Bright Cluster Manager 8.0

Developer Manual

Revision: 61f93b0

Date: Fri Jul 5 2024

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 Monitoring Data Producers **15**
2.1 Measurables 15
2.2 Measurables Classes 15
2.3 Metric Monitoring Data Producers 15
2.4 Health Check Monitoring Data Producers 16
2.5 Collection Monitoring Data Producers 16
2.6 Node Execution Filters 17
2.7 Execution Multiplexers 18
2.8 Monitoring Resources 19
2.9 Collection Monitoring Data Producers With Filter And Multiplexer 19
2.10 Collection Monitoring Data Producers For Standalone Entities 21

3 Monitoring Actions **23**
3.1 Actions And Triggers 23
3.2 Time Restrictions 24
3.2.1 Time Restriction Syntax In BNF Notation 24
3.3 CMDaemon Environment Variables 24

3.3.1 Standard Environment Variables Available In Action Scripts . . . 24
3.3.2 Extended Environment Variables Available To Action Scripts . . . 26

4 Bright Cluster Manager JSON API 39
4.1 Services . . . 39
4.1.1 auth . . . 39
4.1.2 ceph . . . 39
4.1.3 cert . . . 39
4.1.4 cloud . . . 39
4.1.5 device . . . 39
4.1.6 etcd . . . 39
4.1.7 gui . . . 39
4.1.8 hadoop . . . 39
4.1.9 job . . . 39
4.1.10 keyvalue . . . 39
4.1.11 kube . . . 39
4.1.12 lustre . . . 39
4.1.13 main . . . 39
4.1.14 mesos . . . 39
4.1.15 mon . . . 39
4.1.16 net . . . 39
4.1.17 openstack . . . 39
4.1.18 part . . . 39
4.1.19 proc . . . 39
4.1.20 prov . . . 39
4.1.21 puppet . . . 39
4.1.22 serv . . . 39
4.1.23 session . . . 39
4.1.24 sync . . . 40
4.1.25 test . . . 40
4.1.26 ticket . . . 40
4.1.27 user . . . 40
4.1.28 zookeeper . . . 40
4.2 Entities . . . 40
4.2.1 AzureDataDisk . . . 40
4.2.2 AzureDisk . . . 40
4.2.3 AzureIntermediateStorage . . . 40
4.2.4 AzureLocation . . . 40
4.2.5 AzureManagedDiskParameters . . . 40
4.2.6 AzureOSDisk . . . 40
4.2.7 AzureProvider . . . 40
4.2.8 AzurePublicIP . . . 40
4.2.9 AzureSettings . . . 40
4.2.10 AzureVMSize . . . 40
4.2.11 BadEntityManagers . . . 40
4.2.12 BasicResource . . . 40
4.2.13 BeeGFSAdmonRole . . . 40

4.2.14 BeeGFSClientRole . . . 40
4.2.15 BeeGFSManagementRole . . . 40
4.2.16 BeeGFSMetadataRole . . . 40
4.2.17 BeeGFSStorageRole . . . 40
4.2.18 BigDataAdditionalTool . . . 40
4.2.19 BigDataAdvancedSettings . . . 40
4.2.20 BigDataCassandra . . . 40
4.2.21 BigDataFileSystemSettings . . . 40
4.2.22 BigDataJobManagementSettings . . . 40
4.2.23 BigDataLoggingSettings . . . 40
4.2.24 BigDataSecurity . . . 40
4.2.25 BigDataSpark . . . 40
4.2.26 BillingHistory . . . 40
4.2.27 BMCSettings . . . 40
4.2.28 BootRole . . . 40
4.2.29 BurnConfig . . . 40
4.2.30 BurnStatus . . . 40
4.2.31 BurnTestStatus . . . 40
4.2.32 Category . . . 40
4.2.33 Ceph . . . 40
4.2.34 CephMDSRole . . . 40
4.2.35 CephMGRRole . . . 40
4.2.36 CephMonitorRole . . . 40
4.2.37 CephOSDBlueStoreConfig . . . 40
4.2.38 CephOSDConfig . . . 40
4.2.39 CephOSDFileStoreConfig . . . 40
4.2.40 CephOSDLegacyConfig . . . 40
4.2.41 CephOSDPool . . . 41
4.2.42 CephOSDRole . . . 41
4.2.43 CephState . . . 41
4.2.44 Certificate . . . 41
4.2.45 CertificateRequest . . . 41
4.2.46 CertificateSubjectName . . . 41
4.2.47 Cgroup . . . 41
4.2.48 CgroupController . . . 41
4.2.49 CgroupControllerBlkio . . . 41
4.2.50 CgroupControllerCpu . . . 41
4.2.51 CgroupControllerCpuacct . . . 41
4.2.52 CgroupControllerCpuset . . . 41
4.2.53 CgroupControllerDevices . . . 41
4.2.54 CgroupControllerFreezer . . . 41
4.2.55 CgroupControllerHugetlb . . . 41
4.2.56 CgroupControllerMemory . . . 41
4.2.57 CgroupControllerNetcls . . . 41
4.2.58 CgroupControllerNetprio . . . 41
4.2.59 CgroupControllerNs . . . 41

4.2.60 CgroupControllerPerf . . . 41
4.2.61 CgroupRule . . . 41
4.2.62 CgroupSupervisorRole . . . 41
4.2.63 Chassis . . . 41
4.2.64 ChronosRole . . . 41
4.2.65 ClientUserData . . . 41
4.2.66 CloudDirectorRole . . . 41
4.2.67 CloudGatewayRole . . . 41
4.2.68 CloudImage . . . 41
4.2.69 CloudJobDescription . . . 41
4.2.70 CloudJobSubmissionStatus . . . 41
4.2.71 CloudNode . . . 41
4.2.72 CloudPrivateCloud . . . 41
4.2.73 CloudProvider . . . 41
4.2.74 CloudRegion . . . 41
4.2.75 CloudSettings . . . 41
4.2.76 CloudStaticIP . . . 41
4.2.77 CloudStorageActionData . . . 41
4.2.78 CloudStorageNodeState . . . 41
4.2.79 CloudType . . . 41
4.2.80 CloudVirtualNetworkInterface . . . 41
4.2.81 ClusterSetup . . . 41
4.2.82 CMDaemonBackgroundTask . . . 41
4.2.83 CMDaemonFailover . . . 41
4.2.84 CMDaemonFailoverGroup . . . 41
4.2.85 CMDaemonFailoverGroupStatus . . . 41
4.2.86 CMDaemonFailoverPeer . . . 41
4.2.87 CMDaemonFailoverStatus . . . 41
4.2.88 CMDaemonStatus . . . 42
4.2.89 CMService . . . 42
4.2.90 CMSubConfig . . . 42
4.2.91 CMSubIntermediateStorage . . . 42
4.2.92 ConfigFileVersion . . . 42
4.2.93 ConfigSum . . . 42
4.2.94 ConfigurationOverlay . . . 42
4.2.95 Consolidator . . . 42
4.2.96 ContainerdHostRole . . . 42
4.2.97 ContainerInfo . . . 42
4.2.98 CustomizationEntry . . . 42
4.2.99 CustomizationFile . . . 42
4.2.100 DellClustat . . . 42
4.2.101 DellClustatGroup . . . 42
4.2.102 DellClustatNode . . . 42
4.2.103 DellDiskGroupInfo . . . 42
4.2.104 DellPhysicalDiskDriveInfo . . . 42
4.2.105 DellRAIDControllerInfo . . . 42

4.2.106 DellSettings . . . 42
4.2.107 DellSettingsFirmware . . . 42
4.2.108 DellSettingsNicDevice . . . 42
4.2.109 DellStorageInfo . . . 42
4.2.110 DellVirtualDiskInfo . . . 42
4.2.111 Device . . . 42
4.2.112 DevStatus . . . 42
4.2.113 DiskAssertion . . . 42
4.2.114 DiskDevice . . . 42
4.2.115 DiskInfo . . . 42
4.2.116 DiskPartition . . . 42
4.2.117 DiskRaid . . . 42
4.2.118 DiskSetup . . . 42
4.2.119 DiskVolume . . . 42
4.2.120 DiskVolumeGroup . . . 42
4.2.121 DockerHostRole . . . 42
4.2.122 DockerRegistryFilesystemStorageDriver . . . 42
4.2.123 DockerRegistryInmemoryStorageDriver . . . 42
4.2.124 DockerRegistryRole . . . 42
4.2.125 DockerRegistryStorageDriver . . . 42
4.2.126 DockerStorageBackend . . . 42
4.2.127 DockerStorageDeviceMapperBackend . . . 42
4.2.128 DrainAction . . . 42
4.2.129 DrainResult . . . 42
4.2.130 EC2AMI . . . 42
4.2.131 EC2AvailabilityZone . . . 42
4.2.132 EC2EBSStorage . . . 42
4.2.133 EC2EphemeralStorage . . . 42
4.2.134 EC2PrivateCloud . . . 42
4.2.135 EC2Provider . . . 43
4.2.136 EC2Region . . . 43
4.2.137 EC2RegionAMI . . . 43
4.2.138 EC2Settings . . . 43
4.2.139 EC2StaticIP . . . 43
4.2.140 EC2Storage . . . 43
4.2.141 EC2Type . . . 43
4.2.142 EC2VirtualNetworkInterface . . . 43
4.2.143 ElasticSearchRole . . . 43
4.2.144 EntityManagersMD5 . . . 43
4.2.145 EtcdCluster . . . 43
4.2.146 EtcdHostRole . . . 43
4.2.147 EthernetSwitch . . . 43
4.2.148 FailoverRole . . . 43
4.2.149 FakeJob . . . 43
4.2.150 FakeJobQueue . . . 43
4.2.151 FakeJobQueueStat . . . 43

4.2.152 FakeWlmClientRole . . . 43
4.2.153 FakeWlmServerRole . . . 43
4.2.154 FileInfo . . . 43
4.2.155 FileSyncConfig . . . 43
4.2.156 FileSyncStatus . . . 43
4.2.157 FlannelConfigurationRole . . . 43
4.2.158 FlannelHostRole . . . 43
4.2.159 FlannelNetworkingBackend . . . 43
4.2.160 FlannelNetworkingUdpBackend . . . 43
4.2.161 FlannelNetworkingVxLanBackend . . . 43
4.2.162 FSExport . . . 43
4.2.163 FSMount . . . 43
4.2.164 FSPart . . . 43
4.2.165 FSPartAssociation . . . 43
4.2.166 FSPartBasicAssociation . . . 43
4.2.167 FSPartProviderAssociation . . . 43
4.2.168 GaleraRole . . . 43
4.2.169 GenericDevice . . . 43
4.2.170 GenericResource . . . 43
4.2.171 GPUInfo . . . 43
4.2.172 GPUSettings . . . 43
4.2.173 GpuUnit . . . 43
4.2.174 GPUUnitInfo . . . 43
4.2.175 GridEngineJob . . . 43
4.2.176 GridEngineJobQueue . . . 43
4.2.177 GridEngineJobQueueStat . . . 43
4.2.178 GridEngineParallelEnvironment . . . 43
4.2.179 Group . . . 43
4.2.180 GuiCephOsdPoolInfo . . . 43
4.2.181 GuiCephOverview . . . 43
4.2.182 GuiCephPgsInfo . . . 44
4.2.183 GuiClusterOverview . . . 44
4.2.184 GuiCompleteOpenStackOverview . . . 44
4.2.185 GuiDiskUsage . . . 44
4.2.186 GuiGpuUnitOverview . . . 44
4.2.187 GuiHadoopHDFSDetailHBase . . . 44
4.2.188 GuiHadoopHDFSDetailHDFS . . . 44
4.2.189 GuiHadoopHDFSDetailMapreduce . . . 44
4.2.190 GuiHadoopHDFSDetailSpark . . . 44
4.2.191 GuiHadoopHDFSDetailYarn . . . 44
4.2.192 GuiHadoopHDFSDetailZooKeeper . . . 44
4.2.193 GuiHadoopHDFSOverview . . . 44
4.2.194 GuiJob . . . 44
4.2.195 GuiKubeClusterOverview . . . 44
4.2.196 GuiNetworkInterface . . . 44
4.2.197 GuiNodeOverview . . . 44

4.2.198 GuiNodeStatus . . . 44
4.2.199 GuiOpenStackOverview . . . 44
4.2.200 GuiOpenStackProjectOverview . . . 44
4.2.201 GuiOpenStackTenantOverview . . . 44
4.2.202 GuiPDUBank . . . 44
4.2.203 GuiPDUOutlet . . . 44
4.2.204 GuiPDUOverview . . . 44
4.2.205 GuiSwitchOverview . . . 44
4.2.206 GuiSwitchPort . . . 44
4.2.207 GuiWorkload . . . 44
4.2.208 HadoopAccumuloMasterHDFSConfiguration . . . 44
4.2.209 HadoopAccumuloMasterRole . . . 44
4.2.210 HadoopAccumuloTabletHDFSConfiguration . . . 44
4.2.211 HadoopAccumuloTabletRole . . . 44
4.2.212 HadoopAlluxioMasterHDFSConfiguration . . . 44
4.2.213 HadoopAlluxioMasterRole . . . 44
4.2.214 HadoopAlluxioWorkerHDFSConfiguration . . . 44
4.2.215 HadoopAlluxioWorkerRole . . . 44
4.2.216 HadoopBaseConfiguration . . . 44
4.2.217 HadoopCassandraHDFSConfiguration . . . 44
4.2.218 HadoopCassandraRole . . . 44
4.2.219 HadoopDataNodeHDFSConfiguration . . . 44
4.2.220 HadoopDataNodeRole . . . 44
4.2.221 HadoopDrillHDFSConfiguration . . . 44
4.2.222 HadoopDrillRole . . . 44
4.2.223 HadoopFlinkJobManagerHDFSConfiguration . . . 44
4.2.224 HadoopFlinkJobManagerRole . . . 44
4.2.225 HadoopFlinkTaskManagerHDFSConfiguration . . . 44
4.2.226 HadoopFlinkTaskManagerRole . . . 44
4.2.227 HadoopHBaseClientHDFSConfiguration . . . 44
4.2.228 HadoopHBaseClientRole . . . 44
4.2.229 HadoopHBaseServerHDFSConfiguration . . . 45
4.2.230 HadoopHBaseServerRole . . . 45
4.2.231 HadoopHDFS . . . 45
4.2.232 HadoopHiveHDFSConfiguration . . . 45
4.2.233 HadoopHiveRole . . . 45
4.2.234 HadoopJob . . . 45
4.2.235 HadoopJobQueue . . . 45
4.2.236 HadoopJobQueueStat . . . 45
4.2.237 HadoopJobTrackerHDFSConfiguration . . . 45
4.2.238 HadoopJobTrackerRole . . . 45
4.2.239 HadoopJournalHDFSConfiguration . . . 45
4.2.240 HadoopJournalRole . . . 45
4.2.241 HadoopKafkaServerHDFSConfiguration . . . 45
4.2.242 HadoopKafkaServerRole . . . 45
4.2.243 HadoopKMServerHDFSConfiguration . . . 45

