NODUS Cloud OS
Quick Start Guide 5.0.0

July 2020

Adaptive COMPUTING

Legal Notices

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

Chapter 1: NODUS Cloud OS Platform Installation

This chapter provides information about the installer, which contains the entire NODUS Cloud Platform.

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 via 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* in the NODUS Cloud OS User Guide for additional information.

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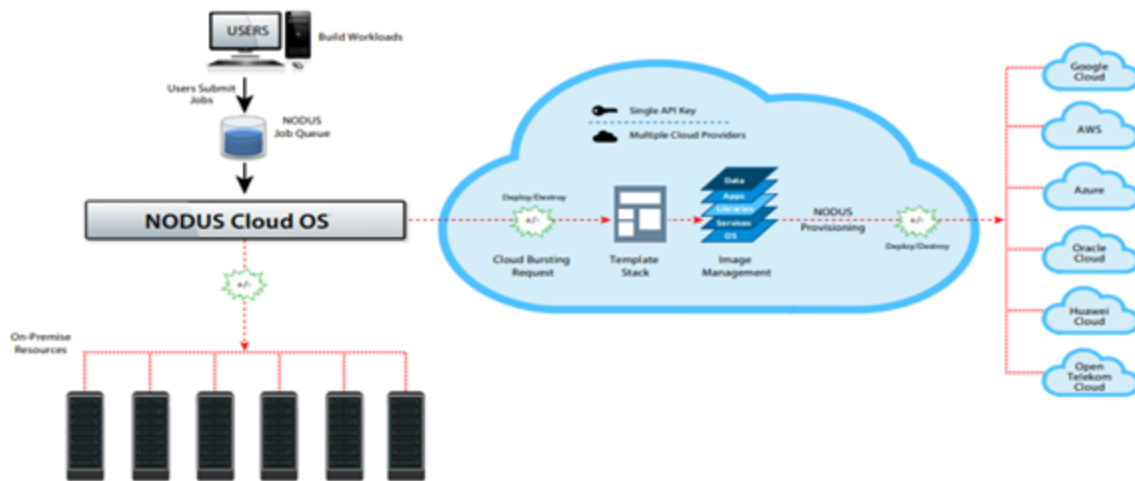
Continue with the steps for your chosen provider below to obtain credentials:

- Google Cloud

- AWS

- Microsoft Azure

- Oracle Cloud

- Huawei Cloud

- Open Telekom Cloud

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.

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 section Stack Commands 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.
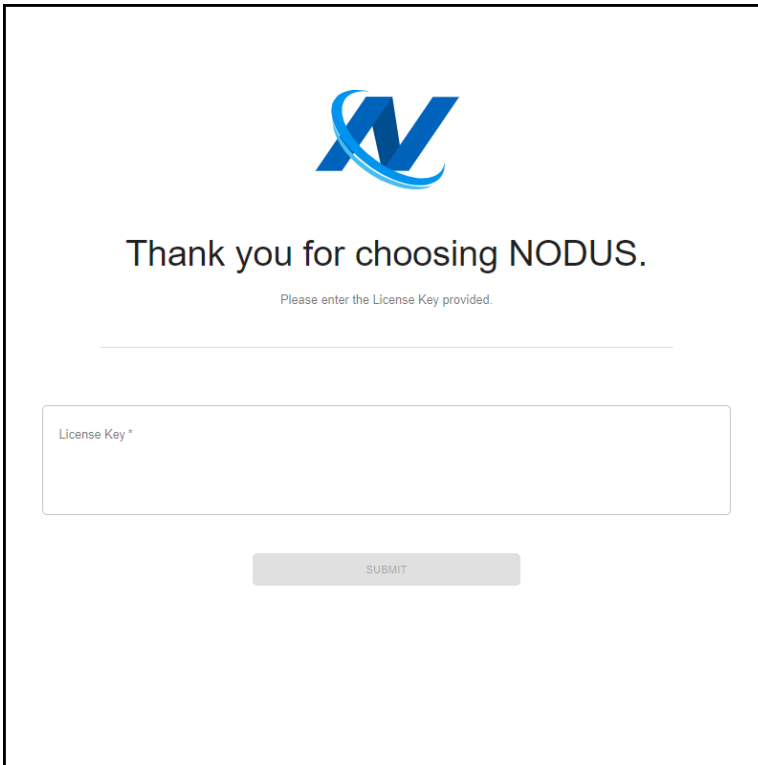
1.4 Adding the Server License Key

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.

