GCP Slurm Cluster Tutorial

By: Michael DeProspo, Nathaniel Law, Thomas Seeley, and Madison Jones

Outline:

- Prerequisites
- Marketplace Deployment (recommended)
- Terraform Deployment
- Running Jobs on the cluster
- Troubleshooting

Prerequisites:

This tutorial offers two options for setup of a HPC with a Slurm scheduler on the Google Cloud Platform. Both setup options, marketplace and terraform, will require the following prerequisites:

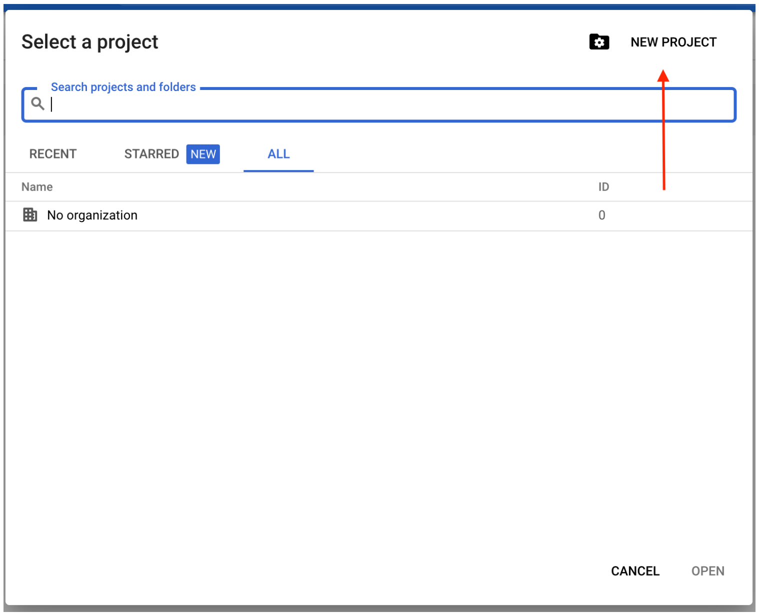
- A GCP Account:
  - Either a student account with credits or a free tier account with starter credits, this tutorial **WILL** incur a charge for running jobs on the slurm cluster, however the education credits or the $300 credits from regular account sign up should be enough unless you are going to run a lot of jobs

Marketplace Deployment:

SchedMD has a marketplace configuration on GCP to create and deploy the Slurm HPC on the cloud. This configuration allows for a streamlined setup for deploying the cluster. Because of the nature of the Slurm configuration it will require the use of compute nodes larger than the f1-micro and **will** create a charge on your google account when jobs are run.

New Project Creation

- Sign into your GCP Cloud Console
- Click select a project in the upper left corner
- This should have a modal pop up, click new project

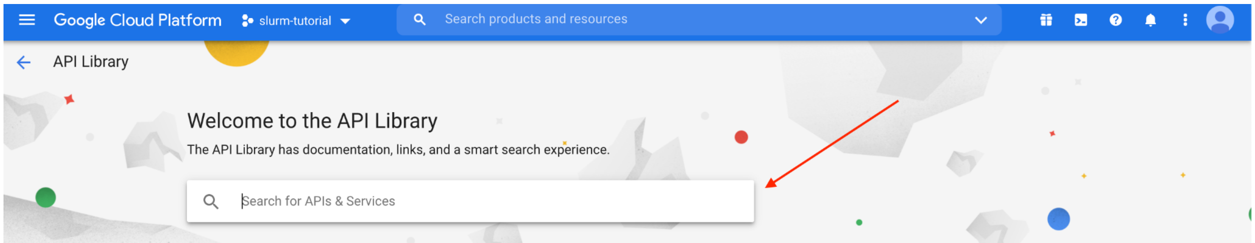


Select a project

- Make a name for your project and press create
  - You do not need to specify an organization for your project

Marketplace deployment

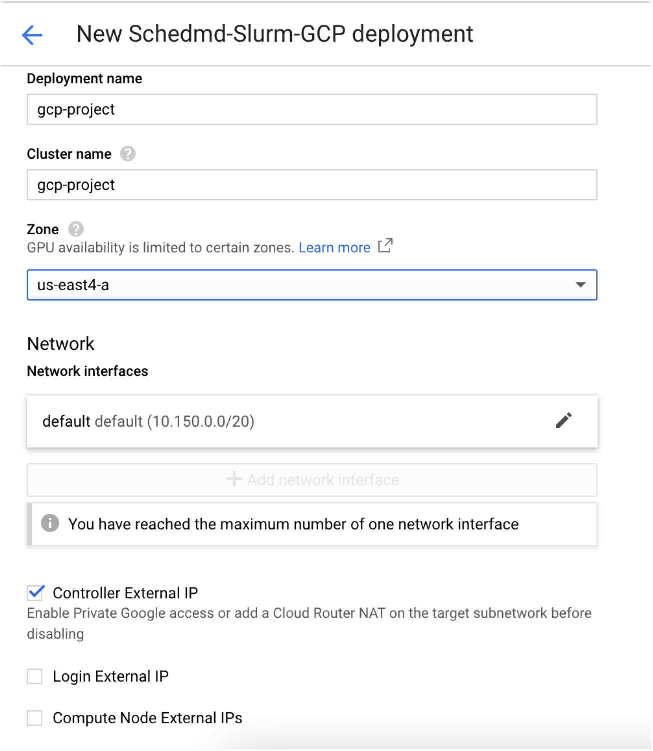
- From the newly created project use the navigation menu on the left go to the marketplace
- In the white search bar type in “SchedMD-Slurm-GCP”



Welcome to the API Library

The API Library has documentation, links, and a smart search experience.

