

# Implementing Serverless PHP

## Under the hood of OpenWhisk

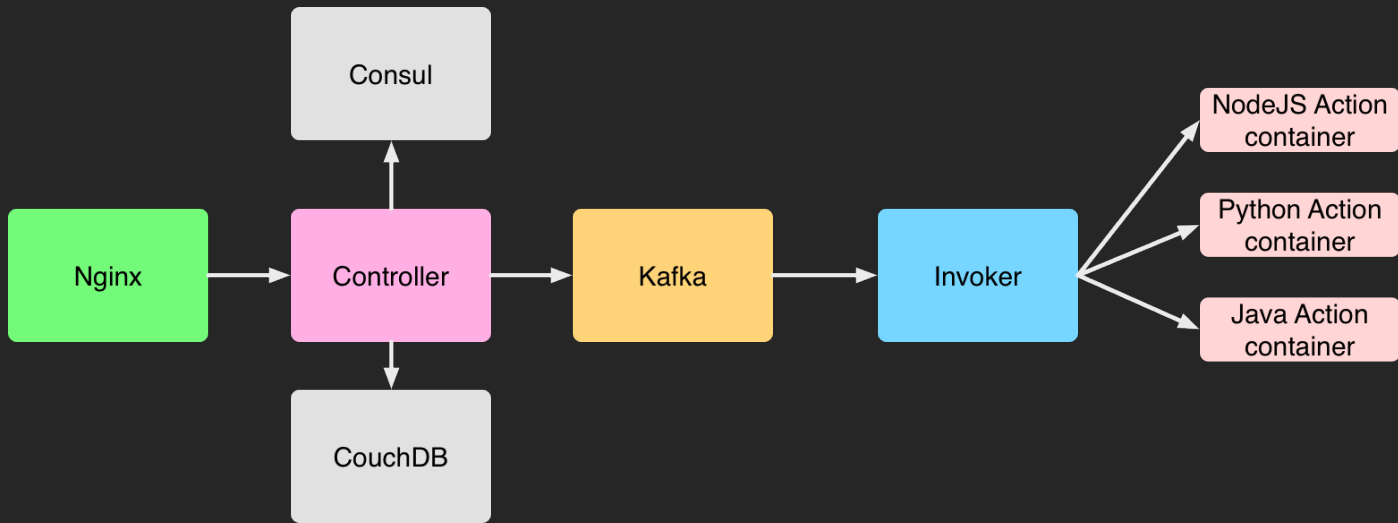
Rob Allen, 19FT

# The goal: PHP native actions

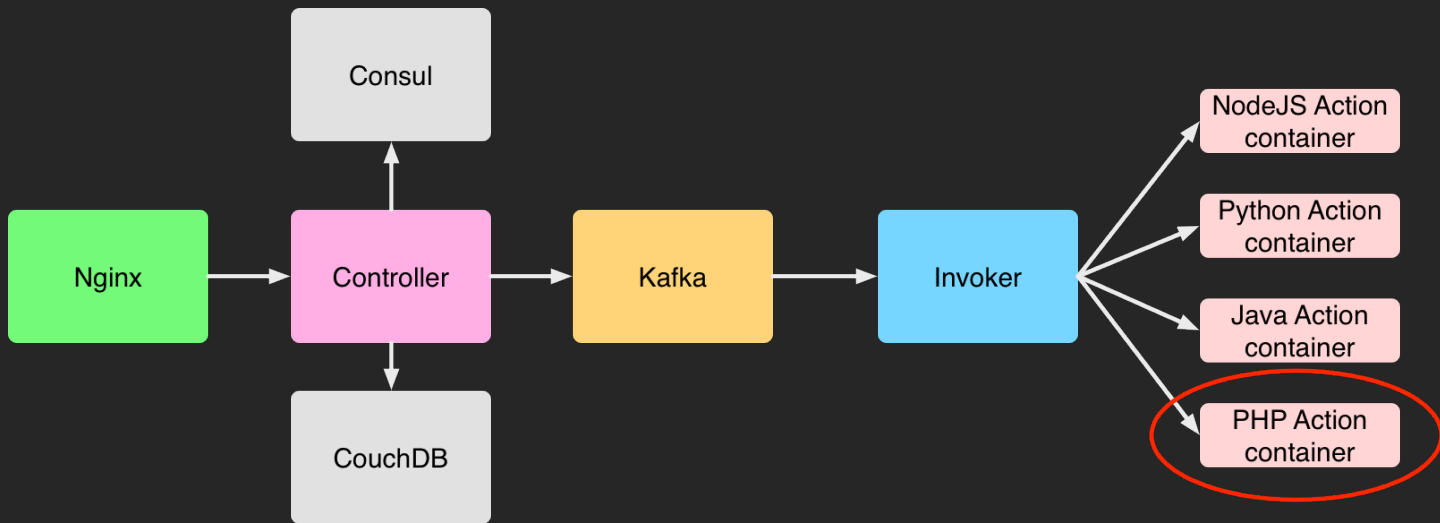
hello.php:

```
1 <?php
2 function main(array $args) : array
3 {
4     $name = $args["name"] ?? "World";
5     return [ "msg" => "Hello $name" ];
6 }
```

