NODUS Cloud OS

User Guide 5.0.0

July 2020

Contents

Welcome

The NODUS Cloud OS User Guide will show you how to use the NODUS Cloud OS user interface and command line 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 service provider. This highly flexible and customizable solution enables HPC or enterprise systems to 'burst' scheduled workloads 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

This documentation and related software are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

This documentation and related software may provide access to or information about content, products, and services from third-parties. Adaptive Computing is not responsible for and expressly disclaims all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Adaptive Computing. Adaptive Computing will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Adaptive Computing.

Adaptive Computing, Moab®, Moab HPC Suite, Moab Viewpoint, Moab Grid, NODUS Cloud OS™, and other Adaptive Computing products are either registered trademarks or trademarks of Adaptive Computing Enterprises, Inc. The Adaptive Computing logo is a trademark of Adaptive Computing Enterprises, Inc. All other company and product names may be trademarks of their respective companies.

The information contained herein is subject to change without notice and is not warranted to be error free. If you find any errors, please report them to us in writing.

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
June 2020	NODUS Cloud OS 5.0.0

Related Documentation

- *Cloud Business Value Process Spreadsheet*
  http://support.adaptivecomputing.com/Cloud-Business-Value-Process-Spreadsheet.xlsx

Chapter 1: 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.

In this chapter:

1.1 Prerequisites

This section contains the following information:

- Operating System
- Supported Browsers
- Linux Packages
- Traffic
- Obtaining Credentials from the Cloud Service Providers
- Stacks

1.1.1 Operating System

NODUS platform can be installed on many different Linux distributions and has been tested 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.

1.1.2 Supported Browsers

The NODUS UI supports these web browsers:

- Google Chromium (latest version)
- Google Chrome (latest version)
- Mozilla Firefox (latest version)
- Microsoft Edge (two latest major versions)
- Apple Safari (two latest major versions)

1.1.3 Linux Packages

The following Linux packages must be installed on the NODUS platform server prior to the software installation:

- mongodb-server
- nginx

1.1.4 Traffic

To enable access to the NODUS platform server, the following ports need to be open with external access:

- 22 (SSH) - user direct access / Moab job routing
- 80 (HTTP) - connection to the NODUS UI
- 443 (HTTPS) - optional, if you secure your HTTP connection
- 12001 (Cluster API) - RESTful API that communicates with the clusters
- 12345 (NODUS-web-API) - the NODUS Platform API server that communicates with your UI

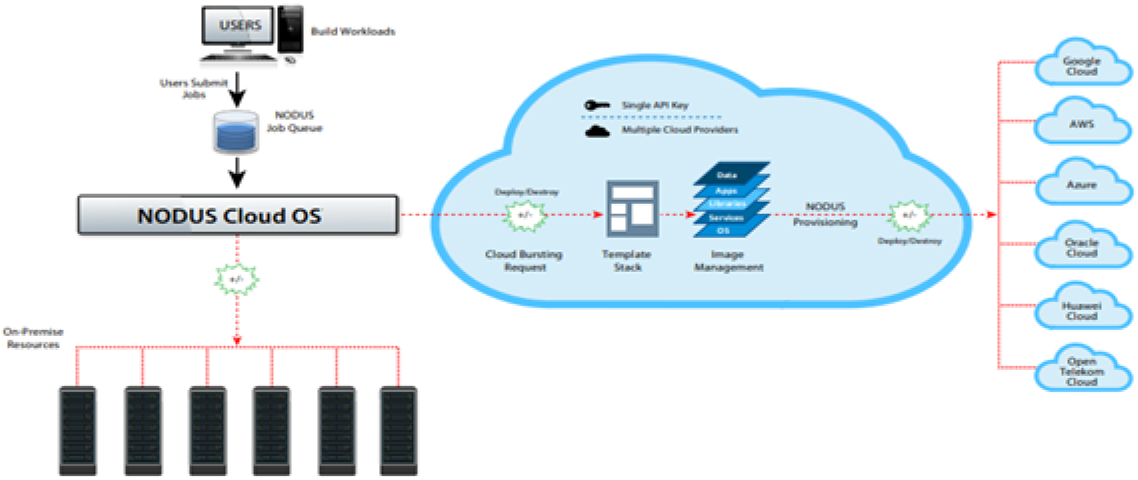
Note: The NODUS platform server installation will enable these ports on the server. Ports 80 and 443 are used for the UI, but can be changed to reflect your requirements.

1.1.5 Obtaining Credentials from the Cloud Service Providers

NODUS requires Programmatic access to deploy nodes and cluster networks on your cloud provider's resources. The steps below will help you obtain the proper cloud credentials. See the appendices Creating Accounts for NODUS Providers and Obtaining Cloud Service Provider Account Information and Credentials for additional information.

Refer to the providers' support pages for detailed information. **Note:** If you need a private IP address created within a provider or you cannot get provider credentials / correct admin level credentials, please contact us at nodus@adaptivecomputing.com.

NODUS Overview



Continue with the steps for your chosen provider below to obtain credentials:

- Google Cloud
- AWS
- Microsoft Azure
- Oracle Cloud
- Huawei Cloud
- Open Telekom Cloud

Google Cloud

1. Create a service account.
2. From the **Service account** list, select **New service account**.
3. In the **Service account name** field, enter a name.
4. From the **Role** list, select **Project / Owner**.
5. Click **Create**. A `JSON` file that contains your key downloads to your computer.

See https://cloud.google.com/docs/authentication/production for additional information.

AWS

AWS has two forms of credentials, both require admin privileges to set up:

- Programmatic access (access key, secret key) is a floating set of coupled values.
- EC2 roles only exist within AWS and are attached to AWS instances.

These accounts must have valid permissions associated in order to use NODUS. NODUS can deploy clusters using either programmatic access or EC2 roles. However, stack builds can only be done with Programmatic access.

NODUS requires Programmatic access to access the provider's resources. For example, if NODUS is outside of the AWS environment, access keys are required. If NODUS is running inside AWS, the best practice is to use IAM roles instead. An IAM role is a defined set of permissions. It is not associated with a specific user or group and any trusted entity can assume the role to perform a specific business task. NODUS supports both access methods.

When creating a cluster via **nodus cluster:aws:create**, the flag `--role` can be used to set the cluster to use EC2 role instead of Programmatic access. If the cluster entry already exists, then **nodus cluster:aws:update** can be used with flag `--role` or `--no-role` to switch on EC2 role use or turn it off, respectfully.

1. Sign in to the AWS Management Console and open the **IAM** console.
2. In the navigation pane, select **Users**.
3. Select the name of the user whose access keys you want to create, and then select the **Security credentials** tab.
4. In the **Access keys** section, select **Create access key**.
5. To view the new access key pair, select **Show**. You will not have access to the secret access key again after this dialog box closes.
6. To download the key pair, select **Download .csv file**. Store the keys in a secure location. You will not have access to the secret access key again after this dialog box closes.
7. After you download the `.csv` file, select **Close**. When you create an access key, the key pair is active by default, and you can use the pair right away.

See https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html for additional information.

Microsoft Azure

1. Run the script **account-scripts_azure.sh**.

See https://docs.microsoft.com/en-us/azure/automation/shared-resources/credentials for additional information.

Oracle Cloud

1. On the Oracle account to be used, create a compartment.
2. Within that compartment, create a VPC (virtual private cloud) network.
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.

1.1.6 Stacks

To launch instances in the cloud, a stack must first be created using credentials with appropriate cloud permissions and policies.

The stack consists of the base OS (CentOS 7), services, libraries, applications, and any data that is needed so that once the instance starts up, it is ready to begin processing jobs. NODUS allows customization options to conform to cloud configurations (proxy, vpc, subnet, etc.).

Two images are generated from the stack build process: a base OS image and a prebuilt/configured image snapshot. You can use a stock image from the marketplace or account to build a private image. Cluster software used is Torque, Ansible, Maui, and Serf.

See the CLI Commands section Stacks for additional information.

1.2 Installation Steps

1. In a browser, access the server via the IP or hostname (e.g., `http://x.x.x.x/`) to verify that `nginx` is started and the default port is available. If the site is unreachable, then the firewall may be enabled or the nginx server is not running.

   A. Connect to the server via a command line and run **sudo systemctl start nginx**.

   B. Access `http://x.x.x.x/` once again:

   - If you get a `Welcome to CentOS` message, the default port number is available and you can proceed with the installation. Or if the site is still unreachable, then the firewall may be enabled and the NODUS install can proceed.

   - If any other data is displayed on the browser, the default port may be in use and your administrator should assign you another port number that will be used in step **6**.

2. Log in to the installation user for the operating system where NODUS is to be installed to get a command line shell.

3. Untar the NODUS distribution file with the command **tar xf Nodus_Cloud_OS-<version>.tgz**. This will produce a directory with the installation files needed to install NODUS.

   A. Look at the files via **ls**. You should see a file called `nodus-<version>`.

   B. Change to the installation directory with the command **cd nodus-<version>**.

4. Because a license key is needed to install NODUS, save the license key in a file on the server. You will also need to provide a username that will be used to access the web server files. For security purposes, never use `root` as the username.

5. Install NODUS with the command **./install-nodus.sh <username> <license file>**.

6. (optional) If there is already a process using port 80 (from step **1**), change the nginx port number with the command **sudo <edit> /etc/nginx/nginx.conf**.

   A. Replace these lines:

   ```
   listen  80 default_server;
   listen  [::] :80 default_server;
   ```

   With these lines:

   ```
   listen  <new port> default_server;
   ```

```
        listen  [::] :<new port> default_server;
```

8. Restart the nginx server with the command **sudo systemctl restart nginx**.

1.3 Verifying the Installation

1. 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 ones you will supply licenses to. The file looks similar to this:

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

2. 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.

3. 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.

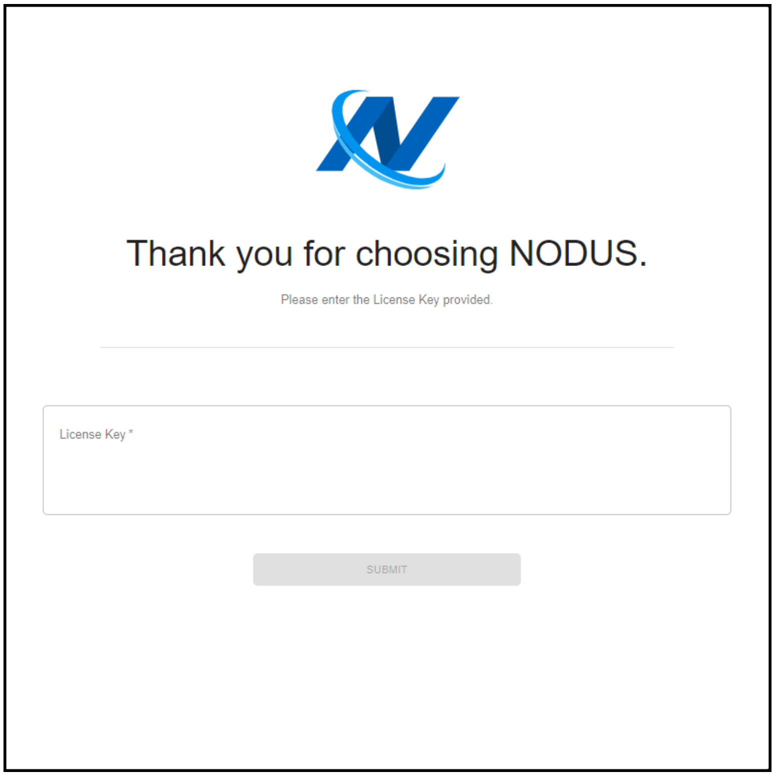
1.4 Adding the Server License Key

Follow one of the procedures below to add a license key.

- Adding the License Key via the NODUS GUI
- Adding the License Key from a Terminal

Adding the License Key via the NODUS GUI

1. Go to `http://<web_server_address>` and enter the license key that was provided. **Note:** web_server_address refers to the server where the NODUS platform is installed.



Thank you for choosing NODUS.

Please enter the License Key provided.

License Key *

Adding the License Key from a Terminal

1. Log in to the `<user_name>@<web_server_address>` that you installed NODUS as in the section Installation Steps.
2. Add the license key that was provided to `/NODUS/.license`. **Note:** If the license type is Multi-User, then add the user license to `~/.nodus/.userKey`.
3. Verify that the license is working by running command **nodus license**.

1.5 Modifying the Code

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

Chapter 2: Using the NODUS GUI

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

In this chapter:

2.1 Creating a NODUS Account

You can have one of two different types of accounts, depending on how your server is set up: Site Account or Multi-User Account.

