NODUS Cloud OS
User Guide 4.0.1

April 2020

Adaptive
COMPUTING

Contents

Welcome

Welcome to the NODUS Cloud OS User Guide. This guide will show you how to use the NODUS Cloud OS user interface.

NODUS Cloud OS for intelligent cloud management gives immediate access to all computational resources, whether on-premises or in the cloud, on any leading cloud provider. This highly flexible and customizable solution enables HPC or enterprise systems to 'burst' the additional workload to an external cloud on demand. NODUS Cloud OS includes all the necessary tools to facilitate moving HPC and enterprise workloads and applications to the cloud and/or extending on-premises resources.

Legal Notices

Adaptive Computing Enterprises, Inc.
1100 5th Avenue South, Suite #201
Naples, FL 34102
+1 (239) 330-6093
www.adaptivecomputing.com

Revision History

Date	Release
February 2018	Moab/NODUS Cloud Bursting 1.1.0
November 2018	NODUS 3.0.0
July 2019	NODUS Cloud OS 3.2.0
October 2019	NODUS Cloud OS 3.2.3
December 2019	NODUS Cloud OS 4.0.0
April 2020	NODUS Cloud OS 4.0.1

Chapter 1: NODUS Cloud OS Prerequisites

Complete the steps below to ensure you are ready to start using NODUS Cloud OS.

NODUS Cloud OS can be installed on these operating systems:

- CentOS 7
- Oracle 7.7
- Red Hat 7

Note: Refer to the chosen operating system's product specifications for minimum requirements.

1. From the computer where the job submission will be made, test your internet access from a browser by going to `http://<web_server_address>` and ensure that you can get to the NODUS Cloud OS home page. Also make sure that your network is configured to allow HTTP requests.
2. Make sure you have the proper cloud credentials and authorization to provision instances in the cloud for your preferred cloud providers. See the appendices Creating Accounts for NODUS Cloud OS Providers and Obtaining Provider Account Information and Credentials for information.
3. Make sure you are using one of these supported browsers:
   - Google Chrome (latest version)
   - Mozilla Firefox (latest version)
   - Microsoft Edge (two latest major versions)
   - Apple Safari (two latest major versions)
4. Make sure the device running the API (e.g., your in-house CentOS 7 NODUS Platform Server) has the following ports open:
   - 22 (SSH) - user direct access / Moab job routing
   - 80 (HTTP) - connection to the NODUS UI
   - 443 (HTTPS) - optional, if you secure your HTTP connection
   - 12345 (NODUS-web-API) - the NODUS Platform API server that your UI will communicate with

Packages Needed for the Installation

The following packages are installed via yum during the installation process; they may already exist on the server, but installation of these packages will be attempted anyway:

- perl - The Perl Programming Language
- epl-release - Extra Packages for Enterprise Linux
- nginx - Open-source web server
- ansible - SSH-based configuration management, deployment, and task execution system
- python3 - Programming language, Version 3
- pip3 - The Python Package Installer
- openssh - An open source implementation of SSH protocol Versions 1 and 2
- openssh-clients - An open source SSH client application
- openssh-server - An open source SSH server daemon
- openssl-libs - A general purpose cryptography library with TLS implementation
- net-tools - Basic networking tools
- vim - The VIM version of the vi editor for the X Window System
- git - Fast Version Control System
- gcc-c++ - C++ support for GCC
- make - A GNU tool that simplifies the build process for users
- unzip - A utility for unpacking zip files
- jq - Command-line JSON processor
- dos2unix - Text file format converters
- libtool - The GNU Portable Library Tool
- wget - A utility for retrieving files using the HTTP or FTP protocols
- crontabs - Root crontab files used to schedule the execution of programs
- nodejs - JavaScript runtime

The NODUS repositories installed will also run an `npm install` to install the node modules to enable NODUS to function. The definition of these products can be found in the `package.json` of each repository in the `/NODUS/` directory.

Chapter 2: NODUS Cloud OS Platform Installation

This chapter provides information about the installer, which contains the entire NODUS Cloud Platform, along with dependencies to install it, and setup scripts to simplify/automate the process.

In this chapter:

2.1 Installation Steps

In the following steps, replace x.x.x with the current release (e.g., 4.0.1).

1. Create a user account on the system where NODUS will be installed. **Note:** This account must have sudo access.
2. Login and transfer the following file to the user account using this command: **Nodus_Cloud_Platform-x.x.x.tgz**
3. Extract the file using this command: **tar xf Nodus_Cloud_Platform-x.x.x.tgz**
4. Run the following installation scripts:

   cd Nodus-x.x.x/

   ./osname-install.sh

   Where `osname` is the operating system name in lowercase.

   For Example

   cd Nodus-4.0.1/

./redhat7-install.sh

Depending on your system, the installation may take up to 45 minutes to complete. If it does not complete successfully, refer to the **install.log** for troubleshooting.

5. When prompted, enter the password.
6. Continue with the section Obtaining Credentials from Providers.

2.2 Verifying the Installation

1. Verify that the API is running by entering **sudo pm2 status**. `nodus4-web-api name` should display with the status as `online`. If not, enter **/NODUS/nodus4-web-api/start.sh**.
2. Check the UI configuration files by editing `/NODUS/nodus4-ui-desktop-app/build/assets/config.json`. This file tells the UI the address and port of the API and which providers will be used. You can edit the providers to only show which providers you will supply licenses to. The file will look like this:

```
{
 "server": "http://<server Address>:12345",
 "providers": ["aws", "google", "azure", "oracle", "huawei",
"otc"]
}
```

3. Verify the API in a browser by entering **http://<server Address>:12345/**. The browser should display `{"reason":"unknown","message":"Not Found"}`. If not, rerun step 1 and/or step 2.
4. Verify the UI in a browser by entering **http://<server Address>/**. You should be prompted to enter a license or to log in. If not, rerun step 1 and/or step 2.

2.3 Starting the Web API

Complete the steps below to start up the Web API at boot.

1. Edit the file `/etc/rc.local` by adding the line `/NODUS/nodus4-web-api/start.sh`.
2. Enter the command **sudo chmod +x /etc/rc.d/rc.local**.

2.4 Modifying the Code

Modification of the code without the approval of Adaptive Computing is discouraged as this may result in an unstable environment.

2.5 Obtaining Credentials from Providers

Continue with the steps for your chosen provider below.

Microsoft Azure

1. Run the script **azure-setup.sh**.

Oracle Cloud

1. On the Oracle account to be used, create a compartment.
2. Within that compartment, create a VPC.
3. Within that VPC, create at least one subnet.

See https://docs.cloud.oracle.com/iaas/Content/home.htm for additional information.

Huawei Cloud

1. On the Huawei account to be used, create a VPC.
2. Within that VPC, create a subnet.

See https://support.huaweicloud.com/intl/en-us/vpc/index.html for additional information.

Open Telekom Cloud

1. On the Open Telekom Cloud (OTC) account to be used, create a VPC.
2. Within that VPC, create a subnet.

See https://open-telekom-cloud.com/en/products-services/virtual-private-cloud for additional information.

Continue to the next section.

2.6 Credentials Manager

Credentials are required by the cloud providers in order to access their environment and resources. Adding and saving them now means that you can easily select them with just one click at any time in the future when creating a cluster.

1. To add credentials, on the **Applications** menu, click **Credentials Manager**. The **Credentials Manager** screen shows this information: Name, Provider, Path, Date/Time Created, and Actions.
2. Select the appropriate cloud provider and enter the desired name for the credentials.
3. Enter your account credentials. Alternatively, click **Upload Credential** (see the section Credentials File Templates).
4. Click **Save** to finish adding the credentials for this provider. Repeat the steps in this section to add credentials for another provider if desired.
5. Continue with the section Cluster Manager to create a new cluster.

2.6.1 Credentials File Templates

Below are credentials file templates for the cloud providers:

- AWS
- Google Cloud
- Microsoft Azure
- Oracle Cloud
- Huawei Cloud
- OTC

AWS

```
{
"access_key": "",
"secret_key": "",
}
```

Google Cloud

```
{
```

```
"type" : "",
"project_id" : "",
"private_key_id" : "",
"private_key" : "",
"client_email" : "",
"client_id" : "",
"auth_uri" : "",
"token_uri" : "",
"auth_provider_x509_cert_url": "",
"client_x509_cert_url" : ""
}
```

Microsoft Azure

```
{
"client_id": "",
"client_secret": "",
"subscription_id": "",
"tenant_id": "",
}
```

Oracle Cloud

```
{
"user_ocid": "",
"tenancy_ocid": "",
"compartment_ocid": "",
"availability_domain": "",
"key_file": "", // oci api key (generated and uploaded into Oracle account)
"fingerprint": "" // fingerprint of oci api key
}
```

Huawei Cloud

```
{
"username": "",
"password": "",
"domain_name": "",
"tenant_name": "",
}
```

OTC

```
{
"username" : "",
"password" : "",
"domain_name": "",
"tenant_name": "",
}
```

2.7 Stack File Templates

Below are stack file templates for the cloud providers:

- AWS
- Google Cloud
- Microsoft Azure
- Oracle Cloud
- Huawei Cloud
- OTC

AWS

```
{
"provider" : "aws",
"nodus_image_prefix": "",
"region" : "us-east-1",
"instance_type" : "t2.micro",
"source_ami_name" : "*CentOS Linux 7 x86_64 HVM EBS*",
}
```

Notes:

provider	[required, must be 'aws']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of region that the VM will be in]
instance_type	[required, instance type used to provision and create the image on]
source_ami_name	[required]

Google Cloud

```
{
"provider" : "google",
"nodus_image_prefix" : "",
"region" : "us-east1",
"zone" : "us-east1-b",
"instance_type" : "n1-standard-2",
"source_image_family": "centos-7",
}
```