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**.

Chapter 2: Using the NODUS CLI

In this chapter:

2.1 Stack Commands

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

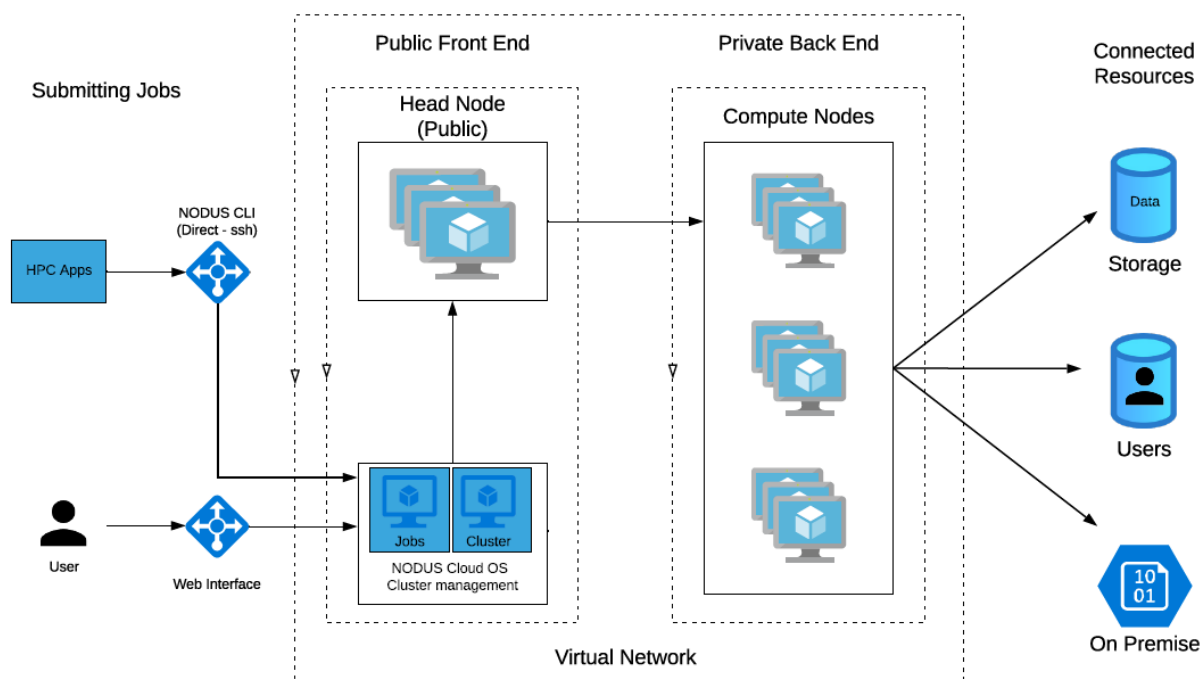| Examples |
| --- |
| $ nodus stack:remove myStack |
| $ nodus stack:remove myStack --no-clean |
| $ nodus stack:remove myStack --force |

