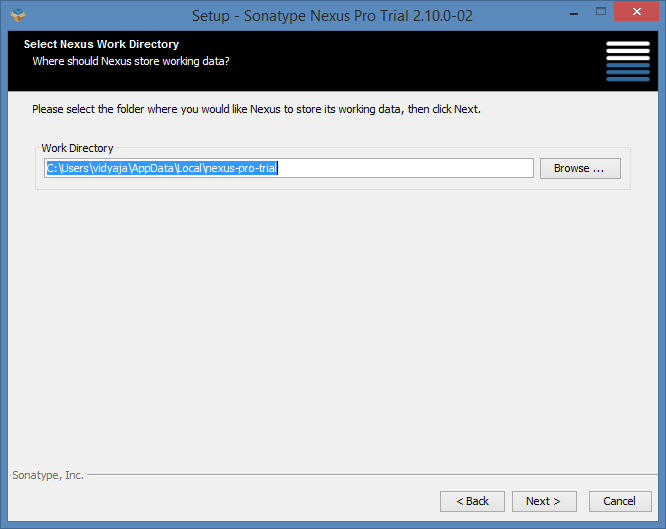
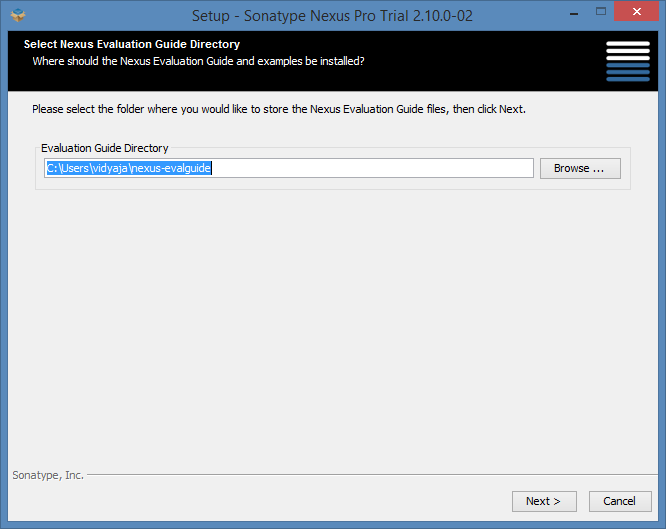
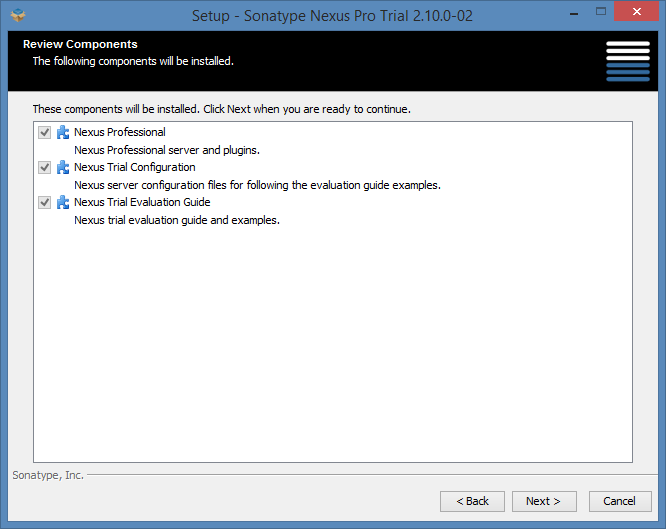
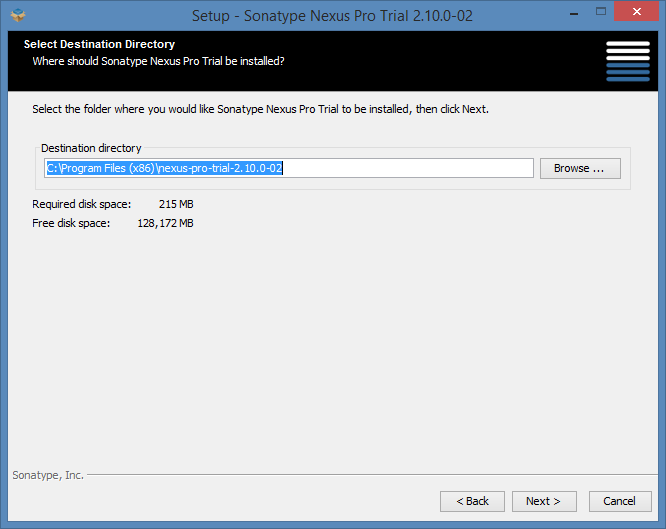
Nexus PRO Installation

