Demo test -- <https://github.com/watchmen-test/webdeploy>

Deployment procedures and different scenarios for getting the website running :

How :

1. Three servers can be created for running as three instances and the web site requests can be routed through these servers and the website page will get appeared.

Architecture Solutions:

Solution-1:

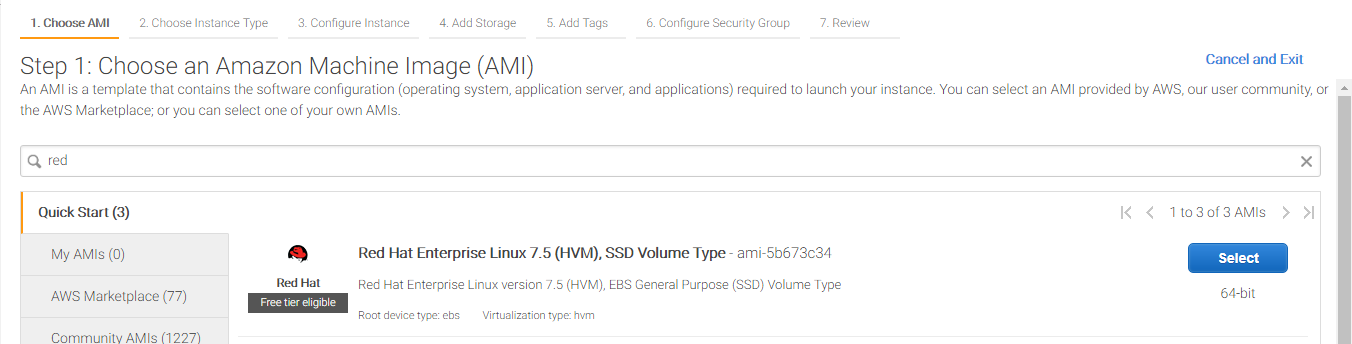
1. Three Linux virtual servers to host the code are to be created , and apache web servers to be installed on each of them to route the static code.

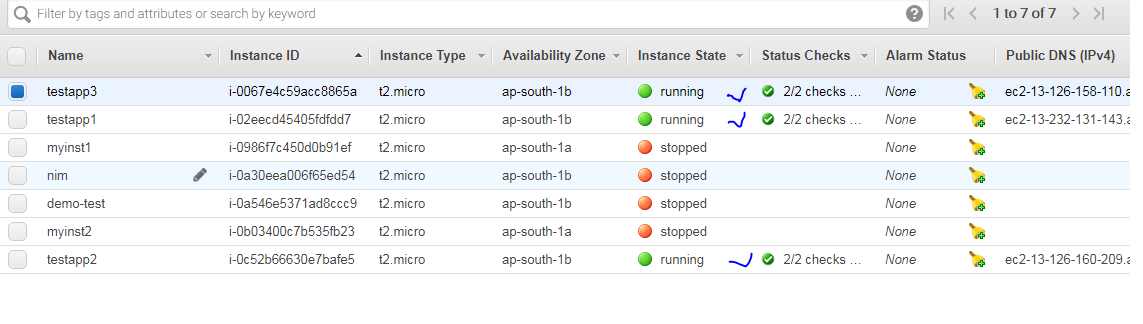
Procedure of code deployment 1:

1. Whenever the code gets checked in to version control , here GIT: an automated build gets created , so that updated code will get dropped in to the following stage server.
2. Once we are good with the code tests, code will get dropped in to the required app servers from this stage server with the help of this shell script.

