**Blazeclan Technologies**

**UTCL CWFM BILL GENERATION PROJECT TECHNICAL DESIGN SPECIFICATION**

**8 July, 2021**

Blazeclan Technologies Pvt. Ltd.

UrbanWrk, 5th Floor, Sai Radhe IT Complex,

Bund Garden Road, Behind Grand Sheraton Hotel,

Off. Raja Bahadur Mill Road, Pune 411001.

Mobile: +91 88888 77048 Office: +91 20 67611100

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | Abhishek Bansal |  | Vijay Kumar Singh |  | Initial Design Definition draft |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

[**1** **APPLICATION ARCHITECTURE Diagram 4**](#_Toc180482593)

[**2** **SYSTEM OVERVIEW 5**](#_Toc180482595)

[2.1 Azure Application Gateway](#_Toc180482596) 5

2.2 VNET Service Endpoints 5

2.3 VNET Peering 5

[**3** **COMPONENTS 6**](#_Toc180482597)

[3.1 Azure SQL 6](#_Toc180482598)

[3.2 Azure Kubernetes Service 7](#_Toc180482599)

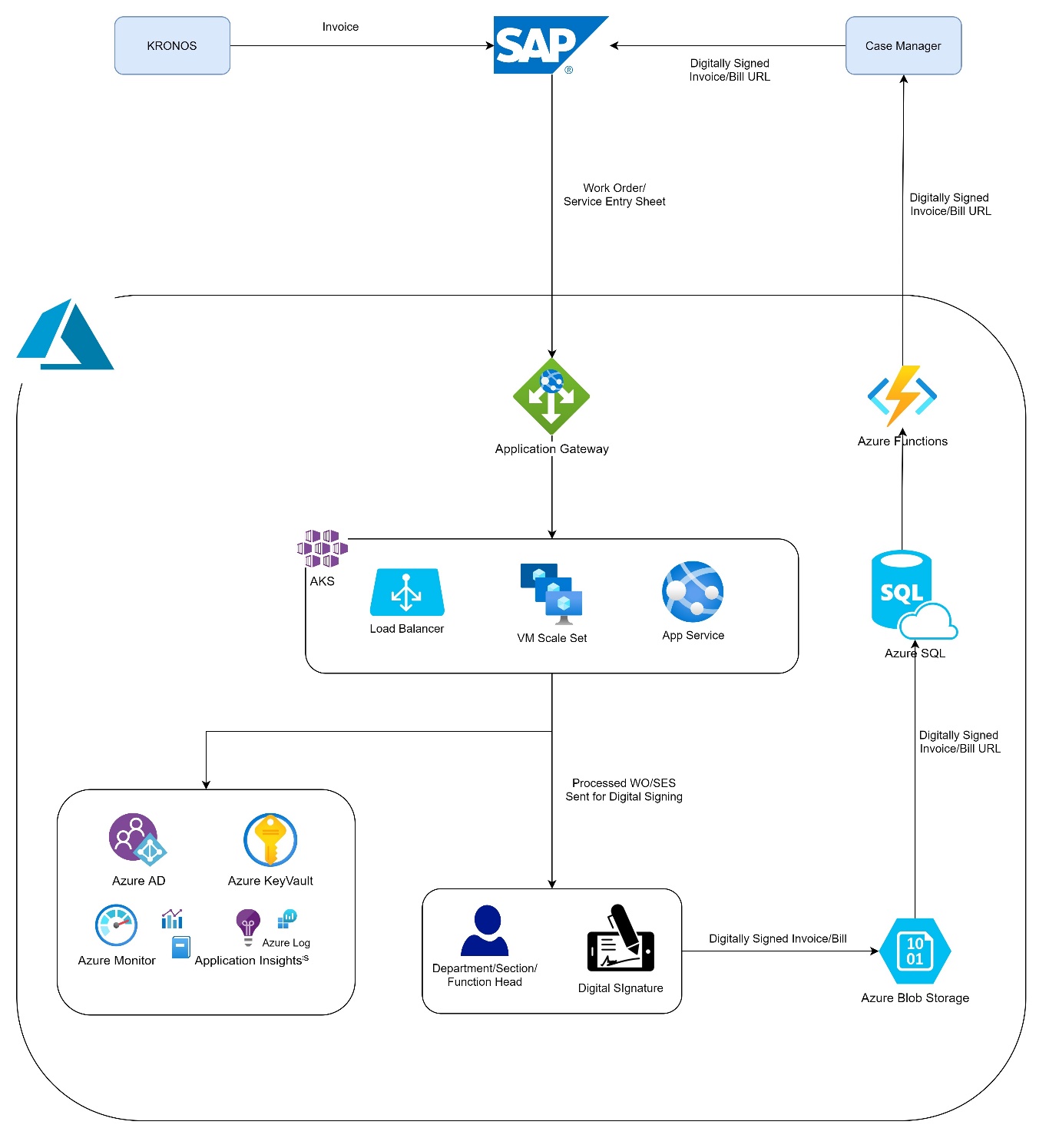
[3.3 Key Vault 8](#_Toc180482600)

[3.4 Log Analytics 8](#_Toc180482601)

[3.5 Application Insights 9](#_Toc180482602)

[3.6 Azure Storage 9](#_Toc180482603)

1. **APPLICATION ARCHITECTURE DIAGRAM**



1. **SYSTEM OVERVIEW**

2.1 Azure Application Gateway

* It would be used for validation of user traffic flow directed towards the Billing application.
* It would be part of a subnet within the Azure Virtual Network.
* It will have two endpoints – Public (Optional) and internal.
* Traffic coming from the internet would be directed to the public endpoint with a public IP.
* Traffic coming from the intranet would be directed to the internal endpoint with a private IP.
* For intranet traffic - configuration would have to be done on the on-premises domain controller (DNS) to direct it to Azure Application Gateway’s internal endpoint.
* For internet traffic - configuration has to be done at the domain registry’s end to direct it to Azure Application Gateway’s public endpoint.
* It would forward the requests to the internal load balancer of the Azure Kubernetes Services.