Notes:

provider	[required, must be 'google']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of region that the VM will be in]
zone	[required, name of zone that the VM will be in]
instance_type	[required, instance type used to provision and create the image on]
source_image_family	[required, name of existing image family to be used as the base]

Microsoft Azure

Account Preparations

- Create Storage Account
- Create Permissions
- Create Resource Group

Marketplace Image

```
{
"provider" : "azure",
"nodus_image_prefix" : "",
"region" : "eastus",
"instance_type" : "Standard_DS2_v2",
"source_image_publisher" : "openLogic",
"source_image_offer" : "CentOs",
"source_image_sku" : "7.5",
"nodus_resource_group_name": "nodus-eastus",
}
```

Notes:	
provider	[required, must be 'azure']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of region that the VM will be in]
instance_type	[required, instance type used to provision and create the image on]
source_image_publisher	[required, specific to the marketplace image]
source_image_offer	[required, specific to the marketplace image]
source_image_sku	[required, specific to the marketplace image]
nodus_resource_group_name	[required, name of the existing resource group where images will be placed (can be found at https://portal.azure.com/)]

Custom Image

```
{
"provider" : "azure",
"nodus_image_prefix" : "",
"region" : "eastus",
"instance_type" : "Standard_DS2_v2",
"source_resource_group_name": "",
"source_image_name" : "",
"ssh_private_key_file" : "",
"nodus_resource_group_name" : "",
}
```

Notes:	
provider	[required, must be 'azure']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of region that the VM will be in]
instance_type	[required, instance type used to provision and create the image on]
source_resource_group_name	[required, name of the existing resource group where source images exist]
source_image_name	[required, name of the existing image to be used as the image base]
ssh_private_key_file	[required, content of the ssh key]
nodus_resource_group_name	[required, name of the existing resource group where images will be placed]

Oracle Cloud

```
{
"provider" : "oracle",
"nodus_image_prefix" : "",
"region" : "us-ashburn-1",
"availability_domain": "",
```

```
"instance_type" : "VM.Standard2.2",
"base_image_ocid" : "",
"subnet_ocid" : "",
"nodus_key_file_path": ""
}
```

Notes:

provider	[required, must be 'oracle']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of region that the VM will be in]
availability_domain	[required, name of the availability domain that the VM will be in (tenant specific hash prefix ":" availability zone)]
instance_type	[required, instance type used to provision and create the image on]
base_image_ocid	[required, ID of the existing image to be used as the base]
subnet_ocid	[required, ID of the existing subnet (can be found at https://www.oracle.com/cloud/sign-in.html)]
nodus_key_file_path	[optional, appended by the NODUS base image build]

Huawei Cloud

```
{
"provider" : "huawei",
"nodus_image_prefix": "",
"region" : "ap-southeast-1",
"availability_zone" : "ap-southeast-1a",
"source_image" : "CentOS 7.6 64bit",
"flavor" : "s2.large.2",
"network" : "",
}
```

Notes:

provider	[required, must be 'huawei']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of region that the VM will be in]
availability_zone	[required, name of zone that the VM will be in]
source_image	[required, name of existing image to be used as the base]
flavor	[required, flavor used to provision and create the image on]
network	[required, ID of existing subnet (can be found at https://intl.huaweicloud.com/en-us/?locale=en-us/)]

OTC

```
{
"provider" : "otc",
"nodus_image_prefix": "",
"region" : "eu-de",
"source_image" : "Standard_CentOS_7_latest",
"flavor" : "s2.medium.1",
"network" : "",
}
```

Notes:

provider	[required, must be 'otc']
nodus_image_prefix	[optional, prefix all created images with (i.e., "test-"; suggest ending in hyphen)]
region	[required, name of the region that the VM will be in]
source_image	[required, name of the existing image to be used as the base]
flavor	[required, flavor used to provision and create the image on]
network	[required, ID of the existing subnet (can be found at https://console.otc.t-systems.com/console/)]

2.8 Images

2.8.1 Building Stacks into Images

The only time you need to build stacks is when switching your cloud credentials (AWS access/secret key), cloud provider, and/or moving regions within AWS (us-east -> us-west). You can see the list of stack images that have been built for a particular region in the AWS console (https://aws.amazon.com/) under the Images section of the sidebar.

If you just want to deploy another cluster in the same region using the same credentials, then stack rebuilds should not be necessary.

By default, NODUS images are built on top of the marketplace image CentOS 7. Alternatively, it can be configured to build NODUS images on top of an existing image. **Note:** See https://wiki.centos.org/About/Product for minimum requirements for CentOS 7.

1. Edit the credential file for the specific provider, enter the desired information, and then save as **<credential name>.json** (see the section Credentials File Templates for details).
2. Edit the stack file for the specific provider, enter the desired information, and then save as **<stack name>.json** (see the section Stack File Templates for details).
3. Edit the cluster file for the specific provider to meet the requirements for the head node and compute nodes, define a unique name for the cluster, and then save as **<cluster name>.json** (see the section Cluster File Templates for details).
4. Add credentials from step 1: **nodus credentials add <credential name>.json**.
5. Add the stack file from step 2: **nodus stack add <stack name>.json**.
6. Add the cluster from step 3: **nodus cluster add <cluster name>.json**.
7. Associate the credentials file with the cluster: **nodus cluster set-credentials <cluster name> <credential name>**.
8. Associate the stack file with the cluster: **nodus cluster set-stack <cluster name> <stack name>**.
9. Create the image: **nodus stack build <cluster name in file>**.

2.8.2 Build Order

Images are built behind the scenes in this order:

0. [user-image]

1. nodus-centos-7

2. nodus-server

3. nodus-node

Note: 0. [user-image] is optional.

A successful build should end with the following:

```
Build 'NODUS Image Builder - nodus-centos-7-1566325733 - ' finished.
==> Builds finished. The artifacts of successful builds are:
--> NODUS Image Builder - nodus-centos-7-1566325733 - : AMIs were created:
us-east-1: ami-07a14470d92be82a2
/home/name/.nodus/resources/aws/cluster/aws-cluster/nodus-centos-7.image
```

Chapter 3: Using NODUS Cloud OS

This chapter provides information about using NODUS Cloud OS to run your workloads in the cloud.

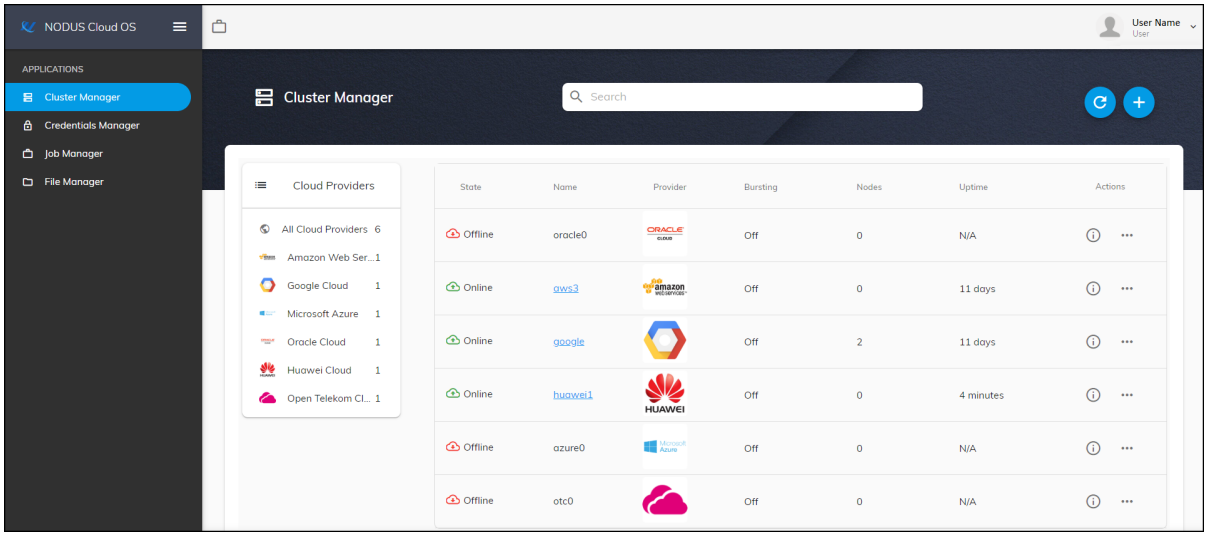
In this chapter:

3.1 Create a NODUS Account

1. Go to `http://<web_server_address>/register` and create an account.
   Note: For most installations, nodusplatform.com refers to the server that the NODUS platform is installed on, for example, the Web server address of the machine that you installed NODUS on.
2. Follow the instructions and complete the registration process. At the end of this step, you will have your NODUS Cloud OS user ID and password. Write them down and/or store them in a secure location. Upon successful registration, you will receive a confirmation email from nodus@adaptivecomputing.com and be redirected to the login screen at `http://<web_server_address>`.

3.2 Cluster Manager

1. Login to your NODUS Cloud OS user account. The main NODUS **Cluster Manager** screen appears:



On this screen, you can configure new clusters and see a list of all your current clusters and information such as their State, Name, Provider, Bursting, Nodes, Uptime, and Actions. From the **Applications** menu, you can also add credentials, submit jobs, and view a job's output file.

Notes:

- Clicking **SSH Key** next to your user name will download the SSH key to gain SSH access to the cluster. A '.pem' file is downloaded and can be used to access the head node of your clusters via SSH. This provides you full control over the cluster.
- Clicking the refresh icon ( ) at the top right of the cluster list, refreshes the list.
- Clicking the add icon ( ) at the top right opens the **New Cluster** screen.
- Clicking the name of a cluster shows this information:
  - **Cluster Info** - Jobs Submitted (Today, This Month, This Year, Blocked Jobs), Queue (Running, Queued, Total Jobs), Nodes (Available, Busy, Down, Offline, Total Nodes), and Activities Log (All, Job).
  - **Queue** - ☐ (Cancel), Name, State (All Jobs, Running, Queued, Completed), Time, Node Count, and Executing Nodes.

