# Server Installation Guide

# Installation Guide

## IMPORTANT:

**Do NOT reboot the machine until you finish step 4**. You will almost certainly crash the server with inconsistent dependencies. You need to finish loading the prepare script and then do the DNF / YUM update before a reboot can happen. After the upgrade to 8.10 (which happens during the update) is done, then rebooting is safe.

# Installation Steps:

1. **Download the AlmaLinux 8.6 x86_64 dvd.iso**
    1. https://repo.almalinux.org/vault/8.6/isos/x86_64/AlmaLinux-8.6-x86_64-dvd.iso
    2. Create a bootable flash drive with **AlmaLinux 8.6**. Create the bootable drive using Rufus - 4.
    3. Plug the drive into your server and load up the BIOS to set the USB drive as your boot media.

2. **Server Resources**
    1. Ensure you have enough server resources when creating your VM or using a physical server. The following is a recommendation for minimum system requirements.
    128 GB Storage Capacity (The repos are massive, and the DB grows over time)
    6+ Core CPU
    16GB Ram (32GB would be ideal)

3. **Start the AlmaLinux Installation**
    1. Set the **root** password
    2. Set the user account with the following details
        1. Username: swg
        2. Full Name: swg
        3. Password: <set to what ever you want>
        4. Make the account an ADMIN account.
    3. Set the Network to connect to your ethernet, write down the IP assigned (this can be changed later)
        1. Set the hostname: swg

4. Select the drive to install AlmaLinux. Delete all partitions.
5. Continue the installation process. This will take several minutes. The server will ask to reboot after.
6. Once rebooted, pull the flash drive out, you're done with that.
7. Accept the terms of service then log into your **swg** account.

4. **First Time Setup (Terminal)**
   1. Update hostname. This step has two commands. The first command will run, then the second command will require you to edit a file using **vim**.

```
sudo hostnamectl set-hostname swg


sudo vim /etc/hosts


## Add your IP address with a TAB then your swg user as
## a new entry to the end of the file, here is an example
## of a new entry
192.168.1.1    swg
```

   2. Update sudoers. This step will require you to edit a file. It uses a form of **vim**.

```
sudo visudo


## Search for the line that looks like this:
## root    ALL=(ALL)
## copy this line and duplicate it. Now change root to swg on the
## duplicated line. here is an example of that (ignoring the ...):


swg ALL=(ALL) ...
```

   3. AlmaLinux 8.6 has outdated **GPG keys** (as of 2024). The first two commands will update those keys.
      Then we will install **git**.
      Finally, we will clone the **SWG Prepare** scripts and run the **main.sh** script which starts the installation process.
      **Note:**
      To use this script from my GitLab you'll need to create the SSH Key and input that into GitLab.

```
## Make a backup of the GPG Keys file
sudo mv /etc/pki/rpm-gpg/RPM-GPG-KEY-AlmaLinux /etc/pki/rpm-gpg/RPM-GPG-KEY-
AlmaLinux.bak
```

```
## Download the new GPG Keys
sudo wget -O /etc/pki/rpm-gpg/RPM-GPG-KEY-AlmaLinux
https://repo.almalinux.org/almalinux/RPM-GPG-KEY-AlmaLinux

## Install GIT (should work now)
sudo dnf instal -y git

## (optional) run the keygen if you need to add your SSH keys to GitHub or GitLab
## only do this if the `git clone` below doesn't work
## copy the ssh-key from id_rsa.pub into GitLab before cloning.
ssh-keygen

## Clone the repository for swg-prepare
git clone https://github.com/SniperFox22/swg-prepare.git

## Now run SWG-Prepare:
./swg-prepare/main.sh
```

**THE MOST CRUCIAL PART IS TO GET THROUGH THE PREPARE SCRIPT (THERE WILL BE ISSUES).**

5. **FIX THE ISSUES (AFTER PREPARE SCRIPT!!!)**
    1. Install 32-bit libraries (Very Important)

```
sudo dnf install -y glibc.i686 libstdc++.i686 zlib.i686 gtk2.i686 libXtst.i686 libXrender.i686
libcanberra-gtk2.i686 libcanberra-gtk3.i686
```

    2. Download and install the **Boost 1.81** from the link provided (or from the tar.gz provided)
        1. https://www.boost.org/
        2. Boost 1.81
        3. **Installation of Boost**

```
## Install development tools
sudo dnf groupinstall "Development Tools"

sudo mkdir -p /opt/boost

sudo mv ~/Downloads/boost_1_81_0.tar.gz /opt/boost/

cd /opt/boost/
```

```
sudo tar -xvf boost_1_81_0.tar.gz

cd boost_1_81_0

sudo ./bootstrap.sh --prefix=/usr/local

sudo ./b2 install --with-program_options
```

