

Documentation for the thesis:

”Comparative Performance Evaluation of
Hadoop on PaaS Proposals by Leveraging
HiBench”

Uluer Emre Özdił

May 1, 2021

Contents

1	Creating Dataproc Cluster on GCP	3
2	Creating HDInsight Cluster on Azure	16
3	Creating e-MapReduce Cluster on Alibaba Cloud	25
4	Running HiBench on GCP Dataproc	39
5	Running HiBench on Azure HDInsight	49
6	Running HiBench on Alibaba Cloud e-MapReduce	52

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: 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.

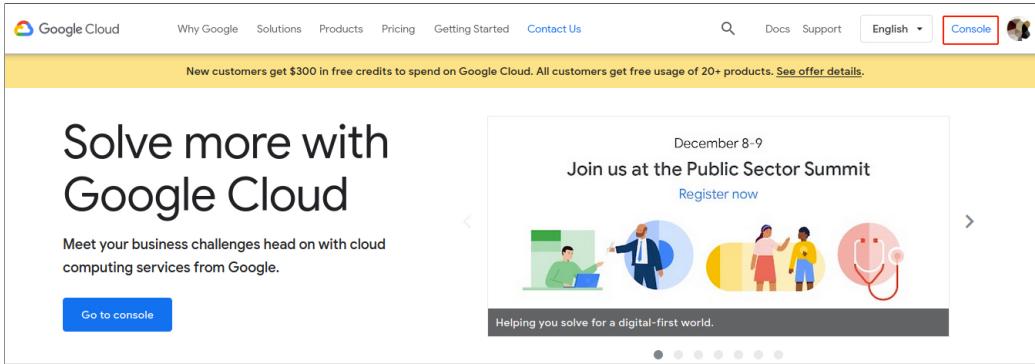


Figure 1: GCP Mainpage

Figure 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.

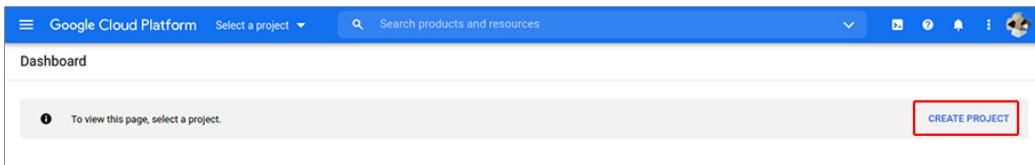


Figure 2: GCP - New Project

Figure 3: Enter project details. At this stage a billing account will be asked to be created as well.

Figure 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 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.

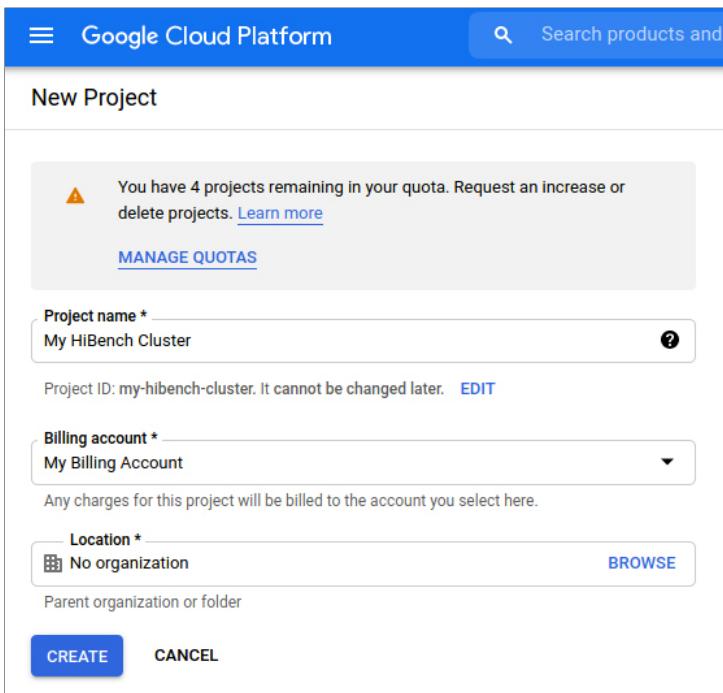


Figure 3: GCP - Create Project

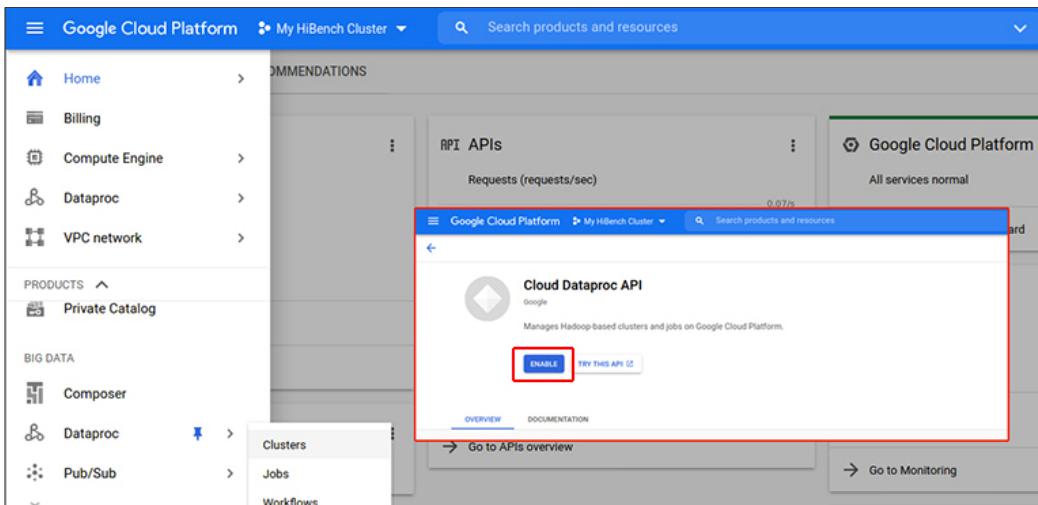


Figure 4: GCP - Enabling Dataproc API

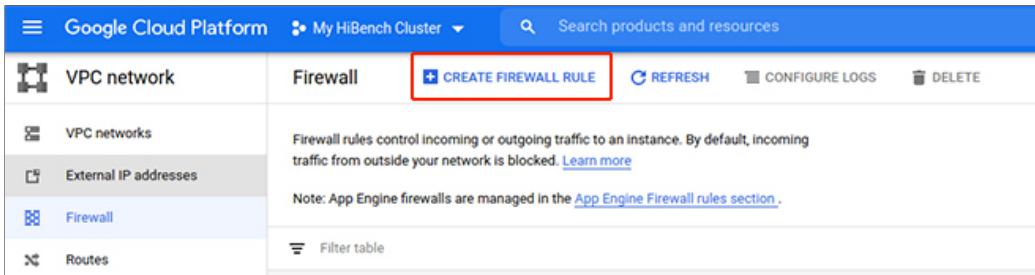


Figure 5: GCP- Create Firewall (Step 1)

Figure 6: Entering following informations:

Name: hadoop-access

Figure 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 8: Clicking on Create button redirects to firewall listings page after successful creation.

Figure 9: To start with the managed Hadoop cluster, click on "Create Cluster"

Figure 10:

Selected configurations:

Setup Cluster Pane

Name: a proper name for the cluster

Location: europe-west3, Frankfurt

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

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

Figure 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 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 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

Google Cloud Platform My HiBench Cluster Search products and resources

VPC network Create a firewall rule

VPC networks External IP addresses Firewall Routes VPC network peering Shared VPC Serverless VPC access Packet mirroring

Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. [Learn more](#)

Name * allow-hadoop Lowercase letters, numbers, hyphens allowed

Description

Logs Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. [Learn more](#)

On Off

Network * default

Priority * 1000 CHECK PRIORITY OF OTHER FIREWALL RULES Priority can be 0 - 65535

Direction of traffic Ingress Egress

Action on match Allow Deny

Figure 6: GCP- Create Firewall (Step 2)

Google Cloud Platform My HiBench Cluster Search products and resources

VPC network <ul style="list-style-type: none"> <input type="checkbox"/> VPC networks <input type="checkbox"/> External IP addresses <input checked="" type="checkbox"/> Firewall <input type="checkbox"/> Routes <input type="checkbox"/> VPC network peering <input type="checkbox"/> Shared VPC <input type="checkbox"/> Serverless VPC access <input type="checkbox"/> Packet mirroring 	<h3>Create a firewall rule</h3> <p>Direction of traffic ?</p> <p><input checked="" type="radio"/> Ingress <input type="radio"/> Egress</p> <p>Action on match ?</p> <p><input checked="" type="radio"/> Allow <input type="radio"/> Deny</p> <p>Targets ? Specified target tags</p> <p>Target tags * ? hadoopaccess X</p> <p>Source filter ? IP ranges</p> <p>Source IP ranges * ? /32 X for example, 0.0.0.0/0, 192.168.2.0/24</p> <p>Second source filter ? None</p> <p>Protocols and ports ?</p> <p><input type="radio"/> Allow all <input checked="" type="radio"/> Specified protocols and ports</p> <p><input checked="" type="checkbox"/> tcp : 8088, 9870</p> <p><input type="checkbox"/> udp : all</p>
--	--

Figure 7: GCP- Create Firewall (Step 3)

Google Cloud Platform My HiBench Cluster Search products and resources

VPC network <ul style="list-style-type: none"> <input type="checkbox"/> VPC networks <input type="checkbox"/> External IP addresses <input checked="" type="checkbox"/> Firewall <input type="checkbox"/> Routes <input type="checkbox"/> VPC network peering <input type="checkbox"/> Shared VPC 	<p>Firewall</p> <p>CREATE FIREWALL RULE REFRESH CONFIGURE LOGS DELETE</p> <p>Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more</p> <p>Note: App Engine firewalls are managed in the App Engine Firewall rules section.</p> <p>Filter table</p> <table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Targets</th><th>Filters</th><th>Protocols / ports</th><th>Action</th><th>Priority</th><th>Network ↑</th><th>Logs</th><th>Hit</th></tr> </thead> <tbody> <tr> <td>allow-hadoop</td><td>Ingress</td><td>hadoopaccess</td><td>IP ranges: X</td><td>tcp:8088,50070</td><td>Allow</td><td>1000</td><td>default</td><td>Off</td><td></td></tr> </tbody> </table>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑	Logs	Hit	allow-hadoop	Ingress	hadoopaccess	IP ranges: X	tcp:8088,50070	Allow	1000	default	Off	
Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑	Logs	Hit												
allow-hadoop	Ingress	hadoopaccess	IP ranges: X	tcp:8088,50070	Allow	1000	default	Off													

Figure 8: GCP- Create Firewall (Step 4)

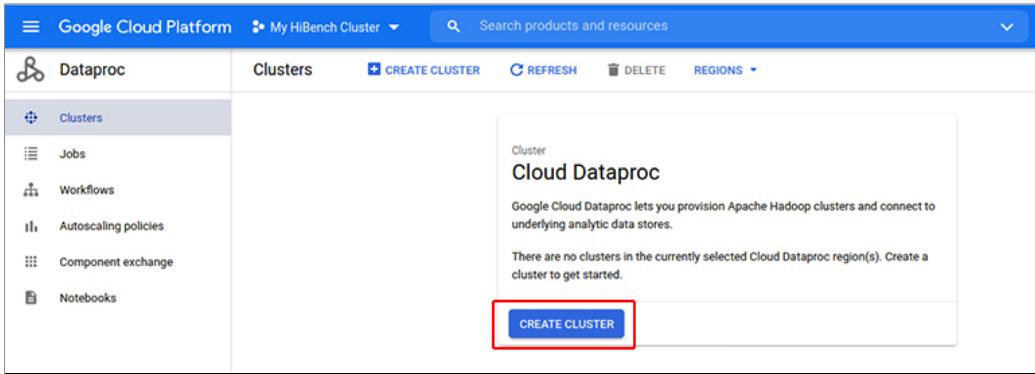


Figure 9: GCP Dataproc - Create Cluster (Step 1)

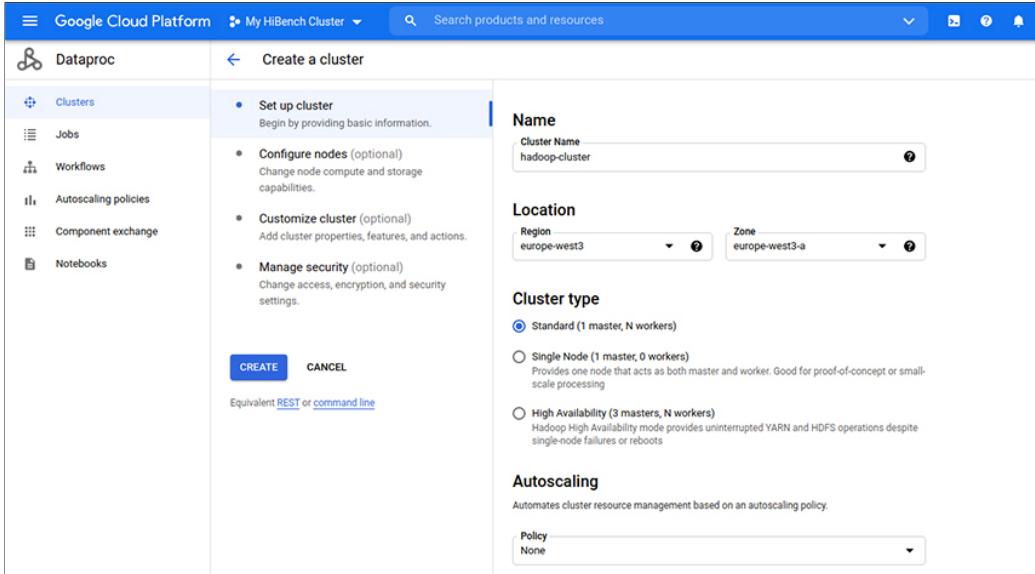


Figure 10: GCP Dataproc - Create Cluster (Step 2)

