

Bright Cluster Manager 8.0

Developer Manual

Revision: f4bb9ed

Date: Mon Dec 10 2018



©2017 Bright Computing, Inc. All Rights Reserved. This manual or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Bright Computing, Inc.

Trademarks

Linux is a registered trademark of Linus Torvalds. PathScale is a registered trademark of Cray, Inc. Red Hat and all Red Hat-based trademarks are trademarks or registered trademarks of Red Hat, Inc. SUSE is a registered trademark of Novell, Inc. PGI is a registered trademark of NVIDIA Corporation. FLEXlm is a registered trademark of Flexera Software, Inc. ScaleMP is a registered trademark of ScaleMP, Inc. All other trademarks are the property of their respective owners.

Rights and Restrictions

All statements, specifications, recommendations, and technical information contained herein are current or planned as of the date of publication of this document. They are reliable as of the time of this writing and are presented without warranty of any kind, expressed or implied. Bright Computing, Inc. shall not be liable for technical or editorial errors or omissions which may occur in this document. Bright Computing, Inc. shall not be liable for any damages resulting from the use of this document.

Limitation of Liability and Damages Pertaining to Bright Computing, Inc.

The Bright Cluster Manager product principally consists of free software that is licensed by the Linux authors free of charge. Bright Computing, Inc. shall have no liability nor will Bright Computing, Inc. provide any warranty for the Bright Cluster Manager to the extent that is permitted by law. Unless confirmed in writing, the Linux authors and/or third parties provide the program as is without any warranty, either expressed or implied, including, but not limited to, marketability or suitability for a specific purpose. The user of the Bright Cluster Manager product shall accept the full risk for the quality or performance of the product. Should the product malfunction, the costs for repair, service, or correction will be borne by the user of the Bright Cluster Manager product. No copyright owner or third party who has modified or distributed the program as permitted in this license shall be held liable for damages, including general or specific damages, damages caused by side effects or consequential damages, resulting from the use of the program or the un-usability of the program (including, but not limited to, loss of data, incorrect processing of data, losses that must be borne by you or others, or the inability of the program to work together with any other program), even if a copyright owner or third party had been advised about the possibility of such damages unless such copyright owner or third party has signed a writing to the contrary.

Table of Contents

Table of Contents	i
0.1 About This Manual	xxi
0.2 About The Manuals In General	xxi
0.3 Getting Administrator-Level Support	xxii
0.4 Getting Developer-Level Support	xxii
0.5 Getting Professional Services	xxii
1 Bright Cluster Manager Python API	1
1.1 Installation	1
1.1.1 Linux Clients	1
1.2 Examples	1
1.2.1 First Program	2
1.3 Methods And Properties	3
1.3.1 Viewing All Properties And Methods	3
1.3.2 Property Lists	3
1.3.3 Creating New Objects	3
1.3.4 List Of Objects	4
1.3.5 Useful Methods	6
1.3.6 Useful Example Program	7
1.4 The Workload Management API	8
1.4.1 Job Submission	8
1.4.2 Job Information And Management	11
1.4.3 Queue Information And Management	12
2 Metric Collections	15
2.1 Metric Collections Added Using <code>cmsh</code>	15
2.2 Metric Collections Initialization	15
2.3 Metric Collections Output During Regular Use	16
2.4 Metric Collections Error Handling	17
2.5 Metric Collections Consolidator Syntax	17
2.6 Metric Collections Environment Variables	18
2.7 Metric Collections Examples	20
2.8 Metric Collections On iDataPlex And Similar Units	20
3 Bright Cluster Manager JSON API	23
3.1 Services	23
3.1.1 <code>auth</code>	23
3.1.2 <code>ceph</code>	23
3.1.3 <code>cert</code>	23
3.1.4 <code>cloud</code>	23
3.1.5 <code>device</code>	23

3.1.6	etcd	23
3.1.7	gui	23
3.1.8	hadoop	23
3.1.9	job	23
3.1.10	keyvalue	23
3.1.11	kube	23
3.1.12	lustre	23
3.1.13	main	23
3.1.14	mesos	23
3.1.15	mon	23
3.1.16	net	23
3.1.17	openstack	23
3.1.18	part	23
3.1.19	proc	23
3.1.20	prov	23
3.1.21	puppet	23
3.1.22	serv	23
3.1.23	session	23
3.1.24	sync	24
3.1.25	test	24
3.1.26	ticket	24
3.1.27	user	24
3.1.28	zookeeper	24
3.2	Entities	24
3.2.1	AzureDataDisk	24
3.2.2	AzureDisk	24
3.2.3	AzureIntermediateStorage	24
3.2.4	AzureLocation	24
3.2.5	AzureManagedDiskParameters	24
3.2.6	AzureOSDisk	24
3.2.7	AzureProvider	24
3.2.8	AzurePublicIP	24
3.2.9	AzureSettings	24
3.2.10	AzureVMSize	24
3.2.11	BadEntityManagers	24
3.2.12	BasicResource	24
3.2.13	BeeGFSAdmonRole	24
3.2.14	BeeGFSClientRole	24
3.2.15	BeeGFSManagementRole	24
3.2.16	BeeGFSMetadataRole	24
3.2.17	BeeGFSStorageRole	24
3.2.18	BigDataAdditionalTool	24
3.2.19	BigDataAdvancedSettings	24
3.2.20	BigDataCassandra	24
3.2.21	BigDataFileSystemSettings	24
3.2.22	BigDataJobManagementSettings	24

3.2.23	BigDataLoggingSettings	24
3.2.24	BigDataSecurity	24
3.2.25	BigDataSpark	24
3.2.26	BillingHistory	24
3.2.27	BMCSettings	24
3.2.28	BootRole	24
3.2.29	BurnConfig	24
3.2.30	BurnStatus	24
3.2.31	BurnTestStatus	24
3.2.32	Category	24
3.2.33	Ceph	24
3.2.34	CephMDSRole	24
3.2.35	CephMGRRole	24
3.2.36	CephMonitorRole	24
3.2.37	CephOSDBlueStoreConfig	24
3.2.38	CephOSDConfig	24
3.2.39	CephOSDFileStoreConfig	24
3.2.40	CephOSDLegacyConfig	24
3.2.41	CephOSDPool	25
3.2.42	CephOSDRole	25
3.2.43	CephState	25
3.2.44	Certificate	25
3.2.45	CertificateRequest	25
3.2.46	CertificateSubjectName	25
3.2.47	Cgroup	25
3.2.48	CgroupController	25
3.2.49	CgroupControllerBlkio	25
3.2.50	CgroupControllerCpu	25
3.2.51	CgroupControllerCpuacct	25
3.2.52	CgroupControllerCpuset	25
3.2.53	CgroupControllerDevices	25
3.2.54	CgroupControllerFreezer	25
3.2.55	CgroupControllerHugetlb	25
3.2.56	CgroupControllerMemory	25
3.2.57	CgroupControllerNetcls	25
3.2.58	CgroupControllerNetprio	25
3.2.59	CgroupControllerNs	25
3.2.60	CgroupControllerPerf	25
3.2.61	CgroupRule	25
3.2.62	CgroupSupervisorRole	25
3.2.63	Chassis	25
3.2.64	ChronosRole	25
3.2.65	ClientUserData	25
3.2.66	CloudDirectorRole	25
3.2.67	CloudGatewayRole	25
3.2.68	CloudImage	25

3.2.69	CloudJobDescription	25
3.2.70	CloudJobSubmissionStatus	25
3.2.71	CloudNode	25
3.2.72	CloudPrivateCloud	25
3.2.73	CloudProvider	25
3.2.74	CloudRegion	25
3.2.75	CloudSettings	25
3.2.76	CloudStaticIP	25
3.2.77	CloudStorageActionData	25
3.2.78	CloudStorageNodeState	25
3.2.79	CloudType	25
3.2.80	CloudVirtualNetworkInterface	25
3.2.81	ClusterSetup	25
3.2.82	CMDaemonBackgroundTask	25
3.2.83	CMDaemonFailover	25
3.2.84	CMDaemonFailoverGroup	25
3.2.85	CMDaemonFailoverGroupStatus	25
3.2.86	CMDaemonFailoverPeer	25
3.2.87	CMDaemonFailoverStatus	25
3.2.88	CMDaemonStatus	26
3.2.89	CMService	26
3.2.90	CMSubConfig	26
3.2.91	CMSubIntermediateStorage	26
3.2.92	ConfigFileVersion	26
3.2.93	ConfigSum	26
3.2.94	ConfigurationOverlay	26
3.2.95	Consolidator	26
3.2.96	ContainerdHostRole	26
3.2.97	ContainerInfo	26
3.2.98	CustomizationEntry	26
3.2.99	CustomizationFile	26
3.2.100	DellClustat	26
3.2.101	DellClustatGroup	26
3.2.102	DellClustatNode	26
3.2.103	DellDiskGroupInfo	26
3.2.104	DellPhysicalDiskDriveInfo	26
3.2.105	DellRAIDControllerInfo	26
3.2.106	DellSettings	26
3.2.107	DellSettingsFirmware	26
3.2.108	DellSettingsNicDevice	26
3.2.109	DellStorageInfo	26
3.2.110	DellVirtualDiskInfo	26
3.2.111	Device	26
3.2.112	DevStatus	26
3.2.113	DiskAssertion	26
3.2.114	DiskDevice	26

3.2.115 DiskInfo	26
3.2.116 DiskPartition	26
3.2.117 DiskRaid	26
3.2.118 DiskSetup	26
3.2.119 DiskVolume	26
3.2.120 DiskVolumeGroup	26
3.2.121 DockerHostRole	26
3.2.122 DockerRegistryFilesystemStorageDriver	26
3.2.123 DockerRegistryInmemoryStorageDriver	26
3.2.124 DockerRegistryRole	26
3.2.125 DockerRegistryStorageDriver	26
3.2.126 DockerStorageBackend	26
3.2.127 DockerStorageDeviceMapperBackend	26
3.2.128 DrainAction	26
3.2.129 DrainResult	26
3.2.130 EC2AMI	26
3.2.131 EC2AvailabilityZone	26
3.2.132 EC2EBSStorage	26
3.2.133 EC2EphemeralStorage	26
3.2.134 EC2PrivateCloud	26
3.2.135 EC2Provider	27
3.2.136 EC2Region	27
3.2.137 EC2RegionAMI	27
3.2.138 EC2Settings	27
3.2.139 EC2StaticIP	27
3.2.140 EC2Storage	27
3.2.141 EC2Type	27
3.2.142 EC2VirtualNetworkInterface	27
3.2.143 ElasticSearchRole	27
3.2.144 EntityManagersMD5	27
3.2.145 EtcCluster	27
3.2.146 EtcHostRole	27
3.2.147 EthernetSwitch	27
3.2.148 FailoverRole	27
3.2.149 FakeJob	27
3.2.150 FakeJobQueue	27
3.2.151 FakeJobQueueStat	27
3.2.152 FakeWlmClientRole	27
3.2.153 FakeWlmServerRole	27
3.2.154 FileInfo	27
3.2.155 FileSyncConfig	27
3.2.156 FileSyncStatus	27
3.2.157 FlannelConfigurationRole	27
3.2.158 FlannelHostRole	27
3.2.159 FlannelNetworkingBackend	27
3.2.160 FlannelNetworkingUdpBackend	27

3.2.161 FlannelNetworkingVxLanBackend	27
3.2.162 FSExport	27
3.2.163 FSMount	27
3.2.164 FSPart	27
3.2.165 FSPartAssociation	27
3.2.166 FSPartBasicAssociation	27
3.2.167 FSPartProviderAssociation	27
3.2.168 GaleraRole	27
3.2.169 GenericDevice	27
3.2.170 GenericResource	27
3.2.171 GPUInfo	27
3.2.172 GPUSettings	27
3.2.173 GpuUnit	27
3.2.174 GPUUnitInfo	27
3.2.175 GridEngineJob	27
3.2.176 GridEngineJobQueue	27
3.2.177 GridEngineJobQueueStat	27
3.2.178 GridEngineParallelEnvironment	27
3.2.179 Group	27
3.2.180 GuiCephOsdPoolInfo	27
3.2.181 GuiCephOverview	27
3.2.182 GuiCephPgsInfo	28
3.2.183 GuiClusterOverview	28
3.2.184 GuiCompleteOpenStackOverview	28
3.2.185 GuiDiskUsage	28
3.2.186 GuiGpuUnitOverview	28
3.2.187 GuiHadoopHDFSDetailHBase	28
3.2.188 GuiHadoopHDFSDetailHDFS	28
3.2.189 GuiHadoopHDFSDetailMapreduce	28
3.2.190 GuiHadoopHDFSDetailSpark	28
3.2.191 GuiHadoopHDFSDetailYarn	28
3.2.192 GuiHadoopHDFSDetailZooKeeper	28
3.2.193 GuiHadoopHDFSOverview	28
3.2.194 GuiJob	28
3.2.195 GuiKubeClusterOverview	28
3.2.196 GuiNetworkInterface	28
3.2.197 GuiNodeOverview	28
3.2.198 GuiNodeStatus	28
3.2.199 GuiOpenStackOverview	28
3.2.200 GuiOpenStackProjectOverview	28
3.2.201 GuiOpenStackTenantOverview	28
3.2.202 GuiPDUBank	28
3.2.203 GuiPDUOutlet	28
3.2.204 GuiPDUOverview	28
3.2.205 GuiSwitchOverview	28
3.2.206 GuiSwitchPort	28

3.2.207 GuiWorkload	28
3.2.208 HadoopAccumuloMasterHDFSConfiguration	28
3.2.209 HadoopAccumuloMasterRole	28
3.2.210 HadoopAccumuloTabletHDFSConfiguration	28
3.2.211 HadoopAccumuloTabletRole	28
3.2.212 HadoopAlluxioMasterHDFSConfiguration	28
3.2.213 HadoopAlluxioMasterRole	28
3.2.214 HadoopAlluxioWorkerHDFSConfiguration	28
3.2.215 HadoopAlluxioWorkerRole	28
3.2.216 HadoopBaseConfiguration	28
3.2.217 HadoopCassandraHDFSConfiguration	28
3.2.218 HadoopCassandraRole	28
3.2.219 HadoopDataNodeHDFSConfiguration	28
3.2.220 HadoopDataNodeRole	28
3.2.221 HadoopDrillHDFSConfiguration	28
3.2.222 HadoopDrillRole	28
3.2.223 HadoopFlinkJobManagerHDFSConfiguration	28
3.2.224 HadoopFlinkJobManagerRole	28
3.2.225 HadoopFlinkTaskManagerHDFSConfiguration	28
3.2.226 HadoopFlinkTaskManagerRole	28
3.2.227 HadoopHBaseClientHDFSConfiguration	28
3.2.228 HadoopHBaseClientRole	28
3.2.229 HadoopHBaseServerHDFSConfiguration	29
3.2.230 HadoopHBaseServerRole	29
3.2.231 HadoopHDFS	29
3.2.232 HadoopHiveHDFSConfiguration	29
3.2.233 HadoopHiveRole	29
3.2.234 HadoopJob	29
3.2.235 HadoopJobQueue	29
3.2.236 HadoopJobQueueStat	29
3.2.237 HadoopJobTrackerHDFSConfiguration	29
3.2.238 HadoopJobTrackerRole	29
3.2.239 HadoopJournalHDFSConfiguration	29
3.2.240 HadoopJournalRole	29
3.2.241 HadoopKafkaServerHDFSConfiguration	29
3.2.242 HadoopKafkaServerRole	29
3.2.243 HadoopKMServerHDFSConfiguration	29
3.2.244 HadoopKMServerRole	29
3.2.245 HadoopNameNodeHDFSConfiguration	29
3.2.246 HadoopNameNodeRole	29
3.2.247 HadoopNFSGatewayHDFSConfiguration	29
3.2.248 HadoopNFSGatewayRole	29
3.2.249 HadoopPigHDFSConfiguration	29
3.2.250 HadoopPigRole	29
3.2.251 HadoopSecondaryNameNodeHDFSConfiguration	29
3.2.252 HadoopSecondaryNameNodeRole	29