3. **(OPTIONAL)** Download and install the Java 17 (for 32-bit) from the link provided (or from the tar.gz provided)
    1. https://www.azul.com/downloads/?package=jdk#zulu
    2. Notes: the 32-bit version will be **i686.**
    3. **Java 17 (32-Bit)**
    4. **Installation of Java**

    *This upgrade is now marked as OPTIONAL. In my late experience Java 17 (i686) is very unstable and will cause the server to crash with memory crashing issues quite often. Upgrade at your own risk. Your alternative is to use the default Java 11, which is zulu-11-x86, the 32-bit version of Java 11.* ***I would highly recommend skipping Java 17 and skipping this step...***

```
sudo mkdir -p /opt/java

sudo mv ~/Downloads/zulu17.58.21-ca-jdk17.0.15-linux_i686.tar.gz /opt/java/

cd /opt/java/

sudo tar -xvf zulu17.58.21-ca-jdk17.0.15-linux_i686.tar.gz

## Update profile
sudo vim /etc/profile.d/jdk-zulu.sh

# Set JAVA_HOME for Zulu OpenJDK 17
export JAVA_HOME=/opt/java/zulu17.58.21-ca-jdk17.0.15-linux_i686
# Add JAVA_HOME/bin to the PATH
export PATH=$JAVA_HOME/bin:$PATH

source /etc/profile.d/jdk-zulu.sh
```

```
## INSTALL JAVA

sudo alternatives --install /usr/bin/java java /opt/java/zulu17.58.21-ca-jdk17.0.15-linux_i686/bin/java 20000

sudo alternatives --install /usr/bin/javac javac /opt/java/zulu17.58.21-ca-jdk17.0.15-linux_i686/bin/javac 20000

sudo alternatives --install /usr/bin/jar jar /opt/java/zulu17.58.21-ca-jdk17.0.15-linux_i686/bin/jar 20000

## Set JAVA 17 as the version to use
sudo alternatives --config java

sudo alternatives --config javac

java -version

## NOTES
## YOU WILL NEED TO UPDATE THE taskManager.cfg FILE (or servercommon.cfg file)
```

4. **(OPTIONAL)** Download and Install Oh My Zsh.
   1. First install `zsh`
      Copy the JAVA path from the previous step into the **.zshrc** file.

      ```
      sudo dnf install zsh -y

      # Then install omzsh

      sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
      ```

   2. Setup `.zshrc` with the theme **`bira`**.
   3. **(OPTIONAL) Setup Java 17**

      *If you upgraded to Java 17 in the optional step above, then you should perform this step, otherwise you don't need to do anything here.*

```
# Set JAVA_HOME for Zulu OpenJDK 17
export JAVA_HOME=/opt/java/zulu17.58.21-ca-jdk17.0.15-linux_i686
# Add JAVA_HOME/bin to the PATH
export PATH=$JAVA_HOME/bin:$PATH
```

4. Add the `swg-main` as the working directory.

```
# Every Login to SSH will take me to the swg-main directory
cd /home/swg/swg-main
```

5. Then `source .zshrc`.

5. **(OPTIONAL)** Setup Tailscale
   1. Disable Expiry
   2. Set the machine name, if applicable, i.e., "swg" --> "swg-secura".
   3. Now start tailscale: `sudo tailscale up`

6. Finally do one last update and reboot.
   1. Update DNF and YUM
      This will take a LONG time as there are like 1300 libraries that will take the AlmaLinux 8.6 to 8.10, but it is necessary to do this update AFTER the prepare script has run at least once.

```
sudo dnf update -y && sudo dnf upgrade -y
sudo yum update -y && sudo yum upgrade -y
sudo reboot now
```

6. **(OPTIONAL)** Add Authorized Keys
   1. Add the generated key to SSH Keys in GitLab, i.e., copy the key from `~/.ssh/id_rsa.pub`.
      1. If you created the `authorized_keys` file then you'll have to chmod it to 600.

```
vim ~/.ssh/authorized_keys


sudo chmod 600 ~/.ssh/authorized_keys
```

   2. Setup sshd_config for PubKeyAuthentication.

```
sudo vim /etc/ssh/sshd_config


PubKeyAuthentication yes


sudo systemctl restart sshd
```

7. **(OPTIONAL)** Setup **Remote Desktop**

1. Install dependencies for remote desktop. This should be done after the big DNF / YUM update as it requires libraries from the 8.10 group.

```
sudo dnf install epel-release -y
sudo dnf groupinstall "Xfce" -y
sudo dnf install xrdp -y
sudo vim /etc/xrdp/startwm.sh

----------

#!/bin/sh

if [ -r /etc/profile ]; then
        . /etc/profile
fi

#test -x /etc/X11/Xsession && exec /etc/X11/Xsession
#exec /bin/sh /etc/X11/Xsession

exec startxfce4

----------

sudo systemctl enable xrdp --now
sudo systemctl status xrdp
```

2. Now you can remote desktop using windows remote desktop tool.

8. **(OPTIONAL)** Install **Google Chrome** on the desktop, so we can log in and get access to the wiki and things.