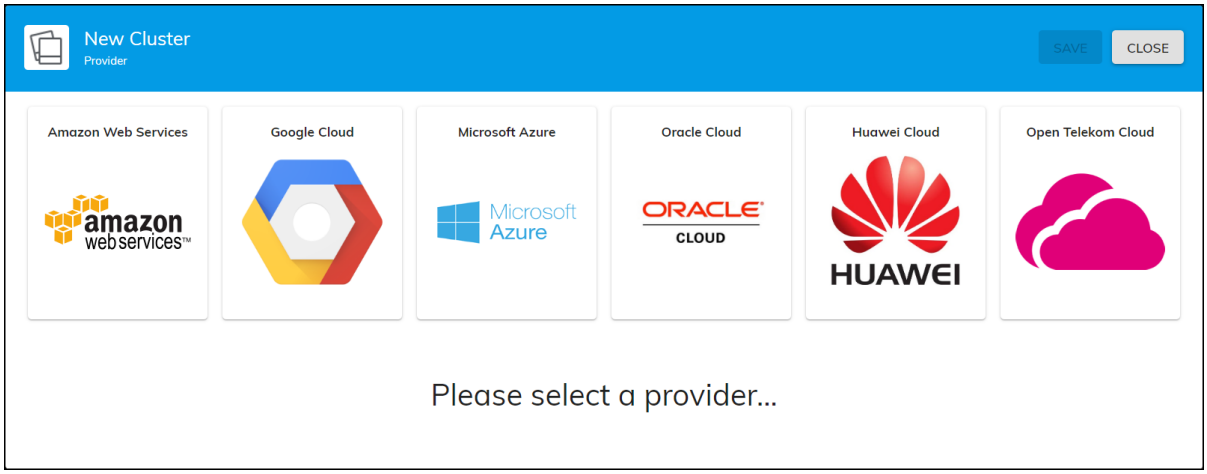
  - **Nodes** - ☐ (Deploy, Destroy), Name, State (All Nodes, Available, Busy, Offline, Down), Cores, and Available Threads.
  - **Configuration** - Cluster ID, Head Node Size, Cluster IP, Date/Time Created, Region, and Availability Zone.

- Clicking the information icon (**i**) at the end of the row shows additional details about the cluster, including: Cluster ID, Date/Time Created, Expected Provisioning Time, Actual Provisioning Time, Uptime, Head Node Public IP, Head Node Size, Compute Node Sizes, and Region. The buttons **Deploy** and **Logs** also show.
- Clicking the horizontal ellipsis (⋯) at the end of the cluster row shows actions that you can take regarding the cluster:
  - **Deploy** - Deploys the cluster into an online state.
  - **Destroy** - Destroys the head node and the linked compute nodes so that they are no longer in a state to do work. This also means that you will not be billed further for these cloud resources.
  - **Logs** - Shows additional information regarding the provisioning. A screen shows information in real-time about the cluster's state. Additionally, these logs contain information for deploy, redeploy, resize, and destroy.
  - **Delete** - Permanently deletes the cluster. **Note:** The cluster must be destroyed first before it can be deleted.

3.2.1 Create a Cluster

If you have already added your cloud credentials for the chosen provider, follow the steps below to configure a new cluster. If you have not yet added your credentials, you must first follow the steps in the section Credentials Manager.

1. To create a new cluster, on the **Cluster Manager** screen, click the add icon (⊕) to open the **New Cluster** screen:



   These are the supported cloud providers:

   - Amazon Web Services (AWS)
   - Google Cloud
   - Microsoft Azure
   - Oracle Cloud
   - Huawei Cloud
   - Open Telekom Cloud (OTC)

   Note: For unsupported cloud providers, please contact nodus@adaptivecomputing.com to inquire about adding the provider into NODUS Cloud OS.

2. Select the cloud provider that you want to deploy a cluster in.
3. Enter the required/optional information: Name, Credentials, Region, Availability Zone, Server Size, Compute Node Instance Sizes, Node Size, Nodes Count, Persistent/Bursting, and Description.
   Note: Multiple Compute Node Instances of different sizes can be added. The available Regions, Availability Zones, Server Instance Sizes, Compute Node Instance Sizes, and quantity of instances may vary based on your provider account status.

4. Click **Save**.
5. Repeat the steps in this section to create additional clusters as desired.
6. To deploy a cluster, click the ellipsis and then click **Deploy**.
7. Continue to the section Job Manager to run a job.

3.3 Job Manager

You can add jobs and submit them to either an existing cluster or an on demand cluster by using Job Manager.

1. To add a job, on the **Applications** menu, click **Job Manager**. The **Job Manager** screen shows this information: Name, Nodes, Cores, Data Files, Walltime, Submit, and Delete.
2. Click the add icon () at the top right to open the **New Job** screen.
3. On the **Job Info** panel, enter the required information: Name, Walltime, Number of Nodes, Number of Cores per Node, and Description. This can be edited after the job is saved by clicking the name of the job.
4. On the **Script** panel, either edit the script for this job or click **Upload Script**. This can be edited after the job is saved by clicking the name of the job.
5. On the **Data Files** panel, click **Upload Data** to upload any data files that the job may require. This can be edited after the job is saved by clicking the name of the job.
6. Click **Save** to finish adding the job.
7. To submit a job, click the submit icon () and then select a cluster to run the job on.

Note: You can also submit one of these two sample jobs:

- **test-job**: A job that is best used to test bursting functionality.
- **mpi-benchmarks**: A job that tests performance of the cluster.

After selecting a sample job, you can then select one of these cluster types:

- **Named Cluster**: If there are no created clusters or ones that meet the hardware requirements of the job, then certain named clusters may not appear.
- **On Demand**: A new temporary cluster that is tailored to the parameters of your job will be deployed to run your job and then destroyed when the job completes. These are the **On Demand Types**:

- **Destroy Full Cluster** - The full cluster is destroyed including the head node.
- **Destroy Compute Nodes** - The head node stays active and the compute nodes are destroyed.
- **Offline Compute Nodes** - The head node stays active and the compute nodes go offline.

---

8. Click **Submit**. The job is queued and runs on the selected cluster. The cluster details screen shows the submitted job information:

   - **Cluster Info** - Jobs Submitted (Over All, Today, This Month, This Year), Queue (Running, Queued), Nodes (Available, Busy, Down, Offline), Activities Log (All, Job), and Nodes/Workload graph (see below).

   This graph shows the status of Available Nodes, Busy Nodes, & Down Nodes, and the status of the Running Jobs, Queued Jobs, & Blocked Jobs. Click to deselect nodes and jobs to narrow the display results. Hover over the graph to see the number of nodes or jobs.

   - **Queue** - ID, Name, State, Time, Node Count, Executing Nodes, and Actions (⊗). **Note:** After a job completes, it only shows here temporarily, but can be viewed in File Manager.
   - **Nodes** - ☐ (Deploy, Destroy), Name, State (All Nodes, Available, Busy, Offline, Down), Cores, and Available Threads, and Actions (Deploy, Destroy).
   - **Configuration** - Cluster Information (Cluster ID, Head Node Size, Cluster IP, SSH Username, Date/Time Created, Region, Availability Zone), Resize (Resize Cluster),

and Bursting Service (View Logs, Burst Once & Enable Service / Disable Service).
Note: See the section Bursting Configurations for information about bursting.

Note: Jobs can also quickly be run from any screen at any time by clicking the **Quick Launch** icon ( ).

3.4 File Manager

You can view or download a job's standard output file or error file using **File Manager**.

1. On the **Applications** menu, click **File Manager**, and then from the **Select Cluster** drop-down list at the top right, select a cluster.
   The **File Manager** screen shows this information: ID, Name, Type, Owner, Size(B), Modified, and Actions.
2. Click the ellipsis at the end of a row and select the desired option: View STD-Out, Download STD-Out, View STD-Err, Download STD-Err, or Delete.

3.5 Bursting Service

3.5.1 Bursting Configurations

The NODUS burst function detects what jobs are in the queue and automatically spins up, takes offline, or shuts down nodes depending on the total requirements for the queue. If there are not enough online nodes to run all jobs, bursting will bring on as many nodes as needed. If there are more nodes than needed, the excess nodes will be taken offline. If the job queue is empty, all nodes will be shut down after a specified period of time.

Min Burst spins up the minimum number of compute nodes required to complete all jobs in the queue, which is ideal for budgeting and controlling cloud costs.

