



**Academic Year: 2025-26**

**Semester: V**

**Class / Branch: TEIT**

**Subject: DevOps Lab**

**Name of Instructor: Ms. Seema Jadhav**

---

### **Experiment No. 4**

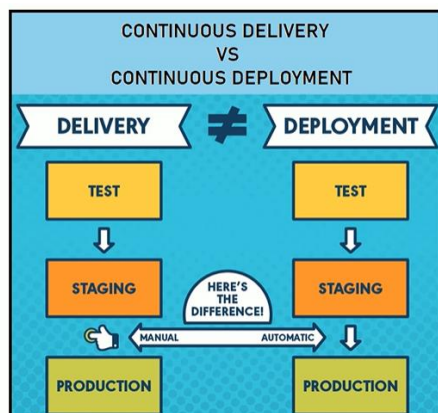
**Aim: To install Jenkins and configure parallel jobs for building pipeline.**

**Theory:** Jenkins is an open-source automation server widely used in Continuous Integration (CI) and Continuous Delivery (CD) pipelines. It allows developers to automate building, testing, and deployment of applications. One powerful feature of Jenkins is the ability to execute parallel jobs in a pipeline, which reduces build time and improves efficiency.

#### **Objectives**

- Understand the process of installing Jenkins.
- Learn how to configure Jenkins for pipeline execution.
- Implement parallel stages in Jenkins pipelines to run tasks simultaneously.

#### **Continuous Delivery vs Continuous deployment**



**Continuous Delivery :** The entire process is segregated into three phases:



Test, Staging and Production. The test and staging are automatic. But when you try to move your project from staging to production you need manual. So, this is called **continuous delivery**.

**Continuous Deployment:** All the phases test, staging and production is completely automatic. There is no manual intervention is required. So, this is called **continuous deployment**.

**Implementation:**

**Create 3 jobs: Test , Staging and Production**

Test→Freestyle→Ok

**Build Steps**  
Automate your build process with ordered tasks like

**Execute shell** ?

Command  
See [the list of available environment variables](#)

```
sleep 10  
echo "This is a Test phase"
```

Advanced ▾

Save Apply

Staging→freestyle→Ok



### Build Steps

Automate your build process with ordered tasks like

≡

Execute shell ?

Command

See [the list of available environment variables](#)

```
sleep 10  
echo "This is a Staging phase"|
```

Advanced ▾

Save Apply

Production → freestyle → ok

### Build Steps

Automate your build process with ordered tasks like

≡

Execute shell ?

Command

See [the list of available environment variables](#)

```
sleep 10  
echo "This is a Production Phase"
```

Advanced ▾

Save Apply

## Part I: Demo on concept of continuous delivery



In continuous delivery you can see the test job and staging is continuous.

**Test→Configure→Add post build action→build other project**

### Post-build Actions

Define what happens after a build completes, li

≡

**Build other projects**

?

Projects to build

Staging,|

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

Add post-build action ▾

Save

Apply

In Continuous delivery, test and staging would be automatic. But when you want to move from staging to production it should be manual.

Now goto staging job. In this staging job I have selected this option build other jobs manual step.

**Staging→Configure→Post-Build Actions → Build other projects (manual step)→production**

So I have created all my jobs and I have interlinked as per the continuous delivery.



## Staging

### Upstream Projects

Test

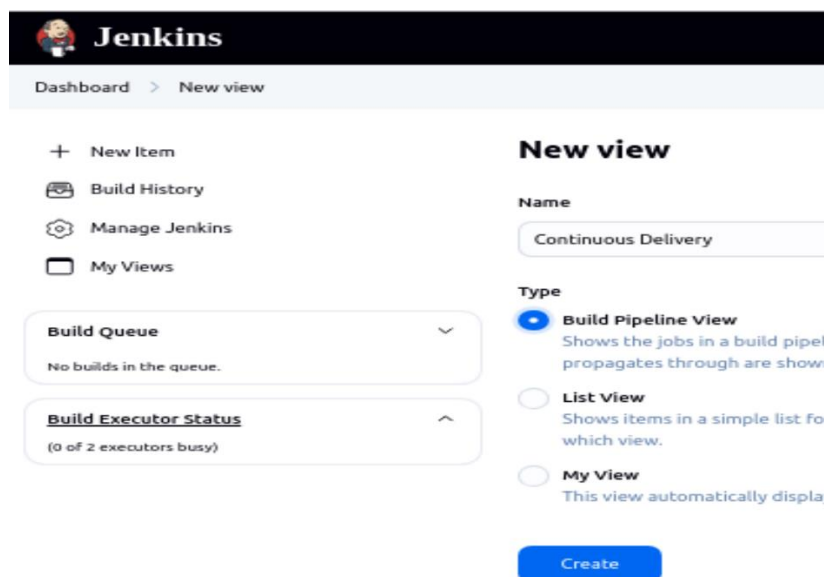
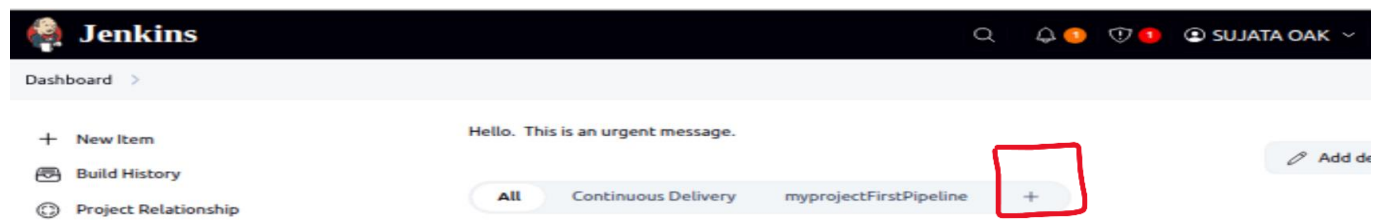
### Downstream Projects

