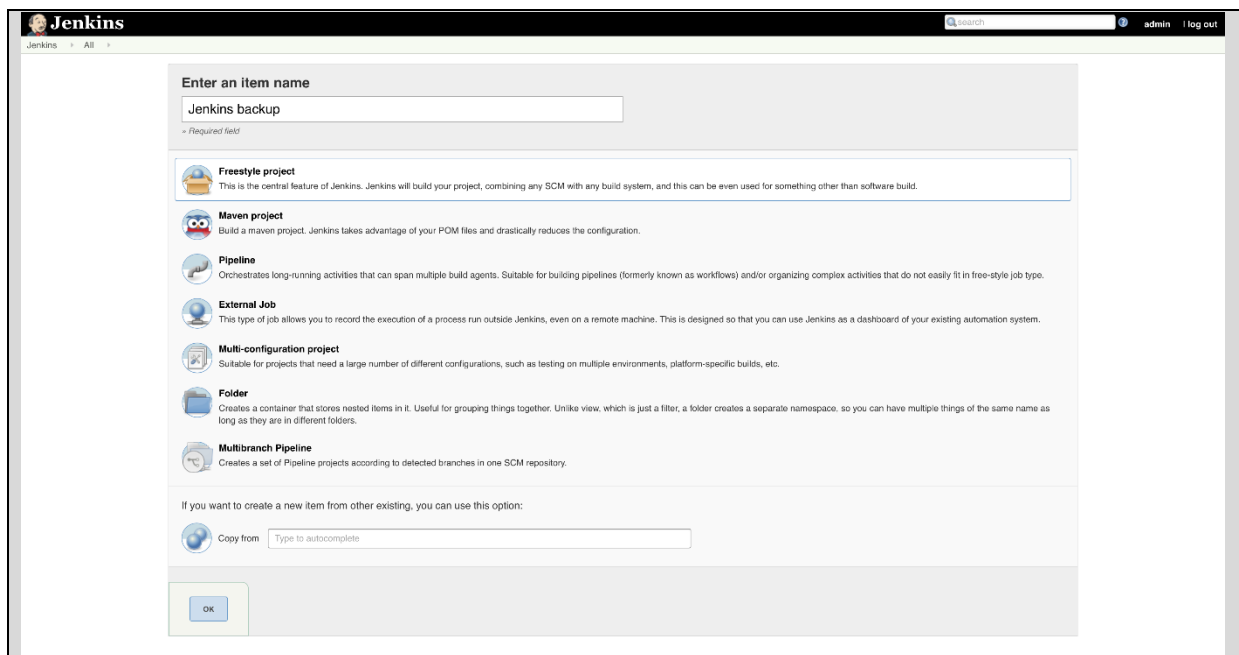


# How to Backup and Restore Jenkins Server

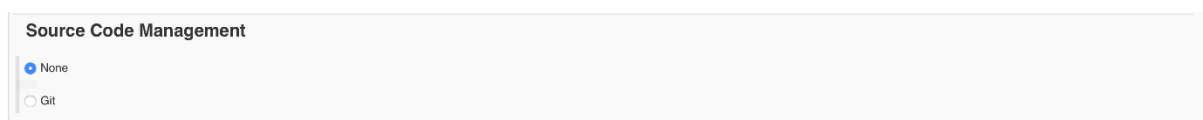
## Configure the regular backup job.

### **Step 1. Create a new Jenkins Job.**

You should choose Freestyle project.



### **Step 2. Mark “None” for Source Control Management.**



**Step 3.** Let's add a job which will completely backup Jenkins including all jobs, playbooks, whatever else you have there. It will backup everything located at **/var/lib/jenkins**. Select the “Build Periodically” build trigger and configure to run as frequently as you like

**Build Triggers**

☐ Trigger builds remotely (e.g., from scripts)  
☐ Build after other projects are built  
☒ Build periodically

Schedule

⚠ Spread load evenly by using 'H 12 \* \* \*' rather than '25 12 \* \* \*'  
 Would last have run at Friday, January 18, 2019 12:25:30 PM UTC; would next run at Saturday, January 19, 2019 12:25:30 PM UTC.

☐ GitHub hook trigger for GITScm polling  
☐ Poll SCM

For example, 25 12 \* \* \* will run a backup job every day at 12.25 pm.

**Step 4. Add a new “Execute Shell” build step and add the content of [this file](#) as the command.**

**Build**

**Execute shell**

Command

```
#!/bin/bash
# Jenkins Configuraitons Directory
cd $JENKINS_HOME

# Add general configurations, job configurations, and user content
git add -- *.xml jobs/*/*.xml userContent/* ansible/*

# only add user configurations if they exist
if [ -d users ]; then
  user_configs="ls users/*/config.xml"
  if [ -n "$user_configs" ]; then
    git add $user_configs
  fi
fi

# mark as deleted anything that's been, well, deleted
to_remove= git status | grep "deleted" | awk '{print $3}'
if [ -n "$to_remove" ]; then
  git rm --ignore-unmatch $to_remove
fi

git commit -m "Automated Jenkins commit"
git push -q -u origin master
```

See [the list of available environment variables](#)

Advanced...

