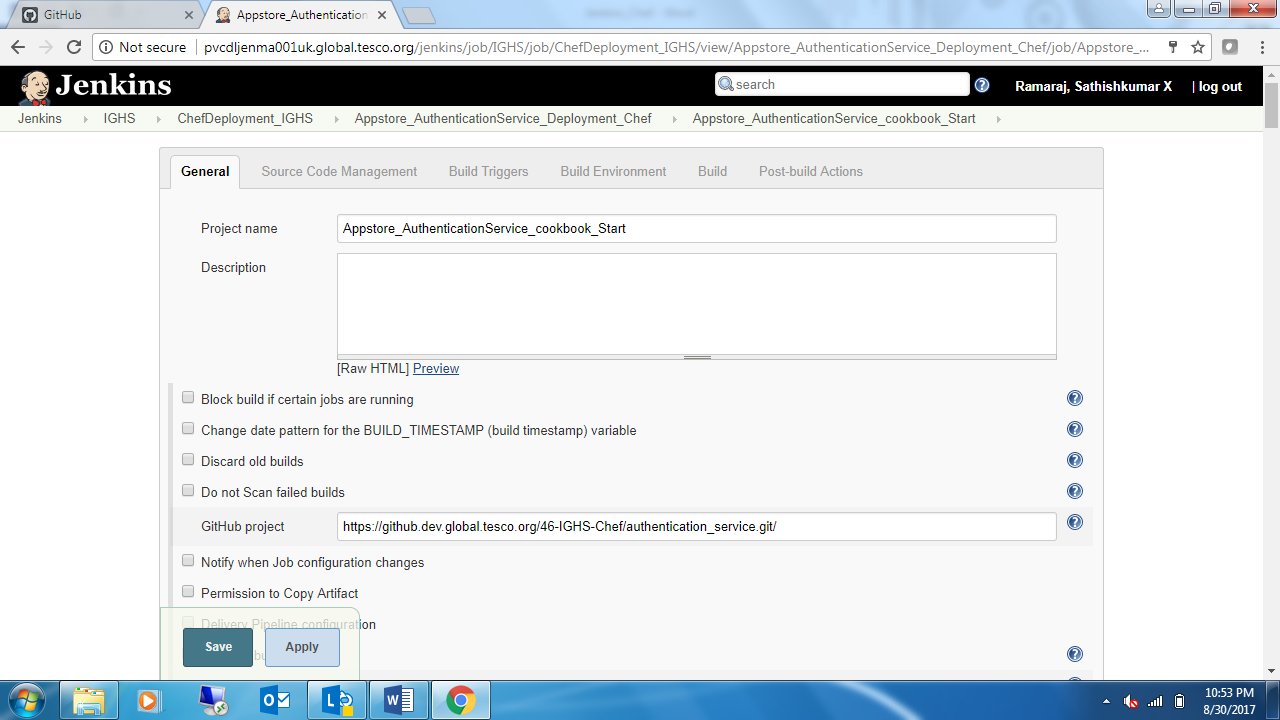
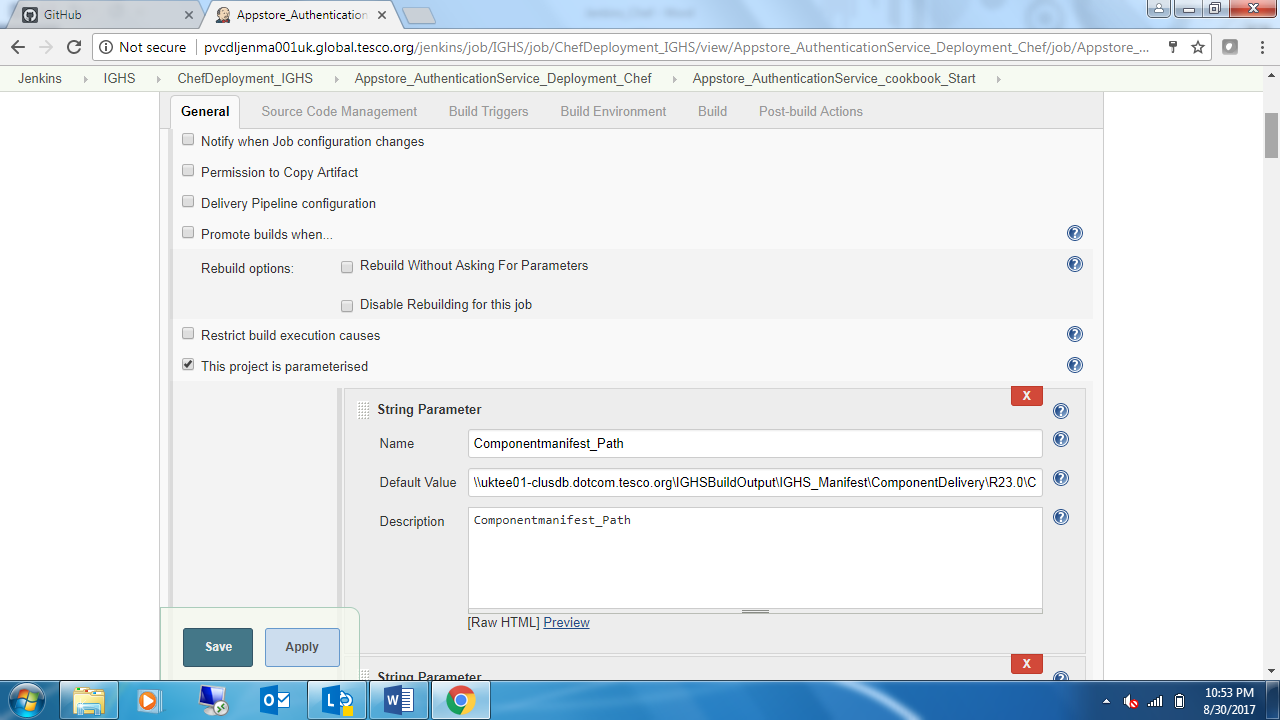
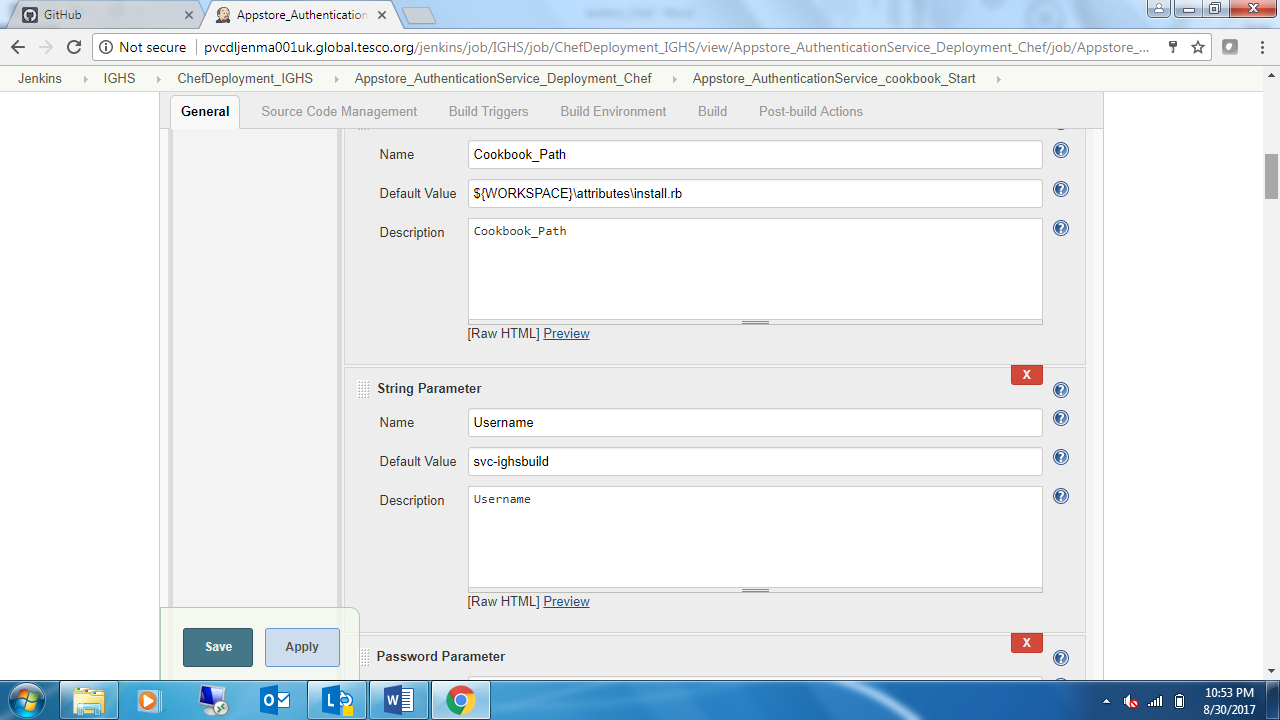
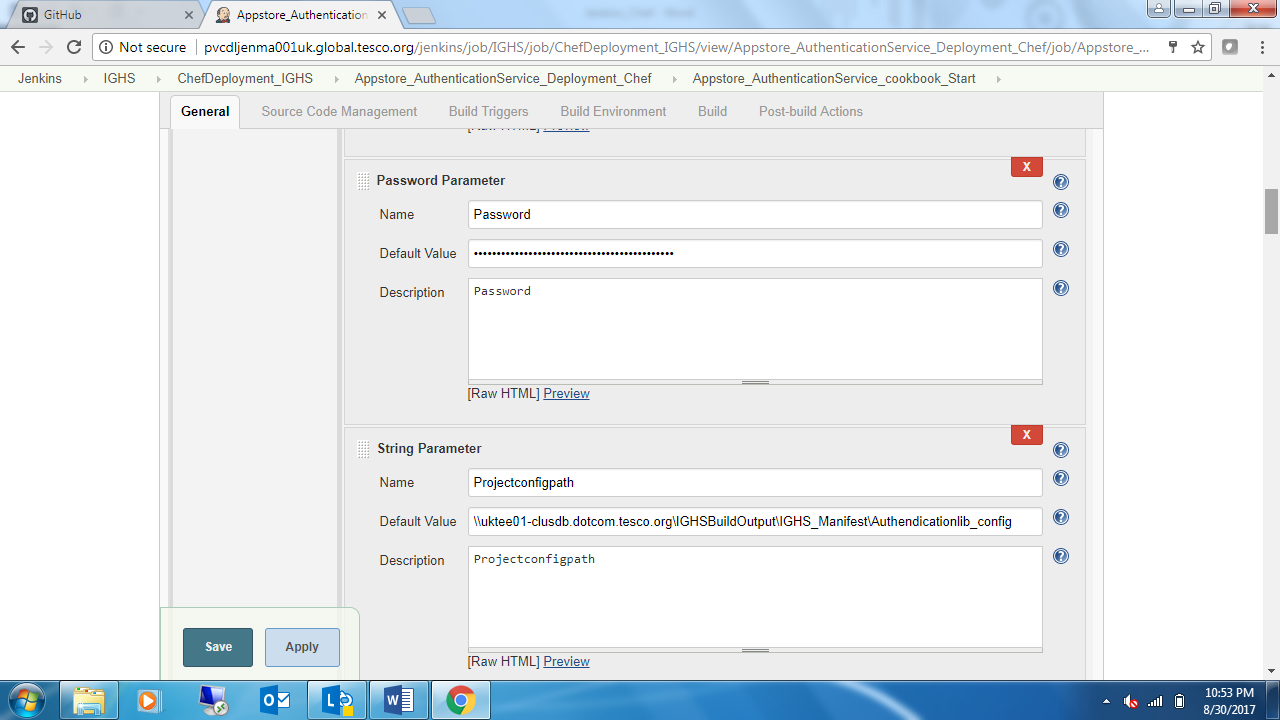