# OpenWhisk's Architecture

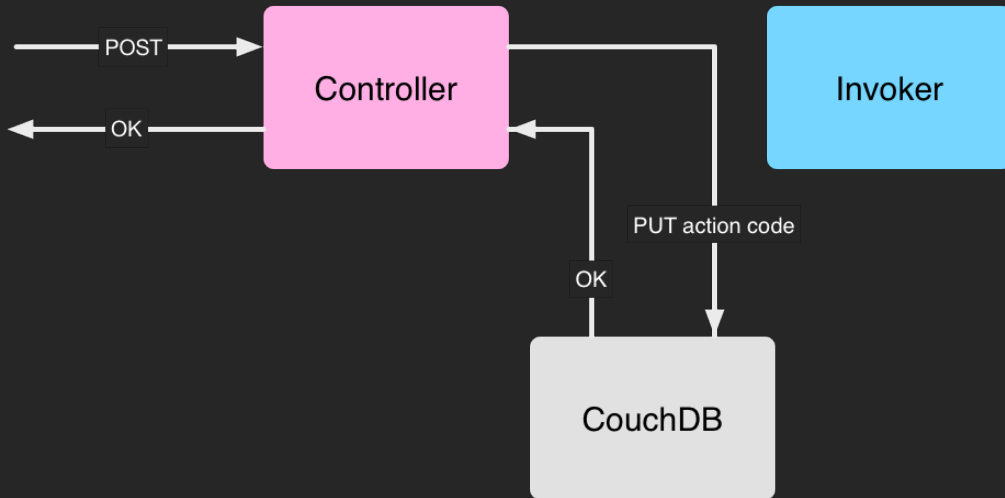


# OpenWhisk's Architecture



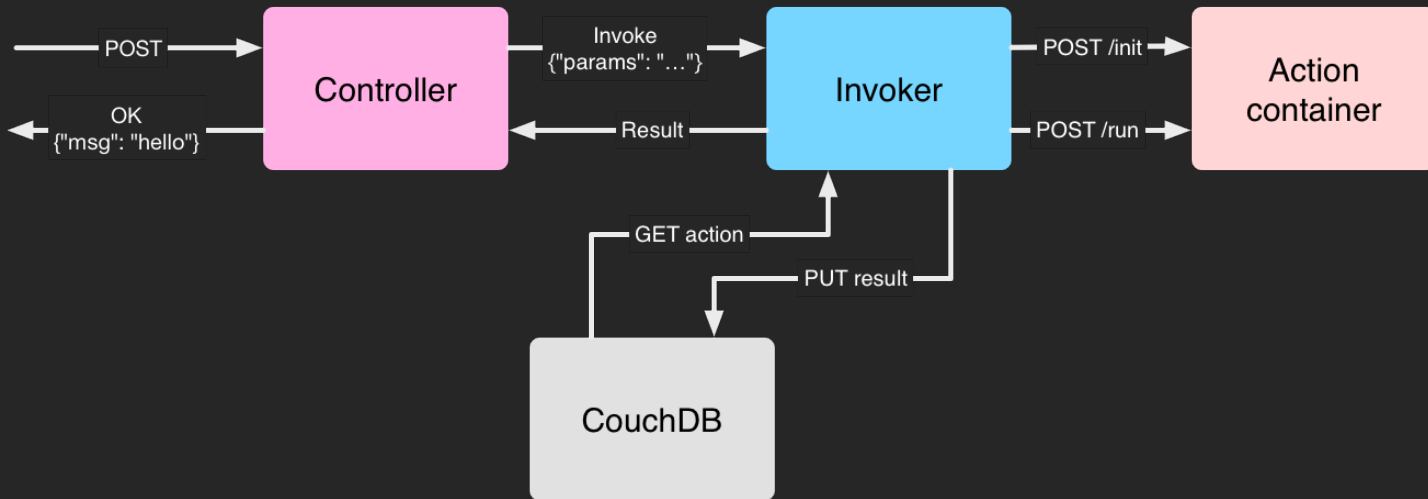
# Create an action

wsk action create hello



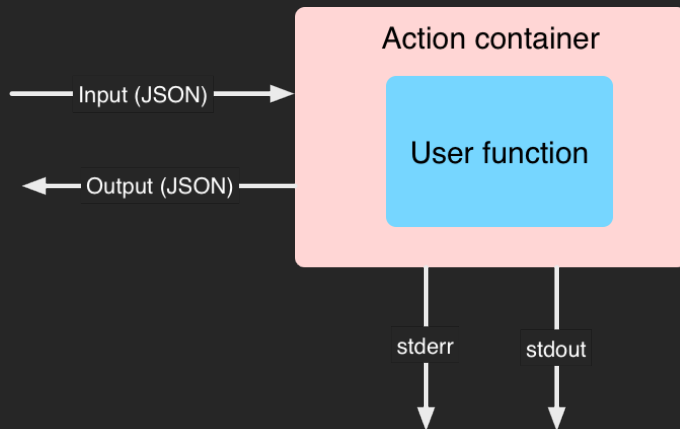
