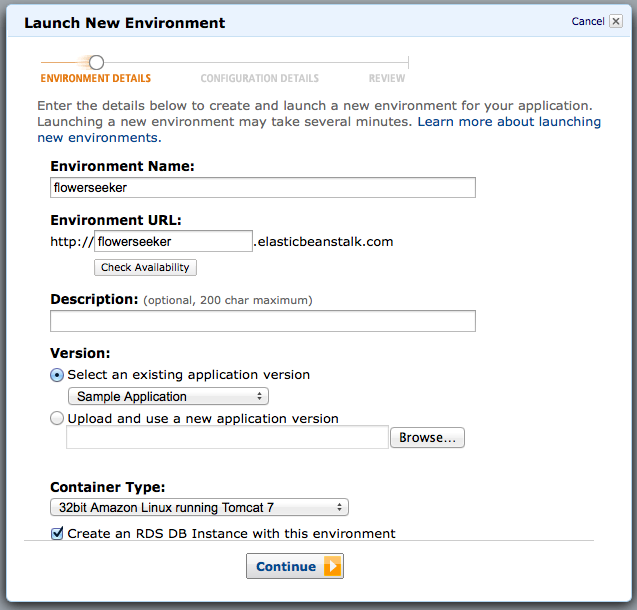
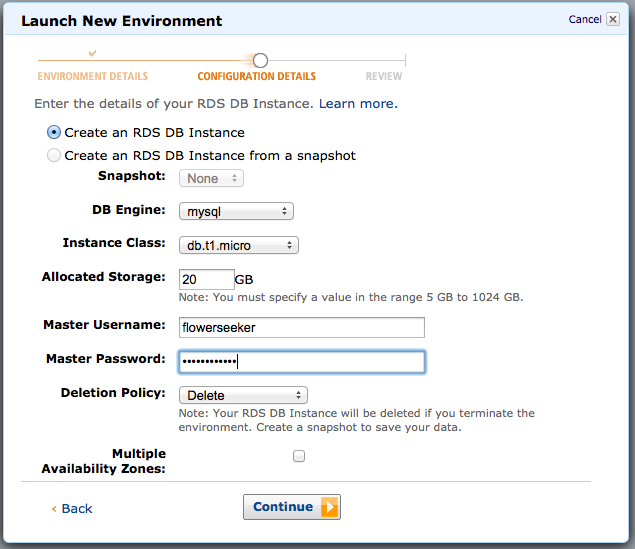
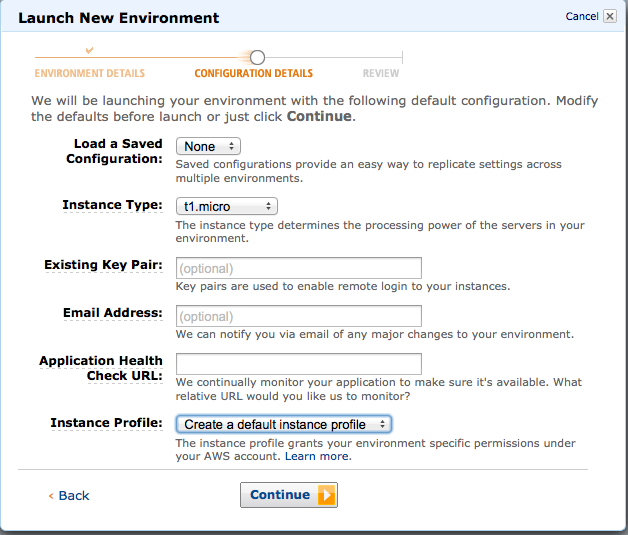
After signing up, go to the ElasticBeanstalk console <https://console.aws.amazon.com/elasticbeanstalk/home>