1. Go to `http://<web_server_address>/register` and create an account.
   Note: web_server_address refers to the server where the NODUS platform is installed.
2. Follow the instructions and complete the registration process. Your username and password are required for future logins.

Site Account

Upon successful registration, you will be logged in to `http://<web_server_address>`.

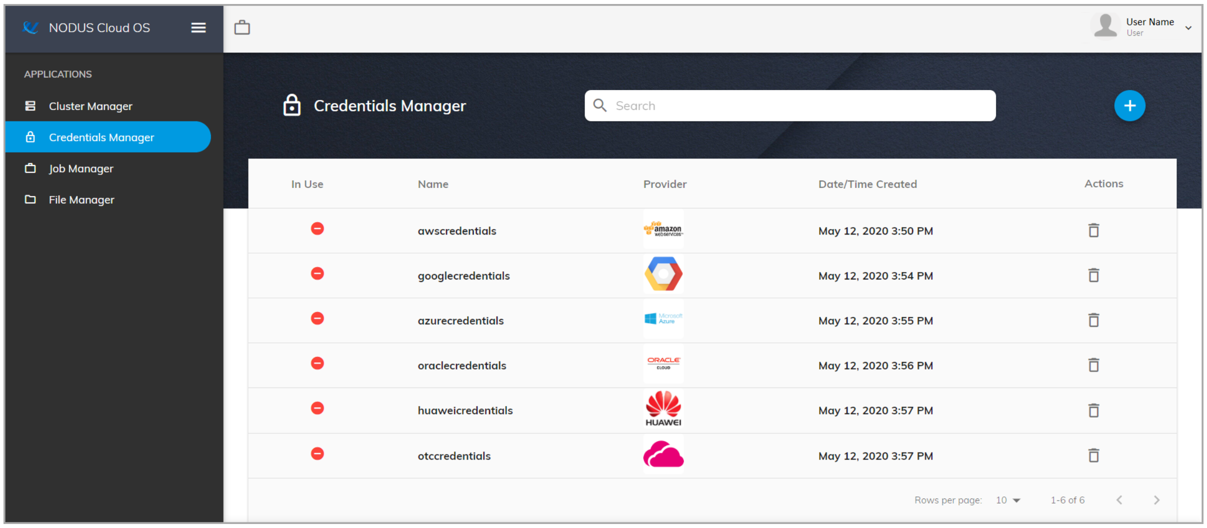
Multi-User Account

This message appears: `Account Creation Successful! In order to access NODUS you will need a USER KEY. If you do not already have one, please contact Adaptive Computing at license@adaptivecomputing.com!`

2.2 Credentials Manager

Credentials are required by the cloud service 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 when creating a cluster.

1. To add credentials, log in to your NODUS Cloud OS user account at `http://<web_server_address>`. The **Cluster Manager** screen opens.
2. On the **Applications** menu, click **Credentials Manager**:



Credentials Manager

In Use	Name	Provider	Date/Time Created	Actions
⊖	awscredentials	amazon web services	May 12, 2020 3:50 PM	🗑
⊖	googlecredentials		May 12, 2020 3:54 PM	🗑
⊖	azurecredentials	Microsoft Azure	May 12, 2020 3:55 PM	🗑
⊖	oraclecredentials	ORACLE CLOUD	May 12, 2020 3:56 PM	🗑
⊖	huaweicredentials	HUAWEI	May 12, 2020 3:57 PM	🗑
⊖	otccredentials		May 12, 2020 3:57 PM	🗑

Rows per page: 10 ▾ 1-6 of 6 ‹ ›

The **Credentials Manager** screen shows this information: In Use, Name, Provider, Date/Time Created, and Actions.

3. Select the appropriate cloud service provider and enter the desired name for the credentials.
4. Enter your account credentials. Alternatively, click **Upload Credential**.
5. Click **Save** to finish adding the credentials for this provider.

Note: Clicking the name of a credential shows information about it.

Repeat the steps in this section to add credentials for another provider if desired.

2.3 Building Stacks into Images

NODUS cluster stacks need to built for your provider and in the regions you want to use with a NODUS cluster.

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.

Successful Build

A successful stack build looks like this:

```
==> Builds finished. The artifacts of successful builds are:
--> market-server: AMIs were created:
us-east-1: ami-00d17452c46306a94

--> market-server:
--> market-node: AMIs were created:
us-east-1: ami-082631c41bb6b0761

--> market-node:

=== BUILD COMPLETE :: Wed Jun 17 2020 09:19:34 GMT-0400 ===
```

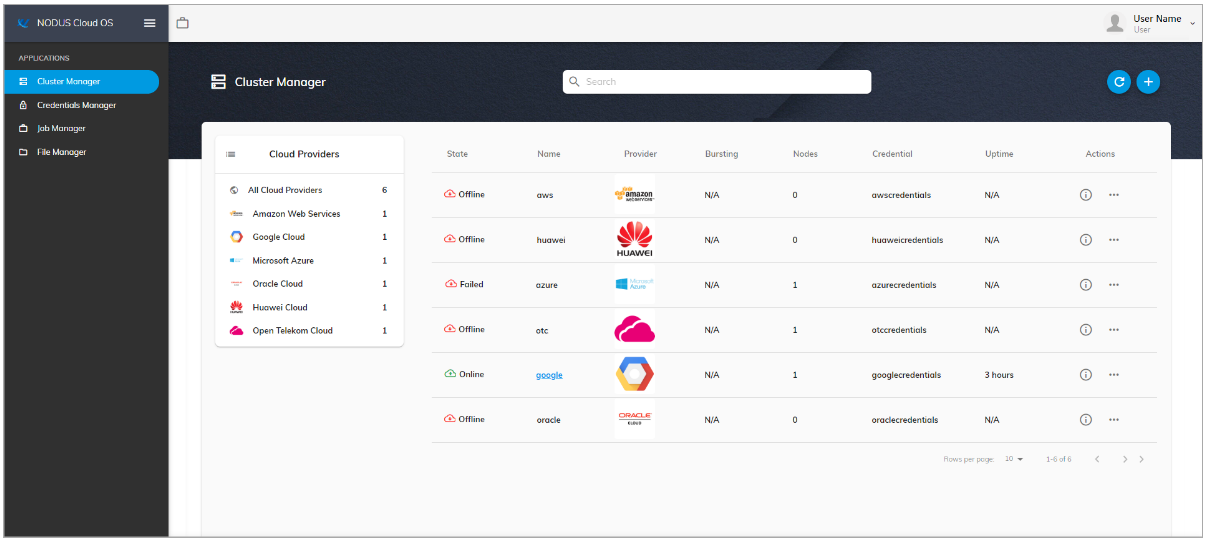
2.4 Cluster Manager

This section contains the following information:

- Viewing Clusters
- Creating a Cluster
- Deploying a Cluster
- Deleting a Cluster

2.4.1 Viewing Clusters

1. On the **Applications** menu, click **Cluster Manager**:



Cloud Providers	
All Cloud Providers	6
Amazon Web Services	1
Google Cloud	1
Microsoft Azure	1
Oracle Cloud	1
Huawei Cloud	1
Open Telekom Cloud	1

State	Name	Provider	Bursting	Nodes	Credential	Uptime	Actions
Offline	aws	amazon webservices	N/A	0	awscredentials	N/A	ⓘ ···
Offline	huawei	HUAWEI	N/A	0	huaweicredentials	N/A	ⓘ ···
Failed	azure	Microsoft Azure	N/A	1	azurecredentials	N/A	ⓘ ···
Offline	otc		N/A	1	otccredentials	N/A	ⓘ ···
Online	google		N/A	1	googlecredentials	3 hours	ⓘ ···
Offline	oracle	ORACLE CLOUD	N/A	0	oraclecredentials	N/A	ⓘ ···

Rows per page: 10 1-6 of 6

On the **Cluster Manager** screen, you can configure new clusters and see a list of all your current clusters and information such as their State (Online / Offline), Name, Provider, Bursting, Nodes, Credential, Uptime, and Actions (Information / Deploy, Logs, & Delete).

From the **Applications** menu, you can also add credentials, submit jobs, and view a job's output file. See the sections Credentials Manager, Job Manager, and File Manager for information.

Notes:

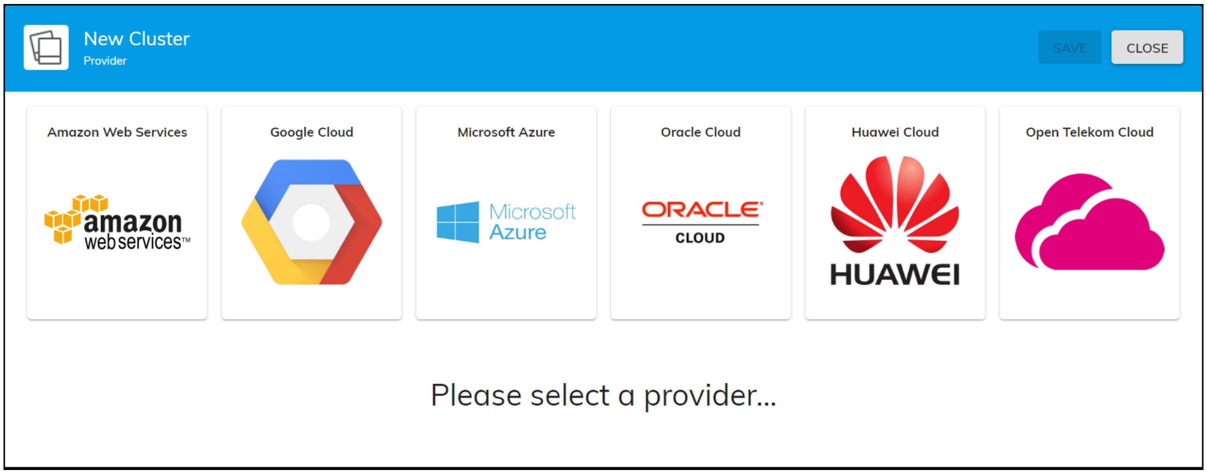
- Clicking **User SSH Key** next to your user name will download a `.pem` file that can be used to access the head node of a cluster via SSH. This provides you full control over the cluster.
- Clicking the refresh icon (↻) at the top right of the cluster list, refreshes the list.
- Clicking the add icon (+) 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** - ID, Name, State (All Jobs, Queued, Running, ✓ Completed), Submitted, Cores, Nodes, and Actions (⊗).

  - **Nodes** - □ (Deploy, Destroy), Name, State (🌐 All Nodes, 🖥 Available, 🖥 Busy, 🖥 Offline, 🖥 Down), Cores, and Available Threads. Nodes are online by default, but can be taken offline by selecting **Destroy**. A node that is down has been deprovisioned in the cloud service provider and does not count as a billable resource. A node that is offline is still provisioned in the cloud service provider and isn't being used to run jobs at the moment, but counts as a billable resource.
  - **Configuration** - Cluster ID, Head Node Size, Image Name, Cluster IP, SSH Username, Date/Time Created, and Download Cluster SSH Key.
- Clicking the information icon (**i**) at the end of the row shows additional details about the cluster, including: Cluster ID, SSH Username, Date/Time Created, Expected Deploy Time, Actual Deploy Time, Uptime, Head Node Public IP, Image Name, Head Node Size, Compute Node Sizes, and Region. The buttons **Deploy**/**Destroy** 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. After a cluster is destroyed, the state shows as `Offline`.
  - **Logs** - Shows information about the provisioning. A screen shows information in real-time about the cluster's state. Additionally, these logs contain information for deploy and destroy.
  - **Delete** - Permanently deletes the cluster. **Note:** The cluster must be destroyed first before it can be deleted.
  - **Cluster SSH Key** - Downloads the cluster SSH key.

2.4.2 Creating a Cluster

If you have already added your cloud credentials for the chosen provider, follow the steps below to create a new cluster, which can then be deployed at any time. 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 (⊕). The **New Cluster** screen shows the cloud service providers that you have credentials for:



Please select a provider...

These are the supported cloud service providers:

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

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

2. Select the cloud service provider that you want to deploy a cluster in.

3. Enter the desired information for the new cluster, such as: Name, Type, Prefix, Credentials, Region, Availability Zone, Server Size, Compute Node Instance Sizes, Node Size, Nodes Count, Persistent/Bursting, and Description.
   Notes: The available fields vary based on the provider. 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**. The state shows `Offline` until it is deployed.
5. Repeat the steps in this section to create additional clusters as desired.

