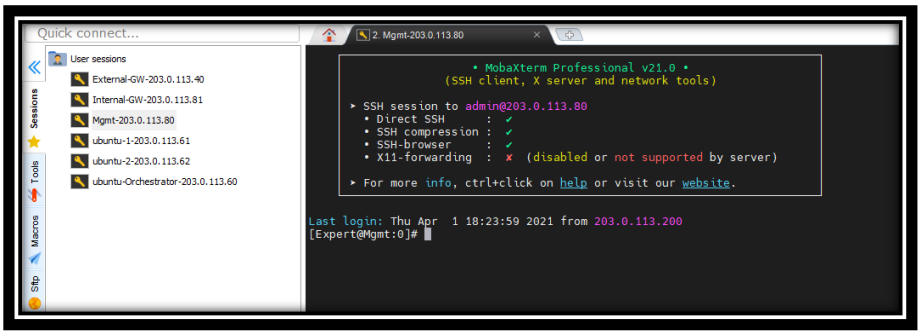
Management CLI Automation

Task - 1: Login to the Environment and API setup

Objective: Connect to the Lab Environment, review the default API settings, configure and start the R81 Management API server.

1. Login to the management server "203.0.113.80". The session is saved in Mobaxterm.



- Username: admin
- Password: vpn123
- IP Address: **203.0.113.80** (NAT address. Real address is **10.1.1.100**)

2. In the SSH session on the management server run the following command "`api status`" .

3. Review the output and check the following settings:
   - API Settings –> Accessibility
   - API Settings –> Automatic Start
   - Overall API Status

☞ Note: By default, the API server is only configured to allow local connections.

```
[Expert@Mgmt:0]# api status

API Settings:
---------------------
Accessibility:                  Require local
Automatic Start:                Enabled

Processes:

Name      State     PID       More Information
-----------------------------------------------------
API       Started   5819
CPM       Started   5819      Check Point Security Management Server is running and ready
FWM       Started   5110
APACHE    Started   4504
```


4. Change directory to the R80_Mgmt_Scripts.

```
[Expert@Mgmt:0]# cd R81_Mgmt_Scripts/
[Expert@Mgmt:0]# pwd
/home/admin/R81 Mgmt Scripts
```

5. Since the R81 API server is setup to only allow local connections by default, we would like to configure it to allow other IP addresses to connect. In addition, we will add an api_user with the password of "vpn123" as an additional user.

   Run the following script:

   ```
   [Expert@Mgmt:0]# bash api_setup
   ```

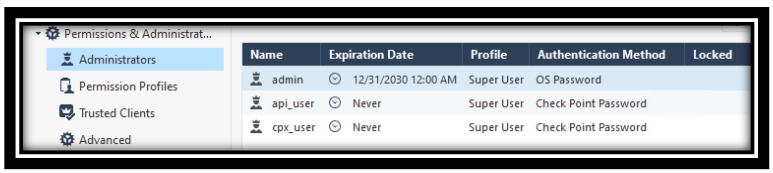
   This script does the following:

   - Adds a new user api_user with a password of vpn123
   - Adds a new user cpx_user with a password of vpn123
   - Adds a new tag "Web_Control"
   - Changes the API server to accept all IP addresses
   - Restarts the api server
   - Runs the api status command

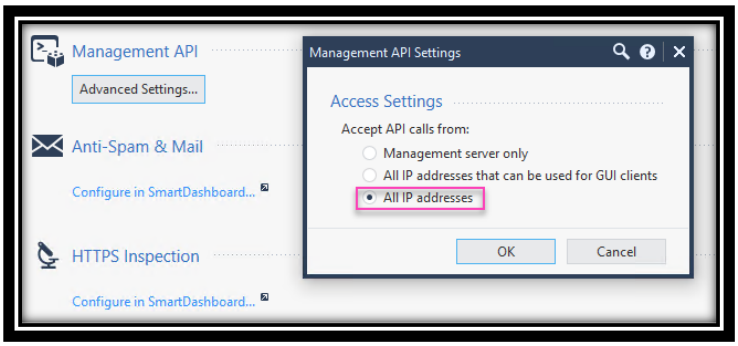
```
[Expert@Mgmt:0]# bash api_setup
Logging into the R81 API Server
SESSIONID KaLFw7V5Vh8b8eDhR6Ur-6y16DCrz4IJKiYW-wLZ9kY
Set R81 API to accept all ip addresses
Add user api_user with password vpn123
Add user cpx_user with password vpn123
Publising User Changes
```