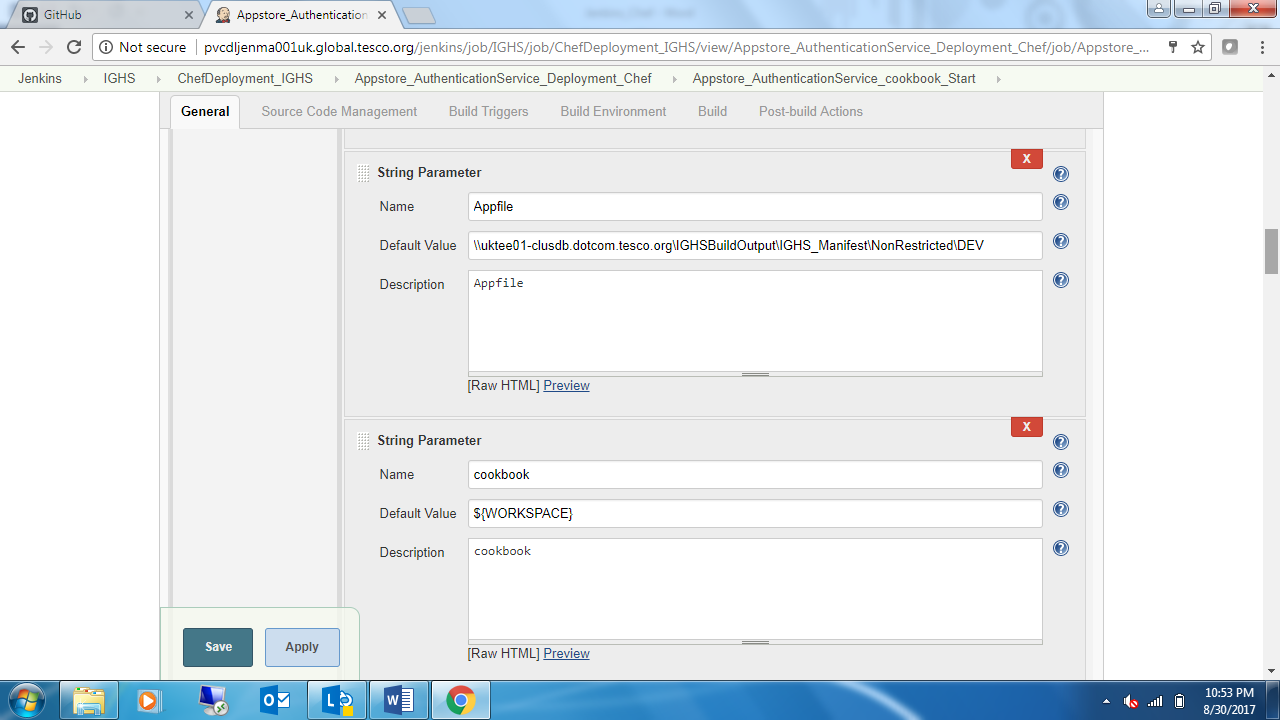
**Appstore\_AuthenticationService\_cookbook\_Start**