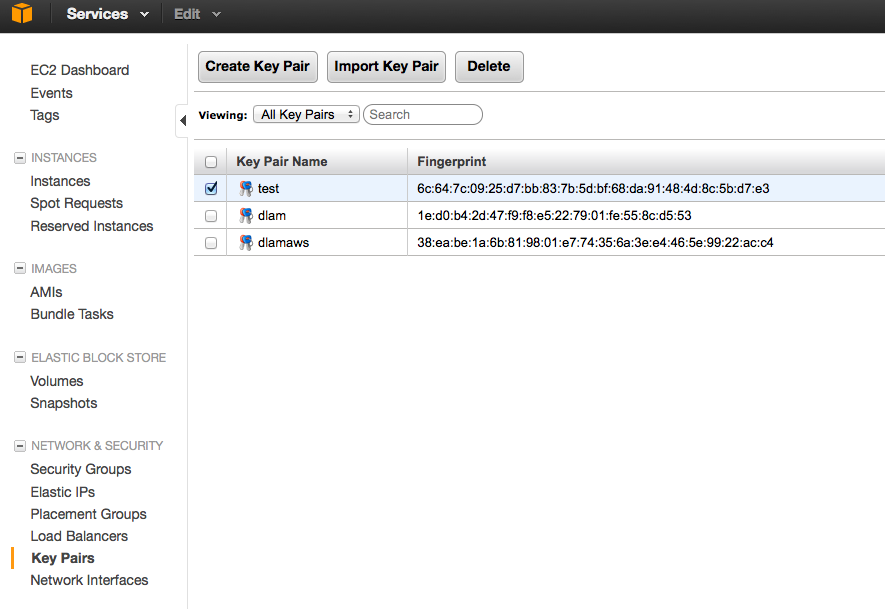
Let it create the sample Tomcat 7 application, wait till it finishes.