4.2.244 HadoopKMServerRole . . . 45
4.2.245 HadoopNameNodeHDFSConfiguration . . . 45
4.2.246 HadoopNameNodeRole . . . 45
4.2.247 HadoopNFSGatewayHDFSConfiguration . . . 45
4.2.248 HadoopNFSGatewayRole . . . 45
4.2.249 HadoopPigHDFSConfiguration . . . 45
4.2.250 HadoopPigRole . . . 45
4.2.251 HadoopSecondaryNameNodeHDFSConfiguration . . . 45
4.2.252 HadoopSecondaryNameNodeRole . . . 45
4.2.253 HadoopSparkMasterHDFSConfiguration . . . 45
4.2.254 HadoopSparkMasterRole . . . 45
4.2.255 HadoopSparkWorkerHDFSConfiguration . . . 45
4.2.256 HadoopSparkWorkerRole . . . 45
4.2.257 HadoopSparkYARNHDFSConfiguration . . . 45
4.2.258 HadoopSparkYARNRole . . . 45
4.2.259 HadoopSqoopHDFSConfiguration . . . 45
4.2.260 HadoopSqoopRole . . . 45
4.2.261 HadoopStormNimbusHDFSConfiguration . . . 45
4.2.262 HadoopStormNimbusRole . . . 45
4.2.263 HadoopStormSupervisorHDFSConfiguration . . . 45
4.2.264 HadoopStormSupervisorRole . . . 45
4.2.265 HadoopTaskTrackerHDFSConfiguration . . . 45
4.2.266 HadoopTaskTrackerRole . . . 45
4.2.267 HadoopYARNClientHDFSConfiguration . . . 45
4.2.268 HadoopYARNClientRole . . . 45
4.2.269 HadoopYARNServerHDFSConfiguration . . . 45
4.2.270 HadoopYARNServerRole . . . 45
4.2.271 HadoopZeppelinHDFSConfiguration . . . 45
4.2.272 HadoopZeppelinRole . . . 45
4.2.273 HadoopZooKeeperHDFSConfiguration . . . 45
4.2.274 HadoopZooKeeperRole . . . 45
4.2.275 HAProxyEntry . . . 45
4.2.276 HAProxyEntryBind . . . 46
4.2.277 HAProxyRole . . . 46
4.2.278 HAProxyServer . . . 46
4.2.279 HAProxySharedSettings . . . 46
4.2.280 IBSwitch . . . 46
4.2.281 IPCPerm . . . 46
4.2.282 IPResource . . . 46
4.2.283 Job . . . 46
4.2.284 JobInfo . . . 46
4.2.285 JobInfoStatistics . . . 46
4.2.286 JobQueue . . . 46
4.2.287 JobQueuePlaceholder . . . 46
4.2.288 JobQueueStat . . . 46
4.2.289 KeepalivedEntry . . . 46

4.2.290 KeepalivedRole . . . 46
4.2.291 KernelModule . . . 46
4.2.292 KeyValuePair . . . 46
4.2.293 KibanaRole . . . 46
4.2.294 KubeCluster . . . 46
4.2.295 KubePodInfo . . . 46
4.2.296 KubernetesApiServerRole . . . 46
4.2.297 KubernetesControllerRole . . . 46
4.2.298 KubernetesDashboardRole . . . 46
4.2.299 KubernetesDNSRole . . . 46
4.2.300 KubernetesHeapsterRole . . . 46
4.2.301 KubernetesNodeRole . . . 46
4.2.302 KubernetesProxyRole . . . 46
4.2.303 KubernetesSchedulerRole . . . 46
4.2.304 KubeUserPolicy . . . 46
4.2.305 LicenseInfo . . . 46
4.2.306 LoginRole . . . 46
4.2.307 LogstashForwarderRole . . . 46
4.2.308 LogstashServerCustomFilter . . . 46
4.2.309 LogstashServerCustomListener . . . 46
4.2.310 LogstashServerCustomOutput . . . 46
4.2.311 LogstashServerElasticOutput . . . 46
4.2.312 LogstashServerFilter . . . 46
4.2.313 LogstashServerListener . . . 46
4.2.314 LogstashServerLocalFileListener . . . 46
4.2.315 LogstashServerLumberjackListener . . . 46
4.2.316 LogstashServerOutput . . . 46
4.2.317 LogstashServerRole . . . 46
4.2.318 LogstashServerRSyslogFilter . . . 46
4.2.319 LogstashServerRSyslogListener . . . 46
4.2.320 LogstashServerStdOutput . . . 46
4.2.321 LSFBaseJob . . . 46
4.2.322 LSFBaseJobQueue . . . 46
4.2.323 LSFBaseJobQueueStat . . . 47
4.2.324 LSFCgroupsSettings . . . 47
4.2.325 LSFClientRole . . . 47
4.2.326 LSFJob . . . 47
4.2.327 LSFJobQueue . . . 47
4.2.328 LSFJobQueueStat . . . 47
4.2.329 LSFServerRole . . . 47
4.2.330 LustreAlert . . . 47
4.2.331 LustreClientMount . . . 47
4.2.332 LustreFileSystem . . . 47
4.2.333 LustreFileSystemTarget . . . 47
4.2.334 LustreLog . . . 47
4.2.335 LustreOverview . . . 47

4.2.336 LustreServer . . . 47
4.2.337 LustreServerProfile . . . 47
4.2.338 LustreSettings . . . 47
4.2.339 LustreTargetMap . . . 47
4.2.340 LustreUser . . . 47
4.2.341 LustreVolume . . . 47
4.2.342 LustreVolumeNode . . . 47
4.2.343 MarathonRole . . . 47
4.2.344 MasterNode . . . 47
4.2.345 MasterRole . . . 47
4.2.346 MemcachedRole . . . 47
4.2.347 MemoryInfo . . . 47
4.2.348 MesosCluster . . . 47
4.2.349 MesosDNSRole . . . 47
4.2.350 MesosMasterRole . . . 47
4.2.351 MesosProxyRole . . . 47
4.2.352 MesosResourceUsage . . . 47
4.2.353 MesosSlaveRole . . . 47
4.2.354 MICHostRole . . . 47
4.2.355 MICInfo . . . 47
4.2.356 MICNode . . . 47
4.2.357 MICNodeCategory . . . 47
4.2.358 MICOverlay . . . 47
4.2.359 MICSettings . . . 47
4.2.360 MonitoringAction . . . 47
4.2.361 MonitoringActionRunData . . . 47
4.2.362 MonitoringCacheSubSystemInfo . . . 47
4.2.363 MonitoringCategoryListExecutionFilter . . . 47
4.2.364 MonitoringCompareExpression . . . 47
4.2.365 MonitoringConsolidator . . . 47
4.2.366 MonitoringDataCacheSubSystemInfo . . . 47
4.2.367 MonitoringDataProducer . . . 47
4.2.368 MonitoringDataProducerAggregateNode . . . 47
4.2.369 MonitoringDataProducerAlertLevel . . . 47
4.2.370 MonitoringDataProducerCGroup . . . 48
4.2.371 MonitoringDataProducerClusterTotal . . . 48
4.2.372 MonitoringDataProducerCMDaemonState . . . 48
4.2.373 MonitoringDataProducerDeviceState . . . 48
4.2.374 MonitoringDataProducerEC2SpotPrices . . . 48
4.2.375 MonitoringDataProducerEthernetSwitch . . . 48
4.2.376 MonitoringDataProducerFuture . . . 48
4.2.377 MonitoringDataProducerGalera . . . 48
4.2.378 MonitoringDataProducerGenerator . . . 48
4.2.379 MonitoringDataProducerGPU . . . 48
4.2.380 MonitoringDataProducerInternal . . . 48
4.2.381 MonitoringDataProducerJob . . . 48

4.2.382 MonitoringDataProducerJobQueue 48
4.2.383 MonitoringDataProducerLua 48
4.2.384 MonitoringDataProducerMonitoringSystem 48
4.2.385 MonitoringDataProducerOpenStack 48
4.2.386 MonitoringDataProducerOpenStackHealth 48
4.2.387 MonitoringDataProducerPerpetual 48
4.2.388 MonitoringDataProducerPowerDistributionUnit 48
4.2.389 MonitoringDataProducerProcMemInfo 48
4.2.390 MonitoringDataProducerProcMount 48
4.2.391 MonitoringDataProducerProcNetDev 48
4.2.392 MonitoringDataProducerProcNetSnmp 48
4.2.393 MonitoringDataProducerProcPidStat 48
4.2.394 MonitoringDataProducerProcStat 48
4.2.395 MonitoringDataProducerProcVMStat 48
4.2.396 MonitoringDataProducerRackSensor 48
4.2.397 MonitoringDataProducerScript 48
4.2.398 MonitoringDataProducerSingleLineHealthCheckScript 48
4.2.399 MonitoringDataProducerSingleLineMetricScript 48
4.2.400 MonitoringDataProducerSingleLineScript 48
4.2.401 MonitoringDataProducerSmart 48
4.2.402 MonitoringDataProducerSysBlockStat 48
4.2.403 MonitoringDataProducerSysInfo 48
4.2.404 MonitoringDataProducerTest 48
4.2.405 MonitoringDataProducerTrustedTool 48
4.2.406 MonitoringDataProducerUserCount 48
4.2.407 MonitoringDeviceStateSubSystemInfo 48
4.2.408 MonitoringDrainAction 48
4.2.409 MonitoringEmailAction 48
4.2.410 MonitoringEventAction 48
4.2.411 MonitoringExecutionFilter 48
4.2.412 MonitoringExecutionMultiplexer 48
4.2.413 MonitoringExpression 48
4.2.414 MonitoringGroupedExpression 48
4.2.415 MonitoringHealthOverview 48
4.2.416 MonitoringImageUpdateAction 48
4.2.417 MonitoringJobMetricSettings 49
4.2.418 MonitoringLuaExecutionFilter 49
4.2.419 MonitoringLuaExecutionMultiplexer 49
4.2.420 MonitoringMeasurable 49
4.2.421 MonitoringMeasurableEnum 49
4.2.422 MonitoringMeasurableHealthCheck 49
4.2.423 MonitoringMeasurableMetric 49
4.2.424 MonitoringNodeListExecutionFilter 49
4.2.425 MonitoringOverlayListExecutionFilter 49
4.2.426 MonitoringPlotterSubSystemInfo 49
4.2.427 MonitoringPowerAction 49

4.2.428 MonitoringPowerOffAction . . . 49
4.2.429 MonitoringPowerOnAction . . . 49
4.2.430 MonitoringPowerResetAction . . . 49
4.2.431 MonitoringRebootAction . . . 49
4.2.432 MonitoringReplicateConfiguration . . . 49
4.2.433 MonitoringReplicateSource . . . 49
4.2.434 MonitoringReplicateSubSystemInfo . . . 49
4.2.435 MonitoringResourceExecutionFilter . . . 49
4.2.436 MonitoringResourceExecutionMultiplexer . . . 49
4.2.437 MonitoringRole . . . 49
4.2.438 MonitoringScriptAction . . . 49
4.2.439 MonitoringServiceAction . . . 49
4.2.440 MonitoringServiceRestartAction . . . 49
4.2.441 MonitoringServiceStartAction . . . 49
4.2.442 MonitoringServiceStopAction . . . 49
4.2.443 MonitoringServiceSubSystemInfo . . . 49
4.2.444 MonitoringShutdownAction . . . 49
4.2.445 MonitoringStorageSubSystemInfo . . . 49
4.2.446 MonitoringSubSystemInfo . . . 49
4.2.447 MonitoringTrigger . . . 49
4.2.448 MonitoringTypeExecutionFilter . . . 49
4.2.449 MonitoringTypeExecutionMultiplexer . . . 49
4.2.450 MonitoringUndrainAction . . . 49
4.2.451 MsgQueue . . . 49
4.2.452 MyrinetSwitch . . . 49
4.2.453 Network . . . 49
4.2.454 NetworkAliasInterface . . . 49
4.2.455 NetworkBmcInterface . . . 49
4.2.456 NetworkBondInterface . . . 49
4.2.457 NetworkBridgeInterface . . . 49
4.2.458 NetworkInterface . . . 49
4.2.459 NetworkNetMapInterface . . . 49
4.2.460 NetworkPhysicalInterface . . . 49
4.2.461 NetworkTunnelInterface . . . 49
4.2.462 NetworkVLANInterface . . . 49
4.2.463 NewNode . . . 49
4.2.464 NginxRole . . . 50
4.2.465 Node . . . 50
4.2.466 NodeCategory . . . 50
4.2.467 NodeGroup . . . 50
4.2.468 OpenLavaCgroupsSettings . . . 50
4.2.469 OpenLavaClientRole . . . 50
4.2.470 OpenLavaJob . . . 50
4.2.471 OpenLavaJobQueue . . . 50
4.2.472 OpenLavaJobQueueStat . . . 50
4.2.473 OpenLavaServerRole . . . 50

4.2.474 OpenStack . . . 50
4.2.475 OpenStackApiAgent . . . 50
4.2.476 OpenStackApiDomain . . . 50
4.2.477 OpenStackApiEndpoint . . . 50
4.2.478 OpenStackApiEntity . . . 50
4.2.479 OpenStackApiFlavor . . . 50
4.2.480 OpenStackApiFloatingIP . . . 50
4.2.481 OpenStackApiGroup . . . 50
4.2.482 OpenStackApiHostAggregate . . . 50
4.2.483 OpenStackApiHypervisor . . . 50
4.2.484 OpenStackApiImage . . . 50
4.2.485 OpenStackApiNetwork . . . 50
4.2.486 OpenStackApiPort . . . 50
4.2.487 OpenStackApiProject . . . 50
4.2.488 OpenStackApiRole . . . 50
4.2.489 OpenStackApiRoleAssignment . . . 50
4.2.490 OpenStackApiRouter . . . 50
4.2.491 OpenStackApiSecurityGroup . . . 50
4.2.492 OpenStackApiServer . . . 50
4.2.493 OpenStackApiService . . . 50
4.2.494 OpenStackApiStack . . . 50
4.2.495 OpenStackApiSubnet . . . 50
4.2.496 OpenStackApiUser . . . 50
4.2.497 OpenStackApiVolume . . . 50
4.2.498 OpenStackApiVolumeSnapshot . . . 50
4.2.499 OpenStackApiVolumeType . . . 50
4.2.500 OpenStackAuthBackend . . . 50
4.2.501 OpenStackAuthBackendHybrid . . . 50
4.2.502 OpenStackAuthBackendLDAP . . . 50
4.2.503 OpenStackAuthBackendLDAPGroupSettings . . . 50
4.2.504 OpenStackAuthBackendLDAPProjectSettings . . . 50
4.2.505 OpenStackAuthBackendLDAPRoleSettings . . . 50
4.2.506 OpenStackAuthBackendLDAPUserSettings . . . 50
4.2.507 OpenStackAuthBackendSQL . . . 50
4.2.508 OpenStackBareMetalApiRole . . . 50
4.2.509 OpenStackBareMetalConductorRole . . . 50
4.2.510 OpenStackBareMetalDiscoverdDNSMasqRole . . . 50
4.2.511 OpenStackBareMetalDiscoverdRole . . . 51
4.2.512 OpenStackBlockStorage . . . 51
4.2.513 OpenStackComputeApiEC2Role . . . 51
4.2.514 OpenStackComputeApiMetadataRole . . . 51
4.2.515 OpenStackComputeApiRole . . . 51
4.2.516 OpenStackComputeConductorRole . . . 51
4.2.517 OpenStackComputeRole . . . 51
4.2.518 OpenStackComputeSchedulerRole . . . 51
4.2.519 OpenStackComputeVNCProxyRole . . . 51

