IBM Spectrum Computing Deployment Guide for LSF onto Amazon Web Services

Table of Contents

Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying IBM Spectrum LSF on the AWS Cloud.

IBM® Spectrum LSF (formerly IBM® Platform™ LSF®) is a complete workload management solution for demanding HPC environments. Featuring intelligent, policy-driven scheduling and easy to use interfaces for job and workflow management, it helps organizations to improve competitiveness by accelerating research and design while controlling costs through superior resource utilization.

Please note, Spectrum LSF is not itself an application in the traditional sense, but instead provides an environment and framework for other applications to be managed and run in a load balanced efficient manner. It is expected that you will install some kind of application(s) into this environment, or use application installed in your on premise environment to make proper evaluation use of the features and benefits of Spectrum LSF.

This Quick Start provides a functional example from which users can use, adapt, build upon for their own needs when using AWS. Our expectation is that users of this Quick Start will have unsteady, predictable or unpredictable spikes in workload which their on premise environments can not process in acceptable time.

Please also note, the term *node* is typically used to refer to any running instance of an operating system. The nodes deployed in this AWS Quick Start are all Amazon Elastic Computing Cloud (Amazon EC2) instances, so this deployment guide will generally use the term *instance* in place of *node.*

Costs & Licenses

The Quick Start builds the IBM Spectrum LSF environment by using pre-built Amazon Machine Images (AMI) with IBM Spectrum LSF pre-installed into the Operating System (see details below in the "Technical Requirements and Design" section).

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. At this time, you must accept a trial license version of IBM Spectrum LSF (trial license) Advanced Edition in order to be able to use the solution deployed by the Quick Start. The use of IBM Spectrum LSF on AWS (including all packages provided via the Quick Start offering and packages derived from these) may only be used for a maximum of 90 consecutive calendar days, and may not be used for production work of any sort. IBM may decide to de-authorize access to the code, and the use of this code, at any time. After the trial period, you are responsible for acquiring

the necessary licenses directly from IBM to use IBM Spectrum LSF. The IBM Spectrum LSF evaluations page will be updated with details on how to proceed with acquiring an IBM Spectrum LSF license after the 90 day trial expires.

The AWS CloudFormation template for this Quick Start includes configuration parameters that you can customize. Some of these settings, such as instance type, will affect the cost of deployment. For cost estimates, see the pricing pages for each AWS instance and service you will be using. Prices are subject to change.

When estimating costs using the pricing pages for AWS services, note any data transferred from Amazon back to an on premise environment can result in data transfer charges which can be significant depending on the amount of data transferred and the timeframe the transfer occurred. See the "Best Practices" section below for more information.

