

**Documentation for the study**

**”An Experimental and Comparative Benchmark Study Examining Resource Utilization in Managed Hadoop Context”**

**Setting Up Experimental Managed Hadoop Environments and  
Running HiBench**

**U. E. Özdil, S. Ayvaz**

## CONTENTS

<b>FIGURES .....</b>	<b>iv</b>
1. Creating Dataproc Cluster on GCP.....	1
2. Creating HDInsight Cluster on Azure .....	13
3. Creating e-MapReduce Cluster on Alibaba Cloud .....	26
4. Running HiBench on GCP Dataproc .....	39
5. Running HiBench on Azure HDInsight .....	46
6. Running HiBench on Alibaba Cloud e-MapReduce .....	49

## FIGURES

Figure 1.1 : GCP Mainpage .....	1
Figure 1.2 : GCP - New Project .....	1
Figure 1.3 : GCP - Create Project .....	2
Figure 1.4 : GCP - Enabling Dataproc API .....	2
Figure 1.5 : GCP- Create Firewall (Step 1) .....	3
Figure 1.6 : GCP- Create Firewall (Step 2) .....	3
Figure 1.7 : GCP- Create Firewall (Step 3) .....	4
Figure 1.8 : GCP- Create Firewall (Step 4) .....	5
Figure 1.9 : GCP Dataproc - Create Cluster (Step 1) .....	5
Figure 1.10 : GCP Dataproc - Create Cluster (Step 2) .....	5
Figure 1.11 : GCP Dataproc - Create Cluster (Step 3) .....	6
Figure 1.12 : GCP Dataproc - Create Cluster (Step 4) .....	7
Figure 1.13 : GCP Dataproc - Create Cluster (Step 5) .....	7
Figure 1.14 : GCP Dataproc - Create Cluster (Step 6) .....	8
Figure 1.15 : GCP Dataproc - Create Cluster (Step 7) .....	8
Figure 1.16 : GCP Dataproc - Create Cluster (Step 8) .....	9
Figure 1.17 : GCP Dataproc - Create Cluster (Step 9) .....	9
Figure 1.18 : GCP Dataproc - Create Cluster (Step 10) .....	10
Figure 1.19 : GCP Dataproc - Create Cluster (Step 11) .....	10
Figure 1.20 : GCP Dataproc - Create Cluster (Step 12) .....	11
Figure 1.21 : GCP Dataproc - Namenode Manager .....	11
Figure 1.22 : GCP Dataproc - Resource Manager .....	12
Figure 2.1 : HDInsight setup - Create a resource group .....	13
Figure 2.2 : HDInsight setup - Register subscription .....	14
Figure 2.3 : HDInsight setup - Navigate to HDInsight .....	14
Figure 2.4 : HDInsight setup - Create Cluster .....	15
Figure 2.5 : HDInsight setup - Create Cluster .....	16
Figure 2.6 : HDInsight setup - Storage .....	17
Figure 2.7 : HDInsight setup - Networking .....	18
Figure 2.8 : HDInsight setup - Configuration .....	19
Figure 2.9 : HDInsight setup - Review .....	20

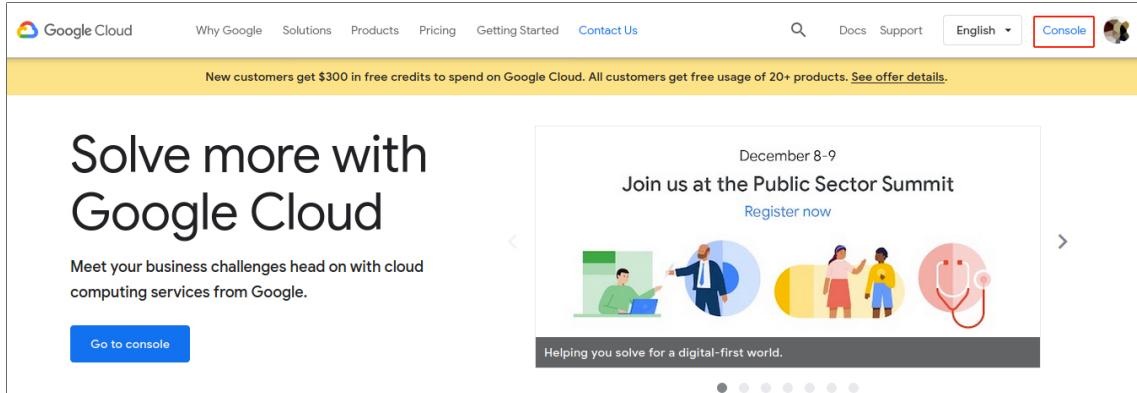
Figure 2.10 : HDInsight setup - In progress .....	21
Figure 2.11 : HDInsight setup - Complete.....	21
Figure 2.12 : HDInsight setup - Overview .....	22
Figure 2.13 : HDInsight - Ambari Dashboard.....	22
Figure 2.14 : HDInsight - Cluster login .....	23
Figure 2.15 : HDInsight - Connecting to master node .....	24
Figure 2.16 : HDInsight - Listing worker nodes .....	24
Figure 2.17 : HDInsight - Ready to benchmark .....	25
Figure 3.1 : e-MapReduce - Create access key .....	26
Figure 3.2 : e-MapReduce - Activate OSS (Step 1) .....	27
Figure 3.3 : e-MapReduce - Activate OSS (Step 2) .....	27
Figure 3.4 : e-MapReduce - Activate OSS (Step 3) .....	28
Figure 3.5 : e-MapReduce - Create OSS bucket .....	28
Figure 3.6 : e-MapReduce - Enabling access logging on OSS .....	29
Figure 3.7 : e-MapReduce - Select Region .....	29
Figure 3.8 : e-MapReduce - Create cluster (Step 1) .....	29
Figure 3.9 : e-MapReduce - Create cluster (Step 2) .....	30
Figure 3.10 : e-MapReduce - Create cluster (Step 3) .....	30
Figure 3.11 : e-MapReduce - Create cluster (Step 4) .....	31
Figure 3.12 : e-MapReduce - Create cluster (Step 5) .....	32
Figure 3.13 : e-MapReduce - Create cluster (Step 6) .....	33
Figure 3.14 : e-MapReduce - Create cluster (Step 7) .....	34
Figure 3.15 : e-MapReduce - Create cluster (Step 8) .....	34
Figure 3.16 : e-MapReduce - Create cluster (Step 9) .....	35
Figure 3.17 : e-MapReduce - Create cluster (Step 10).....	35
Figure 3.18 : e-MapReduce - Connecting to the cluster .....	37
Figure 3.19 : e-MapReduce - Connecting to worker nodes.....	38
Figure 3.20 : e-MapReduce - Handling broken pipe error.....	38
Figure 3.21 : e-MapReduce - Ready to benchmark .....	38
Figure 4.1 : HiBench Dataproc - Uploading datacollector.sh .....	40
Figure 4.2 : HiBench Dataproc - Verify Maven installation.....	40
Figure 4.3 : HiBench Dataproc - Success compiled .....	42
Figure 4.4 : HiBench Dataproc - Hadoop configurations.....	43

Figure 4.5 : HiBench Dataproc - HiBench configurations .....	44
Figure 4.6 : HiBench Dataproc - Profile from an ongoing benchmark execution	45
Figure 5.1 : Azure HDInsight - HiBench execution process configurations.....	48
Figure 6.1 : Alibaba Cloud e-MapReduce - Maven version.....	50
Figure 6.2 : Alibaba Cloud e-MapReduce - HiBench success .....	51
Figure 6.3 : Alibaba Cloud e-MapReduce - Hadoop configurations .....	52
Figure 6.4 : Alibaba Cloud e-MapReduce - HiBench configurations .....	53
Figure 6.5 : Alibaba Cloud e-MapReduce - HiBench running .....	53

## 1. CREATING DATAPROC CLUSTER ON GCP

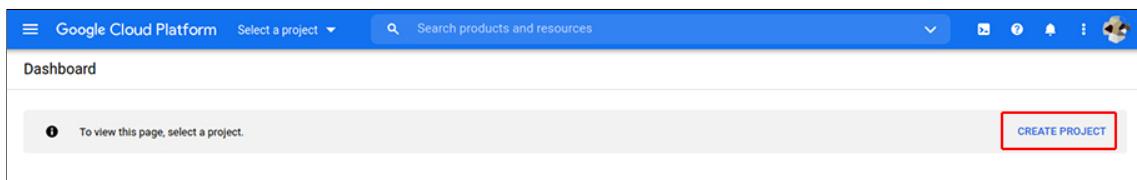
To operate on GCP, a Google account is required. Having a Google account set, go to:  
<https://cloud.google.com>

Figure 1.1: Click on Sign In, giving requested account credentials will redirect to GCP main page as below. Click on the Console link on the top right of the page.



**Figure 1.1: GCP Mainpage**

Figure 1.2: In the dashboard the user is requested to create a project first. A project is needed to go with GCP services. Click on Create Project.



**Figure 1.2: GCP - New Project**

Figure 1.3: Enter project details. At this stage a billing account will be asked to be created as well.

Figure 1.4: Clicking on "Dataproc" in the GCP menu below the group "Big Data" will redirect to Dataproc since the API for Dataproc needs to be enabled to run. Clicking on "Enable" will set Dataproc ready to operate.

Figure 1.5: Before starting Dataproc cluster installation, we create a firewall rule to allow accessing Hadoop web UI over internet. From the VPC network dashboard within GCP, click on Create Firewall Rule.

Figure 1.6: Entering following informations:

Name: hadoop-access

You have 4 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*  [?](#)

Project ID: my-hibench-cluster. It cannot be changed later. [EDIT](#)

Billing account \*  [▼](#)

Any charges for this project will be billed to the account you select here.

Location \*  [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Figure 1.3: GCP - Create Project

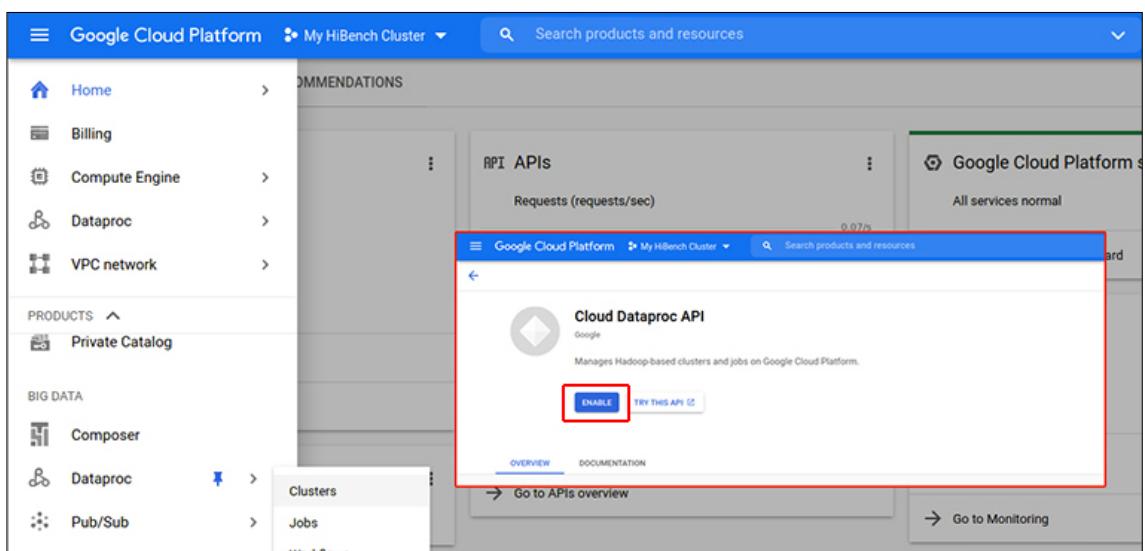


Figure 1.4: GCP - Enabling Dataproc API

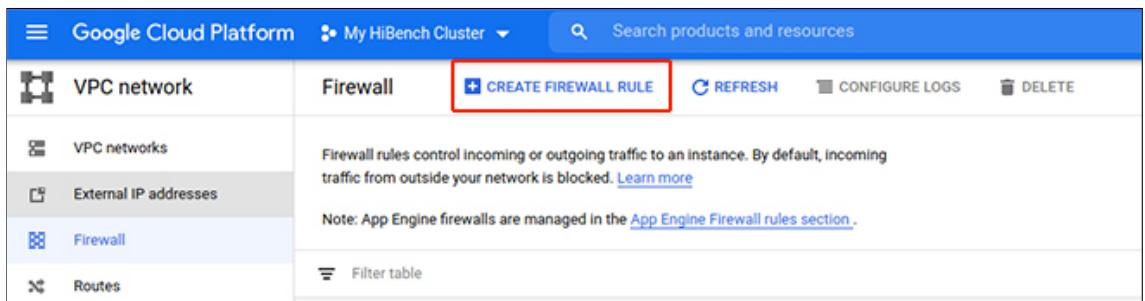


Figure 1.5: GCP- Create Firewall (Step 1)

A screenshot of the 'Create a firewall rule' dialog box. The left sidebar shows 'VPC network' selected. The main form has the following fields:

- Name \***: allow-hadoop
- Description**: (empty)
- Logs**: Off
- Network \***: default
- Priority \***: 1000
- Direction of traffic**: Ingress
- Action on match**: Allow

Figure 1.6: GCP- Create Firewall (Step 2)

Figure 1.7: Direction of traffic: ingress (for incoming traffic)

Specify a user friendly tag name (hadoopaccess in this case) for determining the access of the dataproc cluster at later stage. To limit the cluster acces only with our local machine, Source IP ranges is given with the IP value of current local machine including subnet mask ”/32” at the end. We specify TCP protocol with port number 9870 and 8088, gateways for NameNode’s and YARN resource manager’s WebUI, respectively.

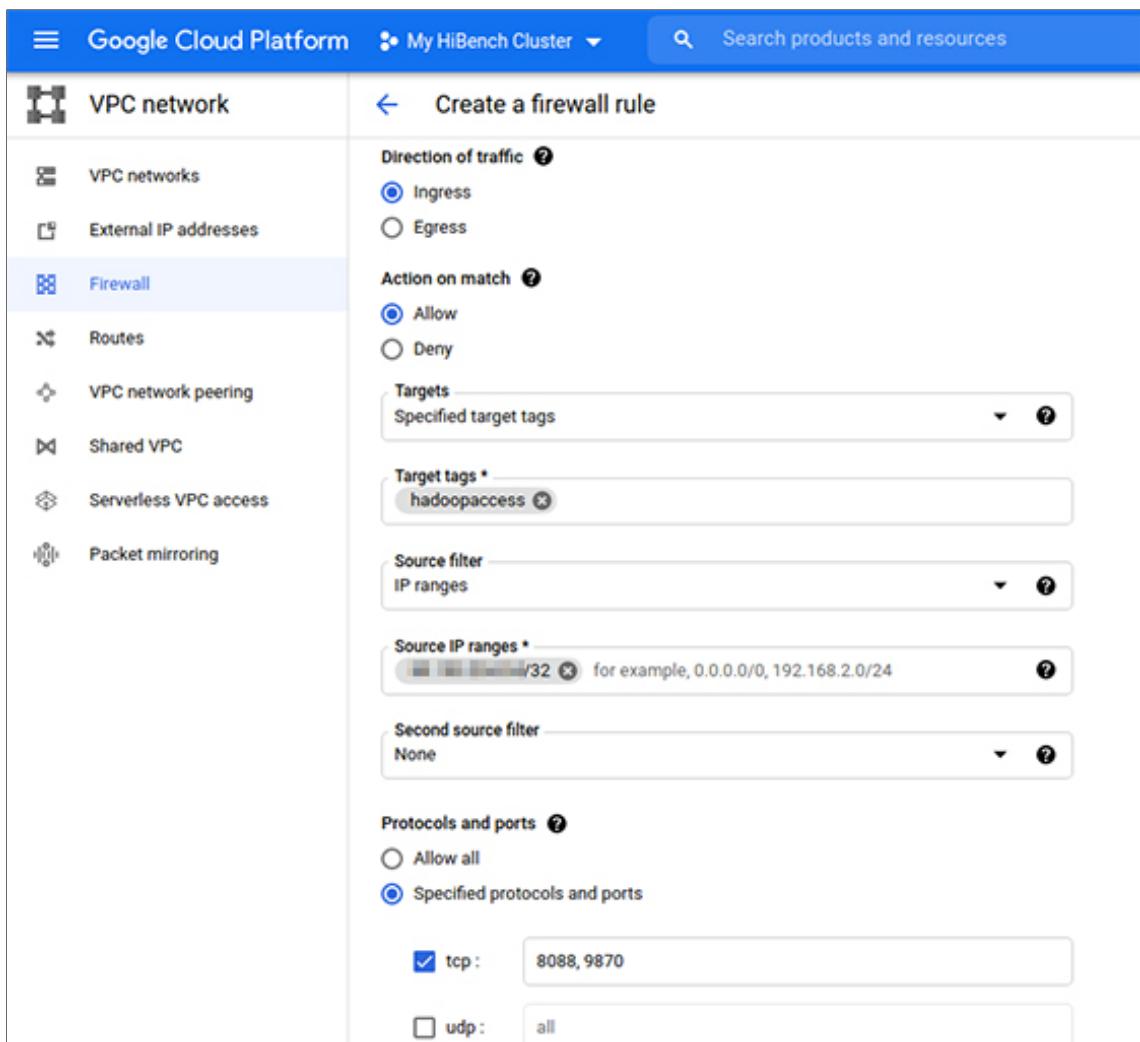


Figure 1.7: GCP- Create Firewall (Step 3)

Figure 1.8: Clicking on Create button redirects to firewall listings page after successful creation.

Figure 1.9: To start with the managed Hadoop cluster, click on "Create Cluster"

Figure 1.10:

Selected configurations:

Setup Cluster Pane

Name: a proper name for the cluster

The screenshot shows the Google Cloud Platform VPC network Firewall interface. The left sidebar lists options like VPC networks, External IP addresses, Firewall, Routes, VPC network peering, and Shared VPC. The main area displays a table for Firewall rules. One rule is listed:

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network	Logs	Hit
allow-hadoop	Ingress	hadoopaccess	IP ranges: 10.128.0.0/14	tcp:8088,50070	Allow	1000	default	Off	

**Figure 1.8: GCP- Create Firewall (Step 4)**

The screenshot shows the Google Cloud Platform Dataproc Clusters interface. The left sidebar lists Clusters, Jobs, Workflows, Autoscaling policies, Component exchange, and Notebooks. The main area shows a summary for a cluster named "Cloud Dataproc" with a "CREATE CLUSTER" button highlighted by a red box.

