

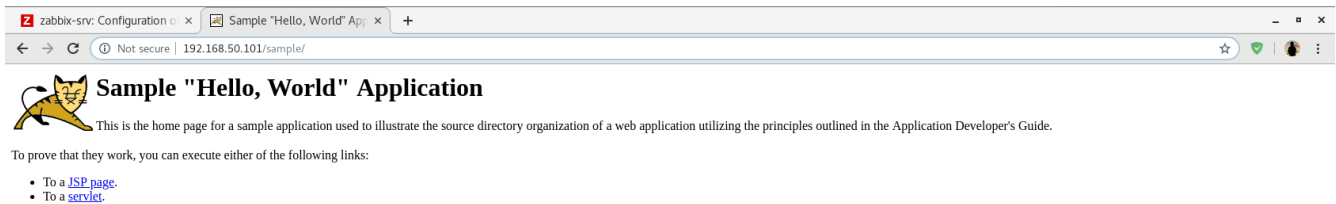
## --- Day 02 ---

### 1: Zabbix WEB

Tasks:

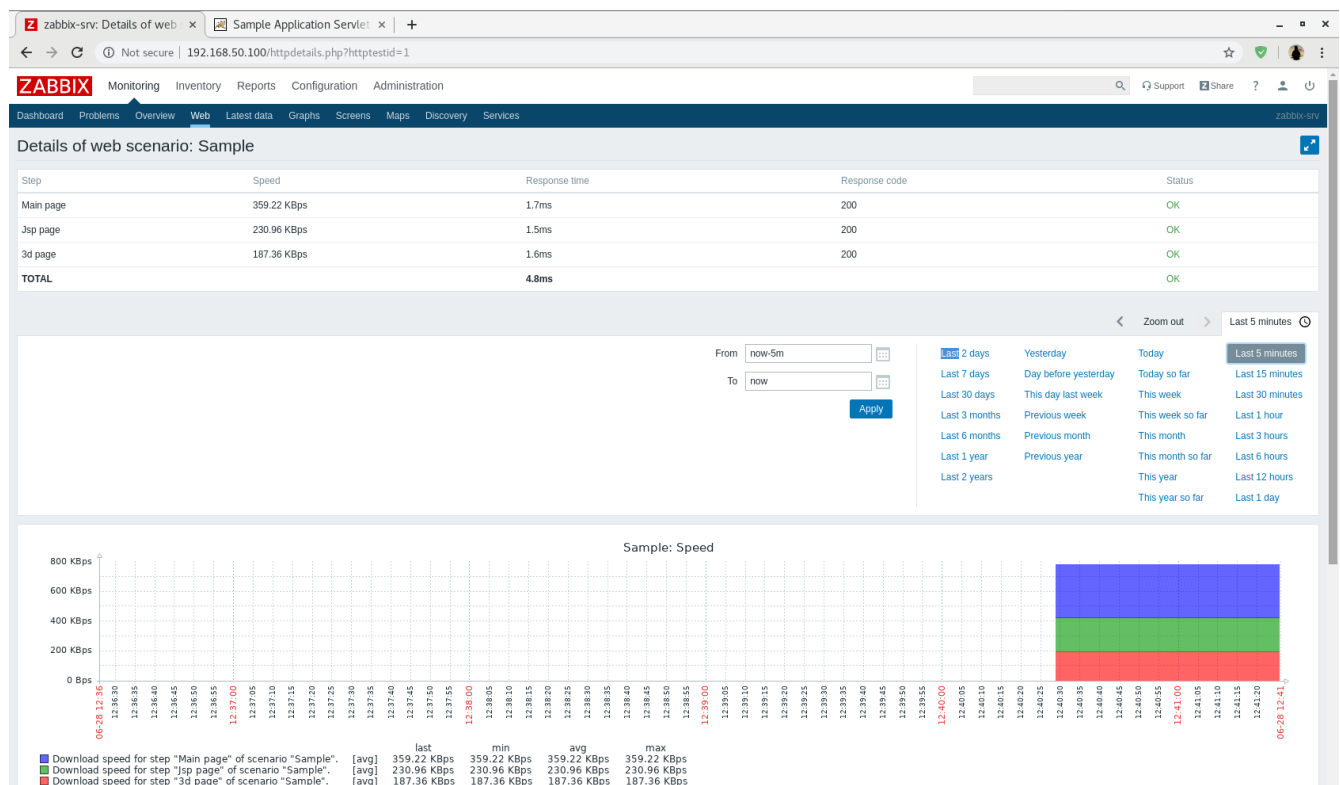
Configure WEB check:

