

Deploying IBM Spectrum LSF auto-scale clusters on Google Cloud



Li Flame

*IBM
May 28, 2021*

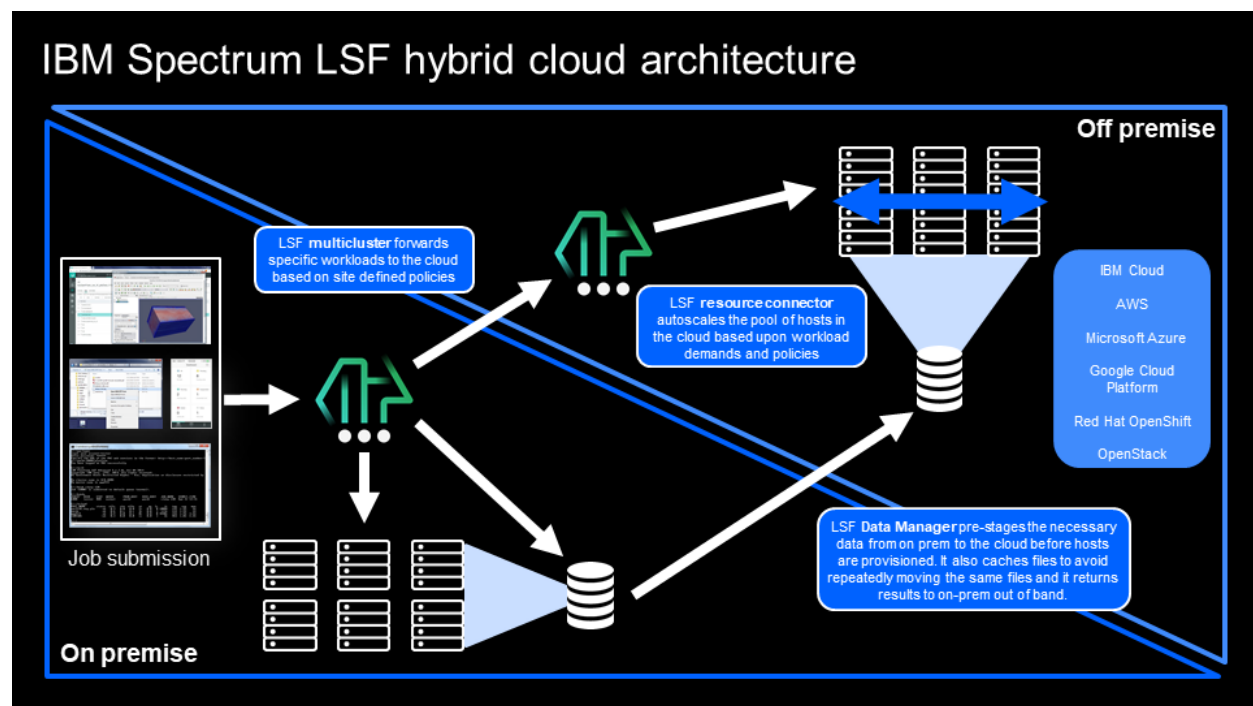
Table of Contents

<i>About IBM Spectrum LSF</i>	<i>2</i>
<i>LSF resource connector overview</i>	<i>3</i>
<i>Creating LSF clusters on Google Cloud and configure for LSF resource connector</i>	<i>3</i>
1. Create a Google Cloud project	3
2. Create a Google Cloud VM instance as an LSF management node.	6
3. Install and configure basic parameters for the LSF management node.	8
4. Create a Google Cloud VM for the compute node and install LSF on it.	11
5. Run the LSF installation script.	11
6. If needed, configure your DNS/NIS server.	12
7. Manually start the LSF daemons on the instance	12
8. Request a VM with the Google Cloud storage image.	12
9. Request a VM with the Google Cloud instance.	15

About IBM Spectrum LSF

IBM Spectrum LSF (“LSF”) is a powerful workload management platform for demanding, distributed HPC environments. LSF provides a comprehensive set of intelligent, policy-driven scheduling features that enables full utilization of your compute infrastructure resources and ensures optimal application performance.

IBM Spectrum LSF features dynamic hybrid cloud capabilities, enabling deployments on-premises, in the cloud, and in the hybrid cloud. Responding to workload, IBM Spectrum LSF dynamically provisions external cloud resources from various cloud providers including Google Cloud Platform. Resources that are no longer needed are returned in time and the administrator can define which containers are permitted to run in the environment. IBM Spectrum LSF hybrid HPC cloud cluster size also dynamically responds to demands. The cluster grows and shrinks accordingly, so you pay only for what you need. The generalized interface supports containerized workloads of Docker, Shifter, Singularity and Enroot technologies. The consistency of deployments on-premises and in the cloud is secured by the help of containerized applications as IBM Spectrum LSF facilitates resource binding, interactive and parallel job support, and automatic container re-running during failure of containerized workloads.



