U-6 (SDN)

* Juniper SDN Framework -

- uses high level data model that self generates & offers a REST API to SDN appln.
- data centre orchestration appln → setup virtual routers on hosts in order to connect/relay instances across network overlay.
- API → coordinate entire system.
  - ↳ portion of it overlaps Openstack Quantum API.
- controller → multinode system → several different subsystems.
  goal → promote high availability, scalability & extensibility.
  sys → scale out server modules for analytics configuration & control by supporting theoritically sepdables modules.

* Analytics → supports query interface & storage interface for statistics / counter reporting.

* Configuration → facilities compiler that uses high level data model to exchange API requests for network actions into low-level data model for implementation via control code distribution.

* Control → BGP speaker → horizontal scale b/w controller & implementer of low level data model.

- BGP → defined protocol enabling horizontal scalability & possibility for multivendor interoperability, used by control node to distribute network state.

DOMS | Page No.
Date / /

IETF SDN Framework –

- Internet Engineering Task Force.
- works on specific, tightly focussed problems with an emphasis on protocol specification.
- foremost standards body for Internet.
- SDN → centralized view of topology of network.
- Appln layer traffic optimization (ALTO) can implemented at controller side in this context.

Open Daylight (ODL) controller –

- JVM software.
- used with any hardware & OS that supports JAVA.

Tools utilized by controller.

* Maven → • easy build automation.
  • uses pom.xml (Project Object Model) to script dependencies among bundles & describe what bundles to load & start.
* OSGi → • framework is backend of OpenDaylight.
  permits dynamically loading bundles & packages JAR files & binding jointly for exchanging information.
* JAVA interfaces → • used for event listening, specification & forming patterns.
  • main way in specific bundles implement call back functions for events & also to indicate awareness of specific state.
* REST APIs → northbound APIs → topology manager, host tracker, flow programmer, static routing, etc
  • Northbound APIs → controller exposes
  • Northbound APIs → bidirectional REST & OSGi framework

DOMS | Page No.
Date / /

* Open Daylight Architecture -

- modular open platform for customizing & automating networks of any size & scale.
- arose out of SDN movement, with clear focus on network programmability
- designed from outset as foundation for commercial solutions.

ODL architecture has three layers -

1) Southbound Plugins - communicate with network devices.
2) Core services - used by service Abstraction Layer (SAL) which is based on OSGi to help components going in & out of controller while controller is running.
3) Northbound interfaces - allows operators to relate high-level policies to network devices or integration of ODL with other platforms

1) Modularity & extensibility (join code modules)
2) Scalability → Cluster scalability
   → scalability of architecture
3) Interfaces → Southbound
   → Northbound
4) Telemetry
5) Resilience & fault tolerance.
6) Programming language (JAVA)
7) Community.

DOMS | Page No.
Date / /

Floodlight Controller -

- open & programmable network by SDN.
- an OpenFlow controller.
- implements various capabilities to address various user demands across network.
- a set of common functionalities to control & query an OpenFlow network.
- made up of appln modules that implement sol^n for various problems & controller modules → essential network functions.

* Features →
  - Developers benefit → Java → quickly adjust s/w &
  - REST APIs → simpler to interact with product programatically against net apps.
    - ↳ coding samples to help developer to create product.
  - tested with physical & virtual OpenFlow compatible switches.
  - handle standard networks, non-open flow switch consists a collection of OpenFlow compatible switches.
  - Openstack s/w toolset → cloud computing platform → public / private clouds.
  - network backend for openstack using Neutron plugin exposes a networking as a service model with REST API that floodlight offers.

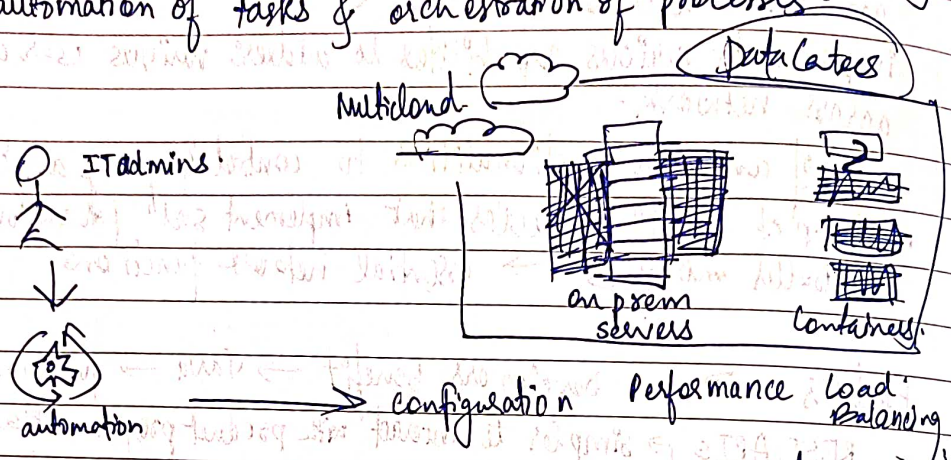
* Flood light Platform -
1) Apache licensed
2) openFlow protocol
3) Java based
4) Enterprise class controller.

Basic functionality
1) Topology discovery.
   - LLDP protocol.
2) Flow installation / deletion -
   - Install / modify / delete a flow on a switch.
   - Flow is defined as packets same match.
3) Stats query -
   - Packet counters.
   - Flow counts
   - Port stats query - Etc.

DOMS | Page No. | Date / /

Data Center Orchestration -

- process driven workflow will help to make data cents more well organised
- Repetitive, slow & error prone manual tasks are replaced by automation of tasks & orchestration of processes.



Adv → streamlined creation → public & private clouds.
→ shorter time lag exists b/w business demand & infrastructure
→ IT department → less time to deliver a domain specific
→ framework controlling data transferred b/w business process.

capabilities → tracking & publishing APIs for automatic updates of metadata
→ updating policy enforcement
→ integrating data services
→ Scheduling & coordination of data services.
→ leveraging of distributed data services.

Data Center Orchestration Tools -

- Chef
- Docker
- Puppet
- Ansible
- Vagrant.