# Job Content

Zuul jobs are implemented as Ansible playbooks. Zuul prepares the repositories used for a job, installs any required Ansible roles, and then executes the job’s playbooks. Any setup or artifact collection required is the responsibility of the job itself. While this flexible arrangement allows for almost any kind of job to be run by Zuul, batteries are included. Zuul has a standard library of jobs upon which to build.

# Working Directory

Before starting each job, the Zuul executor creates a directory to hold all of the content related to the job. This includes some directories which are used by Zuul to configure and run Ansible and may not be accessible, as well as a directory tree, under work/, that is readable and writable by the job. The hierarchy is:

**work/**

The working directory of the job.

**work/src/**

Contains the prepared git repositories for the job.

**work/logs/**

Where the Ansible log for the job is written; your job may place other logs here as well.

# Git Repositories

The git repositories in work/src contain the repositories for all of the projects specified in the required-projects section of the job, plus the project associated with the queue item if it isn’t already in that list. In the case of a proposed change, that change and all of the changes ahead of it in the pipeline queue will already be merged into their respective repositories and target branches. The change’s project will have the change’s branch checked out, as will all of the other projects, if that branch exists (otherwise, a fallback or default branch will be used). If your job needs to operate on multiple branches, simply checkout the appropriate branches of these git repos to ensure that the job results reflect the proposed future state that Zuul is testing, and all dependencies are present.

The git repositories will have a remote origin with refs pointing to the previous change in the speculative state. This means that e.g. a git diff origin/<branch>..<branch> will show the changes being tested. Note that the origin URL is set to a bogus value (file:///dev/null) and can not be used for updating the repository state; the local repositories are guaranteed to be up to date.

The repositories will be placed on the filesystem in directories corresponding with the canonical hostname of their source connection. For example:

work/src/git.example.com/project1

work/src/github.com/project2

Is the layout that would be present for a job which included project1 from the connection associated to git.example.com and project2 from GitHub. This helps avoid collisions between projects with the same name, and some language environments, such as Go, expect repositories in this format.

Note that these git repositories are located on the executor; in order to be useful to most kinds of jobs, they will need to be present on the test nodes. The base job in the standard library (see [zuul-base-jobs documentation](https://zuul-ci.org/docs/zuul-base-jobs/jobs.html" \l "job-base) for details) contains a pre-playbook which copies the repositories to all of the job’s nodes. It is recommended to always inherit from this base job to ensure that behavior.