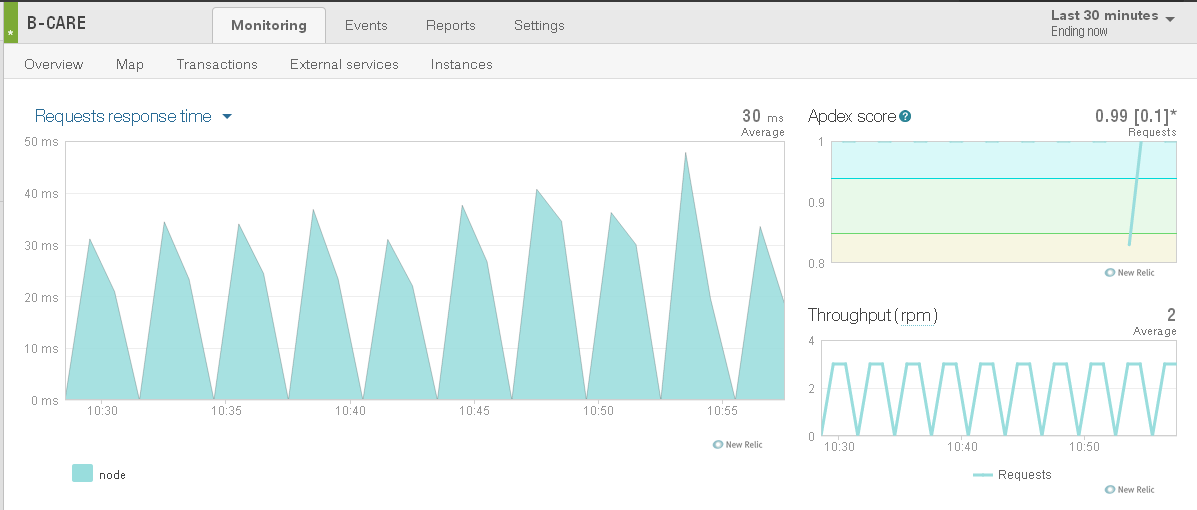
Load Testing of B-CARE Helpdesk App

The B-CARE Helpdesk app specs state that there may be up to 250 concurrent users upon launch. In order to give a good margin of error, we are load testing the website with 500 concurrent sessions. Overall testing specs:

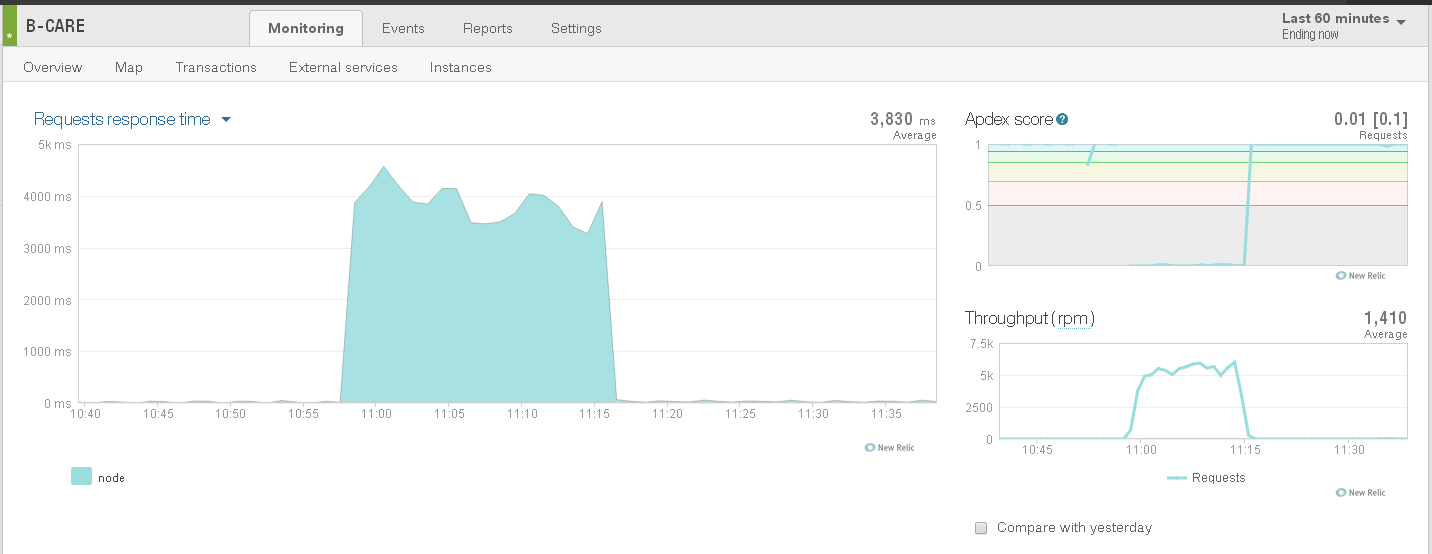
* Using the **siege** tool from a single Linux box running Ubuntu on EC2 on AWS
  + Siege parameters in .siegerc file:
    - Using csv logging
    - HTTP 1.1 protocol
    - No caching
    - Connection directive: close (using keep-alive may affect size of database storing sessions, this was not investigated at this time)
    - Concurrent users: 500
    - Repetitions: 600 (this results in a total of 500\*600 = 300,000 hits to website per test
    - URLS: only PUBLIC URLS are being tested as siege cannot automate logging in. The URLS tested are:
      * Public URLs on site itself:
        + http://bcare.herokuapp.com/
        + http://bcare.herokuapp.com/contact
        + http://bcare.herokuapp.com/privacy
        + http://bcare.herokuapp.com/about
      * Slideshare resources accessible publicly:
        + https://www.slideshare.net/slideshow/embed\_code/28886281
        + http://cdn.slidesharecdn.com/ss\_thumbnails/110713briefingorientation-131204090252-phpapp02-thumbnail.jpg?cb=1392046383
        + https:// www.slideshare.net/slideshow/embed\_code/28886303
        + <http://cdn.slidesharecdn.com/ss_thumbnails/110813introb-care-131204090336-phpapp02-thumbnail.jpg?cb=1392058118>
      * URLs are accessed in a random order; no guarantee each will be accessed equally but presumably with high reps the distribution will approach even. (Internet=true option in .siegerc)
    - Delay: 1 second between repetitions
    - Benchmark: false
  + The command line looks like this: siege –l”LOGNAME” –v > “VERBOSEOUTPUTLOGNAME”
  + We viewed the Apdex score on New Relic and also analyzed logs for issues.
  + The parameters we can play with are number of dynos, and database plan

