Juniper Event Driven Infrastructure

Junos configuration auto remediation demo

Khelil Sator
ksator@juniper.net
November 2017

OVERVIEW OF THE DEMO

- At each new Junos commit
  - Automatically backup the Junos configuration files on a Gitlab server
  - Automatically run a JSNAPy test to check if the new junos configuration is inline with the company policy
    - If the new Junos configuration is not inline with the company policy, then automatically
      - Create a ticket
      - Fix the junos configuration to make it compliant
      - Update the ticket

Juniper Event Driven Infrastructure (J-EDI)

- Uses regular/native SaltStack building blocks
  - Salt master, minions, event bus, reactor, …
  - Salt proxies for Junos (Juniper contribution to SaltStack)
  - Salt sls and execution modules for Junos (Juniper contribution to SaltStack)
  - Salt junos_syslog engines (Juniper contribution to SaltStack)

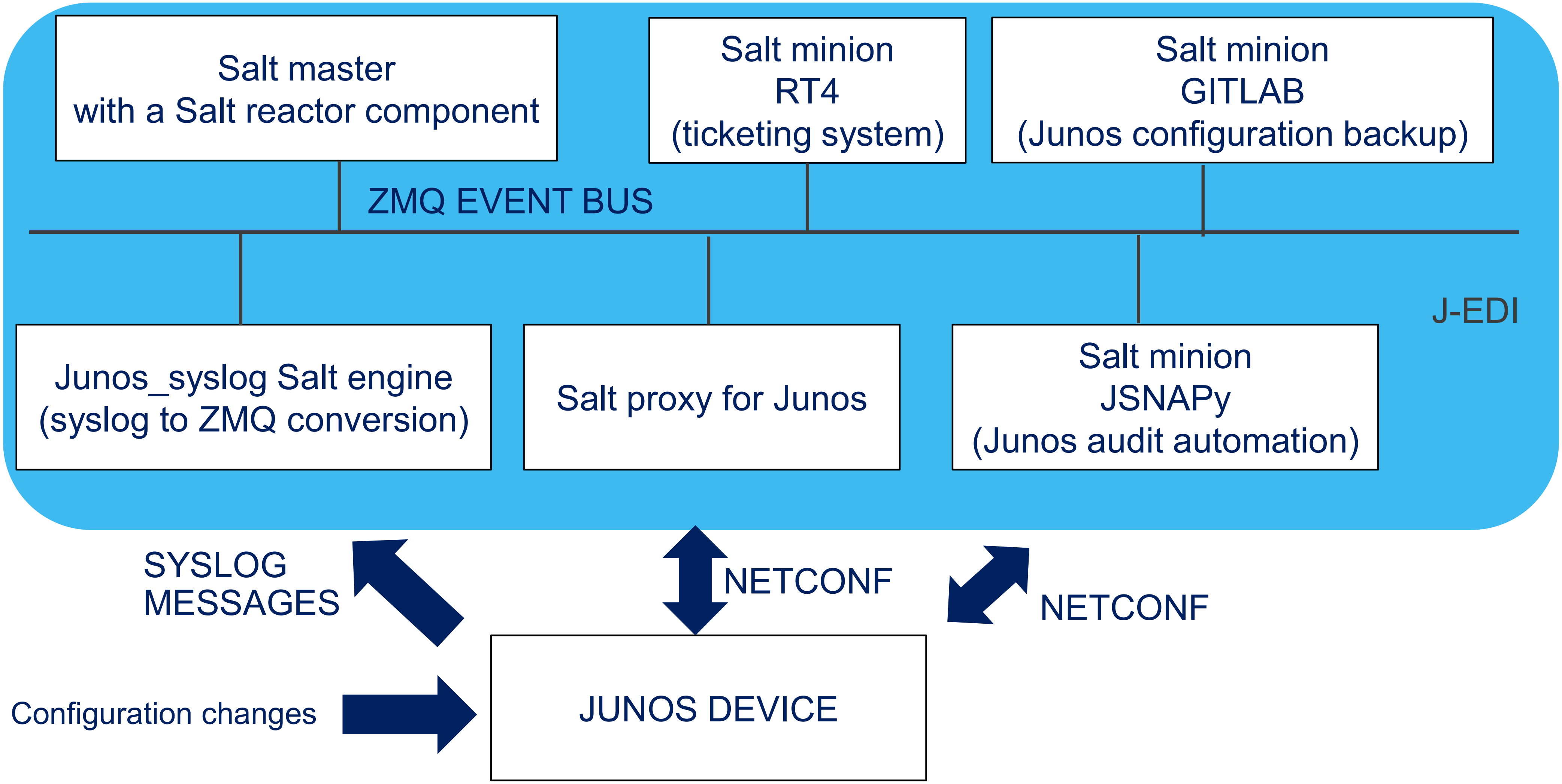
- Uses the event bus as an automation backplane

- Loosely couples a growing collection of open-source and Juniper maintained tools
  - Junos Space, JSNAPy, OpenNTI, Appformix, Juniper Secure Analytics, Request Tracker 4, …

- Is developed by Juniper. Is installed, configured, and supported by Juniper Professional Services.

