

Jira-Rundeck scenarios

**www.zippyops.com**

172-172, 5th floor Old Mahabalipuram Road

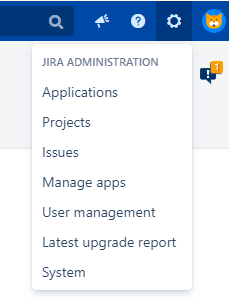
(Above Axis Bank-PTC Bus Stop)  
Thuraipakkam  
Chennai 600097

**🖂**[**zippyops@gmail.com**](mailto:zippyops@gmail.com)

**✆+91 7010585768**

# adding plugins in jira for rundeck.

Login to jira service disk and click setting and in that click Manage apps

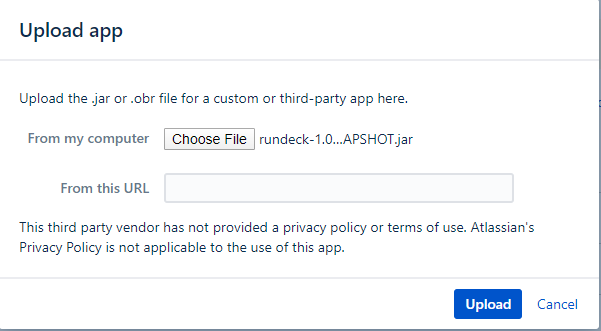


Click Find new apps

Search for jira cli and install this package **Bob Swift Atlassian Apps - Jira Command Line Interface (CLI).** Also search for Approvals and install for **Approvals – advanced issue acceptance.** Also search for Approvals and install for **Adaptavist ScriptRunner for JIRA.**

First download the Rundeck plugin by this link, <https://marketplace.atlassian.com/download/apps/1219585/version/1000000>

Now click Manage apps, there click upload apps, here choose the rundeck plugin for your computer and upload and then install.

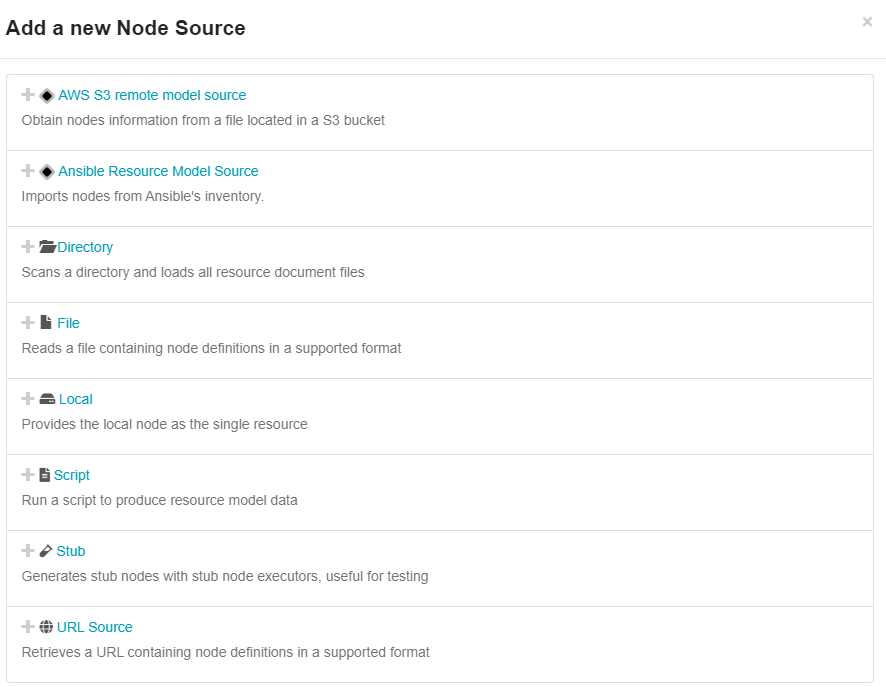


Adding nodes,

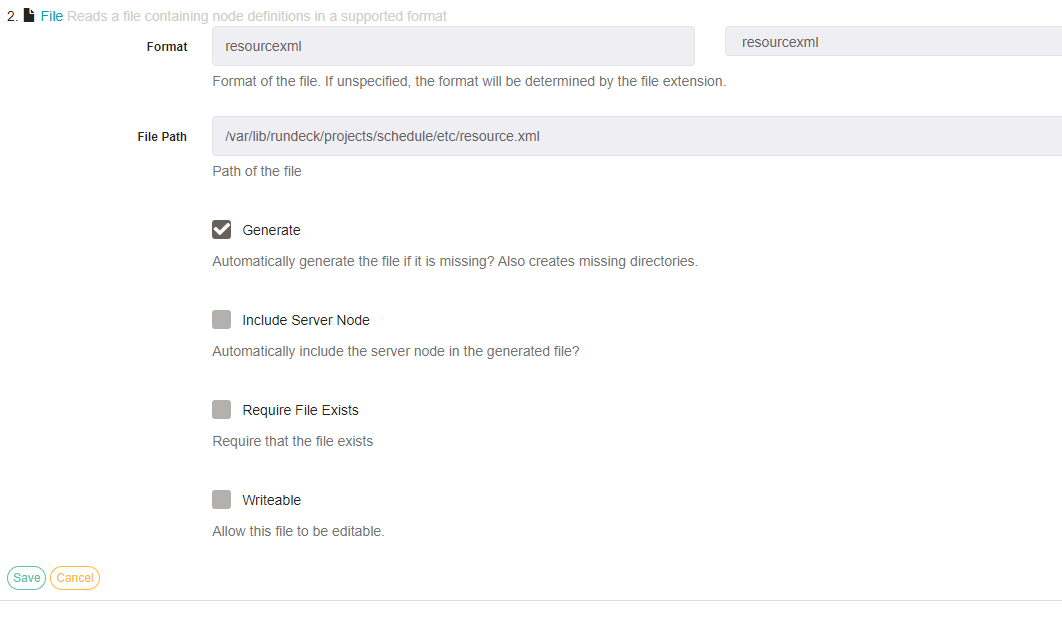
Click Edit Nodes… and in that click Add the new nodes



Select file,



Mention and save it,



Then open terminal of rundeck master,

Download **Jira-cli-jar** and copy it to the rundeck home path.

Download **Py-winrm-plugin** for windows connectivity and copy to libext directory in the home path of rundeck.

[root@rundeck rundeck]# cat /var/lib/rundeck/projects/schedule/etc/resource.xml

<?xml version="1.0" encoding="UTF-8"?>

<project>

<node name="rundeckcentosnode" hostname="rundeckcentosnode.zippyops.com" osName="Centos" ssh-keypath="/var/lib/rundeck/projects/schedule/etc/core.pem" ssh-authentication="privateKey" username="centos"/>

<node name="rundeckubuntunode" hostname="rundeckubuntunode.zippyops.com" osName="Ubuntu" ssh-keypath="/var/lib/rundeck/projects/schedule/etc/core.pem" ssh-authentication="privateKey" username="ubuntu"/>

<node name="rundeckwindowsnode" hostname="rundeckwindowsnode.zippyops.com" osName="Windows" username="rundeck" ps-authentication-type="CredSSP" winrm-protocol="http" winrm-cmd="powershell" winrm-password-storage-path="keys/winpasswd" file-copier="WinRMcpPython" node-executor="WinRMPython"/>

<node name="Nfs\_Machine" hostname="enduserwin10.zippyops.com" osName="Windows" username="zippyops" ps-authentication-type="CredSSP" winrm-protocol="https" winrm-cmd="powershell" winrm-password-storage-path="keys/nfspasswd" file-copier="WinRMcpPython" node-executor="WinRMPython"/>

</project>

Install mailx in rundeck master,

[root@rundeck resetpass]# yum install mailx

Loaded plugins: fastestmirror

Determining fastest mirrors

epel/x86\_64/metalink

rundeck-release-bintray

Package mailx-12.5-19.el7.x86\_64 already installed and latest version

Nothing to do

Go to this file and configure the mail address by adding this lines in the mail.rc file.

[root@rundeck resetpass]# cat /etc/mail.rc

set from=xxxxxxxxxx@zippyops.in

set smtp=smtp://mail.zippyops.in:25

set smtp-auth=login

set smtp-auth-user=xxxxxx@zippyops.in

set smtp-auth-password=xxxxxxxxxx

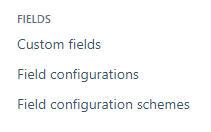
set ssl-verify=ignore

# password reset

Creating custom field in jira, click setting and in that click issue,

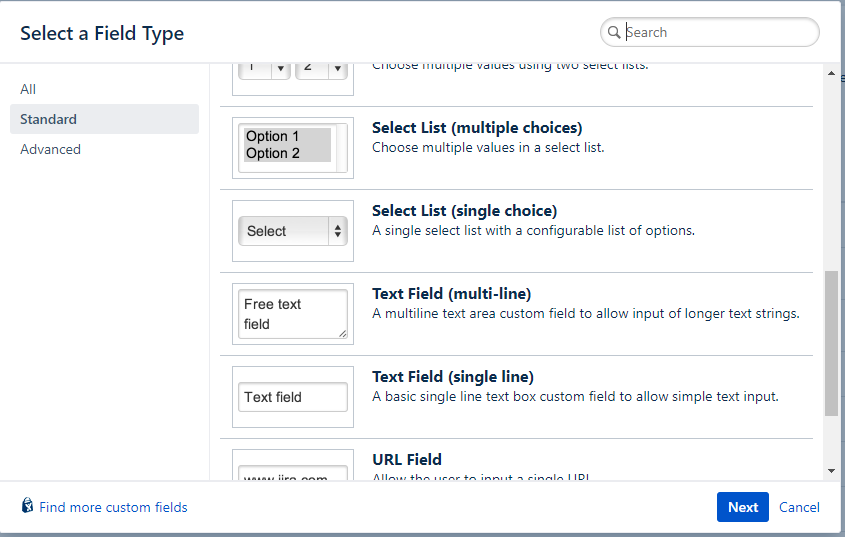


Click on Custom fields,

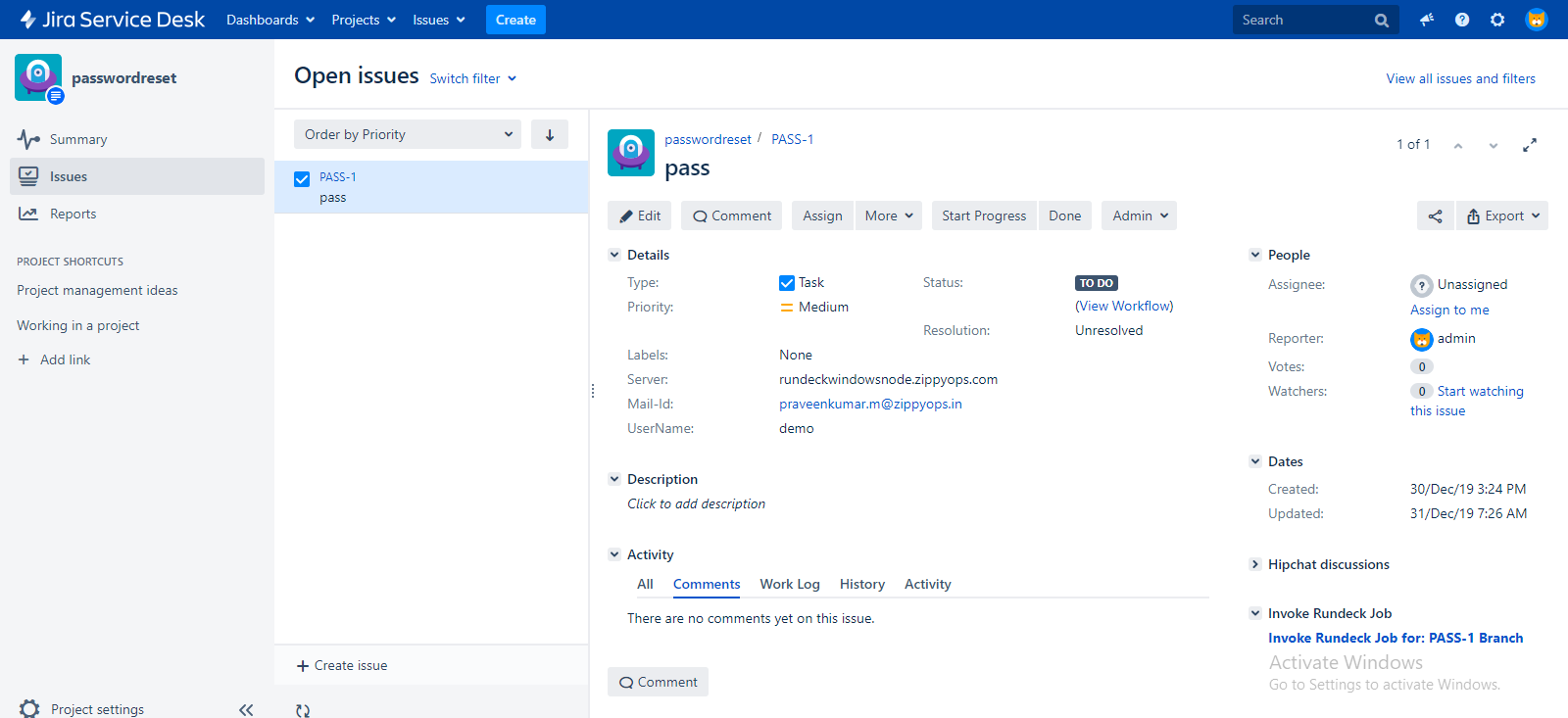


In that click Add Custom field in the right corner, In that create 3 field

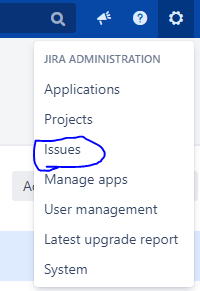
1. Username field in Text field (Single line)
2. Mail\_Id field in Text field (Single line)
3. Server field in Select List (Multiple choice)



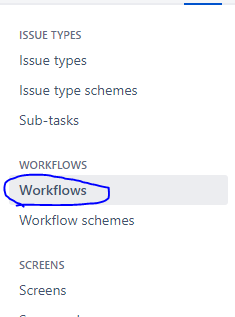
Create the new project with the name passwordreset and issue id as PASS.



Create the workflow, click on issue,

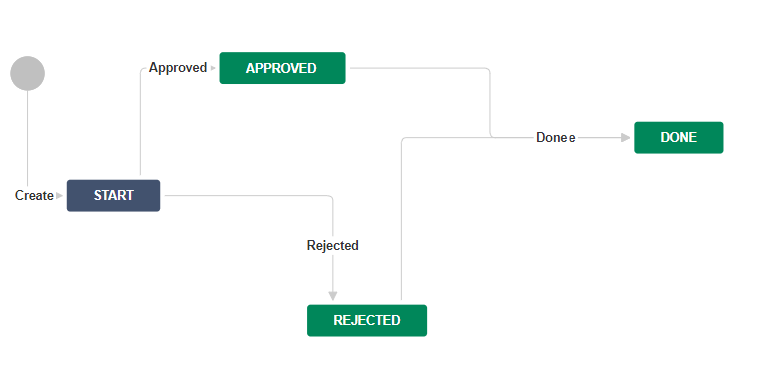


Click on workflow, there click on edit in which project you have to make change in workflow,

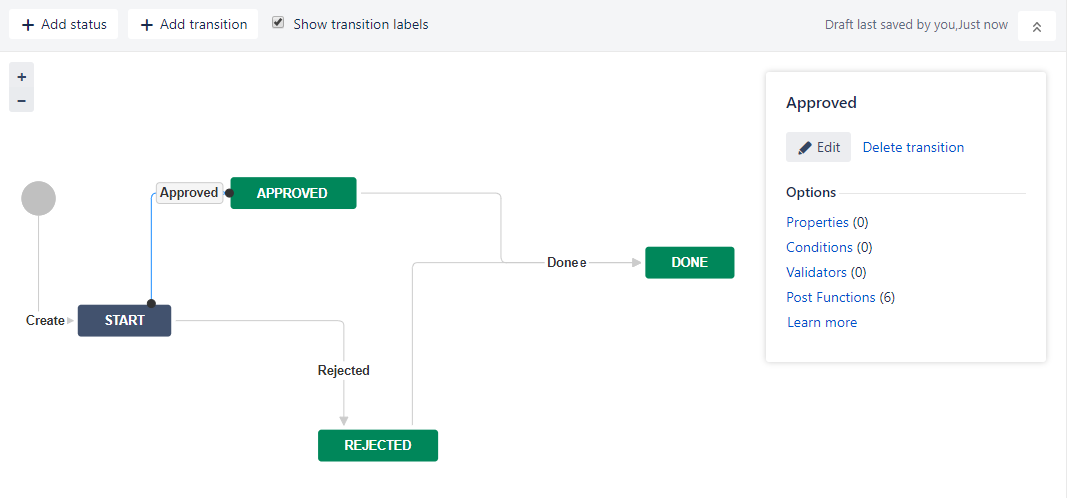




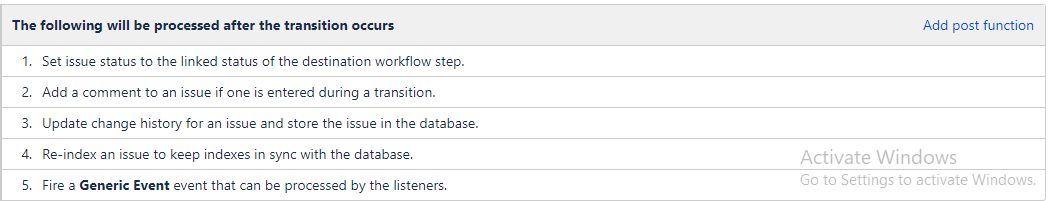
Create the workflow like this,



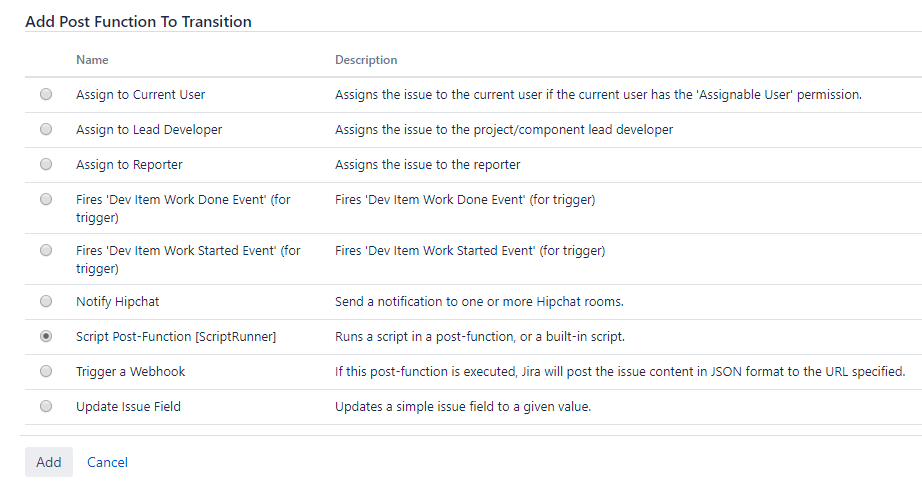
Click on the approve transition, the new dialog box will open there click post function,



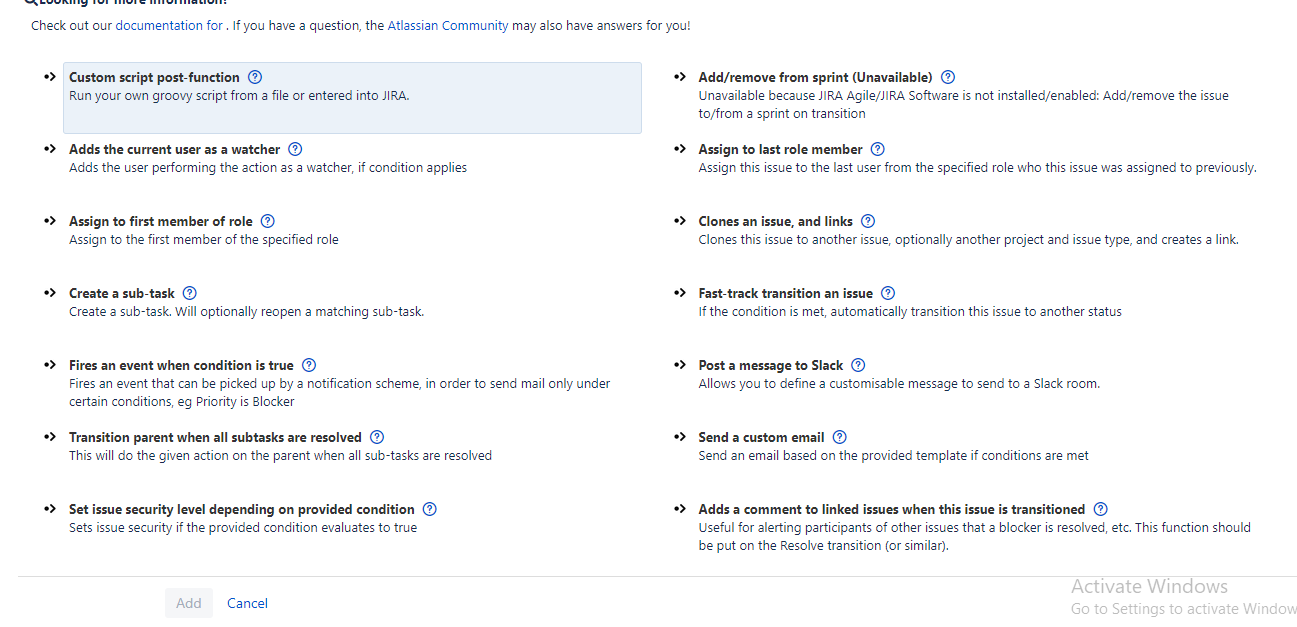
Click on add post function



Click on script post-function and click Add.



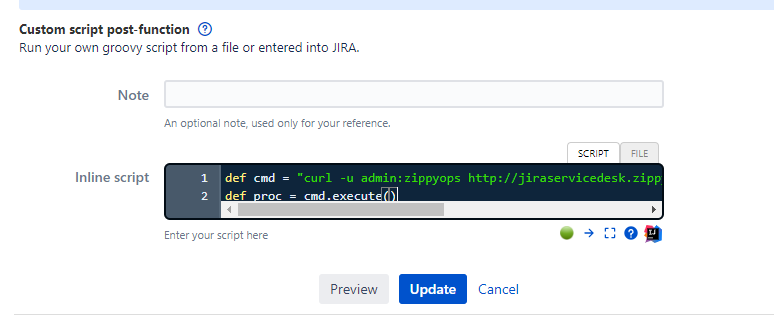
Click Custom script post function,



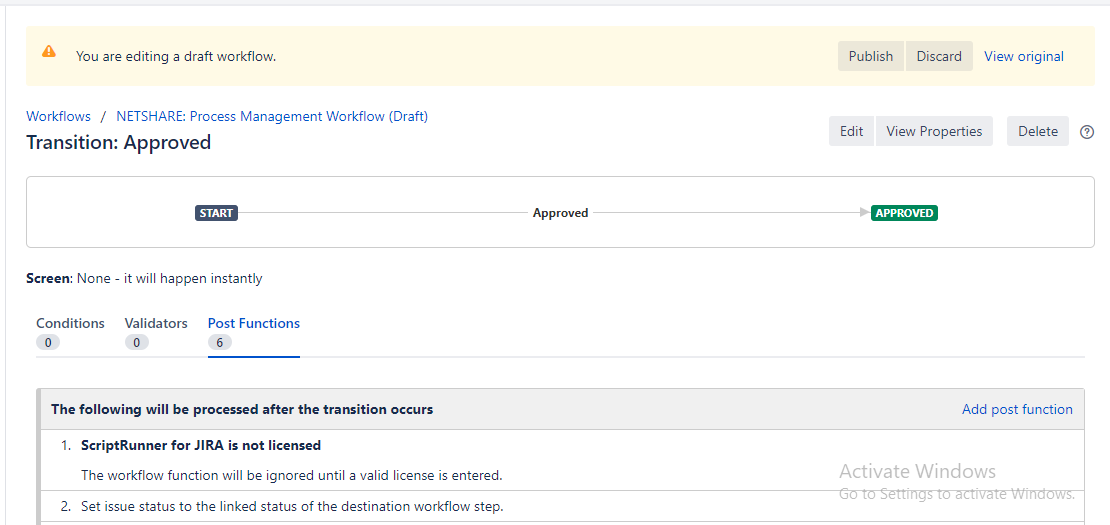
Type the script and Click on update,

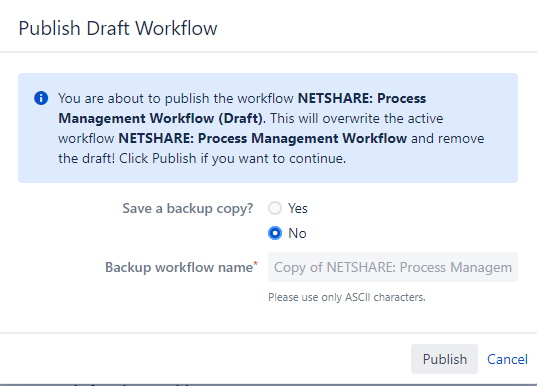
def cmd = "curl -u admin:zippyops http://jiraservicedesk.zippyops.com:8080/plugins/servlet/rundeck?issueKey=${issue.key}"

def proc = cmd.execute()



Click on publish,

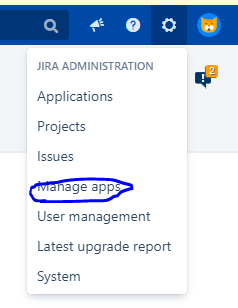




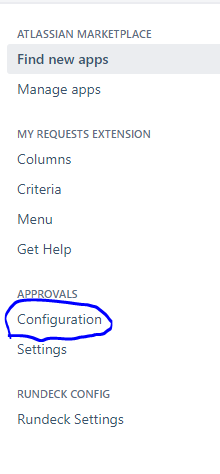
Like this do for the rejection transition,

Approval setup,

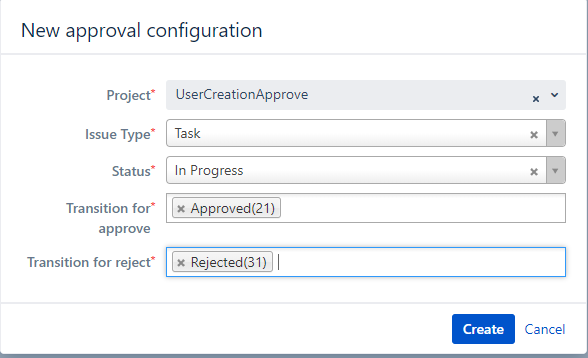
Go to settings, and click manage apps,



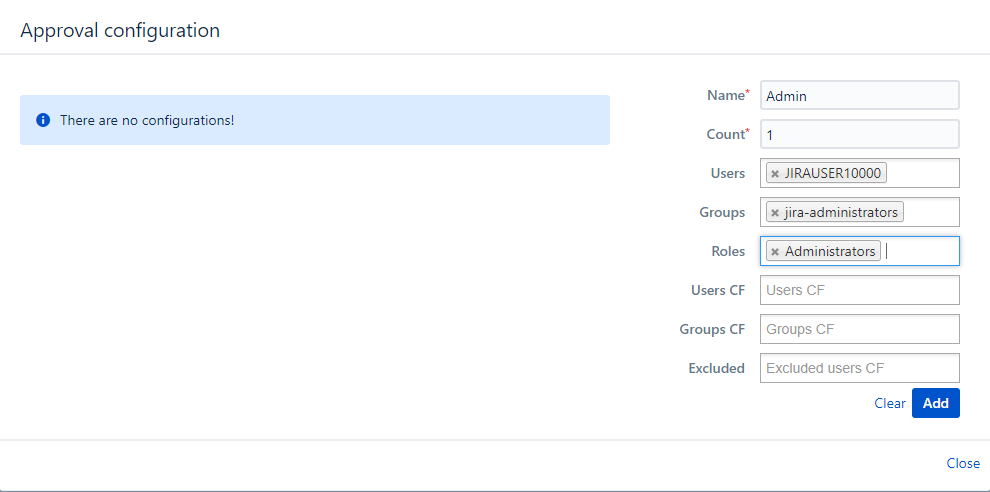
Click configuration below approval, if it not available install the plugins.



Click ADD, then click create



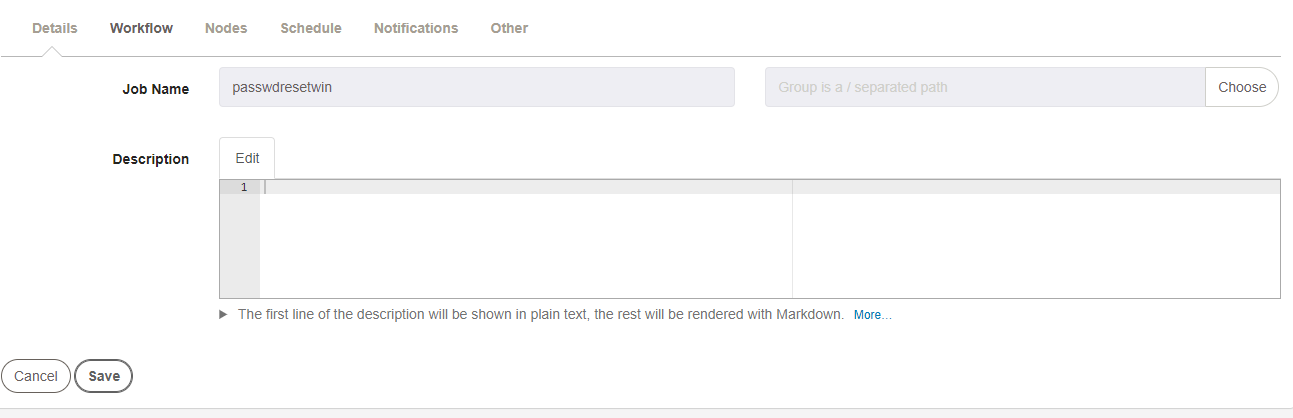
After creating click configuration, and click add



Create the two job for Passwd reset in rundeck,

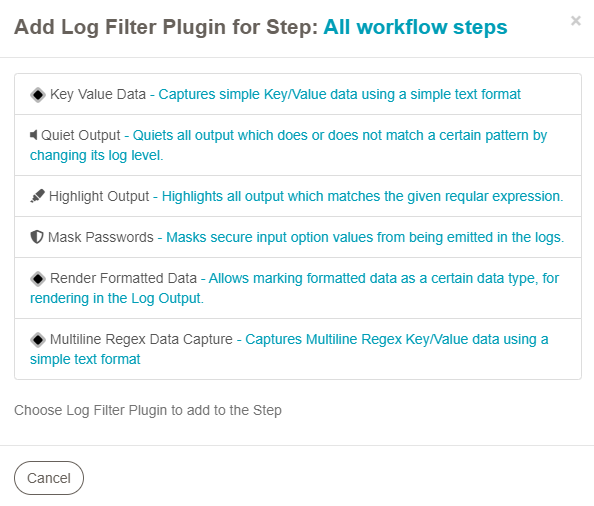
One for linux and another for windows,

First we going to create jobs for windows,

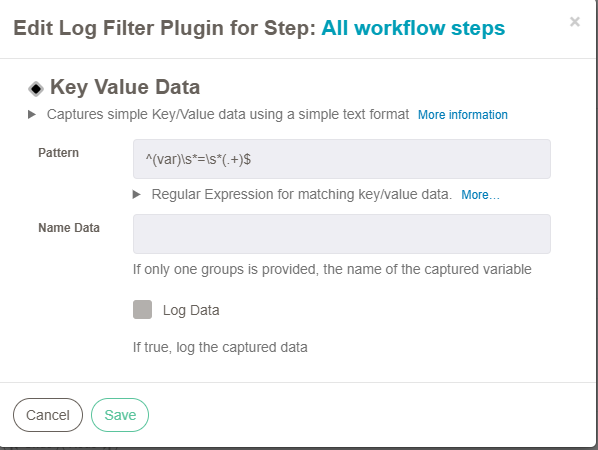


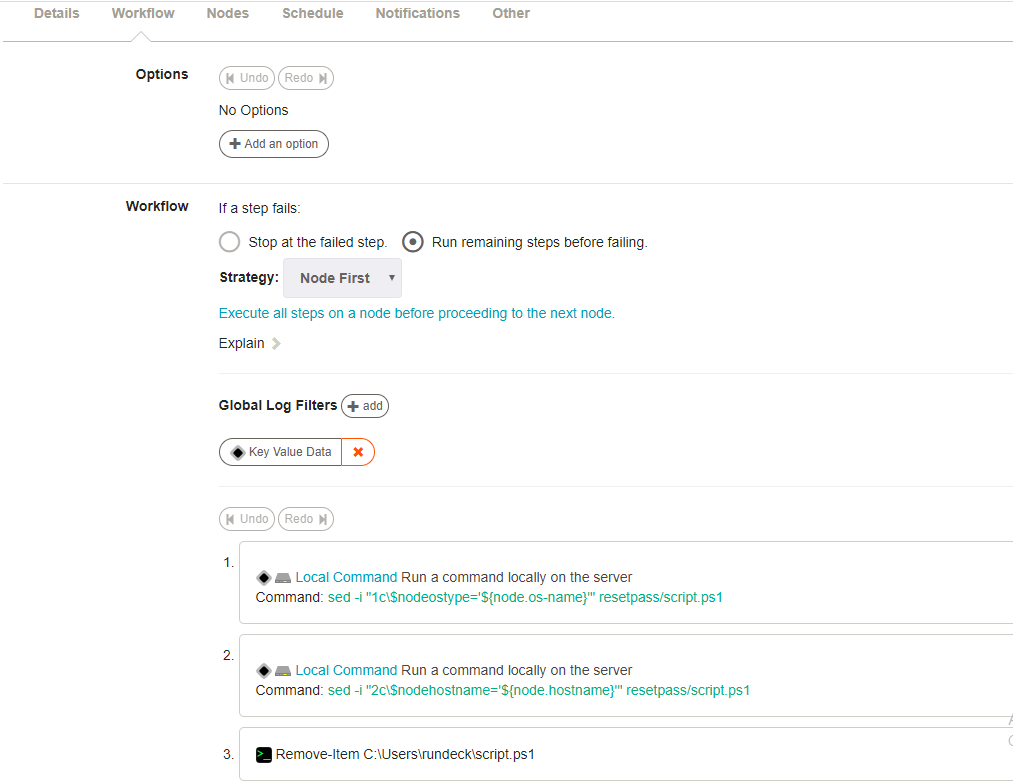
Workflows,

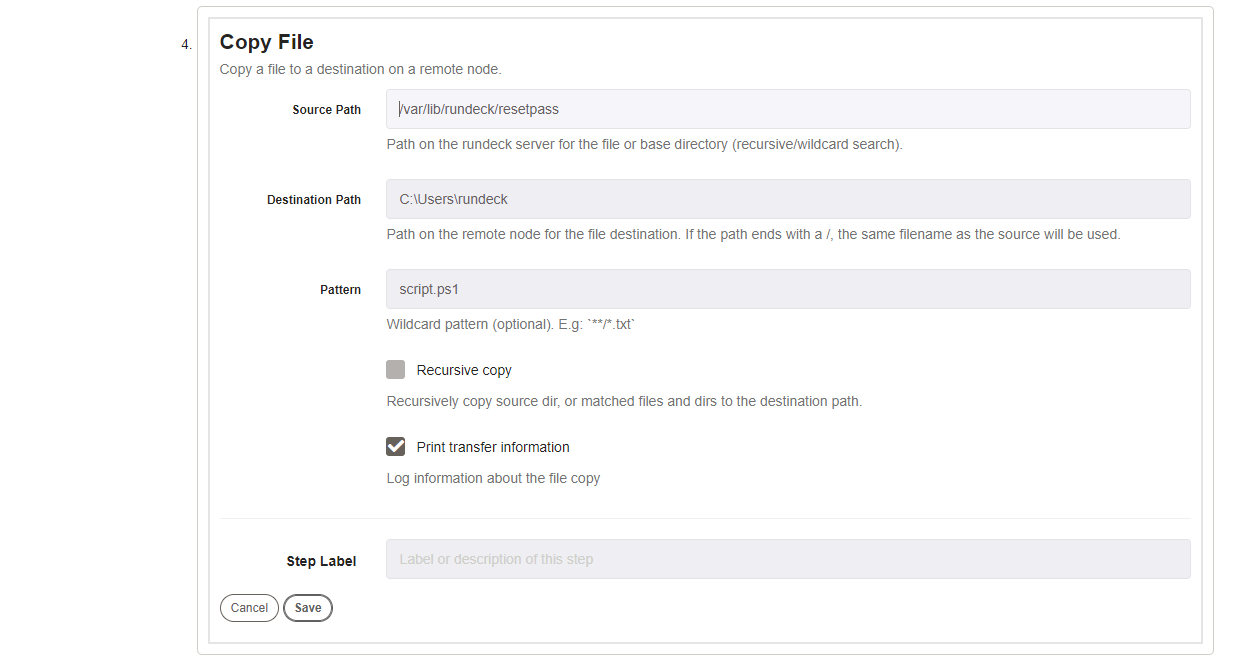
Click on Add Log Filter, and in that click Key Value Data.