1. Scenario to test Tomcat availability as well as Application health

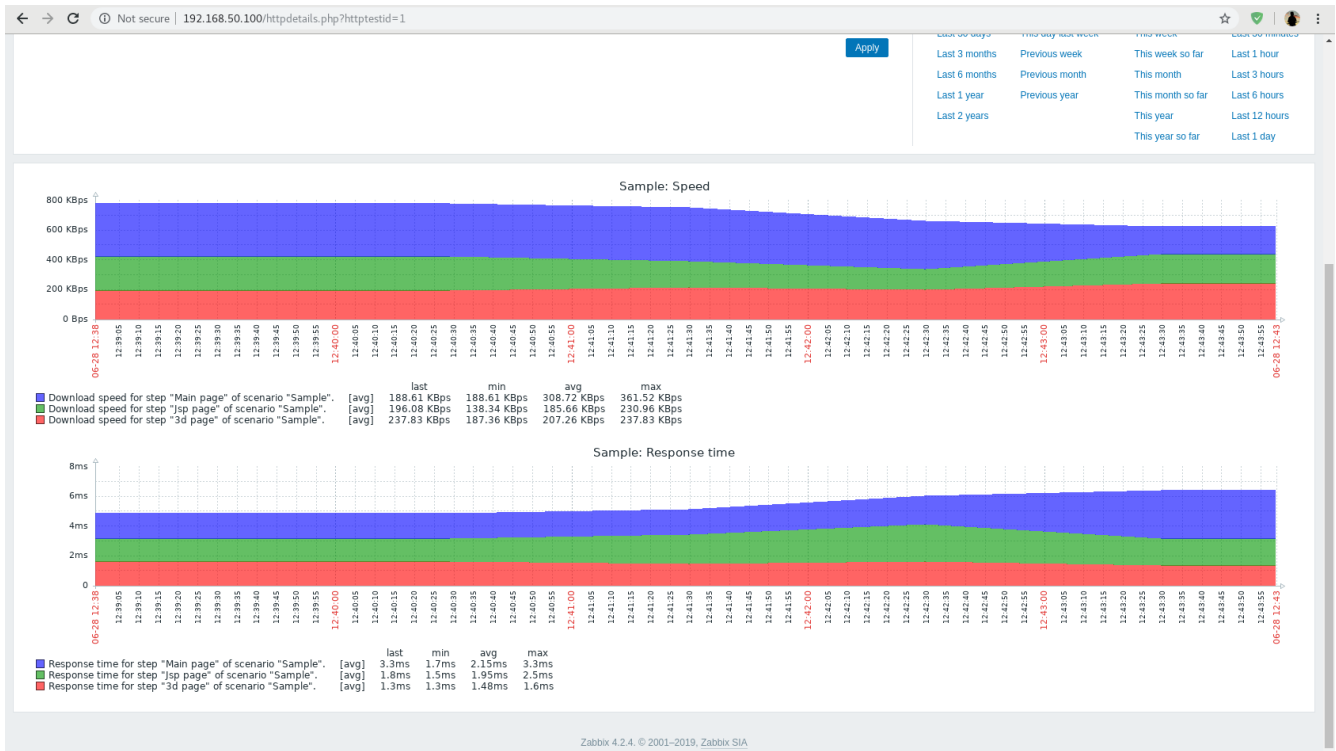


Picture 1

2. Configure Triggers to alert once WEB resources become unavailable



Picture 2.1

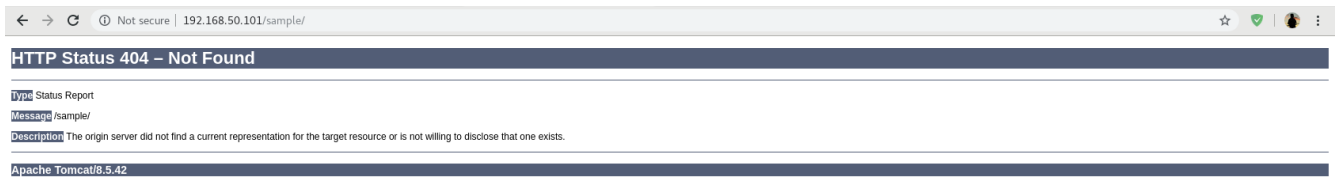