Architecture

Stretched Cluster

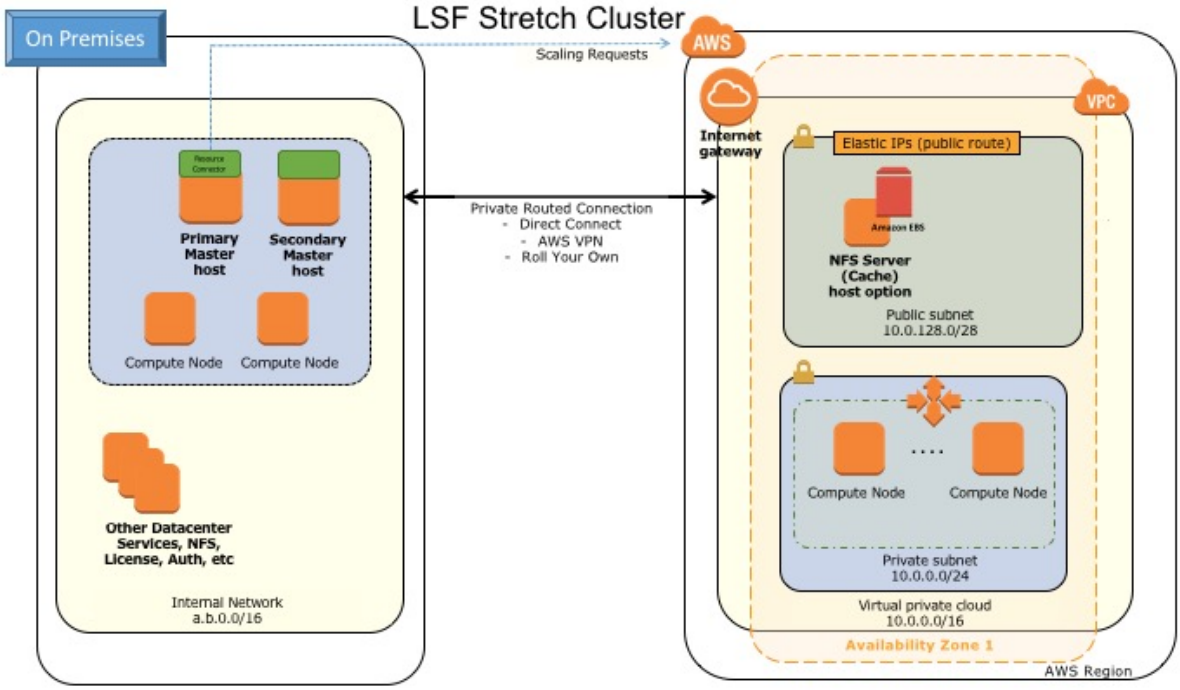

Figure 1: LSF Stretch Cluster

Multi Cluster

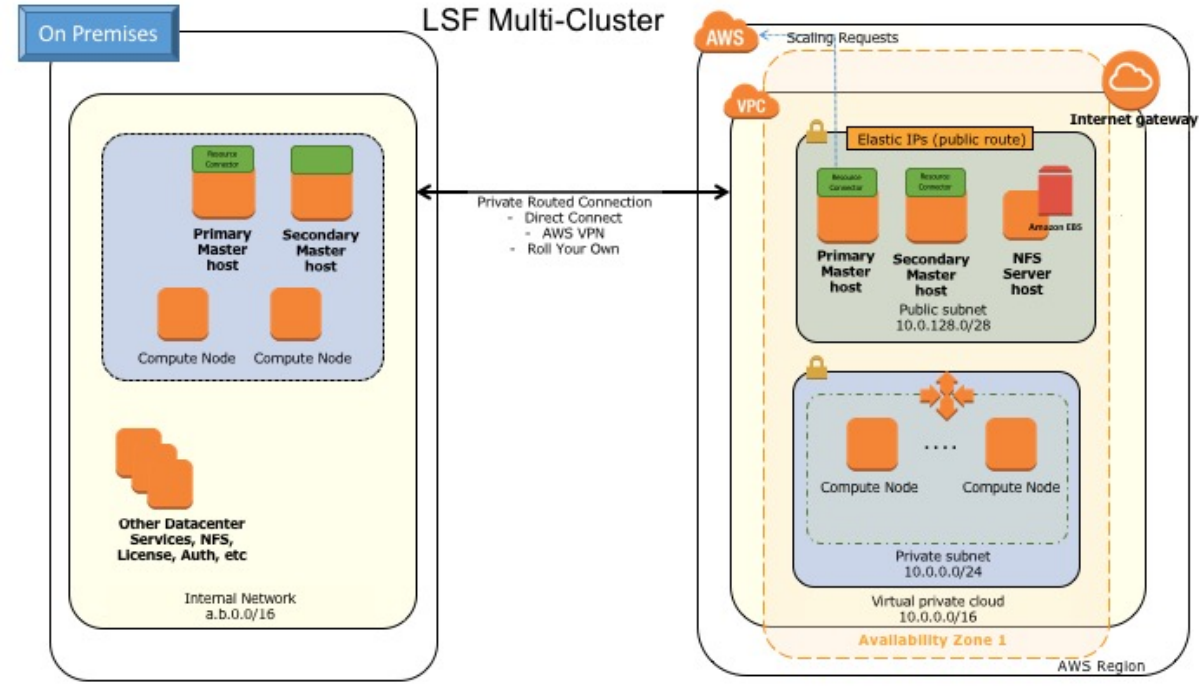

Figure 2: LSF Multi Cluster

Best Practices for using IBM Spectrum LSF on AWS

Planning

This Quick Start covers the installation of LSF Stretch Clusters and LSF Multi Clusters. Each has different use cases and characteristics. These need to be understood in order to select the type of LSF cluster to deploy.

LSF Stretch Clusters, seen in figure 1, are an extension to an existing on premises LSF cluster. Additional compute resources are added to the LSF cluster by using EC2 instances that are joined to the LSF cluster. They are given access to the on premises datacenter systems they need to run the workload. Typically:

- Intended for dealing with predictable on premises resource shortfalls
- Small number of on Cloud machines
- Only used to run specific workload types:
    - High CPU with low IO
    - Little demand for datacenter services
    - Not latency sensitive

LSF Multi Cluster deployments, shown in figure 2, differ in that there is a separate LSF cluster built from EC2 instances. The on premises LSF master

forwards workload to the on Cloud LSF master. The on Cloud LSF master uses the LSF Resource Connector to start EC2 instances on demand. The on-demand EC2 instances are used to run the workload that has been forwarded from the on premises LSF master. The size of the on cloud LSF cluster will change depending on the load on the cluster. This type of configuration typically:

- Used to offload workload from on premises LSF cluster
- Used to support larger workload demand
- Possible needs on cloud storage
- Dynamically resizes based on LSF Resource Manager policies
- Can use Data Manager to move workload data to/from cloud
- Best for specific job types

The type of cluster to deploy will depend on the resource demands and workload characteristics. The quick start guide covers the installation of both types of clusters and will allow evaluation of both.