**Figure 1.9: GCP Dataproc - Create Cluster (Step 1)**

Location: europe-west3, Frankfurt

Cluster type: Select "Standard (1 master, N workers)"

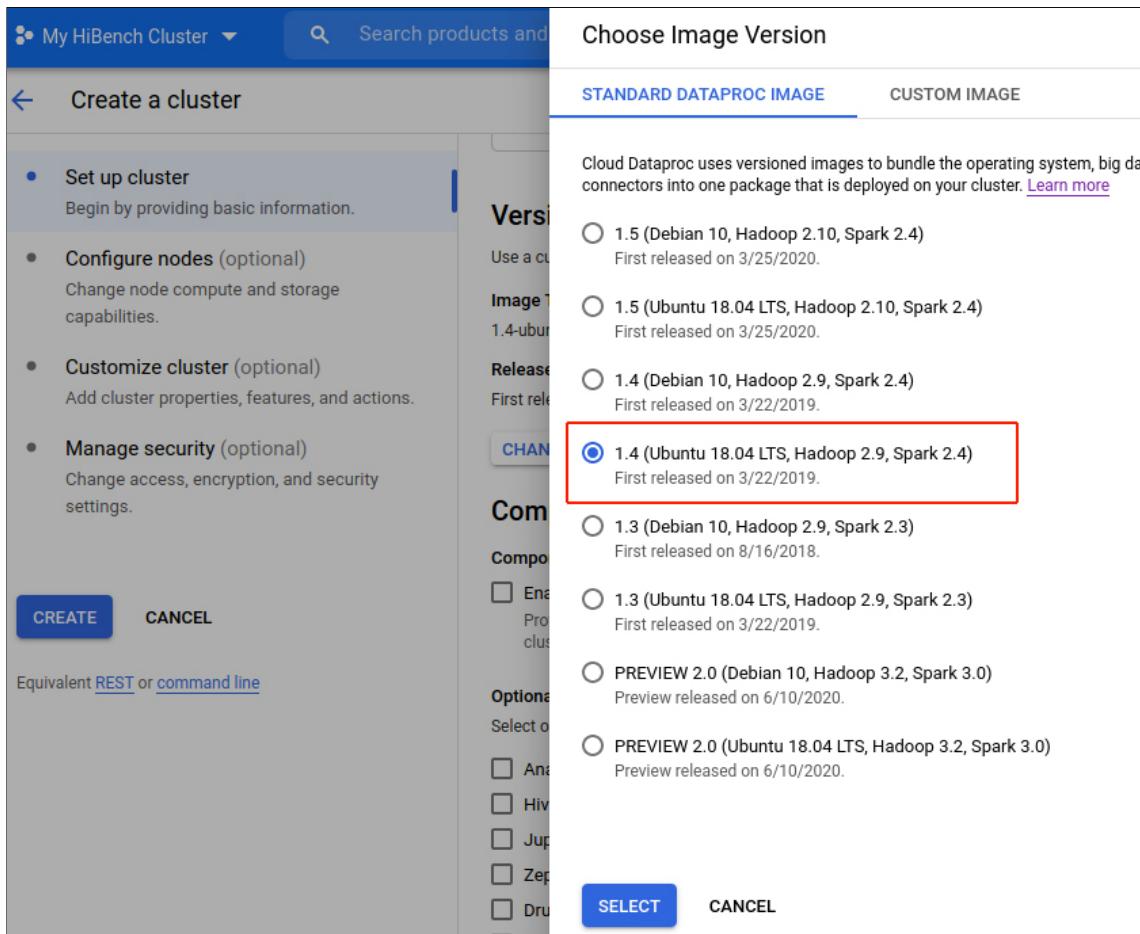
Autoscaling policy is left blank since we are not dealing autoscaling.

The screenshot shows the "Create a cluster" form for Google Cloud Platform Dataproc. The left sidebar lists Clusters, Jobs, Workflows, Autoscaling policies, Component exchange, and Notebooks. The main form has the following fields:

- Set up cluster**: Step 1 of 5. Sub-steps: Configure nodes (optional), Customize cluster (optional), Manage security (optional).
- Name**: Cluster Name: hadoop-cluster
- Location**: Region: europe-west3, Zone: europe-west3-a
- Cluster type**: Standard (1 master, N workers) is selected. Other options: Single Node (1 master, 0 workers) and High Availability (3 masters, N workers).
- Autoscaling**: Automates cluster resource management based on an autoscaling policy. Policy: None.

**Figure 1.10: GCP Dataproc - Create Cluster (Step 2)**

Figure 1.11: The pre-installed and pre-configured Dataproc image 1.4 fits our rules since it fits our rule satisfying HiBench's prerequisite of Hadoop 2.X. The OS is ubuntu 18.04.



**Figure 1.11: GCP Dataproc - Create Cluster (Step 3)**

Figure 1.12: Regarding our aim to benchmarks managed systems as they come out of the box, we don't want to use any optional components, so leaving the Components boxes blank.

Figure 1.13: For the master node we select General Purpose *e2-highmem-8* configuration allocating 64GB memory with 8 cores for the namenode. Master node is lasted mostly in memory operations so we allocate large space. Leave the primary disk size in its default value of 500 GB.

Figure 1.14: For worker nodes number we specify 3, for machine type selecting *e2-highmem-4* configuration provides 4 CPUs and 32 GB memory per worker node which totals in 12 processors and 96 GB memory for the cluster.

Figure 1.15: Below the configuration settings page worker nodes' available systems resources to YARN is listed. The local fraction available to YARN is 0.8 which makes 76.8 of 96 GB available to Hadoop.

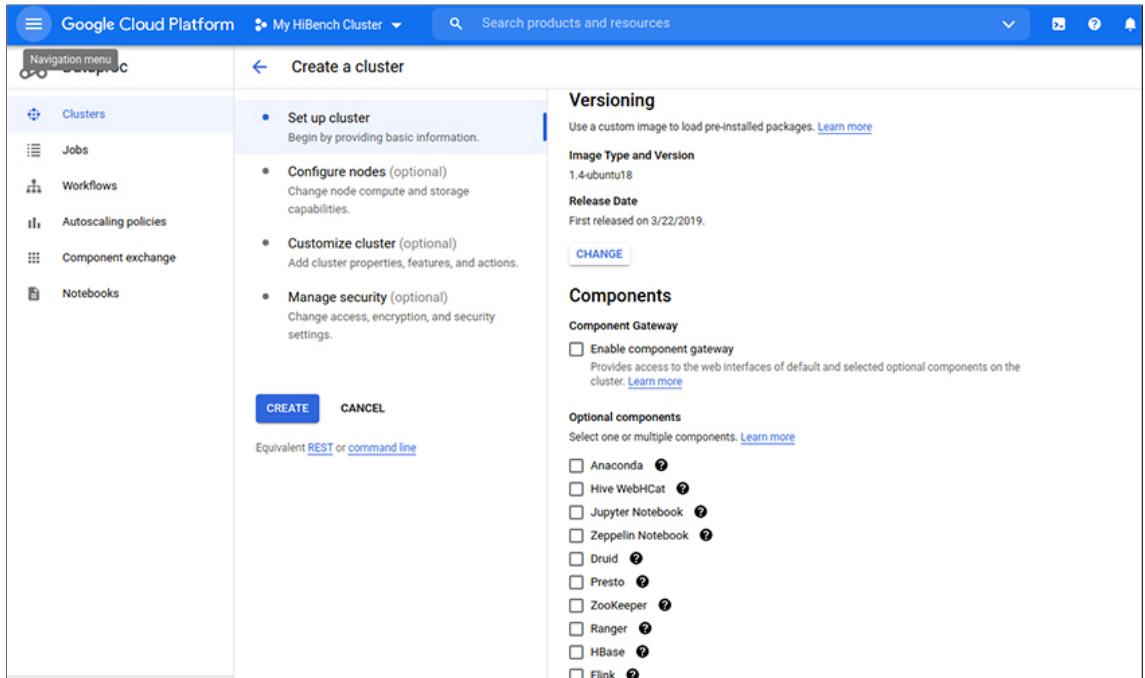


Figure 1.12: GCP Dataproc - Create Cluster (Step 4)

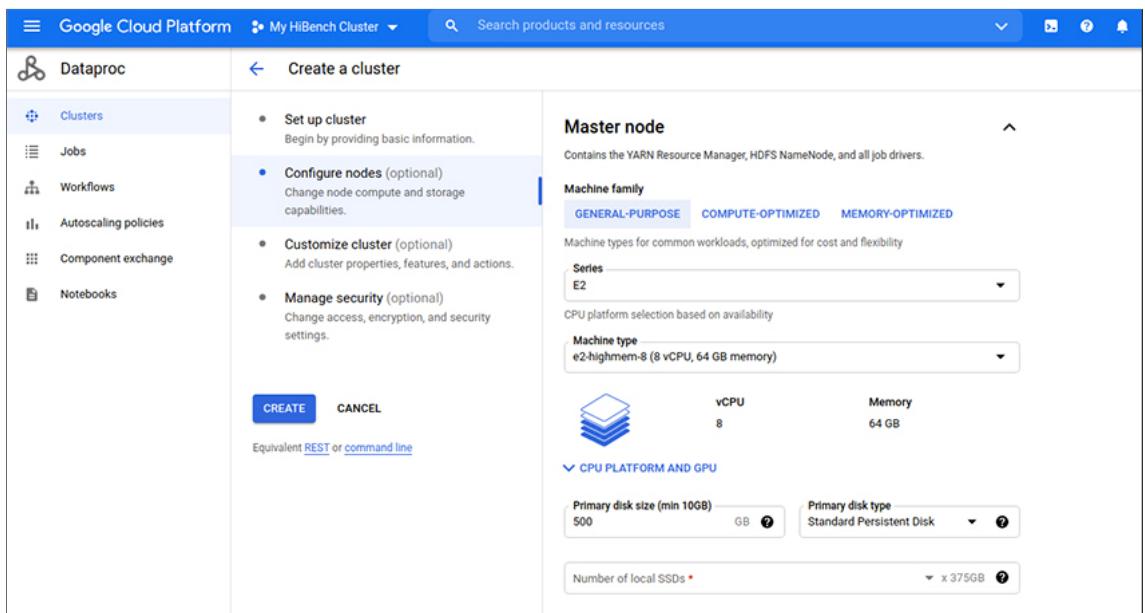
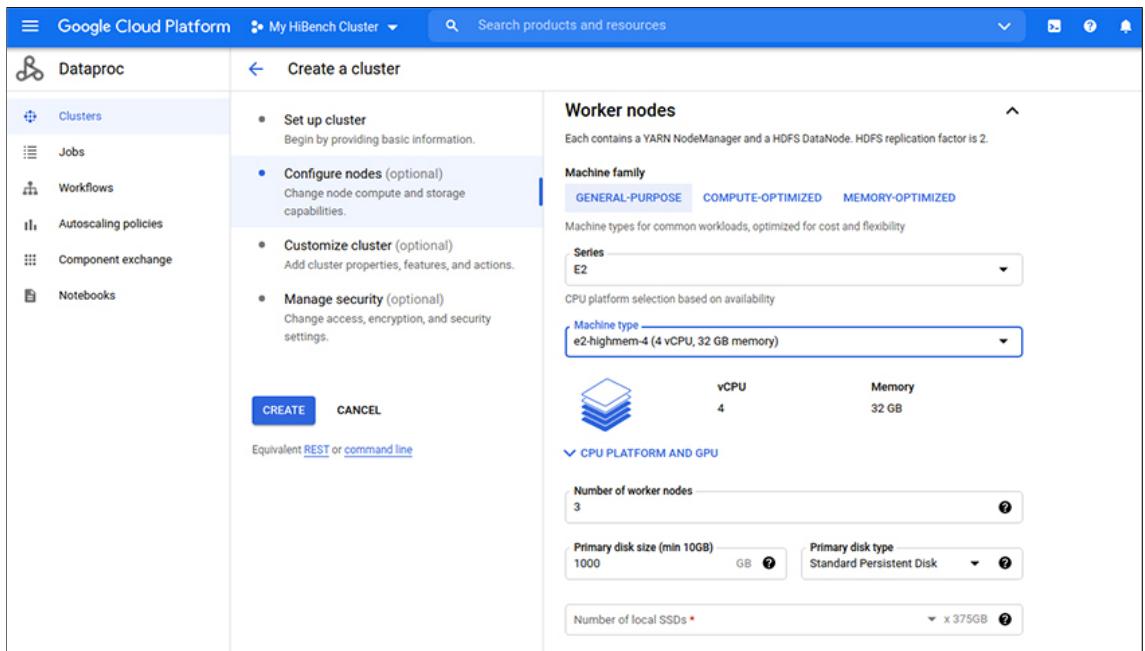


Figure 1.13: GCP Dataproc - Create Cluster (Step 5)



**Figure 1.14: GCP Dataproc - Create Cluster (Step 6)**

**Secondary worker nodes**

Each contains a YARN NodeManager. HDFS does not run on secondary worker nodes. Secondary worker VMs are preemptible by default. A preemptible VM costs less, but lasts only 24 hours, and can be terminated at any time due to system demands. [Learn more](#)

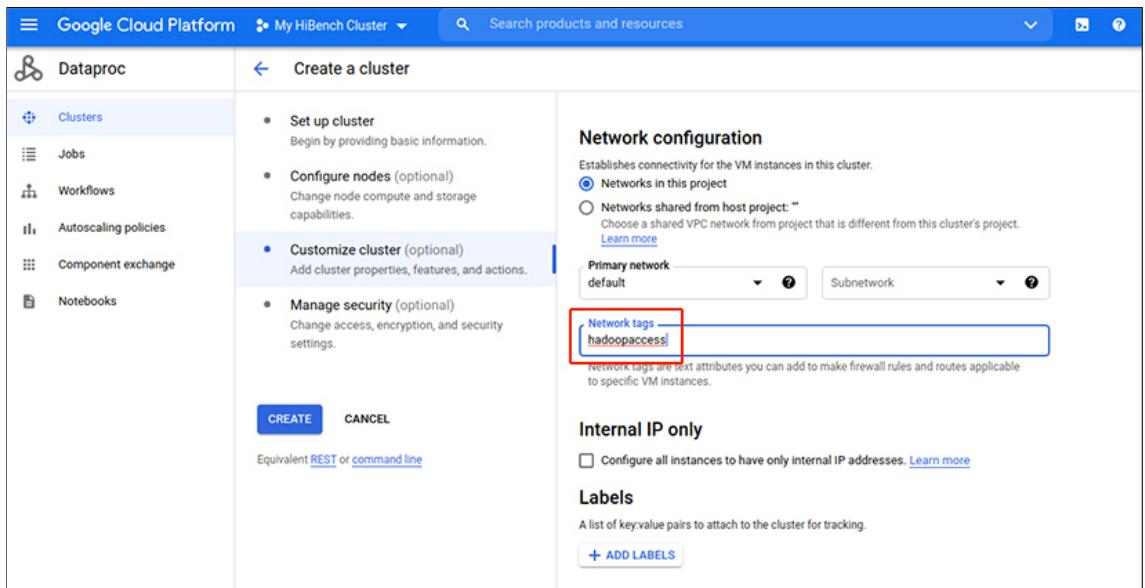
---

**Total YARN usage**

<b>YARN cores</b> <small>?</small>	<b>YARN memory</b> <small>?</small>
12	76.8 GB

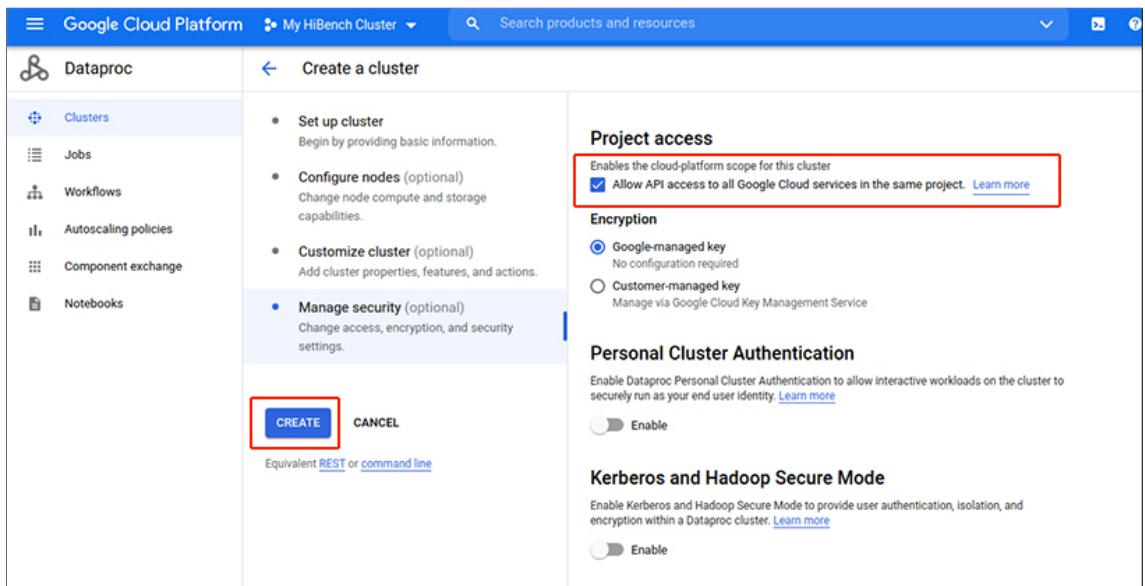
**Figure 1.15: GCP Dataproc - Create Cluster (Step 7)**

Figure 1.16: For the network configuration we provide the tag we created during firewall creation for Hadoop.



**Figure 1.16: GCP Dataproc - Create Cluster (Step 8)**

Figure 1.17: As final step we check Allow API access, and after reviewing our settings, we click on Create.



**Figure 1.17: GCP Dataproc - Create Cluster (Step 9)**

Figure 1.18 and Figure 1.19: The creation process can be followed from Dataproc and VM dashboards.

Figure 1.20: Once the installation is completed, by navigating to Compute Engine & VM Instances

The screenshot shows the Google Cloud Platform interface for the Dataproc service. The main header includes the 'Google Cloud Platform' logo, a dropdown for 'My HiBench Cluster', a search bar labeled 'Search products and resources', and various navigation icons. On the left, there's a sidebar with sections for Clusters, Jobs, Workflows, Autoscaling policies, Component exchange, and Notebooks. The main content area is titled 'Clusters' and contains a table with one row. The table columns are 'Name', 'Region', 'Zone', 'Total worker nodes', 'Scheduled deletion', 'Cloud Storage staging bucket', and 'Created'. The single entry is 'hadoop-cluster' located in 'europe-west3' with '3' total worker nodes, created on 'Dec 12, 2020, 1:10:43 PM'. A search bar at the top of the content area says 'Search clusters, press Enter'.