3.2.253 HadoopSparkMasterHDFSConfiguration	29
3.2.254 HadoopSparkMasterRole	29
3.2.255 HadoopSparkWorkerHDFSConfiguration	29
3.2.256 HadoopSparkWorkerRole	29
3.2.257 HadoopSparkYARNHDFSConfiguration	29
3.2.258 HadoopSparkYARNRole	29
3.2.259 HadoopSqoopHDFSConfiguration	29
3.2.260 HadoopSqoopRole	29
3.2.261 HadoopStormNimbusHDFSConfiguration	29
3.2.262 HadoopStormNimbusRole	29
3.2.263 HadoopStormSupervisorHDFSConfiguration	29
3.2.264 HadoopStormSupervisorRole	29
3.2.265 HadoopTaskTrackerHDFSConfiguration	29
3.2.266 HadoopTaskTrackerRole	29
3.2.267 HadoopYARNClientHDFSConfiguration	29
3.2.268 HadoopYARNClientRole	29
3.2.269 HadoopYARNServerHDFSConfiguration	29
3.2.270 HadoopYARNServerRole	29
3.2.271 HadoopZeppelinHDFSConfiguration	29
3.2.272 HadoopZeppelinRole	29
3.2.273 HadoopZooKeeperHDFSConfiguration	29
3.2.274 HadoopZooKeeperRole	29
3.2.275 HAProxyEntry	29
3.2.276 HAProxyEntryBind	30
3.2.277 HAProxyRole	30
3.2.278 HAProxyServer	30
3.2.279 HAProxySharedSettings	30
3.2.280 IBSSwitch	30
3.2.281 IPCPerm	30
3.2.282 IPResource	30
3.2.283 Job	30
3.2.284 JobInfo	30
3.2.285 JobInfoStatistics	30
3.2.286 JobQueue	30
3.2.287 JobQueuePlaceholder	30
3.2.288 JobQueueStat	30
3.2.289 KeepalivedEntry	30
3.2.290 KeepalivedRole	30
3.2.291 KernelModule	30
3.2.292 KeyValuePair	30
3.2.293 KibanaRole	30
3.2.294 KubeCluster	30
3.2.295 KubePodInfo	30
3.2.296 KubernetesApiServerRole	30
3.2.297 KubernetesControllerRole	30
3.2.298 KubernetesDashboardRole	30

3.2.299 KubernetesDNSRole	30
3.2.300 KubernetesHeapsterRole	30
3.2.301 KubernetesNodeRole	30
3.2.302 KubernetesProxyRole	30
3.2.303 KubernetesSchedulerRole	30
3.2.304 KubeUserPolicy	30
3.2.305 LicenseInfo	30
3.2.306 LoginRole	30
3.2.307 LogstashForwarderRole	30
3.2.308 LogstashServerCustomFilter	30
3.2.309 LogstashServerCustomListener	30
3.2.310 LogstashServerCustomOutput	30
3.2.311 LogstashServerElasticOutput	30
3.2.312 LogstashServerFilter	30
3.2.313 LogstashServerListener	30
3.2.314 LogstashServerLocalFileListener	30
3.2.315 LogstashServerLumberjackListener	30
3.2.316 LogstashServerOutput	30
3.2.317 LogstashServerRole	30
3.2.318 LogstashServerRSyslogFilter	30
3.2.319 LogstashServerRSyslogListener	30
3.2.320 LogstashServerStdOutput	30
3.2.321 LSFBaseJob	30
3.2.322 LSFBaseJobQueue	30
3.2.323 LSFBaseJobQueueStat	31
3.2.324 LSFGroupsSettings	31
3.2.325 LSFClientRole	31
3.2.326 LSFJob	31
3.2.327 LSFJobQueue	31
3.2.328 LSFJobQueueStat	31
3.2.329 LSFServerRole	31
3.2.330 LustreAlert	31
3.2.331 LustreClientMount	31
3.2.332 LustreFileSystem	31
3.2.333 LustreFileSystemTarget	31
3.2.334 LustreLog	31
3.2.335 LustreOverview	31
3.2.336 LustreServer	31
3.2.337 LustreServerProfile	31
3.2.338 LustreSettings	31
3.2.339 LustreTargetMap	31
3.2.340 LustreUser	31
3.2.341 LustreVolume	31
3.2.342 LustreVolumeNode	31
3.2.343 MarathonRole	31
3.2.344 MasterNode	31

3.2.345 MasterRole	31
3.2.346 MemcachedRole	31
3.2.347 MemoryInfo	31
3.2.348 MesosCluster	31
3.2.349 MesosDNSRole	31
3.2.350 MesosMasterRole	31
3.2.351 MesosProxyRole	31
3.2.352 MesosResourceUsage	31
3.2.353 MesosSlaveRole	31
3.2.354 MICHostRole	31
3.2.355 MICInfo	31
3.2.356 MICNode	31
3.2.357 MICNodeCategory	31
3.2.358 MICOverlay	31
3.2.359 MICSettings	31
3.2.360 MonitoringAction	31
3.2.361 MonitoringActionRunData	31
3.2.362 MonitoringCacheSubSystemInfo	31
3.2.363 MonitoringCategoryListExecutionFilter	31
3.2.364 MonitoringCompareExpression	31
3.2.365 MonitoringConsolidator	31
3.2.366 MonitoringDataCacheSubSystemInfo	31
3.2.367 MonitoringDataProducer	31
3.2.368 MonitoringDataProducerAggregateNode	31
3.2.369 MonitoringDataProducerAlertLevel	31
3.2.370 MonitoringDataProducerCGroup	32
3.2.371 MonitoringDataProducerClusterTotal	32
3.2.372 MonitoringDataProducerCMDaemonState	32
3.2.373 MonitoringDataProducerDeviceState	32
3.2.374 MonitoringDataProducerEC2SpotPrices	32
3.2.375 MonitoringDataProducerEthernetSwitch	32
3.2.376 MonitoringDataProducerFuture	32
3.2.377 MonitoringDataProducerGalera	32
3.2.378 MonitoringDataProducerGenerator	32
3.2.379 MonitoringDataProducerGPU	32
3.2.380 MonitoringDataProducerInternal	32
3.2.381 MonitoringDataProducerJob	32
3.2.382 MonitoringDataProducerJobQueue	32
3.2.383 MonitoringDataProducerLua	32
3.2.384 MonitoringDataProducerMonitoringSystem	32
3.2.385 MonitoringDataProducerOpenStack	32
3.2.386 MonitoringDataProducerOpenStackHealth	32
3.2.387 MonitoringDataProducerPerpetual	32
3.2.388 MonitoringDataProducerPowerDistributionUnit	32
3.2.389 MonitoringDataProducerProcMemInfo	32
3.2.390 MonitoringDataProducerProcMount	32

3.2.391 MonitoringDataProducerProcNetDev	32
3.2.392 MonitoringDataProducerProcNetSnmp	32
3.2.393 MonitoringDataProducerProcPidStat	32
3.2.394 MonitoringDataProducerProcStat	32
3.2.395 MonitoringDataProducerProcVMStat	32
3.2.396 MonitoringDataProducerRackSensor	32
3.2.397 MonitoringDataProducerScript	32
3.2.398 MonitoringDataProducerSingleLineHealthCheckScript	32
3.2.399 MonitoringDataProducerSingleLineMetricScript	32
3.2.400 MonitoringDataProducerSingleLineScript	32
3.2.401 MonitoringDataProducerSmart	32
3.2.402 MonitoringDataProducerSysBlockStat	32
3.2.403 MonitoringDataProducerSysInfo	32
3.2.404 MonitoringDataProducerTest	32
3.2.405 MonitoringDataProducerTrustedTool	32
3.2.406 MonitoringDataProducerUserCount	32
3.2.407 MonitoringDeviceStateSubSystemInfo	32
3.2.408 MonitoringDrainAction	32
3.2.409 MonitoringEmailAction	32
3.2.410 MonitoringEventAction	32
3.2.411 MonitoringExecutionFilter	32
3.2.412 MonitoringExecutionMultiplexer	32
3.2.413 MonitoringExpression	32
3.2.414 MonitoringGroupedExpression	32
3.2.415 MonitoringHealthOverview	32
3.2.416 MonitoringImageUpdateAction	32
3.2.417 MonitoringJobMetricSettings	33
3.2.418 MonitoringLuaExecutionFilter	33
3.2.419 MonitoringLuaExecutionMultiplexer	33
3.2.420 MonitoringMeasurable	33
3.2.421 MonitoringMeasurableEnum	33
3.2.422 MonitoringMeasurableHealthCheck	33
3.2.423 MonitoringMeasurableMetric	33
3.2.424 MonitoringNodeListExecutionFilter	33
3.2.425 MonitoringOverlayListExecutionFilter	33
3.2.426 MonitoringPlotterSubSystemInfo	33
3.2.427 MonitoringPowerAction	33
3.2.428 MonitoringPowerOffAction	33
3.2.429 MonitoringPowerOnAction	33
3.2.430 MonitoringPowerResetAction	33
3.2.431 MonitoringRebootAction	33
3.2.432 MonitoringReplicateConfiguration	33
3.2.433 MonitoringReplicateSource	33
3.2.434 MonitoringReplicateSubSystemInfo	33
3.2.435 MonitoringResourceExecutionFilter	33
3.2.436 MonitoringResourceExecutionMultiplexer	33

3.2.437 MonitoringRole	33
3.2.438 MonitoringScriptAction	33
3.2.439 MonitoringServiceAction	33
3.2.440 MonitoringServiceRestartAction	33
3.2.441 MonitoringServiceStartAction	33
3.2.442 MonitoringServiceStopAction	33
3.2.443 MonitoringServiceSubSystemInfo	33
3.2.444 MonitoringShutdownAction	33
3.2.445 MonitoringStorageSubSystemInfo	33
3.2.446 MonitoringSubSystemInfo	33
3.2.447 MonitoringTrigger	33
3.2.448 MonitoringTypeExecutionFilter	33
3.2.449 MonitoringTypeExecutionMultiplexer	33
3.2.450 MonitoringUndrainAction	33
3.2.451 MsgQueue	33
3.2.452 MyrinetSwitch	33
3.2.453 Network	33
3.2.454 NetworkAliasInterface	33
3.2.455 NetworkBmcInterface	33
3.2.456 NetworkBondInterface	33
3.2.457 NetworkBridgeInterface	33
3.2.458 NetworkInterface	33
3.2.459 NetworkNetMapInterface	33
3.2.460 NetworkPhysicalInterface	33
3.2.461 NetworkTunnelInterface	33
3.2.462 NetworkVLANInterface	33
3.2.463 NewNode	33
3.2.464 NginxRole	34
3.2.465 Node	34
3.2.466 NodeCategory	34
3.2.467 NodeGroup	34
3.2.468 OpenLavaCgroupsSettings	34
3.2.469 OpenLavaClientRole	34
3.2.470 OpenLavaJob	34
3.2.471 OpenLavaJobQueue	34
3.2.472 OpenLavaJobQueueStat	34
3.2.473 OpenLavaServerRole	34
3.2.474 OpenStack	34
3.2.475 OpenStackApiAgent	34
3.2.476 OpenStackApiDomain	34
3.2.477 OpenStackApiEndpoint	34
3.2.478 OpenStackApiEntity	34
3.2.479 OpenStackApiFlavor	34
3.2.480 OpenStackApiFloatingIP	34
3.2.481 OpenStackApiGroup	34
3.2.482 OpenStackApiHostAggregate	34

3.2.483 OpenStackApiHypervisor	34
3.2.484 OpenStackApiImage	34
3.2.485 OpenStackApiNetwork	34
3.2.486 OpenStackApiPort	34
3.2.487 OpenStackApiProject	34
3.2.488 OpenStackApiRole	34
3.2.489 OpenStackApiRoleAssignment	34
3.2.490 OpenStackApiRouter	34
3.2.491 OpenStackApiSecurityGroup	34
3.2.492 OpenStackApiServer	34
3.2.493 OpenStackApiService	34
3.2.494 OpenStackApiStack	34
3.2.495 OpenStackApiSubnet	34
3.2.496 OpenStackApiUser	34
3.2.497 OpenStackApiVolume	34
3.2.498 OpenStackApiVolumeSnapshot	34
3.2.499 OpenStackApiVolumeType	34
3.2.500 OpenStackAuthBackend	34
3.2.501 OpenStackAuthBackendHybrid	34
3.2.502 OpenStackAuthBackendLDAP	34
3.2.503 OpenStackAuthBackendLDAPGroupSettings	34
3.2.504 OpenStackAuthBackendLDAPProjectSettings	34
3.2.505 OpenStackAuthBackendLDAPRoleSettings	34
3.2.506 OpenStackAuthBackendLDAPUserSettings	34
3.2.507 OpenStackAuthBackendSQL	34
3.2.508 OpenStackBareMetalApiRole	34
3.2.509 OpenStackBareMetalConductorRole	34
3.2.510 OpenStackBareMetalDiscoverdDNSMasqRole	34
3.2.511 OpenStackBareMetalDiscoverdRole	35
3.2.512 OpenStackBlockStorage	35
3.2.513 OpenStackComputeApiEC2Role	35
3.2.514 OpenStackComputeApiMetadataRole	35
3.2.515 OpenStackComputeApiRole	35
3.2.516 OpenStackComputeConductorRole	35
3.2.517 OpenStackComputeRole	35
3.2.518 OpenStackComputeSchedulerRole	35
3.2.519 OpenStackComputeVNCProxyRole	35
3.2.520 OpenStackDashboardRole	35
3.2.521 OpenStackDataProcessingApiRole	35
3.2.522 OpenStackDBaaSRole	35
3.2.523 OpenStackDefaultUserRole	35
3.2.524 OpenStackIdentityApiRole	35
3.2.525 OpenStackImageApiRole	35
3.2.526 OpenStackImageBackend	35
3.2.527 OpenStackImageBackendCeph	35
3.2.528 OpenStackImageBackendFS	35

3.2.529 OpenStackImageRegistryRole	35
3.2.530 OpenStackMessageQueueServerRole	35
3.2.531 OpenStackNetworkApiRole	35
3.2.532 OpenStackNetworkDHCPAgentRole	35
3.2.533 OpenStackNetworkL3AgentRole	35
3.2.534 OpenStackNetworkMetadataAgentRole	35
3.2.535 OpenStackNetworkOVSAgentRole	35
3.2.536 OpenStackNovaImageBackend	35
3.2.537 OpenStackNovaImageBackendCeph	35
3.2.538 OpenStackNovaImageBackendCow	35
3.2.539 OpenStackObjectAccountRole	35
3.2.540 OpenStackObjectApiRole	35
3.2.541 OpenStackObjectContainerRole	35
3.2.542 OpenStackObjectStoreRole	35
3.2.543 OpenStackOrchestrationApiRole	35
3.2.544 OpenStackOrchestrationRole	35
3.2.545 OpenStackSettings	35
3.2.546 OpenStackSettingsAdvanced	35
3.2.547 OpenStackSettingsAuthentication	35
3.2.548 OpenStackSettingsCMDaemonInteractions	35
3.2.549 OpenStackSettingsCollection	35
3.2.550 OpenStackSettingsCompute	35
3.2.551 OpenStackSettingsCredentials	35
3.2.552 OpenStackSettingsDatabase	35
3.2.553 OpenStackSettingsLogging	35
3.2.554 OpenStackSettingsNetworking	35
3.2.555 OpenStackSettingsPorts	35
3.2.556 OpenStackSettingsQuota	35
3.2.557 OpenStackSettingsUserPortal	35
3.2.558 OpenStackSettingsUsers	36
3.2.559 OpenStackStorage	36
3.2.560 OpenStackTelemetryAgentCentralRole	36
3.2.561 OpenStackTelemetryAgentComputeRole	36
3.2.562 OpenStackTelemetryAgentIpmiRole	36
3.2.563 OpenStackTelemetryAgentNotificationRole	36
3.2.564 OpenStackTelemetryAlarmEvaluatorRole	36
3.2.565 OpenStackTelemetryAlarmNotifierRole	36
3.2.566 OpenStackTelemetryApiRole	36
3.2.567 OpenStackTelemetryCollectorRole	36
3.2.568 OpenStackUserRole	36
3.2.569 OpenStackVolumeApiRole	36
3.2.570 OpenStackVolumeBackend	36
3.2.571 OpenStackVolumeBackend3PAR	36
3.2.572 OpenStackVolumeBackendCeph	36
3.2.573 OpenStackVolumeBackendDellStorageCenter	36
3.2.574 OpenStackVolumeBackendGPFS	36