2.4.3 Deploying a Cluster

1. To deploy a cluster, click the ellipsis and then click **Deploy**. The state changes to `Online`.

2.4.4 Deleting a Cluster

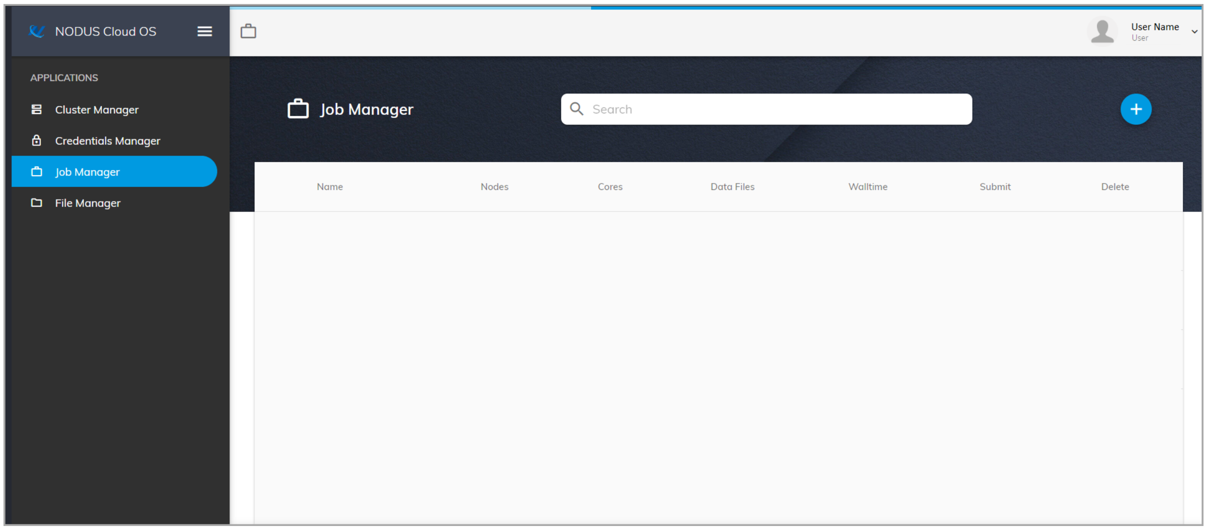
To delete a cluster, it must first be destroyed.

1. Click the ellipsis for the desired cluster and then click **Destroy**.
2. After the state changes to `Offline`, click **Delete**.

2.5 Job Manager

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

1. To add a job, on the **Applications** menu, click **Job Manager**. The **Job Manager** screen opens:



   This 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 desired information for the new job: 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.

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

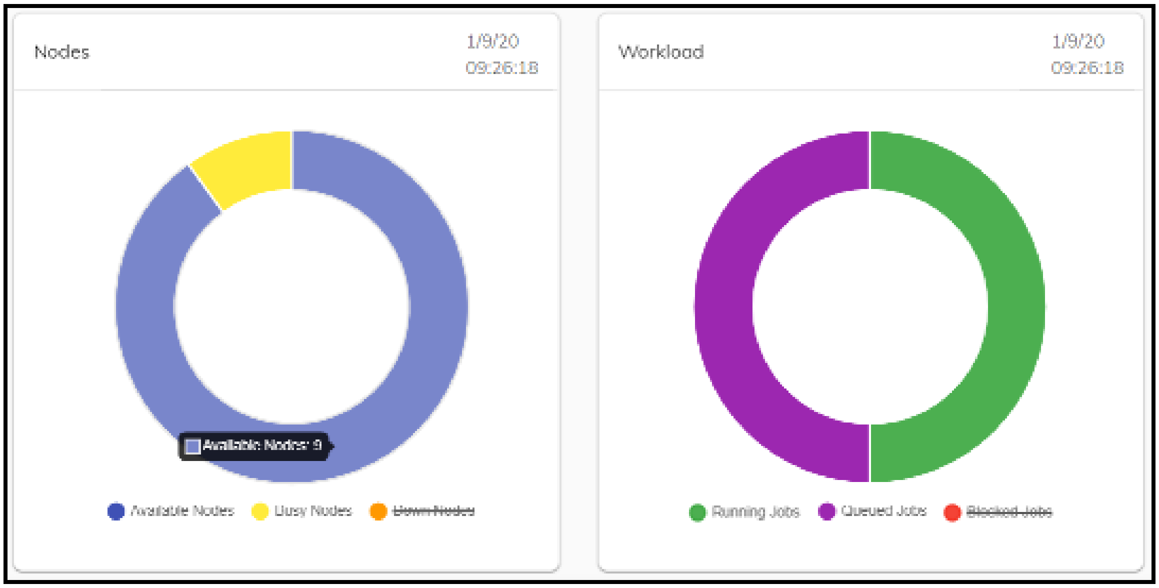
- **Named Cluster**: If there are no created clusters or ones that meet the hardware requirements of the job, then certain named clusters may not appear.

- **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. Perform the normal cluster configuration as if creating new, then select an on demand type.
  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.

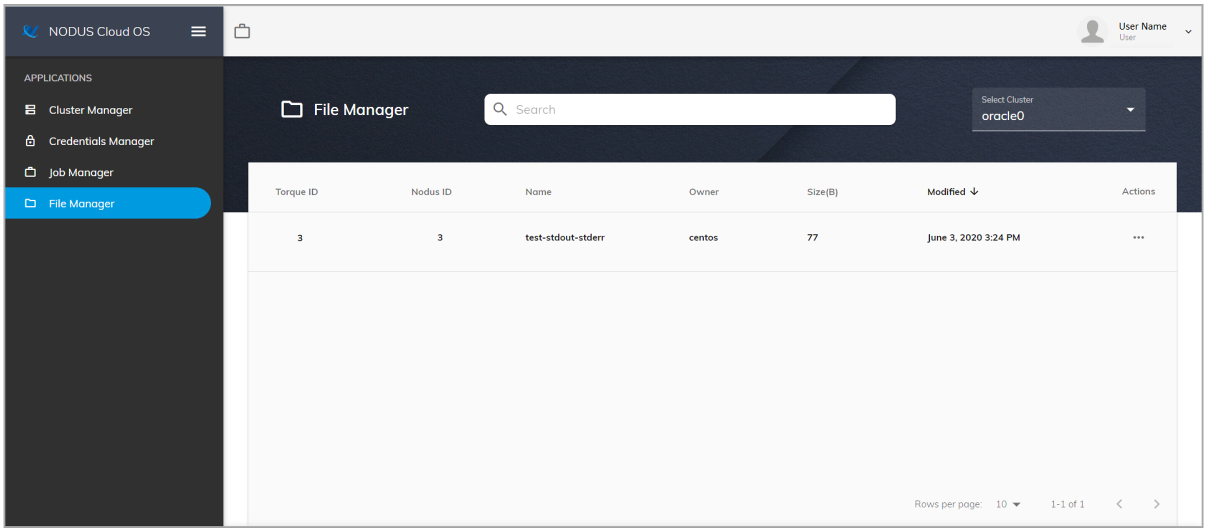
- **Nodes** - ☐ (Deploy, Destroy), Name, State (🌐 All Nodes, 🖥 Available, 🖥 Busy, 🖥 Offline, 🖥 Down), Cores, and Available Threads, and Actions (Deploy, Destroy).
- **Configuration** - Cluster Information (Cluster ID, Head Node Size, Image Name, Cluster IP, SSH Username, Date/Time Created, and Download Cluster SSH Key), Cluster Compute Nodes (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 (🗀) at the top of the screen.

2.6 File Manager

You can view, download, and delete 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:



Torque ID	Nodus ID	Name	Owner	Size(B)	Modified ↓	Actions
3	3	test-stdout-stderr	centos	77	June 3, 2020 3:24 PM	···

The **File Manager** screen shows this information: Torque ID, Nodus ID, Name, 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.

2.7 Bursting

This section contains the following information:

- Bursting Configurations
- Bursting Service

2.7.1 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.

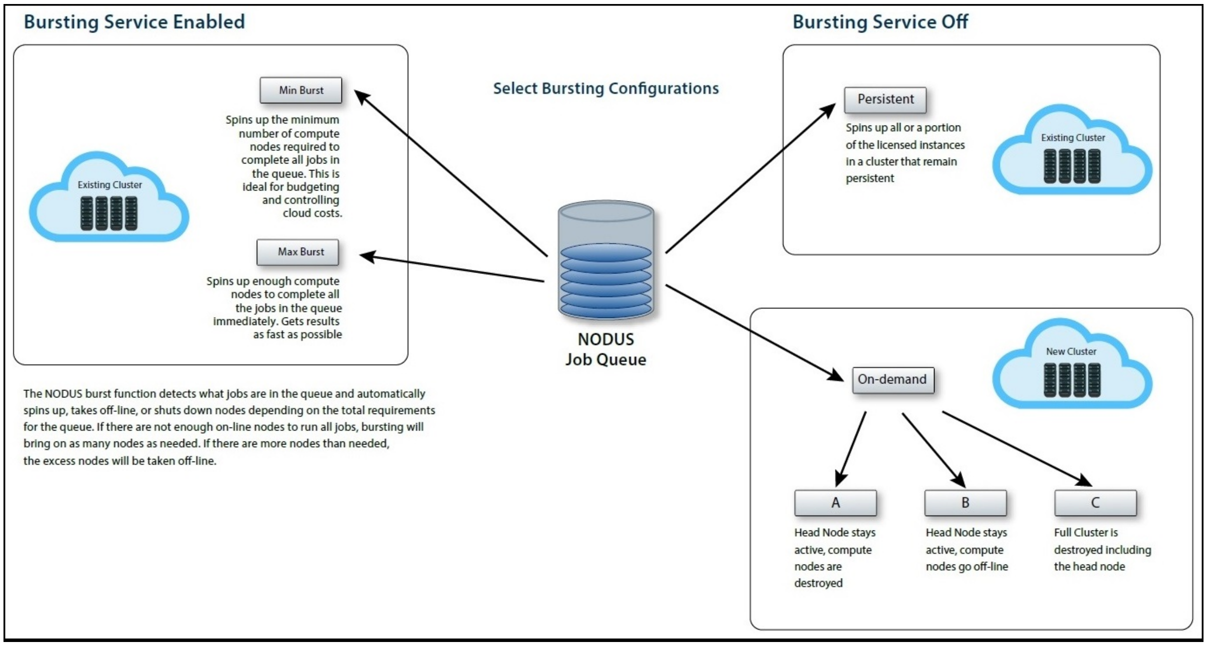
Min Burst spins up the minimum number of compute nodes required to complete all jobs in the queue, which is ideal for budgeting and controlling cloud costs.

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.



Spins up the minimum number of compute nodes required to complete all jobs in the queue. This is ideal for budgeting and controlling cloud costs.

Spins up enough compute nodes to complete all the jobs in the queue immediately. Gets results as fast as possible

Spins up all or a portion of the licensed instances in a cluster that remain persistent

Head Node stays active, compute nodes are destroyed

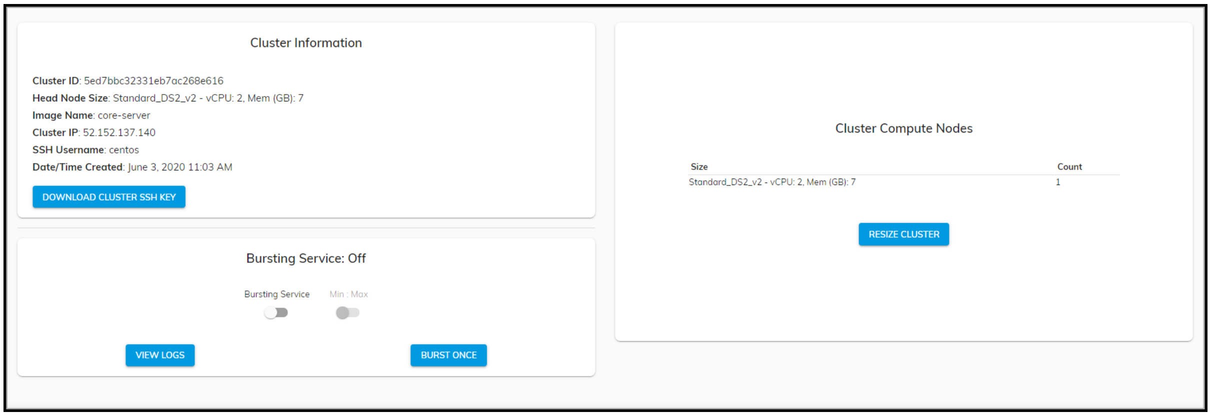
Head Node stays active, compute nodes go off-line

Full Cluster is destroyed including the head node

The NODUS burst function detects what jobs are in the queue and automatically spins up, takes off-line, or shuts down nodes depending on the total requirements for the queue. If there are not enough on-line 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 off-line.

