

Karthik Pradeep Hegadi

2KE20CS032

Assignment 30

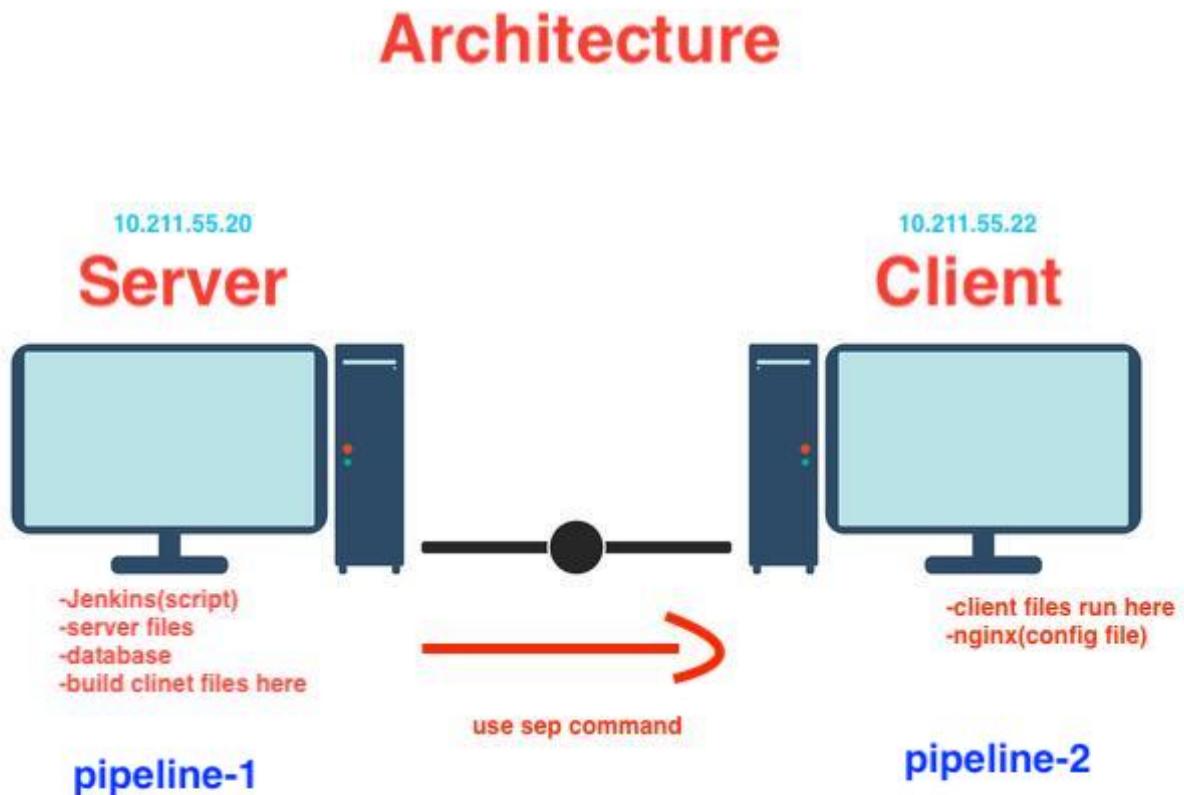
Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

Jenkins-6

Assignment - 6- 3 Tier architecture app involving NodeJS,

ReactS and MYSOL

(You can refer the manual setup of server & client side, and you need to implement this in pipeline script)



PIPELINE-1(server-side)

1 creating the project

The screenshot shows the Jenkins configuration interface for the 'nodejs_server' project. The 'General' tab is selected. In the 'Description' field, the text 'nodejs project server side automation' is entered. The 'Enabled' switch is turned on. On the left sidebar, 'Pipeline' is selected. Below the main area, there are several checkboxes for build options: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', 'Pipeline speed/durability override', and 'Preserve stashes from completed builds'. A 'Plain text' link is available for previewing the configuration.