3.2.575 OpenStackVolumeBackendNetApp	36
3.2.576 OpenStackVolumeBackendNFS	36
3.2.577 OpenStackVolumeBackendSolidFire	36
3.2.578 OpenStackVolumeBackupBackend	36
3.2.579 OpenStackVolumeBackupBackendCeph	36
3.2.580 OpenStackVolumeBackupRole	36
3.2.581 OpenStackVolumeRole	36
3.2.582 OpenStackVolumeSchedulerRole	36
3.2.583 OpenvSwitchRole	36
3.2.584 OsapiPortIP	36
3.2.585 OsapiSecurityGroupRule	36
3.2.586 OsapiStackResource	36
3.2.587 OsapiSubnetAllocationPool	36
3.2.588 OSService	36
3.2.589 OSServiceArray	36
3.2.590 OSServiceConfig	36
3.2.591 ParentJob	36
3.2.592 Partition	36
3.2.593 PBSJob	36
3.2.594 PBSJobQueue	36
3.2.595 PBSJobQueueStat	36
3.2.596 PbsProCgroupsSettings	36
3.2.597 PbsProClientRole	36
3.2.598 PbsProJob	36
3.2.599 PbsProJobQueue	36
3.2.600 PbsProJobQueueStat	36
3.2.601 PbsProServerRole	36
3.2.602 PDUPort	36
3.2.603 PhysicalNode	36
3.2.604 PowerDistributionUnit	36
3.2.605 PowerOperation	37
3.2.606 PowerStatus	37
3.2.607 Process	37
3.2.608 Processor	37
3.2.609 Profile	37
3.2.610 ProgramRunnerInput	37
3.2.611 ProgramRunnerKill	37
3.2.612 ProgramRunnerOutput	37
3.2.613 ProgramRunnerStatus	37
3.2.614 ProvisioningNodeStatus	37
3.2.615 ProvisioningProcessorJob	37
3.2.616 ProvisioningRequestStatus	37
3.2.617 ProvisioningRole	37
3.2.618 ProvisioningStatus	37
3.2.619 Puppet	37
3.2.620 PuppetApplyOnNodeRequest	37

3.2.621 PuppetApplyResult	37
3.2.622 PuppetApplySession	37
3.2.623 PuppetClass	37
3.2.624 PuppetClassDeclaration	37
3.2.625 PuppetClassFactory	37
3.2.626 PuppetConfigurationEntry	37
3.2.627 PuppetForgeInstallation	37
3.2.628 PuppetForgePagination	37
3.2.629 PuppetForgeSearchRequest	37
3.2.630 PuppetForgeSearchResult	37
3.2.631 PuppetKeyValuePair	37
3.2.632 PuppetModule	37
3.2.633 PuppetModuleDependency	37
3.2.634 PuppetModuleRelease	37
3.2.635 PuppetOperatingSystemSupport	37
3.2.636 PuppetParameterFactory	37
3.2.637 PuppetRescanResult	37
3.2.638 PuppetResourceDeclaration	37
3.2.639 PuppetRole	37
3.2.640 PuppetRoleChange	37
3.2.641 Rack	37
3.2.642 RackPosition	37
3.2.643 RackSensor	37
3.2.644 RadosGatewayRole	37
3.2.645 RemoteNodeInstallerInteraction	37
3.2.646 RemoteSetupExecution	37
3.2.647 ResourcePool	37
3.2.648 ResourcePoolStatus	37
3.2.649 Role	37
3.2.650 S3BucketIntermediateStorage	37
3.2.651 ScaleDynamicNodesProvider	37
3.2.652 ScaleEngine	38
3.2.653 ScaleHpcEngine	38
3.2.654 ScaleHpcQueueTracker	38
3.2.655 ScaleMesosEngine	38
3.2.656 ScaleMesosLoadTracker	38
3.2.657 ScalePendingWorkload	38
3.2.658 ScaleResourceProvider	38
3.2.659 ScaleServerRole	38
3.2.660 ScaleStaticNodesProvider	38
3.2.661 ScaleTracker	38
3.2.662 Semaphore	38
3.2.663 Sensor	38
3.2.664 Session	38
3.2.665 SGEClientRole	38
3.2.666 SGEJob	38

3.2.667 SGEJobQueue	38
3.2.668 SGEJobQueueStat	38
3.2.669 SGEParallelEnvironment	38
3.2.670 SGEServerRole	38
3.2.671 SharedMemory	38
3.2.672 SlaveNode	38
3.2.673 SlurmCgroupsSettings	38
3.2.674 SlurmClientRole	38
3.2.675 SlurmJob	38
3.2.676 SlurmJobQueue	38
3.2.677 SlurmJobQueueStat	38
3.2.678 SlurmServerRole	38
3.2.679 SoftwareImage	38
3.2.680 SoftwareImageProxy	38
3.2.681 SoftwareImageRevisionInfo	38
3.2.682 StandaloneMonitoredEntity	38
3.2.683 StaticRoute	38
3.2.684 StorageNodePolicy	38
3.2.685 StorageRole	38
3.2.686 StringListObject	38
3.2.687 SubnetManagerRole	38
3.2.688 SubSystemInfo	38
3.2.689 Switch	38
3.2.690 SwitchPort	38
3.2.691 SysInfoCollector	38
3.2.692 Ticket	38
3.2.693 TorqueCgroupsSettings	38
3.2.694 TorqueClientRole	38
3.2.695 TorqueJob	38
3.2.696 TorqueJobQueue	38
3.2.697 TorqueJobQueueStat	38
3.2.698 TorqueServerRole	38
3.2.699 UCSAdaptorEthCompQueueProfile	39
3.2.700 UCSAdaptorEthGenProfile	39
3.2.701 UCSAdaptorEthInterruptProfile	39
3.2.702 UCSAdaptorEthOffloadProfile	39
3.2.703 UCSAdaptorEthRecvQueueProfile	39
3.2.704 UCSAdaptorEthUSNICProfile	39
3.2.705 UCSAdaptorEthWorkQueueProfile	39
3.2.706 UCSAdaptorExtEthIf	39
3.2.707 UCSAdaptorExtIpV6RssHashProfile	39
3.2.708 UCSAdaptorFcCdbWorkQueueProfile	39
3.2.709 UCSAdaptorFcErrorRecoveryProfile	39
3.2.710 UCSAdaptorFcGenProfile	39
3.2.711 UCSAdaptorFcInterruptProfile	39
3.2.712 UCSAdaptorFcPortFLogiProfile	39

3.2.713 UCSAdaptorFcPortPLogiProfile	39
3.2.714 UCSAdaptorFcPortProfile	39
3.2.715 UCSAdaptorFcRecvQueueProfile	39
3.2.716 UCSAdaptorFcWorkQueueProfile	39
3.2.717 UCSAdaptorHostEthIf	39
3.2.718 UCSAdaptorHostFcIf	39
3.2.719 UCSAdaptorIpV4RssHashProfile	39
3.2.720 UCSAdaptorIpV6RssHashProfile	39
3.2.721 UCSAdaptorPortProfiles	39
3.2.722 UCSAdaptorRssProfile	39
3.2.723 UCSBase	39
3.2.724 UCSBiosBootDev	39
3.2.725 UCSBiosBootDevGrp	39
3.2.726 UCSBiosSettings	39
3.2.727 UCSBiosVfAdjacentCacheLinePrefetch	39
3.2.728 UCSBiosVfAltitude	39
3.2.729 UCSBiosVfASPMsSupport	39
3.2.730 UCSBiosVfConsoleRedirection	39
3.2.731 UCSBiosVfCoreMultiProcessing	39
3.2.732 UCSBiosVfCPUEnergyPerformance	39
3.2.733 UCSBiosVfCPUFrequencyFloor	39
3.2.734 UCSBiosVfCPUPerformance	39
3.2.735 UCSBiosVfCPUPowerManagement	39
3.2.736 UCSBiosVfDCUPrefetch	39
3.2.737 UCSBiosVfDemandScrub	39
3.2.738 UCSBiosVfDirectCacheAccess	39
3.2.739 UCSBiosVfDRAMClockThrottling	39
3.2.740 UCSBiosVfDramRefreshRate	39
3.2.741 UCSBiosVfEnhancedIntelSpeedStepTech	39
3.2.742 UCSBiosVfExecuteDisableBit	39
3.2.743 UCSBiosVfFRB2Enable	39
3.2.744 UCSBiosVfHardwarePrefetch	39
3.2.745 UCSBiosVfIntelHyperThreadingTech	39
3.2.746 UCSBiosVfIntelTurboBoostTech	40
3.2.747 UCSBiosVfIntelVirtualizationTechnology	40
3.2.748 UCSBiosVfIntelVTForDirectedIO	40
3.2.749 UCSBiosVfLegacyUSBsSupport	40
3.2.750 UCSBiosVfLOMPortOptionROM	40
3.2.751 UCSBiosVfLvDIMMSupport	40
3.2.752 UCSBiosVfMemoryInterleave	40
3.2.753 UCSBiosVfMemoryMappedIOAbove4GB	40
3.2.754 UCSBiosVfNUMAOptimized	40
3.2.755 UCSBiosVfOnboardStorage	40
3.2.756 UCSBiosVfOnboardStorageSWStack	40
3.2.757 UCSBiosVfOSBootWatchdogTimer	40
3.2.758 UCSBiosVfOSBootWatchdogTimerPolicy	40

3.2.759 UCSBiosVfOSBootWatchdogTimerTimeout	40
3.2.760 UCSBiosVfPatrolScrub	40
3.2.761 UCSBiosVfPCIOptionROMs	40
3.2.762 UCSBiosVfPCISlotOptionROMEnable	40
3.2.763 UCSBiosVfProcessorC1E	40
3.2.764 UCSBiosVfProcessorC6Report	40
3.2.765 UCSBiosVfPStateCoordType	40
3.2.766 UCSBiosVfQPIConfig	40
3.2.767 UCSBiosVfSelectMemoryRASConfiguration	40
3.2.768 UCSBiosVfTPMSupport	40
3.2.769 UCSBiosVfUCSMBootOrderRuleControl	40
3.2.770 UCSBiosVfUSBEmulation	40
3.2.771 UCSBiosVfUSBPortsConfig	40
3.2.772 UCSBiosVfVgaPriority	40
3.2.773 UCSCommNtpProvider	40
3.2.774 UCSCommSyslog	40
3.2.775 UCSCommSyslogClient	40
3.2.776 UCSEquipmentIndicatorLed	40
3.2.777 UCSEquipmentLocatorLed	40
3.2.778 UCSFaultInst	40
3.2.779 UCSFirmwareRunning	40
3.2.780 UCSInfo	40
3.2.781 UCSLogs	40
3.2.782 UCSLsbootDef	40
3.2.783 UCSLsbootEfi	40
3.2.784 UCSLsbootLan	40
3.2.785 UCSLsbootStorage	40
3.2.786 UCSLsbootVirtualMedia	40
3.2.787 UCSStatus	40
3.2.788 UGECgroupsSettings	40
3.2.789 UGEClientRole	40
3.2.790 UGEJob	40
3.2.791 UGEJobQueue	40
3.2.792 UGEJobQueueStat	40
3.2.793 UGEParallelEnvironment	41
3.2.794 UGEServerRole	41
3.2.795 User	41
3.2.796 Validation	41
3.2.797 VersionInfo	41
3.2.798 VirtualNode	41
3.2.799 VirtualNodeSettings	41
3.2.800 VirtualSMPNode	41
3.2.801 VScaleMPSettings	41
3.2.802 VsmPSettings	41
3.2.803 WillChange	41
3.2.804 WlmCgroupsSettings	41

3.2.805 XeonPhiSettings	41
3.2.806 ZooKeeperCluster	41
3.2.807 ZooKeeperHostRole	41
3.3 JSON Examples	41

Preface

Welcome to the *Developer Manual* for Bright Cluster Manager 8.0.

0.1 About This Manual

This manual is aimed at helping developers who would like to program the Bright Cluster Manager in order to enhance or alter its functionality. It is not intended for end users who simply wish to submit jobs that run programs to workload managers, which is discussed in the *User Manual*. The developer is expected to be reasonably familiar with the parts of the *Administrator Manual* that is to be dealt with—primarily CMDaemon, of which `cmsh` and `cmgui` are the front ends.

This manual discusses the Python API to CMDaemon, and also covers how to program for metric collections.

0.2 About The Manuals In General

Regularly updated versions of the Bright Cluster Manager 8.0 manuals are available on updated clusters by default at `/cm/shared/docs/cm`. The latest updates are always online at <http://support.brightcomputing.com/manuals>.

- The *Administrator Manual* describes the general management of the cluster.
- The *Installation Manual* describes installation procedures for a basic cluster.
- The *User Manual* describes the user environment and how to submit jobs for the end user.
- The *Cloudbursting Manual* describes how to deploy the cloud capabilities of the cluster.
- The *Developer Manual* has useful information for developers who would like to program with Bright Cluster Manager.
- The *OpenStack Deployment Manual* describes how to deploy OpenStack with Bright Cluster Manager.
- The *Big Data Deployment Manual* describes how to deploy Big Data with Bright Cluster Manager.
- The *UCS Deployment Manual* describes how to deploy the Cisco UCS server with Bright Cluster Manager.
- The *Machine Learning Manual* describes how to install and configure machine learning capabilities with Bright Cluster Manager.

If the manuals are downloaded and kept in one local directory, then in most pdf viewers, clicking on a cross-reference in one manual that refers to a section in another manual opens and displays that section in the second manual. Navigating back and forth between documents is usually possible with keystrokes or mouse clicks.

For example: `<Alt>-<Backarrow>` in Acrobat Reader, or clicking on the bottom leftmost navigation button of `xpdf`, both navigate back to the previous document.

The manuals constantly evolve to keep up with the development of the Bright Cluster Manager environment and the addition of new hardware and/or applications. The manuals also regularly incorporate customer feedback. Administrator and user input is greatly valued at Bright Computing. So any comments, suggestions or corrections will be very gratefully accepted at manuals@brightcomputing.com.

0.3 Getting Administrator-Level Support

If the reseller from whom Bright Cluster Manager was bought offers direct support, then the reseller should be contacted.

Otherwise the primary means of support is via the website <https://support.brightcomputing.com>. This allows the administrator to submit a support request via a web form, and opens up a trouble ticket. It is a good idea to try to use a clear subject header, since that is used as part of a reference tag as the ticket progresses. Also helpful is a good description of the issue. The followup communication for this ticket goes via standard e-mail. Section 13.2 of the *Administrator Manual* has more details on working with support.

0.4 Getting Developer-Level Support

Developer support is given free, within reason. For more extensive support, Bright Computing can be contacted in order to arrange a support contract.

0.5 Getting Professional Services

Bright Computing normally differentiates between professional services (customer asks Bright Computing to do something or asks Bright Computing to provide some service) and support (customer has a question or problem that requires an answer or resolution). Professional services can be provided after consulting with the reseller, or the Bright account manager.

1

Bright Cluster Manager Python API

This chapter introduces the Python API of Bright Cluster Manager. For a head node `bright80`, the API reference documentation for all available objects is available in a default cluster via browser access to the URL:

```
https://bright80/userportal/downloads/python
```

The preceding access is via the User Portal (section 12.7 of the *Administrator Manual*).

The documentation is also available directly on the head node itself at:

```
file:///cm/local/docs/cmd/python/index.html
```

1.1 Installation

The Python cluster manager bindings are pre-installed on the head node.

1.1.1 Linux Clients

For Linux clients, a redistributable source package is supplied in the `pythoncm-dist` package installed on the cluster. The file at `/cm/shared/apps/pythoncm/dist/pythoncm-8.0-r18836-src.tar.bz2`—the exact version number may differ—is copied and untarred to any directory.

The `build.sh` script is then run to compile the source. About 4GB of memory is usually needed for compilation, and additional packages may be required for compilation to succeed. A list of packages needed to build Python cluster manager bindings can be found in the `README` file included with the package.

The `headnodeinfo.py` example supplied with the untarred files is edited as for in the earlier windows client example, for the `clustermanager.addCluster` line.

The path to the remote cluster manager library is added:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:remotecm
```

To verify everything is working, the following can be run:

```
python ./headnodeinfo.py
```

1.2 Examples

A set of examples can be found in `/cm/local/examples/cmd/python/` on the head node of the cluster.

1.2.1 First Program

A Python script is told to use the cluster manager bindings by importing `pythoncm` at the start of the script:

```
import pythoncm
```

If not working on the cluster, the administrator needs to set the path where the shared libraries can be found (`pythoncm.so` in Linux, or `python.pyd` in windows). This is done by adding the following to the start of the script:

```
import sys
sys.path.append(".") # path to pythoncm.so/python.pyd
```

Python cluster manager bindings allow for simultaneous connections to several clusters. For this reason the first thing to do is to create a `ClusterManager` object:

```
clustermanager = pythoncm.ClusterManager()
```

A connection to a cluster can now be made. There are two possible ways of connecting.

The first is using the certificate and private key file that `cmsh` uses by default when it authenticates from the head node.