2.7.2 Bursting Service

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



Cluster Information

Cluster ID: 5ed7bbc32331eb7ac268e616
Head Node Size: Standard_DS2_v2 - vCPU: 2, Mem (GB): 7
Image Name: core-server
Cluster IP: 52.152.137.140
SSH Username: centos
Date/Time Created: June 3, 2020 11:03 AM

Bursting Service: Off

Bursting Service ○ Min : Max ○

Cluster Compute Nodes

Size	Count
Standard_DS2_v2 - vCPU: 2, Mem (GB): 7	1

Bursting Service (Off/On) - 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 (as set from Job Manager) 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.

2.8 Disaster Recovery

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

2.9 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 Workload Considerations for additional information.

Chapter 3: NODUS CLI Procedures

In this chapter:

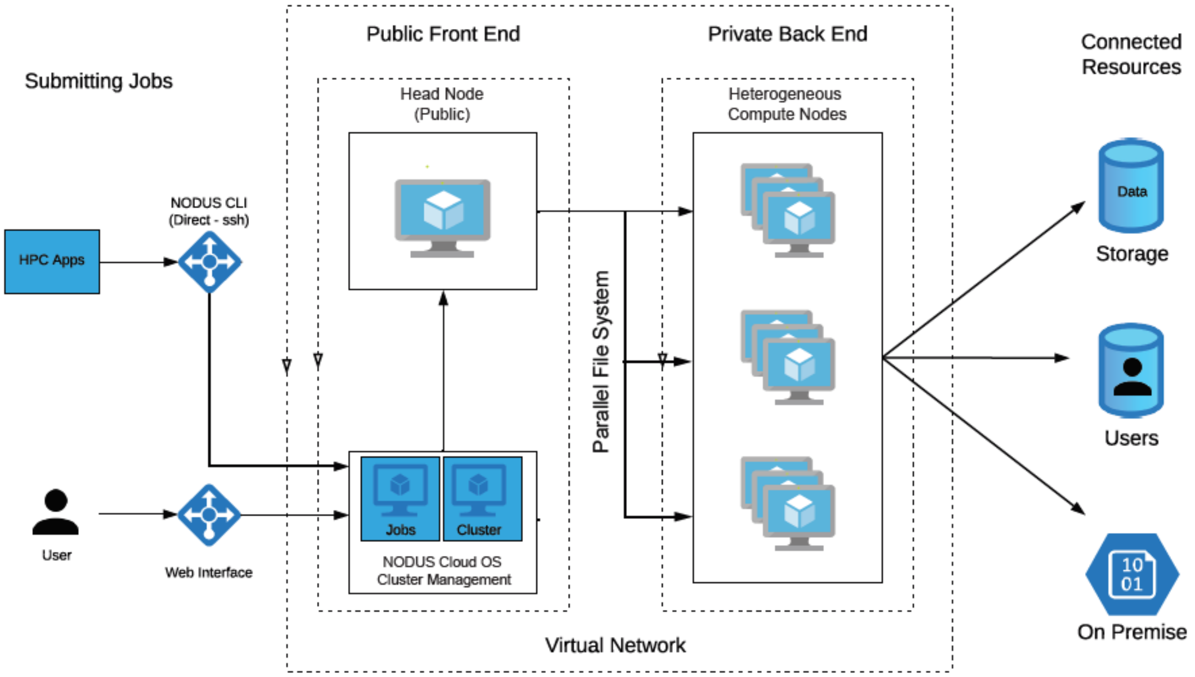
3.1 Users

A NODUS user is just a registered Linux system user that enables using NODUS in various capacities, permissions, and NODUS license permitting.

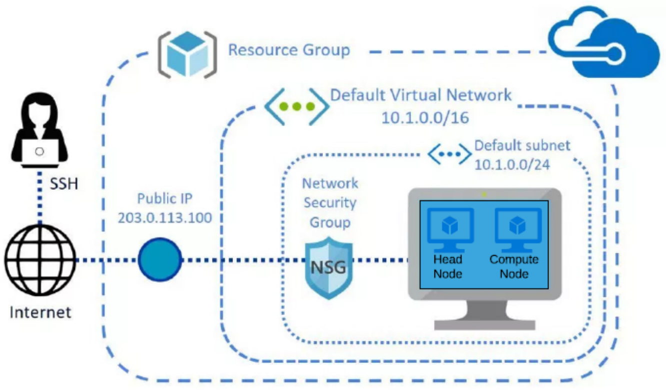
See the CLI Commands section Users for information about user commands.

3.2 Architecture

Typical Cloud Provider



NODUS Cluster Resources



3.3 CLI Usage

This section contains the following information:

- Jobs
- Building Images

3.3.1 Jobs

Show standard error messages: **$ nodus job:output ID CLUSTER [OPTIONS]**

3.3.2 Building Images

After a cluster is associated with a set of credentials and a stack file, use the command **$ nodus stack:build STACK [OPTIONS]** to create the image.

3.4 Cluster Actions

This section contains the following information:

- Connecting to a Cluster
- Destroying a Cluster
- Viewing All Clusters

3.4.1 Connecting to a Cluster

1. To connect to a deployed cluster via SSH, use the command **nodus cluster:ssh CLUSTER**.

3.4.2 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 [OPTIONS]**.
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, use the command **nodus cluster:remove CLUSTER**.

3.4.3 Viewing All Clusters

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

3.5 Job Actions

1. To submit a job to a cluster, use the command **nodus job:submit JOB CLUSTER**.

3.6 Bursting Actions

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, use the command **$ nodus cluster:burst CLUSTER --start**.
2. To stop the bursting service, use the command **$ nodus cluster:burst CLUSTER --stop**.
3. To run a single burst cycle, use the command **$ nodus cluster:burst CLUSTER**.

3.7 Common CLI Commands

This section contains common NODUS CLI commands:

- Cluster Management
- Cluster Monitoring
- Cluster Jobs
- Admin Commands

See the chapter NODUS CLI Commands for detailed information.

3.7.1 Cluster Management

This section contains the following information:

- Cluster Deploy
- Cluster Destroy
- Cluster SSH
- Cluster Resize

3.7.1.A Cluster Deploy

This deploys nodes on a cluster:

```
$ nodus cluster:deploy CLUSTER --target HOSTNAME
```

See cluster:deploy for additional information.

3.7.1.B Cluster Destroy

This destroys nodes on a cluster:

```
$ nodus cluster:destroy CLUSTER --target HOSTNAME
```

See cluster:destroy for additional information.

3.7.1.C Cluster SSH

This connects to a cluster:

```
$ nodus cluster:ssh CLUSTER [OPTIONS]
```

See cluster:ssh for additional information.

3.7.1.D Cluster Resize

This resizes a cluster:

```
$ nodus cluster:resize CLUSTER [[INSTANCE COUNT] ...] [OPTIONS]
```

See cluster:resize for additional information.

3.7.2 Cluster Monitoring

This section contains the following information:

- Cluster Show
- Cluster Jobs Queue
- Cluster List
- Cluster Members
- Cluster pbsnodes

3.7.2.A Cluster Show

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

$ nodus cluster:show CLUSTER

3.7.2.B Cluster Jobs Queue

This displays JSON information about the jobs queue:

$ nodus cluster:qstat CLUSTER

3.7.2.C Cluster List

This lists all user clusters:

$ nodus cluster:list

3.7.2.D Cluster Members

This displays serf members from the head node:

$ nodus cluster:members CLUSTER [OPTIONS]

3.7.2.E Cluster pbsnodes

This displays a detailed description of compute nodes on the cluster:

$ nodus cluster:pbsnodes CLUSTER [OPTIONS]

3.7.3 Cluster Jobs

This section contains the following information:

- Cluster Burst
- Job Submit

3.7.3.A Cluster Burst

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

```
$ nodus cluster:burst CLUSTER [OPTIONS] --mode max --start
```

Note: [--mode max] is optional.

3.7.3.B Job Submit

This remotely runs a local job script on a cluster:

```
$ nodus job:submit JOB CLUSTER [OPTIONS]
```

3.7.4 Admin Commands

This section contains the following information:

- Delete User

3.7.4.A Delete User

This deletes a user from the system:

```
sudo nodus admin delete-users <username>
```

Chapter 4: NODUS CLI Commands

This chapter contains a list of CLI commands.

In this chapter:

4.1 Actions

This section contains the following commands:

- action:list
- action:remove
- action:show

action:list

Description	
List action entries from database.	
Usage	
$ nodus action:list [OPTIONS]	
Arguments	
NAME	Name Identifier
TYPE	(centos-7) Image Type
CLUSTER	Cluster Name or ID

Options		
-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

	--id=id	Filter by Affected ID
	--type=action|cluster|credential |job|provider|stack|user	Filter by Action Type

Examples

```
$ nodus action:list
$ nodus action:list --format json
```

action:remove

Description

Remove action entries from database.

Usage

```
$ nodus action:remove ACTION [[ACTION] …] [OPTIONS]
```

Argument

ACTION	Action ID

Option

-h	--help	Show Help

Example

```
action:remove 5e8f6a8ac86619500ab573da
```

action:show

Description

Display details of a particular action entry from the database.

Usage

```
$ nodus action:show ACTION [OPTIONS]
```

Argument

ACTION	Action ID

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus action:show 5ea06014df12871f264716eb
$ nodus action:show 5ea06014df12871f264716eb --format json
```

4.2 Users

This section contains the following commands:

- user:add
- user:list
- user:remove
- user:show
- user:update

user:add

Description

Registers linux user as a NODUS user.

Usage

```
$ nodus user:add USERNAME [OPTIONS]
```

Argument

USERNAME	Linux Username

Options

-e	--email=email	Email Address
-f	--first=first	First Name (Given Name)
-h	--help	Show Help
-l	--last=last	Last Name (Family Name)
	--[no-]admin	Make User Admin
	--[no-]ui	Allow User to Login via Web App / UI

Examples

```
$ nodus user:add jdoe
$ nodus user:add jdoe --admin
```

user:list

Description

List user entries from database.

Usage

```
$ nodus user:list [OPTIONS]
```

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help
Examples		
$ nodus user:list		
$ nodus user:list --format json		

user:remove

Description		
Remove NODUS users.		
Usage		
$ nodus user:remove USERNAME [OPTIONS]		
Argument		
USERNAME	Username of the system user	
Options		
-h	--help	Show Help
	--clean	Remove User Home Directory from Filesystem
	--force	Force Remove User Home Directory from Filesystem
Examples		
$ nodus user:remove jdoe		
$ nodus user:remove jdoe --clean		
$ nodus user:remove jdoe --force		

user:show

Description		
Display details of a particular user entry from the database.		
Usage		
$ nodus user:show USER [OPTIONS]		
Argument		
USERNAME	User Name or ID	
Options		
-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help
Examples		
$ nodus user:show jdoe		
$ nodus user:show jdoe --format json		

user:update

Description

Update NODUS user details.

Usage

$ nodus user:update USERNAME [OPTIONS]

Argument

USERNAME	Linux Username

Options

-a	--[no-]admin	Make User Admin
-e	--email=email	Email Address
-f	--first=first	First Name (Given Name)
-h	--help	Show Help
-l	--last=last	Last Name (Family Name)
-u	--[no-]ui	Allow User to Login via Web App / UI
	--pass	Change Password

Example

$ nodus user:update jdoe --admin

4.3 Credentials

This section contains the following commands:

- credential:add
- credential:list
- credential:remove
- credential:show

credential:add

Description

Create credential entry with information from file.

Usage

$ nodus credential:add FILE NAME --provider PROVIDER

Arguments

FILE	Credential File (Absolute / Relative) Path

NAME	Credential Name

Options

-d	--description=description	Description of Credential
-h	--help	Show Help
-p	--provider=aws|gcp|huawei|azure|oracle|otc	(required) Provider Code

Examples

```
$ nodus credential:add /home/user/Downloads/credential.json myCredential --provider aws
```

```
$ nodus credential:add ~/Downloads/credential.json myCredential --provider aws
```

credential:list

Description

List credential entries from database.

Usage

```
$ nodus credential:list [OPTIONS]
```

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus credential:list [OPTIONS]
```

```
$ nodus credential:list --format json
```

credential:remove

Description

Remove credential entries from database.

Usage

```
$ nodus credential:remove CREDENTIAL [OPTIONS]
```

Argument

CREDENTIAL	Credential Name or ID

Options

-f	--force	Force Removal from Database
-h	--help	Show Help

Example

```
credential:remove myCredential
```

credential:show

Description		
Display details of a particular credential entry from the database.		
Usage		
$ nodus credential:show CREDENTIAL [OPTIONS]		
Argument		
CREDENTIAL	Credential Name or ID	
Options		
-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help
Examples		
$ nodus credential:show myCredential		
$ nodus credential:show myCredential --format json		

4.4 Stacks

This section contains the following commands:

- stack:aws:create
- stack:azure:create
- stack:gcp:create
- stack:huawei:create
- stack:oracle:create
- stack:otc:create
- stack:set
- stack:build
- stack:log
- stack:list
- stack:show
- stack:remove

stack:aws:create

Description

Create new Amazon Web Services (AWS) stack configuration entry.