J-EDI PLUGINS USED FOR THIS DEMO

- Junos Backup plugin
  - To automate JUNOS configuration backup on a GITLAB server at each JUNOS commit.
- JSNAPy plugin
  - JSNAPy is a tool to automate verifications on Junos devices (operational state verifications and configuration verifications).
  - To automate JUNOS configuration compliance check, J-EDI uses JSNAPy python library to run a test at each JUNOS commit, and send a 0MQ message to the SaltStack event Bus with the JSNAPy test result details.
  - Auto remediation is done for compliance check failures using a SaltStack proxy for Junos.
- Request Tracker 4 plugin
  - RT4 is a popular ticketing system. It has an API that can be used for the CRUD operations against tickets.
  - J-EDI interacts with RT4 using its API to automate tickets manipulation (tickets creation and update)



Salt master
with a Salt reactor component

Salt minion
RT4
(ticketing system)

Salt minion
GITLAB
(Junos configuration backup)

ZMQ EVENT BUS

J-EDI

Junos_syslog Salt engine
(syslog to ZMQ conversion)

Salt proxy for Junos

Salt minion
JSNAPy
(Junos audit automation)

SYSLOG
MESSAGES

NETCONF

NETCONF

Configuration changes

JUNOS DEVICE

SALT PROXY for JUNOS

- For help about Junos automation with SaltStack, you can visit this repository https://github.com/ksator/junos-automation-with-saltstack

- SaltStack supports Junos automation with a Salt proxy
  - Proxy controls junos devices without installing salt on device.
  - It uses Junos API: junos-eznc python library (pyez) and NETCONF on the device.

- It provides execution modules for Junos so you can run commands on various machines in parallel with a flexible targeting system
  - https://docs.saltstack.com/en/latest/ref/modules/all/salt.modules.junos.html

- It provides state modules for Junos so you can apply sls files
  - https://docs.saltstack.com/en/latest/ref/states/all/salt.states.junos.html

- Junos facts are stored in salt grains

SALT PROXY for JUNOS

```
root@JEDI-cluster-demo:~# salt 'vsrx' test.ping
vsrx:
    True

root@JEDI-cluster-demo:~# salt 'vsrx' junos.cli 'show version'
vsrx:
    ----------
    message:

        Hostname: vsrx
        Model: vsrx
        Junos: 15.1X49-D100.6
        JUNOS Software Release [15.1X49-D100.6]
    out:
        True
```

JUNOS SYSLOG SALT ENGINE

- Listens to syslog events
- Extracts events information
- Sends information on the master/minion event bus.
- Control the type of events to be sent.
- Salt reactors has the ability to take actions according to these events (event driven automation).

- Junos_syslog engine configuration

```
root@JEDI-cluster-demo:~# more /etc/salt/master
...
engines:
  - junos_syslog:
      port: 516
...
```

JUNOS SYSLOG CONFIGURATION

- For junos_syslog engine to receive events, configure the junos devices this syslog host:
  - The ip address is the one of the server running the syslog engine
  - The port is the port where the engine is listening for events.

```
root@JEDI-cluster-demo:~# salt -G 'junos_facts:model:VMX' state.apply junos.syslog
```

```
lab@dc-vmx-2> show system commit
0   2017-11-27 12:17:33 UTC by SaltStack via netconf
    configured with SaltStack using the model syslog

lab@dc-vmx-2> show configuration system syslog host 172.30.52.150
any any;
match UI_COMMIT_COMPLETED;
port 516;
```