4.2.520 OpenStackDashboardRole . . . 51
4.2.521 OpenStackDataProcessingApiRole . . . 51
4.2.522 OpenStackDBaaSRole . . . 51
4.2.523 OpenStackDefaultUserRole . . . 51
4.2.524 OpenStackIdentityApiRole . . . 51
4.2.525 OpenStackImageApiRole . . . 51
4.2.526 OpenStackImageBackend . . . 51
4.2.527 OpenStackImageBackendCeph . . . 51
4.2.528 OpenStackImageBackendFS . . . 51
4.2.529 OpenStackImageRegistryRole . . . 51
4.2.530 OpenStackMessageQueueServerRole . . . 51
4.2.531 OpenStackNetworkApiRole . . . 51
4.2.532 OpenStackNetworkDHCPAgentRole . . . 51
4.2.533 OpenStackNetworkL3AgentRole . . . 51
4.2.534 OpenStackNetworkMetadataAgentRole . . . 51
4.2.535 OpenStackNetworkOVSAgentRole . . . 51
4.2.536 OpenStackNovaImageBackend . . . 51
4.2.537 OpenStackNovaImageBackendCeph . . . 51
4.2.538 OpenStackNovaImageBackendCow . . . 51
4.2.539 OpenStackObjectAccountRole . . . 51
4.2.540 OpenStackObjectApiRole . . . 51
4.2.541 OpenStackObjectContainerRole . . . 51
4.2.542 OpenStackObjectStoreRole . . . 51
4.2.543 OpenStackOrchestrationApiRole . . . 51
4.2.544 OpenStackOrchestrationRole . . . 51
4.2.545 OpenStackSettings . . . 51
4.2.546 OpenStackSettingsAdvanced . . . 51
4.2.547 OpenStackSettingsAuthentication . . . 51
4.2.548 OpenStackSettingsCMDaemonInteractions . . . 51
4.2.549 OpenStackSettingsCollection . . . 51
4.2.550 OpenStackSettingsCompute . . . 51
4.2.551 OpenStackSettingsCredentials . . . 51
4.2.552 OpenStackSettingsDatabase . . . 51
4.2.553 OpenStackSettingsLogging . . . 51
4.2.554 OpenStackSettingsNetworking . . . 51
4.2.555 OpenStackSettingsPorts . . . 51
4.2.556 OpenStackSettingsQuota . . . 51
4.2.557 OpenStackSettingsUserPortal . . . 51
4.2.558 OpenStackSettingsUsers . . . 52
4.2.559 OpenStackStorage . . . 52
4.2.560 OpenStackTelemetryAgentCentralRole . . . 52
4.2.561 OpenStackTelemetryAgentComputeRole . . . 52
4.2.562 OpenStackTelemetryAgentIpmiRole . . . 52
4.2.563 OpenStackTelemetryAgentNotificationRole . . . 52
4.2.564 OpenStackTelemetryAlarmEvaluatorRole . . . 52
4.2.565 OpenStackTelemetryAlarmNotifierRole . . . 52

4.2.566 OpenStackTelemetryApiRole . . . 52
4.2.567 OpenStackTelemetryCollectorRole . . . 52
4.2.568 OpenStackUserRole . . . 52
4.2.569 OpenStackVolumeApiRole . . . 52
4.2.570 OpenStackVolumeBackend . . . 52
4.2.571 OpenStackVolumeBackend3PAR . . . 52
4.2.572 OpenStackVolumeBackendCeph . . . 52
4.2.573 OpenStackVolumeBackendDellStorageCenter . . . 52
4.2.574 OpenStackVolumeBackendGPFS . . . 52
4.2.575 OpenStackVolumeBackendNetApp . . . 52
4.2.576 OpenStackVolumeBackendNFS . . . 52
4.2.577 OpenStackVolumeBackendSolidFire . . . 52
4.2.578 OpenStackVolumeBackupBackend . . . 52
4.2.579 OpenStackVolumeBackupBackendCeph . . . 52
4.2.580 OpenStackVolumeBackupRole . . . 52
4.2.581 OpenStackVolumeRole . . . 52
4.2.582 OpenStackVolumeSchedulerRole . . . 52
4.2.583 OpenvSwitchRole . . . 52
4.2.584 OsapiPortIP . . . 52
4.2.585 OsapiSecurityGroupRule . . . 52
4.2.586 OsapiStackResource . . . 52
4.2.587 OsapiSubnetAllocationPool . . . 52
4.2.588 OSService . . . 52
4.2.589 OSServiceArray . . . 52
4.2.590 OSServiceConfig . . . 52
4.2.591 ParentJob . . . 52
4.2.592 Partition . . . 52
4.2.593 PBSJob . . . 52
4.2.594 PBSJobQueue . . . 52
4.2.595 PBSJobQueueStat . . . 52
4.2.596 PbsProCgroupsSettings . . . 52
4.2.597 PbsProClientRole . . . 52
4.2.598 PbsProJob . . . 52
4.2.599 PbsProJobQueue . . . 52
4.2.600 PbsProJobQueueStat . . . 52
4.2.601 PbsProServerRole . . . 52
4.2.602 PDUPort . . . 52
4.2.603 PhysicalNode . . . 52
4.2.604 PowerDistributionUnit . . . 52
4.2.605 PowerOperation . . . 53
4.2.606 PowerStatus . . . 53
4.2.607 Process . . . 53
4.2.608 Processor . . . 53
4.2.609 Profile . . . 53
4.2.610 ProgramRunnerInput . . . 53
4.2.611 ProgramRunnerKill . . . 53

4.2.612 ProgramRunnerOutput . . . 53
4.2.613 ProgramRunnerStatus . . . 53
4.2.614 ProvisioningNodeStatus . . . 53
4.2.615 ProvisioningProcessorJob . . . 53
4.2.616 ProvisioningRequestStatus . . . 53
4.2.617 ProvisioningRole . . . 53
4.2.618 ProvisioningStatus . . . 53
4.2.619 Puppet . . . 53
4.2.620 PuppetApplyOnNodeRequest . . . 53
4.2.621 PuppetApplyResult . . . 53
4.2.622 PuppetApplySession . . . 53
4.2.623 PuppetClass . . . 53
4.2.624 PuppetClassDeclaration . . . 53
4.2.625 PuppetClassFactory . . . 53
4.2.626 PuppetConfigurationEntry . . . 53
4.2.627 PuppetForgeInstallation . . . 53
4.2.628 PuppetForgePagination . . . 53
4.2.629 PuppetForgeSearchRequest . . . 53
4.2.630 PuppetForgeSearchResult . . . 53
4.2.631 PuppetKeyValuePair . . . 53
4.2.632 PuppetModule . . . 53
4.2.633 PuppetModuleDependency . . . 53
4.2.634 PuppetModuleRelease . . . 53
4.2.635 PuppetOperatingSystemSupport . . . 53
4.2.636 PuppetParameterFactory . . . 53
4.2.637 PuppetRescanResult . . . 53
4.2.638 PuppetResourceDeclaration . . . 53
4.2.639 PuppetRole . . . 53
4.2.640 PuppetRoleChange . . . 53
4.2.641 Rack . . . 53
4.2.642 RackPosition . . . 53
4.2.643 RackSensor . . . 53
4.2.644 RadosGatewayRole . . . 53
4.2.645 RemoteNodeInstallerInteraction . . . 53
4.2.646 RemoteSetupExecution . . . 53
4.2.647 ResourcePool . . . 53
4.2.648 ResourcePoolStatus . . . 53
4.2.649 Role . . . 53
4.2.650 S3BucketIntermediateStorage . . . 53
4.2.651 ScaleDynamicNodesProvider . . . 53
4.2.652 ScaleEngine . . . 54
4.2.653 ScaleHpcEngine . . . 54
4.2.654 ScaleHpcQueueTracker . . . 54
4.2.655 ScaleMesosEngine . . . 54
4.2.656 ScaleMesosLoadTracker . . . 54
4.2.657 ScalePendingWorkload . . . 54

4.2.658 ScaleResourceProvider . . . 54
4.2.659 ScaleServerRole . . . 54
4.2.660 ScaleStaticNodesProvider . . . 54
4.2.661 ScaleTracker . . . 54
4.2.662 Semaphore . . . 54
4.2.663 Sensor . . . 54
4.2.664 Session . . . 54
4.2.665 SGEClientRole . . . 54
4.2.666 SGEJob . . . 54
4.2.667 SGEJobQueue . . . 54
4.2.668 SGEJobQueueStat . . . 54
4.2.669 SGEParallelEnvironment . . . 54
4.2.670 SGEServerRole . . . 54
4.2.671 SharedMemory . . . 54
4.2.672 SlaveNode . . . 54
4.2.673 SlurmCgroupsSettings . . . 54
4.2.674 SlurmClientRole . . . 54
4.2.675 SlurmJob . . . 54
4.2.676 SlurmJobQueue . . . 54
4.2.677 SlurmJobQueueStat . . . 54
4.2.678 SlurmServerRole . . . 54
4.2.679 SoftwareImage . . . 54
4.2.680 SoftwareImageProxy . . . 54
4.2.681 SoftwareImageRevisionInfo . . . 54
4.2.682 StandaloneMonitoredEntity . . . 54
4.2.683 StaticRoute . . . 54
4.2.684 StorageNodePolicy . . . 54
4.2.685 StorageRole . . . 54
4.2.686 StringListObject . . . 54
4.2.687 SubnetManagerRole . . . 54
4.2.688 SubSystemInfo . . . 54
4.2.689 Switch . . . 54
4.2.690 SwitchPort . . . 54
4.2.691 SysInfoCollector . . . 54
4.2.692 Ticket . . . 54
4.2.693 TorqueCgroupsSettings . . . 54
4.2.694 TorqueClientRole . . . 54
4.2.695 TorqueJob . . . 54
4.2.696 TorqueJobQueue . . . 54
4.2.697 TorqueJobQueueStat . . . 54
4.2.698 TorqueServerRole . . . 54
4.2.699 UCSAdaptorEthCompQueueProfile . . . 55
4.2.700 UCSAdaptorEthGenProfile . . . 55
4.2.701 UCSAdaptorEthInterruptProfile . . . 55
4.2.702 UCSAdaptorEthOffloadProfile . . . 55
4.2.703 UCSAdaptorEthRecvQueueProfile . . . 55

4.2.704 UCSAdaptorEthUSNICProfile . . . 55
4.2.705 UCSAdaptorEthWorkQueueProfile . . . 55
4.2.706 UCSAdaptorExtEthIf . . . 55
4.2.707 UCSAdaptorExtIpV6RssHashProfile . . . 55
4.2.708 UCSAdaptorFcCdbWorkQueueProfile . . . 55
4.2.709 UCSAdaptorFcErrorRecoveryProfile . . . 55
4.2.710 UCSAdaptorFcGenProfile . . . 55
4.2.711 UCSAdaptorFcInterruptProfile . . . 55
4.2.712 UCSAdaptorFcPortFLogiProfile . . . 55
4.2.713 UCSAdaptorFcPortPLogiProfile . . . 55
4.2.714 UCSAdaptorFcPortProfile . . . 55
4.2.715 UCSAdaptorFcRecvQueueProfile . . . 55
4.2.716 UCSAdaptorFcWorkQueueProfile . . . 55
4.2.717 UCSAdaptorHostEthIf . . . 55
4.2.718 UCSAdaptorHostFcIf . . . 55
4.2.719 UCSAdaptorIpV4RssHashProfile . . . 55
4.2.720 UCSAdaptorIpV6RssHashProfile . . . 55
4.2.721 UCSAdaptorPortProfiles . . . 55
4.2.722 UCSAdaptorRssProfile . . . 55
4.2.723 UCSBase . . . 55
4.2.724 UCSBiosBootDev . . . 55
4.2.725 UCSBiosBootDevGrp . . . 55
4.2.726 UCSBiosSettings . . . 55
4.2.727 UCSBiosVfAdjacentCacheLinePrefetch . . . 55
4.2.728 UCSBiosVfAltitude . . . 55
4.2.729 UCSBiosVfASPMSupport . . . 55
4.2.730 UCSBiosVfConsoleRedirection . . . 55
4.2.731 UCSBiosVfCoreMultiProcessing . . . 55
4.2.732 UCSBiosVfCPUEnergyPerformance . . . 55
4.2.733 UCSBiosVfCPUFrequencyFloor . . . 55
4.2.734 UCSBiosVfCPUPerformance . . . 55
4.2.735 UCSBiosVfCPUPowerManagement . . . 55
4.2.736 UCSBiosVfDCUPrefetch . . . 55
4.2.737 UCSBiosVfDemandScrub . . . 55
4.2.738 UCSBiosVfDirectCacheAccess . . . 55
4.2.739 UCSBiosVfDRAMClockThrottling . . . 55
4.2.740 UCSBiosVfDramRefreshRate . . . 55
4.2.741 UCSBiosVfEnhancedIntelSpeedStepTech . . . 55
4.2.742 UCSBiosVfExecuteDisableBit . . . 55
4.2.743 UCSBiosVfFRB2Enable . . . 55
4.2.744 UCSBiosVfHardwarePrefetch . . . 55
4.2.745 UCSBiosVfIntelHyperThreadingTech . . . 55
4.2.746 UCSBiosVfIntelTurboBoostTech . . . 56
4.2.747 UCSBiosVfIntelVirtualizationTechnology . . . 56
4.2.748 UCSBiosVfIntelVTForDirectedIO . . . 56
4.2.749 UCSBiosVfLegacyUSBSupport . . . 56