- Click on the result for SchedMD-Slurm-GCP
- Click on the blue launch button
  - This may take up to 5-10 minutes to enable the necessary APIs
- Change the deployment name and cluster name to your desired name
- Change the region to us-east4-a
  - To run from a different zone consult the zone list and enter the desired value instead of us-east4-a



New Schedmd-Slurm-GCP deployment

Deployment name

gcp-project

Cluster name

gcp-project

Zone

GPU availability is limited to certain zones. Learn more

us-east4-a

Network

Network interfaces

default default (10.150.0.0/20)

+ Add network interface

You have reached the maximum number of one network interface

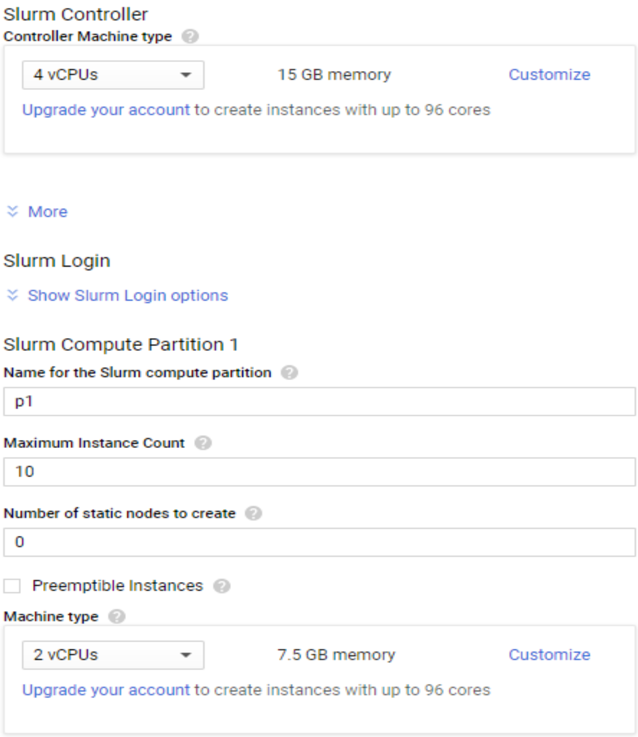
☑ Controller External IP

Enable Private Google access or add a Cloud Router NAT on the target subnetwork before disabling

□ Login External IP

□ Compute Node External IPs

- You can leave all the below values as the defaults



Slurm Controller

Controller Machine type

4 vCPUs 15 GB memory Customize

Upgrade your account to create instances with up to 96 cores

More

Slurm Login

Show Slurm Login options

Slurm Compute Partition 1

Name for the Slurm compute partition

p1

Maximum Instance Count

10

Number of static nodes to create

0

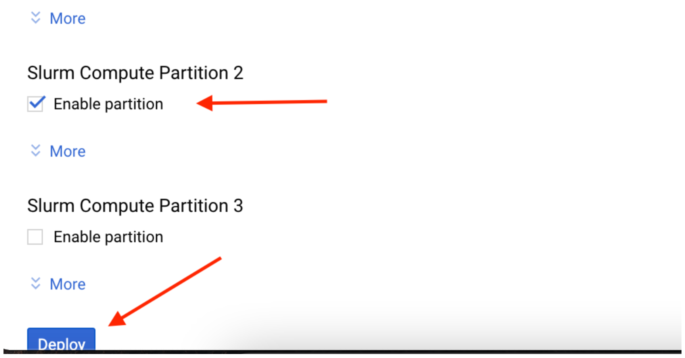
□ Preemptible Instances

Machine type

2 vCPUs 7.5 GB memory Customize

Upgrade your account to create instances with up to 96 cores

- You **do not** need to enable the 2nd and 3rd partitions but this tutorial does to show how it would look during deployment
- If applicable, agree to the terms of service for marketplace
- Press the blue deploy button



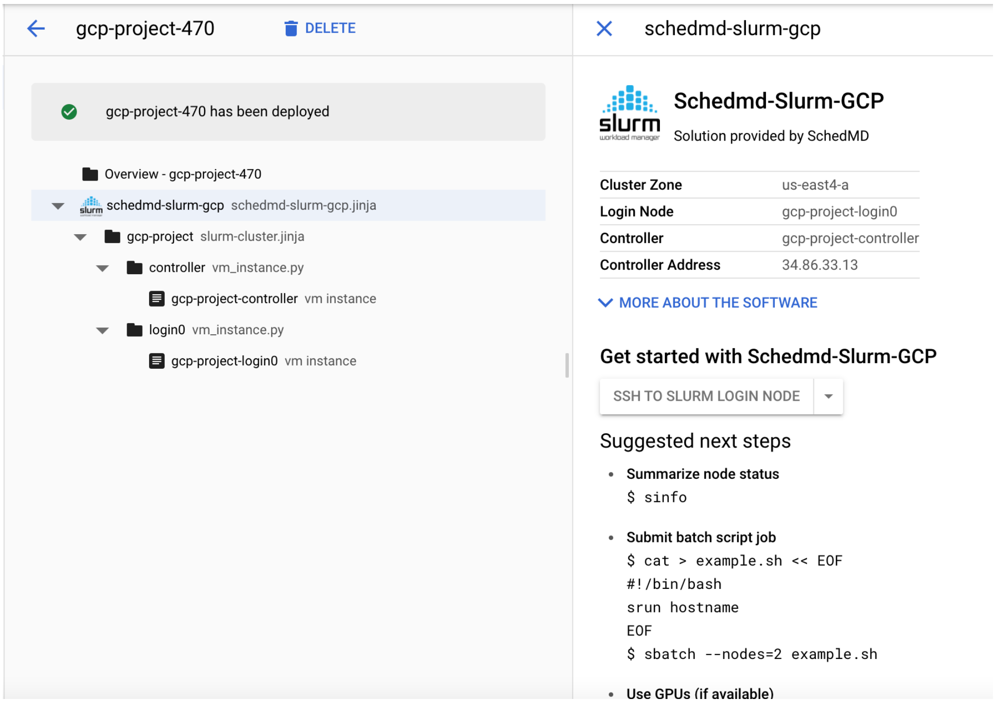
Slurm Compute Partition 2

☑ Enable partition

Slurm Compute Partition 3

□ Enable partition

- This should take you to a deployment screen - deployment of the cluster will take 5-10 minutes to complete
- Once deployment is completed your screen should look like this:



gcp-project-470 has been deployed

- Overview - gcp-project-470
- schedmd-slurm-gcp schedmd-slurm-gcp.jinja
  - gcp-project slurm-cluster.jinja
    - controller vm_instance.py
      - gcp-project-controller vm instance
    - login0 vm_instance.py
      - gcp-project-login0 vm instance

