Unable to create windows nodepool on GKE cluster #632

Closed **abdelhegazi** opened this issue on Aug 17, 2020 · 9 comments

abdelhegazi commented on Aug 17, 2020 • edited

In GKE "This is my first time to provision a cluster with windows node_pool"
I am calling module

```
  source  = "terraform-google-modules/kubernetes-engine/google//modules/beta-private-
cluster-update-variant"
  version = "9.2.0"
```

I had to define two pools one for linux pool required by GKE and the windows one we require, terraform always succeeds in provisioning the linux node_pool but fails to provision the windows one and the error message

```
module.gke.google_container_cluster.primary: Still modifying... [id=projects/uk-xxx-
xx-xxx-b821/locations/europe-west2/clusters/gke-nonpci-dev, 24m31s elapsed]
module.gke.google_container_cluster.primary: Still modifying... [id=projects/uk-xxx-
xx-xxx-b821/locations/europe-west2/clusters/gke-nonpci-dev, 24m41s elapsed]
module.gke.google_container_cluster.primary: Still modifying... [id=projects/uk-xxx-
xx-xxx-b821/locations/europe-west2/clusters/gke-nonpci-dev, 24m51s elapsed]
module.gke.google_container_cluster.primary: Modifications complete after 24m58s
[id=projects/xx-xxx-xx-xxx-b821/locations/europe-west2/clusters/gke-nonpci-dev]
module.gke.google_container_node_pool.pools["windows-node-pool"]: Creating...

Error: error creating NodePool: googleapi: Error 400: Workload Identity is not
supported on Windows nodes. Create the nodepool without workload identity by
specifying --workload-metadata=GCE_METADATA., badRequest

  on .terraform\modules\gke\terraform-google-kubernetes-engine-9.2.0\modules\beta-
private-cluster-update-variant\cluster.tf line 341, in resource
"google_container_node_pool" "pools":
 341: resource "google_container_node_pool" "pools" {
```

I tried many places to set this metadata values but I coldn't get it right:

From terraform side :

I tried also setting `enable_shielded_nodes = false` but this didn't really help much.

I tried to test this if it is doable even through the command line this was my command line

```
C:\>gcloud container node-pools --region europe-west2 list
NAME                    MACHINE_TYPE   DISK_SIZE_GB  NODE_VERSION
default-node-pool-d916  n1-standard-2  100           1.17.9-gke.600


C:\>gcloud container node-pools --region europe-west2 create window-node-pool --
cluster=gke-nonpci-dev --image-type=WINDOWS_SAC --no-enable-autoupgrade --machine-
type=n1-standard-2
WARNING: Starting in 1.12, new node pools will be created with their legacy Compute
Engine instance metadata APIs disabled by default. To create a node pool with legacy
instance metadata endpoints disabled, run `node-pools create` with the flag `--
metadata disable-legacy-endpoints=true`.
This will disable the autorepair feature for nodes. Please see
https://cloud.google.com/kubernetes-engine/docs/node-auto-repair for more information
on node autorepairs.
ERROR: (gcloud.container.node-pools.create) ResponseError: code=400, message=Workload
Identity is not supported on Windows nodes. Create the nodepool without workload
identity by specifying --workload-metadata=GCE_METADATA.

C:\>gcloud container node-pools --region europe-west2 create window-node-pool --
cluster=gke-nonpci-dev --image-type=WINDOWS_SAC --no-enable-autoupgrade --machine-
type=n1-standard-2 --workload-metadata=GCE_METADATA --metadata disable-legacy-
endpoints=true
This will disable the autorepair feature for nodes. Please see
https://cloud.google.com/kubernetes-engine/docs/node-auto-repair for more information
on node autorepairs.
ERROR: (gcloud.container.node-pools.create) ResponseError: code=400, message=Service
account "874988475980-compute@developer.gserviceaccount.com" does not exist.

C:\>gcloud auth list
                      Credentialed Accounts
ACTIVE  ACCOUNT
*       tf-xxx-xxx-xx-xxx@xx-xxx-xx-xxx-xxxx.iam.gserviceaccount.com
```

This service account from running `gcloud auth list` is the one I am running terraform with but I don't know where is this one in the error message coming from, even though trying to create the windows nodepool through command line as shown above also didn't work I am a bit stuck and I don't know what to do.