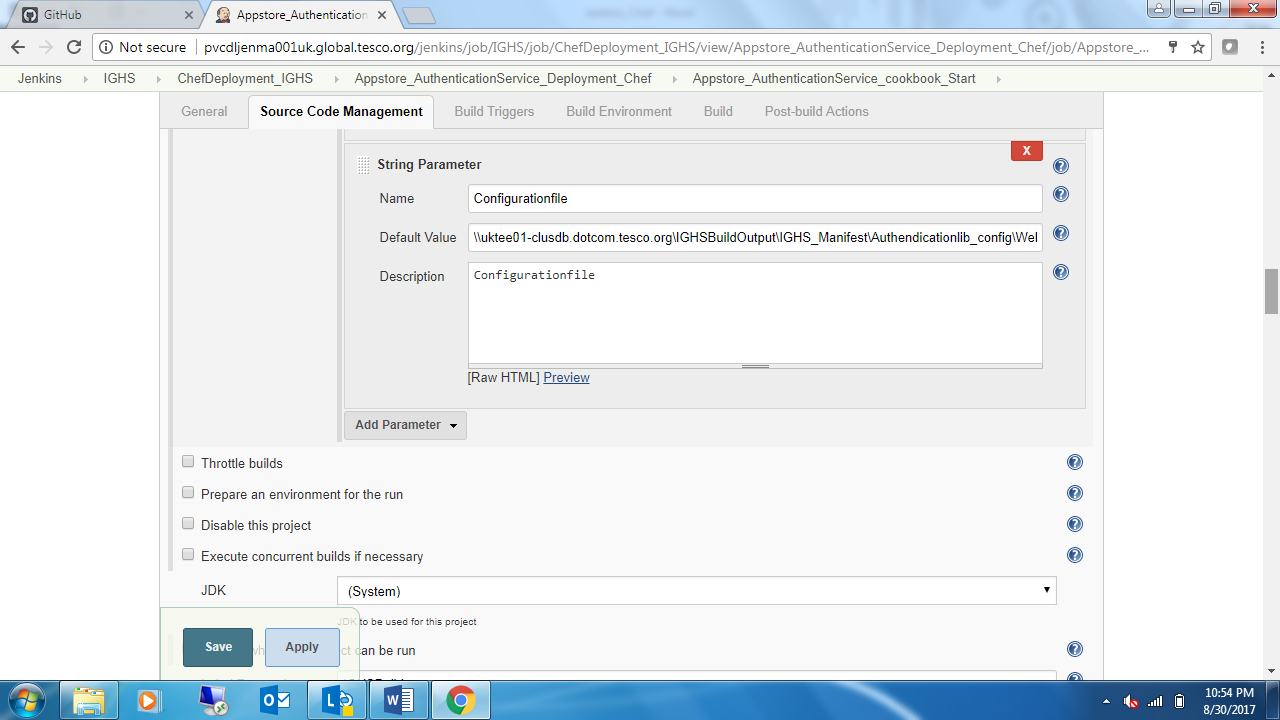


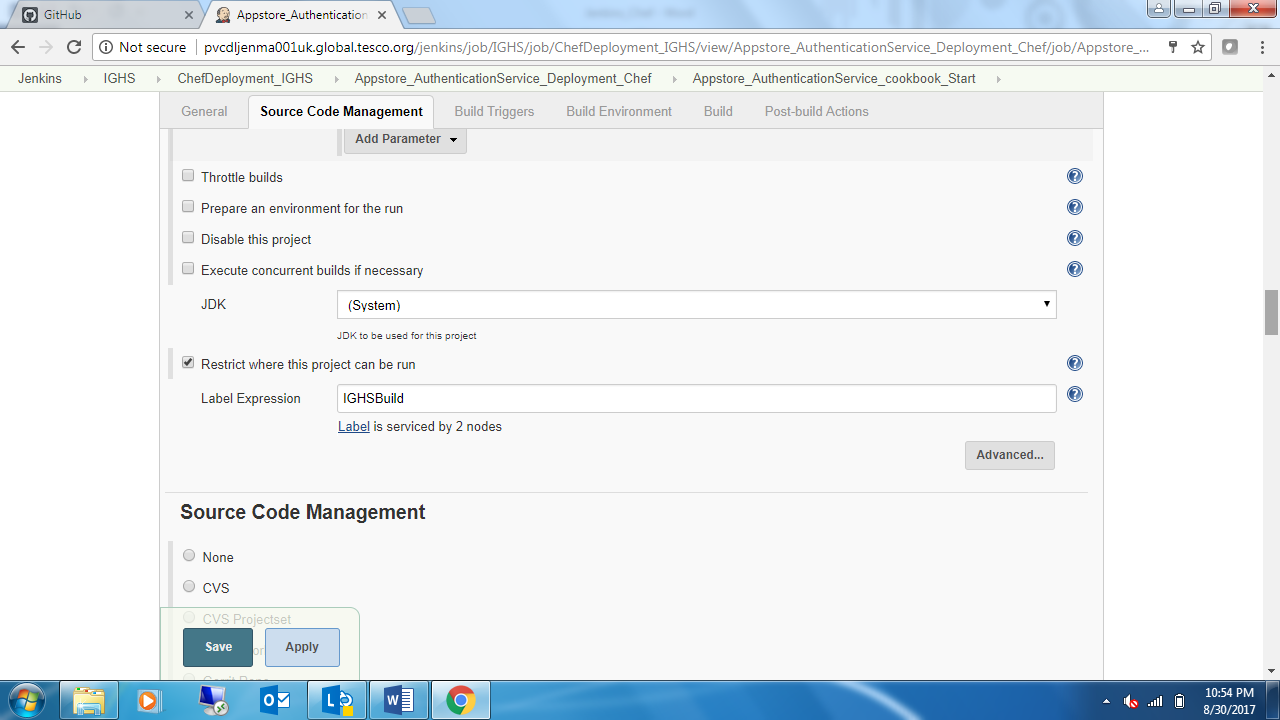


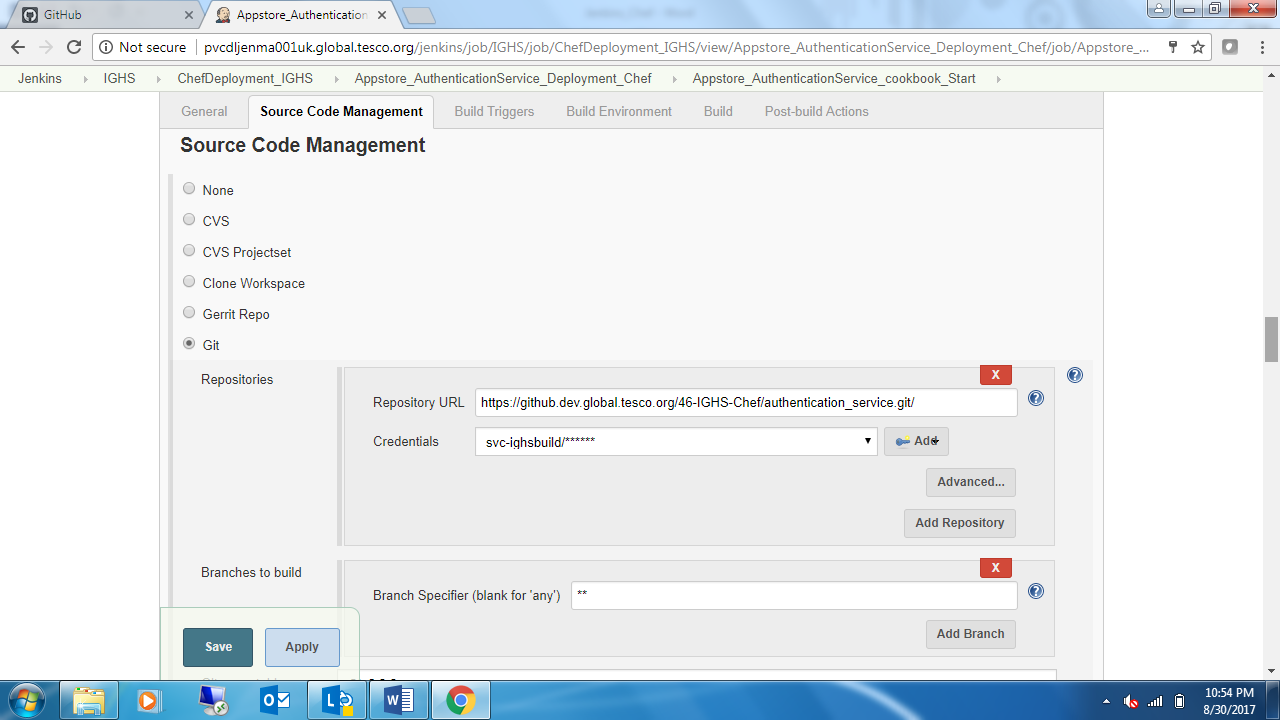


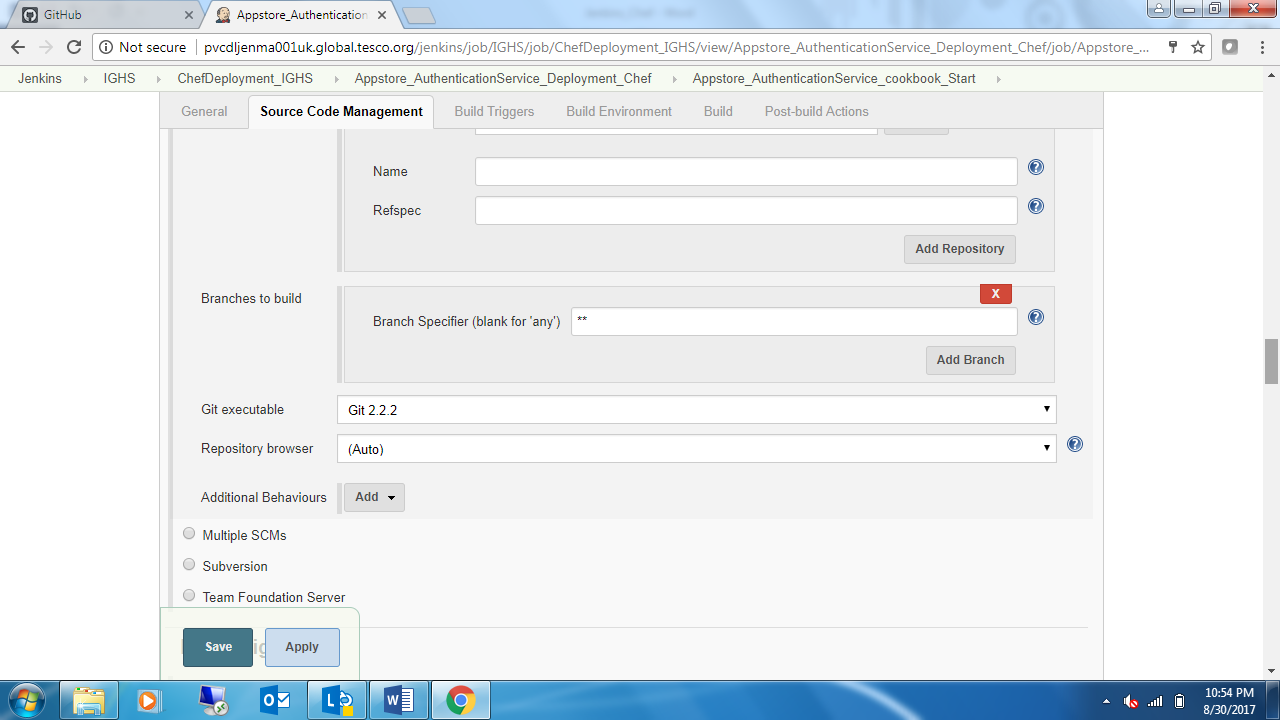


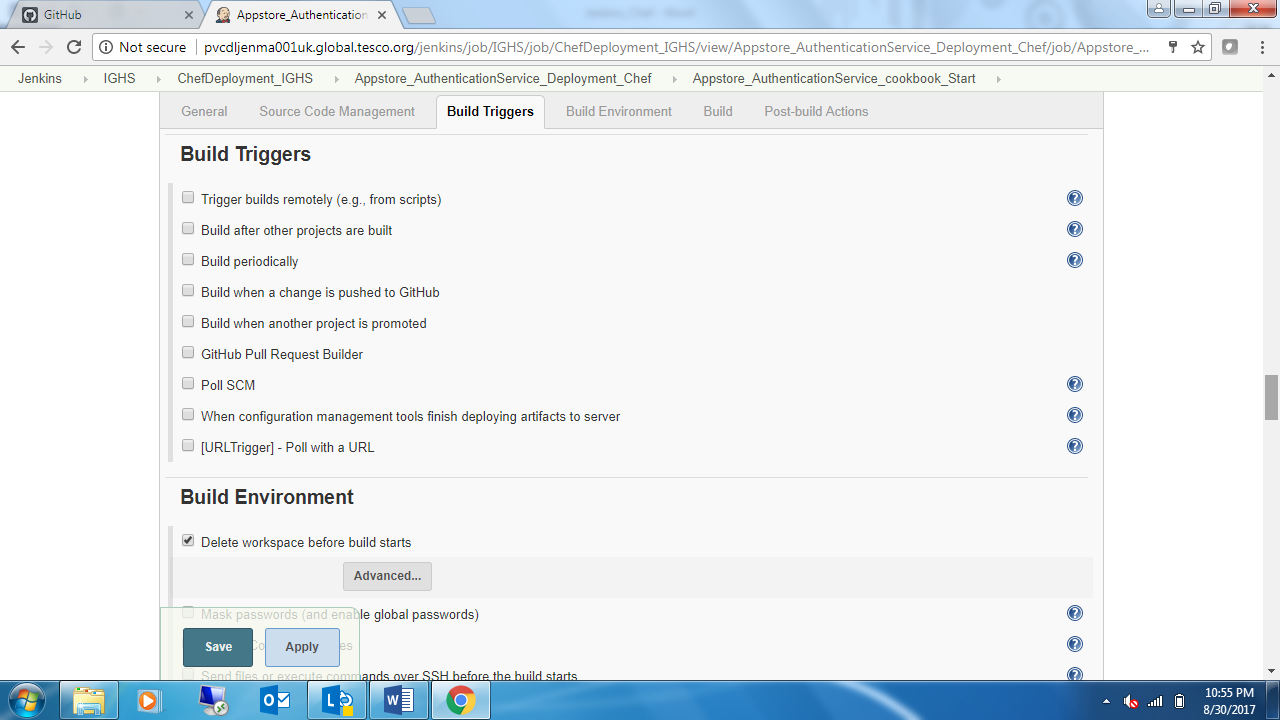


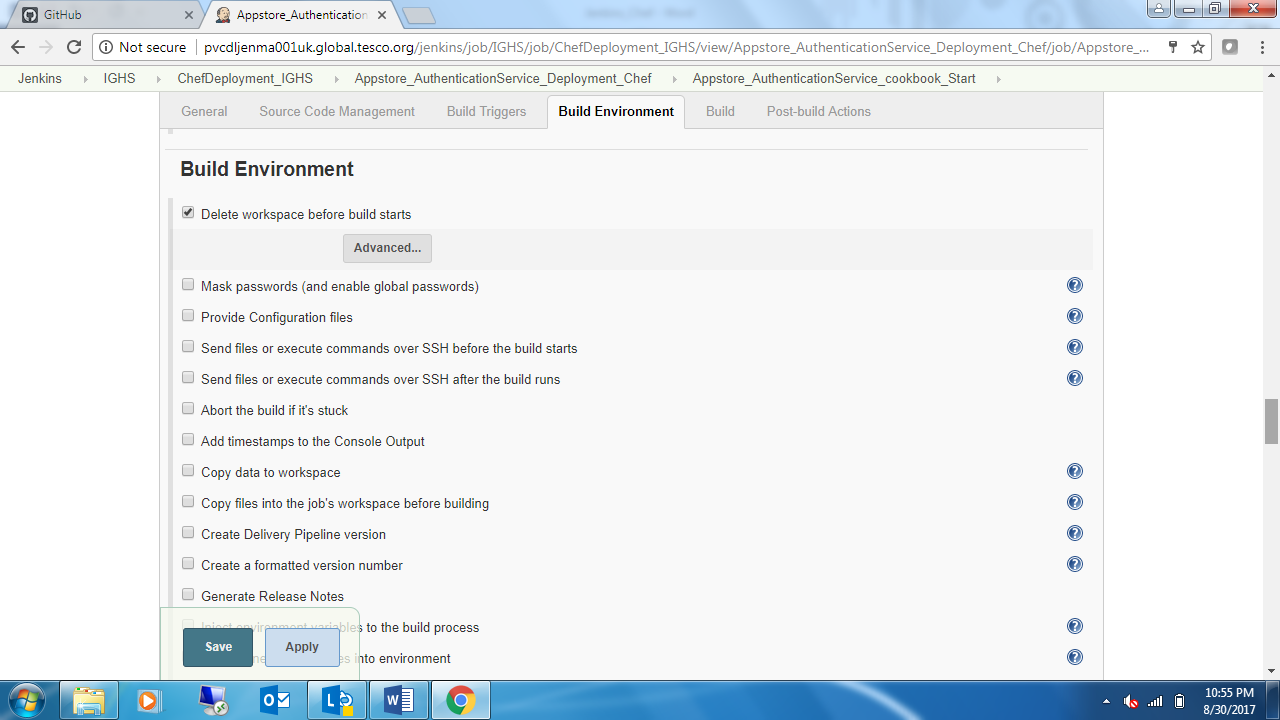


