|  |
| --- |
| Home Start Realty |
| Securing MySQL |

To: Brendan McAlister

From: Chad Ballay

Some background on MySQL’s fit for our goals:

* Open Source so as we develop inhouse expertise we can avoid vendor lockin.
* Oracle offers MySQL as a service solutions within their Cloud
* Feature set meets our needs. (See discussion about GAP analysis)
* Data replication ability will allow us to maintain reliability targets
* Data backup/restore will also allow us to look into shifting to a hybrid solution of cloud/on-prem to even a completely on-premise hosting with cloud as a hot spare/backup solution.

We’ve selected MySQL for the DB tier of the new infrastructure. In doing so we’ll have to audit and address some security concerns specific to MySQL. We don’t have the inhouse ability to start off with it secure. I’m advocating that we spend money on forestalling this part of the problem for now. Here is the overview of my plan that’ll help mitigate our lack of depth resulting in a catastrophe.

**Tools and Best Practices**

* Standardize on using DBForge for administration. (Will help prevent human error)
* Budget for hourly snapshots and build a pipeline for backups to be encrypted and stored in another provider. (Maintain 6 month’s worth of images in cold storage.)
* Put out a request for proposal now for a third party DB audit in 6 months. We don’t have the expertise in house yet so we’ll have to pay for it. This will be in addition to using Oracle in the following step. This will be to also train our nascent inhouse team.
* Start off with using Oracle’s inhouse support staff for our initial design/allocation. The money spent here will be worth it over the long run due to Subject Matter Expertise. It’ll be a large bill so we’ll need to review the deliverable documentation expectations.

**Stage 1**

We’ll start off with Oracle’s MySQL as a service since we really don’t have the staffing yet to handle the fulltime support needs. This’ll help cover the majority of our basic needs for securing the install since they have multiple best practices that we can follow. For this stage we’ll need to spend large sums of money mainly on the maintenance cost and cloud costs. (<https://www.oracle.com/mysql/>)

**Stage 2**

For this stage we’ll want to develop a backup/restore process that’ll allow us to build a test environment in either in Infrastructure as a Service environment or here on-premise. This will reduce our cloud cost at the expense of having developed local expertise. The Oracle hosted solution will continue to be our record of truth, but this stage will be building a safer/cheaper answer for testing/provisioning. At this point I would recommend we invest in some of the thirdparty automated scanning tools. We’ll learn from what they capture as well as what we learn from building our test environment.

**Stage 3**

If we reach this stage it’ll either be that we’ve grown enough that the Oracle costs run high enough to warrant building our own. Or the latency/experience has gotten bad enough to need to be addressed. For this we’ll need to migrate to a IaaS or an on-premise solution as the record of truth with the Oracle instance being scaled back to seeing less traffic and perhaps being a hot spare or just an offsite solution. I don’t forsee us reaching this point or if so in any appreciable short term. But gearing to this point will be the target so as we can develop the inhouse knowledge for securing MySQL.