Launch New Environment, you can try 64 bit, but I had problems with it

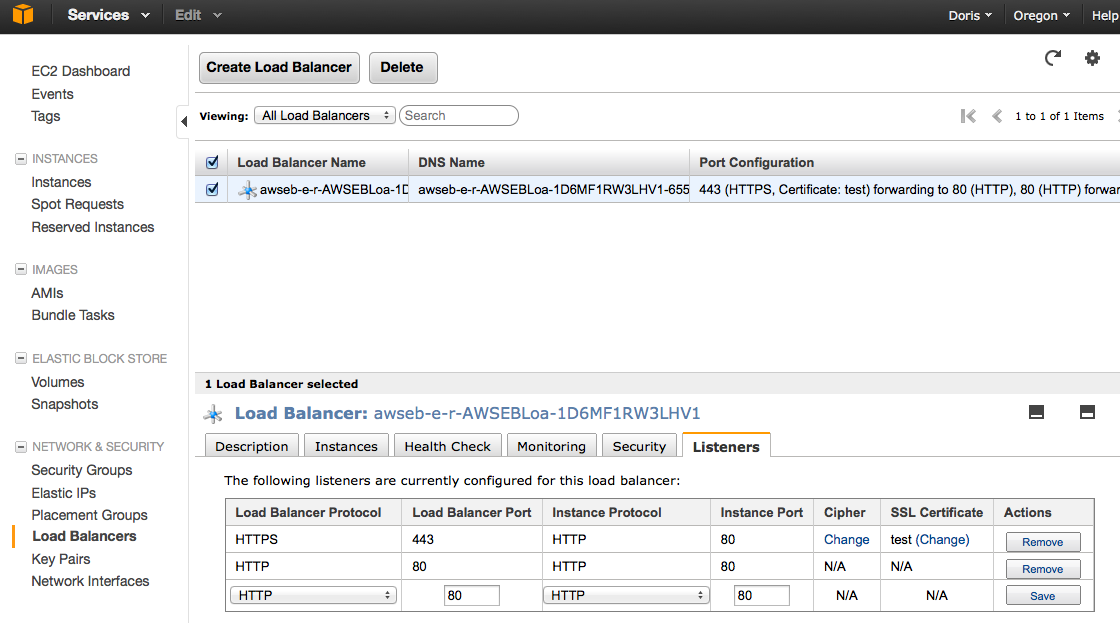


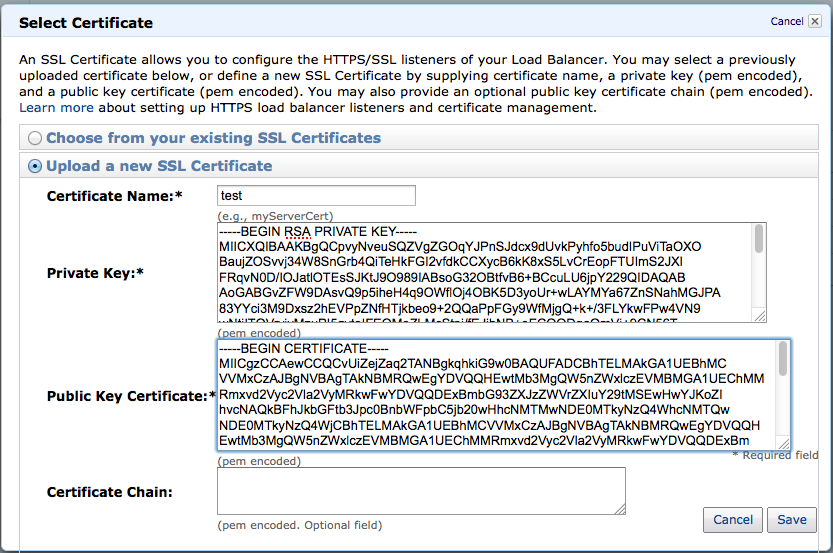
Continue and finish, let it do stuff, wait a while until it says successfully running version Sample Application.

Go to EC2 console and create key pair so you can access the instance using ssh in the future. Save the private key somewhere safe.

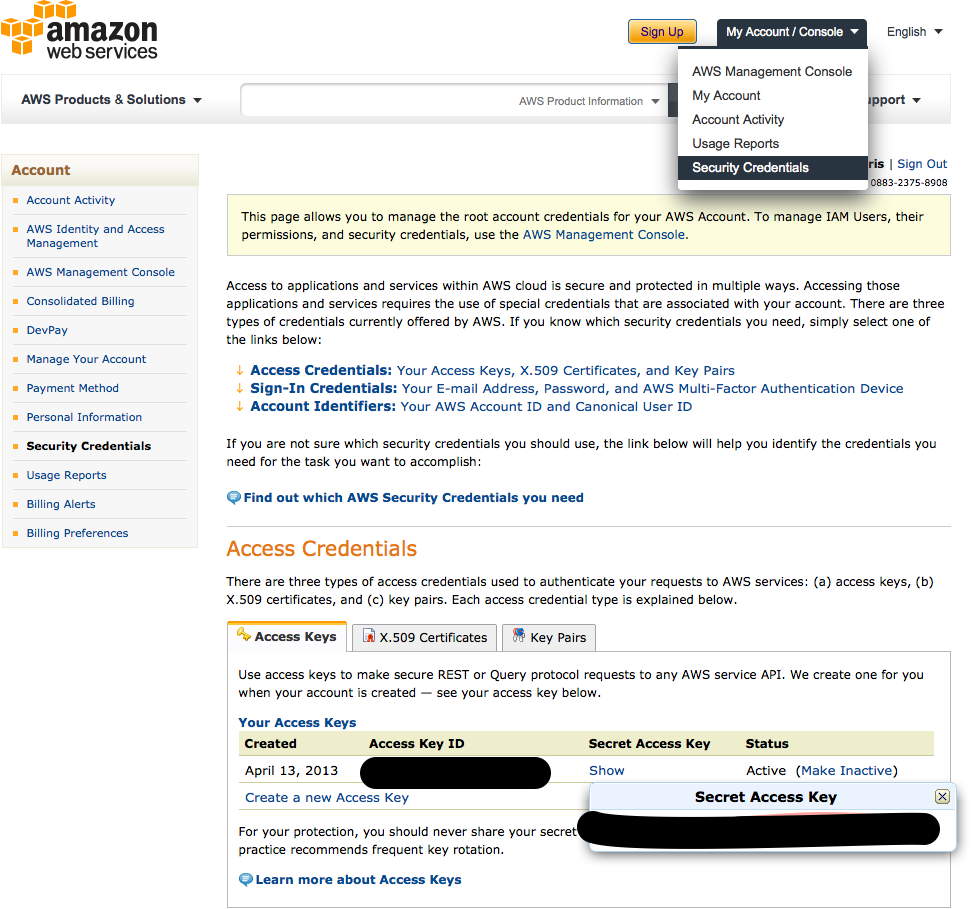


Go to load balancers and enable https forwarding to port 80, set the ssl certificate for it. You can create a self signed certificate for now while waiting for the verisign one, instructions here: <http://www.nczonline.net/blog/2012/08/15/setting-up-ssl-on-an-amazon-elastic-load-balancer/>