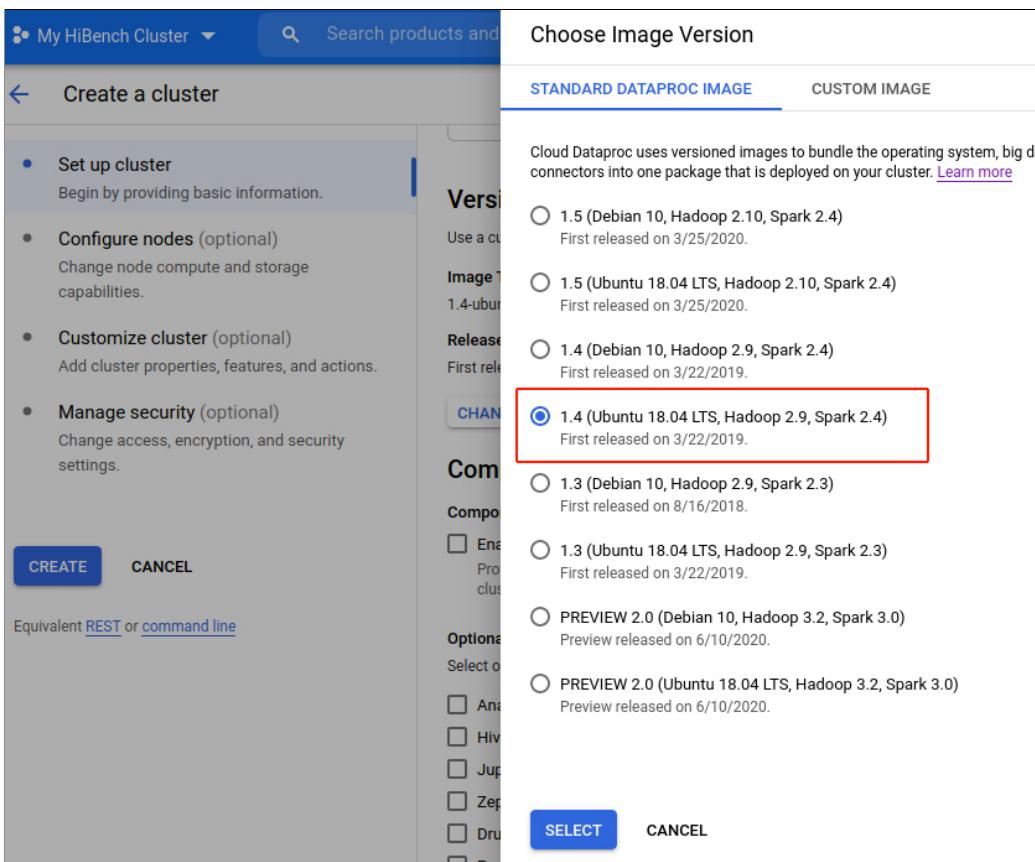


Figure 11: GCP Dataproc - Create Cluster (Step 3)

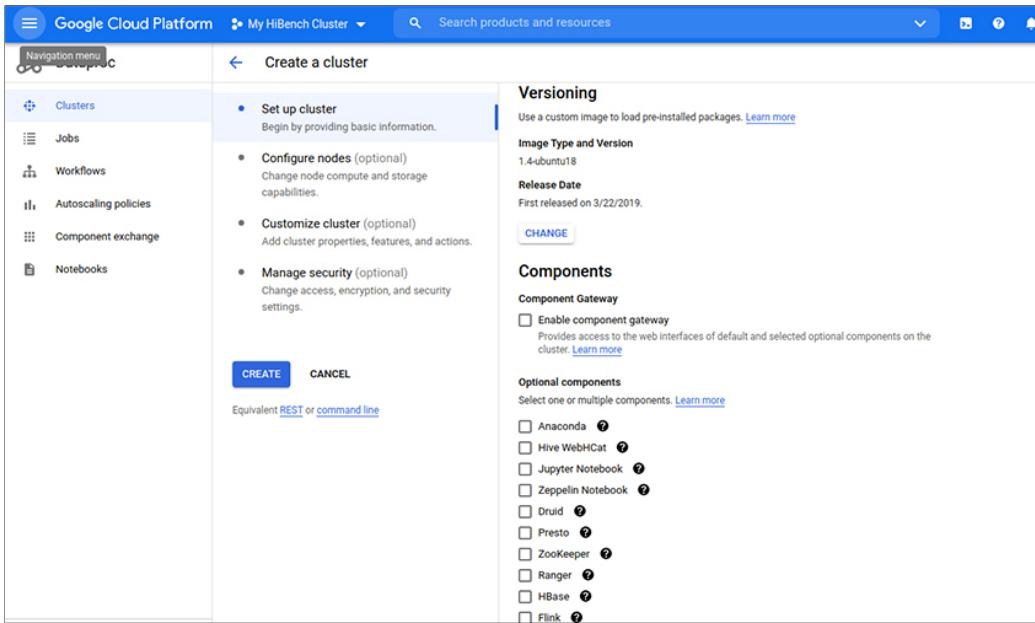


Figure 12: GCP Dataproc - Create Cluster (Step 4)

disk size in its default value of 500 GB.

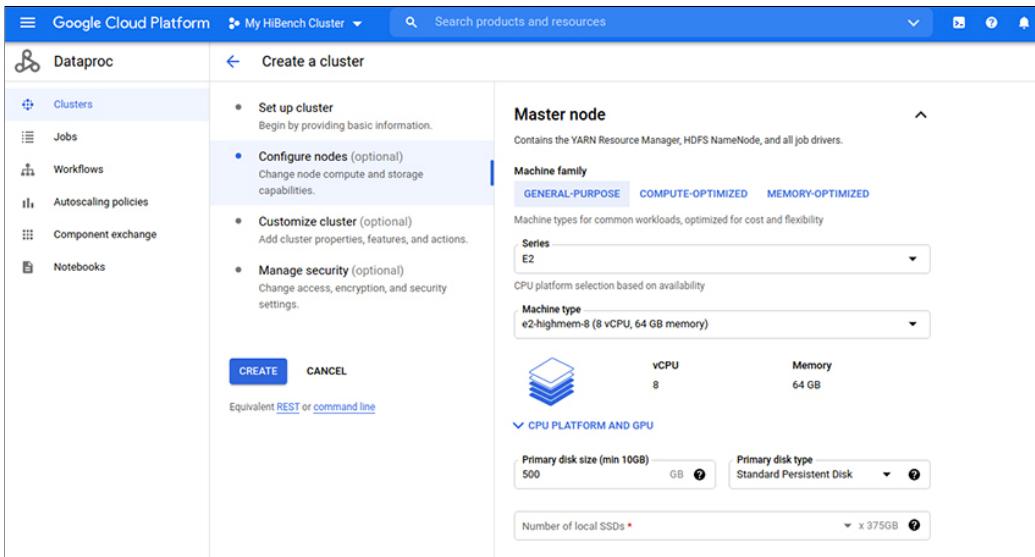


Figure 13: GCP Dataproc - Create Cluster (Step 5)

Figure 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.

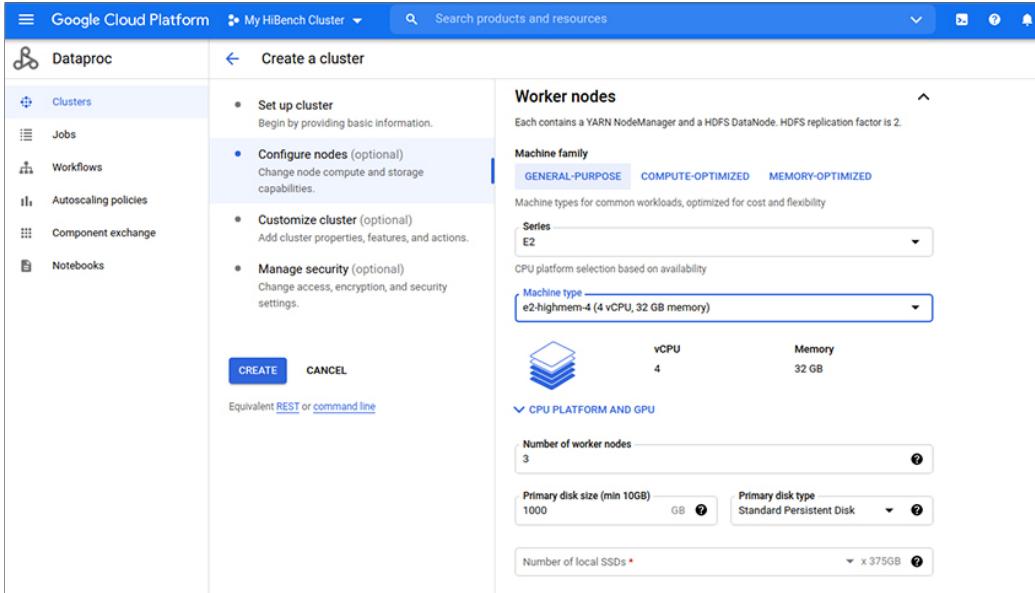


Figure 14: GCP Dataproc - Create Cluster (Step 6)

Figure 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 16: For the network configuration we provide the tag we created during firewall creation for Hadoop.

Figure 17: As final step we check Allow API access, and after reviewing our settings, we click on Create.

Figure 18 and Figure 19: The creation process can be followed from Dataproc and VM dashboards.

Figure 20: Once the installation is completed, by navigating to Compute Engine & VM Instances page we can access master and worker nodes' CLIs by clicking on relevant SSH buttons. Our Dataproc cluster is ready for benchmarking operations.

Figure 21 and Figure 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.

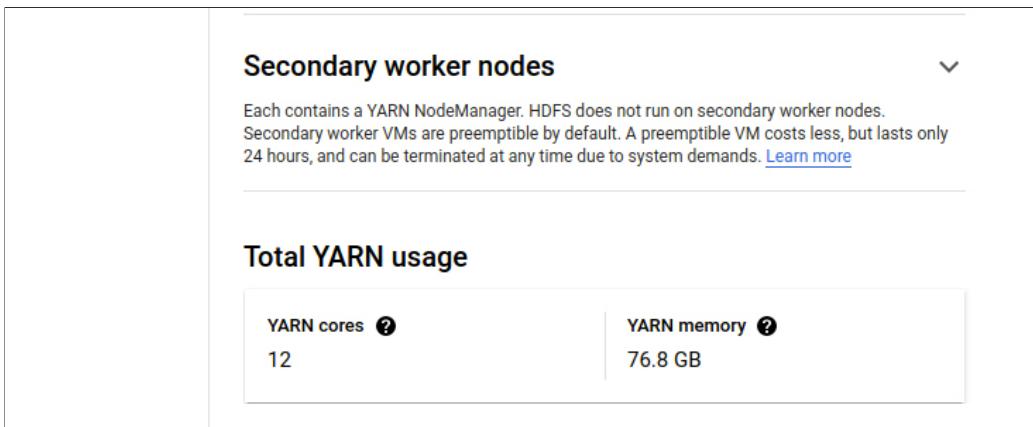


Figure 15: GCP Dataproc - Create Cluster (Step 7)

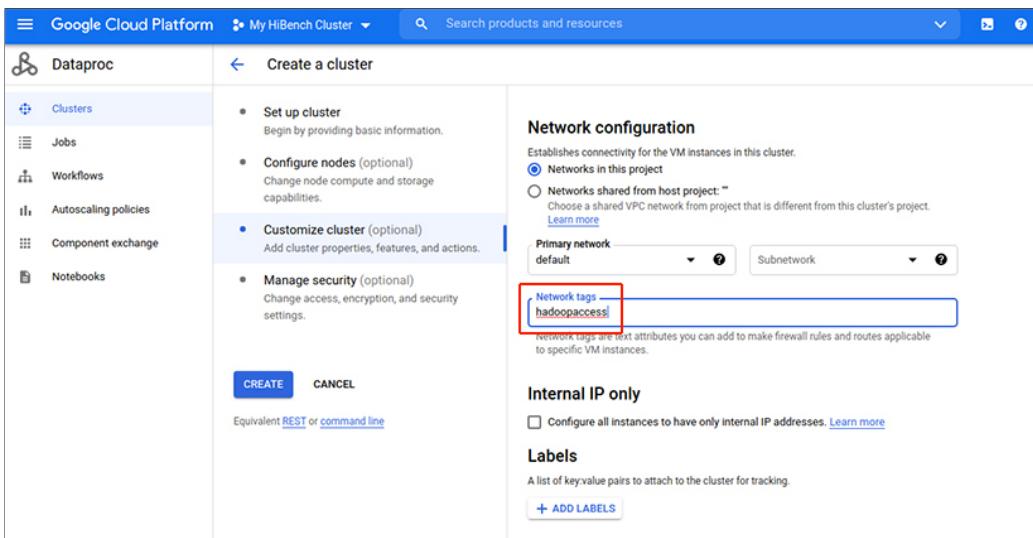


Figure 16: GCP Dataproc - Create Cluster (Step 8)

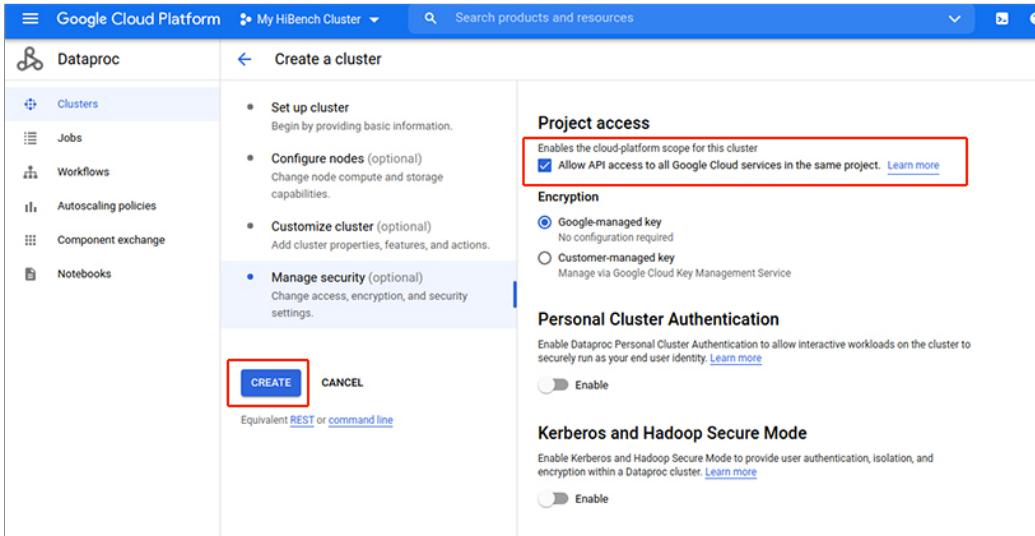


Figure 17: GCP Dataproc - Create Cluster (Step 9)

Clusters							
		CREATE CLUSTER	REFRESH	DELETE	REGIONS		
Clusters							
Jobs							
Workflows							
Autoscaling policies							
Component exchange							
Notebooks							

Figure 18: GCP Dataproc - Create Cluster (Step 10)

VM instances							
		CREATE INSTANCE	REFRESH	DELETE	MANAGE ACCESS	SHOW INFO PANEL	
Virtual machines							
VM instances							
Instance templates							
Sole-tenant nodes							
Machine images							
TPUs							
Migrate for Compute Engi...							