Schedmd-Slurm-GCP

Solution provided by SchedMD

Cluster Zone	us-east4-a
Login Node	gcp-project-login0
Controller	gcp-project-controller
Controller Address	34.86.33.13

Get started with Schedmd-Slurm-GCP

Suggested next steps

- Summarize node status

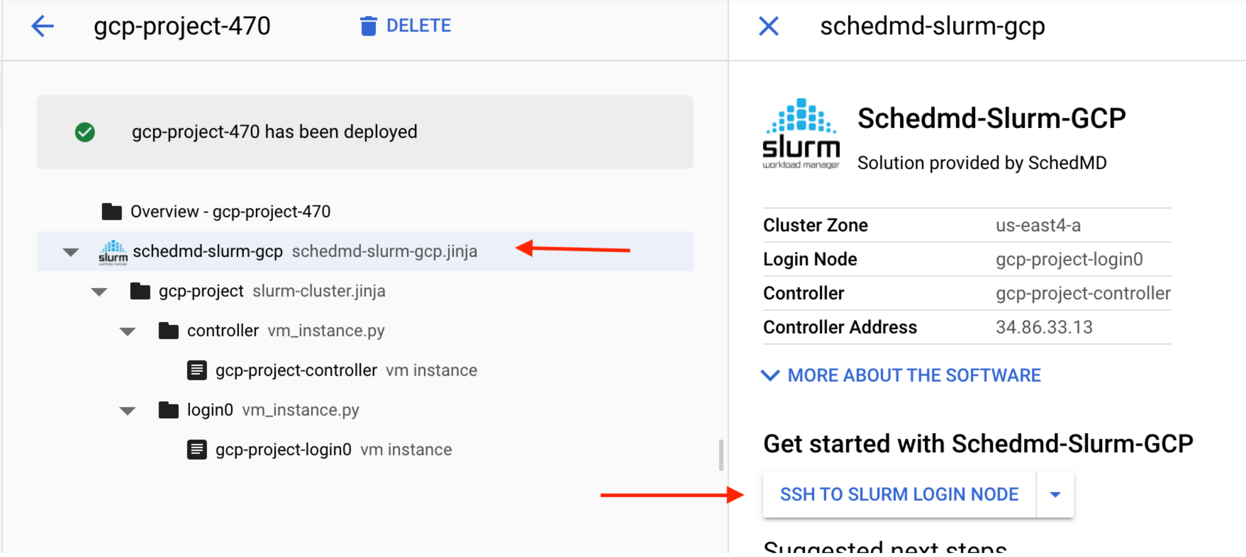
```
$ sinfo
```

- Submit batch script job

```
$ cat > example.sh << EOF
#!/bin/bash
srun hostname
EOF
$ sbatch --nodes=2 example.sh
```

- Use GPUs (if available)

- The final part of the deployment may run in the background, click on the folder that says “schedmd-slurm-gcp” with the logo next to it and wait for the “SSH TO SLURM LOGIN NODE” button to turn blue



gcp-project-470 DELETE

gcp-project-470 has been deployed

- Overview - gcp-project-470
- schedmd-slurm-gcp schedmd-slurm-gcp.jinja
  - gcp-project slurm-cluster.jinja
    - controller vm_instance.py
      - gcp-project-controller vm instance
    - login0 vm_instance.py
      - gcp-project-login0 vm instance

schedmd-slurm-gcp

Schedmd-Slurm-GCP

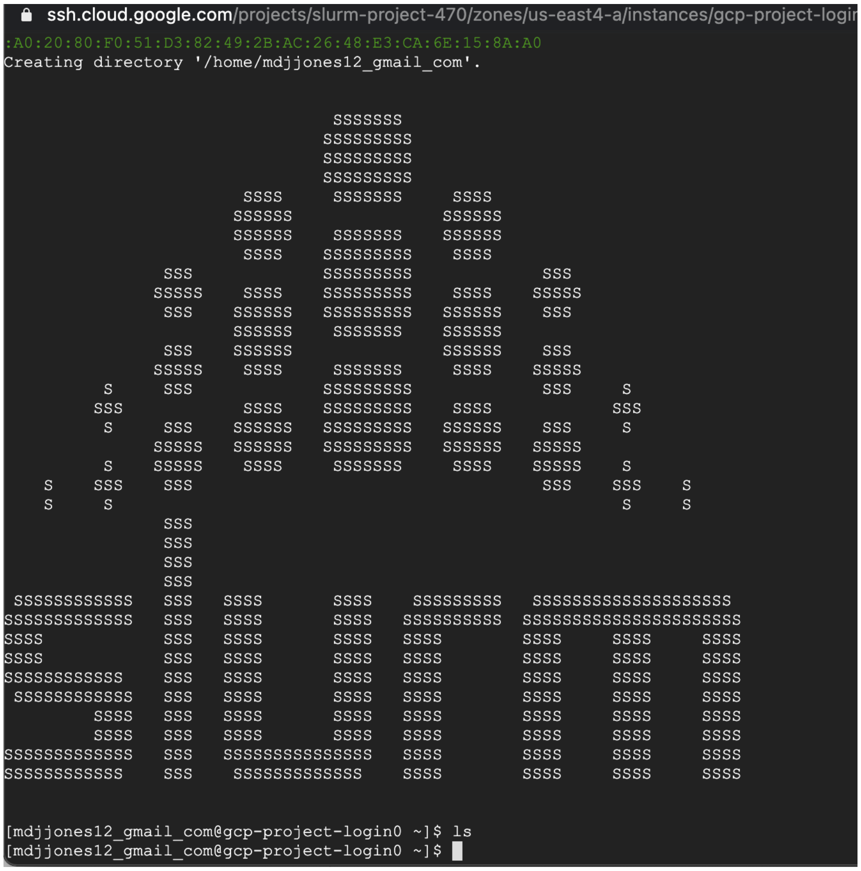
Solution provided by SchedMD

Cluster Zone	us-east4-a
Login Node	gcp-project-login0
Controller	gcp-project-controller
Controller Address	34.86.33.13

