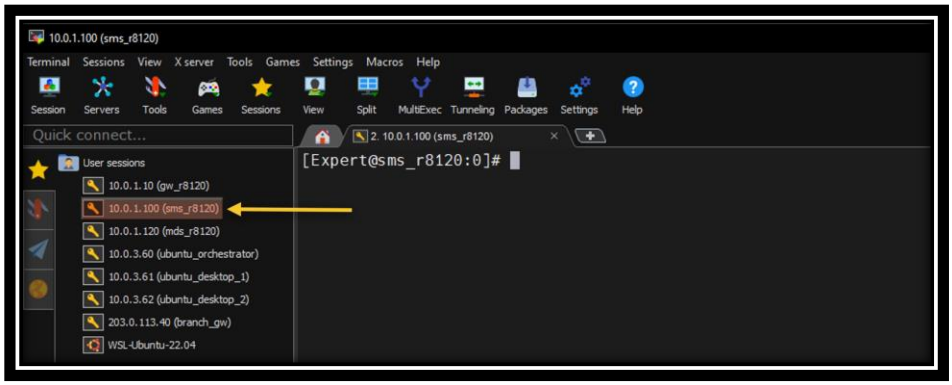
Management CLI Automation

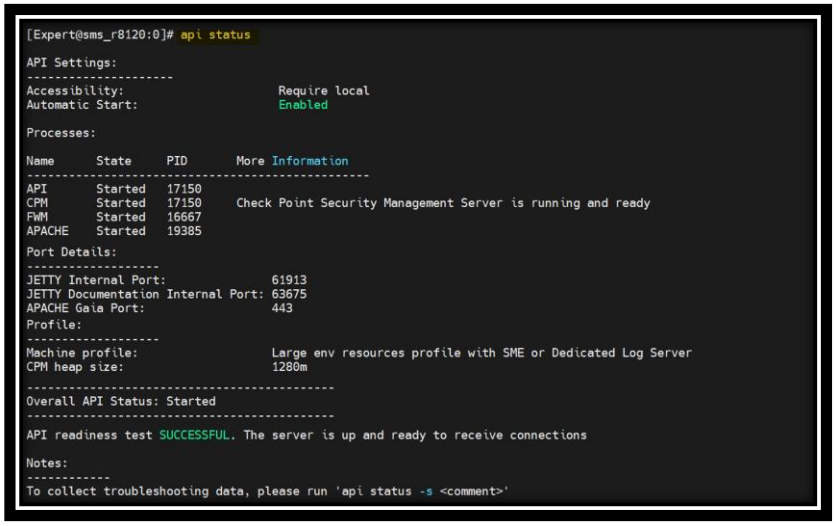
Task 1: Login to the Environment and API setup

Objective: Connect to the Lab Environment, review the default API settings, and configure and start the R81.20 Management API server.

1. Login to the management server `10.0.1.100(sms_r8120)` over `SSH`. The session is saved in `MobaXterm`.



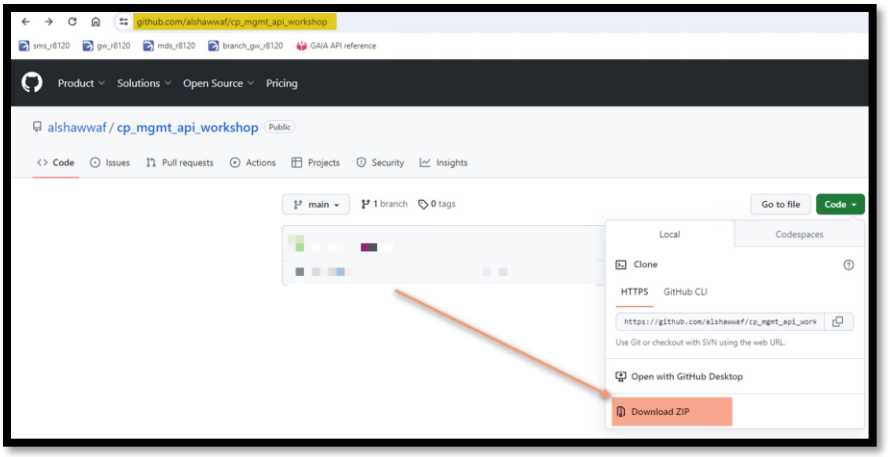
2. Run the following command `api status`.