2 script

The screenshot shows the Jenkins configuration interface for the 'nodejs_server' project. The 'Pipeline' tab is selected. In the 'Definition' dropdown, 'Pipeline script' is chosen. The 'Script' editor contains the following Groovy code:

```
1+ node {
2+   try {
3+     // Stage 1: Install Node.js
4+     stage('Install Node.js') {
5+       sh 'sudo yum install -y nodejs'
6+     }
7+
8+     // Stage 2: Install MySQL
9+     stage('Install MySQL') {
10+
11+
12+   }
13+
14+   // Stage 3: Clone Git Repository
15+   stage('Clone Git Repository') {
16+     sh 'git clone https://github.com/username/repo.git'
17+   }
18+ }
```

Below the script, the 'Use Groovy Sandbox' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons. A preview window on the right shows the Jenkins interface with the pipeline stages listed.

3 steps in script

nodejs deployment
deployment at server-side

steps

1. install nodejs
2. install mysqld
3. git clone
 1. link :https://github.com/kartikhegadi/nodejs_server.git
 2. as main branch
4. auto login mysqld with credentials
 1. as root user
 2. password is "password"
 3. create a database doctor_appointment;
 4. exit
5. import data base
 1. doctor_appointment to mysql
 2. as root user
 3. password is "password"
6. open app.js
 1. go to const pool function
 2. in that go to password = "password" change to password
 3. save it and come back
7. in this directory /home/usy/nodejs_server
 1. run npm start

4 whole script

```
===== (pipeline)
node {
    try {
        // Stage 1: Install Node.js
        stage('Install Node.js') {
            sh 'sudo yum install -y nodejs'
        }

        // Stage 2: Install MySQL
        stage('Install MySQL') {
            sh 'already installed'
        }

        // Stage 3: Clone Git Repository
        stage('Clone Git Repository') {
            checkout([$class: 'GitSCM', branches: [[name: 'main']], userRemoteConfigs: [[url: 'https://github.com/kartikhegadi/nodejs_server.git']]])
        }

        // Stage 4: Auto-login to MySQL and create a database
        stage('Create MySQL Database') {
            sh 'mysql -u root --password="password" -e "CREATE DATABASE IF NOT EXISTS doctor_appointment;"' creating data-base if it is not exist
        }

        // Stage 5: Import the database
        stage('Import Database') {
            sh 'mysql -u root --password="password" doctor_appointment < doctor_appointment.sql'
        }
    }
}
```

```

// Stage 6: Modify app.js
stage('Modify app.js') {
    sh 'sed -i \'s/password = "root"/password = "password"/\' /var/lib/jenkins/workspace/nodejs_server/app.js'
}

// Stage 7: Run the Node.js Application
stage('Run Node.js Application') {
    dir('/var/lib/jenkins/workspace/nodejs_server/') {
        sh 'npm install'      start server
        sh 'npm start'
    }
}

} catch (Exception e) {
    currentBuild.result = 'FAILURE'
    throw e
} finally {

}

```

5 build it

The screenshot shows the Jenkins Pipeline nodejs_server stage view. It displays a timeline of stages: Install Node.js, Install MySQL, Clone Git Repository, Create MySQL Database, Import Database, Modify app.js, Run Node.js Application, and Cleanup. Stage #14 is shown with a duration of 1min 11s. Stage #13 is shown with a duration of 3min 8s. The 'Build Now' button is highlighted.

Stage	Average stage times
Install Node.js	1s
Install MySQL	16ms
Clone Git Repository	6s
Create MySQL Database	291ms
Import Database	280ms
Modify app.js	287ms
Run Node.js Application	2min 37s
Cleanup	12ms

6 out put

The screenshot shows the Jenkins Pipeline nodejs_server #14 build log. The log output includes npm audit results, npm install, and npm start commands. A red arrow points from the 'server started and listening' message to the 'listening on port 3001' message at the bottom of the log.

```

+ sed -i 's/password = "root"/password = "password"/\' /var/lib/jenkins/workspace/nodejs_server/app.js
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Node.js Application)
[Pipeline] dir('/var/lib/jenkins/workspace/nodejs_server')
Running in /var/lib/jenkins/workspace/nodejs_server
[Pipeline] {
[Pipeline] sh
+ npm install

up to date, audited 116 packages in 6s

11 packages are looking for funding
  run `npm fund` for details

3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
[Pipeline] sh
+ npm start

> server-mysql@1.0.0 start
> node app.js

listening on port 3001