Give the pattern like this, and save.

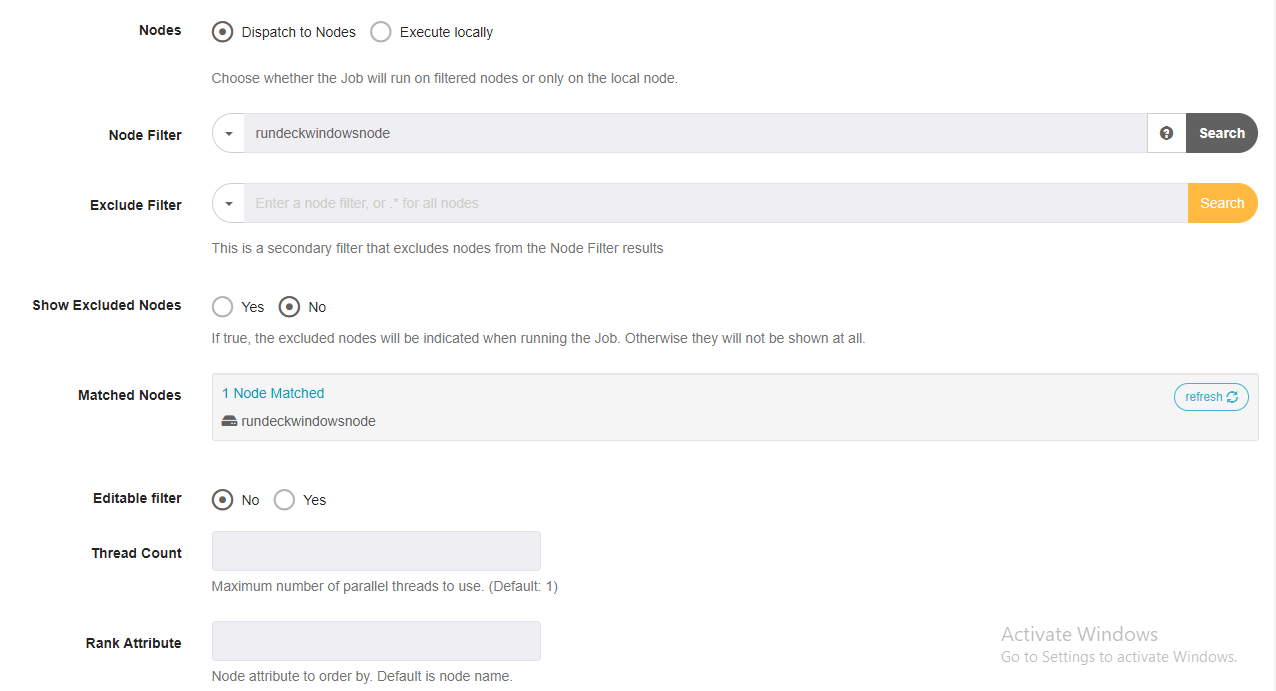


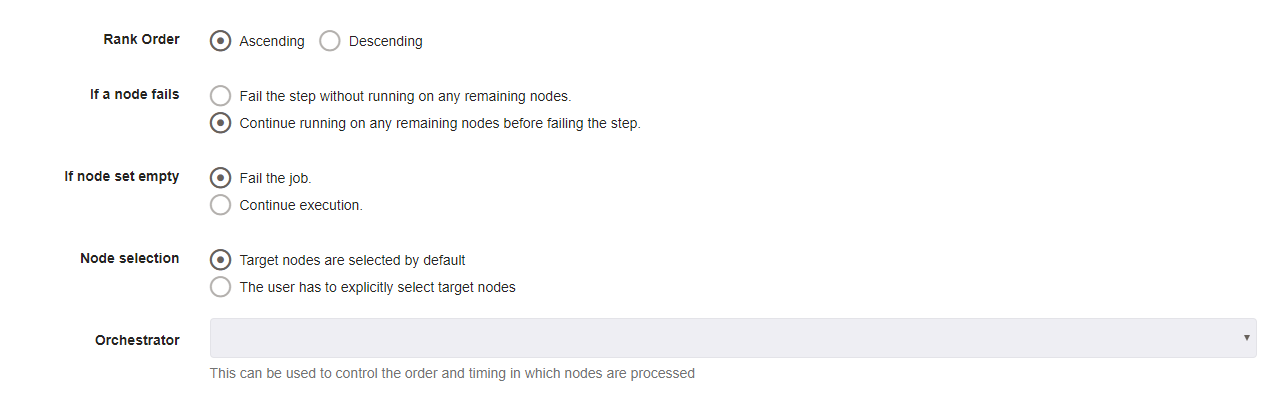






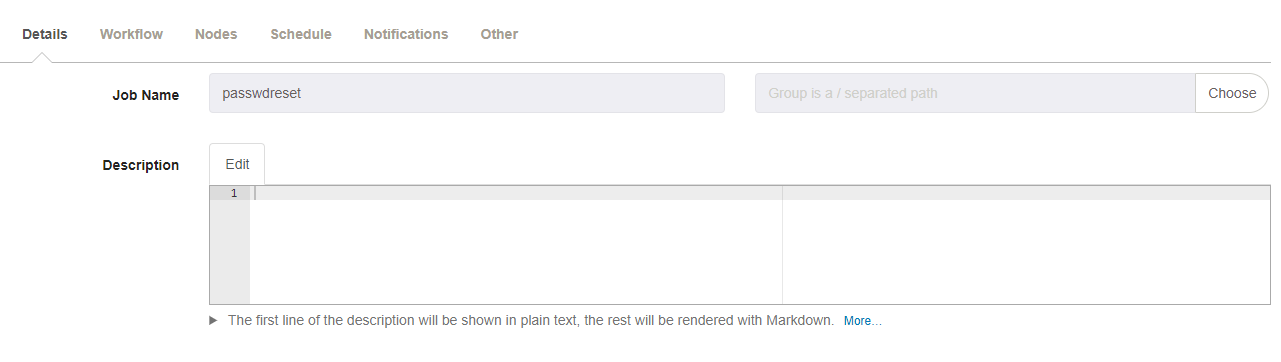
Nodes,





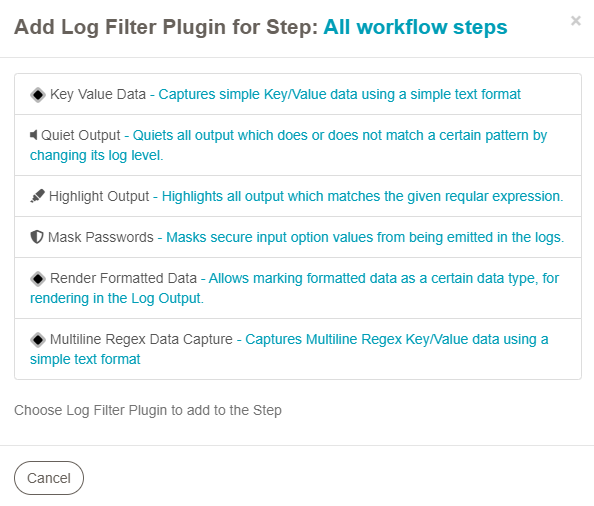
Click create.

Now create the job for linux machine,

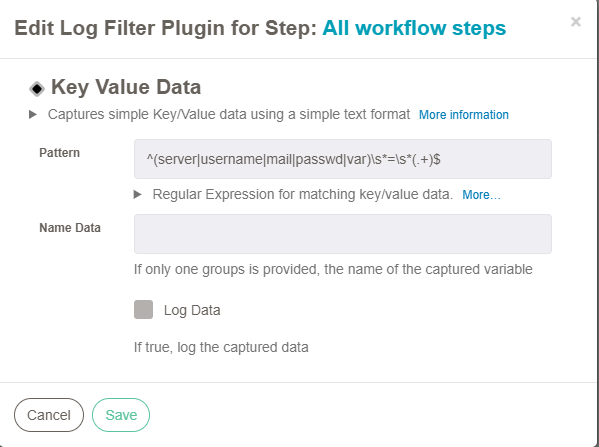


Workflow,

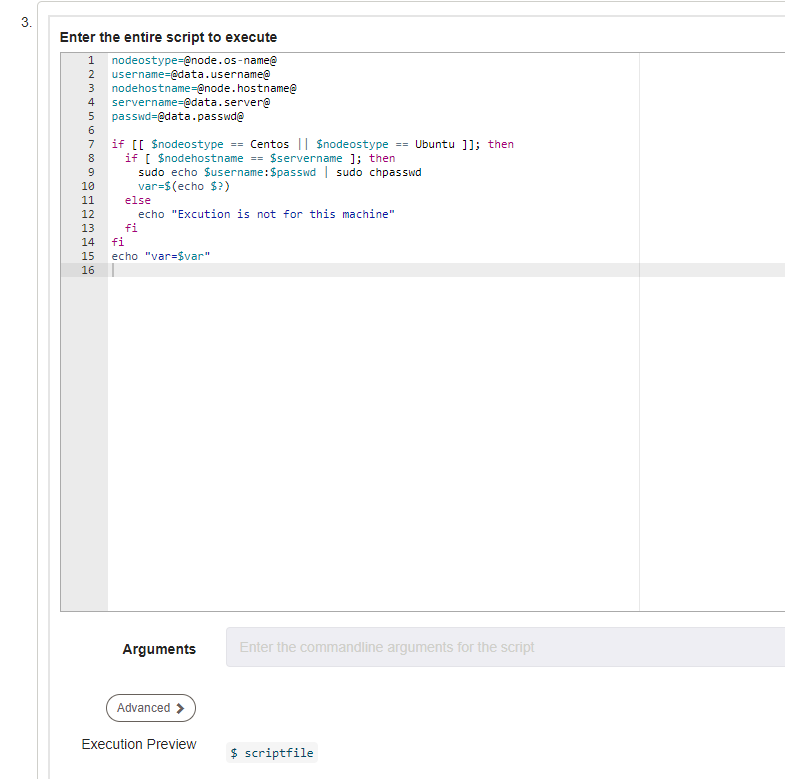
Click on Add Log Filter, and in that click Key Value Data.

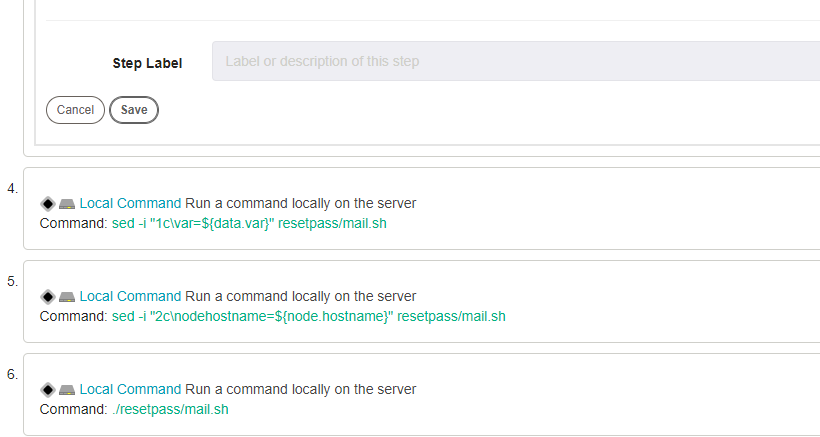


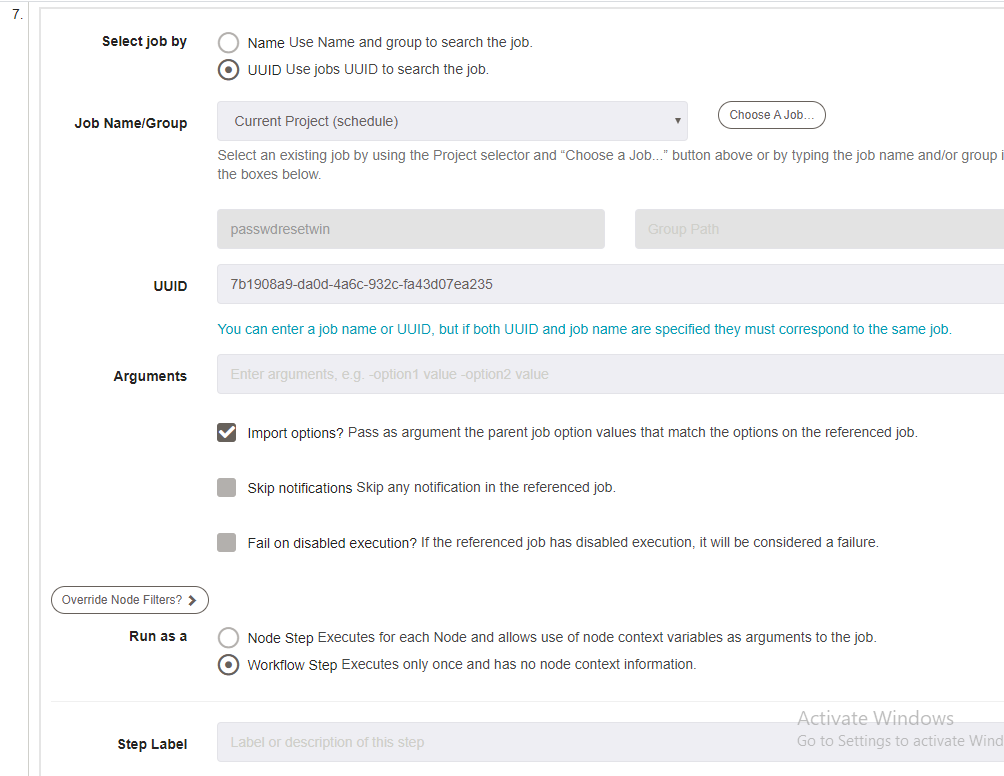
Give the pattern like this, and save.



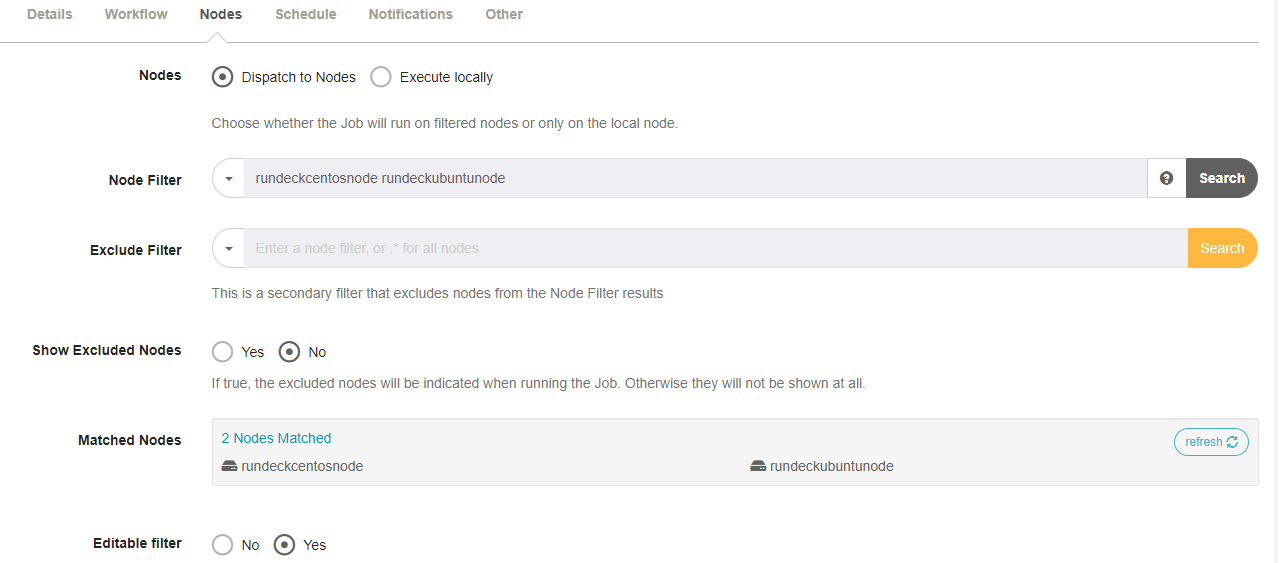


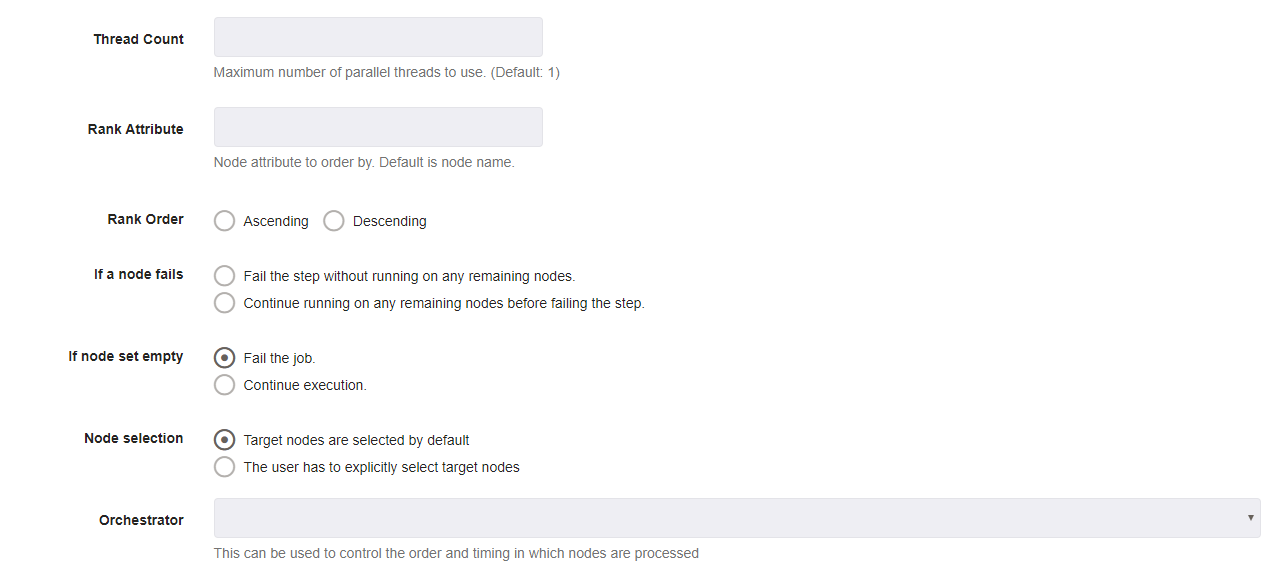






Nodes,





Now click create,

Open the Rundeck terminal,

[root@rundeck ~]# cd /var/lib/rundeck/

[root@rundeck rundeck]# mkdir resetpass

[root@rundeck rundeck]# cd resetpass

[root@rundeck resetpass]# cat resetpass.sh

issue=PASS-1

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --ser ver http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops " --action getFieldValue --issue "$issue" --field "Server" | tail -1)

user=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --serve r http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "UserName" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --serve r http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

pass=$(openssl rand -base64 8)

echo "server=$server"

echo "username=$user"

echo "mail=$mail"

echo "passwd=$pass"

sed -i "3c\$servername='$server'" resetpass/script.ps1

sed -i "4c\$username='$user'" resetpass/script.ps1

sed -i "5c\$passwd='$pass'" resetpass/script.ps1

sed -i "3c\servername=$server" resetpass/mail.sh

sed -i "4c\username=$user" resetpass/mail.sh

sed -i "5c\passwd=$pass" resetpass/mail.sh

sed -i "6c\mail=$mail" resetpass/mail.sh

[root@rundeck resetpass]# cat resetpass.sh

issue=PASS-1

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Server" | tail -1)

user=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "UserName" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

pass=$(openssl rand -base64 8)

echo "server=$server"

echo "username=$user"

echo "mail=$mail"

echo "passwd=$pass"

sed -i "3c\$servername='$server'" resetpass/script.ps1

sed -i "4c\$username='$user'" resetpass/script.ps1

sed -i "5c\$passwd='$pass'" resetpass/script.ps1

sed -i "3c\servername=$server" resetpass/mail.sh

sed -i "4c\username=$user" resetpass/mail.sh

sed -i "5c\passwd=$pass" resetpass/mail.sh

sed -i "6c\mail=$mail" resetpass/mail.sh

[root@rundeck resetpass]# cd ..

[root@rundeck rundeck]# cd resetpass/

[root@rundeck resetpass]# cat resetpass.sh

issue=PASS-1

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Server" | tail -1)

user=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "UserName" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

pass=$(openssl rand -base64 8)

echo "server=$server"

echo "username=$user"

echo "mail=$mail"

echo "passwd=$pass"

sed -i "3c\$servername='$server'" resetpass/script.ps1

sed -i "4c\$username='$user'" resetpass/script.ps1

sed -i "5c\$passwd='$pass'" resetpass/script.ps1

sed -i "3c\servername=$server" resetpass/mail.sh

sed -i "4c\username=$user" resetpass/mail.sh

sed -i "5c\passwd=$pass" resetpass/mail.sh

sed -i "6c\mail=$mail" resetpass/mail.sh

[root@rundeck resetpass]# cat script.ps1

$nodeostype='Windows'

$nodehostname='rundeckwindowsnode.zippyops.com'

$servername='rundeckwindowsnode.zippyops.com'

$username='demo'

$passwd='kT66KS8B+Lc='

if($nodeostype -eq "Windows"){

if($nodehostname -eq $servername){

net user $username $passwd

$var=$(echo $?)

echo "var=$var"

}else{

echo "Excution is not for this machine"

}

}

[root@rundeck resetpass]# cat mail.sh

var=False

nodehostname=rundeckwindowsnode.zippyops.com

servername=rundeckwindowsnode.zippyops.com

username=demo

passwd=kT66KS8B+Lc=

mail=praveenkumar.m@zippyops.in

if [ $nodehostname == $servername ]; then

if [[ $var == 0 || $var == True ]]; then

echo "This is your password $passwd for the user $username in the specfied server $servername" | mail -s "password reset done" $mail

elif [[ $var == 1 || $var == False ]]; then

echo "$username user not found in the specfied server $servername" | mail -s "password reset not done" $mail

else

echo "Execution is not for this machine"

fi

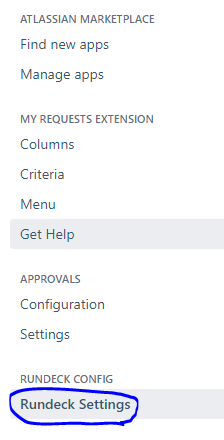
fi

[root@rundeck resetpass]#

Invoke Rundeck job setting, click the Manage apps

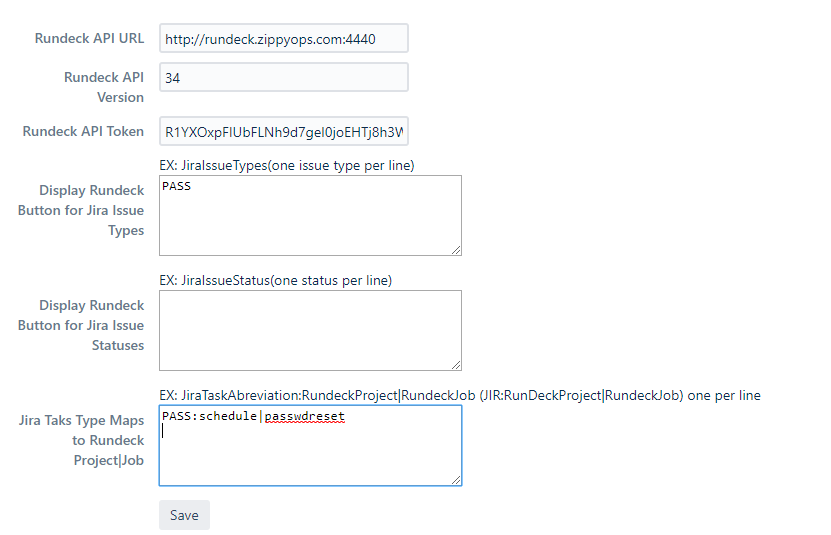


Click Rundeck setting,



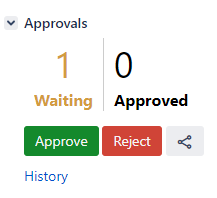
First mention Rundeck API URl and then API verion

Then generate the Api Token and paste here.



Come back to issue,

Click on Approve or Reject.

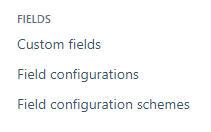


# usercreationApproval

Creating custom field in jira, click setting and in that click issue,

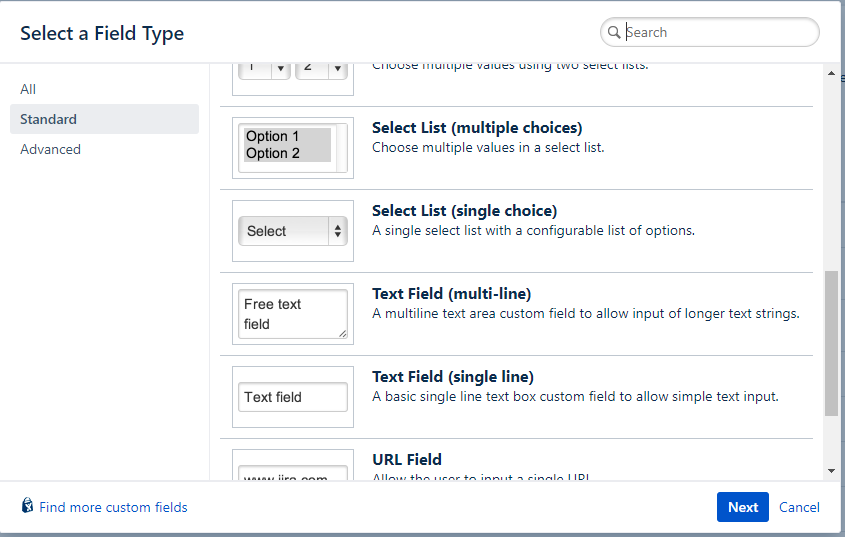


Click on Custom fields,



In that click Add Custom field in the right corner, In that create 3 field

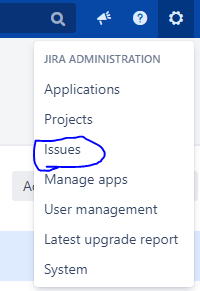
1. Username field in Text field (Single line)
2. Mail\_Id field in Text field (Single line)
3. Server field in Select List (Multiple choice)
4. Access field in Radio Button



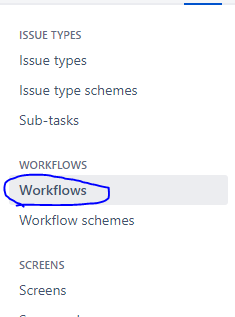
Create the new project with the name UsercreationApproval and issue id as ADMINACCES.