JSNAPy TEST FILE

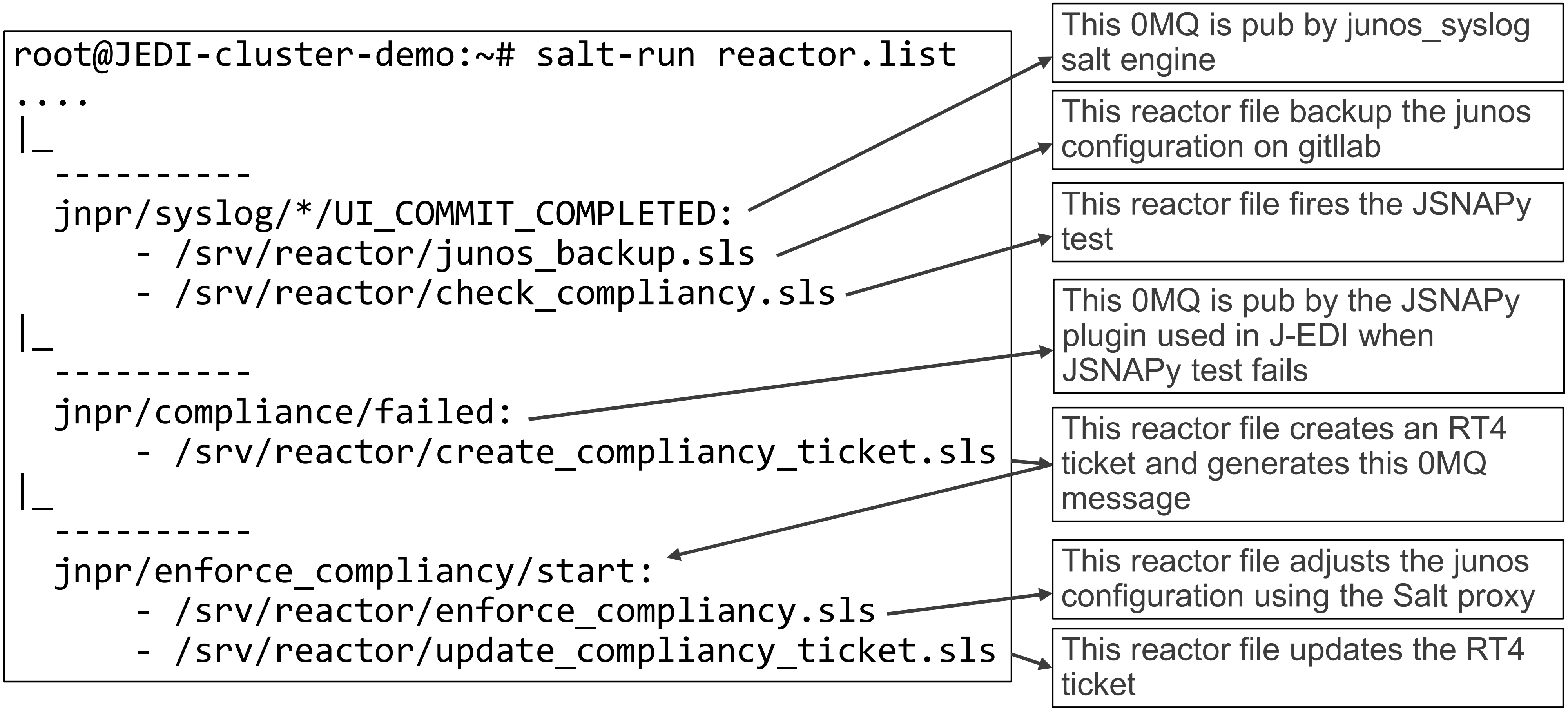
- For help about Junos verification with JSNAPy, you can visit this repository https://github.com/ksator/junos-verifications-automation-with-jsnapy

- So telnet configuration is not allowed

```
root@JEDI-cluster-demo:~# more /etc/jsnapy/testfiles/test_telnet.yml
tests_include:
  - test_telnet_config

test_telnet_config:
  - rpc: get-config
  - kwargs:
      filter_xml: configuration/system
  - item:
      xpath: system/services
      tests:
        - not-exists: telnet
          err: "Test Failed!! telnet is configured"
          info: "Test succeeded!! telnet is not configured"
```

SALTSTACK REACTOR CONFIGURATION



```
root@JEDI-cluster-demo:~# salt-run reactor.list
....
|_
  ----------
  jnpr/syslog/*/UI_COMMIT_COMPLETED:
      - /srv/reactor/junos_backup.sls
      - /srv/reactor/check_compliancy.sls
|_
  ----------
  jnpr/compliance/failed:
      - /srv/reactor/create_compliancy_ticket.sls
|_
  ----------
  jnpr/enforce_compliancy/start:
      - /srv/reactor/enforce_compliancy.sls
      - /srv/reactor/update_compliancy_ticket.sls
```

This 0MQ is pub by junos_syslog salt engine

This reactor file backup the junos configuration on gitllab

This reactor file fires the JSNAPy test

This 0MQ is pub by the JSNAPy plugin used in J-EDI when JSNAPy test fails

This reactor file creates an RT4 ticket and generates this 0MQ message

This reactor file adjusts the junos configuration using the Salt proxy

This reactor file updates the RT4 ticket

DEMO

EVENT DRIVEN AUTOMATION

- A human or a process commits a configuration change on a junos device
  - The junos device sends a UI_COMMIT_COMPLETED syslog message to SaltStack
    - The SaltStack junos_syslog engine publishes a 0MQ message
- The reactor component of the master is subscribing to this 0MQ topic
  - So it executes sls files
    - To backup the new configuration file in a git repository
    - To run a configuration compliance test using JSNAPy.
      - This configuration change is not permitted, so the compliance test fails and a 0MQ message is published
- The reactor component of the master is subscribing to this 0MQ topic
  - So it executes an sls file to create an auto remediation ticket and to publishes a 0MQ message
- The reactor component of the master is subscribing to this 0MQ topic
  - So it executes sls files
    - To enforce the configuration compliancy (auto-remediation) on the device using a SaltStack proxy for Junos (so a new JUNOS commit happens)
    - To update the ticket to track the auto-remediation action