2.2 Users

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

2.3 Architecture

NODUS Platform Overview

Chapter 3: Using the NODUS GUI

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

3.1 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>`.
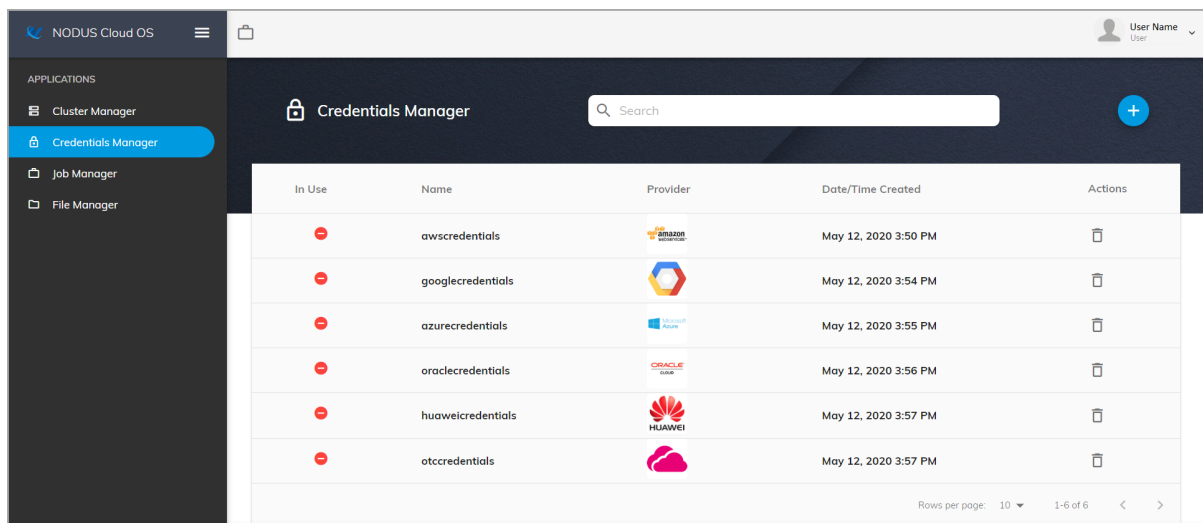
   Multi-User Account