You can now deploy IBM Spectrum LSF on Google Cloud as an auto-scaling cluster based on job requirements. The latest versions of LSF support the various specification and selection

properties of Google Cloud VMs, including scheduling computing jobs on VMs with GPUs, creating VMs in specific zones and regions, and selecting minimum CPU platforms. LSF also supports other Google Cloud features including instance templates, bulk APIs, preemptible VMs, and local SSD disks. Note that the bulk API is used by default by LSF.

LSF resource connector overview

The resource connector for IBM Spectrum LSF (previously referred to as *host factory*) enables LSF clusters to borrow resources from Google Cloud. LSF clusters can borrow hosts from Google Cloud to satisfy pending workload. The borrowed resources join the LSF cluster as hosts. When the resources become idle, the LSF resource connector returns them to Google Cloud.

The resource connector generates requests for extra hosts from Google Cloud and dispatches jobs to dynamic hosts that join the LSF cluster. When the Google Cloud reclaims the hosts, the resource connector requeues the jobs that are running on the LSF hosts, shuts down LSF daemons, and releases the hosts to Google Cloud.

The following is a use case of how the LSF resource connector runs jobs on Google Cloud VM instances.

```
[user@lsfmanage log]$ bhosts
HOST_NAME      STATUS      JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
lsfmanage.us-centr ok          -      2     0      0    0      0      0
[user@lsfmanage log]$ bsub -R googlehost sleep 100
Job <652> is submitted to default queue <normal>.
[user@lsfmanage log]$ bjobs
JOBID  USER  STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
652    root  RUN   normal    lsfmanage.u compute-oet  sleep 100  Apr  9 06:58
[user@lsfmanage log]$ bhosts
HOST_NAME      STATUS      JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
compute-oetct0001. closed    -      1     1      1    0      0      0
lsfmanage.us-centr ok          -      2     0      0    0      0      0
[root@lsfmanage log]#
```

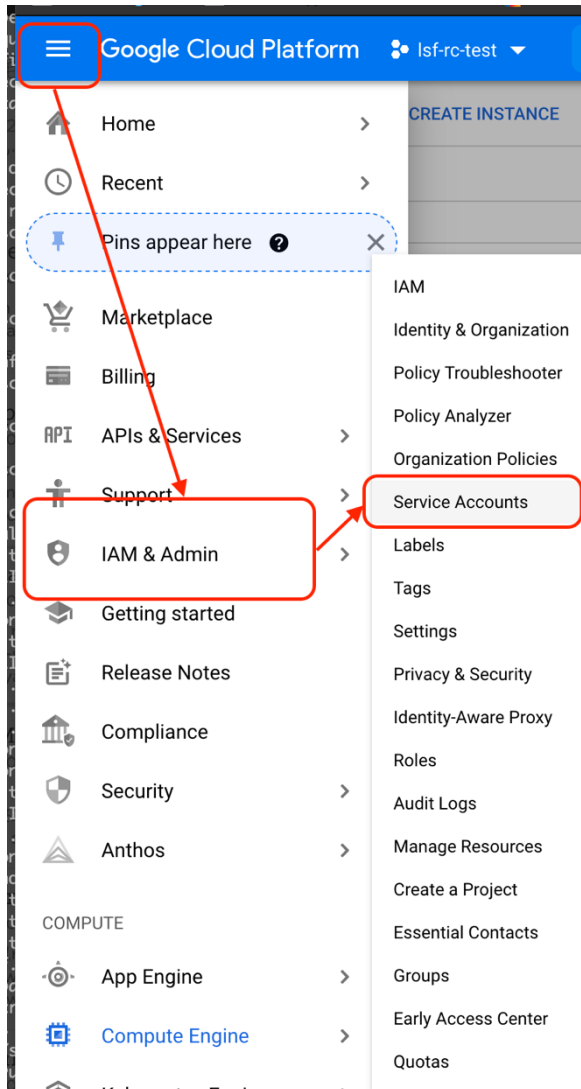
Creating LSF clusters on Google Cloud and configure for LSF resource connector

1. Create a Google Cloud project
 - a. Log in to the Google Cloud console and create a new project or select an existing project.

Your future work is based on this project.

- b. After creating or selecting a project, configure the service account for the project.

Select **IAM & Admin > Service Accounts**.



