

GNU Screen:

GNU Screen is a terminal multiplexer, a software application that can be used to multiplex several virtual consoles, allowing a user to access multiple separate login sessions inside a single terminal window, or detach and reattach sessions from a terminal. It is useful for dealing with multiple programs from a command line interface, and for separating programs from the session of the Unix shell that started the program, particularly so a remote process continues running even when the user is disconnected.

It's like a tab.

Command to install screen in Linux.

```
[] sudo apt install screen.
```

Screen commands

```
[] screen -version:
```

```
[] screen -S sessionName: This Screen command to start a new session named sessionName.
```

```
[] screen -ls: This screen command helps to list active screen
```

```
[] screen -r sessionName: This command helps to reattach to a running Screen session.
```

```
[] Ctrl-a + D: This command helps to detach the running screen.
```

```
[] echo $STY : To check which screen session.
```

[] screen -XS sessionName kill: To delete screen

[] pkill screen: To delete multiple screen

Jenkins pipeline code for create the two screen and run the two-python script, inside the two different screens

stage first: git clone

stage two: [] creating screen, if screen is there it will ignore. Here the screens name is my_screen_session and my_screen_session2

[] moving inside the screen, stop the python script and restart the python. Here my script name is in my_screen_session hello.py will run and in my_screen_session2 hlo.py will run.

```
pipeline {  
  
    agent any  
  
    stages {  
        stage(checkout) {  
            steps {  
                git branch: 'main', credentialsId: 'GIT_PATH', url:  
'https://github.com/kanchana08/testing.git'  
  
            }  
  
        }  
    }  
}
```

```

stage('Hello'){

    steps {

        script {

            // creating screen

            sh'''

            if ! sudo -u jenkins bash -c "screen -list | grep -q 'my_screen_session';then

                sudo -u jenkins bash -c "screen -dmS my_screen_session"

            fi

            if ! sudo -u jenkins bash -c "screen -list | grep -q 'my_screen_session2';then

                sudo -u jenkins bash -c "screen -dmS my_screen_session2"

            fi

            '''

            // moving inside the screen and stop the script.

            sh"screen -S my_screen_session -X stuff '^c'"

            sh "screen -S my_screen_session2 -X stuff '^C'"

            // moving inside the screen and restart the script.

            sh "screen -S my_screen_session -X stuff 'python3 hello.py\\n'"

            sh "screen -S my_screen_session2 -X stuff 'python3 hlo.py\\n'"

            sh 'screen -ls'

```

```
    }  
  
    }  
  }  
}  
}
```

Jenkins pipeline code for run the two-python script, inside the two different screens

[] we created screen manually and here I mentioned screen name with ID 5454.my_screen_session, with ID will help when we have two screen with same name at that time we can use with ID

```
pipeline {  
    agent any  
    stages {  
        stage(checkout){  
            steps{  
  
                git branch: 'main', credentialsId: 'GIT_PATH', url:  
'https://github.com/kanchana08/testing.git'  
            }  
        }  
    }  
}
```

```

    stage('Hello') {
        steps {
            // moving inside the screen and stop the python script
            sh "screen -S 5454.my_screen_session -X stuff '^c'"
            sh "screen -S my_screen_session -X stuff '^C'"
            // moving inside the screen and start the python script
            sh "screen -S 5454.my_screen_session -X stuff 'python3
hello.py\\n'"
            sh "screen -S my_screen_session -X stuff 'python3 hlo.py\\n'"

        }
    }
}

```

Jenkins pipeline code for to create screen, if the screen is there it will ignore not there it will create, if any of one screen is deleted it will create deleted screen.

```

pipeline {

    agent any

```

```

stages {
    stage(checkout){
        steps{
            git branch: 'main', credentialsId: 'GIT_PATH', url:
'https://github.com/kanchana08/testing.git'

        }

    }

    stage('Hello') {

        steps {

            script {

                sh'''

                    if ! sudo -u jenkins bash -c "screen -list | grep -wq
'my_screen_session';then
                        sudo -u jenkins bash -c "screen -dmS
my_screen_session"
                    fi
                    ...

                sh'''

                    if ! sudo -u jenkins bash -c "screen -list | grep -wq
'my_screen_session2';then

```

