2022.06.17 - Setup Jenkins - by Tim Carr

Friday, June 17, 2022 8:52 AM

I followed this video: https://www.jenkins.io/doc/book/installing/windows/

Passwords: Build server 10.0.193.10, the Jenkins service, and the root MySQL user use the same user/pwd. User=lexagene_admin / Pwd=<see BitWarden>

1. **RDP into VM 10.0.193.10 as lexagene_admin**
2. Download Java (JDK 11) - Go to https://adoptium.net/temurin/releases/ and download the .msi of Temurin v11 for Windows x64.
3. Download Jenkins (2.355) - Go to https://www.jenkins.io .
    a. Originally tried v2.332.3 LTS but had issues upgrading Credentials plugin.
4. Install JDK(Temurin 11):
    a. JDK with Hotspot - ON
    b. Add to PATH - ON
    c. Associate .jar - ON
    d. Set JAVA_HOME variable - ON
    e. JavaSoft (Oracle) - OFF
    f. Change installation location to C:\tools\jdk\jdk-11.0.15.10-hotspot
5. Java environment tweeks:
    a. Open Environment Variables:
        i. JAVA_HOME - Remove the trailing slash from JAVA_HOME in System variable list.
        ii. PATH - replace "C:\Tools\JDK\jdk-11.0.15.10-hotspot\bin" with "%JAVA_HOME%\bin"
6. Open Local Security Policy
    a. Go to Local Policies -> User Rights Assignment -> Log on as a service.
    b. Add lexagene_admin
7. Install Jenkins
    a. Run jenkins.exe installer from downloads folder
    b. Change install directory to C:\tools\Jenkins_x_xxx\
    c. Run service as local or domain user. Insert user/pwd for lexagene_admin.
    d. Test credentials.
    e. Leave default port to 8080, click Test Port.
    f. Custom Setup, set Start Service to be **unavailable** (we have more config to do first)
    g. Custom Setup, leave Firewall Exception - OFF.
8. Configure Jenkins (before its' first start)
    a. Open C:\tools\Jenkins\jenkins.xml
        i. Set data location: <env name="JENKINS_HOME" value="C:\data\jenkins_home"/>
        ii. Set executable: <executable>%JAVA_HOME%\bin\java.exe</executable>
        iii. Set arguments: <arguments>-Xrs -Xms1g -Xmx2g -Djava.awt.headless=true -Djava.net.preferIPv4Stack=true -Djava.io.tmpdir=C:\tools\Jenkins_x_xxx\tmp\ -Dorg.apache.commons.jelly.tags.fmt.timezone=American/New_York -Duser.timezone=American/New_York -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle -jar "C:\tools\Jenkins_x_xxx\jenkins.war" --httpPort=8080 --webroot="C:\tools\Jenkins_x_xxx\war" --pluginroot="C:\tools\Jenkins_x_xxx\plugins"</arguments>

            1) Memory from 256 to 1GB min, 2GB max (video suggested setting both to 3GB but we have only 4 total)
            2) Add headless = true
            3) Add prefer IPv4
            4) Specify temp directory for Java so it doesn't use Windows'.
            5) Set timezones to New York.
            6) Change webroot folder.
            7) Add plugins folder.
        iv. Set PID file: <pidfile>C:\tools\Jenkins_x_xxx\jenkins.pid</pidfile>
    b. Open Services and start Jenkins service.
        i. Will say running, but when you refresh, did not start.
        ii. C:\tools\Jenkins now has some log files:
            1) jenkins.out.log - tells us it ran from jenkins.war. That's great.
            2) jenkins.wrapper.log - tells us some expected things, and that the PID wasn't found but then it created and saved it.
            3) jenkins.err.log - tells us Jenkins failed to create a file in our new specified but not created tmp folder. (need to create folder)
                a) Create folder: C:\tools\Jenkins\tmp
        iii. Start Jenkins service again. (err file will tell it started and installing some plugins.)
    c. Open browser: localhost:8080
        i. Use password from jenkins.err.log
        ii. Tell browser to never save.
        iii. Click "Install Suggested Plugins"
        iv. Set first Jenkins Admin to the same user/pwd as server. (lexagene_admin)
9. Open Task Manager - Processes, and sort by memory
    a. Notice the "OpenJDK Platform binary" is using between what we set in jenkins.xml (1-2GB).
10. THINGS TO KEEP BACKED UP:
    a. Jenkins installer
    b. JDK installer
    c. C:\data\jenkins_home
    d. C:\tools\Jenkins\jenkins.xml

WEBHOOKS (GitHub/Jenkins server integration):
This video was also good and included setting up secrets: How to Forward Webhooks to Jenkins Behind a Firewall using Smee.io
SMEE GitHub docs: https://github.com/probot/smee.io#faq