- c. Create a new service account in this project, then select **Manage permissions** in the action menu to set the service account's permissions.

Service accounts [+ CREATE SERVICE ACCOUNT](#) [DELETE](#) [MANAGE ACCESS](#)

Service accounts for project "lsf-rc-test"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Filter Enter property name or value

	Email	Status	Name ↑	Description	Key ID	Key creation date	Actions
<input type="checkbox"/>	810721966646-compute@developer.gserviceaccount.com	✓	Compute Engine default service account		af9e765bb22f2a065e2cce33a3b3d0d3833d11e	Mar 18, 2021	⋮
<input type="checkbox"/>	test2-888@lsf-rc-test.iam.gserviceaccount.com	✓	test2		a0120aa4900d10b3c4ed6edfd028b009acab3814ba05bd62da3cedee332680431c3601f400c9d		

Manage details
Manage permissions
 Manage keys
 View metrics
 View logs
 Disable
 Delete

- d. Make sure that you set the proper permissions, and that the role of the current service account is **Editor**.

test2

DETAILS **PERMISSIONS** KEYS METRICS LOGS

Resources this service account can access

You can create a policy report for this service account to view a summary of all resources this service account has access to.

[VIEW ACCESS](#)

Members with access to this service account

You can allow specific users to have ownership and access to service accounts and their settings. Users with primitive project owner and project editor roles can already modify service accounts, but you might want to restrict access for some users so that they can take only specific actions against service account resources. [Learn how to grant users permission to service accounts](#)

[GRANT ACCESS](#)

View By: **MEMBERS** ROLES ☐ Include Google-provided role grants

Filter Enter property name or value

Type	Member ↑	Name	Role	Inheritance	
	810721966646-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor	lsf-rc-test	
	810721966646@cloudservices.gserviceaccount.com	Google APIs Service Agent	Editor	lsf-rc-test	
			Editor	lsf-rc-test	
			Editor	lsf-rc-test	
			Editor	lsf-rc-test	
			Owner	lsf-rc-test	

- e. Download a credential key for later use.
Click **KEYS**, then click **ADD KEY** and select **Create new key**.

The screenshot shows the Google Cloud IAM console interface. At the top, there's a breadcrumb trail with a back arrow and the text 'try'. Below this is a horizontal tab bar with five tabs: 'DETAILS', 'PERMISSIONS', 'KEYS' (which is selected and underlined), 'METRICS', and 'LOGS'. On the left side, there is a vertical sidebar with various icons, including a shield, a person, a key, a document, a list, a folder, a key, a right arrow, and a gear. The main content area is titled 'Keys' and contains the following text: 'Add a new key pair or upload a public key certificate from an existing key pair. Please note that public certificates need to be in RSA_X509_PEM format. [Learn more about upload key formats](#)'. Below this, it says 'Block service account key creation using [organization policies](#). [Learn more about setting organization policies for service accounts](#)'. There is a button labeled 'ADD KEY' with a dropdown arrow. A dropdown menu is open below this button, showing two options: 'Create new key' and 'Upload existing key'. To the right of the dropdown menu, there is a table with two columns: 'Key creation date' and 'Key expiration date'.

In the next pane, select **JSON**, then click **CREATE** to download a JSON file to your computer. Keep this file, which you will use when creating Google Cloud VM instances for LSF jobs.

Create private key for "try"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

☒ JSON

Recommended

☐ P12

For backward compatibility with code using the P12 format

CANCEL


CREATE

2. Create a Google Cloud VM instance as an LSF management node.
 - a. Click the **Compute Engine** menu item, then select **VM instances**.
 - b. Click **CREATE INSTANCE**.

VM instances						
CREATE INSTANCE Download Refresh Play Stop Pause Restart Delete						
MANAGE ACCESS SHOW INFO PANEL LEARN						
Filter VM instances						
Columns						
<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/> lsfcompute1	us-central1-a			10.128.0.6 (nic0)	None	SSH - ⋮
<input checked="" type="checkbox"/> lsfmanage	us-central1-a			10.128.0.4 (nic0)	34.122.95.99 ↗	SSH - ⋮

- c. Select **New VM instance** and specify the required information.
- Specify the name and labels according to your requirements.
 - Select a region and zone for location of the VM. Different regions and zones may have different available resources and only certain zones have certain resources such as GPUs. This may impact how you select your machine properties by selecting different zone and regions.
 - For **Boot disk**, you can select the OS type and boot disk size.
 - If required, click **Management, security, disks, networking, sole tenancy** to expand the configuration panel, where you can perform tasks such as setting SSH security or attach extra data disks.

