Backend Engineer Homework Assignment

The goal here is to see how you deal with a breadth of requirements, and also to give you a taste of the various sorts of things you might need to do on this job. It's best to set up a Github project and then share it when you are done (details below). You should only spend up to about 3 hours on this; we don't want to ruin your weekend! For the onsite interview, we'll be reviewing this and talking about what you did and why.

Spring Boot

Create a Spring Boot project. Feel free to use the Initializr. It needs (at least):

- Domain Object: Person(id: long, name: string, age: int)
- Controller
- Service Object
- Data Access Object
    - Please don't take the trouble to connect this to a real database; instead, keep a Map<Long, Person> as the "database"
    - Initialize with 2 or 3 Person objects

Create 3 APIs:

- Fetch Person by id
- Fetch all Persons below a certain age
- Create a new Person from input values

The APIs should return JSON, and accept JSON if necessary. Design the URL scheme however you like.

Scripting

Write a script (hopefully in Groovy or Python) (or bash if you must!) that calculates the percentage of cumulative time for each Controller, and prints them out in order from highest to lowest. The cumulative time is the last column in each row. Only do the Controllers; filter out the others!

AdviceController	1	349	1.50	87.510	471	899,588	1,352,000
AdviceService	1	2,196	1.49	87.520	471	899,627	1,339,000
AlertController	0	289	1.37	15.310	2	157,323	214,700
AlertService	0	148	0.91	124.100	809	1,275,337	1,166,000
AlexaController		291	892	333.00	0.008	0	81	26,970
AlexaService	0	892	9.51	0.838	3	8,615	81,950
BasicErrorController	0	60	0.03	2.572	0	26,434	923

ConfigController	0	153	0.07	53.160	0	546,476	39,970	
ContentController	0	10,749	9.86	206.100		11	2,118,942	20,890,000
ContentfulService	2	10,748	17.21	1.626	3	16,713	287,600	
ContentService	0	4,653	22.11	345.100		590	3,547,619	78,450,000
CsToolsController	0	29,809	160.10	3.341	6	34,338	5,496,000	
DailyContentController		0	325	1.08	10.020	0	103,028	111,500
DiaperController	1	877	11.87	5.500	150	56,534	670,900	
DiaperService 0	2,470	12.11	114.200		150	1,173,741	14,220,000	
FeedingController	1	2,290	12.54	19.930	501	204,814	2,569,000	

Output format should be like:

FeedingController 27%
AlexaController 13%
…
(those are not the real answers; just showing the format)

DevOps

Support I have an AWS Application Load Balancer that distributes load to 3 EC2 instances running identical server processes. The ALB is assigned a DNS A Record by AWS, and we have a CNAME alias to that A Record.

It's essential that we are notified if this service is not working. Create a list of things you might monitor (with something like Nagios, Cloudwatch, Prometheus, etc) with some service that will page us if something goes wrong.

SQL

Given the following schema

```
CREATE TABLE `user_debug` (
  `id` int NOT NULL AUTO_INCREMENT,
  `version` mediumint unsigned NOT NULL DEFAULT '1',
  `member_id` int NOT NULL,
  `debug_level` varchar(20) COLLATE utf8mb4_general_ci DEFAULT NULL,
  `create_date` datetime NOT NULL,
  `update_date` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  UNIQUE KEY `ud_umid` (`member_id`),
  CONSTRAINT `ud_mid` FOREIGN KEY (`member_id`) REFERENCES `member` (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=11802 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci
```

We need to set `debug_level` to 'NONE' if the last time the row was updated is more than 10 days in the past. Write the necessary SQL.

Test Parameters:

- Feel free to use any libraries, or dependencies,
- We don't expect it to be production ready code, but try to take reasonable procedures you would normally take when setting up and starting a new project,
- Take notes, especially if there are any false starts or bad paths you went down, this happens and being able to explain your process is the key,
- Try not to spend too much time, over 3 hours is too much
- Feel free to add any extras that might be necessary to enhance the requirements

Test Review:

- After you finish, please notify your recruiter and submit your test by:
    - uploading your project to Github and sharing the repo with Ken
        - https://github.com/kenwdelong/
    - OR
    - emailing a ZIP (as a secondary option)
- At your interview you will present what you have completed of your work. You will walk us through the code, explain your thinking, show us what you set up, and demo what you accomplished.