

# we write about the things we build and the things we consume



written by Tom McAdam on 22 October 2014 in engineering, MongoDB

## mongo and aws availability zones

If you follow this blog you'll know that we use both MongoDB and AWS. We span multiple availability zones—roughly equivalent to data centres in non-AWS speak—but recently made a change to our set-up which necessitated our systems being availability zone aware.

### the story so far

We already had databases in multiple availability zones for resilience reasons of course (you'd be mad not to), but we only had a couple. With one having the role of the primary database, the other of secondary, we simply used Mongo's `ReadPreference.primary()` and `secondaryPreferred()` options. For the unfamiliar, these allow you to, respectively, connect to the primary database, or a secondary if one is available.

### what's new?

We added an additional secondary database in another region. So we now needed to connect to the closest database in each region. Thankfully, Mongo makes this easy and flexible using `tags`. These are key/value pairs you can associate with each database instance, and then a connecting client can specify a tag or set of tags to match to a database.

All we need to do, therefore, is tag each database with its region, and tell each client to prefer a database with a tag matching its region.

### tagging the databases

We'll be automating this through `Puppet` in the near future but in the meantime it's as easy as running something along the lines of

```
conf = rs.conf() conf.members[n].tags = { "region": "eu-west-1a" } rs.reconfig(conf)
```

on the primary database, replacing `n` with the appropriate array index of each host.

### clients

The Mongo Java client's `ReadPreference.secondaryPreferred()` allows you to provide the tags you'd like a database to have, so we just add a property with the required tags. `Puppet`, with its `ec2 facts`, comes to the rescue to do this automatically. It's as easy as

```
mongo.db.tag: "az:%{:ec2_placement_availability_zone}"
```

in our hiera configuration file. Behind the scenes, Puppet uses the ec2 metadata endpoints to obtain each host's availability zone and substitute it in.

This set-up has a few benefits:-

1. We don't incur unnecessary inter-availability zone data transfer costs
2. Things will be a bit faster
3. Things will be more reliable: in the old set-up, an availability zone outage may lead to some requests failing as components connected to the remaining database, but with availability-zone alignment, if the whole availability zone goes down only in-flight requests to that availability zone would be affected.

### **the happy couple: puppet and aws**

We were pleasantly surprised by the presence of the EC2 facts in Puppet by default; it made our lives a lot easier. If you've any tips on making AWS, Mongo or Puppet work in harmony do drop us a line.