



SUMMARY OF **AWS EXPERTISE**



www.nix-united.com





Introduction

NIX is a leading system integration and software development service provider since 1994. We deliver innovative IT solutions for businesses and cover a broad spectrum of technologies and domains. NIX provides business analysis, solutions architecture, application development, remote administration, and system integration services that allow deployment and operating of any cloud-based solutions you need.

AWS Certificates:

- AWS Certified Solutions Architects - 11
- AWS Technical Professional - 50
- AWS Business Professional - 15

AWS Partner Status:



Service Delivery Designation:





Selected Case Studies

AWS CLUSTER FOR BIOMETRIC- AND VR-ENABLED HEALTHCARE PLATFORM

The main idea of the system is to activate healthy behaviors and improve health outcomes. The platform leverages VR technology to create targeted immersive experience for the patients to carry out tailored programs in treatment, lifestyle, educational, motivation, and research. The system collects, aggregates, and analyzes signals from individual sensors to dynamically personalize user experience using bio-metrics and data-driven algorithms. The server-side relies on 3 API services – each located on its own cluster.

Main goals

1. Ensure high availability and scalability of the system
2. Enforce HIPPA compliance.

Technologies

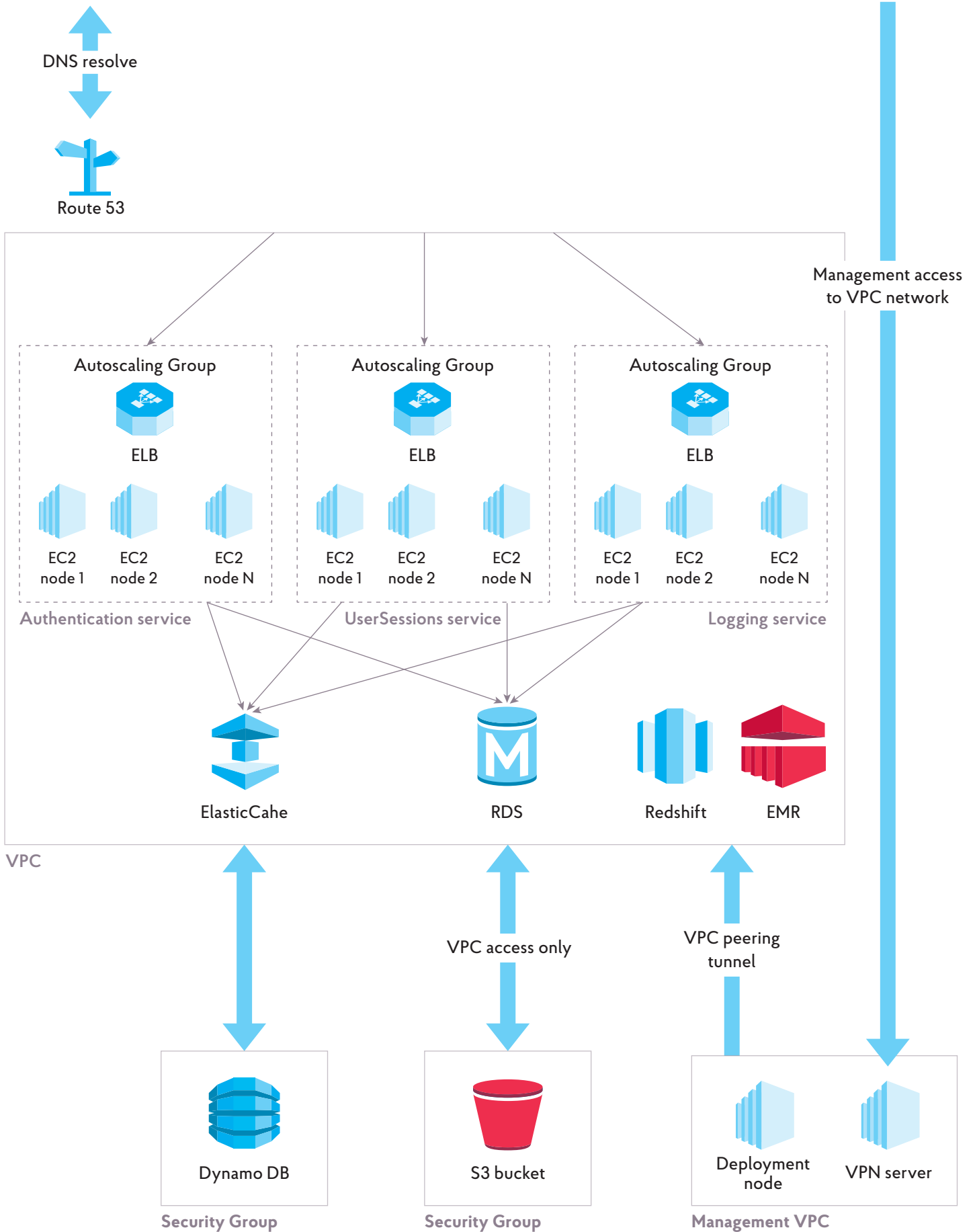
- Route53
- EC2
- ELB, EBS
- VPC
- ElastiCache
- S3
- DynamoDB
- Autoscaling
- RedShift
- EMR





Selected Case Studies

AWS CLUSTER FOR BIOMETRIC- AND VR-ENABLED HEALTHCARE PLATFORM





MARKETPLACE SOLUTION

The marketplace application allows you to connect restaurants to the system and solves the queue problem in restaurants and shops. Furthermore, users can place an order in the nearest restaurant or store of interest through the application, pay for it, and receive a notification when it is ready.

The app works through the API and allows the user to accept transactions, maintain cash, and transfer readiness statuses. The aim was to ensure the simplicity of adding new restaurants and smooth operation for connected points.

NIX team developed a **high-available cluster** using container orchestration on AWS.