Figure 19: GCP Dataproc - Create Cluster (Step 11)

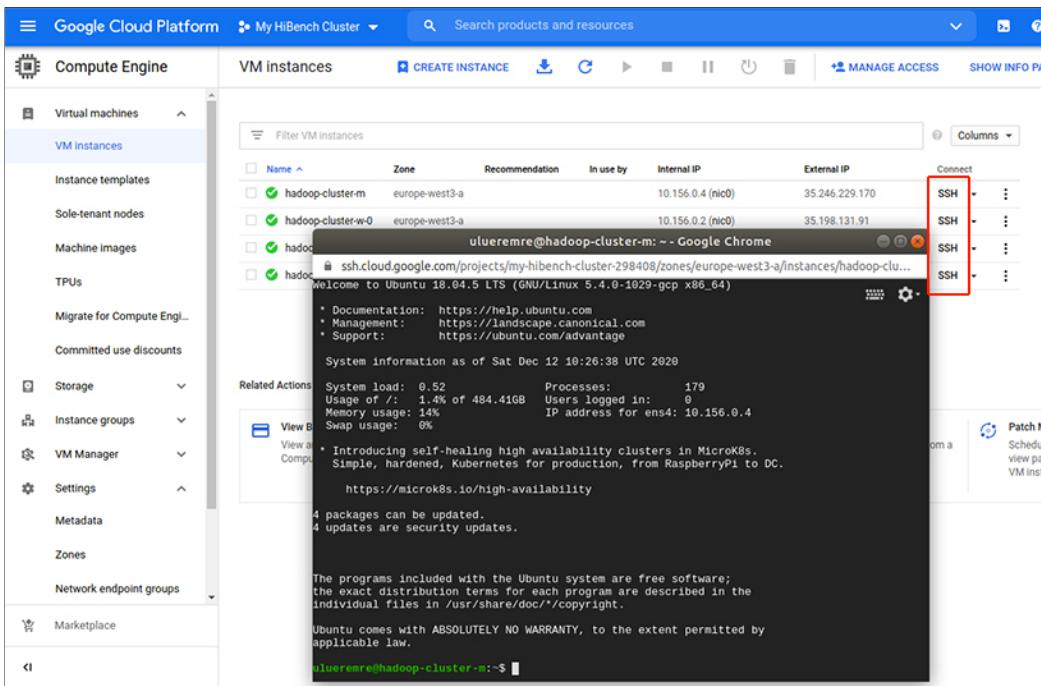


Figure 20: GCP Dataproc - Create Cluster (Step 12)

Overview 'hadoop-cluster-m:8020' (active)

Started:	Sat Dec 12 13:13:16 +0300 2020
Version:	2.9.2, r70f8e666032c815a80625f3b0d5ab0a070fe5d
Compiled:	Sun Oct 18 11:39:00 +0300 2020 by bigtop from (no branch)
Cluster ID:	CID-05ee8590-0640-49f3-8d77-1079006b1421
Block Pool ID:	BP-1972814622-10.156.0.4-1607767989931

Summary

Security is off.
Safemode is off.
1,063 files and directories, 256 blocks = 1,319 total filesystem object(s).
Heap Memory used 360.99 MB of 972.5 MB Heap Memory. Max Heap Memory is 12.5 GB.
Non Heap Memory used 58.44 MB of 59.72 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

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 21: GCP Dataproc - Namenode Manager

All Applications

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	Status	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	ulverence-random-test-writer	MAPREDUCE	default	0	Set Def	12	N/A	RUNNING	UNDEFINED	12	12	73728	0	0	100.0	100.0	ApplicationMaster	0	

Figure 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 23: 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.

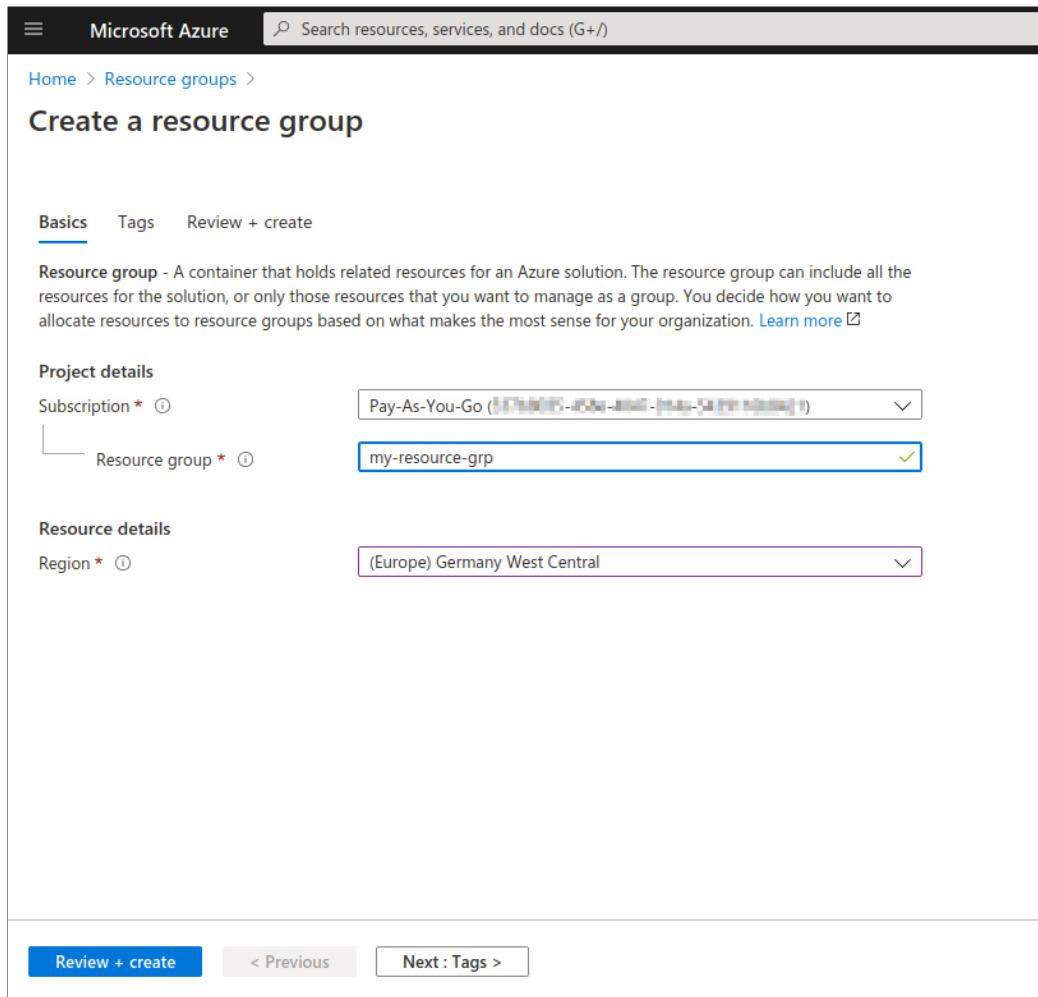


Figure 23: HDInsight setup - Create a resource group

Figure 24: The subscription has to be registered with Microsoft.HDInsight. From the subscription's Resource Provider page we can locate HDInsight and

register it.

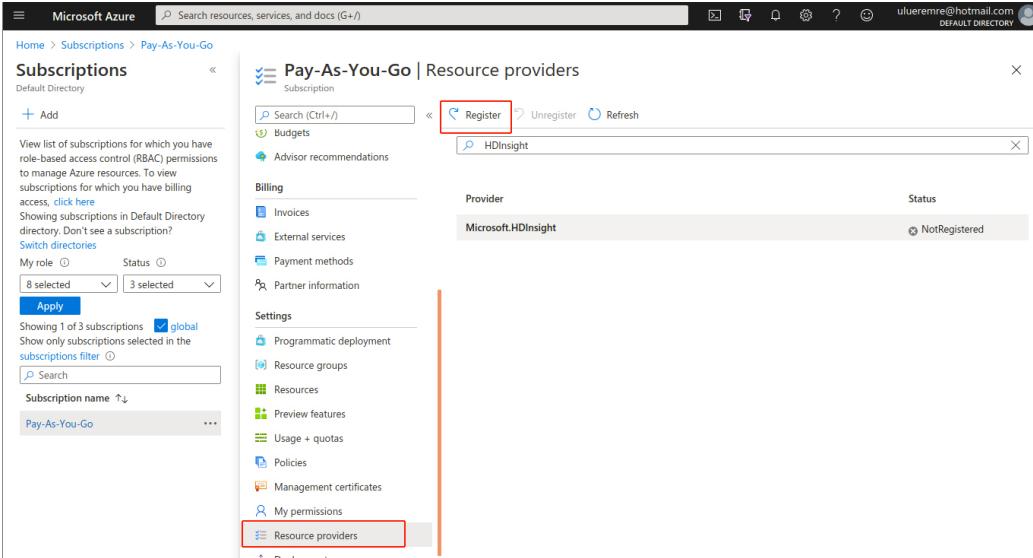


Figure 24: HDInsight setup - Register subscription

Figure 25: 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.

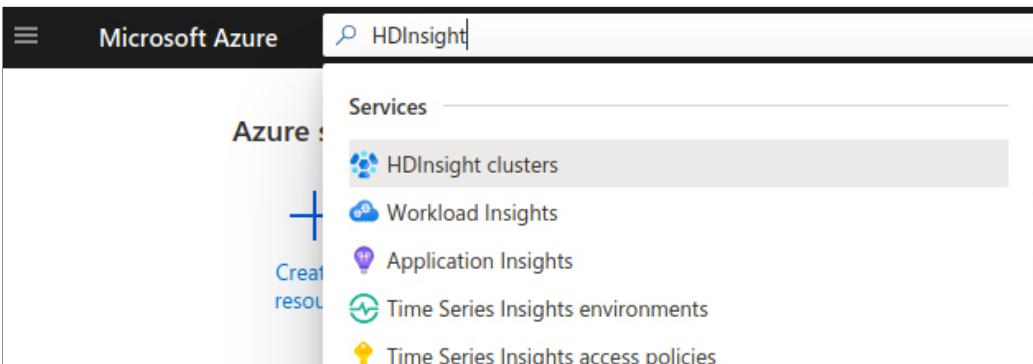


Figure 25: HDInsight setup - Navigate to HDInsight

Figure 26: From the HDI console, click on "Create HDInsight Cluster"

Figure 27: Select subscription and resource group. Give cluster a name and select *Germany West Central* as Region. The cluster type is Hadoop, version

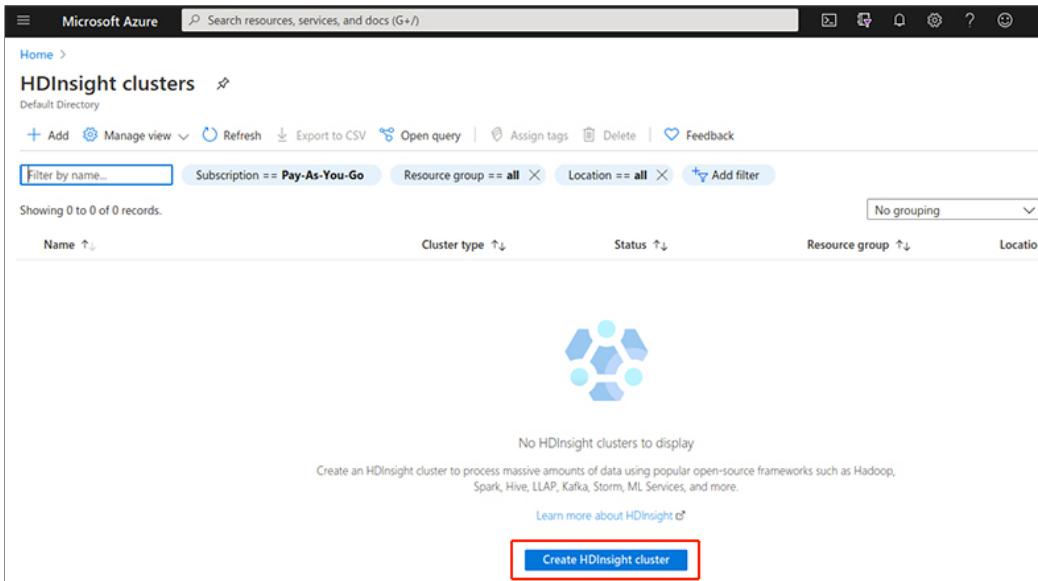


Figure 26: HDInsight setup - Create Cluster

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 28: Leave the storage settings with their defaults.