Boot disk ?



New 10 GB balanced persistent disk

Image

Debian GNU/Linux 10 (buster)

Change

Identity and API access ?

Service account ?

Compute Engine default service account

Access scopes ?

☒ Allow default access

☐ Allow full access to all Cloud APIs

☐ Set access for each API

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

☐ Allow HTTP traffic

☐ Allow HTTPS traffic

⌵ Management, security, disks, networking, sole tenancy

You will be billed for this instance. [Compute Engine pricing](#) ↗

[Create](#) [Cancel](#)



Equivalent [REST](#) or [command line](#)

- In the **Security** section, if you add the ssh-key that you generated from a local host, you can use ssh to log in to the created VM from a

local directory using the external IP address of that VM. This makes it convenient to install and configure LSF on the VM instance.

- d. After setting all the required properties for the instance, click **Create** to create the VM, and the VM will automatically start up after it is created.

Boot disk ?

 New 10 GB balanced persistent disk
Image
 Debian GNU/Linux 10 (buster) Change

Identity and API access ?

Service account ?

Compute Engine default service account

Access scopes ?

☒ Allow default access

☐ Allow full access to all Cloud APIs

☐ Set access for each API

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

☐ Allow HTTP traffic

☐ Allow HTTPS traffic

☒ Management, security, disks, networking, sole tenancy

You will be billed for this instance. [Compute Engine pricing](#)

Create Cancel

Equivalent [REST](#) or [command line](#)

3. Install and configure basic parameters for the LSF management node.
 - a. Get the external IP address from the VM list.
 - b. Transfer the LSF installation package and entitlement file into the new VM instance.
 - c. Log in to the new VM instance.
 - d. Install LSF 10.1.

For more information on installing LSF, see [IBM Spectrum LSF documentation](#).
If you created the VM with RHEL8, you must install libnsl first.
 - e. Configure the following parameters in the following files in the new LSF management node:
 - lsf.conf

```
LSF_ROOT_USER=Y
LSB_RC_EXTERNAL_HOST_FLAG=googlehost
LSB_RC_EXTERNAL_HOST_IDLE_TIME=2
LSB_GPU_NEW_SYNTAX=extend
```

```
LSF_GPU_AUTOCONFIG=Y
```

- **lsbatch/*clustername*/configdir/lsb.modules**
In the PluginModule section, define the following:

```
Schmod_demand () ()
```

- **lsf.shared:**

```
Begin Resource
RESOURCENAME    TYPE      INTERVAL  INCREASING  DESCRIPTION
Googlehost      Boolean   ()         ()           (googlehost
flag)
rc_account      String   ()         ()           (RC required)
instanceID      String   ()         ()           (RC required)
templateID      String   ()         ()           (RC required)
clusterName     String   ()         ()           (RC required)
providerName    String   ()         ()           (RC required)
End Resource
```

- **lsbatch/*clustername*/configdir/lsb.queues:**
Configure a queue for running jobs in Google Cloud VMs, then configure the following parameter in the queue:

```
RC_HOSTS=googlehost
```

- **Enable user_data.sh by running the following commands:**

```
cp
$LSF_TOP/10.1/resource_connector/google/scripts/example_user_d
ata.sh
$LSF_TOP/10.1/resource_connector/google/scripts/user_data.sh
```

- **Edit the user_data.sh file and change the following parameter values:**

```
LSF_TOP=lsf_top_directory
LSF_CONF_FILE=$LSF_TOP/conf/lsf.conf
. $LSF_TOP/conf/profile.lsf
Env >> $logfile
```

Make sure user_data.sh will start LSF daemons in the end. If the LSF daemons fail to start, check the contents of the defined logfile.

- **Enable Google in the \$LSF_TOP/conf/resource_connector/hostProviders.json file by adding the Google provider:**

```
{
  "providers": [
    {
      "name": "google",
      "type": "googleProv",
      "path": "resource_connector/google/provider.json"
    }
  ]
}
```

```
]
}
```

- Create the \$LSF_TOP/conf/resource_connector/google/provider.json file:

```
{
  "host_type": "google_host",
  "interfaces":
  [{
    "name": "getAvailableTemplates",
    "action":
"resource_connector/google/scripts/getAvailableTemplates.sh"
  },
  {
    "name": "getReturnRequests",
    "action":
"resource_connector/google/scripts/getReturnRequests.sh"
  },
  {
    "name": "requestMachines",
    "action":
"resource_connector/google/scripts/requestMachines.sh"
  },
  {
    "name": "requestReturnMachines",
    "action":
"resource_connector/google/scripts/requestReturnMachines.sh"
  },
  {
    "name": "getRequestStatus",
    "action":
"resource_connector/google/scripts/getRequestStatus.sh"
  }]
}
```