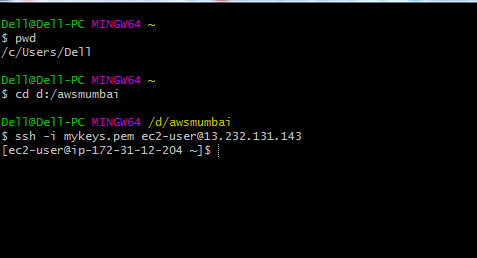
Screenshots for setting up the infrastructure as per solution 1:

1. Created the vitual servers ( RHEL AMI ) in AWS.

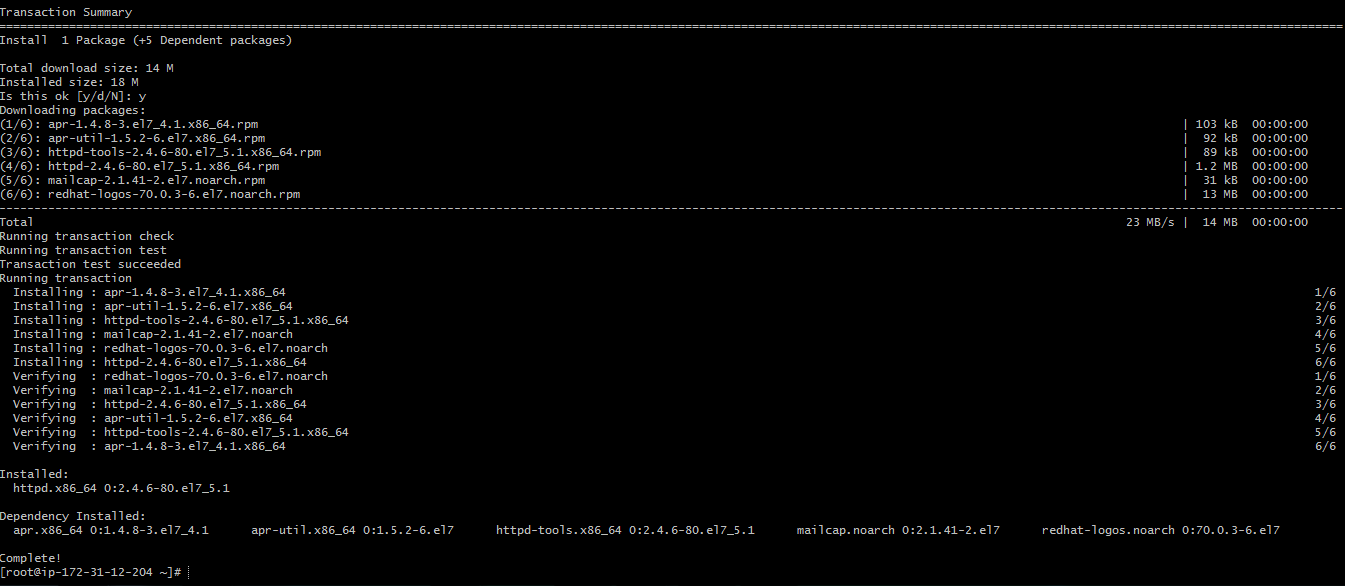




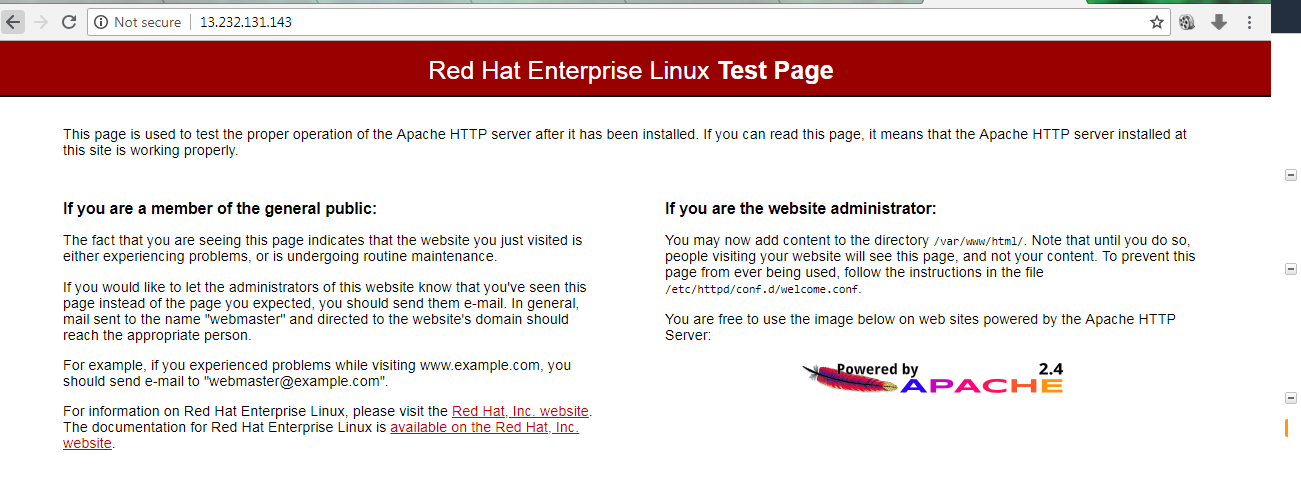
Logging in to testapp1 ( linux Box ) procedure would be same for the rest of the app servers .