Figure 29: Step 3: As with storage options, we leave Network settings with their defaults.

Figure 30: 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 31: 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 32 The creation process takes some time and can be tracked from the HDInsight cluster overview page.

Figure 33 After successful installation a message appears.

Figure 34 HDInsight comes with an HDP (Hortonworks Data Platform) cluster.

Microsoft Azure

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) ▾

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 ✓

Region * Germany West Central ▾

Cluster type * Hadoop
[Change](#)

Version * Hadoop 2.7.3 (HDI 3.6) ▾

Cluster credentials

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

Cluster login username * admin

Cluster login password * ✓

Confirm cluster login password * ✓

Secure Shell (SSH) username * sshuser

Use cluster login password for SSH

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

Figure 27: HDInsight setup - Create Cluster

Microsoft Azure Search resources, services

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 [Configure access settings](#)

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

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

Figure 28: HDInsight setup - Storage

Microsoft Azure

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 ○

Network settings

Resource provider connection ○

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 29: HDInsight setup - Networking

The screenshot shows the 'Create HDInsight cluster' configuration page in the Microsoft Azure portal. The top navigation bar includes 'Microsoft Azure' and a search bar. Below the navigation, the breadcrumb trail shows 'Home > HDInsight clusters > Create HDInsight cluster'. The main content area has tabs for 'Basics', 'Storage', 'Security + networking', 'Configuration + pricing' (which is selected), 'Tags', and 'Review + create'. A sub-section titled 'Node configuration' allows setting cluster size and performance, with a note about estimated costs. A callout box provides information about core usage. The 'Node type' section lists three types: Head node, Zookeeper node, and Worker node, each with its respective node size and estimated cost per hour. An option to enable autoscale is available. The total estimated cost per hour is displayed as 16.91 TRY. The 'Script actions' section allows adding custom PowerShell or Bash scripts. At the bottom, there are buttons for 'Review + create', '« Previous', and 'Next: Tags »'.

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

Figure 30: HDInsight setup - Configuration

The screenshot shows the 'Create HDInsight cluster' review step in the Azure portal. At the top, there's a green validation message: 'Validation succeeded.' Below it, the 'Review + create' tab is selected. The page displays the following information:

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-2020-12-24t13-30-30-287z
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)

At the bottom, there are buttons for 'Create', '< Previous', 'Next >', and 'Download a template for automation'.

Figure 31: HDInsight setup - Review

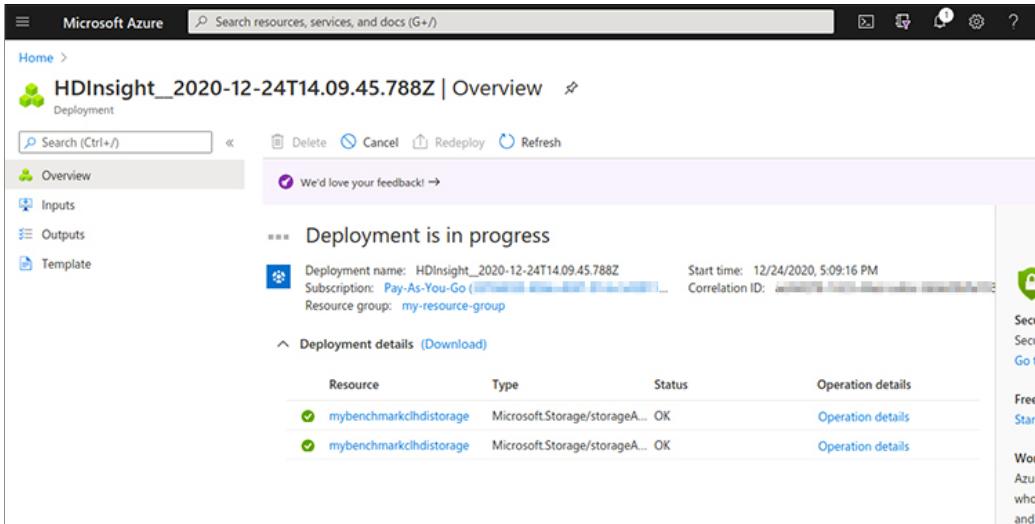


Figure 32: HDInsight setup - In progress

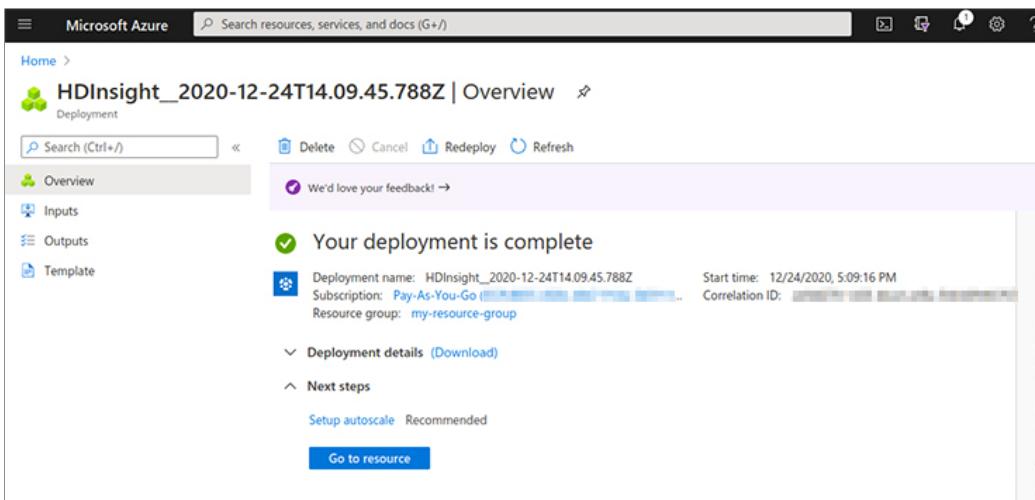


Figure 33: HDInsight setup - Complete

The HDInsight overview page gives access to Ambari web UI.

The screenshot shows the Microsoft Azure portal interface for managing HDInsight clusters. On the left, there's a sidebar with 'HDInsight clusters' and a search bar. The main area is titled 'my-benchmark-cluster' and contains a summary card with the following information:

- Resource group: my-resource-group
- Status: Running
- Location: Germany West Central
- Subscription: Pay-As-You-Go
- Cluster ID: (redacted)
- Tags: (change) Click here to add tags

Below the summary card, there are two buttons: 'Ambari home' and 'Ambari views', with 'Ambari home' being highlighted with a red box. At the bottom, there are 'Overview' and 'Get started' tabs, and a 'Dashboards' section with icons for Ambari home and Ambari views.

Figure 34: HDInsight setup - Overview

Figure 35 Ambari WebUI gives an overview to the current state of the cluster.

Figure 36 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 37 Using the password created on the HDInsight installation process connects the CLI to the master node.

Figure 38 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 39 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.

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>.

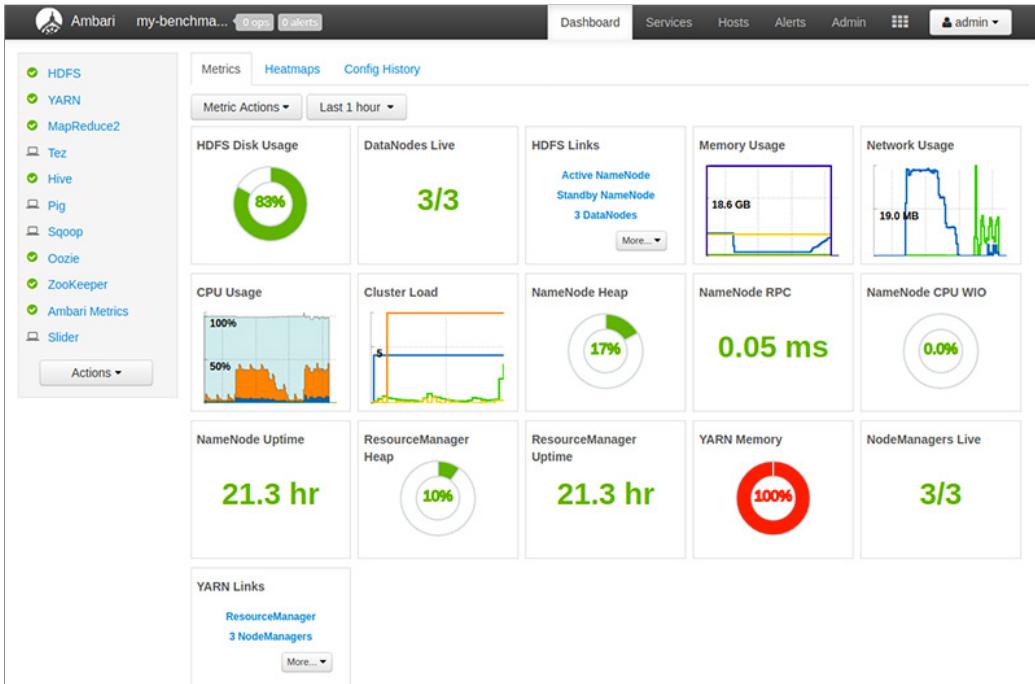


Figure 35: HDInsight - Ambari Dashboard

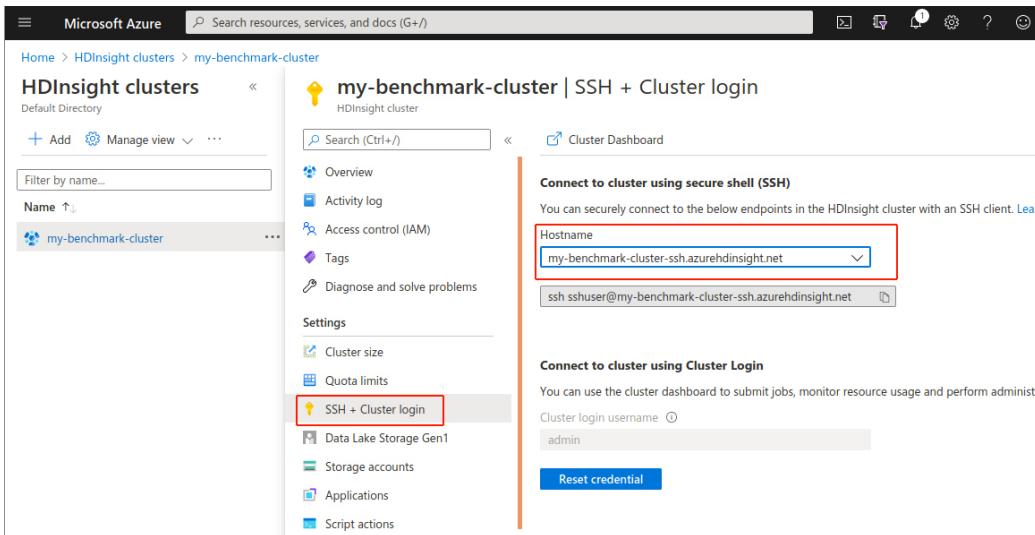
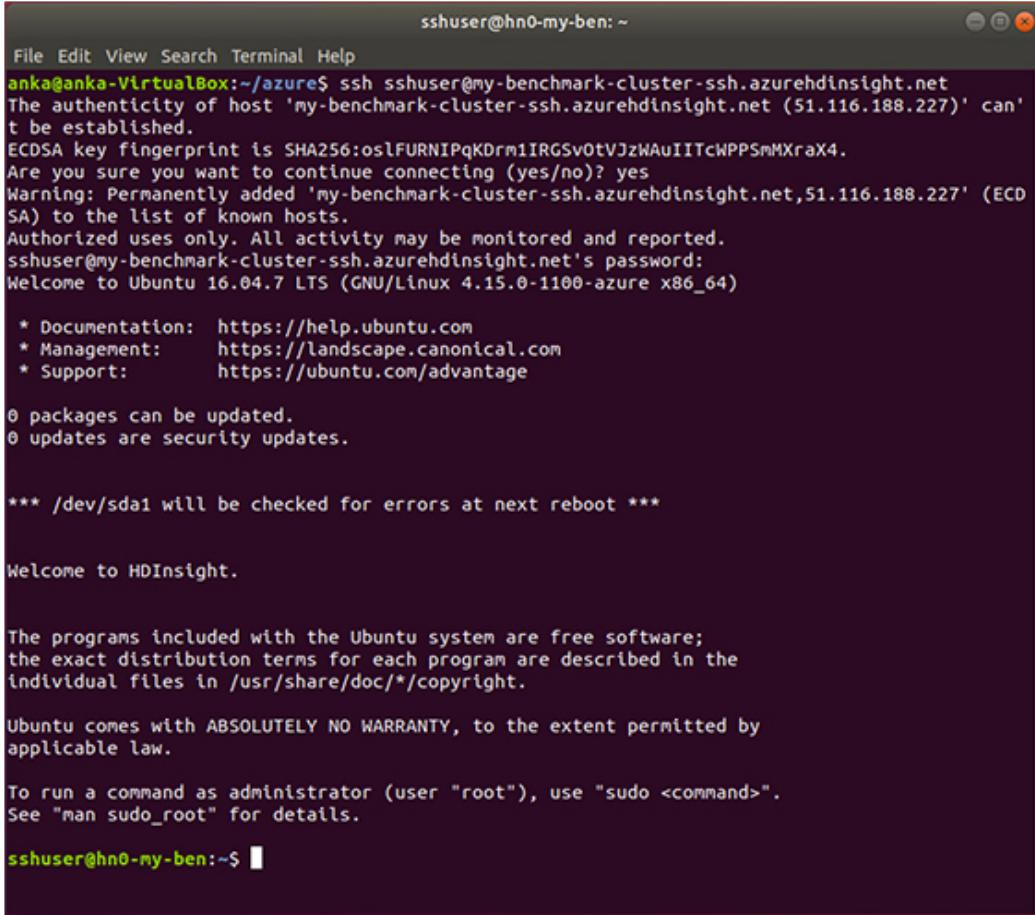