Clusters		<a href="#">CREATE CLUSTER</a>	<a href="#">REFRESH</a>	<a href="#">DELETE</a>	REGIONS	S
<input type="text"/> Search clusters, press Enter						
Name	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created
hadoop-cluster	europe-west3	europe-west3-a	3	Off	dataproc-staging-europe-west3-976470371707-ujyal1j3	Dec 12, 2020, 1:10:43 PM

**Figure 1.18: GCP Dataproc - Create Cluster (Step 10)**

The screenshot shows the Google Cloud Platform interface for the Compute Engine service. The main header includes the 'Google Cloud Platform' logo, a dropdown for 'My HiBench Cluster', a search bar labeled 'Search products and resources', and various navigation icons. On the left, there's a sidebar with sections for Virtual machines, Instance templates, Sole-tenant nodes, Machine images, and TPUs. The main content area is titled 'VM instances' and contains a table with four rows. The table columns are 'Name', 'Zone', 'Recommendation', 'In use by', 'Internal IP', 'External IP', and 'Connect'. The instances listed are 'hadoop-cluster-m' (Zone: europe-west3-a, Internal IP: 10.156.0.4, External IP: 35.246.229.170), 'hadoop-cluster-w-0' (Zone: europe-west3-a, Internal IP: 10.156.0.2, External IP: 35.198.131.91), 'hadoop-cluster-w-1' (Zone: europe-west3-a, Internal IP: 10.156.0.5, External IP: 34.107.4.110), and 'hadoop-cluster-w-2' (Zone: europe-west3-a, Internal IP: 10.156.0.3, External IP: 35.242.246.14). Each instance has an 'SSH' button next to its external IP. A search bar at the top of the content area says 'Filter VM instances'.

VM instances		<a href="#">CREATE INSTANCE</a>						<a href="#">MANAGE ACCESS</a>	SHOW INFO PANEL
<input type="text"/> Filter VM instances									
Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect			
hadoop-cluster-m	europe-west3-a			10.156.0.4 (nic0)	35.246.229.170	SSH	⋮		
hadoop-cluster-w-0	europe-west3-a			10.156.0.2 (nic0)	35.198.131.91	SSH	⋮		
hadoop-cluster-w-1	europe-west3-a			10.156.0.5 (nic0)	34.107.4.110	SSH	⋮		
hadoop-cluster-w-2	europe-west3-a			10.156.0.3 (nic0)	35.242.246.14	SSH	⋮		

**Figure 1.19: GCP Dataproc - Create Cluster (Step 11)**

page we can acces master and worker nodes' CLIs by clicking on relevant SSH buttons. Our Dataproc cluster is ready for benchmarking operations.

Figure 1.21 and Figure 1.22: Using the external IP of the master node and specified ports 8088 and 9870 in the firewall settings, Hadoop Web UI portals shall be available.

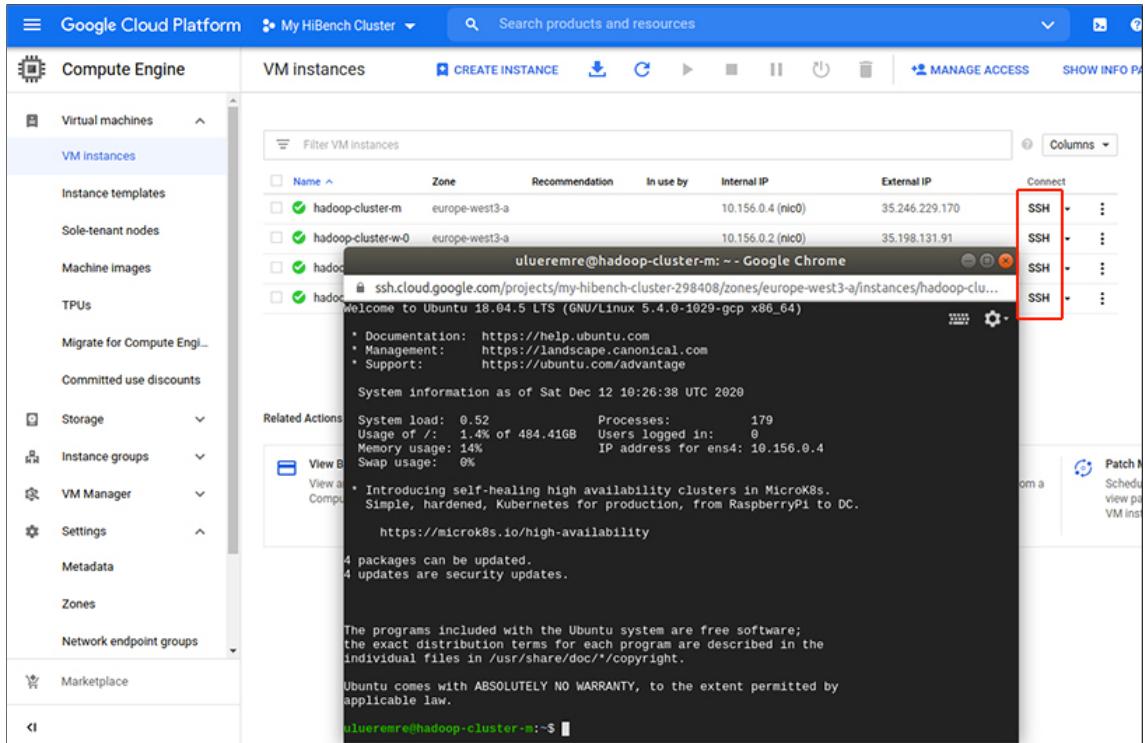


Figure 1.20: GCP Dataproc - Create Cluster (Step 12)

Configured Capacity:	2.84 TB
DFS Used:	61.67 GB (2.12%)
Non DFS Used:	19.36 GB
DFS Remaining:	2.76 TB (97.21%)
Block Pool Used:	61.67 GB (2.12%)
DataNodes usages% (Min/Median/Max/stdDev):	2.01% / 2.06% / 2.30% / 0.13%
Live Nodes	3 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)

Figure 1.21: GCP Dataproc - Namenode Manager

The screenshot shows the 'All Applications' page of the GCP Dataproc Resource Manager. The page title is 'All Applications'. The left sidebar includes sections for Cluster (status: NEW, NEW\_WAITING, ACCEPTED, PENDING, ENQUEUED, RUNNING, FINISHED, CANCELLED), Scheduler (Capacity Scheduler), and Tools. The main content area displays cluster metrics and application details.

**Cluster Metrics**

App Submitted	Apps Pending	Apps Planning	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	1	0	12	72 GB	72 GB	0 B	12	12	0

**Cluster Nodes Metrics**

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
3	0	0	0	0	0	0

**Scheduler Metrics**

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:24576, vCores:4>	0

**Applications**

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1607767990070_0001	uluerenre	random-text-writer	MAPREDUCE	default	0	Sat Dec 12 14:44:49 +0300 2020	N/A	RUNNING	UNDEFINED	12	12	73728	0	0	100.0	100.0	0	0	applicationMaster

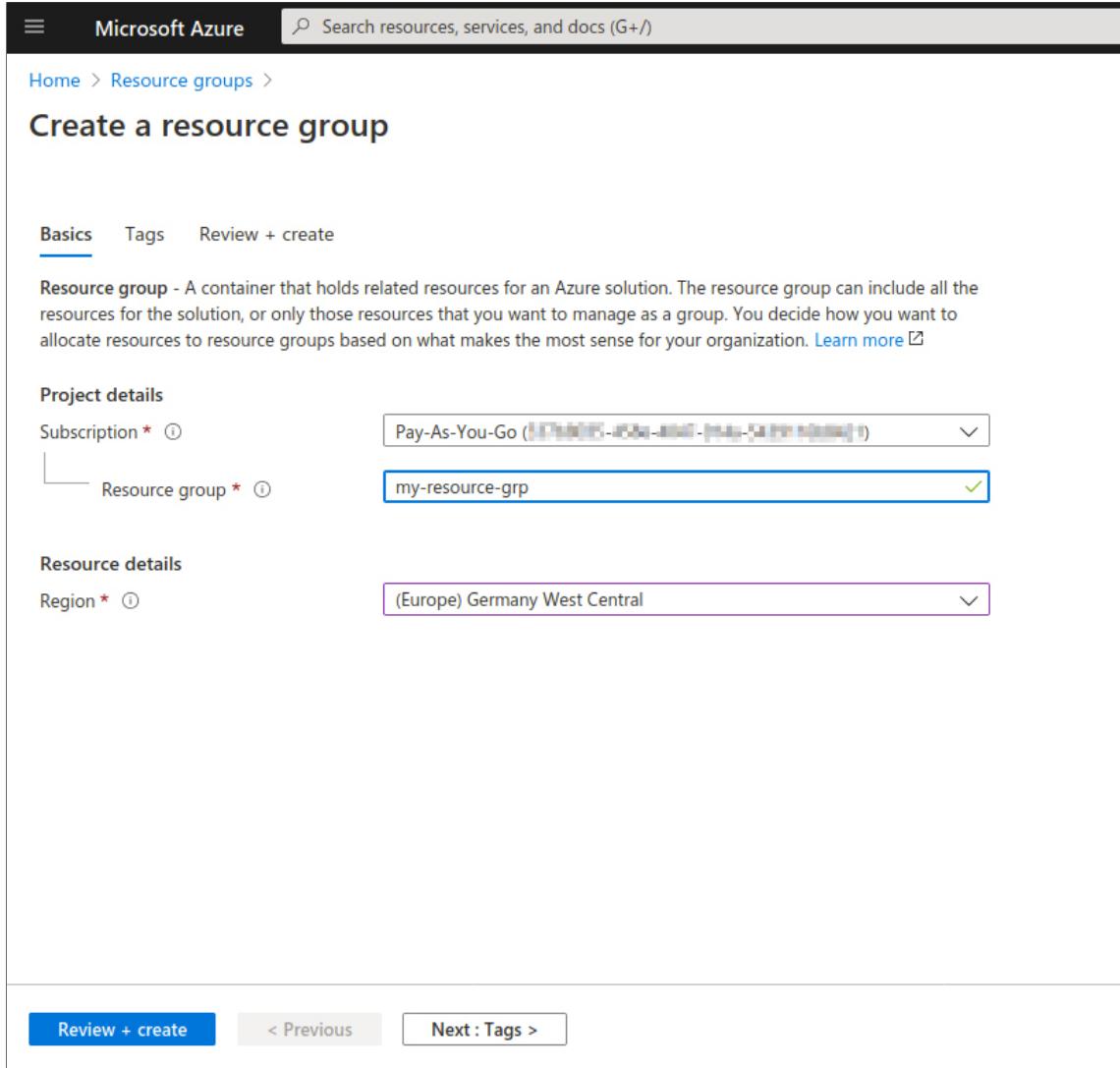
Showing 1 to 1 of 1 entries

Figure 1.22: GCP Dataproc - Resource Manager

## 2. CREATING HDINSIGHT CLUSTER ON AZURE

Working on Azure requires a Microsoft account; an account can be created at <https://login.live.com>.

Figure 2.1: On the Azure portal, we first create a resource group. A resource group is a collection of resources that share the same lifecycle, permissions, and policies.



**Figure 2.1: HDInsight setup - Create a resource group**

Figure 2.2: The subscription has to be registered with Microsoft.HDInsight. From the subscription's Resource Provider page we can locate HDInsight and register it.

Figure 2.3: Next step is to create an HDI cluster. To do so, we first navigate to HDI console by entering the name HDInsight into the searchbar on the top of the page.

The screenshot shows the Microsoft Azure Subscriptions page under the 'Pay-As-You-Go' section. On the left, there's a sidebar with 'Subscriptions' and a search bar. The main area displays a table of subscriptions. One row for 'Microsoft.HDInsight' is selected, showing its status as 'NotRegistered'. A red box highlights the 'Register' button at the top of the table. Another red box highlights the 'Resource providers' link in the bottom navigation menu.

Figure 2.2: HDInsight setup - Register subscription

The screenshot shows the Microsoft Azure Services page. The search bar at the top contains 'HDInsight'. Below the search bar, there's a 'Services' section with a list of items: 'HDInsight clusters', 'Workload Insights', 'Application Insights', 'Time Series Insights environments', and 'Time Series Insights access policies'. The 'HDInsight clusters' item is highlighted with a red box.

Figure 2.3: HDInsight setup - Navigate to HDInsight

Figure 2.4: From the HDI console, click on "Create HDInsight Cluster"

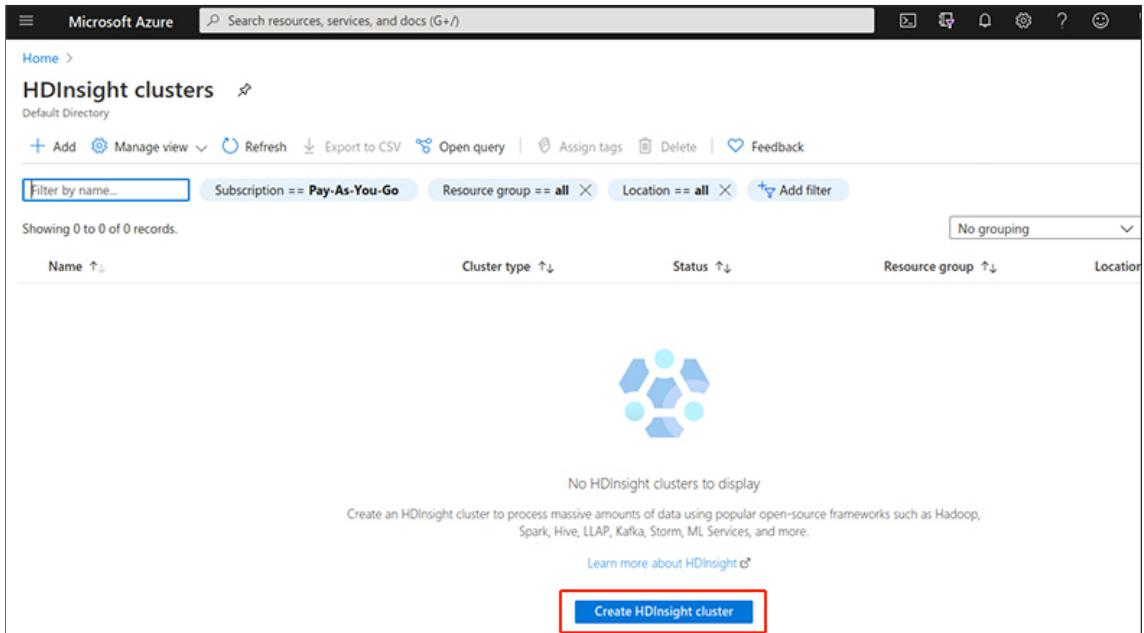


Figure 2.4: HDInsight setup - Create Cluster

Figure 2.5: Select subscription and resource group. Give cluster a name and select *Germany West Central* as Region. The cluster type is Hadoop, version will be *HDI 3.6*. For remote CLI operations, remote logon with an SSH Key is recommended, for the sake of the purpose, we check *Use cluster login as password for SSH*.

Figure 2.6: Leave the storage settings with their defaults.

Figure 2.7: Step 3: As with storage options, we leave Network settings with their defaults.

Figure 2.8: HDInsight assigns an obligatory High Availability option for the master node which doubles the resource usage. By the date the study has been made Zookeeper configuration has also been assigned to 3 free of charge nodes by HDInsight. We specify the master and worker nodes' machine types by the following choices: *A8m v2* for master node, and *A4m v2* for 3 worker nodes. Note: Due to resource usage regulations it is likely that the user faces issues on core availability on this page. Opening a support ticket for rising the quota limit may solve this issue within 24 to 48 hours.

Figure 2.9: Review the validated settings and click on Create button. Any issues related with the settings are displayed here so that they can be fixed before creation process.

Figure 2.10 The creation process takes some time and can be tracked from the HDInsight cluster overview page.

Figure 2.11 After successful installation a message appears.

Microsoft Azure Search resources, services & more

Home > HDInsight clusters >

## Create HDInsight cluster

[Basics](#) [Storage](#) [Security + networking](#) [Configuration + pricing](#) [Tags](#) [Review + create](#)

New to HDInsight? Get started with our [training resources](#).  
Create a managed HDInsight cluster. Select from Spark, Kafka, Hadoop, Storm, and more. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* [Pay-As-You-Go \(my-subscription\)](#) [Create new](#)

Resource group \* [my-resource-group](#) [Create new](#)

**Cluster details**

Name your cluster, pick a region, and choose a cluster type and version. [Learn more](#)

Cluster name \* my-benchmark-cluster [Change](#)

Region \* Germany West Central [Change](#)

Cluster type \* Hadoop [Change](#)

Version \* Hadoop 2.7.3 (HDI 3.6) [Change](#)

**Cluster credentials**

Enter new credentials that will be used to administer or access the cluster.