Create the workflow, click on issue,

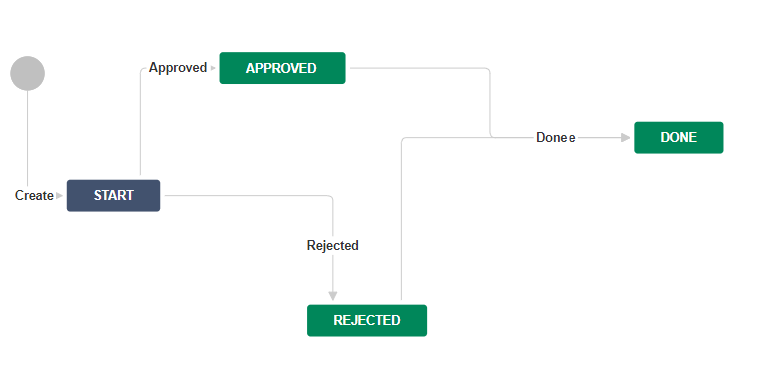


Click on workflow, there click on edit in which project you have to make change in workflow,

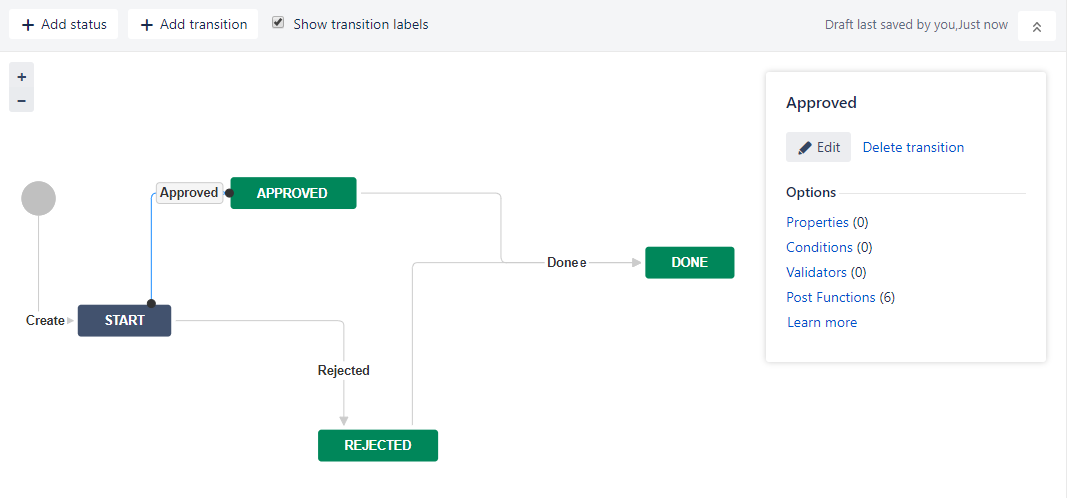




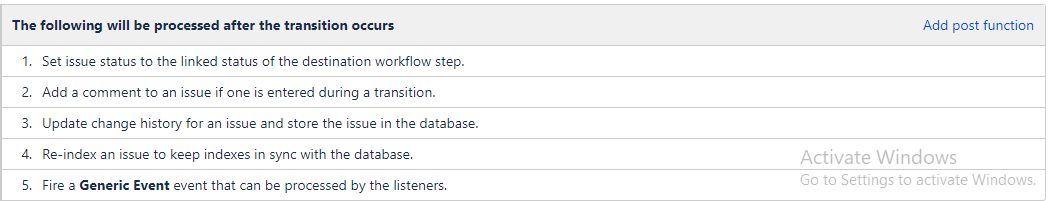
Create the workflow like this,



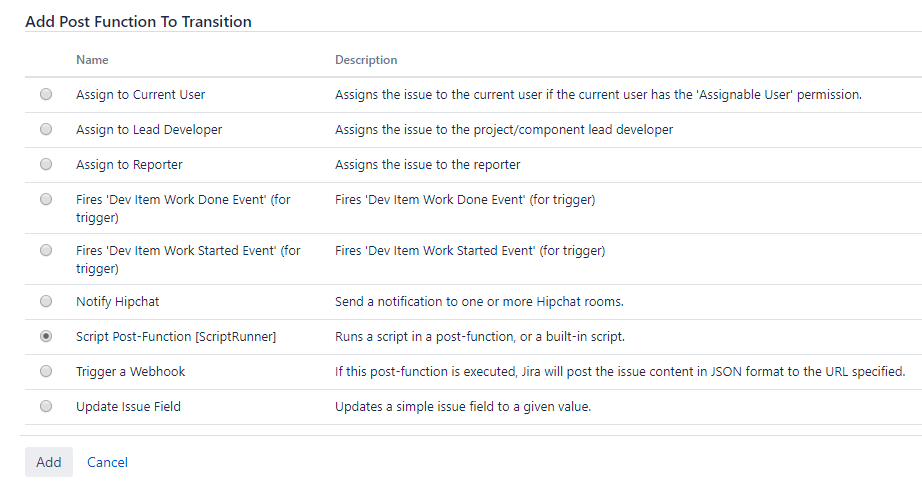
Click on the approve transition, the new dialog box will open there click post function,



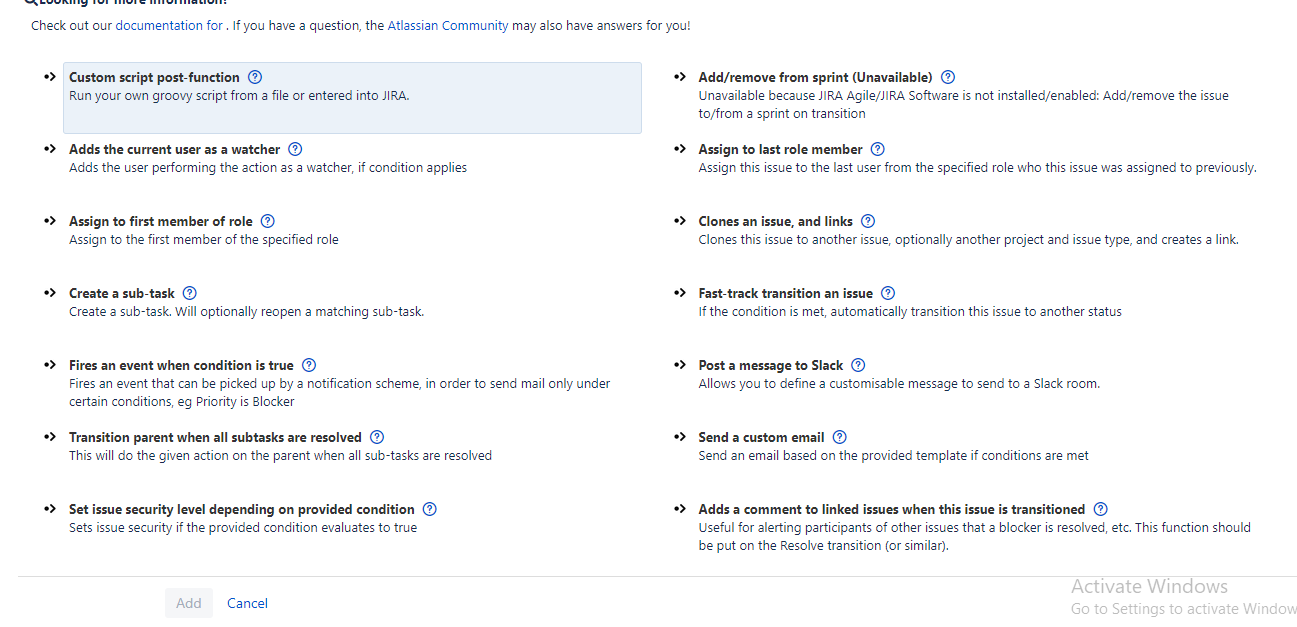
Click on add post function



Click on script post-function and click Add.



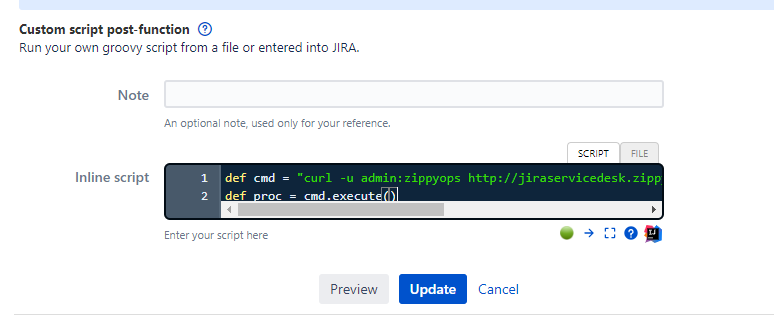
Click Custom script post function,



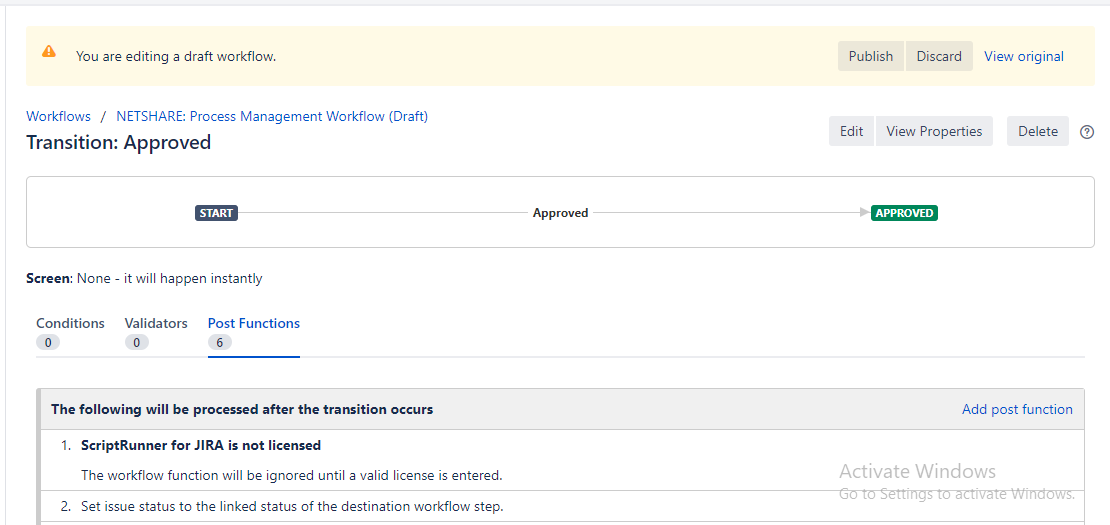
Type the script and Click on update,

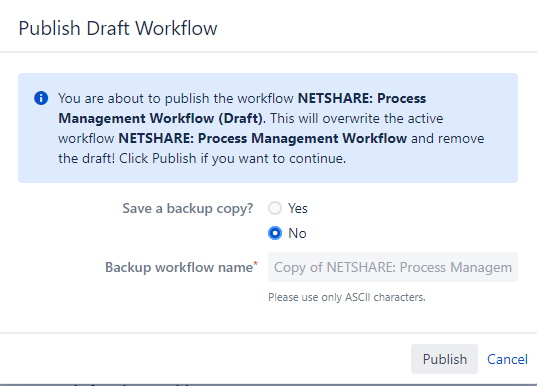
def cmd = "curl -u admin:zippyops http://jiraservicedesk.zippyops.com:8080/plugins/servlet/rundeck?issueKey=${issue.key}"

def proc = cmd.execute()



Click on publish,

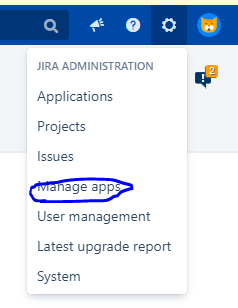




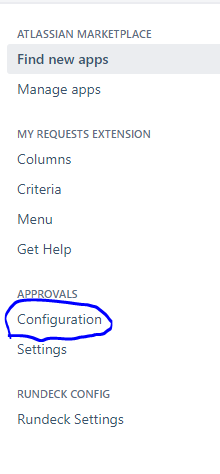
Like this do for the rejection transition,

Approval setup,

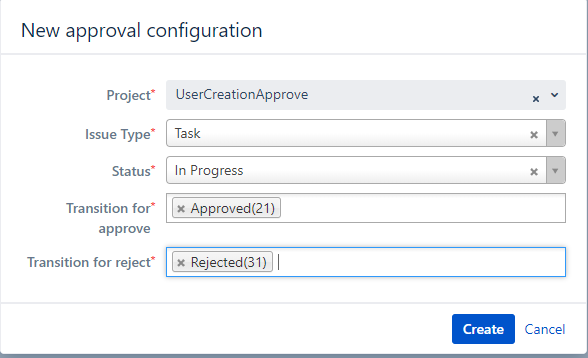
Go to settings, and click manage apps,



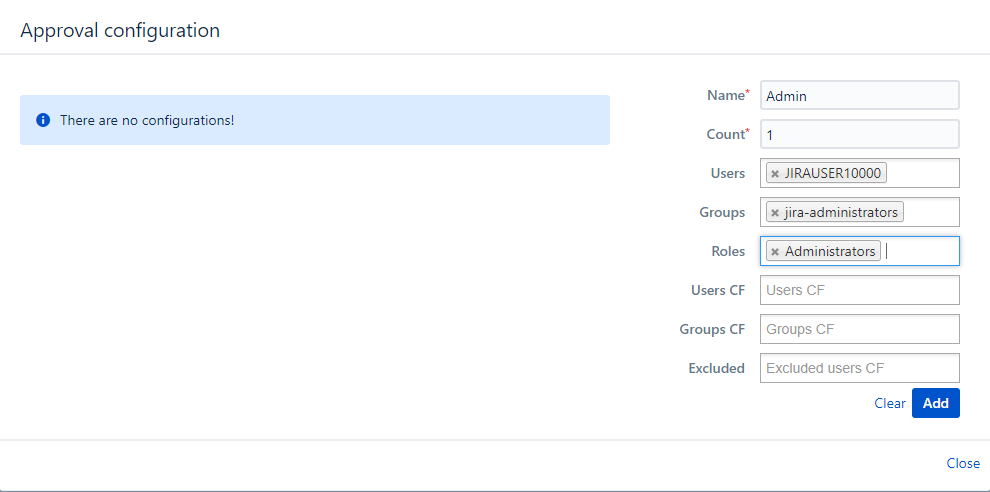
Click configuration below approval, if it not available install the plugins.



Click ADD, then click create



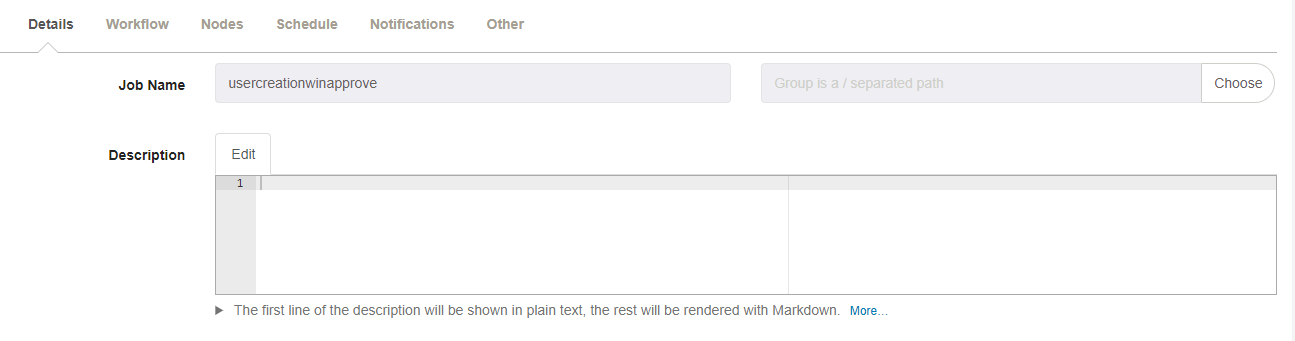
After creating click configuration, and click add



Create the two job for Usercreation Approve in rundeck,

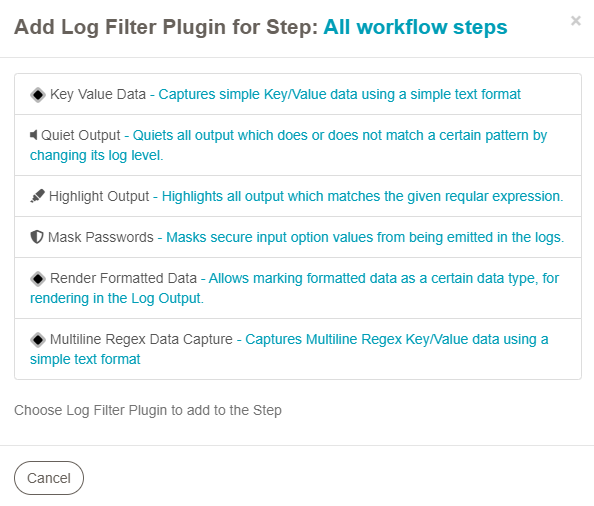
One for linux and another for windows,

First we going to create jobs for windows,

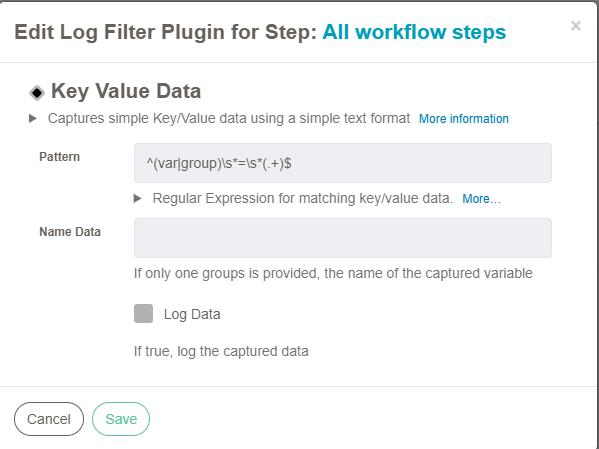


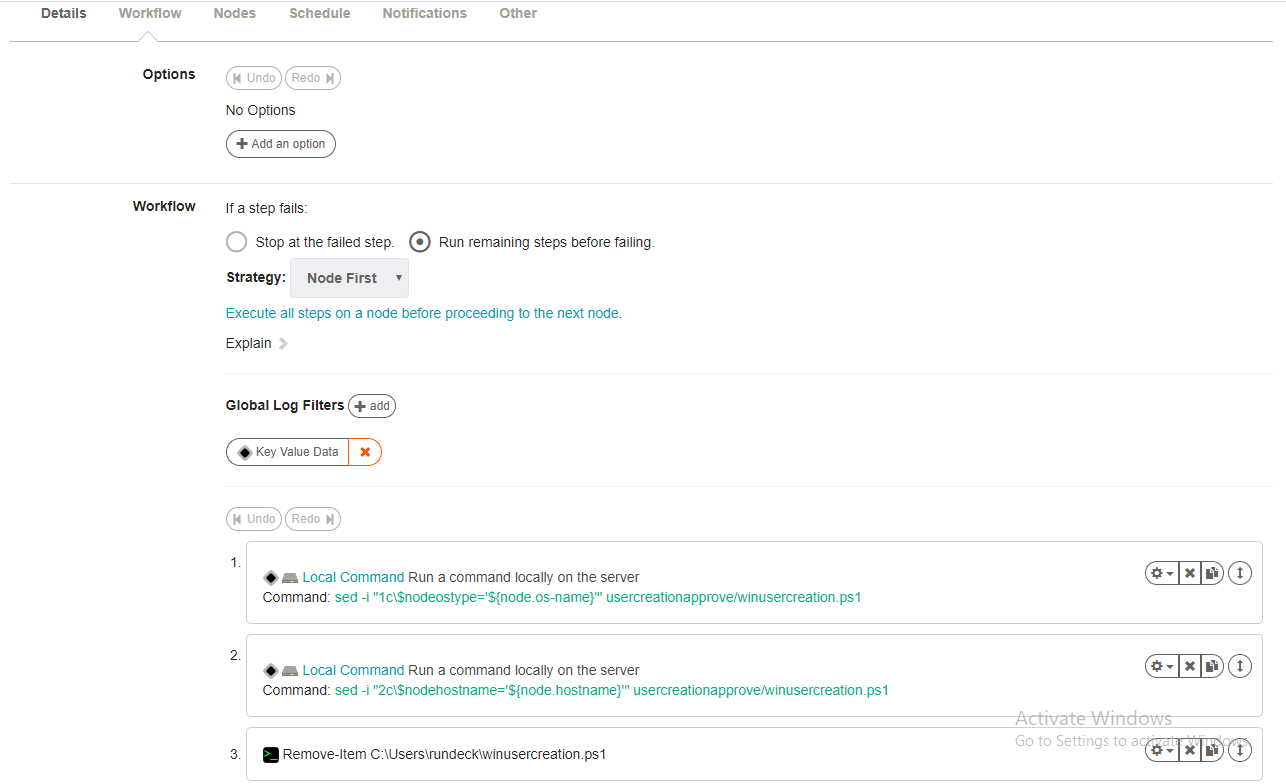
Workflow,

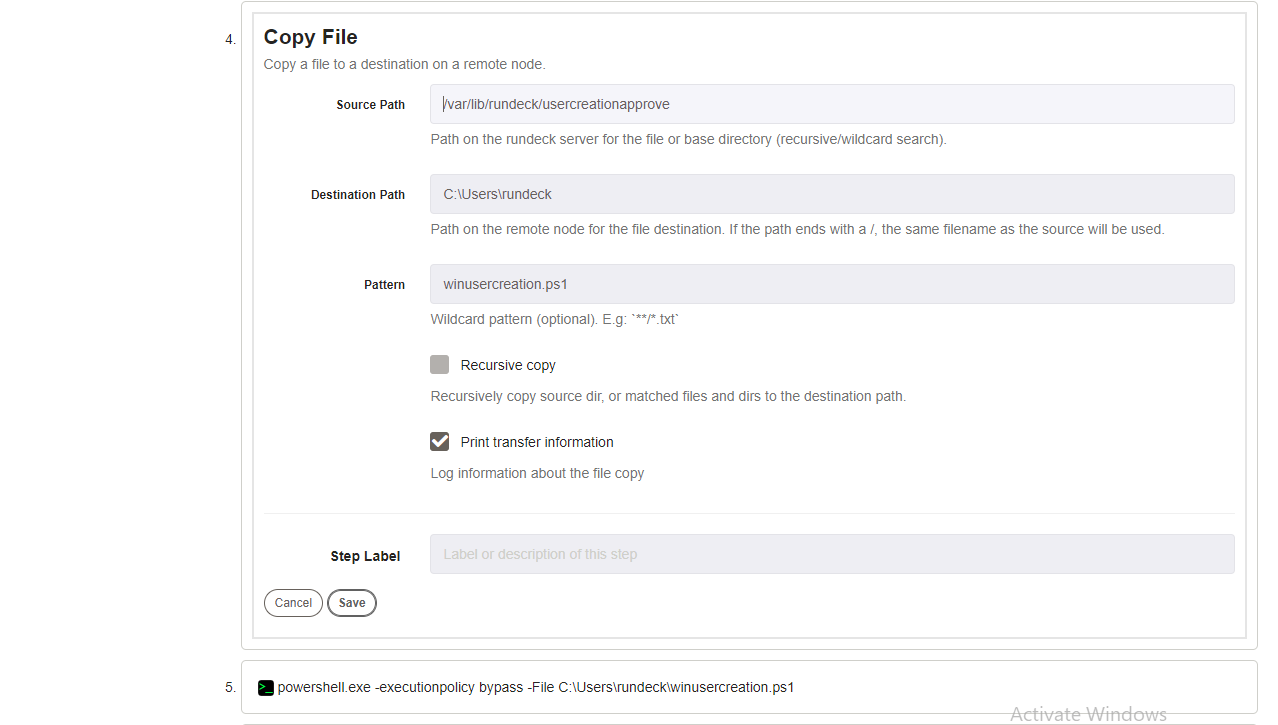
Click on Add Log Filter, and in that click Key Value Data.



Give the pattern like this, and save.

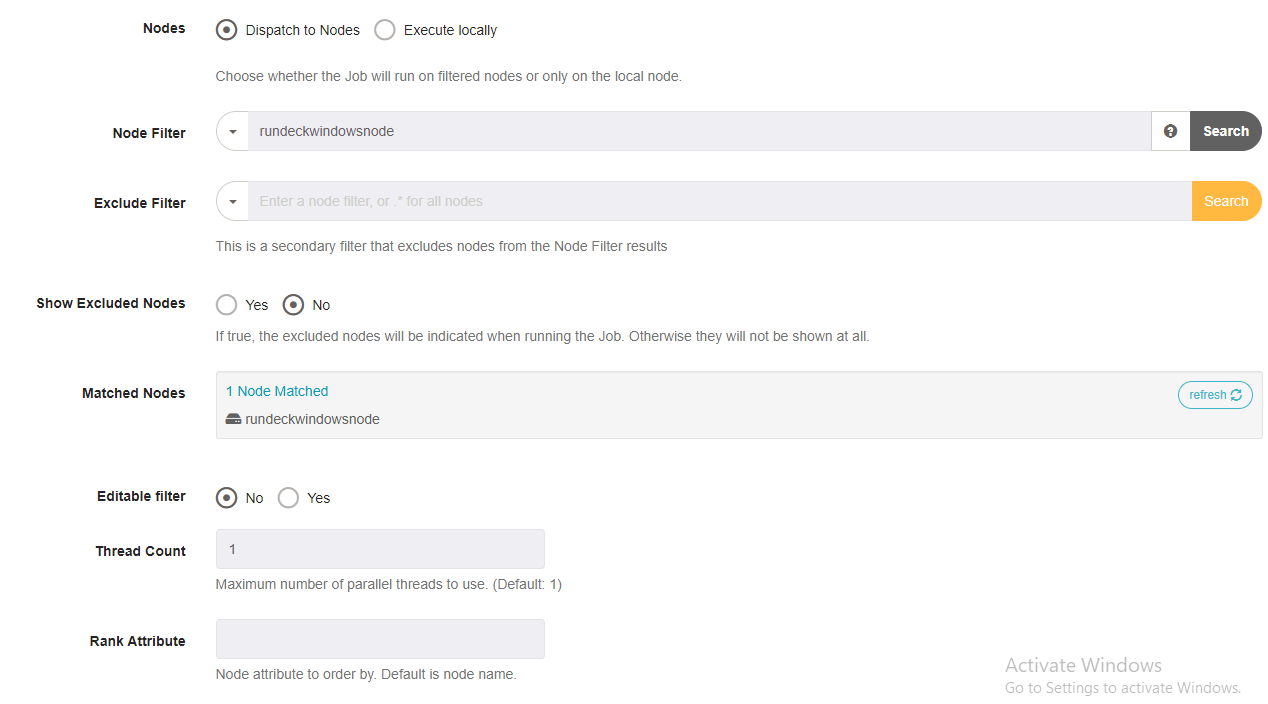


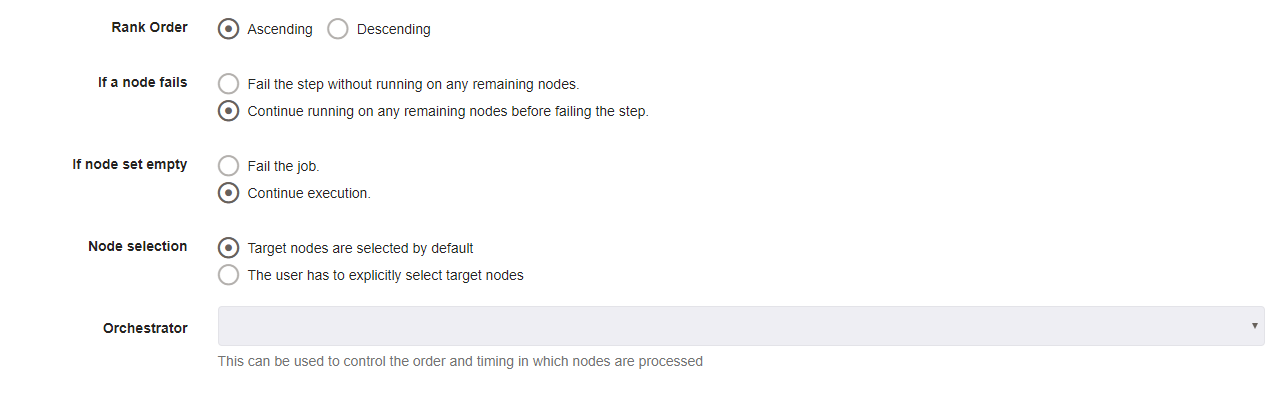






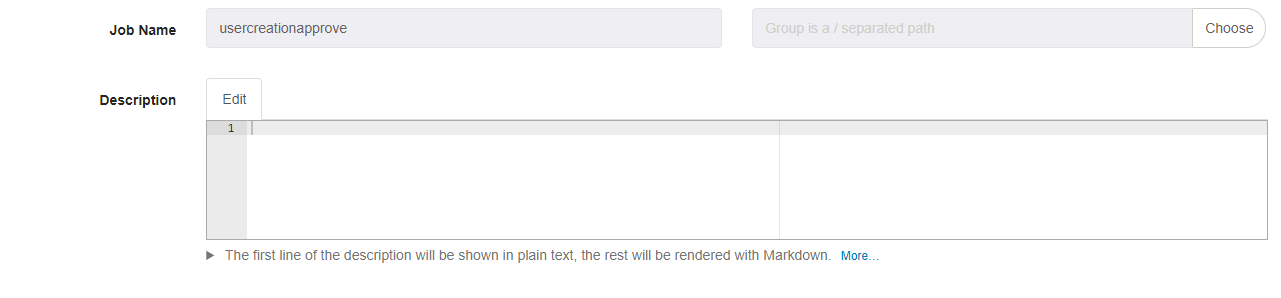
Nodes,





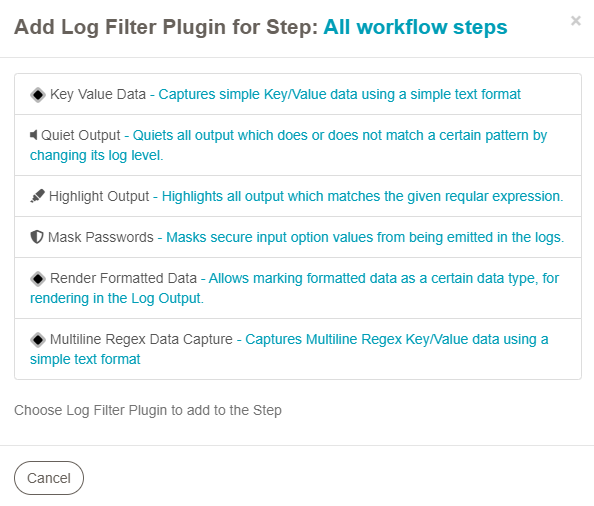
Click create.

Now create the new job for linux,

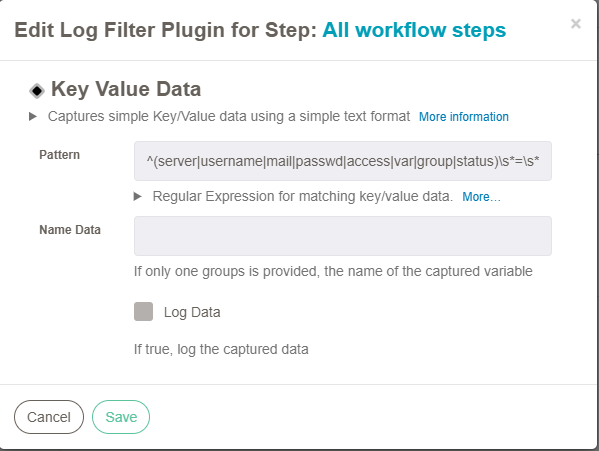


Workflow,

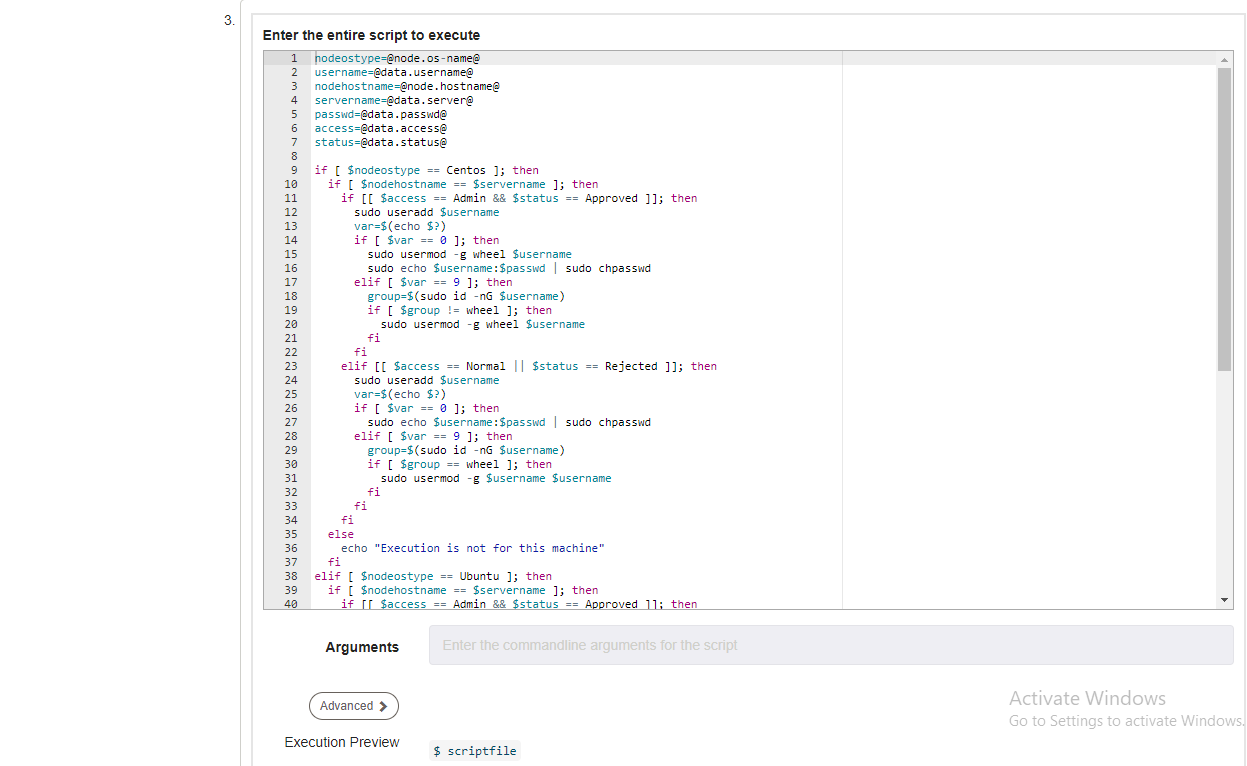
Click on Add Log Filter, and in that click Key Value Data.



