

A background image of a golden wheat field under a bright, warm light, possibly at sunrise or sunset. The wheat stalks are in focus in the foreground and become more blurred towards the background.

OLF

NETWORKING

LFN Developer & Testing Forum



OLF NETWORKING

LFN Developer & Testing Forum

The Path to a Production-Grade ONAP

**Ahmad Khalil –
Ali Fouladgar –
Min Sang Yoon –
Mohammad Nurujjaman –**

Anti-Trust Policy Notice

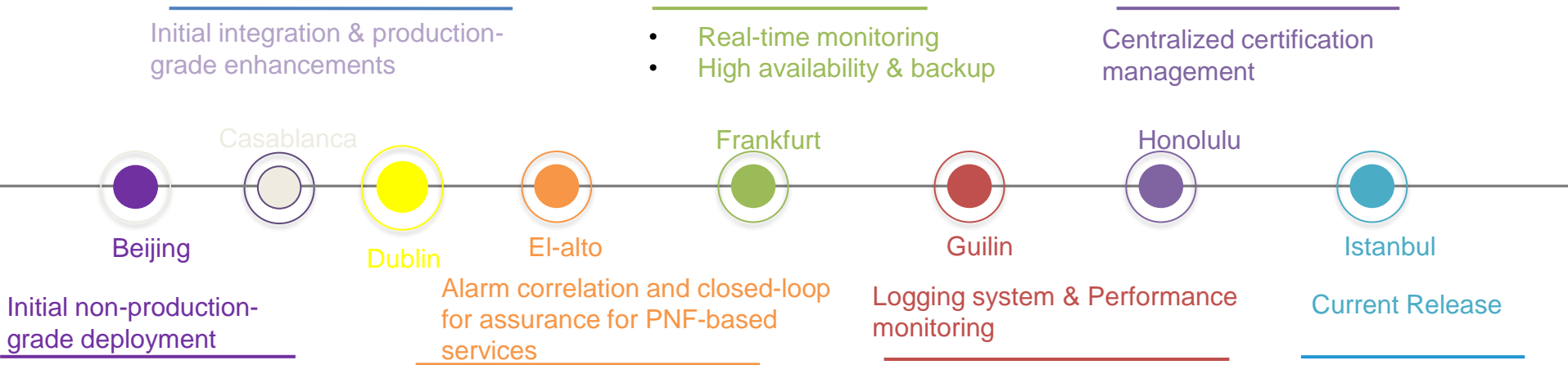
- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrustpolicy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

CONTENTS

- Overview
- Deployment Timeline
- Deployment & Support
- LOGS & Metrics: Architecture
- Monitoring & Troubleshooting
- Demo (Video)
- Q&A

