# Common Operation

You can run any zuul process with the **-f** option to make it not daemonize and stay in the foreground, logging to your terminal. It’s a good idea at first to check for issues with your configuration. There’s also a **-d** option to engage verbose debug logging, but be careful in busy deployments as this can generate very large logs.

Before Zuul can run any jobs, it needs to load its configuration, most of which is in the git repositories that Zuul operates on. Start an executor to allow zuul to do that:

## **Scheduler**

### **Operation**

To start the scheduler, run zuul-scheduler. To stop it, run zuul-scheduler stop.

To start, simply run:

**zuul-scheduler**

### **Reconfiguration**

Most of Zuul’s configuration is automatically updated as changes to the repositories which contain it are merged. However, Zuul must be explicitly notified of changes to the tenant config file, since it is not read from a git repository. Zuul supports two kinds of reconfigurations.

The full reconfiguration refetches and reloads the configuration of all tenants. To do so, run zuul-scheduler full-reconfigure. For example this can be used to fix eventual configuration inconsistencies after connection problems with the code hosting system.

## **Merger**

The Zuul Merger is a separate component which communicates with the main Zuul server. Its purpose is to speculatively merge the changes for Zuul in preparation for testing. The resulting git commits also must be served to the test workers, and the server(s) running the Zuul Merger are expected to do this as well. Because both of these tasks are resource intensive, any number of Zuul Mergers can be run in parallel on distinct hosts.

### **Operation**

To start the merger, run zuul-merger.

In order to stop the merger and under normal circumstances it is best to pause and wait for all currently running tasks to finish before stopping it. To do so run zuul-merger pause.

To stop the merger, run zuul-merger stop. This will wait for any currently running merge task to complete before exiting. As a result this is always a graceful way to stop the merger. zuul-merger graceful is an alias for zuul-merger stop to make this consistent with the executor.

## **Executor**

### **Operation**

To start the executor, run zuul-executor.

There are several commands which can be run to control the executor’s behavior once it is running.

To pause the executor and prevent it from running new jobs you can run zuul-executor pause.

### **Ansible and Python**

As noted above, the executor runs Ansible playbooks against the remote node(s) allocated for the job.

## **Web Server**

### **Operation**

To start the web server, run zuul-web. To stop it, kill the PID which was saved in the pidfile specified in the configuration.

## **Finger Gateway**

### **Operation**

To start the finger gateway, run zuul-fingergw. To stop it, kill the PID which was saved in the pidfile specified in the configuration.