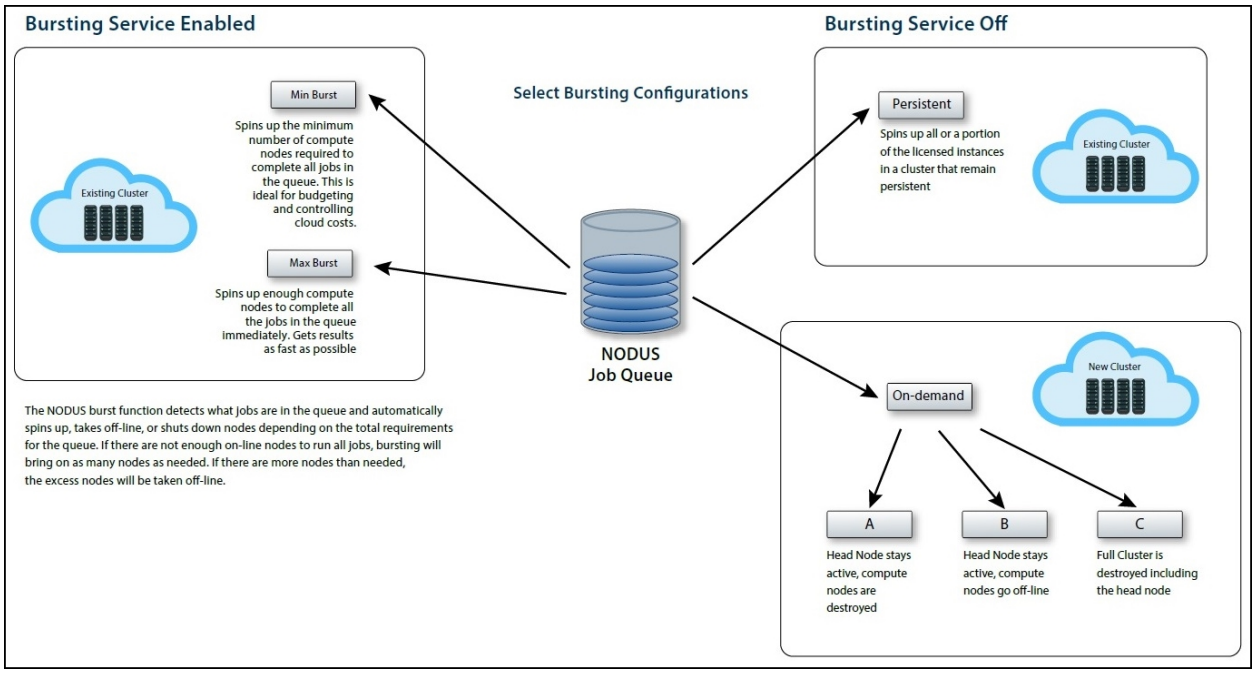
Max Burst spins up enough compute nodes to complete all the jobs in the queue immediately; this gets results as fast as possible. Max Burst is limited by the size of the cluster and will not create new nodes.

Persistent bursting spins up all or a portion of the licensed instances in a cluster that remain persistent for a period of time and brings nodes online or shuts them down as needed.

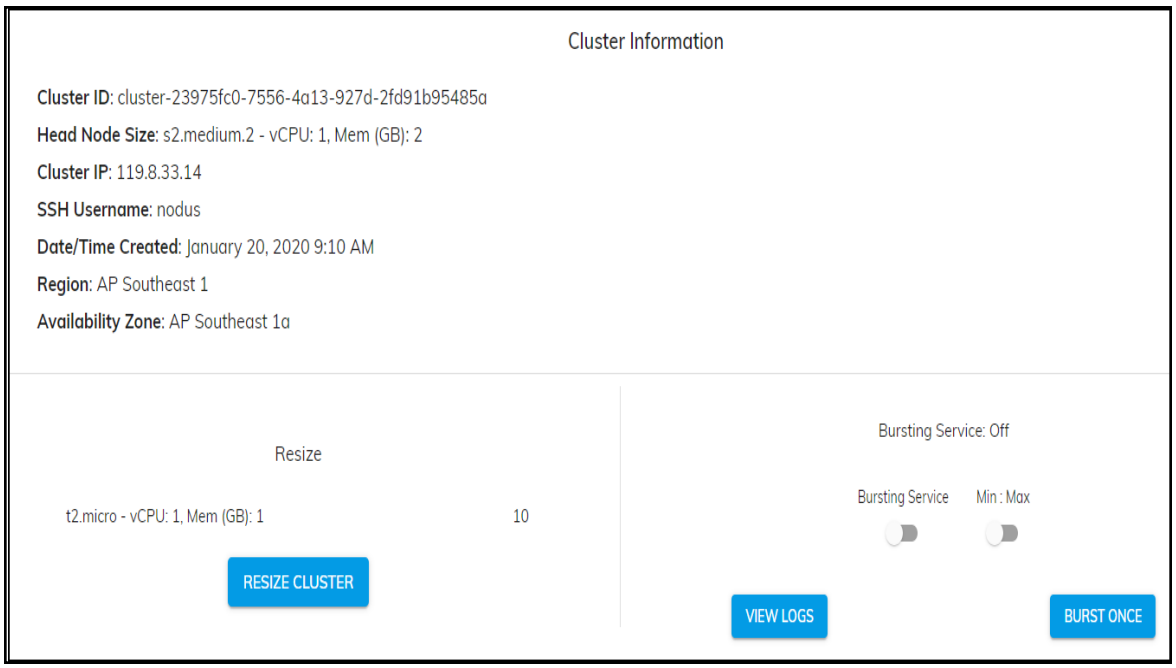
On demand bursting spins up the number of nodes required to run one job now; this is an isolated cluster, not for sharing with other jobs. The on demand types are: Destroy Compute Nodes (the head node stays active and the compute nodes are destroyed), Offline

Compute Nodes (the head node stays active and the compute nodes go offline), and Destroy Full Cluster (the full cluster is destroyed including the head node).

See the diagram below for details.



To manage cluster cloud bursting configurations, on the **Cluster Manager** screen, select a cluster and click the **Configuration** tab. The **Cluster Information** screen shows the following information:



Bursting Service - Disable or enable bursting functionality for a single cluster:

- **Disabled** - The cluster remains in its current state.
- **Enabled** - Jobs are run and clusters are provisioned to handle the jobs:
  - **Min** - Min Burst spins up the minimum number of nodes required to complete all jobs in the queue. This is ideal for budgeting and controlling cloud costs.
  - **Max** - Max Burst spins up enough nodes to complete all the jobs in the queue immediately. This gets results as fast as possible.

Burst Once - This spins up or tears down nodes as required to complete all the jobs in the queue. Select a one time bursting size - Min or Max.

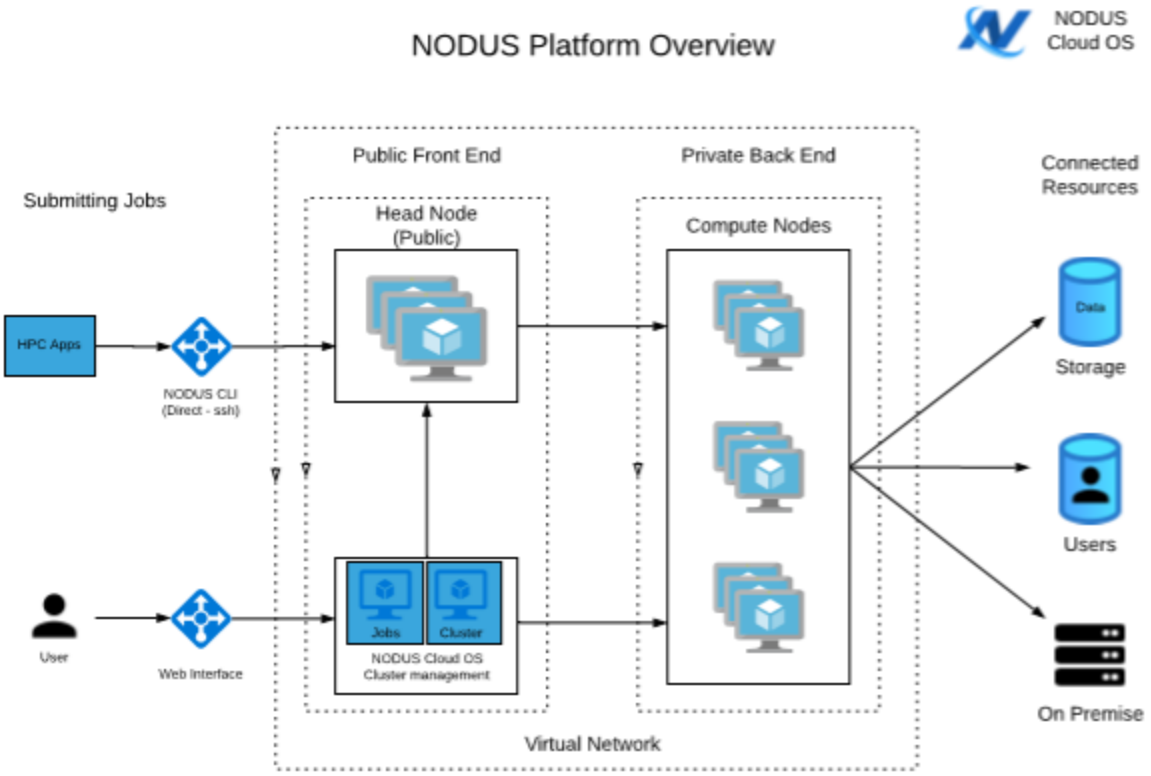
3.5.2 Disaster Recovery

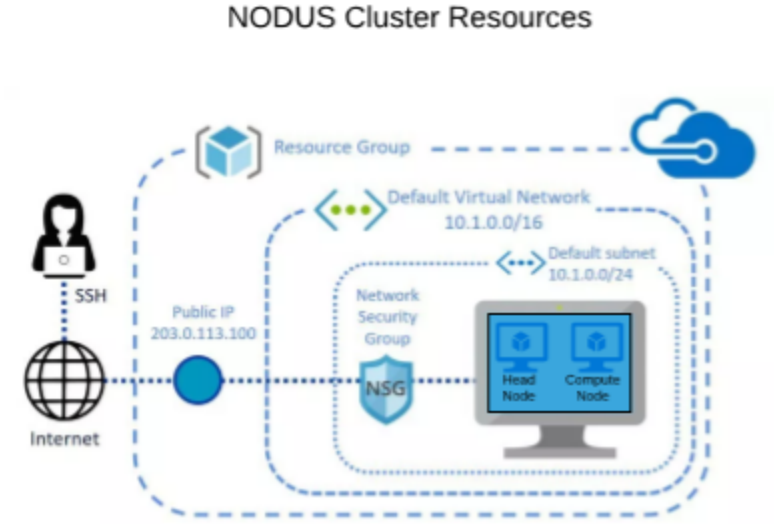
NODUS helps facilitate disaster recovery by allowing users to move workloads to different regions within the same cloud provider or to another cloud provider, automatically or manually.