As module 9.2.0 is a stable module for us through all our linux based clusters we setup before, hence I thought this may be an old version for a windows node_pool I used the `11.0.0` instead to see if this would make it any different but ended up with a different error

```
module.gke.google_container_node_pool.pools["default-node-pool"]: Refreshing state...
[id=projects/uk-tix-p1-npe-b821/locations/europe-west2/clusters/gke-nonpci-
dev/nodePools/default-node-pool-d916]
```

```
.terraform/modules/gke.gcloud_delete_default_kube_dns_configmap/terraform-google
gcloud-1.4.1/scripts/check_env.sh: %1 is not a valid Win32 application.

  on .terraform\modules\gke.gcloud_delete_default_kube_dns_configmap\terraform-google-
gcloud-1.4.1\main.tf line 70, in data "external" "env_override":
  70: data "external" "env_override" {

Error: failed to execute ".terraform/modules/gke.gcloud_wait_for_cluster/terraform-
google-gcloud-1.3.0/scripts/check_env.sh": fork/exec
.terraform/modules/gke.gcloud_wait_for_cluster/terraform-google-gcloud-
1.3.0/scripts/check_env.sh: %1 is not a valid Win32 application.

  on .terraform\modules\gke.gcloud_wait_for_cluster\terraform-google-gcloud-
1.3.0\main.tf line 70, in data "external" "env_override":
  70: data "external" "env_override" {
```

This how I set node_pools parameters

```
  node_pools = [
    {
      name               = "linux-node-pool"
      machine_type       = var.nodepool_instance_type
      min_count          = 1
      max_count          = 10
      disk_size_gb       = 100
      disk_type          = "pd-standard"
      image_type         = "COS"
      auto_repair        = true
      auto_upgrade       = true
      service_account    = google_service_account.gke_cluster_sa.email
      preemptible        = var.preemptible
      initial_node_count = 1
    },
    {
      name               = "windows-node-pool"
      machine_type       = var.nodepool_instance_type
      min_count          = 1
      max_count          = 10
      disk_size_gb       = 100
      disk_type          = "pd-standard"
      image_type         = var.nodepool_image_type
      auto_repair        = true
      auto_upgrade       = true
      service_account    = google_service_account.gke_cluster_sa.email
      preemptible        = var.preemptible
      initial_node_count = 1

    }
  ]

  cluster_resource_labels = var.cluster_resource_labels

  # health check and webhook firewall rules
  node_pools_tags = {
```

```
  }

  node_pools_metadata = {
    all = {
//      workload-metadata = "GCE_METADATA"
    }

    linux-node-pool = {
      ssh-keys = join("\n", [for user, key in var.node_ssh_keys : "${user}:${key}"])
      block-project-ssh-keys = true
    }

    windows-node-pool = {
      workload-metadata = "GCE_METADATA"
    }

  }
```

- this is a shared VPC where I provision my cluster with cluster version: 1.17.9-gke.600

Also scrolling through some SOF posts someone suggested setting `enable_shielded_nodes = false` but it didn't really make much of a help with same error message I got earlier regarding the setting the metadata to `GCE_METADATA`

bharathkkb commented on Aug 18, 2020

Hi **@abdelhegazi**
Have you tried setting `identity_namespace = null` and `node_metadata="SECURE"` in the beta-private-cluster-update-variant module? This should disable WI.

abdelhegazi commented on Aug 18, 2020 • edited

Hi **@bharathkkb** Thanks a lot but I tired this before and today again it didn't really work out.

