OPENSEARCH LOGGING

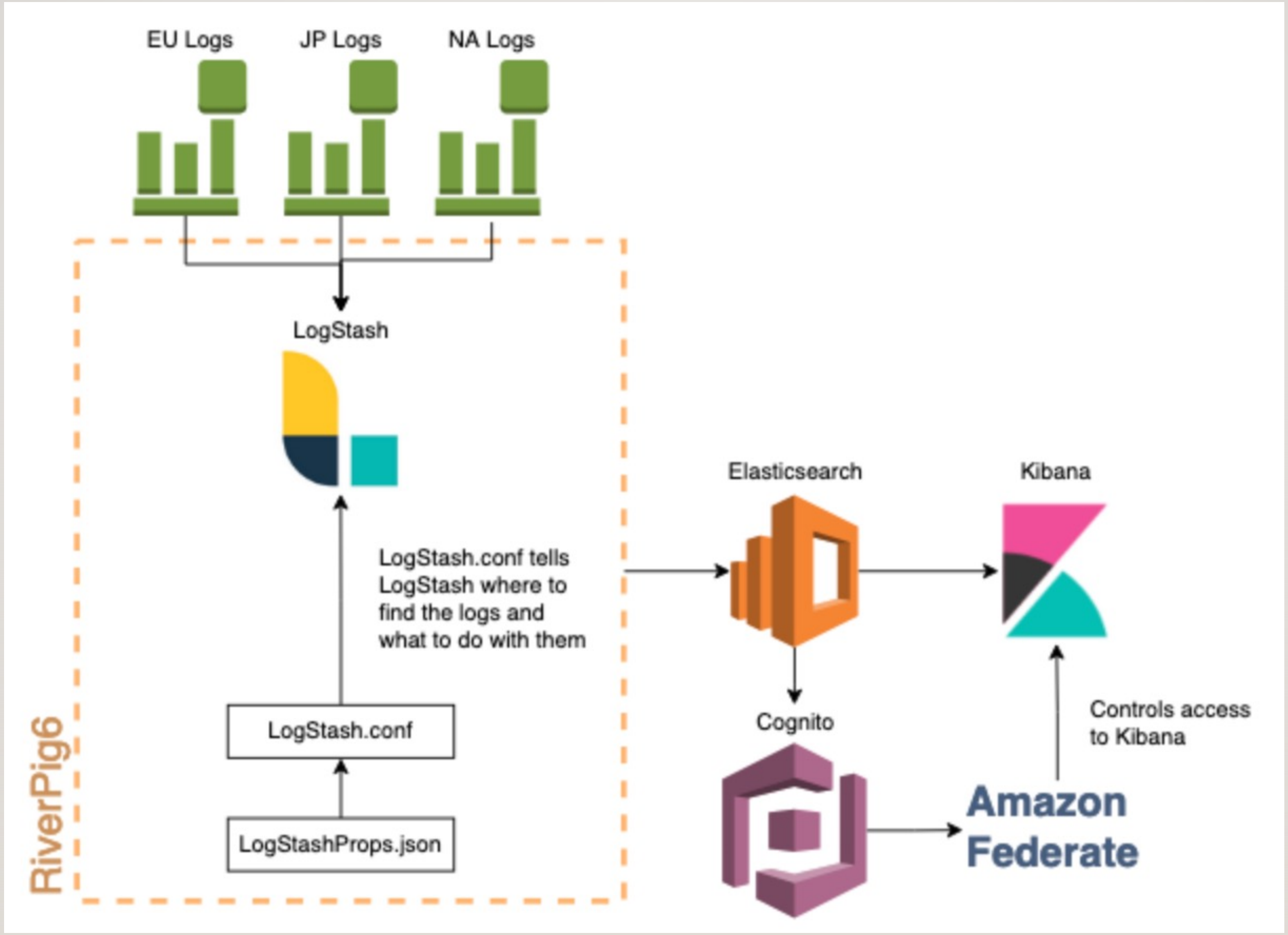
INTERN PROJECT BY EKENE ILLOH

BACKGROUND

- ADAM-Galaxy has several services (Arches, AIS, GRS, GAMS etc.) involved in registering an advertising account
- Sometimes, tracking the status of an account registration can be very difficult since the logs for the services are stored in different places, formats and systems
- It becomes difficult and time-consuming to query the different databases but all of them frequently have critical information about an account registration
- The goal was to build an OpenSearch cluster that will aggregate application logs from account registration from adam services into an easily queryable place

