

Harvester supports global configuration of resource overload percentages on CPU, memory, and storage. By setting [overcommit-config](#), this will allow scheduling of additional virtual machines even when physical resources are fully utilized.

Harvester allows you to overcommit CPU and RAM on compute nodes. This allows you to increase the number of instances running on your cloud at the cost of reducing the performance of the instances. The Compute service uses the following ratios by default:

- CPU allocation ratio: 1600%
- RAM allocation ratio: 150%
- Storage allocation ratio: 200%

:::note

Classic memory overcommitment or memory ballooning is not yet supported by this feature. In other words, memory used by a virtual machine instance cannot be returned once allocated.

:::

Configure the global setting [overcommit-config](#)

Users can modify the global `overcommit-config` by following the steps below, and it will be applied to each newly created virtual machine after the change.

1. Go to the **Advanced > Settings** page.

 overcommit page

1. Find the `overcommit-config` setting.
2. Configure the desired CPU, Memory, and Storage ratio.

 overcommit panel

Configure overcommit for a single virtual machine

In situations where you require specific configurations for individual virtual machines without affecting the global settings, you can easily achieve this by modifying the `spec.template.spec.domain.resources.limits.<memory|cpu>` value on the corresponding virtual machine spec directly.

 vm overcommit config

Reserve more memory for the system overhead

By default, the Harvester reserves a certain amount of system management overhead memory from the memory allocated for the virtual machine. In most cases, this will not cause any problems. However, some operating systems, such as Windows 2022, will request more memory than is reserved.

To address the issue, Harvester provides an annotation `harvesterhci.io/reservedMemory` on `VirtualMachine` custom resource to let you specify the amount of memory to reserve. For instance, add

`harvesterhci.io/reservedMemory: 200Mi` if you decide to reserve 200 MiB for the system overhead of the VM.

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  annotations:
+   harvesterhci.io/reservedMemory: 200Mi
    kubevirt.io/latest-observed-api-version: v1
    kubevirt.io/storage-observed-api-version: v1alpha3
    network.harvesterhci.io/ips: '[]'
  ...
  ...
```

Why my virtual machines are scheduled unevenly?

The scheduling of virtual machines depends on the underlying behavior of the kube-scheduler. We have a dedicated article explaining the details. If you would like to learn more, check out: [Harvester Knowledge Base: VM Scheduling](#).