```

Note*-> before going to server we should conf some things in Jenkins and we need to set the credentials to use gmail and ssh (to move the files form one machine to another)**

a) sonar cube

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name	sonar
Server URL	Default is http://localhost:9000



b) Java Development Toolkit

JDK installations

JDK installations ^ Edited

Add JDK

JDK	Name
	jdk11
<input checked="" type="checkbox"/> Install automatically	?
≡ Install Oracle Java SE Development Kit from the website ?	
Version	Java SE Development Kit 9.0.4

c) Node js extension



d) Credentials

Credentials

T	P	Store ↓	Domain	ID	Name
💻	👤	System	(global)	sonarcube_key	sonarcube_key
💻	👤	System	(global)	ea2bd294-296b-4a4d-8392-5b88eec4bb74	2ke20cs086_@kleit.ac.in /***** (Hey kartihik .. this is message from jenkins.)
⌚	👤	System	(global)	MySSHKey	Iusy
💻	👤	System	(global)	fdbed7f5-c072-4b6b-ac6a-a18ce9b17eeb	2ke20cs086_t@kleit.ac.in/***** (Hey kartihik .. this is message from jenkins.)
💻	👤	System	(global)	gmail.conf	2ke20cs086_t@kleit.ac.in/***** (gmail.conf)

Note***-> Email configurations

a) downloading plugins

The screenshot shows the Jenkins plugin manager interface. The 'Installed' tab is selected. A search bar at the top has the text 'email' entered. Below the search bar, a table lists a single plugin: 'Email Ext Recipients Column Plugin' (version 27.vb_9404db_b_018d). The plugin is described as a sample to explain how to write a Jenkins plugin. It has an 'Enabled' status indicated by a green checkmark icon. There are also 'Report an issue with this plugin' and 'Edit' buttons.

b) Setting the app password after verifying the 2 step verification

c) E-mail notification settings

d) Extended Email notifications

The screenshot shows the Jenkins system configuration page under 'Email Notifications'. It includes fields for 'SMTP Port' (set to 465), 'Reply-To Address' (empty), 'Charset' (set to UTF-8), and a checkbox for 'Test configuration by sending test e-mail' which is checked. A 'Test e-mail recipient' field contains the email address 'karthikhegdi143@gmail.com'. Below the form, there is a success message 'Email was successfully sent' and a red error message 'testing'.

Post-build Actions

≡ Editable Email Notification ?

Allows the user to disable the publisher, while maintaining the settings

Disable Extended Email Publisher ?

Project From

Project Recipient List ?

Comma-separated list of email address that should receive notifications for this project.

2ke20cs086_t@kleit.ac.in

Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

\$DEFAULT_REPLYTO

Content Type ?

HTML (text/html)

Default Subject ?