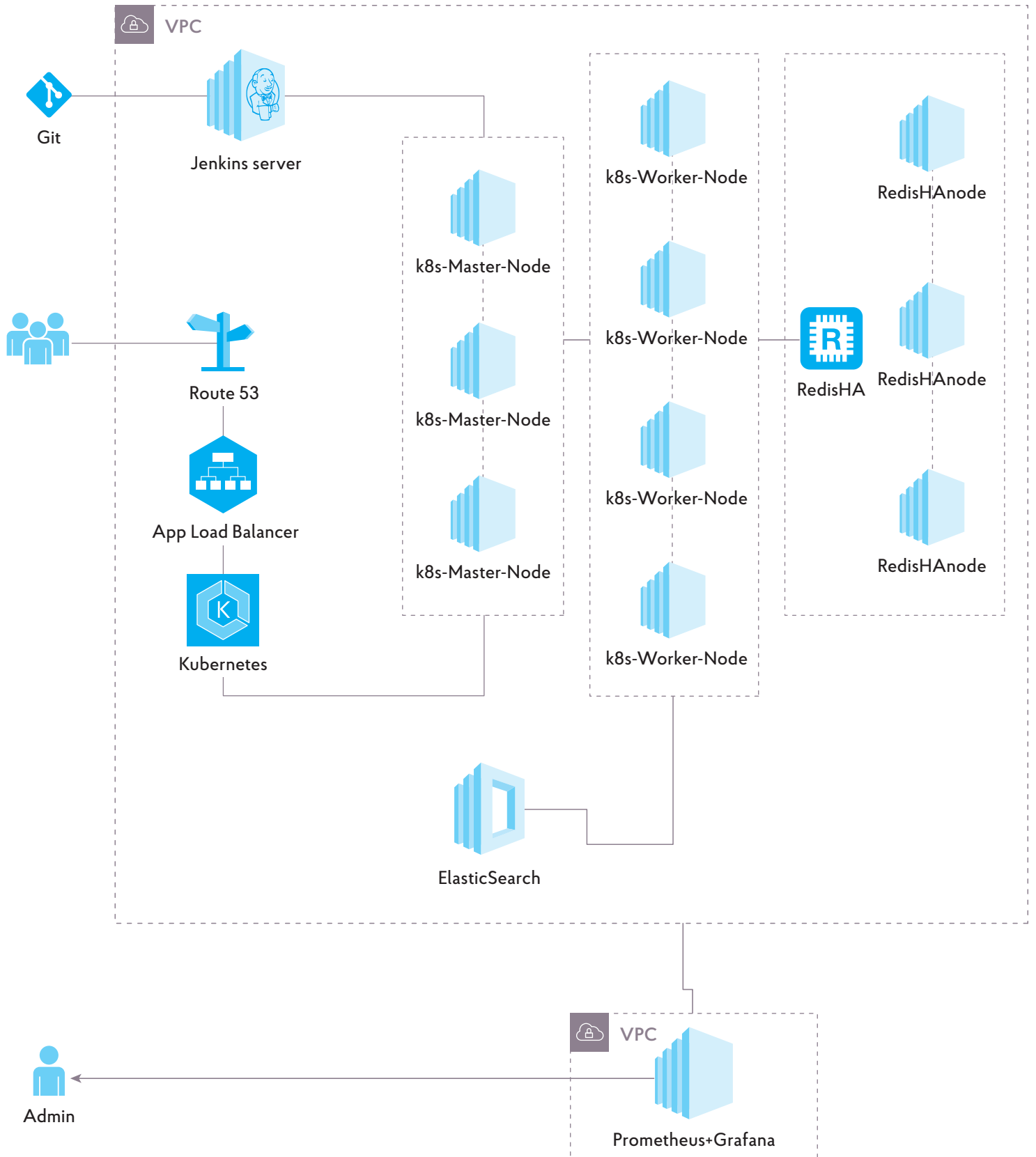
Main goals

1. Fast scaling – in case of a traffic spike
2. High-availability – to handle high ads and traffic
3. Easy management – the number of nodes control, scaling of required resources, and costs optimization.

Technologies

- AWS Load Balancer
- Kubernetes with EC2 usage for nodes:
 - 3 x master nodes
 - 4 x workers
- RedisHA (3 containers)
- MondoDB replica set on 3 nodes
- Jenkins + Git for code deploy
- Fluentd (logs for Nginx) + Elasticsearch + Kibana
- Prometheus + Graphana for monitoring and metrics show







BROADCASTING SERVICE

Broadcast service provides access to various media resources on multiple smart devices. The team realized **serverless infrastructure** for highly-loaded systems and fast roll-out of updates. Infrastructure is capable to automatically scale and withstand high peak loads in case of any unforeseen events. It allows using typical development procedures to push deploys in multi-environment structures with roll-back functions. Furthermore, developers can make code changes working in Bitbucket and push modifications to CircleCI.

The environment grows according to specific requirements – the number of running nodes can be up to 99 under single AWS Load Balancer (with HTTPS protocol enable).

