Frequently Asked Questions About F5 Technology

Colin Walker, 2012-11-12

F5 is the creator of and home to many different, powerful technologies. We offer some of the most powerful, advanced application delivery technology in the world. As with any advanced technology, however, there are often times terms and names flying around in documentation, publications or online communities that aren't necessarily part of your daily speech. Until you're a bit "in the know" some of these things can be confusing or even daunting. As such, we've gone through and created a bit of a glossary to describe, albeit briefly, what each of these highly used F5 terms means.

Ideally this will help you get your feet wet and get started digging deeper into F5 technology and terms, or acquaint you with some background or information you may not have already known. For deeper information on each subject there are links to go learn more if you're a "roll up your sleeves" type person. We encourage that, after all, so feel free to jump in where applicable and we'll do our best to offer any questions you may have.

iRules

iRules is a custom built scripting language that allows you complete access to the traffic on the wire, in a programmatic sense. If you're looking to inspect, analyze, modify, route, re-direct, discard, manipulate, re-play, mirror or…just about anything else you can think of, chances are you can do it with an iRule. While based largely on the open source Tcl language, a huge amount of customization and improvement by specificity has been baked into iRules in the many years it has been in place in our products. The ability to write business logic into your network layer in a fully fledged programming language is a truly powerful and liberating thing. This unique and formidable technology allows you to effectively extend your application layer into the network, better marrying the two and sharing the load where it makes sense.

While there is a near endless amount that can be done with iRules, you don't have to be an expert to get started. You don't even need any prior programming experience. If you're comfortable writing very basic (meaning 2 or 3 line) scripts in any language, shell or otherwise, iRules will be simple to pick up. We have a host of documentation to get you started, a vibrant community of users eager to answer questions or swap code, and hundreds of examples in our CodeShare. To check out the power that iRules can offer, and read more about how it is changing the role the network plays for thousands of companies around the world, there's more info below.

https://devcentral.f5.com/wiki/iRules.HomePage.ashx

iControl

As iRules is to network traffic, iControl is to F5 configurations. iControl is our open, web services based API that allows complete, dynamic, programmatic control of F5 configuration objects. This means you can add, modify or remove bits from your F5 device on the fly, automatically. Whether you're looking to add a list of Virtual Servers, shut down half of the 400 members in a pool for an upgrade, or track more advanced stats by writing a script to poll certain data.; the uses for iControl are near limitless. Allowing this level of fine grained control and access has proven useful time and time again, as many of our users have come to truly rely on the API for automating management tasks in the large scale environments in which BIG-IP is often deployed. Some even use it to design custom interfaces for particular groups of users, or to integrate directly with an existing portal to allow their applications to tie straight into F5 technology.

https://devcentral.f5.com/wiki/iControl.HomePage.ashx

iApps

iApps are a relatively new player in the F5 world, having only been introduced in version 11. iApps are a powerful new way to allow you to automate tasks and template-ize sets of functionality on your F5 device. Whether you're automating the way you add HTTP virtual servers, so you don't have to go through the manual steps every time you add a new application, or building a custom iApp to manage your iRules inventory, this new, powerful, question driven, customizable functionality offers a huge amount of possibilities. The way it works is that you are able to write what's called "APL" (Application Presentation Language) which is a simple scripting language used to define the interface with which users will interact with your iApp application. You decide which questions to ask them, which data to enter or objects within your configuration to select, etc. Then, with that data, you're able to automate nearly any task on the device. This can easily allow you to take a 50 or 60 step process down to 2 or 3 easy steps, which is invaluable if you're either automating repeat tasks, or looking to obfuscate some complexity from users. This is only the tip of the iceberg of what iApps can be used for. There's a plethora of possibilities, and users exploring and sharing every day the new things that are coming in the world of iApps

https://devcentral.f5.com/wiki/iApp.HomePage.ashx

tmsh

tmsh (Traffic Management Shell) is the powerful new way in which you are able to control your F5 devices from the command line (CLI). If you're the type that wants to skip the GUI and get straight into SSH, I can't blame you, and tmsh is precisely how you can do that. tmsh was developed internally by F5 to allow complete access to all of the advanced features F5 devices are capable of, while incorporating user friendly features like tab completion. tmsh also offers a powerful way to script and automate tasks directly on your F5 device. Running scheduled tasks, repeating functions on demand and the like have never been easier on F5 devices. If you're not familiar yet, and want to be, DevCentral is a great place to get started learning about what tmsh has to offer.

https://devcentral.f5.com/wiki/TMSH.HomePage.ashx

TMOS/TMM

One of the things that makes F5 devices so powerful, as well as unique, is the fact that we have designed and built, from the ground up, a completely custom microkernel designed to do one thing, and one thing only: Process traffic. It is robust, powerful, and widely functional and it is extremely efficient. This kernel is known as our Traffic Management Microkernel (TMM), which is the heart of our Traffic Management Operating System (TMOS). These terms are important as they are at nerve center of F5's application delivery magic. Any data passed in or out on the wire passes through one of the TMMs on your device, and there are likely several, before arriving at its destination. There's a massive amount of technology packed into the core of our devices, and we're happy to help those that want a deeper understanding to get just that.

https://devcentral.f5.com/weblogs/macvittie/archive/2012/02/10/f5-friday-whatrsquos-inside-an-f5.aspx

Host OS

As much as TMM is amazing and powerful for passing traffic, it is purpose built for precisely that. Its very nature makes it incapable of things like hosting a web interface, dealing with a syslog daemon, or handling SSH connections. As such, every F5 device works on a Host OS/TMOS architecture where we have both running simultaneously, handling different tasks. The Host OS, which is a **nix** based OS, manages all of the "user" processes. Things that don't affect data being passed in real time, such as system logs, management access, clock management, etc. are all handled in the Host OS. There isn't as much to learn about the Host OS itself, but understanding the architecture of the technology is a powerful thing.

CMP

There are papers that already exist that depict CMP in a far more thorough and articulate manner than I could hope to achieve, so I will give the primer version to elucidate the very basics of the concept. CMP is "Clustered Multi-Processing". This is F5's proprietary way of dealing with multiple core devices. In essence, in a very rough sense, each core in a device is assigned its own individual TMM (see above) to handle processing of traffic for that core. There is then a custom disaggregator (DAG) built into the system that decides which TMM, and as such, which core to send traffic to for processing. In this way F5 is able to achieve massively linear scalability in multiple CPU, multiple core systems. That barely skims the surface of this technology, but when discussing iRules it is important to know that CMP is a good thing, and breaking or demoting from CMP is, generally speaking, bad. Hopefully this explains why that is, and what CMP is at a basic level.

http://www.f5.com/pdf/white-papers/viprion-clustered-multiprocessing-wp.pdf

VCMP

VCMP takes the concept of CMP one step further. Where CMP allows us to use multiple cores on a device to run multiple instances of our kernel, thereby effectively scaling traffic handling, VCMP allows us to use each of those cores to not only run a TMM instance controlled by the same Host OS, but to effectively turn the F5 device into a hypervisor, and run multiple, fully independent and insulated versions of F5 software on a single device. This means you're able to run multiple versions of LTM, GTM, or just about any other F5 product within the same device, and have them 100% logically separate. No crossing over of VLANs, resources, or any impact whatsoever on resource utilization. This also means that you can allow access to each device separately; manage individual upgrades and downtime, etc. Every instance is very much treated like its own stand-alone device in almost every way.

http://www.f5.com/pdf/white-papers/vcmp-wp.pdf