COMMIT A NOT PERMITTED CHANGE ON JUNOS

- Commit a non permitted configuration change on a junos device:

```
[edit]
ksator@vsrx# set system services telnet

[edit]
ksator@vsrx# commit and-quit
commit complete
Exiting configuration mode

ksator@vsrx> show system commit
0   2017-11-06 00:11:27 UTC by ksator via cli
```

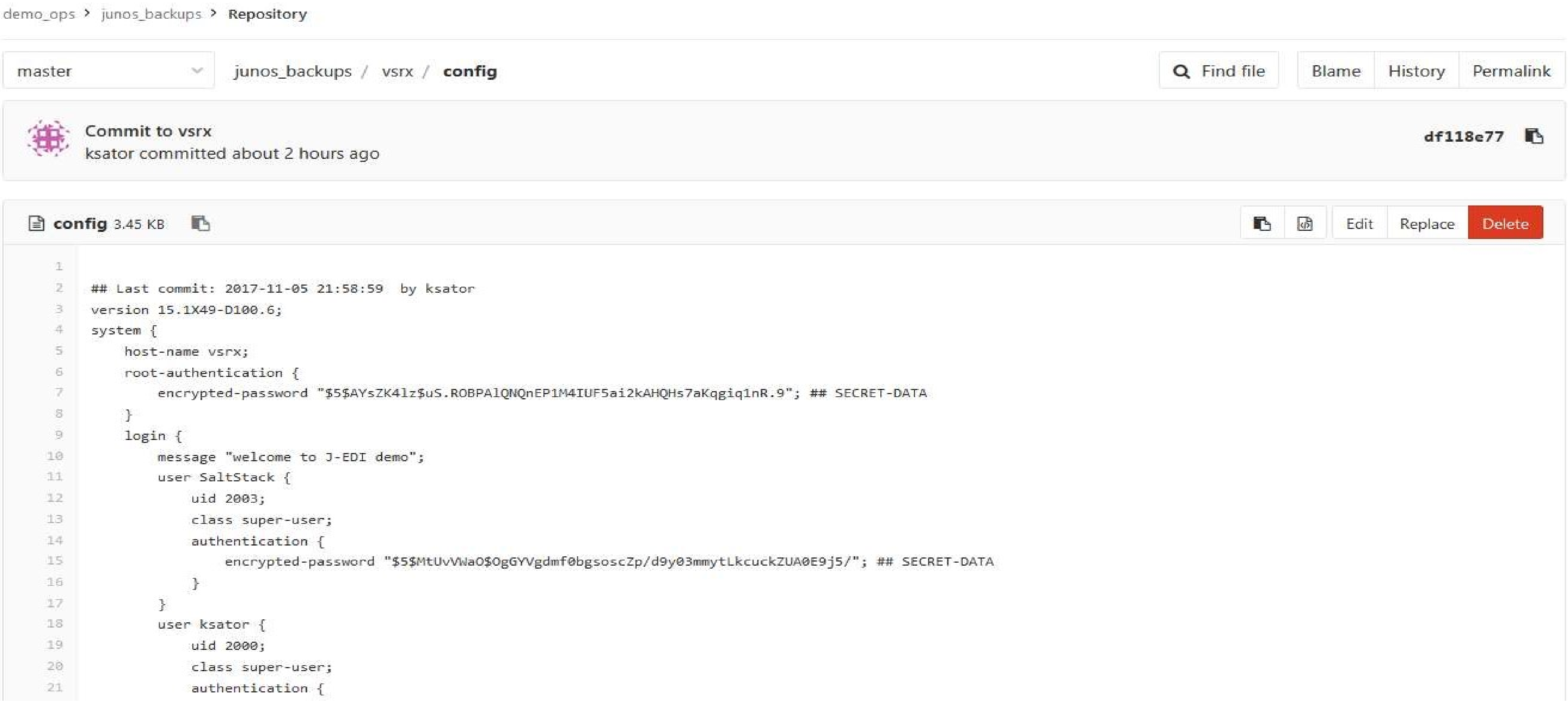
TCPDUMP OUTPUT ON JUNOS_SYSLOG ENGINE

```
root@JEDI-cluster-demo:~# tcpdump -i ens33 port 516 -XX
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
14:00:02.817674 IP 192.168.233.165.syslog > cluster.516: SYSLOG
local7.warning, length: 74
        0x0000:  000c 29fd f54a 000c 2973 7ad1 0800 4500  ..)..J..)sz...E.
        0x0010:  0066 8967 0000 4011 9c5c c0a8 e9a5 c0a8  .f.g..@..\......
        0x0020:  e9cc 0202 0204 0052 68a4 3c31 3838 3e4e  .......Rh.<188>N
        0x0030:  6f76 2020 3620 3030 3a31 313a 3238 2076  ov..6.00:11:28.v
        0x0040:  7372 7820 6d67 645b 3833 3731 325d 3a20  srx.mgd[83712]:.
        0x0050:  5549 5f43 4f4d 4d49 545f 434f 4d50 4c45  UI_COMMIT_COMPLE
        0x0060:  5445 443a 2063 6f6d 6d69 7420 636f 6d70  TED:.commit.comp
        0x0070:  6c65 7465                                lete
```

