



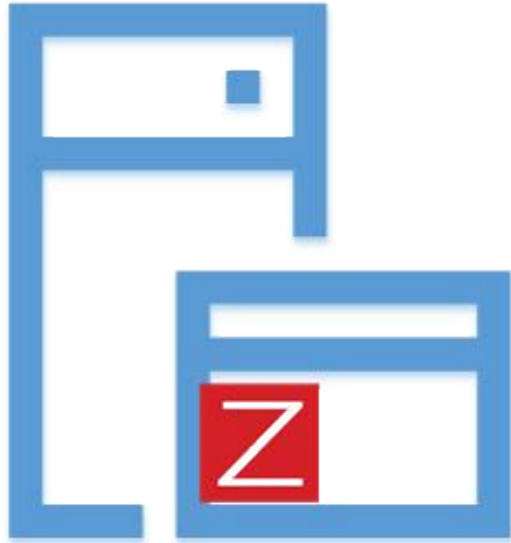
ZABBIX MONITORING STORAGE DEVICES

Dēnnīs Anānīēv
Zabbix enthusiast



Type to enter a caption.

SNMP MONITORING



161.162 UDP

```
ubuntu@NMS-SERVER: ~  
ubuntu@NMS-SERVER:~$ snmpwalk -c public -v 1 172.16.0.5 -m IF-MIB ifName  
IF-MIB::ifName.1 = STRING: lo  
IF-MIB::ifName.2 = STRING: enp0s3  
IF-MIB::ifName.3 = STRING: enp0s8  
IF-MIB::ifName.4 = STRING: enp0s9  
IF-MIB::ifName.5 = STRING: enp0s10  
IF-MIB::ifName.6 = STRING: sw0  
IF-MIB::ifName.7 = STRING: vlan1  
ubuntu@NMS-SERVER:~$ snmpwalk -c public -v 1 172.16.0.5 -m IF-MIB ifType  
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)  
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)  
IF-MIB::ifType.3 = INTEGER: ethernetCsmacd(6)  
IF-MIB::ifType.4 = INTEGER: ethernetCsmacd(6)  
IF-MIB::ifType.5 = INTEGER: ethernetCsmacd(6)  
IF-MIB::ifType.6 = INTEGER: ethernetCsmacd(6)  
IF-MIB::ifType.7 = INTEGER: ethernetCsmacd(6)  
ubuntu@NMS-SERVER:~$ snmpwalk -c public -v 1 172.16.0.5 -m IF-MIB ifSpeed  
IF-MIB::ifSpeed.1 = Gauge32: 10000000  
IF-MIB::ifSpeed.2 = Gauge32: 100000000  
IF-MIB::ifSpeed.3 = Gauge32: 100000000  
IF-MIB::ifSpeed.4 = Gauge32: 100000000  
IF-MIB::ifSpeed.5 = Gauge32: 100000000  
IF-MIB::ifSpeed.6 = Gauge32: 0  
IF-MIB::ifSpeed.7 = Gauge32: 0  
ubuntu@NMS-SERVER:~$
```

REST API



```
if args["item"] == "Discovery":
    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
    systemArray = list()
    fcArray = list()
    trayArray = list()

    systemID = "https://{0}:{1}/devmgr/v2/storage-systems/".format(hostIP, Port)
    req = requests.get(systemID, auth=HTTPBasicAuth(user, passwd), verify=False)
    for item in req.json():
        systemArray.append((item['id'], item['name'], item['status']))

    # Hardware
    for item in systemArray:
        sysID = item[0]
        sysName = item[1]
        hardID = "https://{0}:{1}/devmgr/v2/storage-systems/{2}/hardware-inventory/".format(hostIP, Port, item[0])
        req = requests.get(hardID, auth=HTTPBasicAuth(user, passwd), verify=False)

        tray = req.json()['trays']
        for item in tray:
            if sysName == str(host):
                trayArray.append((sysID, sysName, item['physicalLocation']['slot'], item['trayRef']))

    items = trayArray
    items_count = len(items) # to remove last comma

    # list of data for item status (state) > /tmp/netapp
    filename = str(args["host"]) + "_" + str(args["type"]) + "_" + "Slot"
    write_to_file(filename, items)

    # create json
    zabbix_inventory_json = '{"data":['

    for item in items:
```

