### Setting Up Laravel Queue Worker with systemd

1. Create a systemd Service File Create a service file for your Laravel queue worker.

 Commands

**#sudo bash**

**#cd /etc/systemd/system/**

**# nano laravel-worker.service**

Type the following text in the nano text editor

**[Unit]**

**Description=Laravel Queue Worker**

**After=network.target**

**[Service]**

**User=www-data**

**Group=www-data**

**Restart=always**

**ExecStart=/usr/bin/php /var/www/html/unza\_payroll/artisan queue:work --sleep=3 --tries=3**

**WorkingDirectory=/var/www/html/unza\_payroll**

**SyslogIdentifier=laravel-worker**

**[Install]**

**WantedBy=multi-user.target**

**Press control + x to exit**

**Type yes to save the file**

 **Press enter to close the file**

1. Reload systemd to Register the Service After creating the service file, reload systemd so it can recognize the new service:

Commands

**#sudo systemctl daemon-reload**

1. Start and Enable the Laravel Worker Service Start the service and enable it to run at boot:

**#sudo systemctl start laravel-worker**

**#sudo systemctl enable laravel-worker**

1. Check the Status of the Service Use this command to check the status and logs of your Laravel queue worker:

Command

**#sudo systemctl status laravel-worker**

Logs for the service will be accessible via journalctl, or you can see real-time logs with:

**#sudo journalctl -u laravel-worker -f**

Why Use systemd Instead of Supervisor?

Native to Linux: systemd is widely supported on most Linux distributions and often comes pre-installed, meaning fewer dependencies.

 More Control: systemd provides better control over service lifecycle and logging integration with journalctl.

Resource Efficiency: systemd tends to use fewer resources compared to Supervisor, which can be useful on resource-constrained environments.

Using systemd for the Laravel queue worker is equivalent to using Supervisor but with some added benefits in resource efficiency and native support on Linux systems.

### **Check Directory Permissions**

Ensure that the /var/www/html/unza\_payroll/storage/app/payslips directory is writable by the user running your PHP application (usually www-data on Ubuntu/Debian or apache on CentOS/Red Hat).

Command

#sudo chown -R www-data:www-data /var/www/html/unza\_payroll/storage/app/payslips

#sudo chmod -R 755 /var/www/html/unza\_payroll/storage/app/payslips