Cluster login username \* admin

Cluster login password \* [Change](#)

Confirm cluster login password \* [Change](#)

Secure Shell (SSH) username \* sshuser

Use cluster login password for SSH

[Review + create](#) [« Previous](#) [Next: Storage »](#)

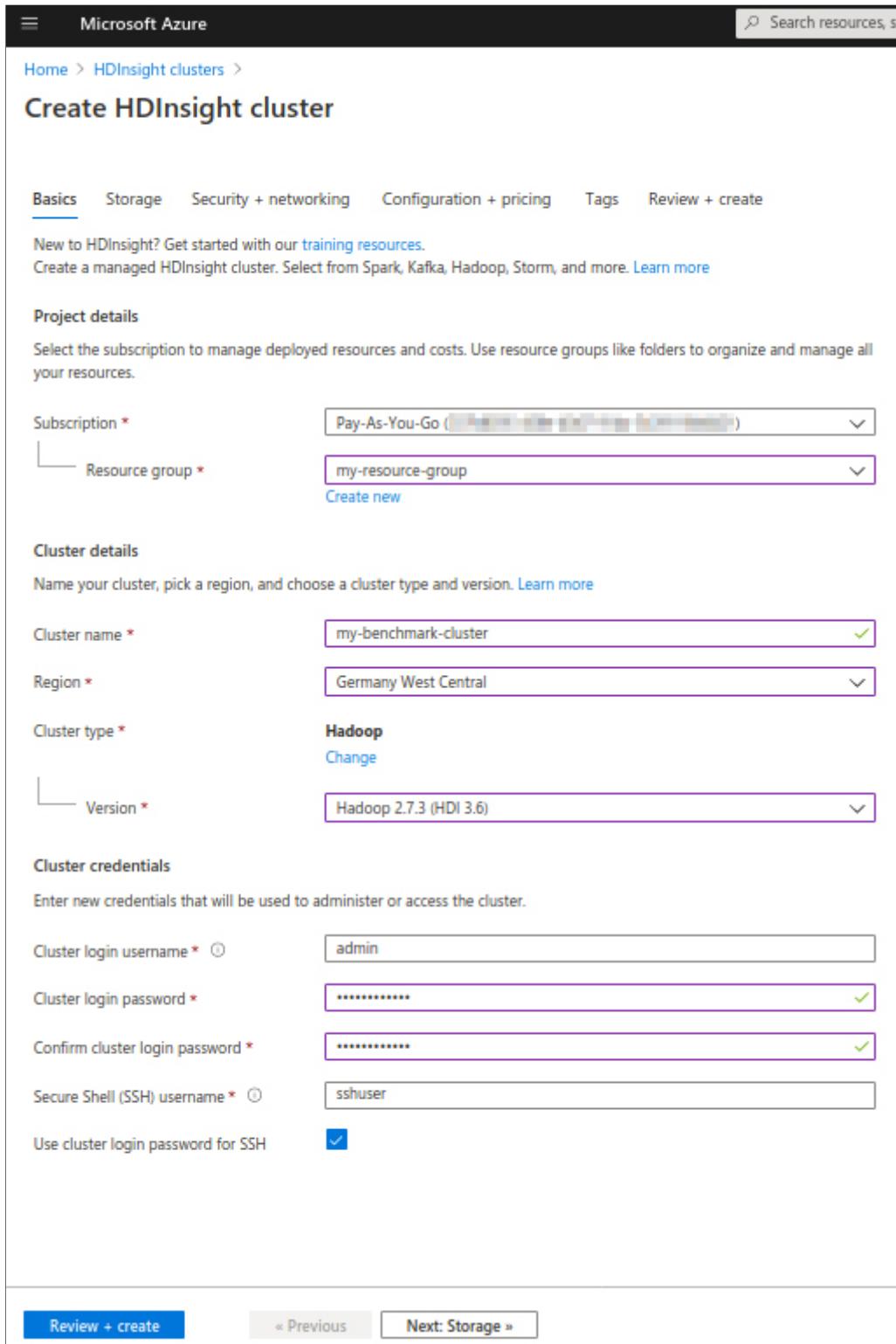


Figure 2.5: HDInsight setup - Create Cluster

Microsoft Azure

Home > HDInsight clusters >

## Create HDInsight cluster

Basics Storage Security + networking Configuration + pricing Tags Review + create

Select or create storage accounts that will be used for the cluster's logs, job input, and job output. Configure the cluster's access to these accounts, if needed.

**Primary storage**

Select or create a storage account that will be the default location for cluster logs and other output.

Primary storage type \*

Selection method \*  Select from list  Use access key

Primary storage account \*    
[Create new](#)

Container \*

**Data Lake Storage Gen1**

Provide details for the cluster to access Data Lake Storage Gen1. The cluster will be able to access any Data Lake Storage Gen1 accounts that the chosen service principal has access to.

Data Lake Storage Gen1 access

**Additional Azure Storage**

Link additional Azure Storage accounts to the cluster.

[Add Azure Storage](#)

**Custom Ambari DB**

Use an external Ambari database for greater flexibility, control, and customization. [Learn More](#)

SQL database for Ambari

**External metadata stores**

To store your Hive and Oozie metadata outside of this cluster, select a SQL database. [Learn More](#)

SQL database for Hive

SQL database for Oozie

Figure 2.6: HDInsight setup - Storage

Microsoft Azure  Search resources, services, an

Home > HDInsight clusters >

## Create HDInsight cluster

Basics Storage **Security + networking** Configuration + pricing Tags Review + create

Configure your cluster's security and network settings.

**Enterprise security package**

Connect this cluster with Active Directory Domain Services (AAD-DS) to have finer control of who can access the cluster. [Learn More](#)

Enable enterprise security package (Adds 0.072 TRY per Core-Hour)

**TLS**

Select the minimum TLS version supported for your cluster. [Learn more](#)

Minimum TLS version  1.2

**Network settings**

Resource provider connection  Inbound

Connect this cluster to a virtual network. [Learn more](#)

Virtual network

**Encryption in transit**

Configure encryption in transit settings. [Learn more](#)

Enable encryption in transit (

**Encryption at rest**

Configure disk encryption settings. [Learn more](#)

Provide your own key from key vault (

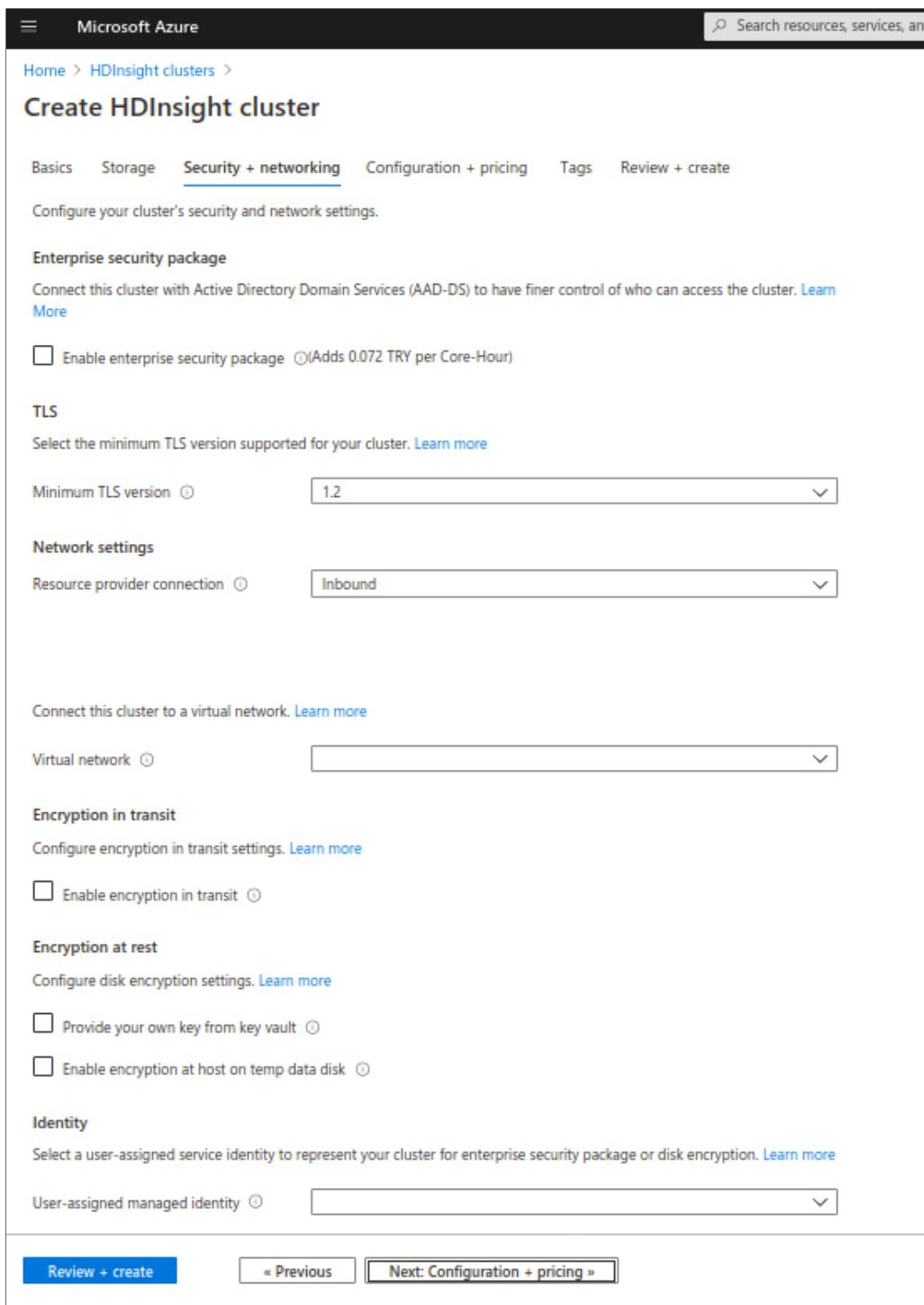
Enable encryption at host on temp data disk (

**Identity**

Select a user-assigned service identity to represent your cluster for enterprise security package or disk encryption. [Learn more](#)

User-assigned managed identity

**Review + create** [« Previous](#) [Next: Configuration + pricing »](#)



**Figure 2.7: HDInsight setup - Networking**

The screenshot shows the 'Create HDInsight cluster' configuration page in the Microsoft Azure portal. The top navigation bar includes 'Microsoft Azure', a search bar, and a 'Home > HDInsight clusters >' breadcrumb. The main title is 'Create HDInsight cluster'. Below it, a navigation bar has tabs: Basics, Storage, Security + networking, Configuration + pricing (which is underlined), Tags, and Review + create.

The 'Configuration + pricing' section contains a note: 'Configure cluster performance and pricing. [Learn more](#)'. A 'Node configuration' section allows setting cluster size and performance, with a note: 'Configure your cluster's size and performance, and view estimated cost information.' It also states: 'The cost estimate represented in the table does not include subscription discounts or costs related to storage, networking, or data transfer.'

A callout box provides information: 'This configuration will use 34 of 34 available cores in the Germany West Central region.' with a link 'View cores usage'.

The 'Node type' table lists three node types:

Node type	Node size	Number of ...	Estimated cost/h...
Head node	A8m v2 (8 Cores, 64 GB RAM), 4.40 TRY/hour	2	9.82 TRY
Zookeeper node	A2 v2 (2 Cores, 4 GB RAM), 0.90 TRY/hour	3	0.00 (FREE)
Worker node	A4m v2 (4 Cores, 32 GB RAM), 2.20 TRY/hour	3	7.08 TRY

An unchecked checkbox 'Enable autoscale' has a 'Learn More' link.

The total estimated cost per hour is 16.91 TRY.

The 'Script actions' section allows running custom PowerShell or Bash scripts on cluster nodes during provisioning, with a link 'Learn about script actions'.

At the bottom, there are buttons for '+ Add application', '+ Add script action', 'Review + create', '« Previous', and 'Next: Tags »'.

**Figure 2.8: HDInsight setup - Configuration**

Microsoft Azure Search resources

Home > HDInsight clusters >

## Create HDInsight cluster

Validation succeeded.

Basics Storage Security + networking Configuration + pricing Tags **Review + create**

Hadoop 2.7.3 (HDI 3.6) **16.91 TRY Total estimated cost/hour**  
This estimate does not include subscription discounts or costs related to storage, networking, or data transfer.

**Basics**

Subscription	Pay-As-You-Go
Resource group	my-resource-group
Region	Germany West Central
Cluster name	(new) my-benchmark-cluster
Cluster type	Hadoop 2.7.3 (HDI 3.6)
Cluster login username	admin
Secure Shell (SSH) username	sshuser
Use cluster login password for SSH	Enabled

**Security + networking**

Minimum TLS version	1.2
Resource provider connection	Inbound
Encryption at rest	Disabled
Encryption in transit	Disabled
Encryption at host on temp data disk	Disabled

**Storage**

Primary storage type	Azure Storage
Primary storage account	(new) mybenchmarkclhdstorage
Container	my-benchmark-cluster-2020-12-24t13-30-287z
Additional Azure Storage	None
Data Lake Storage Gen1 access	Disabled

**Cluster configuration**

Head	2 nodes, A8m v2 (8 Cores, 64 GB RAM)
Zookeeper	3 nodes, A2 v2 (2 Cores, 4 GB RAM)
Worker	3 nodes, A4m v2 (4 Cores, 32 GB RAM)

**Create** [« Previous](#) [Next »](#) [Download a template for automation](#)

Figure 2.9: HDInsight setup - Review

The screenshot shows the Microsoft Azure portal interface for an HDInsight deployment. The title bar reads "Microsoft Azure" and "Search resources, services, and docs (G+/-)". The main content area is titled "HDInsight\_2020-12-24T14.09.45.788Z | Overview". On the left, a navigation menu includes "Overview", "Inputs", "Outputs", and "Template". The "Overview" tab is selected. A message box says "We'd love your feedback! →". Below it, a section titled "Deployment is in progress" displays deployment details: Deployment name: HDInsight\_2020-12-24T14.09.45.788Z, Subscription: Pay-As-You-Go, Resource group: my-resource-group. It also shows the start time: 12/24/2020, 5:09:16 PM and Correlation ID: [redacted]. A table titled "Deployment details (Download)" lists two resources: "mybenchmarkclhdstorage" (Type: Microsoft.Storage/storageAccounts) and "mybenchmarkclhdstorage" (Type: Microsoft.Storage/storageAccounts), both marked as "OK".

Figure 2.10: HDInsight setup - In progress

The screenshot shows the Microsoft Azure portal interface for an HDInsight deployment. The title bar reads "Microsoft Azure" and "Search resources, services, and docs (G+/-)". The main content area is titled "HDInsight\_2020-12-24T14.09.45.788Z | Overview". On the left, a navigation menu includes "Overview", "Inputs", "Outputs", and "Template". The "Overview" tab is selected. A message box says "We'd love your feedback! →". Below it, a section titled "Your deployment is complete" displays deployment details: Deployment name: HDInsight\_2020-12-24T14.09.45.788Z, Subscription: Pay-As-You-Go, Resource group: my-resource-group. It also shows the start time: 12/24/2020, 5:09:16 PM and Correlation ID: [redacted]. A table titled "Deployment details (Download)" is present. At the bottom, there is a "Next steps" section with a "Setup autoscale Recommended" link and a "Go to resource" button.

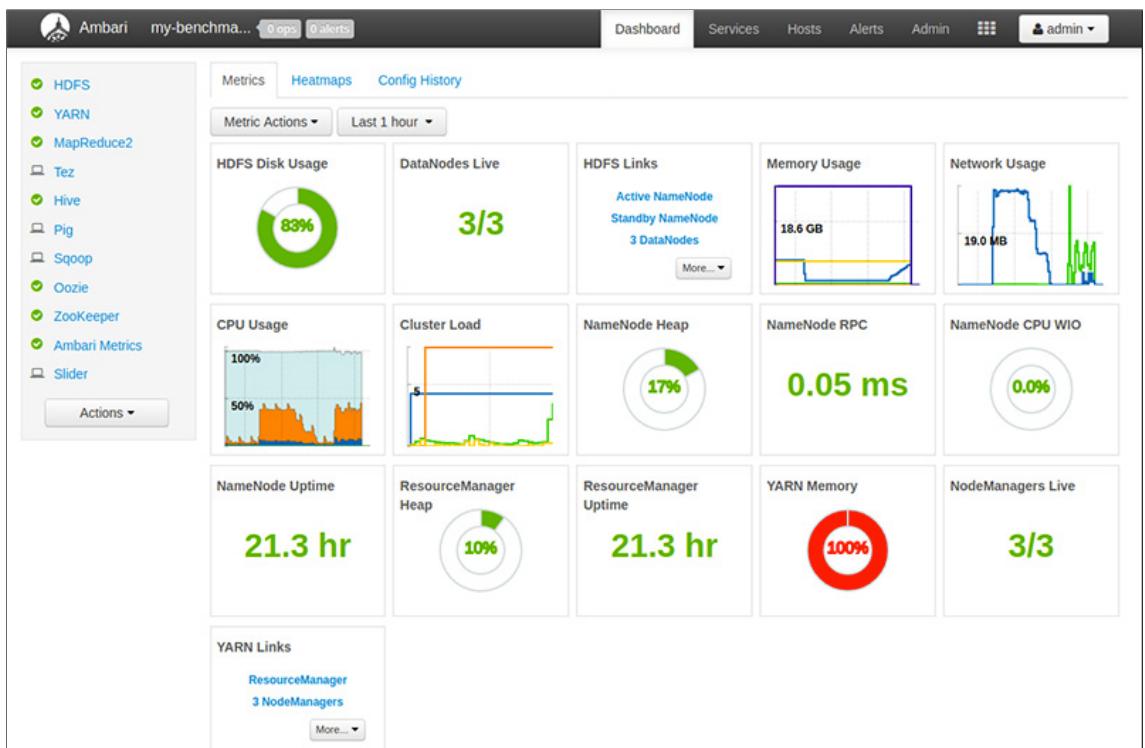
Figure 2.11: HDInsight setup - Complete

Figure 2.12 HDInsight comes with an HDP (Hortonworks Data Platform) cluster. The HDInsight overview page gives access to Ambari web UI.

The screenshot shows the Azure portal interface for managing HDInsight clusters. On the left, a sidebar lists 'my-benchmark-cluster' under 'HDInsight clusters'. The main panel displays the 'my-benchmark-cluster' details, including its status as 'Running', location 'Germany West Central', and Hadoop version '2.7 (HDI 3.6)'. A 'Dashboards' section contains two links: 'Ambari home' and 'Ambari views', with 'Ambari home' being highlighted by a red box.

**Figure 2.12: HDInsight setup - Overview**

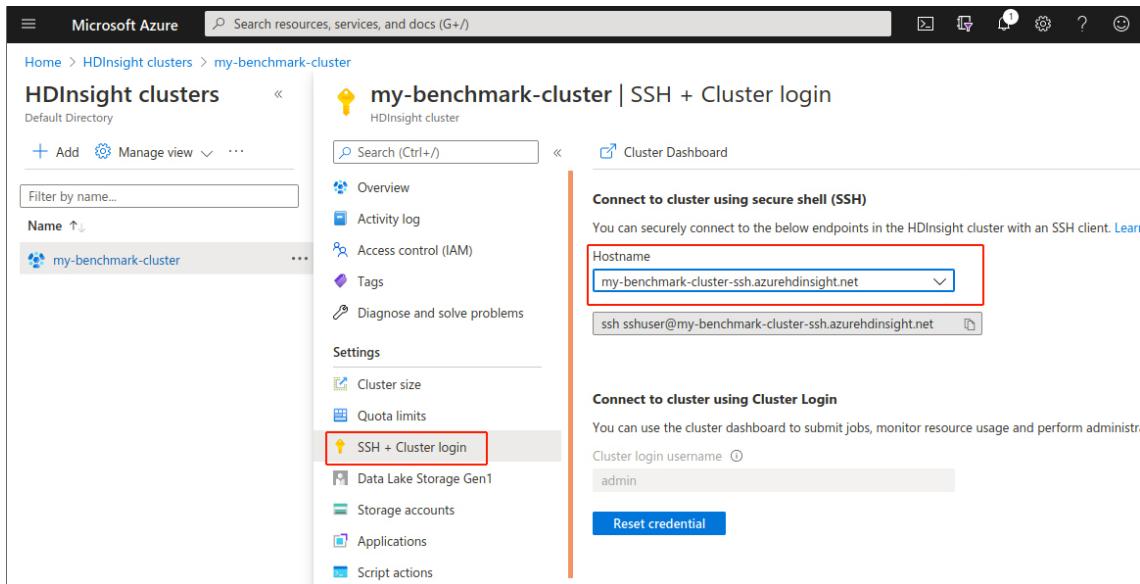
Figure 2.13 Ambari WebUI gives an overview to the current state of the cluster.



