

FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

Locked Out of a Windows VM After a Cloud-Admin Restart — Cloudbase-Init and the Password Nobody Set

A presales war story about an Administrator password that changed with "SYSTEM" as the subject, the OpenStack service almost nobody remembers is still installed, and the one log line I now read before anyone says "we were hacked".

Daanial Farrukh · 30 October 2026 · <https://quabin.com/windows-vm-locked-out-cloudbase-init-random-password/>

The call started with two groups of people certain they had done nothing.

"We did not touch the password. We did not touch the network card. Your team did."

That was the customer. Their application server, a Windows VM on a private OpenStack cloud, had been shut down the previous evening by their own administrator, from the console. The cloud team started it again a few minutes later. When the customer tried to log in the next morning, the Administrator password was wrong. The machine name had changed. The network adapter was disabled.

"We only pressed Start. Nothing else," said the cloud team.

I asked for one thing before anyone said the word "breach".

"Can you export the Security log from the VM?"

"It says SYSTEM did it. So it was not a person. So it was the platform."

"SYSTEM is not a person. SYSTEM is a service. Which service?"

Silence.

Forty minutes later I had the answer, and it was neither group. It was a service that had been sitting on that VM since the day the image was built, doing nothing for months, waiting for a reason to run again.

The Trap, Named

That's the trap. When a Windows event says the subject was **SYSTEM**, people read it as "no human was involved" and stop there. Some read it as proof of a hack. Some read it as proof the hypervisor did something. Almost nobody asks the next question: which process, running as SYSTEM, made that change at that second.

A SYSTEM-subject password reset is not the end of the investigation. It is the first clue about which service to go and read.

What Cloudbase-Init Actually Is

Here's the part people miss. Every Windows image built for OpenStack carries a small service called **Cloudbase-Init**. It is the Windows equivalent of cloud-init on Linux. On the first boot of a new instance it reads metadata from the cloud and does the things that make a fresh VM usable:

Cloud metadata → Cloudbase-Init service → set hostname → create user → set password → extend disk → run user-data → reboot

Each of those is a plugin. The ones that matter in this story:

- **SetHostNamePlugin** renames Windows to the instance name the cloud knows, then schedules a reboot.
- **CreateUserPlugin** makes sure the admin account exists.
- **SetUserPasswordPlugin** sets that account's password. If the cloud passed one in metadata, it uses it. If not, it generates a random one.

The design is correct for a brand-new instance. The random password is meant to be encrypted with the SSH public key the VM was launched with and posted back to the metadata service, so the owner can retrieve it from the cloud. If there is no key, there is nothing to encrypt with, and Cloudbase-Init says so in its log and moves on.

Cloudbase-Init remembers which plugins have already run. It keeps a per-instance status in the Windows registry, under HKLM\SOFTWARE\Cloudbase Solutions\Cloudbase-Init\<instance-id>\Plugins. On every later boot the service starts, checks the status, and for each plugin logs "execution already done, skipping". That is the normal life of the service: start, skip everything, stop. For months.

Until one boot where it does not skip.

Why Nobody Sees It

Three reasons.

First, the service is invisible by design. It runs for a few seconds at boot, writes to a log under C:\Program Files\Cloudbase Solutions\, and exits. Nobody opens that folder on a server that has been in production for a year.

Second, the symptom looks exactly like an intruder. A password reset you did not do. A hostname you did not choose. A network card you did not disable. If you are the customer, that is a compromise. If you are the cloud team, that is the customer forgetting what they did. Both readings are emotionally satisfying and both are wrong.

Third, the Security log is honest but incomplete. Event 4724, "An attempt was made to reset an account's password", records the subject as SYSTEM. It does not record the process. The one log that names the process is the one nobody reads.

Absence of a human in the event log is not absence of a cause. It is a pointer to a service.

How to Actually Prove It

Everything here is read-only. Run it from an elevated PowerShell on the affected VM, or against an exported log on another machine.

Step 1: Find the password resets and the exact second they happened.

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4724} -MaxEvents 20 | Format-List  
TimeCreated, Message
```

In this case there were two 4724 events ten seconds apart, both with Subject Account Name SYSTEM, both targeting Administrator. Note the timestamps to the second. You will match them in the next step.

Also pull 4723, "An attempt was made to change an account's password". That is the event you get when a logged-in user changes a password themselves. If you see 4723 with a named user, a person was at the keyboard. In this case there was one 4723, the next evening, from an interactive Administrator session, and that one was the customer's own recovery attempt. Keep the two event IDs apart in your head: **4724 is a reset done to an account; 4723 is a change done by the account.**

Step 2: Find out who rebooted the machine and why.