6. Log in to R81 SmartConsole and review the changes that were made:
   - user: admin
   - password: vpn123
   - Management Server IP: 10.1.1.100

   a. Start with Manage & Settings and check the new user under Permissions & Administrators –> Administrators and look for api_user.



b. Check the API settings under Blades –> Management API



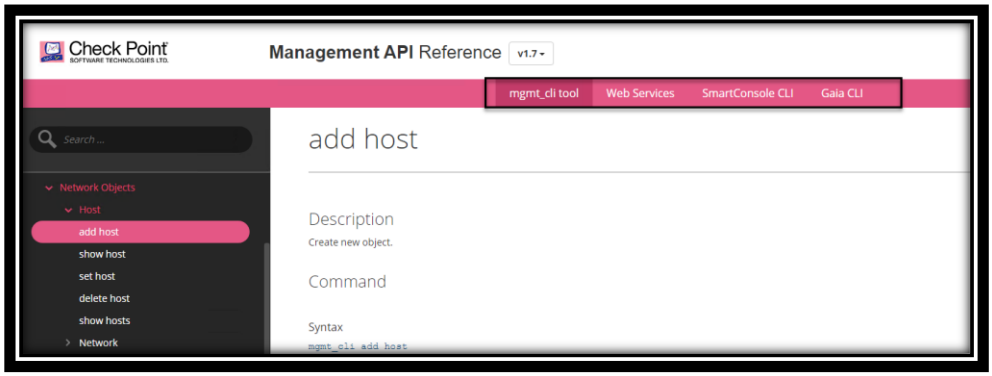
Task - 2: Add a Host using the SmartConsole CLI

Objective: Create a host with the R81 API using the SmartConsole CLI. Use the R81 Management API guide as a reference.

1. Open Chrome on the desktop and navigate to (https://203.0.113.80/api_docs) to access the API reference.

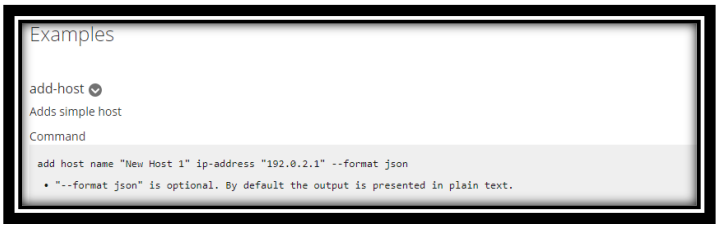
2. Select Network Objects –> Host –> add host in the navigation tree on the left side of the page.

3. Notice the four options for what method is used to access the API. The four methods are
    a. **SmartConsole CLI**: Used in SmartConsole
    b. **Mgmt_cli** tool: Used from any Gaia or Windows system with the local mgmt_cli command
    c. **Gaia CLI**: Used from within CliSH in Gaia to access the API
    d. **Web Services**: Restful access to the API – remember that we need GUI client access to use this method of the API
        i. Each method is listed in the API guide. Click SmartConsole CLI

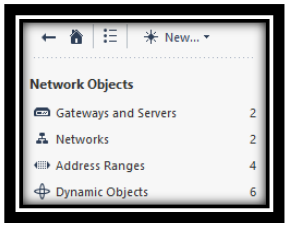


4. Review the options of how to add a host using the SmartConsole CLI. Check the example at the bottom of the page.

☞ **Note**: *Only certain parameters are **required** for each command as noted in the API guide.*

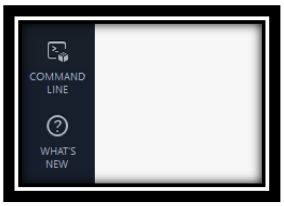


5. Move back to the SmartConsole and open Network Objects on the right side of the page.



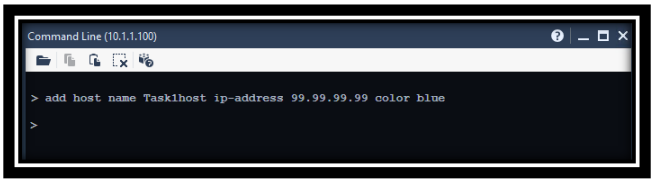
6. Notice there are no hosts configured on the management server since there are no counts for hosts listed in this view.