This is what New Relic looks like when the site is idle  


The spikes are from New Relic’s availability monitoring.

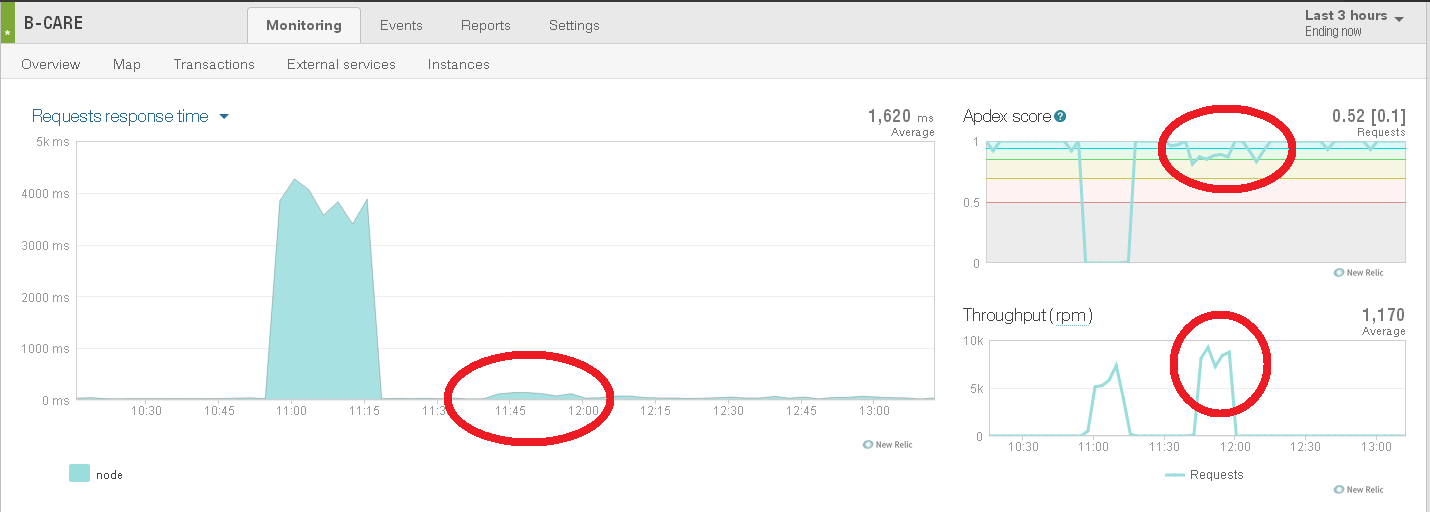
# Run 1: Baseline

* Dynos: 1 Small
* Database Plan: Yanari Standard
* Command line: siege -c 500 -l"siege-baseline.log" -v > siegeoutput-baseline.log

Results with one dyno:

# Run 2: Five Dynos

Because during previous testing we found that 7 dynos handled the load fine, we’ll try 5 dynos as our next step. If it works fine, we’ll try 3; if it doesn’t we’ll try 6.