Get started with Schedmd-Slurm-GCP

Suggested next steps

- Click the SSH to Login node button once it has turned blue to be connected to the login node via the Google Cloud Shell
- The cloud shell should look like this if it has deployed correctly



ssh.cloud.google.com/projects/slurm-project-470/zones/us-east4-a/instances/gcp-project-logir

```
:A0:20:80:F0:51:D3:82:49:2B:AC:26:48:E3:CA:6E:15:8A:A0
Creating directory '/home/mdjjones12_gmail_com'.


                                 SSSSSSS
                                SSSSSSSSS
                                SSSSSSSSS
                                SSSSSSSSS
                        SSSS     SSSSSSS     SSSS
                       SSSSSS               SSSSSS
                       SSSSSS    SSSSSSS    SSSSSS
                        SSSS    SSSSSSSSS    SSSS
                SSS             SSSSSSSSS             SSS
               SSSSS    SSSS    SSSSSSSSS    SSSS    SSSSS
                SSS    SSSSSS   SSSSSSSSS   SSSSSS    SSS
                       SSSSSS    SSSSSSS    SSSSSS
                SSS    SSSSSS               SSSSSS    SSS
               SSSSS    SSSS     SSSSSSS     SSSS    SSSSS
          S     SSS             SSSSSSSSS             SSS     S
         SSS            SSSS    SSSSSSSSS    SSSS            SSS
          S     SSS    SSSSSS   SSSSSSSSS   SSSSSS    SSS     S
               SSSSS   SSSSSS   SSSSSSSSS   SSSSSS   SSSSS
          S    SSSSS    SSSS     SSSSSSS     SSSS    SSSSS    S
    S    SSS    SSS                                   SSS    SSS    S
    S     S                                                   S     S
                SSS
                SSS
                SSS
                SSS
 SSSSSSSSSSSS   SSS   SSSS       SSSS    SSSSSSSSS   SSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSS   SSS   SSSS       SSSS   SSSSSSSSSS  SSSSSSSSSSSSSSSSSSSSSS
SSSS            SSS   SSSS       SSSS   SSSS        SSSS     SSSS     SSSS
SSSS            SSS   SSSS       SSSS   SSSS        SSSS     SSSS     SSSS
SSSSSSSSSSSS    SSS   SSSS       SSSS   SSSS        SSSS     SSSS     SSSS
 SSSSSSSSSSSS   SSS   SSSS       SSSS   SSSS        SSSS     SSSS     SSSS
         SSSS   SSS   SSSS       SSSS   SSSS        SSSS     SSSS     SSSS
         SSSS   SSS   SSSS       SSSS   SSSS        SSSS     SSSS     SSSS
SSSSSSSSSSSSS   SSS   SSSSSSSSSSSSSSS   SSSS        SSSS     SSSS     SSSS
SSSSSSSSSSSS    SSS    SSSSSSSSSSSSS    SSSS        SSSS     SSSS     SSSS


[mdjjones12_gmail_com@gcp-project-login0 ~]$ ls
[mdjjones12_gmail_com@gcp-project-login0 ~]$ █
```

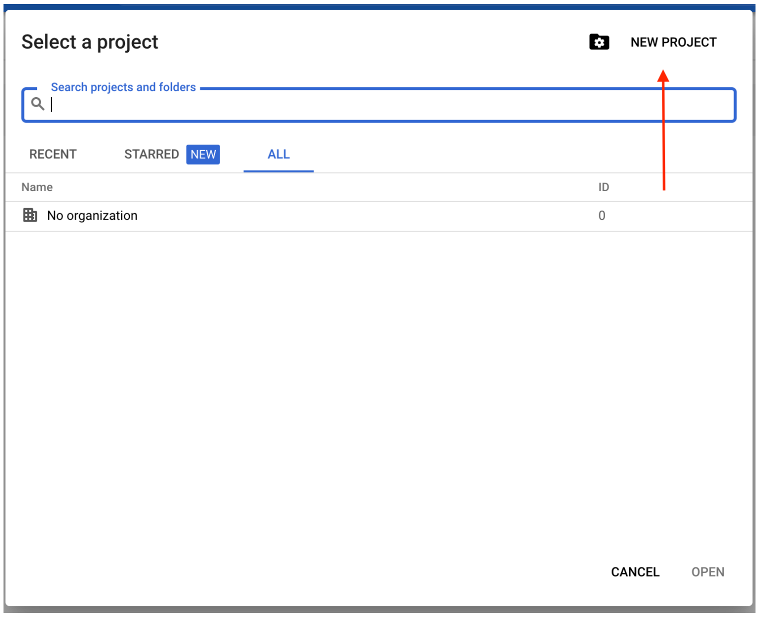
- To run jobs with this deployment go to the running jobs section

Terraform Deployment:

This section covers getting the project setup in your GCP console and allocating resources necessary to configure and run the Slurm scheduler through Terraform. The benefit of running a terraform deployment is that it allows for more customization of the Slurm configuration although the setup can be more complex and by extension error prone. This part of the tutorial is included for clarity but if doing the default deployment we recommend using the Marketplace deployment. Because of the nature of the Slurm configuration we will be using it will require that compute nodes that are larger than the f1-micro (free tier) are used which **will** create a charge to your GCP account.