Usage

```
$ nodus stack:aws:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Associated Name
TYPE	(centos-7) Image Type

Options

-c	--custom	Use Managed Image
-h	--help	Show Help
-i	--instance=instance	(required) [default: t2.micro] Instance Type
-p	--prefix=prefix	(required) [default: nodus] Output Image Prefix
-r	--region=region	(required) Region ID
-s	--source=source	Source Image ID
-u	--username=username	SSH Username
	--proxy=proxy	HTTP(S) Proxy URL
	--security=security	Security Group ID
	--subnet=subnet	Subnet ID
	--volume=volume	Storage Volume (GB)

Examples

```
$ nodus stack:aws:create myStack centos-7 --region=us-east-2
```

```
$ nodus stack:aws:create myStack centos-7 --prefix=myNodus --region=us-east-2 --instance=t2.large
```

```
$ nodus stack:aws:create myStack centos-7 --region us-east-2 --proxy $http_proxy
```

```
$ nodus stack:aws:create myStack centos-7 --region=us-east-2 --custom --source=ami-04152c3a27c49a944 --username=ec2-user --instance=t2.xlarge
```

stack:azure:create

Description

Create new Microsoft Azure stack configuration entry.

Usage

```
$ nodus stack:azure:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Associated Name
TYPE	(centos-7) Image Type

Options		
-c	--custom	Use Managed Image
-h	--help	Show Help
-i	--instance=instance	(required) [default: Standard_DS2_v2] Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Output Image Prefix
-s	--source=source	Source Image ID
-u	--username=username	SSH Username
	--os-type=os-type	Source Image OS Type
	--publisher=publisher	Source Image Publisher
	--resource=resource	(required) Managed Resource Group Name
	--sku=sku	Source Image SKU
	--source-resource=source-resource	Source Resource Group Name

Examples
$ nodus stack:azure:create myStack centos-7 --resource=myResourceGroup --instance-e=Standard_DS2_v2
$ nodus stack:azure:create myStack centos-7 --prefix=myNodus --resource-e=myResourceGroup --instance=Standard_DS2_v2

stack:gcp:create

Description
Create new Google Cloud Platform (GCP) stack configuration entry.

Usage
$ nodus stack:gcp:create NAME TYPE [OPTIONS]

Arguments	
NAME	Associated Name
TYPE	(centos-7) Image Type

Options		
-c	--custom	Managed Image
-d	--disk=disk	Disk Size (GB)
-h	--help	Show Help
-i	--instance=instance	(required) [default: n1-standard-1] Instance Type ID
-n	--subnet=subnet	Subnet Self Link ID
-p	--prefix=prefix	(required) [default: nodus] Output Image Prefix
-s	--source=source	Source AMI ID
-u	--username=username	SSH Username

-z	--zone=zone	(required) Availability Zone ID
	--account	Auth via Service Account

Examples

```
$ nodus stack:gcp:create myStack centos-7 --zone=us-east1-b --instance=n1-standard-1
```

```
$ nodus stack:gcp:create myStack centos-7 --prefix=myNodus --zone=us-east1-b --instance-
e=n1-standard-1
```

stack:huawei:create

Description

Create new Huawei Cloud stack configuration entry.

Usage

```
$ nodus stack:huawei:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Associated Name
TYPE	(centos-7) Image Type

Options

-c	--custom	Use Managed Image
-h	--help	Show Help
-i	--instance=instance	(required) [default: s2.medium.2] Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Output Image Prefix
-r	--region=region	(required) Region ID
-s	--source=source	Source Image ID
-u	--username=username	SSH Username
-v	--volume=volume	Storage Volume (GB)
-z	--zone=zone	(required) Availability Zone ID
	--security=security	Security Group ID
	--subnet=subnet	(required) Subnet Network ID

Examples

```
$ nodus stack:huawei:create myStack centos-7 --region=ap-southeast-1 --zone=ap-
southeast-1a --subnet=41dcc08c-b044-4af1-b298-8770a479d60a --instance=s2.medium.2
```

```
$ nodus stack:huawei:create myStack centos-7 --prefix=myNodus --region=ap-southeast-1
--zone=ap-southeast-1a --subnet=41dcc08c-b044-4af1-b298-8770a479d60a --
instance=s2.medium.2
```

stack:oracle:create

Description

Create new Oracle Cloud stack configuration entry.

Usage

```
$ nodus stack:oracle:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Associated Name
TYPE	(centos-7) Image Type

Options

-c	--custom	Use Managed Image
-h	--help	Show Help
-i	--instance=instance	(required) [default: VM.Standard2.1] Instance Type ID
-n	--subnet=subnet	(required) Subnet OCID
-p	--prefix=prefix	(required) [default: nodus] Output Image Prefix
-r	--region=region	(required) Region ID
-s	--source=source	Source Image ID
-u	--username=username	SSH Username

Example

```
$ nodus stack:oracle:create myStack centos-7 --prefix=myNodus --region=us-ashburn-1 --subnet=ocid1.subnet.oc1.iad.aaaaaaaa7v2znysxlclowk7rhgkgldvjaxmkkjpiktnahnusgknhw5pcvi6q
```

stack:otc:create

Description

Create new OpenTelekom Cloud stack configuration entry.

Usage

```
$ nodus stack:otc:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Associated Name
TYPE	(centos-7) Image Type

Options

-c	--custom	Use Managed Image
-h	--help	Show Help
-i	--instance=instance	(required) [default: s2.medium.2] Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Output Image Prefix

-s	--source=source	Source Image ID
-u	--username=username	SSH Username
-v	--volume=volume	Storage Volume (GB)
	--security=security	Security Group ID
	--subnet=subnet	Subnet ID

Examples

```
$ nodus stack:otc:create myStack centos-7 --subnet 3bc69be8-3ebc-434c-b828-9ae-b5a8ec78a --instance s2.medium.2
```

```
$ nodus stack:otc:create myStack centos-7 --prefix myNodus --subnet 3bc69be8-3ebc-434c-b828-9aeb5a8ec78a --instance s2.medium.2
```

stack:set

Description

Associate stack entry with credential entry.

Usage

```
$ nodus stack:set STACK CREDENTIAL [OPTIONS]
```

Arguments

STACK	Stack Name or ID
CREDENTIAL	Credential Name or ID

Option

-h	--help	Show Help

Example

```
$ nodus stack:set myStack myCredential
```

stack:build

Description

Generate images compatible with NODUS clusters. Generate two images: 'nodus-server'; 'nodus-node'.

Usage

```
$ nodus stack:build STACK [OPTIONS]
```

Argument

STACK	Stack Name or ID

Options

-S	--slow	Sequential Builds
-h	--help	Show Help

-s	--stack=server|node	Sub-Stack

Examples

```
$ nodus stack:build myStack
$ nodus stack:build myStack --stack server
```

stack:log

Description

Display stack build logs.

Usage

```
$ nodus stack:log CLUSTER TYPE [OPTIONS]
```

Arguments

STACK	Stack Name or ID
TYPE	(master|build) [default: master] Log Type

Options

-d	--debug	Debug Logs
-h	--help	Show Help
-n	--lines=lines	Last N Lines
	--follow	Follow File Output Stream

Example

```
$ nodus stack:log myCluster
```

stack:list

Description

List stack entries from database.

Usage

```
$ nodus stack:list [OPTIONS]
```

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus stack:list
$ nodus stack:list --format json
```

stack:show

Description		
Display details of a particular stack entry from the database.		
Usage		
$ nodus stack:show STACK [OPTIONS]		
Argument		
STACK	Stack Name or ID	
Options		
-f	--format=text|json|xml	[default: text] Output Format
-h	--help	Show Help
Examples		
$ nodus stack:show myStack		
$ nodus stack:show myStack --format json		

stack:remove

Description		
Remove stack entries from database.		
Usage		
$ nodus stack:remove STACK [OPTIONS]		
Argument		
STACK	Stack Name or ID	
Options		
-h	--help	Show Help
	--[no-]clean	Remove Files Created for Cluster
Examples		
$ nodus stack:remove myStack		
$ nodus stack:remove myStack --no-clean		
$ nodus stack:remove myStack --force		

4.5 Clusters

This section contains the following commands:

- cluster:aws:create
- cluster:azure:create
- cluster:burst
- cluster:copy
- cluster:deploy
- cluster:destroy
- cluster:gcp:create
- cluster:huawei:create
- cluster:list
- cluster:log
- cluster:members
- cluster:oracle:create
- cluster:otc:create
- cluster:pbsnodes
- cluster:qstat
- cluster:releasehold
- cluster:remove
- cluster:resize
- cluster:set
- cluster:show
- cluster:ssh

cluster:aws:create

Description

Create new Amazon Web Services (AWS) cluster configuration entry.

Usage

```
$ nodus cluster:aws:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Name Identifier
TYPE	(centos-7) Image Type

Options

-h	--help	Show Help
-i,	--instance=instance	(required) Server Instance Type ID
-p,	--prefix=prefix	(required) [default: nodus] Source Image Root Prefix (example: nodus)
-r,	--region=region	(required) Region ID
	--burstOnDeploy-y=burstOnDeploy	Initiate Burst Service post Deploy
	--role	Auth via EC2 Role
	--security=security	Security Group ID
	--subnet=subnet	Subnet ID
	--vpc=vpc	VPC Network Self Link ID

Examples

```
$ nodus cluster:aws:create myCluster centos-7 --region=us-east-2 --instance=t2.large
```

```
$ nodus cluster:aws:create myCluster centos-7 --region=us-east-2 --instance=t2.large --vpc=vpc-03ceecbf61bb37c99 --subnet=subnet-05aa121e98148d0ce --security=sg-09cda11597d6fd926
```

cluster:azure:create

Description

Create new Microsoft Azure cluster configuration entry.

Usage

```
$ nodus cluster:azure:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Name Identifier
TYPE	(centos-7) Image Type

Options

-R	--resource=resource	(required) Image Resource Group Name
-h	--help	Show Help
-i,	--instance=instance	(required) [default: Standard_DS2_v2] Server Instance Type ID
-p,	--prefix=prefix	(required) [default: nodus] Source Image Root Prefix (example: nodus)
-r,	--region=region	(required) Region ID
	--burstOnDeploy-y=burstOnDeploy	Initiate Burst Service post Deploy

Example

```
$ nodus cluster:azure:create myCluster centos-7 --prefix=myNodus --region=eastus --resource=myResourceGroup
```

cluster:burst

Description

Manage cluster bursting state or manual burst.

Usage

```
$ nodus cluster:burst CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-f	--force	Skip State Checks
-h	--help	Show Help
-m	--mode=min|max	Bursting Mode
	--raw	Raw Details
	--start	Start bursting service Daemon
	--status	Check Service Status
	--stop	Stop bursting service Daemon

Examples

```
$ nodus cluster:burst myCluster
$ nodus cluster:burst myCluster --start
$ nodus cluster:burst myCluster --stop
$ nodus cluster:burst myCluster --status
$ nodus cluster:burst myCluster --mode max --start
```

cluster:copy

Description

Updates NODUS Cluster provisioning files. Used for version upgrades, maintenance, and development purposes.

Usage

```
$ nodus cluster:copy CLUSTER TYPE [OPTIONS]
```

Arguments

CLUSTER	Cluster Name or ID
TYPE	(api|scripts) Provision File Type

Option

-h	--help	Show Help
Examples		
$ nodus cluster:copy myCluster api		
$ nodus cluster:copy myCluster scripts		

cluster:deploy		
Description		
Modify cloud resources based on cluster configuration entry.		
Usage		
$ nodus cluster:deploy CLUSTER [OPTIONS]		
Argument		
CLUSTER	Cluster Name or ID	
Options		
-h	--help	Show Help
-t	--target=target	Node Hostname
	--[no-]plan	Show Plan
Examples		
$ nodus cluster:deploy myCluster		
$ nodus cluster:deploy myCluster --no-plan		

cluster:destroy		
Description		
Destroy cloud resources based on cluster configuration entry.		
Usage		
$ nodus cluster:destroy CLUSTER [OPTIONS]		
Argument		
CLUSTER	Cluster Name or ID	
Options		
-f	--force	Skip Checks
-h	--help	Show Help
-t	--target=target	Node Hostname
	--plan	Show Plan
Examples		
$ nodus cluster:destroy myCluster		
$ nodus cluster:destroy myCluster --no-plan		

cluster:gcp:create

Description

Create new Google Cloud Platform (GCP) cluster configuration entry.

Usage

```
$ nodus cluster:gcp:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Name Identifier
TYPE	(centos-7) Image Type

Options

-h	--help	Show Help
-i	--instance=instance	(required) [default: n1-standard-1] Server Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Source Image Root Prefix (example: nodus)
-r	--region=region	(required) Region ID
-z	--zone=zone	(required) Availability Zone ID
	--account	Auth via Service Account
	--burstOnDeploy-y=burstOnDeploy	Initiate Burst Service post Deploy
	--firewall=firewall	Firewall Self Link ID
	--subnet=subnet	Subnet Self Link ID
	--vpc=vpc	VPC Network Self Link ID

Example

```
$ nodus cluster:gcp:create myCluster centos-7 --prefix=myNodus --region=us-east1 --zone=us-east1-b
```

cluster:huawei:create

Description

Create new Huawei Cloud cluster configuration entry.