production

### Permalinks

Goto Dashboard,

Now you need to click on this plus icon. New view: Name: continuous delivery.  
And type: build pipeline view and click on create.





and scroll down and look for your initial job.  
My initial job is the test job.

### Pipeline Flow

#### Layout

Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the jobs. This is the only out-of-the-box supported layout mode.

#### Upstream / downstream config

Select Initial Job ?

Test

### Trigger Options

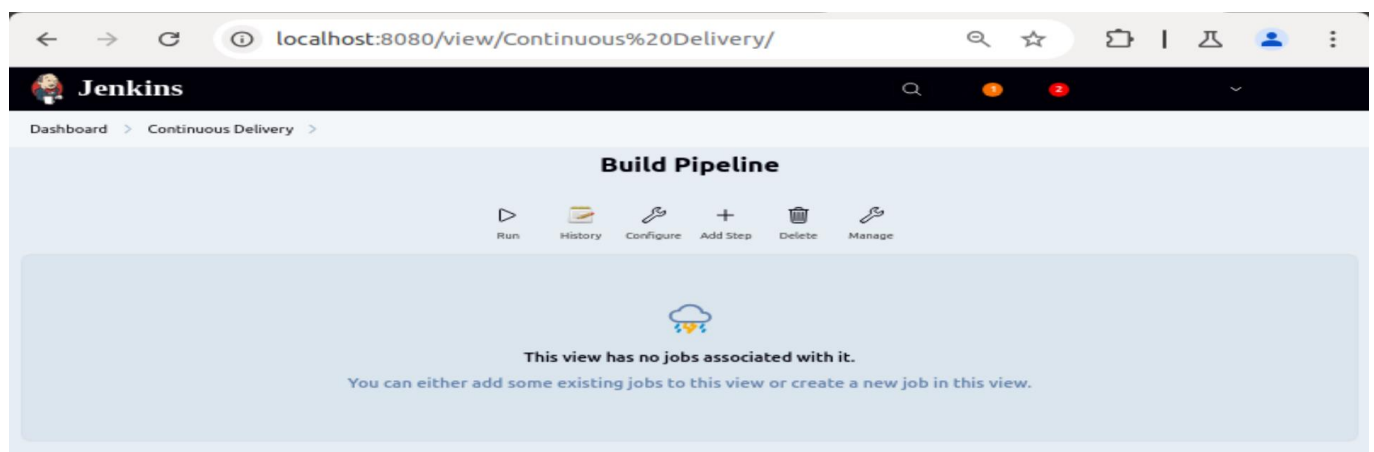
#### Build Cards

Standard build card

Use the default build cards

Save

Apply

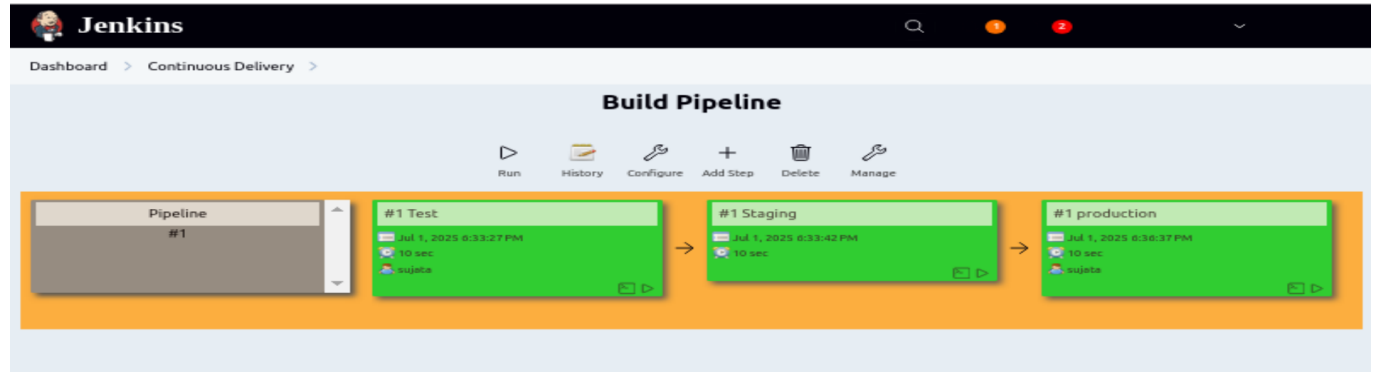




PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
Department of Information Technology  