- Follow this tutorial: https://www.jenkins.io/blog/2019/01/07/webhook-firewalls/
- **Get new SMEE url**: (smee.io)
    - LexaGene Webhook Proxy URL is: https://smee.io/1gMzR0h37pLwWYcv
- **Install node.js and npm** via Windows installer: https://nodejs.org/en/download/ - node-v16.15.1-x64.msi
    - Leave all defaults (except install location=c:\tools\nodejs), including installing additional things after (python, choco, ...)
    - **LET IT RUN. Takes some time. Seems stuck but the PS window is doing things.**
    - *Does install 2019 VS Build tools. Jenkins is set to point to different VS Build Tool installation (2022), so shouldn't effec t build.
    - **npm needs to be run as admin in cmd.exe
    - Remove User Env Var.
    - Verify/Add: System Env Var = c:\tools\nodejs\
    - Add System Env Var: PM2_HOME = C:\tools\nodejs
    - Configure npm to be able to run from any user:
        - npm config set cache "C:\\tools\\nodejs\\npm-cache"
        - npm config set prefix "C:\\tools\\nodejs"
        - npm config set globalconfig "C:\\tools\\nodejs\\etc\\npmrc"

    - Install log file at: C:\ProgramData\chocolatey\logs\chocolatey.log
    - Update npm: npm install -g npm
- **Install SMEE client** (webhook proxy app/layer, to avoid opening ports on build server):
    - npm install --global smee-client
- **Test SMEE client on build server**:
    - smee --url https://smee.io/1gMzR0h37pLwWYcv --target http://localhost:8080/github-webhook/
    - smee --url https://smee.io/1gMzR0h37pLwWYcv --path /github-webhook/ --port 8080 (also works but use other)

- Schedule Task: (start SMEE on system startup and run forever. Retry if dies)
    - Open Windows Task Scheduler
    - Right-click on Task Scheduler Library and select, Create Basic Task.
        - Name = Start SMEE client on system startup
        - Trigger = When the computer starts
        - Action = Start a program
        - Finish = C:\tools\smee\start_smee.bat
        - Select to open Properties on finish
    - Properties:
        - General tab:
            - □ Name = Start SMEE client on system startup
            - □ User = lexagene_user
            - □ Run whether user is logged on or not
        - Conditions tab (check only what is listed below, uncheck all others):
            - □ Start the task only if the computer is on AC power
        - Settings tab (check only what is listed below, uncheck all others):
            - □ Allow task to be run on demand.
            - □ If the task fails, restart every - 1 minute, and attempt 999 times.
            - □ If the running task does not end when requested, force it to stop.
            - □ If the task is already running - Do not start another instance.
- GitHub Status Checks: (I DON'T THINK THIS ONE IS NECESSARY SINCE NOW USING GitHub-Branch-Source plugin)
    - Create GitHub App named "Jenkins - LeXml"
        - Follow: https://github.com/jenkinsci/github-branch-source-plugin/blob/master/docs/github-app.adoc
- GitHub Branch Source Plugin (Jenkins)
    - I finally realized this is what most DevOps teams will migrate to.
    - https://docs.cloudbees.com/docs/cloudbees-ci/latest/cloud-admin-guide/github-branch-source-plugin

- Things to discuss:
    - ☑ Get PRs pushing to 10.0.0.5
    - ☑ **Full clean** or just pull for each build? (2 sec vs 20 sec checkout)
    - ☑ Artifact structure:

Branch commits:	10.0.0.5\builds\{REPOSITORY}\{BRANCH_NAME}\Major.Minor.Patch.Build	
PRs:	10.0.0.5\builds\{REPOSITORY}\{TARGET_BRANCH_NAME}\Major.Minor.Patch.Build-PR-XXX	Source branch not as easy to know.

    - ☑ Use same build number across branches. unique per repo.
    - ☑ Remove old builds from LexmlInstaller folder?
    - ☑ Remove LexmlInstaller project form Lexml.sln? (should I replace with EXE project to just replace files in c:\ProgFiles\Lexagene\LeXml ?)
    - ☑ Backup build server files.
    - ☑ How many old builds to keep on build server? (delete builds on branch delete)
        - Cron to clean N: drive occasionally? Or every time we push, allow X on master/release/hotfix builds and delete rest?
        - On delete branch action, remove builds for that branch.

☑ ○ Switch creds to Software@lexagene.com

□ ○ Once in a while I noticed initial build failed checkout creds, then second always worked. Keep eye out to see if persists.

Jenkins runs as a service.
SMEE runs as Task Scheduler.

☑ MySQL no pwd on build machine. Repo table and buildnumber table.