7. Open the SmartConsole CLI from the left side of the SmartConsole –> the icon is named Command Line.

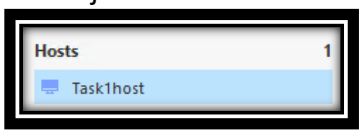


☞ ***Note**: The Command Line option is not the Gaia Command Line. It is the SmartConsole CLI used with the API only!*

8. In the SmartConsole CLI add a host with the following
    a. Name Task1host
    b. IP address 99.99.99.99



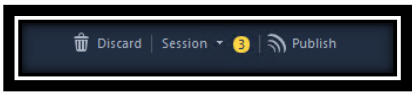
9. Review the host that was added using the Networks Objects on the right side of the screen. Notice that a host is now listed on the object tree.



10. Now using the R81 API Guide, delete the host Task1host that was created using the SmartConsole CLI.

☞ *Note: Each command has different **required parameters**. In the case of the delete command, the only required parameter is either **name** or **UID**.*

11. Publish your changes using the publish button at the top of the SmartConsole.
    All changes done with the SmartConsole CLI will use the publish/discard options from SmartConsole.



📖 *Using the publish command from the SmartConsole CLI is **not supported**. The only method to publish changes from the SmartConsole CLI is to use the Publish button within the SmartConsole.*

Task – 3: Add a Host using the mgmt_cli

Objective: Create a host with the R81 API using mgmt_cli from the management server. Use the R81 Management API guide as a reference.

1. Using the R81 API Guide, review the command to add host using the mgmt_cli tool.

📖 Reminder: Examples of each API command is listed at the bottom of each page and all four methods of using the API are listed.

2. Switch back to the SSH window to the R81 Management server and add a host named Task2host with an IP address of 22.22.22.22
   a. Use the username api_user
   b. Use the password vpn123

```
[Expert@Mgmt:0]# mgmt_cli add host name Task2host ip-address 22.22.22.22
Username: api_user
Password:


-----------------------------------------------
Time: [10:43:31] 19/4/2021
-----------------------------------------------
"Publish operation"  in progress  (80%)


-----------------------------------------------
Time: [10:43:41] 19/4/2021
-----------------------------------------------
"Publish operation"  succeeded  (100%)
```

- *Notice that the **publish** operation happens **automatically** as part of this command.*

📖

- *When a single API command or command without a session ID is run using the mgmt_cli an implicit publish will happen when the command completes.*

3. Next, delete the host. Use the API guide as a reference for removal. This time pass the user id "api_user" to the command using the -u flag.

```
[Expert@Mgmt:0]# mgmt_cli delete host name Task2host -u api_user
Password:


----------------------------------------------
Time: [10:47:14] 19/4/2021
----------------------------------------------
"Publish operation"  succeeded  (100%)
```

4. Now run the same command to add the host, but use the following flags when running the command "-r true"

```
[Expert@Mgmt:0]# mgmt_cli -r true add host name Task2host ip-address 22.22.22.22


----------------------------------------------
Time: [10:48:12] 19/4/2021
----------------------------------------------
"Publish operation"  succeeded  (100%)
```

📖 *The **–r true** option allows the user to receive **SuperUser** permissions. No additional login credentials are required. **This flag will only function if running on the local management server**.*

5. Delete the host Task2host using the mgmt_cli

```
[Expert@Mgmt:0]# mgmt_cli -r true delete host name Task2host


----------------------------------------------
Time: [10:50:34] 19/4/2021
----------------------------------------------
"Publish operation"  succeeded  (100%)
```

Task – 4: Add multiple Hosts from a CSV file

Objective: Learn how to use the R81 Management API to add multiple elements from a CSV file. To use CSV we utilize a batch function in the API to add multiple elements.

1. In the SSH session make sure you are in the R81_Mgmt_Scripts directory on the management server. Use the "*pwd*" command to verify.

