**Question2:** How can you establish a connection between two or many distinct jobs (hosted on different Jenkins servers) to pass the parameters/upstream- downstream jobs etc. (Without using the master-slave connection in a secured way)

**Assumptions:** Jenkins version 2.1 and above. Jenkins Admin access is provided.

**Solution:**

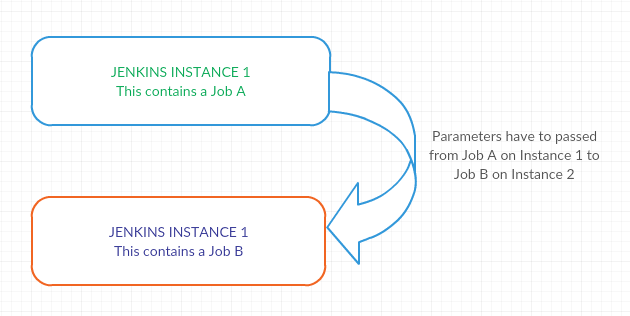
This can be achieved using **JENKINS PARAMETERIZED REMOTE TRIGGER PLUGIN**.

**Architecture:**

Number of Jenkins Instances: 2 (Names: Jenkins instance 1 and Jenkins instance 2)

Jenkins Instance 1: Job A configured on this Jenkins server.

Jenkins Instance 2: Job B configured on this Jenkins server. The parameters should be passed from Job A to Job B.



**Steps:**

1. Install the PARAMETERIZED REMOTE TRIGGER PLUGIN on INSTANCE 1 using the Manage Jenkins> Manage Plugins Section.
2. Since our INSTANCE 2 has a password based access, these have to be fed while we connect from Job A to Job B. Below are the steps:
3. On INSTANCE 1, go to the Manage Credentials Page and then add a credential used for JOB2.
4. Add the Hostname of JOB 2 and select the username and password authentication under the Credential box.
5. Specify the scope and Save.
6. Select Configure System (on INSTANCE 1) option after selecting Manage Jenkins option.  Under Parameterized Remote Trigger Configuration, click on Add. The field should automatically be filled up with the Display Name and Authentication. We need to provide the INSTANCE 2 hostname and port number in the Remote Address and port section.
7. JOB 1 Configuration: Below steps have to be executed in the Job configuration page:
8. Under the Build section, choose “Trigger a remotely parameterized job” option in the drop down menu of “Add build step” option.
9. We can choose our remote Jenkins server in “Select a remote host” field.
10. Type the name of the job i.e. “JOB2”. Specify the parameter i.e. “NAME” required by the “JOB2” job. Also, we need to use the value defined here in “name” parameter as $name.
11. BUILD THE JOB1. You can see that JOB1 is able to run the JOB2 by passing the parameters we injected in JOB1.

Note: The above steps can be done for multiple connections, and scheduling upstream and downstream jobs also.

**Question4:** Prepare a Jenkins pipeline job (DSL/Groovy Script) to satisfy the following CI/CD pipepline:

1. Send approval email to release Manager after the application is packaged.
2. Trigger Jenkins job on reception of the approval email for the above job.

**Assumptions:** Jenkins version 2.1 and above. Jenkins Admin access is provided.

**Solution:**

1. Install POLL MAILBOX TRIGGER PLUGIN.
2. Global tools configuration> Enter the SMTP details.
3. Create a JOB A which has to be triggered upon the receiving the email.
4. Create a pipeline from the new Item tab.
5. Select Configure pipeline
6. Insert below groovy script in the pipeline section to send out email post Job Completion.

stage 'Send mail'

node() {

mail bcc: '', body: 'Build Completed, please approve.', cc: '', from: 'jenkinsadmin', replyTo: '', subject: 'PLEASE APPROVE', to: 'btanandk@gmail.com'

}

1. Go to the JOB A configuration and then under the build trigger, choose POLL email trigger.
2. Enter \* \* \* \* \*. This will start the job as soon as the email is received.

**Question3:** Write a script or convey a proven mechanism to retrieve the URL of artifacts uploaded in Nexus/Artifactory via the Jenkins Job.

**Solution:** As post-build step in the build process Add a task "Archive the artifacts". And specify the files to be made accessible.

On the project dashboard page we can see a link to "Last successful artifacts".