Picture 2.2

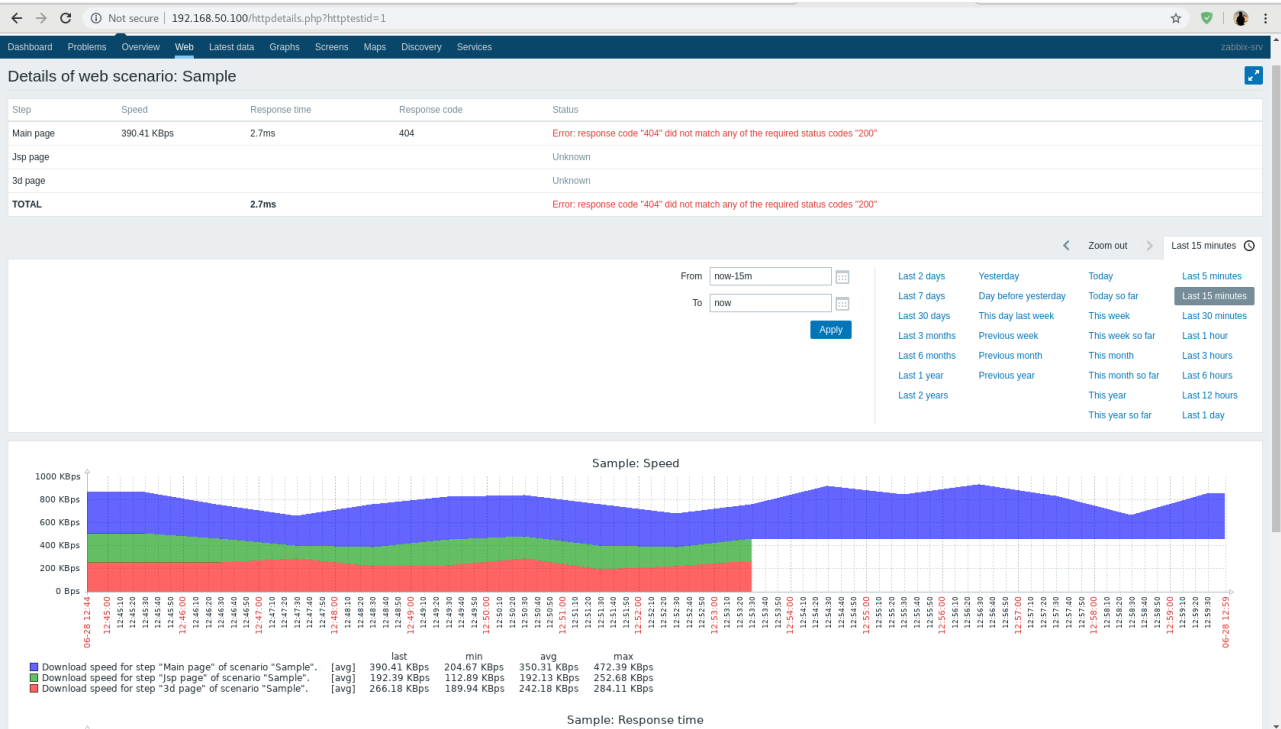
### 3. Stop simple.war

name	name	name	name	name	name
/sample	None specified	Hello, World Application	false	0	Expire sessions with idle ≥ 30 minutes
Deploy					
Start Stop Reload Undeploy					