Usage

```
$ nodus cluster:huawei:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Name Identifier
TYPE	(centos-7) Image Type

Options

-h	--help	Show Help

-i	--instance=instance	(required) [default: s2.medium.2] Server Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Source Image Root Prefix (example: nodus)
-r	--region=region	(required) Region ID
-z	--zone=zone	(required) Availability Zone ID
	--burstOnDeploy-y=burstOnDeploy	Initiate Burst Service post Deploy

Examples

```
$ nodus cluster:huawei:create myCluster centos-7 --prefix=myNodus --instance=s2.medium.2
```

cluster:list

Description

List cluster entries from database.

Usage

```
$ nodus cluster:list [OPTIONS]
```

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus cluster:list
```

```
$ nodus cluster:list --format json
```

cluster:log

Description

Display cluster logs.

Usage

```
$ nodus cluster:log CLUSTER TYPE [OPTIONS]
```

Arguments

CLUSTER	Cluster Name or ID
TYPE	(init|plan|deploy|destroy|master|burst) [default: master] Log Type

Options

-d	--debug	Debug Logs
-f	--follow	Follow File Output Stream

-h	--help	Show Help
-n	--lines=lines	Last N Lines

Examples

```
$ nodus cluster:log myCluster
$ nodus cluster:log myCluster burst
```

cluster:members

Description

Get cluster information as reported by 'serf members'.

Usage

```
$ nodus cluster:members CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Example

```
$ nodus cluster:members myCluster
```

cluster:oracle:create

Description

Create new Oracle Cloud cluster configuration entry.

Usage

```
$ nodus cluster:oracle:create NAME [OPTIONS]
```

Arguments

NAME	Name Identifier
TYPE	(centos-7) Image Type

Options

-h	--help	Show Help
-i	--instance=instance	(required) Server Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Source Image Root Prefix (example: nodus)
-r	--region=region	(required) Region ID
	--burstOnDeploy-y=burstOnDeploy	Initiate Burst Service post Deploy

Example

```
$ nodus cluster:oracle:create myCluster --prefix=myNodus
```

cluster:otc:create

Description

Create new OpenTelekom Cloud cluster configuration entry.

Usage

```
$ nodus cluster:otc:create NAME TYPE [OPTIONS]
```

Arguments

NAME	Name Identifier
TYPE	(centos-7) Image Type

Options

-h	--help	Show Help
-i	--instance=instance	(required) [default: s2.medium.2] Server Instance Type ID
-p	--prefix=prefix	(required) [default: nodus] Source Image Root Prefix (example: nodus)
	--burstOnDeploy-y=burstOnDeploy	Initiate Burst Service post Deploy

Example

```
$ nodus cluster:otc:create myCluster centos-7 --prefix=myNodus --instance=s2.medium.2
```

cluster:pbsnodes

Description

Get cluster information as reported by 'pbsnodes' via PBS/TROQUE.

Usage

```
$ nodus cluster:pbsnodes CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Example

```
$ nodus cluster:pbsnodes myCluster
```

cluster:qstat

Description

Get cluster information as reported by 'qstat' via PBS/TROQUE.

Usage

```
$ nodus cluster:qstat CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Example

```
$ nodus cluster:qstat myCluster
```

cluster:releasehold

Description

Release all job holds.

Usage

```
$ nodus cluster:releasehold CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Option

-h	--help	Show Help

Example

```
$ nodus cluster:releasehold myCluster
```

cluster:remove

Description

Remove cluster entries from database.

Usage

```
$ nodus cluster:remove CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-h	--help	Show Help

	--[no-]clean	Remove Files Created for Cluster
	--force	Force Removal from Database
Examples		
cluster:remove myCluster		
cluster:remove myCluster --no-clean		
cluster:remove myCluster --force		

cluster:resize

Description		
Alter cluster configuration by modifying instance types and their counts.		
Usage		
$ nodus cluster:resize CLUSTER [[INSTANCE COUNT] ...] [OPTIONS]		
Arguments		
CLUSTER		Cluster Name or ID
INSTANCE		Instance ID Code
COUNT		Number of Instances [0,inf)
Options		
-f	--force	Ignore Checks
-h	--help	Show Help
-p	--purge	Purge All Existing Node Groups Except Given Args
-s	--server=server	New Instance ID Code for Server
Examples		
$ nodus cluster:resize myCluster t2.micro 4 t2.xlarge 1		
$ nodus cluster:resize myCluster t2.nano 5 --purge		

cluster:set

Description		
Associate cluster entry with credential entry.		
Usage		
$ nodus cluster:set CLUSTER CREDENTIAL [OPTIONS]		
Arguments		
CLUSTER		Cluster Name or ID
CREDENTIAL		Credential Name or ID
Option		
-h	--help	Show Help

Example

```
$ nodus cluster:set myCluster myCredential
```

cluster:show

Description

Display details of a particular cluster entry from the database.

Usage

```
$ nodus cluster:show CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-b	--backup	Show TF State Backup
-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help
-k	--keys	Show Keys
-s	--state	Show TF State

Example

```
$ nodus cluster:show myCluster
```

cluster:ssh

Description

SSH into the head node of cluster.

Usage

```
$ nodus cluster:ssh CLUSTER [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-f	--force	Ignore Checks
-h	--help	Show Help

Example

```
$ nodus cluster:ssh myCluster
```

4.6 Jobs

This section contains the following commands:

- job:cancel
- job:create
- job:list
- job:output
- job:remove
- job:show
- job:submit
- job:update

job:cancel

Description		
Cancel Submitted Job.		
Usage		
$ nodus job:cancel ID CLUSTER [OPTIONS]		
Arguments		
ID	Torque Job ID	
CLUSTER	Cluster Name or ID	
Option		
-h	--help	Show Help
Example		
$ nodus job:cancel 0.nodus-server myCluster		

job:create

Description	
Create new job entry.	
Usage	
$ nodus job:create SCRIPT NAME [OPTIONS]	
Arguments	
SCRIPT	(Absolute/Relative) File Path to Job Script

NAME	Job Name

Options

-c	--cores=cores	(required) [default: 1] Number of required cores per node
-d	--description=description	Description of the job
-f	--dataFile=dataFile	(Absolute/Relative) File path to data file
-h	--help	Show Help
-n	--nodes=nodes	(required) [default: 1] Number of required nodes
-w	--walltime=walltime	[default: 24:00:00] Job maximum wall-time

Examples

```
$ nodus job:create ~/Documents/myScript.sh myJob
$ nodus job:create ~/Documents/myScript.sh myJob --nodes=2 --cores=16
$ nodus job:create ~/Documents/myScript.sh myJob --dataFile=~/Documents/data0.dat --dataFile=~/Documents/data1.dat
```

job:list

Description

List job entries from database.

Usage

```
$ nodus job:list [CLUSTER] [OPTIONS]
```

Argument

CLUSTER	Cluster Name or ID

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus job:list
$ nodus job:list --format json
$ nodus job:list myCluster
```

job:output

Description

Display stdout or stderr of from.

Usage

```
$ nodus job:output ID CLUSTER [OPTIONS]
```

Arguments

JOB	NODUS Job ID
CLUSTER	Cluster ID or Name

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help
-t	--type=stdout|stderr	[default: stdout] Output Type
	--remove	Remove Job Directory Instead

Examples

```
$ nodus job:output 0 myCluster
$ nodus job:output 0 myCluster --format json
```

job:remove

Description

Remove job entries from database.

Usage

```
$ nodus job:remove JOB [OPTIONS]
```

Argument

JOB	Job Name or ID

Option

-h	--help	Show Help

Example

```
job:remove myJob
```

job:show

Description

Display job entry.

Usage

```
$ nodus job:show JOB [OPTIONS]
```

Argument

JOB	Job ID or Name

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus job:show myJob
$ nodus job:show myJob --format json
```

job:submit

Description

Submit a job to the cluster.

Usage

```
$ nodus job:submit JOB CLUSTER [OPTIONS]
```

Arguments

JOB	Job Name or ID
CLUSTER	Cluster Name or ID

Option

-h	--help	Show Help

Example

```
$ nodus job:submit myJob myCluster
```

job:update

Description

Update job configuration entry.

Usage

```
$ nodus job:update:update NAME [OPTIONS]
```

Argument

JOB	Job Name

Options

-a	--add-df=add-df	Absolute Path to any data files to be added
-c	--cores=cores	Number of required cores per node
-d	--description=description	Description of the job
-h	--help	Show Help
-n	--nodes=nodes	Number of required nodes
-r	--remove-df=remove-df	Absolute Path to any data files to be removed
-s	--script=script	Absolute File path to job script
-w	--walltime=walltime	Job walltime

Examples

```
$ nodus job:update myJob --description="My Job" --script-
t=/home/jdoe/Documents/myScript.sh
```

```
$ nodus job:update myJob --description="My Job" --
script=/home/jdoe/Documents/myScript.sh --nodes=2 --cores=16 --walltime="00:30:00" -
-remove-df=/home/jdoe/Documents/file0.dat --add-df=/home/jdoe/Documents/file1.dat
```

4.7 Providers

This section contains the following commands:

- provider:instance
- provider:region

provider:instance

Description

List provider instances.

WARNING Non-exhaustive list.

WARNING Availability is dependent on region and demand.

WARNING Eligibility is strictly dependent on your account.

Usage

```
$ nodus provider:instance [OPTIONS]
```

Argument

PROVIDER	(aws|gcp|huawei|azure|oracle|otc) Cloud Provider

Options

-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples

```
$ nodus provider:instance
$ nodus provider:instance --format json
```

provider:region

Description

List provider regions and zones.

WARNING Non-exhaustive list.

WARNING Availability and eligibility is strictly dependent on your account.

Usage

```
$ nodus provider:region [OPTIONS]
```

Argument	
PROVIDER	(aws|gcp|huawei|azure|oracle|otc) Cloud Provider

Options		
-f	--format=text|json	[default: text] Output Format
-h	--help	Show Help

Examples
$ nodus provider:region
$ nodus provider:region --format json

4.8 Misc

This section contains the following commands:

- daemon
- help
- license

daemon

Description
Manage Master Bursting Daemon. Run 'daemon --init' to initialize the bursting daemon.

Usage
$ nodus daemon [OPTIONS]

Options	
--init	Initializes and Starts Bursting Service
--restart	Restart Bursting Service
--start	Start Bursting Service
--status	Check Bursting Service Status
--stop	Stop Bursting Service

Examples
$ nodus daemon --init
$ nodus daemon --status

help

Description

Display help for nodus.

Usage

```
$ nodus help [COMMAND]
```

Argument

COMMAND	Command to show help for

Option

--all	See all commands in CLI

license

Description

Show current license information.

Usage