Launching

The LSF Stretch cluster and LSF Multi clusters are all created using a series of Ansible playbooks. These playbooks are used to perform the following steps in order:

1. Prepare the on premises LSF master to deploy the EC2 instances by installing the necessary software prerequisites.
2. Optionally creating a VPC from some minimal configuration
3. Optionally bringing up a VPN connection
4. Marshalling and preparing configuration files for the cloud machines
5. Launching EC2 instances for the LSF cluster
6. Optionally accessing on premises storage
7. Installing the LSF Stretch cluster or LSF Multi cluster

These playbooks are provided as a framework for customization. Initially they can be run to create a simple on cloud cluster, but they are intended to be taken and customized to meet particular site needs.

Using

Both the LSF Stretch Cluster and LSF Multi Cluster clusters will create a job queue on the LSF master on premises. The LSF Stretch cluster queue is called “awsexample”. The LSF Multi Cluster queue is called “send2cloud”. Workload sent to these queues will be run on the cloud EC2 instances. A user should define which queue to use at job submission time e.g.

```
$ bsub -q {name of queue} ….
```

Or they can specify the queue in the LSF Application Center GUI, or desktop submission user interface. The LSF master will then schedule those workload on the EC2 instances.

It may not be appropriate to run all workload types on the cloud instances. The best types of workload to run on the cloud will depend on the characteristics of the workloads, such as:

- Software licenses needed
- IO characteristics
- Datacenter service access
- Storage

The most cost effective workloads to run this way need to be determined by the LSF administrator.

Maintaining

Both the LSF Stretch and Multi clusters use NFS to host the LSF binaries and log directories. Care should be taken to ensure there is sufficient free space in the shared filesystem. The default instance is providing 8GB.

Decommissioning

The on cloud resources created by running the Quick Start can be easily removed using the provided playbook. It will remove all EC2 instances created, and optionally remove the VPC and associated subnet, internet gateway, network ACL, security group.

Before decommissioning the on cloud resources it is best to stop additional work from being queued by running:
$ badmin qclose –C “Decommissioning the machines” {Name of queue}

Check to see if there are any pending or running workload using:
$ bqueues –w {Name of queue}

Once they are finished run the following to terminate all EC2 instances created by the playbooks:

```
# ansible-playbook –i lsf-inventory Cleanup.yml
```

This will terminate all instances that were created by the initial playbook, but will not terminate instances that were dynamically created in the LSF Multi Cluster. Dynamically created instances will be terminated by the Resource Connector on the on Cloud LSF Master. They can also be terminated in the EC2 Management Console.

> **NOTE:** A bug currently exists in the LSF Multi Cluster, where the volumes associated with the dynamic hosts are not deleted when the instance is terminated. These need to be deleted from the EC2 Management Console to avoid additional charges.

Prerequisites

Specialized Knowledge

Before you deploy this Quick Start, we recommend that you become familiar with the following AWS services. (If you are new to AWS, see Getting Started with AWS). We generally assume also that you are familiar with Linux, and are comfortable on the command line to execute basic commands, edit files, and so forth.

- Amazon EC2 - The Amazon EC2 service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.

- Amazon VPC – The Amazon VPC service lets you provision a private, isolated section of the AWS Cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, subnet creation, and configuration of route tables and network gateways.

- IAM – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can manage users, security credentials such as access keys, and permissions that control which AWS resources users can access, all from a central location.

Optional Specialized Knowledge

Though not strictly required for success using Spectrum LSF in the Amazon cloud, the following areas of knowledge could prove very useful in improving the performance, data security, and robustness of the environment you deploy.

EBS - Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads.

S3 - Amazon S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices. It is designed to deliver 99.999999999% durability, and stores data for millions of applications used by market leaders in every industry. S3 provides comprehensive security and compliance capabilities that meet even the most stringent regulatory requirements.

S3 Managed Keys (SSE) - Server-side encryption protects data at rest. Server-side encryption with Amazon S3-managed encryption keys (SSE-S3) uses strong multi-factor encryption. Amazon S3 encrypts each object with a unique key. As an additional safeguard, it encrypts the key itself with a master key that it rotates regularly. Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt your data.

Technical Requirements and Design Considerations

Sizing IBM Spectrum LSF Instance Types and OS – Management Node

The following guidance is only relevant to the Multi Cluster Deployment option described below. The Quick Start supports a large selection of EC2 instance types for the IBM Spectrum LSF cluster instances.