Picture 3.1



Picture 3.2

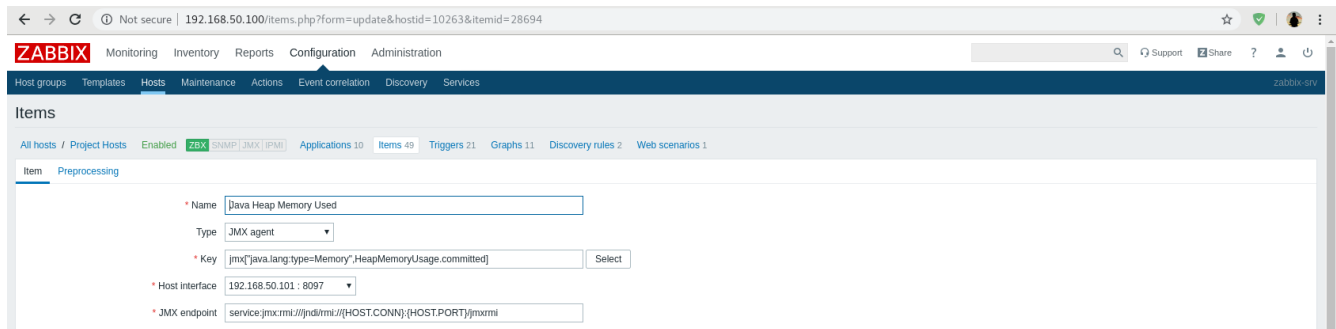


Picture 3.3

## 2: Zabbix JAVA

### Tasks:

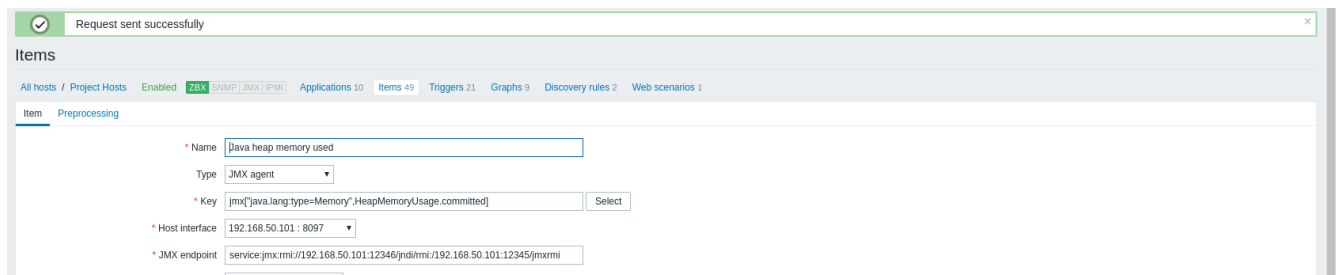
1. Configure Zabbix to examine Java parameters via Java Gateway (<http://jmxmonitor.sourceforge.net/jmx.html>)



The screenshot shows the Zabbix web interface with the 'Items' tab selected. The 'Preprocessing' sub-tab is active. The configuration form for a new item is displayed with the following fields:

- Name: Java Heap Memory Used
- Type: JMX agent
- Key: jmx[\"java.lang.type=Memory\";HeapMemoryUsage.committed]
- Host interface: 192.168.50.101 : 8097
- JMX endpoint: service:jmx:mi://jndi.rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi

Picture 4.1

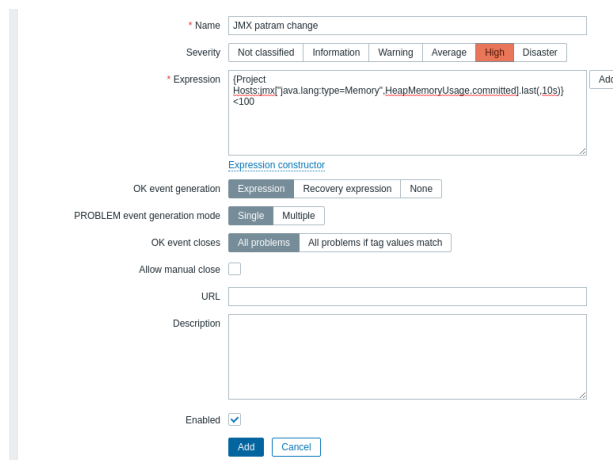


The screenshot shows the Zabbix web interface with the 'Items' tab selected. The 'Preprocessing' sub-tab is active. The configuration form for a new item is displayed with the following fields:

- Name: Java heap memory used
- Type: JMX agent
- Key: jmx[\"java.lang.type=Memory\";HeapMemoryUsage.committed]
- Host interface: 192.168.50.101 : 8097
- JMX endpoint: service:jmx:mi://192.168.50.101:12346/jndi.rmi://192.168.50.101:12345/jmxrmi

Picture 4.2

2. Configure triggers to alert once these parameters changed.



The screenshot shows the Zabbix web interface with the 'Triggers' tab selected. The configuration form for a new trigger is displayed with the following fields:

- Name: JMX param change
- Severity: Not classified, Information, Warning, Average, High, Disaster
- Expression: (Project Hosts jmx[\"java.lang.type=Memory\";HeapMemoryUsage.committed].last(10s)) <100
- OK event generation: Expression, Recovery expression, None
- PROBLEM event generation mode: Single, Multiple
- OK event closes: All problems, All problems if tag values match
- Allow manual close: ☐
- URL:
- Description:
- Enabled: ☒

Picture 5.1

ZABBIX

MonitoringInventoryReportsConfigurationAdministration

Host groupsTemplatesHostsMaintenanceActionsEvent correlationDiscoveryServices

Actions

ActionOperationsRecovery operationsUpdate operations

Name

Report false problem with application

Type of calculation

And/Or

A and B and C

Conditions

Label	Name	Action
A	Trigger equals Project Hosts: JMX patram change	<a href="#">Remove</a>
B	Host equals Project Hosts	<a href="#">Remove</a>
C	Trigger severity equals High	<a href="#">Remove</a>

New condition

Trigger severity

equals

Not classified

Add

Enabled

☒

At least one operation, recovery operation or update operation must exist.

