Prodapt

Maximize value from cloud migration

Implement a holistic approach for effective transition of intricate online charging system and network service order management to cloud

Lakshmiprasad A | Baskar Ramasamy | Balakrishnan N | Priyankaa A

Current state: Service providers are reluctant to cloudify their Online Charging System (OCS) and network Service Order Management (SOM)

OCS helps to charge customers for the services they use in real-time and network SOM ensure that quality services are delivered to customers on time. These critical systems are the frontiers that need **cloud migration.**
Gartner states, “Cloud will be the centerpiece of new digital services and experiences, which is why 40% of all enterprise workloads will be **deployed in Cloud** over the next few years”.

Moving OCS and network SOM to the cloud gives a phenomenal advantage with the evolving 5G landscape and virtualization

- Convenient scalability for multi-national service providers with M&As
- Easy integration with new 5G services
- Economies of scale
- Flexibility to quickly adapt new technologies & services with Cloud-based infrastructure
- Out of the box AI, analytics and monitoring
- Improved operational efficiency

Despite the advantages, service providers are hesitant to move their OCS and network SOM to the cloud. Here is why:

- It contains **sensitive data** such as invoices, Call Detail Records (CDRs), history of customers’ usage, financial transactions, and porting information
- Handing heavy payloads and workflows juggling through the **integration-heavy** architecture with least to nil latency is cumbersome
- Ensuring adherence to both local and national **data regulatory norms** is difficult

Moving OCS and network SOM to Cloud presents significant challenges, due to a heavy dependency of telco architecture on complex networks and integrations

- Lack of mechanisms to integrate with "distant" network elements - handing provisioning, activation, and number porting with minimum latency
- Unavailability of cloud data center near customer location resulting in latency, impacting real-time rating and charging
- Lack of methodology to address the complexities in setting up a multi-country deployment architecture for billing, rating, charging, and networks
- Threats of sensitive data breaches

How does this impact service providers' operations?

- Increasing costs of operating and maintaining BSS
- Complex processes and elongated releases

Taking billing, online charging, and network SOM to the cloud offers several benefits in terms of agility, elasticity, and process efficiency. However successful cloud migration requires a design ground up, end-to-end automation, and continuous security to adapt to real-time changes in the environment quickly.

Key focus areas for successful cloudification of complex OCS and network SOM

1

Design a deployment blueprint for Cloud

- Enable high availability, resilience, disaster recovery and zero latency in online charging and rating and network SOM applications

2

Set up infrastructure automation and Cloud CI/CD

- Adapt to rapid changes in the OCS and network SOM ecosystem and accelerate time-to-market with fully automated pipelines

3

Leverage DevSecOps to secure the critical data

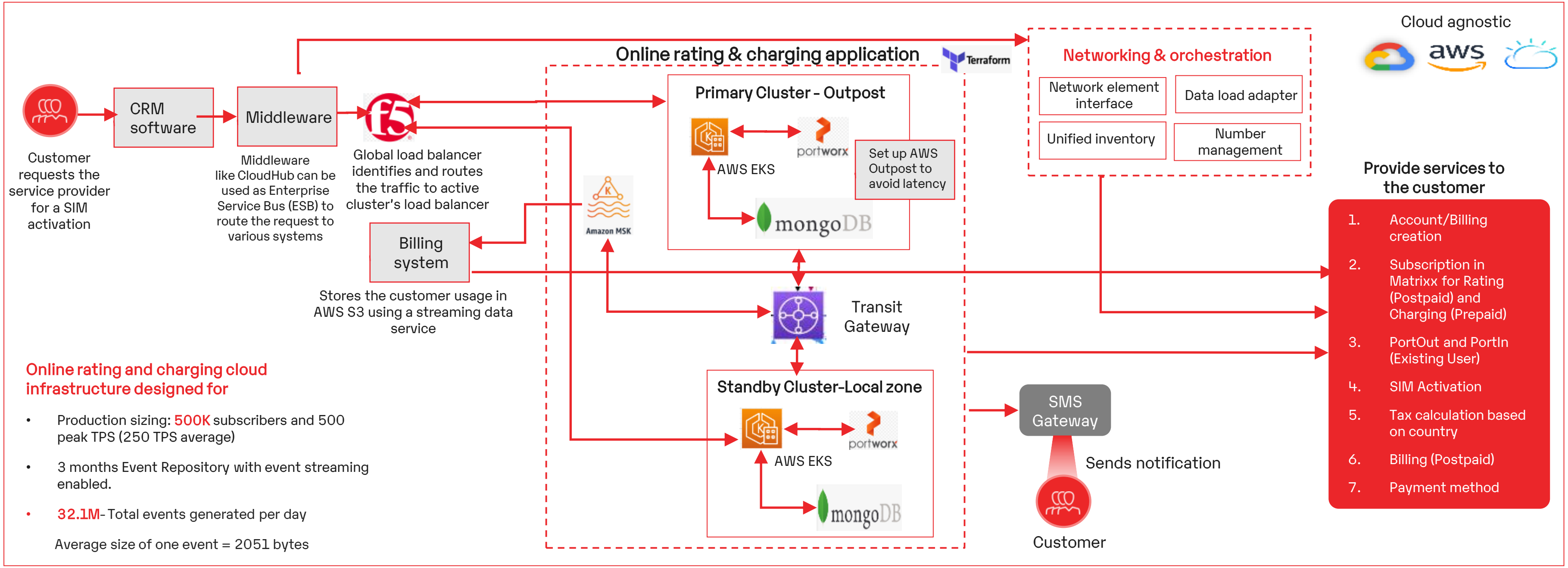
- Enable proactive identification of vulnerabilities and fix them before they impact the business