- Copy the \$LSF_TOP/conf/resource_connector/google/conf directory to the management node VM if not already on the management node VM:

```
cp -rpf $LSF_TOP/10.1/resource_connector/google/conf
$LSF_TOP/conf/resource_connector/google/
```

- Configure the project and credential files.
Transfer the credential file in json format into the LSF management node VM, configure the project name and the credential file location in the \$LSF_TOP/conf/resource_connector/google/conf/googleprov_config.json file:

```
{
  "LogLevel": "INFO",
  "GCLOUD_PROJECT_ID": "lsf-rc-test",
```

```
"GCLOUD_CREDENTIAL_FILE": "/home/yanli/lsf-rc-test-
af9e765bb22f.json"
}
```

- Start up the LIM, RES, and sbatchd daemons on the LSF management node by running the lsfstartup command:
lsfstartup

4. Create a Google Cloud VM for the compute node and install LSF on it.

- a. Create another VM to install the LSF compute node and use this VM as the source disk to generate VMs for LSF jobs in the future.
- b. Create a VM in same subnet with LSF management node.
- c. Log in to the instance that you created.
- d. Copy the LSF packages and the entitlement file to the instance.

For example, copy the following files:

- lsf10.1_linux2.6-glibc2.3-x86_64.tar.Z
- lsf10.1_lsfinstall.tar.Z
- lsf_std_entitlement.dat

- e. If your VM does not have ed installed, use the `yum install` command to install it.

```
yum install ed
```

- f. Edit the server.config file in the LSF installer and enable the following parameters:

- LSF_TOP: Specify the same location as the LSF management node.
- LSF_ADMINS: Specify the same users as the LSF management node.
- LSF_TARDIR: Specify the location of the LSF packages.
- LSF_ENTITLEMENT_FILE: Specify the location of the entitlement file.
- LSF_SERVER_HOSTS: Specify the name of the LSF management node.
- LSF_LOCAL_RESOURCES: Specify "[resource googlehost]" to define googlehost as a local resource.
- LSF_LIM_PORT: Specify the same LIM port as the LSF management node.

For example:

```
LSF_TOP="/opt/lsf"
LSF_ADMINS="lsfadmin"
LSF_TARDIR="/opt/install/"
LSF_LICENSE="/opt/install/lsf_std_entitlement.dat"
LSF_SERVER_HOSTS="management.myserver.com"
LSF_LOCAL_RESOURCES="[resource googlehost]"
LSF_LIM_PORT="7869"
LSF_GET_CONF=lim
```

5. Run the LSF installation script.

```
./lsfinstall -s -f server.config
```

After installation, make sure the `<LSF_TOP>/conf/lsf.conf` configuration file contains the "googlehost" resource:

```
cat /opt/lsf/conf/lsf.conf
LSF_GET_CONF=lim
LSF_CONFDIR=/opt/lsf/conf
LSF_LIM_PORT=7869
LSF_SERVER_HOSTS="management.myserver.com"
LSF_VERSION=10.1
LSF_LOCAL_RESOURCES="[resource googlehost]"
LSF_TOP=/opt/lsf/
...
```

6. If needed, configure your DNS/NIS server.

The default DNS server for Google Cloud works with LSF.

7. Manually start the LSF daemons on the instance

Manually start the LSF daemons on the instance with the `lsfstartup` command and make sure that the instance can join the LSF cluster as a dynamic host. If the instance does not join the cluster, check the following items, then restart the LSF daemons:

- The instance firewall or firewall rules of the VPC network.
- The management host and the instance can ping each other using the private IP address.

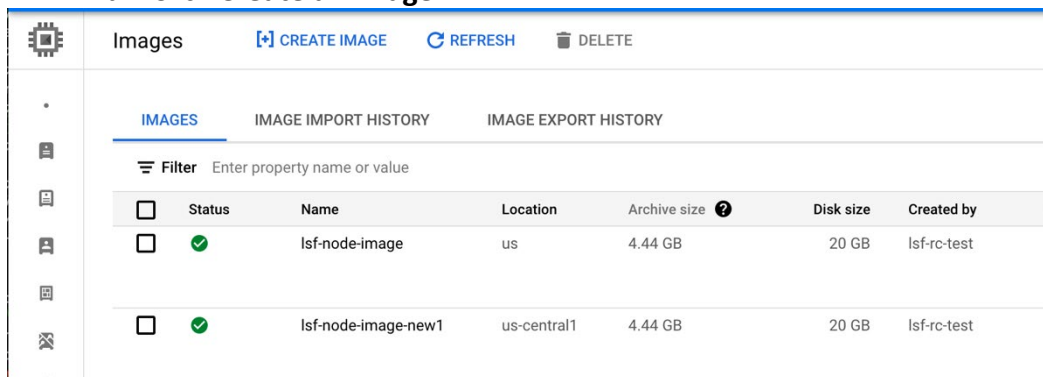
If the instance joins the cluster, you can shut down the compute node VM.