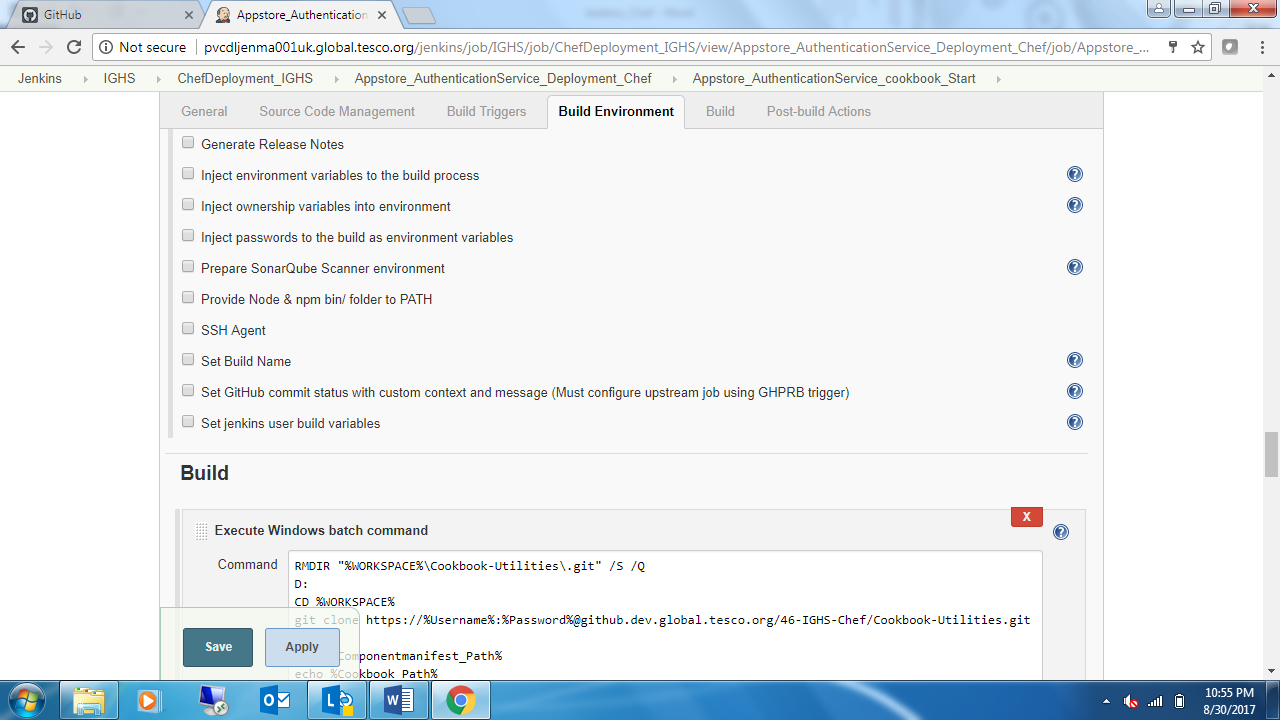


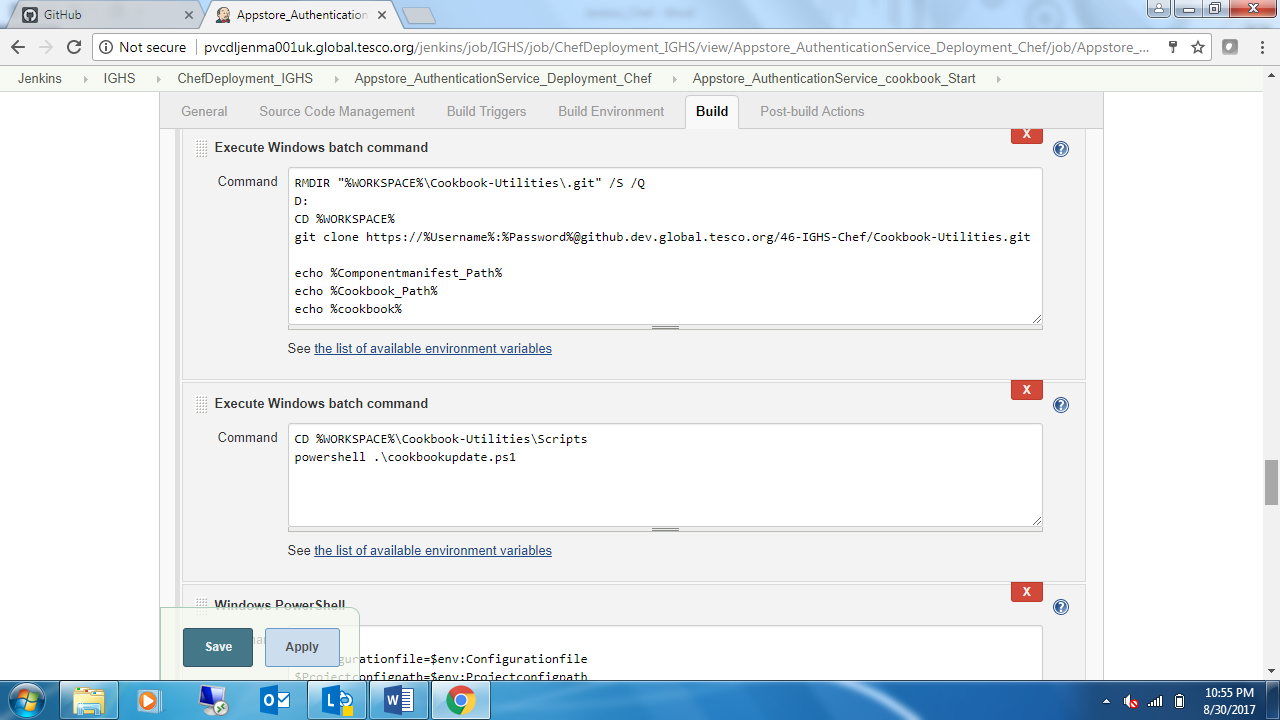












RMDIR "%WORKSPACE%\Cookbook-Utilities\.git" /S /Q

D:

CD %WORKSPACE%

git clone https://%Username%:%Password%@github.dev.global.tesco.org/46-IGHS-Chef/Cookbook-Utilities.git