EVENT PUBLISHED BY JUNOS_SYSLOG SALT ENGINE

```
jnpr/syslog/vsrx/UI_COMMIT_COMPLETED    {
    "_stamp": "2017-11-15T13:00:02.849128",
    "daemon": "mgd",
    "event": "UI_COMMIT_COMPLETED",
    "facility": 23,
    "hostip": "192.168.233.165",
    "hostname": "vsrx",
    "message": "commit complete",
    "pid": "83712",
    "priority": 188,
    "raw": "<188>Nov  6 00:11:28 vsrx mgd[83712]: UI_COMMIT_COMPLETED: commit
complete",
    "severity": 4,
    "timestamp": "2017-11-15 14:00:02"
}
```

NEW CONFIGURATION IS PLACED INTO GITLAB



demo_ops › junos_backups › Repository
master
junos_backups / vsrx / config
Find file
Blame
History
Permalink

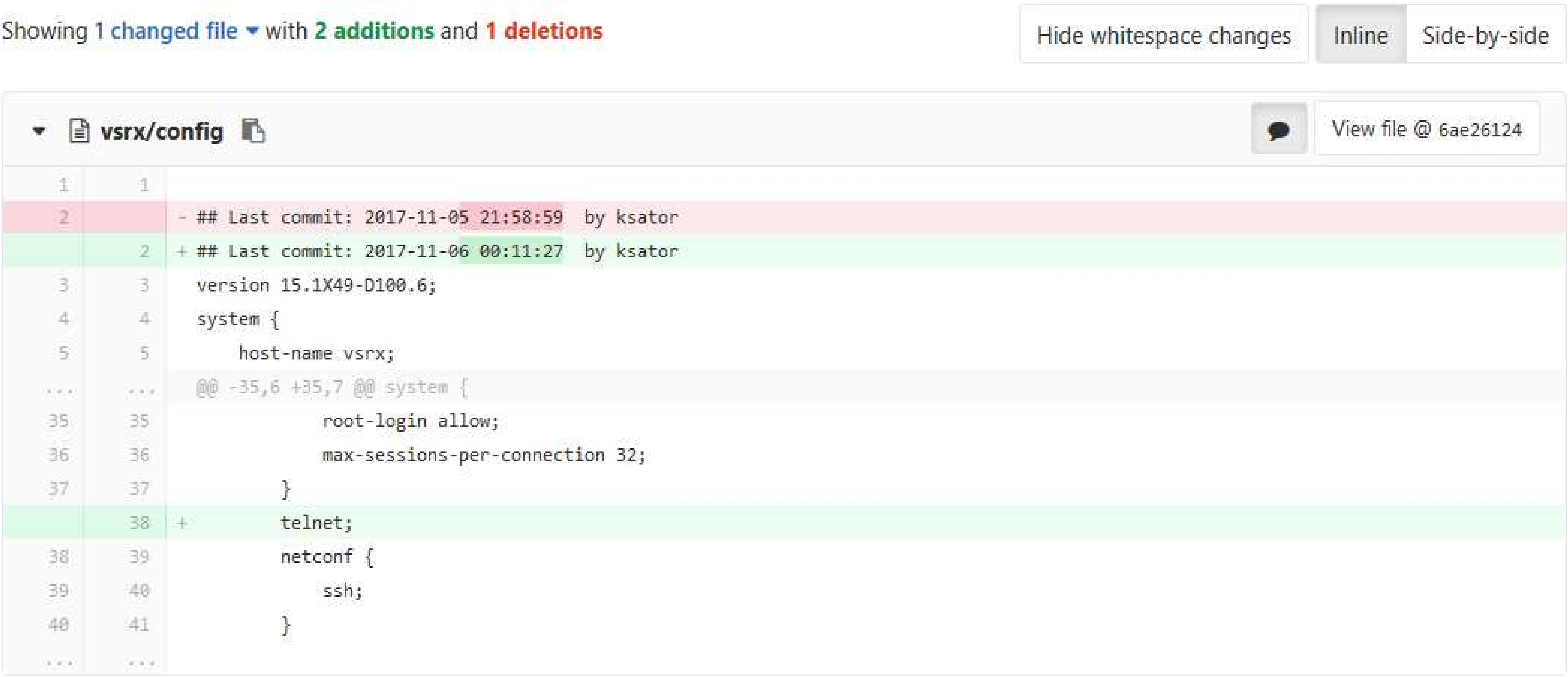
Commit to vsrx
ksator committed about 2 hours ago