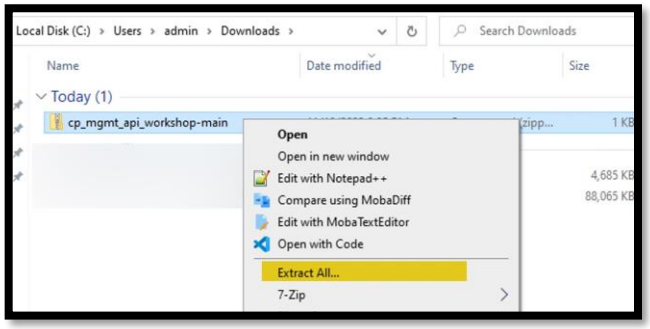
3. Review the output and check the following settings:
   - `API Settings -> Accessibility`
   - `API Settings -> Automatic Start`
   - `Overall API Status.`

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

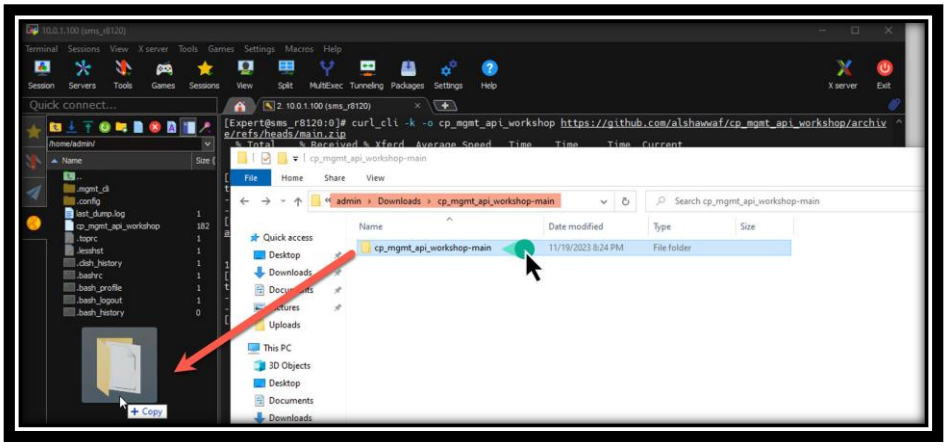
4. Download the demo files from the following GitHub link:
   `https://github.com/alshawwaf/cp_mgmt_api_workshop.git`



5. The file is downloaded to the `Download` folder. Extract the zip file.



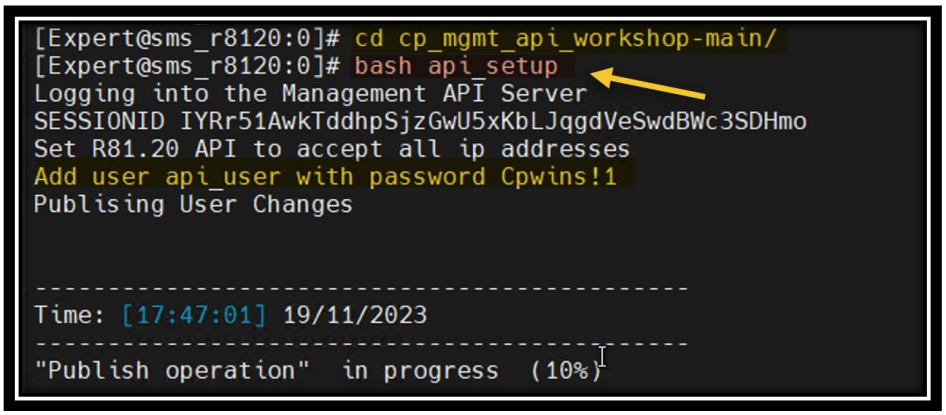
6. To copy the folder to the management server, drag the **sub** folder (*there is sub folder with the same name)* into the management SCP session in `MobaXterm`.



