Jenkins & Nexus Setup

This doc as of now doesn't include the steps to install Jenkins or Nexus but to configure it for usage.

Nexus setup

Nexus is not exposed via public IP. As of now only Jenkins can push or pull packages from it via private IP.

1. Create argoid user in nexus so it can be configured in Jenkins to push packages and also to fetch dependent packages.
   a. Server administration > Users > Create local user
      i. ID as argoid, setup password and set role as nx-admin
2. Configure maven proxy so that all artifacts are fetched via Nexus. Nexus will cache the packages so that subsequent fetch by same or other Jenkins pipelines are faster.
   a. Server administration > Repositories > maven2(proxy)
      i. name - maven-proxy
      ii. Remote storage - https://repo1.maven.org/maven2/
      iii. Blob storage - default
3. Configure sbt plugins proxy repo
   a. Server administration > Repositories > maven2(proxy)
      i. name - sbt-plugins-proxy
      ii. Layout - Permissive (**Very important**)
      iii. Remote storage - https://scala.jfrog.io/artifactory/sbt-plugin-releases/ or https://repo.scala-sbt.org/scalasbt/sbt-plugin-releases/ only one, use whichever works
      iv. Blob storage - default
4. **We may have to create one more repo for ivy-releases similar to sbt-plugins-repo, only change is**
   a. **name - ivy-releases-proxy**
   b. **Remote storage - https://scala.jfrog.io/ui/native/ivy-releases or https://repo.typesafe.com/typesafe/ivy-releases**
5. For internal release and snapshot artifacts(created by Argoid) will be using the default Nexus repos - maven-releases and maven-snapshots
6. Create snapshot cleanup task to remove older snapshots for same version
   a. Server administration> System> Tasks> Create Task
      i. name - cleanup_maven_snapshots
      ii. Repository - maven-snapshots
      iii. Minimum snapshot count - 1
      iv. Snapshot Retention(days) - 7
      v. Task frequency - Daily
      vi. Time to run this task - 00:00

Jenkins setup

Prerequisites

- Install maven on /opt/maven3 or any other location
- Install sbt
  - `curl -L https://www.scala-sbt.org/sbt-rpm.repo > sbt-rpm.repo`
  - sudo mv sbt-rpm.repo /etc/yum.repos.d/
  - sudo yum install sbt
- Below steps assume Jenkins home directory is /var/lib/jenkins. If not change below steps accordingly.
- sudo -u jenkins mkdir /var/lib/jenkins/.sbt
- sudo -u jenkins vi /var/lib/jenkins/.sbt/repositories

```
[repositories]
    local
    sbt-plugins-proxy: http://repo1.internal.argoid.com:8081
/repository/sbt-plugins-proxy/, [organization]/[module]/(scala_
[scalaVersion]/)(sbt_[sbtVersion]/)[revision]/[type]s/[artifact](-
[classifier]).[ext], allowInsecureProtocol
    maven-proxy-releases: http://repo1.internal.argoid.com:8081
/repository/maven-proxy/, allowInsecureProtocol
    maven-internal-releases: http://repo1.internal.argoid.com:8081
/repository/maven-releases/, allowInsecureProtocol
    maven-internal-snapshots: http://repo1.internal.argoid.com:8081
/repository/maven-snapshots/, allowInsecureProtocol
```

- In case ivy-releases repo is created as mentioned in repo creation process in nexus above section, then add one more line after sbt-plugins-proxy above like

```
• ivy-releases-proxy: http://repo1.internal.argoid.com:8081/repository
  /ivy-releases-proxy/, [organization]/[module]/(scala_[scalaVersion]
  /)(sbt_[sbtVersion]/)[revision]/[type]s/[artifact](-[classifier]).
  [ext], allowInsecureProtocol
```

- sudo -u jenkins vi /var/lib/jenkins/.sbt/.credentials

  -
    ```
    realm=Sonatype Nexus Repository Manager

    host=repo1.internal.argoid.com

    user=argoid

    password=<password>
    ```
- sudo -u jenkins mkdir /var/lib/jenkins/.sbt/{1.0,boot}
- sudo -u jenkins vi /var/lib/jenkins/.sbt/1.0/credentials.sbt