   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!
```

3.2 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**:

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.

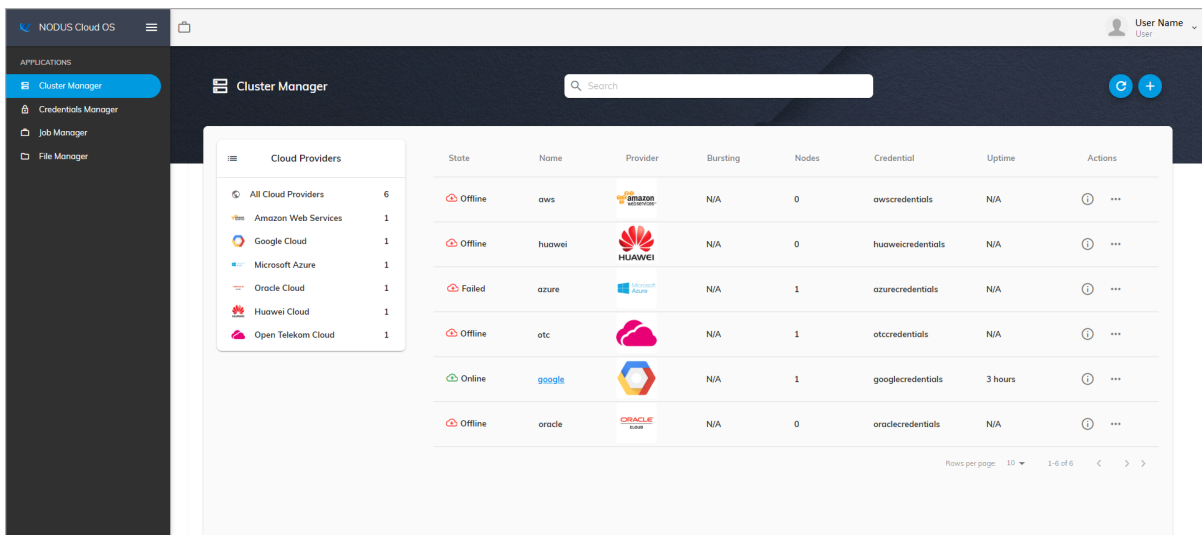Repeat the steps in this section to add credentials for another provider if desired.

3.3   Cluster Manager

This section contains the following information:

- Viewing Clusters

- Creating a Cluster

- Deploying a Cluster

- Deleting a Cluster

3.3.1 Viewing Clusters

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

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
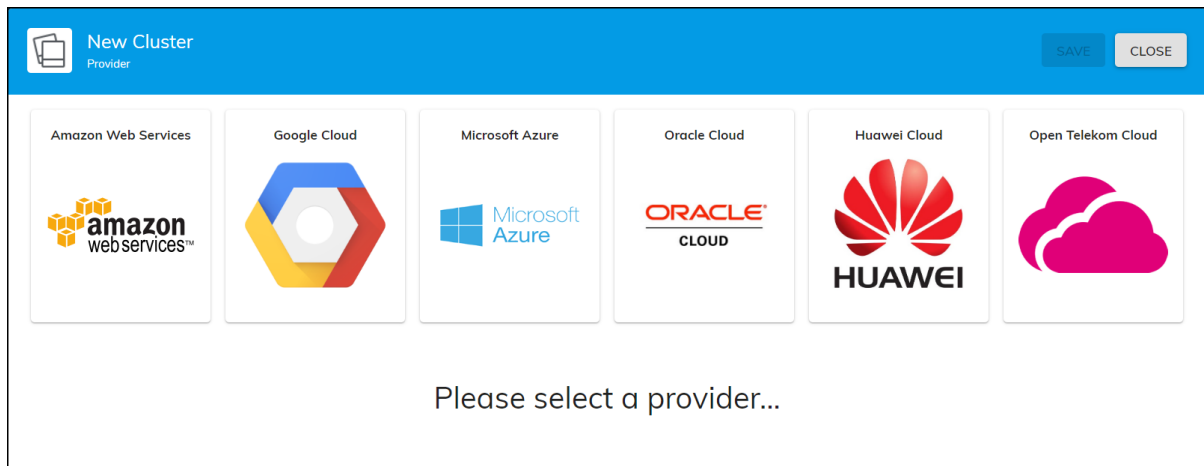
the cluster.

- 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.

3.3.2 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:

These are the supported cloud service providers:

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