CLI



```
if args["item"] == "Data":  
  
    # Vars  
    lunDataStates = ('Normal', 'Read only', 'Inaccessible', 'Virtual space fault', 'Indeterminate data loss',  
                    'Rank failed', 'Rank Repairing', 'Rank Repaired', 'Global inaccessible', 'Global lost data', 'NVS data inaccessible')  
    lunConfStates = ('Normal', 'Configuring', 'Reconfiguring', 'Deconfiguring', 'Configuration error', 'Merging', 'Migrating', 'Migration Cancelled',  
                    'Migration Paused', 'Migration Error', 'Reconfiguration error', 'Deconfiguration error', 'Transposition Error')  
  
    itemArray = list()  
    lssiArray = list()  
    metricsArray = list()  
    command1 = "{0} -hmc1 {1} -user {2} -passwd {3} lssi -s".format(ibmcli_path, hostIP, user, pwd)  
    filename = str(args["host"]) + "_script_luns"  
    lun_metric_script = str(script_dir) + "/" + str(script_name) + " --host " + str(args["host"]) + " --type lun --item LUNmetrics"  
  
    # LUNs  
    # [0] - Name, [1] - ID, [2] - accstate, [3] - datastate, [4] - configstate, [5] - DS  
    #  
    p = subprocess.Popen(command1, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, shell=True)  
    linesarray = p.stdout.readlines()  
    i = 0  
    for line in linesarray:  
        if i > 2:  
            lssiArray.append(line.rstrip())  
            i += 1  
  
    for item in lssiArray:  
        lssi = item  
        command2 = "{0} -hmc1 {1} -user {2} -passwd {3} lsfbvol -dev {4}".format(ibmcli_path, hostIP, user, pwd, item)  
        p = subprocess.Popen(command2, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, shell=True)  
        linesarray = p.stdout.readlines()  
        for line in linesarray:  
            for item in lunDataStates:  
                if item in line:  
                    dst1 = item  
                    for item in lunConfStates:  
                        if item in line:  
                            dst2 = item  
                            break  
                    line1 = line.split()  
                    itemArray.append((line1[0], line1[1], line1[2], dst1, dst2, lssi))  
                    break
```

CLI AND EXPECT



cli
commands



```
petrovy@VPLEXA:~> cat list_some_folders
#!/usr/bin/expect -f

# connect to vplexcli

spawn vplexcli

# Look for login prompt
expect -re "Name:"

# Send login
send "petrovy\r"

# Look for password prompt
expect -re "Password:"

# Send password
send "1q2e3\r"

expect -re "VPlexcli:/> "

send "ll -t /clusters/cluster-2/exports/storage-views/rsb-dsabsdb2\r"

expect -re "VPlexcli:/> "

expect eof
petrovy@VPLEXA:~> █
```

PYTHON IS OUR FUTURE



netapp-ontap 9.6.0

```
pip install netapp-ontap
```



Cisco MDS 9000 Series Programmability Guide, Release 8.x

 Book Contents

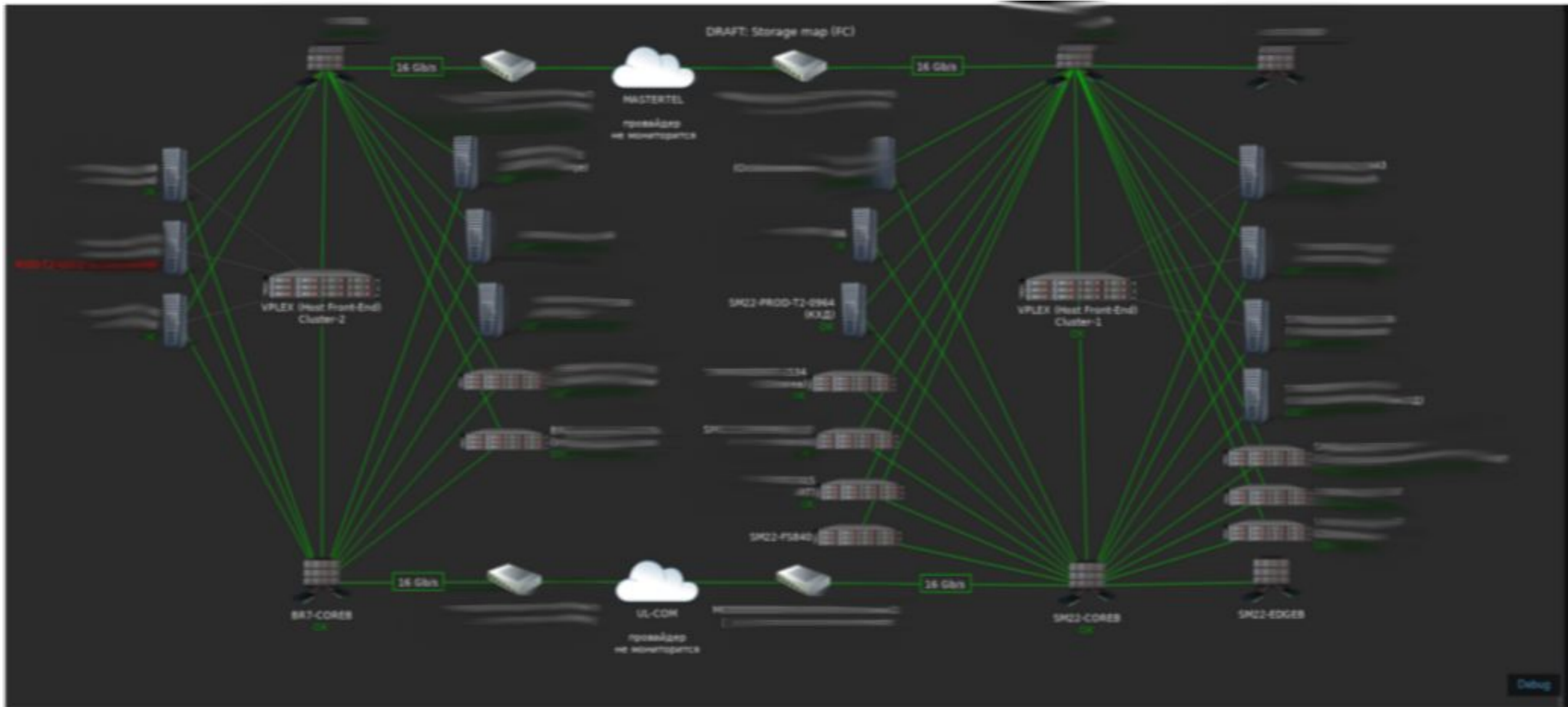
[Book Title Page](#)