Figure 36: HDInsight - Cluster login



```

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:oslFURNIPqKDrn1IRGSv0tVJzWAuIITcWPPSmMXraX4.
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 uses 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.

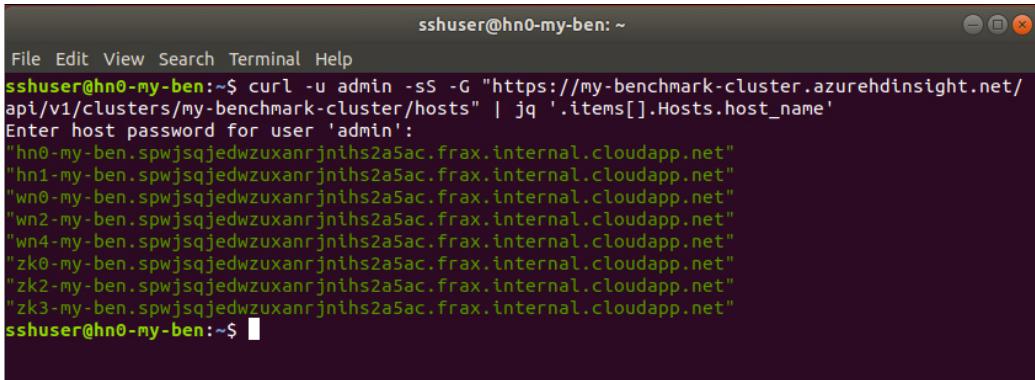
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

sshuser@hn0-my-ben:~$ 