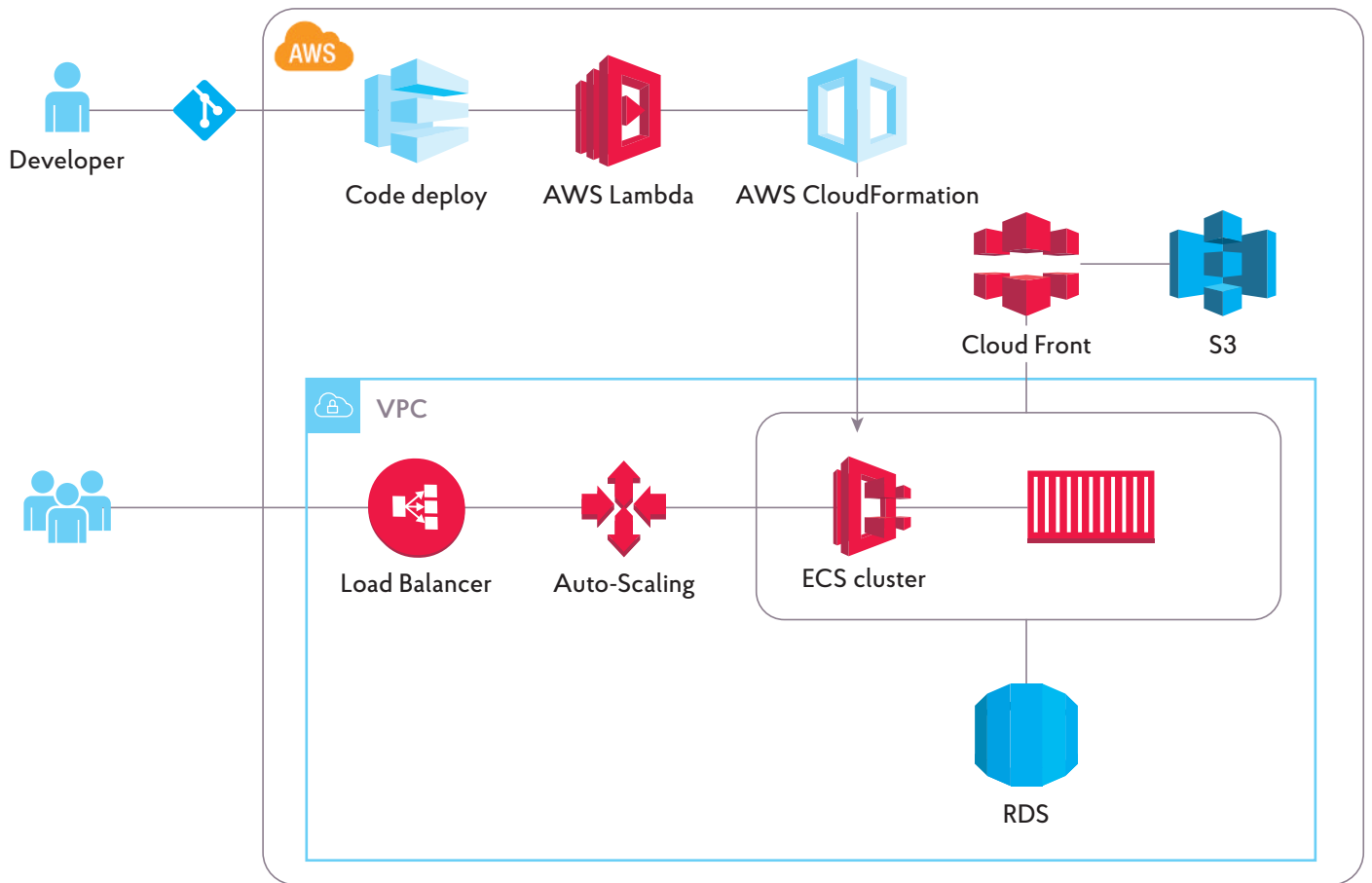
Due to rigorous user's access system each IAM user has its policy and permissions.

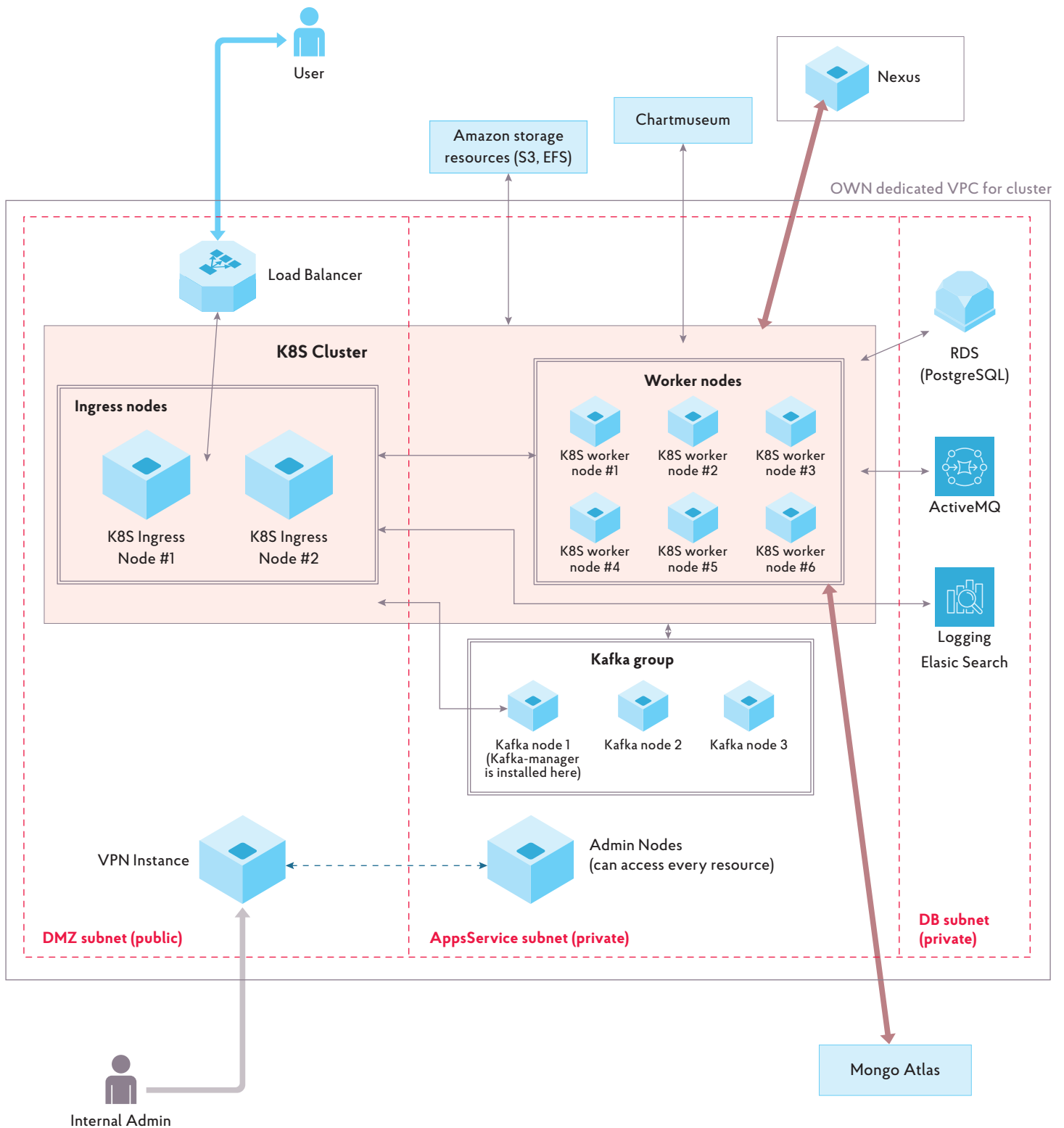
Main goals

1. Serverless reaching – direct access without static servers
2. Auto-scaling option realized on the CPU spike trigger
3. S3 and AWS Cloud Front usage as content storage and delivery
4. AWS RDS usage to store databases
5. CircleCI with AWS Cloud Formation for required Lambda functions
6. Ability to control all process by a set of AWS Lambda functions
7. Fast and easy deployment with health checks
8. Swift and controllable resources scaling
9. Transparent and straightforward costs management.

