Architecture_for_platform

Last edited by **czerniga-gft** 8 months ago

Architecture for platform

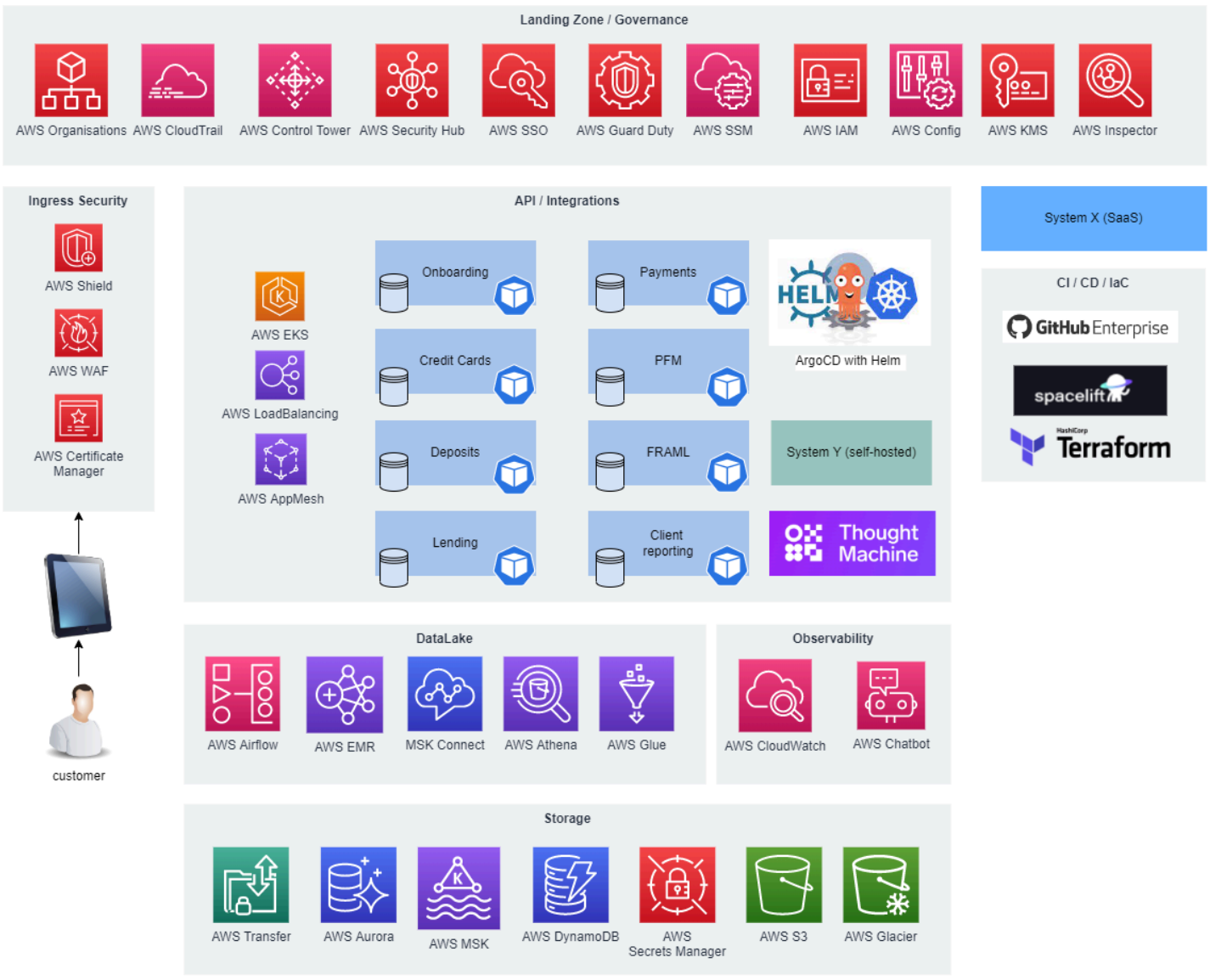
High-level platform overview

Source files:

Draw.io source: cloud-platform-asset-high-level-arch-examplev2.drawio

PDF blx-platform-20230508.drawio.pdf

SVG blx-platform-20230508.drawio.svg



Architecture summary

General principles

Architecture design is focused on the following principles:

- use AWS-managed services whenever it allows to save costs and shift to a pay-per-use vs pay per subscription model
- use cloud agnostic patterns whenever it is globally recognized standards (i.e. Kubernetes over other container models)
- use services promoting delivering AWS cloud-native security using AWS IAM security model
- prepare for using integrations with external SaaS providers though allow to also support self-hosting models if required

Landing Zone and governance

Architecture aims to use AWS Control Tower to use the best native tool for creating landing zones and deliver fundamental guardrails for aws account structure on top of AWS Organisations. From the beginning, AWS CloudTrail is delivering trail services for auditing while the general security posture is managed using AWS Security Hub using aggregation from the whole account structure and enabling broad range of tooling and services like GuardDuty, Config, Inspector. For human access, AWS SSO is enabled along with typical integration with external Identity Provider.

Ingress security

Bank customer typically operates from Public Internet. To expose services in a controlled and secure manner multiple layer of protection is applied. This starts from using secure DNS services through edge protection which is using DDoS protection along with AWS Web Application Firewall. Additional expansion capabilities exist for additional Network Firewall solutions depending on the organisation's security strategy with the potential use of AWS Network Firewall or 3rd party virtual firewall protections. Full support to end-to-end encryption in transit starts with using AWS Certification Manager with full control over transit protocols and lowering cost of managing certification rotation.

API and integrations

integration and application layer is using AWS EKS which is managed kubernetes service. We recommend using service mesh functionality in API and integration layer which might be delivered using AWS Service Mesh for functionalities likes mtls authentication in cluster along with encryption in transit and retries capabilities and network policies. Native load balancing services are managing traffic and integration between EKS and the rest of services.

Storage

We recommend using aws managed services hence for storage we rely on managed services to provide RDBMS and messaging (kafka) functionalities as long as S3 for object storage. Banks typically operate for transfer with SFTP protocol for which we recommend to use AWS Transfer family which is SFTP managed service. For special usecases DynamoDB is offered as highly scalable key value store.

DataLake

Data lake architecture is focused on lowering the cost of modern analytical and reporting capabilities required for virtual banking and pay per use vs pay per subscription model. For this purpose object storage is used with real time processing using streaming from operational database area. For transformations and aggregations EMR with Athena is used driven by scheduling capabilities delivered by Athena operating using AWS Glue schema registry. This architecture is highly scalable and prepared even for largest organisations and highest volumes of data processed.

Observability

Default observability stack relies on providing services in areas like:

- logs browsing and search from both kubernetes pods and infrastructure
- metrics provided from multiple sources (custom applications, prometheus, infrastructure)
- tracing support
- alerting support
- integration with modern notifications and optional interactive channels including Slack (through AWS ChatBot support) and or other channels like email, PagerDuty or ServiceNow

CI / CD / IaC

Continuous integrations and delivery might be constructed using many frameworks. We assume the most mature standards here are used like github enterprise along with github actions and Terraform potentially using some managed control planes like spacelift for managing IaC. We support the reuse of the libraries along with rich RBAC for infrastructure delivery pipeline. For docker images delivery we aim to operate using static controls and shift left approach for automated testing and static analysis. Vulnerabilities protections are enabled by integration with AWS Inspector scans and reporting.

Attachments:

cloud-platform-asset-high-level-arch-examplev2.drawio.svg (image/svg+xml) cloud-platform-asset-high-level-arch-examplev2.drawio.png (image/png) cloud-platform-asset-high-level-arch-examplev3.drawio.png (image/png) blx-platform-20230508.drawio.pdf (application/pdf) blx-platform-20230508.drawio.svg (image/svg+xml) cloud-platform-asset-high-level-arch-examplev2.drawio (application/octet-stream) blx-platform-20230508.drawio.pdf (application/pdf)