**GAP ANALYSIS DOCUMENT**

**Missing Details:**

1. Naming conventions not provided for Azure Resources.
2. How Many different environments will be there? Dev, QA, Stage, Prod and Hotfix etc.
3. Which Azure Resources need to be used.
4. VNET details, Subnet details, NSG , ASG or routing details.
5. Firewall Rules and App Gateway, LB requirements.
6. Secret Management and Key vault integration.
7. Information regarding RBAC
8. Monitoring and Logging Setup information.
9. Artifact/ Container image storage information.
10. Custom Domains and SSL cert setup.
11. CI/CD Tools used for deployment.
12. Any Architecture or Design doc.
13. Different stakeholders list like DB, networking etc.
14. IAC tools information
15. Integration of backend service with Databricks jobs information.
16. What are the tagging policies?
17. Information regarding namespaces.
18. Deployment strategy to be adopted for release.

**Questions:**

1. What is the naming convention and tagging policy to be followed?
2. What are the different types on env? Should each have its own AKS?
3. What is the CI/CD tool that we are going to use?
4. Are we going to use Ingress or LB to expose our application to end user?
5. What are the alerting rules and mechanisms?
6. What metrics and logs we need to be monitor?
7. Are there any landing zones already setup in the organization?
8. Are we going to use Azure Keyvaults or any other vault?
9. How is the backend API authenticated and authorized?
10. Will the backend service communicate with Databricks using Service endpoint or private endpoint?
11. Are there any existing policies Azure Policy?
12. What will be the branching strategy?
13. List of approvers or change managers?

**DevOps Best Practices:**

1. Creating all the Infra using IAC like Terraform or ARM templates.
2. Using Remote Backend for storing statefile and creating separate module for each components.
3. Doing Secret Management using Keyvaults.
4. Restricting the access to Azure resources using the principle of least privilege.
5. Using Azure DevOps for tracking the project sprint status using Boards.
6. Restricting the pipeline trigger access.
7. Enabling Azure Monitor, Log Analytics and app insights reports.
8. Using Helm charts in ADO for deploying services and pipelines on AKS.
9. Using ACR to store the Docker Images.
10. Setting up Highly scalable and Available AKS cluster using VMSS and HPA.
11. Document everything related to Project, Architecture and pipelines on wiki or Confluence.
12. Enforcing proper tagging and naming conventions.