

08. Multi-Node Script

Multi-Node Script

Video Transcript

Hello, this is Amir Shehata with another quick tip on the LUTF.

Today we'll cover how a script can be written to execute on multiple nodes.

Let's jump right into it.

Script Description

Here is a simple script which gathers the interfaces on the remote and returns them to the main script.

Let's do a quick overview of the script

The LUTF provides a BastTest class which implements the remote execution feature.

The script needs to define a class which inherits from the BaseTest class.

The magic happens when you define a constructor for your inherited class which calls the constructor of the parent class, passing it the absolute path to the script and the target to execute on.

This is standard for every script. To make life easier the LUTF provides an "add" feature to create a template script which you can then modify. I'll quickly show that at the end.

This class has a get_intfs_and_add() method. It collects the interfaces from the me variable, does a sum and returns the values.

In the run function:

- I'll find any available agent.
- Instantiate my class passing it the name of that agent
- call the get_intfs_and_add() method
- Return the results back to the LUTF using the lutfrc() function.

That's it.

Let's look at the local interfaces first before we look at the results to prove that the method was actually executed on the remote.

Let's run the script

Now let's look at the Global YAML result

It's a simple as that.

Creating a New Script

Let's show how you can create a new script:

```
suites['sample'].add("sample_11")
suites['sample'].scripts['sample_11'].edit()
```

Now you can develop the script to your heart's desire.

In the next tutorial we'll go over how the LUTF integrates in the lustre test infrastructure.