**Figure 2.13: HDInsight - Ambari Dashboard**

Figure 2.14 To conduct HiBench and system utilization data we need to connect to the cluster's master and worker nodes. From within the newly created HDI clusters's management portal,

clicking on *SSH + Cluster Login*, then selecting the host name will provide us with an ssh link to connect to the master node.



**Figure 2.14: HDInsight - Cluster login**

Figure 2.15 Using the password created on the HDInsight installation process connects the CLI to the master node.

Figure 2.16 To connect to the worker nodes we list the nodes within the cluster to find out the machine names. In this case our worker node 0, 1, and 2 names are wn0-my-ben, wn2-my-ben, and wn4-my-ben respectively.

Figure 2.17 In new CLI windows for each worker node we first connect to the master node, then from within the master node we connect to the respective worker nodes. The master node screen on top of the other windows is where HiBench benchmarks are executed. Within the worker nodes scripts capturing system utilization during benchmarking are executed.

```
sshuser@hn0-my-ben: ~
File Edit View Search Terminal Help
anka@anka-VirtualBox:~/azure$ ssh sshuser@my-benchmark-cluster-ssh.azurehdinsight.net
The authenticity of host 'my-benchmark-cluster-ssh.azurehdinsight.net (51.116.188.227)' can't be established.
ECDSA key fingerprint is SHA256:oslFURNIPqKDrm1IRGSv0tVJzWAuIITcWPPSmMXraX4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'my-benchmark-cluster-ssh.azurehdinsight.net,51.116.188.227' (ECDSA) to the list of known hosts.
Authorized users only. All activity may be monitored and reported.
sshuser@my-benchmark-cluster-ssh.azurehdinsight.net's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-1100-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

*** /dev/sda1 will be checked for errors at next reboot ***

Welcome to HDInsight.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sshuser@hn0-my-ben:~$
```

Figure 2.15: HDInsight - Connecting to master node

```
sshuser@hn0-my-ben: ~
File Edit View Search Terminal Help
sshuser@hn0-my-ben:~$ curl -u admin -sS -G "https://my-benchmark-cluster.azurehdinsight.net/api/v1/clusters/my-benchmark-cluster/hosts" | jq '.items[].Hosts.host_name'
Enter host password for user 'admin':
"hn0-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"hn1-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"wn0-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"wn2-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"wn4-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"zk0-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"zk2-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
"zk3-my-ben.spwjsqjedwzuxanrjnths2a5ac.frax.internal.cloudapp.net"
sshuser@hn0-my-ben:~$
```

Figure 2.16: HDInsight - Listing worker nodes

The screenshot shows a terminal window with four tabs, each representing a different HDInsight node:

- sshuser@hn0-my-ben:** Shows the command: `curl -v -sS -G "https://my-benchmark-cluster.azurehdinsight.net/api/v1/clusters/my-benchmark-cluster/hosts" | jq '.items[].Hosts.host_name'`. It lists several host names: "hn0-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net", "hn1-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net", "wn0-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net", "wn2-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net", "wn4-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net", "zk2-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net", and "zk3-my-ben.spwjsqjedwzuxanrjnth52a5ac.frax.internal.cloudapp.net".
- sshuser@hn2-my-ben:** Shows the command: `curl -v -sS -G "https://my-benchmark-cluster.azurehdinsight.net/api/v1/clusters/my-benchmark-cluster/hosts" | jq '.items[].Hosts.host_name'`.
- sshuser@vn4-my-ben:** Shows the command: `curl -v -sS -G "https://my-benchmark-cluster.azurehdinsight.net/api/v1/clusters/my-benchmark-cluster/hosts" | jq '.items[].Hosts.host_name'`.
- sshuser@hn0-my-ben:** Shows the command: `curl -v -sS -G "https://my-benchmark-cluster.azurehdinsight.net/api/v1/clusters/my-benchmark-cluster/hosts" | jq '.items[].Hosts.host_name'`.

The bottom part of the terminal shows a standard Ubuntu welcome message and help text for the terminal:

```

Ubuntu comes with some pre-installed programs and documentation.
Welcome to HDInsight.

To run a command
See "man sudo_root". Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law. The exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

To run a command
See "man sudo". Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sshuser@hn2-my-ben:~$ 

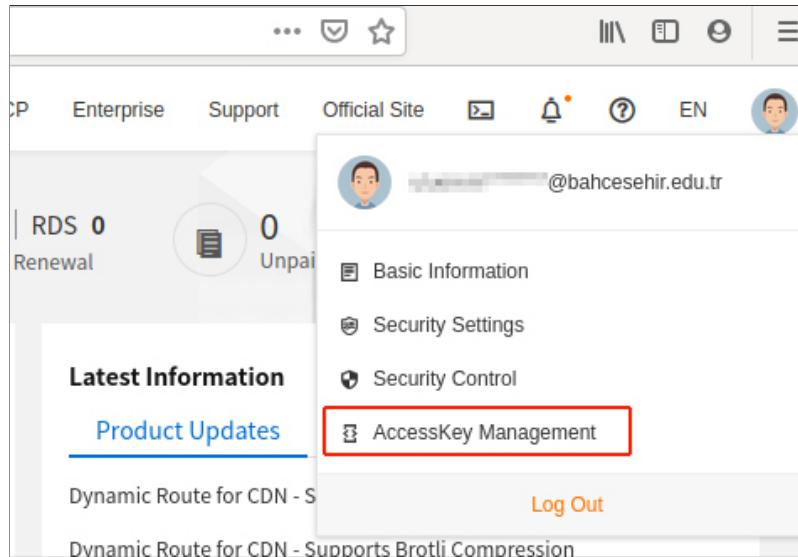
```

Figure 2.17: HDInsight - Ready to benchmark

### 3. CREATING E-MAPREDUCE CLUSTER ON ALIBABA CLOUD

An Alibaba Cloud account is required and can be created at the url <https://www.alibabacloud.com>.

Figure 3.1 First we create an access key by clicking AccessKey Management link on the upper right hand from the user menu and following the instructions.



**Figure 3.1: e-MapReduce - Create access key**

Figure 3.2, 3.3, 3.4 For e-MapReduce to run, OSS (Object Storage Service) also needs to be activated.

Figure 3.5 In OSS, setup a bucket for e-MapReduce to store logs.

Figure 3.6 Alibaba leverages OSS to store access logs. Within the created bucket's settings enable logging access to the bucket.

Figure 3.7 Before starting with e-MapReduce, select *Germany, Frankfurt* as region, where cluster will be created.

Figure 3.8 From Alibaba Cloud dashboard, select e-MapReduce.

Figure 3.9 From Alibaba Cloud EMR console, click on Cluster Wizard to start cluster installation.

Figure 3.10 *EMR-3.32.0* is the image version supporting Hadoop 2, and subject to selection.

Figure 3.11 Make selections below regarding Hardware settings. Existing VPC switch and a security group are required, these can be created by clicking on relevant links on the below screenshot (CreateVPC/V switch and Create Security Group). High availability of Master Node is not re-

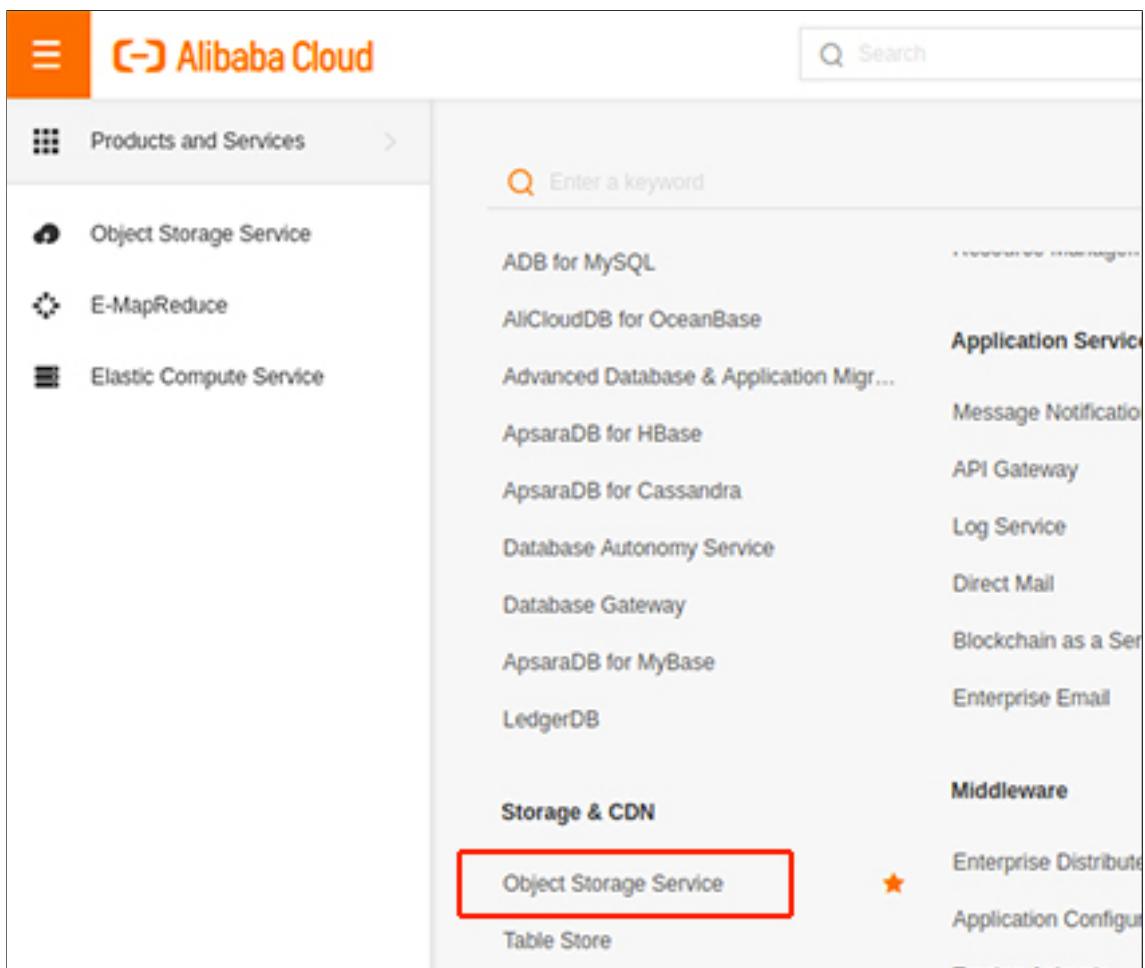


Figure 3.2: e-MapReduce - Activate OSS (Step 1)

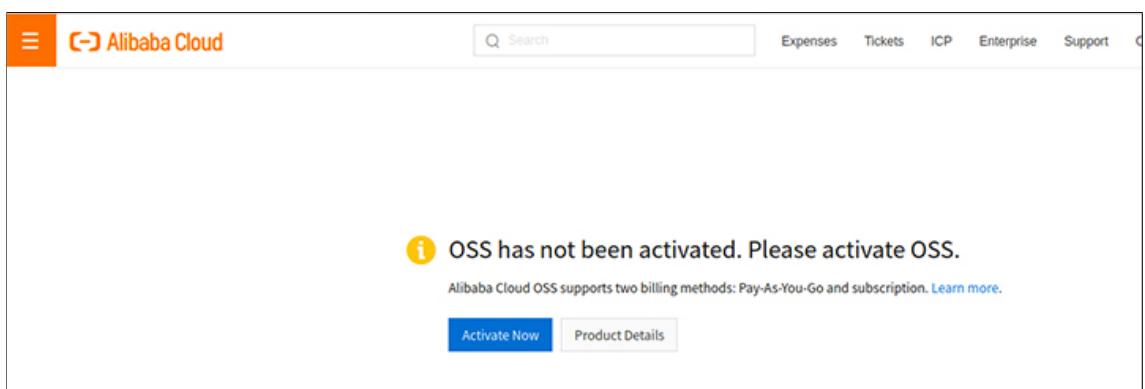
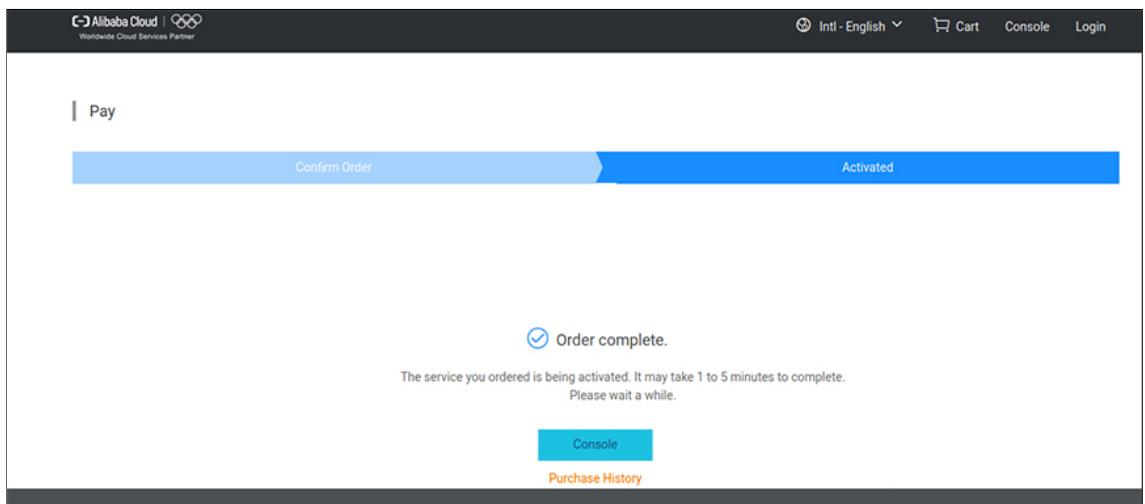
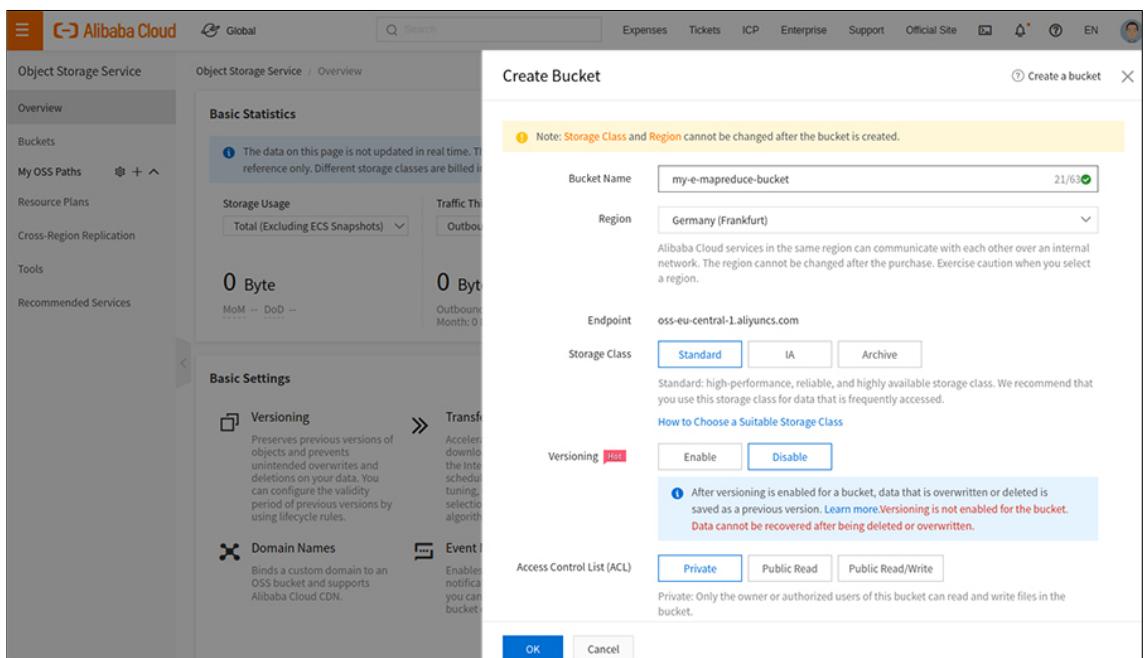


Figure 3.3: e-MapReduce - Activate OSS (Step 2)



**Figure 3.4: e-MapReduce - Activate OSS (Step 3)**

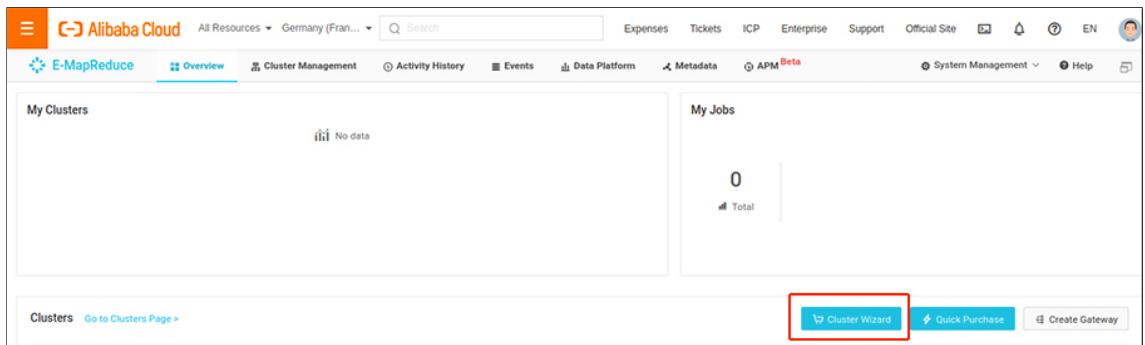


**Figure 3.5: e-MapReduce - Create OSS bucket**

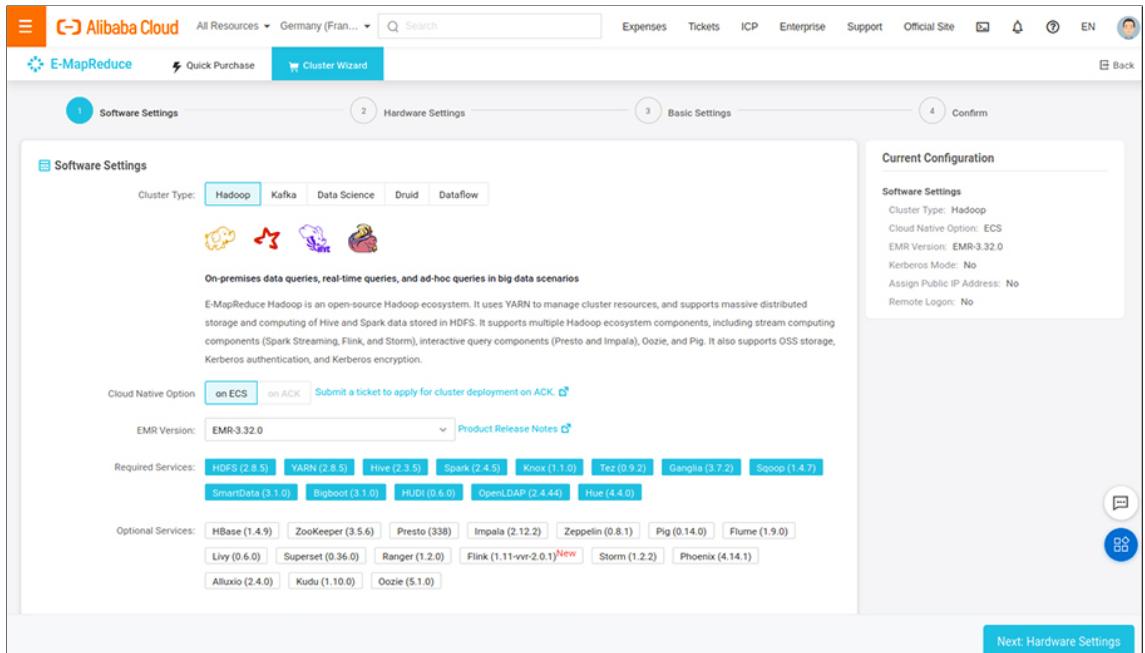
**Figure 3.6: e-MapReduce - Enabling access logging on OSS**

