**Question 1: Outline your approach in terms of what DevOps practices and principles you would implement as the project kicks off.**

**[Shraddha]:** A new project is good opportunity to start using a DevOps approach as there are no earlier development procedures to follow.

If we look at the internet, there are dozens of articles available which talk about how to implement the DevOps. However below are my recommendations which I firmly believe in and have experience in:

DevOps is not only a technology adaption but also a cultural adaptation. If you do not have the buy-in from your team and the team is not keen to come out of their comfort zone, it is obvious that the adaptation would fail. About 90% of the projects fail to adapt to DevOps because they fail to change the mindset of their technical teams. Hence it is always good to start with some detailed workshops with the technical teams to understand the current pain areas and showcase how DevOps comes to rescue to these traditional pain areas. It helps to develop the DevOps mindset amongst the team and creates a healthy atmosphere where everyone wants to contribute.

Transparency in communication and collaboration are the most important building blocks of a successful DevOps team. Hence, selecting the right tools for Communication and Collaboration is the next most important thing to look at when we are looking to adapt DevOps.

In this competitive marketplace, Customer is always looking for improved product with faster delivery. If we want to achieve this it is very important that we keep a keen eye on Automation of the redundant tasks. We should always look for automating as much of the development process and operations tasks.

Further going into the details, adopting DevOps in any Web App or Mobile App development projects requires implementing the most important elements- profoundly known as “The 6 C’s of DevOps Adoption”.

1. Continuous Planning
2. Continuous Integration
3. Continuous Testing
4. Continuous Monitoring
5. Continuous Delivery
6. Continuous Deployment

As per the use case provided in the DevOps Challenge Document, I recommend below to be brought into practice.

1. Azure DevOps (<https://dev.azure.com/> )
2. Azure App Center (<https://appcenter.ms/> )

Azure DevOps to take care of the Web and Backend Development & testing. While App Center for the Mobile App Development and testing.

Since, we do not have a dedicated QA teams, we should utilize the Testing feature(Azure test plans) in Azure DevOps to our benefit which will reduce the dependencies from the QA team and will also ensure the quality of the development is not compromised.

**Question 2: What is the first (or most critical) area that you would choose to focus on?**

**[Shraddha]:** I would team up with the right group who are ready to come out of their traditional working ways/project life cycle and accept new mindset to work in a changed culture. Because when a company desire to adopt DevOps, there are a lot of changes that we need to do. As in Agile methodology with DevOps approach we are going to:

**Reduce time** 🡪 when it comes to development and deployment.

**Reduce approval** 🡪 a lot of unnecessary approvals may hinder our fast or automation approach. For ex. Approvals from business/stack holder or senior managements to deploy an application on staging environment. Too many approvals at different stages of a project life cycle can be reviewed.

**Revisit processes** 🡪 Processes mapped to traditional approach (again when it comes to development and deployment) needs to be revisited to be fit for DevOps.

At the same time, we need to ensure that **we do not compromise on quality and everyone must take accountability.** Which says: need to change the way we think, build, and implement.

And the approach towards the implementation would be selecting a small milestone or a development cycle to implement with DevOps rather than try to transform whole project to DevOps in one stab. It improves our chances to succeed and creates an example model for the whole team to adopt. This approach is well tested by implementing it practically in one of my previous projects and has functioned as expected.

We should select a small feature from the product backlog and try to implement the complete CI-CT-CD pipeline using Azure DevOps.

From the technical front, my first focus would be on understanding the infrastructure requirement and the application so that our DevOps implementation is creating value to the business.

Important aspects to be noted while understanding the infrastructure would be –

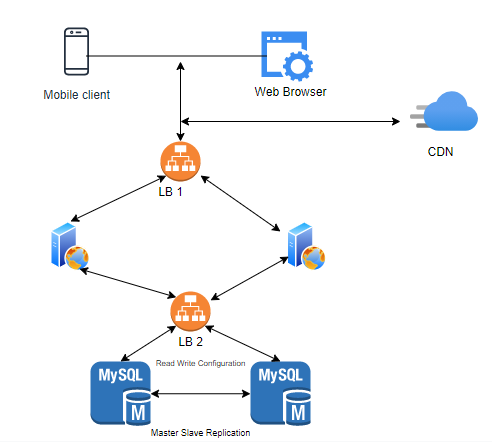
1. Environment Versions – Dev/Test/Staging/Prod
2. Infrastructure as a Code
3. Software delivery Cycle time
4. Infrastructure Security

* We should select the tool for implementing DevOps. As per my understanding, in 1Nebula team has already decided upon using Azure DevOps.
* We should implement containerization (use IaC) to ensure that individual components of the software are running without any dependencies. It would also help the Ops team to manage application quickly and reduce ops workload by creating self-healing applications.
* Integrate complete infrastructure with the CI/CD pipeline, using Azure DevOps we can easily resolve the configuration dependencies for development and deployment.
* Work with the QA team to automate the test cases to achieve faster delivery. Functional testing of the App can be done manually by the QA team.
* Configure monitoring on the complete environment which will help us preventing downtime, identify application bugs and defects.

**Question 3: How would you monitor and alert on downtime on any of the given aspects of the system?**

**[Shraddha]:** Since the Application monitoring must be configured based on the architecture and type of application. Below is simple Application architecture I have kept under consideration before deciding upon the monitoring and alert configurations. (Not considering K8S implementation for now)

**Assumed Architecture:**



Following Monitoring should be in place.

1. Web and App response time. Example: Login, Webpage Loading time
2. Functional monitoring- All the functionalities are working as expected.
3. Response failure of the API’s. (Can be configured on the error response code.)
4. API response time against all API’s Web & App. (Based on the business agreement, threshold can be set. If the API response is beyond the threshold, it should trigger alerts.
5. CPU Utilization- Auto scaling configuration on Azure. (Considering the current infrastructure is already on Azure.)
6. Website and App Uptime.
7. Log Monitoring
8. Crash Analytics to track and trace Mobile Application Crashes.

Alert Implementation:

1. Alert emails can be configured to automatically send mail emails to Ops team distribution list.
2. SMS Notifications can be automatically triggered to team Ops distribution lists.
3. Free Google Firebase Dashboard configuration to Monitor the performance.
4. App Dynamics would be my suggested tool to monitor the App and Website together if we are looking for a monitoring solution.