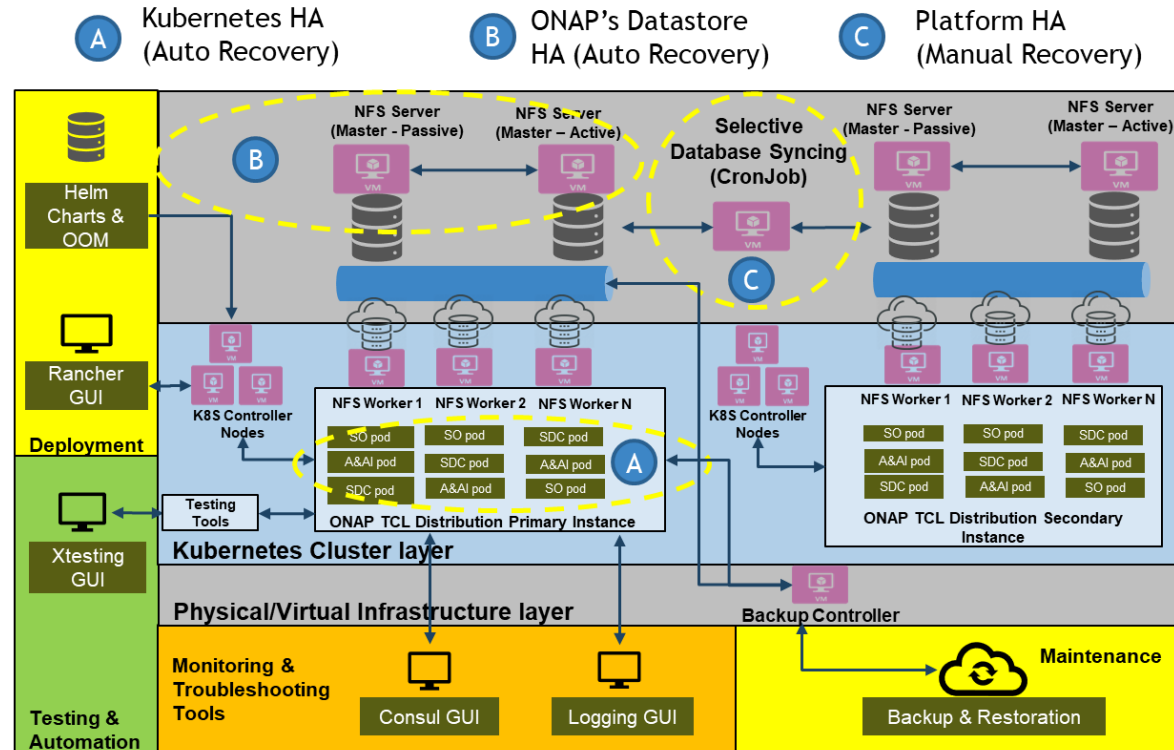
- TATA Communications has deployed all ONAP versions since Beijing release
- Deployed & configured cloud Infrastructure and Kubernetes to support ONAP
- Deployed ONAP core components needed for use case (e.g., SDC, SO, SDN-C, A&AI, DCAE, OOM, Consul, External APIs, DMaaP, MSB, AAF, Portal, LOG, etc...) and identify what requires development and what can be used as-is.
- Deployed and configured monitoring & troubleshooting tools needed for production-grade readiness
- Enhanced the centralized logging system to capture information to operate, troubleshoot and report on performance of the platform and its components

DEPLOYMENT TIMELINE



DEPLOYMENT & SUPPORT

- Deployed high available ONAP on application/pod level, Kubernetes/datastore level as well as platform level
- Disaster recovery is ensured by Frequent backup of the states of the platform to external storages
- Developed Helm Charts as well as manual procedure to support rolling upgrade from old to new release including data migration of developed artifacts. This used to automate upgrade process and impacted: MariaDB, Cassandra, PostgreSQL, A&AI, SDC, DMAAP, SDNC
- Consul GUI: Developed/configured scripts to ensure a complete and healthy deployment of ONAP custom image that is specific to TCL use case with proper real-time monitoring for fast support, including the deployment of the required disaster recovery options.
- Logging GUI: Developed/configured tools and integrate them with ONAP to ensure proper and fully functional ONAP centralized logging system (Elasticsearch, Logstash, and Kibana)



Real-Time Health Check Monitoring

- Consul is fully customized and is used to monitor the Health-check of 20+ micro-services related to use case
- Developed scripts and configured consul to collect and show outcomes in dashboard (Network discovery, websocket client for 3rd party controller - DMaaP integration, verification of deployed services on underlying network, etc...)
- Customized severity of alarms to - Critical (Red): service is not responding at all, Warning (Yellow): service is responding but not working as desired and Healthy (Green): Service is running
- More comprehensive customization can be achieved based operational rules

