



Application Development for Mobile and Ubiquitous Computing

Seminar Task

Final Presentation

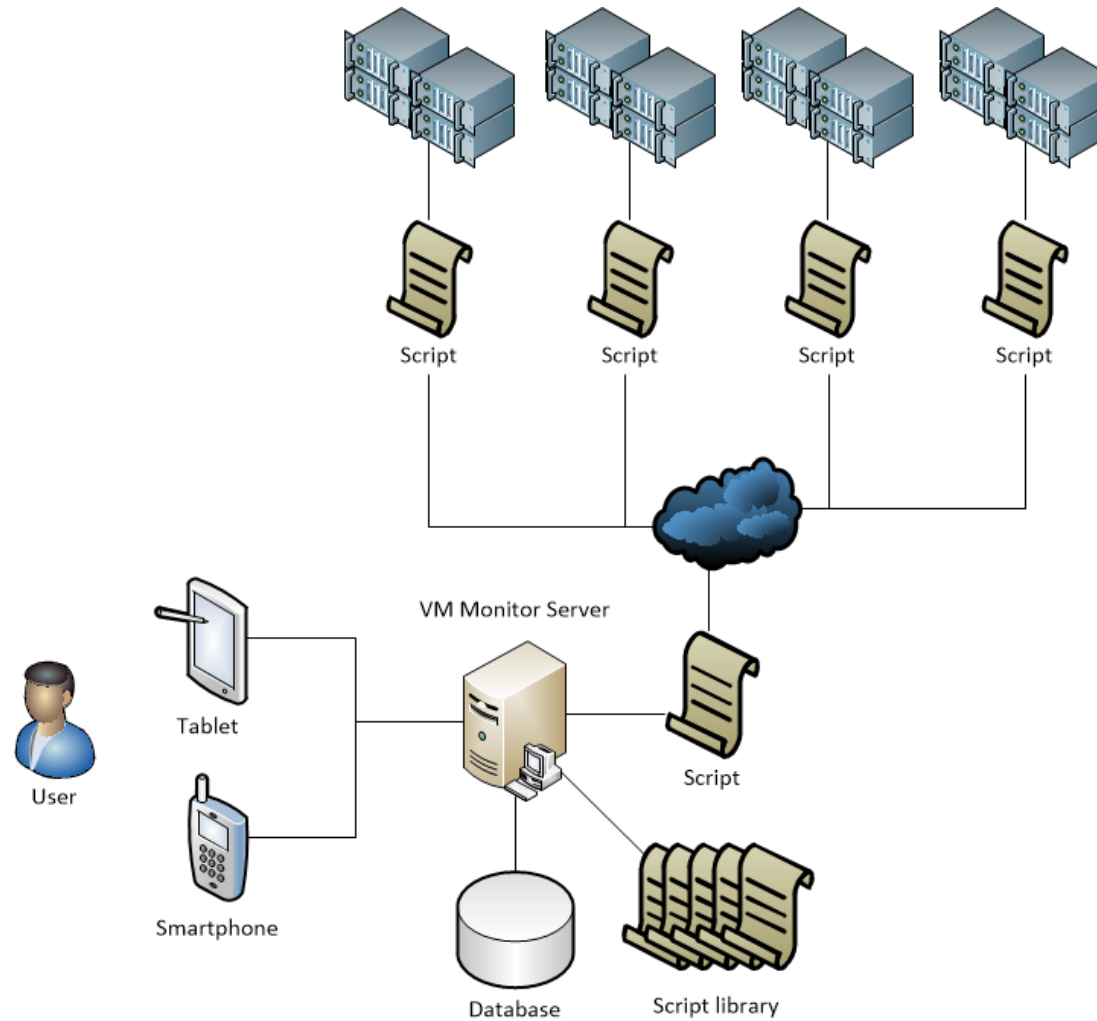
GroupNo. 2 – VM Resource Monitor
Team: Pradeep Kumar,
Rodrigo Lins de Oliveira



VMMonitor

- Make it easy to view the resources consumed by the VMs on mobile device
- Manage the software that are installed on the VMs
- Get alerts on mobile device upon some events like resource usage reached to threshold level

- Client
 - Add the Monitor-Server on the mobile device
 - Get the VMs that are registered to monitor and update the list on mobile device
 - Sync with the monitor for any new scripts or VMs added to monitor
 - Getting notification from monitor upon events
 - Run the script and get result
 - Displaying info about the VM
- Server
 - Add , Store VM and related scripts,
 - Execute script on VM and send result back to mobile device
 - Send alert notification to mobile device.