```

Figure 37: 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.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"hn1-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"wn0-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"wn2-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"wn4-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"zk0-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"zk2-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
"zk3-my-ben.spwjsqjedwzuxanrnths2a5ac.frax.internal.cloudapp.net"
sshuser@hn0-my-ben:~$ 

```

Figure 38: HDInsight - Listing worker nodes

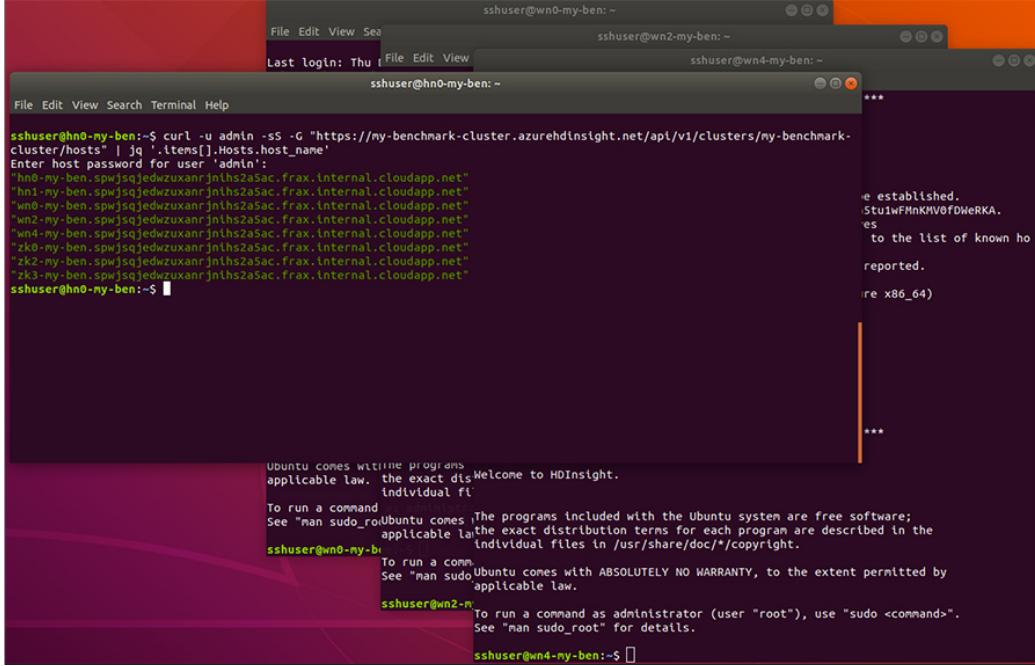


Figure 39: HDInsight - Ready to benchmark

Figure 40 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 41, 42, 43 For e-MapReduce to run, OSS (Object Storage Service) also needs to be activated.

Figure 44 In OSS, setup a bucket for e-MapReduce to store logs.

Figure 45 Alibaba leverages OSS to store access logs. Within the created bucket's settings enable logging access to the bucket.

Figure 46 Before starting with e-MapReduce, select *Germany, Frankfurt* as region, where cluster will be created.

Figure 47 From Alibaba Cloud dashboard, select e-MapReduce.

Figure 48 From Alibaba Cloud EMR console, click on Cluster Wizard to start cluster installation.

Figure 49 *EMR-3.32.0* is the image version supporting Hadoop 2, and subject to selection.

Figure 50 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 required in our case, so we leave

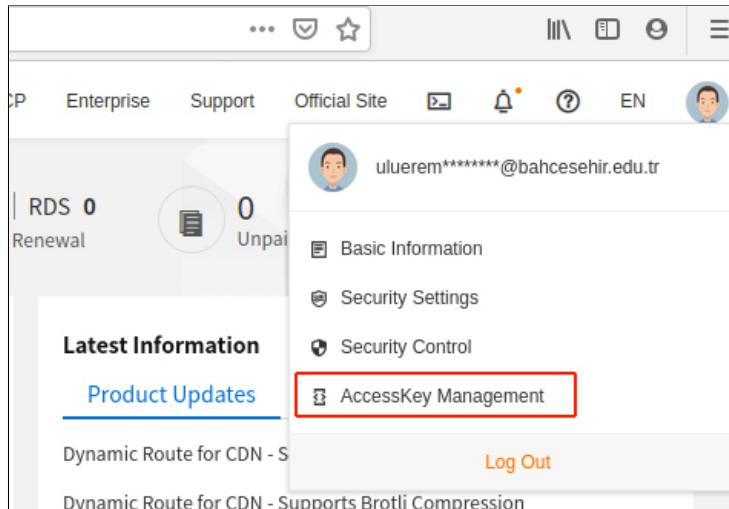


Figure 40: e-MapReduce - Create access key

it unchecked as its default.

Figure 51 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 52 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 53 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 54 Review settings. On the bottom of the page activate checkbox to accept terms and click on *Create*.

Figure 55 The creation process is displayed on e-MapReduce cluster page.

Figure 56 Within Alibaba Cloud's *Elastic Compute Service, Instances* page, single VMs of the cluster are also available.

Figure 57 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.

The screenshot shows the Alibaba Cloud homepage. The top navigation bar includes the Alibaba Cloud logo, a search bar, and links for Expenses, Tickets, ICP, Enterprise, Support, and Help Center. The main content area features a sidebar with 'Products and Services' and icons for Object Storage Service, E-MapReduce, and Elastic Compute Service. The main panel has a search bar labeled 'Enter a keyword'. Below it, a grid of service cards includes: ADB for MySQL, AliCloudDB for OceanBase (Application Services), Advanced Database & Application Migr... (Application Services), ApsaraDB for HBase (Message Notification), ApsaraDB for Cassandra (API Gateway), Database Autonomy Service (Log Service), Database Gateway (Direct Mail), ApsaraDB for MyBase (Blockchain as a Service), LedgerDB (Enterprise Email), Storage & CDN (Middleware), Object Storage Service (Enterprise Distribution), and Table Store (Application Configuration). The 'Object Storage Service' card is highlighted with a red border.

Figure 41: e-MapReduce - Activate OSS (Step 1)

The screenshot shows a page titled 'Activate OSS'. It features a large message: 'OSS has not been activated. Please activate OSS.' Below this, a note states: 'Alibaba Cloud OSS supports two billing methods: Pay-As-You-Go and subscription. Learn more.' At the bottom are two buttons: 'Activate Now' (in blue) and 'Product Details'.

Figure 42: e-MapReduce - Activate OSS (Step 2)

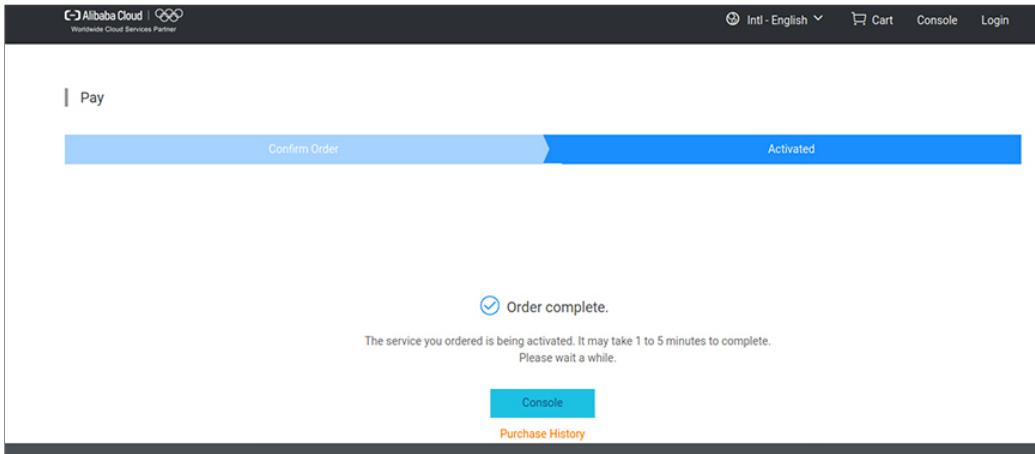


Figure 43: e-MapReduce - Activate OSS (Step 3)

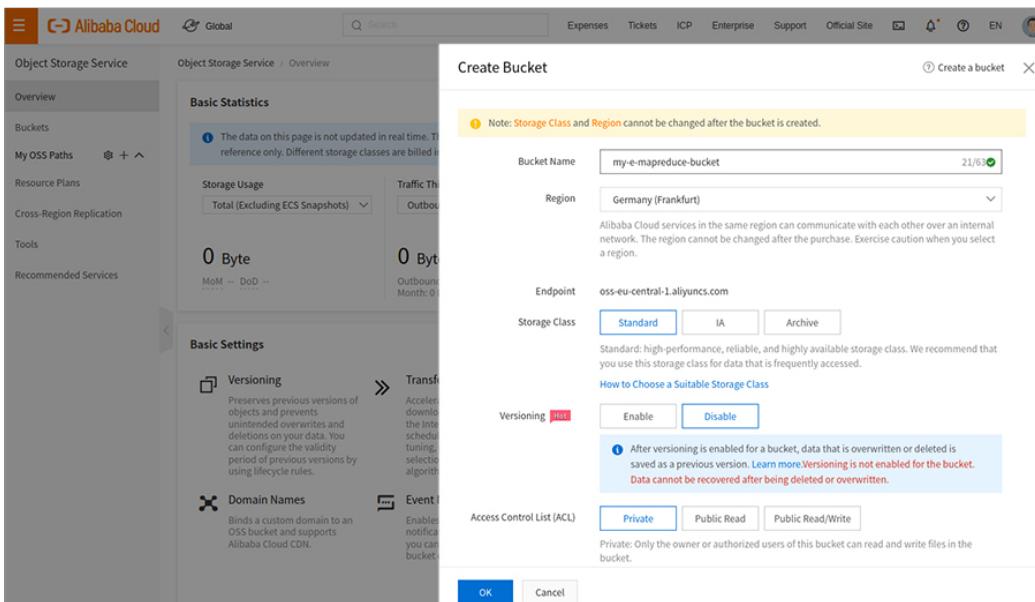


Figure 44: e-MapReduce - Create OSS bucket

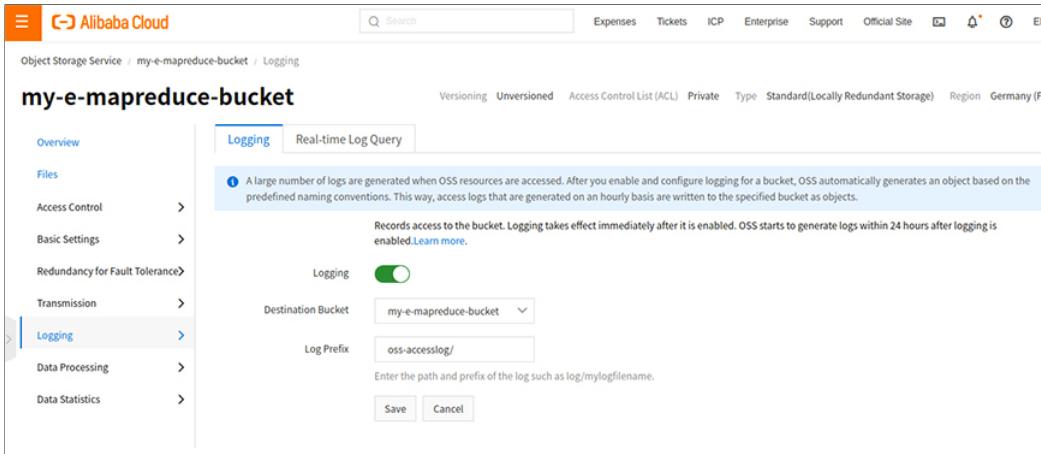


Figure 45: e-MapReduce - Enabling access logging on OSS

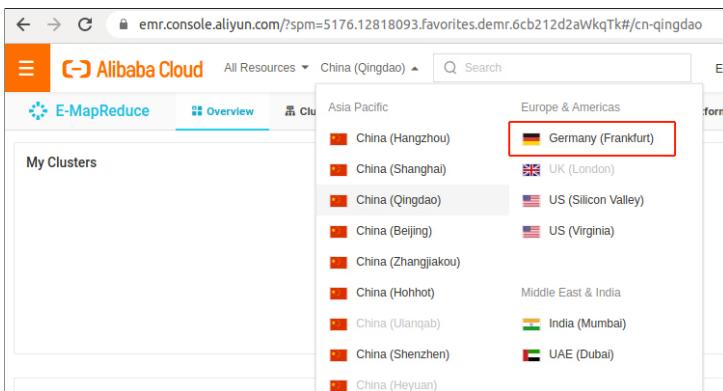


Figure 46: e-MapReduce - Select Region

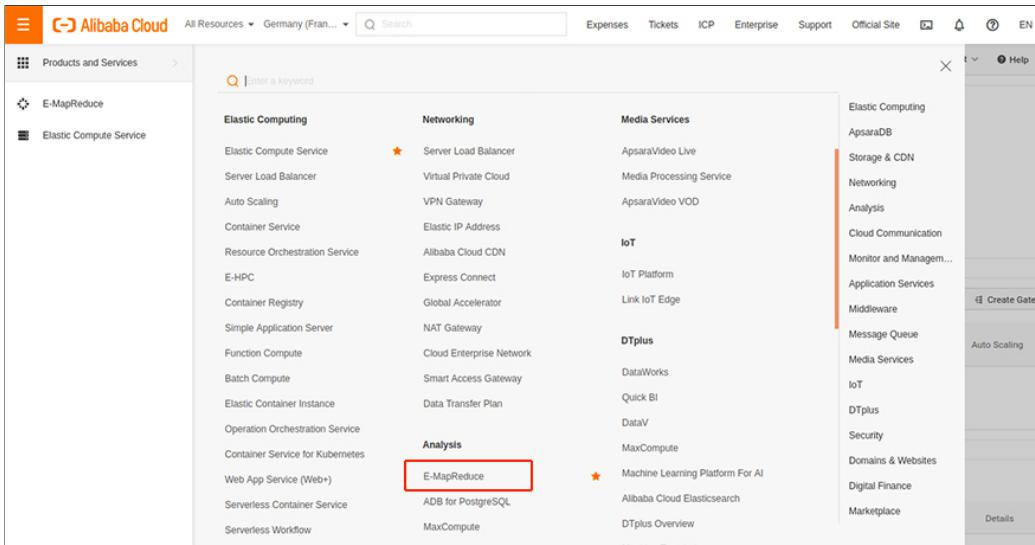


Figure 47: e-MapReduce - Create cluster (Step 1)

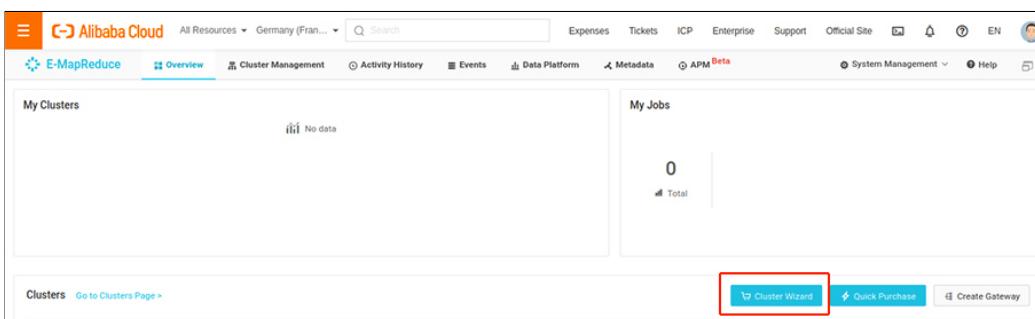


Figure 48: e-MapReduce - Create cluster (Step 2)

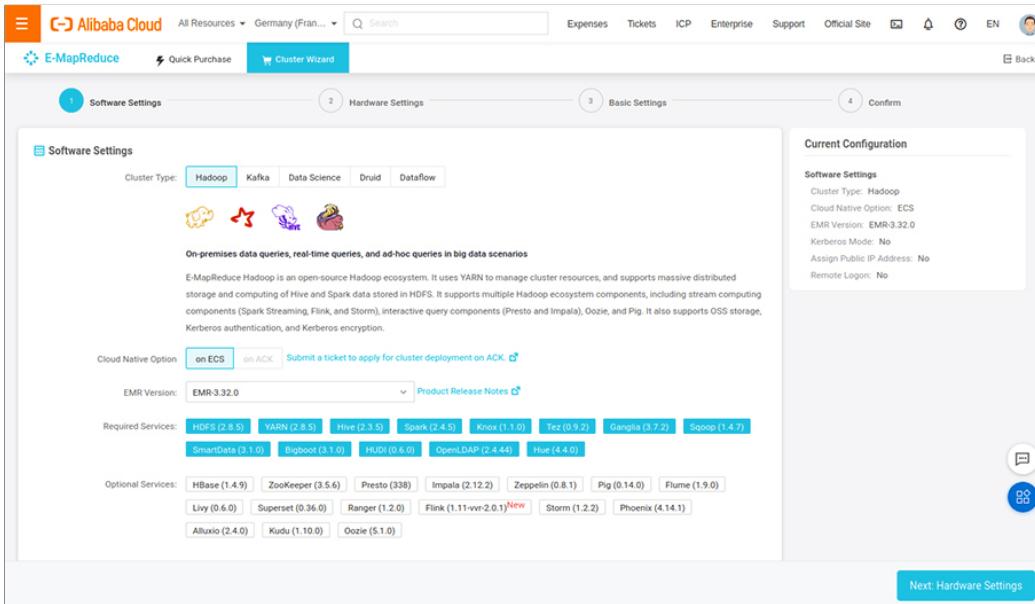


Figure 49: e-MapReduce - Create cluster (Step 3)

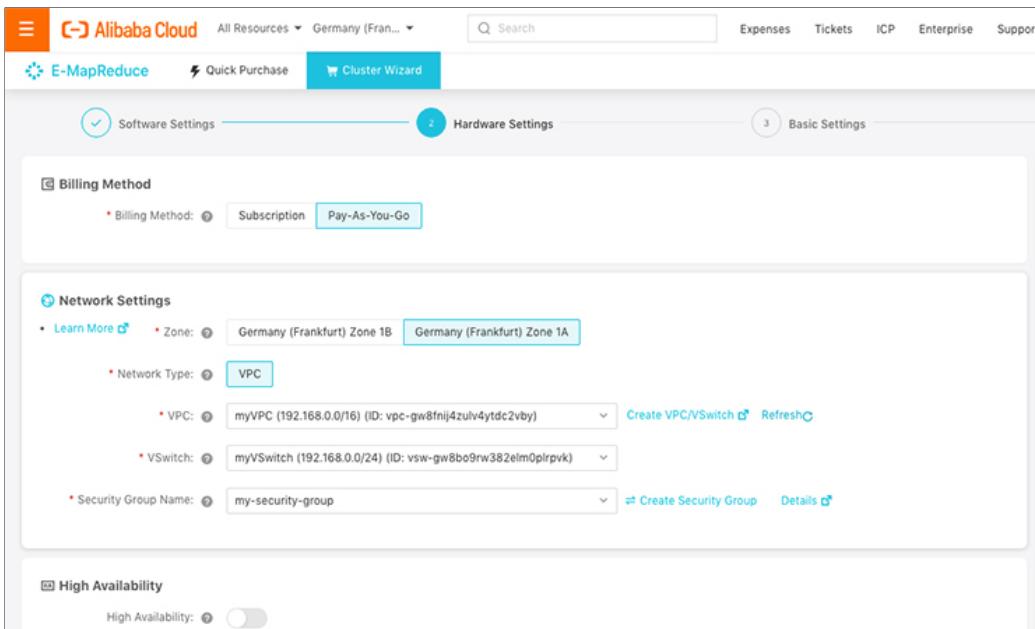


Figure 50: e-MapReduce - Create cluster (Step 4)

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 the tabs, there are three main categories: 'Master Instance', 'Core Instance', and 'Task Instance'. The 'Master Instance' tab is selected. Under the 'Master Instance' tab, there are six filter buttons: '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.

Instance Type	Region	vCore	vMem	Band Width
Memory Optimized ecs.r6e.xlarge	ecs.r6e.xlarge	4 vCPU	32 GiB	2.048 Gbps
Memory Optimized ecs.r6e.2xlarge	ecs.r6e.2xlarge	8 vCPU	64 GiB	3.072 Gbps
Memory Optimized ecs.r6e.4xlarge	ecs.r6e.4xlarge	16 vCPU	128 GiB	6.144 Gbps
Memory Optimized ecs.r6e.8xlarge	ecs.r6e.8xlarge	32 vCPU	256 GiB	10.240 Gbps
Memory Optimized ecs.r6e.13xlarge	ecs.r6e.13xlarge	52 vCPU	384 GiB	16.384 Gbps
Memory Optimized ecs.r6	ecs.r6.xlarge	4 vCPU	32 GiB	1.536 Gbps
Memory Optimized ecs.r6.2xlarge	ecs.r6.2xlarge	8 vCPU	64 GiB	2.560 Gbps
Memory Optimized ecs.r6.3xlarge	ecs.r6.3xlarge	12 vCPU	96 GiB	4.096 Gbps
Memory Optimized ecs.r6.4xlarge	ecs.r6.4xlarge	16 vCPU	128 GiB	5.120 Gbps
Memory Optimized ecs.r6.6xlarge	ecs.r6.6xlarge	24 vCPU	192 GiB	7.680 Gbps
Memory Optimized ecs.r6.8xlarge	ecs.r6.8xlarge	32 vCPU	256 GiB	10.240 Gbps
Memory Optimized ecs.r6.13xlarge	ecs.r6.13xlarge	52 vCPU	384 GiB	12.800 Gbps
Memory Optimized ecs.se1	ecs.se1.xlarge	4 vCPU	32 GiB	1.536 Gbps
Memory Optimized ecs.se1.2xlarge	ecs.se1.2xlarge	8 vCPU	64 GiB	2.048 Gbps
Memory Optimized ecs.se1.4xlarge	ecs.se1.4xlarge	16 vCPU	128 GiB	3.072 Gbps
Memory Optimized ecs.se1.8xlarge	ecs.se1.8xlarge	32 vCPU	256 GiB	6.144 Gbps

Below the table, it says 'Current Master' is set to 'ecs.se1.2xlarge' and 'Node Type' is selected. There are two sections for disk configuration: 'System Disk Type' (SSD selected) with a size of '120 GB * 1 Disks (Capacity Range: 40 ~ 500 GB) IOPS 5400' and 'Data Disk Type' (SSD selected) with a size of '500 GB * 1 Disks (Capacity Range: 40 ~ 32768 GB) IOPS 16800'. At the bottom, it shows 'Master Nodes: 1 Nodes'.

Figure 51: e-MapReduce - Create cluster (Step 5)

The screenshot shows the Alibaba Cloud E-MapReduce Cluster Wizard interface. At the top, there are tabs for 'E-MapReduce', 'Quick Purchase', and 'Cluster Wizard' (which is highlighted). Below the tabs, there are three main categories: 'Master Instance', 'Core Instance' (which is selected), and 'Task Instance'. Under 'Core Instance', there are several tabs: 'General Purpose', 'Compute Optimized', 'Memory Optimized' (which is selected and highlighted in blue), 'Big Data', 'Local SSD', 'High Clock Speed', 'Entry-Level (Shared)', and 'GPU'. The main content area displays a list of available instance types under the 'Memory Optimized' tab. The selected instance is 'Memory Optimized ecs.se1 ecs.se1.xlarge'.

	Instance Type	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 checked="" type="radio"/>	Memory Optimized ecs.se1 ecs.se1.xlarge	4 vCPU	32 GiB	0.819 Gbps
<input 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

Current Core Node: ecs.se1.xlarge
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 52: e-MapReduce - Create cluster (Step 6)

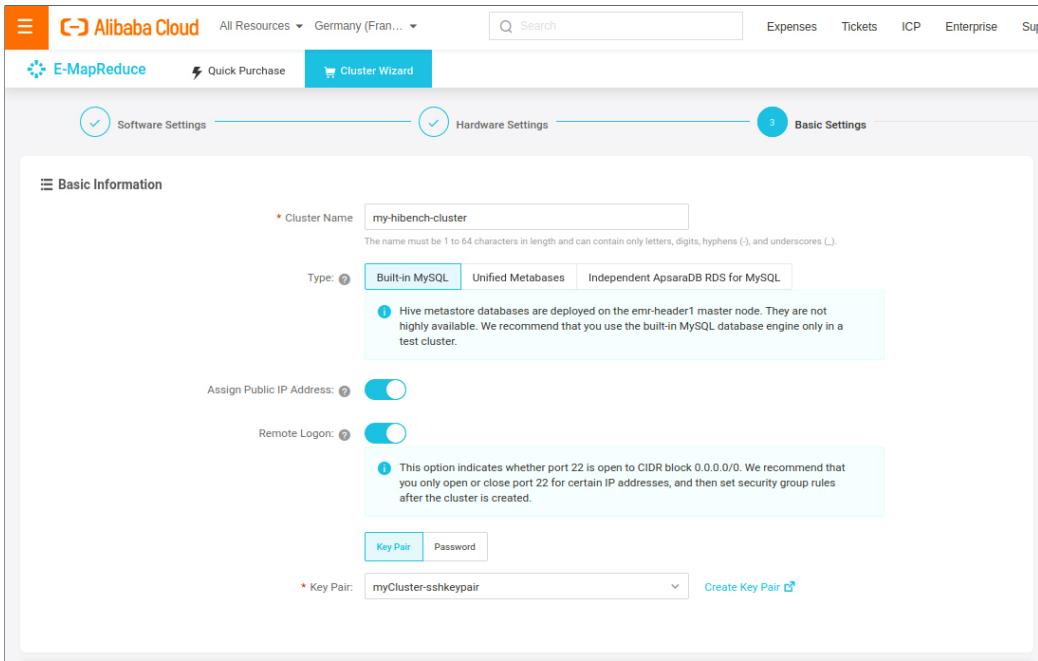


Figure 53: e-MapReduce - Create cluster (Step 7)

```
$ 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 58 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 59 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.