Give the pattern like this, and save.







nodeostype=@node.os-name@

username=@data.username@

nodehostname=@node.hostname@

servername=@data.server@

passwd=@data.passwd@

access=@data.access@

status=@data.status@

if [ $nodeostype == Centos ]; then

if [ $nodehostname == $servername ]; then

if [[ $access == Admin && $status == Approved ]]; then

sudo useradd $username

var=$(echo $?)

if [ $var == 0 ]; then

sudo usermod -g wheel $username

sudo echo $username:$passwd | sudo chpasswd

elif [ $var == 9 ]; then

group=$(sudo id -nG $username)

if [ $group != wheel ]; then

sudo usermod -g wheel $username

fi

fi

elif [[ $access == Normal || $status == Rejected ]]; then

sudo useradd $username

var=$(echo $?)

if [ $var == 0 ]; then

sudo echo $username:$passwd | sudo chpasswd

elif [ $var == 9 ]; then

group=$(sudo id -nG $username)

if [ $group == wheel ]; then

sudo usermod -g $username $username

fi

fi

fi

else

echo "Execution is not for this machine"

fi

elif [ $nodeostype == Ubuntu ]; then

if [ $nodehostname == $servername ]; then

if [[ $access == Admin && $status == Approved ]]; then

sudo useradd -s /bin/bash -d /home/$username/ -m $username

var=$(echo $?)

if [ $var == 0 ]; then

sudo usermod -g sudo $username

sudo echo $username:$passwd | sudo chpasswd

elif [ $var == 9 ]; then

group=$(sudo id -nG $username)

if [ $group != sudo ]; then

sudo usermod -g sudo $username

fi

fi

elif [[ $access == Normal || $status == Rejected ]]; then

sudo useradd -s /bin/bash -d /home/$username/ -m $username

var=$(echo $?)

if [ $var == 0 ]; then

sudo echo $username:$passwd | sudo chpasswd

elif [ $var == 9 ]; then

group=$(sudo id -nG $username)

if [ $group == sudo ]; then

sudo usermod -g $username $username

fi

fi

fi

else

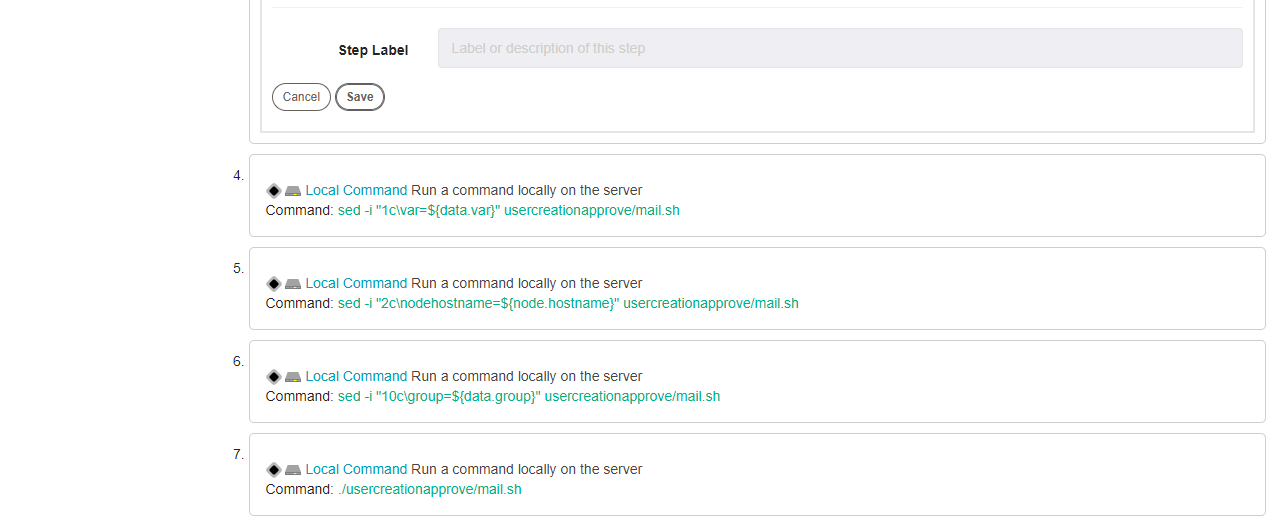
echo "Execution is not for this machine"

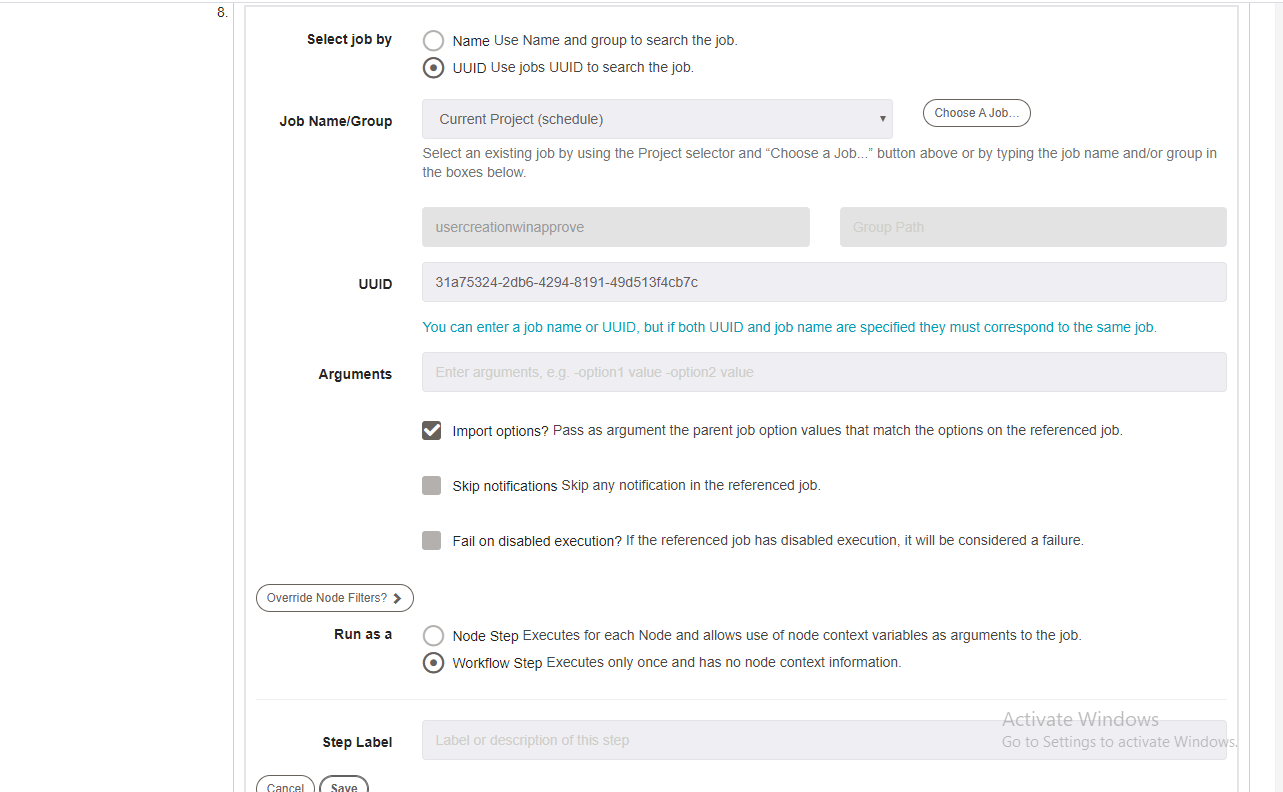
fi

fi

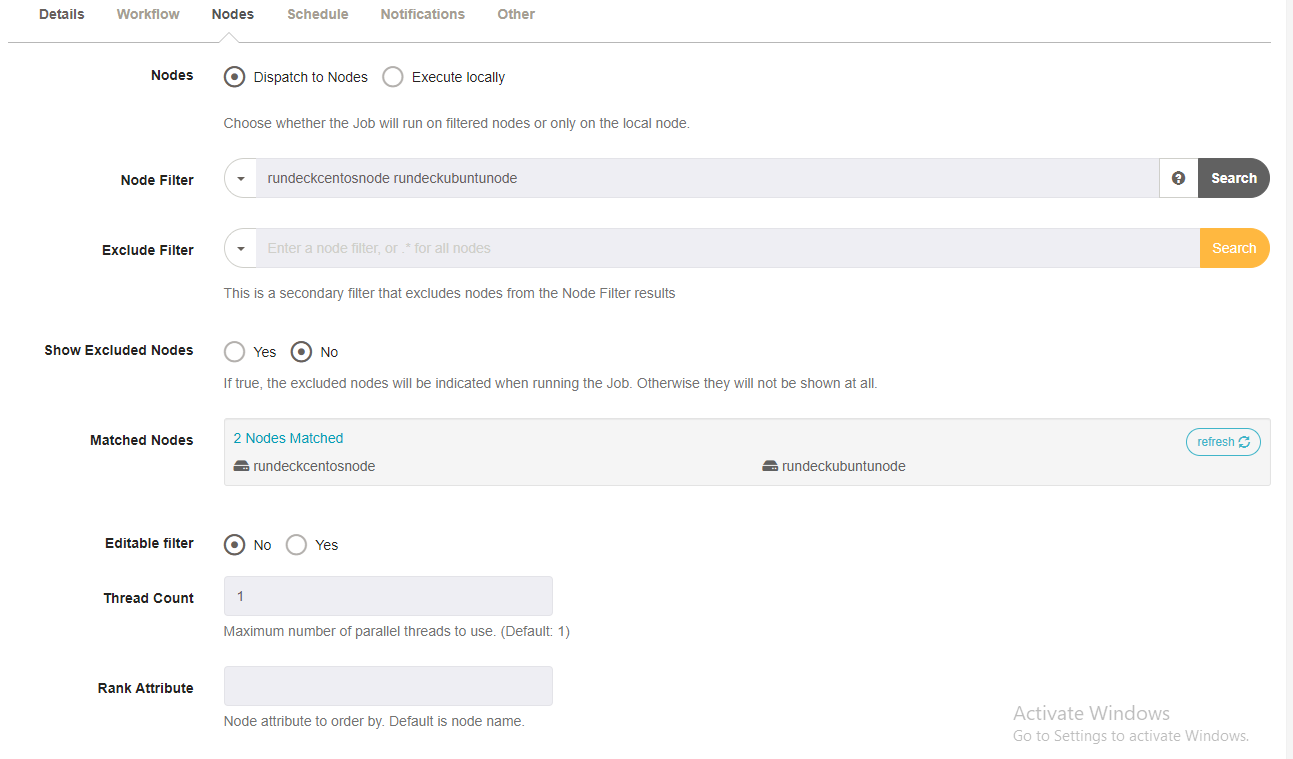
echo "var=$var"

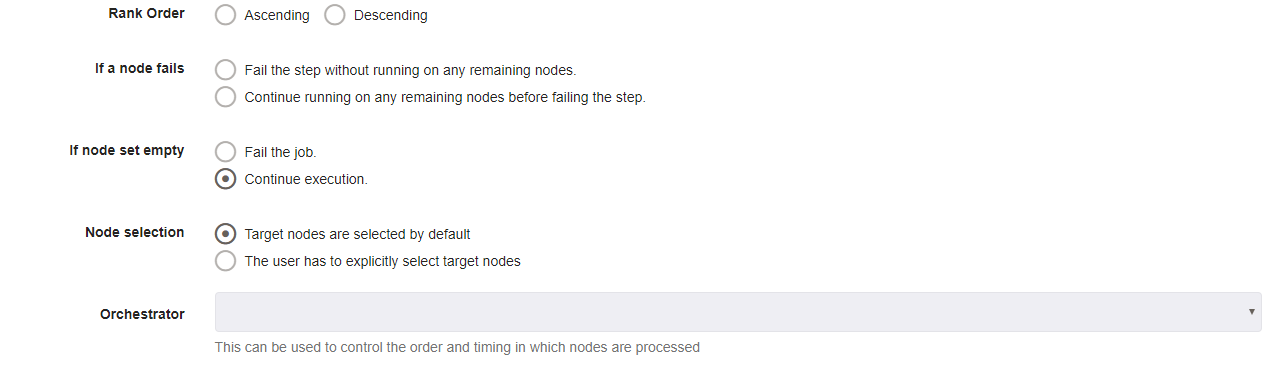
echo "group=$group"





Nodes,





Click create,

Open rundeck terminal,

[root@rundeck ~]# cd /var/lib/rundeck

[root@rundeck rundeck]# mkdir usercreationapprove

[root@rundeck rundeck]# cd usercreationapprove

[root@rundeck usercreationapprove]# cat usercreation.sh

issue=ADMINACCES-8

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Server" | tail -1)

user=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "UserName" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

pass=$(openssl rand -base64 8)

access=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Access" | tail -1)

status=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Status" | tail -1)

echo "server=$server"

echo "username=$user"

echo "mail=$mail"

echo "passwd=$pass"

echo "access=$access"

echo "status=$status"

sed -i "3c\$servername='$server'" usercreationapprove/winusercreation.ps1

sed -i "4c\$username='$user'" usercreationapprove/winusercreation.ps1

sed -i "5c\$passwd='$pass'" usercreationapprove/winusercreation.ps1

sed -i "6c\$acces='$access'" usercreationapprove/winusercreation.ps1

sed -i "7c\$statu='$status'" usercreationapprove/winusercreation.ps1

sed -i "3c\servername=$server" usercreationapprove/mail.sh

sed -i "4c\username=$user" usercreationapprove/mail.sh

sed -i "5c\passwd=$pass" usercreationapprove/mail.sh

sed -i "6c\mail=$mail" usercreationapprove/mail.sh

sed -i "7c\access=$access" usercreationapprove/mail.sh

sed -i "8c\issue=$issue" usercreationapprove/mail.sh

sed -i "9c\status=$status" usercreationapprove/mail.sh

[root@rundeck usercreationapprove]# cat winusercreation.ps1

$nodeostype='Windows'

$nodehostname='rundeckwindowsnode.zippyops.com'

$servername='rundeckwindowsnode.zippyops.com'

$username='mani'

$passwd='KP43v6bUiyM='

$acces='Normal'

$statu='Done'

if("$nodeostype" -eq "Windows"){

if("$nodehostname" -eq "$servername"){

if(("$acces" -eq "Admin") -and ("$statu" -eq "Approved")){

NET USER $username $passwd /ADD

$var=$(echo $?)

if("$var" -eq "True"){

NET LOCALGROUP Administrators $username /ADD

}else{

$s = $(net user $username | find "Local Group Memberships")

$p = 'Administrators'

$group=$($s -match $p)

if("$group" -eq "False"){

NET LOCALGROUP Administrators $username /ADD

}

}

}elseif(($acces -eq "Normal") -or ($statu -eq "Rejected")){

NET USER $username $passwd /ADD

$var=$(echo $?)

echo "pppZZ"

if("$var" -eq "False"){

echo "pppZZhgghkj"

$s = $(net user $username | find "Local Group Memberships")

$p = 'Administrators'

$group=$($s -match $p)

if("$group" -eq "True"){

NET LOCALGROUP Administrators $username /delete

}

}

}

}else{

echo "Excution is not for this machine"

}

}

echo "var=$var"

echo "group=$group"

[root@rundeck usercreationapprove]# cat mail.sh

var=False

nodehostname=rundeckwindowsnode.zippyops.com

servername=rundeckwindowsnode.zippyops.com

username=mani

passwd=KP43v6bUiyM=

mail=praveenkumar.m@zippyops.in

access=Normal

issue=ADMINACCES-8

status=Done

group=False

if [ $nodehostname == $servername ]; then

if [[ $var == 0 || $var == True ]]; then

if [[ $access == Admin && $status == Approved ]]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "New User $username created with the password $passwd with Admin access in the server $servername" | mail -s "New User Creation done" $mail

else

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "New User $username created with the password $passwd with Normal access in the server $servername" | mail -s "New User Creation done" $mail

fi

elif [[ $var == 9 || $var == False ]]; then

if [[ $access == Admin && $status == Approved && $group == $username || $access == Admin && $status == Approved && $group == False ]]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "$username user already exists in the server $servername and now the user has given Admin access" | mail -s "New User Creation done" $mail

elif [[ $status == Rejected && $group == wheel || $access == Normal && $group == wheel || $status == Rejected && $group == sudo || $access == Normal && $group == sudo || $status == Rejected && $group == True || $access == Normal && $group == True ]]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "$username user already exists in the server $servername and now the user has given Normal access" | mail -s "New User Creation done" $mail

else

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "$username user already exists in the server $servername" | mail -s "New User Creation not done" $mail

fi

fi

else

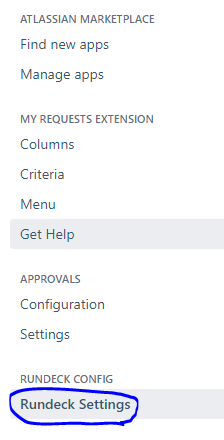
echo "Execution is not for this machine"

fi

Invoke Rundeck job setting, click the Manage apps

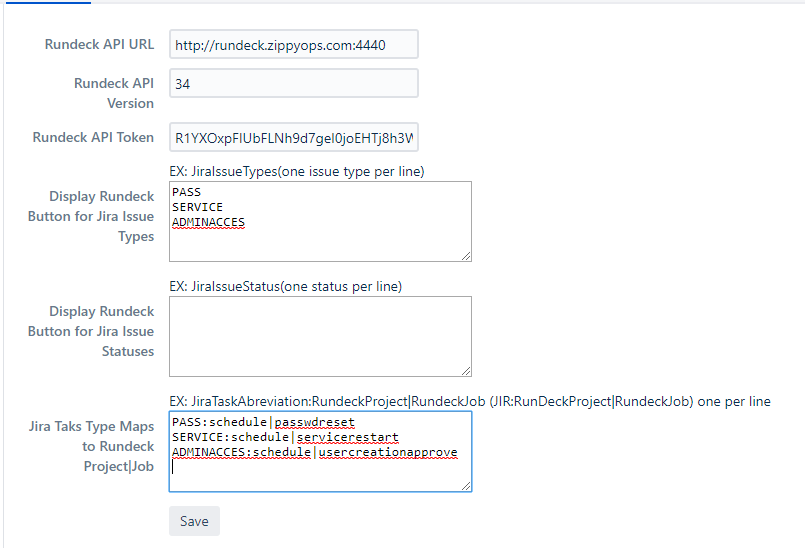


Click Rundeck setting,



First mention Rundeck API URl and then API verion

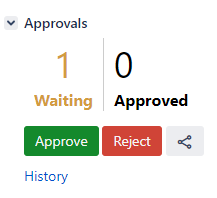
Then generate the Api Token and paste here.



Save,

Come back to issue,

Click on Approve or Reject.

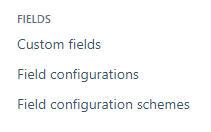


# Logfilecopy

Creating custom field in jira, click setting and in that click issue,

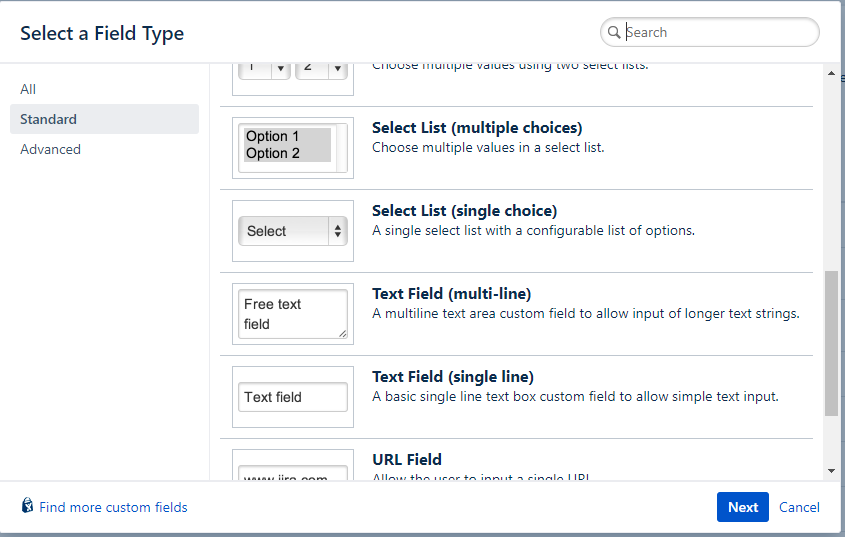


Click on Custom fields,

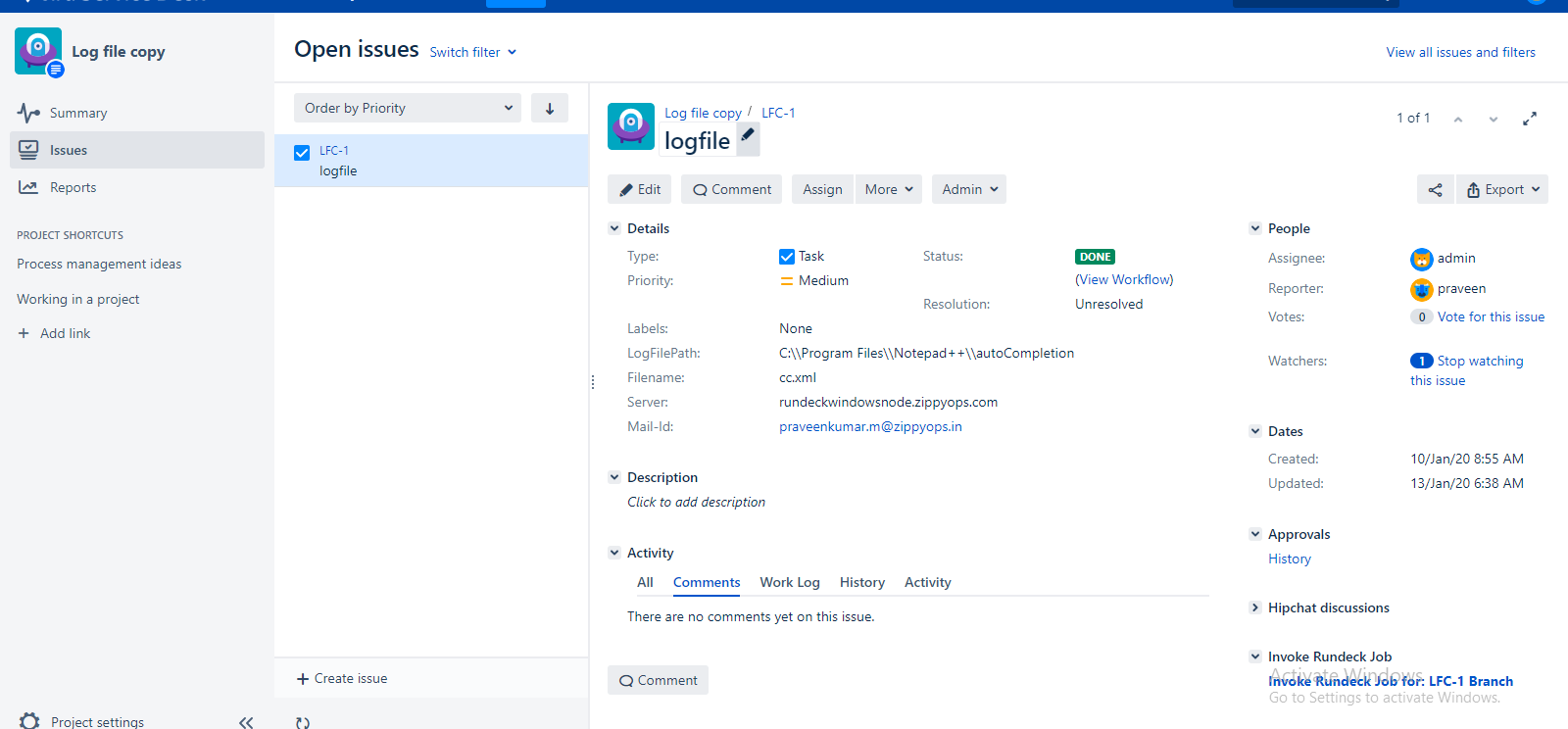


In that click Add Custom field in the right corner, In that create 3 field

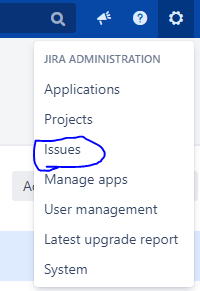
1. Filename field in Text field (Single line)
2. Mail\_Id field in Text field (Single line)
3. Server field in Select List (Multiple choice)
4. LogFilePath field in Text field (Single line)



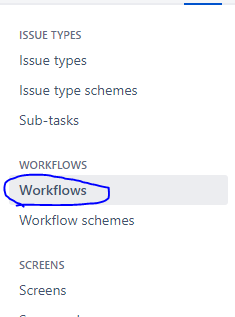
Create the new project with the name Log file copy and issue id as LFC.



Create the workflow, click on issue,

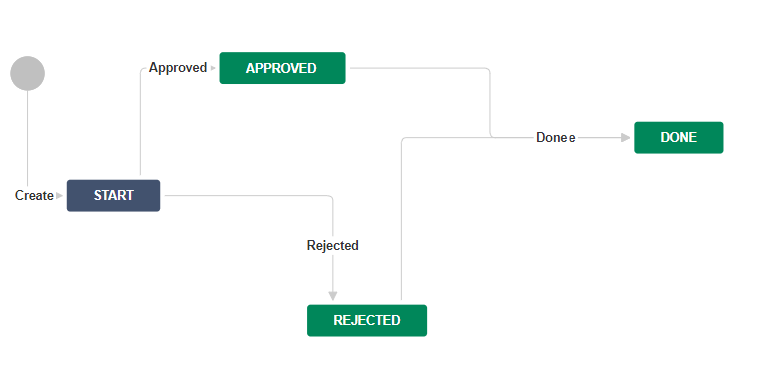


Click on workflow, there click on edit in which project you have to make change in workflow,

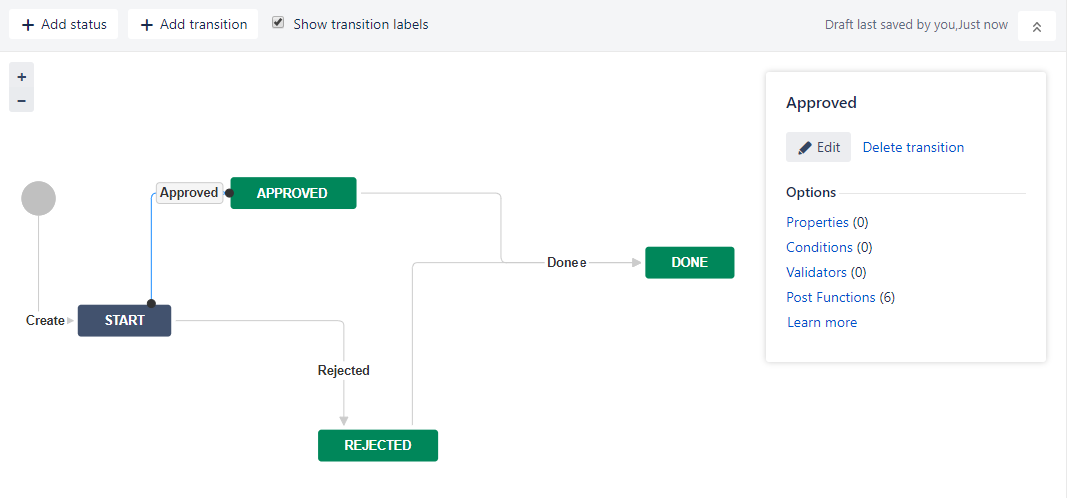




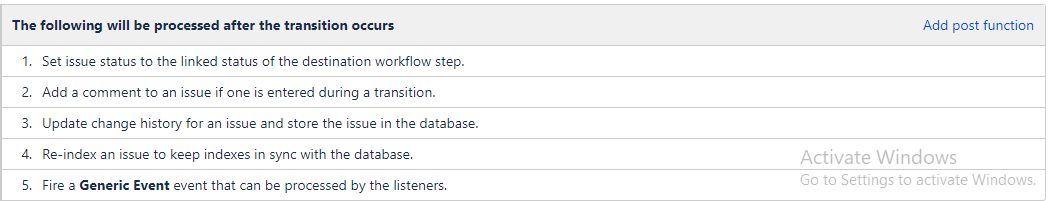
Create the workflow like this,



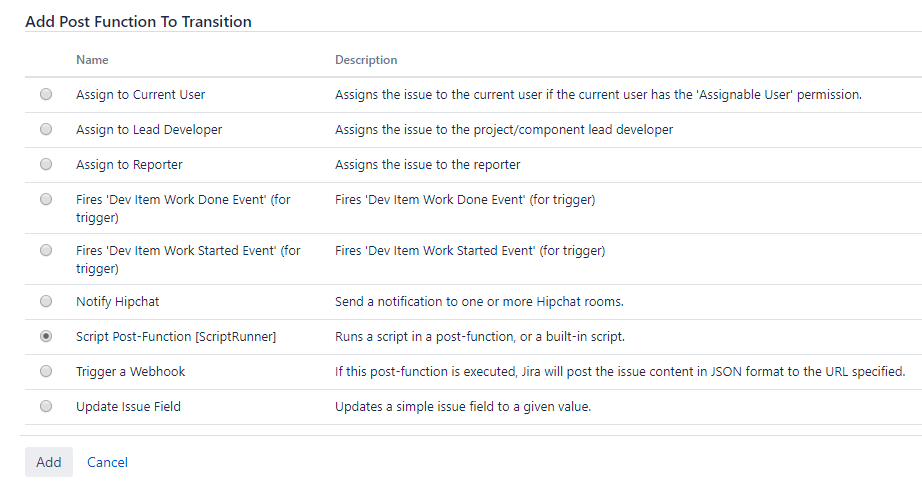
Click on the approve transition, the new dialog box will open there click post function,



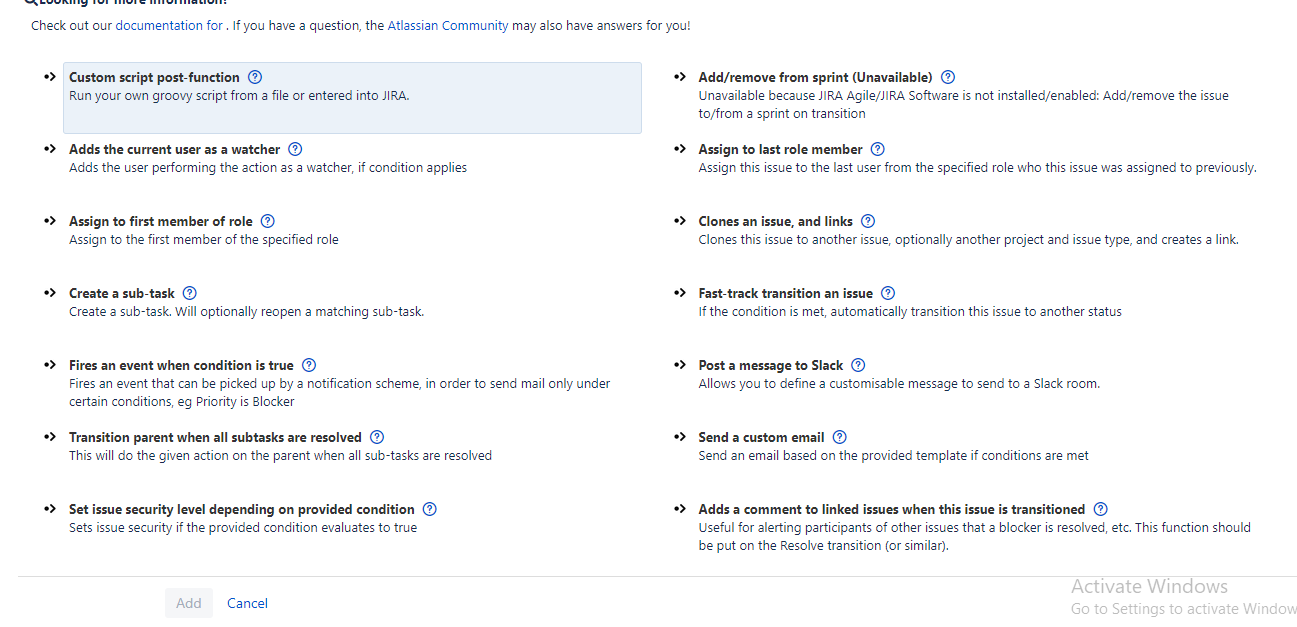
Click on add post function



Click on script post-function and click Add.