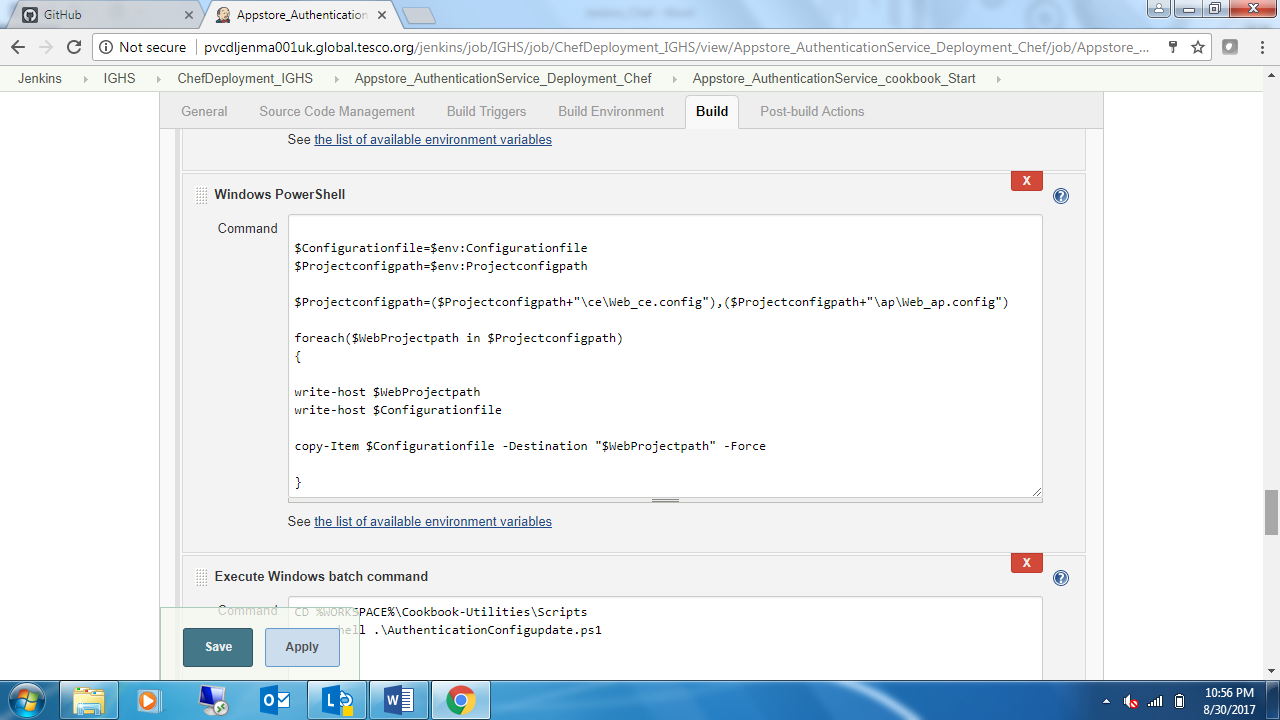
echo %Componentmanifest\_Path%

echo %Cookbook\_Path%

echo %cookbook%

CD %WORKSPACE%\Cookbook-Utilities\Scripts

powershell .\cookbookupdate.ps1



$Configurationfile=$env:Configurationfile

$Projectconfigpath=$env:Projectconfigpath

$Projectconfigpath=($Projectconfigpath+"\ce\Web\_ce.config"),($Projectconfigpath+"\ap\Web\_ap.config")

foreach($WebProjectpath in $Projectconfigpath)

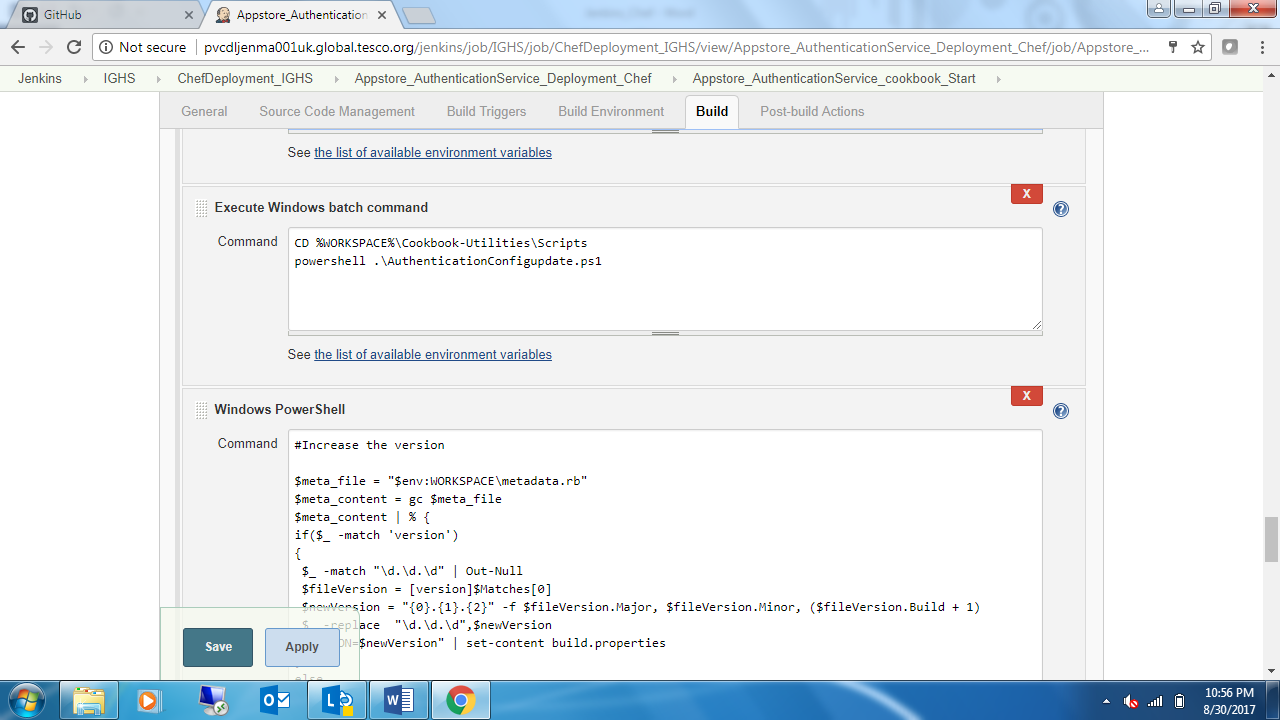
{

write-host $WebProjectpath

write-host $Configurationfile

copy-Item $Configurationfile -Destination "$WebProjectpath" -Force

}



CD %WORKSPACE%\Cookbook-Utilities\Scripts

powershell .\AuthenticationConfigupdate.ps1

#Increase the version

$meta\_file = "$env:WORKSPACE\metadata.rb"

$meta\_content = gc $meta\_file

$meta\_content | % {

if($\_ -match 'version')

{

$\_ -match "\d.\d.\d" | Out-Null

$fileVersion = [version]$Matches[0]

$newVersion = "{0}.{1}.{2}" -f $fileVersion.Major, $fileVersion.Minor, ($fileVersion.Build + 1)