9. **(!!!IMPORTANT!!!) Database Setup & Fixes**
    1. The install process has issues every time you re-run the prepare script. It'll always add this line at the bottom of the file, so you may have to go in and edit it so that it shows the **/usr/lib/jvm/zulu-11**. It's important we use this version as the **sqldeveloper.sh** requires a java 11, 32-bit version, it won't run otherwise.

```
/opt/sqldeveloper/sqldeveloper.sh

If you run into an issue opening this then modify the file:
sudo vim /opt/sqldeveloper/sqldeveloper/bin/sqldeveloper.conf
```

```
# Update this line to look like this:
SetJavaHome /usr/lib/jvm/zulu-11
```

2. When we run the **swginit.sh** script from the **swg-prepare** directory, it'll try to overwrite the `servercommon.cfg` file. We need to do some commands so that our **ant swg** command will run properly, essentially we may need to finish the process.

```
## AFTER the servercommon.cfg file is moved
cd /home/swg/swg-main/exe/linux/;

git config --global user.name "SWG User"
git config --global user.email "swg-user@swg.com"

git add . ;
git commit -m "Initializing servercommon.cfg"
```

3. Occasionally the database may become locked for what ever reason. Here is the process to unlock it. Essentially we have to log into the oracle user, use the database password set by the **swg-prepare** scripts and then unlock the **swg** account every time it gets locked.

```
sudo -i -u oracle
sqlplus sys as sysdba

## PASSWORD IS : swgevolve OR swgsecura (swgsecura if used my prepare script)
## USERNAME IS: oracle (if you ever needed to know)

ALTER USER swg ACCOUNT UNLOCK;

## this will reset the password back to swg (IDENTIFIED BY is the password).
ALTER USER swg IDENTIFIED BY swg;

exit
```

10. **FINAL STEPS**
    1. The Database is a huge pain in the ass so we may need to run the following commands over and over until minor details are fixed. It's **IMPORTANT TO NOTE** that we only need the prepare script to run to get the basic dependencies for the server. i.e., oracle, and other compilation dependencies. But everything else can be

resolved by using the **ant swg** command, and starting from this point.

This is why we run the prepare script first before we do the **dnf update**. So that we get the proper dependencies that are only available from 8.6. After we finish with that we can update all we want. In fact we shouldn't reboot until we update as this will crash the server.

```
# this command will initialize the repository and database, pulling all the code
# and loading all the default database stuff
ant swg

# if we just want to reload the database to resolve those issues then just run this one.
ant create_database

# finally, when the database is all sorted out and we've pulled all the code we can compile it all
ant compile

# additionally we can use the clean function to clear all compilation and builds, this sometimes helps
# but then you have to run compile again.
ant clean
```

2. **Open the Firewall Ports**
   For those that are using a firewall, you'll need to open the ports:

```
# Open each port 44451-44453, 44462-44464 for TCP and UDP
sudo firewall-cmd --add-port=44451/tcp --permanent
sudo firewall-cmd --add-port=44451/udp --permanent
sudo firewall-cmd --add-port=44452/tcp --permanent
sudo firewall-cmd --add-port=44452/udp --permanent
sudo firewall-cmd --add-port=44453/tcp --permanent
sudo firewall-cmd --add-port=44453/udp --permanent
sudo firewall-cmd --add-port=44462/tcp --permanent
sudo firewall-cmd --add-port=44462/udp --permanent
sudo firewall-cmd --add-port=44463/tcp --permanent
sudo firewall-cmd --add-port=44463/udp --permanent
sudo firewall-cmd --add-port=44464/tcp --permanent
sudo firewall-cmd --add-port=44464/udp --permanent

# Reload the firewall
```

```
sudo firewall-cmd --reload
```

11. **(OPTIONAL) Upgrade to AlmaLinux 9.6**
    This is a completely optional part, but I have validated that you can upgrade to AlmaLinux 9.6 which is a much more stable operating system than AlmaLinux 8.10. I would HIGHLY recommend using an AI to walk you through the process as it'll be slightly different per machine. Just use the following prompt.

    You must be absolutely certain you want to go through this process as there is always a small chance that things will break upgrading to a major version i.e., 8 to 9. GOOD LUCK.

    > I want to upgrade my AlmaLinux 8.10 (Desktop) server to AlmaLinux 9.6 using the CLI. I am using zsh as my shell. Please walk me through the process step by step to perform this upgrade using leapp.

# Server Startup

```
sudo -i -u oracle
sqlplus / as sysdba
startup
exit
lsnrctl start
exit

cd /home/swg/swg-main/chat
./stationchat &

cd /home/swg/swg-main
ant start &
```

---

Revision #32
Created 31 December 2023 01:52:13 by Brandon Calabro
Updated 30 May 2025 17:12:25 by Brandon Calabro