

In Jenkins, **Upstream** and **Downstream** refer to the relationship between different jobs (projects), especially when they are configured to **trigger each other** as part of a pipeline or build chain.

Definitions:

Upstream Job:

- A job that **triggers another job**.
- Think of it as the **source** job in a pipeline.
- Example: If **Job A** triggers **Job B**, then **Job A is the upstream of Job B**.

Downstream Job:

- A job that is **triggered by another job**.
- It executes **after** the upstream job finishes.
- In the same example, **Job B is the downstream of Job A**.

Real-World Example:

You have a CI/CD pipeline with the following jobs:

Job A: Code Compile

Job B: Run Unit Tests

Job C: Deploy to Dev

Relationship:

- **Job A triggers Job B** → A is upstream of B, B is downstream of A
- **Job B triggers Job C** → B is upstream of C, C is downstream of B

Use Cases:

Use Case	Upstream Job Role	Downstream Job Role
CI Pipeline	Compile source code	Run tests, create artifacts
CD Pipeline	Package application	Deploy to servers
Automated Testing	Trigger test data generation	Run automation scripts
Build Promotion	Trigger after successful build	Promote to staging/prod

How to Configure:

1. From Jenkins Job Configuration:

- Open a job (e.g., **Job A**)
- Go to **Post-build Actions**
- Choose "**Build other projects**"
- Enter downstream job names (e.g., **Job B**)

2. Or from Downstream Job:

- Open **Job B**
- In "**Build Triggers**" section, check:
 - ☒ **Build after other projects are built**
 - Set **Job A** as the upstream

 Summary:

Term	Triggers	Triggered By
Upstream	Others	No
Downstream	No	Others

This relationship helps **create automated build chains**, ensuring that jobs are executed in the right sequence based on dependencies.

 Jenkins Job Chain: **JobA** → **JobB** → **JobC**

◆ Step-by-Step Execution Flow

◆ Step 1: GitHub Webhook or Manual Trigger Starts JobA

 What Happens:

- Jenkins detects a GitHub push event or you manually click “Build Now” for JobA.
- The **GitHub hook trigger for GITScm polling** is enabled, so:
 - Jenkins verifies that the webhook came from the correct repository.
 - Git plugin polls the repository and finds a new commit.

 Configuration:

- **SCM Git URL:** **<https://github.com/AbdulRehaman082493/mahalogin.git>**
- **Branch:** **[*/master](#)**

←

→

↻

Not Secure

18.119.163.94:8080/job/JobA/configure

☆

Finish update

Dashboard > JobA > Configuration

Configure

⚙️ General

🔑 Source Code Management

🕒 Triggers

🌐 Environment

☰ Build Steps

🏗️ Post-build Actions

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/AbdulRehaman082493/mahalogin.git

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master

Save

Apply

←

→

↻

Not Secure

18.119.163.94:8080/job/JobA/configure

☆

Finish update

Dashboard > JobA > Configuration

Configure

⚙️ General

🔑 Source Code Management

🕒 Triggers

🌐 Environment

☰ Build Steps

🏗️ Post-build Actions

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☒ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☒ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

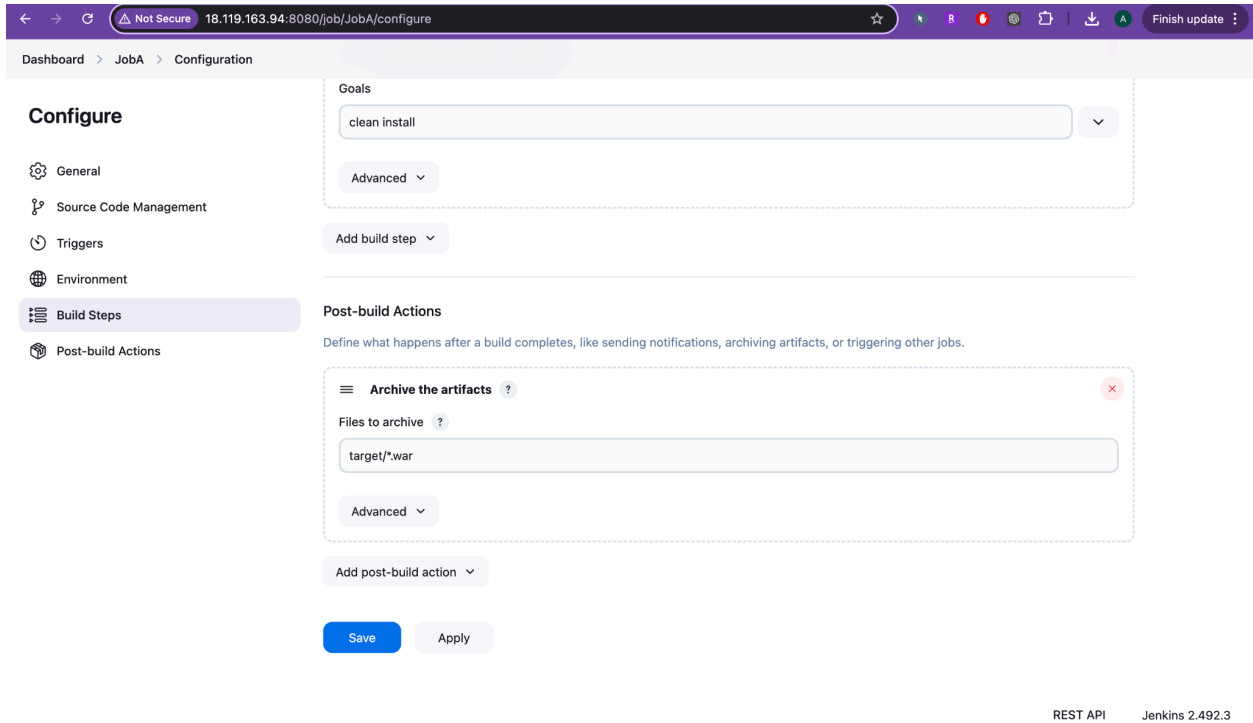
☐ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Save