Technologies

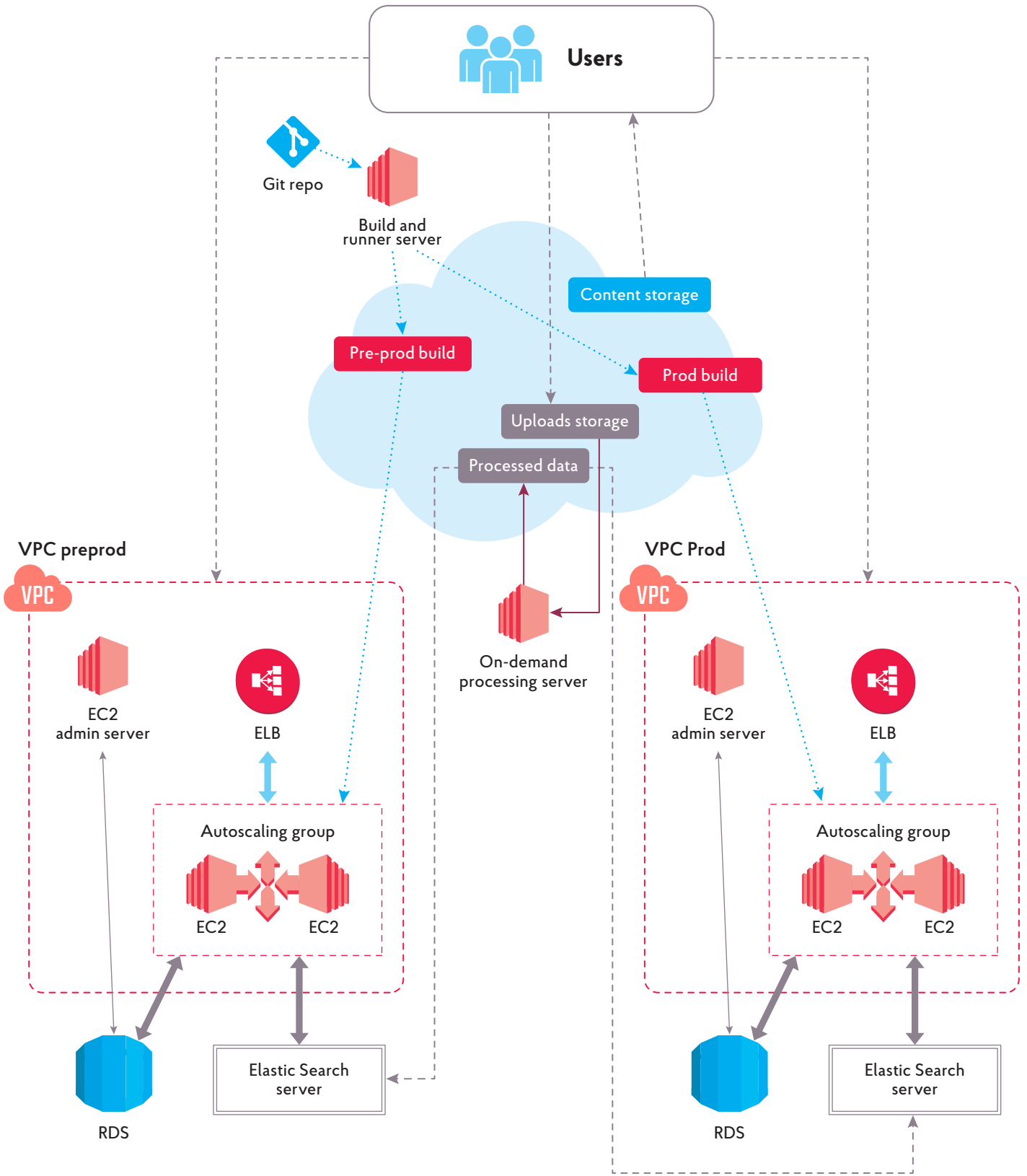
- CPU spike trigger
- S3 and AWS Cloud Front usage
- AWS Load Balancer
- AWS RDS
- AWS Lambda

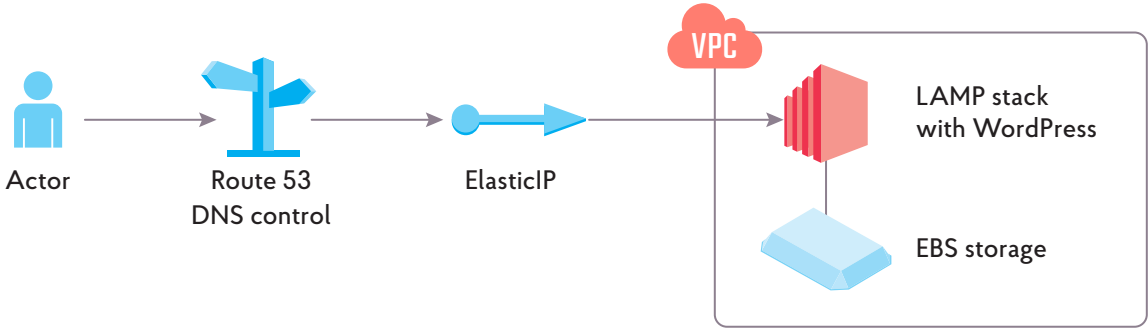




LEGEND

- | | | | | | |
|--|---------------------------------------|--|---------------------------------|--|--|
| | External VPN access | | Internal connection through VPN | | Subnet area |
| | External public access (live user) | | VPN peering connection | | Autoscale group area |
| | Internal connection between resources | | VPC area | | Other types of resources (type of resource is specified in the form) |





