

[Resetting vSphere 6.x ESXi Account Lockouts via SSH](#)

VMware vSphere has had a good security feature added since vSphere ESXi 6.0 to add a root account lockout for safety. After a number of failed login attempts, the server will trigger a lockout. This is a good safety measure for when you have public facing servers and is even important for internally exposed servers on your corporate network. We can't always assume that it's external bad actors who are the only ones attempting to breach your devices.

Using the vSphere web client shows us the settings which are used to define the lockout count and duration. The parameters under the Advanced settings are as follows:

```
Security.AccountLockFailures  
Security.AccountUnlockTime
```



Resetting your Failed Login Attempts with pam_tally2

There is a rather simple but effective tool to help you do this. It's called `pam_tally2` and is baked in with your ESXi installation. The command line to clear the lockout status and reset the count to zero for an account is shown here with the root account as an example:

```
pam_tally2 --user root --reset
```



In order to gain access to do this, you will need to have SSH access or console access to your server. Console access could be at a physical or virtual console. For SSH access, you need to use SSH keys to make sure that you won't fall victim to the lockouts for administrative users. In fact, this should be a standard practice. Setting up the SSH keys is relatively simple and is nicely documented in the Knowledge Base article **Allowing SSH access to ESXi/ESX hosts with public/private key authentication (1002866)**

https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1002866

Uploading a key can be done with the `vifs` command as shown here:

<https://docs.vmware.com/en/VMware-vSphere/6.0/com.vmware.vsphere.security.doc/GUID-392ADDE9-FD3B-49A2-BF64-4ACBB60EB149.html>

The real question will come as to why you have the interface exposed publicly. This is a deeper question that we have to make sure to ask ourselves at all times. It's generally not recommended as you can imagine. Ensuring you always use complex passwords and 2-factor authentication is another layer which we will explore. Hopefully this quick tip to safely reset your accounts for login is a good first step.

[Installing PowerCLI 6.5.x on Windows Server 2012 R2 after Find-Module Error](#)

Now that PowerCLI is part of the PowerShell Gallery, you can install it using the native module installer...but there's a catch. Windows Server 2012 R2 requires a couple of minor updates to get this process underway. You'll know really quickly if you open up your PowerShell terminal or PowerShell ISE (as Administrator) and try the following command:

```
Find-Module -name VMware.PowerCLI
```



The issue is easily solved by deploying a more recent installer for the PackageManagement PowerShell Modules. Download the installer using this link and run the install:

<https://www.microsoft.com/en-us/download/details.aspx?id=51451>

Select the Download within the page once you're there:



Choose the x64 version (assuming you're running a 64-bit OS):



Run through the installation and accept the defaults. Nothing significant to worry about with this file as it's a necessary update for what we need to do.

If you run the `Find-Module` command again, you'll see a much better result. You'll be prompted to update your NuGet components which are used to pull resources from the PowerShell Gallery. Accept the update and then we can keep going:



Time to get back to the issues. Just relaunch your PowerShell terminal or ISE as an Administrator. We are running as Administrator so that we install the module for all users of the server. If you only want to run for your user then run your PowerShell session as your regular user and add `-scope CurrentUser` to the `Install-Module` command. Run the following to install for all users:

```
Install-Module -name VMware.PowerCLI
```



Now we have to import the module into our session using the `Import-Module -name VMware.PowerCLI` command:



Just like that, you're up to date and running the latest and greatest PowerCLI goodness. Happy

scripting!

Getting Terraform Provisioning Parameters from the Packet.net API

Provisioning on [Packet.net](#) is super easy using Terraform. One of the tricks you will need to know up front is that for Terraform and for many other provisioning tools, you need to provide a minimum set of parameters to launch.

As a minimum, you need to provide these following parameters as shown in the [Terraform docs for the Packet provisioner](#):

- **hostname** - gotta name 'em all
- **project_id** - you need to know, or create the project to launch into
- **facility** - which location are you deploying into? (EWR1, SJC1, etc.)
- **plan** - which node type?
- **billing_cycle** - hourly or monthly
- **operating_system** - which OS will the node run?

Some are simple to use because they are your criteria. We choose the hostname, and we choose the billing cycle as either a static choice of hourly or monthly. How can we get the other details about our deployment? You can gather some data using a browser such as browsing to your project and then pulling the project ID from the URL. That still leaves us in search of the plan type, operating_system, and facility.

For completeness, let's learn how to simply gather all four items (operating system list, project ID, plan types, facility) from the Packet.net API.

You'll need a terminal session, your API key to query the Packet.net API, and the JQ tool for parsing out JSON results into something a little more friendly.

Querying the API is as easy as sending your token to the API using the cURL command and selecting which entities you want to query. This is the basic framework:

```
curl -s -X GET -H 'X-Auth-Token: YOURAPITOKEN'  
'https://api.packet.net/OBJECT'
```

Now we can dig into the four easy examples we have.