  ```
  credentials += Credentials(Path.userHome / ".sbt" / ".credentials")
  ```
- sudo -u jenkins vi /var/lib/jenkins/.sbt/boot/credentials.sbt

  ```
  credentials += Credentials(Path.userHome / ".sbt" / ".credentials")
  ```
- sudo vi /etc/sbt/sbtopts and append at end

```
• -Dsbt.boot.credentials="/var/lib/jenkins/.sbt/.credentials"
  -Dsbt.override.build.repos=true
  -Dsbt.repository.config="/var/lib/jenkins/.sbt/repositories"
```

1. Create required credentials in Jenkins - Manage Jenkins > Manage Credentials > Stores scoped to Jenkins(Jenkins) > Global credentials > Add Credentials
   a. Nexus credentials(Username with password)
      i. ID: nexus_credentials
      ii. username: argoid
      iii. password: same password set in Nexus
   b. Create specific user for each project for both prod and stating environment to deploy packages like reco-cache and reco-rest (SSH username with private key). Please refer to **Jenkins User to Deploy** column in google sheet : https://docs.google.com/spreadsheets/d/1MZ06FFugNoIcQBS2VAu4LCSXWpDYZBByBizJlGHi6pE/edit#gid=1962507149

      Ex:

SSH credentials of prod-fun2 user

ID: ssh-prod-fun2

username: prod-fun2

Private key > Enter directly - past the private key

Note: use different ssh keys for each user.

1. Bitbucket credentials to checkout repo(Username with password)
    a. ID:
    b. Username: devops@argoid.com
    c. Password: password set in bitbucket
    d. Note: the above password has to be generated from Bitbucket after logging into Bitbucket as devops@argoid.com user. Go to user profile, app passwords, then generate a password with git read/write privileges and name as 'jenkins'. Also need to go to SSH keys, then add public key of jenkins user after generating ssh keys for jenkins user in jenkins instance. Also for first time host key verification, need to try to ssh to Bitbucket as jenkins user from jenkins instance -

```
sudo -u jenkins ssh bitbucket.org
```

2. Set Nexus global environment variables
    a. Manage Jenkins > Configure System > Global properties > Environment variables
    b. Set below variables
        i. NEXUS_CREDENTIAL_ID = nexus_credentials
        ii. NEXUS_PROTOCOL = http
        iii. NEXUS_RELEASES_REPO = maven-releases
        iv. NEXUS_SNAPSHOTS_REPO = maven-snapshots
        v. NEXUS_URL = <nexus_ip>:<nexus_port>
        vi. NEXUS_VERSION = nexus3
3. Configure maven as tool if not already done
    a. Manage Jenkins > Global Tool Configuration > Maven(Maven Installations)
    b. Configure as
        i. Name: maven_38
        ii. MAVEN_HOME: /opt/maven3(change accordingly if required)
4. Configure maven to use Nexus as default repo and also to fetch from internal Argoid repos configured in Nexus. Modify /opt/maven3/conf/settings.xml
    a. Under <servers> add
        i.
```
  <server>
      <id>nexus</id>
      <username>argoid</username>
      <password>{as configured in Nexus}</password>
    </server>
    <server>
      <id>internal-releases</id>
      <username>argoid</username>
      <password>{as configured in Nexus}</password>
    </server>
    <server>
      <id>internal-snapshots</id>
      <username>argoid</username>
      <password>{as configured in Nexus}</password>
    </server>
```
    b. Under <mirrors> add. NOTE: May have to disable http blocker mirror as well.
        i.
```
  <mirror>
```

```
            <id>nexus</id>

            <mirrorOf>central</mirrorOf>

            <url>http://repo1.internal.argoid.com:8081/repository/maven-proxy/</url>

          </mirror>
```

c. Under <profiles> add

   i.