7. Change the directory to the folder we just copied `cp_mgmt_api_workshop-main`. Run the script to enable the management API external access using the command `bash api_setup`

8. The API server is configured to allow only 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 `Cpwins!1` as an additional user. Run the following script:

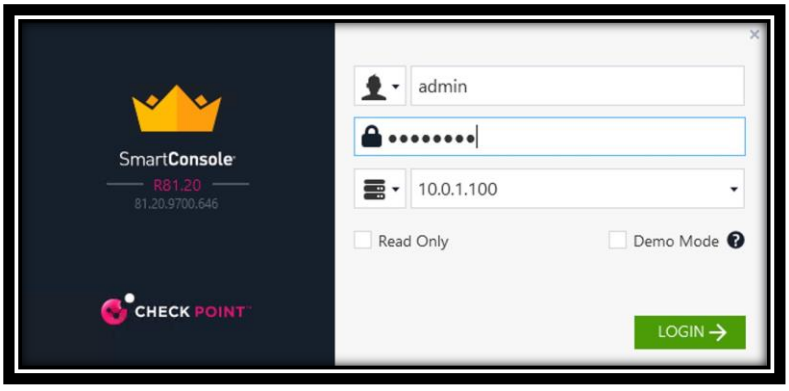
```
[Expert@sms_r8120:0]# bash api_setup
```



This script does the following:

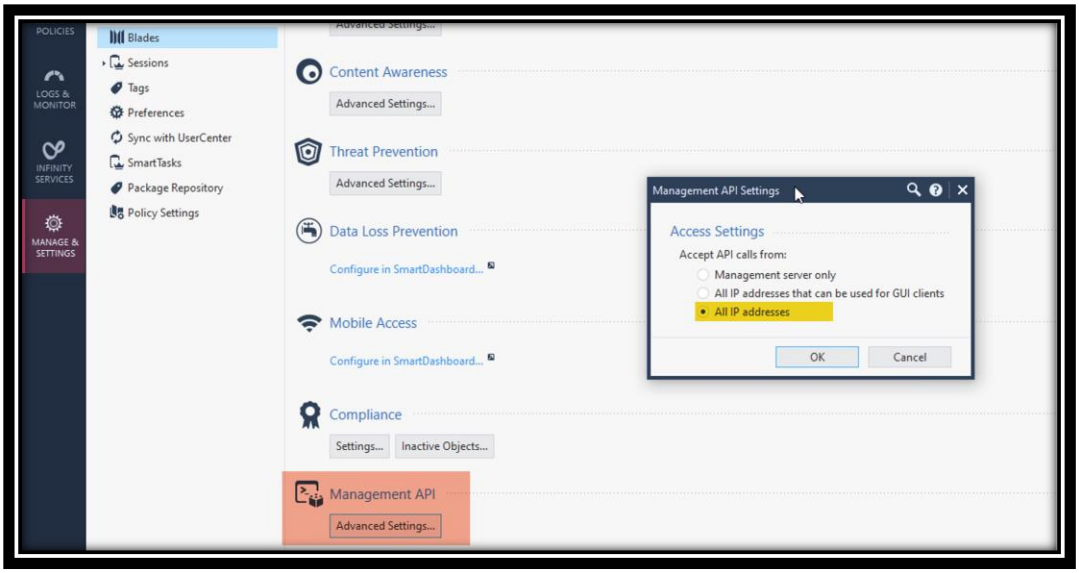
- Adds a new user `api_user` with the password `Cpwin!1`
- Changes the API server to accept `all IP addresses`.
- Apply the new API server using the command `api reconf`.
- Runs the `api status` command.