Add build step ▾

**Step 5. Save all changes**

**Post-build Actions**

Add post-build action ▾

Save Apply

**Step 6. Go to the directory /var/lib/jenkins and initiate a new git repository.**

```
sudo su -

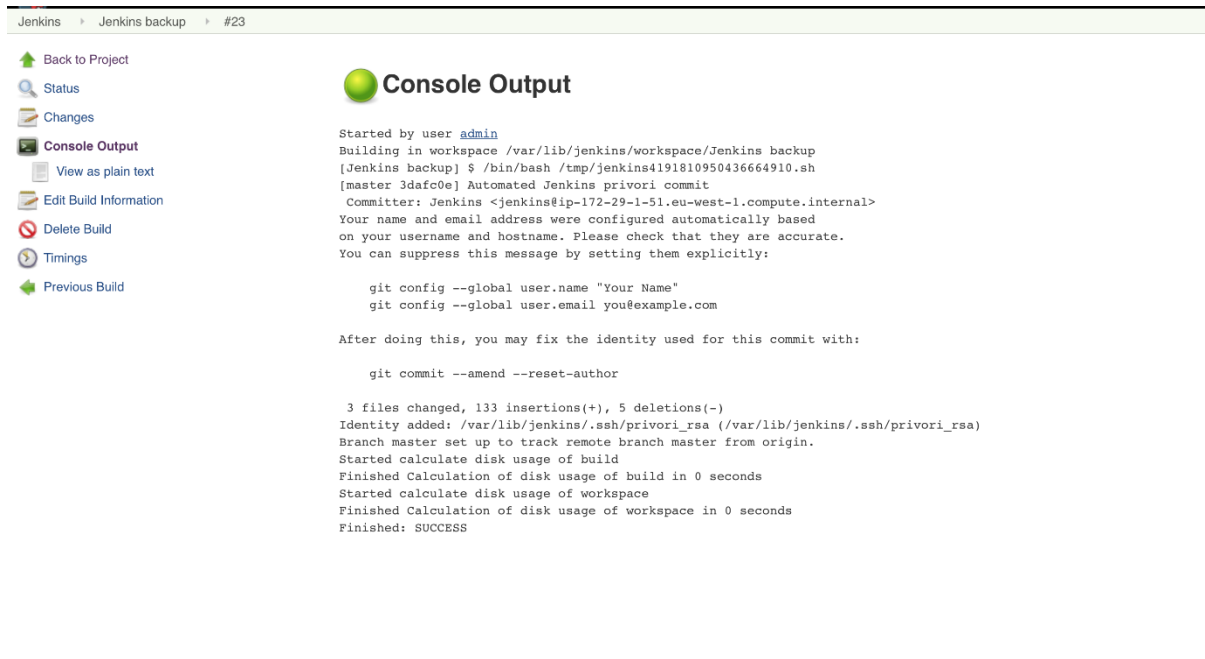
cd /var/lib/jenkins && git init
```

## Step 7. Connect your local repository to GitHub.

```
git remote add origin git@github.com:prakashk0301/new_repo
```

## Step 8. Test your Jenkins backup job

Click “Build now” button. If successful, you should see something like this:



The screenshot shows the Jenkins web interface for a job named 'Jenkins backup'. The left sidebar contains links: 'Back to Project', 'Status', 'Changes', 'Console Output' (selected), 'View as plain text', 'Edit Build Information', 'Delete Build', 'Timings', and 'Previous Build'. The main area is titled 'Console Output' and displays the following text:

```
Started by user admin
Building in workspace /var/lib/jenkins/workspace/Jenkins backup
[Jenkins backup] $ /bin/bash /tmp/jenkins4191810950436664910.sh
[master 3dafc0e] Automated Jenkins privori commit
Committer: Jenkins <jenkins@ip-172-29-1-51.eu-west-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

3 files changed, 133 insertions(+), 5 deletions(-)
Identity added: /var/lib/jenkins/.ssh/privori_rsa (/var/lib/jenkins/.ssh/privori_rsa)
Branch master set up to track remote branch master from origin.
Started calculate disk usage of build
Finished Calculation of disk usage of build in 0 seconds
Started calculate disk usage of workspace
Finished Calculation of disk usage of workspace in 0 seconds
Finished: SUCCESS
```

## How to restore Jenkins from backup:

### Step 1. Delete all from jenkins home directory

```
cd /var/lib/jenkins && rm -rf
```

**Step 2. Go to the directory /var/lib/jenkins and initiate a new git repository**  
(or you can create new Jenkins Instance in case if old Jenkins Instance already crashed)

```
cd /var/lib/jenkins && git init
```

**Step 3. Cleans the working tree by recursively removing files that are not under version control**

```
git clean -df
```

**Step 4. Add a new remote**

```
git remote add origin git@github.com:prakashk0301/new_repo
```

**Step 5. Pull all data from GitHub**

```
git pull origin master
```

**Step 6. Restart Jenkins daemon as root user**

```
service jenkins restart
```

---

There is another method to backup jenkins data.

thin backup plugin can be used to backup jenkins