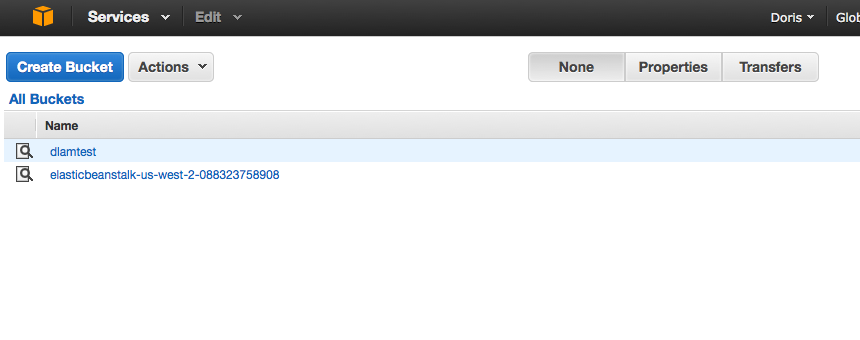




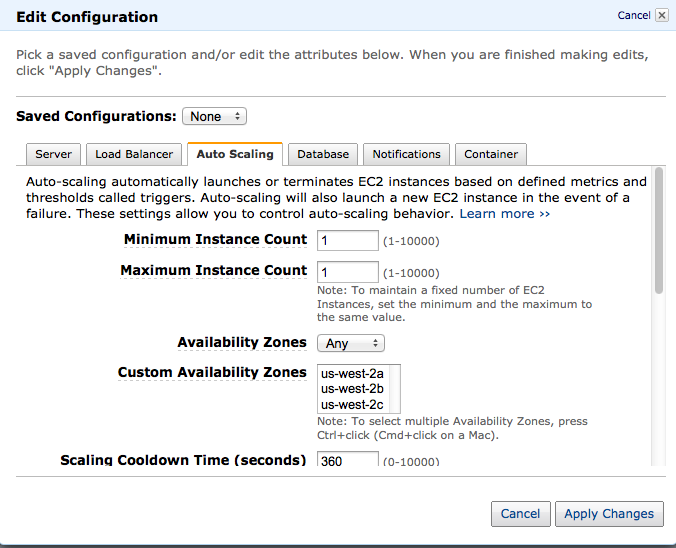
Take note of your Access Key ID and Secret Access Key on the credentials page.



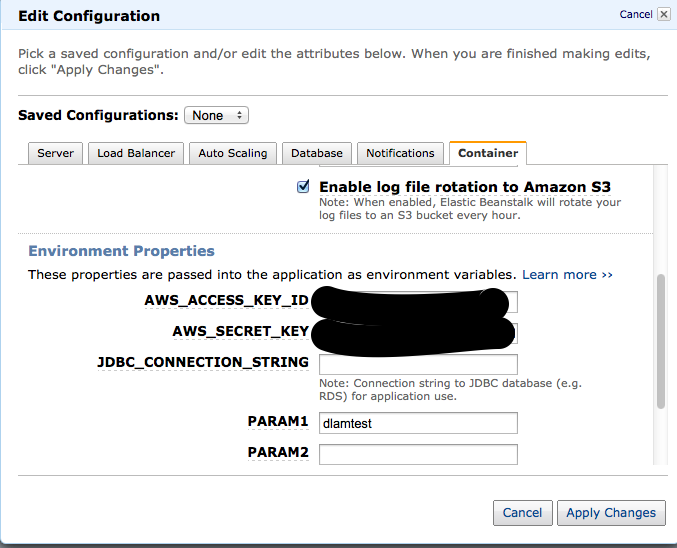
Go to the S3 console and create a bucket, maybe call flowerseeker to store uploaded pictures.