[Preface](#)

 Find Matches in This Book

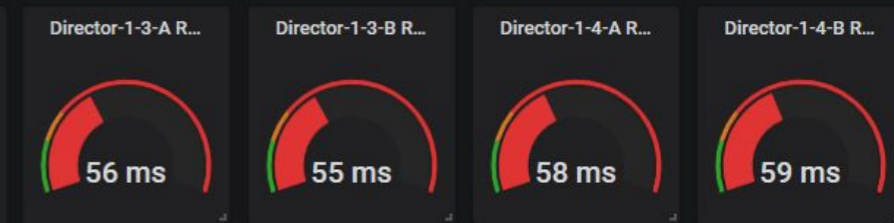
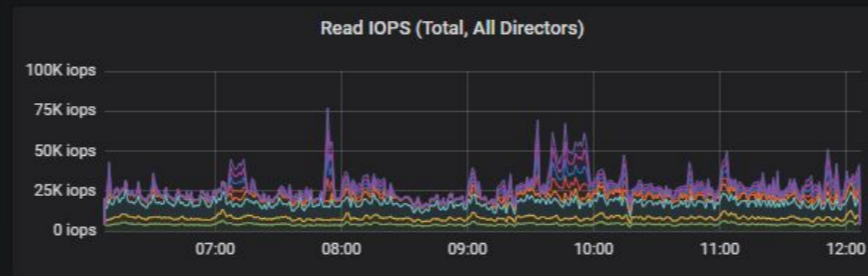
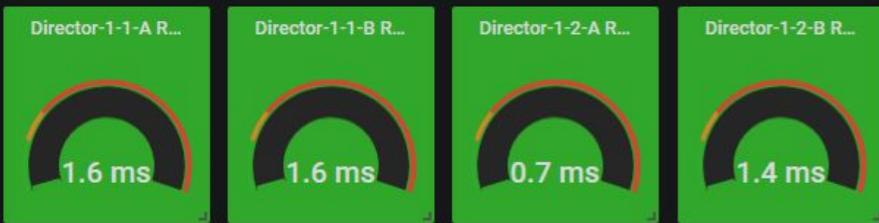
Chapter: Python API