9. Log in to R81.20 SmartConsole and review the changes that were made:
    - username: `admin`
    - password: `Cpwins!1`
    - 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`



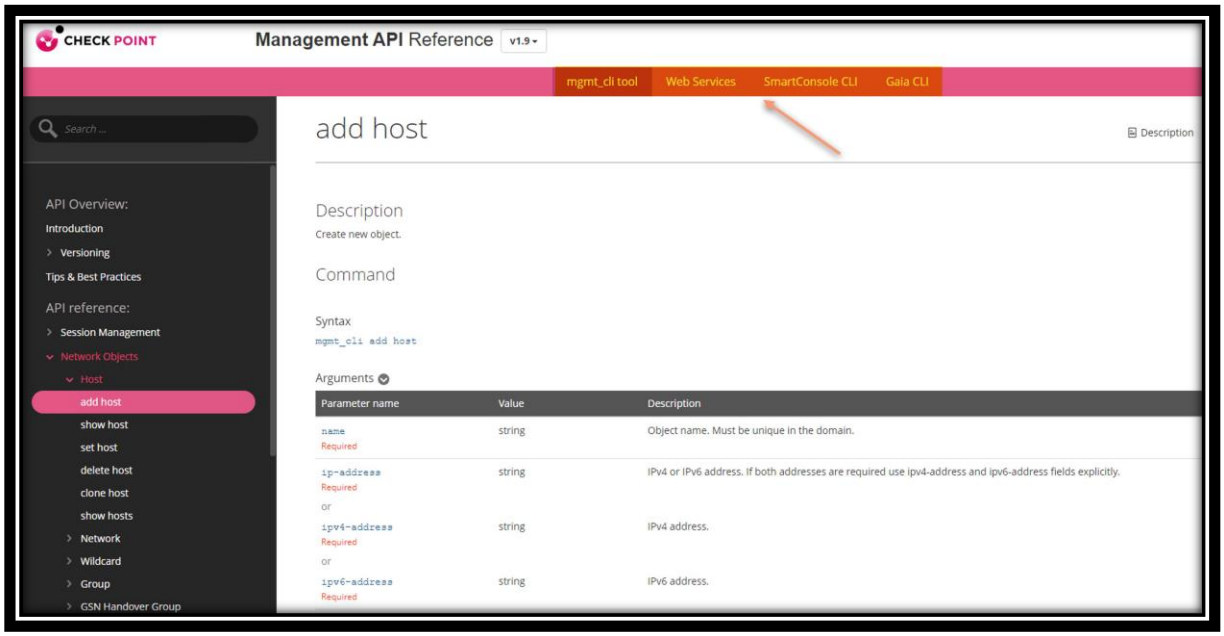
Task - 2: Add a Host using the SmartConsole CLI

Objective: Create a host with the R81.20 API using the SmartConsole CLI. Use the Management API guide as a reference.

1. Open Chrome on the desktop and navigate to `https://10.0.1.100/api_docs/` to access the API reference.

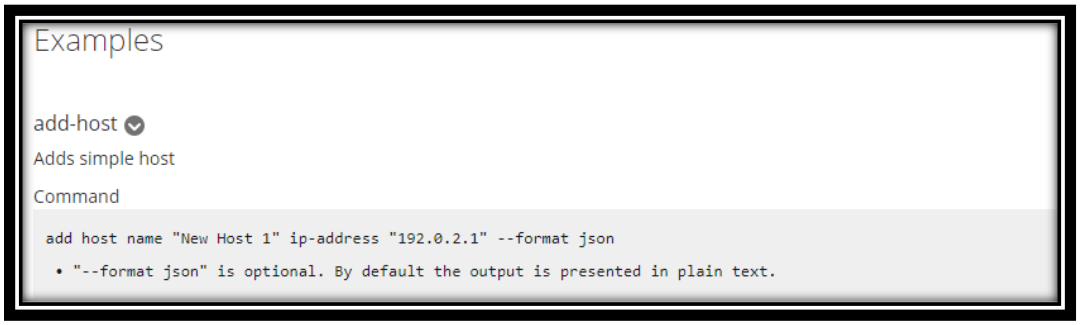
2. Select `Network Objects -> Host -> add host` in the navigation tree on the left side of the page.