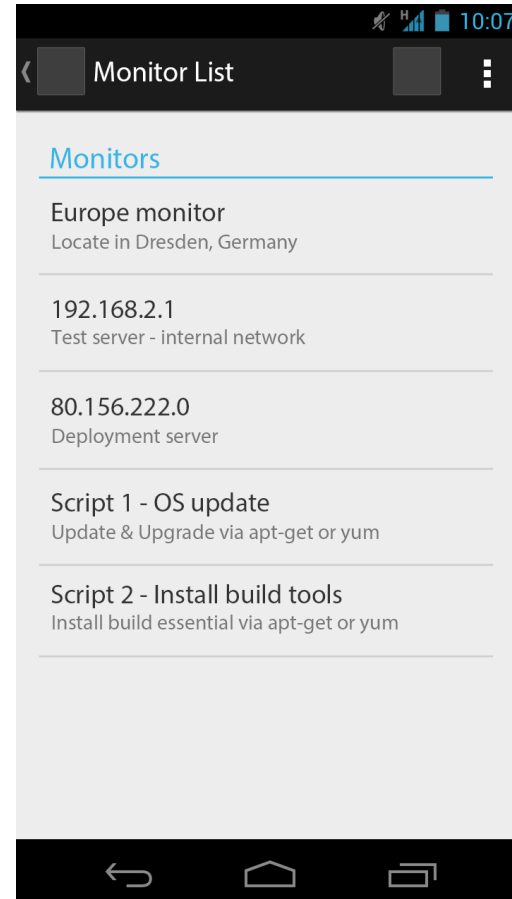
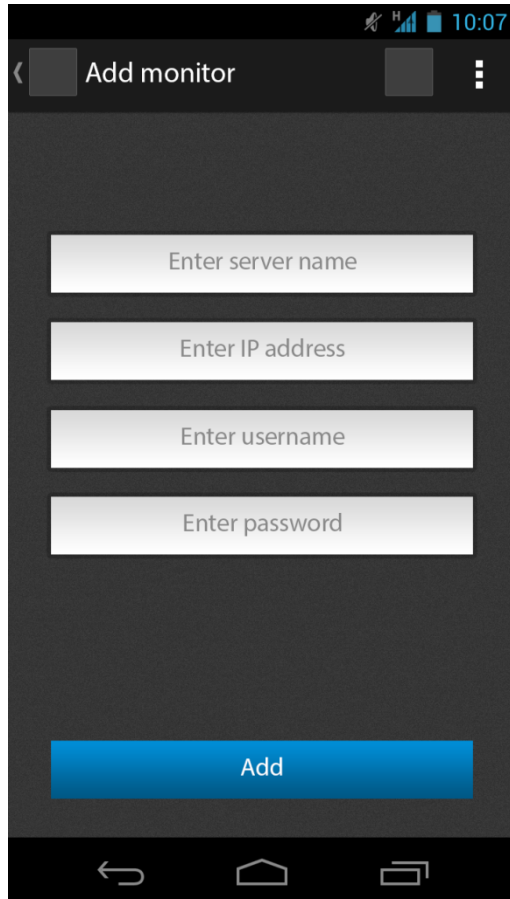