Apply



♦ Step 2: JobA Runs Build & Archives Artifact

🔧 What Happens:

1. Jenkins clones the GitHub repo (checked out at master).
2. Maven build runs: mvn clean install

This compiles your code and produces a **.war** file.

3. Shell command: echo "This is first build"
4. Jenkins archives the output: **target/mahaLogin-2.0.war** is saved as a build artifact.
5. Post-build Action:

JobA triggers JobB (✅ as downstream) only if the build is stable.

← → ↻ Not Secure 18.119.163.94:8080/job/JobA/configure ☆

Dashboard > JobA > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Archive the artifacts ?

Files to archive ?

target/*.war

Advanced ▾

Build other projects ?

Projects to build

JobB,

☒ 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

REST API Jenkins 2.492.3

✓ You See:

- Last Successful Artifacts: mahaLogin-2.0.war
- Downstream Projects: JobB

♦ Step 3: JobB Starts Automatically After JobA Success

🔧 What Happens:

Dashboard > JobB > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

Execute shell ?

Command

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

```
echo "This is second build"
```

Advanced ▾

Add build step ▾

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action ▾

Save Apply

REST API Jenkins 2.492.3

- JobB is configured as a **downstream job** in JobA's "Post-build Actions."
- Once JobA finishes, Jenkins **automatically triggers JobB**.

JobB Configuration:

- Build Step: echo "This is second build"
- Post-build Action: Trigger **JobC** (next downstream job)

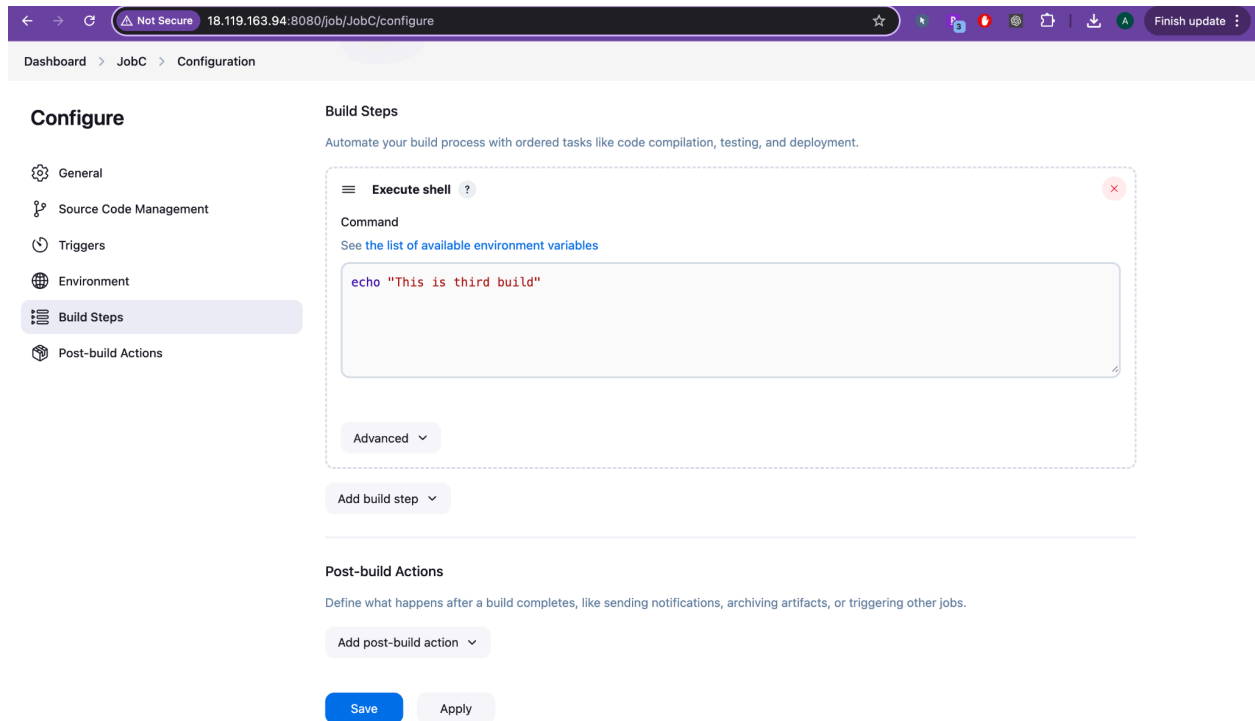
The screenshot shows the Jenkins web interface for configuring JobB. The left sidebar contains the 'Configure' section with sub-items: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions (which is highlighted). The main content area is titled 'Configure' and has a tab for 'Advanced'. Below this is a button 'Add build step'. The 'Post-build Actions' section is expanded, showing a description: 'Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.' A card titled 'Build other projects' is visible, with a text input field containing 'JobC,' and three radio button options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. Below the card is a button 'Add post-build action'. At the bottom are 'Save' and 'Apply' buttons. The bottom right corner of the page shows 'REST API' and 'Jenkins 2.492.3'.