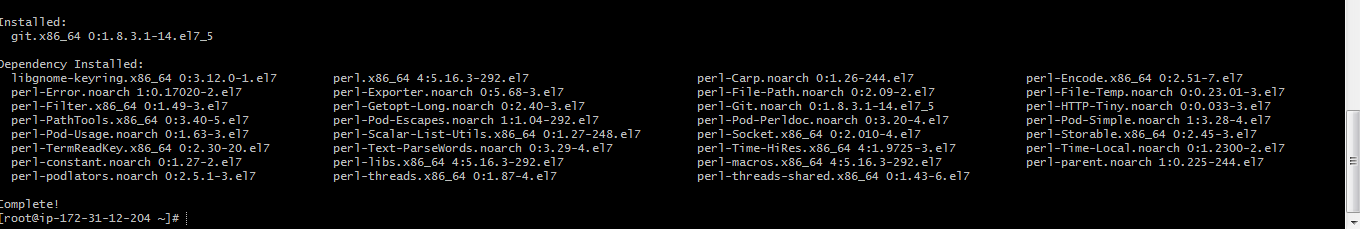
Connected to server, and installed apache webserver in the box.

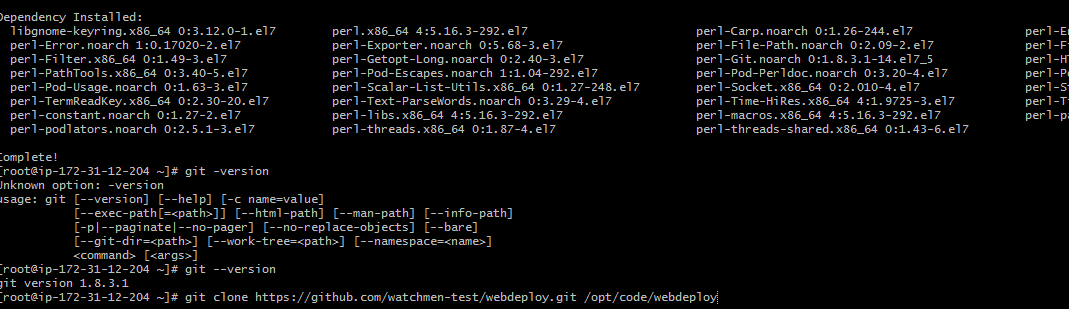


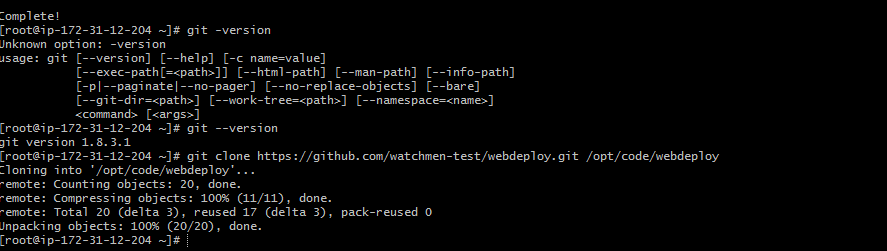
Tested the Apache webserver , and it is working fine