(NBA Accredited)



Refresh it , still the production stage will not start Since it is a manual process in continuous delivery. We need to trigger it manually.



## Part II: Demo on concept of continuous deployment

All 3 stages has to be automated

**Goto Dashboard→Staging→Configure**

### Post-build Actions

Define what happens after a build completes,

≡

**Build other projects**

?

Projects to build

production,

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

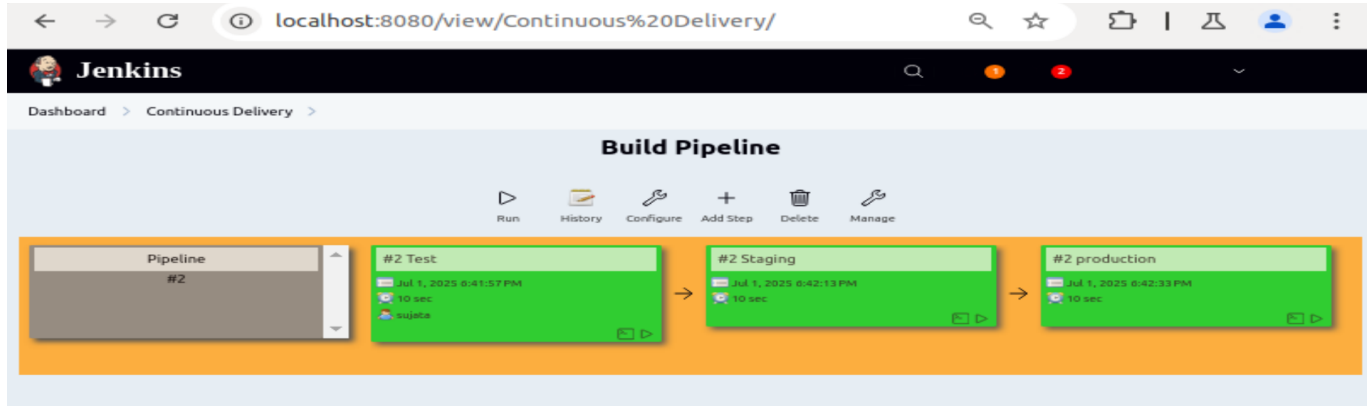
☐ Trigger even if the build fails

Add post-build action

Save

Apply





### Part III: Parallel Jobs in Jenkins Build Pipeline

In Test job → Configure

#### Post-build Actions

Define what happens after a build completes.

Build other projects ?