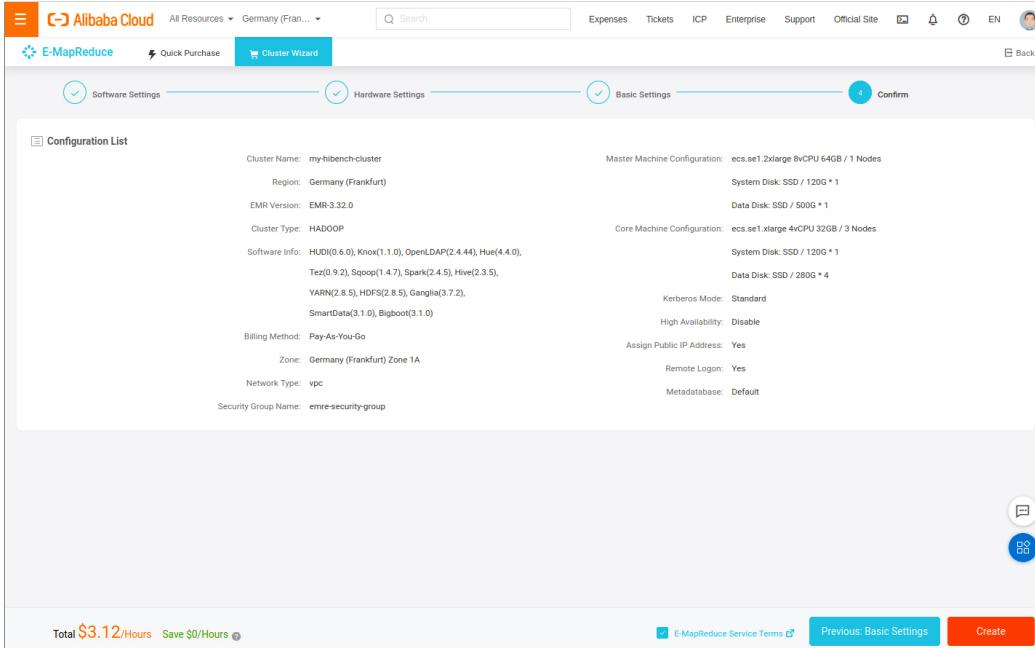


Figure 54: e-MapReduce - Create cluster (Step 8)

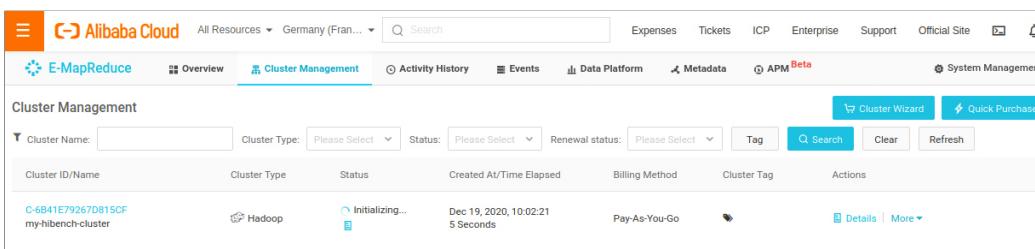


Figure 55: e-MapReduce - Create cluster (Step 9)

Figure 56: e-MapReduce - Create cluster (Step 10)

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

Figure 60 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.

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
```

```
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+fGwjq5ydBxLEEWdNsNY8q07YiMb9oXRMA70zCZLc.  
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 57: 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@172.31.19.104  
Last login: Sat Dec 19 04:43:26 2020  
  
Welcome to Alibaba Cloud Elastic Compute Service !  
  
[root@emr-header-1 ~]# su hadoop  
[hadoop@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 58: e-MapReduce - Connecting to worker nodes

```
GNU nano 2.9.3 /home/anka/.ssh/config  
Host *  
  ServerAliveInterval 30  
  ServerAliveCountMax 5  
█
```

Figure 59: e-MapReduce - Handling broken pipe error

```

hadoop@emr-worker-1:~                                                 hadoop@emr-worker-2:~                                                 hadoop@emr-worker-3:~                                                 
208 [anka@anka-Virtua] File Edit View Terminal Help             208 [anka@anka-Virtua] File Edit View Search Terminal Help             208 [anka@anka-Virtua] File Edit View Search Terminal Help
Last login: Sat Dec 19 15:41:22 2020                               Last login: Sat Dec 19 15:41:22 2020                               Last login: Sat Dec 19 15:41:22 2020
Welcome to Alibaba Cloud Elastic Compute Service !               Welcome to Alibaba Cloud Elastic Compute Service !               Welcome to Alibaba Cloud Elastic Compute Service !
[root@emr-header-1 ~]# su hadoop                                [root@emr-header-1 ~]# ssh emr-worker-3                      [root@emr-header-1 ~]# ssh emr-worker-3
Are you sure you want to continue? (y/n) y                     Host key verification successful.                               Host key verification successful.
[root@emr-header-1 ~]# ssh emr-worker-3                      [root@emr-header-1 ~]# ssh emr-worker-3
[root@emr-header-1 ~]# ssh emr-worker-3                      [root@emr-header-1 ~]# ssh emr-worker-3
Last login: Sat Dec 19 15:11:20 2020                               Last login: Sat Dec 19 15:11:20 2020                               Last login: Sat Dec 19 15:11:20 2020
The authenticity of host 'emr-worker-3 (105.236.144.12)' can't be established.
ECDSA key fingerprint: SHA256:KJLjwvZCQDfXWzqkPQHdVnGJLc=.
Are you sure you want to continue? (y/n) y                     Welcome to Alibaba Cloud Elastic Compute Service !               Welcome to Alibaba Cloud Elastic Compute Service !
[root@emr-worker-3 ~]# 
[root@emr-worker-3 ~]# 

root@emr-header-1:~                                                 
File Edit View Search Terminal Help
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

[root@emr-header-1 ~]#

```

Figure 60: e-MapReduce - Ready to benchmark

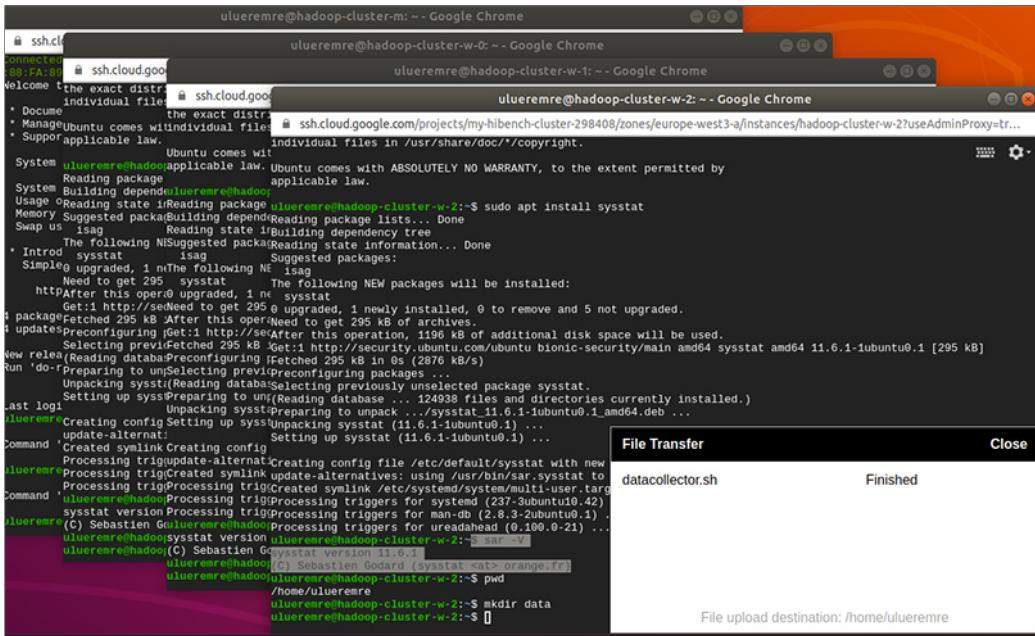


Figure 61: HiBench Dataproc - Uploading datacollector.sh

Figure 61 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 62 Verify maven installation.

```
$ mvn --version
```

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

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

Download HiBench 7.1.1

```
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 62: HiBench Dataproc - Verify Maven installation

```
$ 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 63 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 configurtions.

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

```
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]
```

Figure 63: HiBench Dataproc - Success compiled

```

$ 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.hadoop.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 64 depicts specified Hadoop settings for HiBench.

Figure 65 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.

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.

```

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 configraution 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

```

File menu: **Get Help**, **Write Out**, **Where Is**, **Cut Text**, **Justify**, **Cur Pos**, **Undo**
 Edit menu: **Exit**, **Read File**, **Replace**, **Uncut Text**, **To Spell**, **Go To Line**, **Redo**

Figure 64: HiBench Dataproc - Hadoop configurations

```

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 65: HiBench Dataproc - HiBench configurations

- 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 on 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 66 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.

```
ulueremre@hadoop-cluster-w-0: ~ - Google Chrome
ssh.cloud.google.com/projects/my-hibench-cluster-298408/zones/europe-west3-a/instances/hadoop-cluster-w-0?useAdminProxy=true&authType=OAuth2
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=true&authType=OAuth2
Bytes Written: 320000000000
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
workloads/               workloads/micro/    workloads/        workloads/storm/
graph micro_m sql streaming websearch
ulueremre@hadoop-cluster-m: ~/HiBench$ ls conf/workloads/micro/
dfsio.com_repartition.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 HadoopTerasort bench
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: /usr/lib/hadoop/bin/hadoop --config /usr/lib/hadoop/etc/hadoop fs -du -s hdfs://hadoop-cluster-m/HiBench/Terasort/Input
Submit MapReduce Job: /usr/lib/hadoop/bin/hadoop --config /usr/lib/hadoop/etc/hadoop jar /usr/lib/hadoop//hadoop-mapreduce/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-w-0: ~ $ ./datacollector.sh -p gcp-uci-w0-g-tera
-rw-rw-r-- 1 ulueremre ulueremre 0 Dec 12 12:39 gcp-uci-w1-g-tera-c7.dat
ulueremre@hadoop-cluster-w-1: ~ $ ./datacollector.sh -p gcp-uci-w1-g-tera
ulueremre@hadoop-cluster-w-2: ~ $ ./datacollector.sh -p gcp-uci-w2-g-tera
```

Figure 66: 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 67 taken during benchmark execution depicts the process on master node and worker nodes.