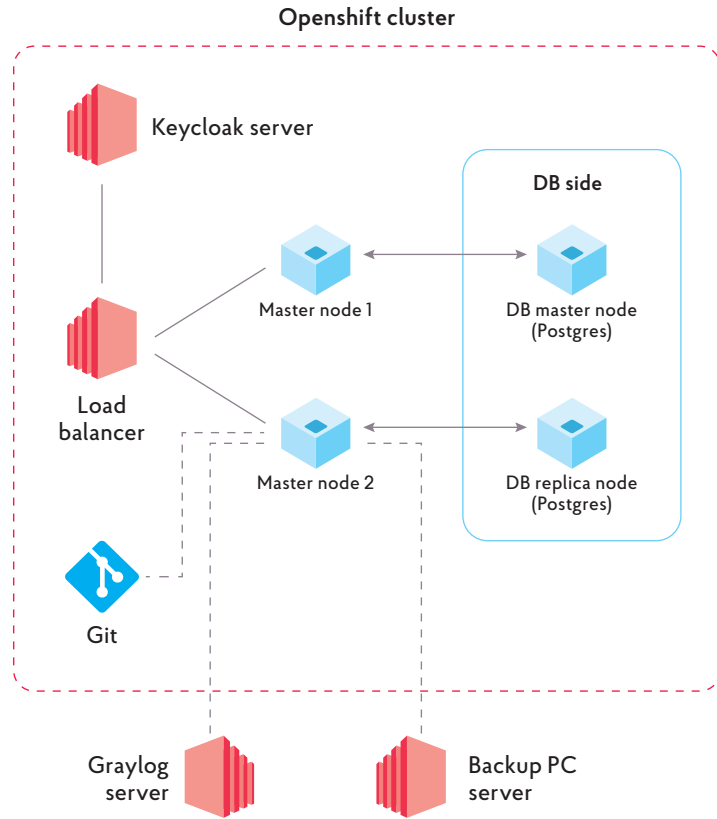
CONTACT US:

+1 (727) 900-8020

ask@nix-United.com

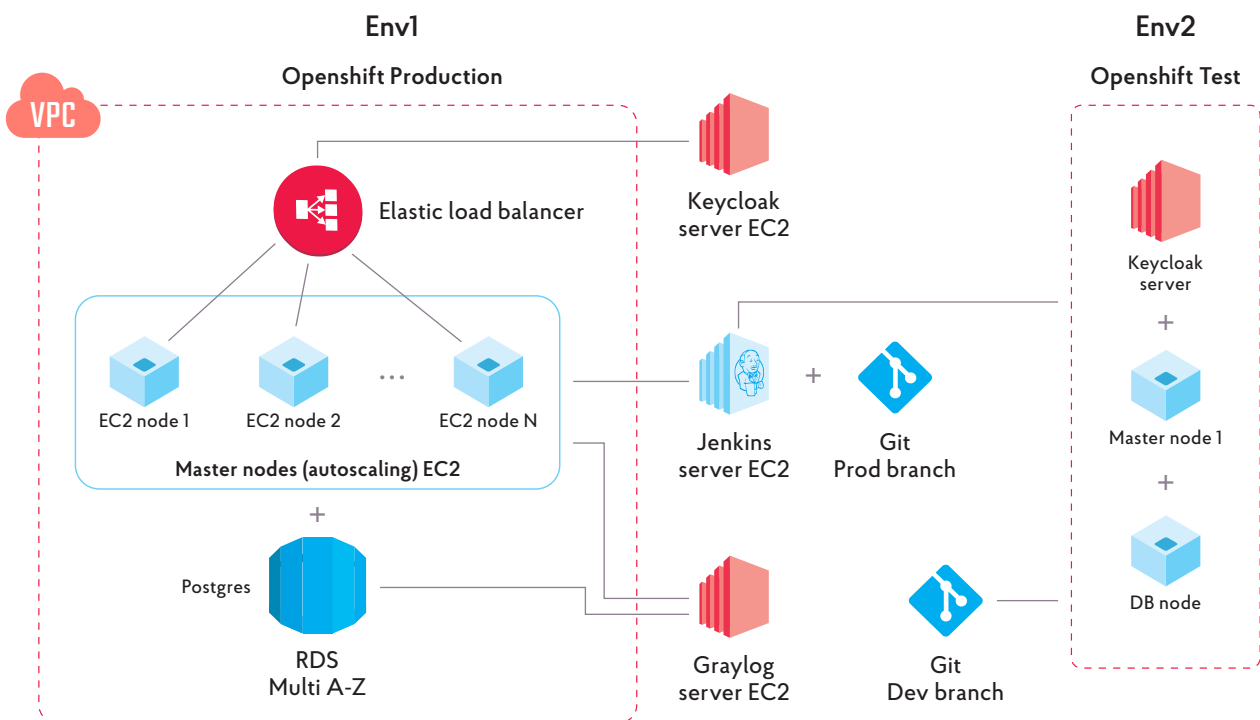
Cloud Sigma

BEFORE



AWS

AFTER





SUMMARY OF **AWS EXPERTISE**

AWS Cloud Migration



CONTACT US:

+1 (727) 900-8020

ask@nix-united.com

www.nix-united.com





CASE STUDY

HEALTH INSURANCE DISTRIBUTION PLATFORM

Migration from bare-metal

Introduction

Health insurance distribution and health plan management platform that helps to decrease administrative expenses, simplify the overall process of choosing the best-fit health insurance and integrate advanced health products. The client maintained its own servers and network since inception.

Reason for migration

The client product development organization of 300+ staff members was substantially limited by the existing bare-metal infrastructure. The In-house IT systems team having limited resources was overloaded with maintaining physical servers and the network and was severely challenged with the rising business needs of speed, scalability, security and disaster recovery. It was difficult to scale and fork new teams and develop new services with CI/CD in place from day one. We considered AWS as the best option in order to achieve the objectives of elasticity and administrative simplicity, as well as meeting HIPAA compliance requirements.