✓ You See:

- **Upstream Project:** JobA
- **Downstream Project:** JobC

◆ **Step 4: JobC Starts Automatically After JobB Success**

🔧 What Happens:



- Jenkins automatically triggers **JobC** after JobB is complete and stable.

JobC Configuration:

- Build Step: echo "This is third build"

✓ You See:

- **Upstream Project:** JobB
- No downstream (final job in the chain)

 Timeline View (Example from Console Logs)

Time	Job	Action
05:31 AM	JobA	Triggered (via GitHub or manual)
05:31 AM	JobA	Build Success + Archived .war
05:32 AM	JobB	Auto-triggered by JobA
05:32 AM	JobB	Executed echo "This is second build"
05:32 AM	JobC	Auto-triggered by JobB
05:32 AM	JobC	Executed echo "This is third build"

Job Relationship Diagram

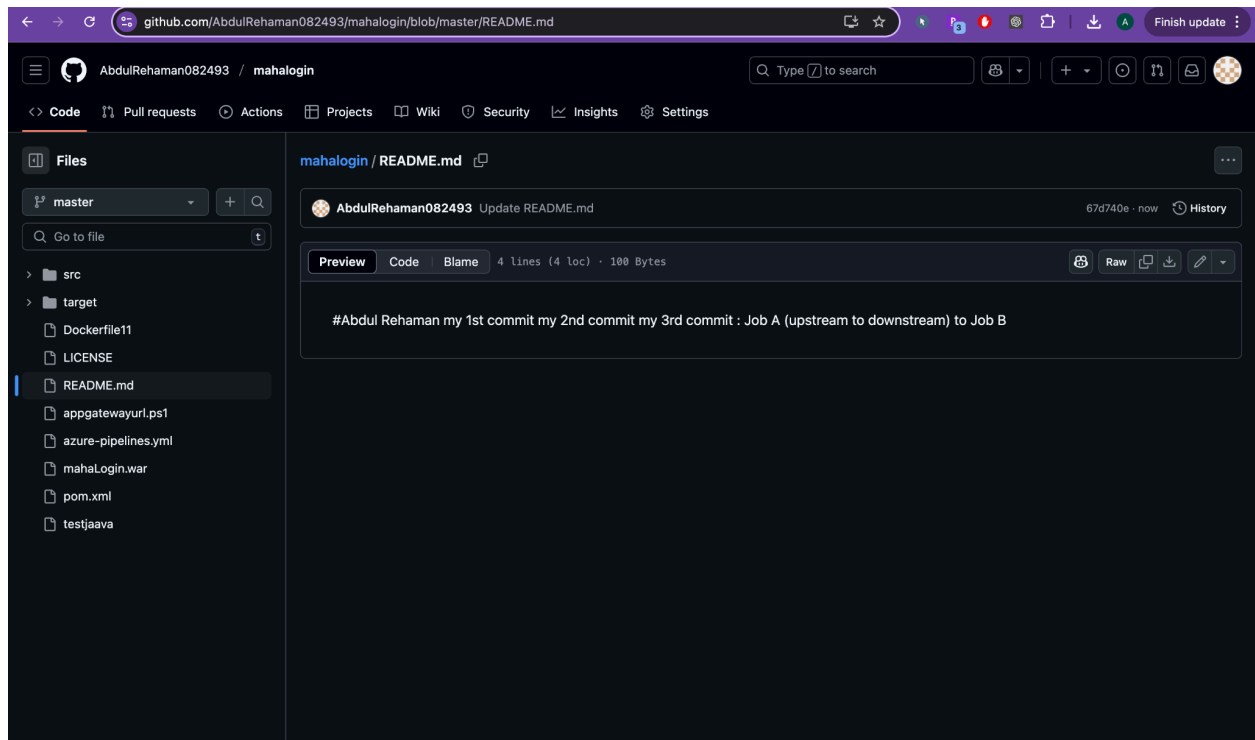
graph TD

A[JobA: Git Checkout + Build + Archive] --> B[JobB: echo 2nd build]