The screenshot displays the Consul web interface. The top navigation bar includes 'Services', 'Nodes', 'Key/Value', 'ACL', and 'Intentions'. The main content area is divided into two panels. The left panel, titled 'Services 23 total', shows a list of services with their health status. A dropdown menu is open, showing options: 'Passing' (green checkmark), 'Warning' (yellow triangle), 'Critical' (red X), and 'No checks' (grey square). The services listed are: 'A&AI Model Loader' (1 Instance, Passing), 'A&AI Synapse Data Routing Service' (1 Instance, Passing), 'A&AI UI Backend Service' (1 Instance, Passing), and 'Health Check: Application Authorization Framework' (1 Instance, Passing). The right panel, also titled 'Services 23 total', shows a similar list with 'Health Check: Network Discover - CronJob' (1 Instance, Passing), 'Health Check: ONAPWEBSOCKET client' (1 Instance, Passing), 'Health Check: SDNC' (1 Instance, Passing), and 'Health Check: SDNC - DB Host' (1 Instance, Passing). Below these panels, a detailed view for the 'Health Check: Network Discover - CronJob' service is shown. It includes the 'Service Name' and 'Node Name'. The 'Health Checks' section is active, showing a search bar and filters. The 'netdiscovery' health check is highlighted, showing its 'ServiceName', 'CheckID', 'Type', and 'Notes'. The output of the check is displayed in a code block, showing a successful status and timing information.

Service Name	Node Name
Health Check: Network Discover - CronJob	stable2-consul-75d99c956f-toplz

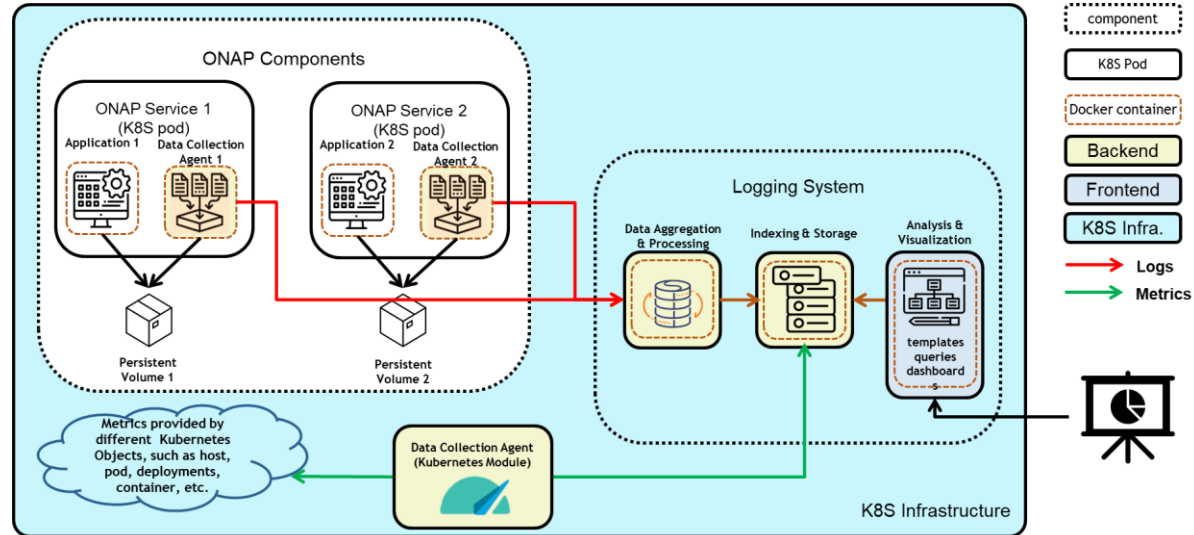
Health Checks	Tags & Meta
netdiscovery	

ServiceName	CheckID	Type	Notes
Health Check: Network Discover - CronJob	networkdiscoverycronjob	script	-

```
onap-auto-network-discovery */30 * * * * False 1 3m48s 2d19h
Success. Network Discovery is scheduled successfully. It is been running for last 2d19h, the last time it ran was 4m26s ago.
```


LOGS & METRICS: ARCHITECTURE

- Data Collection by “Filebeat” & “Metricbeat”
 - Both agents are installed on servers and custom-configured
- Data Aggregation & Processing by “Logstash”
 - Deployed to be used as a data pipeline for Elasticsearch
- Indexing & Storage by “Elasticsearch”
 - Elasticsearch is installed and custom-configured
- Analysis & Visualization by “Kibana”