```
cluster = clustermanager.addCluster('https://mycluster:8081', \
'/root/.cm/admin.pem', '/root/.cm/admin.key');
```

The second way uses the password protected `admin.pfx` file, which is generated with the `cmd -c` command. A Python script could ask for the password and store it in a variable for increased security.

```
cluster = clustermanager.addCluster('https://mycluster:8081', \
'/root/.cm/cmgui/admin.pfx', '', '<password>');
```

Having defined the cluster, a connection can now be made to it:

```
isconnected = cluster.connect()
if !isconnected:
    print "Unable to connect"
    print cluster.getLastError()
    exit(1)
```

If a connection cannot be made, the function `cluster.connect()` returns false. The function `cluster.getLastError()` shows details about the problem. The two most likely problems are due to a wrong password setting or a firewall settings issue.

Similar to `cmgui` and `cmsh`, the cluster object contains a local cache of all objects. This cache will be filled automatically when the connection is established. All changes to properties will be done on these local copies and will be lost after the Python scripts exits, unless a `commit` operation is done.

The most common operation is finding specific objects in the cluster.

```
active = cluster.find('active')
if active == None:
    print "Unable to find active head node"
    exit(1)
else:
    print "Hostname of the active head node is %s" % active.hostname
```

If creating an automated script that runs at certain times, then it is highly recommended to check if objects can be found. During a failover, for instance, there will be a period over a few minutes in which the active head node will not be set.

It is good practice to disconnect from the cluster at the end of the script.

```
cluster.disconnect()
```

When connecting to a cluster with a failover setup, it is the shared IP address that should be connected to, and not the fixed IP address of either of the head nodes.

1.3 Methods And Properties

1.3.1 Viewing All Properties And Methods

All properties visible in `cmsh` and `cmgui` are also accessible from Python cluster manager bindings. The easiest way to get an overview of the methods and properties of an object is to define the following function:

```
import re
def dump(obj):
    print "--- DUMP ---"
    for attr in dir(obj):
        p = re.compile('^__.*__$')
        if not p.match(attr):
            print "%s = %s" % (attr, getattr(obj, attr))
```

An overview of all properties and methods for the active head node can be obtained with:

```
active = cluster.find('active')
dump(active)
```

1.3.2 Property Lists

Most properties are straightforward and their names are almost identical to the `cmsh` equivalent.

For instance:

```
node.mac = '00:00:00:00:00:00'
category.softwareimage = cluster.find('testimage')
```

Properties that contain lists, like `node.roles`, `node.interfaces`, `category.fsmounts` and several others, are trickier to deal with. While iterating over a list property is simple enough:

```
for role in node.roles:
    print role.name
```

because of an implementation restriction, adding a new role requires that a local copy of the roles list be made:

```
roles = node.roles
provisioningrole = pythoncm.ProvisioningRole() # Create a new pro\
                                              visioning role object
roles.append(provisioningrole)
node.roles = roles # This will update the internal\
                  roles list with the local copy
```

1.3.3 Creating New Objects

Creating a new node can be done with:

```
node = pythoncm.Node()
```

This is valid command, but fairly useless because a node has to be a `MasterNode`, `PhysicalNode` or `VirtualSMPNode`. So to create a normal compute or login node, the object is created as follows:

```
node = pythoncm.PhysicalNode()
```

The first thing to do after creating a new object is to add it to a cluster.

```
cluster.add(node)
```

It is impossible to add one node to more than one cluster.

After the node has been added its properties can be set. In `cmsh` and `cmgui` this is semi-automated, but in Python cluster manager bindings it has to be done by hand.

```
node.hostname = 'node001'
node.partition = cluster.find('base')
node.category = cluster.find('default')
```

Similar to the node object, a NetworkInterface object has several subtypes: NetworkPhysicalInterface, NetworkVLANInterface, NetworkAliasInterface, NetworkBondInterface, and NetworkIPMIInterface.

```
interface = pythoncm.NetworkPhysicalInterface()
interface.name = 'eth0'
interface.ip = '10.141.0.1'
interface.network = cluster.find('internalnet')
node.interfaces = [interface]
node.provisioningInterface = interface
```

Having set the properties of the new node, it can now be committed.

```
cr = node.commit()
```

If a commit fails for some reason, the reason can be found:

```
if not cr.result:
    print "Commit of %s failed:" % node.resolveName()
    for j in range(cr.count):
        print cr.getValidation(j).msg
```

1.3.4 List Of Objects

In the following lists of objects:

- Objects marked with (*) cannot be used
- Trees marked with (+) denote inheritance

Roles

```
Role (*)
+ BackupRole
+ BootRole
+ DatabaseRole
+ EthernetSwitch
+ LoginRole
+ LSFClientRole
+ LSFServerRole
+ MasterRole
+ PbsProClientRole
+ PbsProServerRole
+ ProvisioningRole
+ SGEClientRole
+ SGEServerRole
+ SlurmClientRole
+ SlurmServerRole
+ SubnetManagerRole
+ TorqueClientRole
+ TorqueServerRole
```

Devices

Device (*)
+ Chassis
+ GpuUnit
+ GenericDevice
+ PowerDistributionUnit
+ Switch (*)
 + EthernetSwitch
 + IBSwitch
 + MyrinetSwitch
Node (*)
+ FSExport
+ FSMount
+ MasterNode
+ SlaveNode (*)
 + PhysicalNode
 + VirtualSMPNode

Network Interfaces

NetworkInterface (*)
+ NetworkAliasInterface
+ NetworkBondInterface
+ NetworkIpmiInterface
+ NetworkPhysicalInterface
+ NetworkVLANInterface

Information Objects

ClusterSetup
GuiClusterOverview
GuiCephOverview
GuiHadoopHDFSOverview
GuaOpenStackOverview
GuiOpenStackTenantOverview
GuiGpuUnitOverview
GuiNodeOverview
GuiNodeStatus
LicenseInfo
SysInfoCollector
VersionInfo

LDAP Objects

User
Group

Category Objects

Category
FSExport
FSMount

Miscellaneous Objects

SoftwareImage

KernelModule

Network

NodeGroup

Partition

+ BurnConfig

Rack

1.3.5 Useful Methods

For The Cluster Object:

Name	Description
<code>find(<name>)</code>	Find the object with a given name, <i><name></i>
<code>find(<name>, <type>)</code>	Because it is possible to give a category and node the same name, sometimes the type <i><type></i> of the object needs to be specified too
<code>getAll(<type>)</code>	Get a list of all objects of a given type: e.g. device, category
<code>activeMaster()</code>	Get the active master object
<code>passiveMaster()</code>	Get the active master object
<code>overview()</code>	Get all the data shown in the <code>cmgui</code> cluster overview
<code>add(<object>)</code>	Add a newly created object <i><object></i> to the cluster. Only after an object is added can it be used
<code>pexec(<nodes>, <command>)</code>	Execute a command <i><command></i> on one or more nodes

For Any Object:

Name	Description
<code>commit()</code>	Save changes to the cluster
<code>refresh()</code>	Undo all changes and restore the object to its last saved state
<code>remove()</code>	Remove an object from the cluster
<code>clone()</code>	Make an identical copy. The newly created object is not added to a cluster yet

For Any Device:

Name	Description
<code>close()</code>	Close a device
<code>open()</code>	Open a device
<code>powerOn()</code>	Power on a device
<code>powerOff()</code>	Power off a device
<code>powerReset()</code>	Power reset a device
<code>latestMonitoringData()</code>	Return a list of the most recent monitoring data

For Any Node:

Name	Description
<code>overview()</code>	Get the data displayed in the <code>cmgui</code> node overview tab
<code>sysinfo()</code>	Get the data displayed in the <code>cmgui</code> node system information tab
<code>pexec(<command>)</code>	Execute a command

1.3.6 Useful Example Program

In the directory `/cm/local/examples/cmd/python` are some example programs using the python API.

One of these is `printall.py`. It displays values for objects in an easily viewed way. With `all` as the argument, it displays resource objects defined in a list in the program. The objects are 'Partition', 'MasterNode', 'SlaveNode', 'Category', 'SoftwareImage', 'Network', 'NodeGroup'. The output is displayed something like (some output elided):

Example

```
[root@bright80 ~]# cd /cm/local/examples/cmd/python
[root@bright80 python]# ./printall all
Partition base
+- revision .....
| name ..... base
| clusterName ..... Bright 8.0 Cluster
...
| burnConfigs
| +- revision .....
| | name ..... default
| | description ..... Standard burn test.
| | configuration ..... < 2780 bytes >
| +- revision .....
| | name ..... long-hpl
...
| provisioningInterface ..... None
| fsmounts ..... < none >
| fsexports
| +- revision .....
| | name ..... /cm/shared@internalnet
| | path ..... /cm/shared
| | hosts ..... !17179869185!
...
Category default
+- revision .....
| name ..... default
| softwareImage ..... default-image
| defaultGateway ..... 10.141.255.253
| nameServers ..... < none >
...
```

The values of a particular resource-level object, such as `softwareimage`, can be viewed by specifying it as the argument:

Example

```
[root@bright80 python]# ./printall.py softwareimage
softwareimage default-image
+- revision .....
| name ..... default-image
| path ..... /cm/images/default-image
| originalImage ..... 0
| kernelVersion ..... 2.6.32-431.11.2.el6.x86_64
| kernelParameters ..... rdblacklist=nouveau
| creationTime ..... 1398679806
| modules
```

```

| +- revision .....
| | name ..... xen-netfront
...
| +- revision .....
| | name ..... hpilo
| | parameters .....
| enableSOL ..... False
| SOLPort ..... ttyS1
| SOLSpeed ..... 115200
| SOLFlowControl ..... True
| notes .....
| fspart ..... 98784247812
| bootfspart ..... 98784247813
...
[root@bright80 python]#

```

1.4 The Workload Management API

The workload management API allows the submission of jobs, the retrieval of information on jobs and queues, and the management of jobs and queues. The methods described in this section are a part of the `cmjob` service. They can also be accessed via the `Cluster` object, with exception of the `getParentJobs` and `getJobsSlice` methods.

Workload management examples for a particular workload manager `<wlm>` in Python can be found on the head node in the directory:

```
/cm/local/examples/cmd/python/workload-<wlm>.py
```

Here, `<wlm>` can take the values `torque`, `slurm`, `sgc`, `pbspro`, `openlava`, or `lsf`. The examples define a job, with different job properties associated with different workload managers. With the right properties set, the job can be submitted and the submitted job outputs are printed to STDOUT.

Details of entities and their properties can be found in the CMDaemon API reference.

1.4.1 Job Submission

Job submission is performed with the `submitJob` method. Its only argument is the `Job` entity that provides the properties and resource requirements of the job that is submitted.

Each workload manager uses its own job properties format, although they usually behave in a similar way. The following table shows the correspondence between `Job` entity parameters and the submission parameters for each workload manager.

Parameter	Slurm	PBS Pro Torque	LSF openlava	UGE OGS (SGE)
queue	-p	-q	-q	-q
jobname	-J	-N	-J	-N

...continues

...continued

Parameter	Slurm	PBS Pro Torque	LSF openlava	UGE OGS (SGE)
account	-A	-A	N/A	-A
project	N/A	-P	-P	-P
rundirectory	-D	-w	N/A	-wd
username	Job script is submitted by this user			
groupname	Job script is submitted with group permissions of this user			
priority	--nice	-p	-sp	-p
stdinfile	-i	N/A	-i	-i
stdoutfile	-o	-o	-o	-o
stderrfile	-e	-e	-e	-e
dependencies	-d	-W depend=	-w	--hold_jid
mailNotify	Enables passing other email options, not used directly			
mailOptions	--mail-type	-m	-B	-m
mailList	--mail-user	-M	-u	-M
resourceList	-C	-l	-R	-l

...continues

...continued

Parameter	Slurm	PBS Pro Torque	LSF openlava	UGE OGS (SGE)
maxWallClock	-t	-l walltime=	-c	-l h_rt=
numberOfProcesses	-n	mpiprocs= ppn=	-n	-pe
numberOfNodes	-N	-l select=	-R 'span[hosts=]'	N/A
nodes	-w	-l select=	-m	-l hostname=

`environmentVariables` All additional environment variables are passed to the job

`commandLineInterpreter`

Interpreter path is added as a first line into the jobscript

`executable` Added as a command at the end of a new created jobscript.

`arguments` Appended to `executable` line

`modules` Module files will be added to job script environment

`userdefined` These lines are added into the jobscript before the `executable` line

`scriptFile` If scriptfile is specified, then only is it submitted

`debug` Return debug info (without submission), including generated script

Notes:

1. In the case of LSF and OpenLava, the `rundirectory` parameter of the `Job` entity is converted into a `cd` command line, that is added to the job script before any commands.
2. The executable file path and its arguments are translated to a single line in the job script. If more complex commands are required then the parameter `userdefined` should be used instead of `executable` and `arguments`. If `userdefined` is not an empty list, then `executable` and `arguments` are ignored.

1.4.2 Job Information And Management

For job manipulation the following functions are used. In these functions, the parameter `<scheduler>` is the name of the workload manager that the operation is applied to, and takes a value of `slurm`, `uge`, `sge`, `openlava`, `lsf`, `torque` or `pbspro`. The parameter `<JobID>` is a string in a format related to that particular workload manager.

getJobs(<scheduler>): returns `Job` entities for the specified scheduler. This function triggers a call to the workload manager utility. The workload manager utility is, for example, `qstat` in the case of SGE or Torque, and `scontrol` in the case of Slurm. In profiles (section 6.4 of the *Administrator Manual*), `GET_JOB_TOKEN` is needed to be able to get all the jobs, while `GET_OWN_JOB_TOKEN` is needed to get just all the jobs belonging to the user making the call.

getJob(<scheduler>, <JobID>): returns a job by job ID. `GET_JOB_TOKEN` is needed to be able to get any job, and `GET_OWN_JOB_TOKEN` is needed to be able to get just the job belonging to the user making the call.

removeJob(<scheduler>, <JobID>): removes the job by job ID and returns the result of job removal. `UPDATE_JOB_TOKEN` is needed to be able to remove any job, and `UPDATE_OWN_JOB_TOKEN` is needed to be able to remove just the job belonging to the user making the call.

getJobsSlice(<scheduler>, <start>, <maxCount>, <parentID>, <allUsers>): returns jobs at the position `<start>` in the global list (sorted by job ID), but only up to `<maxCount>` items. That is, if the value of the parameter `<start>` is a number `n`, then jobs starting from the `n`th item in the global list are returned, up to `<maxCount>` times. `<parentID>` is a method to group jobs by a keyword in the comment string of the jobs. `<allUsers>` specifies, using the value `True` or `False`, whether the jobs of all users should be considered—a value of `False` means that only the jobs owned by the requestor are considered. `GET_JOB_TOKEN` is needed to get any job slice, while `GET_OWN_JOB_TOKEN` is needed to get just the job slices belonging to the user making the call.

getParentJobs(<scheduler>, <start>, <maxCount>, <parentID>, <allUsers>): returns `parentJob` entities at the position `<start>` in the global list (sorted by parent job ID), but only up to `maxCount` items. That is, if the value of the parameter `<start>` is a number `n`, then jobs starting from the `n`th item in the global list are returned, up to `<maxCount>` times. `<parentID>` is a method to group jobs by a keyword in the comment string of the jobs. By default it has an empty value passed to it. If `<parentID>` is given a parent ID value, then the parent job is treated as owned by particular user if and only if all jobs with this tag (parent id) are submitted by that user. Setting `<allUsers>` specifies, using the value `True` or `False`, whether the jobs of all users should be considered—a value of `False` means that only the jobs owned by the requestor are considered. `GET_JOB_TOKEN` is needed to get any job slice, while `GET_OWN_JOB_TOKEN` is needed to get just the job slices belonging to the user making the call.

requeueJob(<scheduler>, <JobID>): requeues job and returns the result of this operation as a string. `REQUEUE_JOB_TOKEN` is needed to be able to requeue any job, while `REQUEUE_OWN_JOB_TOKEN` is needed to be able to requeue just the job belonging to the user making the call.

holdJob(<scheduler>, <JobID>): holds the job and returns the result of this operation as a string. `HOLD_JOB_TOKEN` is needed to be able to hold any job, while `HOLD_OWN_JOB_TOKEN` is needed to be able to hold just the job belonging to the user making the call.

suspendJob(*<scheduler>*, *<JobID>*): suspends the job and returns the result of this operation as a string. `SUSPEND_JOB_TOKEN` is needed to be able to suspend any job, while `SUSPEND_OWN_JOB_TOKEN` is needed to be able to suspend just the job belonging to the user making the call.

resumeJob(*<scheduler>*, *<JobID>*): resumes the job and returns the result of this operation as a string. `RESUME_JOB_TOKEN` is needed to be able to resume any job, while `RESUME_OWN_JOB_TOKEN` is needed to be able to resume just the job belonging to the user making the call.

releaseJob(*<scheduler>*, *<JobID>*): release the job and returns the result of this operation as a string. `RELEASE_JOB_TOKEN` is needed to be able to release any job, while `RELEASE_OWN_JOB_TOKEN` is needed to be able to release just the job belonging to the user making the call.

updateJob(*<scheduler>*, *<JobID>*): update the job and returns result of this operation as a string. `UPDATE_JOB_TOKEN` is needed to be able to update any job, while `UPDATE_OWN_JOB_TOKEN` is needed to be able to update just the job belonging to the user making the call.

isNodeAllocatedForUser(*<scheduler>*, *<username>*, *<hostname>*): returns true if at least one job owned by the user, as specified by the value of *<username>* allocates the host, as specified by the value of *<hostname>*.

Parent job is an entity introduced in Bright 7.3 and serves a goal of jobs clusterization. The jobs can be united by a tag surrounded by square brackets (for example "[*workflow1*]"). The tag is parsed by CMDaemon from the job comment line. The first entry of such a tag in the job comment is considered as the parent job ID. CMDaemon caches parent jobs, and an API client can request all the parent jobs or just some particular one. This allows the client to unite jobs by some user-defined property in a workflow, even if the workload manager does not support the workflow.

1.4.3 Queue Information And Management

For queue manipulation the following functions are used.

getJobQueues(*<scheduler>*): retrieves all `JobQueue` entities. Requires `GET_JOBQUEUE_TOKEN`.

getJobQueue(*<queuename>*): retrieves a particular `JobQueue` entity. Here *<queuename>* is a string. Requires `GET_JOBQUEUE_TOKEN`.

getParallelEnvs(*<scheduler>*): retrieves a list of `ParallelEnvironment` entities associated with a particular workload manager. Requires `GET_PE_TOKEN`.

getJobQueueStates(*<scheduler>*): retrieves a list of `JobQueueStat` entities. Requires `GET_JOBQUEUE_TOKEN`.

updateJobQueue(*<JobQueue>*, *<force>*): updates job queue properties defined by `JobQueue` entity. Parameter *<force>* is ignored for now. Requires `UPDATE_JOBQUEUE_TOKEN`.

addJobQueue(*<JobQueue>*, *<force>*): adds a new job queue to workload manager. If *<force>* has the value `True`, then the existing queue is recreated. Requires `ADD_JOBQUEUE_TOKEN`.

removeJobQueue(*<queueKey>*, *<force>*): removes queue by key. The key can be retrieved from the `JobQueue` entity requested by the `getJobQueue` method. Parameter *<force>* is ignored for now. Requires `UPDATE_JOBQUEUE_TOKEN`.

drainNodes(*<scheduler>*, *<queue>*, *<nodes>*, *<drain>*): drains nodes (as defined by a list of hostnames or uniqueKeys) or a particular queue (if supported by the workload manager) in the workload manager. If *<drain>* has the value 1, then the nodes will be drained, otherwise they are undrained. Returns a list `DrainResult` entities. Requires `DRAIN_TOKEN`.

drainOverview(*<scheduler>*, *<nodes>*): returns `DrainResult` entities with current drain state of the nodes. The nodes are defined by a list of hostnames or uniqueKeys. Requires `DRAIN_OVERVIEW_TOKEN`.

2

Metric Collections

This chapter covers how to add a metric collections script with `cmsh`. It also describes the output specification of a metric collections script, along with example outputs, so that a metric collections script can be made by the administrator.

2.1 Metric Collections Added Using `cmsh`

A metric collections script, `responsiveness`, is added in the `monitoring metrics` mode just like any other metric.

Example