**Download Sonatype Nexus pro from following link and follow the steps as mentioned below :-**

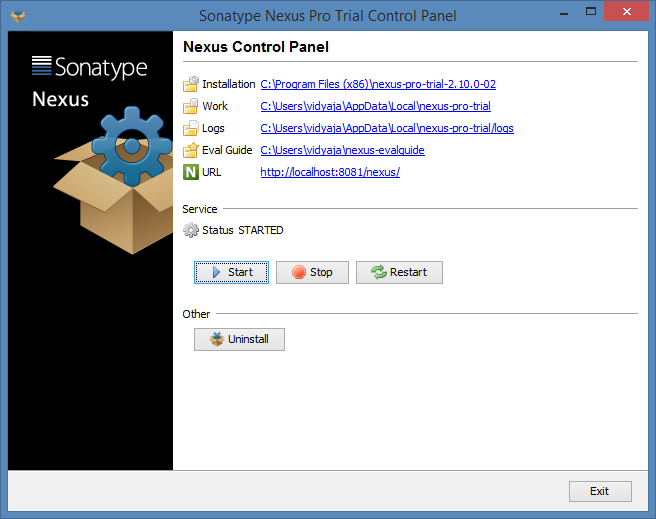
[**http://www.sonatype.org/nexus/go/**](http://www.sonatype.org/nexus/go/)







As shown below Nexus control panel can be used to start Nexus :-



Configuration for downloading dependencies from nexus:

**In pom.xml you have to specify**

<repository>

<id>central</id>

<name>admin</name>

<url>[http://localhost:8082/nexus/content/repositories/central/</url](http://localhost:8082/nexus/content/repositories/central/%3c/url)