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

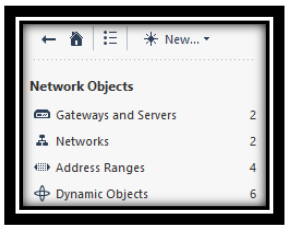


4. Review the options of how to add a host using the SmartConsole CLI. Check the example at the bottom of the page.

☞ **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.



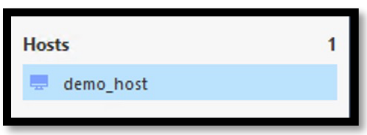
☞ *Notice there are no hosts configured on the management server since there are no counts for hosts listed in this view.*

6. Open the SmartConsole CLI from the left side of the `SmartConsole -> the icon is named Command Line`.

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

7. In the SmartConsole CLI add a host with the following
   a. `Name: demo_host`
   b. `IP address: 99.99.99.99`

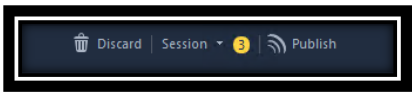
8. Review the host that was added using the Networks Objects on the right side of the screen. Notice that a host is now listed on the object tree.



9. Now using the API Guide, delete the host `demo_host` 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**.*

10. 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 R81.20 API using mgmt_cli from the management server. Use The Management API guide as a reference.

1. Using the API reference 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 R81.20 Management server and add a host named cp_host with an IP address of 22.22.22.22
   a. Use the username `api_user`

b. Use the password `Cpwins!1`

```
[Expert@sms_r8120:0]# mgmt_cli add host name cp_host ip-address 22.22.22.22
Username: api_user
Password:

----------------------------------------------
Time: [08:59:05] 20/11/2023
----------------------------------------------
"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@sms_r8120:0]# mgmt_cli delete host name cp_host -u api_user
Password:

----------------------------------------------
Time: [09:01:44] 20/11/2023
----------------------------------------------
"Publish operation"  succeeded  (100%)
```

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

```
[Expert@sms_r8120:0]# mgmt_cli -r true add host name cp_host ip-address 22.22.22.22

----------------------------------------------
Time: [09:03:48] 20/11/2023
----------------------------------------------
"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` tool.

```
[Expert@sms_r8120:0]# mgmt_cli -r true delete host name cp_host

----------------------------------------------
Time: [09:05:41] 20/11/2023
----------------------------------------------
"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 directory we copied to the management server. Use the "*pwd*" command to verify.

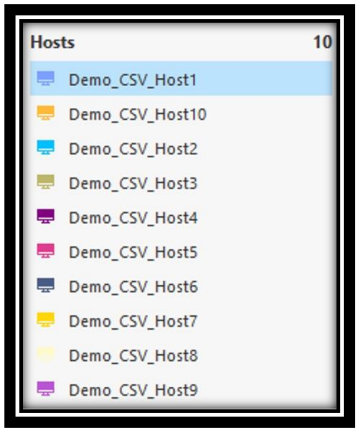
2. Review the hosts.csv file by running the command: `cat hosts.csv`

```
[Expert@sms_r8120:0]# cat hosts.csv
name,ip-address,color,tags
Demo_CSV_Host1,10.1.1.1,blue,myhost1
Demo_CSV_Host2,10.2.1.1,cyan,myhost2
Demo_CSV_Host3,10.3.1.1,khaki,myhost3
Demo_CSV_Host4,10.4.1.1,orchid,myhost4
Demo_CSV_Host5,10.5.1.1,pink,myhost5
Demo_CSV_Host6,10.6.1.1,dark blue,myhost6
Demo_CSV_Host7,10.7.1.1,gold,myhost7
Demo_CSV_Host8,10.8.1.1,lemon chiffon,myhost8
Demo_CSV_Host9,10.9.1.1,purple,myhost9
```