Implementing the key focus areas above ensures a successful cloud migration of OCS and network SOM applications. Reduce the time spent spinning up a new environment by up to 80%, decrease resources and effort by 30%, and accelerate secure releases by 2x.

Design a cloud-driven deployment blueprint to achieve high availability and zero latency in OCS and network SOM

While moving billing, online charging, and network SOM to the cloud are critical for service providers, designing a deployment blueprint to address the challenges of handling complex payloads and integrations in real-time is vital. Further, a **multi-country model** includes numerous rule-based workflows and geo-specific components, which require an architecture supporting **high availability** and **minimal latency**.

Reference deployment architecture for Cloud to achieve real-time rating and charging



Customer requests the service provider for a SIM activation
CRM software
Middleware
Middleware like CloudHub can be used as Enterprise Service Bus (ESB) to route the request to various systems
Global load balancer identifies and routes the traffic to active cluster's load balancer
Billing system
Stores the customer usage in AWS S3 using a streaming data service
Amazon MSK
Online rating & charging application
Terraform
Primary Cluster - Outpost
AWS EKS
portworx
mongoDB
Set up AWS Outpost to avoid latency
Transit Gateway
Standby Cluster-Local zone
AWS EKS
portworx
mongoDB
Networking & orchestration
Network element interface
Data load adapter
Unified inventory
Number management
Cloud agnostic
aws
SMS Gateway
Sends notification
Customer

Provide services to the customer

1. Account/Billing creation
2. Subscription in Matrixx for Rating (Postpaid) and Charging (Prepaid)
3. PortOut and PortIn (Existing User)
4. SIM Activation
5. Tax calculation based on country
6. Billing (Postpaid)
7. Payment method

Online rating and charging cloud infrastructure designed for

- Production sizing: 500K subscribers and 500 peak TPS (250 TPS average)
- 3 months Event Repository with event streaming enabled.
- 32.1M- Total events generated per day

Average size of one event = 2051 bytes

Design a cloud-driven deployment blueprint to achieve high availability and zero latency in OCS and network SOM

1 A B 2 3

Recommendations for robust cloud design

1

Ensure minimal latency

Leverage tools such as **AWS Outpost** and **Google Anthos** to overcome the **unavailability of cloud data centers** near the customer. Record customer usage details and create Call Detail Records (CDRs) in real-time, thus avoiding latency

2

Achieve high availability

Deploy the micro-services using **helm chart** in Kubernetes cluster as PODs to achieve high availability in OCS and network SOM applications

3

Curb data loss

Implement **Portworx**, a persistent storage that converts EBS to NFS storage depending on the application demand. It aligns with AWS Outpost and distributes the EBS storage across applications, avoiding huge data loss during application/server termination

4

Ensure Disaster Recovery

Scatter the active and standby Kubernetes clusters across two locations-primary in the customer data center and secondary in the cloud to achieve Disaster Recovery (DR)