B --> C[JobC: echo 3rd build]

Summary of Purpose

Job	Function	Triggers
JobA	Pull from GitHub, build <code>.war</code> file	GitHub webhook/manual
JobB	Confirm 2nd step in chain (echo)	Triggered by JobA
JobC	Final confirmation (echo)	Triggered by JobB




← → ↻

Not Secure 18.119.163.94:8080

☆

🔍 📄 📁 📌 📄 📄 📄

👤 Finish update ⋮

 **Jenkins**

🔍

🔔 1

🛡️ 1

👤 Abdul Rehman Shaik ▾

🚪 log out

Dashboard >

+ New Item

📄 Build History

⚙️ Manage Jenkins

📌 My Views

Build Queue ▾

No builds in the queue.

Build Executor Status 0/2 ▾

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀️	Assignment05	1 hr 47 min #1	N/A	26 sec	▶
⋮	☀️	JobA	N/A	N/A	N/A	▶
⋮	☀️	JobB	N/A	N/A	N/A	▶
⋮	☀️	JobC	N/A	N/A	N/A	▶

Icon: S M L

⋮

REST API

Jenkins 2.492.3


← → ↻

Not Secure 3.142.197.132:8080/job/JobA/

☆

🔍 📄 📁 📌 📄 📄 📄

👤 Finish update ⋮

 **Jenkins**

🔍

🔔 1

🛡️ 1

👤 Abdul Rehman Shaik ▾

🚪 log out

Dashboard > JobA >

📄 Status

</> Changes

📁 Workspace

▶ Build Now

⚙️ Configure

🗑️ Delete Project

📄 GitHub Hook Log

✏️ Rename

Builds

Filter

Today

✓ #2 5:31 AM

✓ JobA

📦 Last Successful Artifacts

📄 mahaLogin-2.0.war 16.64 MiB view

Downstream Projects

✓ JobB

Permalinks

- Last build (#2), 7 min 36 sec ago
- Last stable build (#2), 7 min 36 sec ago
- Last successful build (#2), 7 min 36 sec ago
- Last completed build (#2), 7 min 36 sec ago

REST API

Jenkins 2.492.3

←

→

↺

⚠ Not Secure

3.142.197.132:8080/job/JobB/

☆

🔍

🔒

🔥

🌐

📦


📄

🔧

👤

Finish update

⋮

**Jenkins**

🔍

🔔

🔒

🔥

👤 Abdul Rehaman Shaik

⌵

🚪 log out

Dashboard > JobB >

📄

Status

</>

Changes

📁

Workspace

▶

Build Now

⚙

Configure

🗑

Delete Project

✏

Rename

Builds

⋮

🔗

🔍

Filter

🔗

Today

🟢

#1 5:32 AM

⌵

🟢

JobB

Add description

Upstream Projects

🟢 JobA

Downstream Projects

🟢 JobC

Permalinks

- [Last build \(#1\), 8 min 59 sec ago](#)
- [Last stable build \(#1\), 8 min 59 sec ago](#)
- [Last successful build \(#1\), 8 min 59 sec ago](#)
- [Last completed build \(#1\), 8 min 59 sec ago](#)

REST APIJenkins 2.492.3

←

→

↺

⚠ Not Secure

3.142.197.132:8080/job/JobC/

☆

🔍

🔒

🔥

🌐

📦


📄

🔧

👤

Finish update

⋮

**Jenkins**

🔍

🔔

🔒

🔥

👤 Abdul Rehaman Shaik

⌵

🚪 log out

Dashboard > JobC >

📄

Status

</>

Changes

📁

Workspace

▶

Build Now

⚙

Configure

🗑

Delete Project

✏

Rename

Builds

⋮

🔗

🔍

Filter

🔗

Today

🟢

#1 5:32 AM

⌵

🟢

JobC

Add description

Upstream Projects

🟢 JobB

Permalinks

- [Last build \(#1\), 10 min ago](#)
- [Last stable build \(#1\), 10 min ago](#)
- [Last successful build \(#1\), 10 min ago](#)
- [Last completed build \(#1\), 10 min ago](#)

REST APIJenkins 2.492.3