New Project Creation

- Sign into your GCP Cloud Console
- If you are already running click select a project in the upper left corner
- This should have a modal pop up, click new project



Select a project

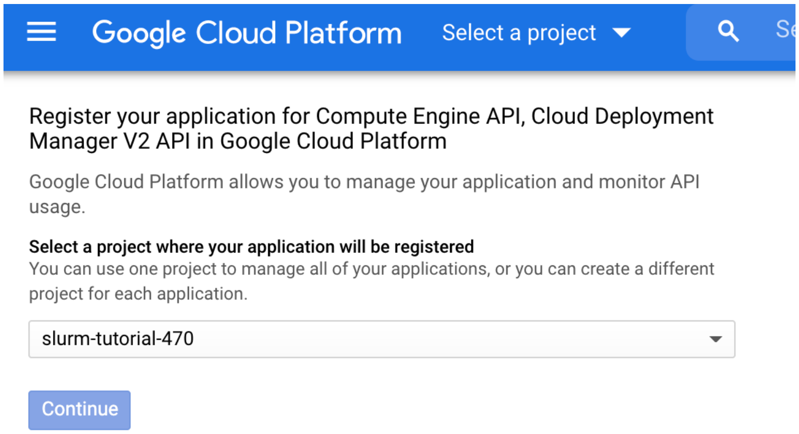
Search projects and folders

Name	ID
No organization	0

- Make a name for your project and press create
    - You do not need to specify an organization for your project

Enabling GCP Compute Engine API

- Use the following link to enable the compute engine API for your project: https://console.cloud.google.com/flows/enableapi?apiid=compute,deploymentmanager.googleapis.com&_ga=2.133478158.1032065615.1619552735-165370126.1616510696
- On the link select the project you just created and press continue



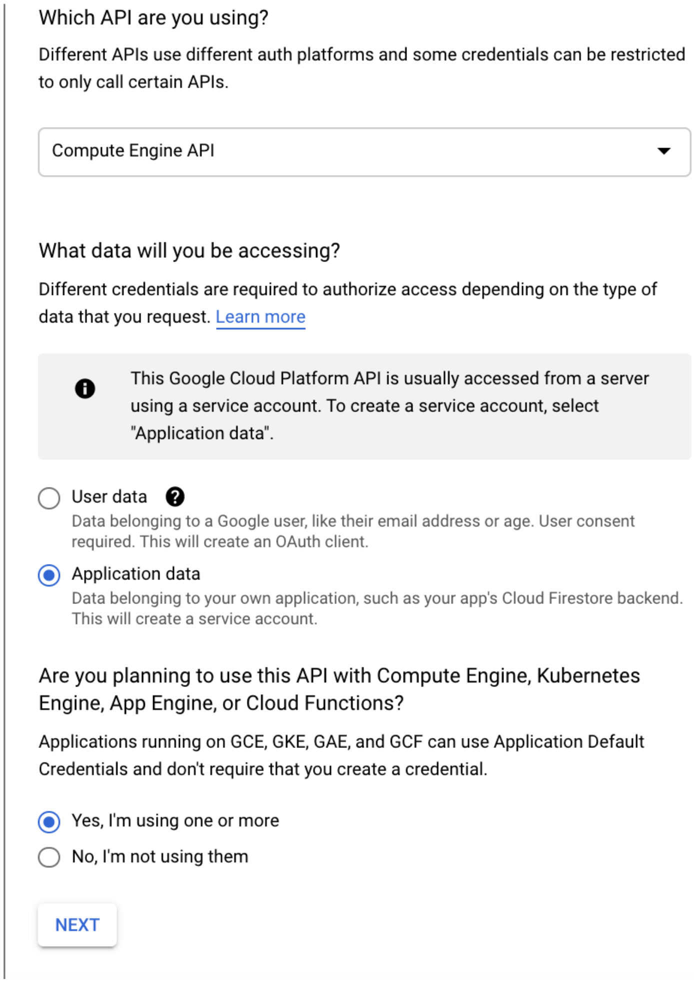
Register your application for Compute Engine API, Cloud Deployment Manager V2 API in Google Cloud Platform

Google Cloud Platform allows you to manage your application and monitor API usage.

Select a project where your application will be registered
You can use one project to manage all of your applications, or you can create a different project for each application.

slurm-tutorial-470

- This will take 2-5 minutes to enable
- Press the button that says go to credentials
- Fill out the credentials page as follows:



Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Compute Engine API

What data will you be accessing?

Different credentials are required to authorize access depending on the type of data that you request. Learn more

This Google Cloud Platform API is usually accessed from a server using a service account. To create a service account, select "Application data".

○ User data
Data belonging to a Google user, like their email address or age. User consent required. This will create an OAuth client.

● Application data
Data belonging to your own application, such as your app's Cloud Firestore backend. This will create a service account.

Are you planning to use this API with Compute Engine, Kubernetes Engine, App Engine, or Cloud Functions?

Applications running on GCE, GKE, GAE, and GCF can use Application Default Credentials and don't require that you create a credential.