3.6 Cloud Budget Analysis

Now that you have successfully run one job in the cloud by completing the steps above, you can run all your jobs in the cloud and begin developing your cloud budget. See the appendix NODUS Cloud OS Considerations for additional information.

Chapter 4: NODUS Cloud OS Platform Overview and Cluster Resources





Chapter 5: NODUS Cloud OS Command Line

After NODUS is installed, set up the command line interface (CLI) by following the steps below.

In this chapter:

5.1 Setup

1. Ensure the NODUS CLI is available by running **nodus --help** from a terminal.

   If this is unsuccessful, run the following commands to link the CLI:

   cd /NODUS/nodus-cli

   npm i

   sudo npm link

   nodus --help

2. Set the location of nodus-cluster-providers for the CLI to use (this folder is located in /NODUS by default):

   nodus packages add nodus-cluster-provider /NODUS/nodus4-cluster-providers

5.2 Cluster Actions

5.2.1 Deploying a Cluster

1. Obtain credentials from the cloud provider you intend to deploy this cluster on.
2. Create a cluster file for the cluster you are deploying.
3. Add the cluster via **nodus cluster add <cluster.json> [credentials.json]**. Cluster add optionally lets you add your credentials when you add a cluster. If you don't, you can still set them using **nodus user set cluster/cluster-name/credentials**.
4. Add credentials for the cluster using **nodus user set cluster/<cluster-name>/credentials credentials.json**, providing the full path to the file.
5. Deploy the cluster with **nodus cluster deploy <cluster-name>**.

Summary

nodus cluster add cluster-file.json

nodus user set cluster/<cluster-name>/credentials /path/to/credentials.json

5.2.2 Connecting to a Cluster

1. To connect to a deployed cluster via SSH, run the command **nodus cluster connect <cluster-name>**.

5.2.3 Destroying a Cluster

1. Assuming the cluster was deployed properly through the above CLI commands, a cluster can be destroyed using the command **nodus cluster destroy <cluster-name>**.
2. The cluster should be destroyed before you delete it, or you will have to destroy it manually. To remove all information about the cluster, run the command **nodus cluster delete <cluster-name>**.

5.2.4 Viewing All Clusters

1. To view all saved clusters, deployed or not, run the command **nodus cluster list**.

5.3 Node Actions

1. To deploy/destroy a cluster's individual nodes manually, run the commands:

   nodus cluster deploy-node <cluster-name> <node-names>

   nodus cluster destroy-node <cluster-name> <node-name>

   Note: Node names are always `nodus-cluster-y-node-x`, where y is the instance size label of the cluster (e.g., t2-micro) and x is a number from 0 to (count - 1).

5.4 Job Actions

1. To submit a job to a cluster, run the command **nodus job submit <cluster-name> <job-name>**.

5.5 Bursting

The NODUS burst function looks at what jobs are queued on your cluster and automatically spins up or tears down nodes depending on the total requirements for the queue. If there are not enough online nodes to run all jobs, bursting will bring on as many nodes as needed. If there are more nodes than needed, the excess nodes will be destroyed. If the job queue is empty, all nodes will be destroyed.

1. To start the bursting service, run the command **nodus cluster bursting-service <cluster-name>**.
2. To stop the bursting service, run the command **nodus cluster stop-bursting <cluster-name>**.
3. To run a single burst cycle, run the command **nodus cluster burst <cluster-name>**.

5.6 User Keys

The NODUS CLI comes with user key, value storage. These resources are used by the CLI commands.

nodus cluster add <cluster-name> creates a user key, **cluster/<cluster-name>**, containing the cluster file.

5.6.1 List All User Keys

1. To view all of your user keys, run the command **nodus user list**.

5.6.2 Get/Set Values

1. To retrieve the value of a key, run the command **nodus user get <key>**.
2. To set the value of a key or create a new key, run the command **nodus user set <key> <value>**.

5.7 CLI Usage

5.7.1 CLI Setup

1. Link the user to the NODUS Cluster Provider Package using **nodus packages add nodus-cluster-provider /NODUS/nodus4-cluster-providers**.
2. Create credential files for the cloud provider as defined in the section Credential File Templates.
3. Copy cluster templates for use in cluster builds to the user: **cp -r /NODUS/nodus-cluster-providers/resource-templates/clusters/ ./**.
4. Copy stack templates for use in cluster builds to the user: **cp -r /NODUS/nodus-cluster-providers/resource-templates/stacks/ ./** as defined in the section Stack File Templates.
5. Create NODUS images for the provider/region (see the section Building Stacks into Images).
6. If **~/.ssh/id_rsa** and **~/.ssh/id_rsa.pub** do not exist, create SSH keys for the users using **ssh-keygen** (use defaults and no paraphrase).

7. Copy the provider template to the working file using **cp cluster-templates/<provider>-cluster.json to <cluster>.json**.
   Where <provider> is the provider name, for example, **cp cluster-templates/aws-cluster.json to <cluster>.json**.
8. Edit the cluster file **<cluster>.json** to meet the requirements for the head node and compute nodes, define a unique name for the cluster **<cluster name>**, and define the provider image size names.

5.7.2 Cluster File Templates

- AWS
- Google Cloud
- Microsoft Azure
- Oracle Cloud
- Huawei Cloud
- OTC

5.7.2.A AWS

```
{
"provider": "aws",
"name": "aws-cluster-0",
"server": {
"image": "nodus-server",
"size": "t2.micro"
},
"node": {
"image": "nodus-node",
"group": [
{
"size": "t2.micro",
"count": 1
}
]
},
"region": "us-east-1",
```

```
"availability_zone": "us-east-1a",
"software": []
}
```

5.7.2.B Google Cloud

```
{
"provider": "google",
"name": "google-cluster-0",
"server": {
"image": "nodus-server",
"size": "n1-standard-1"
},
"node": {
"image": "nodus-node",
"group": [
{
"size": "n1-standard-1",
"count": 1
}
]
},
"region": "us-east-1",
"availability_zone": "us-east1-b",
"software": []
}
```

5.7.2.C Microsoft Azure

```
{
"provider": "azure",
"name": "azure-cluster-0",
"server": {
"image": "nodus-server",
"size": "Standard_DS1_v2"
},
"node": {
"image": "nodus-node",
```

```
"group": [
{
"size": "Standard_DS1_v2",
"count": 1
}
]
},
"region": "eastus",
"image_resource_group_name": "nodus-eastus",
"software": []
}
```

5.7.2.D Oracle Cloud

```
{
"provider": "oracle",
"name": "oracle-cluster-0",
"server": {
"image": "nodus-server",
"size": "VM.Standard1.1"
},
"node": {
"image": "nodus-node",
"group": [
{
"size": "VM.Standard1.1",
"count": 1
}
]
},
"region": "us-ashburn-1",
"software": []
}
```

5.7.2.E Huawei Cloud

```
{
"provider": "huawei",
```

```
"name": "huawei-cluster-0",
"server": {
"image": "nodus-server",
"size": "s2.small.1"
},
"node": {
"image": "nodus-node",
"group": [
{
"size": "s2.small.1",
"count": 1
}
]
},
"region": "ap-southeast1",
"availability_zone": "ap-southeast-1a",
"software": []
}
```

5.7.2.F OTC

```
{
"provider": "otc",
"name": "otc-cluster-0",
"server": {
"image": "nodus-server",
"size": "s2.medium.1"
},
"node": {
"image": "nodus-node",
"group": [
{
"size": "s2.medium.1",
"count": 1
}
]
},
```

```
"region": "eu-de",
"software": []
}
```

5.7.3 Managing Clusters

Add a cluster: **nodus cluster add <cluster>.json <credential>.json**

Show all clusters: **nodus cluster list**

Deploy a cluster: **nodus cluster deploy <cluster>.json**

Show cluster members: **nodus cluster members <cluster name>**

Resize a cluster: **nodus cluster resize <cluster-name> <instance-size> <instance-size-count>** (this creates a file called **<cluster>json.resize** that will be used if you issue a **nodus cluster deploy <cluster name>** again).

Destroy a cluster: **nodus cluster destroy <cluster-name>**

5.7.4 Jobs

Use the following job json template for each job file to create:

```
{
"name": "",
"walltime": "24:00:00",
"description": "",
"nodes": 1,
"cores": 1,
"script": "/full/path/to/script.sh",
"dataFiles": []
}
```

Notes:

name	(required, creates a unique name <job name>)
walltime	(required, time limit for the job)
description	(optional, job description)
nodes	(required, number of nodes needed for the job)
cores	(required, number of cores per node needed for the job)
script	(required, the source code for the script or the absolute path to the script to run)
dataFiles	(optional, array of file names to data files needed for the job)

Add a job: **nodus job add <file>.json**

Show all jobs: **nodus job list**

Show job definition: **nodus job get <job name>**

Submit a job: **nodus job submit <cluster name> <job name>**

Show the job queue: **nodus cluster queue <cluster name>** (use **| jq** to pretty print)

Show all job outputs: **nodus job outputs <cluster name>** (**<job number>** is significant for future commands)

Show job output: **nodus job output <cluster name> <job number>**

5.7.5 Build Images

After a cluster is associated with a set of credentials and a stack file, use the command: **nodus stack build <cluster name>**.