Finding the Packet.net Facility Name

The simple one-liner will pull a JSON result that gives you the locations and subsequent Facility name that you can use and then parses out just the location codes to use. If you remove the `'.facilities[].code'` portion of the command it will show you the full pretty-printed JSON results including the full facility descriptions.

```
curl -s -X GET -H 'X-Auth-Token: YOURAPITOKEN'  
'https://api.packet.net/facilities' | jq '.facilities[].code'
```

Finding the Packet.net Project ID

You'll want the full JSON result so you can choose from your active projects if you have more than one. Just drill into the JSON results and you can locate the **id** field:

```
curl -s -X GET -H 'X-Auth-Token: YOURAPITOKEN'  
'https://api.packet.net/projects' | jq
```

Finding the Packet.net Plan Names

Plans don't shift around too much, just like facilities. Here is the simple query to get all the plan names and match them to what node type you want to use:

```
curl -s -X GET -H 'X-Auth-Token: YOURAPITOKEN' 'https://api.packet.net/plans'  
| jq '.plans[].slug'
```

Finding the Packet.net Operating System Types

By now, you can guess where we are going with the next one. Query the API, parse out the results, and provide the slugs for the Operating System names which we will use for Terraform and other provisioning tools which consume the Packet API.

```
curl -s -X GET -H 'X-Auth-Token: YOURAPITOKEN'  
'https://api.packet.net/operating-systems' | jq '.operating_systems[].slug'
```

The result will give you all of the slug names that are usable as the `operating_system` parameter. In the case of vSphere 6.5, it happens to be **vmware_esxi_6_5** which may not have been obvious if you were to try guessing it out.



Now you can take those easy JSON results and feed them into a Terraform file or you may also use these raw queries as part of other configuration management and provisioning solutions. Hope you find this helpful!

Also, you can sign up for Packet.net to kick the tires on this goodness and you can use **VDM25** as a referral code to get a 25\$ credit to use. Make sure you tell them DiscoPosse and the Virtual Design Master crew sent you!

[The Goal Graphic Novel is here!](#)

As a long-time fan of all things related to the Theory of Constraints, I was extremely pleased and honored to be able to join the early review program for [The Goal: A Business Graphic Novel](#). This book has been the foundation of so much that has driven manufacturing to new levels and then into any of a number of industries which have also benefitted from the concepts from the writings of Eli

Goldratt.



The format is very interesting because the book itself is a very character-driven story. The narrative comes across very well in the graphic novel format, so if you're a fan of this style of reading then The Goal in graphic format will definitely be one to add to your collection.



The next book I can definitely see going this way would be The Phoenix Project. The story of the Phoenix Project is a derivative of the style and teachings of The Goal with the focus on DevOps methodologies rather than manufacturing.

I can say that this was a great read and if you're looking for a book that adds a very interesting visual element to a profoundly important story of the Theory of Constraints in action, this is a must-read. It's a business book, a personal growth book, and if you look around our IT communities, it is effectively the story of our every day.



You can head on over to the North River Press site to read up on the book and get your copy ordered: <http://northriverpress.com/the-goal-a-business-graphic-novel/>

[Setting up Turbonomic Action Notifications to Slack Channels](#)

An interesting use-case that I've bumped into lately is where folks want to enable automation, but they also need to know when automated things happen. Email was the common platform for notifications, and still is, but there are many more organizations adoption Slack for day-to-day activity monitoring and building out interesting interactive ways to enable the ChatOps approach to IT operations management.

Since you may have followed along my first article which showed you how to set up a custom WebHook integration for your Slack team channel, we will take that one step further and show you how to configure Turbonomic to send notifications of actions to your Slack channel.

Setting up Action Scripts in Turbonomic

One of the cool features within Turbonomic is something called Action Scripts. These are scripts that are run when a particular actions happens on a particular entity within the environment. Action scripts run at different times in the process including before (PRE) and after (POST) the action so that you can either get notification or to trigger some interaction with the action.

Action Scripts run for every action time available including moves, scale/resize, and more. The naming of each Action Script is relative to the timing (PRE/POST) and the action type. You only need to create one Action Script which is hosted on your Turbonomic control instance and launched by

the Turbonomic engine as actions are triggered.

The [official documentation on using Action Scripts is here](#), but for our purposes here I will give you a crash course in creating a PRE move script so that we can send Slack notifications when an application workload is about to move.

Variables Accessible during Action Script Execution

There are a number of environment variables which are generated when a Turbonomic action is instantiated. Some of these include:

\$VMT_TARGET_NAME - the entity which is subject to the move action

\$VMT_CURRENT_NAME - the source location where the entity is located

\$VMT_NEW_NAME - the destination where the entity will be moved

\$VMT_ACTION_NAME - the unique ID for the action

These are the ones that I've chosen to include for my Slack notifications because I will want to know the workload which is subject to the move, the source location, target location, and then having the ID of the action is helpful for auditing and also for more deeper integration with a true ChatOps approach that we will dive into further in another post.