   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, IP-address, Color and Tags.
   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@sms_r8120:0]# 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 API server. Remember that access to the API could be restricted and can be setup to mirror the SmartConsole GUI client access.*

Task – 5: Run a script against a remote gateway

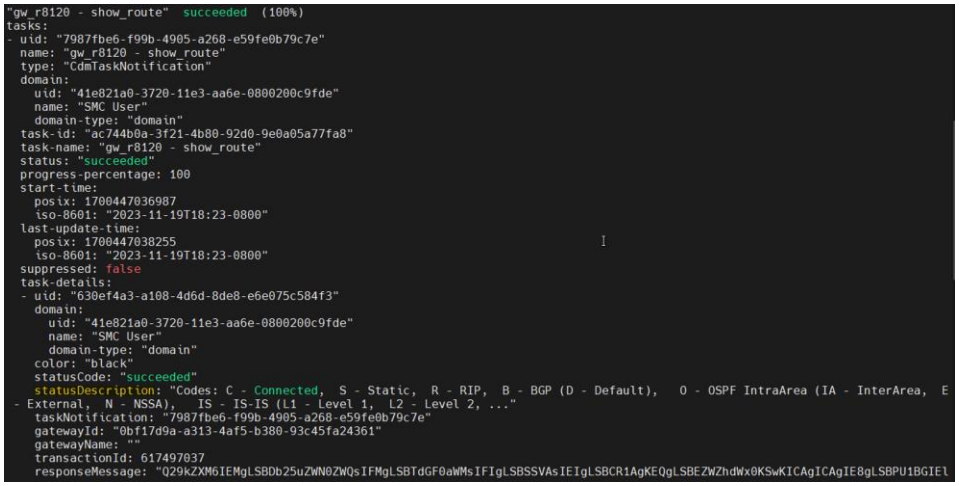
Objective: This task will show how we can interact with a gateway using the API function run-script of the R81 Management API. Using this method opens many interesting options on gateway tasks from configuration to verification of clish settings in a programmatic way.