--- IGNORE BELOW. Notes taken while heading down pitfall.--

Maybe try this for folder to have full version: https://stackoverflow.com/questions/5248597/how-to-get-jenkins-to-copy-artifacts-to-a-dynamic-directory

Manage Jenkins -> Manage Credentials:

- Manage Jenkins -> Credentials -> click Jenkins Store -> Global Credentials (unrestricted) -> Add Credential
- Scope=Global
- Kind = SSH Username with private key
- Description=GitHub to Jenkins - SSH
- Username=timcarr
- Private Key - select Enter directly. Copy entire private key file contents into text box.
- Did not use a passphrase for this key. Maybe should.

Creating a Jenkins Freestyle Project:
https://www.codeproject.com/Articles/878203/Integrate-Jenkins-with-MSBuild-and-NuGet

- Install nuget v6.2.1 (www.nuget.org/downloads)
- Install MSBuild (via Build Tools for Visual Studio 2022) - https://visualstudio.microsoft.com/downloads/
  - Only need to include ".NET Desktop Development" during install.
- Install Jenkins "MSBuild" plugin.
- Go to Global Tool Configuration -> MSBuild -> Add MSBuild
  - Name=MSBuild 2022 (17.2.1.25201)
  - Path=C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools\MSBuild\Current\Bin
  - Save.
- Dashboard -> New Item
  - Description=LeXml application build.
  - GitHub project - Project url=https://github.com/LexaGene/LeXml/
  - Source Code Management:
    - Git
      - Repository URL=git@github.com:LexaGene/LeXml.git
      - Credentials=timcarr (GitHub to Jenkins - SSH)
    - Branches to build
      - Branch specifier=*/jenkins-test
  - Build Environment
    - Delete workspace before build - Advanced
    - Pattern for files to be deleted
      - Add - Include - **/bin/**
      - Add - Include - **/obj/**
      - Add - Include - **/LeXmlInstaller/Release/**   (in case we build this in future)
      - Add - Include - **/LeXmlInstaller/Debug/**   (in case we build this in future)
  - Build:
    - Update AssemblyVersion/AssemblyFileVersion files: https://stackoverflow.com/questions/33496781/how-to-set-assembly-version-to-jenkins-build-number
      - Add build step -> Change Assembly Version (change-assembly-version-plugin)
      - Assembly Version=$BUILD_NUMBER
      - RegexPattern=Assembly(\w*)Version\("(\d+).(\d+).(\d+).(*)"\)
      - ReplacementPattern=Assembly$1Version("$2.$3.$4.%s")
    - Nuget restore: https://www.codeproject.com/Articles/878203/Integrate-Jenkins-with-MSBuild-and-NuGet
      - Add build step -> Execute Windows batch command
      - C:\tools\nuget\nuget.exe restore C:\data\jenkins_home\workspace\LeXml\Lexml.sln
    - Build Lexml.sln
      - Add build step -> Build a Visual Studio project or solution using MSBuild
      - MSBuild Version=MSBuild 2022 (17.2.1.25201)
      - MSBuild Build File=Lexml.sln
      - Command Line Arguments=/t:Clean,Build /p:Configuration=Release /p:Platform="Any CPU"
  - Post-build Actions
    - Archive the artifacts
      - Files to archive=Lexml\bin\Release\*

            - Advanced - Archive artifacts only if build is successful - check. (leave other defaults)

- Install PM2 and set to start SMEE on server startup (not user login)
  Follow tutorial: https://stackoverflow.com/questions/42758985/windows-auto-start-pm2-and-node-apps
  - You already installed node and npm to location accessible to all users, so skip "**Prerequisites (part # 1)**" (it messed me up a few times so don't touch)
  - I also skipped Prereq #2 part b. Trying to avoid touching what node installer did since when I did, npm installation got messed up.
  - *FYI: Because of user vars, npm\node_modules is at: C:\Users\lexagene_admin\AppData\Roaming\npm\node_modules
  - cd "C:\Users\lexagene_admin\AppData\Roaming\npm\node_modules\pm2-windows-service"
  - pm2-service-install -n SMEE_PM2_AUTORUN

- Set SMEE to start with windows as a service:
  - npm install pm2 -g
  - pm2 start C:\Users\lexagene_admin\AppData\Roaming\npm\smee --url https://smee.io/1gMzR0h37pLwWYcv --target http://localhost:8080/github-webhook/
  - .....this is turning out to be impossible. this webpage doesn't work out: https://github.com/probot/smee-client/issues/118
  - Use node.js and smee-client's programmatic API to write a node program, that starts smee, that can be run by pm2, and then set pm2 to start that shit at Windows startup. (not user log in.)
  - https://www.npmjs.com/package/smee-client
  - Restart pm2 and my app on server reboot:
    - https://stackoverflow.com/questions/42758985/windows-auto-start-pm2-and-node-apps
    - My app: C:\Users\lexagene_admin\AppData\Roaming\npm\start_smee.js
  - removed this from User vars: C:\Users\lexagene_admin\AppData\Roaming\npm