**Figure 3.7: e-MapReduce - Select Region**

**Figure 3.8: e-MapReduce - Create cluster (Step 1)**

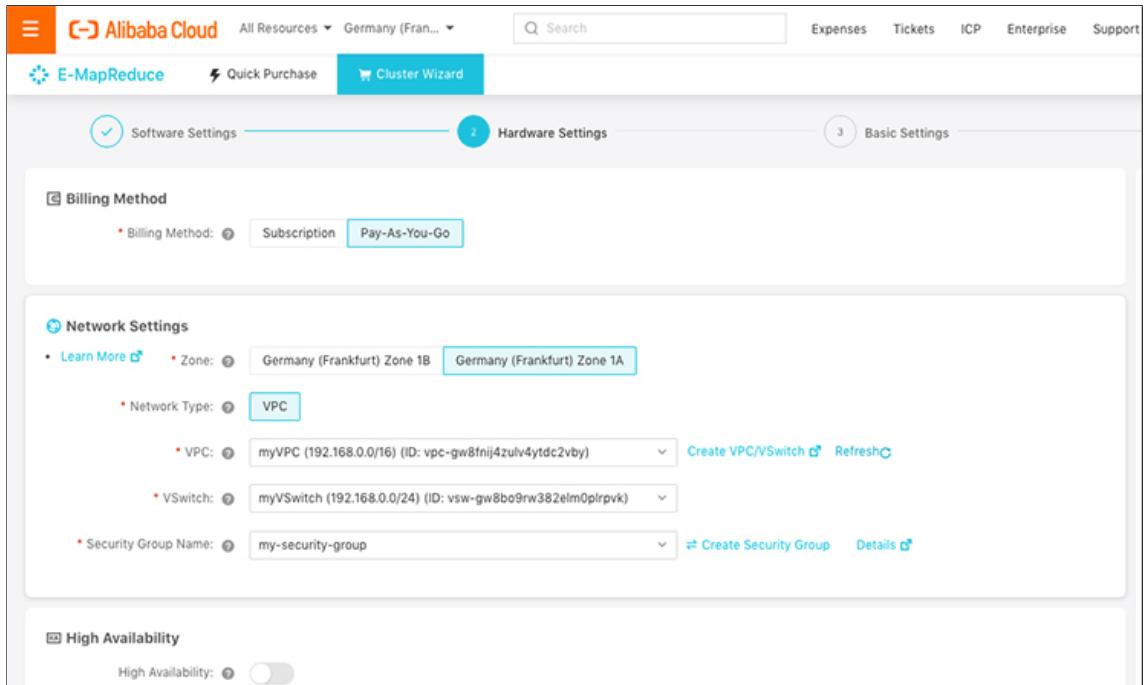


**Figure 3.9: e-MapReduce - Create cluster (Step 2)**



**Figure 3.10: e-MapReduce - Create cluster (Step 3)**

quired in our case, so we leave it unchecked as its default.



**Figure 3.11: e-MapReduce - Create cluster (Step 4)**

Figure 3.12 For Master Instance selections we chose memory optimized pre-configurations due to namenode's high memory load. Select *ecs.se1.2xlarge* as machine type, specify 500 GB for Master Node data storage.

Figure 3.13 For the worker nodes (inside the tab Core Instances) select *ecs.se1.xlarge*, in order to leverage 4 CPUs and 32 GB RAM per worker node. Storage capacity is set to 1000 GB (280 GB \* 4 disks, considering that 120 GB system disk size reserved for each node) for data. Leaving system storage capacity with its default of 120 GB. Number of worker nodes is specified as 3.

Figure 3.14 Final step before confirming installation is to give a name for the cluster and specify password for operations. Enable the checkbox *Assign Public IP Address* for later accessing the cluster over internet. Enable Remote Logon and select ssh key created before, if there is no existent ssh key pair, *Create Key Pair* next to the key pair selection box to do so.

Figure 3.15 Review settings. On the bottom of the page activate checkbox to accept terms and click on *Create*.

Figure 3.16 The creation process is displayed on e-MapReduce cluster page.

Figure 3.17 Within Alibaba Cloud's *Elastic Compute Service, Instances* page, single VMs of the cluster are also available.

The screenshot shows the Alibaba Cloud E-MapReduce Cluster Wizard interface. At the top, there are tabs for 'E-MapReduce' (selected), 'Quick Purchase', and 'Cluster Wizard'. Below that, there are tabs for 'Master Instance', 'Core Instance', and 'Task Instance', with 'Master Instance' being the active tab. Under 'Master Instance', there are several tabs: 'General Purpose', 'Compute Optimized', 'Memory Optimized' (which is selected and highlighted in blue), 'High Clock Speed', 'Entry-Level (Shared)', and 'GPU'. A search bar is located at the top right.

The main area displays a list of available master instance types, each with its name, configuration details, and bandwidth information:

	Name	Configuration	vCore	vMem	Band Width
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.xlarge	4 vCPU	32 GiB	2.048 Gbps
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.2xlarge	8 vCPU	64 GiB	3.072 Gbps
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.4xlarge	16 vCPU	128 GiB	6.144 Gbps
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.8xlarge	32 vCPU	256 GiB	10.240 Gbps
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.13xlarge	52 vCPU	384 GiB	16.384 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.xlarge	4 vCPU	32 GiB	1.536 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.2xlarge	8 vCPU	64 GiB	2.560 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.3xlarge	12 vCPU	96 GiB	4.096 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.4xlarge	16 vCPU	128 GiB	5.120 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.6xlarge	24 vCPU	192 GiB	7.680 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.8xlarge	32 vCPU	256 GiB	10.240 Gbps
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.13xlarge	52 vCPU	384 GiB	12.800 Gbps
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.xlarge	4 vCPU	32 GiB	1.536 Gbps
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.2xlarge	8 vCPU	64 GiB	2.048 Gbps
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.4xlarge	16 vCPU	128 GiB	3.072 Gbps
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.8xlarge	32 vCPU	256 GiB	6.144 Gbps
<input type="radio"/>	Memory Optimized ecs.se1	ecs.se1.xlarge	4 vCPU	32 GiB	0.819 Gbps
<input checked="" type="radio"/>	Memory Optimized ecs.se1	ecs.se1.2xlarge	8 vCPU	64 GiB	1.536 Gbps
<input type="radio"/>	Memory Optimized ecs.se1	ecs.se1.4xlarge	16 vCPU	128 GiB	3.072 Gbps
<input type="radio"/>	Memory Optimized ecs.se1	ecs.se1.8xlarge	32 vCPU	256 GiB	6.144 Gbps

Below the instance list, it shows the current master node type as 'ecs.se1.2xlarge' and the node type as 'ecs.se1ne.2xlarge'. It also shows disk configuration options: System Disk Type (SSD selected), Disk Size (120 GB \* 1 Disks), Data Disk Type (SSD selected), Data Disk Size (500 GB \* 1 Disks), and Master Nodes (1).

**Figure 3.12: e-MapReduce - Create cluster (Step 5)**

The screenshot shows the Alibaba Cloud E-MapReduce Cluster Wizard interface. At the top, there are tabs for 'Master Instance', 'Core Instance' (which is selected), and 'Task Instance'. Below these tabs is a table of memory-optimized instances:

	General Purpose	Compute Optimized	Memory Optimized	Big Data	Local SSD	High Clock Speed	Entry-Level (Shared)	GPU
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.xlarge	vCore: 4 vCPU vMem: 32 GiB Band Width: 2.048 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.2xlarge	vCore: 8 vCPU vMem: 64 GiB Band Width: 3.072 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.4xlarge	vCore: 16 vCPU vMem: 128 GiB Band Width: 6.144 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.8xlarge	vCore: 32 vCPU vMem: 256 GiB Band Width: 10.240 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6e	ecs.r6e.13xlarge	vCore: 52 vCPU vMem: 384 GiB Band Width: 16.384 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.xlarge	vCore: 4 vCPU vMem: 32 GiB Band Width: 1.536 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.2xlarge	vCore: 8 vCPU vMem: 64 GiB Band Width: 2.560 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.3xlarge	vCore: 12 vCPU vMem: 96 GiB Band Width: 4.096 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.4xlarge	vCore: 16 vCPU vMem: 128 GiB Band Width: 5.120 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.6xlarge	vCore: 24 vCPU vMem: 192 GiB Band Width: 7.680 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.8xlarge	vCore: 32 vCPU vMem: 256 GiB Band Width: 10.240 Gbps					
<input type="radio"/>	Memory Optimized ecs.r6	ecs.r6.13xlarge	vCore: 52 vCPU vMem: 384 GiB Band Width: 12.800 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.xlarge	vCore: 4 vCPU vMem: 32 GiB Band Width: 1.536 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.2xlarge	vCore: 8 vCPU vMem: 64 GiB Band Width: 2.048 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.4xlarge	vCore: 16 vCPU vMem: 128 GiB Band Width: 3.072 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1ne	ecs.se1ne.8xlarge	vCore: 32 vCPU vMem: 256 GiB Band Width: 6.144 Gbps					
<input checked="" type="radio"/>	Memory Optimized ecs.se1	ecs.se1.xlarge	vCore: 4 vCPU vMem: 32 GiB Band Width: 0.819 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1	ecs.se1.2xlarge	vCore: 8 vCPU vMem: 64 GiB Band Width: 1.536 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1	ecs.se1.4xlarge	vCore: 16 vCPU vMem: 128 GiB Band Width: 3.072 Gbps					
<input type="radio"/>	Memory Optimized ecs.se1	ecs.se1.8xlarge	vCore: 32 vCPU vMem: 256 GiB Band Width: 6.144 Gbps					

Below the table, the 'Current Core Node' is listed as 'ecs.se1.xlarge' with 'Type:'. Configuration options include:

- System Disk Type:  SSD  Ultra Disk [Details](#)
- Disk Size:  GB \* 1 Disks (Capacity Range: 40 ~ 500 GB) IOPS 5400
- Data Disk Type:  SSD  Ultra Disk [Details](#)
- Disk Size:  GB \* 4 Disks (Capacity Range: 40 ~ 32768 GB) IOPS 10200
- Core Nodes:  Nodes

**Figure 3.13: e-MapReduce - Create cluster (Step 6)**

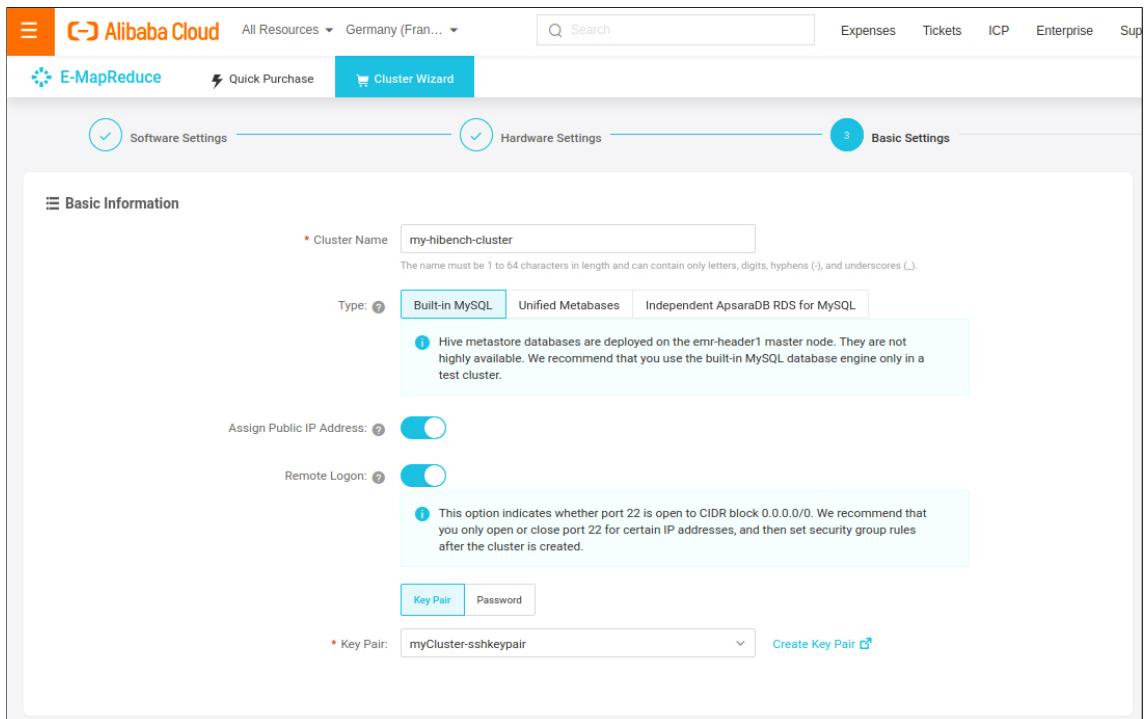


Figure 3.14: e-MapReduce - Create cluster (Step 7)

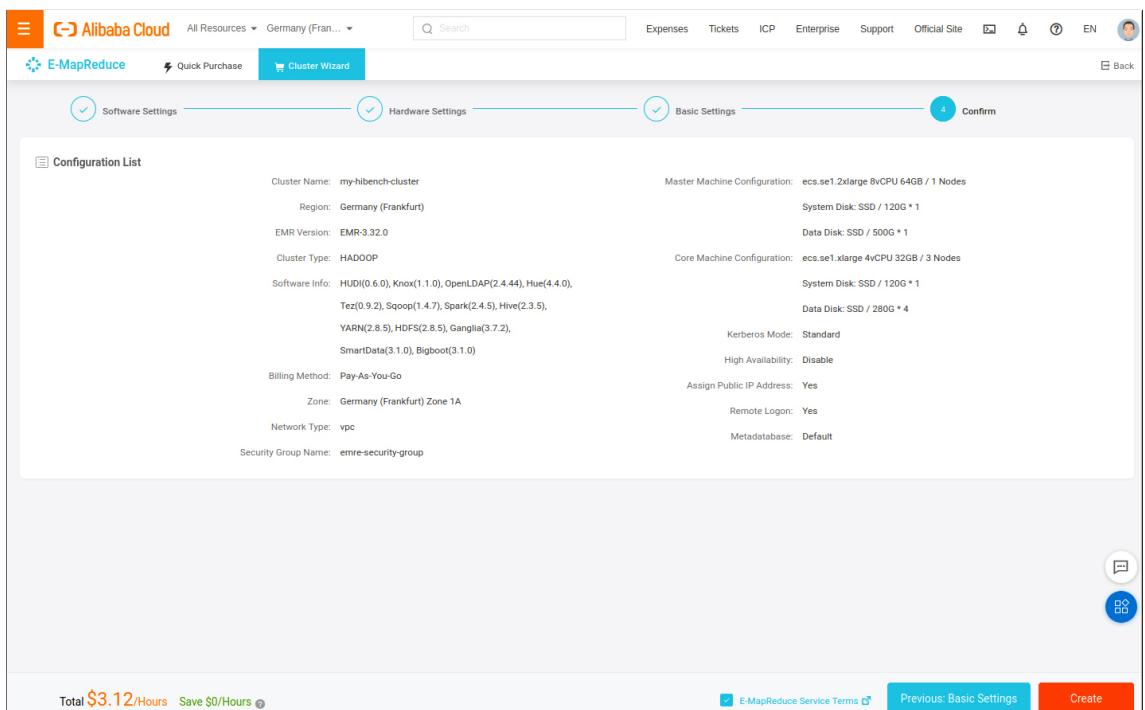


Figure 3.15: e-MapReduce - Create cluster (Step 8)

**Figure 3.16: e-MapReduce - Create cluster (Step 9)**

**Figure 3.17: e-MapReduce - Create cluster (Step 10)**

Figure 3.18 During the SSH key pair creation, respective pem file is downloaded to the local machine. Using the pem file a passwordless secure connection from Linux machine is possible.

```
$ chmod 400 ssh-keypair-name.pem
```

Within the terminal ssh connection is made by following command (instead of 0.0.0.0 the public IP address of the master node is entered):

```
$ ssh -i ssh-keypair-name.pem root@0.0.0.0
```

Entering following command from master node to check the Hadoop cluster:

```
$ hdfs dfsadmin -report
```

Figure 3.19 Worker nodes of the cluster are not accessible directly over the internet; connection to worker nodes is made over master node with following commands:

```
$ su hadoop  
$ ssh emr-worker-1
```

Figure 3.20 Handling broken pipe error: The CLI may fall into timeout causing connection break. To prevent this, we set up the ssh configuration file to send every 30 seconds an empty packet to the server, this will hold the connection live.

```
$ sudo nano ~/.ssh/config
```

