	Requirement	Est. Time	Notes
1a	Stable Solution	5 days	Coding: Extract / Parameterise the *crash* & *failure_rate* code so that it cannot apply in production. Infrastructure: Implement a load balancing solution
1b	Secure solution	0.5 days	Obtain LetsEncrypt SSL cert and move production endpoints to HTTPS Include authentication in the API
1c	Maintainable Solution	5 days	Rebuild service before moving production to new code.
2	Monitoring	2 days	Use AWS Cloudwatch alarms to alert on system availability. Publish Prometheus data as custom Cloudwatch metrics to alert on app-specific issues
3a	Production-ready HTTP Server	2 days	Deploy using AWS ElasticBeanstalk, (uses Apache with mod_wsgi)
3b	Good API	?	Code solution, flask plugin ?
4a	Add authentication	3 days ?	Implement authentication API endpoint
4b	Add request history	3 days	Create AWS RDS backend to store user/search/searchdata
5	Backwards Compat.	?	Ensure feature parity when rebuilding the new service. Maintain the existing API method in the new service, with an interpreter (if necessary) to transparently convert the old input data format into any new format required.
6	Efficient Algorithm	?	Rebuild solver code; didn't review that specifically for efficiency
7a	Deploy to Cloud	3 days	Deploy to AWS ElasticBeanstalk
7b	Automated Deployment	3 days	Use AWS CodeDeploy/Code Pipeline to automatically commit to EB Configure EB to stage rolling deployments and roll back on deployment failure
8	Auto-healing instances	1 day	Various approaches; - Use autoscaler settings to automatically terminate dead instances and start new ones - Use a cloudwatch alarm to trigger a "restartAppServer" call in EB
9a	Integrate with Auto-scaling	1 day	Use AWS Elasticbeanstalk w/ ELB & Auto-scaling
9b	What scaling rules?	-	Probably latency; customers don't care about how the code performs so long as it's fast. Maintain 2 instances at all times, scale up by one when latency per request exceeds 500ms for 5 minutes. Scale down by one, when it drops below 250ms again. Figures plucked from the air – baseline tests should establish the typical/desired latency of the application under