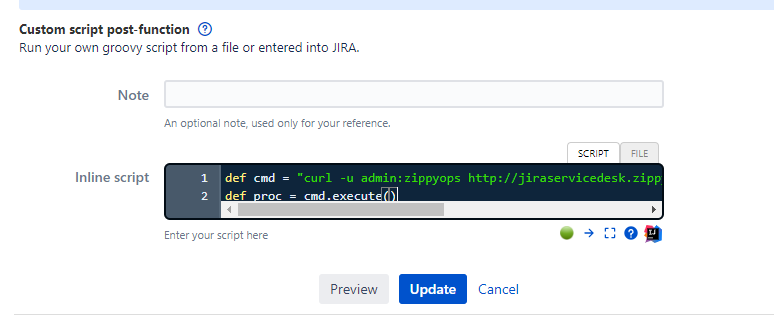
Click Custom script post function,



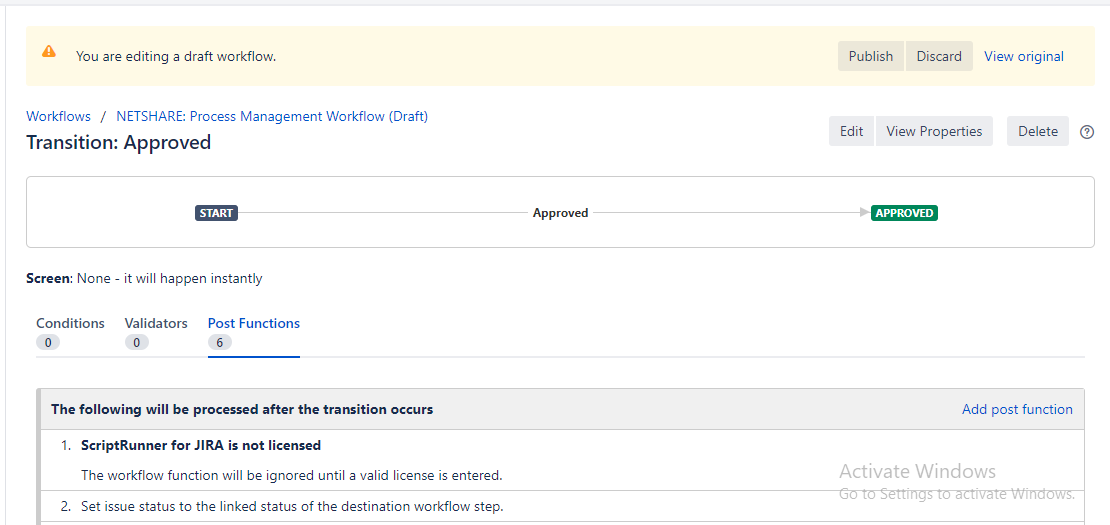
Type the script and Click on update,

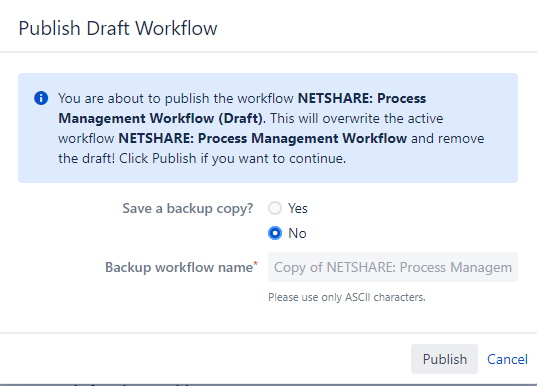
def cmd = "curl -u admin:zippyops http://jiraservicedesk.zippyops.com:8080/plugins/servlet/rundeck?issueKey=${issue.key}"

def proc = cmd.execute()



Click on publish,

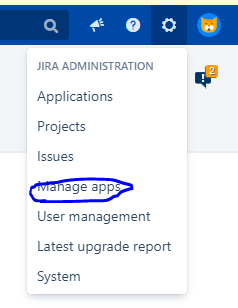




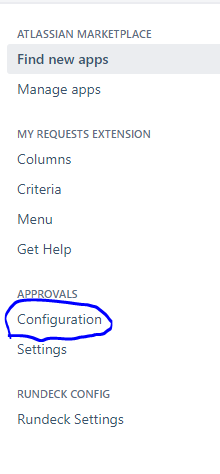
Like this do for the rejection transition,

Approval setup,

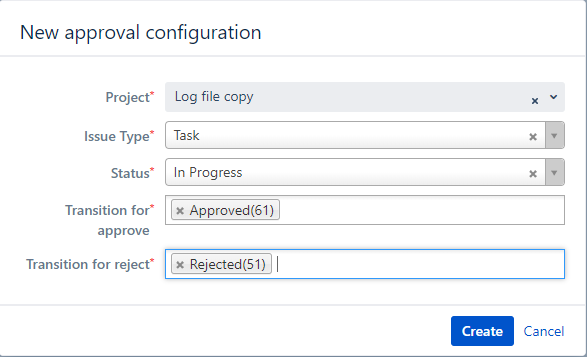
Go to settings, and click manage apps,



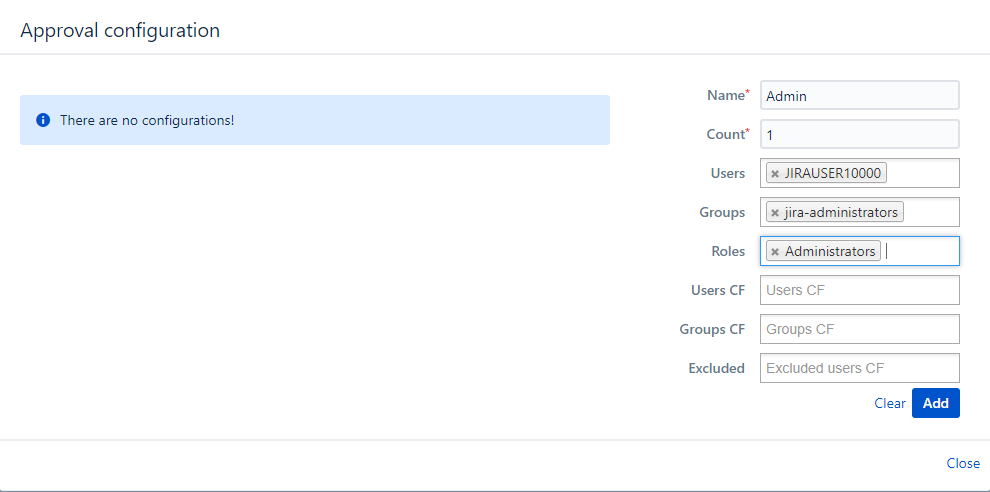
Click configuration below approval, if it not available install the plugins.



Click ADD, then click create



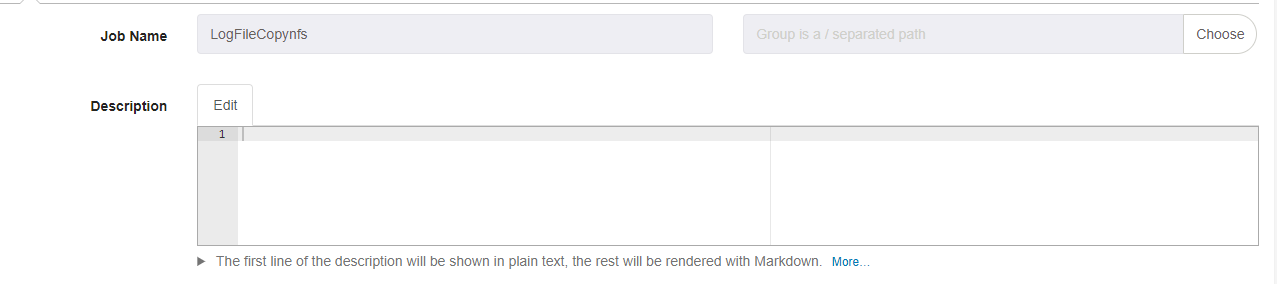
After creating click configuration, and click add



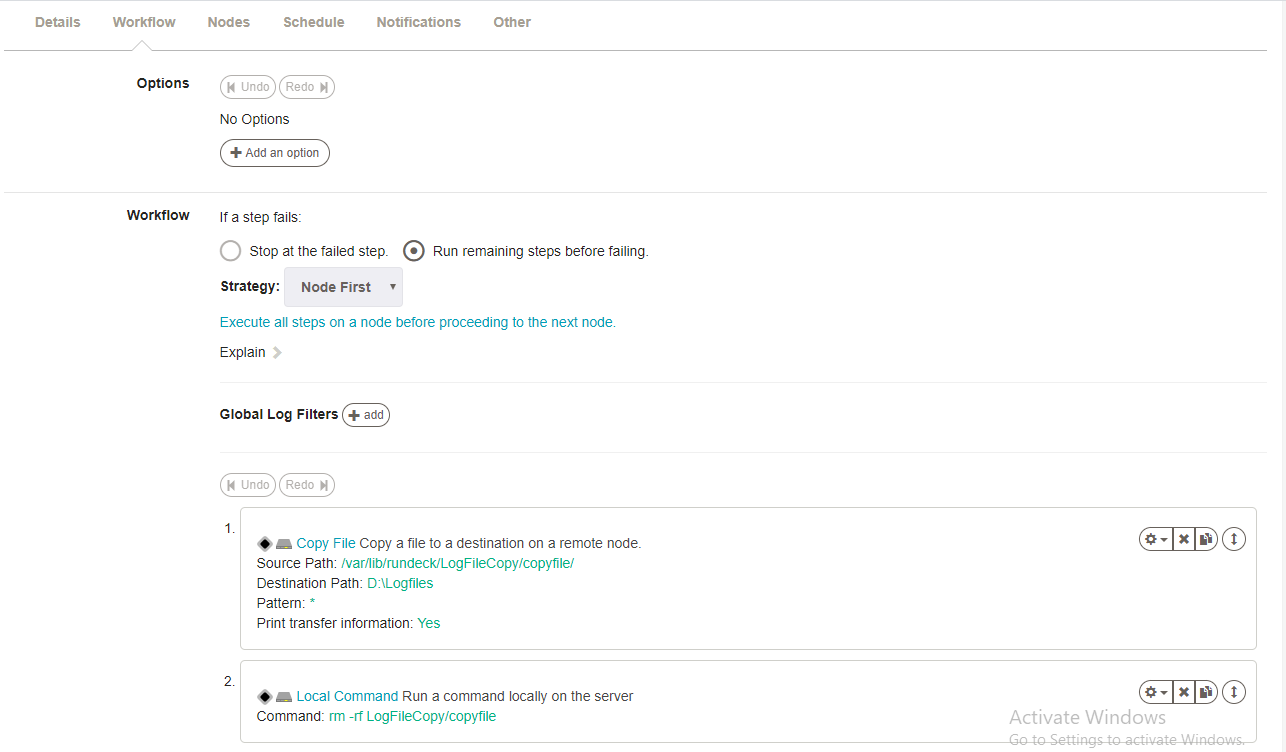
Create the three job for Log file copy Approve in rundeck,

One for linux and one for windows and another for NFS share.

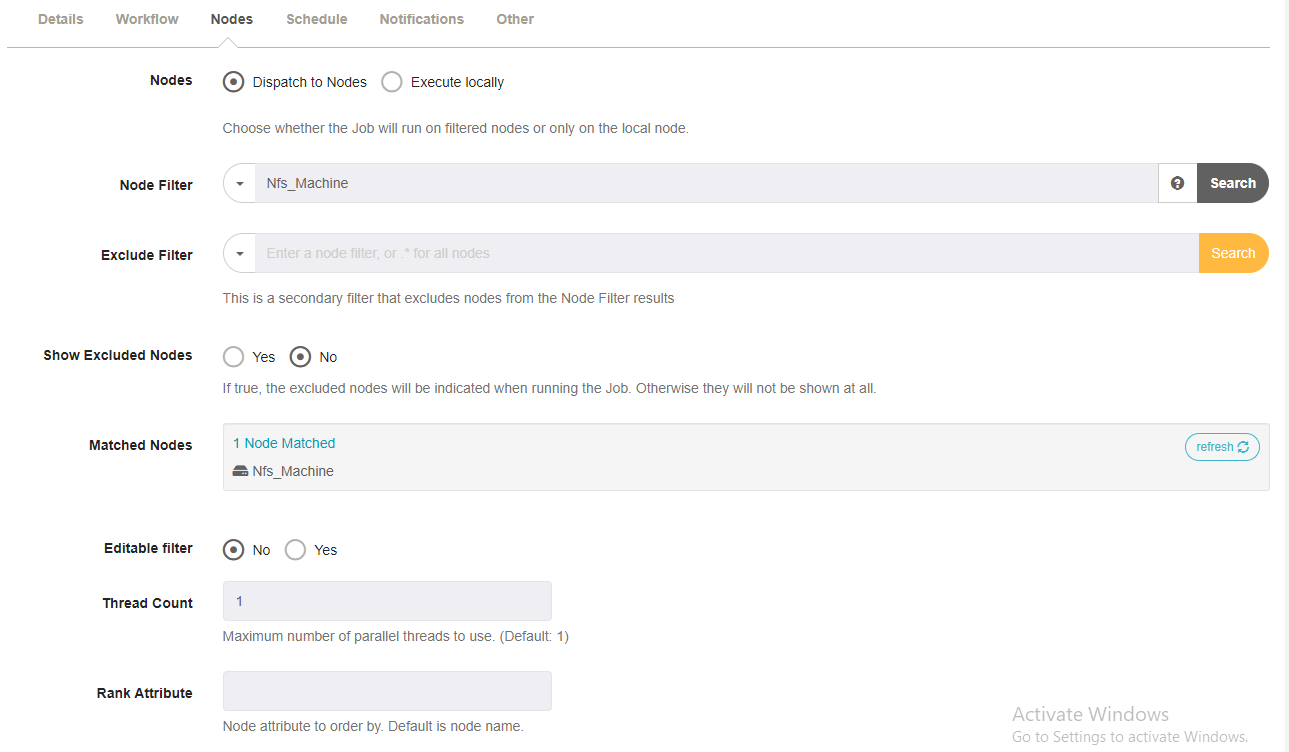
First we going to create jobs for NFS share,

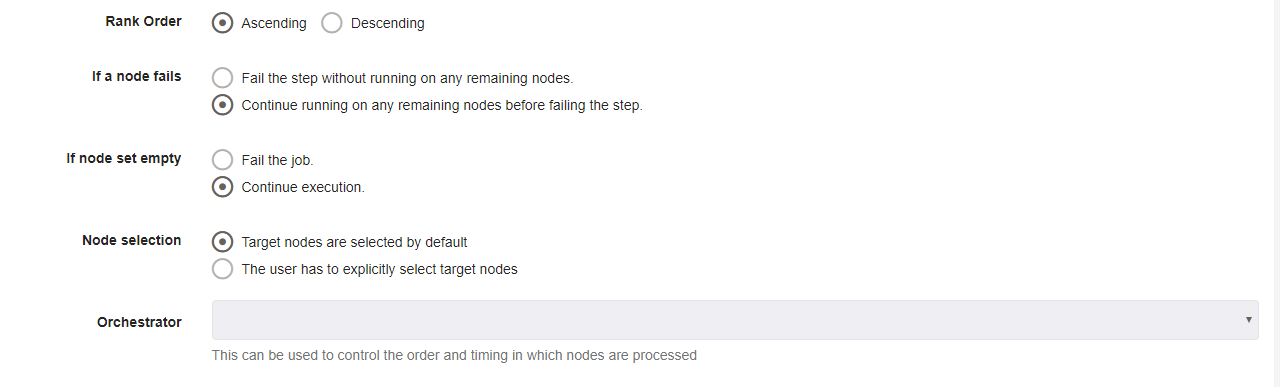


Workflows,



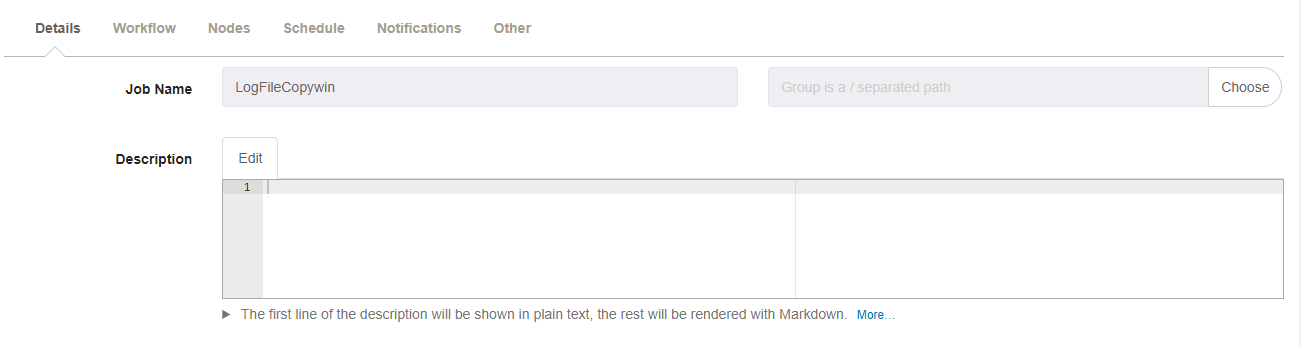
Nodes,





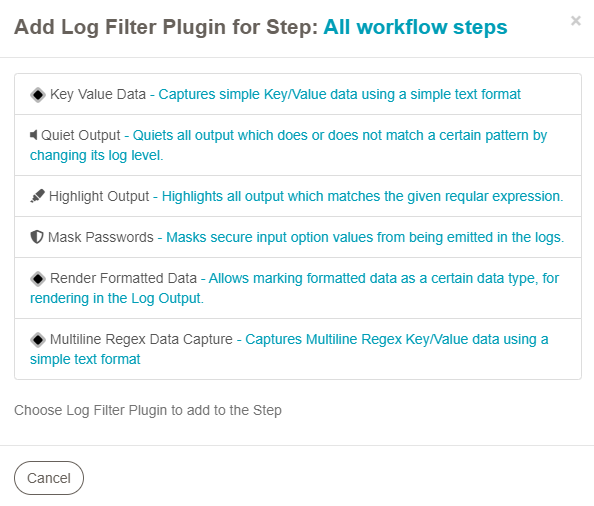
Click create,

Now for windows,

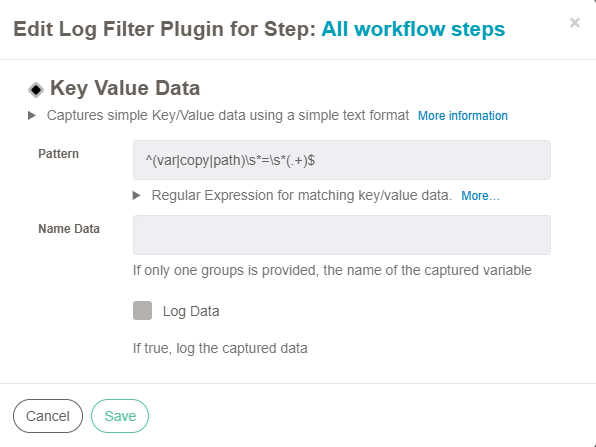


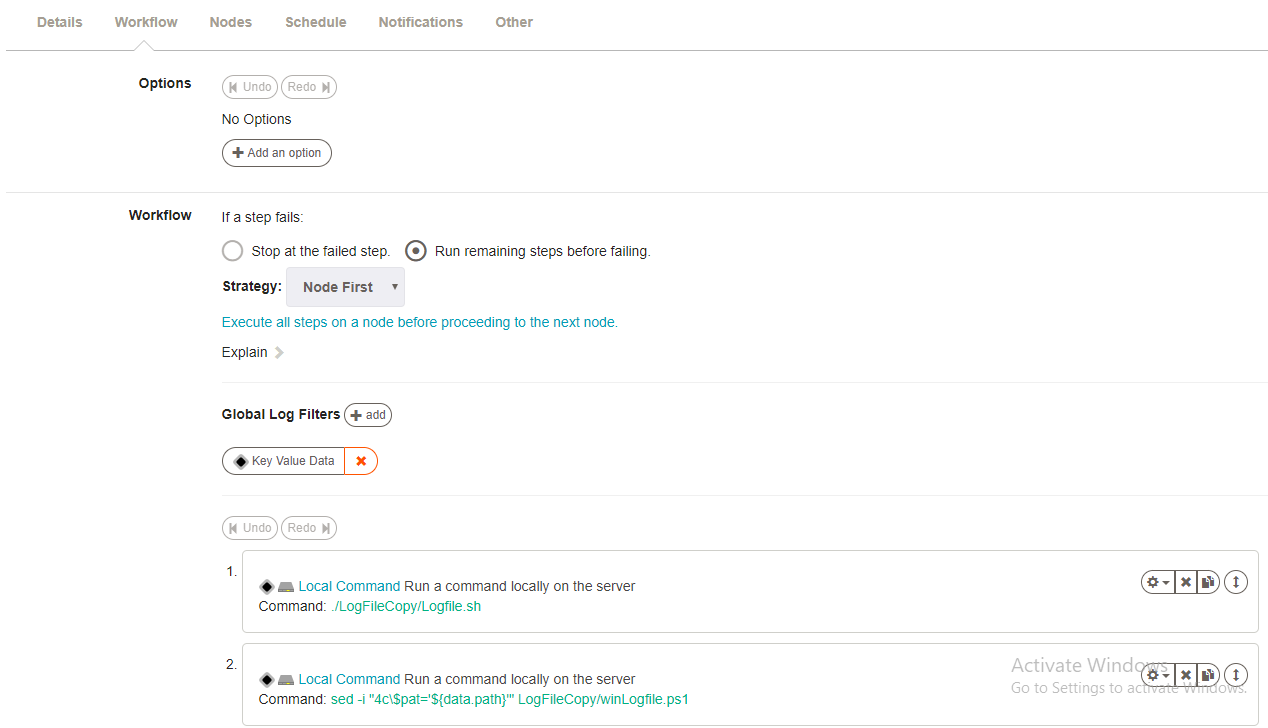
Workflow,

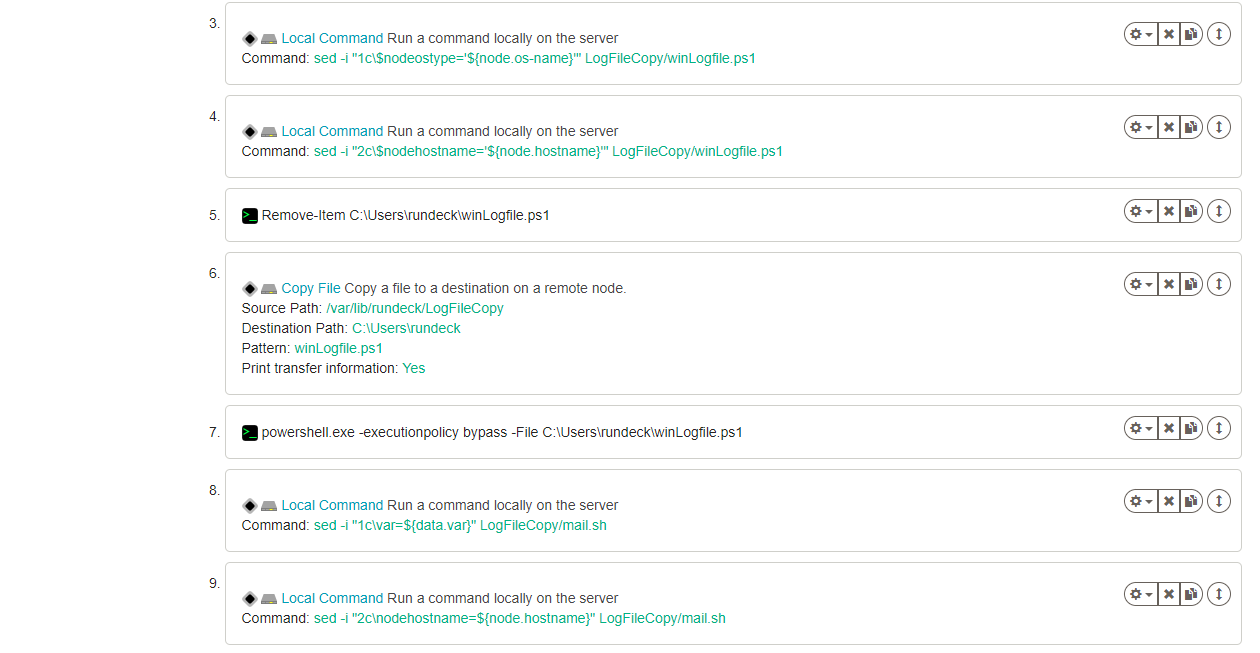
Click on Add Log Filter, and in that click Key Value Data.



Give the pattern like this, and save.

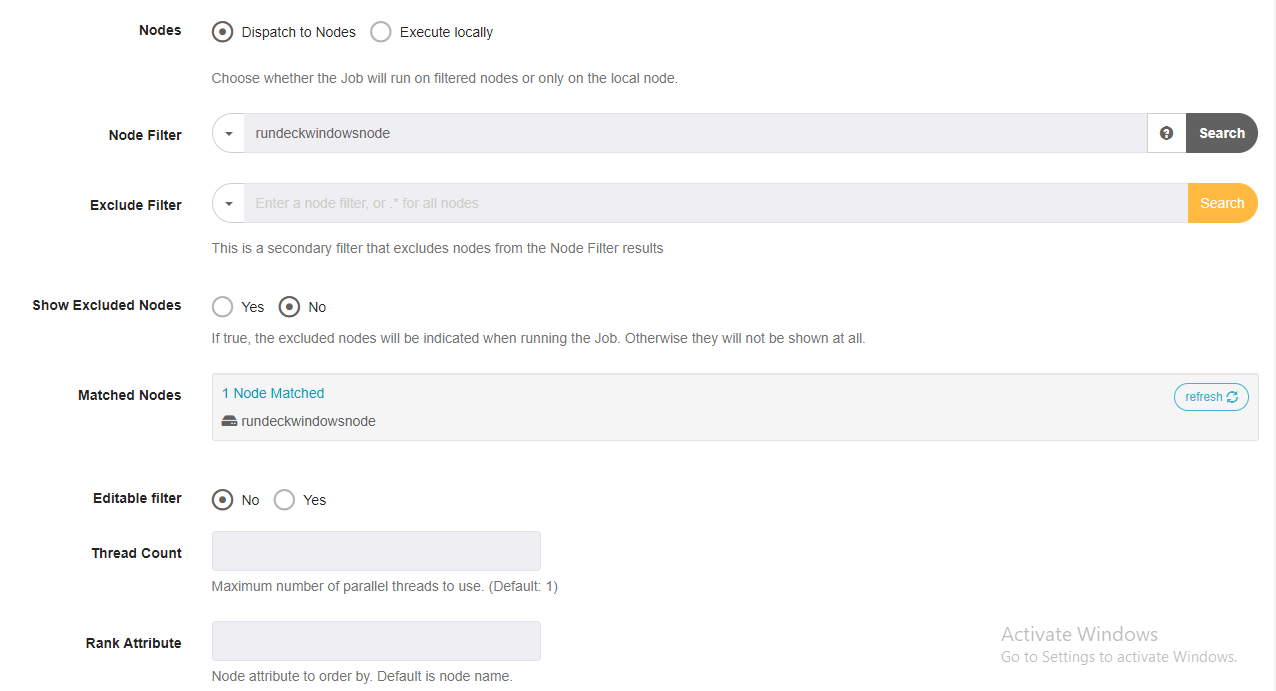


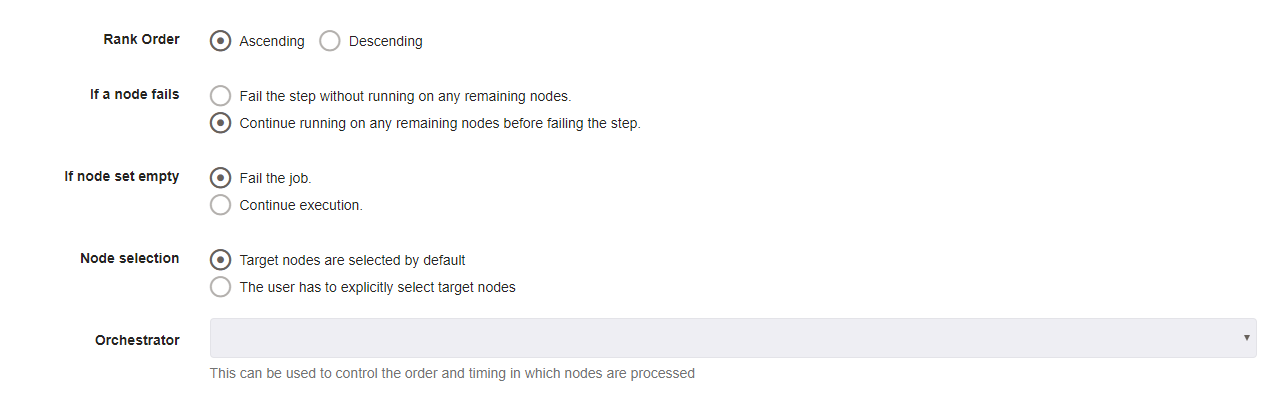






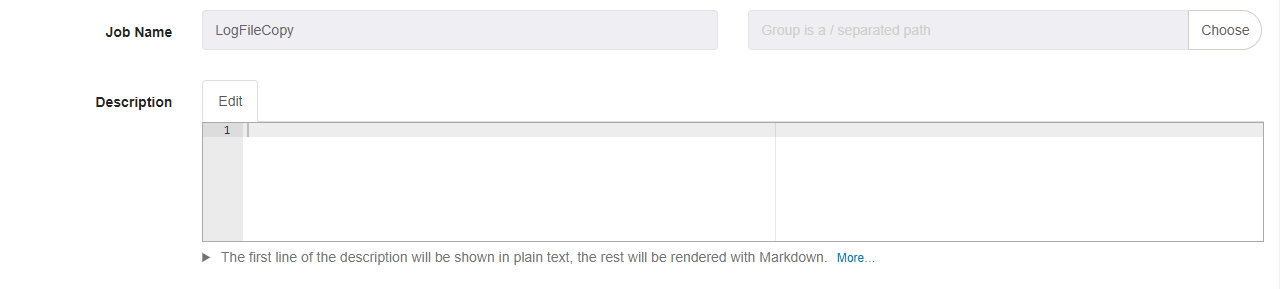
Nodes,





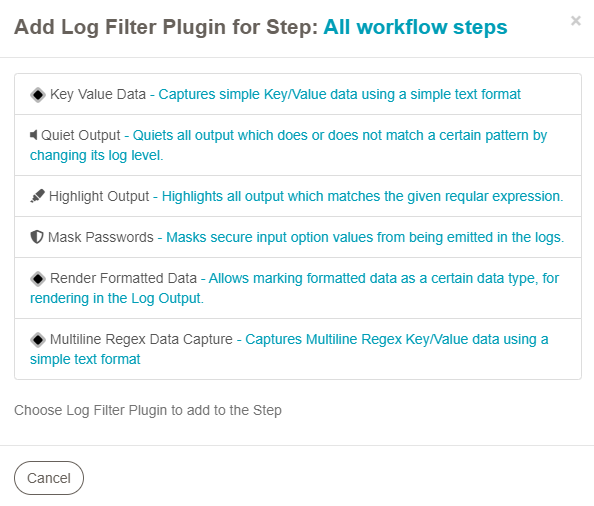
Click create.