8. Request a VM with the Google Cloud storage image.

In Google Cloud, you can create a storage image from the installed LSF compute node, then the LSF resource connector calls the Google API to create a VM on which to execute LSF jobs.

To use the VM in this manner, first create a storage image on the Google Cloud Console.

- a. Select **Compute Engine** from the menu, then select **STORAGE > Images**.
- b. Click **Create an image**.



Status	Name	Location	Archive size	Disk size	Created by
Success	lsf-node-image	us	4.44 GB	20 GB	lsf-rc-test
Success	lsf-node-image-new1	us-central1	4.44 GB	20 GB	lsf-rc-test

- c. Specify a name for the storage image, then select **Source > Disk** and select the name of the LSF compute node VM from the **Source disk** list.

Create an image

Name ⓘ
Name is permanent
image-1

Source ⓘ
Disk

Source disk ⓘ

- compute-w46ix0-1
- compute-wfwch0-1
- lsfcompute1
- lsfmanage
- lsfnode-ssd-temp1-1-1

Labels ⓘ (Optional)
+ Add label

Encryption
Data is encrypted automatically. Select an encryption key management solution.

- ☒ Google-managed key
No configuration required
- ☐ Customer-managed key
Manage via Google Cloud Key Management Service
- ☐ Customer-supplied key
Manage outside of Google Cloud

You will be billed for this image. [Compute Engine pricing](#)

Create Cancel

- d. Click **Create** and wait for Google Cloud to create the storage image.
- e. Log in to the LSF management node VM
- f. Configure the LSF_TOP/conf/resource_connector/google/conf/googleprov_templates.json file to define a template for the created storage image.
For example,

```
{
  "templateId": "gcloud-VM-1",
  "maxNumber": 100,
  "attributes": {
    "type": ["String", "X86_64"],
    "ncores": ["Numeric", "1"],
```

```

        "ncpus": ["Numeric", "1"],
        "nthreads": ["Numeric", "2"],
        "ngpus": ["Numeric", "1"],
        "ngpus_physical": ["Numeric", "1"],
        "gpuextend": ["String",
"ngpus=1;nnumas=1;gbrand=Tesla;gmodel=K80;gmem=10240;nvlink=yes"
],
        "define_ncpus_threads": ["Boolean", "1"],
        "mem": ["Numeric", "3840"],
        "zone": ["String", "us-east1-d"],
        "googlehost": ["Boolean", "1"]
    },
    "imageId": "lsf-gcloud-dynamic-vm",
    "region": "us-east1",
    "zone": "us-east1-d",
    "vmType": "n1-standard-1",
    "gpuType": "nvidia-tesla-k80",
    "gpuNumber": "1",
    "instanceTags" : "lsf-vpn-instance=gcloud-VM-1",
    "userData": "zone=us-east1-d"
}

```

- `templateID` defines the template name that is used within LSF (gcloud-VM-1).
- `attributes` defines the LSF resources that are used for LSF scheduling to select a template that most fits the pending job's requirements if there are multiple templates and storage names.
- Parameter values that are defined outside of `attributes` are used while calling the Google Cloud API to trigger VM. These parameters must follow the Google Cloud API.

After defining this template, you can run the following LSF command to submit a job:

```
bsub -R "googlehost" -gpu "num=1" myjob
```

