Step 1 – Cluster starts in Default (best effort)

Cluster (RF2, so 1 host failure for Guarantee):

- 3 AHV hosts, each 256 GB RAM → 768 GB total.
- HA mode: Default (Best Effort).

You deploy VMs until the cluster is effectively full:

- Total VM vRAM in use: 700 GB.
- There is no reserved HA capacity; all 3 hosts are heavily utilized.

In this state:

- A host failure would require ~350 GB of free capacity across the remaining 2 hosts, but you have almost none.
- Result: many VMs will *not* restart if a node fails, but Default mode allows you to run this way.

Step 2 – You switch to Guarantee on an overfilled cluster

Now you enable Guarantee in Prism or via aCLI, with reservation type = ReserveSegments and 1 host failure to tolerate.

Guarantee math (simplified):

- Total RAM: 768 GB.
- Must reserve the equivalent of 1 host = 256 GB.
- Effective HA-safe capacity for powered-on, *protected* VMs ≈ 768 − 256 = 512 GB.

But you already have 700 GB powered on. So:

- Current usage (700 GB) > allowed HA-safe capacity (512 GB).
- The cluster becomes "HA overcommitted": HA is enabled, but it cannot actually reserve full failover capacity.

What Nutanix does not do:

- It does not automatically power off VMs.
- It does not shuffle VMs to magically create 256 GB free.
- All existing VMs keep running, still effectively unprotected.

What changes immediately:

- HA state shows that 1 host failure is configured, but remaining host failures to tolerate might drop to 0.

- You may see warnings/alerts indicating that not enough resources are available to satisfy the HA reservation.

Step 3 – Behavior for new operations with Guarantee on

After Guarantee is enabled in this overfilled state:

- Existing powered-on VMs continue to run.
- New actions become constrained by admission control:
  - Powering on additional VMs will fail once they would push the "protected" usage beyond 512 GB.
  - Increasing vRAM on existing VMs can fail for the same reason.

If a host fails now:

- AHV still attempts to restart VMs best-effort on remaining hosts.
- Because real free memory is insufficient, some VMs will stay down despite HA being configured in Guarantee mode.

This is a good talking point: "Guarantee doesn't retroactively fix an oversubscribed environment; it just prevents you from making it *worse* going forward."

Step 4 – Make Guarantee actually work (with ha_priority)

To convert this into a truly protected cluster, you have two levers: reduce protected load or add capacity.

Option A – Use ha_priority to exclude low-importance VMs

Let's say out of the 700 GB:

- 400 GB = important prod VMs you want protected.
- 300 GB = test/dev VMs you can treat as non-HA.

You mark test/dev VMs as HA-disabled:

```
bash

nutanix@CVM$ acli vm.update dev-vm-1 ha_priority=-1
nutanix@CVM$ acli vm.update dev-vm-2 ha_priority=-1
# ... and so on for all test/dev VMs
```

Effect on the math:

- Total powered-on RAM is still 700 GB.
- But HA only counts 400 GB (prod) toward its reservation calculation; the 300 GB dev is ignored for HA capacity.
- Required reservation is still 256 GB.
- HA-safe capacity for *protected* VMs: 512 GB.
- Protected usage = 400 GB < 512 GB → now the cluster can genuinely tolerate 1 host failure for those protected VMs.

What happens on host failure now:

- HA attempts to restart *protected* VMs first; they all fit within remaining capacity.
- Test/dev (ha_priority = –1) VMs:
    - Are not guaranteed to restart; they may stay down if there's no room.

This gives you a clean slide story:

"Before: 700 GB of VMs, no spare; enabling Guarantee alone gives a false sense of safety.
After: Mark 300 GB of test/dev as non-HA; HA only promises the 400 GB prod set, which can now be fully restarted after one host failure."

Option B – Add capacity instead

Alternatively, if you don't want to mark any VMs as non-HA:

- Add a 4th host with 256 GB RAM → total 1024 GB.
- Reservation for 1 host failure is 256 GB.
- HA-safe capacity = 1024 – 256 = 768 GB.
- Your current usage (700 GB) now fits within the HA-safe envelope, so all VMs can be protected without changing ha_priority.

Would you like a second example that shows the same idea but for RF3/FT2 with 2 host failures and slightly different numbers for another slide?