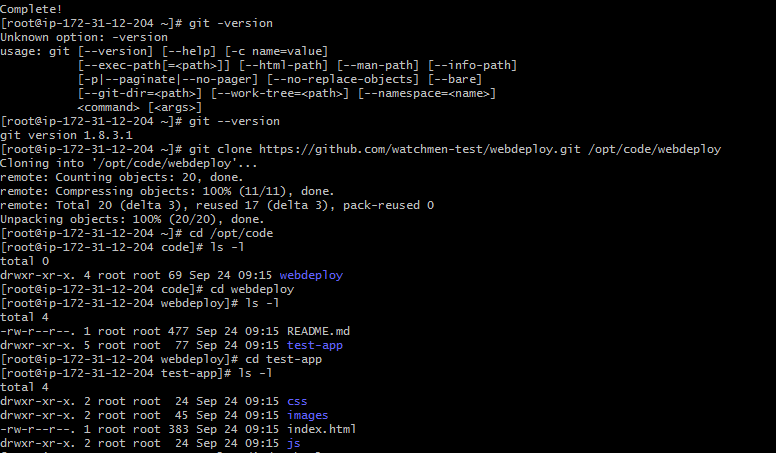


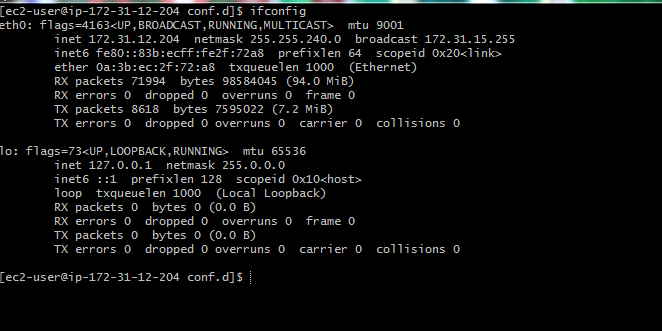
Installed Git in on the server for cloning the code in to the server.