We recommend that you benchmark the environment with your workload to fully understand the requirements and optimal configuration for the management node in Amazon EC2.

The Quick Start launches the Management Node running CentOS 7

Sizing IBM Spectrum LSF Instance Types and OS – Computation Node

We recommend that you benchmark the environment with your workload to fully understand the requirements and optimal configuration for the management node in Amazon EC2.

The Quick Start launches Computation Nodes running CentOS 7

Restrictions Associated with the Trial Evaluation Quick Start

The Quick Start deploys a trial version of the IBM Spectrum LSF software. (For details on the license terms, see the "Costs and Licenses section") This version doesn't support the following features of Spectrum LSF:

- LSF Explorer i.e. system monitoring will not be installed on the EC2 instances
- LSF Application Center GUI will not be installed on the LSF Multi Cluster on cloud LSF master
- Spectrum MPI, and Platform MPI are not installed on the EC2 instances

LSF Data Manager, and LSF License Scheduler may be available in the LSF Suite edition that was installed, however there configuration is not covered by this quick start guide.

Deployment Options

Assumptions

- You have familiarity with LSF
- You have a running LSF Suite cluster already
- The running LSF cluster has applications, licenses, users, and project data available to it for executing tasks

Stretched Cluster

This architecture assumes that you have a cluster in another location – either on premise or even running in another cloud or cloud location. The "stretched cluster" architecture is defined as a single cluster stretched over a WAN so that compute nodes in the cloud communicate with a master scheduling host on the originating location.

Generally, though much simpler in concept than "Multi-Cluster", this means that all LSF daemon communication with the master scheduler happens over the WAN which can be a source of extra cost or lowered reliability.

Multi Cluster

This is a more complex architecture which adds a master scheduler running in the cloud. By adding a master scheduler in the cloud, the architecture eliminates all the communication from cloud compute node to the on premise master.

The two master schedulers instead exchange task meta-data in a "job forwarding" model. In this model, users on premise submit workload to a queue on premise, which in turn forwards that workload to the cloud for execution. Upon task completion, the master in the cloud communicates completion, and status with the on premise master and the user is notified.

Deployment Recommendation

Spectrum Computing, formerly known as Platform Computing has been working with clients transitioning to the cloud for the better part of a decade at the time this document is being written. That foundation and breadth of experience with several products and IaaS partners has helped us form a recommendation that we provide to our clients making the first and sometimes most important hybrid cloud decision on their journey to the cloud – this recommendation should help you answer "which architecture should I use?"

Three main factors impact this decision:

1. Network latency between your on premise environment and the primary cloud environment being considered
2. The location of required services (Project data, user authentication, application binaries, etc)
3. Expected data traffic exiting the cloud over the internet

The first item can be measured directly and experimentally which we recommend to anyone considering the "Stretched Cluster" architecture. We have found that clients who try to use the "Stretched Cluster" architecture when the latency between datacenters is greater than 40 miliseconds have had less than robust experiences using the cloud.

Second, consider the method you plan on using to connect your datacenter to the cloud. There are a few options, and those options trade off cost, speed of deployment, and reliability.

- VPN connection over the internet
  - Less expensive to start
  - More expensive for large data egress
  - Quick to deploy
  - Relies upon internet connectivity & bandwidth at both ends
- Direct Connect
  - More expensive to start
  - No data egress charges
  - Takes months (minimum) to deploy
  - More robust than internet connectivity
  - Dedicated bandwidth between on premise and the cloud

If you are using a direct connection, then the "Stretched Cluster" architecture is more feasible and reliable. If you plan to use the VPN connection, then the "Multi Cluster" architecture our recommendation.

Finally, with respect to costs, the workload and the amount of data that will exit the cloud onto the internet (presumably bound for the on premise datacenter) is a billable item. However, it's important to consider the number of compute nodes in the cloud as well, because in the "Stretched Cluster" architecture, all of the communication between the master scheduler and each compute node must traverse the internet link, a little less than half being data egressing the cloud.

For large numbers of compute nodes (greater than 100) in the cloud, this can become problematic, and sometimes a significant amount of bandwidth on it's own. So again, we recommend the "Multi Cluster" architecture for the situation of many compute nodes as well.

Deployment Steps

The deployment process is arranged in a series of steps that layer on successive functions. They are provided as Ansible playbooks that are functional examples, but are intended to be taken and enhanced to meet site specific needs.

Obtaining the Ansible Playbooks

The Ansible playbooks are hosted on a public git repository. Clone or download the repository on to the LSF master. Login to the LSF master and:

1. Go to the /opt/ibm/ directory
2. Clone the git repository e.g
   # git clone https://github.com/IBMSpectrumComputing/lsf-hybrid-cloud.git
3. Copy the lsf-inventory file from the machine used to deploy the LSF cluster to /opt/ibm/lsf-hybrid-cloud

The playbook used in the rest of the steps is now ready.

