Documentations on CPU Utilization Shell Script

CPU utilization Script Detailed explanation below

```
#!/bin/bash

# To collect CPU, Memory and Disk usage report from slave (devops minion) and send alert to master (devops play) server
# Format: Hostname, Date&Time, CPU%, MEM%, DiskUsage%
```

- *To collect CPU, Memory and disk usage report from slave and send alert to master server, we've used the standard format as Hostname, date & time, CPU%, MEM%, disk-usage%.*

```
i=$(cat /opt/all_assignments/backup/cpu_metrics_script_original/hostlist)
```

- *Here in the above shell script command, cat is used for concatenate and calling out what is written in the hostlist file, also we've used variable called "i" in that specified the absolute path as /opt/all_assignments/backup/cpu_metrics_script_original/hostlist*

```
SLAVE_HOST_NAME=$(ssh "$i" hostname)
SLAVE_DATE_TIME=$(ssh "$i" date)
SLAVE_CPU_USAGE=$(ssh "$i" top -b -n 1 d1 | grep "Cpu(s)" | awk '{print $2}' |
awk -F. '{print $1}')
SLAVE_MEMORY_USAGE=$(ssh "$i" free |grep Mem |awk '{print $3/$2 *
100.0}')
SLAVE_DISK_USAGE=$(df -P |column -t |awk '{print $5}' | tail -n 1 |sed 's/%//g')
```

- Here, In the above block we’ve called SLAVE_HOST_NAME, SLAVE_DATE_TIME, SLAVE_CPU_USAGE, SLAVE_MEMORY_USAGE, SLAVE_DISK_USAGE as a variable and in that specified Linux commands such as hostname, date, top, free, df –P. We will discuss these commands briefly in later points.
- ssh used here is (Secure shell) for having secured connection between two servers which is generated by ssh-keygen & ssh-copy-id command.
- ssh "$i" top -b -n 1 d1 | grep "Cpu(s)" | awk '{print $2}' | awk -F. '{print $1}' is used for separating Cpu(s) information from the complete command. (Note: Here grep command searches a file for a particular pattern or characters and displays all lines that contain pattern. “awk” allows users to perform various operations on an input file or text and “print” allows us to print vertical column that we specify)
- ssh "$i" free |grep Mem |awk '{print $3/$2 * 100.0}' free command prints a quick summary of memory usage in terminal. While print $3/$2 * 100.0 will give us the result in %
- df -P |column -t |awk '{print $5}' | tail -n 1 |sed 's/%//g' “df” will come into picture when we want to know the amount of space consumed by a particular file system or how much space is available in your system while tail is used to check a file line from bottom. Sed is used for find, replace, insertion or deletion.

```
ABS_PATH=$"/opt/all_assignments/backup/cpu_metrics_script_original/out.txt"
```

- Here ABS_PATH is declared as a variable where out.txt is the file for storing output.

```
echo "SLAVE_HOST_NAME, SLAVE_DATE_TIME, SLAVE_CPU_USAGE (%), SLAVE_MEMORY_USAGE (%),
SLAVE_DISK_USAGE (%)" > "${ABS_PATH}"
```

```
echo "$SLAVE_HOST_NAME, $SLAVE_DATE_TIME, $SLAVE_CPU_USAGE, $SLAVE_MEMORY_USAGE,
$SLAVE_DISK_USAGE" >> "${ABS_PATH}"
```

- Here “echo” is used for display the text passed in an argument. All the output of the commands will be stored in out.txt file and declared in variable ${ABS_PATH}.

```
WARNING=80
ex=/
for ex in $ex;
do
```

- Here we’ve specified limit for CPU consumption to 80 and used for loop while declaring a variable $ex.

```
MASTER_HOST_NAME=$(hostname)
MASTER_DATE_TIME=$(date)
MASTER_CPU_USAGE=$(top -b -n 1 d1 | grep "Cpu(s)" | awk '{print $2}' | awk -F. '{print $1}')
MASTER_MEMORY_USAGE=$(free |grep Mem |awk '{print $3/$2 * 100.0}')
SLAVE_DISK_USAGE=$(df -P |column -t |awk '{print $5}' | tail -n 1 |sed 's/%//g')
```