Go back to ElasticBeanstalk console, expand Environment Details, Edit Configuration.

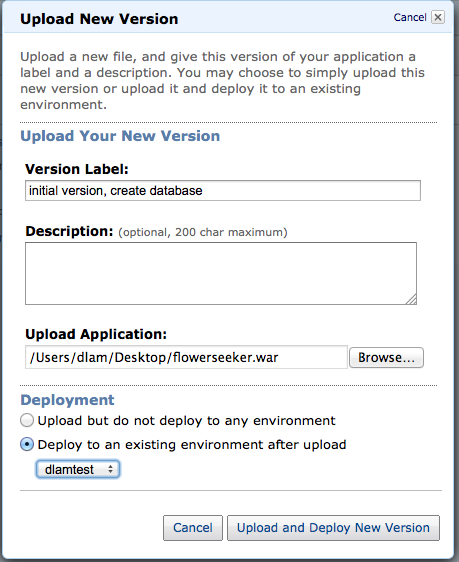
Change Auto Scaling max instance count to 1 for now. 

Enter in the Access key id and AWS secret key and the s3 bucket name as parameters to tomcat.



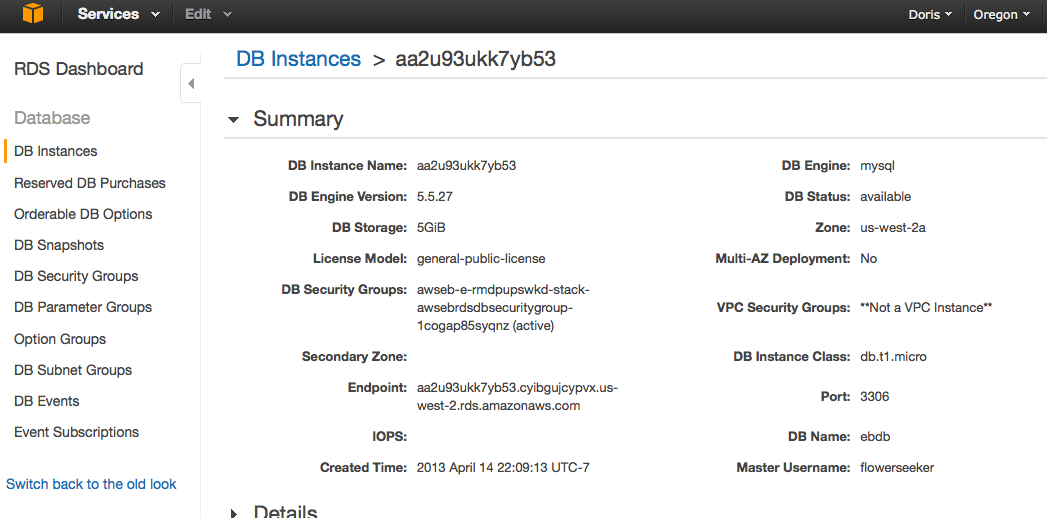
Apply changes, checkout the aws branch from bitbucket, set the database to “create” in applicationContext.xml, export to war in Eclipse. The awscredential bean and jdbc properties config should already be set to take values from the environment vars.

Upload new version



terminate any other environments it may be running so you don’t go over the free tier. Next versions change db create to update or validate so you don’t wipe out the db next time…