df118e77

```

## Last commit: 2017-11-05 21:58:59  by ksator
version 15.1X49-D100.6;
system {
    host-name vsrx;
    root-authentication {
        encrypted-password "$5$AYsZK4lz$uS.ROBPAlQNQnEP1M4IUF5ai2kAHQHs7aKqgiq1nR.9"; ## SECRET-DATA
    }
    login {
        message "welcome to J-EDI demo";
        user SaltStack {
            uid 2003;
            class super-user;
            authentication {
                encrypted-password "$5$MtUvVWaO$OgGYVgdmf0bgsoscZp/d9y03mmytLkcuckZUA0E9j5/"; ## SECRET-DATA
            }
        }
        user ksator {
            uid 2000;
            class super-user;
            authentication {
```

BACKUP HISTORY SHOW THE CHANGE DETAILS



Showing 1 changed file with 2 additions and 1 deletions

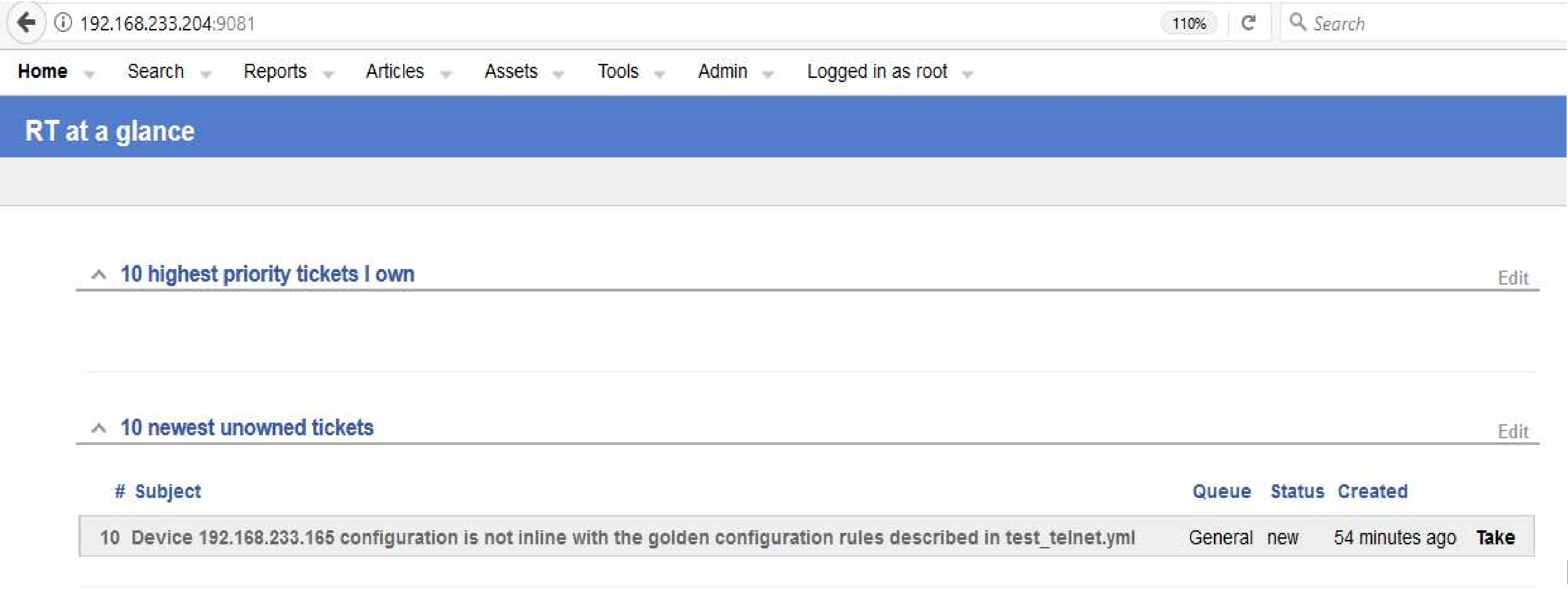
```

- ## Last commit: 2017-11-05 21:58:59  by ksator
+ ## Last commit: 2017-11-06 00:11:27  by ksator
  version 15.1X49-D100.6;
  system {
      host-name vsrx;
  @@ -35,6 +35,7 @@ system {
              root-login allow;
              max-sessions-per-connection 32;
          }
+         telnet;
          netconf {
              ssh;
          }
```

