|  |  |
| --- | --- |
| **capellaStreamConfig.yaml** | **Explanation** |
| # workersUseGPU: allow workers to use nvidia GPU  **workersUseGPU:** false | If working with GPUs / NVENC workflows, set this value to **true**. Otherwise, leave as **false**. |
| # nbGPUs: how many GPUs are on the nodes. Only has an effect if workersUseGPU is set  **nbGPUs:** 1 | If **workersUseGPU** is set to **true**, set this to the number of GPUs that are available on the worker nodes.  **Note:** your Cambria Stream license must include support for multiple GPUs |
| # managerInstanceType: database and Cambria StreamManager will be restricted  **managerInstanceType:** "g6-dedicated-4"  … | The instance type for the Cambria Stream Manager machines. Change this to the instance type that you chose when creating the Cambria Stream Manager nodes (Eg. g6-dedicated-4) |
| # workerInstanceType: instance type for the worker nodes...  **workerInstanceType:** "g6-dedicated-8"  … | The instance type for the Cambria Stream nodes. Change this to the instance type that you chose when creating the Cambria Stream nodes / that you plan to use for your Cambria Stream nodes (Eg. g6-dedicated-8) |
| # maxWorkerInstances: maximum number of worker instances (ie replicas).  **maxWorkerInstances:** 20 | The max number of Cambria Stream machines that can be spawned. By default, this is 20, but we recommend starting out with 2 for testing. |
| # workerRolloutStep1Weight: when upgrading Cambria Stream version  **workerRolloutStep1Weight:** 10  # workerRolloutStep1Weight: when upgrading Cambria Stream version  **workerRolloutStep2Weight:** 40 | For upgrades, Capella uses argo-rollouts to incrementally upgrade nodes. The first value is the percentage of all the nodes that should be upgraded first. The second value is the percentage of all the nodes that should be upgraded second. See the **Upgrading Kubernetes Cluster** section for more information. |
| #workerRolloutMaxUnavailable: during argo-rollout Cambria Stream upgrade…  **workerRolloutMaxUnavailable:** 1 | For upgrades, argo-rollouts will use this value to determine how many Cambria Stream worker pods can be offline at a time during an upgrade. It is recommended to set this value to:  [ maxWorkerInstances ] - [ min # of active nodes ] + 1  Eg.  maxWorkerInstances = 20  Min # of nodes active = 2  workerRolloutMaxUnavailable = 19 |
| # pgInstances: number of postgresql database instances (ie replicas)  **pgInstances:** 3  # managerReplicas: number of Cambria StreamManager instances (ie replicas)  **managerReplicas:** 3  … | The number of instances for Cambria Stream Manager and postgres database. These two values must match each other and also the amount of nodes for Stream Manager created in step 1. In this case, the value will be 3. |
| # routingInputPortBegin: the beginning input port for Akamai Cloud Routing.  **routingInputPortBegin:** 1935  # routingInputPortCount: the number of ports for Akamai Cloud Routing.  **routingInputPortCount:** 3 | These are settings for routing RTMP streams to Cambria Stream instances. |
| # maximum allowed instances for each new machine. Default: -1 (up to the license limit)  **defaultMachineMaximumInstances:** -1 | This option allows users to configure the maximum number of channels a Cambria Stream node / machine can run (up to the amount in the Cambria license). |
| **externalAccess:**  # exposeStreamServiceExternally: main Cambria Stream service  **exposeStreamServiceExternally:** true | Set to **true** if you want to be able to access Cambria Stream Manager externally. Otherwise, set to **false** |
| # enableIngress: enable ngingx ingress  **enableIngress:** true  # hostName: use for the nginx ingress. Replace this with your domain name.  **hostName:** myhost.com  # acmeRegistrationEmail: email for Automated Certificate Management  **acmeRegistrationEmail:** test@example.com  # acmeServer: server to get TLS certificate from  **acmeServer:** https://acme-staging-v02.api.letsencrypt.org/directory | These fields allow you to have a self-signed certificate and use a purchased domain name for your Kubernetes Cluster. If you don’t plan to use this feature, you can either set **enableIngress** to **false** or leave the fields as is. If planning to use this feature, follow these steps: <https://www.dropbox.com/scl/fi/p6i363hz5n0s9uney1zxf/Cambria_Kubernetes_Domain_DNS_Guide.pdf?rlkey=k8zjj0zqdwvsjtp3yozr7uzh8&st=2xuzj2qj&dl=1> |
| externalStreamIPs:  # limitExternalAccess: Firewall, limit which external IPs can send streams…  **limitExternalAccess:** false  # allowedExternalStreamIPs: list of allowed external IPs which can send streams…  # note that "10.0.0.0/8" and "192.168.255.0/24" must exist…  **allowedExternalStreamIPs:**  - 10.0.0.0/8  - 192.168.255.0/24  - ... | This setting is for controlling what ip addresses can send streams to the Cambria Stream instances running on Kubernetes. By default, this setting is set to **false** to indicate that there are no ip restrictions. Set this value to **true** if you want to restrict to certain ip addresses. Add the ip addresses to allow in the list of **allowedExternalStreamIPs**. |
| secrets:  # pgClusterPassword: password for the postgresql database  **pgClusterPassword:** "xrtVeQ4nN82SSiYHoswqdURZ…" | The password for the postgres database. It is recommended to change the default value to something more secure. |
| # cpLicenseKey: license key  **cpLicenseKey:** "XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX" | The Cambria Stream product license. The Capella team should have provided this value for you. Replace the “XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX” with the corresponding license key. |
| # cpLicenseKeyManager: license key for the manager nodes  **cpLicenseKeyManager:** "YYYYYY-YYYYYY-YYYYYY-YYYYYY-YYYYYY-YYYYYY" | The Cambria Stream Manager product license. The Capella team should have provided this value for you. Replace the “YYYYYY-YYYYYY-YYYYYY-YYYYYY-YYYYYY-YYYYYY” with the corresponding license key. |
| # cambriaStreamManagerAPIToken: API token for Cambria StreamManager  **cambriaStreamManagerAPIToken:** "12345678-1234-43f8-b4fc-53afd3893d5f" | This is a token that is needed to make API calls to Capella’s Stream Manager web server. Change this to any string value. |
| # cambriaStreamManagerWebUIUser: user/password to access Cambria Stream...  **cambriaStreamManagerWebUIUser:** "admin,defaultWebUIUser,RZvSSd3ffsElsCEEe9" | This is the login credentials for Cambria Stream Manager’s Web UI. Each user is  listed in the form:  role,username,password  Allowed roles:  **admin** - can view/create/edit/delete anything on the WebUI. Can also create/manage WebUI users.  **superuser** - can view/create/edit/delete anything on the WebUI.  **user** - can only view anything on the WebUI.  For multiple users, separate each by a comma. Example:  admin,admin,changethispassword1234,user,guest,password123 |
| # akamaiCloudAPIToken: Akamai Cloud API token, used for horizontal scaling...  **akamaiCloudAPIToken:** "d02732530a2bcfd4d028425eb55f366f746317eb7a8054…" | This is your Akamai Cloud account’s API Token. See <https://www.linode.com/docs/products/tools/api/guides/manage-api-tokens/> |