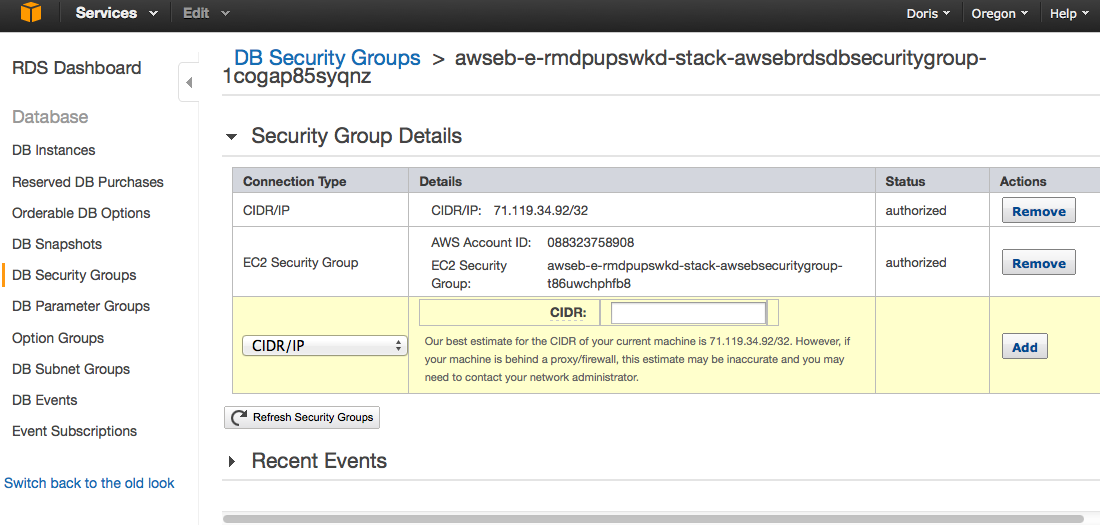
To connect to the database directly, go to the RDS console



note the endpoint, db name, port

go to security groups, edit the group the database is using

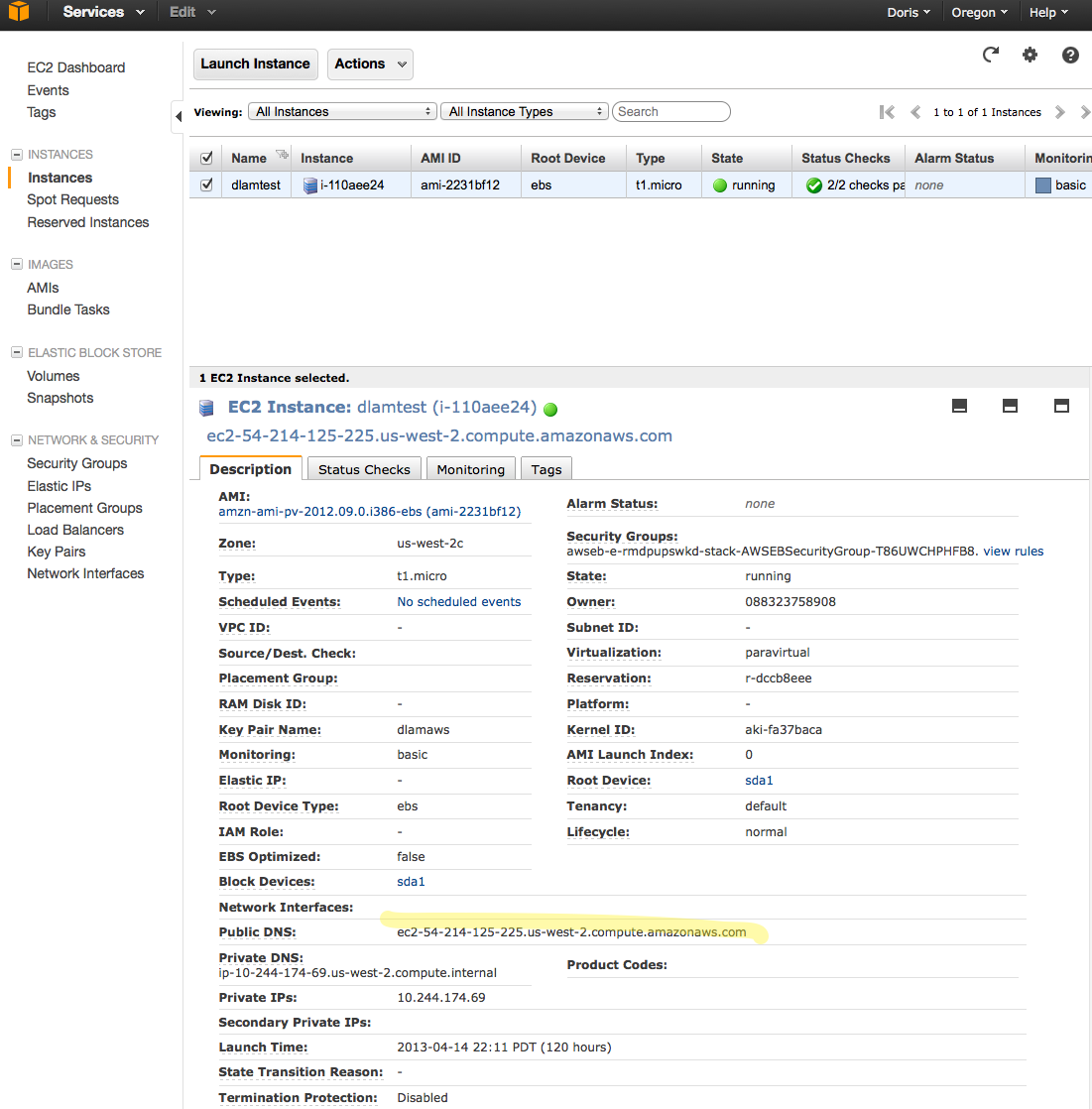
add the ip you want to connect from (it should tell you which ip you’re using now)