Now for linux,

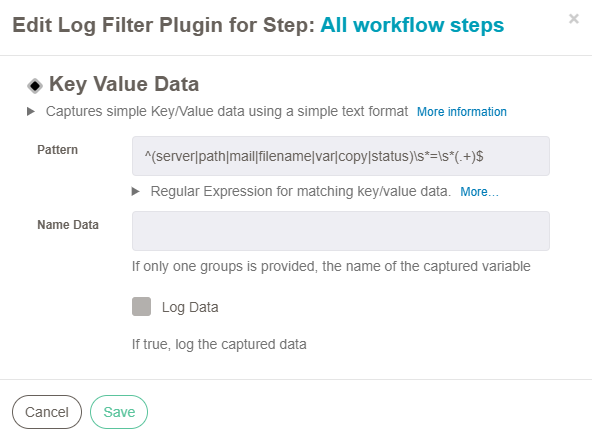


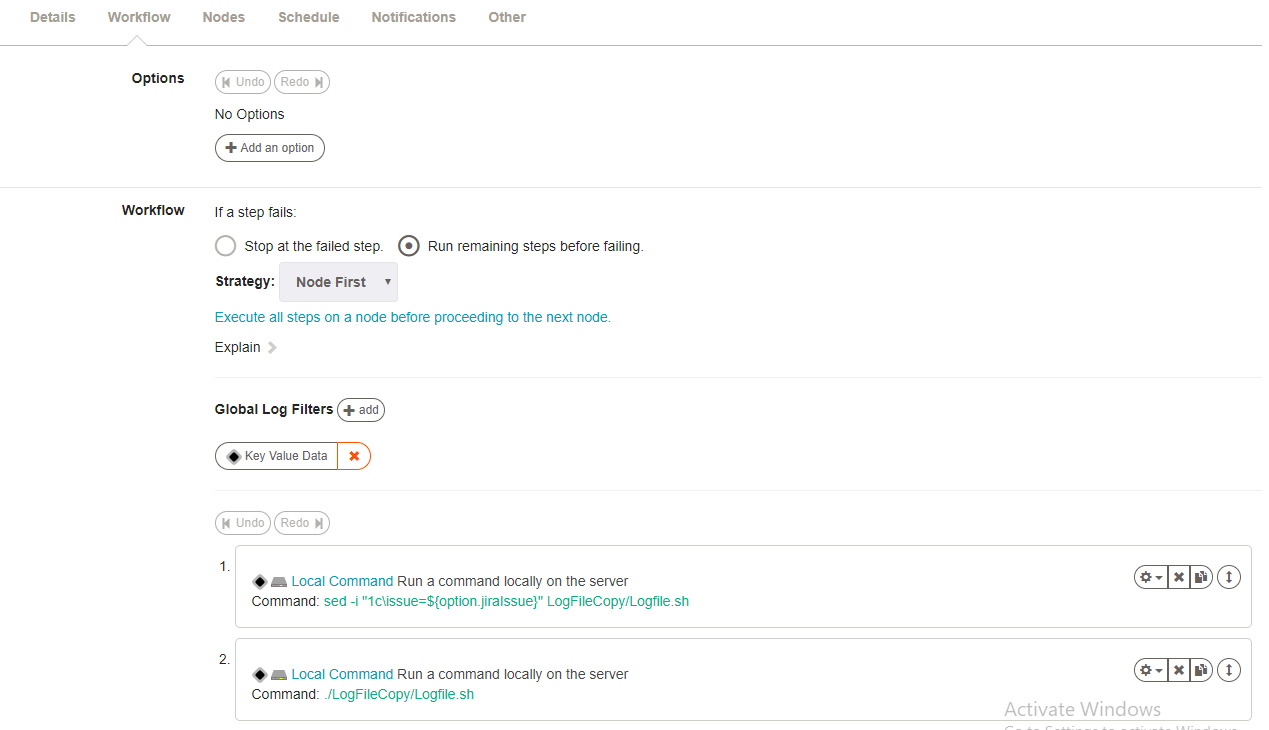
Workflow,

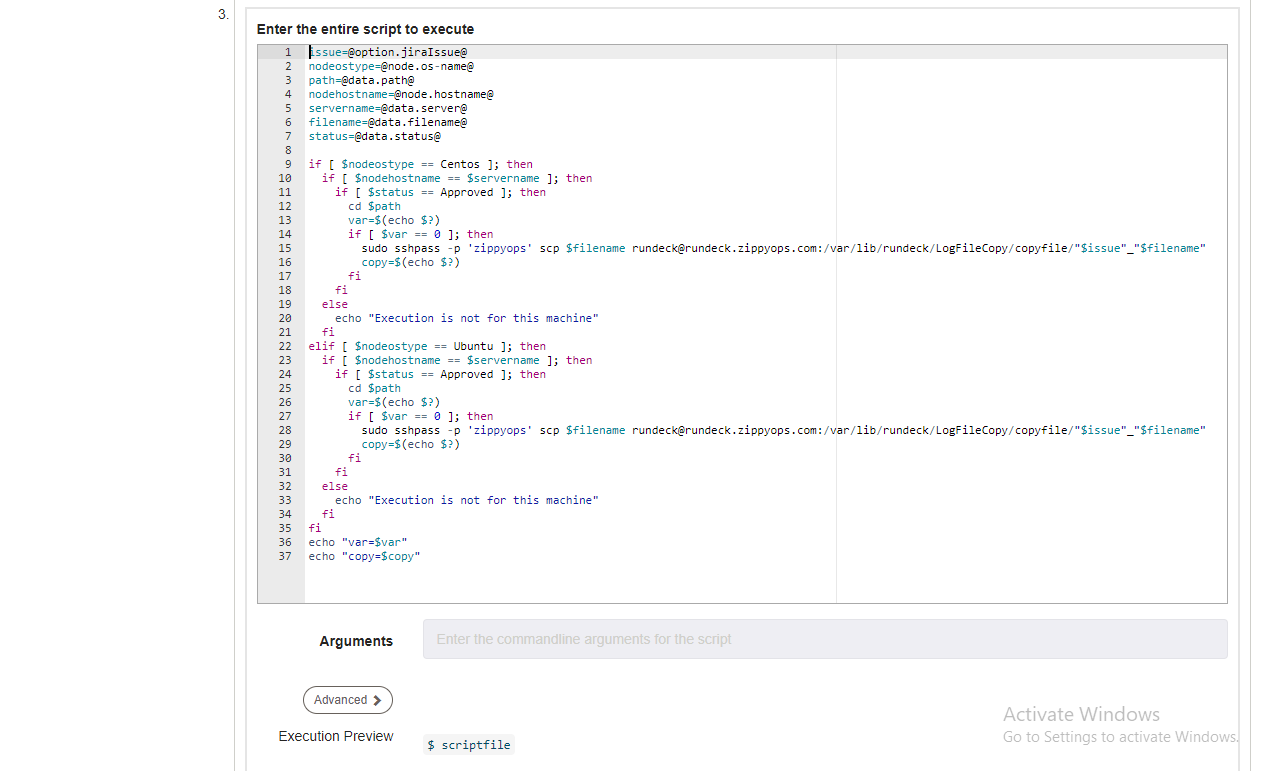
Click on Add Log Filter, and in that click Key Value Data.



Give the pattern like this, and save.







issue=@option.jiraIssue@

nodeostype=@node.os-name@

path=@data.path@

nodehostname=@node.hostname@

servername=@data.server@

filename=@data.filename@

status=@data.status@

if [ $nodeostype == Centos ]; then

if [ $nodehostname == $servername ]; then

if [ $status == Approved ]; then

cd $path

var=$(echo $?)

if [ $var == 0 ]; then

sudo sshpass -p 'zippyops' scp $filename rundeck@rundeck.zippyops.com:/var/lib/rundeck/LogFileCopy/copyfile/"$issue"\_"$filename"

copy=$(echo $?)

fi

fi

else

echo "Execution is not for this machine"

fi

elif [ $nodeostype == Ubuntu ]; then

if [ $nodehostname == $servername ]; then

if [ $status == Approved ]; then

cd $path

var=$(echo $?)

if [ $var == 0 ]; then

sudo sshpass -p 'zippyops' scp $filename rundeck@rundeck.zippyops.com:/var/lib/rundeck/LogFileCopy/copyfile/"$issue"\_"$filename"

copy=$(echo $?)

fi

fi

else

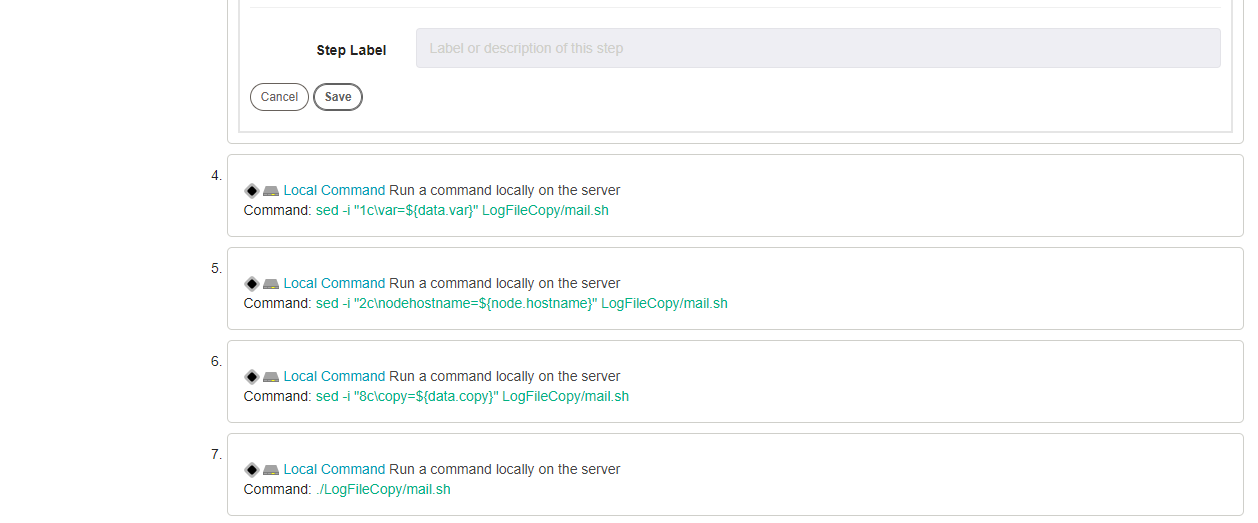
echo "Execution is not for this machine"

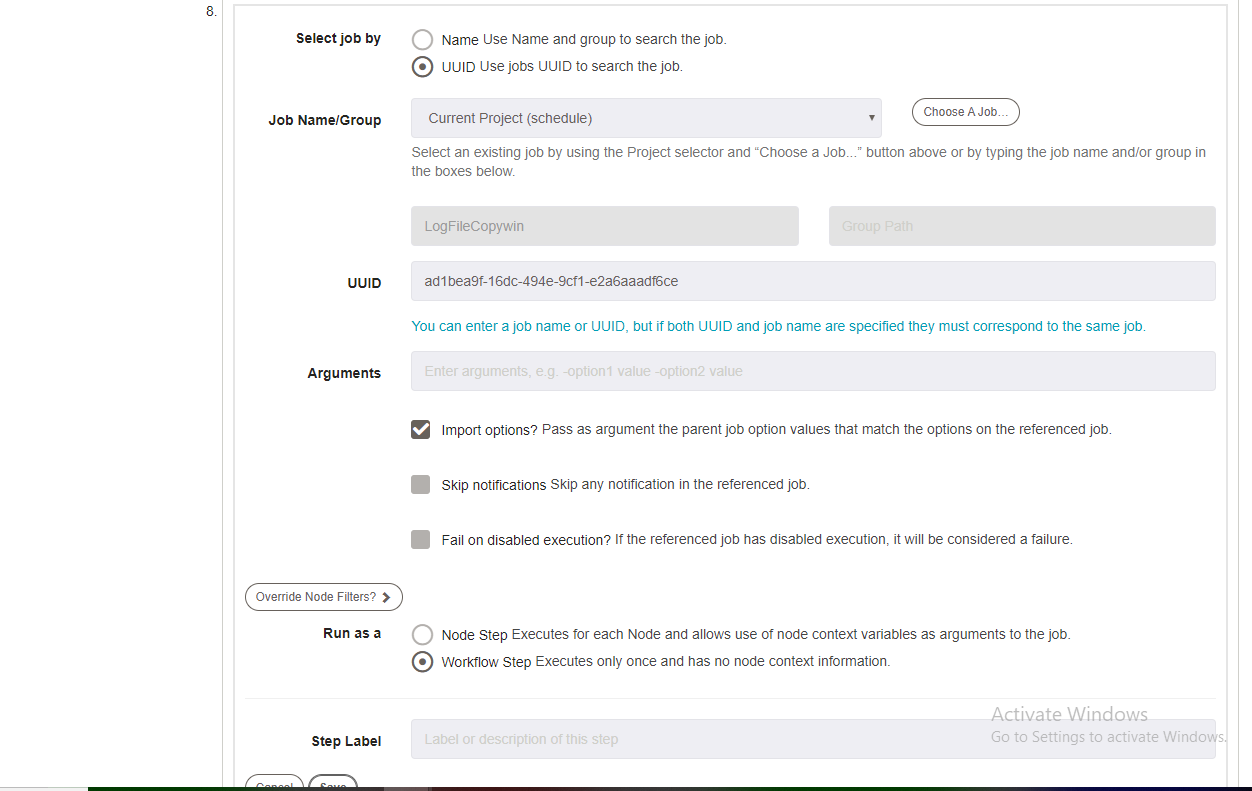
fi

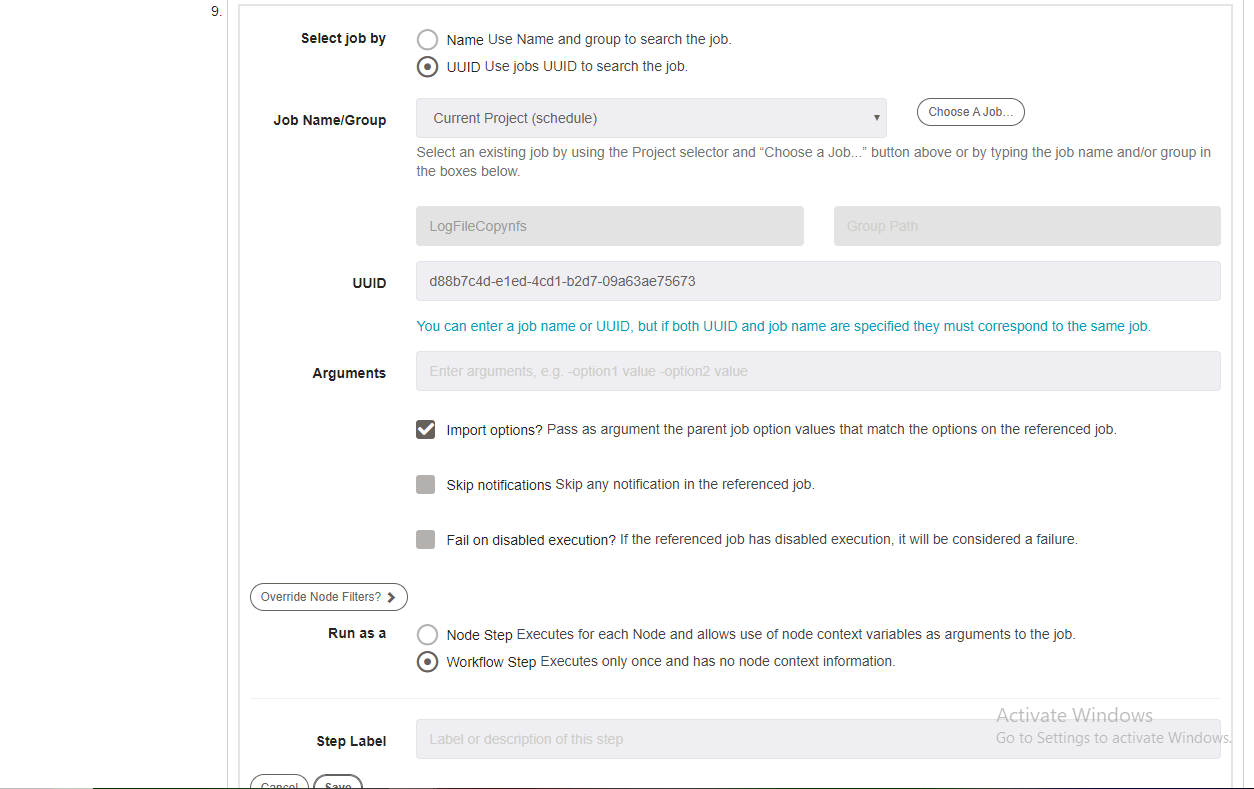
fi

echo "var=$var"

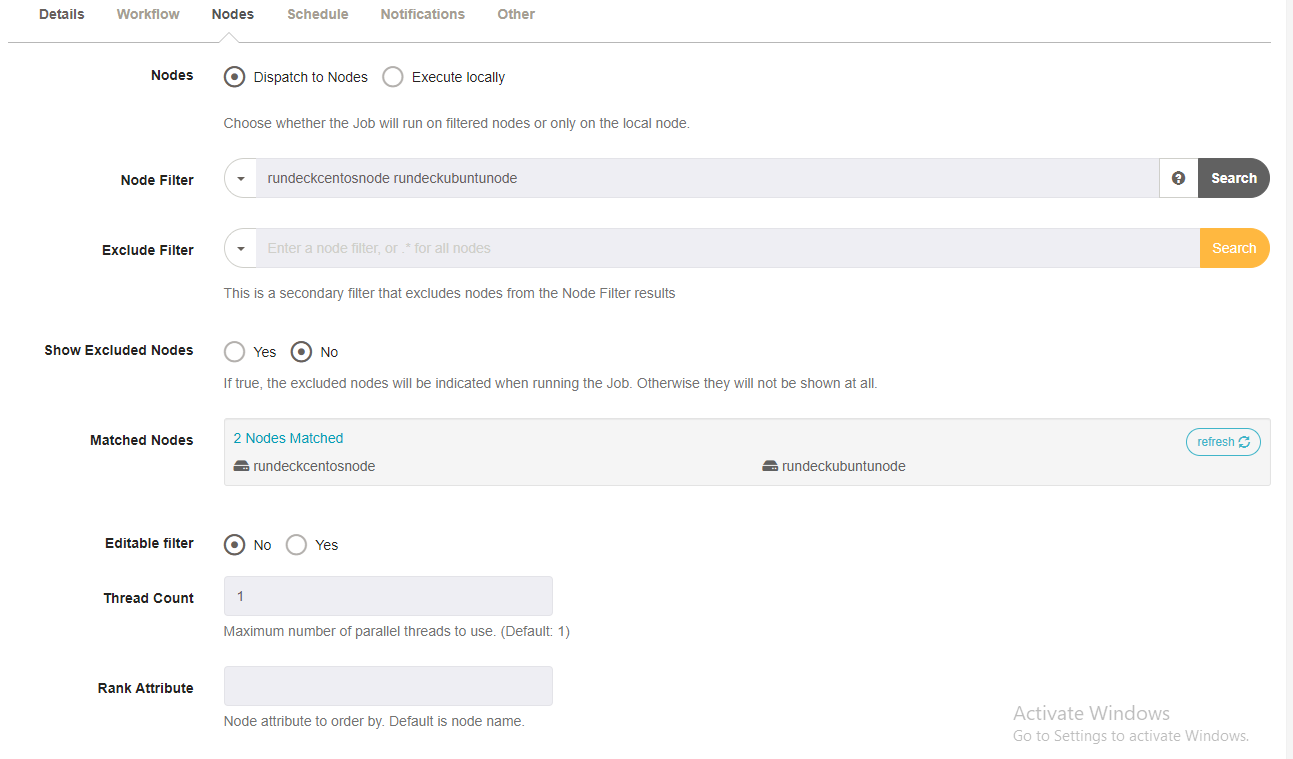
echo "copy=$copy"

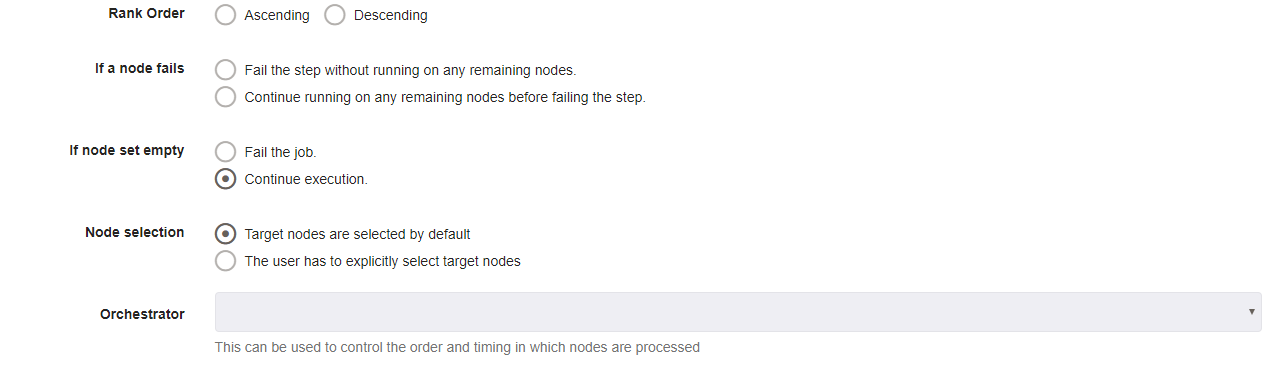






Nodes,





Click create,

Open rundeck terminal,

[root@rundeck ~]# cd /var/lib/rundeck/

[root@rundeck rundeck]# mkdir LogFileCopy/

[root@rundeck rundeck]# cd LogFileCopy/

[root@rundeck LogFileCopy]# cat Logfile.sh

issue=LFC-1

path=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "LogFilePath" | tail -1)

filename=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Filename" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

status=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Status" | tail -1)

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Server" | tail -1)

echo "path=$path"

echo "filename=$filename"

echo "status=$status"

echo "server=$server"

sed -i "3c\$servername='$server'" LogFileCopy/winLogfile.ps1

sed -i "4c\$pat='$path'" LogFileCopy/winLogfile.ps1

sed -i "5c\$filenam='$filename'" LogFileCopy/winLogfile.ps1

sed -i "6c\$statu='$status'" LogFileCopy/winLogfile.ps1

sed -i "7c\$issu='$issue'" LogFileCopy/winLogfile.ps1

sed -i "2c\path=$path" LogFileCopy/mail.sh

sed -i "3c\servername=$server" LogFileCopy/mail.sh

sed -i "4c\status=$status" LogFileCopy/mail.sh

sed -i "5c\filename=$filename" LogFileCopy/mail.sh

sed -i "6c\mail=$mail" LogFileCopy/mail.sh

sed -i "7c\issue=$issue" LogFileCopy/mail.sh

mkdir LogFileCopy/copyfile || true

[root@rundeck LogFileCopy]# cat winLogfile.ps1

$nodeostype='Windows'

$nodehostname='rundeckwindowsnode.zippyops.com'

$servername='rundeckwindowsnode.zippyops.com'

$pat='C:\Program Files\Notepad++\autoCompletion'

$filenam='cc.xml'

$statu='Approved'

$issu='LFC-1'

if("$nodeostype" -eq "Windows"){

if("$nodehostname" -eq "$servername"){

if("$statu" -eq "Approved"){

cd "$pat"

$var=$(echo $?)

if("$var" -eq "True"){

pscp -pw zippyops $filenam rundeck@rundeck.zippyops.com:/var/lib/rundeck/LogFileCopy/copyfile/"$issu"\_"$filenam"

$copy=$(echo $?)

}

}

}else{

echo "Excution is not for this machine"

}

}

echo "var=$var"

echo "copy=$copy"

[root@rundeck LogFileCopy]# cat mail.sh

var=True

nodehostname=rundeckwindowsnode.zippyops.com

servername=rundeckwindowsnode.zippyops.com

status=Approved

filename=cc.xml

mail=praveenkumar.m@zippyops.in

issue=LFC-1

copy=False

if [ $status == Approved ]; then

if [[ $var == 0 && $copy == 0 || $var == True && $copy == True ]]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "Log file copy is done and transfer to Nfs server" | mail -s "Log file copy done" $mail

elif [[ $var == 0 && $copy == 1 || $var == True && $copy == False ]]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "File not found in the specfied path in the specfied server $servername" | mail -s "Log file copy not done" $mail

elif [[ $var == 1 || $var == False ]]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "Path not found in the specfied server $servername" | mail -s "Log file copy not done" $mail

fi

elif [ $status == Rejected ]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "Log file copy is not Approved for you" | mail -s "Log file copy not done" $mail

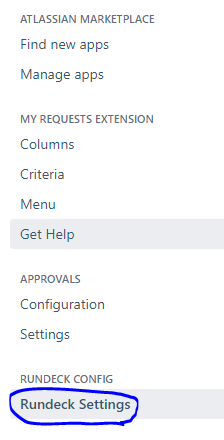
fi

[root@rundeck LogFileCopy]#

Invoke Rundeck job setting, click the Manage apps

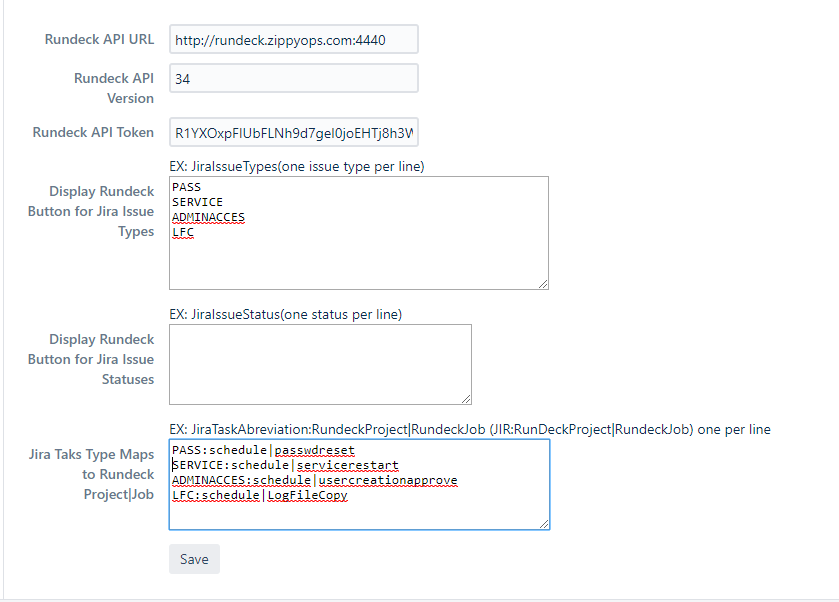


Click Rundeck setting,



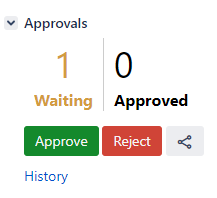
First mention Rundeck API URl and then API verion

Then generate the Api Token and paste here.



Come back to issue,

Click on Approve or Reject.

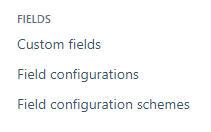


# Mysql Dump

Creating custom field in jira, click setting and in that click issue,

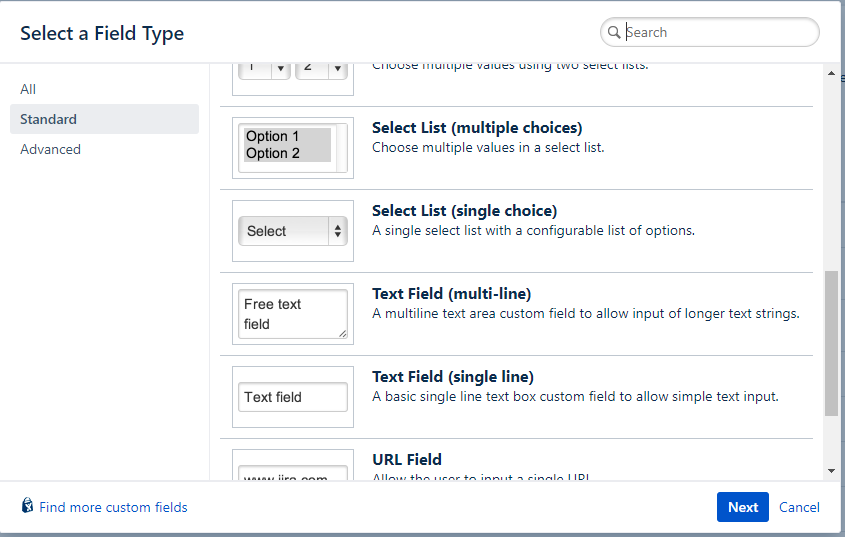


Click on Custom fields,

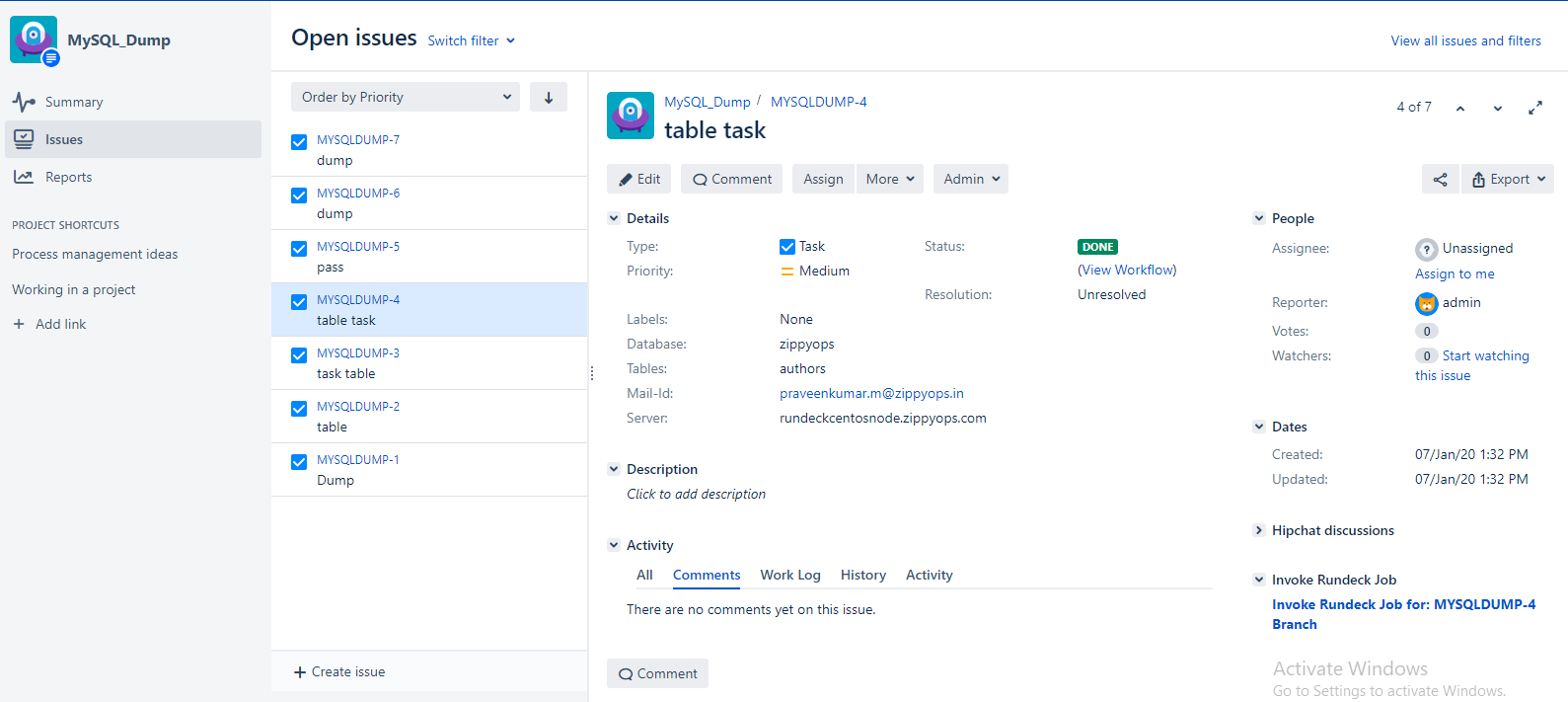


In that click Add Custom field in the right corner, In that create 3 field

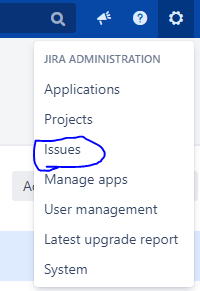
1. Tables field in Select List (Multiple choice)
2. Mail\_Id field in Text field (Single line)
3. Server field in Select List (Multiple choice)
4. Database field in Radio Button



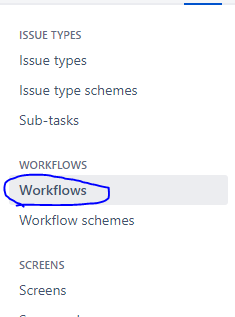
Create the new project with the name Mysql Dump and issue id as MYSQLDUMP.