VISUALISATION - ZABBIX

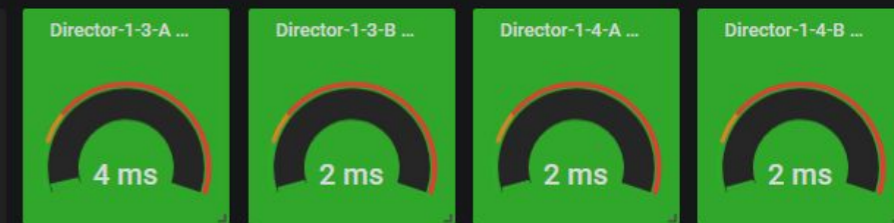
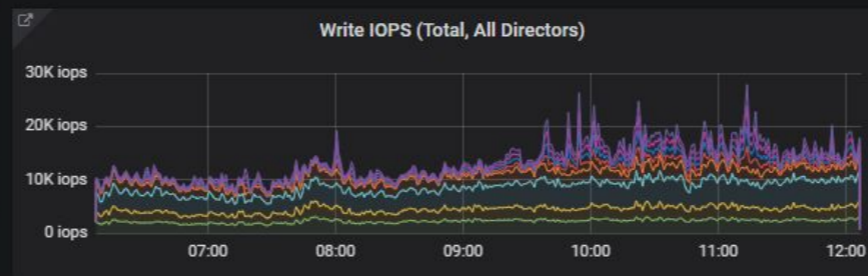
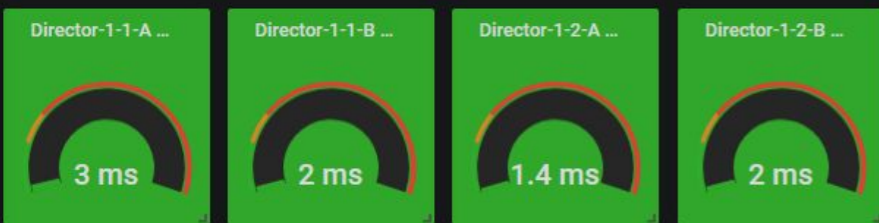


VISUALISATION - GRAFANA

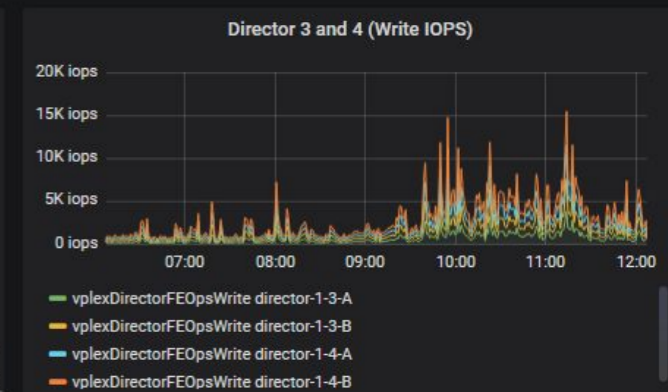
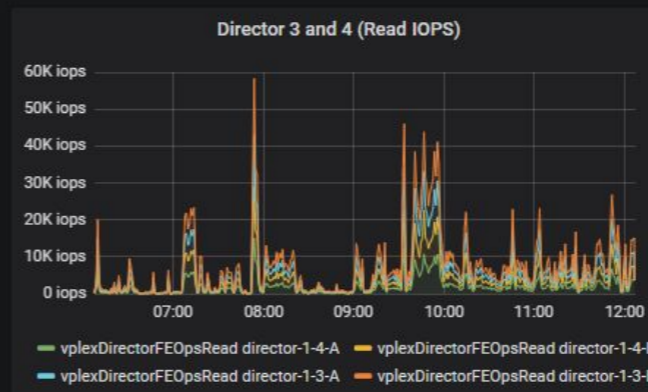
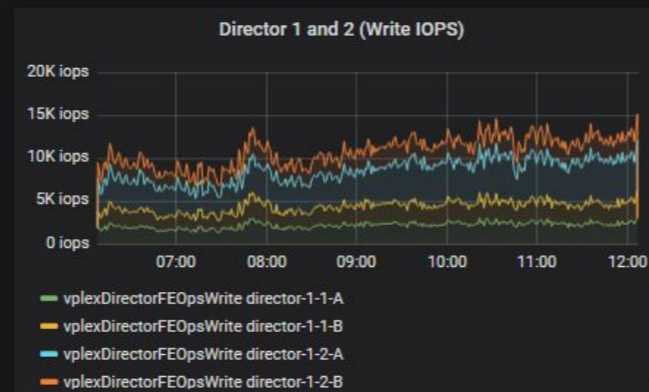
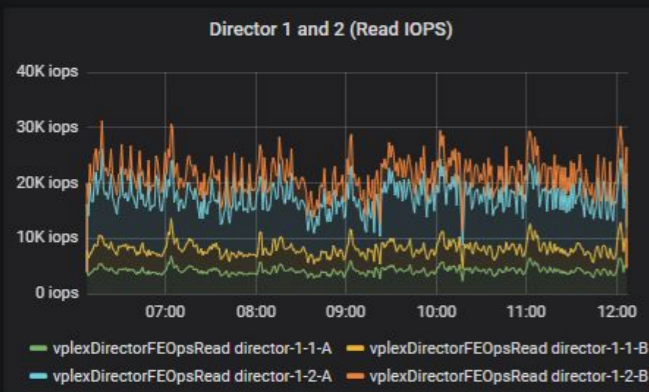
Row title



Dashboard Row



Input output per second per Ddirector



Latency per Ddirector



Q & A



THANK
YOU