$\_ -replace "\d.\d.\d",$newVersion

"VERSION=$newVersion" | set-content build.properties

}

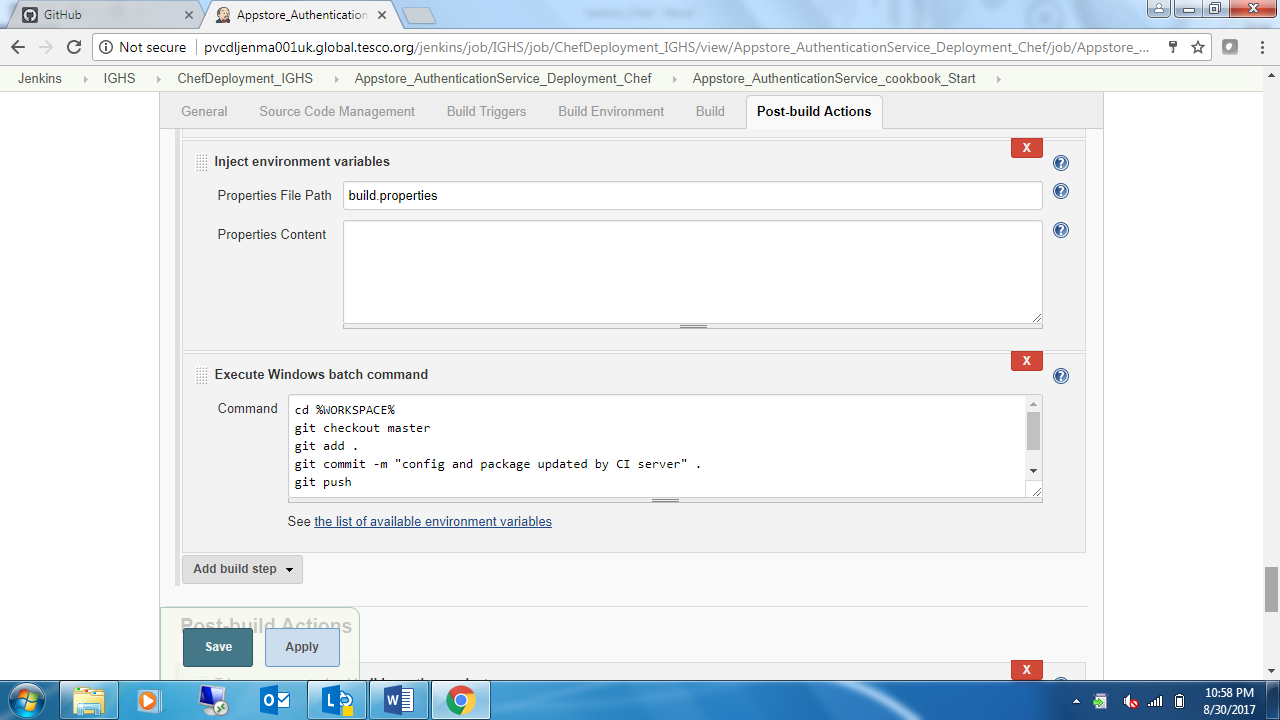
else

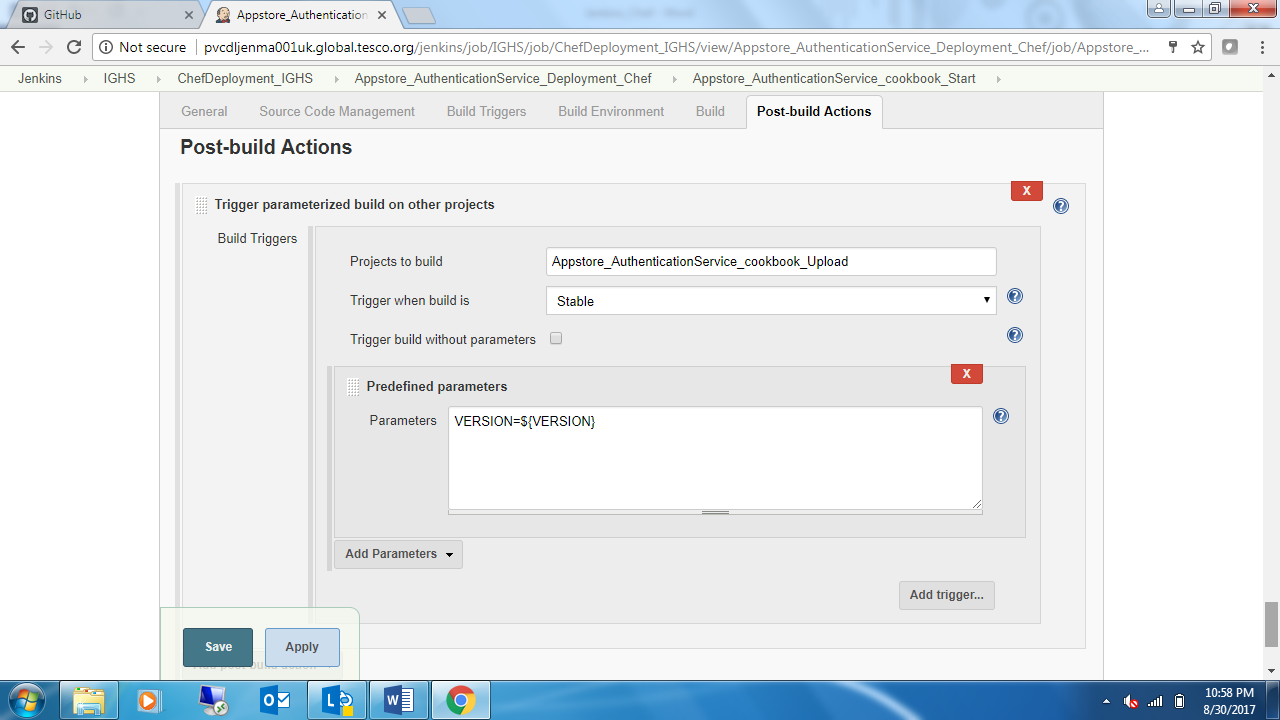
{

$\_

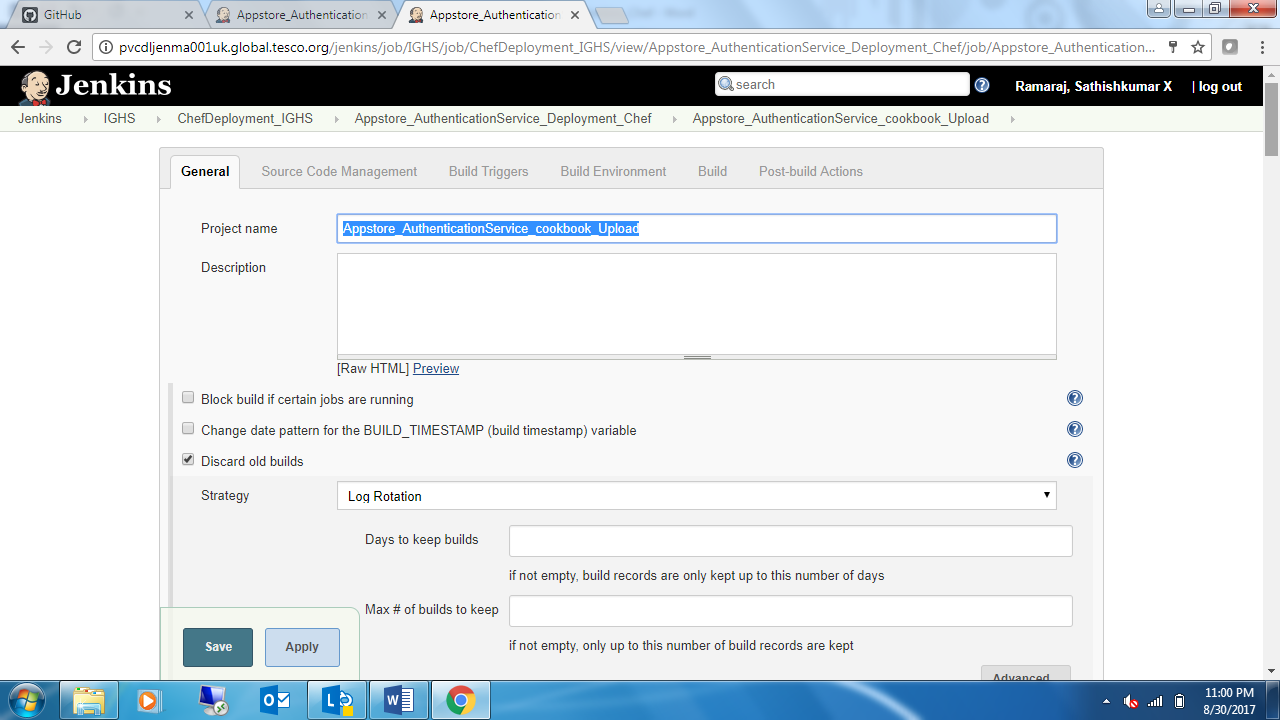
}

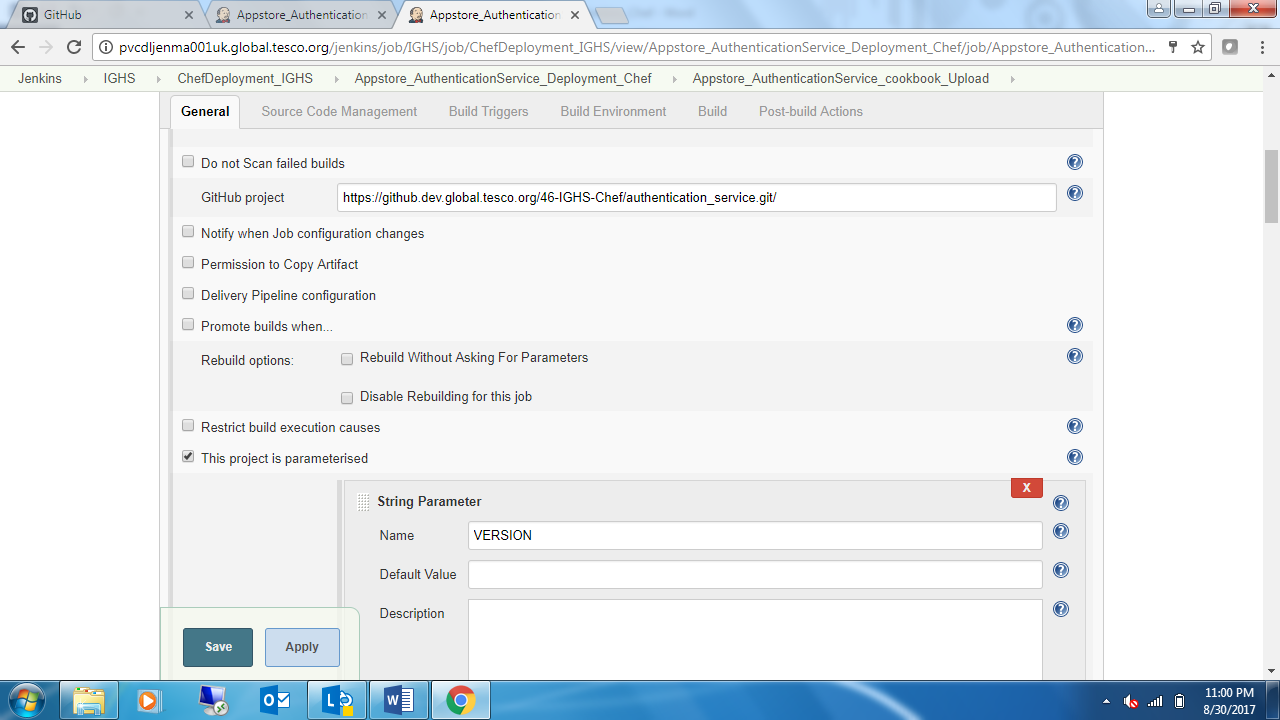
} | Set-Content $meta\_file

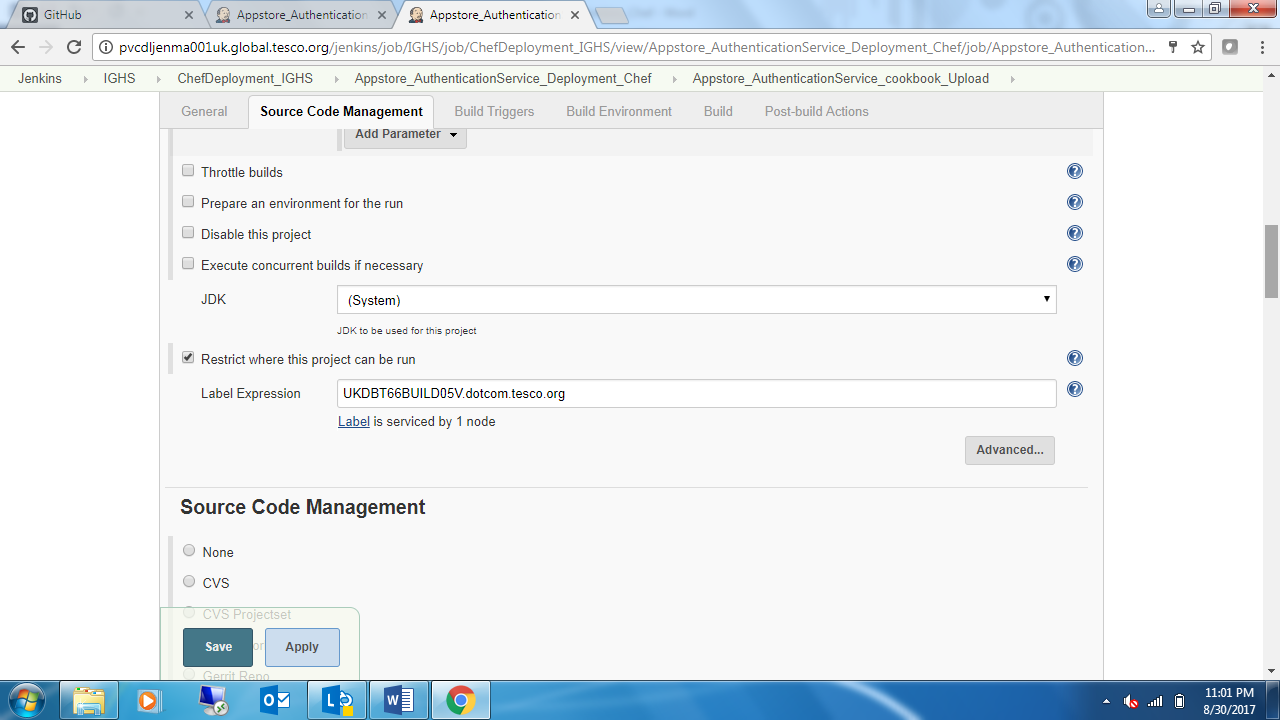


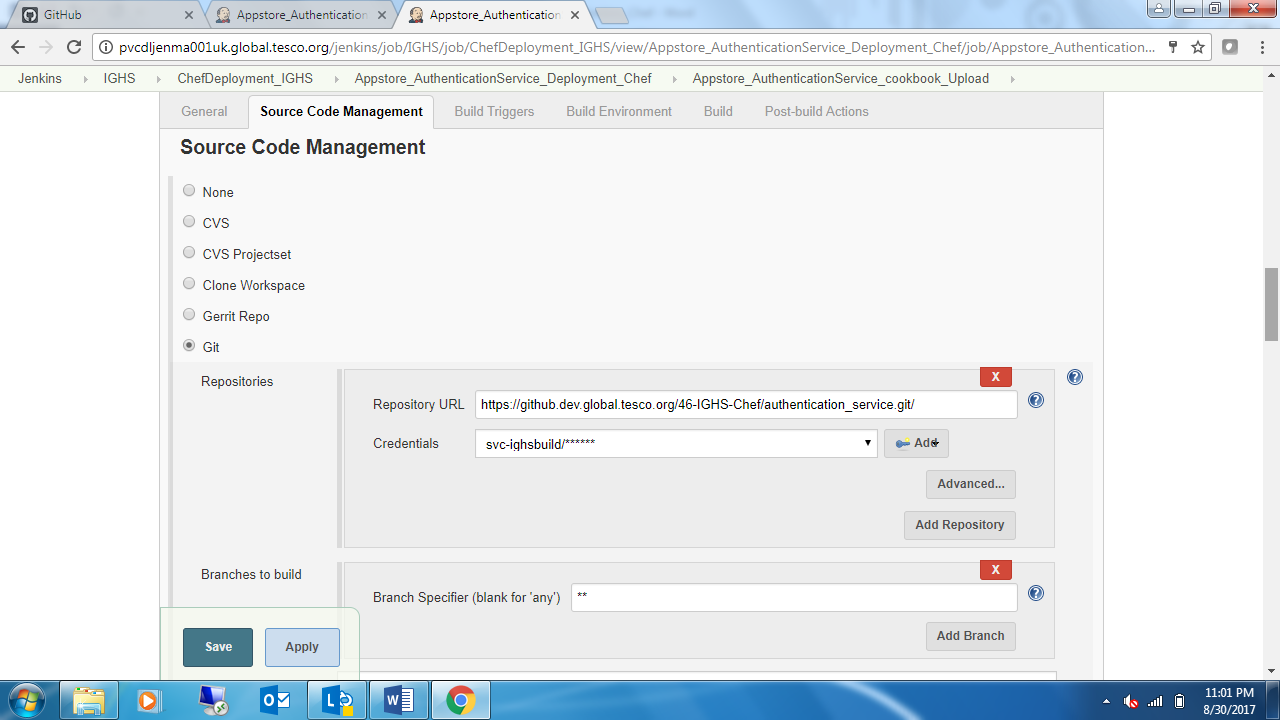


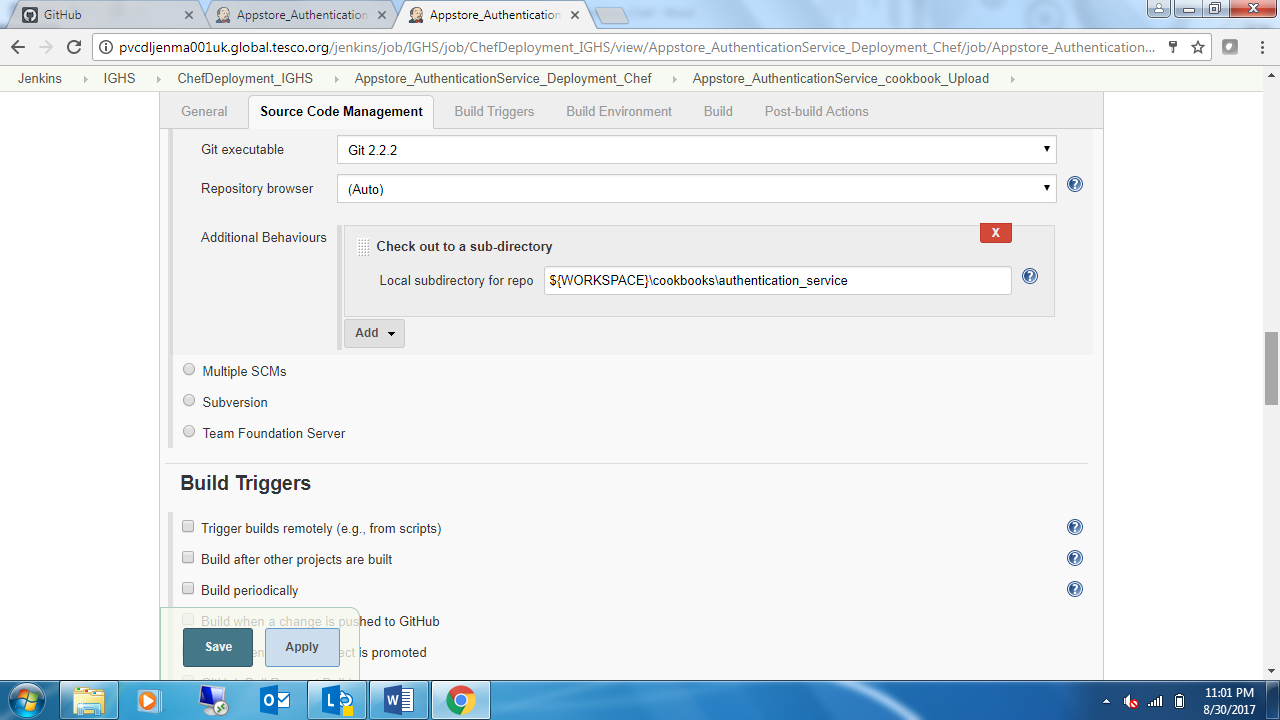
**Appstore\_AuthenticationService\_cookbook\_Upload**