4.2.750 UCSBiosVfLOMPortOptionROM . . . 56
4.2.751 UCSBiosVfLvDIMMSupport . . . 56
4.2.752 UCSBiosVfMemoryInterleave . . . 56
4.2.753 UCSBiosVfMemoryMappedIOAbove4GB . . . 56
4.2.754 UCSBiosVfNUMAOptimized . . . 56
4.2.755 UCSBiosVfOnboardStorage . . . 56
4.2.756 UCSBiosVfOnboardStorageSWStack . . . 56
4.2.757 UCSBiosVfOSBootWatchdogTimer . . . 56
4.2.758 UCSBiosVfOSBootWatchdogTimerPolicy . . . 56
4.2.759 UCSBiosVfOSBootWatchdogTimerTimeout . . . 56
4.2.760 UCSBiosVfPatrolScrub . . . 56
4.2.761 UCSBiosVfPCIOptionROMs . . . 56
4.2.762 UCSBiosVfPCISlotOptionROMEnable . . . 56
4.2.763 UCSBiosVfProcessorC1E . . . 56
4.2.764 UCSBiosVfProcessorC6Report . . . 56
4.2.765 UCSBiosVfPStateCoordType . . . 56
4.2.766 UCSBiosVfQPIConfig . . . 56
4.2.767 UCSBiosVfSelectMemoryRASConfiguration . . . 56
4.2.768 UCSBiosVfTPMSupport . . . 56
4.2.769 UCSBiosVfUCSMBootOrderRuleControl . . . 56
4.2.770 UCSBiosVfUSBEmulation . . . 56
4.2.771 UCSBiosVfUSBPortsConfig . . . 56
4.2.772 UCSBiosVfVgaPriority . . . 56
4.2.773 UCSCommNtpProvider . . . 56
4.2.774 UCSCommSyslog . . . 56
4.2.775 UCSCommSyslogClient . . . 56
4.2.776 UCSEquipmentIndicatorLed . . . 56
4.2.777 UCSEquipmentLocatorLed . . . 56
4.2.778 UCSFaultInst . . . 56
4.2.779 UCSFirmwareRunning . . . 56
4.2.780 UCSInfo . . . 56
4.2.781 UCSLogs . . . 56
4.2.782 UCSLsbootDef . . . 56
4.2.783 UCSLsbootEfi . . . 56
4.2.784 UCSLsbootLan . . . 56
4.2.785 UCSLsbootStorage . . . 56
4.2.786 UCSLsbootVirtualMedia . . . 56
4.2.787 UCSStatus . . . 56
4.2.788 UGECgroupsSettings . . . 56
4.2.789 UGEClientRole . . . 56
4.2.790 UGEJob . . . 56
4.2.791 UGEJobQueue . . . 56
4.2.792 UGEJobQueueStat . . . 56
4.2.793 UGEParallelEnvironment . . . 57
4.2.794 UGEServerRole . . . 57
4.2.795 User . . . 57

4.2.796 Validation . . . 57
4.2.797 VersionInfo . . . 57
4.2.798 VirtualNode . . . 57
4.2.799 VirtualNodeSettings . . . 57
4.2.800 VirtualSMPNode . . . 57
4.2.801 VScaleMPSettings . . . 57
4.2.802 VsmpSettings . . . 57
4.2.803 WillChange . . . 57
4.2.804 WlmCgroupsSettings . . . 57
4.2.805 XeonPhiSettings . . . 57
4.2.806 ZooKeeperCluster . . . 57
4.2.807 ZooKeeperHostRole . . . 57
4.3 JSON Examples . . . 57

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

For Any Object:

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

For Any Device:

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

For Any Node:

Name	Description
`overview()`	Get the data displayed in the `cmgui` node overview tab
`sysinfo()`	Get the data displayed in the `cmgui` node system information tab
`pexec(<command>)`	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`, `sge`, `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(): 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(): 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

Monitoring Data Producers

This chapter covers how to add a new metrics and health checks scripts with `cmsh`.

Three different types of Monitoring Data Producers can be added:

- `metric`: a script which produces a single value.
- `health check`: a script which produces a `PASS`, `FAIL`, `UNKNOWN`, or `no data` value.
- `collection`: a script that produces zero or more metrics, health checks, or a combination of both.

A monitoring data producer cannot be plotted in `cmsh` or Bright View, because it contains no data. A producer defines measurables: metrics and/or health checks. It also generates data for these measurables, which can be plotted.

2.1 Measurables

There are three types of measurable:

- `metric`: a numeric value, or `no data`.
- `health check`: `PASS/FAIL/UNKNOWN/no data`.
- `enum metric`: one of a set of user-defined string based values, or `no data`.

2.2 Measurables Classes

All measurables are grouped into classes. A class is a user-defined free string field, with `/` as delimiters. Bright View uses this class to build a tree for easy search and access.

2.3 Metric Monitoring Data Producers

A metric data producer script generates one data point.

For example, as in the following script:

Example

```
[root@bright80 ~]# cat /path/to/my/metric
#!/bin/bash
echo $((RANDOM))
# Optionally provide extra information
echo "Extra information" >&3
```

The script can be defined as a metric script via the `monitoring setup` mode of `cmsh`:

Example

```
[bright80]% monitoring setup
[bright80->monitoring->setup]% add metric my-metric
[...my-metric]% set script /path/to/my/metric
[...my-metric]% set class My/Class
[...my-metric]% set unit B
[...my-metric]% set interval 1m
[...my-metric]% commit
```

All nodes then execute the script every minute, and produce a random number.

2.4 Health Check Monitoring Data Producers

A health check data producer script generates one data point. The data point can be one of four possible values expected of it: `PASS`, `FAIL`, `UNKNOWN`, or `no data`. Other file descriptors can be used to provide extra information.

For example, as in the following script:

Example

```
[root@bright80 ~]# cat /path/to/my/health-check
#!/bin/bash
if [ $((RANDOM)) -gt 8000 ]; then
  echo "PASS"
else
  echo "FAIL"
  # Optionally provide extra information
  echo "Extra information" >&3
fi
```

The script can be defined as a health check script via the `monitoring setup` mode of `cmsh`:

Example

```
[bright80]% monitoring setup
[bright80->monitoring->setup]% add healthcheck my-health-check
[...my-check]% set script /path/to/my/health-check
[...my-check]% set class My/Class
[...my-check]% set interval 1m
[...my-check]% commit
```

All nodes then execute the script every minute, and produce data values with roughly 75% `PASS` and 25% `FAIL`.

2.5 Collection Monitoring Data Producers

A *collection data producer* script can generate multiple data points in one run. Data points can be a combination of metrics and health checks. Collection scripts are also allowed to produce no data.

A collection script has two modes: initialize mode and sample mode.

- `initialize`: defines the measurables that data values are generated for.
- `sample`: returns the data values for all the measurables defined in initialize mode.

During normal cluster operation the initialize mode is called only once, during boot. Afterwards, the script is called in sample mode at the desired interval.

The following example combines both of the metric and health check examples from earlier on. However, this time it is written as a single script, using JSON as the output format:

Example

```
[root@bright80 ~]# cat /path/to/my/collection
#!/usr/bin/python

import sys
import json
import random

def initialize():
    metric = {"metric": "my.collection.metric",
              "unit": "B",
              "class": "My/Collection"}
    check = {"check": "my.collection.check",
             "class": "My/Collection"}
    return [metric, check]

def sample():
    metric = {"metric": "my.collection.metric",
              "value": random.randint(0, 32767)}
    check = {"check": "my.collection.check",
             "info": "random with 25% failure rate",
             "value": 'PASS' if random.randint(0, 32767) > 8000 else 'FAIL'}
    return [metric, check]

def main():
    if len(sys.argv) > 1 and sys.argv[1] == "--initialize":
        data = initialize()
    else:
        data = sample()
    print json.dumps(data, indent=4)

if __name__ == '__main__':
    main()
```

The script can be defined as a collection script via the `monitoring setup` mode of `cmsh`:

Example

```
[bright80]% monitoring setup
[bright80->monitoring->setup]% add collection my-collection
[...my-collection]% set script /path/to/my/collection
[...my-collection]% set format JSON
[...my-collection]% set interval 1m
[...my-collection]% commit
```

All nodes then execute the script every minute and produce two data points upon each execution. That is, one metric and one health check per execution.

2.6 Node Execution Filters

By default a monitoring data producer script is executed on every node. When this is not desirable, a node execution filter should be created. A node execution filter defines the nodes on which the producer script should be executed.

For example, a filter to execute the script only on cloud nodes can be configured as follows:

Example

```
[bright80]% monitoring setup use my-check
[...my-check]% nodeexecutionfilters
[...nodeexecutionfilters]% add type Cloud
[...nodeexecutionfilters*[Cloud*]]% set cloudnode yes
[...nodeexecutionfilters*[Cloud*]]% show
Parameter          Value
------------------ --------------------------
Base type          MonitoringExecutionFilter
Name               Cloud
Type               Type
Head node          no
Physical node      no
Cloud node         yes
Virtual node       no
Lite node          no
[...nodeexecutionfilters*[Cloud*]]% commit
```

It is also possible to filter based on the specific resources associated with a node:

Example

```
[bright80]% monitoring setup use my-IB-check
[...my-IB-check]% nodeexecutionfilters
[...nodeexecutionfilters]% add resource IB
[...nodeexecutionfilters*[IB*]]% set resources IB
[...nodeexecutionfilters*[IB*]]% commit
```

Because of high availability, a special resource, `active`, is defined for the active head node.

Example

```
[bright80]% monitoring setup use my-metric
[...my-metric]% nodeexecutionfilters
[...nodeexecutionfilters]% active
Added active resource filter
[...nodeexecutionfilters*]% commit
```

2.7 Execution Multiplexers

By default a monitoring data producer script is executed once: the node executes the script only for itself.

However, some scripts, such as BMC samplers, must be sampled from the active head node for all nodes.

In the following example a BMC script is run on each node that has the `ipmi` or `drac` resource:

Example

```
[bright80]% monitoring setup use my-ipmi-collection
[...my-ipmi-collection]% executionmultiplexers
[...executionmultiplexers]% add resource ipmi
[...executionmultiplexers*[ipmi*]]% set resources ipmi drac
[...executionmultiplexers*[ipmi*]]% set operator OR
[...executionmultiplexers*[ipmi*]]% commit
```

If an execution multiplexer <*multiplexer*> is defined, then there should also be a node execution filter <*filter*> associated with it to restrict the number of nodes on which the script runs.

This is because having the script run on many nodes for many other nodes is unlikely to be a desired configuration.

The combination of the execution filter and the multiplexer should be read as:

for every node that matches *filter*, run script, for each node that matches *multiplexer*.

A more specific example, using two of the preceding examples, with a filter based on the resource `IB`, and multiplexers based on the IPMI/Drac resources, the combination should be read as:

for every node that matches `IB`, run script, for each node that matches `ipmi` or `drac`.

2.8 Monitoring Resources

Every device in Bright Cluster Manager has one or more resources. These resources are automatically calculated from:

- Roles
- Hardware
- Settings

Resources for a specific node can be viewed as follows:

Example

```
[bright80]% device use node001
[bright80]% monitoringresources
CentOS7u5
Ethernet
category:default
```

It is possible to add one or more custom resources to a device:

Example

```
[bright80]% device use node001
[bright80]% add userdefinedresources MyResource
[bright80]% append userdefinedresources MyOtherResource
[bright80]% # wait ~10 seconds for the settings to propagate
[bright80]% monitoringresources
CentOS7u5
Ethernet
category:default
MyResource
MyOtherResource
```

Any of these resources can be used to filter and multiplex monitoring data producers.

If a resources changes because of a settings change, then monitoring automatically stops or starts sampling.

2.9 Collection Monitoring Data Producers With Filter And Multiplexer

If a script has an execution multiplexer set, then it needs to determine for which nodes the script runs:

Example

```
[root@bright80~]# cat /path/to/my/collection
#!/usr/bin/python

import sys
import json
import random

def initialize(entity):
    metric = {"metric": "my.collection.metric",
              "entity": entity,
              "unit": "B",
              "class": "My/Collection"}
    check = {"check": "my.collection.check",
             "entity": entity,
             "class": "My/Collection"}
    return [metric, check]

def sample(entity):
    metric = {"metric": "my.collection.metric",
              "entity": entity,
              "value": random.randint(0, 32767)}
    check = {"check" : "my.collection.check",
             "entity": entity,
             "value" : 'PASS' if random.randint(0, 32767) > 8000 else 'FAIL'}
    return [metric, check]

def main():
    try:
        # determine for which node we are sampling
        entity = os.environ['CMD_HOSTNAME']
    except:
        sys.stderr.write('Target device not specified in environment\n')
        return

    if len(sys.argv) > 1 and sys.argv[1] == "--initialize":
        data = initialize(entity)
    else:
        data = sample(entity)
    print json.dumps(data, indent=4)

if __name__ == '__main__':
    main()
```

It can be defined with a filter to run on the active head for all nodes in the GPU category:

Example

```
[bright80]% monitoring setup
[bright80->monitoring->setup]% add collection my-collection
[...my-collection]% set script /path/to/my/collection
[...my-collection]% set format JSON
[...my-collection]% set interval 1m
[...my-collection]% nodeexecutionfilters
[...nodeexecutionfilters]% active
Added active resource filter
[...nodeexecutionfilters]% exit
```

```
[...my-collection]% executionmultiplexers
[...executionmultiplexers]% add category
[...executionmultiplexers*[GPU*]% add category GPU
[...executionmultiplexers*[GPU*]% commit
```

The script is then executed on the head, once for each node in the category of `GPU`.

2.10 Collection Monitoring Data Producers For Standalone Entities

Sometimes monitoring data does not belong to a Bright Cluster Manager entity.

For this reason the standalone monitored entity was added in Bright Cluster Manager 8.0.

This entity can be anything with a name and custom type.

Bright Cluster Manager does nothing with this kind of entity, except allow it to store monitoring data.