Choosing Which Type of LSF Cluster to Deploy

It is necessary to choose the type of LSF cluster to deploy early in the process. This is done by editing the AWS/AWS-config.yml file. For a LSF Stretch cluster change the file as follows:

```
# What type of cluster to deploy.  Uncomment one of these
#multi_cluster: true
hybrid_cluster: true
```

For a LSF Multi Cluster deployment change the file setting to:

```
# What type of cluster to deploy.  Uncomment one of these
multi_cluster: true
#hybrid_cluster: true
```

If these values are changed it will be necessary to use the Cleanup.yml playbook to reset the configuration.

Installation of prerequisites

The Ansible playbooks need to be run from the existing LSF master. If the LSF master host was not used as the deployment host for installing the existing LSF cluster, the following additional steps will be necessary:

1. Download the LSF Suite package to the LSF master. The edition to use will depend on which features you need. The LSF Suite Workgroup edition will work for stretch clusters, but will not work for Multi-cluster configurations. The packages are named:
   lsfsent10.2.0.6-x86_64.bin - This is the Enterprise package
   lsfshpc10.2.0.6-x86_64.bin - This is the HPC package
   lsfswg10.2.0.6-x86_64.bin - This is the Workgroup package

Use the latest available package. If using the 10.2.0.6 packages, LSF Multi cluster deployments will also need iFix fix ID: Suite-10.2.0.6-ifix-499974 or higher.

2. Run the ".bin" file and accept the license. This will install needed components on the LSF master.
3. Copy the /opt/ibm/lsf_installer/playbook/lsf-inventory from the deployment machine to the LSF master.

The LSF Master will need access to the internet to download rpms from the EPEL repository. It will install the packages needed for the Ansible AWS modules. As root from the installation directory run:

```
# ansible-playbook –i lsf-inventory Step0-setup-prereqs.yml
```

On successful completion the LSF master machine can now run the steps needed to create a VPC.

Create a VPC

This step will create a VPC, with associated Security Group, Subnets, routes, etc, from a minimal configuration file. If you have an existing VPC, it is possible to skip this step by taking the related information and populating the AWS-config.yml file.

Make a backup copy of the AWS/AWS-config.yml file. Edit the AWS/AWS-config.yml file, and set the appropriate values.

`AWS_Region:`
: Set this to the region you wish to deploy in

`AWS_Access_Key:`
: Set this to the Access Key for the AWS user account that is being used to deploy to the cloud. This is needed for the duration of the deployment. **Once the cluster is deployed on the cloud these values can be deleted.** Begins with: `AK`

`AWS_Secret_Key:`
: Set this to the Secret Key for the AWS user account that is being used to deploy to the cloud. This is needed for the duration of the deployment. **Once the cluster is deployed on the cloud these values can be deleted.**

`AWS_Instance_Type:`
: Set this to the size of the instance you want to create e.g. `t2.micro`

`AMS_Image:`

Set this to AMI ID for the image you want to deploy. The default is a CentOS 7 image e.g. `ami-77724e12`

`AWS_VPC_CIDR:`
Set this to the IPv4 address block you wish to use for the VPC. This address block must not overlap with any addresses on the on-premises network, or the VPN network, e.g. `10.1.0.0/16`

`AWS_VPC_PUB_CIDR:`
Set this to the IPv4 address block for the private network on EC2. This subnet must be inside the AWS_VPC_CIDR address block e.g. `10.1.0.0/24`

`CLIENT_NET: 10.10.10.0`
`CLIENT_MASK: 255.255.255.0`
Set these to the IPv4 network address and subnet mask for the on premises network that will be routed to the cloud servers. The LSF master must be part of this network. If Direct Connect is used this data is ignored.

`SERVER_IP: 10.0.11.1`
`SERVER_NET: 10.0.11.0`
`SERVER_MASK: 255.255.255.0`
These values are only used to control the VPN IP address of the on cloud instance providing the VPN. Make sure these values do not overlap with any other networks. If Direct Connect is used this data is ignored.

The following values need to be set when an existing VPC is to be used:

`AWS_VPC:`
Set this to the VPC ID, or leave it as none to have the playbook generate it

`AWS_VPC_PRV_Subnet:`
Set this to the Subnet ID of the private network of the EC2 instances, or leave it as none to have the playbook generate it.

`AWS_VPC_IGW:`
Set this to the Internet Gateway ID in the VPC, or leave it as none to have the playbook generate it.

`AWS_VPC_Routes:`
Set this to the VPC Routes ID, or leave it as none to have the playbook generate it.

`AWS_VPC_NACLs: none`

Set this to the VPC Network ACLs ID, or leave it as none to have the playbook generate it.