# Invoke an action

wsk action invoke hello -r

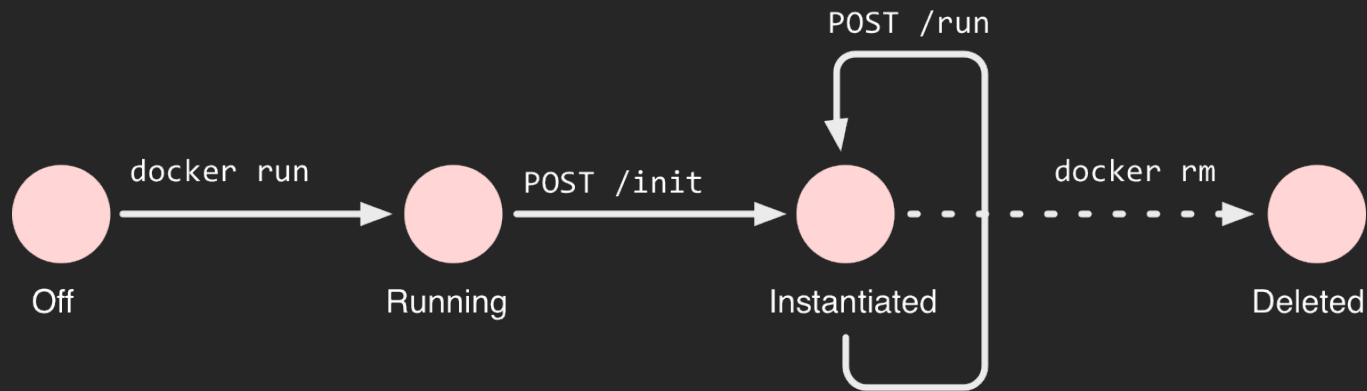


# Action containers

- Hosts the user-written code
- Controlled via two end points
  - `/init` - on creation
  - `/run` - on invocation
- Logs `stdout` and `stderr`