This is how i set these two parameters where I am calling the module also I tried even to be more specific by setting the default values of these variable from within the module itself after it has been called by `terraform init` but still no difference same error message comes out.

```
...
  istio                   = var.istio
  remove_default_node_pool = true
  enable_shielded_nodes = false

  node_metadata = "SECURE"
  identity_namespace = null

  node_pools = [
    {
```

```
      max_count          = 10
      disk_size_gb       = 100
      disk_type          = "pd-standard"
      image_type         = "COS"                                  # GCP-GKE-006
      auto_repair        = true                                   # GCP-GKE-035
      auto_upgrade       = true                                   # GCP-GKE-008
      service_account    = google_service_account.gke_cluster_sa.email
      preemptible        = var.preemptible
      initial_node_count = 1
    },
    {
      name               = "windows-node-pool"
      machine_type       = var.nodepool_instance_type
      min_count          = 1
      max_count          = 10
      disk_size_gb       = 100
      disk_type          = "pd-standard"
      image_type         = var.nodepool_image_type                # GCP-GKE-006
      auto_repair        = true
      auto_upgrade       = true                                   # GCP-GKE-008
      service_account    = google_service_account.gke_cluster_sa.email
      preemptible        = var.preemptible
      initial_node_count = 1

    }
  ]

  cluster_resource_labels = var.cluster_resource_labels

  node_pools_metadata = {
    all = {
      workload-metadata = "GCE_METADATA"
    }

    linux-node-pool = {
      ssh-keys = join("\n", [for user, key in var.node_ssh_keys : "${user}:${key}"])
      block-project-ssh-keys = true
      workload-metadata = "GCE_METADATA"
    }

    windows-node-pool = {
      workload-metadata = "GCE_METADATA"
    }
  }
...
```

And this is the error message, same as before as usual it only provisioned the linux one but not the windows node_pool

```
module.gke.google_container_node_pool.pools["linux-node-pool"]: Creation complete
after 1m32s [id=projects/xxx-xxx-xxx-xxx/locations/europe-west2/clusters/gke-nonpci-
dev/nodePools/linux-node-pool-dd44]

Error: error creating NodePool: googleapi: Error 400: Workload Identity is not
supported on Windows nodes. Create the nodepool without workload identity by
```

```
"google_container_node_pool" "pools":
 336: resource "google_container_node_pool" "pools" {
```

Looks to me this didn't actually disable WIs on neither on the cluster nor the nodes

abdelhegazi commented on Aug 19, 2020 • edited

just an update, this command seems to have worked finally but from the command line, how on earth this is different to what I have done before to the terraform parameters, its all the same, and here surethe workload-metadata is very specific to that node-pool but I really struggle to get this through to terraform

```
gcloud container node-pools --region europe-west2 create windows-nodepool --image-
type=WINDOWS_SAC --no-enable-autoupgrade --machine-type=n1-standard-2  --workload-
metadata=GCE_METADATA --service-account=xx-xx-xx-p1-xxx@xx-xx-xx-xx-
xx.iam.gserviceaccount.com --cluster gke-nonpci-dev --metadata disable-legacy-
endpoints=true
```

Any idea on translating these to terraform, thanks ?

bharathkkb commented on Aug 19, 2020

@abdelhegazi I was able to make it would with these settings. Notable changes include `enable_integrity_monitoring = false` on Windows pool, `node_metadata = "EXPOSE"` , `network_policy = false` , `identity_namespace = null` etc. Full config below. I think the error being surfaced is not entirely correct, but it probably requires a fix at the GKE API layer. Let me know if this works for you.

```
region                    = var.region
  network                 = var.network
  subnetwork              = var.subnetwork
  ip_range_pods           = var.ip_range_pods
  ip_range_services       = var.ip_range_services
  service_account         = var.compute_engine_service_account
  enable_private_endpoint = true
  enable_private_nodes    = true
  network_policy = false
  master_ipv4_cidr_block  = "172.16.0.0/28"
  enable_shielded_nodes = false
  node_metadata = "EXPOSE"
  identity_namespace = null


  master_authorized_networks = [
    {
      cidr_block   = data.google_compute_subnetwork.subnetwork.ip_cidr_range
```

```
    node_pools = [
      {
        name               = "linux-node-pool"
        min_count          = 1
        max_count          = 10
        disk_size_gb       = 100
        disk_type          = "pd-standard"
        image_type         = "COS"
        initial_node_count = 1
      },
      {
        name               = "windows-node-pool"
        min_count          = 1
        max_count          = 10
        disk_size_gb       = 100
        disk_type          = "pd-standard"
        image_type         = "WINDOWS_SAC"
        initial_node_count = 1
        enable_integrity_monitoring = false
      }
    ]
```