JSNAPy automated test fails (0MQ)

```
salt/run/20171127232607303165/ret       {
    "fun": "runner.jsnapy.check_compliance",
    "fun_args": [
        {
            "ip": "172.30.52.86",
            "test_file": "test_telnet.yml"
        }
    ],
    "return": [
        {
            "details": {
                "message": "Test Failed!! telnet is configured",
            },
            "device": "172.30.52.86",
            "failed": 1,
            "passed": 0,
            "result": "Failed"
        }
    ],
…
```

J-EDI CREATES A NEW RT4 TICKET



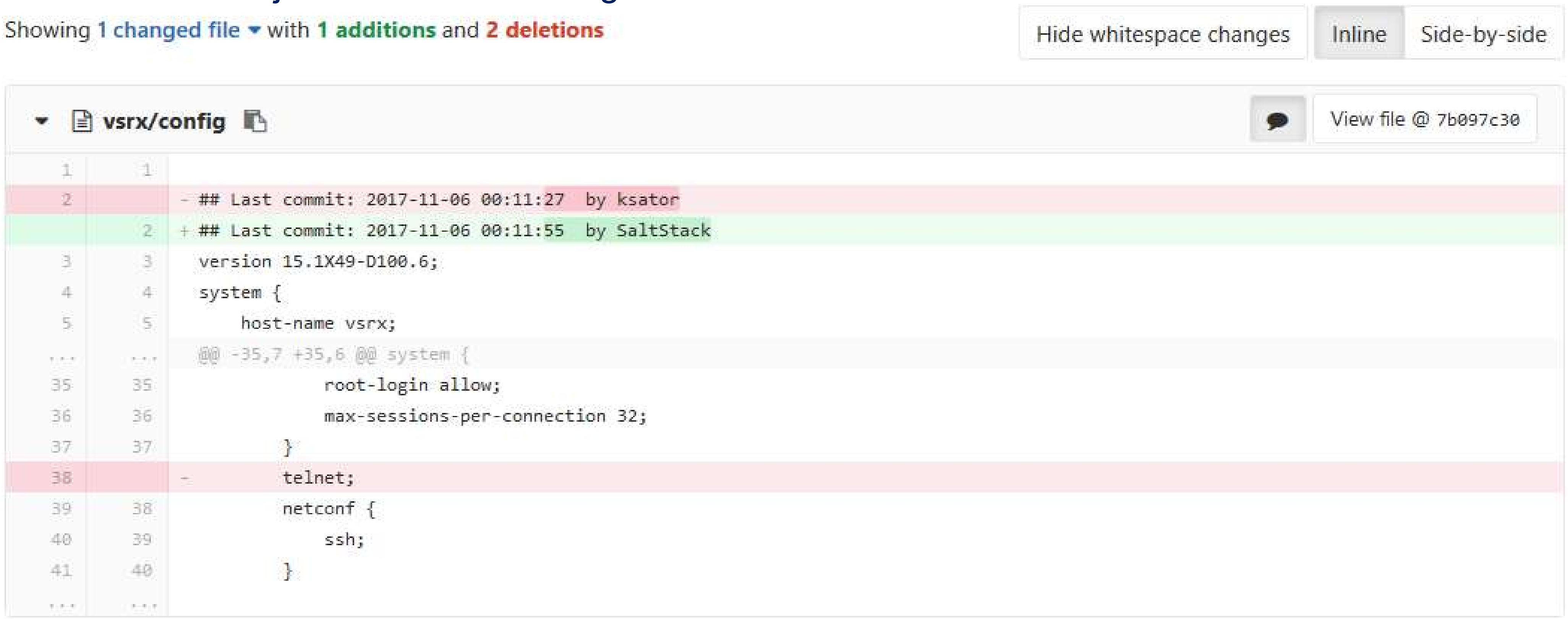
10 highest priority tickets I own

10 newest unowned tickets

#	Subject	Queue	Status	Created	
10	Device 192.168.233.165 configuration is not inline with the golden configuration rules described in test_telnet.yml	General	new	54 minutes ago	Take