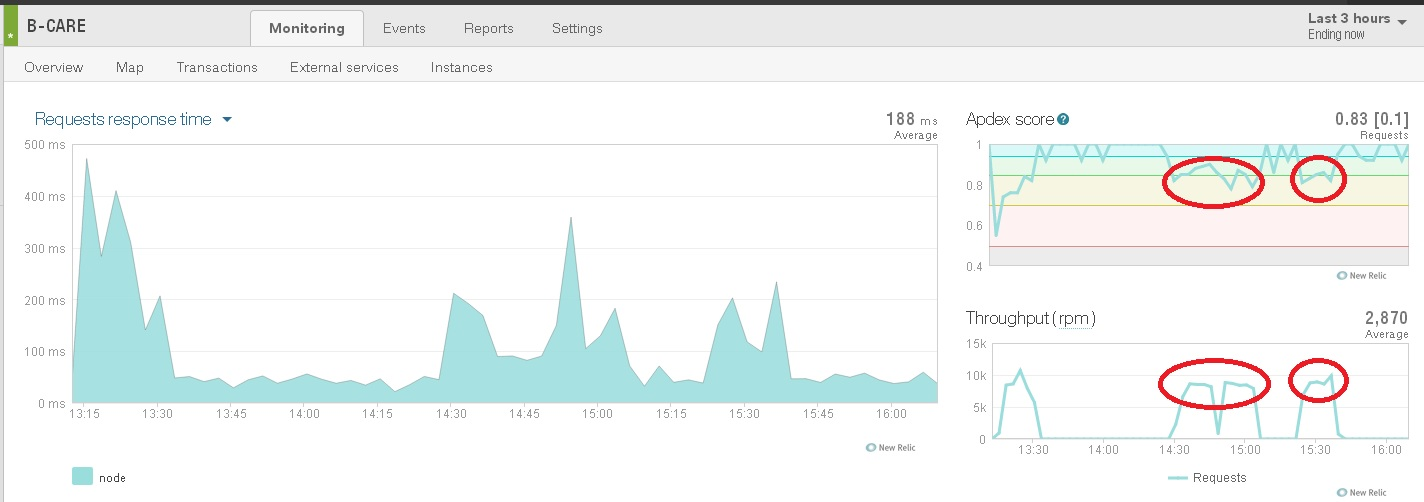


Not quite there, there are a couple of valleys that poke into the yellow.

# Run 3

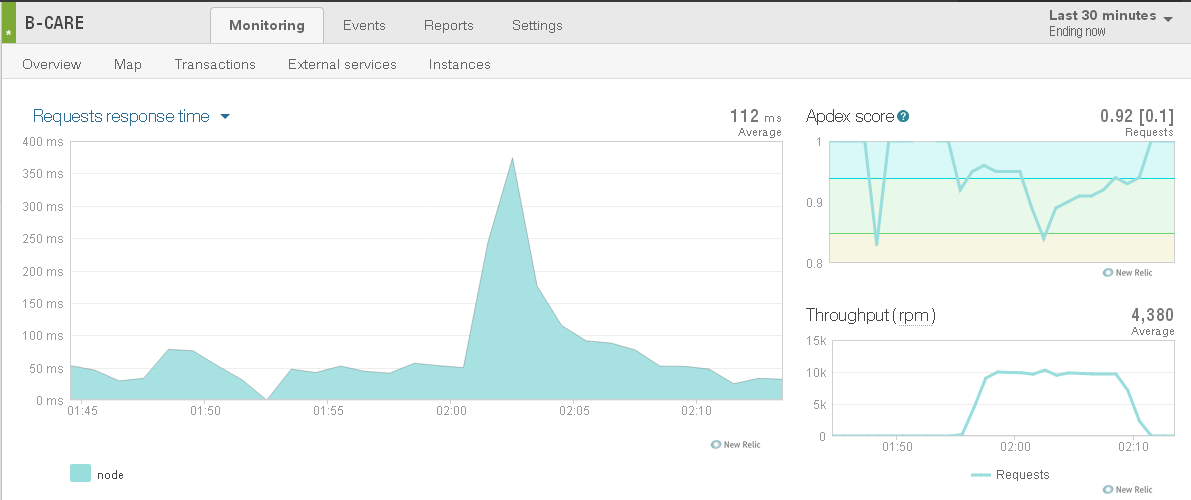
Six dynos—ran three times due to various reasons (kept getting disconnected from terminal session running siege; session running top did not disconnect)

(next attempt is with a bigger database package)