The Management API can be used for more than just manipulating the management database 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 gw_r8120 at 10.0.1.100 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 "show_route" script "clish -c 'show route'" targets.1 "gw_r8120"
```

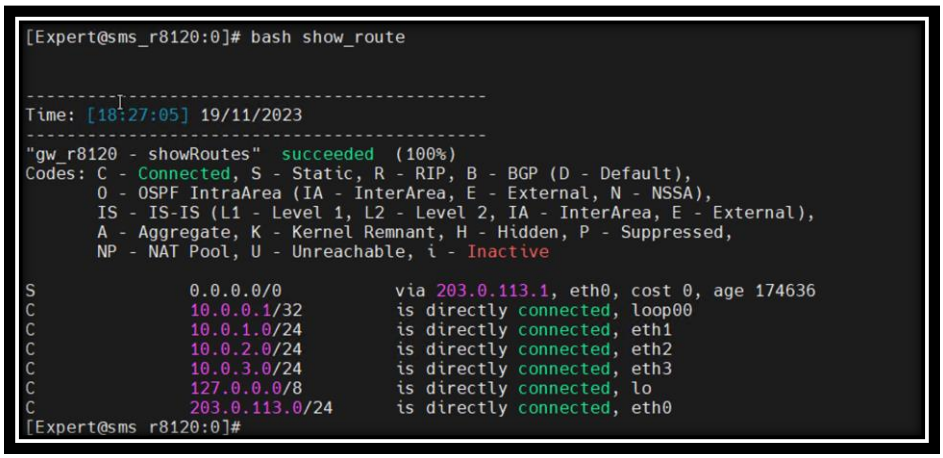
2. Review the output. Does the output in the SessionDescription look complete?



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 bash show_route on the R81.20 Management server in the SSH session
   `bash show_route`

   - 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

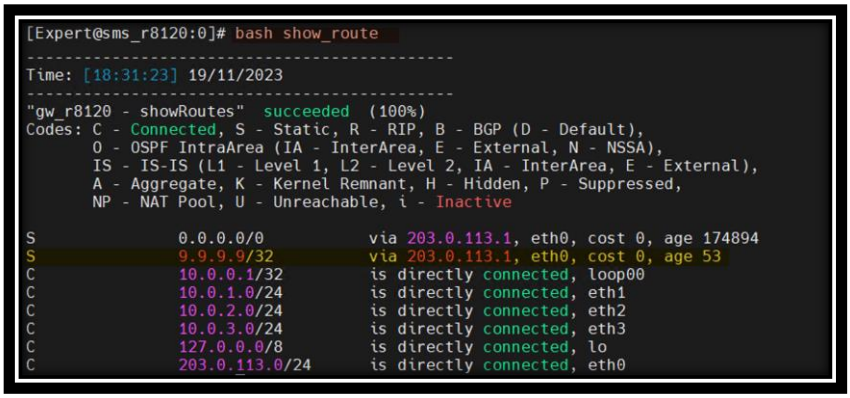


5. Now we will add a route to the gateway via the API. An example of this command is available in the script `add_route`. Run this command:
   `bash add_route`

6. Review the command that was executed.
   `mgmt_cli -r true run-script script-name "add_route" script "clish -c ' set static-route 9.9.9.9/32 nexthop gateway address 203.0.113.1 on'" targets.1 "gw_r8120"`

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

7. Now re-run the show route API command: `bash show_route`



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

Task - 6: Use the mgmt_cli with a session

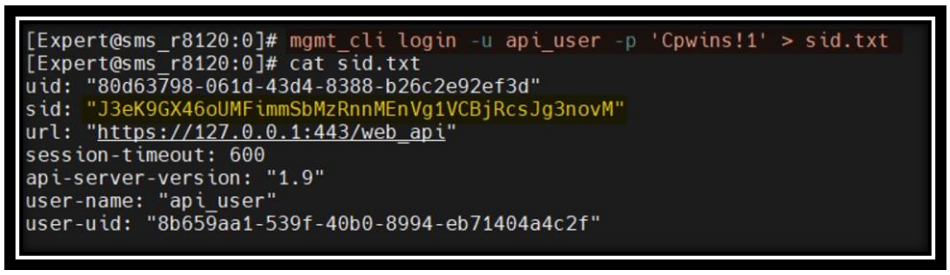
Objective: The API can be used for more than one command. Using the API with sessions allows multiple changes to be made within the session before it is published or discarded.

1. Run the following command to login

```
mgmt_cli login -u api_user -p 'Cpwins!1' > 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.



3. Now we will add new TCP service as an example. Run the following command:
   `mgmt_cli add service-tcp name new_tcp port 2024 -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 can see and use the new service. Run the following command:
   `mgmt_cli publish -s sid.txt`


```
[Expert@sms_r8120:0]# mgmt_cli publish -s sid.txt


----------------------------------------------
Time: [18:39:43] 19/11/2023
----------------------------------------------
"Publish operation"  succeeded  (100%)
tasks:
- task-id: "01234567-89ab-cdef-bac1-10ab73acfa1e"
  task-name: "Publish operation"
  status: "succeeded"
  progress-percentage: 100
  suppressed: false
  task-details:
  - publishResponse:
      numberOfPublishedChanges: 1
      mode: "async"
    revision: "80d63798-061d-43d4-8388-b26c2e92ef3d"
```

6. 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.

7. 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 R81.20.*

8. Now run the logout command as shown below to clear our session:

```
mgmt_cli logout -s sid.txt
```

End of Lab 1

