

**DevOps Assessment**

Imagine we are going to build a web application for one of our clients. We are going to use the traditional [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) stack, but we want to run everything in AWS. It is a b2b aggregator of catering services for business. Whenever you want to organize a meetup, a party for your employees or something like this, you can use our client’s website. It is focused only on businesses who are interested in professional catering and have quite complicated requirements in terms of level of service and so on.

Let’s imagine that the whole website is pretty easy - it has a calculator of approximate catering costs (our data scientists trained a machine learning model and we built an api serving the model) and a list of providers available near you (and filtered by certain conditions). So the user (business’s representative) enters the site and can search via filters and select different providers (in order to contact them directly) or he might enter some details (number of guests, menu, …) and the site will preliminarily estimate the costs.

1. Please suggest an architecture in terms of AWS services. We like the micro service approach pretty much. How would you split the teams working under the project?
   1. I’ve attached a copy of the rough architecture I’ve drafted for the project
   2. I would assign resources to the project with a function oriented approach whereby the teams will be responsible for their service areas, viz. SecOps, DevOps, Data Management services, BI and AI services, Infrastructure services and Development
2. We also like “you build you run it” approach. How would you recommend to organize the process of software development and deployment in this case? What tools would you suggest to use?
   1. Agile Software Development
   2. Atlassian Jira for workflow management
   3. Github/Gitlabs for code repository, versioning and team collaboration
   4. Jenkins for build and deploy code pipeline with teams contributing to feature branches
   5. Atlassian Confluence for document library and wiki
3. Our client cares about security a lot and he does not trust cloud services. How will you approach this issue? What are you going to do to make the service secure enough and eliminate the most threats (and which)?
   1. Ensure GDPR compliance.
   2. Secure data at rest with an encryption key from an on premise security module passed to AWS KMS for disk encryption.
   3. Secure data in flight by SSL encryption.
   4. All backups will be made to S3 and will be encrypted with the onsite HSM encryption key.
   5. I additionally suspected there are other serious security considerations and I’ve therefore segregated the data into an internal VPC VLAN and ensured that the Web/API layer resides in a separate DMZ VLAN fronted by an enterprise class F5 Big IP load balancer with a public IP address and this will be fronted by a Route53 DNS entry. The API/Web layer has an additional internal Application Load Balancer as a back plane for a service and messaging layer.
4. Could you please implement some part of proposed infrastructure? E.g., it would be enough to have a web-server returning “hello world” and “hello you” pages but we expect you to demonstrate your passion to routine automation and we’d like to see the implementation of some fundamental layers on your choice (e.g., networking, etc.)
   1. <https://github.com/nolanvenkiah/KIProjectRepo>

Could you please estimate the implementation effort of the proposed infrastructure if you were to do it today? Feel free to add as much details as necessary for a colleague to deploy this.

1. Our data scientists are going to improve the model constantly. How would you organize the deployment of new model trained to production? What main risks and obstacles do you expect?
   1. I would request that the data scientist follow the SDLC approach by committing code to segregated Dev/Test, Staging, Production
   2. Additionally I recommend that code be segregated by feature sets and these features follow the above mentioned process. These features will be merged into the different environments upward into production eventually following peer code reviews and team lead manual merge acceptance.

Please feel free to make any assumption about the project you find appropriate but do not forget to share them with us :). And do not hesitate to contact us in case you have any questions.