```
[bright80]% monitoring metrics
[bright80->monitoring->metrics]% add responsiveness
[...[responsiveness]]% set command /cm/local/apps/cmd/scripts/metrics/s\
ample_responsiveness
[...*[responsiveness*]]% set classofmetric prototype; commit
```

For `classofmetric`, the value `prototype` is the class used to distinguish metric collections from normal metrics.

2.2 Metric Collections Initialization

When a metric collections script is added to `CMDaemon` for the first time, `CMDaemon` implicitly runs it with the `--initialize` flag. The output is used to define the collections table header structure. The structure is composed of the component metrics in the collections script, and the resulting structure is placed in the `CMDaemon` monitoring database. After the initialization step, data values can be added to the collections table during regular use of the script.

The displayed output of a metric collections script when using the `--initialize` flag is a list of available metrics and their parameter values. The format of each line in the list is:

```
metric <name[:parameter]> <unit> <class> "<description>" <cumulative> <min> <max>
```

where the items in the line are:

- `metric`: A bare word.
- `<name[:parameter]>`: The name of the metric, with for certain metrics a parameter value. For example, the metric `AlertLevel` can have the parameter `sum` assigned to it with the `“:”` character.
- `<unit>`: The unit of measurement that the metric uses.
- `<class>`: Any of:

- ceph,
- cgroups/blkio, cgroups/cpu, cgroups/memory, cgroups/network,
- cluster,
- cpu,
- dellnss,
- disk,
- env,
- gpu,
- hadoop/dfs, hadoop/mapred, hadoop/metricsystem, hadoop/rpcdetailed, hadoop/ugu
- internal,
- lustre,
- madoop/rpc,
- mem,
- misc
- net,
- openstack, openstack/api/allservices, openstack/api/compute, openstack/api/identity, openstack/api/image, openstack/api/network, openstack/api/orchestration, openstack/api/volume, openstack/hypervisors, openstack/vm/blockdevice, openstack/vm/cpu, openstack/vm/memory, openstack/vm/network, openstack/vm/other
- os,
- prototype,
- workload.

- *<description>*: This can contain spaces, but should be enclosed with quotes.
- *<cumulative>*: Either *yes* or *no*. This indicates whether the metric increases monotonically (e.g., bytes received) or not (e.g., temperature).
- *<min>* and *<max>*: The minimum and maximum numeric values of this metric are determined dynamically based on the values so far.

Example

```
[root@myheadnode metrics]# ./sample_responsiveness --initialize
metric util_sda % internal "Percentage of CPU time during which I/O
requests were issued to device sda" no 0 100
metric await_sda ms internal "The average time (in milliseconds) for
I/O requests issued to device sda to be served" no 0 500
```

2.3 Metric Collections Output During Regular Use

The output of a metric collection script without a flag is a list of outputs from the available metrics. The format of each line in the list is:

```
metric <name[:parameter]> <value> [infomessage]
```

where the parameters to the `metric` bare word are:

- `<name [:parameter]>`: The name of the metric, with optional parameter for some metrics.
- `<value>`: The numeric value of the measurement.
- `[infomessage]`: An optional infomessage.

Example

```
[root@myheadnode metrics]# ./sample_responsiveness
metric await_sda 0.00
metric util_sda 0.00
[root@myheadnode metrics]#
```

If the output has more metrics than that suggested by when the `--initialize` flag is used, then the extra sampled data is discarded. If the output has fewer metrics, then the metrics are set to NaN (not a number) for the sample.

A metric or health check inside a metric collection appears as a check when viewing metrics or healthcheck lists. Attempting to remove such a check specifically using `cmsh` or `cmgui` only succeeds until the node is updated or rebooted. It is the metric collection itself that should have the check removed from within it, in order to remove the check from the list of checks permanently.

Setting a node that is UP to a CLOSED state, and then bringing it out of that state with the `open` command (section 5.5.4 of the *Administrator Manual*) also has CMDaemon run the metric collections script with the `--initialize` flag. This is useful for allowing CMDaemon to re-check what metrics in the collections can be sampled, and then re-configure them.

2.4 Metric Collections Error Handling

If the exit code of the script is 0, CMDaemon assumes that there is no error. So, with the `--initialize` flag active, despite no numeric value output, the script does not exit with an error.

If the exit code of the script is non-zero, the output of the script is assumed to be a diagnostic message and is passed to the head node. This shows up as an event in `cmsh` or `cmgui`.

For example, the `sample_ipmi` script uses the `ipmi-sensors` binary internally. Calling the binary directly returns an error code if the device has no IPMI configured. However, the `sample_ipmi` script in this case simply returns 0, and no output. The rationale here being that the administrator is aware of this situation and would not expect data from that IPMI anyway, let alone an error.

2.5 Metric Collections Consolidator Syntax

Metric collections can have a consolidator format defined per metric. The consolidator definition must be placed as an output in the line immediately preceding the corresponding metric initialization output line. The consolidator definition line can take the following forms:

```
consolidators default
consolidators none
consolidators CONSOLIDATORNAME FORMAT SPECIFICATION
```

The meanings of the texts after `consolidators` are as follows:

- `default`: The metrics follow the default consolidator names and interval values (page 432 of the *Administrator Manual*). That is, consolidator names take the value of Hour, Day, Week, while the interval values are the corresponding durations in seconds.
- `consolidators none`: No consolidation is done, only raw data values are collected for the metrics.

- *CONSOLIDATORNAME FORMAT SPECIFICATION*: This has the form:
`<name : interval [: kind [: tablelength]] > . . .`
 - *name*: the consolidator name. A special feature here is that it can also define a new consolidator if the name does not already exist. Multiple consolidators can be defined in each consolidator definition line, with *name* separated from any preceding definition on the same line by a space.
 - *interval*: the duration in seconds, between consolidation, for the consolidator.
 - *kind*: an optional value of *min*, *max*, or *average*. By default it is *average*.
 - *tablelength*: an optional value for the length of the table, if *kind* has been specified. By default it is 1000.

2.6 Metric Collections Environment Variables

The following environment variables are available for a metric collection script, as well as for custom scripts, running from CMDaemon:

On all devices:

`CMD_HOSTNAME`: name of the device. For example:

```
CMD_HOSTNAME=myheadnode
```

Only on non-node devices:

`CMD_IP`: IP address of the device. For example:

```
CMD_IP=192.168.1.33
```

Only on node devices:

Because these devices generally have multiple interfaces, the single environment variable `CMD_IP` is often not enough to express these. Multiple interfaces are therefore represented by these environment variables:

- `CMD_INTERFACES`: list of names of the interfaces attached to the node. For example:

```
CMD_INTERFACES=eth0 eth1 ipmi0 BOOTIF
```

- `CMD_INTERFACE_<interface>_IP`: IP address of the interface with the name `<interface>`. For example:

```
CMD_INTERFACE_eth0_IP=10.141.255.254
CMD_INTERFACE_eth1_IP=0.0.0.0
```

- `CMD_INTERFACE_<interface>_TYPE`: type of interface with the name `<interface>`. For example:

```
CMD_INTERFACE_eth1_TYPE=NetworkPhysicalInterface
CMD_INTERFACE_ipmi0_TYPE=NetworkBmcInterface
```

Possible values are:

- `NetworkBmcInterface`

- NetworkPhysicalInterface
 - NetworkVLANInterface
 - NetworkAliasInterface
 - NetworkBondInterface
 - NetworkBridgeInterface
 - NetworkTunnelInterface
 - NetworkNetMapInterface
- CMD_BMCUSERNAME: username for the BMC device at this node (if available).
 - CMD_BMCPASSWORD: password for the BMC device at this node (if available).

To parse the above information to get the BMC IP address of the node for which this script samples, one could use (in Perl):

```
my $ip;
my $interfaces = $ENV{"CMD_INTERFACES"};
foreach my $interface ( split( " ", $interfaces ) ) {
    if( $ENV{"CMD_INTERFACE_" . $interface . "_TYPE"} eq
        "NetworkBmcInterface" ) {
        $ip = $ENV{"CMD_INTERFACE_" . $interface . "_IP"};
        last;
    }
}
# $ip holds the bmc ip
```

A list of environment variables available under the CMDaemon environment can be found by running a script under CMDaemon and exporting the environment variables to a file for viewing. For example, the `/cm/local/apps/cmd/scripts/healthchecks/testhealthcheck` script can be modified and run to sample on the head node, with the added line: `set>/tmp/environment`. The resulting file `/tmp/environment` that is generated as part of the healthcheck run then includes the `CMD_*` environment variables.

Example

```
CMD_BMCPASSWORD
CMD_BMCUSERNAME
CMD_CLUSTERNAME
CMD_CMDSTARTEDTIME
CMD_DEVICE_TYPE
CMD_EXPORTS
CMD_FSEXPRT__SLASH_cm_SLASH_shared_AT_internalnet_ALLOWWRITE
CMD_FSEXPRT__SLASH_cm_SLASH_shared_AT_internalnet_HOSTS
CMD_FSEXPRT__SLASH_cm_SLASH_shared_AT_internalnet_PATH
CMD_FSEXPRT__SLASH_home_AT_internalnet_ALLOWWRITE
CMD_FSEXPRT__SLASH_home_AT_internalnet_HOSTS
CMD_FSEXPRT__SLASH_home_AT_internalnet_PATH
CMD_FSEXPRT__SLASH_var_SLASH_spool_SLASH_burn_AT_internalnet_ALLOWWRITE
CMD_FSEXPRT__SLASH_var_SLASH_spool_SLASH_burn_AT_internalnet_HOSTS
CMD_FSEXPRT__SLASH_var_SLASH_spool_SLASH_burn_AT_internalnet_PATH
CMD_HOSTNAME
CMD_INTERFACES
CMD_INTERFACE_eth0_IP
CMD_INTERFACE_eth0_MTU
CMD_INTERFACE_eth0_SPEED
CMD_INTERFACE_eth0_STARTIF
```

```

CMD_INTERFACE_eth0_TYPE
CMD_INTERFACE_eth1_IP
CMD_INTERFACE_eth1_MTU
CMD_INTERFACE_eth1_SPEED
CMD_INTERFACE_eth1_STARTIF
CMD_INTERFACE_eth1_TYPE
CMD_IP
CMD_MAC
CMD_METRICNAME
CMD_METRICPARAM
CMD_MOUNTS
CMD_NODEGROUPS
CMD_PARTITION
CMD_PORT
CMD_PROTOCOL
CMD_ROLES
CMD_SCRIPTTIMEOUT
CMD_STATUS
CMD_STATUS_CLOSED
CMD_STATUS_HEALTHCHECK_FAILED
CMD_STATUS_HEALTHCHECK_UNKNOWN
CMD_STATUS_MESSAGE
CMD_STATUS_RESTART_REQUIRED
CMD_STATUS_STATEFLAPPING
CMD_STATUS_USERMESSAGE
CMD_SYSINFO_SYSTEM_MANUFACTURER
CMD_SYSINFO_SYSTEM_NAME
CMD_USERDEFINED1
CMD_USERDEFINED2

```

2.7 Metric Collections Examples

Bright Cluster Manager has several scripts in the `/cm/local/apps/cmd/scripts/metrics` directory. Among them are the metric collections scripts `testmetriccollection` and `sample_responsiveness`. A glance through them while reading this chapter may be helpful.

2.8 Metric Collections On iDataPlex And Similar Units

IBM's iDataPlex is a specially engineered dual node rack unit. When the term iDataPlex is used in the following text in this section, it also implies any other dual node units that show similar behavior.

This section gives details on configuring an iDataPlex if IPMI metrics retrieval seems to skip most IPMI values from one of the nodes in the unit.

When carrying out metrics collections on an iDataPlex unit, Bright Cluster Manager should work without any issues. However, it may be that due to the special paired node design of an iDataPlex unit, most IPMI metrics of one member of the pair are undetectable by the `sample_ipmi` script sampling on that particular node. The missing IPMI metrics can instead be retrieved from the second member in the pair (along with the IPMI metrics of the second member).

The output may thus look something like:

Example

```

[root@master01 ~]# cmsh
[master01]% device latestmetricdata node181 | grep Domain
Metric                               Value
-----

```