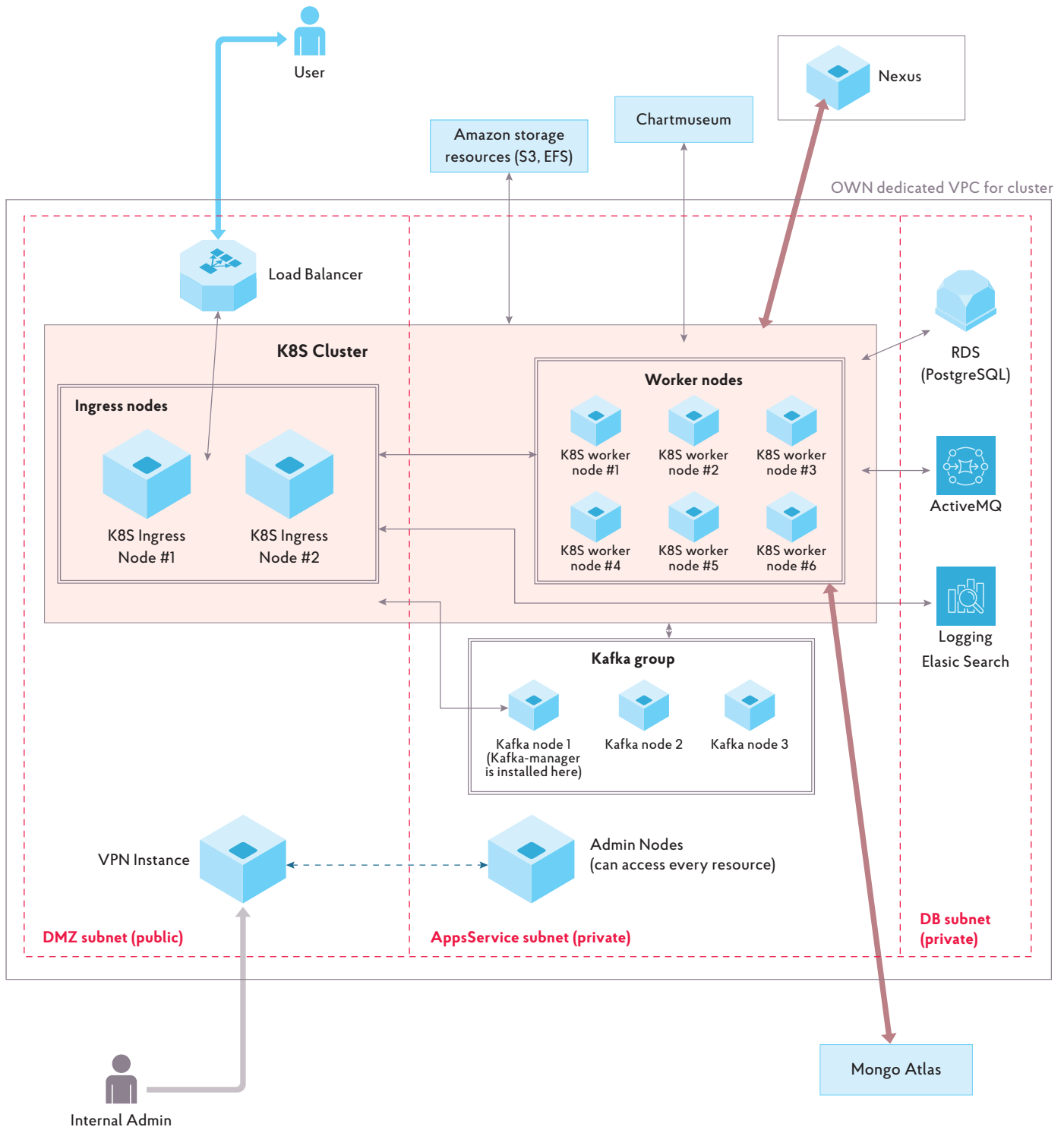
Solution

To ensure the deployment of cluster additional capacities, we created the automated deployment template to accelerate the setup of new instances from the initial 40-80 hours of manual labor to just 2-4 hrs of automated process.

We transferred the infrastructure from bare-metal servers to the Amazon province cluster (EKS), which is incorporated into two subnets of one VPC — public for the world and private for the team. Cuber's nodes are in autoscale groups and spread over public/private parts of the network. Worker nodes with the app are located in the private zone.

Throughout the process of moving to the AWS cloud, we did the following:

- Kubernetes cluster in own VPC
- Elastic Load Balancer and Public Ingress Node as for the public part
- OpenVPN for access to the private part of the network
- For AppsServices subnet — a Kafka cluster consisting of 3 servers
- Implementation of RDS PSQL, ElasticSearch, ActiveMQ, S3, CloudFront, Route53
- Special protection measures - App+DB_s were hidden by a private network and closed for access from the Internet
- Visible part — only LoadBalancers
- Access to cluster restricted by IAM+VPC endpoints
- MongoDB Atlas as a service for DB
- Auto-Scalable function and Userdata deploy function
- Supported by three CloudFormation templates — one for VPC, one for RedHat SSO, and one for everything else
- Use of peering connections, security groups, and IAM roles/policies.



LEGEND

- External VPN access
- External public access (live user)
- Internal connection between resources
- Internal connection through VPN
- VPN peering connection
- Subnet area
- Autoscale group area
- VPC area
- Other types of resources (type of resource is specified in the form)

Migration summary

Project Duration - 5 months (initial system review, new environment configuration, data migration, deployment, DNS switching to work on new IPs).

BARE-METAL	AWS
Dedicated bare-metal servers with Kubernetes and virtual machines	Separate server per service
10 simultaneous users	About 100 simultaneous users in a minimal state and up to 10,000 in a boosted state
40-80 hours for environment deployment	4 hours max for environment deployment
Manual deployment	Automated deployment recipes





CASE STUDY

SAAS DEVOPS PLATFORM

Migration from CloudSigma

Introduction

An online service that allows inexperienced software developers or DevOps experts that have a lack of time/resources acquiring access to advanced dev-ops technology, without an administrative burden, in just a few clicks. It allows developers to deploy and scale their pet projects using an extensive registry of development and CI/CD tools available from the cloud-based service.