`AWS_VPC_Security_Group:`
Set this to the VPC Security Group ID to use, or leave it as none to have the playbook generate it.

`AWS_Key_Name:`
Set this to the name of the SSH key that was generated in IAM for the AWS user you are using to deploy the LSF cluster. If you do not have one, one will be generated. The associated ".pem" file should be downloaded and placed in the AWS directory.

After editing the file the VPC can be created by running the following:

 # ansible-playbook –i lsf-inventory Step1-make-vpc.yml

Once completed the resulting VPC and associated subnet, security group, routes, etc, can be inspected using the AWS console. The AWS/AWS-config.yml file will also have been updated with those values that were not provided in the initial file. .

Bring up the VPN

This step will create a VPN connection between the on premises network and the VPC private subnet. This step is not needed if you have Direct Connect, or have previously setup a VPN.

The VPN created by the Ansible playbook is for demonstration purposes. It will quickly and easily setup a VPN connection between sites, however it is not intended for a production configuration. Amazon Direct Connect or Amazon VPN would provide a more robust configuration.

This step will use the data populated in the AWS/AWS-config.yml file and VPN directory to create an EC2 instance, and configure it as a VPN server for other EC2 instances. It will also install the VPN software on the LSF master and configure it as a VPN client. To bring up the VPN run:

 # ansible-playbook –i lsf-inventory Step2-vpn-bring-up.yml

Once this step is complete the VPN will be setup. Look at the inventory-ec2.yml file to see the IP address of the VPN host. From the LSF master you

should be able to ping the private interface. As root you should also be able to ssh into the EC2 VPN server.

Gather local config

This step will gather the local configuration for users, groups and hosts and prepare it for the EC2 instances. To complete this step run:

```
# ansible-playbook –i lsf-inventory Step3-setup-env.yml
```

The step should complete without error. It will have added files to copy to the EC2 instances to the files/ directory.

Bring up EC2 instances for LSF

This step will start the EC2 instances that will be used to create the LSF cluster. How those instances will be used will depend on the type of cluster created. When run, the playbook will prompt the user with two questions:

1. Should the EC2 VPN instance also be used for LSF?
    - For a LSF Stretch Clusters the EC2 VPN instance will function as a LSF Server.
    - For LSF Multi Clusters the EC2 VPN instance will function as a LSF Master, with LSF Resource Connector configured

2. How many additional EC2 instances should be initially created?
    - For a LSF Stretch Clusters the additional EC2 instances will be used as LSF Servers and will run workload under control of the on premises LSF Master.
    - For a LSF Multi Cluster there must be one machine to host the LSF Master. If the EC2 VPN instance is not to be used, or some other VPN solution is provided, then one additional EC2 instance should be created.

The playbook will take the user input and create the necessary Ec2 instances and copy the configuration files from the previous step. It will also install prerequisites for LSF.

Configure Storage

This step is optional, and will configure the EC2 instances to mount on premises NFS or Spectrum Scale filesystems. This approach is primarily for demonstration purposes. Use of a Spectrum Scale filesystem requires installation of additional packages not covered by this installation guide.

Define the NFS storage to mount by editing the Storage-config.yml file. For each filesystem to mount add the following lines to the file:

```
- export: 10.10.10.10:/export
```

```
mountpnt: /nfs
type: nfs
args: defaults
```

Indentation is significant. Set the export value to the NFS servers IP followed by the export point. The mountpnt is where to mount the filesystem on the EC2 instances. The type will be nfs, and the args can be defaults, or other values. This data will be used to create the /etc/fstab entries.

> **NOTE:** All NFS servers must have a route to the EC2, and VPN subnet. The playbooks provided do not configure this.

After the Storage-config.yml is edited the playbook can be run with:

```
# ansible-playbook –i lsf-inventory Step5-access-storage.yml
```

If there are issues mounting the filesystem, such as routing, the playbook will hang when it attempt to mount the filesystem. Should this happen check that:

- The NFS server has a route to the EC2 subnet
- The NFS server is exporting the filesystem to the EC2 subnet

Once corrected the playbook can be re-run.