Chapter 6: NODUS Cloud OS CLI Commands

In this chapter:

6.1 Cluster Management

6.1.1 Cluster Deploy

This deploys a cluster using credentials from the bash environment variable $CREDENTIALS_FILE or from **nodus user get cluster/cluster-name/credentials**:

$ nodus cluster deploy <cluster name>

6.1.2 Cluster Deploy-Node

This deploys nodes on a cluster:

$ nodus cluster deploy-node <cluster name> <node names>

6.1.3 Cluster Destroy-Node

This destroys nodes on a cluster:

$ nodus cluster destroy-node <cluster name> <node names>

6.1.4 Cluster Connect

This connects to a cluster:

$ nodus cluster connect <cluster name>

6.1.5 Cluster Resize

This resizes a cluster:

$ nodus cluster resize <cluster name> <instance-size> <instance-size-count>

6.1.6 Cluster Destroy

This destroys a cluster:

$ nodus cluster destroy <cluster name>

6.2 Cluster Monitoring

6.2.1 Cluster Info

This gets general information about the cluster (i.e., cluster username, public IP, node list...):

$ nodus cluster info <cluster name>

6.2.2 Cluster Jobs

This displays JSON information about the jobs queue:

$ nodus cluster queue <cluster name>

6.2.3 Cluster JSON

This displays the cluster definition JSON file for a cluster:

$ nodus cluster json <cluster name>

6.2.4 Cluster List

This lists all user clusters:

$ nodus cluster list

6.2.5 Cluster Members

This displays serf members from the head node:

$ nodus cluster members <cluster name>

6.3 Cluster Jobs

6.3.1 Cluster Burst

This runs a single burst cycle that spins up and tears down nodes depending on workload:

$ nodus cluster burst <cluster name> [--max]

Note: [--max] is optional.

6.3.2 Cluster Run-Job

This remotely runs a local job script on a cluster:

$ nodus job submit <cluster-name> <job-name>

6.4 Resource Management

6.4.1 User Set

This sets a user's key, value pair for resources:

$ nodus user set <name of resource> <path to resource>

6.4.2 User Get

This gets the value of a user's resource:

nodus user get <name of resource>

6.4.3 User List

This lists all resource keys for a user:

nodus user list

6.4.4 User Delete

This deletes a user's resource:

nodus user delete <name of resource>

6.4.5 Cluster Add

This adds a cluster using the provided cluster definition file:

nodus cluster add <cluster file> [credentials-name] [stack name]

Note: [credentials-name] [stack name] is optional.

6.4.6 Cluster Delete

This deletes the cluster from user storage (does not destroy the cluster):

$ nodus cluster delete <cluster name>

6.4.7 Cluster Set-Credentials

This associates a credentials file with a cluster:

$ nodus cluster set-credentials <cluster-name> <credentials-name>

6.4.8 Cluster Set-Stack

This associates a stack file with a cluster:

$ nodus cluster set-stack <cluster-name> <stack-name>

Appendix A: Creating Accounts for NODUS Cloud OS Providers

If you don't have credentials through your company, you can get them by following the procedures below.

In this appendix:

A.1 Creating an AWS Account

1. Go to https://aws.amazon.com/ and click **Create an AWS account**.
2. Follow the steps and enter your account information and then click **Continue**.
3. Choose **Personal** or **Professional**.
4. Enter your company information and accept the customer agreement.
5. Choose **Create Account and Continue**.
6. Provide a payment method and verify your phone number.
7. Click **My Security Credentials** > **Access keys (access key ID and secret key)** > **Create New Access Key** and create a new one, then download the file with the access key and secret key. **Note:** If you forget the secret key, you will have to delete it and create a new one.

A.2 Creating a Google Cloud Account

1. Go to https://cloud.google.com/ and click **Get started for free.**
2. Follow the steps on the screen to complete account registration.

A.3 Creating a Huawei Cloud Account

1. Go to https://intl.huaweicloud.com/en-us/ and click **Register**.
2. Follow the steps on the screen to complete account registration.

A.4 Creating a Microsoft Azure Account

1. Go to https://azure.microsoft.com/en-us/free/ and click **Start Free**.
2. On the **Sign in** page, click **Create one!.**
3. Click **Next** and then provide the rest of your user information (i.e., username, passwords, credit/debit, etc.).
4. Run the script **Azure.sh** to configure the Azure account.

A.5 Creating an OTC Account

1. Go to https://open-telekom-cloud.com/en and click **Open Telekom Cloud Console**.
2. Click **Register** and follow the steps on the screen to complete account registration.

A.6 Creating an Oracle Cloud Account

1. Go to https://www.oracle.com/cloud/ and click **View Accounts**.
2. Click **Create an account** and follow the steps on the screen to complete account registration.

See the appendix Obtaining Provider Account Information and Credentials below for additional information.

Appendix B: Obtaining Provider Account Information and Credentials

Follow the checklists below to prepare accounts and get credentials for the chosen provider.

In this appendix:

B.1 AWS

- Create Key: IAM > Security Status > Manage Security Credentials > Access Keys > Create New Access Key
- Copy and fill Credentials File Template

B.2 Google Cloud

- Select Project > IAM > Service Accounts > Create Key (JSON)

B.3 Huawei Cloud

Create Images (see https://support-intl.huaweicloud.com/en-us/):

- Create a virtual private cloud (VPC) network
- Create a subnet on the VPC
- Create a stack file (where: network = {network_id} of the subnet)

B.4 Microsoft Azure

Azure Template Credential File

This is a template credential file for Azure:

```
{
"client_id": " ",
"client_secret": " ",
"subscription_id": " ",
"tenant_id": " "
"region": " "
}
```

Obtaining Azure Credential Information

Follow these steps to obtain credential information.

client_id/tenant_id

1. On the Azure homepage (https://portal.azure.com/), under the **Azure services** section, click **App registrations**.
2. Under **Display name**, click the requested application (e.g., nodus) and copy/paste the **Application (client) ID** into the template between the quotes for the **client_id** variable, and copy/paste the **Directory (tenant) ID** into the template between the quotes for the **tenant_id** variable.
   Note: If you have not previously registered an application, for this step, first click **Register an application** and enter the required information.

client_secret

Since Azure hides the client key, we can make another one that coexists with the one that is hidden by Azure.

3. Under the **Azure services** section, click **App registrations**.
4. Under **Display name**, click the requested application (e.g., nodus).
5. Under the **Manage** section on the left side of the page, click **Certificates & secrets**.
6. In the **Client secrets** section, click **New client secret**.
7. Add a **Description** and specify an **Expiration** time frame.
8. Copy/paste the **Value** into the template between the quotes for the **client_secret** variable.

subscription_id

9. Under the **Navigate** section, click **Subscriptions**.
10. Copy/paste the **Subscription ID** into the template between the quotes for the **subscription_id** variable.

region

11. Enter the desired region into the template between the quotes for the **region** variable. For example, the region can be **eastus** or **westus**.

B.5 OTC

Create Images (see https://open-telekom-cloud.com/en/products-services/virtual-private-cloud):

- Create VPC network
- Create subnet on VPC
- Create stack file (where: network = {network_id} of the subnet)

B.6 Oracle Cloud

Create Images (see https://docs.cloud.oracle.com/iaas/Content/home.htm):

- Create VPC network
- Create subnet on VPC

Appendix C: Job Preparation

The Job script can be run using standard Shell scripts such as batch, Perl, Geant4, or Python. The Job script files are stored in the `Input` folder and the results are stored in the `Output` (head node) folder.

Example Job Script

```
#!/bin/bash

set -x

cd input

PROCESSORS=1

TASKNAME=geant4

cp Dockerfile.template Dockerfile

sed -i -e "s/__FILE__/$FILE/g" Dockerfile

sed -i -e "s/__PROCESSORS__/$PROCESSORS/g" Dockerfile

sudo docker build -t $TASKNAME .

sudo docker run -v $PWD/../output:/app/output $TASKNAME
```

Example Input Dockerfile.template Contents

```
FROM ifurther/geant4

RUN mkdir BUILD

RUN mkdir output

RUN sed -i -e "s/g4root.hh/g4csv.hh/g"
/src/geant4.10.05.p01/examples/extended/radioactivedecay/rdecay01/include/HistoManager.hh

RUN cd BUILD; cmake -DGeant4_DIR=/app/geant4.10.5.1-install/lib/Geant4-10.5.1/
/src/geant4.10.05.p01/examples/extended/radioactivedecay/rdecay01/

RUN cd BUILD; make -j __PROCESSORS__ rdecay01

RUN cd BUILD; make install

RUN . /app/geant4.10.5.1-install/bin/geant4.sh ; /usr/local/bin/rdecay01 ./BUILD/__FILE_
_.mac

CMD mv *.csv output
```

Note: This program creates CSV files and moves them to the Output folder according to the script.

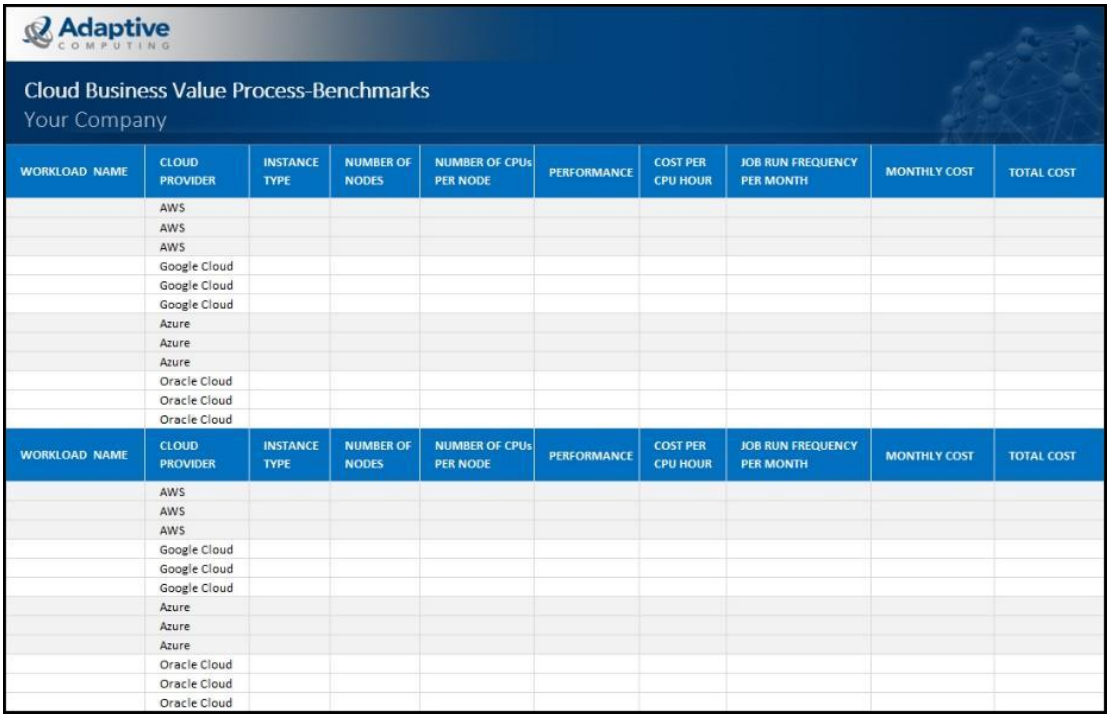
Appendix D: NODUS Cloud OS Workload Considerations

Listed below are some of the considerations and questions that need to be answered when running workloads in the cloud. All of these may not be appropriate for each of your workloads, or you may have additional considerations.