Each standalone entity which needs to be monitored should be added:

Example

```
[bright80]% monitoring standalone
[bright80->monitoring->standalone]% add MSD.0
[...standalone*[MSD.0*]]% set type Lustre
[...standalone*[MSD.0*]]% commit
[...standalone*[MSD.0*]]% add MSD.1
[...standalone*[MSD.1*]]% set type Lustre
[...standalone*[MSD.1*]]% commit
```

A script can be created that produces data for all MSD entities:

Example

```
[root@bright80 ~]# cat /path/to/my/collection
#!/usr/bin/python

import sys
import json

def initialize():
    msd_0 = {"metric": "lustre.free.space",
             "entity": "MSD.0",
             "unit": "B",
             "class": "Lustre"}
    msd_1 = {"metric": "lustre.free.space",
             "entity": "MSD.1",
             "unit": "B",
             "class": "Lustre"}
    return [msd_0, msd_1]

def sample():
    msd_0 = {"metric": "lustre.free.space",
             "entity": "MSD.0",
             "value": 12345}
             "class": "Lustre"}
    msd_1 = {"metric": "lustre.free.space",
             "entity": "MSD.1",
             "value": 54321}
    return [msd_0, msd_1]
```

```
def main():
    if len(sys.argv) > 1 and sys.argv[1] == "--initialize":
        data = initialize()
    else:
        data = sample()
    print json.dumps(data, indent=4)

if __name__ == '__main__':
    main()
```

It can be defined to run on only the active head node:

Example

```
[bright80]% monitoring setup
[bright80->monitoring->setup]% add collection my-collection
[...my-collection]% set script /path/to/my/collection
[...my-collection]% set format JSON
[...my-collection]% set interval 5m
[...my-collection]% nodeexecutionfilters
[...nodeexecutionfilters]% active
Added active resource filter
[...nodeexecutionfilters]% commit
```

The script is then executed on the active head every 5 minutes and collects one data point for each MSD.

Data for a standalone script can be viewed with the same commands as for regular Bright Cluster Manager nodes.

Example

```
[bright80]% monitoring standalone
[bright80->monitoring->standalone]% use MSD.0
[...standalone*[MSD.0*]]% latestmetricdata
...
lustre.free.space        12345         3m 47s
```

3

Monitoring Actions

This chapter covers how to manage monitoring-driven actions with `cmsh`.

3.1 Actions And Triggers

A monitoring action is a script that is executed by CMDaemon. It runs when triggered by the monitored data.

An action by itself does nothing—it needs a trigger (section 12.4.5 of the *Administrator Manual*) to be defined to execute the action.

By default, several actions (section 12.4.4 of the *Administrator Manual*) are predefined:

- `Drain`: Drain node (node refuses new WLM jobs)
- `Event`: Send an event to users with connected client
- `ImageUpdate`: Update the image on the node
- `PowerOff`: Power off a device
- `PowerOn`: Power on a device:
- `PowerReset`: Power reset a device
- `Reboot`: Reboot a node
- `Send e-mail to administrators`: Send e-mail
- `Shutdown`: Shutdown a node
- `Undrain`: Undrain node (node accepts new WLM jobs)
- `killprocess`: `/cm/local/apps/cmd/scripts/actions/killprocess.pl`
- `remount`: `/cm/local/apps/cmd/scripts/actions/remount`
- `testaction`: `/cm/local/apps/cmd/scripts/actions/testaction`

A new action script can be created as follows:

Example

```
[bright80]% monitoring action
[bright80->monitoring->action]% add script MyScript
[...MyScript*]% set script /path/to/MyScript
[...MyScript*]% commit
```

3.2 Time Restrictions

It is possible to allow actions to only be executed at certain times, with the `allowedtime` setting.

Example

```
[bright80]% monitoring action
[bright80->monitoring->action]% add script MyScript
[...MyScript*]% set script /path/to/MyScript
[...MyScript*]% set allowedtime "9:00-17:00"
[...MyScript*]% commit
```

More complex timing restrictions are possible:

Example

```
monday-friday{9:00-17:00}
monday-friday{00:00-09:00;17:00-00:00};saturday-sunday
november-march{monday-saturday{13:00-17:00}}
may-september{monday-friday{09:00-18:00};saturday-sunday{13:00-17:00}}
```

Further examples can be seen in section 12.4.4 of the *Administrator Manual*, page 441.

3.2.1 Time Restriction Syntax In BNF Notation

The allowed values can be written as a BNF grammar:

Example

```
<start> =
  time_intervals
  | ""
<time_intervals> = <time_interval> (; <time_interval>)*
<time_interval> = <inner_time_interval>{<time_intervals}
<inner_time_interval> =
  <day_of_week_interval>
  | <time_of_day_interval>
  | <day_of_month_interval>
  | <month_interval>
<day_of_week_interval> =
  (<day_of_week>-<day_of_week>)
  | (<day_of_week> (, day_of_week)*)
<day_of_week> = sunday | monday | tuesday | wednesday | thursday | friday | saturday
<time_of_day_interval> = <time_of_day>-<time_of_day>
<time_of_day>= \d?\d:\d\d
<month_interval> = (<month>-<month>)
  | (<month> (, month)*)
<month> = january | february | march | april | may | june | july | august | september
  | october | november | december
<day_of_month_interval> = (<day_of_month>-<day_of_month>)
  | (<day_of_month> (, day_of_month)*)
<day_of_month> = \d?\d
```

3.3 CMDaemon Environment Variables

3.3.1 Standard Environment Variables Available In Action Scripts

Name	Description
CMD_ENTITY_KEY	The unique key of the entity that triggered the action.
CMD_ENTITY_NAME	The name of the entity that triggered the action.
CMD_ENTITY_TYPE	The type of entity that triggered the action.
CMD_MEASURABLE_NAME	The name of the measurable that triggered the action.
CMD_MEASURABLE_PARAMETER	The parameter of the measurable that triggered the action.
CMD_MEASURABLE_TYPE	The type of the measurable.
CMD_VALUE	The value that triggered the action.
CMD_RAW_VALUE	The raw value.
CMD_VALUE_TIME	The time on which the value was measured.
CMD_INFO_MESSAGE	Extra information sampled along with the value.
CMD_PRODUCER_NAME	The name of the monitoring data producer that samples the measurable.

...continues

...continued

Name	Description
CMD_ACTION_NAME	The name of the action that was triggered.
CMD_TRIGGER_NAME	The name of the trigger.
CMD_TRIGGER_EXPRESSION	The expression that was evaluated.
CMD_VALUE_EVAL	The result of the evaluated expression.
CMD_VALUE_COUNT	The number of times the expression evaluated to the same value.
CMD_SEVERITY	The assigned severity of the trigger.

All action scripts have the preceding standard environment variables set.

In `cmsh`, if the action object has its `node environment` parameter set to the value `yes`, then scripts running on a node are enabled with an extended environment that provides many more `CMD_*` environment variables. Otherwise they run in the standard environment.

A list of the standard or extended environment variables can be dumped by running the system command `env > /tmp/dumpfile` within an action script, such as the test example script, and triggering the script to run.

Many of the environment variables are similar to the ones used by initialize and finalize scripts (section E.3 of the *Administrator Manual*) in the node-installer environment.

3.3.2 Extended Environment Variables Available To Action Scripts

If the action object has its `node environment` parameter set to the value `yes`, then scripts run in an extended environment that provides many more `CMD_*` environment variables. Otherwise they run in the standard environment of section 3.3.1.

The following table shows the additionally available environment variables with some example values:

Table 3.3.2: Environment Variables For Nodes In The Extended Environment

Variable	Example Value
CMD_ACTIVE_MASTER_IP	10.141.255.254

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_ADDED_NODES	
CMD_BASE_TYPE	
CMD_BMCIP	
CMD_BMCPASSWORD	doQNeV1qksXr59O
CMD_BMCUSERID	4
CMD_BMCUSERNAME	
CMD_BMC_TYPE	2
CMD_CATEGORY	default
CMD_CEPH_MDS_SOCKET	
CMD_CEPH_MGR_SOCKET	
CMD_CEPH_MON_SOCKET	
CMD_CEPH_NAME	
CMD_CEPH_OSD_ID	
CMD_CEPH_OSD_SOCKET	
CMD_CHASSIS	chassis01
CMD_CHASSIS_IP	10.141.1.1
CMD_CHASSIS_MEMBERS	
CMD_CHASSIS_PASSWORD	secr3t
CMD_CHASSIS_SLOT	1

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_CHASSIS_USERNAME	ADMIN
CMD_CHILD_TYPE	
CMD_CLUSTERNAME	Bright 8.0 Cluster
CMD_CONFIGURATION_CREATE_DIRECTORY	
CMD_CONFIGURATION_FILENAME	
CMD_CONFIGURATION_GROUP_NAME	
CMD_CONFIGURATION_MASK	
CMD_CONFIGURATION_NAME	
CMD_CONFIGURATION_USER_NAME	
CMD_CREATE_RAMDISK_TOKEN_CATS	
CMD_CREATE_RAMDISK_TOKEN_NODES	
CMD_CURRENT_NODES	
CMD_DATA	
CMD_DELLFW_FTP_PASSWORD	
CMD_DELLFW_FTP_USERNAME	
CMD_DELLFW_PATH	
CMD_DESTINATION_REVISION	
CMD_DESTINATION_VERSION	
CMD_DEVICE_HEIGHT	1

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_DEVICE_POSITION	10
CMD_DEVICE_TYPE	ComputeNode
CMD_DIRECTOR	
CMD_DIRECTOR_IP	
CMD_DOCKER_ENDPOINTS	
CMD_EDGE_SITE	
CMD_ETCD_CA	
CMD_ETCD_CAKEY	
CMD_ETCD_CLIENT_CA	
CMD_ETCD_CLIENT_CERT	
CMD_ETCD_CLIENT_KEY	
CMD_ETCD_MEMBER_CERT	
CMD_ETCD_MEMBER_KEY	
CMD_ETHERNETSWITCH	switch01:1
CMD_EXISTING_REVISION	
CMD_EXISTING_VERSION	
CMD_EXPORTS	
CMD_FAILONMISSINGBMC	
CMD_FAIL_ON_FAILED_BMCCOMMAND	YES

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_FSEXPORTS	
CMD_FSEXPORT_<*name*>_ALLOWWRITE	
CMD_FSEXPORT_<*name*>_HOSTS	
CMD_FSEXPORT_<*name*>_PATH	
CMD_FSMOUNTS	
CMD_FSMOUNT_<*name*>_DEVICE	

where <*name*> takes these SLASH substitutions:

<*name*>	**example value**
_SLASH_cm_SLASH_shared	$localnfsserver:/cm/shared
_SLASH_dev_SLASH_pts	*none*
_SLASH_dev_SLASH_shm	*none*
_SLASH_home	$localnfsserver:/home
_SLASH_proc	*none*
_SLASH_sys	*none*

CMD_FSMOUNT_<*name*>_FILESYSTEM

where <*name*> takes these SLASH substitutions:

<*name*>	**example value**
_SLASH_cm_SLASH_shared	nfs
_SLASH_dev_SLASH_pts	devpts
_SLASH_dev_SLASH_shm	tmpfs
_SLASH_home	nfs
_SLASH_proc	proc
_SLASH_sys	sysfs

CMD_FSMOUNT_<*name*>_MOUNTPOINT

where <*name*> takes these SLASH substitutions:

<*name*>	**example value**
_SLASH_cm_SLASH_shared	/cm/shared
_SLASH_dev_SLASH_pts	/dev/pts
_SLASH_dev_SLASH_shm	/dev/shm
_SLASH_home	/home
_SLASH_proc	/proc
_SLASH_sys	/sys

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_FSMOUNT_<*name*>_OPTIONS	

where <*name*> takes these SLASH substitutions:

<name>	example value
_SLASH_cm_SLASH_shared	rsize=32768,wsize=32768,hard,intr,async
_SLASH_dev_SLASH_pts	gid=5,mode-620
_SLASH_dev_SLASH_shm	defaults
_SLASH_home	rsize=32768,wsize=32768,hard,intr,async
_SLASH_proc	defaults,nosuid
_SLASH_sys	/defaults

CMD_GATEWAY	10.141.255.254
CMD_GUID	
CMD_HAPROXY_HOST	
CMD_HOSTNAME	node004
CMD_INITRD	
CMD_INITRD_KERNEL_PARAMS	
CMD_INITRD_KERNEL_VERSION	
CMD_INITRD_TMPFS_SIZE	
CMD_INSTALLMODE	AUTO
CMD_INSTANCE_ID	
CMD_INTERFACES	BOOTIF
CMD_INTERFACE_<*interface*>_BOND	
CMD_INTERFACE_<*interface*>_BRIDGE	
CMD_INTERFACE_<*interface*>_DHCP	
CMD_INTERFACE_<*interface*>_GATEWAY	

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_INTERFACE_<*interface*>_IP	10.141.0.5
CMD_INTERFACE_<*interface*>_LANCHANNEL	
CMD_INTERFACE_<*interface*>_MAC	00:00:00:00:00:00
CMD_INTERFACE_<*interface*>_MODE	
CMD_INTERFACE_<*interface*>_MTU	1500
CMD_INTERFACE_<*interface*>_NETMASK	
CMD_INTERFACE_<*interface*>_REVISION	
CMD_INTERFACE_<*interface*>_SLAVES	
CMD_INTERFACE_<*interface*>_SPEED	
CMD_INTERFACE_<*interface*>_STARTIF	ALWAYS
CMD_INTERFACE_<*interface*>_TYPE	NetworkPhysicalInterface
CMD_INTERFACE_<*interface*>_VLANID	

In the preceding CMD_INTERFACE_* variables, <*interface*> can take the following substitutions for the network interface:

possible values for <*interface*>
BOOTIF
drac0, drac1, drac2...
cimc0, cimc1, cimc2...
eth0, eth1, eth1...
ib0, ib1, ib2...
ilo0, ilo1, ilo2...
ipmi0, ipmi1, ipmi2...
rf0, rf1, rf2...
eno1, enp0s18f2, and other names consistent with the RHEL7 interface naming convention

CMD_IP	10.141.0.1
CMD_JOBNODELIST	
CMD_KUBERNETES_ADMIN_CERT	

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_KUBERNETES_ADMIN_CERT_KEY	
CMD_KUBERNETES_ADMIN_KUBECONFIG	
CMD_KUBERNETES_APISERVER_ENDPOINT	
CMD_KUBERNETES_CACERT	
CMD_KUBERNETES_CLIENT_CERTIFICATE	
CMD_KUBERNETES_CLIENT_KEY	
CMD_KUBERNETES_ETCD_ACTIVE	
CMD_KUBERNETES_ETCD_CLIENT_ENDPOINTS	
CMD_KUBERNETES_KUBELET_CERTIFICATE	
CMD_KUBERNETES_KUBELET_ENDPOINT	
CMD_KUBERNETES_KUBELET_KEY	
CMD_KUBE_DNS_IP	
CMD_KUBE_DOMAIN	
CMD_KUBE_INTERNAL_NETWORK_CIDR	
CMD_KUBE_POD_NETWORK_CIDR	
CMD_KUBE_SERVICE_NETWORK_CIDR	
CMD_LOGGING_CONFIG	
CMD_MAC	FA:16:3E:64:8E:1E
CMD_MODEL	