For now, the Slack notifications will be simply to log for us using our Slack channel whenever there are moves occurring. You can select from any of the different actions in the Action Scripts, so this is a good place to start.

The PRE_MOVE_VirtualMachine.sh Script

The simplest view of the script is as follows. Simply create a file named **PRE_MOVE_VirtualMachine.sh** which is the one that is called by a move action. This could be anything from a VM migration across hosts, clusters, or also container pod changes and more.

We need to leverage the action variables that we have been given and pass them into the our Slack API call. The simplest method for this is to inject a cURL command into the Action Script that will run using the native cURL command available on your Turbonomic instance.

The command to post to the API for Slack requires your WebHook URL [which you can get by following this guide that helps you get the WebHook set up](#).

This is the full GitHub Gist of the code. If you have existing Action Scripts in the folder, you can simply append these lines to your existing script.

Take note of the use of quotes within the command line as we need to pass the variables into the cURL command which requires additional double-quotes around the entire command.

Last step - Enable Action Script for Moves in Turbonomic

At the time of this writing, the Action Scripts features are still in the traditional flash UI. Go to the **Policy view** in your Turbonomic instance and expand the **Action | VM** section where we will enable the Action Scripts for Virtual Machines in this case.



Simply check off the **Action Script Settings** setting for the **PreMove** action and you are all set. In the image above we can see that I also have Move actions automated which may be set to Manual for your environment.

NOTE: Enabling policy changes within Turbonomic will trigger a refresh of the actions. This is because the state of your policies has changed and the entities in the environment must shop for the appropriate resources to satisfy their demand based on the newly formed policy configuration. This is the nature of the system being real-time so that no actions are held when they could be stale or unnecessary due to other environmental changes that have occurred.

The Slack View

Under your Slack channel, you will now begin seeing notifications whenever an action occurs. This is what your channel will start to look like as the moves take place:



In my case, I have enabled full automation; This means that these actions are triggered and the notification is done as the action is about to occur. We can also do POST_MOVE script which is handy if we are building out other hooks.

The goal with Action Scripts is to be able to integrate with any application lifecycle management process or product. Look for much more in the coming weeks as we walk through some more integrations that can be done with this method.

[Why Google Needs Consistency for Enterprise Cloud Customers](#)

Remember Google Buzz? Orkut? Wave? Reader? Google Talk? Then there was Google Picasa...which became photos...so far. There are sites dedicated to what we call the Google Graveyard. This doesn't even get into the Google Glass, Site Search, Search Appliance and others. I logged into my Google Analytics platform today and found it to be a completely different UI and UX than I have ever seen before...without warning. I used to use Google Hangouts On Air for the [Virtual Design Master event every year](#) until this year when HOA no longer works, so I have had to move to using Zoom and pushing to a Youtube Live Event.

The reason that I bring these up is that we have an optics problem with Google which may affect how many potential enterprise cloud customers choose to adopt, or rather to not adopt, Google Cloud Platform. One of the big things that traditional enterprise customers enjoy is the warm embrace of platforms that have consistency. Google has tended to have some challenges around product changes and the public face of those changes. Google most likely has lots of data backing the decision to shift or sunset a product.

Can GCP make Enterprises Greene with Envy?

Diane Greene has come over to Google by way of her most recent startup Bebop being acquired. It's

my opinion that the startup was the packaging in which they could acquire the real value, which is Diane herself. Diane has a proven past success in launching a little virtualization concept into the juggernaut that became VMware. The most recent Google Cloud Next event featured a strong presence of a new focus on the enterprise with an aim to become the number 1 public cloud provider within five years.

A quote that stood out from the event was “I actually think we have a huge advantage in our data centers, in our infrastructure, availability, security and how we automate things. We just haven’t packaged it up perfectly yet.” which highlights the challenge that Google will face. The need for many enterprises is a packaged and neatly consumable product that we know we can adopt and maintain with long support plans and clean deprecation.

There is little doubt of the ability of Google to develop incredible products which will give birth to next-generation application infrastructure that few can rival. The only doubt comes around whether enterprise audiences are going to be ready to adapt to the speed at which Google innovates their product set. If Kubernetes is any sign of how well we are leaning in, then it is very easy to see that Google can take the market on and win a significant share.

Google Cloud Platform will be a juggernaut in the public cloud realm. That is a fact which is being proven out by some major customers moving into the platform already and many more dabbling. Multi-cloud is the new cloud, so GCP will inevitably become a key player in that strategy because of its underlying GKE product to support Kubernetes workloads. In my opinion, the multi-cloud approach enabled by containerized workloads with an enterprise-grade scheduler is going to become the goal we should strive for.

The only question is how long it will take before we can all put our trust in one product that Google has lacked in, which is consistency.