# Action container lifecycle





# Action Container API

	POST /init	POST /run
input	<pre>{   "value": {     "name" : "helloPHP",     "main" : "main",     "binary": false,     "code" : "&lt;?php ...",   } }</pre>	<pre>{   "value": {     "name" : "Rob",   } }</pre>
output	<pre>{ "OK": true }</pre>	<pre>{ "msg": "Hello Rob" }</pre>

# Writing a PHP action container

We need:

- A container
- Code to handle endpoints (`router`)
- Code to execute the action code (`runner`)

# Dockerfile

```
FROM php:7.1-alpine
```

```
# copy required files
```

```
ADD router.php /action
```

```
ADD runner.php /action
```

```
# Start webserver on port 8080
```

```
EXPOSE 8080
```

```
CMD [ "php", "-S", "0.0.0.0:8080", "/action/router.php" ]
```

# router.php

```
1 <?php
2 if ($_SERVER['REQUEST_URI'] == '/init') {
3     $result = init();
4 } elseif ($_SERVER['REQUEST_URI'] == '/run') {
5     $result = run();
6 }
7
8 /* send response */
9 header('Content-Type: application/json');
10 echo json_encode((object)$result);
```

# router.php: init()

```
1 function init()
2 {
3     $post = file_get_contents('php://input');
4     $data = json_decode($post, true)['value'];
5
6     file_put_contents('index.php', $data['code']);
7
8     $config = ['function' => $data['main'], 'file' => 'index.php'];
9     file_put_contents('config.json', json_encode($config));
10
11     return ["OK" => true];
12 }
```

# router.php: run()

```
1 function run()
2 {
3     $config = file_get_contents('config.json');
4     $args = file_get_contents('php://input'), true);
5
6     list($code, $out, $err) = runPHP("php runner.php $config", $args);
7
8     $pos = strrpos($stdout, PHP_EOL) + 1;
9     $lastLine = trim(substr($stdout, $pos));
10
11     file_put_contents("php://stderr", $stderr);
12     file_put_contents("php://stdout", $stdout);
13
14     return $lastLine;
15 }
```

# router.php: runPHP()

```
1 function runPHP(string $cmd, string $stdin = '') : array
2 {
3     $process = proc_open($cmd,
4         [['pipe', 'r'], ['pipe', 'w'], ['pipe', 'w']], $pipes,
5         __DIR__, $_ENV);
6
7     fwrite($pipes[0], $stdin);
8     $stdout = stream_get_contents($pipes[1]);
9     $stderr = stream_get_contents($pipes[2]);
10
11     $returnCode = proc_close($process);
12     return [$returnCode, $stdout, $stderr];
13 }
```

# runner.php

Runs the user's code in a separate process

```
1 <?php
2 $config = json_decode($argv[1], true);
3 $filename = $config['file'] ?? 'index.php';
4 $functionName = $config['function'] ?? 'main';
5
6 $args = json_decode(file_get_contents('php://stdin'), true);
7
8 require '/action/vendor/autoload.php';
9 require "/action/src/$filename";
10
11 $result = $functionName($args);
12
13 echo json_encode((object)$result);
```



# Hello world in PHP

hello.php:

```
1 <?php
2 function main(array $args) : array {
3     $name = $args["name"] ?? "World";
4     return [ "msg" => "Hello $name" ];
5 }
```

# Hello world in PHP

```
$ wsk action create helloPHP hello.php --kind php:7.1  
ok: updated action hello
```

```
$ wsk action invoke -r helloPHP -p name Rob  
{  
  "msg": "Hello Rob"  
}
```

# Resources

## PHP on OpenWhisk:

- <http://github.com/apache/incubator-openwhisk/pull/2415>
- <https://github.com/akrabat/ow-php-ftime>

## OpenWhisk:

- <http://www.openwhisk.org>
- <https://medium.com/openwhisk>

Thank you!

Rob Allen ~ @akrabort