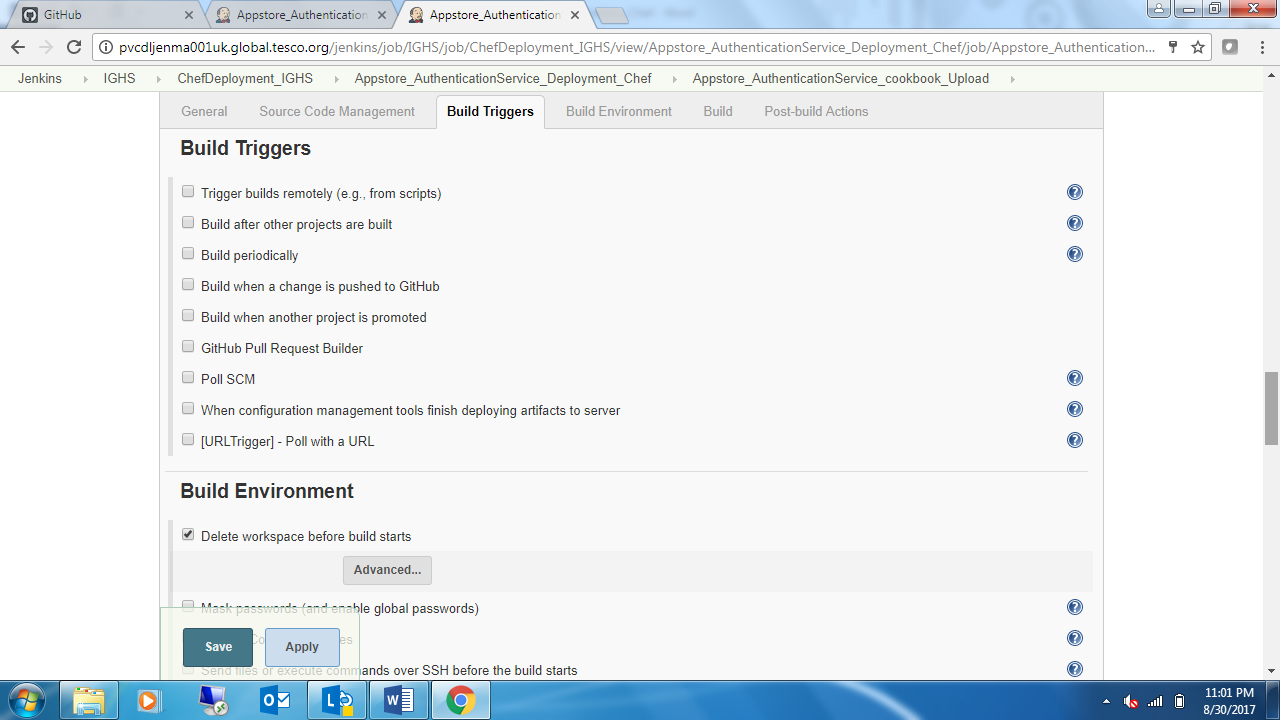


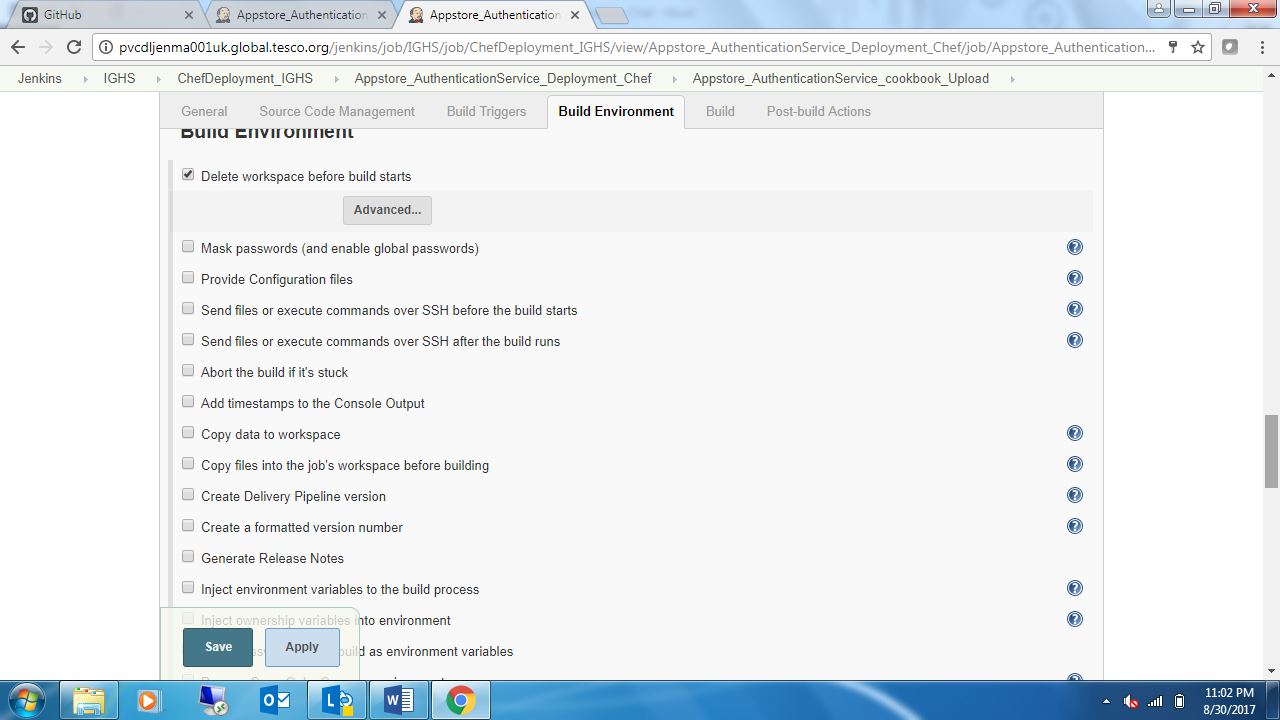


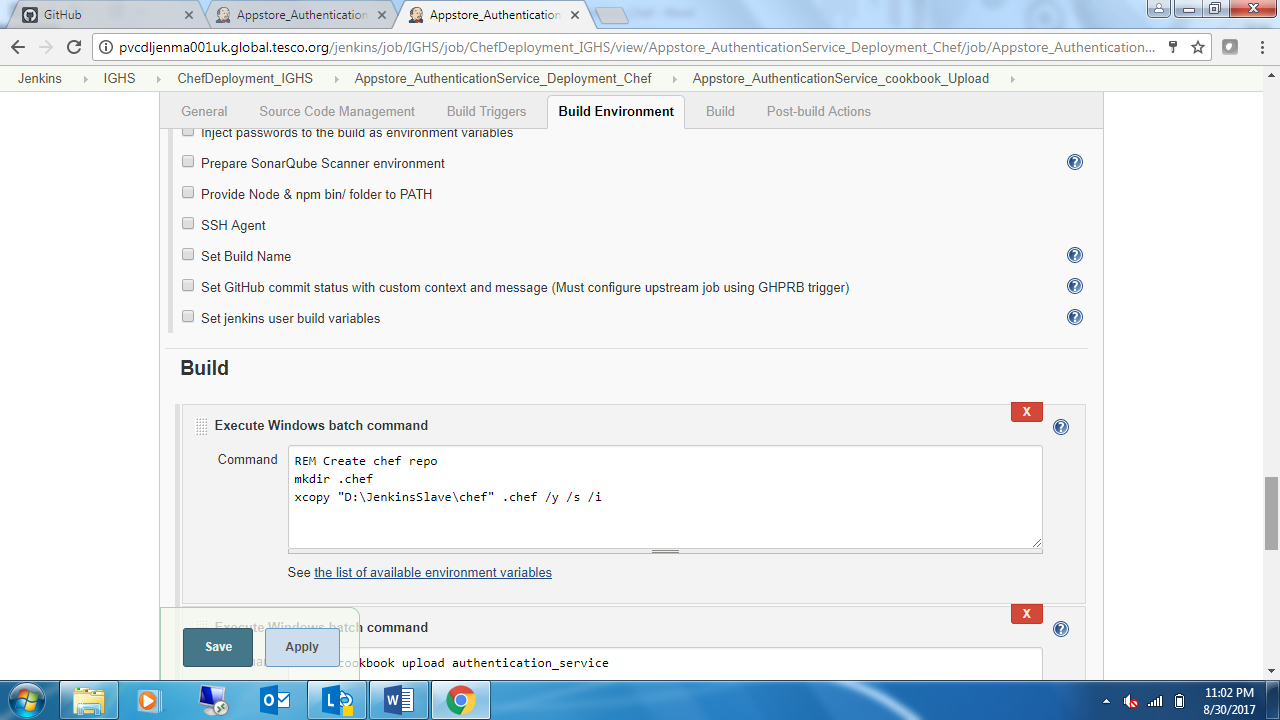








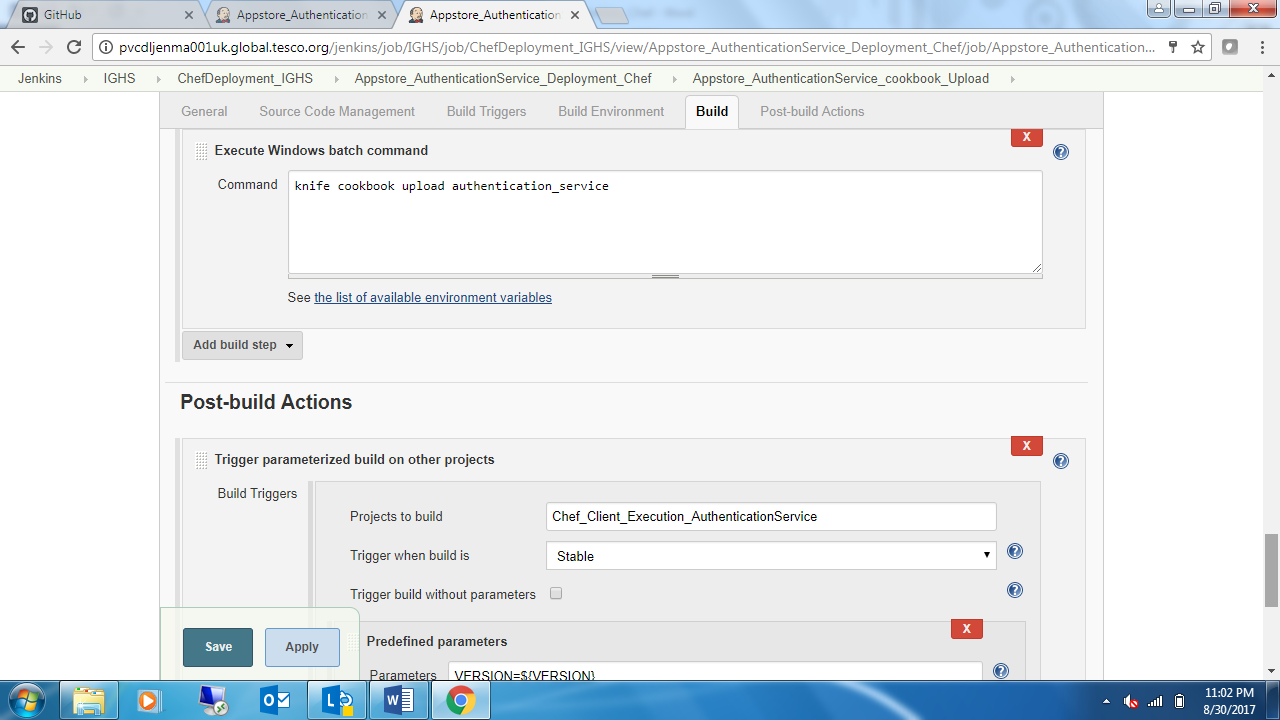


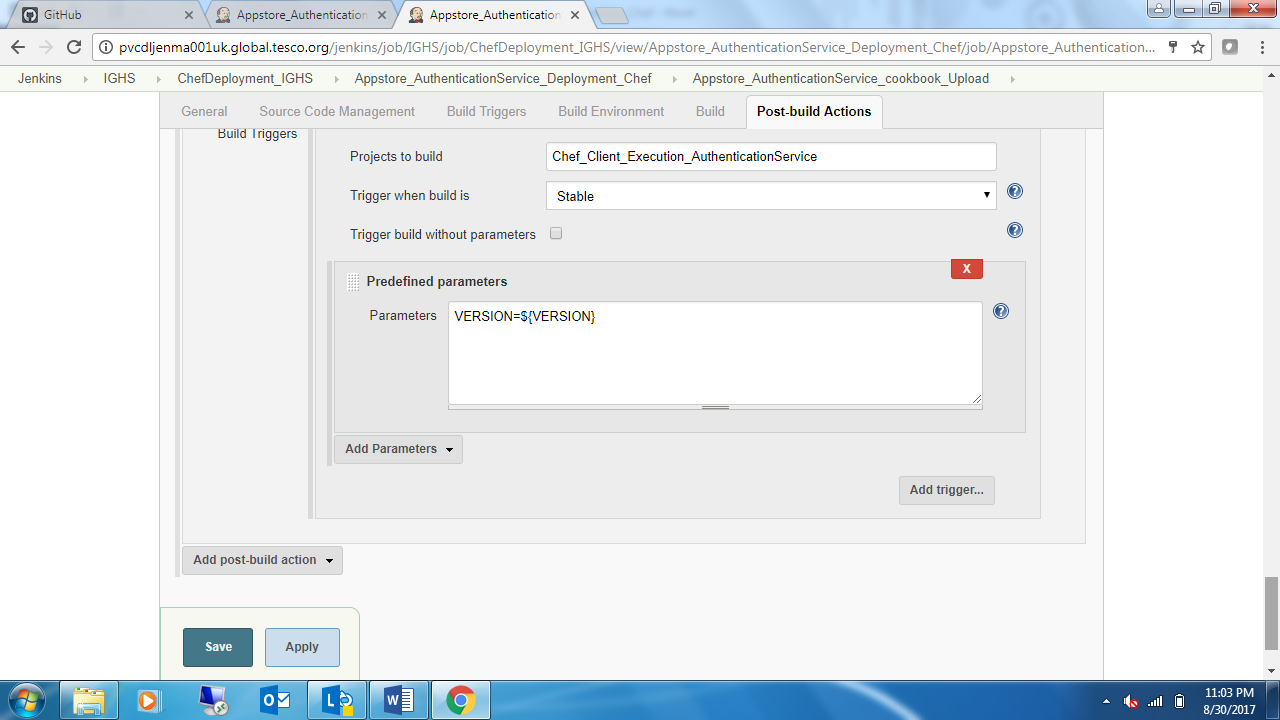


REM Create chef repo

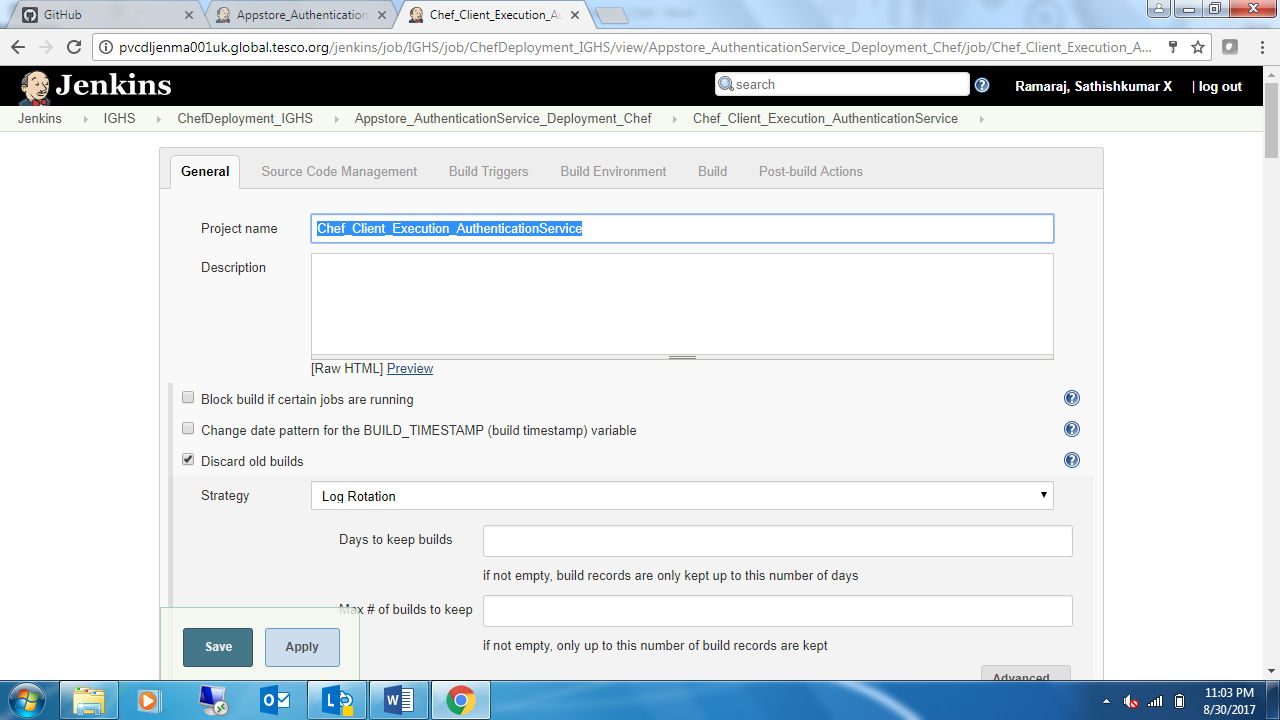
mkdir .chef

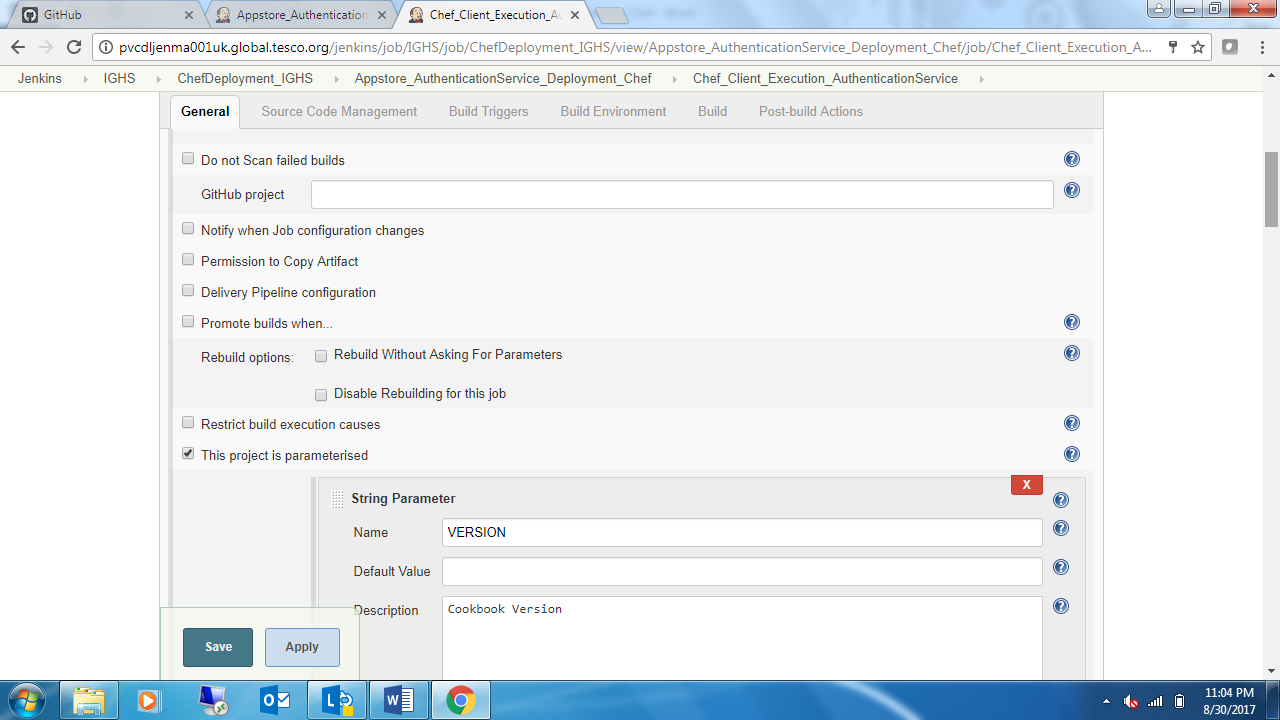
xcopy "D:\JenkinsSlave\chef" .chef /y /s /i