Prodapt

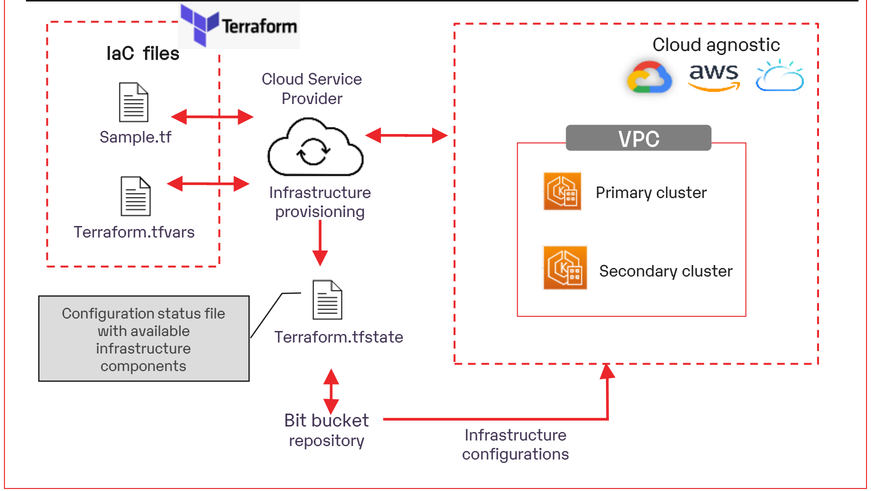
Set up infrastructure automation and Cloud CI/CD

Adapt to rapid changes in OCS and network SOM ecosystem

1 2 A B 3

Managing the scale of today's online charging and network SOM infrastructure is challenging, and these critical systems undergo numerous changes. Further running multiple releases in parallel, juggling numerous lines of environments, balancing feature development, and exhaustive migrations are cumbersome. Hence rapid infrastructure spin-up and Cloud CI/CD on a multi-tenanted distributed architecture are required.

Automation of OCS & network SOM infrastructure on cloud using Terraform



Recommendations

- Leverage open-source Infrastructure-as-Code software such as **Terraform** to automatically cloudify the OCS and network SOM infrastructure. It enables rapid **spin-up of multiple environments** such as prepaid, postpaid, and B2B in **parallel**, thus reducing the time to spin up new environments
- Deploy network SOM application binaries in **AWS EKS** or **Google GKE.** It enables addition of multiple new users and economies of scale without the infrastructure bottleneck
- Leverage **Katalon** to automate daily regression test cases and quickly notify when OCS and network SOM functionalities fail. It reduces the testing effort by 30%, and improves service fulfillment and customer experience

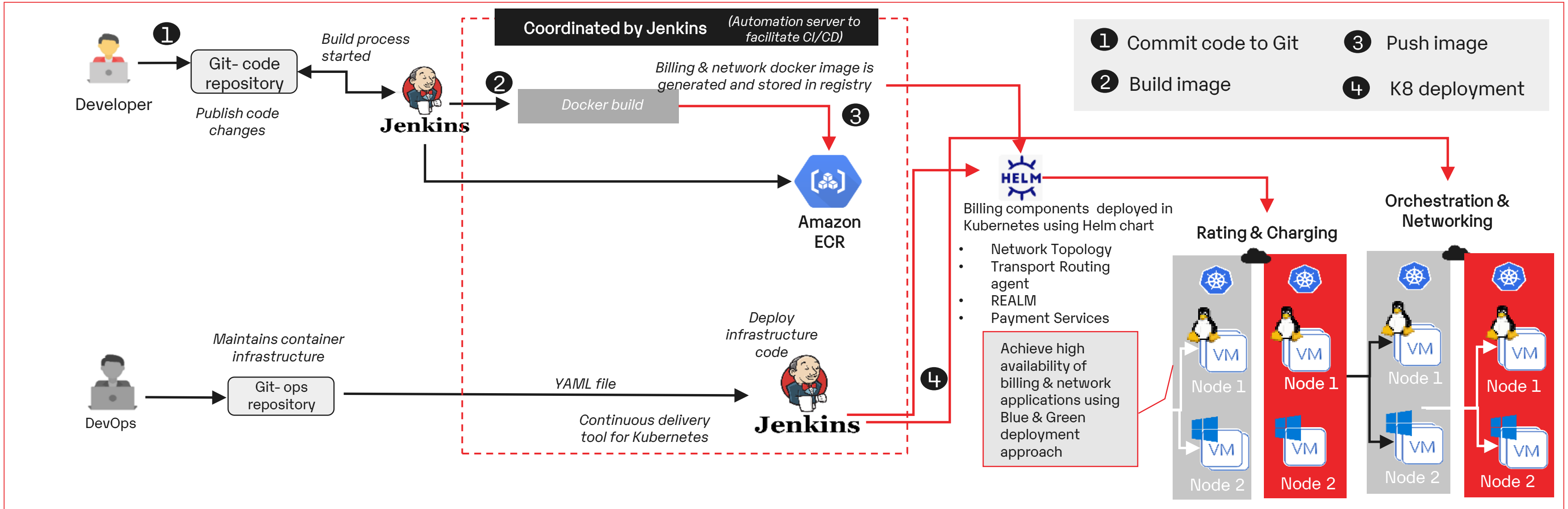
Setting up infrastructure automation helps to auto-scale the OCS and network SOM applications and reduce the time to spin up a new environment by up to 80%.

