Steelmesh Documentation

Release 0.9.8

Damon Oehlman

June 02, 2014

Contents

Steelmesh is a Node.js development / deployment platform that is designed to work in conjuction with systems like CouchDB (i.e. distributed document stores).

CHAPTER 1

Why Use Steelmesh?

Steelmesh has primarily been designed for hosting "behind the firewall" node applications in, dare I say it, enterprise environments. It really hasn't been designed for your next gazillion user startup or other such webscale deployments.

What Steelmesh does do a bang up job on though, is making working with horizontally scaled, homogeneous clusters, really easy. Heck it's almost fun.

CHAPTER 2

Find Out More

2.1 Steelmesh Installation

2.1.1 Install Dependencies

Before installing steelmesh, you will require the following:

- *Node.js (0.6.x)*
- *CouchDB (>= 1.1)*
- *Redis*

Node.js (0.6.x)

CouchDB (>= 1.1)

Redis

2.2 Steelmesh Design Principles

Steelmesh is designed to be *different* to other Node.js hosting options.

While other Node hosting solutions are primarily geared towards hosting a single (and hopefully popular) node application, Steelmesh aims to make hosting multiple applications on the same system (physical or virtualized) a more manageable task.

2.2.1 Application Multi-tenancy

Current techniques and processes for building Node.js applications focus on creating a webserver, binding to particular routes, and then binding that webserver instance to a particular port on the machine.

Shown below, for instance, is a trivial example that uses the excellent Express.js framework:

```
var app = express.createServer();

app.get('/', function(req, res){
    res.send('Hello World');
});
```

```
app.listen(3000);
```

If you are hosting your application on your own server, then you might be selecting port 80 as the port to bind to, or potentially hosting your node application behind nginx and proxying traffic through to your application.

While these solutions are ok, older and much more boring systems have provided application multi-tenancy for quite some time, and allow you to deploy apps A,B & C into a single server instance.

In actual fact, Express also provides this functionality by allowing express server instances to be mounted within other server instances. Steelmesh uses these application mountpoints to allow you to run multiple applications within the one instance.

2.2.2 Application Distribution / Deployment

2.2.3 Graceful Restarts

When application updates are intercepted, at the moment Steelmesh performs a restart to ensure that applications are correctly loaded. This is done gracefully by downloading the new application files, spawning new workers using the new application files, and once the new workers are available, instructing the old workers to shutdown.

2.3 Steelmesh Components

Steelmesh is made up of three primary components:

2.3.1 Steelmesh Application Server

The server is the primary component of Steelmesh, and is responsible for initiating the other supporting processes. Additionally, Steelmesh is written using the cluster module of Node, and thus forks a worker process to serve incoming HTTP requests on port `6633` (by default).

2.3.2 Steelmesh Monitor Process

The Monitor process is responsible for distributing both application and data updates in the Steelmesh 1.0 platform. This process is handled thanks to ChangeMachine which responds to change notifications from CouchDB and takes appropriate action.

Core application updates are captured through listing for *_changes* against the steelmesh db configured on Couch. When an application update is detected, the monitor simply sends a message to the steelmesh core to restart the Steelmesh App Server. As application updates are processed on startup of Steelmesh, and workers are restarted gracefully this works well to ensure traffic is served while keeping applications up-to-date.

In addition to application updates, applications that specify CouchDB data connections are also monitored for changes, and when an update is detected these are passed through to the registered change handler within the application.

It should be noted, however, that this functionality will not be implemented in Steelmesh 2.0 as this will be considered the responsibility of an individual application to monitor updates.

2.3.3 Steelmesh Dashboard

The Steelmesh Dashboard was developed as a part of Steelmesh 1.0 to assist organisations that aren't able to interact with CouchDB directly. At this stage, the Dashboard is very much a beta component of Steelmesh.

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*