```
$ nodus license [OPTIONS]
```

Option

--main	Check main license

Example

```
$ nodus license
```

Appendix A: Creating Accounts for NODUS 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 Cloud Service Provider Account Information and Credentials below for additional information.

Appendix B: Obtaining Cloud Service Provider Account Information and Credentials

Follow the checklists below to prepare accounts and get credentials for the chosen cloud service 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" /sr-
c/geant4.10.05.p01/examples/ex-
tended/radioactivedecay/rdecay01/include/HistoManager.hh

RUN cd BUILD; cmake -DGeant4_DIR=/app/geant4.10.5.1-
install/lib/Geant4-10.5.1/ /sr-
c/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 ; /us-
r/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: License Manager

Follow the steps below to create a License Manager account and to view a list of users.

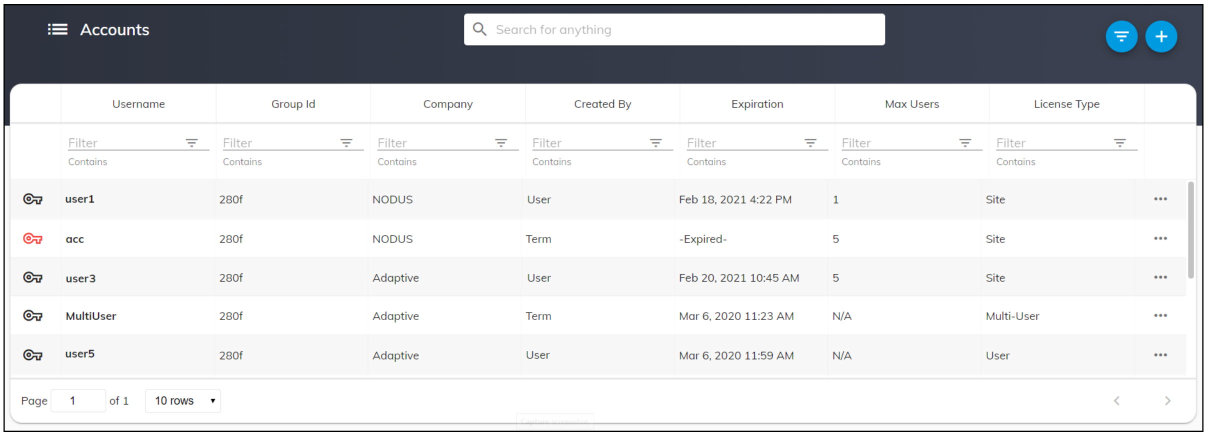
In this appendix:

D.1 Creating a License Manager Account

1. To create a License Manager account, go to `http://<web_server_address>/register`.
2. Follow the instructions and complete the registration process. At the end of this step, you will have your License Manager account 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>`.

D.2 Viewing Accounts

1. Log in to your License Manager account:



	Username	Group Id	Company	Created By	Expiration	Max Users	License Type	
	Filter Contains	Filter Contains	Filter Contains	Filter Contains	Filter Contains	Filter Contains	Filter Contains	
	user1	280f	NODUS	User	Feb 18, 2021 4:22 PM	1	Site	•••
	acc	280f	NODUS	Term	-Expired-	5	Site	•••
	user3	280f	Adaptive	User	Feb 20, 2021 10:45 AM	5	Site	•••
	MultiUser	280f	Adaptive	Term	Mar 6, 2020 11:23 AM	N/A	Multi-User	•••
	user5	280f	Adaptive	User	Mar 6, 2020 11:59 AM	N/A	User	•••

The **Accounts** screen shows this information: Username, Group ID, Company, Created By, Expiration, Max Users, and License Type. You can filter these columns by entering criteria in the Filter field and then at the end of the field, selecting Contains, Starts with, or Ends with.

After creating an account, clicking the ellipsis at the end of the row shows actions that you can take regarding the account: Details, Copy Key, Send Key, and Deactivate.

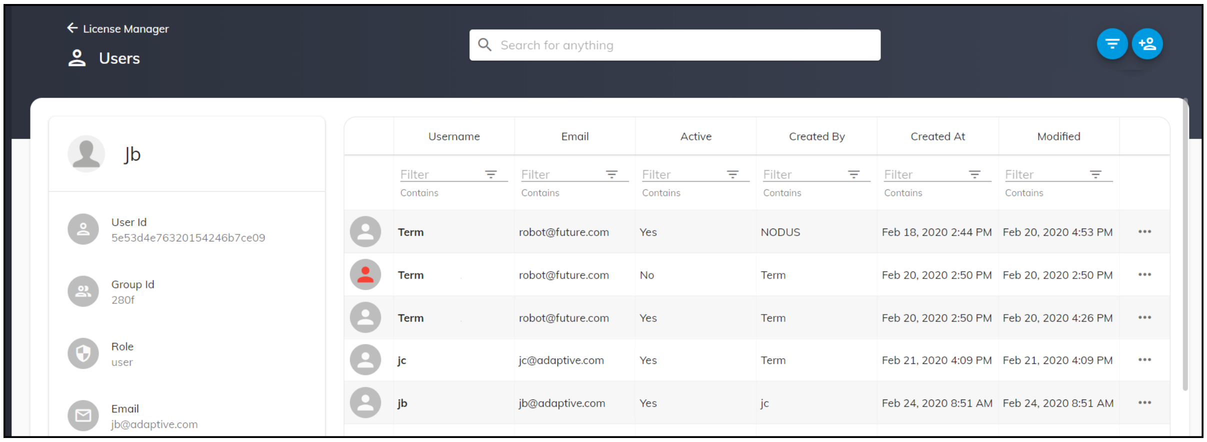
You can filter accounts by clicking the filter icon (☰) and then selecting All, Active, or Inactive.

D.3 Creating a New Account

1. To create a new account, click the add icon (+) to open the **New Account** screen. This screen shows this information: Username, Company, First Name, Last Name, MAC Address, Email, Expiration (Unlimited, 2 Weeks, 1 Month, 6 Months, & 1 Year), License Type (Multi-User, Site, & User), and Max Users (for Site only). This information can be edited after it is saved by clicking the ellipsis / Details / Edit.
2. Enter the required information for the new account and click **Save**.
3. Repeat the steps in this section to create additional accounts as desired.

D.4 Viewing Users

1. To view a list of users, click **Users** next to your user name. The **Users** screen opens:



Jb

User Id
5e53d4e76320154246b7ce09

Group Id
280f

Role
user

Email
jb@adaptive.com

Username	Email	Active	Created By	Created At	Modified	
Filter Contains	Filter Contains	Filter Contains	Filter Contains	Filter Contains	Filter Contains	
Term	robot@future.com	Yes	NODUS	Feb 18, 2020 2:44 PM	Feb 20, 2020 4:53 PM	•••
Term	robot@future.com	No	Term	Feb 20, 2020 2:50 PM	Feb 20, 2020 2:50 PM	•••
Term	robot@future.com	Yes	Term	Feb 20, 2020 2:50 PM	Feb 20, 2020 4:26 PM	•••
jc	jc@adaptive.com	Yes	Term	Feb 21, 2020 4:09 PM	Feb 21, 2020 4:09 PM	•••
jb	jb@adaptive.com	Yes	jc	Feb 24, 2020 8:51 AM	Feb 24, 2020 8:51 AM	•••

This screen shows this information: Username, Email, Active, Created By, Created At, and Modified. You can filter these columns by entering criteria in the Filter field and then at the end of the field, selecting Contains, Starts with, or Ends with.

Clicking the ellipsis at the end of the row shows actions that you can take regarding the account: Details and Deactivate. The password and email can be changed by clicking **Details**.

Appendix E: Configuring NODUS to be a Secure Server

Follow the steps below to configure NODUS to be a secure server.

1. Edit the file **/NODUS/nodus4-ui-desktop-app/build/assets/config.json**:
   - **A.** Replace `http://` with `https://`.
   - **B.** Save the file.
2. Edit the file **/NODUS/nodus4-web-api/.env**:
   - **A.** Add the following lines:

```
HTTPS=true

HTTPS_CERT_FILE=<location_of_the_SSL_certificate_file>

HTTPS_KEY_FILE=<location_of_the_SSL_private_key_file>

HTTPS_CA_FILE=<location_of_the_SSL_certificate_chain_file>
```

   - **B.** Save the file.
3. Enter **/NODUS/nodus4-web-api/restart.sh** to restart the API.

Appendix F: Credential JSON Examples

Credential JSON files should look like the below:

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

AWS

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

Google Cloud

```
{
"type": "",
"project_id": "",
"private_key_id": "",
"private_key": ""
}
```

Huawei Cloud

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

Microsoft Azure

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

OTC

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

Oracle Cloud

```
{
"user_ocid" : "",
"tenancy_ocid" : "",
"compartment_ocid" : "",
"availability_domain": "",
"key_file" : ""
"fingerprint"
}
```

Appendix G: Known Issues and Troubleshooting

This appendix contains information on known issues and troubleshooting:

- Error refreshing state
- WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!
- Error applying plan

Error refreshing state

Issue

While deploying a cluster, the logs display the following error.

```
Error: Error refreshing state: 1 error occurred:

* provider.azurerm: Error building account: Error getting authen-
ticated object ID: Error listing Service Principals: autorest.De-
tailedError{Original:adal.tokenRefreshError{message:"adal: Refresh
request failed. Status Code = '400'. Response body: {\"er-
ror\":\"unauthorized_client\",\"error_descrip-
tion\":\"AADSTS700016: Application with identifier 'f46f18e7-2047-
4b18-b801-1579f731f057' was not found in the directory '82c76c0f-
4c88-4dfa-af71-055fa67876fd'. This can happen if the application
has not been installed by the administrator of the tenant or con-
sented to by any user in the tenant. You may have sent your authen-
tication request to the wrong tenant.
```

Solution

Cross-check the client_secret for the given client_id with the system administrator to determine the issue. The client_secret either no longer exists or has expired; therefore, another client_secret needs to be generated or given to the user, and their credentials file updated to reflect the new client_secret.

WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

Issue

Providers reuse a pool of IP addresses for the head node. You may want to connect to this node via Secure Shell (SSH). Since the SSH key is tied to your account on the NODUS server, a message similar to the following may display when connecting to an IP address that has been reused.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle
attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:TPfwqpRrL2HB34ln4bar5hP+FiN3tTsecR51IBOHRJk.

Please contact your system administrator.
Add correct host key in ~/.ssh/known_hosts to get rid of this mes-
sage.
Offending ECDSA key in ~/.ssh/known_hosts:36
 remove with:
 ssh-keygen -f "~/.ssh/known_hosts" -R "#.#.#.#"
ECDSA host key for #.#.#.# has changed and you have requested
strict checking.
Host key verification failed.
```

Solution

Follow the instructions in your message to correct this issue.

Error applying plan

Issue

While deploying or resizing a cluster, the logs display an error similar to the following.

```
Error: Error applying plan:
1 error occurred:
* oci_core_instance.nodus-cluster-VM-Standard2-2-node-1: 1 error
occurred:
* oci_core_instance.nodus-cluster-VM-Standard2-2-node-1: Service
error:LimitExceeded. The following service limits were exceeded:
vm-standard2-2-count. Request a service limit increase from the
service limits page in the console.
. http status code: 400. Opc request id: 3dd1b-
cd1a550b6f1148ea32ac0986512/0780130618F448D9EB693213AF75B0A9/61396-
56939DE186B0C9C22D172B925DE
```

Solution

The services limits were exceeded by the provider. Clean up existing resources that could be hanging, lost, or currently deployed, and try again.

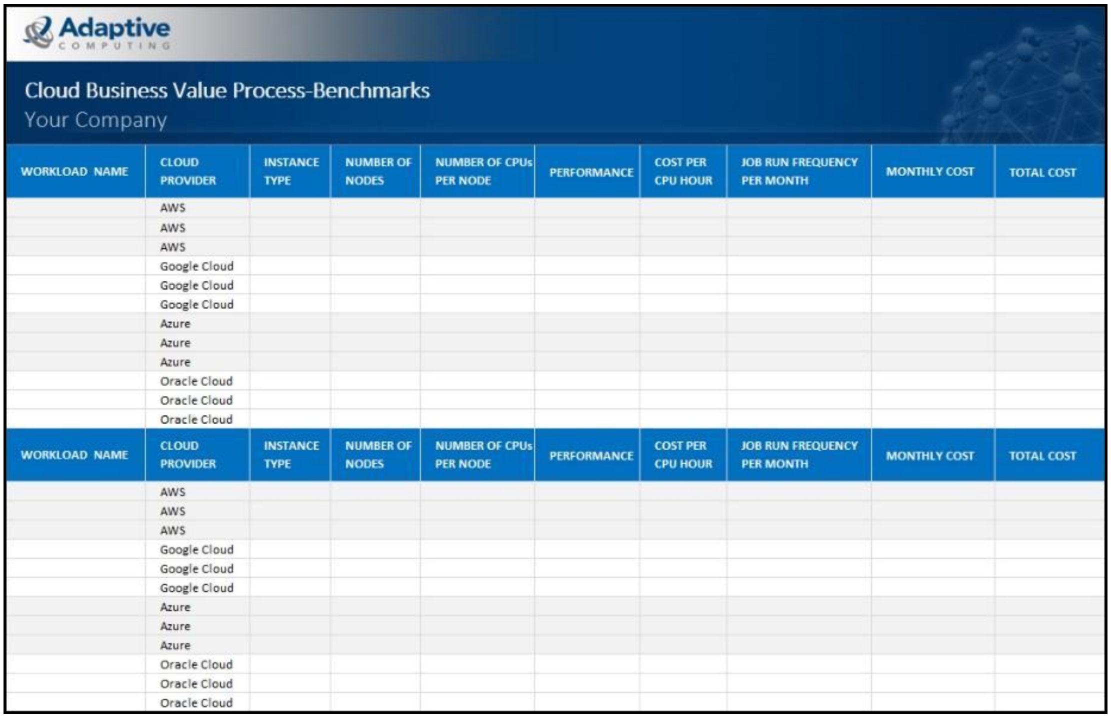
Appendix H: NODUS Workload Considerations

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 service providers.
    - Run your workload on each cloud service 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*. It is also available on adaptivecomputing.com (Support Portal Home / Product Download / NODUS Cloud OS).