Create the workflow, click on issue,

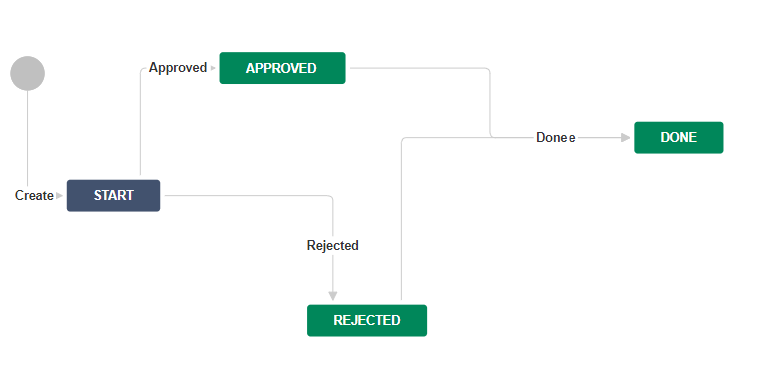


Click on workflow, there click on edit in which project you have to make change in workflow,

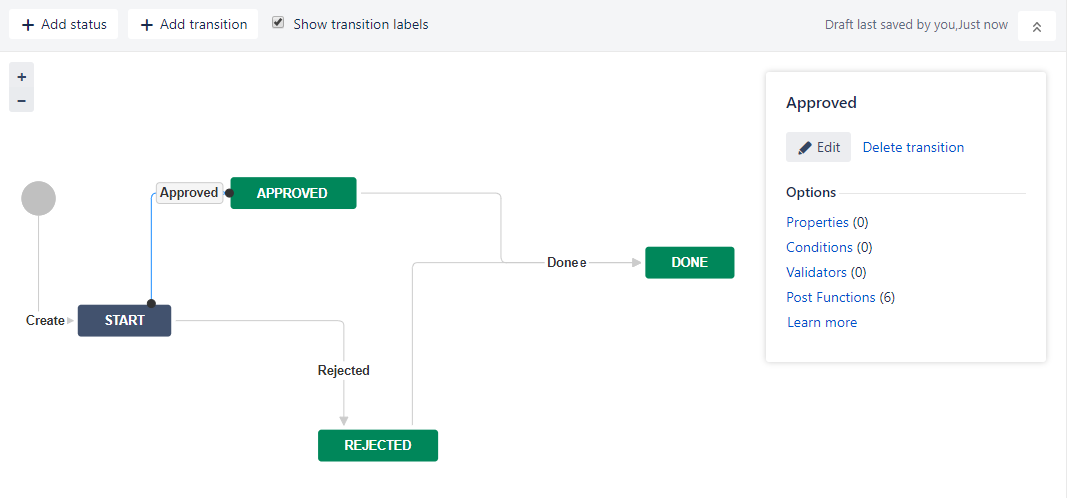




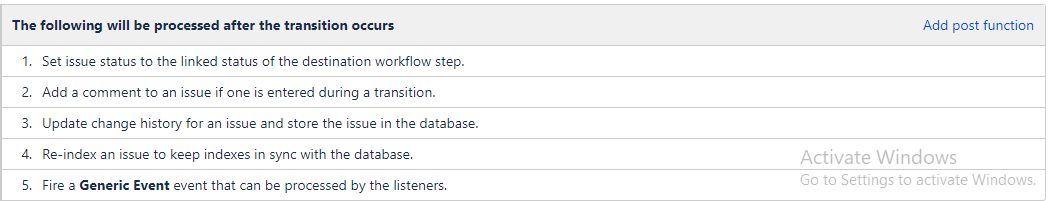
Create the workflow like this,



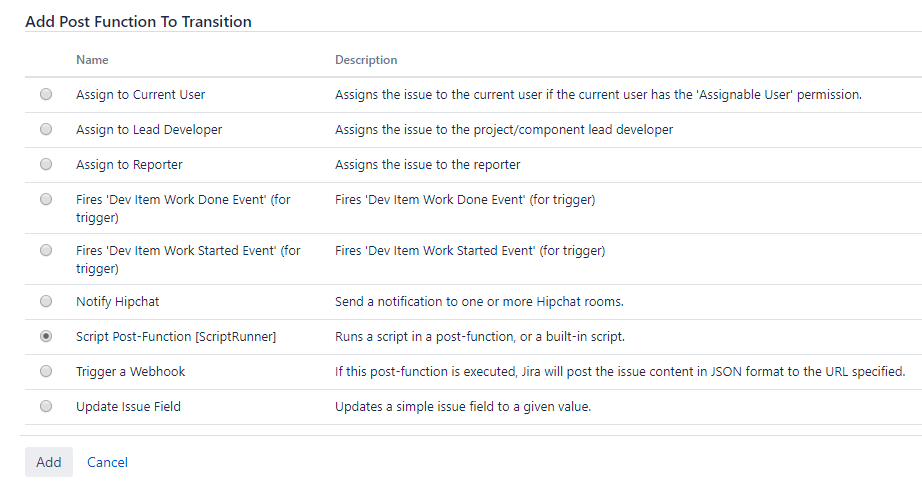
Click on the approve transition, the new dialog box will open there click post function,



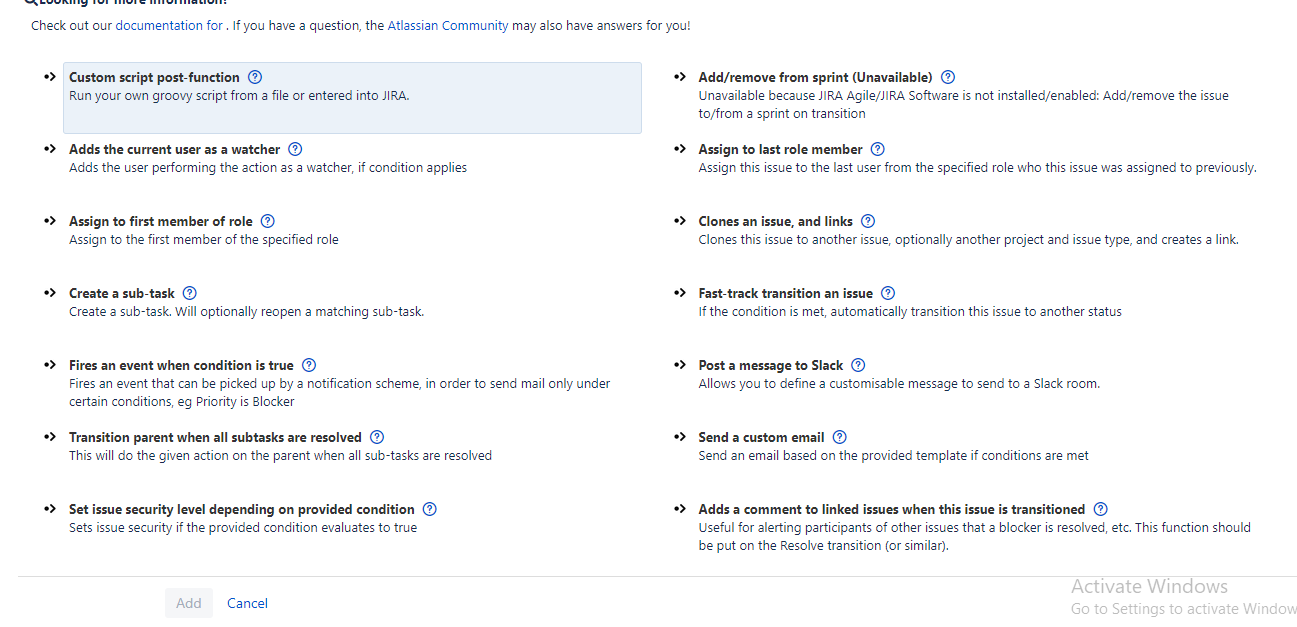
Click on add post function



Click on script post-function and click Add.



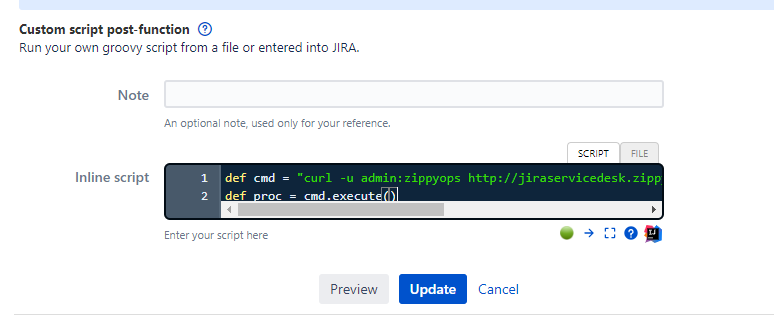
Click Custom script post function,



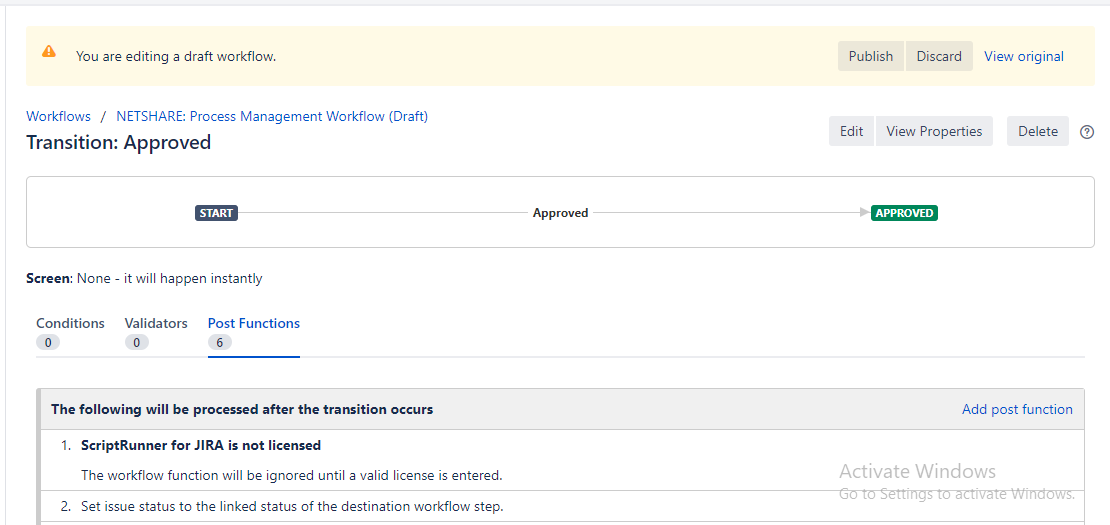
Type the script and Click on update,

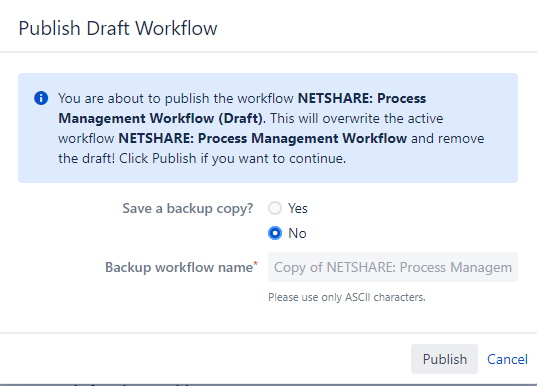
def cmd = "curl -u admin:zippyops http://jiraservicedesk.zippyops.com:8080/plugins/servlet/rundeck?issueKey=${issue.key}"

def proc = cmd.execute()



Click on publish,

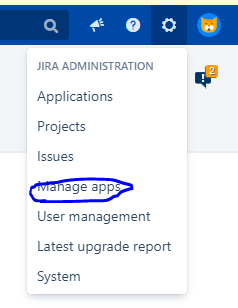




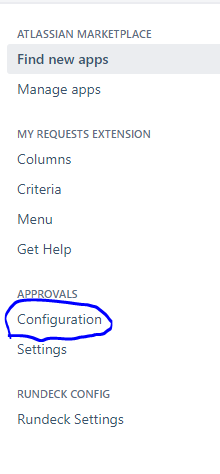
Like this do for the rejection transition,

Approval setup,

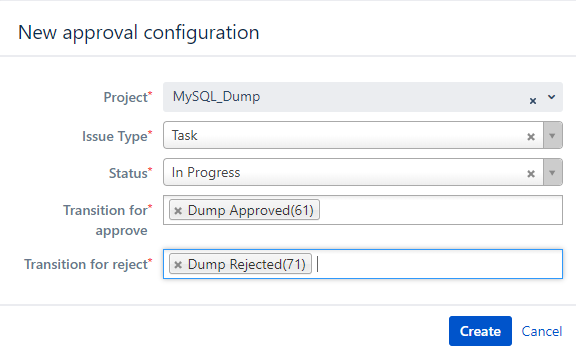
Go to settings, and click manage apps,



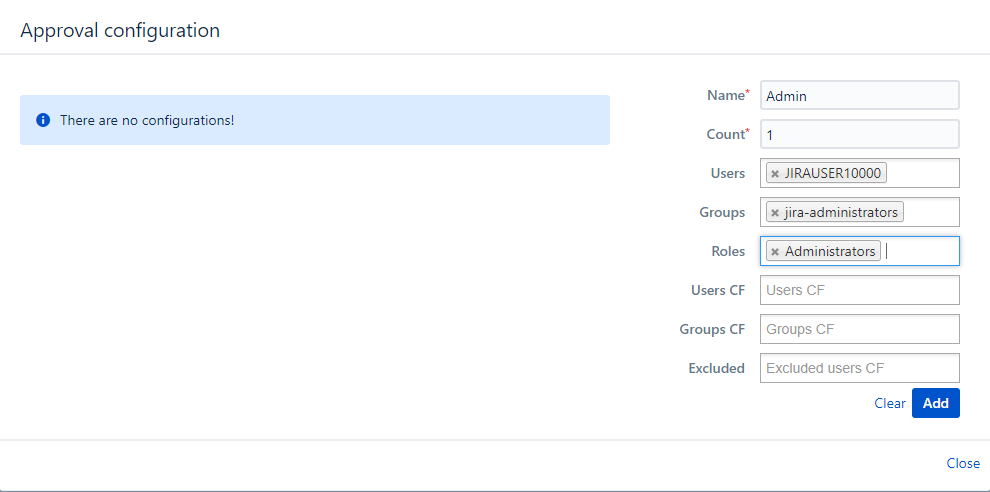
Click configuration below approval, if it not available install the plugins.



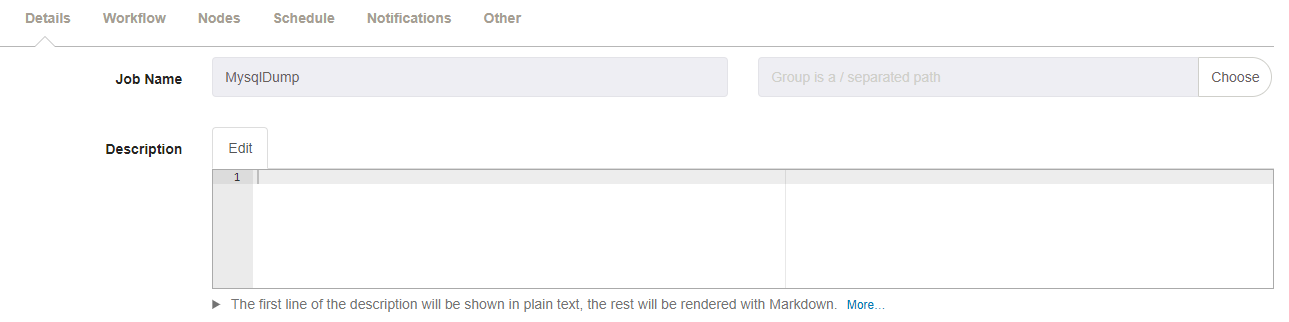
Click ADD, then click create



After creating click configuration, and click add

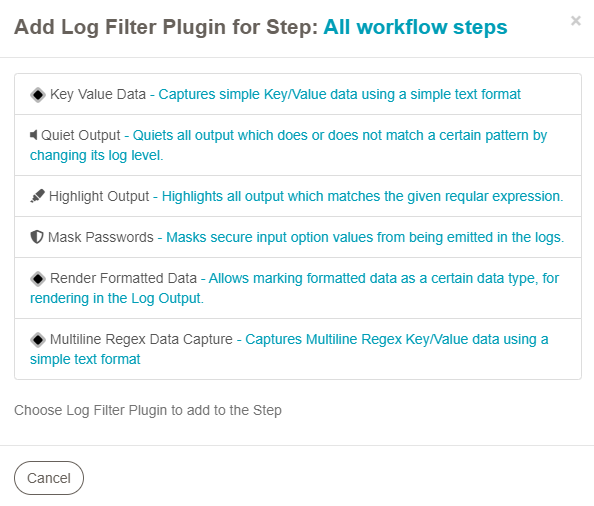


Create the job for Mysql Dump Approve in rundeck,

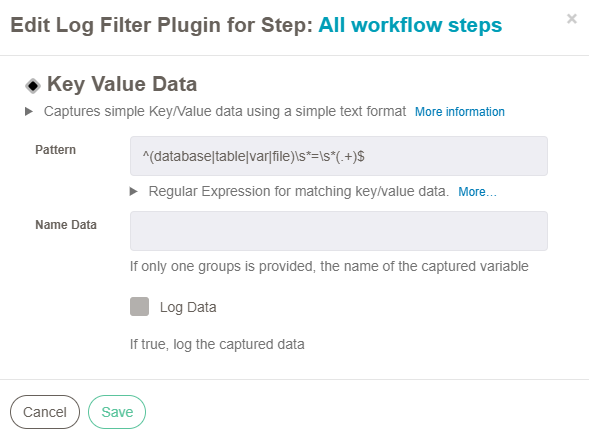


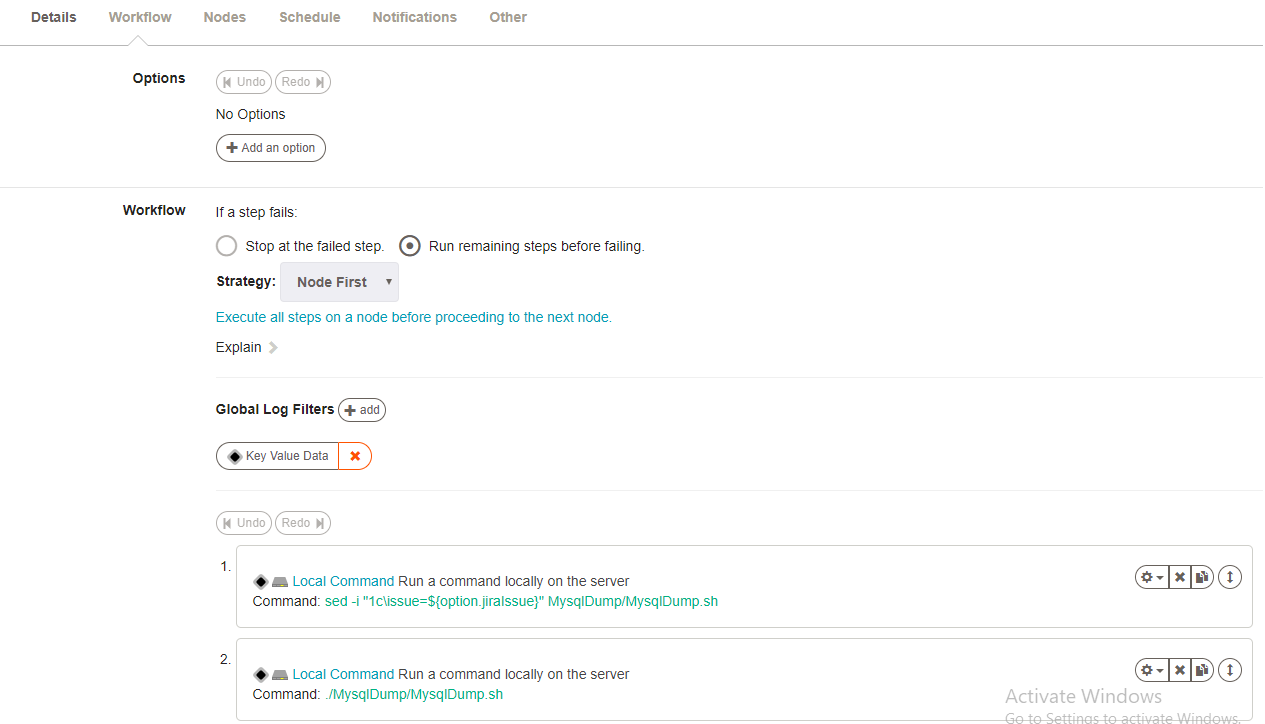
Workflow,

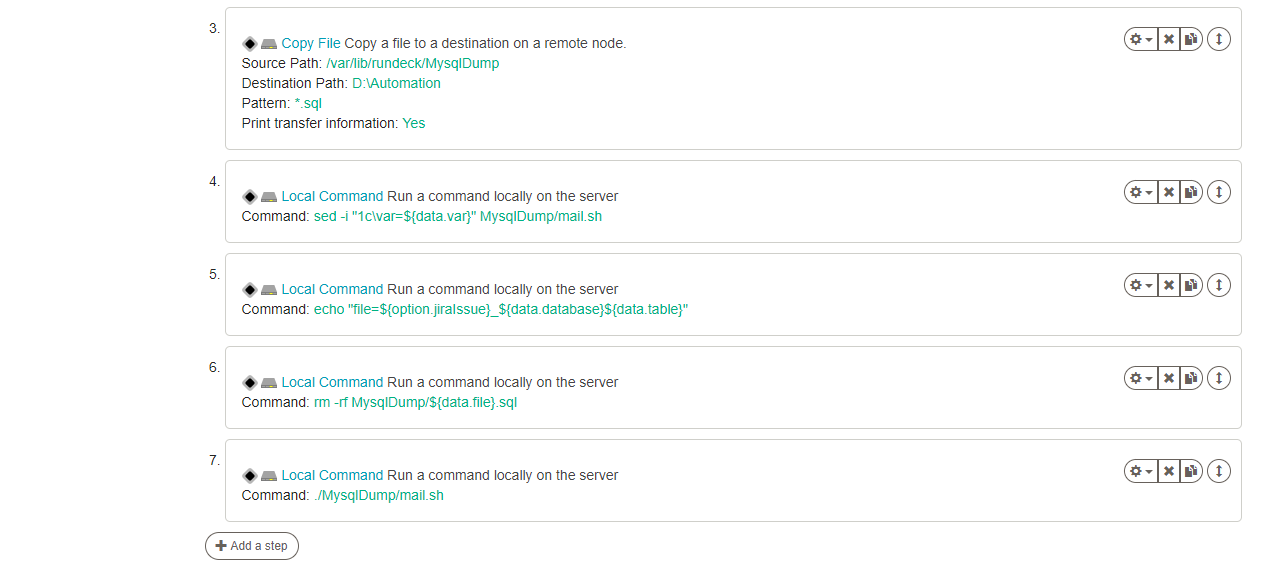
Click on Add Log Filter, and in that click Key Value Data.



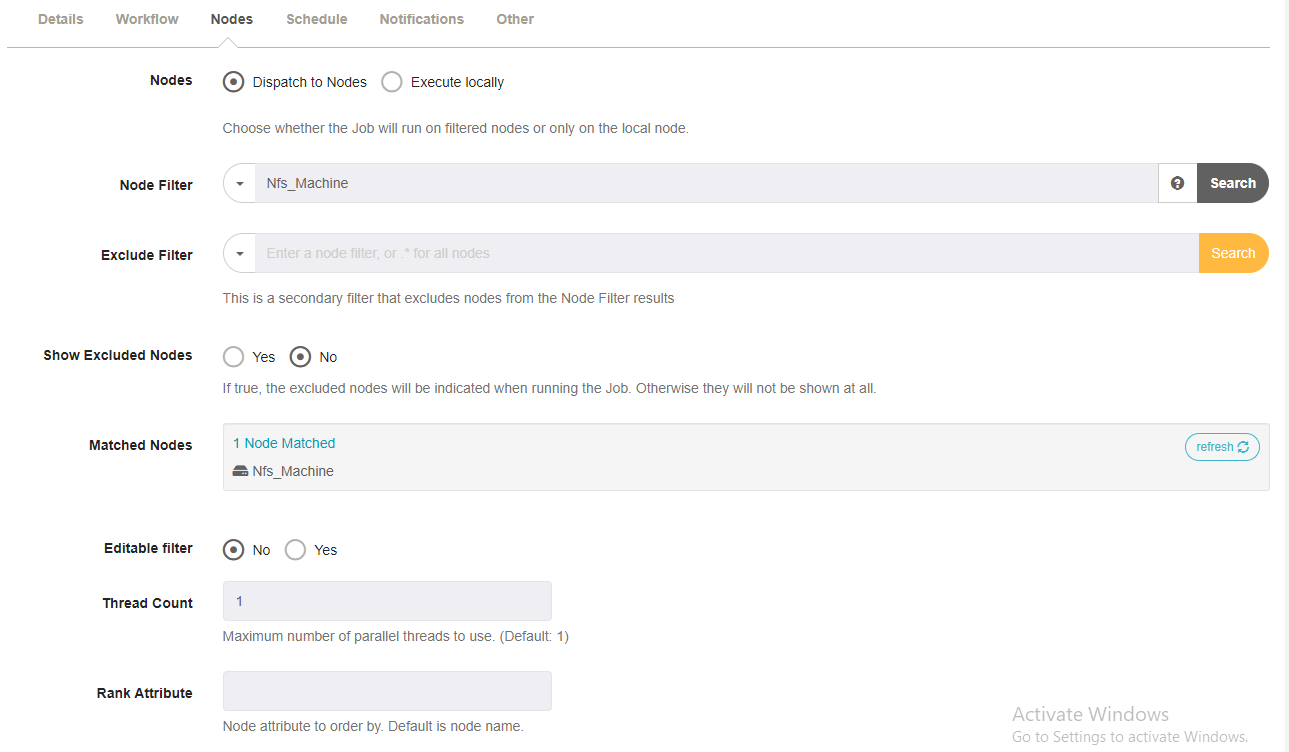
Give the pattern like this, and save.

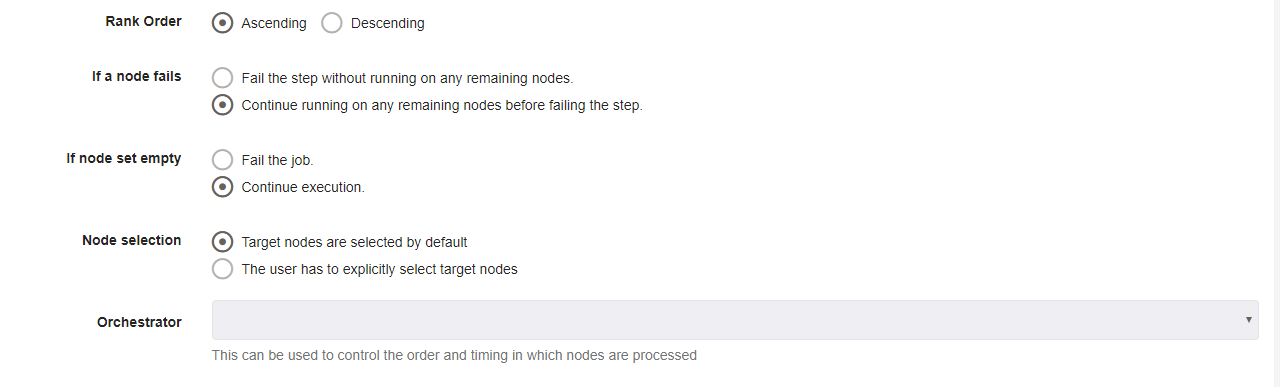






Nodes,





Click create,

Open terminal,

[root@rundeck ~]# cd /var/lib/rundeck/

[root@rundeck rundeck]# mkdir MysqlDump

[root@rundeck rundeck]# cd MysqlDump/

[root@rundeck MysqlDump]# cat MysqlDump.sh

issue=MYSQLDUMP-4

database=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Database" | tail -1)

table=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Tables" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

status=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Status" | tail -1)

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Server" | tail -1)

if [ $status == Approved ]; then

mysqldump -h $server -u rundeck -pzippyops $database $table > MysqlDump/"$issue"\_"$database""$table".sql

var=$(echo $?)

if [ $var != 0 ]; then

rm -rf MysqlDump/"$issue"\_"$database""$table".sql

fi

else

echo "Not Approved for Dumping data"

fi

echo "database=$database"

echo "table=$table"

echo "var=$var"

#echo "file='$issue'\_'$database''$table'.sql"

sed -i "2c\database=$database" MysqlDump/mail.sh

sed -i "3c\servername=$server" MysqlDump/mail.sh

sed -i "4c\status=$status" MysqlDump/mail.sh

sed -i "5c\table=$table" MysqlDump/mail.sh

sed -i "6c\mail=$mail" MysqlDump/mail.sh

sed -i "7c\issue=$issue" MysqlDump/mail.sh

[root@rundeck MysqlDump]# cat mail.sh

var=6

database=zippyops

servername=rundeckcentosnode.zippyops.com

status=Approved

table=authors

mail=praveenkumar.m@zippyops.in

issue=MYSQLDUMP-4

if [ $status == Approved ]; then

if [ $var == 0 ]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "MySQL Dump is done and transfer to Nfs server" | mail -s "Mysql Dump done" $mail

elif [ $var == 2 ]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "Database $database not found in the specfied database server $servername" | mail -s "Mysql Dump not done" $mail

elif [ $var == 6 ]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "Tables $table is not found in the database $database in the specfied database server $servername" | mail -s "Mysql Dump not done" $mail

fi

elif [ $status == Rejected ]; then

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "Dump the database is not Approved for you" | mail -s "Mysql Dump not done" $mail

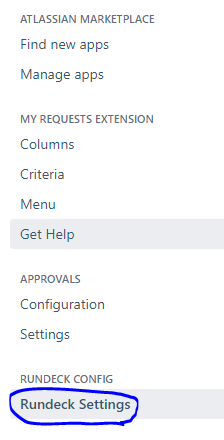
fi

[root@rundeck MysqlDump]#

Invoke Rundeck job setting, click the Manage apps

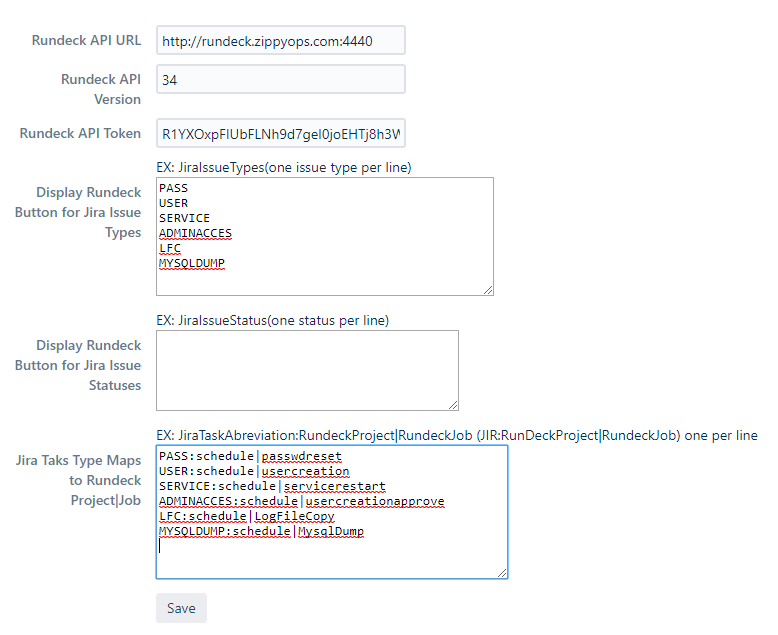


Click Rundeck setting,

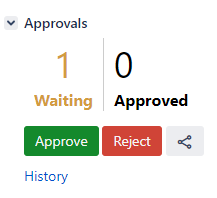


First mention Rundeck API URl and then API verion

Then generate the Api Token and paste here.



Come back to issue, and Click on Approve or Reject.

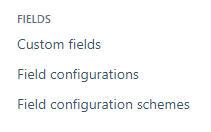


# mysql user creation

Creating custom field in jira, click setting and in that click issue,

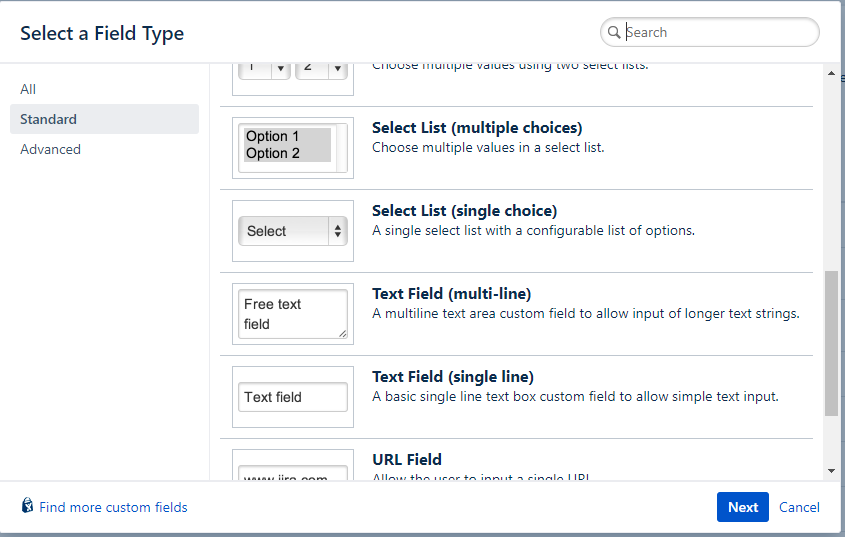


Click on Custom fields,

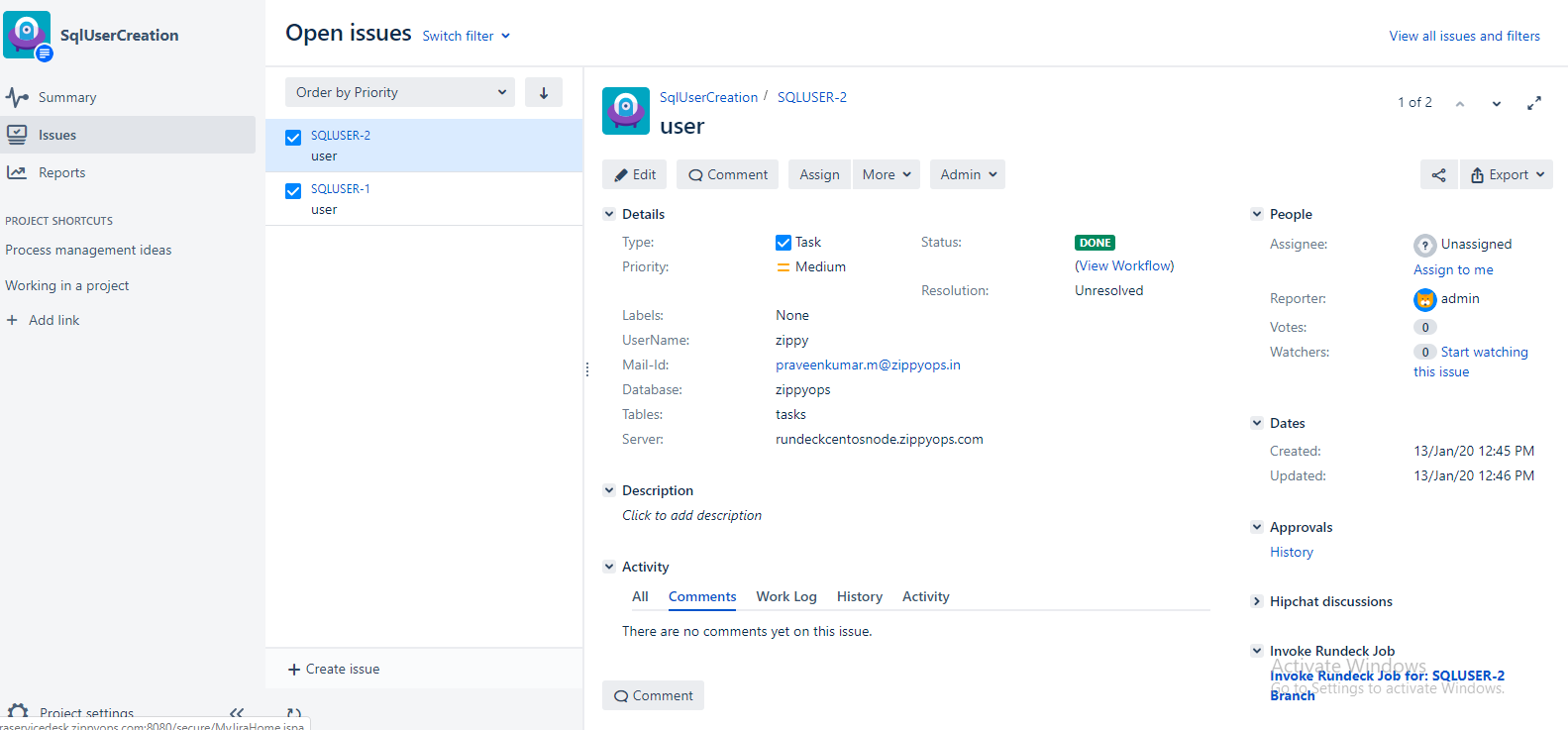


In that click Add Custom field in the right corner, In that create 5 field

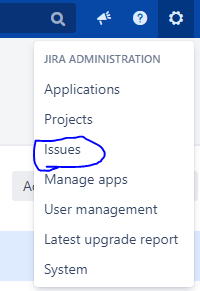
1. Tables field in Select List (Multiple choice)
2. Mail\_Id field in Text field (Single line)
3. Server field in Select List (Multiple choice)
4. Database field in Radio Button
5. Username field in Text field (Single line)



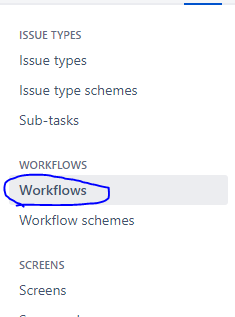
Create the new project with the name Mysql User Creation and issue id as SQLUSER.