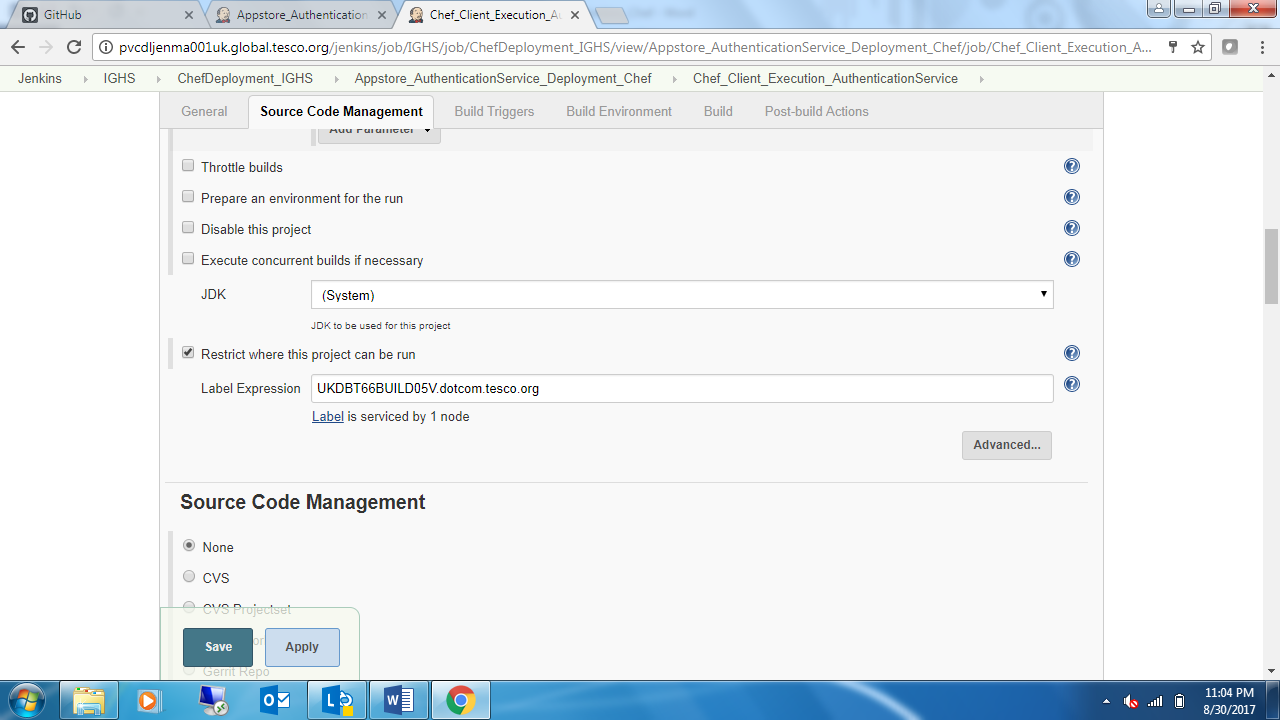


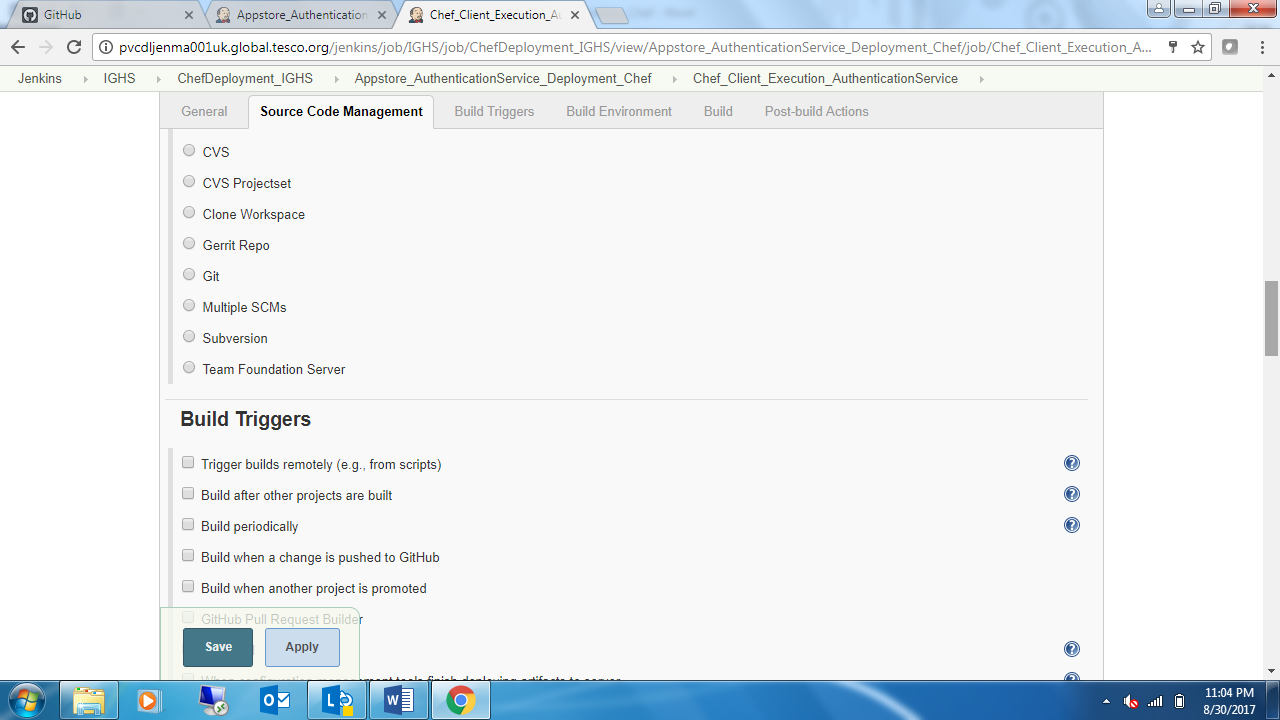


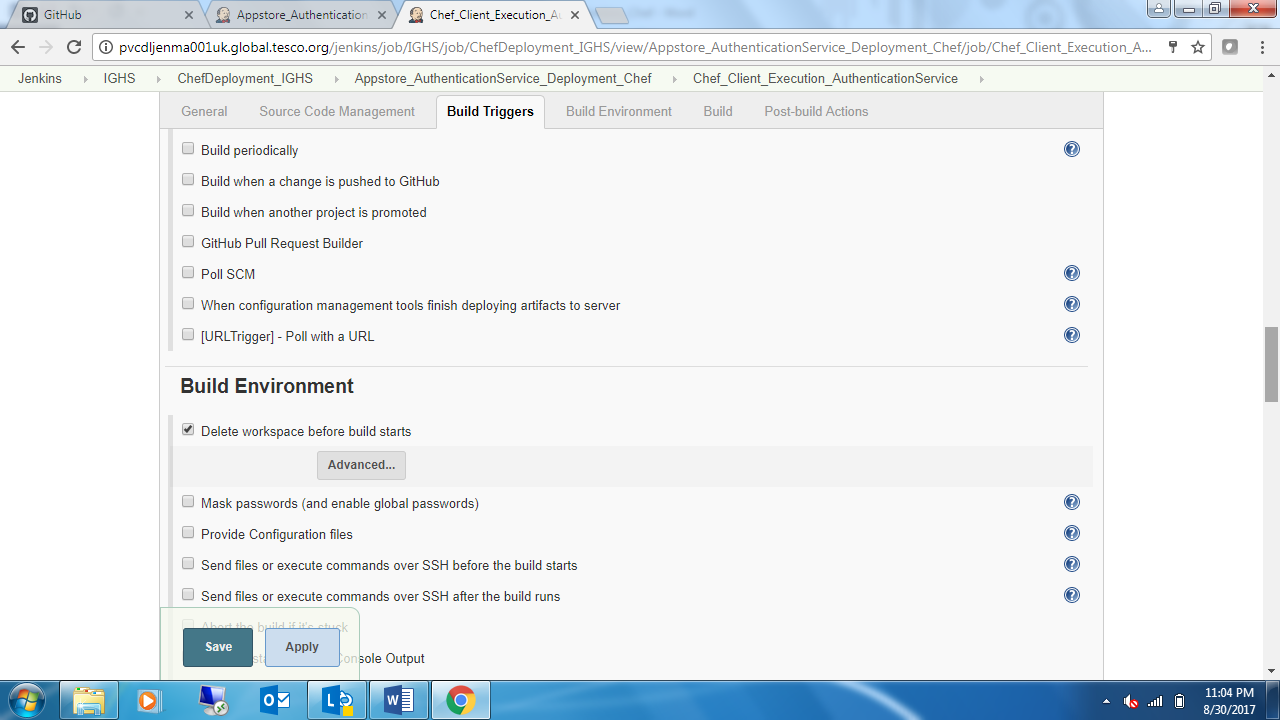
**Chef\_Client\_Execution\_AuthenticationService**

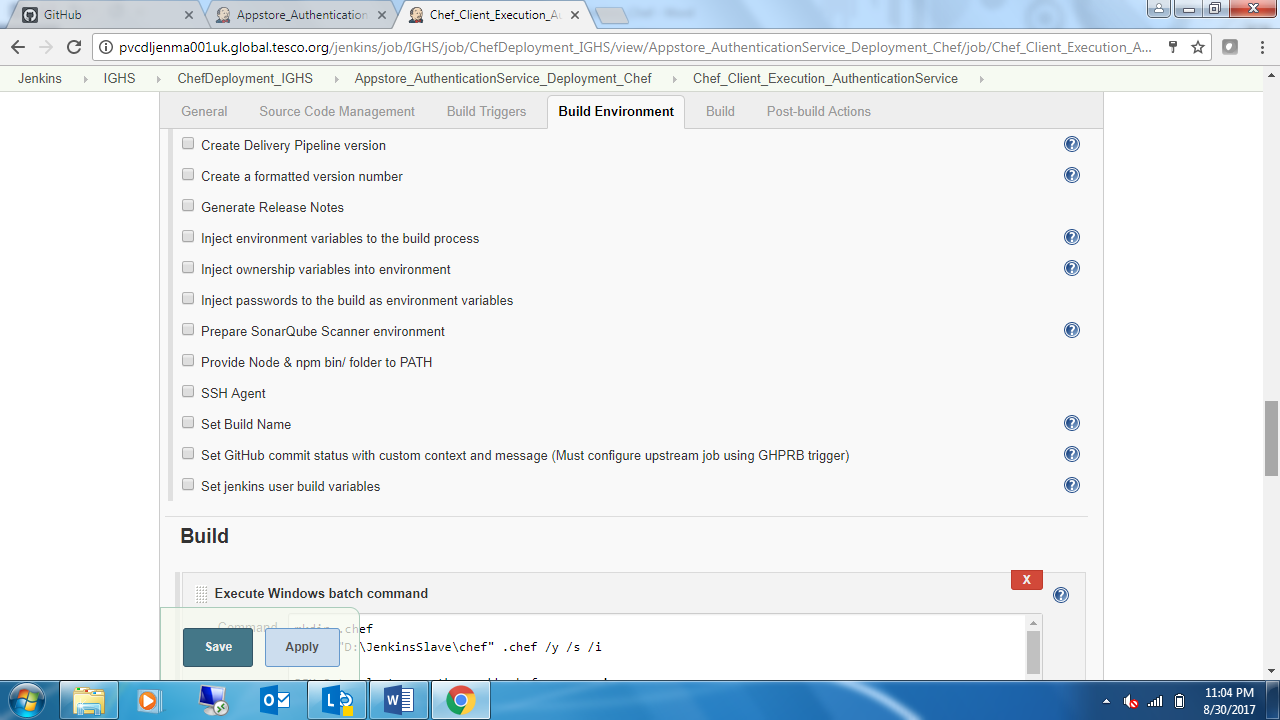


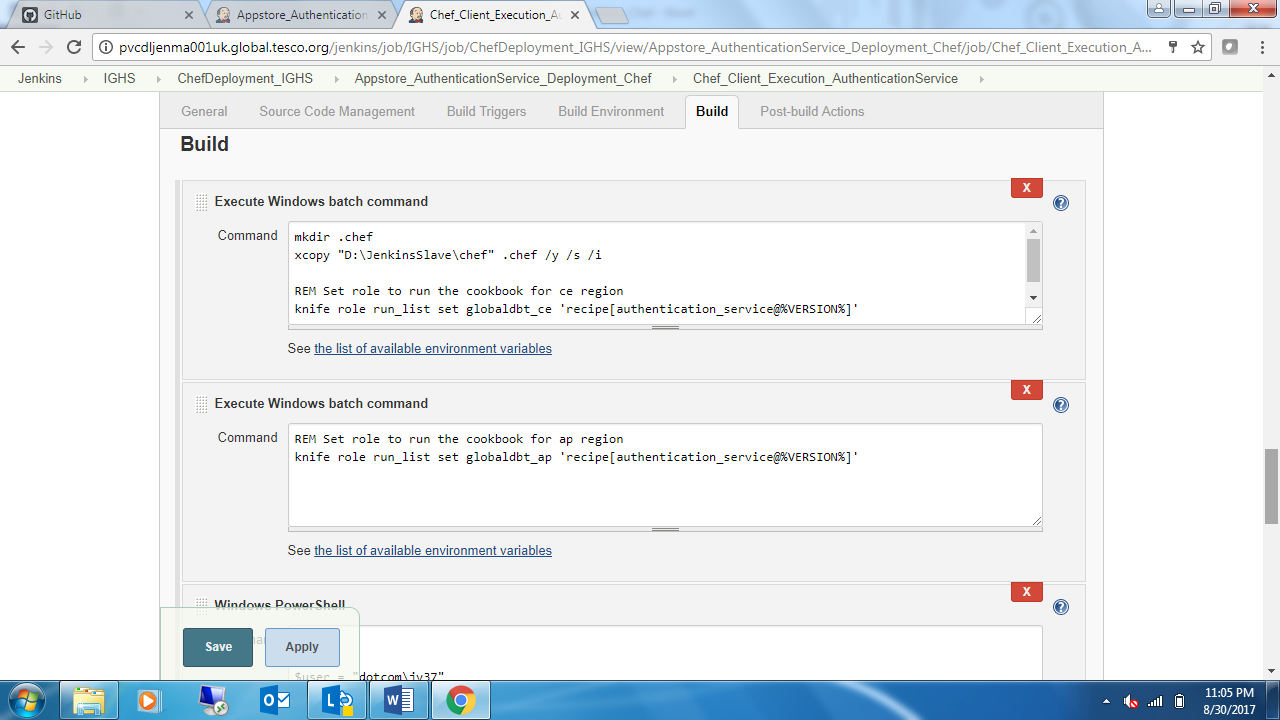












**mkdir .chef**

**xcopy "D:\JenkinsSlave\chef" .chef /y /s /i**

**REM Set role to run the cookbook for ce region**

**knife role run\_list set globaldbt\_ce 'recipe[authentication\_service@%VERSION%]'**

**REM Set role to run the cookbook for ap region**

**knife role run\_list set globaldbt\_ap 'recipe[authentication\_service@%VERSION%]'**



**try**

**{**

**$user = "dotcom\jv37"**

**#$password = $env:jv37**

**Write-Host $password**

**$password='Tesco##!!1234'**

**$Computers = @("UKDEV66MIGAP01V.dotcom.tesco.org","UKDEV66APING01V.dotcom.tesco.org")**

**#create script block to run chef client**

**$run\_chef\_client = {**

**Param($ComputerName, $user, $password)**

**try**

**{**

**cd $env:WORKSPACE**

**chef shell-init powershell | Invoke-Expression**

**knife winrm $ComputerName 'chef-client -c c:/chef/client.rb' -m -x $user -P $password**

**if($lastexitcode -ne 0) { throw }**

**iisreset /start $ComputerName**

**}**

**catch**

**{**

**$Errormessage=$\_.Exception.Message**

**throw $Errormessage**

**}**

**}**

**#Start all client execution**

**ForEach($Computer in $Computers)**

**{**

**Start-Job -scriptblock $run\_chef\_client -ArgumentList $Computer, $user, $password**

**}**

**#wait job until all clients execution completed**

**Get-Job | Wait-Job**

**Get-Job | Receive-Job**

**Get-Job | % {if($\_.state -eq 'Failed'){throw}}**

**}**

**catch**

**{**

**$Errormessage=$\_.Exception.Message**

**throw $Errormessage**

**}**