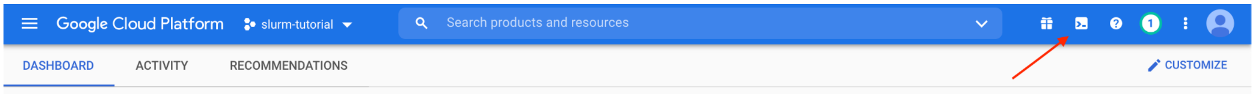
● Yes, I'm using one or more
○ No, I'm not using them

- Press the next button, this should indicate that you do not need any credentials

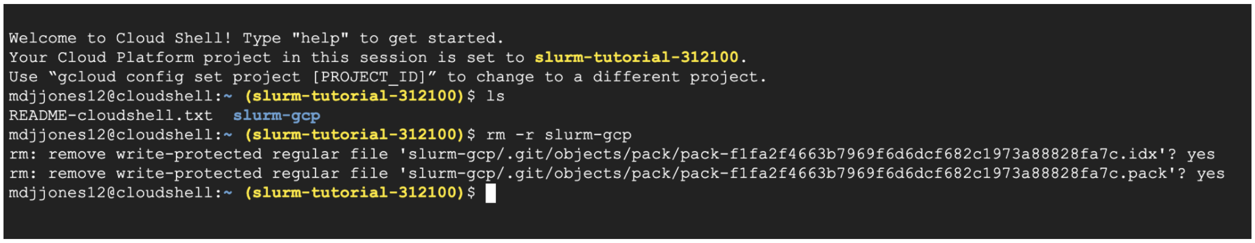
- Press the done button which should take you back to your project dashboard
- Your Compute Engine API is now registered with Google

Cluster Configuration

- Open up the cloud shell using the terminal button in the upper righthand corner of your dashboard
    - This may take a minute to load if this is your first time activating the cloud shell for this project
    - At some point the shell may ask you to confirm that you wish to proceed with a pop up, if this occurs select the affirmative button



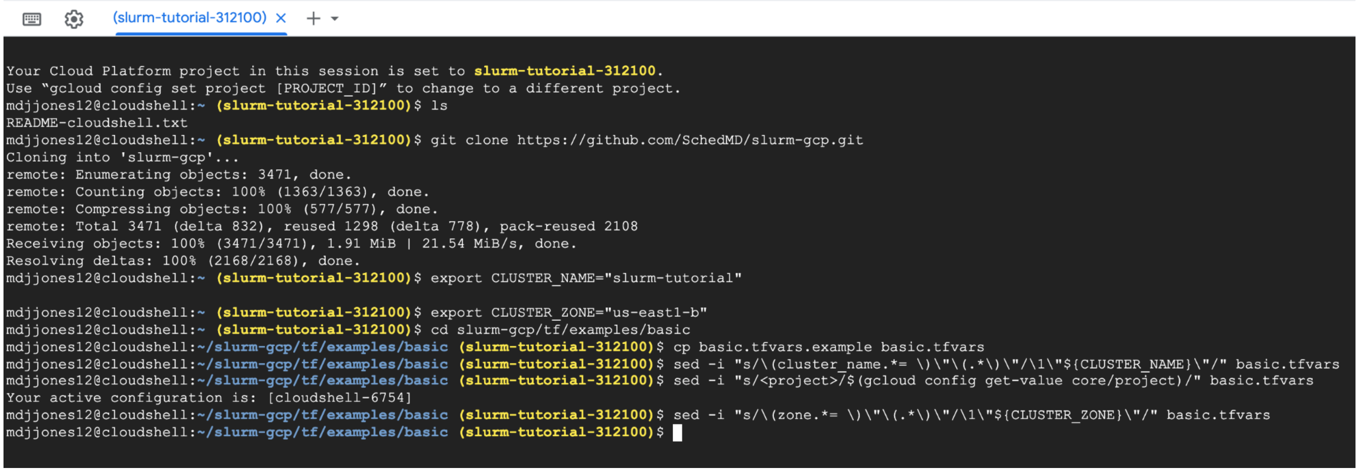
- In the shell **if you have already done this tutorial or have setup a slurm scheduler on your own with GCP** and want to start over from scratch you must remove the `slurm-gcp` directory, this can be done by running the command “`rm -r slurm-gcp`” then answering “`yes`” to the two prompted questions
    - If do not want to start over completely you can continue to the [terraform deployment]() section
    - Terminal output if choosing to remove the folder looks like this:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to slurm-tutorial-312100.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ ls
README-cloudshell.txt  slurm-gcp
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ rm -r slurm-gcp
rm: remove write-protected regular file 'slurm-gcp/.git/objects/pack/pack-f1fa2f4663b7969f6d6dcf682c1973a88828fa7c.idx'? yes
rm: remove write-protected regular file 'slurm-gcp/.git/objects/pack/pack-f1fa2f4663b7969f6d6dcf682c1973a88828fa7c.pack'? yes
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$
```

- In your cloud shell run the following commands - note each bullet point should be run as a single command:
    - `git clone https://github.com/SchedMD/slurm-gcp.git`
    - `export CLUSTER_NAME="`**`<YOUR-CLUSTER-NAME>`**`"`
    - `export CLUSTER_ZONE="`**`us-east1-b`**`"`
        - To run from a different zone consult the [zone list]() and enter the desired value instead of us-east1-b
    - `cd slurm-gcp/tf/examples/basic`
    - `cp basic.tfvars.example basic.tfvars`
    - `sed -i "s/\(cluster_name.*= \)\"\(.*\)\"/\1\"${CLUSTER_NAME}\"/" basic.tfvars`

    - `sed -i "s/<project>/$(gcloud config get-value core/project)/" basic.tfvars`
    - `sed -i "s/\(zone.*= \)\"\(.*\)\"/\1\"${CLUSTER_ZONE}\"/" basic.tfvars`
- See picture of cloud shell below for reference:



```
Your Cloud Platform project in this session is set to slurm-tutorial-312100.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ ls
README-cloudshell.txt
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ git clone https://github.com/SchedMD/slurm-gcp.git
Cloning into 'slurm-gcp'...
remote: Enumerating objects: 3471, done.
remote: Counting objects: 100% (1363/1363), done.
remote: Compressing objects: 100% (577/577), done.
remote: Total 3471 (delta 832), reused 1298 (delta 778), pack-reused 2108
Receiving objects: 100% (3471/3471), 1.91 MiB | 21.54 MiB/s, done.
Resolving deltas: 100% (2168/2168), done.
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ export CLUSTER_NAME="slurm-tutorial"

mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ export CLUSTER_ZONE="us-east1-b"
mdjjones12@cloudshell:~ (slurm-tutorial-312100)$ cd slurm-gcp/tf/examples/basic
mdjjones12@cloudshell:~/slurm-gcp/tf/examples/basic (slurm-tutorial-312100)$ cp basic.tfvars.example basic.tfvars
mdjjones12@cloudshell:~/slurm-gcp/tf/examples/basic (slurm-tutorial-312100)$ sed -i "s/\(cluster_name.*= \)\"\(.*\)\"/\1\"${CLUSTER_NAME}\"/" basic.tfvars
mdjjones12@cloudshell:~/slurm-gcp/tf/examples/basic (slurm-tutorial-312100)$ sed -i "s/<project>/$(gcloud config get-value core/project)/" basic.tfvars
Your active configuration is: [cloudshell-6754]
mdjjones12@cloudshell:~/slurm-gcp/tf/examples/basic (slurm-tutorial-312100)$ sed -i "s/\(zone.*= \)\"\(.*\)\"/\1\"${CLUSTER_ZONE}\"/" basic.tfvars
mdjjones12@cloudshell:~/slurm-gcp/tf/examples/basic (slurm-tutorial-312100)$ 
```

- You can open the basic.tfvars with an editor to see what other variables can be modified.
    - The slurm deployment does not support compute nodes of the f1-micro machine type
    - Advanced users can uncomment or modify this file for a more custom version of the terraform deployment. Commonly modified items are:
        - `machine_type`
        - `compute_disk_type`
        - `compute_disk_size_gb`

Terraform Deployment

- From the cloud shell run the following commands:
    - `terraform init`
    - `terraform apply -var-file=basic.tfvars`
        - This command will create a large amount of output and should end with two outputs of the controller IP and the login IP

    -
      ```
      gcloud compute ssh ${CLUSTER_NAME}-controller \
      --command "sudo journalctl -fu google-startup-scripts.service" \
      --zone $CLUSTER_ZONE
      ```
        - The cluster configuration script should run for up to 10 minutes (it may complete much faster than this)

- When it is completed it will not exit but the last output should say “Started Google Compute Engine Startup Scripts.” - once you see this command you can press `ctrl+c` to exit return to the terminal

- From the cloud shell you can check deployment and get into your login node by running the commands:

    -
      ```
      export CLUSTER_LOGIN_NODE=$(gcloud compute instances list \
          --zones ${CLUSTER_ZONE} \
          --filter="name ~ .*login." \
          --format="value(name)" | head -n1)
      ```
        - This creates a constant variable for the login sequence

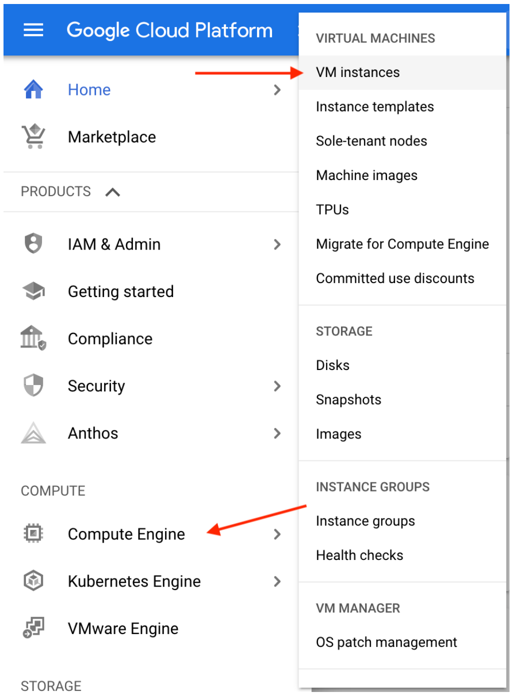
    -
      ```
      gcloud compute ssh ${CLUSTER_LOGIN_NODE} \
          --zone $CLUSTER_ZONE
      ```
- You should now see the slurm startup logo and be in the login node and ready to run jobs
- To exit the login node and return to the cloud shell for the project you can press `ctrl+d`

Running Jobs on the Cluster:

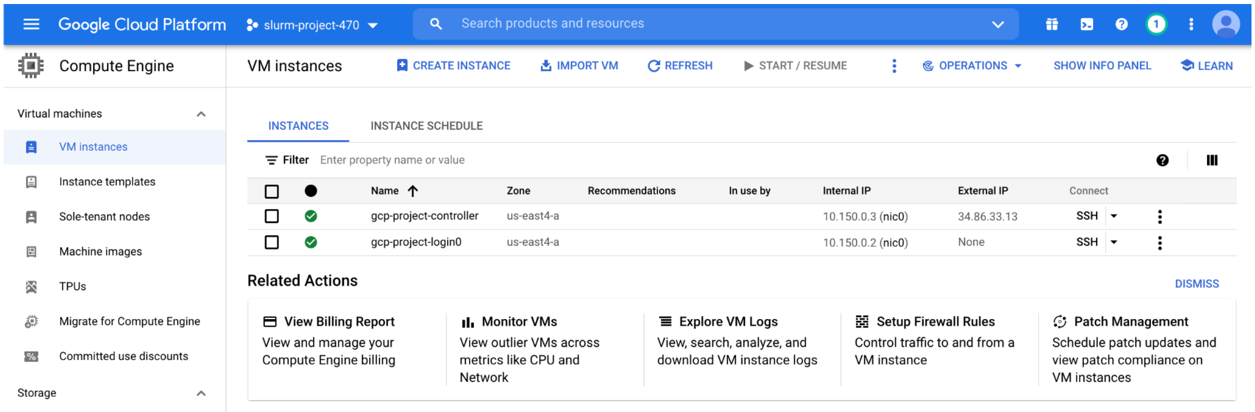
Running jobs on the cluster can be done via the login node. Do not run jobs directly from the login node, the most efficient way to run these jobs is to submit them with `sbatch`. Running jobs on the cluster will allow it to autoscale. Jobs that are submitted will create the required number of nodes, then after the job is completed a short delay will occur then the cluster will scale back down to the controller and login nodes. Because of the autoscaling feature of these Slurm setups it will create some overhead in the amount of time it takes to run jobs which is why using `sbatch` is recommended.

Monitoring VM Instances

- To see the VM instances deploy and be destroyed open another tab (you still need the cloud shell open) with the GCP Cloud Console
- Ensure that you are on the correct project in the upper left corner
- Using the navigation menu on the left go to Compute Engine and then VM Instances



- On this page you will be able to see new instances deploy in real time. If no jobs are running the screen should look like this with just the controller and login nodes



□	●	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	gcp-project-controller	us-east4-a			10.150.0.3 (nic0)	34.86.33.13	SSH
□	✓	gcp-project-login0	us-east4-a			10.150.0.2 (nic0)	None	SSH

Related Actions

View Billing Report
View and manage your Compute Engine billing

Monitor VMs
View outlier VMs across metrics like CPU and Network

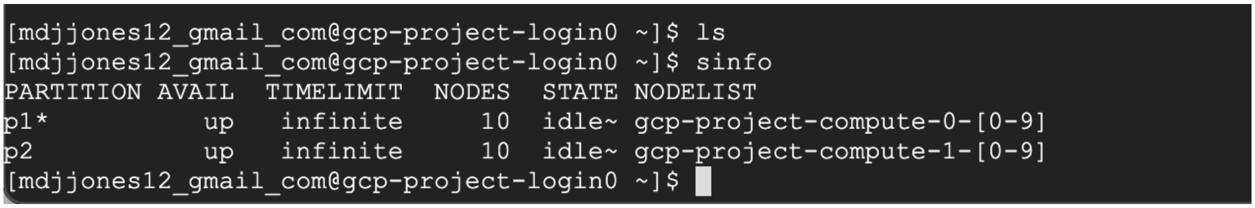
Explore VM Logs
View, search, analyze, and download VM instance logs

Setup Firewall Rules
Control traffic to and from a VM instance

Patch Management
Schedule patch updates and view patch compliance on VM instances

Submitting Batch Jobs on the Cluster

- From the cloud shell you can run the `sinfo` command to ensure that your login and controller nodes are active (this can also be checked in the VM Instances dashboard
    - There should be a row for each partition that you enabled

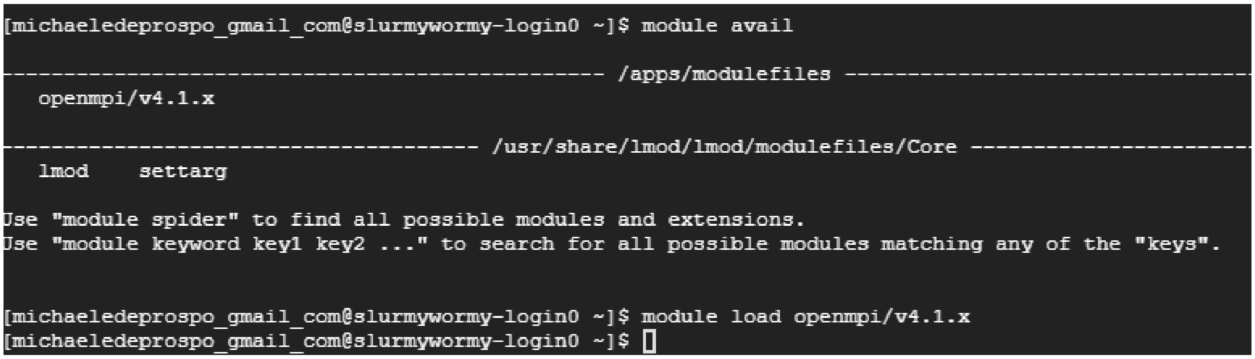