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_MODULES	
CMD_MODULE_<*name*>	
CMD_MOUNTS	
CMD_NAME	
CMD_NODEGROUPS	
CMD_NODEGROUP_NAME	
CMD_NODEGROUP_UID	
CMD_OPENSTACK_CINDER_PASSWORD	
CMD_OPENSTACK_CINDER_USERNAME	
CMD_OPENSTACK_ENABLED	
CMD_OPENSTACK_GLANCE_PASSWORD	
CMD_OPENSTACK_GLANCE_USERNAME	
CMD_OPENSTACK_KEYSTONE_INTERNAL_URL	
CMD_OPENSTACK_KEYSTONE_PASSWORD	
CMD_OPENSTACK_KEYSTONE_PUBLIC_URL	
CMD_OPENSTACK_KEYSTONE_URL	
CMD_OPENSTACK_KEYSTONE_USERNAME	
CMD_OPENSTACK_MANAGER_PASSWORD	
CMD_OPENSTACK_MANAGER_USERNAME	

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_OPENSTACK_MESSAGE_QUEUE_HOSTS	
CMD_OPENSTACK_MESSAGE_QUEUE_PASSWORD	
CMD_OPENSTACK_MESSAGE_QUEUE_USERNAME	
CMD_OPENSTACK_NOVA_PASSWORD	
CMD_OPENSTACK_NOVA_USERNAME	
CMD_OPENSTACK_REGIONS	
CMD_OPENSTACK_SERVICE_TENANT	
CMD_OPENSTACK_SSL_ENABLED	
CMD_OPENSTACK_TENANTS	
CMD_OWNED_INDEX	
CMD_PARTITION	base
CMD_PASSIVE_MASTER_IP	10.141.255.253
CMD_PDUS	
CMD_PORT	8081
CMD_PORTS	
CMD_POWER_CONTROL	custom
CMD_PROTOCOL	https
CMD_RACADM_PATH	
CMD_RACK	rack01

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_RACK_HEIGHT	42
CMD_RACK_ROOM	serverroom
CMD_READ_STRING	
CMD_REMOVED_NODES	
CMD_RESOLVE_NAME	
CMD_ROLES	
CMD_SCRIPTTIMEOUT	5
CMD_SCRIPT_TIMEOUT	5
CMD_SHARED_MASTER_IP	10.141.255.252
CMD_SKIPBMC	
CMD_SOFTWAREIMAGE	default-image
CMD_SOFTWAREIMAGE_PATH	/cm/images/default-image
CMD_STATE	
CMD_STATUS	
CMD_STATUS_CLOSED	NO
CMD_STATUS_HEALTHCHECK_FAILED	NO
CMD_STATUS_HEALTHCHECK_UNKNOWN	NO
CMD_STATUS_MESSAGE	
CMD_STATUS_RESTART_REQUIRED	NO

...continues

Table 3.3.2: Environment Variables For Nodes In The Extended Environment ...continued

Variable	Example Value
CMD_STATUS_STATEFLAPPING	NO
CMD_STATUS_USERMESSAGE	
CMD_STRICTUSERID	
CMD_SUBNET_MANAGER	
CMD_SWITCH_CONTROL_SCRIPT	
CMD_SWITCH_CONTROL_SCRIPT_TIMEOUT	
CMD_SYSINFO_SYSTEM_MANUFACTURER	RDO
CMD_SYSINFO_SYSTEM_NAME	OpenStack Compute
CMD_TAG	00000000a000
CMD_TARGET_NAME	
CMD_TARGET_NODES	
CMD_TYPE	
CMD_TYPES	
CMD_UCS_DN	sys/rack-unit-1
CMD_USERDEFINED1	var1
CMD_USERDEFINED2	var2
CMD_VMLINUZ	
CMD_WRITE_STRING	

4

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 4.3

4.1 Services

4.1.1 auth
4.1.2 ceph
4.1.3 cert
4.1.4 cloud
4.1.5 device
4.1.6 etcd
4.1.7 gui
4.1.8 hadoop
4.1.9 job
4.1.10 keyvalue
4.1.11 kube
4.1.12 lustre
4.1.13 main
4.1.14 mesos
4.1.15 mon
4.1.16 net
4.1.17 openstack
4.1.18 part
4.1.19 proc
4.1.20 prov
4.1.21 puppet
4.1.22 serv
4.1.23 session

4.1.24 sync
4.1.25 test
4.1.26 ticket
4.1.27 user
4.1.28 zookeeper

4.2 Entities

4.2.1 AzureDataDisk
4.2.2 AzureDisk
4.2.3 AzureIntermediateStorage
4.2.4 AzureLocation
4.2.5 AzureManagedDiskParameters
4.2.6 AzureOSDisk
4.2.7 AzureProvider
4.2.8 AzurePublicIP
4.2.9 AzureSettings
4.2.10 AzureVMSize
4.2.11 BadEntityManagers
4.2.12 BasicResource
4.2.13 BeeGFSAdmonRole
4.2.14 BeeGFSClientRole
4.2.15 BeeGFSManagementRole
4.2.16 BeeGFSMetadataRole
4.2.17 BeeGFSStorageRole
4.2.18 BigDataAdditionalTool
4.2.19 BigDataAdvancedSettings
4.2.20 BigDataCassandra
4.2.21 BigDataFileSystemSettings
4.2.22 BigDataJobManagementSettings
4.2.23 BigDataLoggingSettings
4.2.24 BigDataSecurity
4.2.25 BigDataSpark
4.2.26 BillingHistory
4.2.27 BMCSettings
4.2.28 BootRole
4.2.29 BurnConfig
4.2.30 BurnStatus
4.2.31 BurnTestStatus
4.2.32 Category
4.2.33 Ceph
4.2.34 CephMDSRole
4.2.35 CephMGRRole
4.2.36 CephMonitorRole
4.2.37 CephOSDBlueStoreConfig
4.2.38 CephOSDConfig
4.2.39 CephOSDFileStoreConfig
4.2.40 CephOSDLegacyConfig

4.2.41 CephOSDPool
4.2.42 CephOSDRole
4.2.43 CephState
4.2.44 Certificate
4.2.45 CertificateRequest
4.2.46 CertificateSubjectName
4.2.47 Cgroup
4.2.48 CgroupController
4.2.49 CgroupControllerBlkio
4.2.50 CgroupControllerCpu
4.2.51 CgroupControllerCpuacct
4.2.52 CgroupControllerCpuset
4.2.53 CgroupControllerDevices
4.2.54 CgroupControllerFreezer
4.2.55 CgroupControllerHugetlb
4.2.56 CgroupControllerMemory
4.2.57 CgroupControllerNetcls
4.2.58 CgroupControllerNetprio
4.2.59 CgroupControllerNs
4.2.60 CgroupControllerPerf
4.2.61 CgroupRule
4.2.62 CgroupSupervisorRole
4.2.63 Chassis
4.2.64 ChronosRole
4.2.65 ClientUserData
4.2.66 CloudDirectorRole
4.2.67 CloudGatewayRole
4.2.68 CloudImage
4.2.69 CloudJobDescription
4.2.70 CloudJobSubmissionStatus
4.2.71 CloudNode
4.2.72 CloudPrivateCloud
4.2.73 CloudProvider
4.2.74 CloudRegion
4.2.75 CloudSettings
4.2.76 CloudStaticIP
4.2.77 CloudStorageActionData
4.2.78 CloudStorageNodeState
4.2.79 CloudType
4.2.80 CloudVirtualNetworkInterface
4.2.81 ClusterSetup
4.2.82 CMDaemonBackgroundTask
4.2.83 CMDaemonFailover
4.2.84 CMDaemonFailoverGroup
4.2.85 CMDaemonFailoverGroupStatus
4.2.86 CMDaemonFailoverPeer
4.2.87 CMDaemonFailoverStatus

4.2.88 CMDaemonStatus
4.2.89 CMService
4.2.90 CMSubConfig
4.2.91 CMSubIntermediateStorage
4.2.92 ConfigFileVersion
4.2.93 ConfigSum
4.2.94 ConfigurationOverlay
4.2.95 Consolidator
4.2.96 ContainerdHostRole
4.2.97 ContainerInfo
4.2.98 CustomizationEntry
4.2.99 CustomizationFile
4.2.100 DellClustat
4.2.101 DellClustatGroup
4.2.102 DellClustatNode
4.2.103 DellDiskGroupInfo
4.2.104 DellPhysicalDiskDriveInfo
4.2.105 DellRAIDControllerInfo
4.2.106 DellSettings
4.2.107 DellSettingsFirmware
4.2.108 DellSettingsNicDevice
4.2.109 DellStorageInfo
4.2.110 DellVirtualDiskInfo
4.2.111 Device
4.2.112 DevStatus
4.2.113 DiskAssertion
4.2.114 DiskDevice
4.2.115 DiskInfo
4.2.116 DiskPartition
4.2.117 DiskRaid
4.2.118 DiskSetup
4.2.119 DiskVolume
4.2.120 DiskVolumeGroup
4.2.121 DockerHostRole
4.2.122 DockerRegistryFilesystemStorageDriver
4.2.123 DockerRegistryInmemoryStorageDriver
4.2.124 DockerRegistryRole
4.2.125 DockerRegistryStorageDriver
4.2.126 DockerStorageBackend
4.2.127 DockerStorageDeviceMapperBackend
4.2.128 DrainAction
4.2.129 DrainResult
4.2.130 EC2AMI
4.2.131 EC2AvailabilityZone
4.2.132 EC2EBSStorage
4.2.133 EC2EphemeralStorage
4.2.134 EC2PrivateCloud

4.2.135 EC2Provider
4.2.136 EC2Region
4.2.137 EC2RegionAMI
4.2.138 EC2Settings
4.2.139 EC2StaticIP
4.2.140 EC2Storage
4.2.141 EC2Type
4.2.142 EC2VirtualNetworkInterface
4.2.143 ElasticSearchRole
4.2.144 EntityManagersMD5
4.2.145 EtcdCluster
4.2.146 EtcdHostRole
4.2.147 EthernetSwitch
4.2.148 FailoverRole
4.2.149 FakeJob
4.2.150 FakeJobQueue
4.2.151 FakeJobQueueStat
4.2.152 FakeWlmClientRole
4.2.153 FakeWlmServerRole
4.2.154 FileInfo
4.2.155 FileSyncConfig
4.2.156 FileSyncStatus
4.2.157 FlannelConfigurationRole
4.2.158 FlannelHostRole
4.2.159 FlannelNetworkingBackend
4.2.160 FlannelNetworkingUdpBackend
4.2.161 FlannelNetworkingVxLanBackend
4.2.162 FSExport
4.2.163 FSMount
4.2.164 FSPart
4.2.165 FSPartAssociation
4.2.166 FSPartBasicAssociation
4.2.167 FSPartProviderAssociation
4.2.168 GaleraRole
4.2.169 GenericDevice
4.2.170 GenericResource
4.2.171 GPUInfo
4.2.172 GPUSettings
4.2.173 GpuUnit
4.2.174 GPUUnitInfo
4.2.175 GridEngineJob
4.2.176 GridEngineJobQueue
4.2.177 GridEngineJobQueueStat
4.2.178 GridEngineParallelEnvironment
4.2.179 Group
4.2.180 GuiCephOsdPoolInfo
4.2.181 GuiCephOverview

4.2.182 GuiCephPgsInfo
4.2.183 GuiClusterOverview
4.2.184 GuiCompleteOpenStackOverview
4.2.185 GuiDiskUsage
4.2.186 GuiGpuUnitOverview
4.2.187 GuiHadoopHDFSDetailHBase
4.2.188 GuiHadoopHDFSDetailHDFS
4.2.189 GuiHadoopHDFSDetailMapreduce
4.2.190 GuiHadoopHDFSDetailSpark
4.2.191 GuiHadoopHDFSDetailYarn
4.2.192 GuiHadoopHDFSDetailZooKeeper
4.2.193 GuiHadoopHDFSOverview
4.2.194 GuiJob
4.2.195 GuiKubeClusterOverview
4.2.196 GuiNetworkInterface
4.2.197 GuiNodeOverview
4.2.198 GuiNodeStatus
4.2.199 GuiOpenStackOverview
4.2.200 GuiOpenStackProjectOverview
4.2.201 GuiOpenStackTenantOverview
4.2.202 GuiPDUBank
4.2.203 GuiPDUOutlet
4.2.204 GuiPDUOverview
4.2.205 GuiSwitchOverview
4.2.206 GuiSwitchPort
4.2.207 GuiWorkload
4.2.208 HadoopAccumuloMasterHDFSConfiguration
4.2.209 HadoopAccumuloMasterRole
4.2.210 HadoopAccumuloTabletHDFSConfiguration
4.2.211 HadoopAccumuloTabletRole
4.2.212 HadoopAlluxioMasterHDFSConfiguration
4.2.213 HadoopAlluxioMasterRole
4.2.214 HadoopAlluxioWorkerHDFSConfiguration
4.2.215 HadoopAlluxioWorkerRole
4.2.216 HadoopBaseConfiguration
4.2.217 HadoopCassandraHDFSConfiguration
4.2.218 HadoopCassandraRole
4.2.219 HadoopDataNodeHDFSConfiguration
4.2.220 HadoopDataNodeRole
4.2.221 HadoopDrillHDFSConfiguration
4.2.222 HadoopDrillRole
4.2.223 HadoopFlinkJobManagerHDFSConfiguration
4.2.224 HadoopFlinkJobManagerRole
4.2.225 HadoopFlinkTaskManagerHDFSConfiguration
4.2.226 HadoopFlinkTaskManagerRole
4.2.227 HadoopHBaseClientHDFSConfiguration
4.2.228 HadoopHBaseClientRole