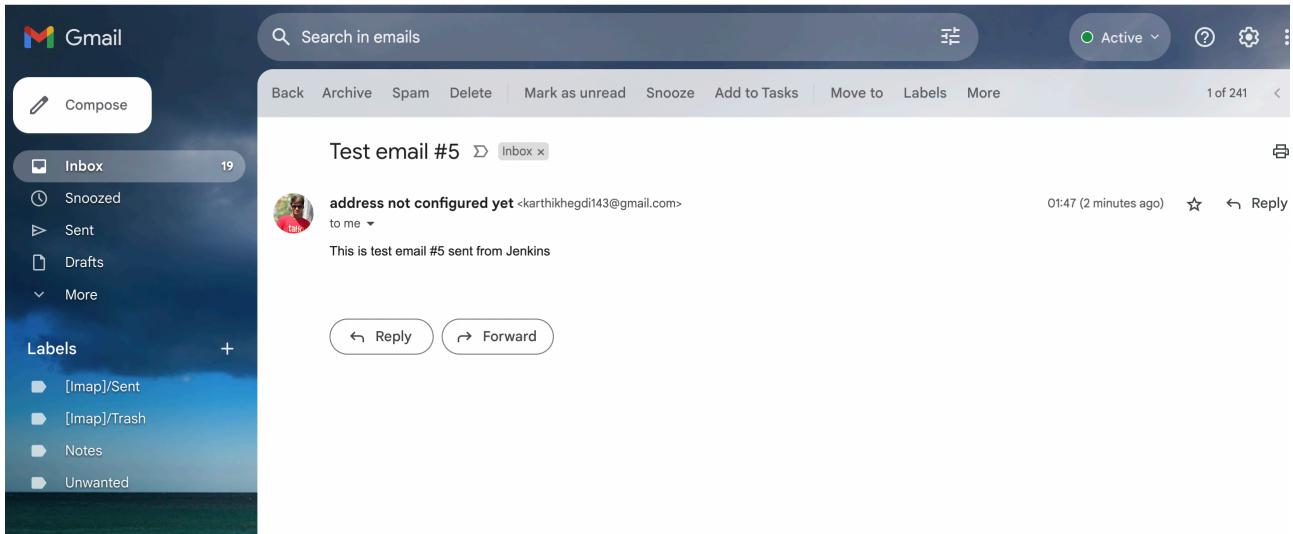
\$DEFAULT SUBJECT

Default Content ?

\$DEFAULT_CONTENT

e) Test mail

(Same steps I have set for for other mail and build the script)



PIPELINE-2(clinet-side)

a) name the project select it as pipeline

A screenshot of a web browser displaying the Jenkins dashboard. The address bar shows the URL 10.211.55.20:8080/view/all/newJob. The page header includes a "Getting Started" link, a search bar, and a notification bar with 1 message and 2 builds. The main content area has a title "Enter an item name" with a required field "nodejs_deployment". Below this, there are four project types listed: "Freestyle project" (represented by a box icon), "Maven project" (represented by an owl icon), "Pipeline" (represented by a circular icon with a flowchart), and "Multi-configuration project" (represented by a gear icon). The "Pipeline" option is currently selected.

b) About the project

The screenshot shows the Jenkins configuration interface for a job named 'nodejs_deployment'. The 'General' tab is selected. The 'Description' field contains the text: 'Assignment 6 : Build Jenkins Pipeline for 3 tier architecture involving Nodejs'. Below the description are several checkboxes: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', and 'GitHub project'. A 'Plain text Preview' link is also present. On the left sidebar, there are links for 'General', 'Advanced Project Options', and 'Pipeline'.

c) Manually conf the nginx config and set default server

```
lousy@localhost.locaLdomain:/etc (0.02s)
cd nginx/

lousy@localhost.locaLdomain /etc/nginx (0.028s)
ls
conf.d  fastcgi.conf.default    koi-utf      mime.types.default  scgi_params      uwsgi_params.default
default.d  fastcgi_params       koi-win      nginx.conf        scgi_params.default  win-utf
fastcgi.conf  fastcgi_params.default  mime.types  nginx.conf.default  uwsgi_params
```

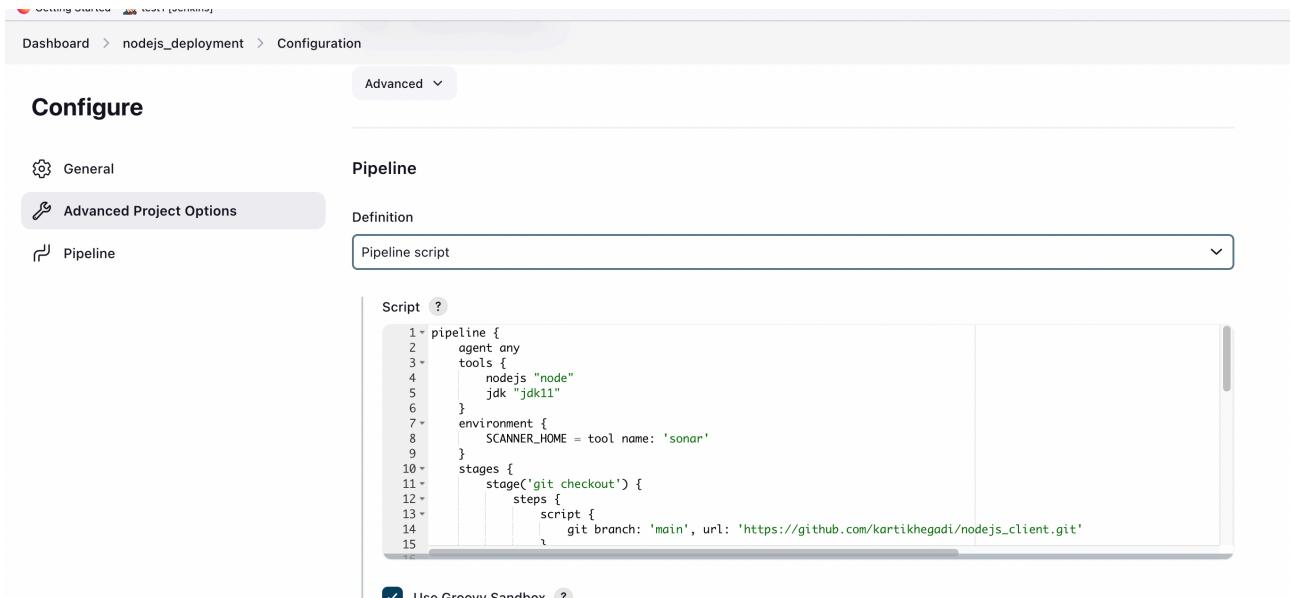
```
lousy@localhost.locaLdomain /etc/nginx/conf.d (0.022s)
cat reactjs.conf
server {
    listen 80;
    listen [::]:80;
    root /usr/share/nginx/html/build;

    location /api/users {
        proxy_pass http://10.211.55.20:3001/api/users;           reverse proxy to trigger
        proxy_set_header Host $host;                                server in which DB is
        proxy_set_header X-Real-IP $remote_addr;                     setup
    }

    location /api/ {
        proxy_pass http://10.211.55.20:3001/api/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

d) Pipeline script



Full script

2 November 2023 at 12:57 AM