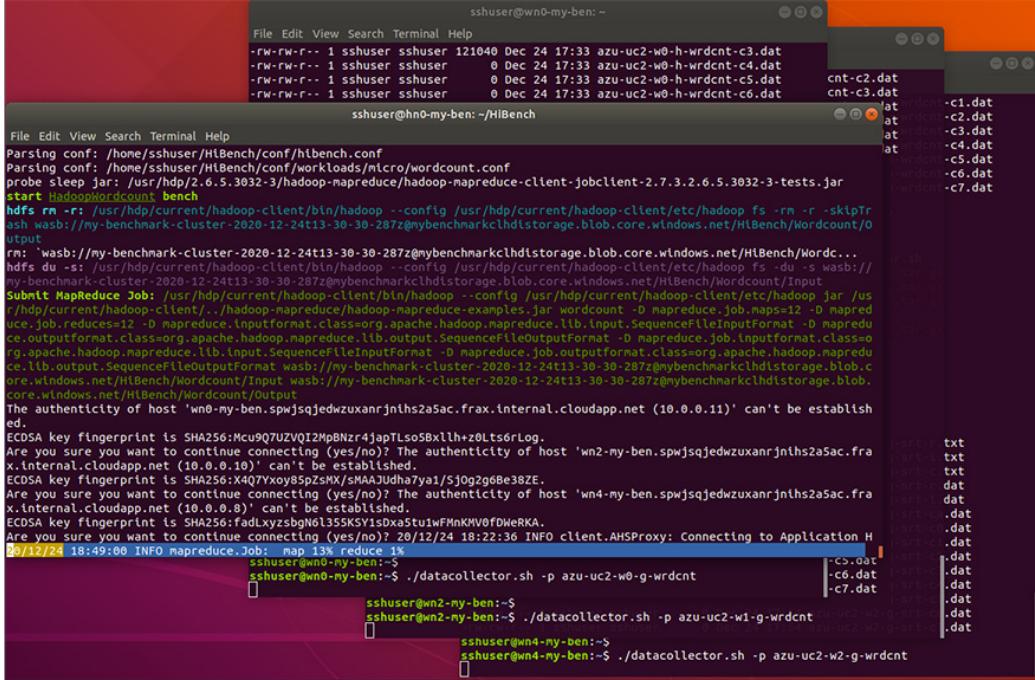


Figure 67: Azure HDInsight - HiBench execution process configurations

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 a datacollector.sh file.

```
# touch datacollector.sh
```

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

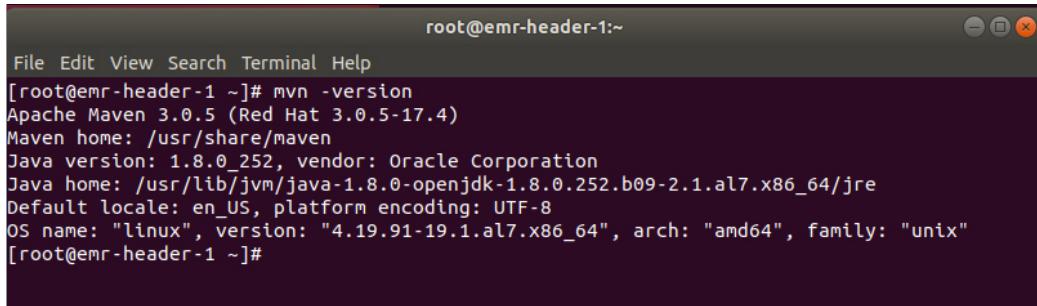
```
# sudo chmod +x datacollector.sh
```

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 68 Verify maven installation

```
# mvn -version
```



```
root@emr-header-1:~#
File Edit View Search Terminal Help
[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 68: Alibaba Cloud e-MapReduce - Maven version

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

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

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
```

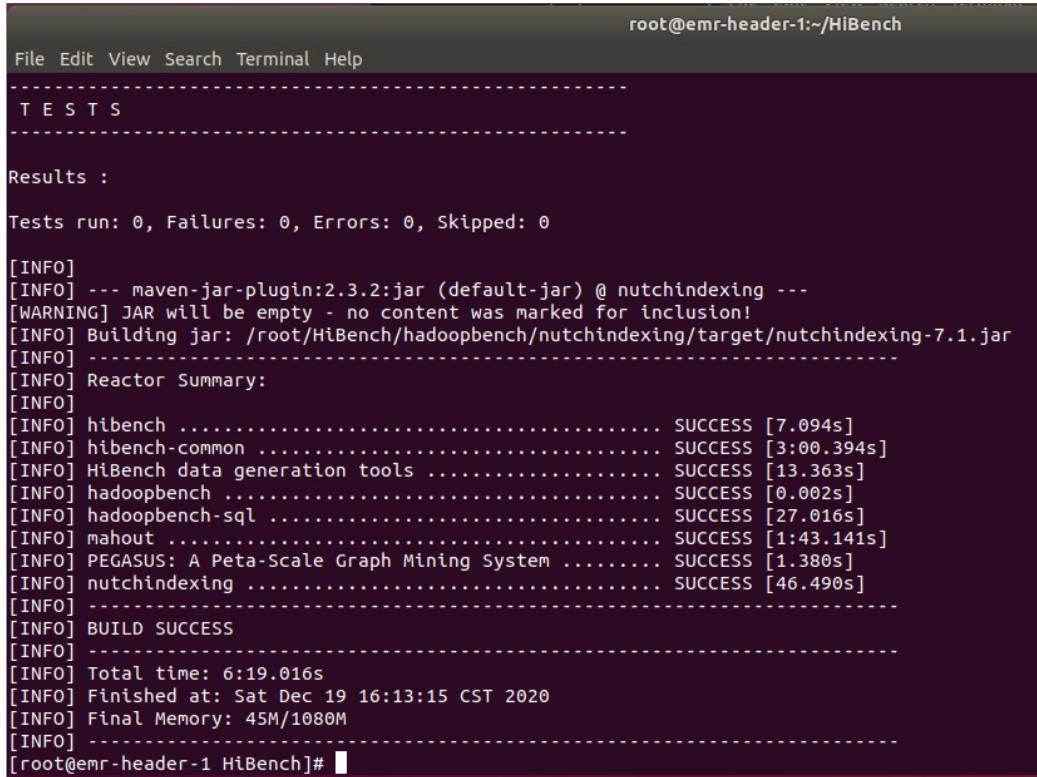
Navigate to HiBench folder

```
# cd HiBench
```

Build HiBench 7.1 for Hadoop

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

Figure 69 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.



```
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 69: Alibaba Cloud e-MapReduce - HiBench success

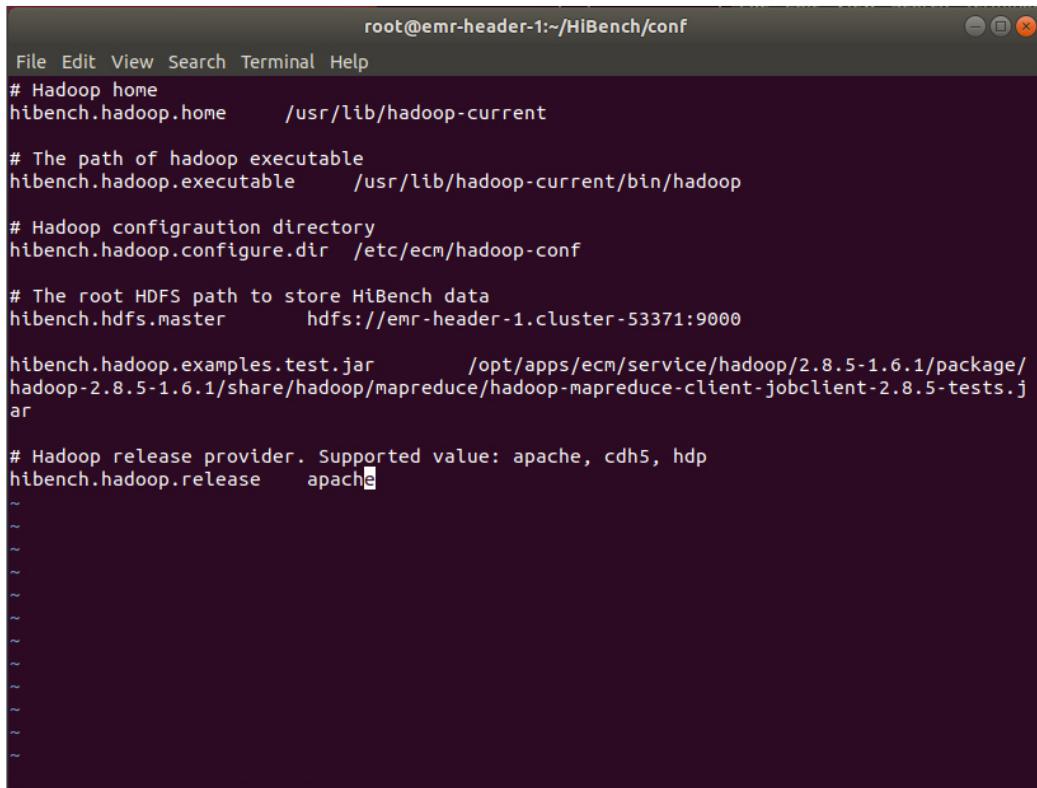
To modify HiBench configuration files, navigate to HiBench's conf folder

```
# 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 70 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 configraution 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, cdhs, hdp
hibench.hadoop.release    apache
~
```

Figure 70: Alibaba Cloud e-MapReduce - Hadoop configurations

Figure 71 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
```

Figure 72 depicts the execution of a HiBench benchmark (Wordcount) and resource utilization capturing process on the worker nodes.

```
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\n"

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

Figure 71: Alibaba Cloud e-MapReduce - HiBench configurations

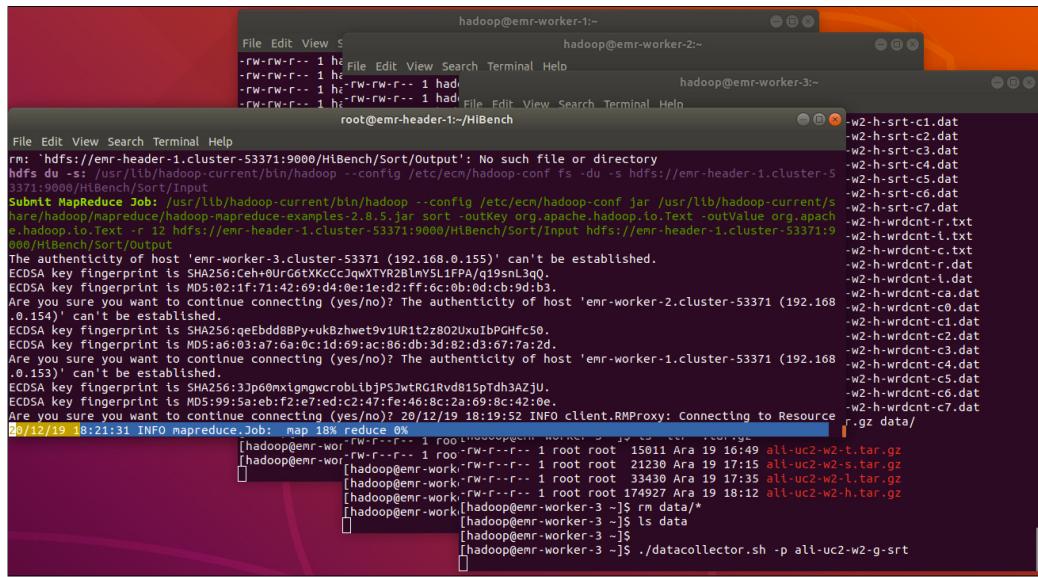


Figure 72: Alibaba Cloud e-MapReduce - HiBench running

List of Figures

1	GCP Mainpage	3
2	GCP - New Project	3
3	GCP - Create Project	4
4	GCP - Enabling Dataproc API	4
5	GCP- Create Firewall (Step 1)	5
6	GCP- Create Firewall (Step 2)	6
7	GCP- Create Firewall (Step 3)	7
8	GCP- Create Firewall (Step 4)	7
9	GCP Dataproc - Create Cluster (Step 1)	8
10	GCP Dataproc - Create Cluster (Step 2)	8
11	GCP Dataproc - Create Cluster (Step 3)	9
12	GCP Dataproc - Create Cluster (Step 4)	10
13	GCP Dataproc - Create Cluster (Step 5)	10
14	GCP Dataproc - Create Cluster (Step 6)	11
15	GCP Dataproc - Create Cluster (Step 7)	12
16	GCP Dataproc - Create Cluster (Step 8)	12
17	GCP Dataproc - Create Cluster (Step 9)	13
18	GCP Dataproc - Create Cluster (Step 10)	13
19	GCP Dataproc - Create Cluster (Step 11)	13
20	GCP Dataproc - Create Cluster (Step 12)	14
21	GCP Dataproc - Namenode Manager	15
22	GCP Dataproc - Resource Manager	15
23	HDInsight setup - Create a resource group	16
24	HDInsight setup - Register subscription	17
25	HDInsight setup - Navigate to HDInsight	17
26	HDInsight setup - Create Cluster	18
27	HDInsight setup - Create Cluster	19
28	HDInsight setup - Storage	20
29	HDInsight setup - Networking	21
30	HDInsight setup - Configuration	22
31	HDInsight setup - Review	23
32	HDInsight setup - In progress	24
33	HDInsight setup - Complete	24
34	HDInsight setup - Overview	25
35	HDInsight - Ambari Dashboard	26
36	HDInsight - Cluster login	26
37	HDInsight - Connecting to master node	27
38	HDInsight - Listing worker nodes	27

39	HDInsight - Ready to benchmark	28
40	e-MapReduce - Create access key	29
41	e-MapReduce - Activate OSS (Step 1)	30
42	e-MapReduce - Activate OSS (Step 2)	30
43	e-MapReduce - Activate OSS (Step 3)	31
44	e-MapReduce - Create OSS bucket	31
45	e-MapReduce - Enabling access logging on OSS	32
46	e-MapReduce - Select Region	32
47	e-MapReduce - Create cluster (Step 1)	33
48	e-MapReduce - Create cluster (Step 2)	33
49	e-MapReduce - Create cluster (Step 3)	34
50	e-MapReduce - Create cluster (Step 4)	34
51	e-MapReduce - Create cluster (Step 5)	35
52	e-MapReduce - Create cluster (Step 6)	36
53	e-MapReduce - Create cluster (Step 7)	37
54	e-MapReduce - Create cluster (Step 8)	38
55	e-MapReduce - Create cluster (Step 9)	38
56	e-MapReduce - Create cluster (Step 10)	39
57	e-MapReduce - Connecting to the cluster	40
58	e-MapReduce - Connecting to worker nodes	41
59	e-MapReduce - Handling broken pipe error	41
60	e-MapReduce - Ready to benchmark	42
61	HiBench Dataproc - Uploading datacollector.sh	43
62	HiBench Dataproc - Verify Maven installation	44
63	HiBench Dataproc - Success compiled	45
64	HiBench Dataproc - Hadoop configurations	47
65	HiBench Dataproc - HiBench configurations	47
66	HiBench Dataproc - Profile from an ongoing benchmark execution	49
67	Azure HDInsight - HiBench execution process configurations	52
68	Alibaba Cloud e-MapReduce - Maven version	53
69	Alibaba Cloud e-MapReduce - HiBench success	54
70	Alibaba Cloud e-MapReduce - Hadoop configurations	55
71	Alibaba Cloud e-MapReduce - HiBench configurations	56
72	Alibaba Cloud e-MapReduce - HiBench running	56