2. Review the hosts.csv file by running the command: `cat hosts.csv`

```
[Expert@Mgmt:0]# cat addhost
mgmt cli -r true add-host --batch hosts.csv
```

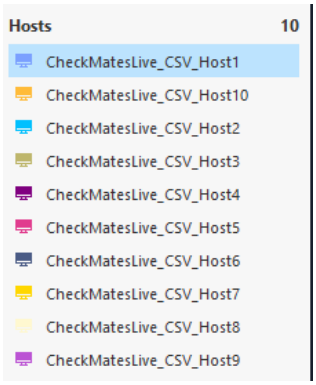
   a. The first line of the file describes the contents of the file and the data that will be run for each host.
   b. There are four items being configured for each host: Name, IPss-address, Color and Tags.

```
[Expert@Mgmt:0]# cat hosts.csv
name,ip-address,color,tags
CheckMatesLive_CSV_Host1,10.1.1.1,blue,myhost1
CheckMatesLive_CSV_Host2,10.2.1.1,cyan,myhost2
CheckMatesLive_CSV_Host3,10.3.1.1,khaki,myhost3
CheckMatesLive_CSV_Host4,10.4.1.1,orchid,myhost4
CheckMatesLive_CSV_Host5,10.5.1.1,pink,myhost5
CheckMatesLive_CSV_Host6,10.6.1.1,dark blue,myhost6
CheckMatesLive_CSV_Host7,10.7.1.1,gold,myhost7
CheckMatesLive_CSV_Host8,10.8.1.1,lemon chiffon,myhost8
CheckMatesLive_CSV_Host9,10.9.1.1,purple,myhost9
CheckMatesLive_CSV_Host10,10.10.1.1,orange,myhost10
```

   c. When adding elements via CSV only **one type** of command **is permitted** per csv file. We **cannot** mix hosts, services, rules in the same file. To accomplish this type of task we would need to run each of them separately.

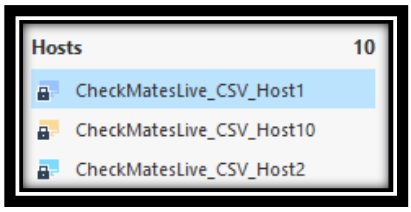
`[Expert@Mgmt:0]# bash addhost`

3. Review the SmartConsole and see the new hosts that were created



4. Overlay the SSH session on top of the SmartConsole so you can see the hosts on the right of SmartConsole. Watch the objects lock and disappear when the following command is run to remove the objects.

```
[Expert@Management:0]# bash deletehost
```



📖 *The mgmt_cli can be used in Gaia as well as on Windows. Any SmartConsole install includes a portable version of **mgmt_cli** to access the R81 management API server. Remember that access to the API could be restricted and can be setup to mirror the SmartConsole GUI client access.*

Task – 5: Add a SmartTask to verify user is authorized to make changes

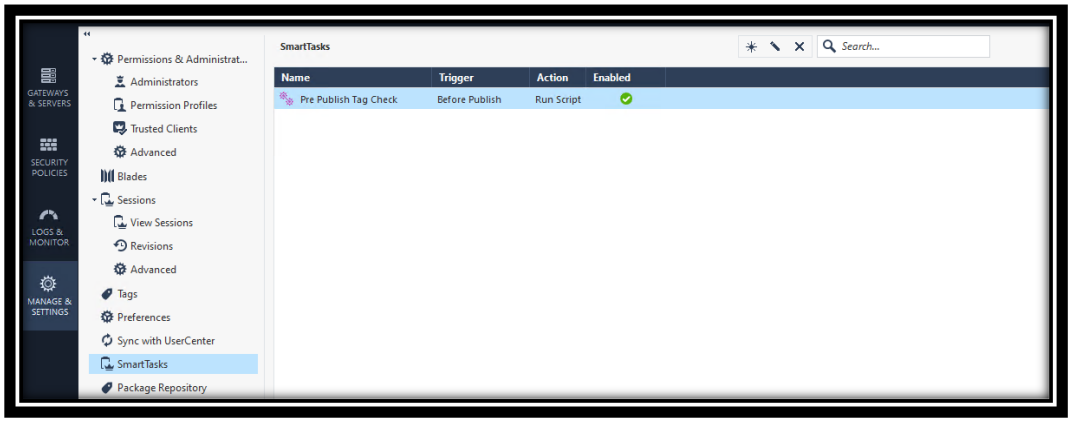
Objective: Using the SmartTask feature of R81, enable a verification process that verifies the user cpx_user is allowed to make changes to hosts and networks with the tag "Web_Control". Other changes should be not permitted and publishes should fail.