Set up infrastructure automation and Cloud CI/CD

Adapt to rapid changes in OCS and network SOM ecosystem

Although moving OCS and network SOM to the cloud offers better velocity and scale, it increases complexity. Hence it requires increased automation, Continuous Integration (CI) and Continuous Delivery (CD) tools designed to support the complexity and accelerate releases.

Cloud CI/CD pipeline with Blue and Green deployment approach to accelerate OCS and network SOM solutions



1 Commit code to Git
2 Build image
3 Push image
4 K8 deployment

Developer
1
Git- code repository
Publish code changes
Build process started
Jenkins

Coordinated by Jenkins *(Automation server to facilitate CI/CD)*

2
Docker build
Billing & network docker image is generated and stored in registry
3
Amazon ECR

DevOps
Maintains container infrastructure
Git- ops repository
YAML file
Deploy infrastructure code
Jenkins
Continuous delivery tool for Kubernetes
4

HELM
Billing components deployed in Kubernetes using Helm chart

- Network Topology
- Transport Routing agent
- REALM
- Payment Services

Achieve high availability of billing & network applications using Blue & Green deployment approach

Rating & Charging
VM Node 1
VM Node 2
VM Node 1
VM Node 2

Orchestration & Networking
VM Node 1
VM Node 2
VM Node 1
VM Node 2

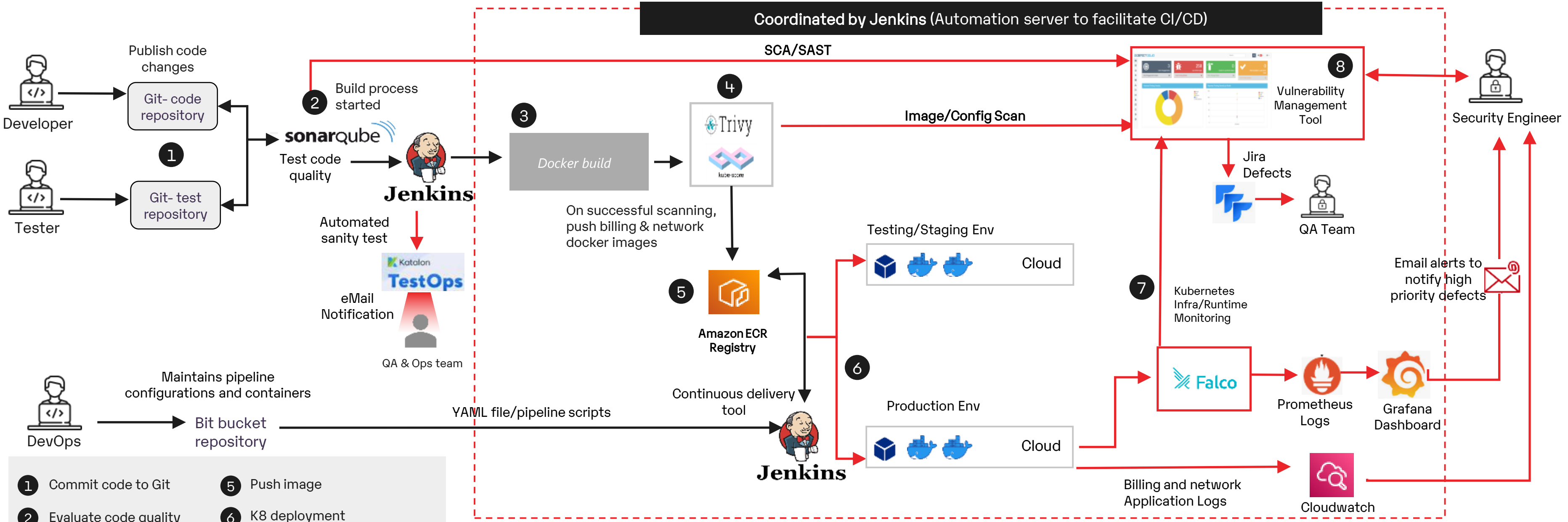
Enables deployment of OCS and network SOM applications into **six** different environments in Kubernetes using Helm chart