```
pipeline {
  agent any
  tools {
    nodejs "node"
    jdk "jdk11"
  }
  environment {
    SCANNER_HOME = tool name: 'sonar'
  }
  stages {
    stage('git checkout') {
      steps {
        script {
          git branch: 'main', url: 'https://github.com/kartikhegadi/nodejs_client.git'
        }
      }
    }
    stage('npm install') {
      steps {
        sh 'npm install'
      }
    }
    stage('npm with SonarQube') {
      steps {
        withSonarQubeEnv('sonar') {
          sh """
            ${SCANNER_HOME}/bin/sonar-scanner \
            -Dsonar.projectName=npmlclient \
            -Dsonar.projectKey=npmclient \
            -Dsonar.sources=.
            -Dsonar.language=js
          """
        }
      }
    }
  }
}
```

```

        }
    }
}

stage('copy build folder and run build (client)') {
    steps {
        sh 'sudo npm run build'
        sh 'sudo scp -r /home/jenkins/workspace/nodejs_client/build/ lusy@10.211.55.22:/usr/share/nginx/html/'
        echo "Files copied"
    }
}
post {
    always {
        emailext body: 'Hey! This email is sent from Jenkins about your build status! 😊', subject: 'Build Status of Jenkins', to: '2ke20cs086_t@kleit.ac.in'
    }
}
=====
===== (script running )

```

e) Build status

Pipeline nodejs_deployment

Average stage times:
(Average full run time: ~2min 38s)

Declarative: Tool Install	git checkout	npm install	npm with SonarQube	copy build folder and run build (client)	Declarative: Post Actions
215ms	6s	2min 23s	756ms	678ms	5s

Build History: #52 Nov 01, 2023, 5:00 AM | trend | Filter builds... | #52 Nov 01, 05:00 No Changes

f) Out-put

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Appointment Booking' and displays the URL '10.211.55.22/index.html/api/user'. The page content includes a form for booking an appointment with fields for Patient Name, Phone Number, Doctor Name, Gender, Date, Age, Time, and a 'Book Appointment' button. Below this, a table lists patient appointments:

Patient	Status	Appointment	Phone	Doctor	Actions
User Img John Doe 28 yrs, Male	Consult	06:00 PM 2 Feb 2021	+91 987654321 Contact	Dr. Ananth	⋮
User Img Mukul Rao yrs, Male	Revisit	2 Feb 2021	+91 987654321 Contact	Dr. Ananth	⋮

g) Final mail

The screenshot shows a Gmail inbox. The left sidebar includes sections for Mail (99+), Chat, Spaces, Meet, and Labels. The main area shows an email from '2ke20cs086_t@kleit.ac.in' with the subject 'Build Status of Jenkins'. The email body reads: 'Hey! This email is sent from Jenkins about your build status! 😊'. The email was sent at '12:38 AM (0 minutes ago)'. Below the email are 'Reply' and 'Forward' buttons.