Install LSF

The type of LSF cluster to install is determined by the AWS/AWS-config.yml file, and was set earlier in the process. The cluster is deployed on the EC2 instances created earlier using:

```
# ansible-playbook –i lsf-inventory Step6-install-LSF.yml
```

Once complete the LSF Master will have an additional job queue to use to run workload on the cloud. On a LSF Stretch Cluster it is called “awsexample”. With LSF Multi Cluster it is called “send2cloud”

To test the LSF Stretch cluster use the following commands:

```
# lshosts
```

The output will look something like below. The machines designated as “awshost” are EC2 instances.

```
HOST_NAME      type    model  cpuf ncpus maxmem maxswp server RESOURCES
lsfmaster    X86_64   PC6000 116.1     4  31.2G   7.9G    Yes (mg)
ip-10-1-1-1  X86_64 Intel_EM  60.0     1   990M      -    Dyn (awshost)
```

```
ip-10-1-1-2  X86_64 Intel_EM  60.0     1    990M      -    Dyn (awshost)
```

To see the machines state in the LSF batch system run:

```
# bhost
```

The output may look something like:

```
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP    RSV
lsfmaster          ok              -      4      0      0      0      0      0
ip-10-1-1-161      ok              -      1      0      0      0      0      0
ip-10-1-1-222      ok              -      1      0      0      0      0      0
```

The host status should be “ok”.

For LSF Multi clusters check the masters connection by running:

```
# lsclusters
```

It will output something like:

```
CLUSTER_NAME   STATUS   MASTER_HOST            ADMIN    HOSTS  SERVERS
myCluster      ok       lsfmaster           lsfadmin        1        1
myCloudCluster ok       ip-10-1-1-99        lsfadmin        1        1
```

Login to the on cloud LSF master and run the commands above to see more details on the EC2 LSF cluster.

As a non-root user jobs can be submitted to the cluster e.g.

```
$ bsub –q {Queue name}  sleep 100
```

To see the jobs and where they are running use the **bjobs** command, or use the LSF Application Console. For example:

```
$ bjobs
```

Output may look like:

```
JOBID   USER    STAT  QUEUE      FROM_HOST  EXEC_HOST   JOB_NAME
SUBMIT_TIME
1293    billy   RUN   send2cloud lsfmaster  ip-10-1-1-99 sleep 3600 Aug
10 13:48
```

Troubleshooting

Should any of the playbooks fail to successfully complete it will be necessary to correct the error before going to the next step. Additional debug information can be had from the ansible playbooks by running them with the “-vv” argument e.g.

 # ansible-playbook –i lsf-inventory –vv …

This will show the files that the playbook is currently executing.

If the problem occurs configuring an EC2 instance they can be accessed via SSH using:
$ ssh -i /opt/ibm/Amazon/AWS/MyAmazonKeyPair.pem centos@{Public IP}

If the VPN is functional it may be possible to use the private IP to access the EC2 instance. The private IP can be found by looking at the contents of the inventory_ec2.yml file. The file has the EC2 id of the VPN node, as well as its public and private IPs. Try to access the private IP as root using:
 # ssh {Private IP}

Once the problem has been corrected the playbook can be re-run.

Operational Guidance

Health assessment of deployment

Use the “lsid”, “lshosts”, “bhosts” and “bqueues” commands to evaluate the health of the LSF Stretch Cluster.

LSF Multi Cluster clusters have an LSF master on premises, and one on EC2. Use the “lsclusters” command to see the state of the EC2 cluster e.g.

```
# lsclusters
```

It will output something like:

```
CLUSTER_NAME   STATUS   MASTER_HOST            ADMIN    HOSTS  SERVERS
myCluster      ok       lsfmaster           lsfadmin        1        1
myCloudCluster ok       ip-10-1-1-99        lsfadmin        1        1
```

Application Deployment

The LSF Stretch cluster will have access to the NFS filesystems listed in the Storage-config.yml file. At a minimum those filesystems should contain the application binaries to run, and ideally the users home directories for application data. Selecting the workloads to run on the EC2 instances is important.

The LSF Multi Cluster as deployed by the playbook has access to the NFS filesystems defined in the Storage-config.yml file, however it is not desirable to

host applications this way. A better approach would be to move the applications on to the cloud. There are several ways to accomplish this:

- Create an image with the applications already loaded
- Use a storage rich instance and host the applications on that
- Use EFS, or EBS to provide an NFS filesystem. LSF users have reported performance issues with large clusters (>500 instances) using EFS.

Periodic Maintenance

The LSF installation on the EC2 instances uses NFS hosted on either the first LSF instance in the cloud. This NFS directory is mounted to /opt/ibm on the other LSF EC2 instances. Care should be taken so that it does not run out of space. The following directories are important for LSF and may needed to be periodically cleaned:

- /opt/ibm/lsfsuite/lsf/log
- /opt/ibm/lsfsuite/lsf/work/myCloudCluster/logdir

If the playbooks that have been provided are taken and extended it becomes an easy matter to takedown and recreate the EC2 clusters.

Decommissioning

A procedure for safely decommissioning the cluster is detailed above in the "Best Practices for using IBM Spectrum LSF on AWS" section. Alternatively the cluster can be deleted via the EC2 Management Console.

The LSF Stretch and Multi-clusters can be deleted from the AWS console. Before deleting the cluster the jobs using those resources should be allowed to finish, otherwise there will be some job failures.