```
            <profile>

            <id>nexus</id>

            <!--Enable snapshots for the built in central repo to direct -->

            <!--all requests to nexus via the mirror -->

            <repositories>

              <repository>

                <id>central</id>

                <url>https://repo1.maven.org/maven2/</url>

                <releases><enabled>true</enabled></releases>

                <snapshots><enabled>true</enabled></snapshots>

              </repository>

              <repository>

                <id>internal-releases</id>

                <name>Internal releases repo</name>

                <url>http://repo1.internal.argoid.com:8081/repository/maven-releases/</url>

                <releases><enabled>true</enabled></releases>

                <snapshots><enabled>false</enabled></snapshots>

              </repository>

              <repository>

                <id>internal-snapshots</id>

                <name>Internal snapshots repo</name>

                <url>http://repo1.internal.argoid.com:8081/repository/maven-snapshots/</url>

                <releases><enabled>false</enabled></releases>

                <snapshots><enabled>true</enabled></snapshots>

              </repository>

            </repositories>

           <pluginRepositories>

              <pluginRepository>

                <id>central</id>

                <url>https://repo1.maven.org/maven2/</url>

                <releases><enabled>true</enabled></releases>

                <snapshots><enabled>true</enabled></snapshots>

              </pluginRepository>

              <pluginRepository>

                <id>internal-releases</id>