```
Get-WinEvent -FilterHashtable @{LogName='System'; Id=1074} -MaxEvents 10 | Format-List  
TimeCreated, Message
```

Event 1074 names the process and the reason for every shutdown and restart. Here it showed two entries in the window. The first was the customer's own shutdown from the console the previous evening. The second, a minute after the cloud team pressed Start, read "**Cloudbase-Init reboot**" in the reason field. That one line is the whole case. A service called Cloudbase-Init rebooted the server, and it only reboots after it has renamed the host.

Step 3: Read the Cloudbase-Init log for the same seconds.

```
Select-String -Path "C:\Program Files\Cloudbase Solutions\Cloudbase-Init\log\cloudbase-init.  
log" -Pattern "Generating a random user password|Password succesfully updated|No SSH public  
key|already done, skipping" | Select-Object -Last 40
```

Yes, "succesfully" is spelled with one s in the log. Search for the misspelling or you will not find it.

What you want to see is the contrast. In every earlier boot, line after line of:

```
[ - ] Plugin 'cloudbaseinit.plugins.common.setuserpassword.SetUserPasswordPlugin' execution  
already done, skipping
```

And then, in the boot after the restart, at the exact second of the first 4724 event:

```
setuserpassword [ - ] Generating a random user password  
setuserpassword [ - ] No SSH public key available for password encryption
```

Read it in words. The service decided to run its first-boot plugins again. It created or confirmed the admin user, generated a random password, found no key to encrypt it with, and set it anyway. The password now exists in exactly one place: the account itself. Nobody can retrieve it. Nobody did anything wrong at a keyboard. The machine is locked.

When the timestamp of "Generating a random user password" matches the timestamp of the 4724 event with subject SYSTEM, the cause is proven. Not a theory. Proven.

Step 4: Confirm the rename came from the same service.

```
Select-String -Path "C:\Program Files\Cloudbase Solutions\Cloudbase-Init\log\cloudbase-init.log" -Pattern "sethostname|hostname" | Select-Object -Last 10
```

The hostname plugin logs the new name it applied and the reboot it requested. Match it to the 1074 "Cloudbase-Init reboot" event. Now you have one author for two of the three symptoms.

The Part I Could Not Prove

I want to be honest here, because this is where it is tempting to make the story tidier than it was.

I proved the password reset. I proved the rename and the reboot. I did **not** prove why the plugins ran again after months of skipping, and I did not prove that the disabled network adapter came from the same service.

The documented mechanism is the per-instance plugin status in the registry. The status lives under a key named after the instance ID the service reads from metadata. If that key is not found, every plugin is new and runs. That can happen when the instance ID the service sees changes, or when the status key is gone. On this VM I could see that the plugins re-ran; I could not get the metadata history to say which of those it was. So in the report I wrote exactly that: cause of lockout proven, trigger of the re-run not proven, network card not attributed.

A report that says "this part is proven and this part is not" is worth more than one that sounds complete. The customer trusted the proven half because I did not stretch it.

How to Actually Fix It

Two jobs. Get back in, then make sure a restart can never do this again.

Step 1: Get back in. This changes state.

If the VM was launched with an SSH key pair and the cloud still has it, the random password is retrievable from the cloud side. On the OpenStack CLI:

```
nova get-password <server> <private-key-file>
```

If, like here, there was no key, the cloud team resets the password from the platform instead. On OpenStack that is a password change through the compute service or the console. Any reset done this way is a new 4724 event. In our log, the following morning's resets were exactly that, and they did not appear in the Cloudbase-Init log at all, which is how you tell an admin's reset from the service's reset.

Step 2: Decide what Cloudbase-Init is allowed to do on a server that is already in production. This changes state.

Open the config file:

```
notepad "C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf"
```

Find the plugins= line. It is a comma-separated list of every plugin the service may run. On a VM that has been handed over to a customer, there is no reason for the service to ever rename the host or set

the admin password again. Remove SetHostNamePlugin, CreateUserPlugin and SetUserPasswordPlugin from the list and save. A restart is not required; the file is read at the next service start.

Also check two settings that control the password behaviour:

```
Select-String -Path "C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf" -Pattern "first_logon_behaviour|inject_user_password|plugins"
```

inject_user_password is what posts the encrypted password back to the cloud; it only helps if the VM has a key. first_logon_behaviour decides whether the user is forced to change the password at first logon. Neither of them stops the plugin from running. Only the plugins= list does that.

If the customer never wants cloud-side changes on this VM again, the stronger option is to disable the service entirely:

