Ansible + Jenkins

Step 1: Primary Needs

1. Jenkins installed
2. Git scm and parameterized build plugin installed in Jenkins
3. One working demo project with successful built.
4. Ansible installed
5. Ssh installed

Step2: Jenkins configuration

1. By default Jenkins uses “jenkins” name user to do its all task.
2. So, give sudo access and remove sudo password for “jenkins” user. This can be achieved by editing /etc/sudoers file. In this file insert following line

jenkins ALL=NOPASSWD: ALL

1. On Terminal switch user to Jenkins

su Jenkins

1. Create SSH public and private keys for “jenkins” user

ssh-keygen

1. Distribute public key to remote machine in which you have to deploy your built.

ssh-copy-id username@ip\_address

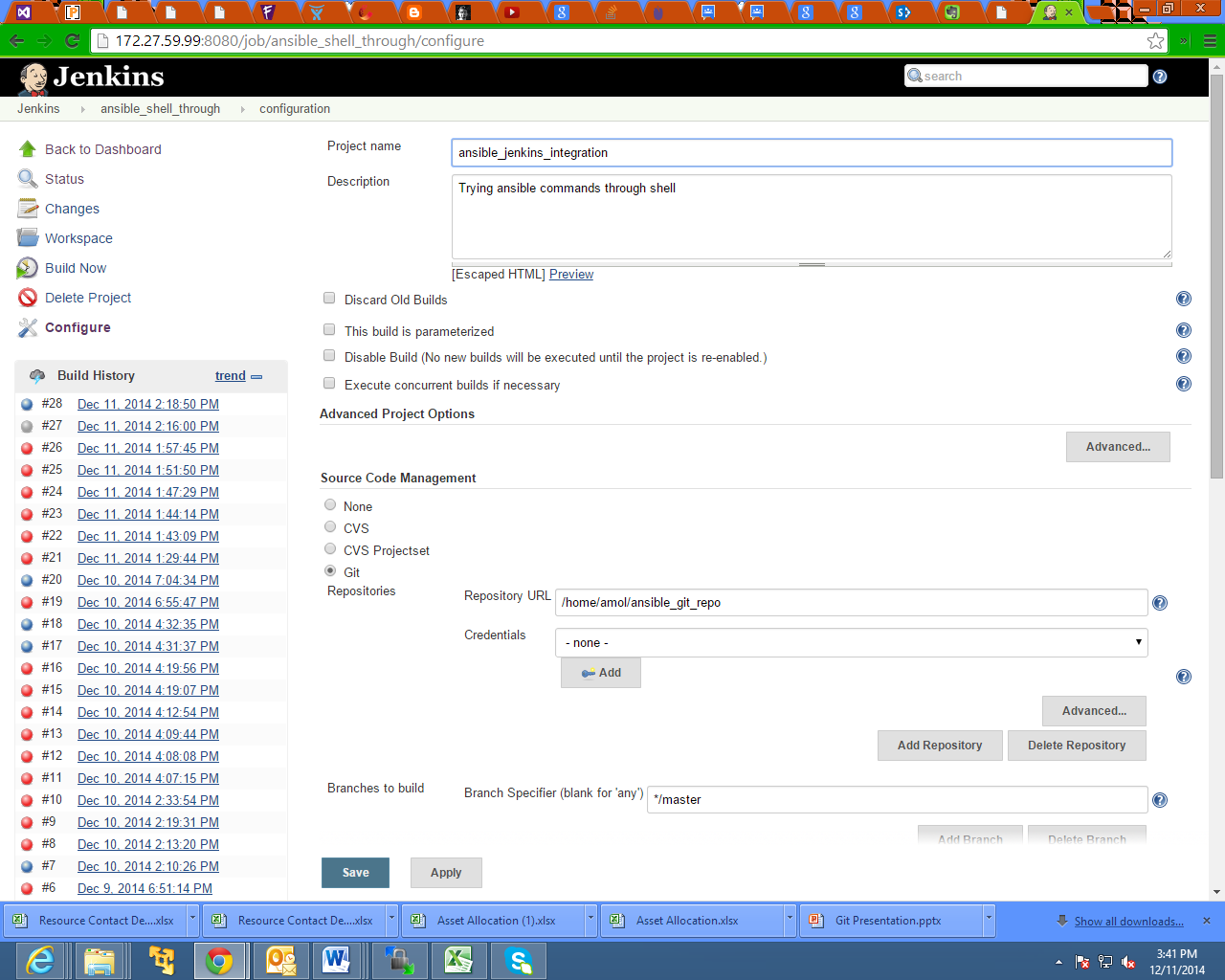
step3: Create git repo “ansible\_git\_repo” and and add ansible playbook in it which you are going to trigger after your project built