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 capital expenses (CapEx) to operational expenses (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 for additional information.

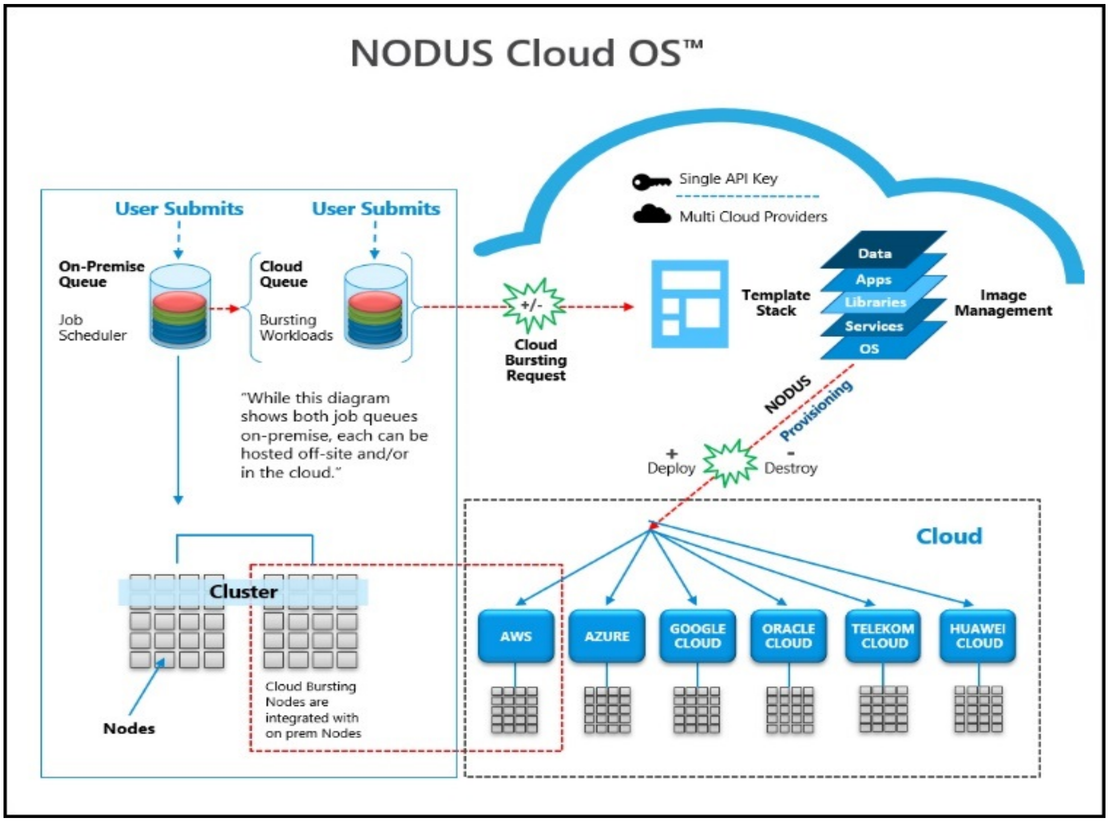


Figure 2: NODUS Cloud OS Bursting Becoming On-Premises

- 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.

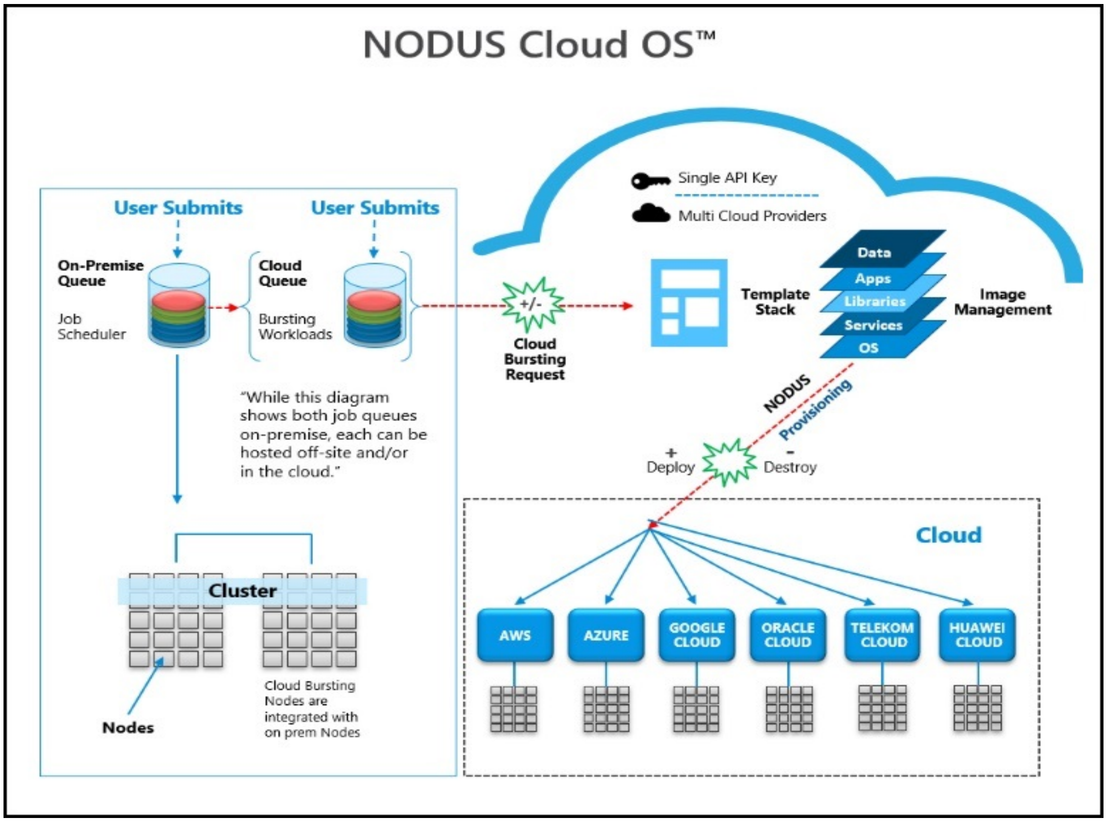


Figure 3: NODUS Cloud OS Bursting Off-Premises

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.

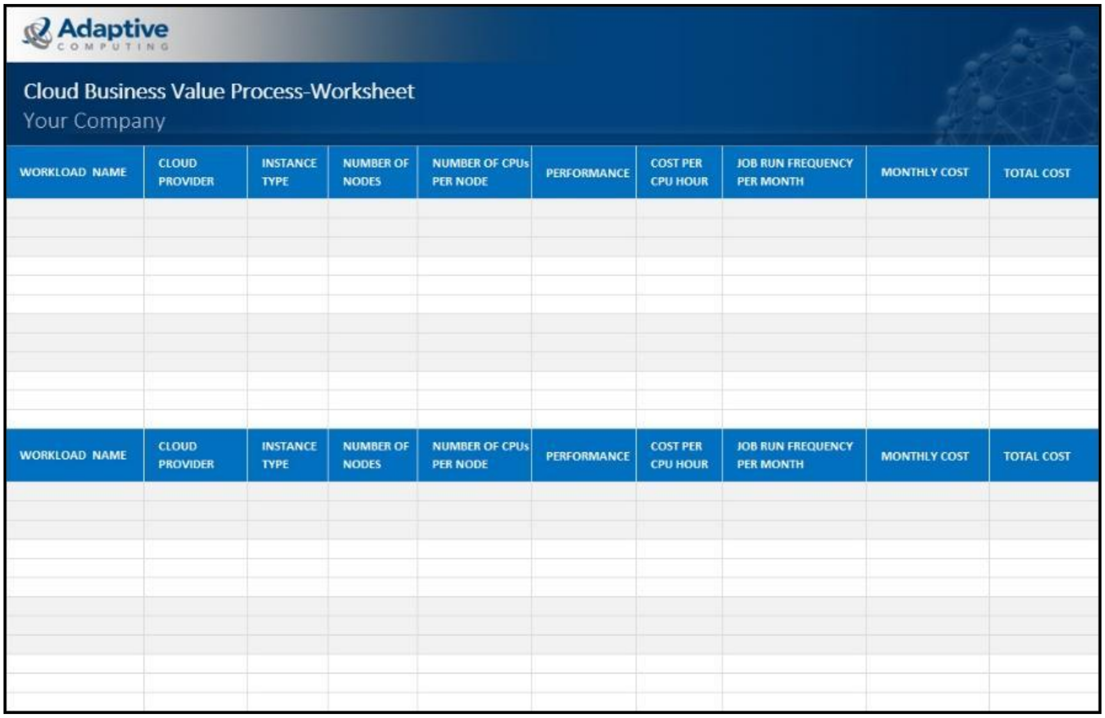
7. Which of your workloads require moving large amounts of data to the cloud and which do not?

- 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 service 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 service providers. Within the NODUS Cloud OS interface, select the cloud service provider that your job will run on. The nodes will shut down when the job completes, and you can then select another cloud service 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 service provider (Figure 1).

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



Cloud Business Value Process-Worksheet

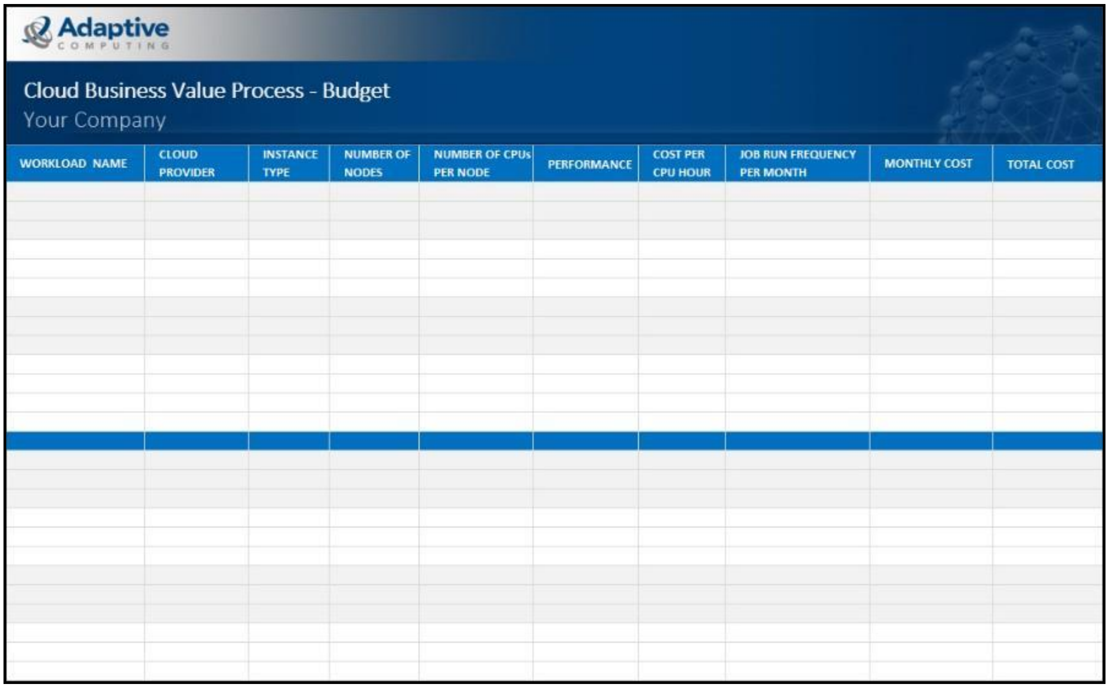
Your Company

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

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

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*.



Cloud Business Value Process - Budget

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

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 service provider using which instance type. We recommend that you run each workload using at least three different instance types for each cloud service 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 being 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, typically designed for fast computations and large amounts of I/O, 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 service 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: A workload submitted to a scheduler for the purpose of scheduling resources on which the workload executes when started up by the scheduler. Typically, a user creates a script that executes the workload (one or more applications) and submits the script to the scheduler where it becomes a job.

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

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.

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

Torque (Terascale Open-source Resource and Queue manager): An industry-standard resource manager solution with a built-in scheduler.

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

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

Index