4.2.229 HadoopHBaseServerHDFSConfiguration
4.2.230 HadoopHBaseServerRole
4.2.231 HadoopHDFS
4.2.232 HadoopHiveHDFSConfiguration
4.2.233 HadoopHiveRole
4.2.234 HadoopJob
4.2.235 HadoopJobQueue
4.2.236 HadoopJobQueueStat
4.2.237 HadoopJobTrackerHDFSConfiguration
4.2.238 HadoopJobTrackerRole
4.2.239 HadoopJournalHDFSConfiguration
4.2.240 HadoopJournalRole
4.2.241 HadoopKafkaServerHDFSConfiguration
4.2.242 HadoopKafkaServerRole
4.2.243 HadoopKMServerHDFSConfiguration
4.2.244 HadoopKMServerRole
4.2.245 HadoopNameNodeHDFSConfiguration
4.2.246 HadoopNameNodeRole
4.2.247 HadoopNFSGatewayHDFSConfiguration
4.2.248 HadoopNFSGatewayRole
4.2.249 HadoopPigHDFSConfiguration
4.2.250 HadoopPigRole
4.2.251 HadoopSecondaryNameNodeHDFSConfiguration
4.2.252 HadoopSecondaryNameNodeRole
4.2.253 HadoopSparkMasterHDFSConfiguration
4.2.254 HadoopSparkMasterRole
4.2.255 HadoopSparkWorkerHDFSConfiguration
4.2.256 HadoopSparkWorkerRole
4.2.257 HadoopSparkYARNHDFSConfiguration
4.2.258 HadoopSparkYARNRole
4.2.259 HadoopSqoopHDFSConfiguration
4.2.260 HadoopSqoopRole
4.2.261 HadoopStormNimbusHDFSConfiguration
4.2.262 HadoopStormNimbusRole
4.2.263 HadoopStormSupervisorHDFSConfiguration
4.2.264 HadoopStormSupervisorRole
4.2.265 HadoopTaskTrackerHDFSConfiguration
4.2.266 HadoopTaskTrackerRole
4.2.267 HadoopYARNClientHDFSConfiguration
4.2.268 HadoopYARNClientRole
4.2.269 HadoopYARNServerHDFSConfiguration
4.2.270 HadoopYARNServerRole
4.2.271 HadoopZeppelinHDFSConfiguration
4.2.272 HadoopZeppelinRole
4.2.273 HadoopZooKeeperHDFSConfiguration
4.2.274 HadoopZooKeeperRole
4.2.275 HAProxyEntry

4.2.276 HAProxyEntryBind
4.2.277 HAProxyRole
4.2.278 HAProxyServer
4.2.279 HAProxySharedSettings
4.2.280 IBSwitch
4.2.281 IPCPerm
4.2.282 IPResource
4.2.283 Job
4.2.284 JobInfo
4.2.285 JobInfoStatistics
4.2.286 JobQueue
4.2.287 JobQueuePlaceholder
4.2.288 JobQueueStat
4.2.289 KeepalivedEntry
4.2.290 KeepalivedRole
4.2.291 KernelModule
4.2.292 KeyValuePair
4.2.293 KibanaRole
4.2.294 KubeCluster
4.2.295 KubePodInfo
4.2.296 KubernetesApiServerRole
4.2.297 KubernetesControllerRole
4.2.298 KubernetesDashboardRole
4.2.299 KubernetesDNSRole
4.2.300 KubernetesHeapsterRole
4.2.301 KubernetesNodeRole
4.2.302 KubernetesProxyRole
4.2.303 KubernetesSchedulerRole
4.2.304 KubeUserPolicy
4.2.305 LicenseInfo
4.2.306 LoginRole
4.2.307 LogstashForwarderRole
4.2.308 LogstashServerCustomFilter
4.2.309 LogstashServerCustomListener
4.2.310 LogstashServerCustomOutput
4.2.311 LogstashServerElasticOutput
4.2.312 LogstashServerFilter
4.2.313 LogstashServerListener
4.2.314 LogstashServerLocalFileListener
4.2.315 LogstashServerLumberjackListener
4.2.316 LogstashServerOutput
4.2.317 LogstashServerRole
4.2.318 LogstashServerRSyslogFilter
4.2.319 LogstashServerRSyslogListener
4.2.320 LogstashServerStdOutput
4.2.321 LSFBaseJob
4.2.322 LSFBaseJobQueue

4.2.323 LSFBaseJobQueueStat
4.2.324 LSFCgroupsSettings
4.2.325 LSFClientRole
4.2.326 LSFJob
4.2.327 LSFJobQueue
4.2.328 LSFJobQueueStat
4.2.329 LSFServerRole
4.2.330 LustreAlert
4.2.331 LustreClientMount
4.2.332 LustreFileSystem
4.2.333 LustreFileSystemTarget
4.2.334 LustreLog
4.2.335 LustreOverview
4.2.336 LustreServer
4.2.337 LustreServerProfile
4.2.338 LustreSettings
4.2.339 LustreTargetMap
4.2.340 LustreUser
4.2.341 LustreVolume
4.2.342 LustreVolumeNode
4.2.343 MarathonRole
4.2.344 MasterNode
4.2.345 MasterRole
4.2.346 MemcachedRole
4.2.347 MemoryInfo
4.2.348 MesosCluster
4.2.349 MesosDNSRole
4.2.350 MesosMasterRole
4.2.351 MesosProxyRole
4.2.352 MesosResourceUsage
4.2.353 MesosSlaveRole
4.2.354 MICHostRole
4.2.355 MICInfo
4.2.356 MICNode
4.2.357 MICNodeCategory
4.2.358 MICOverlay
4.2.359 MICSettings
4.2.360 MonitoringAction
4.2.361 MonitoringActionRunData
4.2.362 MonitoringCacheSubSystemInfo
4.2.363 MonitoringCategoryListExecutionFilter
4.2.364 MonitoringCompareExpression
4.2.365 MonitoringConsolidator
4.2.366 MonitoringDataCacheSubSystemInfo
4.2.367 MonitoringDataProducer
4.2.368 MonitoringDataProducerAggregateNode
4.2.369 MonitoringDataProducerAlertLevel

4.2.370 MonitoringDataProducerCGroup
4.2.371 MonitoringDataProducerClusterTotal
4.2.372 MonitoringDataProducerCMDaemonState
4.2.373 MonitoringDataProducerDeviceState
4.2.374 MonitoringDataProducerEC2SpotPrices
4.2.375 MonitoringDataProducerEthernetSwitch
4.2.376 MonitoringDataProducerFuture
4.2.377 MonitoringDataProducerGalera
4.2.378 MonitoringDataProducerGenerator
4.2.379 MonitoringDataProducerGPU
4.2.380 MonitoringDataProducerInternal
4.2.381 MonitoringDataProducerJob
4.2.382 MonitoringDataProducerJobQueue
4.2.383 MonitoringDataProducerLua
4.2.384 MonitoringDataProducerMonitoringSystem
4.2.385 MonitoringDataProducerOpenStack
4.2.386 MonitoringDataProducerOpenStackHealth
4.2.387 MonitoringDataProducerPerpetual
4.2.388 MonitoringDataProducerPowerDistributionUnit
4.2.389 MonitoringDataProducerProcMemInfo
4.2.390 MonitoringDataProducerProcMount
4.2.391 MonitoringDataProducerProcNetDev
4.2.392 MonitoringDataProducerProcNetSnmp
4.2.393 MonitoringDataProducerProcPidStat
4.2.394 MonitoringDataProducerProcStat
4.2.395 MonitoringDataProducerProcVMStat
4.2.396 MonitoringDataProducerRackSensor
4.2.397 MonitoringDataProducerScript
4.2.398 MonitoringDataProducerSingleLineHealthCheckScript
4.2.399 MonitoringDataProducerSingleLineMetricScript
4.2.400 MonitoringDataProducerSingleLineScript
4.2.401 MonitoringDataProducerSmart
4.2.402 MonitoringDataProducerSysBlockStat
4.2.403 MonitoringDataProducerSysInfo
4.2.404 MonitoringDataProducerTest
4.2.405 MonitoringDataProducerTrustedTool
4.2.406 MonitoringDataProducerUserCount
4.2.407 MonitoringDeviceStateSubSystemInfo
4.2.408 MonitoringDrainAction
4.2.409 MonitoringEmailAction
4.2.410 MonitoringEventAction
4.2.411 MonitoringExecutionFilter
4.2.412 MonitoringExecutionMultiplexer
4.2.413 MonitoringExpression
4.2.414 MonitoringGroupedExpression
4.2.415 MonitoringHealthOverview
4.2.416 MonitoringImageUpdateAction

4.2.417 MonitoringJobMetricSettings
4.2.418 MonitoringLuaExecutionFilter
4.2.419 MonitoringLuaExecutionMultiplexer
4.2.420 MonitoringMeasurable
4.2.421 MonitoringMeasurableEnum
4.2.422 MonitoringMeasurableHealthCheck
4.2.423 MonitoringMeasurableMetric
4.2.424 MonitoringNodeListExecutionFilter
4.2.425 MonitoringOverlayListExecutionFilter
4.2.426 MonitoringPlotterSubSystemInfo
4.2.427 MonitoringPowerAction
4.2.428 MonitoringPowerOffAction
4.2.429 MonitoringPowerOnAction
4.2.430 MonitoringPowerResetAction
4.2.431 MonitoringRebootAction
4.2.432 MonitoringReplicateConfiguration
4.2.433 MonitoringReplicateSource
4.2.434 MonitoringReplicateSubSystemInfo
4.2.435 MonitoringResourceExecutionFilter
4.2.436 MonitoringResourceExecutionMultiplexer
4.2.437 MonitoringRole
4.2.438 MonitoringScriptAction
4.2.439 MonitoringServiceAction
4.2.440 MonitoringServiceRestartAction
4.2.441 MonitoringServiceStartAction
4.2.442 MonitoringServiceStopAction
4.2.443 MonitoringServiceSubSystemInfo
4.2.444 MonitoringShutdownAction
4.2.445 MonitoringStorageSubSystemInfo
4.2.446 MonitoringSubSystemInfo
4.2.447 MonitoringTrigger
4.2.448 MonitoringTypeExecutionFilter
4.2.449 MonitoringTypeExecutionMultiplexer
4.2.450 MonitoringUndrainAction
4.2.451 MsgQueue
4.2.452 MyrinetSwitch
4.2.453 Network
4.2.454 NetworkAliasInterface
4.2.455 NetworkBmcInterface
4.2.456 NetworkBondInterface
4.2.457 NetworkBridgeInterface
4.2.458 NetworkInterface
4.2.459 NetworkNetMapInterface
4.2.460 NetworkPhysicalInterface
4.2.461 NetworkTunnelInterface
4.2.462 NetworkVLANInterface
4.2.463 NewNode

4.2.464 NginxRole
4.2.465 Node
4.2.466 NodeCategory
4.2.467 NodeGroup
4.2.468 OpenLavaCgroupsSettings
4.2.469 OpenLavaClientRole
4.2.470 OpenLavaJob
4.2.471 OpenLavaJobQueue
4.2.472 OpenLavaJobQueueStat
4.2.473 OpenLavaServerRole
4.2.474 OpenStack
4.2.475 OpenStackApiAgent
4.2.476 OpenStackApiDomain
4.2.477 OpenStackApiEndpoint
4.2.478 OpenStackApiEntity
4.2.479 OpenStackApiFlavor
4.2.480 OpenStackApiFloatingIP
4.2.481 OpenStackApiGroup
4.2.482 OpenStackApiHostAggregate
4.2.483 OpenStackApiHypervisor
4.2.484 OpenStackApiImage
4.2.485 OpenStackApiNetwork
4.2.486 OpenStackApiPort
4.2.487 OpenStackApiProject
4.2.488 OpenStackApiRole
4.2.489 OpenStackApiRoleAssignment
4.2.490 OpenStackApiRouter
4.2.491 OpenStackApiSecurityGroup
4.2.492 OpenStackApiServer
4.2.493 OpenStackApiService
4.2.494 OpenStackApiStack
4.2.495 OpenStackApiSubnet
4.2.496 OpenStackApiUser
4.2.497 OpenStackApiVolume
4.2.498 OpenStackApiVolumeSnapshot
4.2.499 OpenStackApiVolumeType
4.2.500 OpenStackAuthBackend
4.2.501 OpenStackAuthBackendHybrid
4.2.502 OpenStackAuthBackendLDAP
4.2.503 OpenStackAuthBackendLDAPGroupSettings
4.2.504 OpenStackAuthBackendLDAPProjectSettings
4.2.505 OpenStackAuthBackendLDAPRoleSettings
4.2.506 OpenStackAuthBackendLDAPUserSettings
4.2.507 OpenStackAuthBackendSQL
4.2.508 OpenStackBareMetalApiRole
4.2.509 OpenStackBareMetalConductorRole
4.2.510 OpenStackBareMetalDiscoverdDNSMasqRole

4.2.511 OpenStackBareMetalDiscoverdRole
4.2.512 OpenStackBlockStorage
4.2.513 OpenStackComputeApiEC2Role
4.2.514 OpenStackComputeApiMetadataRole
4.2.515 OpenStackComputeApiRole
4.2.516 OpenStackComputeConductorRole
4.2.517 OpenStackComputeRole
4.2.518 OpenStackComputeSchedulerRole
4.2.519 OpenStackComputeVNCProxyRole
4.2.520 OpenStackDashboardRole
4.2.521 OpenStackDataProcessingApiRole
4.2.522 OpenStackDBaaSRole
4.2.523 OpenStackDefaultUserRole
4.2.524 OpenStackIdentityApiRole
4.2.525 OpenStackImageApiRole
4.2.526 OpenStackImageBackend
4.2.527 OpenStackImageBackendCeph
4.2.528 OpenStackImageBackendFS
4.2.529 OpenStackImageRegistryRole
4.2.530 OpenStackMessageQueueServerRole
4.2.531 OpenStackNetworkApiRole
4.2.532 OpenStackNetworkDHCPAgentRole
4.2.533 OpenStackNetworkL3AgentRole
4.2.534 OpenStackNetworkMetadataAgentRole
4.2.535 OpenStackNetworkOVSAgentRole
4.2.536 OpenStackNovaImageBackend
4.2.537 OpenStackNovaImageBackendCeph
4.2.538 OpenStackNovaImageBackendCow
4.2.539 OpenStackObjectAccountRole
4.2.540 OpenStackObjectApiRole
4.2.541 OpenStackObjectContainerRole
4.2.542 OpenStackObjectStoreRole
4.2.543 OpenStackOrchestrationApiRole
4.2.544 OpenStackOrchestrationRole
4.2.545 OpenStackSettings
4.2.546 OpenStackSettingsAdvanced
4.2.547 OpenStackSettingsAuthentication
4.2.548 OpenStackSettingsCMDaemonInteractions
4.2.549 OpenStackSettingsCollection
4.2.550 OpenStackSettingsCompute
4.2.551 OpenStackSettingsCredentials
4.2.552 OpenStackSettingsDatabase
4.2.553 OpenStackSettingsLogging
4.2.554 OpenStackSettingsNetworking
4.2.555 OpenStackSettingsPorts
4.2.556 OpenStackSettingsQuota
4.2.557 OpenStackSettingsUserPortal

