

# Experiment No.4

Name:- Bilal Mirje

Roll no:- 64

Div:-A

Batch :- A3

## 1. create jobs :

### New Item

Enter an item name

samplebuildJob1

Select an item type



#### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



#### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



#### Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



#### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

### Build Steps

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

#### Execute shell ?

Command

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

```
echo "Build process Successfully done...."
```

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

# New Item

Enter an item name

sampledeployJob1

Select an item type



**Freestyle project**

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

## Build Steps

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

≡ Execute shell ?

Command

See the list of available environment variables

echo"deploy job successfully executed..."

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

## New Item

Enter an item name

Select an item type



### Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



### Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



### Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



### Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

## Build Steps

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



Execute shell ?



Command

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

```
echo "Test Job Successfully Executed..."
```

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

# New Item

Enter an item name

SampleReleaseJob1

Select an item type



**Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

## Build Steps

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

Execute shell ?

Command

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

```
echo"Release job successfully completed..."
```

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

 Add description

All

+

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		samplebuildJob1	1 min 1 sec <a href="#">#3</a>	7 min 48 sec <a href="#">#2</a>	67 ms	
		sampledeployJob1	29 sec <a href="#">#2</a>	5 min 46 sec <a href="#">#1</a>	68 ms	
		SampleReleaseJob1	22 sec <a href="#">#2</a>	5 min 39 sec <a href="#">#1</a>	51 ms	
		SampleTestJob1	5.7 sec <a href="#">#3</a>	5 min 33 sec <a href="#">#2</a>	65 ms	

Icon:

S

M

L

...

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 [?](#)

Projects to watch

samplebuildJob1,

- ☒ Trigger only if build is stable
- ☐ Trigger even if the build is unstable
- ☐ Trigger even if the build fails
- ☐ Always trigger, even if the build is aborted
- ☐ 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 remote build agent [?](#)

Save

Apply

## samplebuildJob1

### Upstream Projects

 samplebuildJob1

### Downstream Projects


 samplebuildJob1


### Permalinks

- [Last build \(#3\), 2 min 5 sec ago](#)
- [Last stable build \(#3\), 2 min 5 sec ago](#)
- [Last successful build \(#3\), 2 min 5 sec ago](#)
- [Last failed build \(#2\), 8 min 52 sec ago](#)
- [Last unsuccessful build \(#2\), 8 min 52 sec ago](#)
- [Last completed build \(#3\), 2 min 5 sec ago](#)

### 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 

Projects to watch

sampledeployJob1,


☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Always trigger, even if the build is aborted

☐ 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 remote build file(s) 

Save

Apply

### 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 ?

Projects to watch

SampleTestJob1,

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Always trigger, even if the build is aborted

☐ 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 remote build agent ?

Save

Apply

### 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 ?

Projects to watch

SampleReleaseJob1,

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Always trigger, even if the build is aborted

☐ 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 remote build agent ?

Save

Apply