1. Not all workloads can be effectively run in the cloud, so each workload should be tested for cloud appropriateness by running them in the cloud and collecting statistics.
   - NODUS Cloud OS has multi-cloud capabilities and will run workloads in the cloud from the UI on one or more of the major cloud providers.
   - Run your workload on each cloud provider and record the statistics for each one in the Cloud Business Value Process Benchmark Report (see Figure 1 Benchmarks).
   - HTC (High-Throughput Computing) workloads and small scale HPC (High-Performance Computing) workloads that are not memory-, communication-, large-data-, or bandwidth-intensive are ideal for migrating to cloud environments.

2. Cloud OpEx costs are difficult to model without actually running the workload in the cloud.

   - Build a cost and performance model based on the statistics from (Figure 1) below. This will help you develop your cloud budget. See the attached Cloud Business Value Process Spreadsheet.



Adaptive COMPUTING

Cloud Business Value Process-Benchmarks

Your Company

WORKLOAD NAME	CLOUD PROVIDER	INSTANCE TYPE	NUMBER OF NODES	NUMBER OF CPUs PER NODE	PERFORMANCE	COST PER CPU HOUR	JOB RUN FREQUENCY PER MONTH	MONTHLY COST	TOTAL COST
	AWS								
	AWS								
	AWS								
	Google Cloud								
	Google Cloud								
	Google Cloud								
	Azure								
	Azure								
	Azure								
	Oracle Cloud								
	Oracle Cloud								
	Oracle Cloud								

WORKLOAD NAME	CLOUD PROVIDER	INSTANCE TYPE	NUMBER OF NODES	NUMBER OF CPUs PER NODE	PERFORMANCE	COST PER CPU HOUR	JOB RUN FREQUENCY PER MONTH	MONTHLY COST	TOTAL COST
	AWS								
	AWS								
	AWS								
	Google Cloud								
	Google Cloud								
	Google Cloud								
	Azure								
	Azure								
	Azure								
	Oracle Cloud								
	Oracle Cloud								
	Oracle Cloud								

Figure 1: Benchmarks

   - When using NODUS Cloud OS, expenditures move from CapEx to OpEx because you are gaining capacity without buying more hardware. This allows for the immediate availability of resources and the ability to scale up clusters instantaneously.

3. Choose the appropriate delivery model for each of your workloads.

   - NODUS Cloud OS accommodates on-premises, even your laptop, SaaS, and cloud-hosted delivery models, which are very easy to implement because of the low barriers to entry using NODUS Cloud OS. The delivery model for each workload could be different.

4. Do you want to run your workload independent from or as part of your on-premises cluster?

   - NODUS Cloud OS offers superior usability when devising cloud and on-premises integration strategies.
   - NODUS Cloud OS works with any HPC or enterprise job scheduler or without a workload scheduler.
   - In the diagram below (Figure 2) the workload has burst to the cloud automatically based on backlog. The job is seen by the scheduler as part of the on-premises cluster. NODUS Cloud OS has the capability to seamlessly integrate on-premises and cloud resources. See the section Bursting Service for additional information.

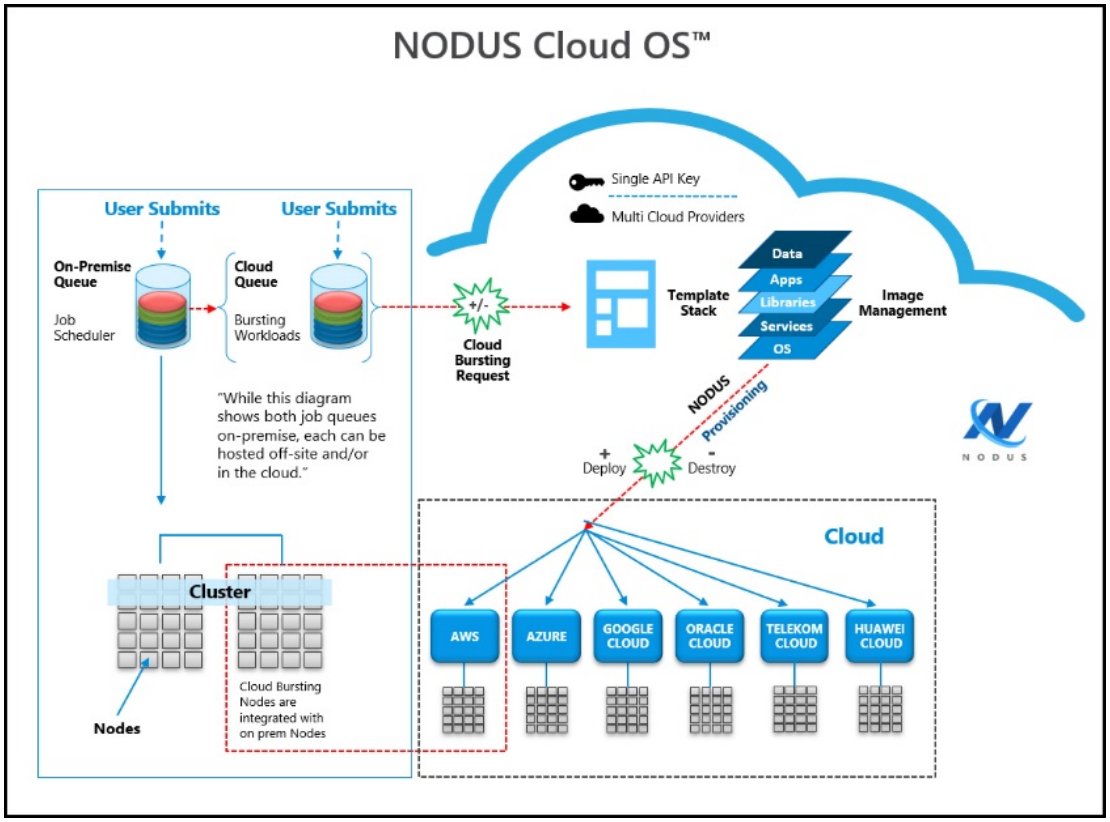


Figure 2: NODUS Cloud OS Bursting Becoming On-Premises

- In the diagram below (Figure 3) the workload has burst to the cloud on demand and is running in the cloud independently of the on-premises cluster.

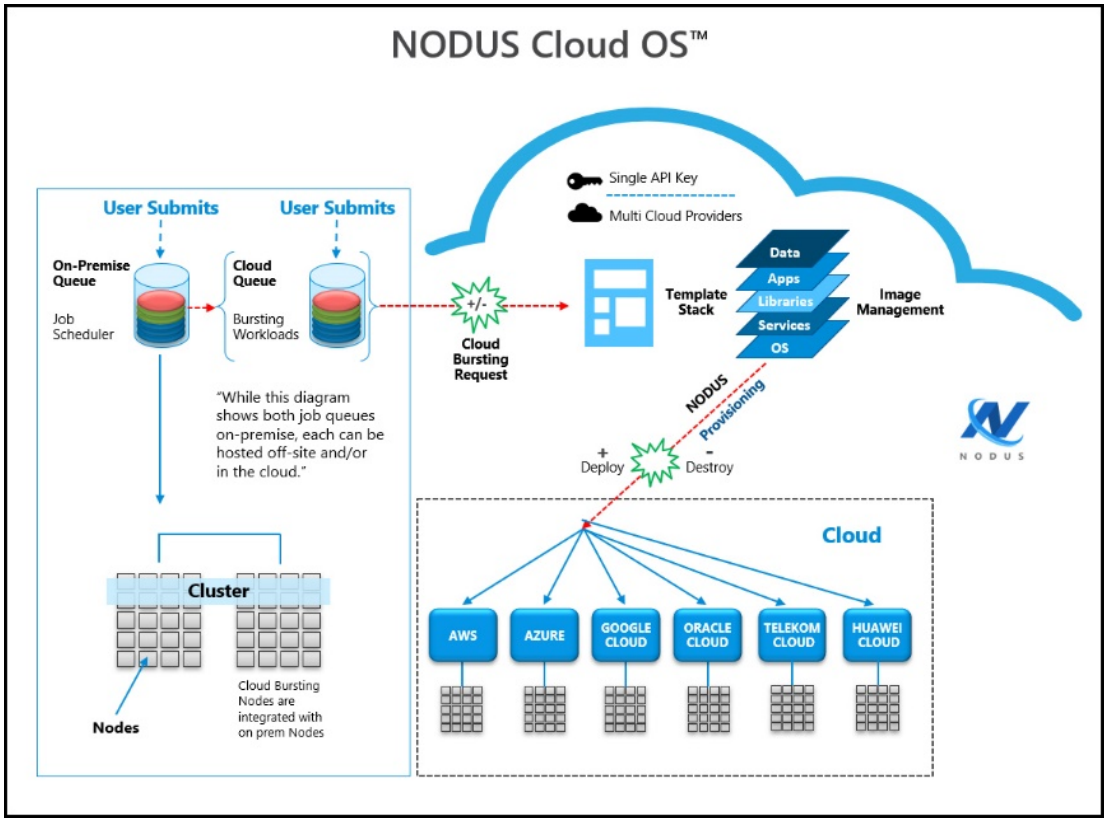


Figure 3: NODUS Cloud OS Bursting Off-Premises

5. Which workloads can be run in the cloud on VMs and which need bare metal?

- Can you get the performance you need from VMs?
- Virtualization layers can often cause slower performance, although VMs can be easily moved to and from similar clouds and are less expensive.
- You may have to run your workload on bare metal to get the performance results required.