4.2.558 OpenStackSettingsUsers
4.2.559 OpenStackStorage
4.2.560 OpenStackTelemetryAgentCentralRole
4.2.561 OpenStackTelemetryAgentComputeRole
4.2.562 OpenStackTelemetryAgentIpmiRole
4.2.563 OpenStackTelemetryAgentNotificationRole
4.2.564 OpenStackTelemetryAlarmEvaluatorRole
4.2.565 OpenStackTelemetryAlarmNotifierRole
4.2.566 OpenStackTelemetryApiRole
4.2.567 OpenStackTelemetryCollectorRole
4.2.568 OpenStackUserRole
4.2.569 OpenStackVolumeApiRole
4.2.570 OpenStackVolumeBackend
4.2.571 OpenStackVolumeBackend3PAR
4.2.572 OpenStackVolumeBackendCeph
4.2.573 OpenStackVolumeBackendDellStorageCenter
4.2.574 OpenStackVolumeBackendGPFS
4.2.575 OpenStackVolumeBackendNetApp
4.2.576 OpenStackVolumeBackendNFS
4.2.577 OpenStackVolumeBackendSolidFire
4.2.578 OpenStackVolumeBackupBackend
4.2.579 OpenStackVolumeBackupBackendCeph
4.2.580 OpenStackVolumeBackupRole
4.2.581 OpenStackVolumeRole
4.2.582 OpenStackVolumeSchedulerRole
4.2.583 OpenvSwitchRole
4.2.584 OsapiPortIP
4.2.585 OsapiSecurityGroupRule
4.2.586 OsapiStackResource
4.2.587 OsapiSubnetAllocationPool
4.2.588 OSService
4.2.589 OSServiceArray
4.2.590 OSServiceConfig
4.2.591 ParentJob
4.2.592 Partition
4.2.593 PBSJob
4.2.594 PBSJobQueue
4.2.595 PBSJobQueueStat
4.2.596 PbsProCgroupsSettings
4.2.597 PbsProClientRole
4.2.598 PbsProJob
4.2.599 PbsProJobQueue
4.2.600 PbsProJobQueueStat
4.2.601 PbsProServerRole
4.2.602 PDUPort
4.2.603 PhysicalNode
4.2.604 PowerDistributionUnit

4.2.605 PowerOperation
4.2.606 PowerStatus
4.2.607 Process
4.2.608 Processor
4.2.609 Profile
4.2.610 ProgramRunnerInput
4.2.611 ProgramRunnerKill
4.2.612 ProgramRunnerOutput
4.2.613 ProgramRunnerStatus
4.2.614 ProvisioningNodeStatus
4.2.615 ProvisioningProcessorJob
4.2.616 ProvisioningRequestStatus
4.2.617 ProvisioningRole
4.2.618 ProvisioningStatus
4.2.619 Puppet
4.2.620 PuppetApplyOnNodeRequest
4.2.621 PuppetApplyResult
4.2.622 PuppetApplySession
4.2.623 PuppetClass
4.2.624 PuppetClassDeclaration
4.2.625 PuppetClassFactory
4.2.626 PuppetConfigurationEntry
4.2.627 PuppetForgeInstallation
4.2.628 PuppetForgePagination
4.2.629 PuppetForgeSearchRequest
4.2.630 PuppetForgeSearchResult
4.2.631 PuppetKeyValuePair
4.2.632 PuppetModule
4.2.633 PuppetModuleDependency
4.2.634 PuppetModuleRelease
4.2.635 PuppetOperatingSystemSupport
4.2.636 PuppetParameterFactory
4.2.637 PuppetRescanResult
4.2.638 PuppetResourceDeclaration
4.2.639 PuppetRole
4.2.640 PuppetRoleChange
4.2.641 Rack
4.2.642 RackPosition
4.2.643 RackSensor
4.2.644 RadosGatewayRole
4.2.645 RemoteNodeInstallerInteraction
4.2.646 RemoteSetupExecution
4.2.647 ResourcePool
4.2.648 ResourcePoolStatus
4.2.649 Role
4.2.650 S3BucketIntermediateStorage
4.2.651 ScaleDynamicNodesProvider

4.2.652 ScaleEngine
4.2.653 ScaleHpcEngine
4.2.654 ScaleHpcQueueTracker
4.2.655 ScaleMesosEngine
4.2.656 ScaleMesosLoadTracker
4.2.657 ScalePendingWorkload
4.2.658 ScaleResourceProvider
4.2.659 ScaleServerRole
4.2.660 ScaleStaticNodesProvider
4.2.661 ScaleTracker
4.2.662 Semaphore
4.2.663 Sensor
4.2.664 Session
4.2.665 SGEClientRole
4.2.666 SGEJob
4.2.667 SGEJobQueue
4.2.668 SGEJobQueueStat
4.2.669 SGEParallelEnvironment
4.2.670 SGEServerRole
4.2.671 SharedMemory
4.2.672 SlaveNode
4.2.673 SlurmCgroupsSettings
4.2.674 SlurmClientRole
4.2.675 SlurmJob
4.2.676 SlurmJobQueue
4.2.677 SlurmJobQueueStat
4.2.678 SlurmServerRole
4.2.679 SoftwareImage
4.2.680 SoftwareImageProxy
4.2.681 SoftwareImageRevisionInfo
4.2.682 StandaloneMonitoredEntity
4.2.683 StaticRoute
4.2.684 StorageNodePolicy
4.2.685 StorageRole
4.2.686 StringListObject
4.2.687 SubnetManagerRole
4.2.688 SubSystemInfo
4.2.689 Switch
4.2.690 SwitchPort
4.2.691 SysInfoCollector
4.2.692 Ticket
4.2.693 TorqueCgroupsSettings
4.2.694 TorqueClientRole
4.2.695 TorqueJob
4.2.696 TorqueJobQueue
4.2.697 TorqueJobQueueStat
4.2.698 TorqueServerRole

4.2.699 UCSAdaptorEthCompQueueProfile
4.2.700 UCSAdaptorEthGenProfile
4.2.701 UCSAdaptorEthInterruptProfile
4.2.702 UCSAdaptorEthOffloadProfile
4.2.703 UCSAdaptorEthRecvQueueProfile
4.2.704 UCSAdaptorEthUSNICProfile
4.2.705 UCSAdaptorEthWorkQueueProfile
4.2.706 UCSAdaptorExtEthIf
4.2.707 UCSAdaptorExtIpV6RssHashProfile
4.2.708 UCSAdaptorFcCdbWorkQueueProfile
4.2.709 UCSAdaptorFcErrorRecoveryProfile
4.2.710 UCSAdaptorFcGenProfile
4.2.711 UCSAdaptorFcInterruptProfile
4.2.712 UCSAdaptorFcPortFLogiProfile
4.2.713 UCSAdaptorFcPortPLogiProfile
4.2.714 UCSAdaptorFcPortProfile
4.2.715 UCSAdaptorFcRecvQueueProfile
4.2.716 UCSAdaptorFcWorkQueueProfile
4.2.717 UCSAdaptorHostEthIf
4.2.718 UCSAdaptorHostFcIf
4.2.719 UCSAdaptorIpV4RssHashProfile
4.2.720 UCSAdaptorIpV6RssHashProfile
4.2.721 UCSAdaptorPortProfiles
4.2.722 UCSAdaptorRssProfile
4.2.723 UCSBase
4.2.724 UCSBiosBootDev
4.2.725 UCSBiosBootDevGrp
4.2.726 UCSBiosSettings
4.2.727 UCSBiosVfAdjacentCacheLinePrefetch
4.2.728 UCSBiosVfAltitude
4.2.729 UCSBiosVfASPMSupport
4.2.730 UCSBiosVfConsoleRedirection
4.2.731 UCSBiosVfCoreMultiProcessing
4.2.732 UCSBiosVfCPUEnergyPerformance
4.2.733 UCSBiosVfCPUFrequencyFloor
4.2.734 UCSBiosVfCPUPerformance
4.2.735 UCSBiosVfCPUPowerManagement
4.2.736 UCSBiosVfDCUPrefetch
4.2.737 UCSBiosVfDemandScrub
4.2.738 UCSBiosVfDirectCacheAccess
4.2.739 UCSBiosVfDRAMClockThrottling
4.2.740 UCSBiosVfDramRefreshRate
4.2.741 UCSBiosVfEnhancedIntelSpeedStepTech
4.2.742 UCSBiosVfExecuteDisableBit
4.2.743 UCSBiosVfFRB2Enable
4.2.744 UCSBiosVfHardwarePrefetch
4.2.745 UCSBiosVfIntelHyperThreadingTech

4.2.746 UCSBiosVfIntelTurboBoostTech
4.2.747 UCSBiosVfIntelVirtualizationTechnology
4.2.748 UCSBiosVfIntelVTForDirectedIO
4.2.749 UCSBiosVfLegacyUSBSupport
4.2.750 UCSBiosVfLOMPortOptionROM
4.2.751 UCSBiosVfLvDIMMSupport
4.2.752 UCSBiosVfMemoryInterleave
4.2.753 UCSBiosVfMemoryMappedIOAbove4GB
4.2.754 UCSBiosVfNUMAOptimized
4.2.755 UCSBiosVfOnboardStorage
4.2.756 UCSBiosVfOnboardStorageSWStack
4.2.757 UCSBiosVfOSBootWatchdogTimer
4.2.758 UCSBiosVfOSBootWatchdogTimerPolicy
4.2.759 UCSBiosVfOSBootWatchdogTimerTimeout
4.2.760 UCSBiosVfPatrolScrub
4.2.761 UCSBiosVfPCIOptionROMs
4.2.762 UCSBiosVfPCISlotOptionROMEnable
4.2.763 UCSBiosVfProcessorC1E
4.2.764 UCSBiosVfProcessorC6Report
4.2.765 UCSBiosVfPStateCoordType
4.2.766 UCSBiosVfQPIConfig
4.2.767 UCSBiosVfSelectMemoryRASConfiguration
4.2.768 UCSBiosVfTPMSupport
4.2.769 UCSBiosVfUCSMBootOrderRuleControl
4.2.770 UCSBiosVfUSBEmulation
4.2.771 UCSBiosVfUSBPortsConfig
4.2.772 UCSBiosVfVgaPriority
4.2.773 UCSCommNtpProvider
4.2.774 UCSCommSyslog
4.2.775 UCSCommSyslogClient
4.2.776 UCSEquipmentIndicatorLed
4.2.777 UCSEquipmentLocatorLed
4.2.778 UCSFaultInst
4.2.779 UCSFirmwareRunning
4.2.780 UCSInfo
4.2.781 UCSLogs
4.2.782 UCSLsbootDef
4.2.783 UCSLsbootEfi
4.2.784 UCSLsbootLan
4.2.785 UCSLsbootStorage
4.2.786 UCSLsbootVirtualMedia
4.2.787 UCSStatus
4.2.788 UGECgroupsSettings
4.2.789 UGEClientRole
4.2.790 UGEJob
4.2.791 UGEJobQueue
4.2.792 UGEJobQueueStat

4.2.793 UGEParallelEnvironment
4.2.794 UGEServerRole
4.2.795 User
4.2.796 Validation
4.2.797 VersionInfo
4.2.798 VirtualNode
4.2.799 VirtualNodeSettings
4.2.800 VirtualSMPNode
4.2.801 VScaleMPSettings
4.2.802 VsmpSettings
4.2.803 WillChange
4.2.804 WlmCgroupsSettings
4.2.805 XeonPhiSettings
4.2.806 ZooKeeperCluster
4.2.807 ZooKeeperHostRole

4.3 JSON Examples

complete.sh

```
#!/bin/bash

URL=https://localhost:8081/json/
user=root
pass=secretrootpassword

echo "========== login ==========="
curl -c curl.cookiest.txt -i -k -X POST -d \
'{"service":"login", "username":"root", "password":"'$pass'"}' $URL; echo

echo "========= master =========="
curl --cookie curl.cookiest.txt -i -k -X POST -d \
'{"service":"cmdevice","call":"getNode", "arg":"master"}' $URL; echo

echo "========== logout ==========="
curl --cookie curl.cookiest.txt -i -k -X POST -d \
'{"service":"logout"}' $URL; echo

echo "========== denied ==========="
curl --cookie curl.cookiest.txt -i -k -X POST -d \
'{"service":"cmdevice","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

URL=https://localhost:8081/json/

if [ -z "$1" ]; then
```

```
  read -p "pass: " -s pass
else
  pass=$1
fi

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

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

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

curl --cookie curl.cookiest.txt -i -k -X POST -d \
'{"service":"cmmain","call":"getSubjectName"}' $URL
```

devices.sh

```
#!/bin/bash
URL=https://localhost:8081/json/

if [ "$1" == "gzip" ]; then
   wget --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key \
   --header='Accept-Encoding: gzip' \
   --no-check-certificate --server-response -qO- $URL \
   --post-data='{"service":"cmdevice","call":"getDevices"}'
else
   wget --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key --no-check-certificate \
   --server-response -qO- $URL --post-data='{"service":"cmdevice","call":"getDevices"}'
fi
```

Tip: run as `./devices.sh | python -mjson.tool`.

loadone.sh

```
#!/bin/bash
URL=https://localhost:8081/json/

# 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'| grep -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 --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key --no-check-certificat
--server-response -qO- $URL --post-data='{"service":"cmdevice","call":"getDevice","arg":"'$node'"}'`
nkey=$(jsonval)
if [ -z $nkey ]; then
  echo $json
```

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

json=`wget --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key \
--no-check-certificate --server-response -qO- $URL \
--post-data='{"service":"cmmon","call":"getMonitoringMeasurable","name":"LoadOne"}'`
mkey=$(jsonval)
echo "loadone.uniqueKey = $mkey"

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

cat <<EOF > /tmp/plot.json
{ "service" : "cmmon",
  "call" : "plot",
  "request" : { "entities" : [$nkey],
                "measurables" : [$mkey],
                "intervals" : 25,
                "rangeStart" : $((day*1000)),
                "rangeEnd" : $((now*1000))
              }
}
EOF
wget --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key \
--no-check-certificate -qO- https://master:8081/json --post-file=/tmp/plot.json | \
python -mjson.tool
```

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
URL=https://localhost:8081/json/
```

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

logout.sh

```
#!/bin/bash
URL=https://localhost:8081/json/
wget --load-cookies cookie.txt --no-check-certificate --server-response -qO- $URL \
--post-data='{"service":"logout"}'
rm cookie.txt
echo
```

node001.sh

```
#!/bin/bash
URL=https://localhost:8081/json/

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

wget --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key \
--no-check-certificate --server-response -qO- $URL \
--post-data='{"service":"cmdevice","call":"getDevice","arg":"'$node'"}' | python -mjson.tool
```

basic_information.sh

```
#!/bin/bash
URL=https://localhost:8081/json/
wget --certificate=$HOME/.cm/admin.pem --private-key=$HOME/.cm/admin.key \
--no-check-certificate --server-response -qO- $URL \
--post-data='{"service":"cmpart","call":"getBasicEntityInformation"}'
```