JUNOS CONFIGURATION AUTO REMEDIATION

SaltStack adjusts the Junos configuration

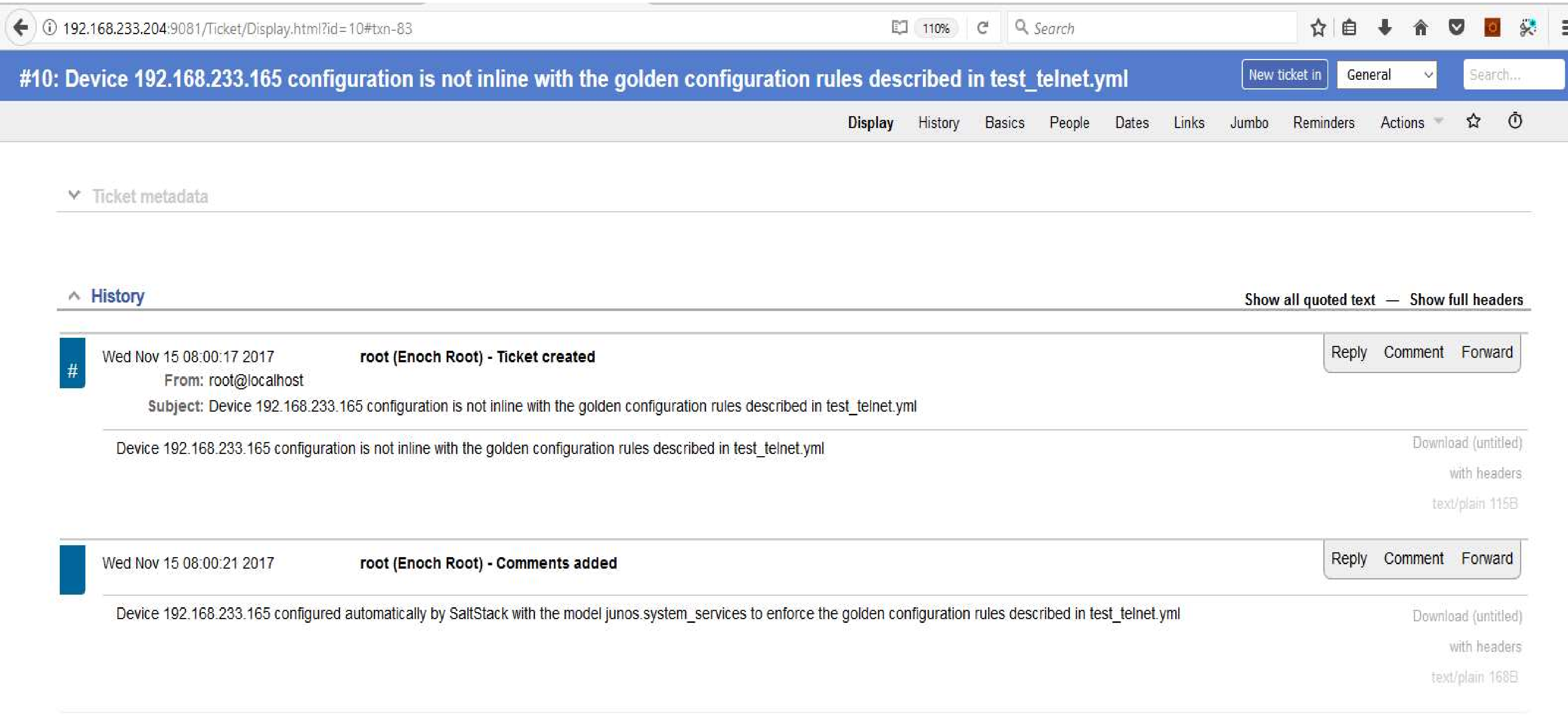


Showing 1 changed file with 1 additions and 2 deletions

vsrx/config

```
- ## Last commit: 2017-11-06 00:11:27  by ksator
+ ## Last commit: 2017-11-06 00:11:55  by SaltStack
  version 15.1X49-D100.6;
  system {
      host-name vsrx;
  @@ -35,7 +35,6 @@ system {
              root-login allow;
              max-sessions-per-connection 32;
          }
-         telnet;
          netconf {
              ssh;
          }
```

TICKET IS UPDATED



#10: Device 192.168.233.165 configuration is not inline with the golden configuration rules described in test_telnet.yml

Wed Nov 15 08:00:17 2017 **root (Enoch Root) - Ticket created**

From: root@localhost

Subject: Device 192.168.233.165 configuration is not inline with the golden configuration rules described in test_telnet.yml

Device 192.168.233.165 configuration is not inline with the golden configuration rules described in test_telnet.yml

Wed Nov 15 08:00:21 2017 **root (Enoch Root) - Comments added**

Device 192.168.233.165 configured automatically by SaltStack with the model junos.system_services to enforce the golden configuration rules described in test_telnet.yml

JUNOS AUTO REMEDIATION DETAILS ON DEVICE

```
ksator@vsrx> show system commit
0   2017-11-06 00:11:55 UTC by SaltStack via netconf
    configured automatically with SaltStack using the model
system_services due to ticket 10
1   2017-11-06 00:11:27 UTC by ksator via cli
2   2017-11-05 21:58:59 UTC by ksator via cli
```

```
ksator@vsrx> show configuration | compare rollback 1
[edit system services]
-    telnet;

ksator@vsrx> show configuration system services | display set
set system services ssh root-login allow
set system services ssh max-sessions-per-connection 32
set system services netconf ssh
set system services web-management http interface fxp0.0

ksator@vsrx>
```

JUNOS AUTOMATION RESOURCES

If you are looking for more details about Junos automation, you can visit these repositories
https://github.com/ksator?tab=repositories
https://gitlab.com/users/ksator/projects
https://gist.github.com/ksator/

Thank you