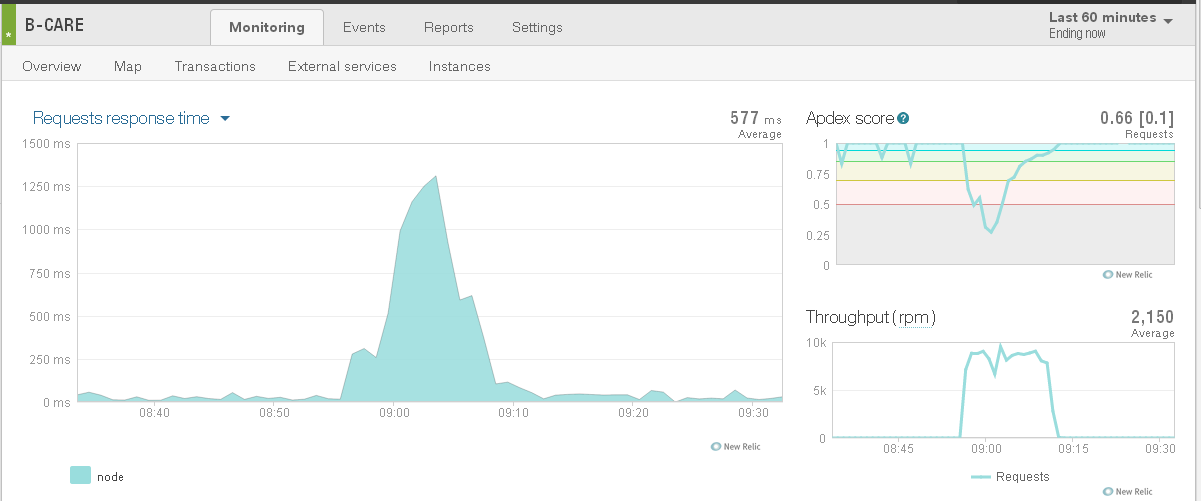
Adding a dyno didn’t keep valleys out of the yellow; need to try upgrading database

# Run 3.5

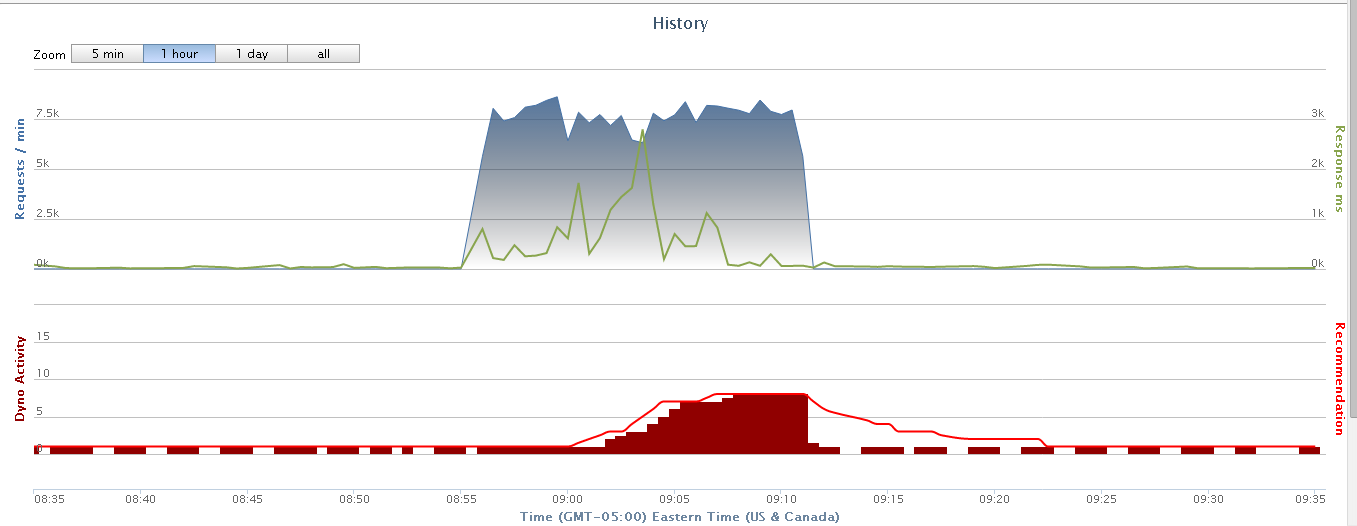


# Run 4

Since I am at home, trying Adept Scale instead of the Postgres thing. Scaled down to 1 dyno, and added the Skiff plan.



As you can see, the performance dipped significantly into the red before improving, due to how slowly the autoscaling works. We can tweak the Adept Scale settings to make it respond more quickly.



It seems like one big question we have to ask is, what is the kind of load pattern we expect to see? If we expect big, precipitous rushes, we’ll want to go with more dynos all the time, because the Apdex score becomes unacceptable while the dynos scale up (relatively) gradually. However that is really expensive. If we expect smoother transitions, Adept should be fine.