Figure 3.21 depicts CLI connection on master and worker nodes of our e-MapReduce cluster, meaning that our system is ready to go with the benchmark processes.

```
root@emr-header-1:~  
File Edit View Search Terminal Help  
anka@anka-VirtualBox:~/Downloads$ chmod 400 myCluster-sshkeypair.pem  
anka@anka-VirtualBox:~/Downloads$ ssh -i myCluster-sshkeypair.pem root@  
The authenticity of host '192.168.0.153 (192.168.0.153)' can't be established.  
ECDSA key fingerprint is SHA256:0z+fGwjq5ydBxLEEWWdsNY8q07YiMb9oXRMA70zCZLc.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.0.153' (ECDSA) to the list of known hosts.  
Last login: Sat Dec 19 15:13:14 2020  
  
Welcome to Alibaba Cloud Elastic Compute Service !  
  
[root@emr-header-1 ~]# hdfs dfsadmin -report  
Configured Capacity: 3177490415616 (2.89 TB)  
Present Capacity: 3177289089024 (2.89 TB)  
DFS Remaining: 3176692304751 (2.89 TB)  
DFS Used: 596784273 (569.14 MB)  
DFS Used%: 0.02%  
Under replicated blocks: 0  
Blocks with corrupt replicas: 0  
Missing blocks: 0  
Missing blocks (with replication factor 1): 0  
Pending deletion blocks: 0  
  
-----  
Live datanodes (3):  
  
Name: 192.168.0.153:50010 (emr-worker-1.cluster-53371)  
Hostname: emr-worker-1.cluster-53371  
Decommission Status : Normal  
Configured Capacity: 1059163471872 (986.42 GB)  
DFS Used: 61530560 (58.68 MB)  
Non DFS Used: 0 (0 B)  
DFS Remaining: 1059034832448 (986.30 GB)  
DFS Used%: 0.01%  
DFS Remaining%: 99.99%  
Configured Cache Capacity: 0 (0 B)  
Cache Used: 0 (0 B)  
Cache Remaining: 0 (0 B)  
Cache Used%: 100.00%  
Cache Remaining%: 0.00%  
Xceivers: 2  
Last contact: Sat Dec 19 15:13:48 CST 2020
```

Figure 3.18: e-MapReduce - Connecting to the cluster

```

hadoop@emr-worker-1:~ 
File Edit View Search Terminal Help 
anka@anka-VirtualBox:~/Downloads$ ssh -i myCluster-sshkeypair.pem root@10.0.2.15 
Last login: Sat Dec 19 04:43:26 2020 

Welcome to Alibaba Cloud Elastic Compute Service ! 

[root@emr-header-1 ~]# su hadoop 
[root@emr-header-1 root]$ ssh emr-worker-1 
Last login: Sat Dec 19 04:28:56 2020 

Welcome to Alibaba Cloud Elastic Compute Service ! 

[hadoop@emr-worker-1 ~]$ 

```

**Figure 3.19: e-MapReduce - Connecting to worker nodes**

```

GNU nano 2.9.3                               /home/anka/.ssh/config

Host * 
  ServerAliveInterval 30 
  ServerAliveCountMax 5 

```

**Figure 3.20: e-MapReduce - Handling broken pipe error**

```

hadoop@emr-worker-1:~ 
hadoop@emr-worker-2:~ 
hadoop@emr-worker-3:~ 
anka@anka-VirtualBox:~/Downloads$ ssh -i myCluster-sshkeypair.pem root@8.209.79.208 
Last login: Sat Dec 19 15:41:22 2020 
Welcome to Alibaba Cloud Elastic Compute Service ! 
[root@emr-header-1 ~]# su hadoop 
[root@emr-header-1 root]$ ssh emr-worker-3 
Last login: Sat Dec 19 15:11:20 2020 
Are you sure you want to continue connecting (yes/no)? yes 
Host key verification failed. 
[root@emr-header-3 ~]# 
[root@emr-header-1 ~]# 
[root@emr-header-1 ~]# 
Name: 192.168.0.155:50010 (emr-worker-3.cluster-53371) 
Hostname: emr-worker-3.cluster-53371 
Decommission Status : Normal 
Configured Capacity: 1059163471872 (986.42 GB) 
DFS Used: 223174702 (212.84 MB) 
Non DFS Used: 0 (0 B) 
DFS Remaining: 1058873188306 (986.15 GB) 
DFS Used%: 0.02% 
DFS Remaining%: 99.97% 
Configured Cache Capacity: 0 (0 B) 
Cache Used: 0 (0 B) 
Cache Remaining: 0 (0 B) 
Cache Used%: 100.00% 
Cache Remaining%: 0.00% 
Xceivers: 2 
Last contact: Sat Dec 19 15:32:21 CST 2020 

```

**Figure 3.21: e-MapReduce - Ready to benchmark**

## 4. RUNNING HiBENCH ON GCP DATAPROC

On all master and worker nodes, we update repositories and dependencies.

```
$ sudo apt update  
$ sudo apt upgrade
```

For data collection of worker nodes' system resource utilization, install sysstat on all 3 worker nodes.

```
$ sudo apt install sysstat  
$ sar -V  
    sysstat version 11.6.1  
    (C) Sebastien Godard (sysstat <at> orange.fr)
```

On worker nodes, create a directory for storing resource utilization outputs:

```
$ mkdir data
```

Figure 4.1 Upload datacollector script on each worker node.

Make datacollector.sh executable.

```
$ sudo chmod +x datacollector.sh
```

HiBench and related processes are executed on the master node. Apache Maven is required to build HiBench. The first step is to install maven:

```
$ sudo apt install maven
```

Figure 4.2 Verify maven installation.

```
$ mvn --version
```

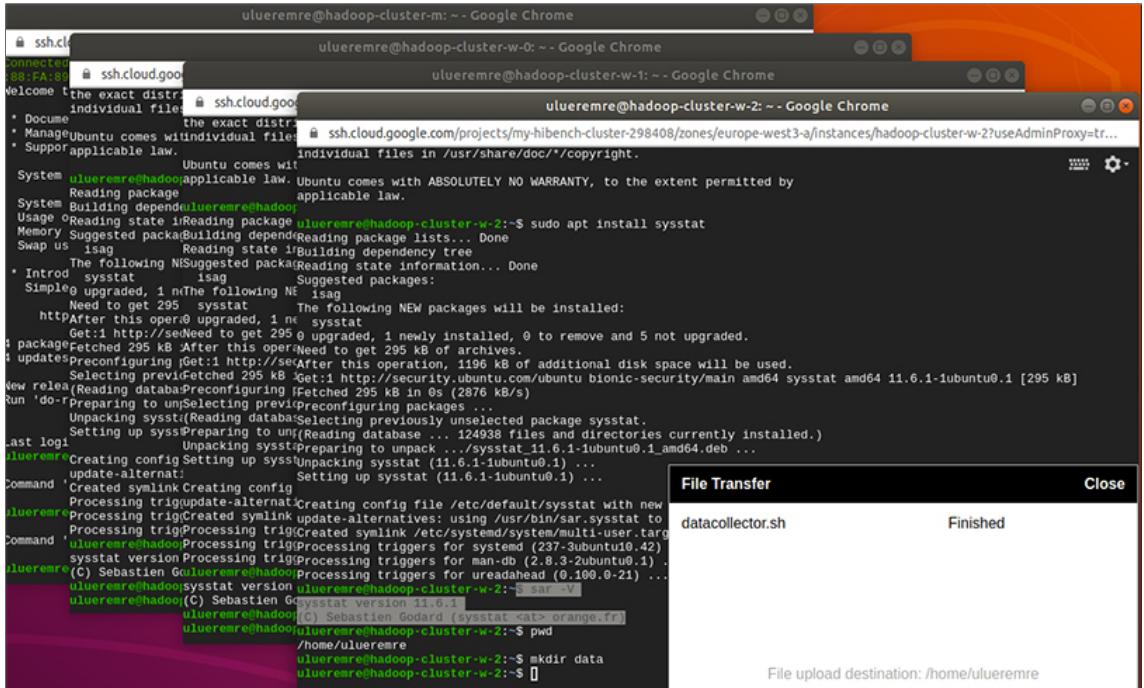


Figure 4.1: HiBench Dataproc - Uploading datacollector.sh

```
ulueremre@hadoop-cluster-m:~$ mvn --version
Apache Maven 3.6.0
Maven home: /usr/share/maven
Java version: 1.8.0_275, vendor: Private Build, runtime: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.4.0-1029-gcp", arch: "amd64", family: "unix"
ulueremre@hadoop-cluster-m:~$
```

Figure 4.2: HiBench Dataproc - Verify Maven installation

HiBench works with python 2, we check if python2 is installed

```
$ python2 -V  
Python 2.7.17
```

Download HiBench 7.1.1

```
$ wget https://github.com/Intel-bigdata/HiBench/archive/  
HiBench-7.1.tar.gz
```

Untar the downloaded file.

```
$ tar -zxf HiBench-7.1.tar.gz
```

Rename the extracted folder to a more user friendly name

```
$ mv HiBench-HiBench-7.1 HiBench
```

Navigate to HiBench folder

```
$ cd HiBench
```

Build HiBench7.1for Hadoop

```
$ mvn -Phadoopbench -Dspark=2.4 -Dscala=2.12 clean package
```

Figure 4.3 Upon successful HiBench compilation, an informative success message occurs. During the compilation failures might occur, re-running the above command would mostly fix this issue.

To modify HiBench's configuration files, navigate to HiBench's conf folder. In here, we will modify Hadoop and HiBench configurations.

```
$ cd conf/  
$ cp hadoop.conf.template hadoop.conf  
$ sudo nano hadoop.conf
```

```

ulueremre@hadoop-cluster-m: ~/HiBench - Google Chrome
ssh.cloud.google.com/projects/my-hibench-cluster-298408/zones/europe-west3-a/instances/hadoop-cluster-m?useAdminProxy...
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/ulueremre/HiBench/hadoopbench/nutchindexing/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.2:compile (default-compile) @ nutchindexing ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ nutchindexing ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/ulueremre/HiBench/hadoopbench/nutchindexing/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.2:testCompile (default-testCompile) @ nutchindexing ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ nutchindexing ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ nutchindexing ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: /home/ulueremre/HiBench/hadoopbench/nutchindexing/target/nutchindexing-7.1.jar
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] hibench 7.1 ..... SUCCESS [ 2.538 s]
[INFO] hibench-common 7.1 ..... SUCCESS [ 43.932 s]
[INFO] HiBench data generation tools 7.1 ..... SUCCESS [ 12.549 s]
[INFO] hadoopbench 7.1 ..... SUCCESS [ 0.002 s]
[INFO] hadoopbench-sql 7.1 ..... SUCCESS [ 6.155 s]
[INFO] mahout 7.1 ..... SUCCESS [01:56 min]
[INFO] PEGASUS: A Peta-Scale Graph Mining System 2.0-SNAPSHOT SUCCESS [ 2.323 s]
[INFO] nutchindexing 7.1 ..... SUCCESS [ 14.238 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 03:18 min
[INFO] Finished at: 2020-12-12T11:09:01Z
[INFO] -----
ulueremre@hadoop-cluster-m:~/HiBench$ 

```

**Figure 4.3: HiBench Dataproc - Success compiled**

hibench.hdfs.master value can be found in /usr/lib/hadoop/etc/hadoop/core-site.xml file (fs.default.name). Here we need to specify the hibench.hadoop.examples.test.jar manually for HiBench to run. Otherwise HiBench raises an Assertion error.

```

$ sudo nano hadoop.conf
# Hadoop home
hibench.hadoop.home      /usr/lib/hadoop

# The path of hadoop executable
hibench.hadoop.executable /usr/lib/hadoop/bin/hadoop

# Hadoop configuration directory
hibench.hadoop.configure.dir /usr/lib/hadoop/etc/hadoop

# The root HDFS path to store HiBench data
# hibench.hdfs.master value can be found in
# /usr/lib/hadoop/etc/hadoop/core-site.xml file
# (fs.defaultFS)
hibench.hdfs.master

```

```

hdfs://hadoop-cluster-m

hibench.hadoop.examples.test.jar /usr/lib/hadoop-mapreduce/
hadoop-mapreduce-client-jobclient-2.9.2-tests.jar

# Hadoop release provider. Supported value:
# apache, cdh5, hdp
hibench.hadoop.release      apache

```

Figure 4.4 depicts specified Hadoop settings for HiBench.

```

ulueremre@hadoop-cluster-m: ~/HiBench/conf - Google Chrome
ssh.cloud.google.com/projects/my-hibench-cluster-298408/zones/europe-west3-a/instances/hadoop-cluster-m?useAdminProxy=true&aut...
GNU nano 2.9.3                              .hadoop.conf                                         Modified
# Hadoop home
hibench.hadoop.home      /usr/lib/hadoop
#/PATH/TO/YOUR/HADOOP/ROOT

# The path of hadoop executable
hibench.hadoop.executable   /usr/lib/hadoop/bin/hadoop
#${hibench.hadoop.home}/bin/hadoop

# Hadoop configuration directory
hibench.hadoop.configure.dir /usr/lib/hadoop/etc/hadoop
#${hibench.hadoop.home}/etc/hadoop

# The root HDFS path to store HiBench data
hibench.hdfs.master        hdfs://hadoop-cluster-m
#hdfs://localhost:8020

hibench.hadoop.examples.test.jar      /usr/lib/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-2.9.2-tests.jar

# Hadoop release provider. Supported value: apache, cdh5, hdp
hibench.hadoop.release      apache

```

**Figure 4.4: HiBench Dataproc - Hadoop configurations**

Figure 4.5 HiBench related configuration settings like data scale and mappers/reducers count are made within *Hibench/conf/hibench.conf* file. Having 12 cores on the worker nodes, we specify a default of 12 mappers and 12 reducers to allocate during benchmark. For each data scale, before we run the benchmarks, *hibench.scale.profile* value has to be updated for the respective data scale:

```
$ sudo nano hibench.conf
```

As a showcase, continue with the manual implementation of UseCase 1.

```

ulueremre@hadoop-cluster-m: ~/HiBench/conf - Google Chrome
ssh.cloud.google.com/projects/my-hibench-cluster-298408/zones/europe-west3-a/instances/hadoop-cluster-m?useAdminProxy...
GNU nano 2.9.3                               hibench.conf                                Modified
# Data scale profile. Available value is tiny, small, large, huge, gigantic and bigdata.
# The definition of these profiles can be found in the workload's conf file i.e. conf/workloads/micro/wordcount
hibench.scale.profile          gigantic
# Mapper number in hadoop, partition number in Spark
hibench.default.map.parallelism    12
#8
#
# Reducer number in hadoop, shuffle partition number in Spark
hibench.default.shuffle.parallelism   12
#8

```

**Figure 4.5: HiBench Dataproc - HiBench configurations**

The benchmarks are executed from within root of HiBench folder. Within Use Case 1 and Use Case 2, benchmark tasks have been executed in an iterative process. We do not put every line of code in here, but to give the idea, the approach is given below:

- Within hibench.conf configuration file, we set up numbers of 12 mappers and 12 reducers.
- Set up hibench.conf configuration file with the respective data scale (tiny, small, large, huge, and gigantic. Due to account limitations we ignore the largest scale bigdata)
- On each worker node we located previously written bash script responsible for system activity collection in a specific directory (data directory). Datacollection script is started manually on all worker nodes short before a benchmark is run (not during preparation of the workload)
- On master node within HiBench, start the preparation script (no running data collectors on worker nodes)
- Once preparation of the workload is finished, start datacollector script o worker nodes
- On the master node start benchmark  
(HiBench/bin/workloads/[benchmark-class]/[benchmarkname]/hadoop/run.sh)
- Once the benchmark completes, stop data collection process on all worker nodes.

Example codes below firstly create the workload for the respective benchmark; 2, 3, and 4 are run across worker nodes one after another starting to capture system utilization by leveraging datacollector.sh script. Immediately in step 5, the respectie benchmark is executed. After the benchmark completion, data collecting activities on the worker nodes are terminated.

```
# 1. Prepare workload for benchmark MICRO-TERASORT
$ bin/workloads/micro/terasort/prepare/prepare.sh
```

```

# 2. Collect system utilization on Worker 0
$ ./datacollector.sh -p gcp-uc1-w0-g-tera

# 3. Collect system utilization on Worker 1
$ ./datacollector.sh -p gcp-uc1-w1-g-tera

# 4. Collect system utilization on Worker 2
$ ./datacollector.sh -p gcp-uc1-w2-g-tera

# 5. Run the benchmark MICRO-TERASORT
$ bin/workloads/micro/terasort/hadoop/run.sh

# 6. STOP data collection at worker nodes by pressing Ctrl+C)

```

The screenshot in Figure 4.6 depicts an ongoing benchmark execution with simultaneous data collection on the worker nodes. The CLI to the master node runs TeraSort benchmark while in parallel, in the worker nodes w0, w1, and w2 *datacollector.sh* script is active.

```

Bytes Written:3200000000000
finish HadoopPrepareTerasort bench
ulueremre@hadoop-cluster-m:~/HiBench$ ls conf/
benchmarks.lst frameworks.lst hadoop.conf hibench.conf storm.conf.template
flink.conf.template gearpump.conf.template hadoop.conf.template spark.conf.template workloads
ulueremre@hadoop-cluster-m:~/HiBench$ ls conf/workloads/
graph micro ml sql streaming websearch
ulueremre@hadoop-cluster-m:~/HiBench$ ls conf/workloads/micro/
dfsio.conf replication.conf sleep.conf sort.conf terasort.conf wordcount.conf
ulueremre@hadoop-cluster-m:~/HiBench$ ls conf/workloads/micro/terasort.conf
conf/workloads/micro/terasort.conf
ulueremre@hadoop-cluster-m:~/HiBench$ sudo nano conf/workloads/micro/terasort.conf
ulueremre@hadoop-cluster-m:~/HiBench$ sudo nano report/hibench.report
ulueremre@hadoop-cluster-m:~/HiBench$ bin/workloads/micro/terasort/hadoop/run.sh
patching args...
Parsing conf: /home/ulueremre/HiBench/conf/hadoop.conf
Parsing conf: /home/ulueremre/HiBench/conf/hibench.conf
Parsing conf: /home/ulueremre/HiBench/conf/workloads/micro/terasort.conf
probe sleep jar: /usr/lib/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-2.9.2-tests.jar
start HadoopPrepareTerasort
hdfs rm -r /usr/lib/hadoop/bin/hadoop --config /usr/lib/hadoop/etc/hadoop fs -rm -r -skipTrash hdfs://hadoop-cluster-m/HiBench/Terasort/Output
rm: 'hdfs://hadoop-cluster-m/HiBench/Terasort/Output': No such file or directory
hdfs du -s: /hadoop-cluster-m/HiBench/Terasort/Output
Submit MapReduce Job: /usr/lib/hadoop/bin/hadoop --config /usr/lib/hadoop/etc/hadoop jar /usr/lib/hadoop/..../hadoop-mapreduce-examples.jar terasort -D mapreduce.job.reduces=9 hdfs://hadoop-cluster-m/HiBench/Terasort/Input hdfs://hadoop-cluster-m/HiBench/Terasort/Output
2012/12/12 13:52:29 INFO mapreduce.Job: map 0% reduce 0%
Average: 3 2.25 65.27 9.32 18.01 0.32 4.82
ulueremre@hadoop-cluster-w0:~$ ./datacollector.sh -p gcp-uc1-w0-g-tera
ulueremre@hadoop-cluster-w1:~$ ./datacollector.sh -p gcp-uc1-w1-g-tera
ulueremre@hadoop-cluster-w2:~$ ./datacollector.sh -p gcp-uc1-w2-g-tera