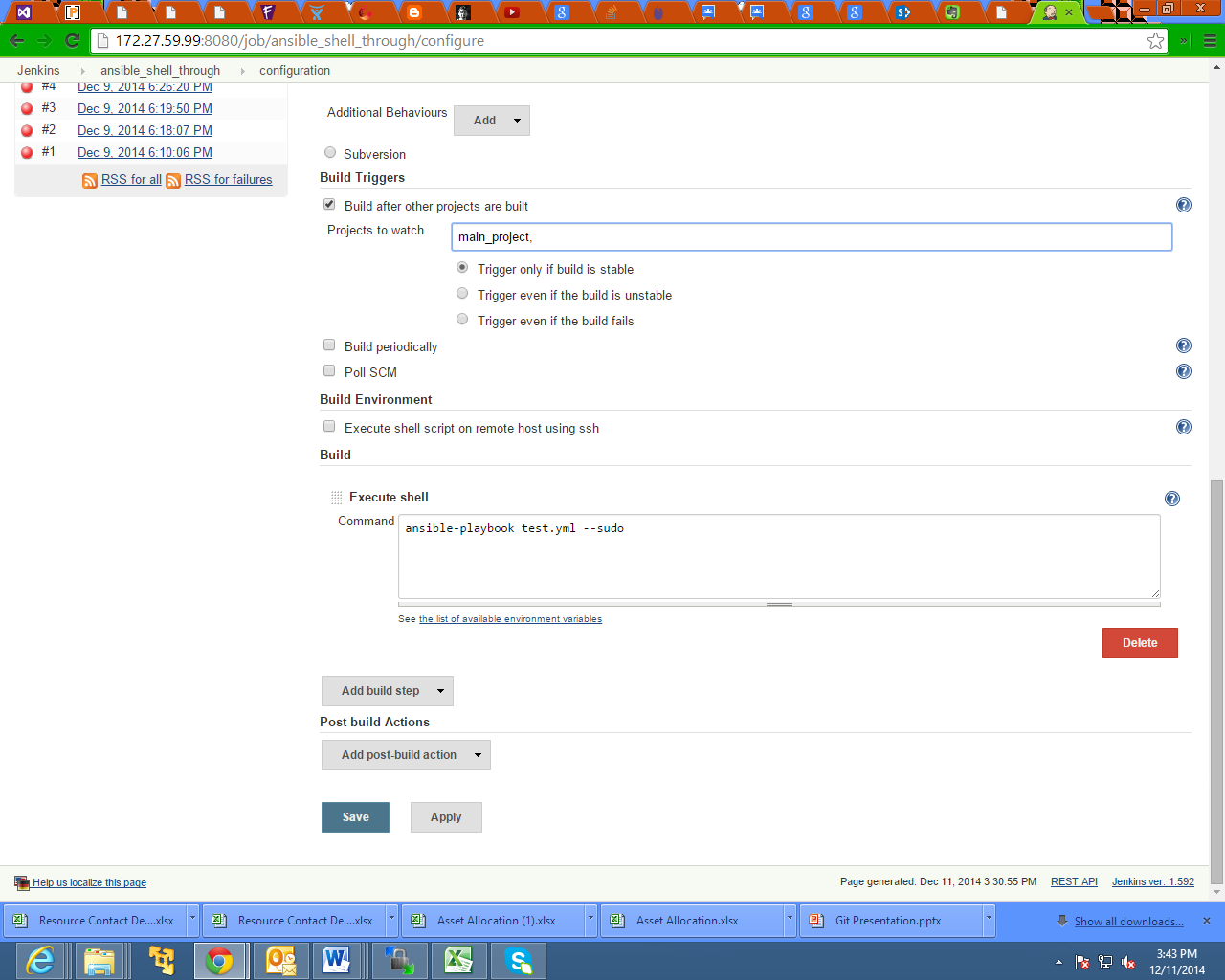
step4: Create job “ansible\_jenkins\_integration” in Jenkins and edit its configuration as follows

1. In this jobs configuration select ‘source code management’ to git and give path to “ansible\_git\_repo”
2. Select Build Trigger as “Build After other projects are built” and give name of job in which your demo project is built say “main\_project”
3. Add Build Step as “Execute Shell” and In Shell box give ansible command which runs your playbook

Ex ansible-playbook test.yml --sudo

1. Save configuration





Step5: This step should be done on remote machine and for remote user (the user through which you are connecting through ansible)

1. Give sudo access to remote user and remove sudo password for remote user (say ‘amol’). This can be achieved by editing /etc/sudoers file. In this file insert following line

amol ALL=NOPASSWD: ALL

Ansible Tower + Jenkins

Step 1: Following should be installed and configured

1. Ansible Tower
2. Jenkins

Step 2: Install and configure tower CLI

pip install ansible-tower-cli

tower-cli config ---> Command for configuring tower cli (you have to pass parameters in key value pair)

Note: The credential which you will give while configuring Tower CLI should match with credential given in Ansible Tower

It creates “.tower\_cli.cfg” in your home directory

Step 3: copy “.tower\_cli.cfg” file to Jenkins user home directory. This file act as key for ansible tower and Jenkins integration.

At this point you should be able to kick off Ansible Tower jobs from the command line within your Jenkins server using the Jenkins user.

Next, all you need to do add a “Execute shell’ build step with the following:

/usr/bin/tower-cli job launch --job-template=XX

\* Where XX is the ID of the Job Template you wish to run.