MONITORING & TROUBLESHOOTING

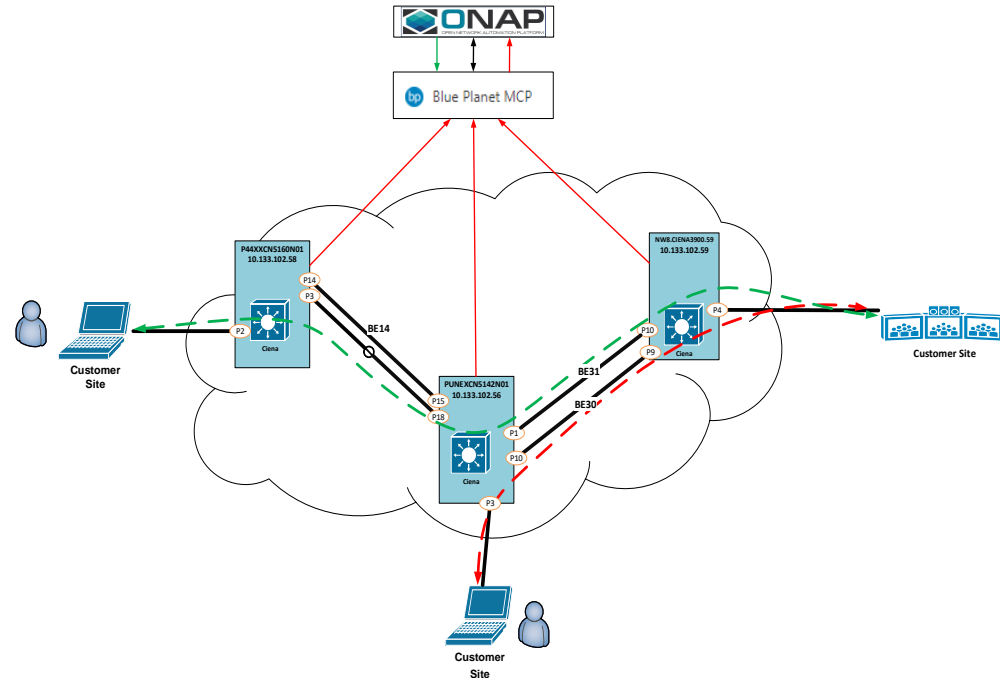
- Performance Monitoring: Configured Kibana to monitor current and historical metrics that show the state of infrastructure components (Node, POD resource allocations, usage and limits, etc...)
 - This will allow us to proactively adjust the required resources for each ONAP component before resource starvation and cause slowness to tasks performed by impacted PODs.
- Infrastructure troubleshooting: Configured the domain controller (also possible for NEs) to route their logs into ONAP's Logging system for a centralized location for troubleshooting to address failures in the network: node, interface, service, LAG, ring, etc...
- ONAP troubleshooting: Configured Filebeat agent to collect logs from ONAP component PODs and send it to Logging system (e.g., during a service provisioning, the operator can see the logs in the backend to get more insights and debug in case of failures)
- K8S troubleshooting using ELK Logging: This enable us to get all the infrastructure-related logs into centralized logging system to facilitate debugging and deep analysis on infrastructure issues on both Worker and Controller nodes, check the real-time logs on deployment issues, etc...

1. Demo Test Case #1: Scheduling a BoD (Customer Experience)

- Feasibility Check to upgrade a provisioned service (i.e., BoD)
- Feasibility Rejection/Approval by user
- Service Provisioning for a BoD
- Disconnection of a provisioned BoD on its end time

2. Demo Test Case #2: Network Monitoring & Troubleshooting (Tata Comm. Operation Experience)

- Enhanced Real-time Monitoring
- Enhanced Logging system
- Performance monitoring: infrastructure & ONAP



Q&A

A background image of a golden wheat field under a bright, hazy sky. The wheat stalks are in sharp focus in the foreground, creating a sense of depth and texture. The overall color palette is warm, dominated by yellows and oranges.

OLF

NETWORKING

LFN Developer & Testing Forum