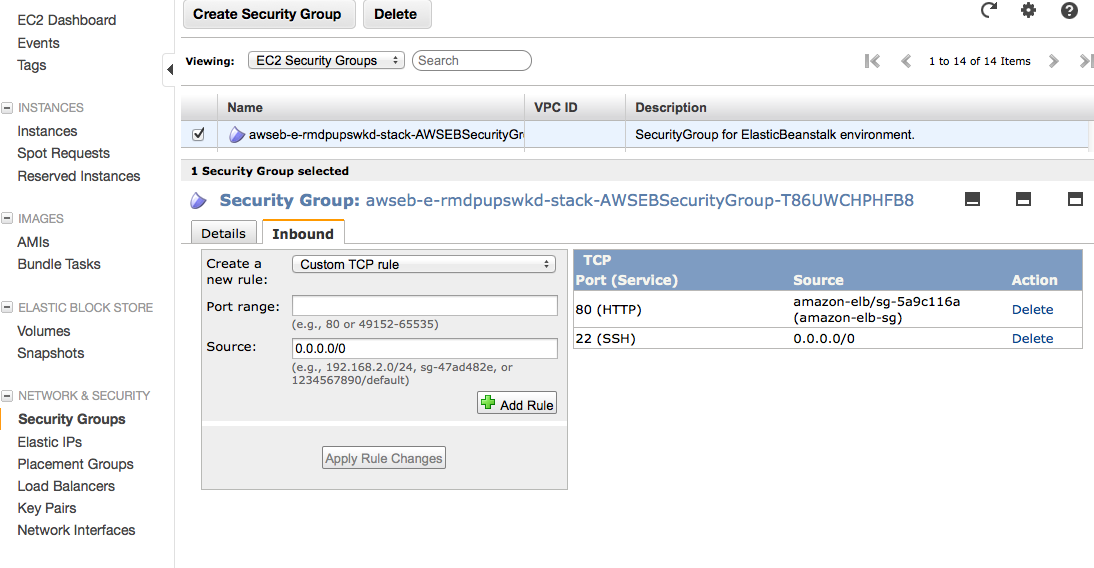
now you should be able to connect using a mysql client

To log into the ec2 instance:

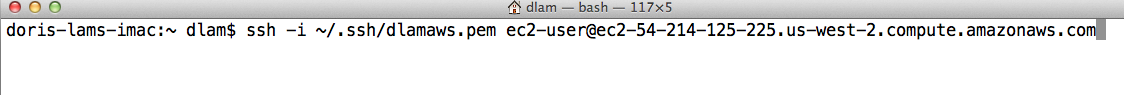
Note the instance public dns in the ec2 console



check the security rule allows inbound ssh



do this in the terminal using the private key you downloaded earlier when generating key pair.



the ec2 user has sudo permission. You can also check tomcat is running.