2.2 VNET Service Endpoints

* These would be leveraged to restrict access to Azure SQL, Key Vault, Storage from the AKS cluster subnet.

2.3 VNET Peering

* Peering would be established between the VNet containing Application Gateway and the VNet for Confluence application.

1. **COMPONENTS** 
   1. Azure SQL

* SLA - 99.995%
* Point in time restore.
* Inbuilt provision for Geo-replication and Failover group.
* As it’s a PaaS offering, we don’t have to manage anything. No maintenance, patch management, security upgrades, etc. are needed.
* Easily compatible with Azure DevOps for setting up continuous deployment.
* Auto Scaling as per the requirement.
* Provision of elastic pools which is suitable for microservices architecture. It automatically allocates the free database transaction units (DTU) to the database which is in need.

Conclusion - Azure SQL elastic pools is the correct choice for Billing application.

E.g. – If the database is getting used more, then resources would get allocated automatically based on the configured DTUs.

Using SQL in Azure VM can provide a maximum of 99.99% SLA, also the latency would increase in this case and elastic pool features would not be present. Thus, not recommended for Billing application.

* 1. Azure Kubernetes Service
* You can easily migrate existing apps to containers and run them with Azure Kubernetes Service. You can also control access via Azure AD integration and SLA-based Azure Services like Azure Database using Open Service Broker for Azure (OSBA).
* Deploy and orchestrate containers that include Windows containers and Linux containers.
* Deploy applications in seconds, at high density with hundreds or thousands of applications or containers per machine.
* Deploy different versions of the same application side by side and upgrade each application independently.
* Manage the lifecycle of your applications without any downtime, including breaking and nonbreaking upgrades.
* Simplifying the configuration and management of microservices-based Apps: You can also simplify the development and management of microservices-based apps as well as streamline load balancing, horizontal scaling, self-healing, and secret management with AKS.
* Bringing DevOps and [Kubernetes](https://kubernetes.io/) together: AKS is also a reliable resource to bring Kubernetes and DevOps together for securing DevOps implementation with Kubernetes. Bringing both together, it improves the security and speed of the development process with Continuous Integration and Continuous Delivery (CI/CD) with dynamic policy controls.
* Ease of scaling: AKS can also be applied in many other use cases such as ease of scaling by using Azure Container Instances (ACI) and AKS. By doing this, you can use AKS virtual node to provision pods inside Azure Container Instance (ACI) that start within a few seconds and enables AKS to run with required resources. If your AKS cluster runs out of resources, if will scale-out additional pods automatically without any additional servers to manage in the Kubernetes environment.
* You can also simplify the development and management of microservices-based apps as well as streamline load balancing, horizontal scaling, self-healing, and secret management with AKS.
  1. Key Vault
* SLA – 99.9%
* Integrated with Azure Active Directory using service principal.
* Secret Management – in a secure manner one can store and control access to tokens, passwords, Certificates, API keys, and other secrets.
* Key Management – As a key management solution, Azure Key Vault simplifies the process of creating and managing encryption keys.
* Certificate Management – It serves as a service to provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates. These can be used with Azure and other internal connected resources.

Conclusion – For any cloud native implementation Key Vault is the recommended option for securely storing all configuration items.

* 1. Log Analytics
* SLA – 99.9%
* Centralized repository for all logs.
* Easy integration with all the Azure services.
* Customized alerting and monitoring as per the requirement.
* Compatible with Azure automation and logic apps for automated scaling, auto start/stop the azure compute resources in non-business hours for non-Prod environment in order to save the cost.
  1. Application Insights
* SLA – 99.9%
* Single platform for monitoring application workloads and interaction with its dependencies/services.
* Inbuilt powerful analytics tools to help you diagnose issues and provide necessary recommendations.
* It helps continuously to improve performance and usability.
* Ping tests can be configured to monitor application availability.

Conclusion – Combination of Log Analytics and Application Insights is recommended to set up application and infrastructure monitoring for Confluence.

* 1. Azure Storage
* SLA – 99.99%
* It enables the users to store their documents, media files, NoSQL data over the cloud that can be accessed using REST APIs.
* Blob Storage is used to store unstructured data such as text or binary data that can be accessed using HTTP or HTTPS from anywhere in the world.
* Common usages of blob storage are as follows:
* For serving documents and images directly to the browser
* For storing data for big data analysis
* For storing data for backup, restore and disaster recovery.
* Security for azure storage data can be provided using encryption.
* Storage account would also be used for storing diagnostic logs of the various Azure components.

Conclusion – Billing application will generate many Bills and Invoices which would be stored on blob storage. The blob URL of these files will be stored in Azure SQL. This implementation would be easier to manage with low latency in performing operations on those files and later accessing them.