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</repository>

**In .m2/settings.xml :**

<server>

<id>nexus</id>

<username>admin</username>

<password>sakshi123</password>

</server>

**In global settings.xml:**

<server>

<id>nexus</id>

<username>admin</username>

<password>sakshi123</password>

</server>

<mirror>

<!--This sends everything else to /public -->

<id>nexus</id>

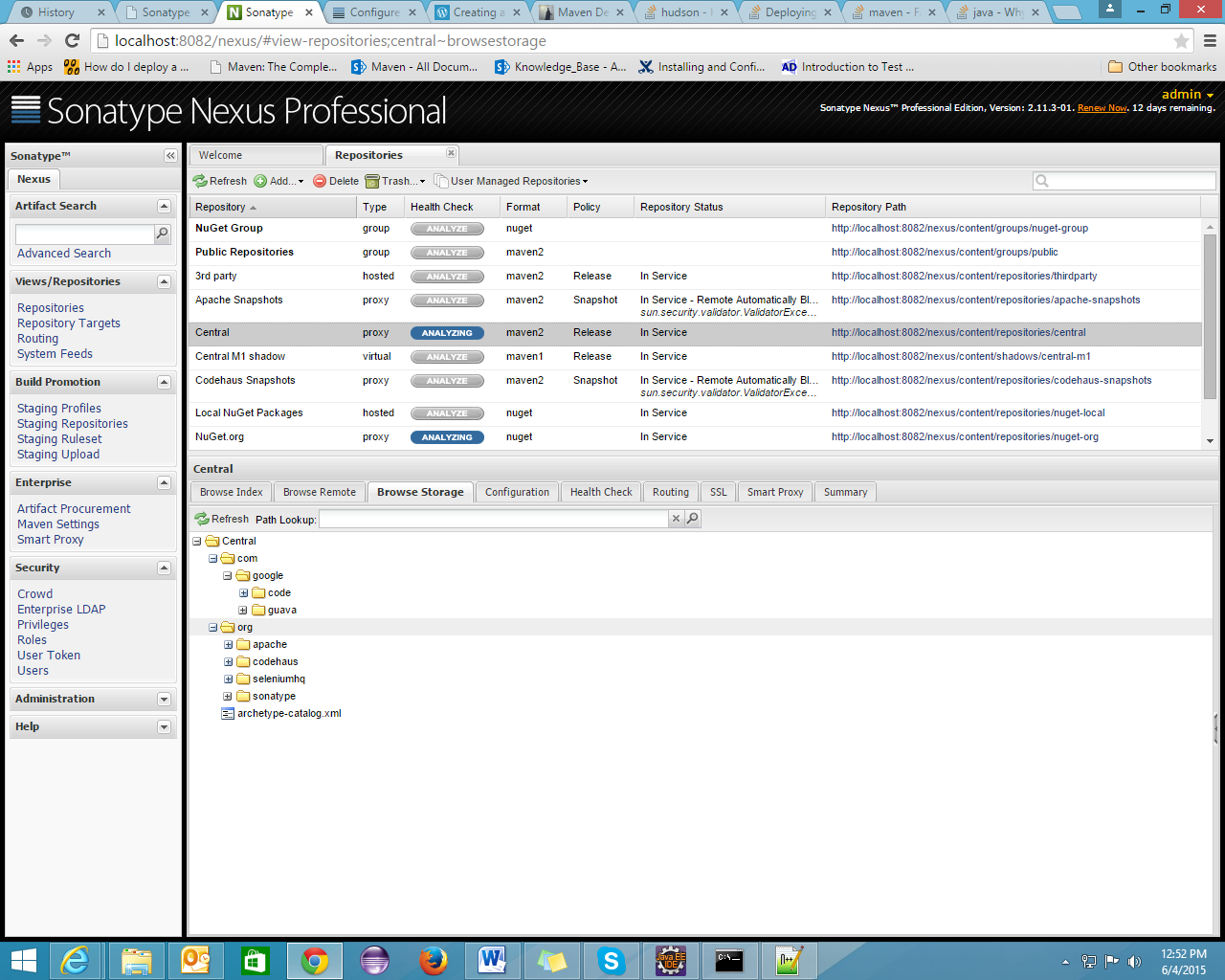
<mirrorOf>\*</mirrorOf>

<url>http://localhost:8082/nexus/content/groups/public</url>

</mirror>

</mirrors>

After these entire configurations you will see the dependencies in nexus dashboard as shown below:



Configuration for deploying dependencies into nexus:

**In pom.xml you have to do the following configuration:**

<distributionManagement>

<repository>

<id>releases</id>

<name>admin</name>

<url>http://localhost:8082/nexus/content/repositories/releases/</url

</repository>

<snapshotRepository>

<id>snapshots</id>

<name>admin</name>

<url>http://localhost:8082/nexus/content/repositories/snapshots/</url>

</snapshotRepository>

</distributionManagement>

**Plugin:**

<plugin>

<groupId>org.sonatype.plugins</groupId>

<artifactId>nexus-staging-maven-plugin</artifactId>

<version>1.6.3</version>

<!-- <extensions>true</extensions> -->

<executions>

<execution>

<phase>deploy</phase>

<goals><goal>deploy</goal></goals>

</execution>

</executions>

<configuration>

<serverId>nexus</serverId>

<name>admin</name>

<nexusUrl>http://localhost:8082/nexus/</nexusUrl>

<skipStaging>true</skipStaging>

</configuration>

</plugin>

You need to skip the maven-deploy plugin

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.7</version>

<executions>

<execution>

<configuration>

<skip>true</skip>

</configuration>

</execution>

</executions>

</plugin>

**In .m2/settings.xml**:

<server>

<id>releases</id>

<username>admin</username>

<password>sakshi123</password>

</server>

<server>

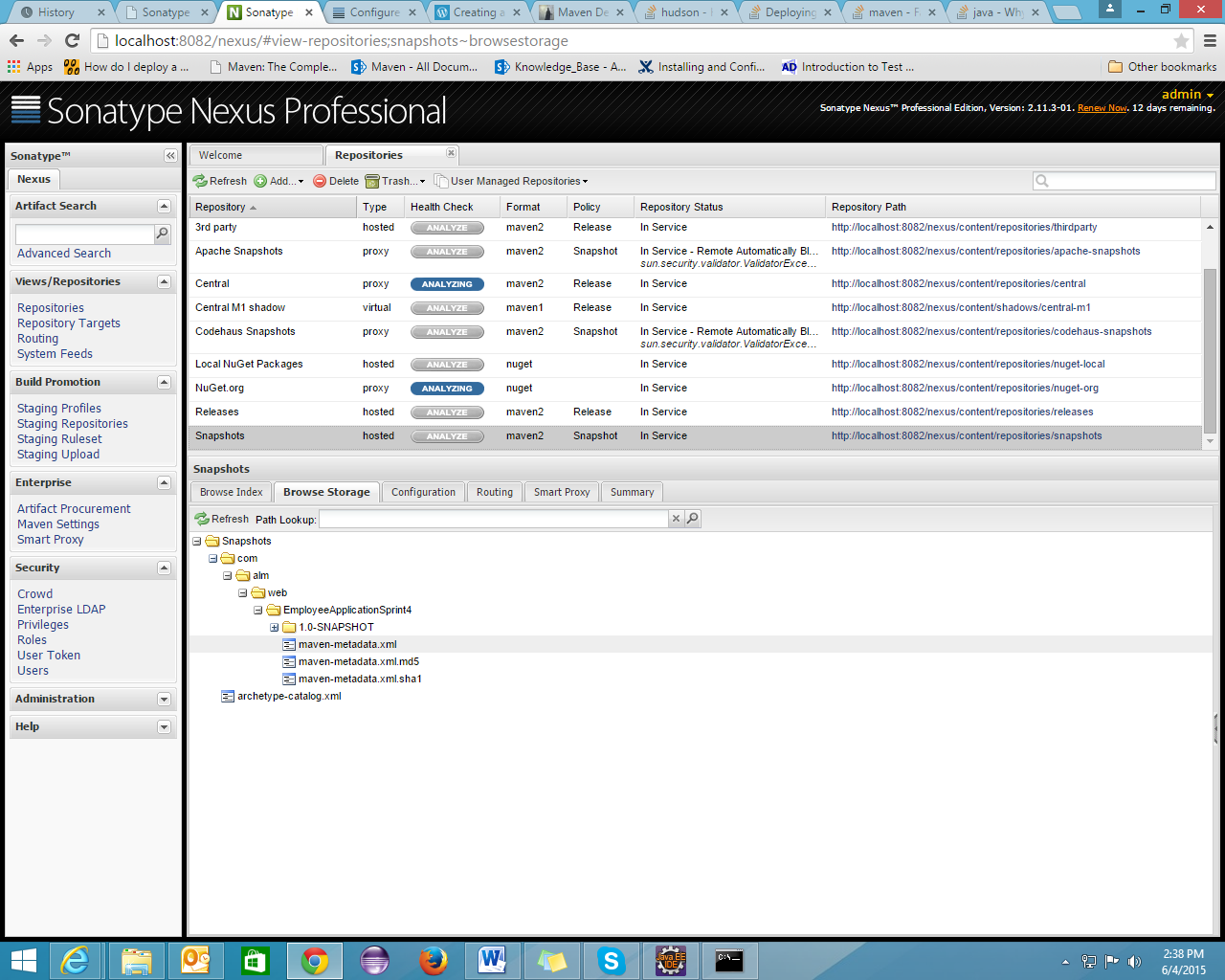
<id>snapshots</id>

<username>admin</username>

<password>sakshi123</password>

</server>

You will see your application in snapshot repository of nexus as shown below:



Repository Connector Plugin – 2 build parameters : Maven repo artifact(not their), artifact resolver

Artifact Promotion Plugin (under development) – to promote artifacts.

Nexus task runner plugins.- scheduled task after your build.

**Artifactory vs Nexus**

**Latest version of nexus** – 2.11.2

**Artifactory** – 3.8.0

**Artifactory** - private Artifactory instance in the cloud with full self-service, pay-per-use model and always up-to-date with all the latest features.

**Nexus Pro-** --

**Artifactory** database based application

**Nexus** writes directly to the file system

Artifactory supports snapshot and release in a single repo

Each repository can only be setup for either snapshot OR release. If we require both snapshot and release in a single repository, we can create 2 repository for snapshot and release than group them in a "group repository" in Nexus.

Directory Structure:

Access: This directory contains a log of all IP addresses accessing Nexus. Database files are there in this folder.

Backup: If you have configured a scheduled job to back up Nexus configuration, this directory is going to

contain a number of ZIP archives that contain snapshots of Nexus configuration.

Conf: This directory contains the Nexus configuration. Settings that define the list of Nexus repositories,

the logging configuration, the staging and procurement configuration, and the security settings are

all captured in this directory.

Plugin repo: additional installed plugins.

Storage: Stores artifacts and metadata for Nexus repositories. Each repository is a subdirectory that contains the artifacts in a repository.

* What is nexus?
* Difference between nexus and artifactory?
* Folder structure.
* Dashboard
* Jenkins plugins
* Bamboo plugin is deprecated.
* Staging Profiles