```
[mdjjones12_gmail_com@gcp-project-login0 ~]$ ls
[mdjjones12_gmail_com@gcp-project-login0 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
p1*          up   infinite     10  idle~ gcp-project-compute-0-[0-9]
p2           up   infinite     10  idle~ gcp-project-compute-1-[0-9]
[mdjjones12_gmail_com@gcp-project-login0 ~]$ 
```

- From the login node you can run `sbatch` scripts directly
- The deployment can be monitored by running the squeue command and by watching the VM Instances dashboard
- Once completed with running jobs and deploying nodes, shut down the project to conserve resources

Loading MPI

- To load MPI type module avail to see what version is available
- Load the module using “module load <version of mpi>”



```
[michaeledeprospo_gmail_com@slurmywormy-login0 ~]$ module avail

------------------------------------------------ /apps/modulefiles --------------------------------
   openmpi/v4.1.x

-------------------------------------- /usr/share/lmod/lmod/modulefiles/Core ----------------------
   lmod    settarg

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".


[michaeledeprospo_gmail_com@slurmywormy-login0 ~]$ module load openmpi/v4.1.x
[michaeledeprospo_gmail_com@slurmywormy-login0 ~]$ 
```

Troubleshooting and notes:

- To delete the project go to home dashboard of your project by clicking on the Google Cloud Platform home button in the upper lefthand corner
  - On the project info card click on go to project settings button and then press the shutdown button on the top row
  - Type in the project name and press the shutdown button
- With the terraform deployment if you have a previous project using the same deployment it may require:
  - Removing the slurm-gcp folder and creating a new one
  - Waiting for any old projects to have all resources deleted completely which can take up to 24 hours
- More information or additional instruction for the Marketplace deployment can be found here: https://github.com/SchedMD/slurm-gcp/blob/master/MP_README.md
- More information or additional instructions for the Terraform deployment can be found here: https://cloud.google.com/architecture/deploying-slurm-cluster-compute-engine