



# Web

Web (MQ)

## 1. RabbitMQ

- - 
  - (AMQP, MQTT, STOMP )
  - 
  -
- - Kafka
  - Erlang
- Web

## 2. Apache Kafka

- - 
  - 
  -
- - 
  - RabbitMQ
- Web

## 3. Redis Streams

- - Redis
  -



- `redis` `redis`

`redis`

# PHP `redis`

`PHP` `redis`

PHP

PHP `redis`

## `redis` PHP `redis`

### 1. Redis Streams + PHP `redis`

- `redis` `redis`
  - PHP `Redis` `phpredis` `redis`
  - `redis`
  - `redis`
- `redis` `redis`
  - `redis`
  - `redis`
  - `Redis` `redis`
- `redis` `redis`

```
$redis = new Redis();
$redis->connect('127.0.0.1', 6379);

// redis
$redis->xAdd('mystream', '*', ['field' => 'value']);

// redis
$messages = $redis->xRead(['mystream' => '$'], 1, 1000);
```

### 2. RabbitMQ + PHP AMQP `php-amqp`

- `redis` `redis`
  - `AMQP` `PHP`
  - `redis`
  - `redis` PHP `redis` (php-amqplib)
- `redis` `redis`



```
$job = $pheanstalk->watch('testtube')->ignore('default')->reserve();
echo $job->getData();
$pheanstalk->delete($job);
```



## 4. Kafka + PHP (rdkafka)

- **PHP**
  - **librdkafka**
  - **php-rdkafka**
- **PHP**
  - **librdkafka** **php-rdkafka**
  - **librdkafka**

## 5. Amazon SQS/Azure Service Bus

- **PHP**
  - **PHP**
  - **PHP**
  - **AWS/Azure**



1. **PHP**
  - **PHP** → **SQS/Service Bus**
  - **PHP** → **2**
2. **PHP**
  - **<10 / PHP** → **Redis Beanstalkd**
  - **10 -100 / PHP** → **RabbitMQ**
  - **100 / PHP** → **Kafka**
3. **PHP** (**PHP**)
  - **PHP** → **RabbitMQ**

- [Redis](#) [Beanstalkd](#)

# PHP [Queue](#)

1. [Queue](#) [PHP](#)
  - [supervisor](#)
  - [Laravel Queue](#)
2. [Queue](#)
  - [amqp](#) [redis](#) [rdkafka](#)
  - [Queue](#)
3. [Queue](#)
  - [Laravel](#): [Redis](#) [Beanstalkd](#) [SQS](#)
  - [Symfony](#): [Messenger](#) [MQ](#)

[Queue](#) ( [Queue](#) )

<a href="#">Queue</a>	<a href="#">Queue</a> ( <a href="#">Queue</a> / <a href="#">Queue</a> )	<a href="#">Queue</a>	PHP <a href="#">Queue</a>
Redis	50,000+	<a href="#">Queue</a>	★★★★★
Beanstalkd	20,000	<a href="#">Queue</a>	★★★★☆
RabbitMQ	10,000-20,000	<a href="#">Queue</a>	★★★★☆
Kafka	100,000+	<a href="#">Queue</a>	★★★☆☆

[Queue](#)

PHP Web [Queue](#) [Redis](#) [RabbitMQ](#)





```
<?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

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

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

## `Queueable`

### 1. `Queueable`

```
// high
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