```

Domain_A_FP_Temp          23
Domain_A_Temp1            39
Domain_A_Temp2            37
Domain_Avg_Power          140
Domain_B_FP_Temp          24
Domain_B_Temp1            40
Domain_B_Temp2            37
[master01]% device latestmetricdata node182 | grep Domain
Metric                    Value
-----
Domain_A_FP_Temp          no data
Domain_A_Temp1            no data
Domain_A_Temp2            no data
Domain_Avg_Power          170
Domain_B_FP_Temp          no data
Domain_B_Temp1            no data
Domain_B_Temp2            no data
[master01]%

```

Because there are usually many iDataPlex units in the rack, the metrics retrieval response of each node pair in a unit should be checked for this behavior.

The issue can be dealt with by Bright Cluster Manager by modifying the configuration file for the `sample_ipmi` script in `/cm/local/apps/cmd/scripts/metrics/configfiles/sample_ipmi.conf`. Two parameters that can be configured there are `chassisContainsLeadNode` and `chassisContainsLeadNodeRegex`.

- Setting `chassisContainsLeadNode` to `on` forces the `sample_ipmi` script to treat the unit as an iDataPlex unit.

In particular:

- `auto` (recommended) means the unit is checked by the IPMI metric sample collection script for whether it behaves like an iDataPlex unit.
 - `on` means the unit is treated as an iDataPlex node pair, with one node being a lead node that has all the IPMI metrics.
 - `off` means the unit is treated as a non-iDataPlex node pair, with each node having normal behavior when retrieving IPMI metrics. This setting may need to be used in case the default value of `auto` ever falsely detects a node as part of an iDataPlex pair.
- The value of `chassisContainsLeadNodeRegex` can be set to a regular expression pattern that matches the system information pattern for the name, as obtained by `CMDaemon` for an iDataPlex unit (or similar clone unit). The pattern that it is matched against is the output of:

```
cmsh -c 'device ; sysinfo master | grep "^System Name"'
```

If the pattern matches, then the IPMI sample collection script assumes the unit behaves like an iDataPlex dual node pair. The missing IPMI data values are then looked for on the lead node.

The value of `chassisContainsLeadNodeRegex` is set to `iDataPlex` by default.

3

Bright Cluster Manager JSON API

This chapter gives an alphabetical list of the JSON API services and entities available for Bright Cluster Manager. The API reference documentation for all available services and entities is available on the head node at:

`/cm/local/apps/cmd/etc/htdocs/userportal/downloads/json/index.html`.

It can also be accessed via the user portal of the cluster by clicking on the JSON API documentation link in the documentation section of the home page (Section 12.7.3 of the *Administrator Manual*).

Some examples of JSON use are given in section 3.3

3.1 Services

- 3.1.1 **auth**
- 3.1.2 **ceph**
- 3.1.3 **cert**
- 3.1.4 **cloud**
- 3.1.5 **device**
- 3.1.6 **etcd**
- 3.1.7 **gui**
- 3.1.8 **hadoop**
- 3.1.9 **job**
- 3.1.10 **keyvalue**
- 3.1.11 **kube**
- 3.1.12 **lustre**
- 3.1.13 **main**
- 3.1.14 **mesos**
- 3.1.15 **mon**
- 3.1.16 **net**
- 3.1.17 **openstack**
- 3.1.18 **part**
- 3.1.19 **proc**
- 3.1.20 **prov**
- 3.1.21 **puppet**
- 3.1.22 **serv**
- 3.1.23 **session**

- 3.1.24 sync
- 3.1.25 test
- 3.1.26 ticket
- 3.1.27 user
- 3.1.28 zookeeper

3.2 Entities

- 3.2.1 AzureDataDisk
- 3.2.2 AzureDisk
- 3.2.3 AzureIntermediateStorage
- 3.2.4 AzureLocation
- 3.2.5 AzureManagedDiskParameters
- 3.2.6 AzureOSDisk
- 3.2.7 AzureProvider
- 3.2.8 AzurePublicIP
- 3.2.9 AzureSettings
- 3.2.10 AzureVMSize
- 3.2.11 BadEntityManagers
- 3.2.12 BasicResource
- 3.2.13 BeeGFSAdmonRole
- 3.2.14 BeeGFSClientRole
- 3.2.15 BeeGFSManagementRole
- 3.2.16 BeeGFSMetadataRole
- 3.2.17 BeeGFSStorageRole
- 3.2.18 BigDataAdditionalTool
- 3.2.19 BigDataAdvancedSettings
- 3.2.20 BigDataCassandra
- 3.2.21 BigDataFileSystemSettings
- 3.2.22 BigDataJobManagementSettings
- 3.2.23 BigDataLoggingSettings
- 3.2.24 BigDataSecurity
- 3.2.25 BigDataSpark
- 3.2.26 BillingHistory
- 3.2.27 BMCSettings
- 3.2.28 BootRole
- 3.2.29 BurnConfig
- 3.2.30 BurnStatus
- 3.2.31 BurnTestStatus
- 3.2.32 Category
- 3.2.33 Ceph
- 3.2.34 CephMDSRole
- 3.2.35 CephMGRRole
- 3.2.36 CephMonitorRole
- 3.2.37 CephOSDBlueStoreConfig
- 3.2.38 CephOSDConfig
- 3.2.39 CephOSDFileStoreConfig
- 3.2.40 CephOSDLegacyConfig

- 3.2.41 CephOSDPool
- 3.2.42 CephOSDRole
- 3.2.43 CephState
- 3.2.44 Certificate
- 3.2.45 CertificateRequest
- 3.2.46 CertificateSubjectName
- 3.2.47 Cgroup
- 3.2.48 CgroupController
- 3.2.49 CgroupControllerBlkio
- 3.2.50 CgroupControllerCpu
- 3.2.51 CgroupControllerCpuacct
- 3.2.52 CgroupControllerCpuset
- 3.2.53 CgroupControllerDevices
- 3.2.54 CgroupControllerFreezer
- 3.2.55 CgroupControllerHugetlb
- 3.2.56 CgroupControllerMemory
- 3.2.57 CgroupControllerNetcls
- 3.2.58 CgroupControllerNetprio
- 3.2.59 CgroupControllerNs
- 3.2.60 CgroupControllerPerf
- 3.2.61 CgroupRule
- 3.2.62 CgroupSupervisorRole
- 3.2.63 Chassis
- 3.2.64 ChronosRole
- 3.2.65 ClientUserData
- 3.2.66 CloudDirectorRole
- 3.2.67 CloudGatewayRole
- 3.2.68 CloudImage
- 3.2.69 CloudJobDescription
- 3.2.70 CloudJobSubmissionStatus
- 3.2.71 CloudNode
- 3.2.72 CloudPrivateCloud
- 3.2.73 CloudProvider
- 3.2.74 CloudRegion
- 3.2.75 CloudSettings
- 3.2.76 CloudStaticIP
- 3.2.77 CloudStorageActionData
- 3.2.78 CloudStorageNodeState
- 3.2.79 CloudType
- 3.2.80 CloudVirtualNetworkInterface
- 3.2.81 ClusterSetup
- 3.2.82 CMDaemonBackgroundTask
- 3.2.83 CMDaemonFailover
- 3.2.84 CMDaemonFailoverGroup
- 3.2.85 CMDaemonFailoverGroupStatus
- 3.2.86 CMDaemonFailoverPeer
- 3.2.87 CMDaemonFailoverStatus

- 3.2.88 **CMDaemonStatus**
- 3.2.89 **CMService**
- 3.2.90 **CMSubConfig**
- 3.2.91 **CMSubIntermediateStorage**
- 3.2.92 **ConfigFileVersion**
- 3.2.93 **ConfigSum**
- 3.2.94 **ConfigurationOverlay**
- 3.2.95 **Consolidator**
- 3.2.96 **ContainerdHostRole**
- 3.2.97 **ContainerInfo**
- 3.2.98 **CustomizationEntry**
- 3.2.99 **CustomizationFile**
- 3.2.100 **DellClustat**
- 3.2.101 **DellClustatGroup**
- 3.2.102 **DellClustatNode**
- 3.2.103 **DellDiskGroupInfo**
- 3.2.104 **DellPhysicalDiskDriveInfo**
- 3.2.105 **DellRAIDControllerInfo**
- 3.2.106 **DellSettings**
- 3.2.107 **DellSettingsFirmware**
- 3.2.108 **DellSettingsNicDevice**
- 3.2.109 **DellStorageInfo**
- 3.2.110 **DellVirtualDiskInfo**
- 3.2.111 **Device**
- 3.2.112 **DevStatus**
- 3.2.113 **DiskAssertion**
- 3.2.114 **DiskDevice**
- 3.2.115 **DiskInfo**
- 3.2.116 **DiskPartition**
- 3.2.117 **DiskRaid**
- 3.2.118 **DiskSetup**
- 3.2.119 **DiskVolume**
- 3.2.120 **DiskVolumeGroup**
- 3.2.121 **DockerHostRole**
- 3.2.122 **DockerRegistryFilesystemStorageDriver**
- 3.2.123 **DockerRegistryInmemoryStorageDriver**
- 3.2.124 **DockerRegistryRole**
- 3.2.125 **DockerRegistryStorageDriver**
- 3.2.126 **DockerStorageBackend**
- 3.2.127 **DockerStorageDeviceMapperBackend**
- 3.2.128 **DrainAction**
- 3.2.129 **DrainResult**
- 3.2.130 **EC2AMI**
- 3.2.131 **EC2AvailabilityZone**
- 3.2.132 **EC2EBSStorage**
- 3.2.133 **EC2EphemeralStorage**
- 3.2.134 **EC2PrivateCloud**

- 3.2.135 EC2Provider
- 3.2.136 EC2Region
- 3.2.137 EC2RegionAMI
- 3.2.138 EC2Settings
- 3.2.139 EC2StaticIP
- 3.2.140 EC2Storage
- 3.2.141 EC2Type
- 3.2.142 EC2VirtualNetworkInterface
- 3.2.143 ElasticSearchRole
- 3.2.144 EntityManagersMD5
- 3.2.145 EtcCluster
- 3.2.146 EtcHostRole
- 3.2.147 EthernetSwitch
- 3.2.148 FailoverRole
- 3.2.149 FakeJob
- 3.2.150 FakeJobQueue
- 3.2.151 FakeJobQueueStat
- 3.2.152 FakeWlmClientRole
- 3.2.153 FakeWlmServerRole
- 3.2.154 FileInfo
- 3.2.155 FileSyncConfig
- 3.2.156 FileSyncStatus
- 3.2.157 FlannelConfigurationRole
- 3.2.158 FlannelHostRole
- 3.2.159 FlannelNetworkingBackend
- 3.2.160 FlannelNetworkingUdpBackend
- 3.2.161 FlannelNetworkingVxLanBackend
- 3.2.162 FSExport
- 3.2.163 FSMount
- 3.2.164 FSPart
- 3.2.165 FSPartAssociation
- 3.2.166 FSPartBasicAssociation
- 3.2.167 FSPartProviderAssociation
- 3.2.168 GaleraRole
- 3.2.169 GenericDevice
- 3.2.170 GenericResource
- 3.2.171 GPUInfo
- 3.2.172 GPUSettings
- 3.2.173 GpuUnit
- 3.2.174 GPUUnitInfo
- 3.2.175 GridEngineJob
- 3.2.176 GridEngineJobQueue
- 3.2.177 GridEngineJobQueueStat
- 3.2.178 GridEngineParallelEnvironment
- 3.2.179 Group
- 3.2.180 GuiCephOsdPoolInfo
- 3.2.181 GuiCephOverview

3.2.182 **GuiCephPgInfo**
3.2.183 **GuiClusterOverview**
3.2.184 **GuiCompleteOpenStackOverview**
3.2.185 **GuiDiskUsage**
3.2.186 **GuiGpuUnitOverview**
3.2.187 **GuiHadoopHDFSDetailHBase**
3.2.188 **GuiHadoopHDFSDetailHDFS**
3.2.189 **GuiHadoopHDFSDetailMapreduce**
3.2.190 **GuiHadoopHDFSDetailSpark**
3.2.191 **GuiHadoopHDFSDetailYarn**
3.2.192 **GuiHadoopHDFSDetailZooKeeper**
3.2.193 **GuiHadoopHDFSOverview**
3.2.194 **GuiJob**
3.2.195 **GuiKubeClusterOverview**
3.2.196 **GuiNetworkInterface**
3.2.197 **GuiNodeOverview**
3.2.198 **GuiNodeStatus**
3.2.199 **GuiOpenStackOverview**
3.2.200 **GuiOpenStackProjectOverview**
3.2.201 **GuiOpenStackTenantOverview**
3.2.202 **GuiPDUBank**
3.2.203 **GuiPDUOutlet**
3.2.204 **GuiPDUOverview**
3.2.205 **GuiSwitchOverview**
3.2.206 **GuiSwitchPort**
3.2.207 **GuiWorkload**
3.2.208 **HadoopAccumuloMasterHDFSConfiguration**
3.2.209 **HadoopAccumuloMasterRole**
3.2.210 **HadoopAccumuloTabletHDFSConfiguration**
3.2.211 **HadoopAccumuloTabletRole**
3.2.212 **HadoopAlluxioMasterHDFSConfiguration**
3.2.213 **HadoopAlluxioMasterRole**
3.2.214 **HadoopAlluxioWorkerHDFSConfiguration**
3.2.215 **HadoopAlluxioWorkerRole**
3.2.216 **HadoopBaseConfiguration**
3.2.217 **HadoopCassandraHDFSConfiguration**
3.2.218 **HadoopCassandraRole**
3.2.219 **HadoopDataNodeHDFSConfiguration**
3.2.220 **HadoopDataNodeRole**
3.2.221 **HadoopDrillHDFSConfiguration**
3.2.222 **HadoopDrillRole**
3.2.223 **HadoopFlinkJobManagerHDFSConfiguration**
3.2.224 **HadoopFlinkJobManagerRole**
3.2.225 **HadoopFlinkTaskManagerHDFSConfiguration**
3.2.226 **HadoopFlinkTaskManagerRole**
3.2.227 **HadoopHBaseClientHDFSConfiguration**
3.2.228 **HadoopHBaseClientRole**

- 3.2.229 HadoopHBaseServerHDFSConfiguration
- 3.2.230 HadoopHBaseServerRole
- 3.2.231 HadoopHDFS
- 3.2.232 HadoopHiveHDFSConfiguration
- 3.2.233 HadoopHiveRole
- 3.2.234 HadoopJob
- 3.2.235 HadoopJobQueue
- 3.2.236 HadoopJobQueueStat
- 3.2.237 HadoopJobTrackerHDFSConfiguration
- 3.2.238 HadoopJobTrackerRole
- 3.2.239 HadoopJournalHDFSConfiguration
- 3.2.240 HadoopJournalRole
- 3.2.241 HadoopKafkaServerHDFSConfiguration
- 3.2.242 HadoopKafkaServerRole
- 3.2.243 HadoopKMServerHDFSConfiguration
- 3.2.244 HadoopKMServerRole
- 3.2.245 HadoopNameNodeHDFSConfiguration
- 3.2.246 HadoopNameNodeRole
- 3.2.247 HadoopNFSGatewayHDFSConfiguration
- 3.2.248 HadoopNFSGatewayRole
- 3.2.249 HadoopPigHDFSConfiguration
- 3.2.250 HadoopPigRole
- 3.2.251 HadoopSecondaryNameNodeHDFSConfiguration
- 3.2.252 HadoopSecondaryNameNodeRole
- 3.2.253 HadoopSparkMasterHDFSConfiguration
- 3.2.254 HadoopSparkMasterRole
- 3.2.255 HadoopSparkWorkerHDFSConfiguration
- 3.2.256 HadoopSparkWorkerRole
- 3.2.257 HadoopSparkYARNHDFSConfiguration
- 3.2.258 HadoopSparkYARNRole
- 3.2.259 HadoopSqoopHDFSConfiguration
- 3.2.260 HadoopSqoopRole
- 3.2.261 HadoopStormNimbusHDFSConfiguration
- 3.2.262 HadoopStormNimbusRole
- 3.2.263 HadoopStormSupervisorHDFSConfiguration
- 3.2.264 HadoopStormSupervisorRole
- 3.2.265 HadoopTaskTrackerHDFSConfiguration
- 3.2.266 HadoopTaskTrackerRole
- 3.2.267 HadoopYARNClientHDFSConfiguration
- 3.2.268 HadoopYARNClientRole
- 3.2.269 HadoopYARNServerHDFSConfiguration
- 3.2.270 HadoopYARNServerRole
- 3.2.271 HadoopZeppelinHDFSConfiguration
- 3.2.272 HadoopZeppelinRole
- 3.2.273 HadoopZooKeeperHDFSConfiguration
- 3.2.274 HadoopZooKeeperRole
- 3.2.275 HAProxyEntry

3.2.276 HAProxyEntryBind
3.2.277 HAProxyRole
3.2.278 HAProxyServer
3.2.279 HAProxySharedSettings
3.2.280 IBSwitch
3.2.281 IPCPerm
3.2.282 IPResource
3.2.283 Job
3.2.284 JobInfo
3.2.285 JobInfoStatistics
3.2.286 JobQueue
3.2.287 JobQueuePlaceholder
3.2.288 JobQueueStat
3.2.289 KeepalivedEntry
3.2.290 KeepalivedRole
3.2.291 KernelModule
3.2.292 KeyValuePair
3.2.293 KibanaRole
3.2.294 KubeCluster
3.2.295 KubePodInfo
3.2.296 KubernetesApiServerRole
3.2.297 KubernetesControllerRole
3.2.298 KubernetesDashboardRole
3.2.299 KubernetesDNSRole
3.2.300 KubernetesHeapsterRole
3.2.301 KubernetesNodeRole
3.2.302 KubernetesProxyRole
3.2.303 KubernetesSchedulerRole
3.2.304 KubeUserPolicy
3.2.305 LicenseInfo
3.2.306 LoginRole
3.2.307 LogstashForwarderRole
3.2.308 LogstashServerCustomFilter
3.2.309 LogstashServerCustomListener
3.2.310 LogstashServerCustomOutput
3.2.311 LogstashServerElasticOutput
3.2.312 LogstashServerFilter
3.2.313 LogstashServerListener
3.2.314 LogstashServerLocalFileListener
3.2.315 LogstashServerLumberjackListener
3.2.316 LogstashServerOutput
3.2.317 LogstashServerRole
3.2.318 LogstashServerRSyslogFilter
3.2.319 LogstashServerRSyslogListener
3.2.320 LogstashServerStdOutput
3.2.321 LSFBaseJob
3.2.322 LSFBaseJobQueue

- 3.2.323 LSFBaseJobQueueStat
- 3.2.324 LSFCgroupsSettings
- 3.2.325 LSFClientRole
- 3.2.326 LSFJob
- 3.2.327 LSFJobQueue
- 3.2.328 LSFJobQueueStat
- 3.2.329 LSFServerRole
- 3.2.330 LustreAlert
- 3.2.331 LustreClientMount
- 3.2.332 LustreFileSystem
- 3.2.333 LustreFileSystemTarget
- 3.2.334 LustreLog
- 3.2.335 LustreOverview
- 3.2.336 LustreServer
- 3.2.337 LustreServerProfile
- 3.2.338 LustreSettings
- 3.2.339 LustreTargetMap
- 3.2.340 LustreUser
- 3.2.341 LustreVolume
- 3.2.342 LustreVolumeNode
- 3.2.343 MarathonRole
- 3.2.344 MasterNode
- 3.2.345 MasterRole
- 3.2.346 MemcachedRole
- 3.2.347 MemoryInfo
- 3.2.348 MesosCluster
- 3.2.349 MesosDNSRole
- 3.2.350 MesosMasterRole
- 3.2.351 MesosProxyRole
- 3.2.352 MesosResourceUsage
- 3.2.353 MesosSlaveRole
- 3.2.354 MICHostRole
- 3.2.355 MICInfo
- 3.2.356 MICNode
- 3.2.357 MICNodeCategory
- 3.2.358 MICOverlay
- 3.2.359 MICSettings
- 3.2.360 MonitoringAction
- 3.2.361 MonitoringActionRunData
- 3.2.362 MonitoringCacheSubSystemInfo
- 3.2.363 MonitoringCategoryListExecutionFilter
- 3.2.364 MonitoringCompareExpression
- 3.2.365 MonitoringConsolidator
- 3.2.366 MonitoringDataCacheSubSystemInfo
- 3.2.367 MonitoringDataProducer
- 3.2.368 MonitoringDataProducerAggregateNode
- 3.2.369 MonitoringDataProducerAlertLevel

3.2.370 MonitoringDataProducerCGroup
3.2.371 MonitoringDataProducerClusterTotal
3.2.372 MonitoringDataProducerCMDaemonState
3.2.373 MonitoringDataProducerDeviceState
3.2.374 MonitoringDataProducerEC2SpotPrices
3.2.375 MonitoringDataProducerEthernetSwitch
3.2.376 MonitoringDataProducerFuture
3.2.377 MonitoringDataProducerGalera
3.2.378 MonitoringDataProducerGenerator
3.2.379 MonitoringDataProducerGPU
3.2.380 MonitoringDataProducerInternal
3.2.381 MonitoringDataProducerJob
3.2.382 MonitoringDataProducerJobQueue
3.2.383 MonitoringDataProducerLua
3.2.384 MonitoringDataProducerMonitoringSystem
3.2.385 MonitoringDataProducerOpenStack
3.2.386 MonitoringDataProducerOpenStackHealth
3.2.387 MonitoringDataProducerPerpetual
3.2.388 MonitoringDataProducerPowerDistributionUnit
3.2.389 MonitoringDataProducerProcMemInfo
3.2.390 MonitoringDataProducerProcMount
3.2.391 MonitoringDataProducerProcNetDev
3.2.392 MonitoringDataProducerProcNetSnmpp
3.2.393 MonitoringDataProducerProcPidStat
3.2.394 MonitoringDataProducerProcStat
3.2.395 MonitoringDataProducerProcVMStat
3.2.396 MonitoringDataProducerRackSensor
3.2.397 MonitoringDataProducerScript
3.2.398 MonitoringDataProducerSingleLineHealthCheckScript
3.2.399 MonitoringDataProducerSingleLineMetricScript
3.2.400 MonitoringDataProducerSingleLineScript
3.2.401 MonitoringDataProducerSmart
3.2.402 MonitoringDataProducerSysBlockStat
3.2.403 MonitoringDataProducerSysInfo
3.2.404 MonitoringDataProducerTest
3.2.405 MonitoringDataProducerTrustedTool
3.2.406 MonitoringDataProducerUserCount
3.2.407 MonitoringDeviceStateSubSystemInfo
3.2.408 MonitoringDrainAction
3.2.409 MonitoringEmailAction
3.2.410 MonitoringEventAction
3.2.411 MonitoringExecutionFilter
3.2.412 MonitoringExecutionMultiplexer
3.2.413 MonitoringExpression
3.2.414 MonitoringGroupedExpression
3.2.415 MonitoringHealthOverview
3.2.416 MonitoringImageUpdateAction

- 3.2.417 **MonitoringJobMetricSettings**
- 3.2.418 **MonitoringLuaExecutionFilter**
- 3.2.419 **MonitoringLuaExecutionMultiplexer**
- 3.2.420 **MonitoringMeasurable**
- 3.2.421 **MonitoringMeasurableEnum**
- 3.2.422 **MonitoringMeasurableHealthCheck**
- 3.2.423 **MonitoringMeasurableMetric**
- 3.2.424 **MonitoringNodeListExecutionFilter**
- 3.2.425 **MonitoringOverlayListExecutionFilter**
- 3.2.426 **MonitoringPlotterSubSystemInfo**
- 3.2.427 **MonitoringPowerAction**
- 3.2.428 **MonitoringPowerOffAction**
- 3.2.429 **MonitoringPowerOnAction**
- 3.2.430 **MonitoringPowerResetAction**
- 3.2.431 **MonitoringRebootAction**
- 3.2.432 **MonitoringReplicateConfiguration**
- 3.2.433 **MonitoringReplicateSource**
- 3.2.434 **MonitoringReplicateSubSystemInfo**
- 3.2.435 **MonitoringResourceExecutionFilter**
- 3.2.436 **MonitoringResourceExecutionMultiplexer**
- 3.2.437 **MonitoringRole**
- 3.2.438 **MonitoringScriptAction**
- 3.2.439 **MonitoringServiceAction**
- 3.2.440 **MonitoringServiceRestartAction**
- 3.2.441 **MonitoringServiceStartAction**
- 3.2.442 **MonitoringServiceStopAction**
- 3.2.443 **MonitoringServiceSubSystemInfo**
- 3.2.444 **MonitoringShutdownAction**
- 3.2.445 **MonitoringStorageSubSystemInfo**
- 3.2.446 **MonitoringSubSystemInfo**
- 3.2.447 **MonitoringTrigger**
- 3.2.448 **MonitoringTypeExecutionFilter**
- 3.2.449 **MonitoringTypeExecutionMultiplexer**
- 3.2.450 **MonitoringUndrainAction**
- 3.2.451 **MsgQueue**
- 3.2.452 **MyrinetSwitch**
- 3.2.453 **Network**
- 3.2.454 **NetworkAliasInterface**
- 3.2.455 **NetworkBmcInterface**
- 3.2.456 **NetworkBondInterface**
- 3.2.457 **NetworkBridgeInterface**
- 3.2.458 **NetworkInterface**
- 3.2.459 **NetworkNetMapInterface**
- 3.2.460 **NetworkPhysicalInterface**
- 3.2.461 **NetworkTunnelInterface**
- 3.2.462 **NetworkVLANInterface**
- 3.2.463 **NewNode**

3.2.464 NginxRole
3.2.465 Node
3.2.466 NodeCategory
3.2.467 NodeGroup
3.2.468 OpenLavaCgroupsSettings
3.2.469 OpenLavaClientRole
3.2.470 OpenLavaJob
3.2.471 OpenLavaJobQueue
3.2.472 OpenLavaJobQueueStat
3.2.473 OpenLavaServerRole
3.2.474 OpenStack
3.2.475 OpenStackApiAgent
3.2.476 OpenStackApiDomain
3.2.477 OpenStackApiEndpoint
3.2.478 OpenStackApiEntity
3.2.479 OpenStackApiFlavor
3.2.480 OpenStackApiFloatingIP
3.2.481 OpenStackApiGroup
3.2.482 OpenStackApiHostAggregate
3.2.483 OpenStackApiHypervisor
3.2.484 OpenStackApiImage
3.2.485 OpenStackApiNetwork
3.2.486 OpenStackApiPort
3.2.487 OpenStackApiProject
3.2.488 OpenStackApiRole
3.2.489 OpenStackApiRoleAssignment
3.2.490 OpenStackApiRouter
3.2.491 OpenStackApiSecurityGroup
3.2.492 OpenStackApiServer
3.2.493 OpenStackApiService
3.2.494 OpenStackApiStack
3.2.495 OpenStackApiSubnet
3.2.496 OpenStackApiUser
3.2.497 OpenStackApiVolume
3.2.498 OpenStackApiVolumeSnapshot
3.2.499 OpenStackApiVolumeType
3.2.500 OpenStackAuthBackend
3.2.501 OpenStackAuthBackendHybrid
3.2.502 OpenStackAuthBackendLDAP
3.2.503 OpenStackAuthBackendLDAPGroupSettings
3.2.504 OpenStackAuthBackendLDAPProjectSettings
3.2.505 OpenStackAuthBackendLDAPRoleSettings
3.2.506 OpenStackAuthBackendLDAPUserSettings
3.2.507 OpenStackAuthBackendSQL
3.2.508 OpenStackBareMetalApiRole
3.2.509 OpenStackBareMetalConductorRole
3.2.510 OpenStackBareMetalDiscoverdDNSSMasqRole

- 3.2.511 OpenStackBareMetalDiscoverdRole
- 3.2.512 OpenStackBlockStorage
- 3.2.513 OpenStackComputeApiEC2Role
- 3.2.514 OpenStackComputeApiMetadataRole
- 3.2.515 OpenStackComputeApiRole
- 3.2.516 OpenStackComputeConductorRole
- 3.2.517 OpenStackComputeRole
- 3.2.518 OpenStackComputeSchedulerRole
- 3.2.519 OpenStackComputeVNCProxyRole
- 3.2.520 OpenStackDashboardRole
- 3.2.521 OpenStackDataProcessingApiRole
- 3.2.522 OpenStackDBaaSRole
- 3.2.523 OpenStackDefaultUserRole
- 3.2.524 OpenStackIdentityApiRole
- 3.2.525 OpenStackImageApiRole
- 3.2.526 OpenStackImageBackend
- 3.2.527 OpenStackImageBackendCeph
- 3.2.528 OpenStackImageBackendFS
- 3.2.529 OpenStackImageRegistryRole
- 3.2.530 OpenStackMessageQueueServerRole
- 3.2.531 OpenStackNetworkApiRole
- 3.2.532 OpenStackNetworkDHCPAgentRole
- 3.2.533 OpenStackNetworkL3AgentRole
- 3.2.534 OpenStackNetworkMetadataAgentRole
- 3.2.535 OpenStackNetworkOVSAgentRole
- 3.2.536 OpenStackNovalImageBackend
- 3.2.537 OpenStackNovalImageBackendCeph
- 3.2.538 OpenStackNovalImageBackendCow
- 3.2.539 OpenStackObjectAccountRole
- 3.2.540 OpenStackObjectApiRole
- 3.2.541 OpenStackObjectContainerRole
- 3.2.542 OpenStackObjectStoreRole
- 3.2.543 OpenStackOrchestrationApiRole
- 3.2.544 OpenStackOrchestrationRole
- 3.2.545 OpenStackSettings
- 3.2.546 OpenStackSettingsAdvanced
- 3.2.547 OpenStackSettingsAuthentication
- 3.2.548 OpenStackSettingsCMDaemonInteractions
- 3.2.549 OpenStackSettingsCollection
- 3.2.550 OpenStackSettingsCompute
- 3.2.551 OpenStackSettingsCredentials
- 3.2.552 OpenStackSettingsDatabase
- 3.2.553 OpenStackSettingsLogging
- 3.2.554 OpenStackSettingsNetworking
- 3.2.555 OpenStackSettingsPorts
- 3.2.556 OpenStackSettingsQuota
- 3.2.557 OpenStackSettingsUserPortal

3.2.558 **OpenStackSettingsUsers**
3.2.559 **OpenStackStorage**
3.2.560 **OpenStackTelemetryAgentCentralRole**
3.2.561 **OpenStackTelemetryAgentComputeRole**
3.2.562 **OpenStackTelemetryAgentIpmiRole**
3.2.563 **OpenStackTelemetryAgentNotificationRole**
3.2.564 **OpenStackTelemetryAlarmEvaluatorRole**
3.2.565 **OpenStackTelemetryAlarmNotifierRole**
3.2.566 **OpenStackTelemetryApiRole**
3.2.567 **OpenStackTelemetryCollectorRole**
3.2.568 **OpenStackUserRole**
3.2.569 **OpenStackVolumeApiRole**
3.2.570 **OpenStackVolumeBackend**
3.2.571 **OpenStackVolumeBackend3PAR**
3.2.572 **OpenStackVolumeBackendCeph**
3.2.573 **OpenStackVolumeBackendDellStorageCenter**
3.2.574 **OpenStackVolumeBackendGPFS**
3.2.575 **OpenStackVolumeBackendNetApp**
3.2.576 **OpenStackVolumeBackendNFS**
3.2.577 **OpenStackVolumeBackendSolidFire**
3.2.578 **OpenStackVolumeBackupBackend**
3.2.579 **OpenStackVolumeBackupBackendCeph**
3.2.580 **OpenStackVolumeBackupRole**
3.2.581 **OpenStackVolumeRole**
3.2.582 **OpenStackVolumeSchedulerRole**
3.2.583 **OpenvSwitchRole**
3.2.584 **OsapiPortIP**
3.2.585 **OsapiSecurityGroupRule**
3.2.586 **OsapiStackResource**
3.2.587 **OsapiSubnetAllocationPool**
3.2.588 **OSService**
3.2.589 **OSServiceArray**
3.2.590 **OSServiceConfig**
3.2.591 **ParentJob**
3.2.592 **Partition**
3.2.593 **PBSJob**
3.2.594 **PBSJobQueue**
3.2.595 **PBSJobQueueStat**
3.2.596 **PbsProCgroupsSettings**
3.2.597 **PbsProClientRole**
3.2.598 **PbsProJob**
3.2.599 **PbsProJobQueue**
3.2.600 **PbsProJobQueueStat**
3.2.601 **PbsProServerRole**
3.2.602 **PDUPort**
3.2.603 **PhysicalNode**
3.2.604 **PowerDistributionUnit**

- 3.2.605 **PowerOperation**
- 3.2.606 **PowerStatus**
- 3.2.607 **Process**
- 3.2.608 **Processor**
- 3.2.609 **Profile**
- 3.2.610 **ProgramRunnerInput**
- 3.2.611 **ProgramRunnerKill**
- 3.2.612 **ProgramRunnerOutput**
- 3.2.613 **ProgramRunnerStatus**
- 3.2.614 **ProvisioningNodeStatus**
- 3.2.615 **ProvisioningProcessorJob**
- 3.2.616 **ProvisioningRequestStatus**
- 3.2.617 **ProvisioningRole**
- 3.2.618 **ProvisioningStatus**
- 3.2.619 **Puppet**
- 3.2.620 **PuppetApplyOnNodeRequest**
- 3.2.621 **PuppetApplyResult**
- 3.2.622 **PuppetApplySession**
- 3.2.623 **PuppetClass**
- 3.2.624 **PuppetClassDeclaration**
- 3.2.625 **PuppetClassFactory**
- 3.2.626 **PuppetConfigurationEntry**
- 3.2.627 **PuppetForgeInstallation**
- 3.2.628 **PuppetForgePagination**
- 3.2.629 **PuppetForgeSearchRequest**
- 3.2.630 **PuppetForgeSearchResult**
- 3.2.631 **PuppetKeyValuePair**
- 3.2.632 **PuppetModule**
- 3.2.633 **PuppetModuleDependency**
- 3.2.634 **PuppetModuleRelease**
- 3.2.635 **PuppetOperatingSystemSupport**
- 3.2.636 **PuppetParameterFactory**
- 3.2.637 **PuppetRescanResult**
- 3.2.638 **PuppetResourceDeclaration**
- 3.2.639 **PuppetRole**
- 3.2.640 **PuppetRoleChange**
- 3.2.641 **Rack**
- 3.2.642 **RackPosition**
- 3.2.643 **RackSensor**
- 3.2.644 **RadosGatewayRole**
- 3.2.645 **RemoteNodeInstallerInteraction**
- 3.2.646 **RemoteSetupExecution**
- 3.2.647 **ResourcePool**
- 3.2.648 **ResourcePoolStatus**
- 3.2.649 **Role**
- 3.2.650 **S3BucketIntermediateStorage**
- 3.2.651 **ScaleDynamicNodesProvider**

3.2.652 ScaleEngine
3.2.653 ScaleHpcEngine
3.2.654 ScaleHpcQueueTracker
3.2.655 ScaleMesosEngine
3.2.656 ScaleMesosLoadTracker
3.2.657 ScalePendingWorkload
3.2.658 ScaleResourceProvider
3.2.659 ScaleServerRole
3.2.660 ScaleStaticNodesProvider
3.2.661 ScaleTracker
3.2.662 Semaphore
3.2.663 Sensor
3.2.664 Session
3.2.665 SGEClientRole
3.2.666 SGEJob
3.2.667 SGEJobQueue
3.2.668 SGEJobQueueStat
3.2.669 SGEParallelEnvironment
3.2.670 SGEServerRole
3.2.671 SharedMemory
3.2.672 SlaveNode
3.2.673 SlurmCgroupsSettings
3.2.674 SlurmClientRole
3.2.675 SlurmJob
3.2.676 SlurmJobQueue
3.2.677 SlurmJobQueueStat
3.2.678 SlurmServerRole
3.2.679 SoftwareImage
3.2.680 SoftwareImageProxy
3.2.681 SoftwareImageRevisionInfo
3.2.682 StandaloneMonitoredEntity
3.2.683 StaticRoute
3.2.684 StorageNodePolicy
3.2.685 StorageRole
3.2.686 StringListObject
3.2.687 SubnetManagerRole
3.2.688 SubSystemInfo
3.2.689 Switch
3.2.690 SwitchPort
3.2.691 SysInfoCollector
3.2.692 Ticket
3.2.693 TorqueCgroupsSettings
3.2.694 TorqueClientRole
3.2.695 TorqueJob
3.2.696 TorqueJobQueue
3.2.697 TorqueJobQueueStat
3.2.698 TorqueServerRole

- 3.2.699 UCSAdaptorEthCompQueueProfile
- 3.2.700 UCSAdaptorEthGenProfile
- 3.2.701 UCSAdaptorEthInterruptProfile
- 3.2.702 UCSAdaptorEthOffloadProfile
- 3.2.703 UCSAdaptorEthRecvQueueProfile
- 3.2.704 UCSAdaptorEthUSNICProfile
- 3.2.705 UCSAdaptorEthWorkQueueProfile
- 3.2.706 UCSAdaptorExtEthIf
- 3.2.707 UCSAdaptorExtIpv6RssHashProfile
- 3.2.708 UCSAdaptorFcCdbWorkQueueProfile
- 3.2.709 UCSAdaptorFcErrorRecoveryProfile
- 3.2.710 UCSAdaptorFcGenProfile
- 3.2.711 UCSAdaptorFcInterruptProfile
- 3.2.712 UCSAdaptorFcPortFLogiProfile
- 3.2.713 UCSAdaptorFcPortPLogiProfile
- 3.2.714 UCSAdaptorFcPortProfile
- 3.2.715 UCSAdaptorFcRecvQueueProfile
- 3.2.716 UCSAdaptorFcWorkQueueProfile
- 3.2.717 UCSAdaptorHostEthIf
- 3.2.718 UCSAdaptorHostFclIf
- 3.2.719 UCSAdaptorIpv4RssHashProfile
- 3.2.720 UCSAdaptorIpv6RssHashProfile
- 3.2.721 UCSAdaptorPortProfiles
- 3.2.722 UCSAdaptorRssProfile
- 3.2.723 UCSBase
- 3.2.724 UCSBiosBootDev
- 3.2.725 UCSBiosBootDevGrp
- 3.2.726 UCSBiosSettings
- 3.2.727 UCSBiosVfAdjacentCacheLinePrefetch
- 3.2.728 UCSBiosVfAltitude
- 3.2.729 UCSBiosVfASPMSupport
- 3.2.730 UCSBiosVfConsoleRedirection
- 3.2.731 UCSBiosVfCoreMultiProcessing
- 3.2.732 UCSBiosVfCPUEnergyPerformance
- 3.2.733 UCSBiosVfCPUFrequencyFloor
- 3.2.734 UCSBiosVfCPUPerformance
- 3.2.735 UCSBiosVfCPUPowerManagement
- 3.2.736 UCSBiosVfDCUPrefetch
- 3.2.737 UCSBiosVfDemandScrub
- 3.2.738 UCSBiosVfDirectCacheAccess
- 3.2.739 UCSBiosVfDRAMClockThrottling
- 3.2.740 UCSBiosVfDramRefreshRate
- 3.2.741 UCSBiosVfEnhancedIntelSpeedStepTech
- 3.2.742 UCSBiosVfExecuteDisableBit
- 3.2.743 UCSBiosVfFRB2Enable
- 3.2.744 UCSBiosVfHardwarePrefetch
- 3.2.745 UCSBiosVfIntelHyperThreadingTech

3.2.746 UCSBiosVfIntelTurboBoostTech
3.2.747 UCSBiosVfIntelVirtualizationTechnology
3.2.748 UCSBiosVfIntelVTFForDirectedIO
3.2.749 UCSBiosVfLegacyUSBSupport
3.2.750 UCSBiosVfLOMPortOptionROM
3.2.751 UCSBiosVfLvDIMMSupport
3.2.752 UCSBiosVfMemoryInterleave
3.2.753 UCSBiosVfMemoryMappedIOAbove4GB
3.2.754 UCSBiosVfNUMAOptimized
3.2.755 UCSBiosVfOnboardStorage
3.2.756 UCSBiosVfOnboardStorageSWStack
3.2.757 UCSBiosVfOSBootWatchdogTimer
3.2.758 UCSBiosVfOSBootWatchdogTimerPolicy
3.2.759 UCSBiosVfOSBootWatchdogTimerTimeout
3.2.760 UCSBiosVfPatrolScrub
3.2.761 UCSBiosVfPCIOptionROMs
3.2.762 UCSBiosVfPCISlotOptionROMEnable
3.2.763 UCSBiosVfProcessorC1E
3.2.764 UCSBiosVfProcessorC6Report
3.2.765 UCSBiosVfPStateCoordType
3.2.766 UCSBiosVfQPIConfig
3.2.767 UCSBiosVfSelectMemoryRASConfiguration
3.2.768 UCSBiosVfTPMSupport
3.2.769 UCSBiosVfUCSMBootOrderRuleControl
3.2.770 UCSBiosVfUSBEmulation
3.2.771 UCSBiosVfUSBPortsConfig
3.2.772 UCSBiosVfVgaPriority
3.2.773 UCSCommNtpProvider
3.2.774 UCSCommSyslog
3.2.775 UCSCommSyslogClient
3.2.776 UCSEquipmentIndicatorLed
3.2.777 UCSEquipmentLocatorLed
3.2.778 UCSFaultInst
3.2.779 UCSFirmwareRunning
3.2.780 UCSInfo
3.2.781 UCSLogs
3.2.782 UCSLsbootDef
3.2.783 UCSLsbootEfi
3.2.784 UCSLsbootLan
3.2.785 UCSLsbootStorage
3.2.786 UCSLsbootVirtualMedia
3.2.787 UCSStatus
3.2.788 UGECgroupsSettings
3.2.789 UGEClientRole
3.2.790 UGEJob
3.2.791 UGEJobQueue
3.2.792 UGEJobQueueStat

- 3.2.793 UGEParallelEnvironment**
- 3.2.794 UGEServerRole**
- 3.2.795 User**
- 3.2.796 Validation**
- 3.2.797 VersionInfo**
- 3.2.798 VirtualNode**
- 3.2.799 VirtualNodeSettings**
- 3.2.800 VirtualSMPNode**
- 3.2.801 VScaleMPSSettings**
- 3.2.802 VsmptSettings**
- 3.2.803 WillChange**
- 3.2.804 WlmCgroupsSettings**
- 3.2.805 XeonPhiSettings**
- 3.2.806 ZooKeeperCluster**
- 3.2.807 ZooKeeperHostRole**

3.3 JSON Examples

complete.sh