Create the workflow, click on issue,

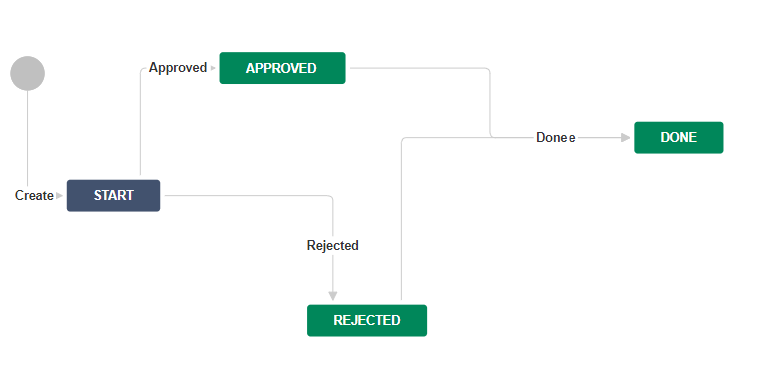


Click on workflow, there click on edit in which project you have to make change in workflow,

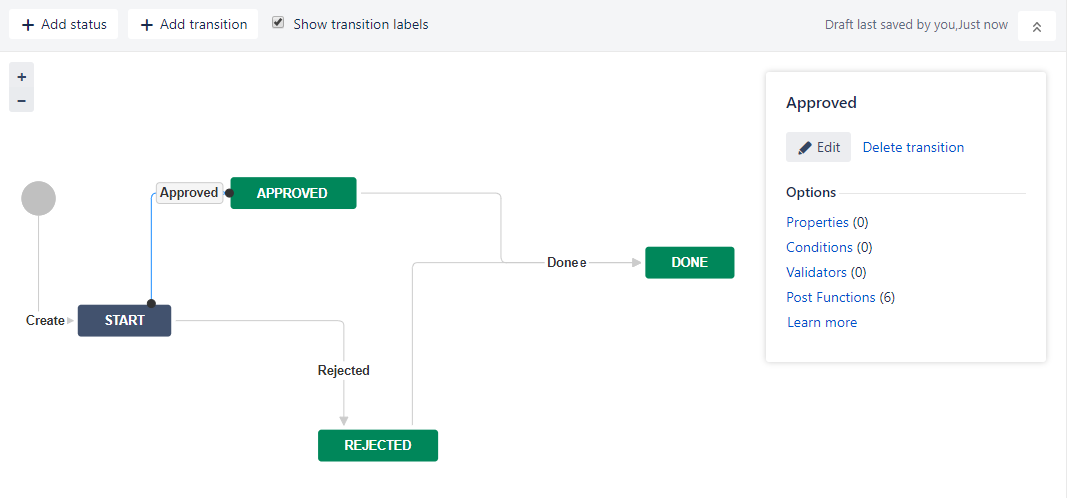




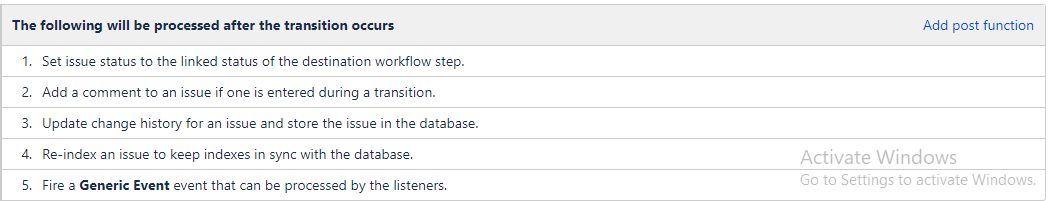
Create the workflow like this,



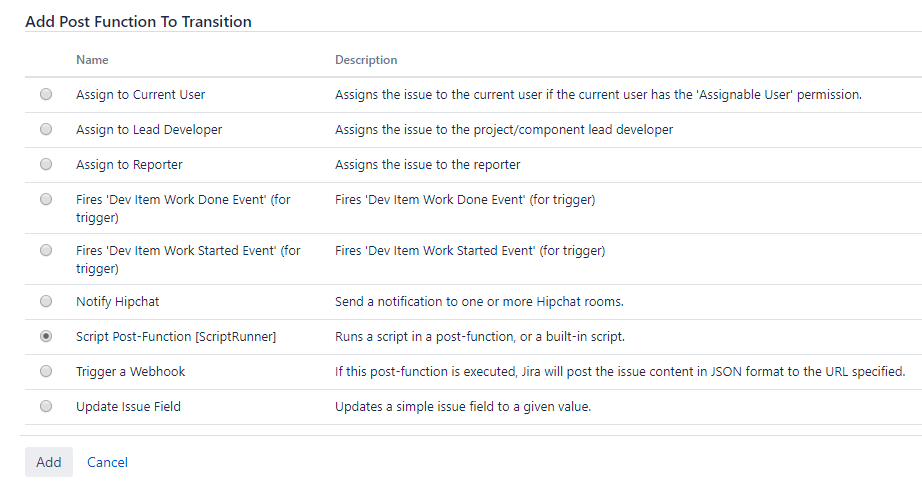
Click on the approve transition, the new dialog box will open there click post function,



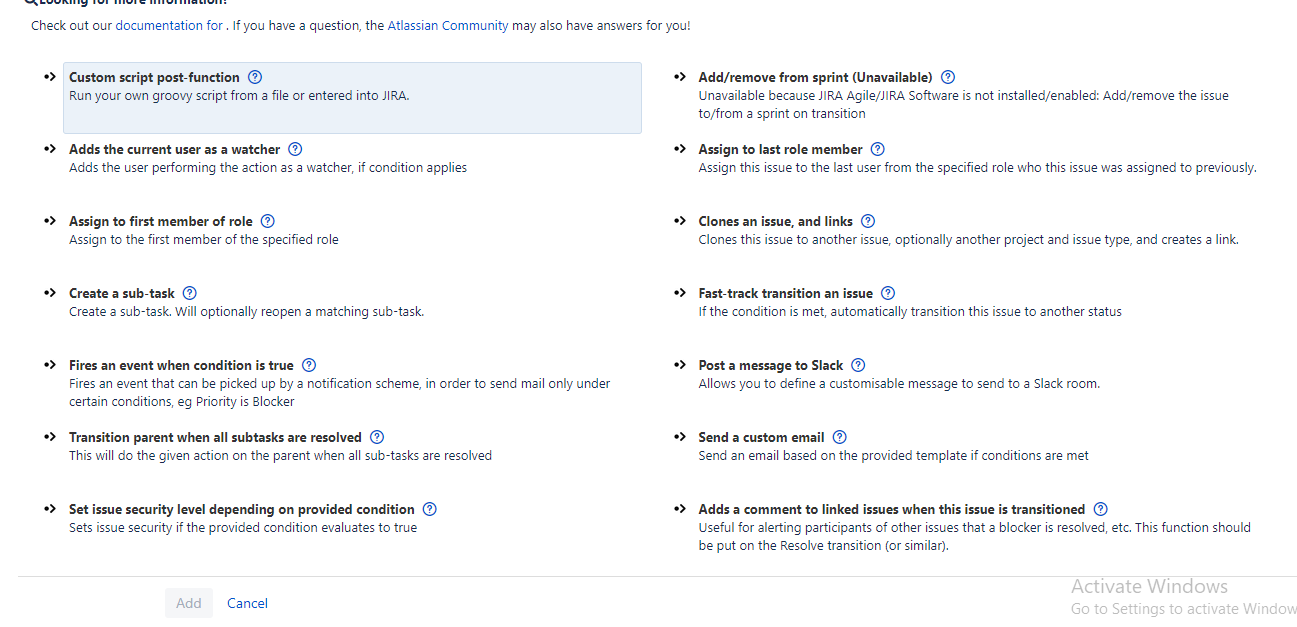
Click on add post function



Click on script post-function and click Add.



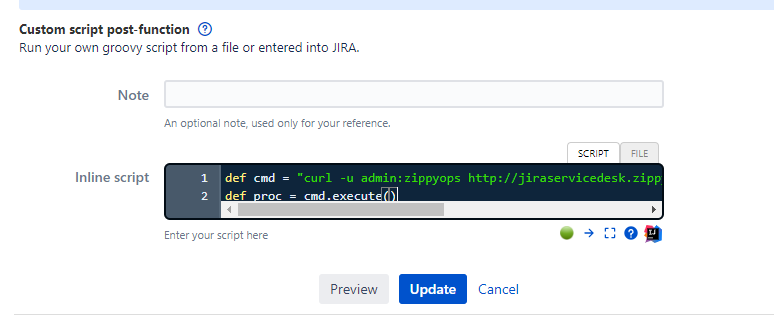
Click Custom script post function,



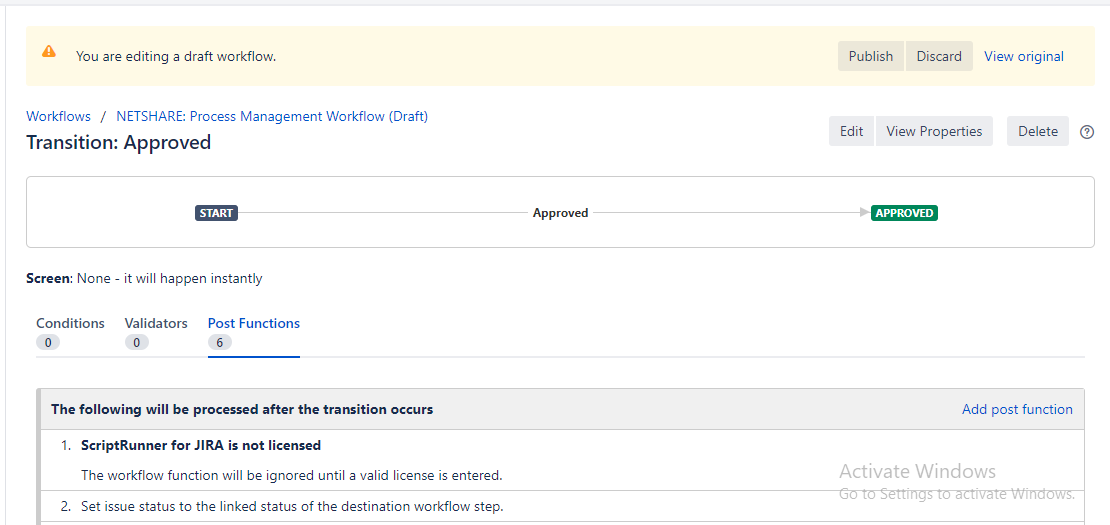
Type the script and Click on update,

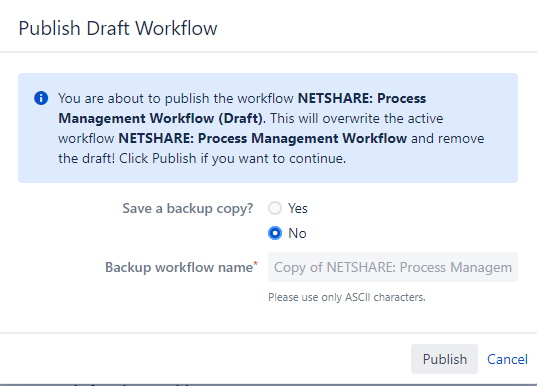
def cmd = "curl -u admin:zippyops http://jiraservicedesk.zippyops.com:8080/plugins/servlet/rundeck?issueKey=${issue.key}"

def proc = cmd.execute()



Click on publish,

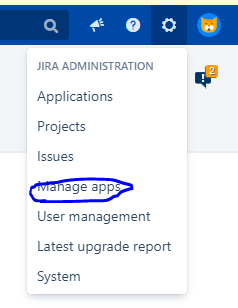




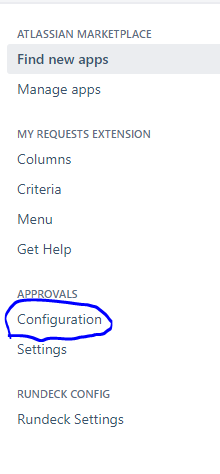
Like this do for the rejection transition,

Approval setup,

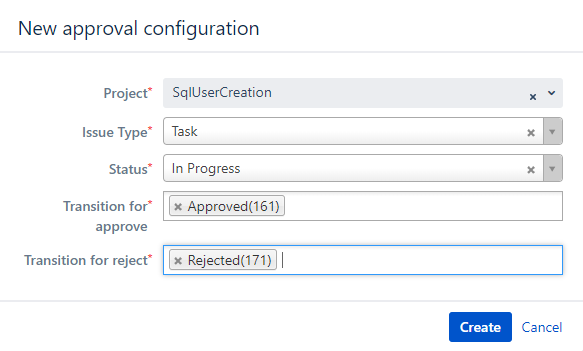
Go to settings, and click manage apps,



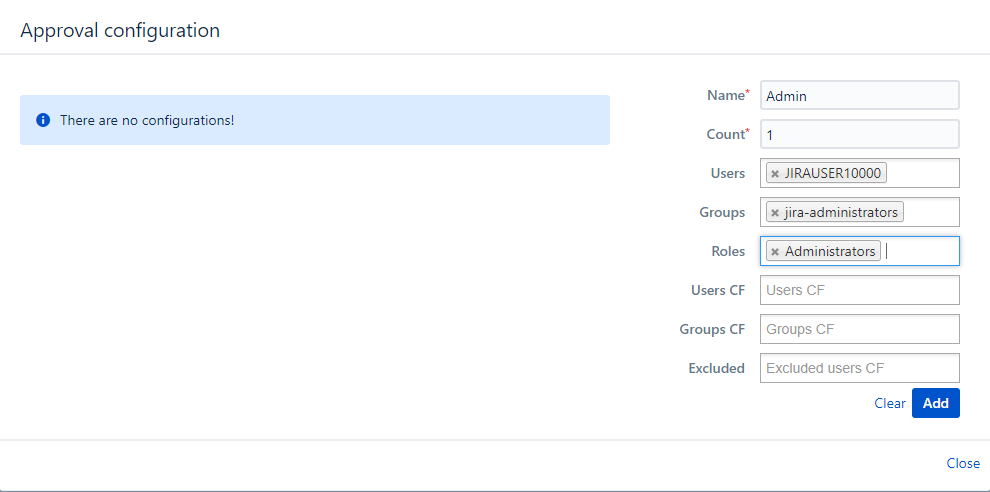
Click configuration below approval, if it not available install the plugins.



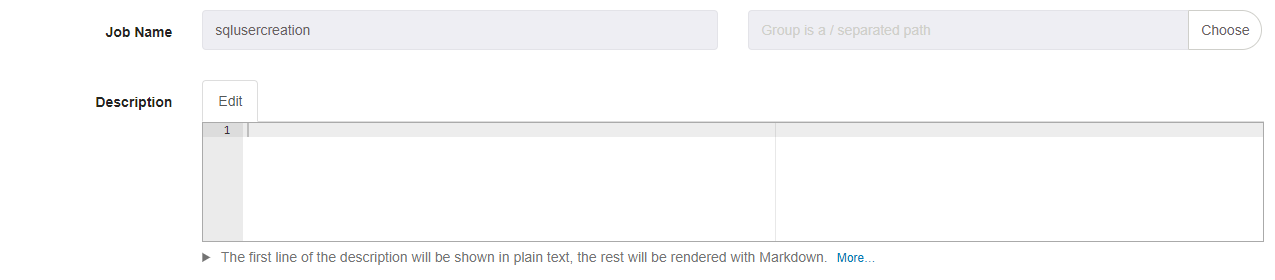
Click ADD, then click create



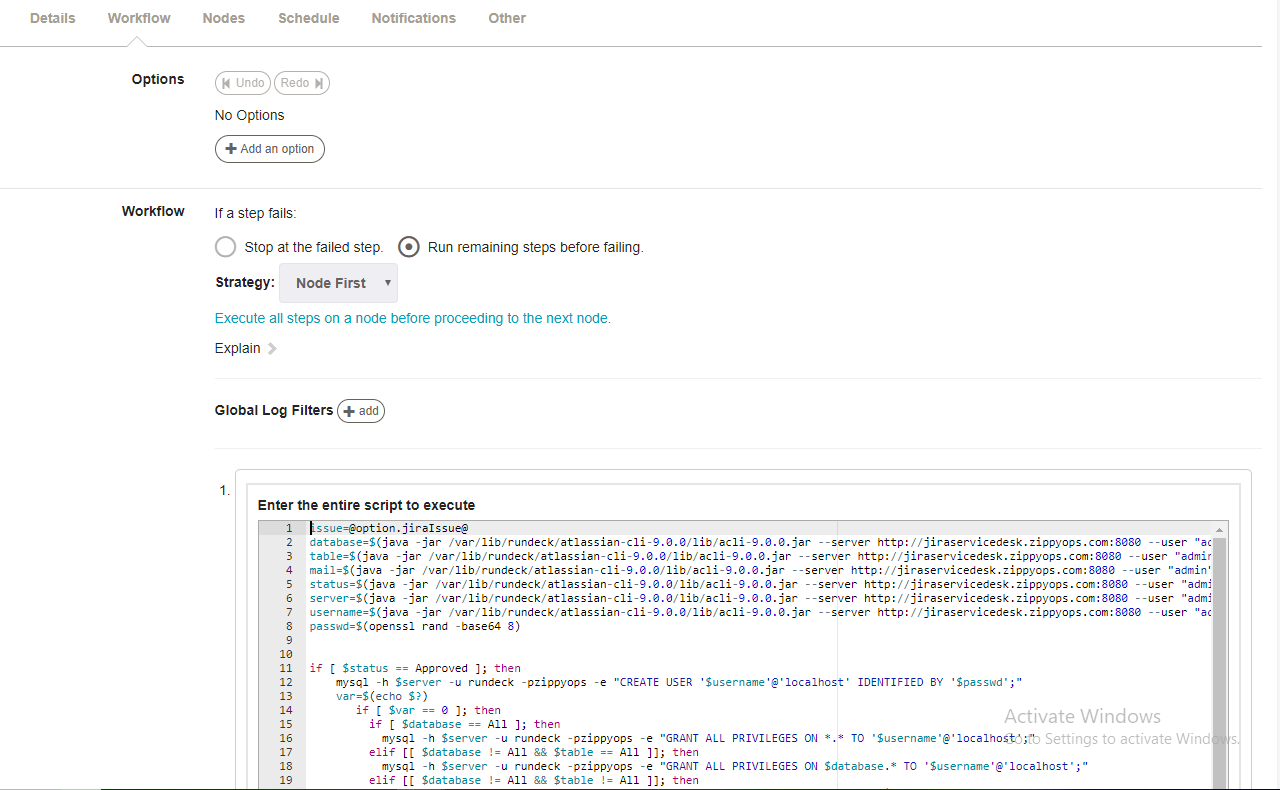
After creating click configuration, and click add

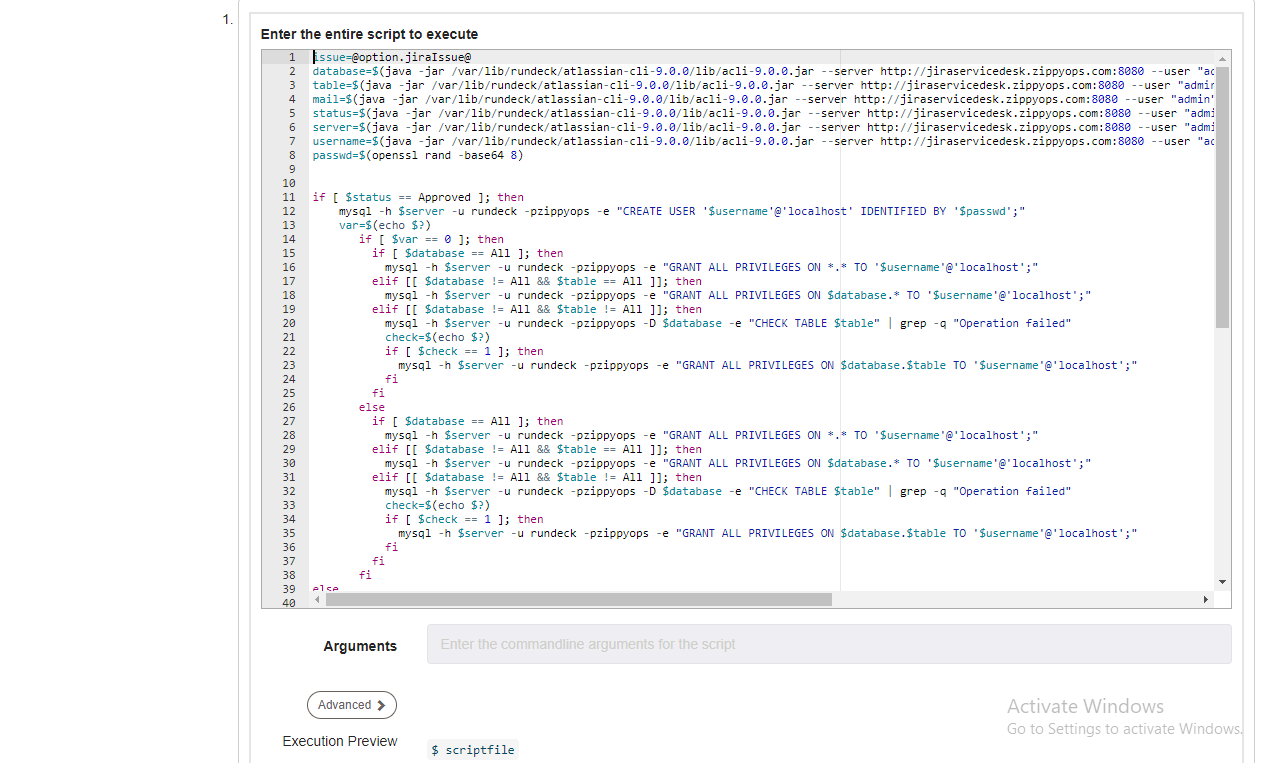


Create the job for Mysql user creation in rundeck,



Workflows,





issue=@option.jiraIssue@

database=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Database" | tail -1)

table=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Tables" | tail -1)

mail=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Mail-Id" | tail -1)

status=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Status" | tail -1)

server=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "Server" | tail -1)

username=$(java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action getFieldValue --issue "$issue" --field "UserName" | tail -1)

passwd=$(openssl rand -base64 8)

if [ $status == Approved ]; then

mysql -h $server -u rundeck -pzippyops -e "CREATE USER '$username'@'localhost' IDENTIFIED BY '$passwd';"

var=$(echo $?)

if [ $var == 0 ]; then

if [ $database == All ]; then

mysql -h $server -u rundeck -pzippyops -e "GRANT ALL PRIVILEGES ON \*.\* TO '$username'@'localhost';"

elif [[ $database != All && $table == All ]]; then

mysql -h $server -u rundeck -pzippyops -e "GRANT ALL PRIVILEGES ON $database.\* TO '$username'@'localhost';"

elif [[ $database != All && $table != All ]]; then

mysql -h $server -u rundeck -pzippyops -D $database -e "CHECK TABLE $table" | grep -q "Operation failed"

check=$(echo $?)

if [ $check == 1 ]; then

mysql -h $server -u rundeck -pzippyops -e "GRANT ALL PRIVILEGES ON $database.$table TO '$username'@'localhost';"

fi

fi

else

if [ $database == All ]; then

mysql -h $server -u rundeck -pzippyops -e "GRANT ALL PRIVILEGES ON \*.\* TO '$username'@'localhost';"

elif [[ $database != All && $table == All ]]; then

mysql -h $server -u rundeck -pzippyops -e "GRANT ALL PRIVILEGES ON $database.\* TO '$username'@'localhost';"

elif [[ $database != All && $table != All ]]; then

mysql -h $server -u rundeck -pzippyops -D $database -e "CHECK TABLE $table" | grep -q "Operation failed"

check=$(echo $?)

if [ $check == 1 ]; then

mysql -h $server -u rundeck -pzippyops -e "GRANT ALL PRIVILEGES ON $database.$table TO '$username'@'localhost';"

fi

fi

fi

else

echo "Not Approved for Creating user"

fi

if [ $status == Approved ]; then

if [ $var == 0 ]; then

if [ $database == All ]; then

echo "User $username is created with the password $passwd and given all database access" | mail -s "Mysql User Creation done" $mail

elif [[ $database != All && $table == All ]]; then

echo "User $username is created with the password $passwd and given all Table access TO the specfied database $database" | mail -s "Mysql User Creation done" $mail

elif [[ $database != All && $table != All ]]; then

if [ $check == 1 ]; then

echo "User $username is created with the password $passwd and given $table Table access in the specfied database $database" | mail -s "Mysql User Creation done" $mail

elif [ $check == 0 ]; then

echo "User $username is created with the password $passwd and table $table not found in the database $database" | mail -s "Mysql User Creation done" $mail

fi

fi

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

else

if [ $database == All ]; then

echo "User $username is already created and given all database access" | mail -s "User Creation not done" $mail

elif [[ $database != All && $table == All ]]; then

echo "User $username is already created and given all Table access in the specfied database $database" | mail -s "Mysql User Creation not done" $mail

elif [[ $database != All && $table != All ]]; then

echo "User $username is already created and given $table Table access in the specfied database $database" | mail -s "Mysql User Creation not done" $mail

if [ $check == 1 ]; then

echo "User $username is already created and given $table Table access in the specfied database $database" | mail -s "Mysql User Creation not done" $mail

elif [ $check == 0 ]; then

echo "User $username is already created and table $table not found in the database $database" | mail -s "Mysql User Creation not done" $mail

fi

fi

java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

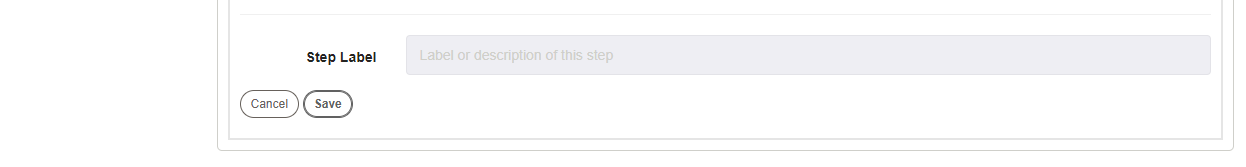
fi

elif [ $status == Rejected ]; then

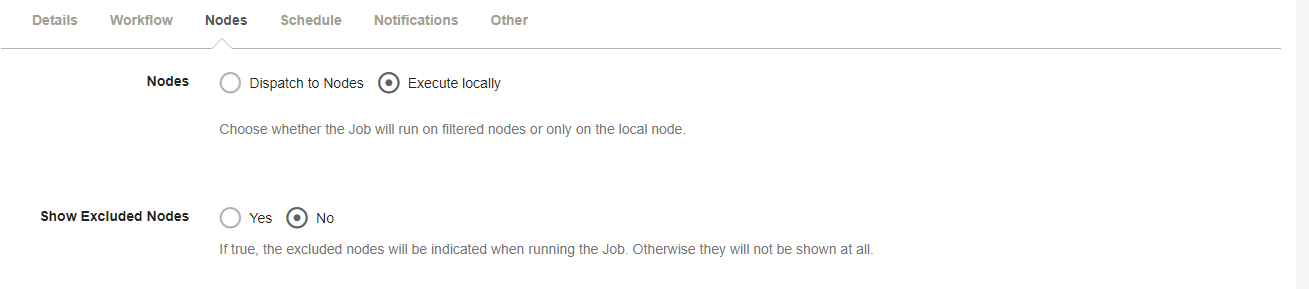
java -jar /var/lib/rundeck/atlassian-cli-9.0.0/lib/acli-9.0.0.jar --server http://jiraservicedesk.zippyops.com:8080 --user "admin" --password "zippyops" --action transitionIssue --issue "$issue" --transition "Done"

echo "User Creation and Access is not Approved for you" | mail -s "Mysql User Creation not done" $mail

fi



Nodes,

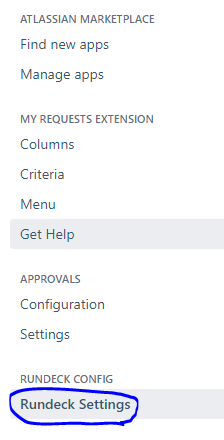


Click Create ,

Invoke Rundeck job setting, click the Manage apps

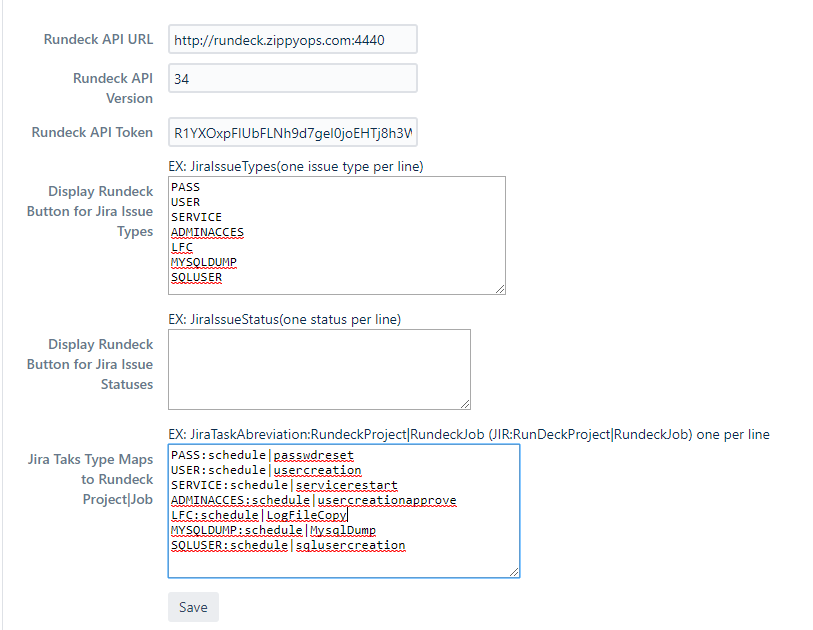


Click Rundeck setting,



First mention Rundeck API URl and then API verion

Then generate the Api Token and paste here.



Come back to issue,

Click on Approve or Reject.