6. Which workloads require additional licenses to run in the cloud and which do not?
   - Determine if your workload requires additional licensing to run in the cloud.
   - License sharing can be difficult in international organizations and will increase costs. License sharing is easier to manage with regional workloads, making these better suited to run in the cloud and are less expensive.
   - Use vendors or open-source software with cloud-friendly terms.

7. Which of your workloads require moving large amounts of data to the cloud and which do not?
   - Look at each job, the amount of data it requires, and then make a determination.
   - Consider compute, memory, and storage when assessing data requirements.
   - Test your data transfer line.
   - Move your data first and then spin up the nodes to run your workload in the cloud.
   - NODUS Cloud OS has a data movement feature, which can automatically move the data and proliferate it.

8. Which public cloud providers match which of your workloads? One size does not fit all.
   - Using NODUS Cloud OS, you can test your workloads on each one of the cloud providers. Within the NODUS Cloud OS interface, select the cloud provider that your job will run on. The nodes will shut down when the job completes, and you can then select another cloud provider to test your workload on. For example, one workload should run on AWS and another should run on Google Cloud.
   - NODUS Cloud OS supports all major public clouds, and can be configured to support any public cloud of any size.
   - After calculating the cost for a workload, multiply that cost by the number of times the workload runs each month. Repeat this for each cloud provider (Figure 1).

- A blank worksheet is provided if you want to test other cloud providers (Figure 4). See the attached Cloud Business Value Process Spreadsheet.

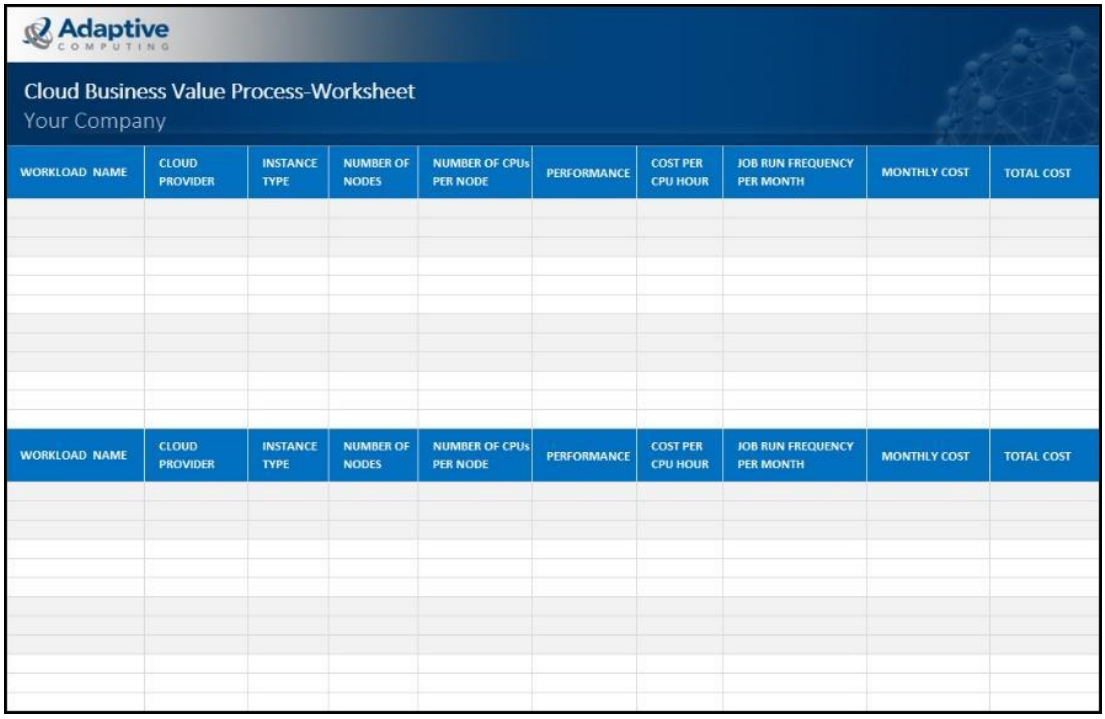


Figure 4: Worksheet

- A budget sheet is included for you to summarize your testing results and prepare your Cloud Business Value Process Budget Report for management (Figure 5). See the attached Cloud Business Value Process Spreadsheet.

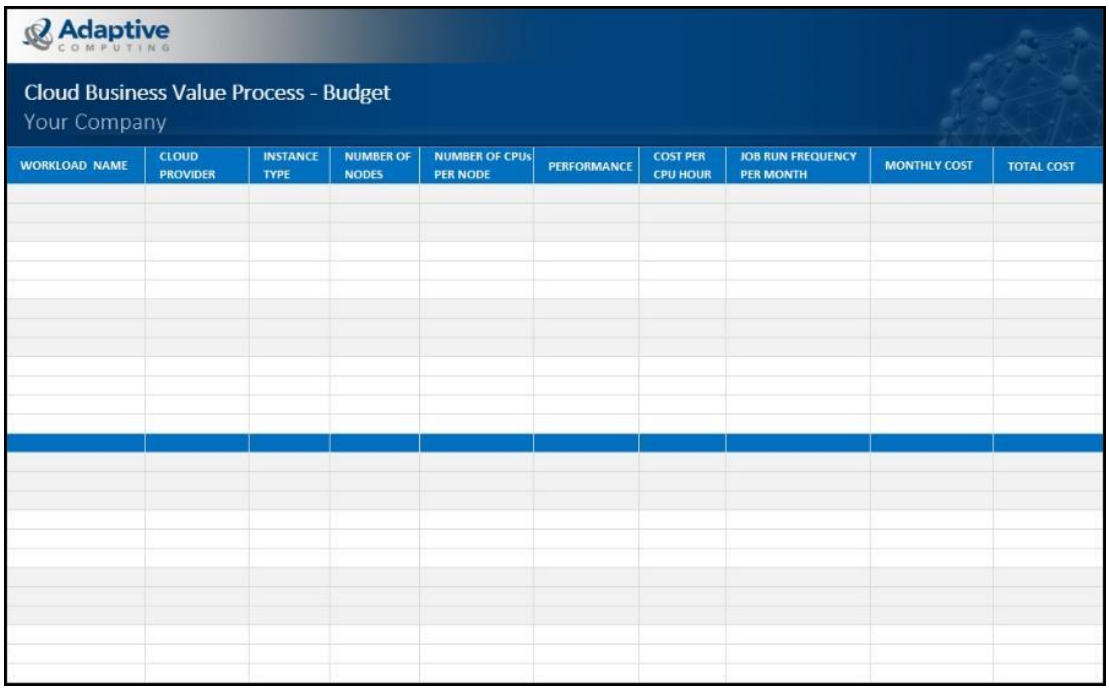


Figure 5: Budget

Test your workloads in the cloud one at a time (the most cloud-friendly ones first) and begin collecting statistics.

Enter the gathered statistical data into the attached spreadsheet. In a very short period, you will know which of your workloads can be run in the cloud, and on which cloud provider using which instance type. We recommend that you run each workload using at least three different instance types for each cloud provider to determine the best cost/performance for each workload (Figure 1).

When testing is concluded, your cloud budget will be determined.

Glossary

Bursting: The event of clusters and nodes being deployed to run jobs, then be destroyed.

Cluster: A collection of compute instances consisting of a head node and compute nodes.

Cluster Size: The number of compute nodes.

Compute Nodes: The servers that provide the storage, networking, memory, and processing resources.

Compute Node Size: An instance type or hardware configuration (for example, n1-standard-2 - vCPU: 2, Mem (GB): 7.50).

Core: An individual hardware-based execution unit within a processor that can independently execute a software execution thread and maintain its execution state separate from the execution state of all other cores within the processor.

Credentials: Authentication information required to access the respective cloud provider from code.

Custom Job: A job that is customizable and configurable.

Head Node: The server that manages the delegation of jobs.

Image: A snapshot of an OS.

Job Script: A program to be run on a cluster (generally a shell script).

mpi-benchmarks: A job that tests the performance of a cluster system, including node performance, network latency, and throughput.

On Demand Cluster: A cluster that carries out a specific job then is removed.

Provisioning: The event of configuring a node or cluster with its stack and getting it into a ready-to-work state.

Scheduler: The specialized software between the user and the HPC cluster/datacenter system that manages submitted workloads or jobs. This includes queuing jobs, prioritizing queued jobs for execution, scheduling and allocating requested resources for each job, and starting jobs when their requested resources become available and the jobs have the highest priority.

Stack: An instance of software packages that defines the operating system components.

test-job: A job that is best used to test bursting functionality; it echos the time and the hostname of the executing machine.

Thread: The quantity of software execution threads the core can simultaneously track.

Walltime: The job's time limit (HH:MM:SS).

Workload: Jobs to be run and/or jobs in the queue.

Index

- D -

- E -

- F -

- G -

- H -

- V -

- W -