```
Set-Service cloudbase-init -StartupType Disabled
```

I prefer trimming the plugin list over disabling the service, because the service also handles useful things like extending the disk after a resize. But both are legitimate. The point is to make a decision, not to leave the default.

Step 3: Keep a break-glass account. This changes state.

Create a second local administrator with a password stored in the customer's vault, not on the VM. Cloudbase-Init only manages the account named in its username= setting. A second account is untouched by anything the service does. On this VM that one account would have turned a two-day outage into a two-minute login.

Step 4: Verify. This step is where most tutorials stop and where mine starts.

Restart the VM once, on purpose, in a maintenance window. Then re-run the Step 1 and Step 3 checks from the proof section:

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4724} -MaxEvents 5 | Format-List  
TimeCreated, Message
```

There must be **no** new 4724 event at boot time. Then:

```
Select-String -Path "C:\Program Files\Cloudbase Solutions\Cloudbase-Init\log\cloudbase-init.log" -Pattern "Generating a random user password" | Select-Object -Last 3
```

The last match must be the old one from the incident, not a fresh one from your test boot. If a new one appears, the plugin is still in the list, or the service is reading a different config file than the one you edited, and you go back to Step 2.

Plain takeaway: a first-boot service you have not told to stop is a first-boot service that is waiting.

The Bigger Lesson — Four Questions Before Anyone Says "Breach"

Whenever a Windows VM in a cloud comes back from a restart with something changed that "nobody did", run these in order.

- 1 **What did the 1074 events say?** The reason field often names the process. "Cloudbase-Init reboot" ends the argument in one line.
- 1 **Who is the subject on the 4724 events?** A named user means a person. SYSTEM means a service, and your next job is to find which one.
- 1 **Does the Cloudbase-Init log have activity at the same second?** If yes, the service is the author. If no, the reset came from the hypervisor side or from a real intruder, and you escalate properly.
- 1 **Is the first-boot service still allowed to run its first-boot plugins?** Check plugins= in the config. On any VM older than its first week, the answer should be no for hostname and password.

If the answers point at Cloudbase-Init, the fix is a config edit and a break-glass account. If they do not, you have a genuine incident and you have reached it in twenty minutes instead of two days.

What I Tell Every Customer Now

In the next review, ask the team one question:

"On your Windows cloud VMs, which Cloudbase-Init plugins are still enabled, and when did you last read its log?"

If the answer is "we trimmed the list at handover and the log shows nothing but skips", they are in good shape. If the answer is "what is Cloudbase-Init?", we are not done yet.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

Why did my Windows VM's Administrator password change after a restart on OpenStack?

Cloudbase-Init, the first-boot service installed in every OpenStack Windows image, re-ran its SetUserPasswordPlugin and generated a random password. The Security log shows this as event 4724 with subject SYSTEM. Confirm it by finding "Generating a random user password" in cloudbase-init.log at the same timestamp.

What does "No SSH public key available for password encryption" mean in cloudbase-init.log?

Cloudbase-Init wanted to encrypt the new random password with the key pair the VM was launched with, so the owner could retrieve it from the cloud. The VM had no key pair, so the password was set but never stored anywhere retrievable. That line is why the account is locked rather than just changed.

What is the difference between Windows event 4724 and 4723?

Event 4724 is a password reset performed on an account by someone or something else, such as an administrator or a service. Event 4723 is a password change made by the account itself from a logged-in session. A 4724 with subject SYSTEM points at a service; a 4723 with a user name points at a person at

the keyboard.

How do I stop Cloudbase-Init from changing the hostname or password again?

Edit cloudbase-init.conf under C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf and remove SetHostNamePlugin, CreateUserPlugin and SetUserPasswordPlugin from the plugins= list. Alternatively disable the cloudbase-init service. Then restart once in a maintenance window and confirm no new 4724 event appears at boot.

Can I recover the random password Cloudbase-Init generated?

Only if the VM was launched with an SSH key pair and inject_user_password was enabled; then the encrypted password is in the metadata service and nova get-password decrypts it with the private key. Without a key pair the password cannot be recovered, and the cloud team must reset it from the platform.

Is a SYSTEM-subject password reset proof that the server was hacked?

No. SYSTEM means a process running as the local system account made the change, which includes legitimate services like Cloudbase-Init. Match the event's timestamp against service logs before treating it as an intrusion. If no service accounts for it, then escalate as a security incident.

Read online: <https://quabin.com/windows-vm-locked-out-cloudbase-init-random-password/>