Reason for migration

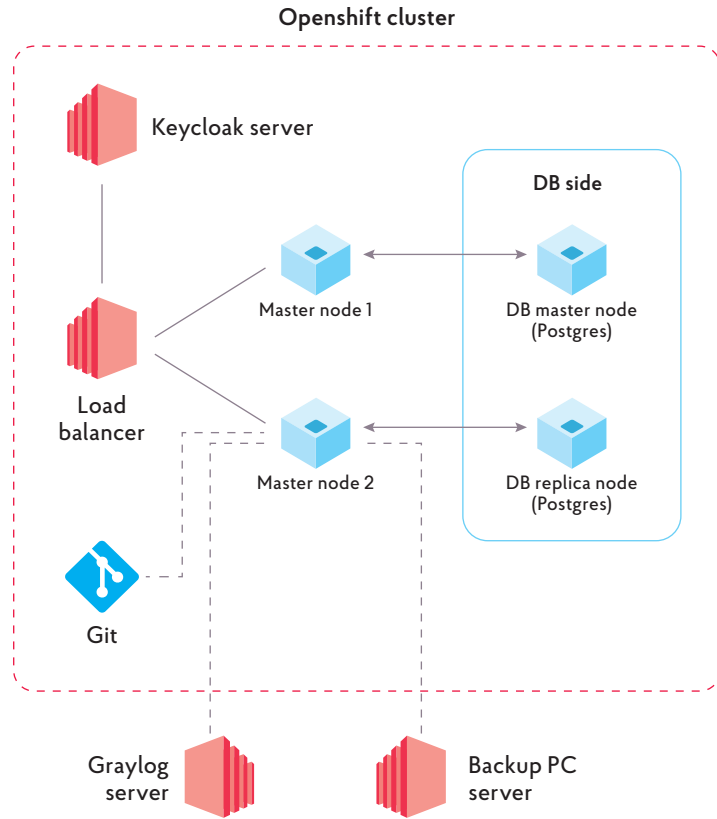
The limited capacity and low performance of the initial hosting didn't support public testing and production purposes. AWS allowed to advance infrastructure, support Openshift, Jenkins and other cutting-edge technologies.

Solution

- Creation of production and development environments
- The development environment had a low version of the production stack and it was needed only for testing new features
- The production environment was created with:
 - Elastic Load Balancer
 - EC2 instances for Keycloak, OpenShift Master nodes, Jenkins, Graylog server side
 - Autoscaling for Openshift nodes
 - Jenkins + Git for CI\CD realization
 - RDS MultiA-Z as a high-loaded database side
- Realization of auto-scalable option
- VPC implementation for better security
- IAM users restriction roles
- Reservation of some instances for cost optimization.

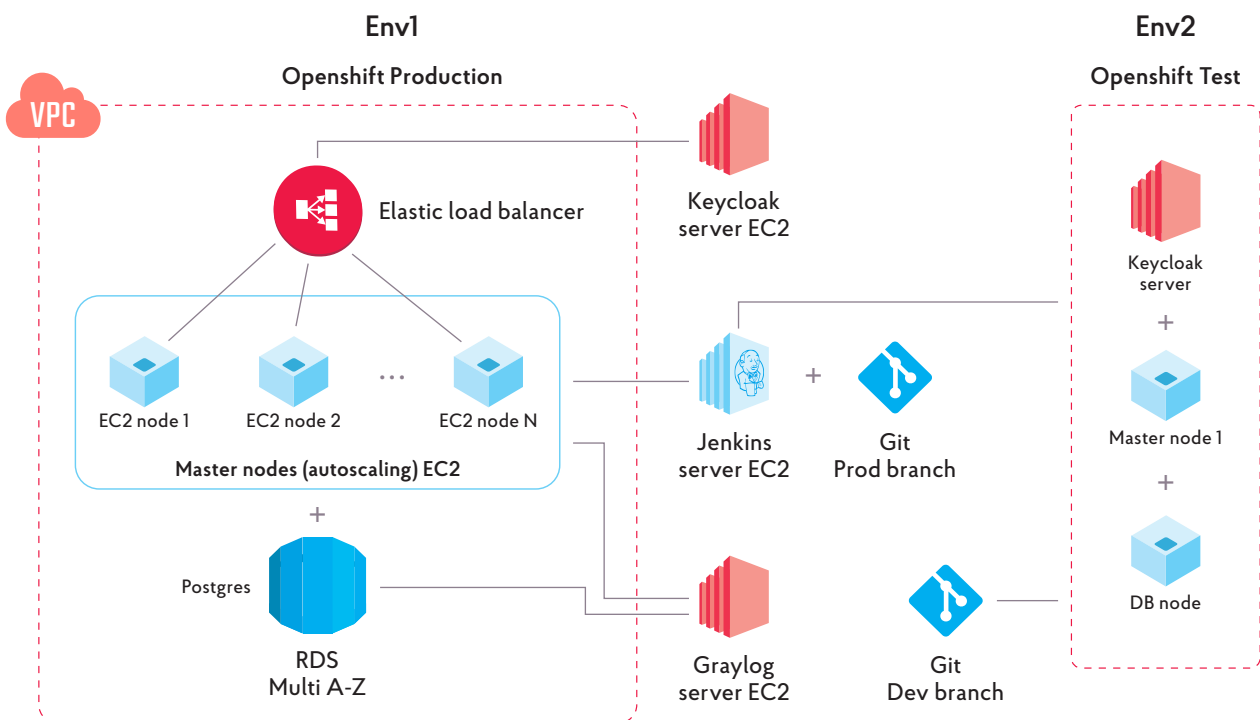
Cloud Sigma

BEFORE



AWS

AFTER



Migration summary

Duration - 2 months (initial system review, new environment configuration, data migration, deployment, and DNS switching to work on new IPs).

CloudSigma	AWS
Constant stack of 6 servers	7 servers + autoscaling for master nodes
Master-Slave DB replica set	MultiA-Z RDS instance as high-redundancy DB set
One network for all users (lack of free IPs)	Restricted network for each user
About 500 simultaneous users and environments	Unlimited amount of simultaneous users and environments due to auto-scaling function
Limited disc space - about 500 Gb storage at all	Unlimited storage, since EBS can be added on the fly





CASE STUDY

CLOUD MIGRATION FOR ADS AGGREGATOR

Migration from Digital Ocean

Introduction

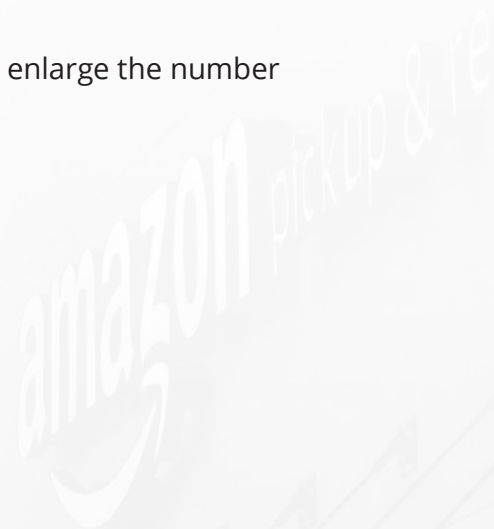
An online automotive marketplace that assists sellers and more than 20,000 dealerships across the U.S., connecting buyers with all models of vehicles for sale. The marketplace consists of three separate websites for different types of users and each requires its own high-efficiency group of resources.

