# Node stats requests

<table>
<thead>
<tr>
<th>Method</th>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/v1/nodes/stats</td>
<td>Get stats for all nodes</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/nodes/stats/{uid}</td>
<td>Get stats for a single node</td>
</tr>
</tbody>
</table>

## Get all nodes stats

**GET /v1/nodes/stats**

Get statistics for all nodes.

## Required permissions

**Permission name**

view_all_nodes_stats

## Request

**Example HTTP request**

GET /nodes/stats?interval=1hour&stime=2014-08-28T10:00:00Z

## Request headers

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>cnm.cluster.fqdn</td>
<td>Domain name</td>
</tr>
<tr>
<td>Accept</td>
<td>application/json</td>
<td>Accepted media type</td>
</tr>
</tbody>
</table>

## Query parameters

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interval</td>
<td>string</td>
<td>Time interval for which we want stats: 1sec/10sec/5min/15min/1hour/12hour/1week (optional)</td>
</tr>
<tr>
<td>stime</td>
<td>ISO_8601</td>
<td>Start time from which we want the stats. Should comply with the <a href="https://en.wikipedia.org/wiki/ISO_8601">ISO_8601</a> format (optional)</td>
</tr>
<tr>
<td>etime</td>
<td>ISO_8601</td>
<td>End time after which we don’t want the stats. Should comply with the <a href="https://en.wikipedia.org/wiki/ISO_8601">ISO_8601</a> format (optional)</td>
</tr>
</tbody>
</table>

## Response
Returns a JSON array of statistics for all nodes.

Example JSON body

```json
[
  {
    "uid": "1",
    "intervals": [
      {
        "interval": "1sec",
        "stime": "2015-05-28T08:40:11Z",
        "etime": "2015-05-28T08:40:12Z",
        "conns": 0.0,
        "cpu_idle": 0.5499999999883585,
        "cpu_system": 0.03499999999985448,
        "cpu_user": 0.38000000000101863,
        "egress_bytes": 0.0,
        "ephemeral_storage_avail": 2929315840.0,
        "ephemeral_storage_free": 3977830400.0,
        "free_memory": 893485056.0,
        "ingress_bytes": 0.0,
        "persistent_storage_avail": 2929315840.0,
        "persistent_storage_free": 3977830400.0,
        "total_req": 0.0
      },
      {
        "interval": "1sec",
        "stime": "2015-05-28T08:40:12Z",
        "etime": "2015-05-28T08:40:13Z",
        "cpu_idle": 1.2,
        "// additional fields...
      }
    ]
  },
  {
    "uid": "2",
    "intervals": [
      {
        "interval": "1sec",
        "stime": "2015-05-28T08:40:11Z",
        "etime": "2015-05-28T08:40:12Z",
        "conns": 0.0,
        "cpu_idle": 0.5499999999883585,
        "cpu_system": 0.03499999999985448,
        "cpu_user": 0.38000000000101863,
        "egress_bytes": 0.0,
        "ephemeral_storage_avail": 2929315840.0,
        "ephemeral_storage_free": 3977830400.0,
        "free_memory": 893485056.0,
        "ingress_bytes": 0.0,
        "persistent_storage_avail": 2929315840.0,
        "persistent_storage_free": 3977830400.0,
        "total_req": 0.0
      }
    ]
  }
]
```
{ "cpu_idle": 1.2, "// additional fields..." }
}

### Status codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>No error</td>
</tr>
<tr>
<td>404 Not Found</td>
<td>No nodes exist</td>
</tr>
</tbody>
</table>

### Get node stats

GET /v1/nodes/stats/{int: uid}

Get statistics for a node.

### Required permissions

**Permission name**

view_node_stats

### Request

**Example HTTP request**

GET /nodes/stats/1?interval=1hour&stime=2014-08-28T10:00:00Z

### Request headers

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>cnm.cluster.fqdn</td>
<td>Domain name</td>
</tr>
<tr>
<td>Accept</td>
<td>application/json</td>
<td>Accepted media type</td>
</tr>
</tbody>
</table>

### URL parameters

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>integer</td>
<td>The unique ID of the node requested.</td>
</tr>
</tbody>
</table>

### Query parameters

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
Time interval for which we want stats:
1sec/10sec/5min/15min/1hour/12hour/1week
(optional)

Start time from which we want the stats.
Should comply with the ISO_8601 format
(optional)

End time after which we don’t want the stats.
Should comply with the ISO_8601 format (optional)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interval</td>
<td>string</td>
<td>Time interval for which we want stats: 1sec/10sec/5min/15min/1hour/12hour/1week (optional)</td>
</tr>
<tr>
<td>stime</td>
<td>ISO_8601</td>
<td>Start time from which we want the stats. Should comply with the ISO_8601 format (optional)</td>
</tr>
<tr>
<td>etime</td>
<td>ISO_8601</td>
<td>End time after which we don’t want the stats. Should comply with the ISO_8601 format (optional)</td>
</tr>
</tbody>
</table>

Response

Returns statistics for the specified node.

Example JSON body

```json
{
  "uid": "1",
  "intervals": [
    {
      "interval": "1sec",
      "stime": "2015-05-28T08:40:11Z",
      "etime": "2015-05-28T08:40:12Z",
      "conns": 0.0,
      "cpu_idle": 0.5499999999883585,
      "cpu_system": 0.0349999999985448,
      "cpu_user": 0.38000000000101863,
      "egress_bytes": 0.0,
      "ephemeral_storage_avail": 2929315840.0,
      "ephemeral_storage_free": 3977830400.0,
      "free_memory": 893485056.0,
      "ingress_bytes": 0.0,
      "persistent_storage_avail": 2929315840.0,
      "persistent_storage_free": 3977830400.0,
      "total_req": 0.0
    },
    {
      "interval": "1sec",
      "stime": "2015-05-28T08:40:12Z",
      "etime": "2015-05-28T08:40:13Z",
      "cpu_idle": 1.2,
      "// additional fields..."
    }
  ]
}
```

Status codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>No error</td>
</tr>
<tr>
<td>404 Not Found</td>
<td>Node does not exist</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>406 Not Acceptable</td>
<td>Node isn't currently active</td>
</tr>
<tr>
<td>503 Service Unavailable</td>
<td>Node is in recovery state</td>
</tr>
</tbody>
</table>

**Updated:** November 29, 2021