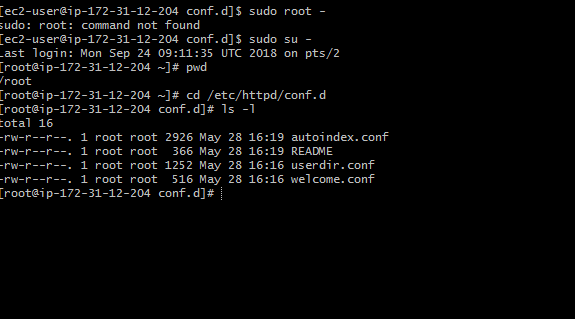


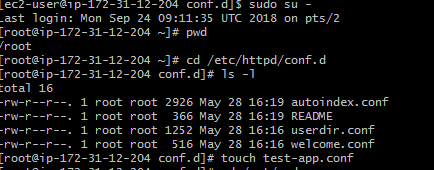


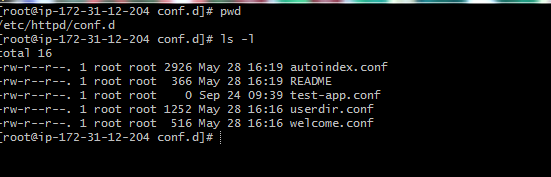


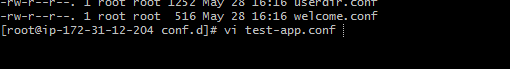


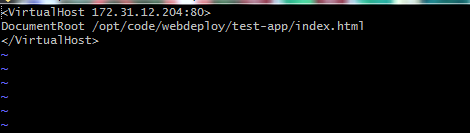


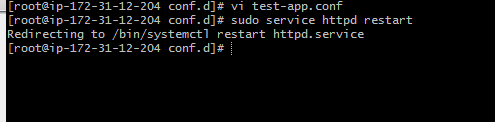


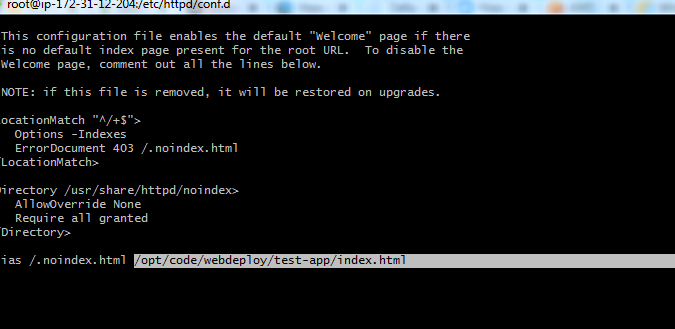








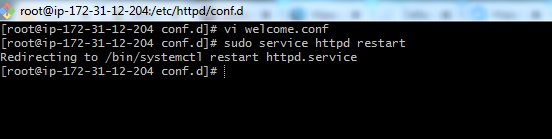




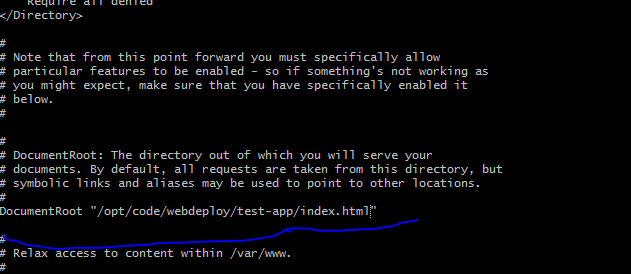
Worked on the configuration’s in the apache httpd.conf file and welcome.conf

Created one empty file test-app.conf file under /etc/httpd/conf.d as well to host the virtual host address of virtual server’s private ip , so that the request from this config file will be routed in to server and index.html will be called , whenever the request comes from the user.

Restarted the httpd service ,







If everything goes fine, whenever i get in to the browser with the public/private ip, i should have see this image like this :





Solution -2 :

1. I will implement Continuous integration here , so Whenever the code gets checked in to version control , here GIT : an automated build gets created , so that updated code will get dropped in to the following stage server.
2. I will have ansible running on this server ( master )
3. When i find i am good with the code stability check, i will push the code in to these required app servers ( ansible hosts ) . from ansible master to ansible hosts .

Solution-3:

I will have Jenkins job set up for this entire process .

Jenkins job fetches the code from the git hub once the code is checked in , and drops the code in to the required application servers ( apache webservers or tomcat application servers ) .

Dropping of the code can be written in linux scripts or app servers can be integrated with the Jenkins as containers.

Solution -4:

I will have vsts set up for the build pipeline ( to get the code from the version control ) and continuos integration is in place .

I will have release pipeline setup , which will fetch this code from the above build pipeline and dropping of this code in to required app servers.

Build pipeline : version control as git and continuos integration is in place and test is also in place.

Release pipeline : drops this build artfacts from the github ( here ) or from the Jenkins jobs in to the required app servers ( IIS, or apache) which can be installed in windows, or linux boxe’s .

We can even have powershell call in place here release pipeline if they are window’s servers.

Solution-5: ( through u deploy , IBM urban code deploy ):

Jenkins jobs can be created to create the application components .

Code can be fetched from the version control tools ( vsts, tfs, github etc ) and this code will be created in to application components from the component process

These components are pieces of the application to make it running and deploying the code in to app servers.

Finally application process can be run in different environments as per the application process designed, i.e deploy, uninstall artifacts, restart/stopping of app services etc.