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.

3.3.3 Deploying a Cluster

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

3.3.4 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**.

3.4 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:
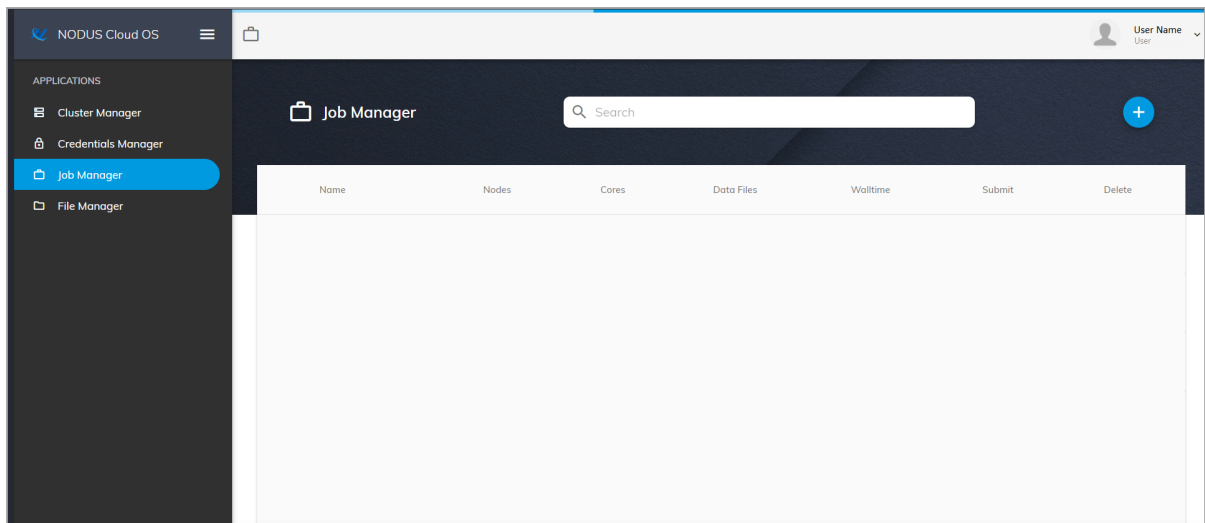
This screen shows this information: Name, Nodes, Cores, Data Files, Walltime, Submit, and Delete.

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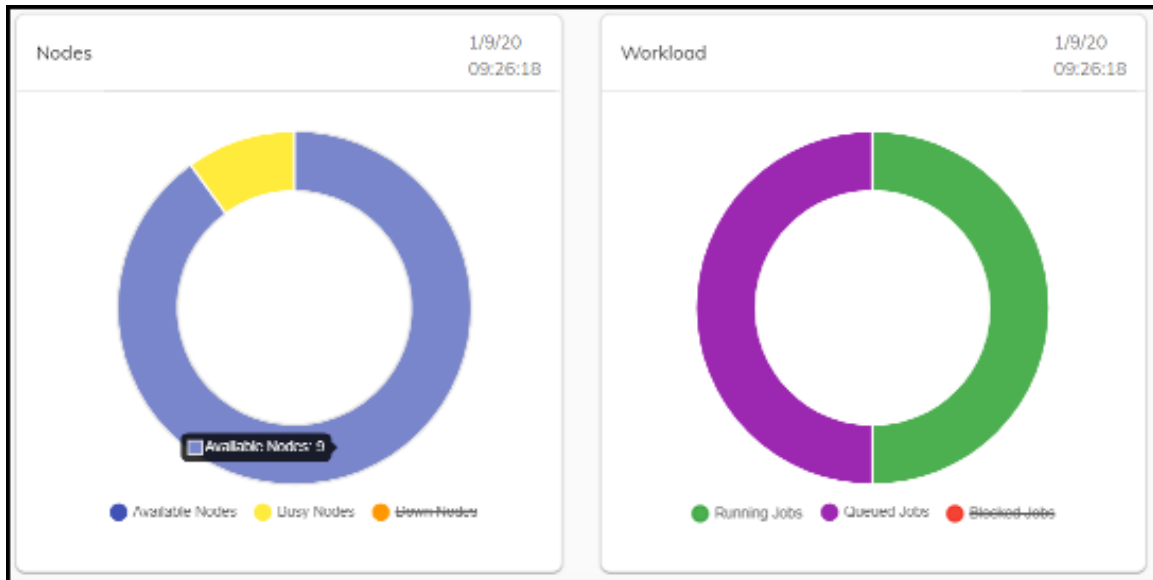
This graph shows the status of Available Nodes, Busy Nodes, & Down Nodes, and the status of the Running Jobs, Queued Jobs, & Blocked Jobs. Click to deselect nodes and jobs to narrow the display results. Hover over the graph to see the number of nodes or jobs.

- **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.

3.5 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:
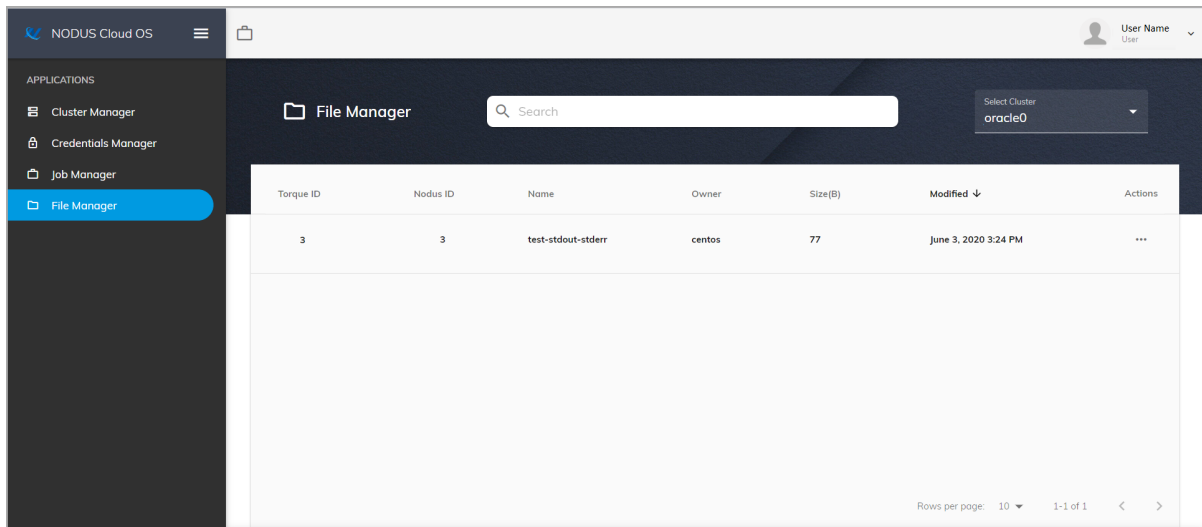
The **File Manager** screen shows this information: Torque ID, Nodus ID, Name, Owner, Size(B), Modified, and Actions.

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.

3.6 Bursting

This section contains the following information:

- Bursting Configurations
- Bursting Service

3.6.1 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.
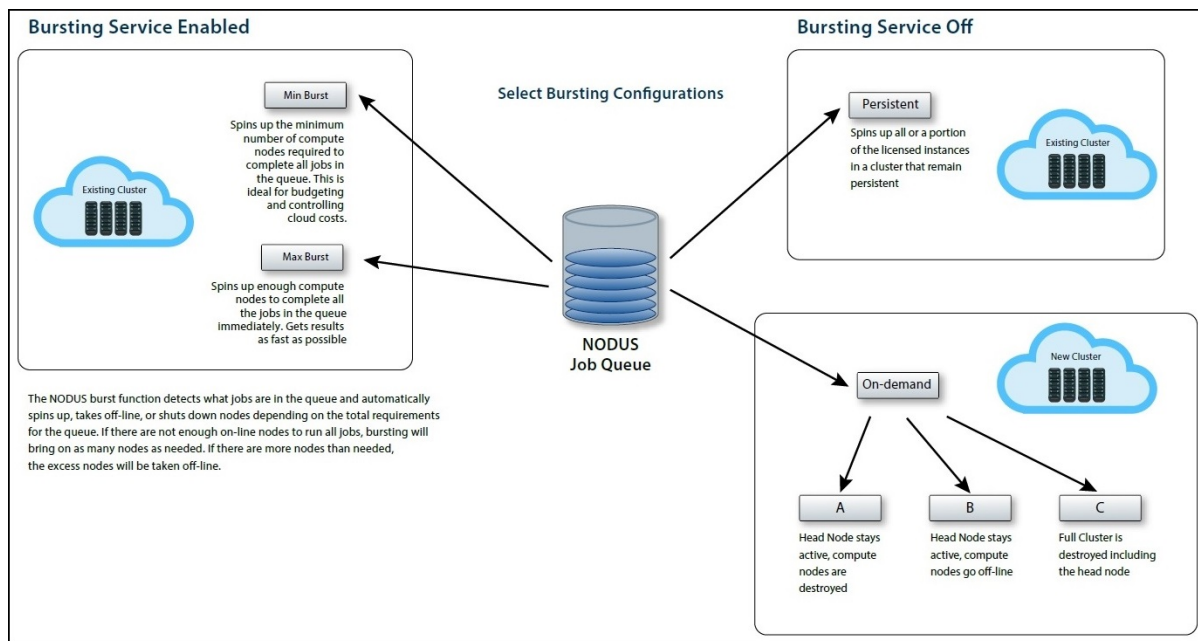
Min Burst spins up the minimum number of compute nodes required to complete all jobs in the queue, which is ideal for budgeting and controlling cloud costs.

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.

3.6.2 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:
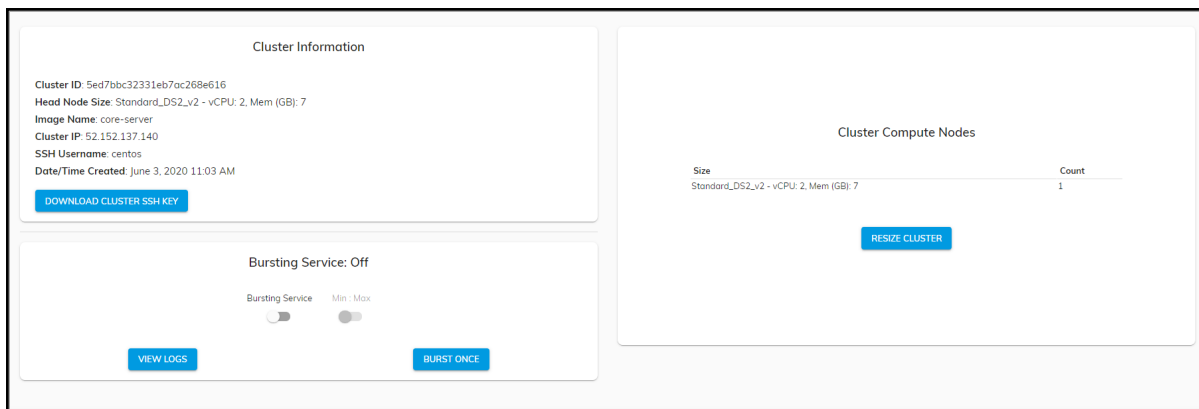
Bursting Service (Off/On) - Disable or enable bursting functionality for a single cluster:

- ⬜ (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.