dmoney panel

Florian Popa

Considerations

- 1h time to spend
- Used AWS instead of Azure

Considerations

- 50 years of customer data and business logic
- Legacy system
  - Leverage existing data and functionality, without compromising stability or performance
- Web and mobile
- Transactions

On-premises - Cloud

- We connect on-premises system to the Cloud
- We have access to existing APIs
- We start moving **relevant** data to the Cloud async

Transactions - approaches

- Call on-premises APIs for real-time transactions
- Data is also stored in the cloud
- Slow
- 100% guarantee

- Transaction happens in the Cloud
- Publish events for on-premises
- Fast
- Consistency analysis required
- This is Target Architecture

Transactions - hybrid approach

- Critical
- Immediate consistency is necessary (e.g., account balance updates, payment processing)
- Use Direct Connect to call on-premises APIs directly.

- Non-Critical
- Do not require immediate consistency with on-premises systems (e.g., user profile updates, non-critical data processing)

Already alleviates (some) Legacy System pain

Events and stream processing

- Data replication from on-premises to the Cloud
  - Readily available for Web, Mobile and Cloud system to use
  - Achieving high availability, reliability, and flexibility
- Update Legacy System via events
- Stream processing
- Event Sourcing

Target Architecture

- Incrementally move transaction processing to the cloud
- Build cloud-native versions of the legacy systems

- Strangler pattern
- Parallel Run: ensure cloud services handle the load and work as expected before decommissioning any legacy components

How to

Host, deploy, and scale the portal and application in the Cloud

- Multi-region, multi AZ
- EKS (Kubernetes platform)
- CloudFront (CDN)

Ensure security and privacy of the customer data and transactions

- Direct Connect
- VPC - public vs private subnet(s)
- AWS Shield, WAF
- IAM
- Encrypted communication

Monitor and troubleshoot

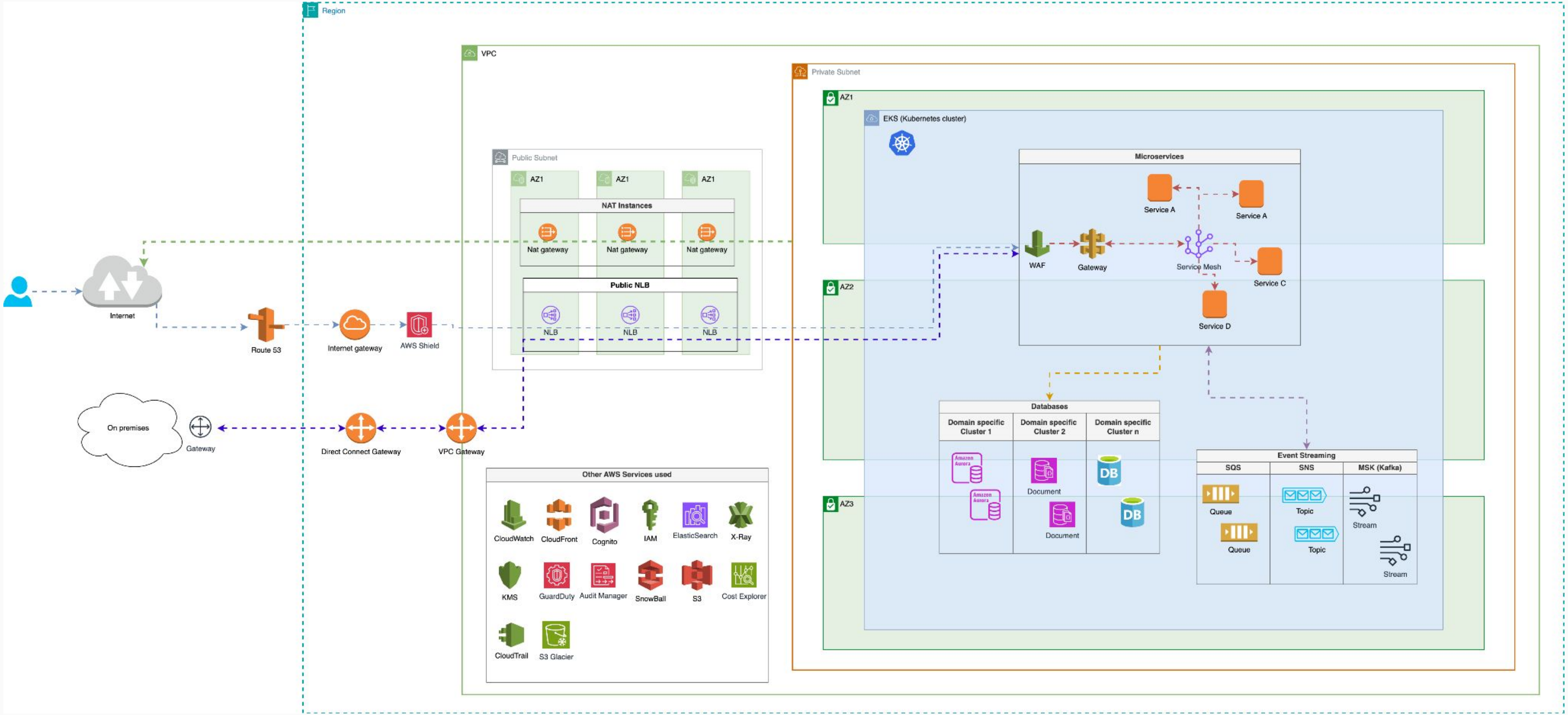
- CloudWatch
  - Metrics, Alarms, Logs, Events
- X-Ray
  - Distributed tracing
- ElasticSearch
- Consider 3rd party tools like Datadog

Optimise the cost and performance of the portal and application

- AWS Cost Explorer
- Auto Scaling to maintain performance and minimize cost

- Once we have patterns of usage we can consider Reserved Instances

Picture


Region
VPC
Private Subnet
AZ1
EKS (Kubernetes cluster)
Public Subnet
AZ1
AZ1
AZ1
NAT Instances
Nat gateway
Nat gateway
Nat gateway
Public NLB
NLB
NLB
NLB
Microservices
Service A
Service A
WAF
Gateway
Service Mesh
Service C
Service D
AZ2
Internet
Route 53
Internet gateway
AWS Shield
On premises
Gateway
Direct Connect Gateway
VPC Gateway
Databases
Domain specific Cluster 1
Domain specific Cluster 2
Domain specific Cluster n
Amazon Aurora
Amazon Aurora
Document
Document
DB
DB
Event Streaming
SQS
SNS
MSK (Kafka)
Queue
Queue
Topic
Topic
Stream
Stream
AZ3
Other AWS Services used
CloudWatch
CloudFront
Cognito
IAM
ElasticSearch
X-Ray
KMS
GuardDuty
Audit Manager
SnowBall
S3
Cost Explorer
CloudTrail
S3 Glacier

Q&A

Thank you