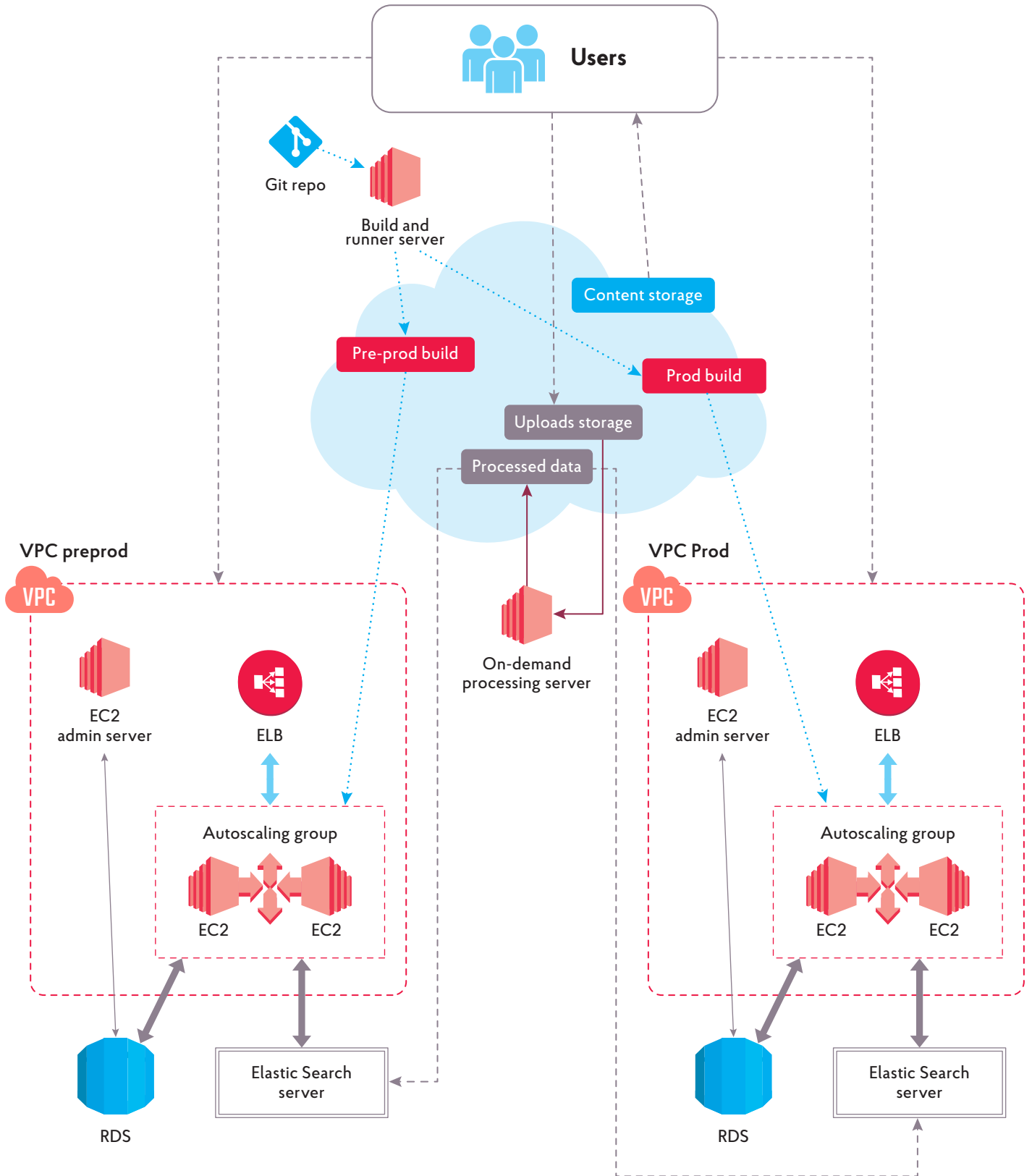
Reason for migration

Realization of cross-region work to ensure high load for different regions and boost performance for each website. AWS supports the cross-region system natively and allowed to do it more cost-effectively than on Digital Ocean. Also, frequent problems with the hoster's stability motivated the client to find more fail-safe hosting.

Solution

- Two auto-scalable environments setup in AWS in different VPCs
- ELB as a load-balancing stage — the ability to divide the load and enlarge the number of nodes to the cluster easily
- EC2 instances were used as a web-server side
- RDS was used as a DB side
- Elasticsearch servers for big requests processing
- CI/CD process realization.





Migration summary

Duration - 4 months (initial system review, new environments configuration (production and pre-production), data migration, deployment, DNS switching to work on new IPs, CI/CD realization).

Digital Ocean	AWS
100,000 simultaneous users	300,000+ simultaneous users
Single zone work, low redundancy	Multi-region infrastructure and cross-region work. High-redundant cluster
8-second search request processing time	4-second search request processing time
Medium-score - about 85% of 100% page speed ranking	High-score page speed ranking
4-7 times failures per year due to hosting hardware problems	No failures were detected due to AWS hardware after the migration was completed





Services

- Cloud applications architecture and development
- Cloud environment design and implementation
- Migration to the Cloud
- Modernization of monolithic systems into service-based solutions
- CI/CD implementation, transformation, and maintenance
- TechOps and 24x7x365 server tech support

Expertise with AWS

Computing	EC2, Lambda, ECS, Fargate, ElasticBeanstalk, Amazon Kubernetes Service
Databases	Aurora, RDS, DynamoDB, ElastiCache
Storages	S3, EBS, EFS, Glacier
Networking	VPC, ApiGateway, CloudFront, Route53, Load Balancing
Analytics	Amazon ElasticSearch, Redshift
App Integration	SNS, SQS
DevOps Tools	CodeCommit, CodeDeploy, CodeBuild, CodePipeline, ContainerRegistry
Management Tools	Cloud Trail, CloudFormation, AWS cli, AutoScaling, VPN, IAM