1. From the R81 management server SSH shell, run the following script.

```
Bash Setup_SmartTask.sh
```

2. Answer the questions with the following information (Case is sensitive!!!).

3. Review the new SmartTask that was created.



4 . Review the "Pre Publish Tag Check". Notice the information we entered is part of the Custom Data. This is the user and tag we will pass to the script when checking when Publishing changes

5. Next, we will add a host without the required tag and see the result. Use the following command and login as cpx_user.

```
[Expert@Mgmt:0]# mgmt_cli -u cpx_user -p vpn123 add-host name Task4Host ip-address 44.44.44.44


----------------------------------------------
Time: [11:21:38] 19/4/2021
----------------------------------------------
"Publish operation"  in progress  (10%)


----------------------------------------------
Time: [11:21:48] 19/4/2021
----------------------------------------------
"Publish operation"  failed  (100%)
Publish failed, the following error has occurred:
tasks:
- task-id: "01234567-89ab-cdef-a264-53833336a949"
  task-name: "Publish operation"
  status: "failed"
  progress-percentage: 100
  suppressed: false
  task-details:
  - fault-message: "You can only modify objects with the tag Web_Control. The following objects don't have this tag: Task4Host "

Discarding the changes.
```

6. Now add the tag to the same command and see the result. The command is all one line.

mgmt_cli -u cpx_user -p vpn123 add-host name Task4Host ip-address 44.44.44.44 tags.1 "Web_Control"

7. The SmartTask allows a publish operation to occur since the tag is specified in the request.

```
[Expert@Mgmt:0]# mgmt_cli -u cpx_user -p vpn123 add-host name Task4Host ip-address 44.44.44.44 tags.1 "Web_Control"


----------------------------------------------
Time: [11:23:51] 19/4/2021
----------------------------------------------
"Publish operation"  in progress  (10%)


----------------------------------------------
Time: [11:24:01] 19/4/2021
----------------------------------------------
"Publish operation"  succeeded  (100%)
```


8. Next see if you can add a network object in the same manner with the cpx_user.

9. **Disable** the SmartTask using the API – use the API guide as a reference to complete this task.

Task – 6: Run a script against a remote gateway

Objective: This task will show how we can interact with a gateway using the API function run-script of the R81 Management API. Using this method opens many interesting options on gateway tasks from configuration to verification of clish settings in a programmatic way.

The R81 API can be used for more than just manipulating the management databases objects and rules. In this task we will show the routing table on the remote gateway. There is already a gateway that is configured with the name Internal-GW at 203.0.113.81 that we will use for this task.

1. Run the following command on the management server (All of the command below is on one line).

```
mgmt_cli -r true run-script script-name "Task5" script "clish -c 'show route'" targets.1 "Internal-GW"
```

```
[Expert@Mgmt:0]# mgmt_cli -r true run-script script-name "Task5" script "clish -c 'show route'" targets.1 " Internal-GW"
```

2. Review the output. Does the output in the SessionDescription look complete?

3. The responseMessage is a **base64** encoded hash of the complete output from the command. Using the next command we will show how to use this output.

4. Next run the command Task5-showroute on the R81 Management server in the SSH session

```
bash Task5-showroute
```

- This command will run the API command and strip out the responseMessage and decode the output using **base64** into the typical output we would see when running the command in clish

```
[Expert@Mgmt:0]# bash Task5-showroute


-----------------------------------------------
Time: [11:37:51] 19/4/2021
-----------------------------------------------
"Internal-GW - Task5"  succeeded  (100%)
Codes: C - Connected, S - Static, R - RIP, B - BGP (D - Default),
       O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA),
       A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed,
       NP - NAT Pool, U - Unreachable, i - Inactive

S              0.0.0.0/0           via 203.0.113.1, eth5, cost 0, age 6459
C              10.0.0.1/32         is directly connected, loop00
                                       License
C              10.1.1.0/24         is directly connected, eth0
C              10.1.2.0/24         is directly connected, eth1
C              10.1.3.0/24         is directly connected, eth2
C              10.1.4.0/24         is directly connected, eth3
C              127.0.0.0/8         is directly connected, lo
C              203.0.113.0/24      is directly connected, eth5
                                       External
```