Recommendations

- Leverage **Blue and Green deployment approach** to achieve zero-downtime and manage **high-availability** of OCS and network SOM applications
- Deploy Rating and Charging functionalities such as AutoPay, Revenue Assurance and Fraud Management as microservices with auto-healing capability using **Helm chart**

Secure critical data and detect up to 70% vulnerabilities by leveraging DevSecOps

Once the deployment architecture is designed to handle heavy payloads and integrations in OCS and network SOM , service providers must **ensure the security of data and the payloads that get transferred beyond the perimeter.** Implement **DevSecOps** to **strengthen the security of sensitive data** such as invoices, Call Detail Records (CDRs), financial transactions, porting information and taxation and detect the vulnerabilities before moving to production.



1. Commit code to Git
2. Evaluate code quality
3. Build code
4. Scan image
5. Push image
6. K8 deployment
7. K8 hardening and monitoring
8. Vulnerability management

Leverage DevSecOps to enhance security, and combat vulnerabilities before they impact the business. In addition, enable continuous security monitoring to protect the environment.

Secure critical data and detect up to 70% vulnerabilities by leveraging DevSecOps

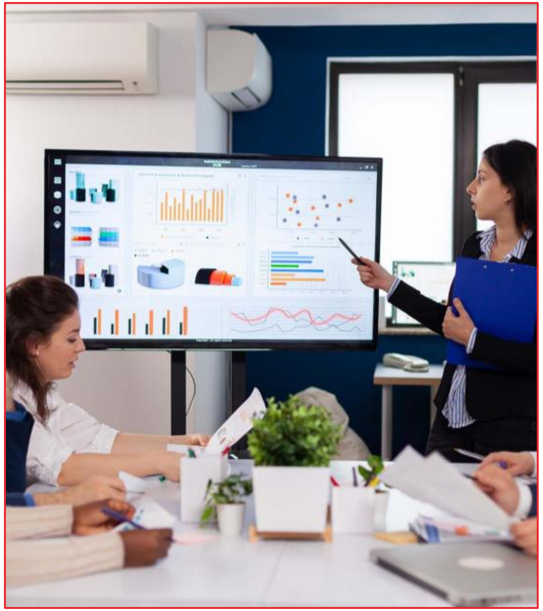
1 2 3 A B

Recommendations

- Implement **Trivy** to scan the billing and network docker images and send the defects to **DefectDojo,** which helps in tracking and managing the vulnerabilities
- Implement **Falco** to continuously monitor and detect unexpected behavior in Amazon EKS cluster containers
- Gain real-time view of vulnerabilities by integrating **Falco** with incident response workflow systems through Alert Manager and Webhooks
- Detect the critical vulnerabilities, push them to **Jira** and assign them to the QA team for better tracking
- Leverage **Amazon managed Prometheus,** a single time-series database to collect metrics for both applications in outpost and standby cluster
- Implement tools such as **AWS Cloudwatch and Azure Monitor** to effectively capture and debug the application logs

Business benefits achieved by a leading service provider post successful cloud migration of OCS and network SOM systems

Implementing the key focus areas as discussed in this insight, resulted in the following benefits.



80% reduction in time to spin up the infrastructure for a new environment

70% vulnerabilities identified before impacting the business

2X faster and secure OCS and network SOM application releases

Accelerated time-to-market by spinning multiple environments quickly

Thank you

insights@prodapt.com