```
#!/bin/bash

URL=https://localhost:2081/json/
user=koen
pass=koen

echo "==== login ====="
curl -c curl.cookiest.txt -i -k -X POST -d '{"service":"login", \
"username":"koen", "password":"'${pass}'"}' $URL; echo
echo "==== master ====="
curl --cookie curl.cookiest.txt -i -k -X POST -d '{"service":"cm\
device", "call":"getNode", "arg":"master"}' $URL; echo
echo "==== logout ====="
curl --cookie curl.cookiest.txt -i -k -X POST -d '{"service":"lo\
gout"}' $URL; echo
echo "==== denied ====="
curl --cookie curl.cookiest.txt -i -k -X POST -d '{"service":"cm\
device", "call":"getNode", "arg":"master"}' $URL; echo
rm -f curl.cookiest.txt

echo "==== cert ====="
curl --cert $HOME/.cm/admin.pem --key $HOME/.cm/admin.key -i -k \
-X POST -d '{"service":"cmdevice", "call":"getNode", "arg":"master\
"}' $URL; echo
```

curl.sh

```
#!/bin/bash

source url
if [ -z "$1" ]; then
    pass=koen
else
```

```

    pass=$1
fi
read -p "pass: " -s -a $pass

curl -c curl.cookie.txt -i -k -X POST -d '{"service":"login", \
"username":"koen", "password":"' $pass'}' $URL

# curl --cookie curl.cookie.txt -i -k -X POST -d '{"service": "\
cmsession", "call": "getLastEvents", "args": [0, 256]}' $URL

curl --cookie curl.cookie.txt -i -k -X POST -d '{"service": "cm\
main", "call": "getProfile"}' $URL
curl --cookie curl.cookie.txt -i -k -X POST -d '{"service": "cm\
main", "call": "getSubjectName"}' $URL

```

devices.sh

```

#!/bin/bash
source url

if [ "$1" == "gzip" ]; then
    wget --load-cookies cookie.txt --header='Accept-Encoding: gzip\
' --no-check-certificate --server-response -qO- $URL --post-dat\
a='{"service": "cmdevice", "call": "getDevices"}'
else
    wget --load-cookies cookie.txt --no-check-certificate --server\
-response -qO- $URL --post-data='{"service": "cmdevice", "call": "g\
etDevices"}'
fi

```

loadone.sh

```

#!/bin/bash
source url

# not perfect but gets the job done
function jsonval {
    temp=`echo $json | sed 's/\\\\\\\\/\\/g' | sed 's/[{}]/g' | awk\
-v k="text" '{n=split($0,a,","); for (i=1; i<=n; i++) print a[i\
]}' | sed 's/\"\":\\\"/\"/g' | sed 's/[\\,]/ /g' | sed 's/\"/\\/g' | g\
rep -w $prop`
    r=$(echo ${temp##*|} | tr ']' ' ' | tr ' ' '\n' | cut -d: -f2 \
| sort -n)
    echo $(echo $r | cut -d' ' -f 1)
}

prop='uniqueKey'

node=master
json=`wget --load-cookies cookie.txt --no-check-certificate --se\
rver-response -qO- $URL --post-data='{"service": "cmdevice", "call\
": "getDevice", "arg1": "' $node'}'`
nkey=$(jsonval)
if [ -z $nkey ]; then
    echo $json

```

```

    exit 1
fi
echo "$node.uniqueKey = $nkey"

json=`wget --load-cookies cookie.txt --no-check-certificate --se\
rver-response -qO- $URL --post-data="{\"service\":\"cmmon\",\"call\":\"\
getMetric\",\"arg1\":\"loadOne\"}"` \
mkey=$(jsonval)
echo "loadone.uniqueKey = $mkey"

now=$(date +%s)
day=$((now-86400))

# echo -----
# wget --load-cookies cookie.txt --no-check-certificate --server\
-response -qO- $URL \
#   --post-data="{\"service\":\"cmmon\",\"call\":\"readDataByIntervalNu\
m\",
#               \"readMonDataIdArray\": [{ \"devId\": '$nkey', \"metric\
Id\": '$mkey',
#                                       \"begTime\": '$day', \"endTi\
me\": '$now' }],
#               \"intervalNum\": 0}"
# echo
echo -----
wget --load-cookies cookie.txt --no-check-certificate --server-r\
esponse -qO- $URL \
  --post-data="{\"service\":\"cmmon\",\"call\":\"readDataByIntervalNum\",
                \"args\": [ [ { \"baseType\": \"ReadMonDataId\", \"uniqueKey\" \
: 0, \"modified\": false, \"toBeRemoved\": false, \"childType\": \"\",
                              \"devId\": '$nkey', \"metricId\": '$mkey',
                              \"begTime\": '$day', \"endTime\": '$now' } ], 0 ] }"

# echo
# echo -----
# data="{\"service\":\"cmmon\",\"call\":\"readDataByIntervalNum\",
#       \"args\": [ [ { \"baseType\": \"ReadMonDataId\", \"uniqueKe\
y\": 0, \"modified\": false, \"toBeRemoved\": false, \"childType\": \"\",
#                     \"devId\": '$nkey', \"metricId\": '$mkey',
#                     \"begTime\": '$day', \"endTime\": '$now' } ], \
0 ] }"
# rm loadone.txt.gz
# echo $data > loadone.txt
# gzip -n loadone.txt
# len=$(wc -c loadone.txt.gz | cut -d" " -f1)
# wget --load-cookies cookie.txt --no-check-certificate --header\
  \"Content-Length: $len\" --header 'Content-Encoding: gzip' --serv\
er-response -O- $URL \
#   --post-file=loadone.txt.gz

```

login.sh

```

#!/bin/bash
source url

```

```
user=$USER
pass=$user
wget --keep-session-cookies --save-cookies cookie.txt --no-check\
-certificate --server-response -qO- $URL \
  --post-data='{"service":"login","username":"' $user' ", "passwor\
d":"' $pass'"}'
echo
```

logout.sh

```
#!/bin/bash
source url
wget --load-cookies cookie.txt --no-check-certificate --server-r\
esponse -qO- $URL --post-data='{"service":"logout"}'
rm cookie.txt
echo
```

node001.sh

```
#!/bin/bash
source url

if [ -z "$1" ]; then
  node=node001
else
  node=$1
fi

wget --load-cookies cookie.txt --no-check-certificate --server-r\
esponse -qO- $URL --post-data='{"service":"cmdevice","call":"get\
Device","arg1":"' $node'"}'
```