# Title: DevOps: Setting up new DB shard

Steps to setup new DB shard

## Step-by-step guide Infra setup

1. **Copy the parameter group** of the current latest shard and create a new one for master.
2. Replicate the same parameter group for slave
3. **Create the master and slave DB** instances (slave)
4. Wait for RDS instances to boot and master & slave should be in sync.
5. Boot **DR slave**, and change **dr parameter group**, and also set **auto backup**  
    and run this command to **enable cdc** for baikal -

* call mysql.rds\_set\_configuration('binlog retention hours', 10);

1. **Add DNS** record(CNAME) for master, slave instances and public DNS record for DR slave (baikal slave)
2. Test connectivity
3. ADD MySQL users, baikal, nocuser, dbuser, migration, proxysql user (writer, reader),mluser, segmentuser with correct passwords and privileges (compare from other RDSs and get passwords from devops team)  
     
   Generate a new password and use it for both mluser and segmentuser  
   Store the root password for Database in AWS Secret Manager   
     
   us-east-1        - freshsales-infra-us  
   eu-central-1    - freshsales-infra-eu  
   ap-south-1       - freshsales-infra-in  
   ap-southeast-2 - freshsales-infra-au

## Product DB setup

1. Follow this doc and create a data-migration job with freshsales master branch and your custom fsales-stacks branch,  
   this will create a data migration pod ([EKS Production - Deployment & Migration Documentation](https://confluence.freshworks.com/pages/viewpage.action?pageId=324489090))
2. Update config/database.yml with new shard configurations  
   Add new shard configuration with root user and update the latest shard user to root instead of migration
3. Verify DB connectivity via mysql client.
4. Run the following command to create the database -  
   **bundle exec rake db:prepare\_shard**[<shard\_number>] --trace
5. Run the following command to seed global data in the shard -  
   **bundle exec rake db:populate\_integrations**[<shard\_number>] --trace
6. Update configurations
   1. proxysql-config in AWS secret manager
   2. create new custom fsales-stacks branch and update db creds in freshsales app & migration and cpq app & migration (ex - <https://github.com/freshdesk/fsales-stacks/pull/712>).
7. Perform EKS node rotation so that the proxysql daemonset starts using new configuration -  
   1. Create new version in launch templates of each node group
   2. Update launch template version from node group page (Please confirm with team before performing rotation, as this will restart all our pods / data migration jobs)
8. Verify connectivity using proxysql host and creds from data-migration pods
9. Merge the custom fsales-stacks branch
10. Add the secrets to FWSS
11. Perform Deployment in both Freshsales and CPQ
12. Informthe below list of teams about the new shard addition -
    1. Freddy-Freshales #fsa-freddy
    2. Microservice team - Segment & Import #fsa-microservices and tag  [Khushi Singla](https://confluence.freshworks.com/display/~khushi.singla) [Arul Pragasam](https://confluence.freshworks.com/display/~arul.pragasam) [Prabu Padmanathan](https://confluence.freshworks.com/display/~prabu.padmanathan) [Kiriti Madakasira](https://confluence.freshworks.com/display/~kiriti.madakasira)
    3. Baikal (raise the below ticket - <https://lighthouse.freshservice.com/support/catalog/items/834>)
    4. Analytics Reconciliation job setup (#analytics-crm-reconciliation)
13. Use new data migration pod to create preprovision accounts in the shard -

* **bundle exec rake db:seed\_pre\_provisioned\_accounts[shard\_number] --trace**  
  **or**  
  **Seeding one plan alone**
* AccountPreProvisioningWorker.new.perform(shard\_name: 'shard\_11122', type: AccountPreProvisioningWorker::TYPE[:manual], sales\_360\_plan\_wise\_limit: {sales360\_clc\_enterprise: 1})

1. **Create test account** on new shard :  
   1. add redis key for the domain  
      $redis\_restriction.hset(FsRedis::RedisKeys::SHARD\_SELECTIVE\_SIGNUP,'subdomain','shard\_1')  
      $redis\_restriction.expire(FsRedis::RedisKeys::SHARD\_SELECTIVE\_SIGNUP, 1.day)
   2. Perform signup
2. Once QA gives approval on the new shard account
3. Configure **freshping** alert for the shard - <https://noc-team.freshping.io/reports?check_id=547082>
4. Update the "**latest\_shard**" value in respective stack settings in fsales-stacks branch (ex - <https://github.com/freshdesk/fsales-stacks/pull/716>)

Add index on updated\_at column [DevOps: Adding index on updated\_at columns in baikal slave RDS](https://confluence.freshworks.com/pages/viewpage.action?pageId=191635561)

## Step-by-step guide - EKS Devstacks

    1. update the database.yml in stacksettings file - eg.  <https://github.com/freshdesk/fsales-stacks/commit/eaf3240835190e32413db6e1880257b34ce2224b>  
    2. Perform fast deploy on the stack  
    3. **Update datebase.yml inside pod and create shard**  
        i. podexec fsales <stackname> eg. podexec fsales i5  
        ii. create tmp directory=>   **mkdir /data/app/tmp**

        iii. vi config/database.yml. 

**database.yml**

staging:  
 adapter: mysql2  
 encoding: utf8  
 reconnect: true  
 port: 3306  
 host: docker-mysql.civuakczrhjv.us-east-1.rds.amazonaws.com  
 database: e8\_master   
 username: root  
 password: 7c41A#9A3PP4I  
 slave:  
 username: fs\_readonly\_user  
 shards:  
 shard\_1:  
 database: e8\_shard1   
 not\_a\_shard: false  
 host: docker-mysql.civuakczrhjv.us-east-1.rds.amazonaws.com  
 username: root  
 password: 7c41A#9A3PP4I  
 slave:  
 username: root  
 password: 7c41A#9A3PP4I  
 shard\_2:  
 database: e8\_shard2   
 not\_a\_shard: false  
 host: docker-mysql.civuakczrhjv.us-east-1.rds.amazonaws.com  
 username: root  
 password: 7c41A#9A3PP4I  
 slave:  
 username: root  
 password: 7c41A#9A3PP4I

      iv. **bundle exec rake db:prepare\_shard**[2] --trace (took around 20min)  
      v. **bundle exec rake db:populate\_integrations**[2] --trace  
   4. perform fast\_deploy in eks stack  
  
   5. **Create test account** on new shard :

* podexec fsales <stackname>
* bundle exec rails c
* add redis key for the domain
* $redis\_restriction.hset(FsRedis::RedisKeys::SHARD\_SELECTIVE\_SIGNUP,'subdomain','shard\_2')  
  $redis\_restriction.expire(FsRedis::RedisKeys::SHARD\_SELECTIVE\_SIGNUP, 1.day)
* make a signup

## Related articles

* Page:
* [DevOps: Setting up new DB shard](/display/freshsales/DevOps%3A+Setting+up+new+DB+shard)
* Page:
* [DevOps: Rebalance Mailbox thread across workers](/display/freshsales/DevOps%3A+Rebalance+Mailbox+thread+across+workers)