                <name>Internal releases repo</name>
```

```
      <url>http://repo1.internal.argoid.com:8081/repository/maven-releases/</url>
      <releases><enabled>true</enabled></releases>
      <snapshots><enabled>false</enabled></snapshots>
    </pluginRepository>
    <pluginRepository>
      <id>internal-snapshots</id>
      <name>Internal snapshots repo</name>
      <url>http://repo1.internal.argoid.com:8081/repository/maven-snapshots/</url>
      <releases><enabled>false</enabled></releases>
      <snapshots><enabled>true</enabled></snapshots>
    </pluginRepository>
  </pluginRepositories>
</profile>
```

d. Under <activeProfiles> add
   i. `<activeProfile>nexus</activeProfile>`

Jenkins Backup

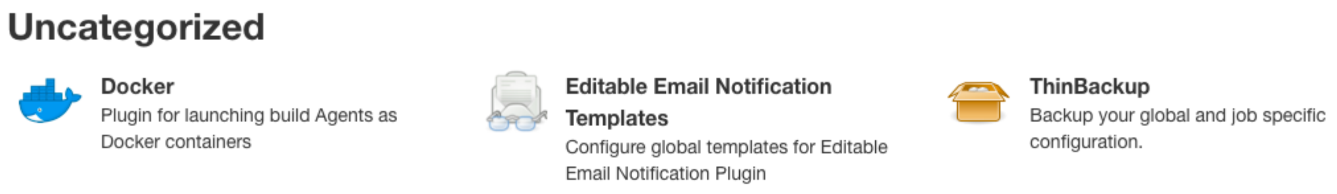
install ThinBackup plugin:

go to Manager Plugins Available search for ThinBackup plugin.

install & restart Jenkins

Configure ThinBackup

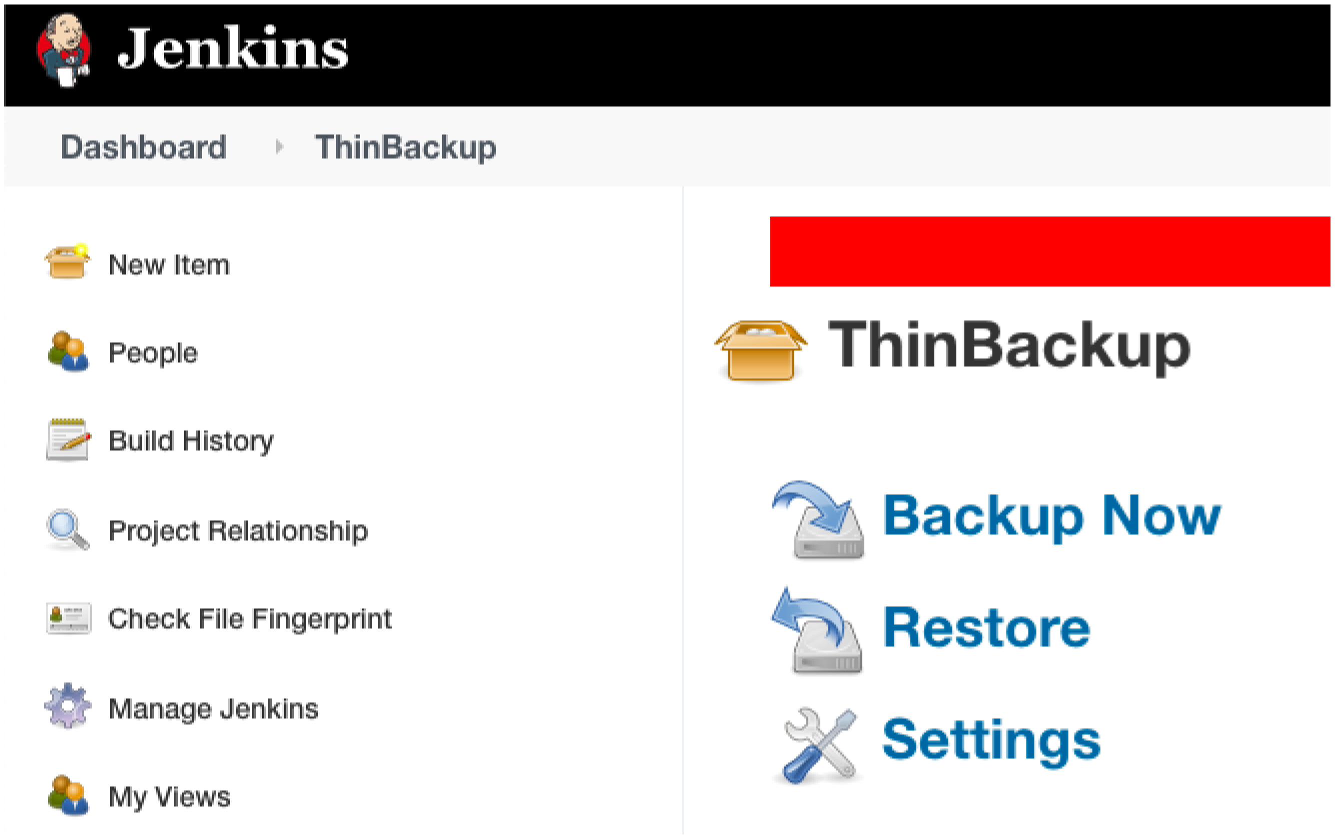
go to Manager Plugins Click on ThinBackup


Uncategorized

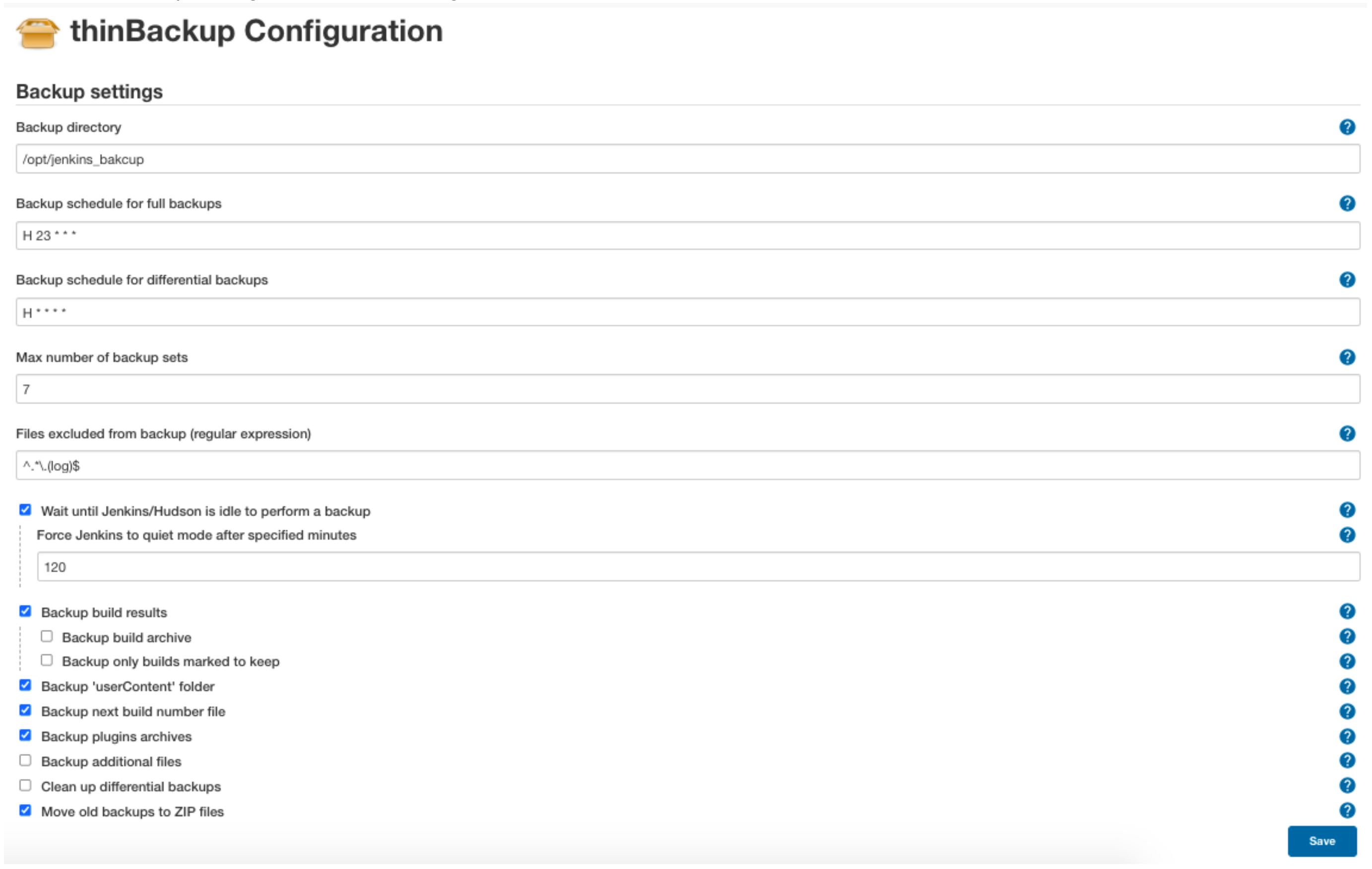
Docker
Plugin for launching build Agents as Docker containers

Editable Email Notification Templates
Configure global templates for Editable Email Notification Plugin

ThinBackup
Backup your global and job specific configuration.



Click on thinBackup Settings and fill the following details.



thinBackup Configuration

Backup settings

Backup directory: /opt/jenkins_bakcup

Backup schedule for full backups: H 23 * * *

Backup schedule for differential backups: H * * * *

Max number of backup sets: 7

Files excluded from backup (regular expression): ^.*\.(log)$

- [✓] Wait until Jenkins/Hudson is idle to perform a backup
  - Force Jenkins to quiet mode after specified minutes: 120