Add

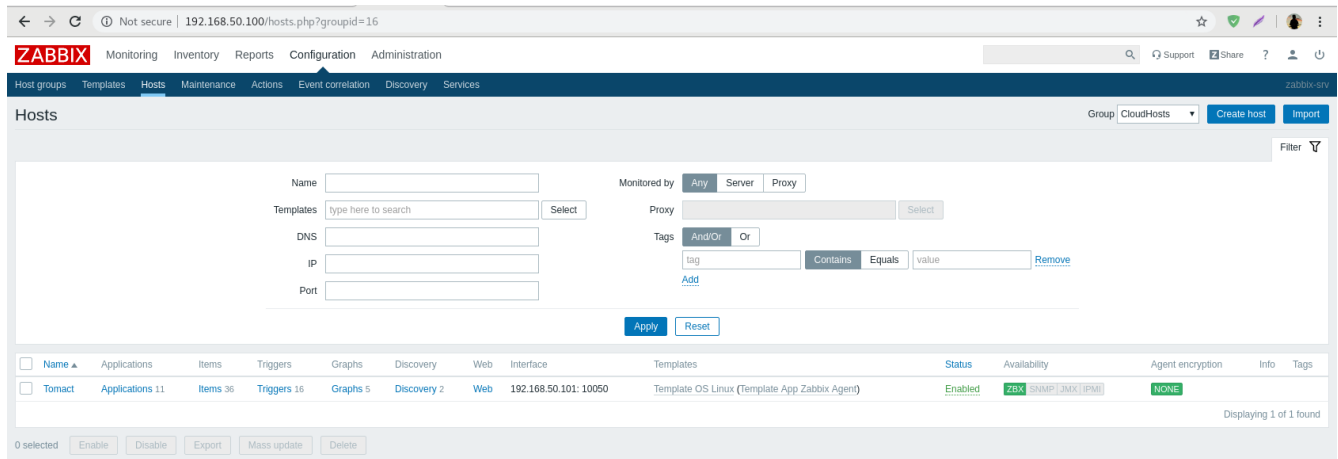
Cancel

Picture 5.2

### 3: Zabbix API

Registering Script requirements:  
Written on Python 2.x

1. Starts at VM startup or on provision phase
2. Host registered in Zabbix server should have Name = Hostname (not IP)
3. Host registered in Zabbix server should belong to "CloudHosts" group
4. Host registered in Zabbix server should be linked with Custom template
5. This script should create group "CloudHosts" if it doesn't exist



Picture 6

Listing main.py:

```
import requests, json

from requests.auth import HTTPBasicAuth

zabbix_server = "192.168.50.100"
zabbix_api_admin_name = "Admin"
zabbix_api_admin_password = "zabbix"

def post(request):
    headers = {'content-type': 'application/json'}
    return requests.post(
        "http://" + zabbix_server + "/api_jsonrpc.php",
        data=json.dumps(request),
        headers=headers,
        auth=HTTPBasicAuth(zabbix_api_admin_name, zabbix_api_admin_password)
    )

auth_token = post({
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "user": zabbix_api_admin_name,
```

```

        "password": zabbix_api_admin_password
    },
    "auth": None,
    "id": 0}
).json()["result"]

def register_group(groupname="CloudHosts"):
    post({
        "jsonrpc": "2.0",
        "method": "hostgroup.create",
        "params": {
            "name": groupname
        },
        "auth": auth_token,
        "id": 1
    })

def register_host(hostname="Tomcat", ip="192.168.50.101"):
    post({
        "jsonrpc": "2.0",
        "method": "host.create",
        "params": {
            "host": hostname,
            "templates": [{
                "templateid": "10001" # Template OS Linux
            }],
            "interfaces": [{
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": ip,
                "dns": "",
                "port": "10050"
            }],
            "groups": [
                {
                    "groupid": "50"
                }
            ]
        },
        "auth": auth_token,
        "id": 1
    })

```