5. Now we will add a route to the gateway via the API. An example of this command is available in the script Task5-step6

6. Run this command:

```
bash Task5-step6
```

7. Review the command that was run.

```
mgmt_cli -r true run-script script-name "Task5-addroute" script "clish -c ' set static-route 9.9.9.9/32 nexthop gateway address 203.0.113.1 on'" targets.1 "Internal-GW"
```

- This command will add a static-route for 9.9.9.9 and save it to the Gaia configuration

8. Now re-run the show route API command: bash Task5-showroute

```
[Expert@Mgmt:0]# bash Task5-showroute


-----------------------------------------------
Time: [11:58:03] 19/4/2021
-----------------------------------------------
"Internal-GW - Task5"  succeeded  (100%)
Codes: C - Connected, S - Static, R - RIP, B - BGP (D - Default),
       O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA),
       A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed,
       NP - NAT Pool, U - Unreachable, i - Inactive

S              0.0.0.0/0           via 203.0.113.1, eth5, cost 0, age 7672
S              9.9.9.9/32          via 203.0.113.1, eth5, cost 0, age 121
C              10.0.0.1/32         is directly connected, loop00
                                       License
C              10.1.1.0/24         is directly connected, eth0
C              10.1.2.0/24         is directly connected, eth1
C              10.1.3.0/24         is directly connected, eth2
C              10.1.4.0/24         is directly connected, eth3
C              127.0.0.0/8         is directly connected, lo
C              203.0.113.0/24      is directly connected, eth5
                                       External
```

When using the run-script function it is easy to run scripts to pull information, see running configurations, or other statuses by using the API.

Task - 6: Use the mgmt_cli with a session

Objective: The API can be used for more than one command. Using the API with sessions allows multiple changes to be made within the session before it is published or discarded.

1. Run the following command to login

```
mgmt_cli login -u api_user -p vpn123 > sid.txt
```

When the session information is sent to a file we can reuse the session information for more API calls before we publish/discard

2. Review the contents of the sid.txt file. Run the following command

```
cat sid.txt
```

- The element we will focus on is the **sid**. This is our session id and will be used in the rest of this task.

```
[Expert@Mgmt:0]# mgmt_cli login -u api_user -p vpn123 > sid.txt
[Expert@Mgmt:0]# cat sid.txt
uid: "dedba69f-68bc-4a67-b70f-6c77515b27c5"
sid: "DAIuqEWtl79aN33yy-A14xgaWS57Fs6-2xehOFeGxj4"
url: "https://127.0.0.1:443/web_api"
session-timeout: 600
last-login-was-at:
  posix: 1618843632690
  iso-8601: "2021-04-19T10:47-0400"
api-server-version: "1.7"
```

3. Now we will add new tcp service as an example. Run the following command:

```
mgmt_cli add service-tcp name newtcp port 2021 -s sid.txt.
```

```
[Expert@Mgmt:0]# mgmt_cli add service-tcp name newtcp port 2021 -s sid.txt
```

4. Check in SmartConsole and see if the service appears
   - **a)** Why is it not in SmartConsole? Did we publish our session changes?
   - **b)** Check in SmartConsole number Manage & Settings/Sessions and look for your change. Do you have locks or changes?

5. Next we will run publish to apply this new TCP service so everyone could see and use the new service.

6. Run the following command:

```
mgmt_cli publish –s sid.txt
```

```
[Expert@Mgmt:0]# mgmt_cli publish -s sid.txt

----------------------------------------------
Time: [12:08:12] 19/4/2021
----------------------------------------------
"Publish operation"  succeeded  (100%)
tasks:
- task-id: "01234567-89ab-cdef-9e64-15b65d313c0c"
  task-name: "Publish operation"
  status: "succeeded"
  progress-percentage: 100
  suppressed: false
  task-details:
  - publishResponse:
      numberOfPublishedChanges: 3
      mode: "async"
    revision: "dedba69f-68bc-4a67-b70f-6c77515b27c5"
```

7. Almost as important as the publish command is the logout. There are a limited number of sessions a user can have. If the logout command is not run, sessions may need to be manually cleared.

8. View the number of sessions on the management server using the following command:

```
mgmt_cli show sessions details-level full -s sid.txt
```

- Review the output from the command
- Look for expired-session and locks

📖 *Any one user has up to 100 sessions that can be used in R81.*

9. Now run the logout command as shown below to clear our sessio

```
mgmt_cli logout –s sid.txt
```

End of Lab 1