Projects to build

Staging,QA-test

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

Add post-build action

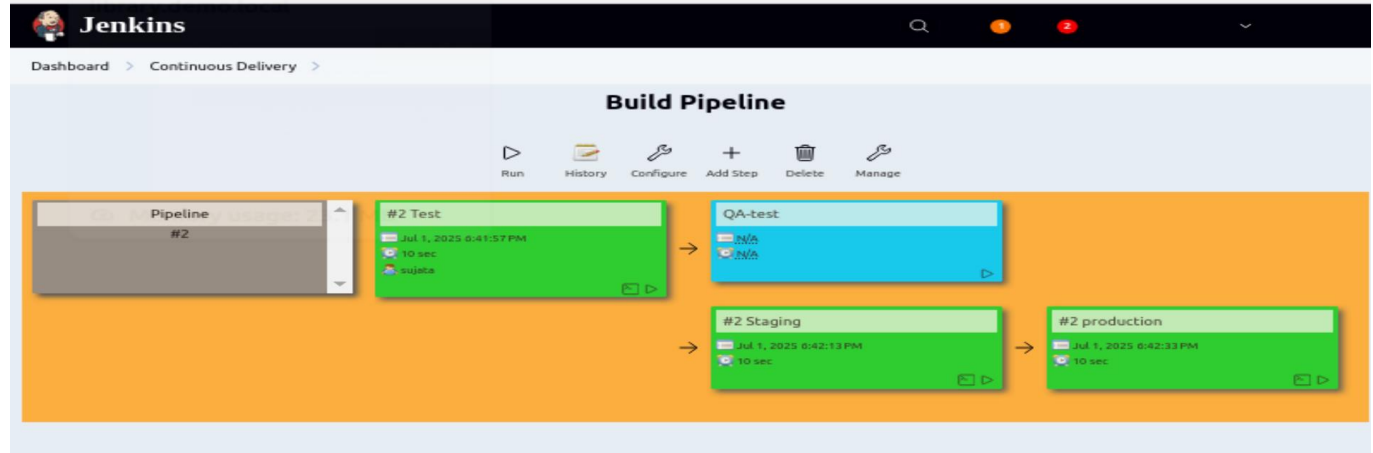
Save

Apply

Goto dashboard → Click on Continuous Delivery



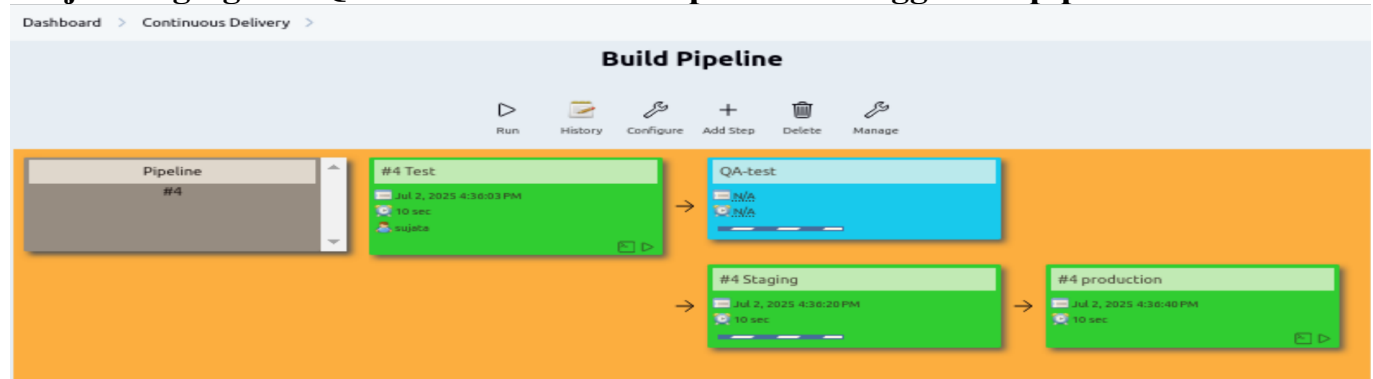
PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
Department of Information Technology  
(NBA Accredited)



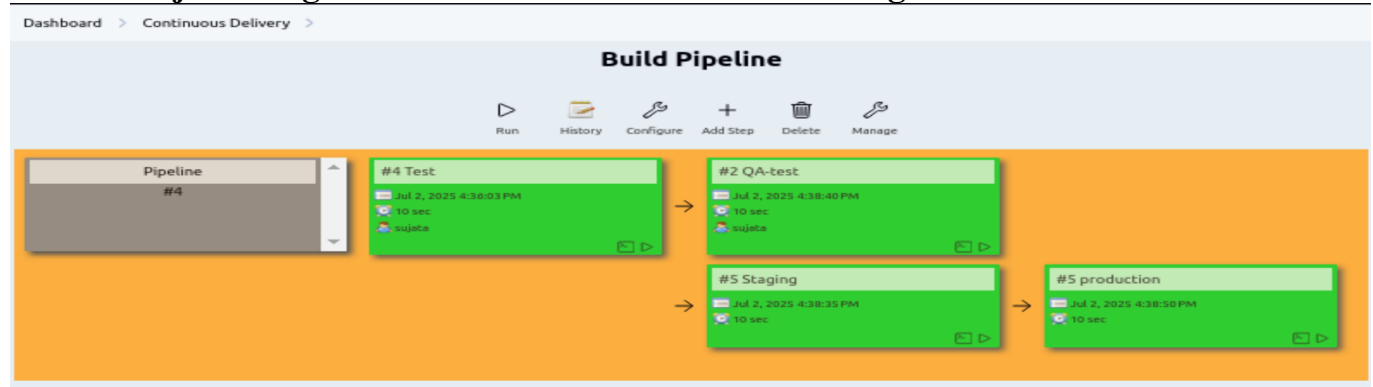
**Refresh and Run**

**1<sup>st</sup> job executed shown in green color**

**2<sup>nd</sup> job staging and QA-test will execute in parallel. Trigger the pipeline**



**At last all job will get executed and the color should be green.**





PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

**Department of Information Technology**

**(NBA Accredited)**



### **Conclusion:**

In this experiment we successfully learned installing jenkins and setting up parallel jobs in a pipeline which significantly improves CI/CD efficiency. This approach helps in faster feedback cycles, quicker releases, and better utilization of resources. With proper pipeline configuration, teams can ensure high productivity while maintaining quality.