



Laravel

Laravel 


Laravel






Laravel 

1. **Database** - 
2. **Redis** - 
3. **Beanstalkd** - 
4. **Amazon SQS** - AWS 
5. **RabbitMQ** - 



1.

```
#  Redis  
composer require predis/predis
```

  `.env` 

```
QUEUE_CONNECTION=redis  
REDIS_HOST=127.0.0.1  
REDIS_PASSWORD=null  
REDIS_PORT=6379
```

2.

```
php artisan make:job ProcessPodcast
```



```
<?php

namespace App\Jobs;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Bus\Dispatchable;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Queue\SerializesModels;

class ProcessPodcast implements ShouldQueue
{
    use Dispatchable, InteractsWithQueue, Queueable, SerializesModels;

    protected $podcast;

    /**
     * [ ] [ ] [ ] [ ] [ ] [ ]
     */
    public function __construct($podcast)
    {
        $this->podcast = $podcast;
    }

    /**
     * [ ] [ ] [ ] [ ]
     */
    public function handle()
    {
        // [ ] [ ] [ ] [ ] ...
        logger()->info('Processing podcast: '.$this->podcast->title);
    }
}
```



1. `ProcessPodcast`

```
use App\Jobs\ProcessPodcast;
use App\Models\Podcast;

// dispatch
ProcessPodcast::dispatch($podcast);

// dispatch 60seconds
ProcessPodcast::dispatch($podcast)->delay(now()->addSeconds(60));

// onQueue
ProcessPodcast::dispatch($podcast)->onQueue('processing');
```

2. `Queueable` worker

```
# artisan worker
php artisan queue:work

# queue:work
php artisan queue:work --queue=high,default --tries=3

# queue:work
php artisan queue:work --daemon
```

`Queueable`

1. `Queueable`

```
// Queueable
public $queue = 'high';
```

`Queueable`

```
ProcessPodcast::dispatch($podcast)->onQueue('high');
```

2.

```
php artisan queue:failed-table  
php artisan migrate
```

```
public function failed(Throwable $exception)  
{  
    //  ...  
}
```

```
php artisan queue:failed
```

```
php artisan queue:retry all
```

3.

Horizon Redis

```
composer require laravel/horizon  
php artisan horizon
```

1. Supervisor

Supervisor

```
sudo apt-get install supervisor
```

□□□□□

/etc/supervisor/conf.d/laravel-worker.conf □

```
[program:laravel-worker]
process_name=%(program_name)s_%(process_num)02d
command=php /path/to/your/project/artisan queue:work --sleep=3 --tries=3
autostart=true
autorestart=true
user=www-data
numprocs=8
redirect_stderr=true
stdout_logfile=/path/to/your/project/storage/logs/worker.log
```

□□ Supervisor□

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start laravel-worker:*
```

2. □□□□□□

1. □□□□□□□□ numprocs □□□□ CPU□□□□ 2-3□□
2. □□ CPU□□□□□□□□□□□□
3. □□ --memory □□□□□□□□
4. □□□□ worker □□□□□□

```
; □ Supervisor□□□□□
stopwaitsecs=3600
```

□□□□□□□□□□

□□	□□	□□	□□□□
Database	□□□□□□□□	□□□□	□□□□□□□□
Redis	□□□□ Laravel□□□□	□□ Redis□□□	□□□□□□□□
Beanstalk	□□□□□□□□	□□□□□□	□□□□□□□□
SQS	□□□□□□□□	AWS□□□□□□□□	AWS□□□□
RabbitMQ	□□□□□□□□	□□□□	□□□□□□□□



1.

-
-
-

2.

```
//   
public $tries = 3;  
public $backoff = [60, 120, 300];
```

3.

```
//   
$batch = Bus::batch([  
    new ProcessPodcast($podcast1),  
    new ProcessPodcast($podcast2),  
])->then(function (Batch $batch) {  
    //  ...  
})->catch(function (Batch $batch, Throwable $e) {  
    //  ...  
})->dispatch();
```

4.

-
-

Laravel