- [✓] Backup build results
  - [ ] Backup build archive
  - [ ] Backup only builds marked to keep
- [✓] Backup 'userContent' folder
- [✓] Backup next build number file
- [✓] Backup plugins archives
- [ ] Backup additional files
- [ ] Clean up differential backups
- [✓] Move old backups to ZIP files

Sync backup folder with Azure Blob store

Create a storage account in azure (Storage account name: **argoidinfrabackups**)

Create Container named **jenkinsbackup** in argoidinfrabackups storage account

Install the Azure CLI on Linux Jenkins box

ref link: https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-linux?pivots=dnf

Import the Microsoft repository key.

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

Create local `azure-cli` repository information.

```
echo -e "[azure-cli]
name=Azure CLI
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc" | sudo tee
/etc/yum.repos.d/azure-cli.repo
```

Install with the `yum install` command.

```
sudo yum install azure-cli
```

Configure Azure blob storage cron in Jenkins box

Configure the cron job as root user

```
[root@argoid-infra-jenkins-1 ~]# crontab -l
SHELL=/bin/bash
AZURE_STORAGE_KEY="
UilImJy48TPIgdwOCWXQAi9ovSEJCg5lqzJw24ZBra4VFYgOeo458HB+OdkLy43w0I9LEhD2
PwlwpoOVovoRiA=="
AZURE_STORAGE_AUTH_MODE="key"
AZURE_STORAGE_ACCOUNT="argoidinfrabackups"
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;
AccountName=argoidinfrabackups;
AccountKey=UilImJy48TPIgdwOCWXQAi9ovSEJCg5lqzJw24ZBra4VFYgOeo458HB+OdkLy
43w0I9LEhD2PwlwpoOVovoRiA==;EndpointSuffix=core.windows.net"

30 23 * * * /usr/bin/az storage azcopy blob sync --container
jenkinsbackup  -s /opt/jenkins_bakcup/
```