- Here, In the above block we’ve called SLAVE_HOST_NAME, SLAVE_DATE_TIME, SLAVE_CPU_USAGE, SLAVE_MEMORY_USAGE, SLAVE_DISK_USAGE as a variable and in that specified Linux commands such as hostname, date, top, free, df –P. We will discuss these commands briefly in later points.
- top -b -n 1 d1 | grep "Cpu(s)" | awk '{print $2}' | awk -F. '{print $1}' is used for separating Cpu(s) information from the complete command. (Note: Here grep command searches a file for a particular pattern or characters and displays all lines that contain pattern. “awk” allows users to perform

various operations on an input file or text and “print” allows us to print vertical column that we specify)

- free |grep Mem |awk '{print $3/$2 * 100.0}' free command prints a quick summary of memory usage in terminal. While print $3/$2 * 100.0 will give us the result in %
- df -P |column -t |awk '{print $5}' | tail -n 1 |sed 's/%//g' “df” will come into picture when we want to know the amount of space consumed by a particular file system or how much space is available in your system while tail is used to check a file line from bottom. Sed is used for find, replace, insertion or deletion.

```
if [[ "$SLAVE_CPU_USAGE" -ge "$WARNING" &&
"$SLAVE_MEMORY_USAGE" -ge "$WARNING" ]]; then

ssh himanshuw@192.168.100.107 'echo CPU and MEMORY consumption
might get full in no time' >> "${ABS_PATH}"

else
ssh himanshuw@192.168.100.107 'echo CPU and MEMORY are in
optimize condition' >> "${ABS_PATH}"


fi
done
```

- Here we will use if else conditional statement for iterative conditional statements in your code. It allows you to control the flow of the scripts execution by evaluating given conditions
- If the CPU and Memory are above 80% then output will be “CPU and MEMORY consumption might get full in no time” and if its vice versa then “CPU and MEMORY are in optimize condition”

Detailed Shell Script for CPU Utilization is shown below:

```
#!/bin/bash

# To collect CPU, Memory and Disk usage report from slave (devops minion) and
send alert to master (devops play) server
# Format : Hostname, Date&Time, CPU%, MEM%, DiskUsage%

i=$(cat /opt/all_assignments/backup/cpu_metrics_script_original/hostlist)

SLAVE_HOST_NAME=$(ssh "$i" hostname)
SLAVE_DATE_TIME=$(ssh "$i" date)
SLAVE_CPU_USAGE=$(ssh "$i" top -b -n 1 d1 | grep "Cpu(s)" | awk '{print $2}' |
awk -F. '{print $1}')
SLAVE_MEMORY_USAGE=$(ssh "$i" free |grep Mem |awk '{print $3/$2 *
100.0}')
SLAVE_DISK_USAGE=$(df -P |column -t |awk '{print $5}' | tail -n 1 |sed 's/%//g')

ABS_PATH=$"/opt/all_assignments/backup/cpu_metrics_script_original/out.txt"

echo "SLAVE_HOST_NAME, SLAVE_DATE_TIME, SLAVE_CPU_USAGE (%),
SLAVE_MEMORY_USAGE (%), SLAVE_DISK_USAGE (%)" > "${ABS_PATH}"
echo "$SLAVE_HOST_NAME, $SLAVE_DATE_TIME, $SLAVE_CPU_USAGE,
$SLAVE_MEMORY_USAGE, $SLAVE_DISK_USAGE" >> "${ABS_PATH}"

WARNING=80
ex=/
for ex in $ex;
do

MASTER_HOST_NAME=$(hostname)
MASTER_DATE_TIME=$(date)
MASTER_CPU_USAGE=$(top -b -n 1 d1 | grep "Cpu(s)" | awk '{print $2}' | awk -
F. '{print $1}')
MASTER_MEMORY_USAGE=$(free |grep Mem |awk '{print $3/$2 * 100.0}')
SLAVE_DISK_USAGE=$(df -P |column -t |awk '{print $5}' | tail -n 1 |sed 's/%//g')

if [[ "$SLAVE_CPU_USAGE" -ge "$WARNING" && "$SLAVE_MEMORY_USAGE" -
ge "$WARNING" ]]; then

ssh himanshuw@192.168.100.107 'echo CPU and MEMORY consumption might
get full in no time' >> "${ABS_PATH}"

else
ssh himanshuw@192.168.100.107 'echo CPU and MEMORY are in optimize
condition' >> "${ABS_PATH}"

fi
done
```