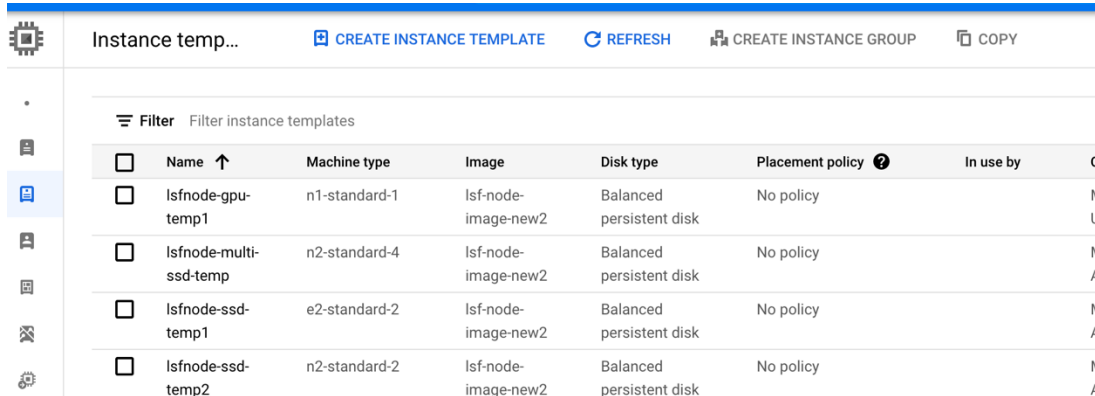
LSF checks to see if there is an available VM with the "googlehost" resource defined that is ready for the job. If there is no VM available, LSF navigates the templates that are defined in the `googleprov_templates.json` file and checks the resources that are defined in `attributes`. If LSF finds a template with `attributes` that fit the pending job's requirements, LSF calls the Google API to create a VM according the requirements for the matching template.

9. Request a VM with the Google Cloud instance.

Google Cloud has a concept of instance templates, and LSF can use this kind of instance template to directly create a VM. First, create an instance template based on the LSF compute node that you successfully installed.

- a. Select **Compute Engine** from the menu, then select **VIRTUAL MACHINES > Instance templates**.

Click “CREATE INSTANCE TEMPLATE”.



<input type="checkbox"/>	Name ↑	Machine type	Image	Disk type	Placement policy ?	In use by	
<input type="checkbox"/>	lsfnode-gpu-temp1	n1-standard-1	lsf-node-image-new2	Balanced persistent disk	No policy		lv U
<input type="checkbox"/>	lsfnode-multi-ssd-temp	n2-standard-4	lsf-node-image-new2	Balanced persistent disk	No policy		lv A
<input type="checkbox"/>	lsfnode-ssd-temp1	e2-standard-2	lsf-node-image-new2	Balanced persistent disk	No policy		lv A
<input type="checkbox"/>	lsfnode-ssd-temp2	n2-standard-2	lsf-node-image-new2	Balanced persistent disk	No policy		lv A

- b. Specify a name for the instance template and select the CPU memory and GPU type/number/size for the template by clicking **GPU**, then selecting the GPU type and number for the template.

TIP: For different regions and zones, the CPU/GPU resources and quotas are also different. Refer to Google Cloud documentation to check which regions and zones have the special CPU/GPU resources that you need.

←

Create an instance template

•

Describe a VM instance once and then use that template to create groups of identical instances [Learn more](#)

•

Name

Name is permanent

instance-template-1

•

Machine configuration

Machine family

General-purpose

Compute-optimized

Memory-optimized

GPU

Optimized for machine learning, high performance computing, and visualization workloads

Series

A2

Powered by NVIDIA A100 graphics cards.

Machine type

a2-highgpu-1g (12 vCPU, 85 GB memory)

	vCPU	Memory	GPUs
	12	85 GB	1 x NVIDIA Tesla A100

•

CPU platform

CPU platform configuration is permanent

Automatic

•

GPU type

NVIDIA Tesla A100

Number of GPUs

1

×

☐ Enable Virtual Workstation (NVIDIA GRID)

To enable Virtual Workstation (NVIDIA GRID), choose a different GPU such as NVIDIA Tesla T4, P4 or P100. [Learn more](#).

•

Display device

Turn on a display device if you want to use screen capturing and recording tools.

☐ Turn on display device

I>

[↑ CPU platform and GPU](#)

- c. Click **Boot disk > Change** to select the image and OS.
- d. In the **Boot disk** page, select **Custom images**, then select the storage image that you created from the LSF compute node VM.

Boot disk

Select an image to create a boot disk. The image determines the operating system installed on the instar solutions in [Marketplace](#).

Public images

Custom images

Show images from
lsf-rc-test

☐ Show deprecated images

Image

lsf-node-image
Created on Mar 22, 2021, 11:04:09 AM

lsf-node-image-new1
Created on Mar 22, 2021, 12:43:58 PM

lsf-node-image-new2
Created on Mar 22, 2021, 12:44:34 PM

- e. Click **Select** to go back to the previous page, then click **Management,security,disks,networking,...** to expand the page and select **Disks**.

Firewall ?

Add tags and firewall rules to allow specific network traffic from the Internet

- ☐ Allow HTTP traffic
☐ Allow HTTPS traffic

Management, security, disks, networking, sole tenancy

The following options have been customized:

On host maintenance

You can create this instance template free of charge

Create

Cancel

- f. In the **Disks** page, you can add extra disks beside the boot disk. Click **Add new disk** to add extra disks.