SQLite

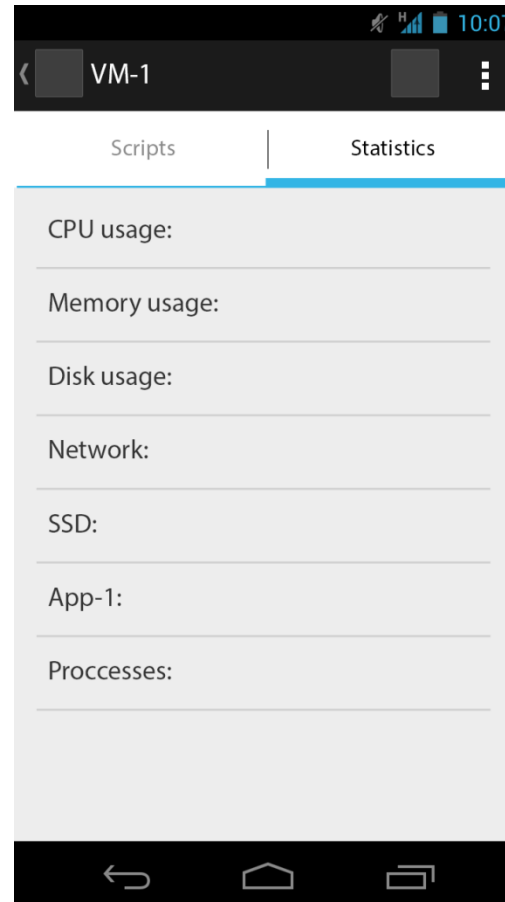
Populate
the list



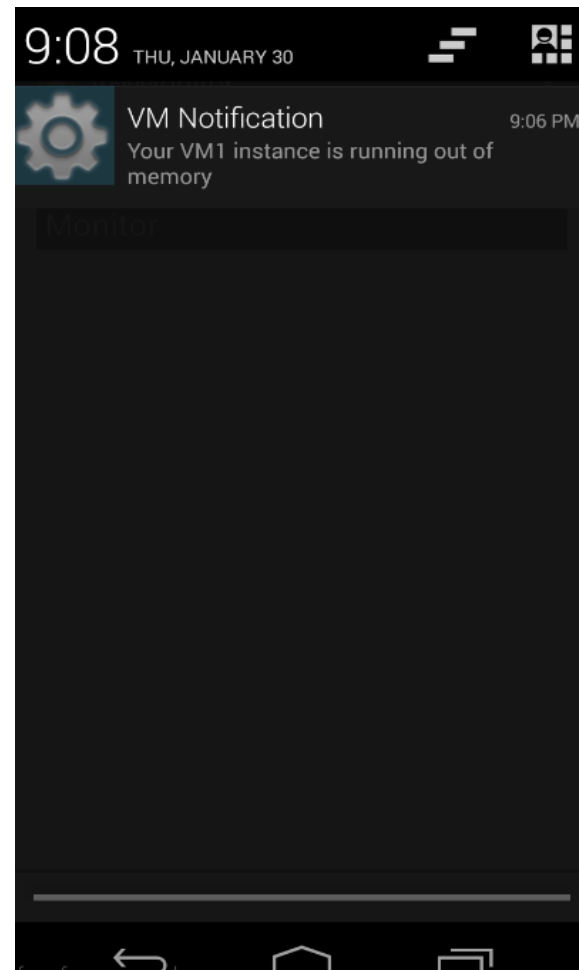
Add Monitor and Monitors List



Taping on the VM
entry will display
resource statistics



Notification from the
server upon an event



127.0.0.1:8001/script-list/

[Create new script](#)

id	name	file	description	servers	Execute	Details	Edit	Delete
2	Test	test.file	Test script.	127.0.0.1, 192.168.2.100,	Run	Details	Edit	Delete
3	Deploy app 1	run.ansible	This script deploys the application 1	192.168.2.100,	Run	Details	Edit	Delete
4	Deploy app 2	run2.ansible	This script deploys application 2.	192.168.2.100,	Run	Details	Edit	Delete
6	List Folder	./dir.cmd	This script list all folder form the target machine.	127.0.0.1,	Run	Details	Edit	Delete
9	Users	./users.sh	This script get all users from the target machine.	127.0.0.1, 192.168.2.100, 192.168.100.113,	Run	Details	Edit	Delete
10	Update	./update.sh	This script update the operating system of the target machine	127.0.0.1,	Run	Details	Edit	Delete

- Keeping the database as small as possible.
- Getting alert notifications through the GCM
- Async task to get the info from monitor through REST APIs
- Django, Tastypie to generate the REST APIs
- Keeping the processing as low as possible while looking for updates.
- Limiting the communication with the server only to run scripts , get VM info and to sync with the monitor.

- Its not a standalone application,it requires internet connection always
- Whenever we add a new VM to monitor we have to sync it to appear on the list
- When we run a script,we dont know how long it takes to run the script and get the result.

Thank You !!!