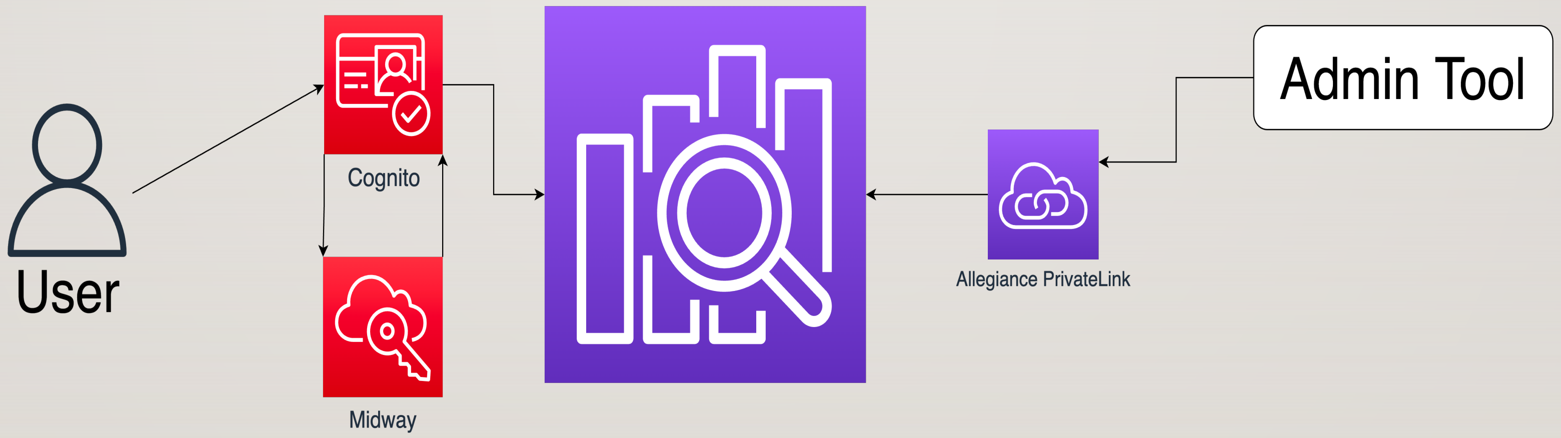
HOW THIS WOULD WORK – ELK STACK

- The logs are transferred from the services to Opensearch through the use of the ELK stack – Elasticsearch, Logstash and Kibana
- Using Logstash you can parse and push the entries in log files to Elasticsearch
- Kibana gives you a powerful web interface with search, filter, and statistical analysis functionality driven by Elasticsearch
- Using this stack we can have real-time access to all log entries from all hosts running a service


EU Logs
JP Logs
NA Logs
LogStash
LogStash.conf tells LogStash where to find the logs and what to do with them
LogStash.conf
LogStashProps.json
RiverPig6
Elasticsearch
Kibana
Cognito
Amazon Federate
Controls access to Kibana

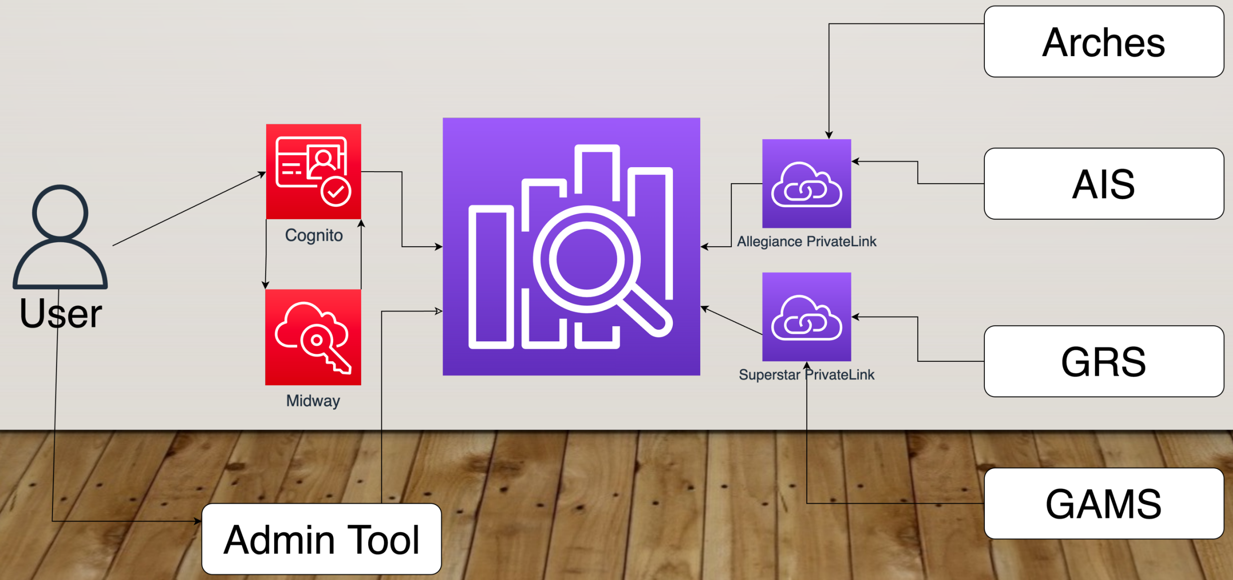
EXISTING SYSTEM

- The POC had Admin Tool onboarded to the cluster through the allegiance private link
- The cluster was set up to receive logs from Admin Tool beta NA



OBJECTIVE

- Expand the current POC to take in logs from the following additional services: GRS, GAMS, AIS and Arches
  - Scaled down to just AIS and Arches
- Build a programmatic API that would allow a user to query a service in the opensearch cluster to retrieve all the logs for a given registration attempt



PROCESS

- Set up the existing POC
- Update the Admin Tool package to onboarding beta logs from all regions
- Make changes to ADAMTimberCDK and GlobalAccountsConstruct packages to allow the onboarding of additional services
- Start onboarding the additional services by making the code changes to their packages

DEMO

CHALLENGES

- Unfamiliar tech stack
- Backend development
- Debugging build errors
- Deploying the ADAM Timber CDK

NEXT STEPS

- Clean up CDK codebase
- Set up endpoints in vpc endpoint service using CDK instead of manually on AWS console
- Build an API for querying the logs in OpenSearch
- Expand the cluster to receive logs from GRS and GAMS
- Expand the system from beta to prod for all the services

THANK YOU!