```

**Figure 4.6: HiBench Dataproc - Profile from an ongoing benchmark execution**

## 5. RUNNING HiBENCH ON AZURE HDINSIGHT

Apache Maven is required to build HiBench. The first step is to install maven on the master node:

```
$ sudo apt install maven
```

Verify maven installation.

```
$ mvn -version
Apache Maven 3.3.9
Maven home: /usr/share/maven
Java version: 1.8.0_275, vendor: Private Build
Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en_US, platform encoding: ANSI_X3.4-1968
OS name: "linux", version: "4.15.0-1100-azure",
arch: "amd64", family: "unix"
```

Download HiBench 7.1.1

```
$ wget https://github.com/Intel-bigdata/HiBench/archive/
HiBench-7.1.tar.gz
```

Untar the downloaded file.

```
$ tar -zxf HiBench-7.1.tar.gz
```

Rename the extracted folder to a more user friendly name.

```
$ mv HiBench-HiBench-7.1 HiBench
```

Navigate to HiBench folder.

```
$ cd HiBench
```

Build HiBench7.1 for Hadoop.

```
$ mvn -Phadoopbench -Dspark=2.4 -Dscala=2.12 clean package
```

To modify HiBench configuration files, navigate to HiBench's conf folder. For HDInsight we also need to specify `hibench.hadoop.examples.test.jar` path and hadoop release (hpd)

```
$ cd conf/
$ cp hadoop.conf.template hadoop.conf
$ sudo nano hadoop.conf
# Hadoop home
hibench.hadoop.home      /usr/hdp/current/hadoop-client

# The path of hadoop executable
hibench.hadoop.executable   /usr/hdp/current/
                                hadoop-client/bin/hadoop

# Hadoop configuration directory
hibench.hadoop.configure.dir    /usr/hdp/current/
                                hadoop-client/etc/hadoop

hibench.hadoop.examples.test.jar /usr/lib/
                                hadoop-mapreduce/
                                hadoop-mapreduce-client-jobclient-2.9.2-tests.jar

# The root HDFS path to store HiBench data
# hibench.hdfs.master value can be found in
# core-site.xml file
hibench.hdfs.master        hdfs://cluster-92af-m

# Hadoop release provider.
# Supported value: apache, cdh5, hdp
hibench.hadoop.release      hdp
```

HiBench related configuration settings like data scale and mappers/reducers count are made within `Hibench/conf/hibench.conf` file. For each data scale, before we run the benchmarks, `hibench.scale.profile` value has to be updated for the respective data scale:

```

$ sudo nano hibench.conf

# Data scale profile. Available value is tiny, small,
# large, huge, gigantic and bigdata.
# The definition of these profiles can be found in
# the workload's conf file
# i.e. conf/workloads/micro/wordcount.conf

hibench.scale.profile          tiny
# Mapper number in hadoop, partition number in Spark
hibench.default.map.parallelism    12

# Reducer nubmer in hadoop, shuffle partition number
# in Spark
hibench.default.shuffle.parallelism    12

```

Figure 5.1 taken during benchmark execution depicts the process on master node and worker nodes.

```

sshuuser@wn0-my-ben: ~
File Edit View Search Terminal Help
File Edit View Search Terminal Help
Parsing conf: /home/sshuuser/HIBench/conf/htbench.conf
Parsing conf: /home/sshuuser/HIBench/conf/workloads/micro/wordcount.conf
probe sleep jar: /usr/hdp/2.6.5.3032-3/hadoop-mapreduce/hadoop-mapreduce-client-jobclient-2.7.3.2.6.5.3032-3-tests.jar
start HadoopWordCount bench
hdfs rm -r: /usr/hdp/current/hadoop-client/bin/hadoop --config /usr/hdp/current/hadoop-client/etc/hadoop fs -rm -r -skipTrash
ssh wasb://my-benchmark-cluster-2020-12-24t13-30-30-287z@mybenchmarkclhdstorage.blob.core.windows.net/HIBench/Wordcount/0
putput
rm: 'wasb://my-benchmark-cluster-2020-12-24t13-30-30-287z@mybenchmarkclhdstorage.blob.core.windows.net/HIBench/Wordc...
hdfs du -s: /usr/hdp/current/hadoop-client/bin/hadoop --config /usr/hdp/current/hadoop-client/etc/hadoop fs -du -s wasb://
my-benchmark-cluster-2020-12-24t13-30-30-287z@mybenchmarkclhdstorage.blob.core.windows.net/HIBench/Wordcount/Input
Submit MapReduce Job: /usr/hdp/current/hadoop-client/bin/hadoop --config /usr/hdp/current/hadoop-client/etc/hadoop jar /us
r/hdp/current/hadoop-client/.hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount -D mapreduce.job.maps=12 -D mapped
use.job.reduces=12 -D mapreduce.inputformat.class=org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat -D mapred
uce.outputformat.class=org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat -D mapreduce.job.inputformat.class=o
rg.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat -D mapreduce.job.outputformat.class=org.apache.hadoop.mapred
uce.lib.output.SequenceFileOutputFormat wasb://my-benchmark-cluster-2020-12-24t13-30-30-287z@mybenchmarkclhdstorage.blob.
core.windows.net/HIBench/Wordcount/Input wasb://my-benchmark-cluster-2020-12-24t13-30-30-287z@mybenchmarkclhdstorage.blob.
core.windows.net/HIBench/Wordcount/Output
The authenticity of host 'wn0-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net (10.0.0.11)' can't be established.
EDSA key fingerprint is SHA256:Mcu9Q7UZVOI2M0BNzr4japTls05Bxllh+z0Lts6RLog.
Are you sure you want to continue connecting (yes/no)? The authenticity of host 'wn2-my-ben.spwjsqjedwzuxanrnths2a5ac.fra
x.internal.cloudapp.net (10.0.0.10)' can't be established.
EDSA key fingerprint is SHA256:X4Q7Yxoy8SpzIsMX/sMAAJUDha7ya1/Sj0g2g68e38ZE.
Are you sure you want to continue connecting (yes/no)? The authenticity of host 'wn4-my-ben.spwjsqjedwzuxanrnths2a5ac.fra
x.internal.cloudapp.net (10.0.0.8)' can't be established.
EDSA key fingerprint is SHA256:fadLxyzsbNg135KSV1sdx5tu1wFMnKMV0fDWeRKA.
Are you sure you want to continue connecting (yes/no)? 20/12/24 18:22:36 INFO client.AHSProxy: Connecting to Application H
20/12/24 18:49:00 INFO mapreduce.Job: map 13% reduce 1%
sshuuser@wn0-my-ben:~$ ./datacollector.sh -p azu-uc2-w0-g-wrdcnt
sshuuser@wn0-my-ben:~$ ./datacollector.sh -p azu-uc2-w1-g-wrdcnt
sshuuser@wn0-my-ben:~$ ./datacollector.sh -p azu-uc2-w2-g-wrdcnt
sshuuser@wn2-my-ben:~$ ./datacollector.sh -p azu-uc2-w1-g-wrdcnt
sshuuser@wn2-my-ben:~$ ./datacollector.sh -p azu-uc2-w2-g-wrdcnt
sshuuser@wn4-my-ben:~$ ./datacollector.sh -p azu-uc2-w2-g-wrdcnt

```

Figure 5.1: Azure HDInsight - HiBench execution process configurations

## 6. RUNNING HiBENCH ON ALIBABA CLOUD E-MAPREDUCE

Aliyun Linux has sysstat package preinstalled, so we don't need to install it.

```
# sar -V
    sysstat version 10.1.5
    (C) Sebastien Godard (sysstat <at> orange.fr)
```

On each worker node, create a directory for storing resource utilization outputs:

```
# mkdir data
```

On each worker node create datacollector.sh file.

```
# touch datacollector.sh
```

Upload datacollector script to each worker node. Make datacollector.sh executable

```
# sudo chmod +x datacollector.sh
```

HiBench and related processes are executed on the master node. Apache Maven is required to build HiBench. The first step is to install maven. Because Aliyun Linux is CentOS based, yum package manager has to be leveraged:

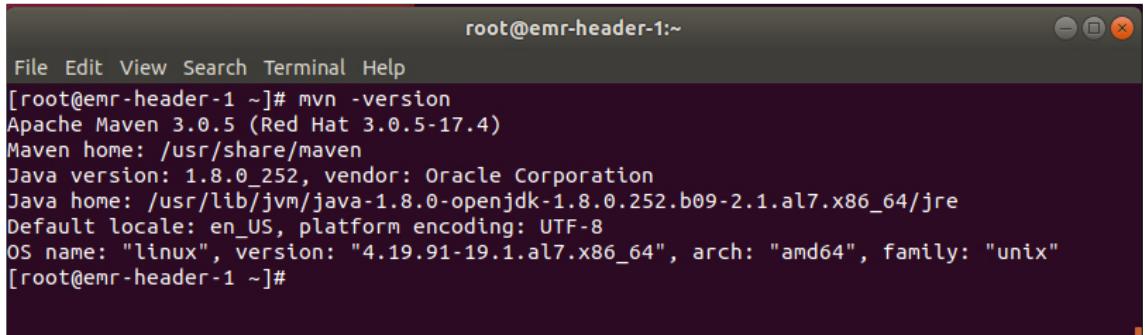
```
# sudo yum install maven
```

Figure 6.1 Verify maven installation

```
# mvn -version
```

HiBench works with python 2, we check if python2 is installed

```
# python2 -V
    Python 2.7.5
```



A screenshot of a terminal window titled "root@emr-header-1:~". The window shows the output of the command "mvn -version". The output displays the Apache Maven version (3.0.5), Java version (1.8.0\_252), Java home ( /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.1.al7.x86\_64/jre ), Default locale (en\_US), platform encoding (UTF-8), OS name (linux), and OS version (4.19.91-19.1.al7.x86\_64). The terminal has a dark theme with white text.

```
root@emr-header-1:~# mvn -version
Apache Maven 3.0.5 (Red Hat 3.0.5-17.4)
Maven home: /usr/share/maven
Java version: 1.8.0_252, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.252.b09-2.1.al7.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.19.91-19.1.al7.x86_64", arch: "amd64", family: "unix"
[root@emr-header-1 ~]#
```

**Figure 6.1: Alibaba Cloud e-MapReduce - Maven version**

Download HiBench 7.1.1

```
# wget https://github.com/Intel-bigdata/HiBench/archive/
HiBench-7.1.tar.gz
```

Untar the downloaded file.

```
# tar -zxf HiBench-7.1.tar.gz
```

Rename the extracted folder to a more user friendly name

```
# mv HiBench-HiBench-7.1 HiBench
```

Navigate to HiBench folder

```
# cd HiBench
```

Build HiBench 7.1 for Hadoop

```
# mvn -Phadoopbench -Dspark=2.4 -Dscala=2.12 clean package
```

Figure 6.2 Upon successful HiBench compilation, an informative success message occurs. During the compilation failures might occur, re-running the above command would mostly fix this issue.

To modify HiBench configuration files, navigate to HiBench's conf folder

```
root@emr-header-1:~/HiBench
File Edit View Search Terminal Help
-----
T E S T S
-----
Results :
Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.3.2:jar (default-jar) @ nutchindexing ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: /root/HiBench/hadoopbench/nutchindexing/target/nutchindexing-7.1.jar
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] hibench ..... SUCCESS [7.094s]
[INFO] hibench-common ..... SUCCESS [3:00.394s]
[INFO] HiBench data generation tools ..... SUCCESS [13.363s]
[INFO] hadoopbench ..... SUCCESS [0.002s]
[INFO] hadoopbench-sql ..... SUCCESS [27.016s]
[INFO] mahout ..... SUCCESS [1:43.141s]
[INFO] PEGASUS: A Peta-Scale Graph Mining System ..... SUCCESS [1.380s]
[INFO] nutchindexing ..... SUCCESS [46.490s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6:19.016s
[INFO] Finished at: Sat Dec 19 16:13:15 CST 2020
[INFO] Final Memory: 45M/1080M
[INFO] -----
[root@emr-header-1 HiBench]#
```

Figure 6.2: Alibaba Cloud e-MapReduce - HiBench success

```
# cd conf/
# cp hadoop.conf.template hadoop.conf
# sudo vi hadoop.conf
```

hibench.hdfs.master value can be found in /usr/lib/hadoop/etc/hadoop/core-site.xml file (fs.default.name). Here we need to specify the hibench.hadoop.examples.test.jar manually for HiBench to run. Otherwise HiBench raises an Assertion error.

Figure 6.3 Editing the hibench configuration file (using vi editor).

```
# sudo vi hadoop.conf
```

```
root@emr-header-1:~/HiBench/conf
File Edit View Search Terminal Help
# Hadoop home
hibench.hadoop.home      /usr/lib/hadoop-current

# The path of hadoop executable
hibench.hadoop.executable    /usr/lib/hadoop-current/bin/hadoop

# Hadoop configuration directory
hibench.hadoop.configure.dir  /etc/ecm/hadoop-conf

# The root HDFS path to store HiBench data
hibench.hdfs.master        hdfs://emr-header-1.cluster-53371:9000

hibench.hadoop.examples.test.jar      /opt/apps/ecm/service/hadoop/2.8.5-1.6.1/package/
hadoop-2.8.5-1.6.1/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.8.5-tests.j
ar

# Hadoop release provider. Supported value: apache, cdh5, hdp
hibench.hadoop.release      apache
```

**Figure 6.3: Alibaba Cloud e-MapReduce - Hadoop configurations**

Figure 6.4 HiBench related configuration settings like data scale and mappers/reducers count are made from the file *Hibench/conf/hibench.conf*. Having 12 cores on worker nodes, we specify 12 mappers and 12 reducers used during benchmark. As the data scale change the changes will be made from this file.

```
$ sudo vi hibench.conf
```

```
root@emr-header-1:~/HiBench/conf
File Edit View Search Terminal Help
# Data scale profile. Available value is tiny, small, large, huge, gigantic and bigdata.
# The definition of these profiles can be found in the workload's conf file i.e. conf/wor
kloads/micro/wordcount.conf
hibench.scale.profile          tiny
# Mapper number in hadoop, partition number in Spark
hibench.default.map.parallelism    12
# Reducer number in hadoop, shuffle partition number in Spark
hibench.default.shuffle.parallelism   12

=====
# Report files
=====
# default report formats
hibench.report.formats        "%-12s %-10s %-8s %-20s %-20s %-20s %-20s\n"

# default report dir path
hibench.report.dir            ${hibench.home}/report

# default report file name
```

**Figure 6.4: Alibaba Cloud e-MapReduce - HiBench configurations**

Figure 6.5 depicts the execution of a HiBench benchmark (Wordcount) and resource utilization capturing process on the worker nodes.

```
hadoop@emr-worker-1:~ hadoop@emr-worker-2:~ hadoop@emr-worker-3:~ root@emr-header-1:~/HiBench
File Edit View S hadoop@emr-worker-2:~
-rw-rw-r-- 1 h  File Edit View Search Terminal Help hadoop@emr-worker-3:~
-rw-rw-r-- 1 h  File Edit View Search Terminal Help hadoop@emr-worker-3:~
-rw-rw-r-- 1 h  File Edit View Search Terminal Help hadoop@emr-worker-3:~
-rw-rw-r-- 1 h  File Edit View Search Terminal Help root@emr-header-1:~/HiBench
root@emr-header-1:~/HiBench
File Edit View Search Terminal Help
rm: 'hdfs://emr-header-1.cluster-53371:9000/HiBench/Sort/Output': No such file or directory
hdfs du -s: /usr/lib/hadoop-current/bin/hadoop --config /etc/ecm/hadoop-conf fs -du -s hdfs://emr-header-1.cluster-53371:9000/HiBench/Sort/Input
Submit MapReduce Job: /usr/lib/hadoop-current/bin/hadoop --config /etc/ecm/hadoop-conf jar /usr/lib/hadoop-current/s
hare/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.5.jar sort -outKey org.apache.hadoop.io.Text -outValue org.apach
e.hadoop.io.Text -r 12 hdfs://emr-header-1.cluster-53371:9000/HiBench/Sort/Input hdfs://emr-header-1.cluster-53371:9
000/HiBench/Sort/Output
The authenticity of host 'emr-worker-3.cluster-53371 (192.168.0.155)' can't be established.
ECDSA key fingerprint is SHA256:Ceh+0URG6txKCcJqWxTYR2BmY5lFPA/q19snL3qQ.
ECDSA key fingerprint is MD5:02:1f:71:42:09:d4:0e:ed:2f:cc:0b:0d:c5:9d:b3.
Are you sure you want to continue connecting (yes/no)? The authenticity of host 'emr-worker-2.cluster-53371 (192.168
.0.154)' can't be established.
ECDSA key fingerprint is SHA256:qeEbddd8BPy+ukBzhwet9v1UR1t2z802UxUbPGHfc50.
ECDSA key fingerprint is MD5:a6:03:a7:6a:0c:id:69:ac:86:db:3d:82:d3:67:7a:2d.
Are you sure you want to continue connecting (yes/no)? The authenticity of host 'emr-worker-1.cluster-53371 (192.168
.0.153)' can't be established.
ECDSA key fingerprint is SHA256:3jP60mxigmqwcroBlibjPSJwRGRVd815pTdh3AZju.
ECDSA key fingerprint is MD5:99:5a:eb:f2:7e:ed:c2:47:f6:46:8c:2a:69:8c:42:0e.
Are you sure you want to continue connecting (yes/no)? 20/12/19 18:19:52 INFO client.RMProxy: Connecting to Resource
r.gz data/
20/12/19 18:21:31 INFO Mapreduce.Job: Map 1% reduce 0%
[hadoop@emr-wor[rw-r--r-- 1 root root 15011 Ara 19 16:49 ali-uc2-w2-t.tar.gz
[hadoop@emr-wor[rw-r--r-- 1 root root 21230 Ara 19 17:15 ali-uc2-w2-s.tar.gz
[hadoop@emr-wor[rw-r--r-- 1 root root 33430 Ara 19 17:35 ali-uc2-w2-l.tar.gz
[hadoop@emr-wor[rw-r--r-- 1 root root 174927 Ara 19 18:12 ali-uc2-w2-h.tar.gz
[hadoop@emr-work[hadoop@emr-worker-3 ~]$ rm data/*
[hadoop@emr-worker-3 ~]$ ls data
[hadoop@emr-worker-3 ~]$ 
[hadoop@emr-worker-3 ~]$ ./datacollector.sh -p ali-uc2-w2-g-srt
```

**Figure 6.5: Alibaba Cloud e-MapReduce - HiBench running**