**TED Search**

TED Search Application:

TED Search is a search engine for TED Talks.

The team writing this search engine uses "feature branches" and a shippable master.

Up to now they did everything manually, and deployed on a clunky machine. It's time that they up their game - and you are going to do it for them!

The application has two components:

- The main process - a spring boot "fat jar" application

- Cache - a memcached server (or cluster)

While the application CAN work without memcached - it will work much more slowly...

Application documentation:

1. To build use ‘ mvn verify’. The artifact is a jar file in the target folder
2. To execute:

java -jar <the-jar-file> --spring.config.location=./application.properties

Configuration is done in the "application.properties" file.

Your missions:

Phase 1: Build & Test

* Build the jar
* Dockerize it
* Execute from within Docker, and see functionality
* Integrate dockerization into maven build by adding docker plugin to pom.xml

Phase 2: E2E Tests

* Write an E2E test using python or bash (curl)

Phase 3: Reverse proxy and FE server

* Create compose file that adds an nginx reverse proxy
* Package all static files in nginx and server them directly
* Update build so that static files are NOT saved inside the jar
* send me an micro service Architecture (write on paper)

Phase 4: Create production (PROD) environment

* Create static production environment in AWS using Terraform
* Use AMIs for FE Server (Nginx), Backend server (JRE) and Memcahched X 3.

Production is NOT going to use Docker!

* Write script to deploy/update production (“mutubale servers”)

Phase 5: Implement CI/CD using Jenkins MBP

- on any branch - build & test

- on master - build, test & deploy to the PROD environment

Phase 6: Dynamic environments

* Implement a Terraform script that provisions a machine for an “all-in-one” docker-compose environment
* Implement the deployment using bash and the AWS cli.  
  Note that this can be a simpler script, as you have a known starting point

Phase 7: Enhanced CI

Each feature branch with #test in commit message will:

- Spin a NEW DYNAMIC TEST environment in AWS with Terraform

- deploy as part of the Terraform script

phase 8: Cleanup and cost maintenance

* Implement a job that runs every 15 minutes and sends you a list of all active environments by mail
* Implement a Jenkins job that fully removes any TEST environment that is older than 15 minutes and sends a report to you by mail