From the AWS console go to the EC2 Dashbord. From here go to the Running Instances and delete them, including the VPN host if it is no longer needed. Once they are deleted, go to the Volumes and delete all of the volumes that are no longer "in use".

The VPC can also be deleted if it is no longer needed. From the AWS Console navigate to the VPC Dashboard. Navigate to the VPC list, and delete the "My_LSF_VPC". This will delete the subnet, routing table, Internet gateway, and security group associated with the VPC. It will also be necessary to restore the AWS/AWS-config.yml file, so subsequent runs of the playbook do not use incorrect values.

Security

AWS Identity and Access Management (IAM)

This solution leverages an IAM role with minimal privileged access. It is not necessary or recommended to store SSH keys, secret keys, or access keys on the provisioned instances.

OS Security

The root user on instances in the cloud can be accessed only by ysing the SSH key specified during the deployment process. AWS doesn't store these SSH keys, so if you lose your SSH key you can lose root access to these instances.

Operating system patches are your responsibility and should be performed on a periodic basis. AWS offers a service called "Systems Manager Patch Manager" (see https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html).

Security Groups

A security group acts as a firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. The new rules are automatically applied to all instances that are associated with the security group.

The security groups created and assigned to the individual instances as part of this solution are restricted as much as possible while allowing access to the various functions needed by IBM Spectrum LSF. We recommend reviewing security groups to further restrict access as needed once the cluster is up and running. The Quick Start creates the following security groups for IBM Spectrum LSF:

AWS_VPC_SEC_GRP
Inbound:

1. Allow ICMP from 0.0.0.0/0
2. Allow SSH from 0.0.0.0/0
3. Allow UDP to 1194 from 0.0.0.0/0
4. Allow all from on premises network
5. Allow all from VPN subnet
6. Allow all from VPC subnet

Outbound: Allow all

Network ACLs

A network ACL acts as a firewall for the subnet it is associated with. If none is provided a network ACL will be created with the following settings:
LSF_VPC_NACL

Inbound:

1. Allow ICMP from 0.0.0.0/0
2. Allow SSH from 0.0.0.0/0
3. Allow UDP to 1194 from 0.0.0.0/0
4. Allow TCP ports 32768-65545 from 0.0.0.0/0
5. Deny All

Outbound:

1. Allow all to 0.0.0.0/0
2. Deny all

IAM Roles & Purpose

The Quick start guide assumes that a user has been created for the purposes of deploying the cluster. This user will need to be able to create and destroy EC2 instances as well as create and destroy VPCs and associated security groups, network ACLs, subnets, etc. The root account should not be used.

Client Data Security

IBM takes our client's security, and most and especially our client's intellectual property security extremely seriously.

As such, we felt it important to emphasize that the Quick Start solution is not meant for production use, and therefore does not include data at rest and data in motion (between nodes in the cloud) encryption. We feel this is important because often IBM Spectrum LSF is used to manage applications which are operating upon crucial intellectual property.

We strongly urge you to not make use of this evaluation with production data so as to minimize or negate the risk of data leakage or theft while running this evaluation.

Further, data security in the deployed environment is your responsibility. 3rd party (data) intellectual property may have restrictions on locations where it may be stored, methods for transfer, etc all of which are also your responsibility to understand and comply with during the use of this evaluation.

Amazon provides best practices for performing EBS encryption here. Server side encryption and S3 Managed Keys

Risk Auditing

Though beyond the scope of the functionality of this deployment guide, Amazon provides tools like CloudTrail and S3 Access Logs (for people using object storage) which can be very useful in performing risk audits.

Backup & Recovery

This Quick Start for automatically deploying IBM Spectrum LSF in the Amazon Cloud (EC2) does **not** include any facilities for backing up or recovering the evaluation environment after failure.

As an evaluation, should there be some kind of datacenter outage, the expectation is that the Quick Start could be run in a separate availability zone and reconstituted in default form in short order. In that scenario, LSF configuration customizations, data uploaded, applications installed, users and security groups customized and defined would all be lost and need to be recreated to return to the state of the evaluation at the time of the outage.

Amazon provides facilities for backup and recovery, see this link.

Additional Resources

AWS Services

Amazon EC2 -
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

Amazon VPC – https://aws.amazon.com/documentation/vpc/

IBM Spectrum LSF
https://www.ibm.com/support/knowledgecenter/en/SSWRJV_10.1.0/lsf_welcome/lsf_welcome.html

Quick Start reference deployments
AWS Quick Start home page – https://aws.amazon.com/quickstart/

Send Us Feedback

For any feedback on this document or your experience, please post something to our discussion forum on IBM Developerworks. Here.

Document Revisions

September 2018	Initial Publication