Boot disk

Deletion rule

☒ Delete boot disk when instance is deleted

Encryption

Data is encrypted automatically. Select an encryption key management solution.

☒ Google-managed key

No configuration required

☐ Customer-managed key

Manage via Google Cloud Key Management Service

Device name [?]

Use the device name to identify this disk.

Based on instance template name (default) ▼

instance-template-1

Additional disks [?] (Optional)

+ Add new disk

+ Attach existing disk

- g. When adding disks, you can select which type of disks to add. LSF also supports **Local SSD scratch disk**. Select this from the list of Type.

Type [?]

Balanced persistent disk

SSD persistent disk

Standard persistent disk

Local SSD scratch disk (maximum 24)

Create snapshot schedules to automatically back up your data. [Learn more about creating snapshot schedules](#) [?]

Dismiss

Source type [?]

Blank disk

Image

Snapshot

Mode

☒ Read/write

☐ Read only

Deletion rule

When deleting instance

☒ Keep disk

☐ Delete disk

Size (GB) [?]

500

Estimated performance [?]

Operation type	Read	Write
Sustained random IOPS limit	3,000.00	3,000.00
Sustained throughput limit (MB/s)	140.00	140.00

Device name [?]

Used to reference the device for mounting or resizing.

- h. After selecting **Local SSD scratch disk**, there are two 2 sub-types (SCSI and NVMe). LSF supports both of these, and in the LSF_TOP/10.1/resource_connector/google/scripts/user_data.sh script file,

LSF demonstrates a way to automatically mount both kinds of local SSD disks to /tmp.

Type ?
Local SSD scratch disk (maximum 24)

☐ SCSI
☒ NVMe

i For optimal NVMe performance, the boot disk image for your instance must include optimized NVMe drivers. [Learn more](#)

Number of disks ?
Multiple Local SSD disks can be created at the same time. (Maximum 24)

1 SSDs

Size (GB) ?
375

Total estimated performance ?

Operation type	Read	Write
Sustained random IOPS limit	170,000.00	90,000.00
Sustained throughput limit (MB/s)	660.00	350.00

Done Cancel

- i. After specifying all the features that you need, click **Create** to create the instance template.

In the LSF_T/OP/conf/resource_connector/google/conf/googleprov_templates.json file, you can use the instance template directly for the Google Cloud API to create the VM.

Use `launchTemplateId` to specify which instance template is used to create the VM.

```
{
  "templateId": "Template-VM-5",
  "launchTemplateId": "lsfnode-gpu-temp1",
  "maxNumber": 2,
  "attributes": {
    "type": ["String", "X86_64"],
    "ncores": ["Numeric", "2"],
    "ncpus": ["Numeric", "2"],
    "nthreads": ["Numeric", "1"],
```

```

        "mem": ["Numeric", "8000"],
        "ngpus": ["Numeric", "2"],
        "ngpus_physical": ["Numeric", "2"],

"gpuextend": ["String", "ngpus=2;nnumas=1;gbrand=Tesla;gmodel=K80;
gmem=10240;nvlank=yes"],
        "googlehost": ["Boolean", "1"]
    },
    "zone": "us-east1-b",
    "instanceTags": "gputemp1"
}

```

To enable launch instance templates in `googleprov_templates.json`, specify the `launchTemplateId` and `zone` attributes. All other attributes become optional after you enable launch instance templates. Define all other attributes in the specified instance template, because all settings that are defined in the `googleprov_templates.json` file override the settings in the instance template.

Within the launch instance template, the attributes are grouped into the following categories:

Disk attributes:

- `imageId`

Network attributes:

- `vpc`
- `subnetId`
- `privateNetworkOnlyFlag`

GPU attributes:

- `gpuType` (see <https://cloud.google.com/compute/docs/gpus>)
- `gpuNumber`

Other attributes:

- `vmType`
- `minCpuPlatform`
- `instanceTags` (Labels must be defined in `googleprov_template.json`. Labels that are defined in the template are overridden by default)

If you define an attribute in the `googleprov_templates.json`, all attributes in the same group override the attribute values in the launch templates, even attributes that you did not define. For example, if you define `gpuType` in `googleprov_templates.json` to override the value in the launch template, the default value of `gpuNumber` also overrides any `gpuNumber` value in the launch

template even if you did not specify a value for `gpuNumber`. To avoid this issue, do not define any attributes in `googleprov_templates.prov` other than `launchTemplateId` and `zone` if you are using launch instance templates.