abdelhegazi commented on Aug 19, 2020 • edited

Thanks a lot **@bharathkkb**
This definitely got the cluster happy with both node_pools provisioned through terraform, I don't how to report this to you guys, is there any process to make this in process or at least to make sure the APIs are addressing this issue. I honestly didn't expect I need to disable WI on the whole cluster I thought disabling it only on the windows one should be enough and also having cluster up without network policy is what do you think about that :)
Really appreciate your responses to my issue.

```
Name | Status | Version | Number of nodes | Machine type | Image type | Autoscaling |

linux-node-pool-88ba | OK | 1.17.9-gke.600 | 3 (1 per zone) | n1-standard-2 |
Container-Optimized OS (cos) | 1 - 10 nodes per zone |
windows-node-pool-060b | Provisioning | 1.17.9-gke.600 | 3 (1 per zone) | n1-standard-
2 | Windows Semi-Annual Channel | 1 - 10 nodes per zone |
```

@abdelhegazi Glad it worked 😅 As this more of feature request/enhancement territory, I believe GCP support would be the best way.

abdelhegazi commented on Aug 19, 2020

Thanks a lot **@bharathkkb** really appreciate your help in sorting this out.
Have a great day and I will close this issue.

abdelhegazi closed this as completed on Aug 19, 2020

abdelhegazi mentioned this issue on Aug 25, 2020

Unable to create GKE Nodepool with windows LTSC nodepool as image_type #587

Closed

bharathkkb mentioned this issue on Oct 4, 2020

initial code commit #695

Closed

kansberry commented on Feb 5, 2021 • edited

I just wanted to comment on this. I have setup a GKE cluster using the "safer-cluster" module with a mix of Linux and Windows Node pools. This module uses the beta-private-cluster module. After reading through this chain of messages and reviewing the scripts, I finally came up with a combination that appears to work without removing workload identity from the entire cluster. I did this by making changes to the node_pools array. I did not have to set network_policy to false or identity_namespace to "null".

@bharathkkb, can you verify that is what I did with settings below?

```
zones                      = var.zones
network                    = var.network_name
network_project_id         = var.network_project_id
add_cluster_firewall_rules = true
subnetwork                 = var.subnet_name
ip_range_pods              = var.ip_range_pods_name
ip_range_services          = var.ip_range_services_name
master_ipv4_cidr_block     = var.master_ipv4_cidr_block
default_max_pods_per_node  = var.default_max_pods_per_node
enable_private_endpoint    = true
master_authorized_networks = [{
  cidr_block   = "${module.bastion.ip_address}/32"
  display_name = "Bastion Host"
}]
```

```
  node_pools_tags = {
    all = ["allow-google-apis", "allow-lb"]
  }

  node_pools = [
    {
      name         = "lin-8c-32g-np1"
      autoscaling  = true
      min_count    = 2
      max_count    = 4
      auto_upgrade = false
      auto_repair  = true
      machine_type = "n2-standard-8"
      node_metadata = "GKE_METADATA_SERVER"
    },
    {
      name         = "lin-4c-16g-np1"
      min_count    = 1
      max_count    = 8
      auto_upgrade = false
      auto_repair  = true
      machine_type = "n2-standard-4"
      node_metadata = "GKE_METADATA_SERVER"
    },
    {
      name         = "lin-2c-8g-np1"
      min_count    = 1
      max_count    = 8
      auto_upgrade = false
      auto_repair  = true
      machine_type = "n2-standard-2"
      node_metadata = "GKE_METADATA_SERVER"
    },
    {
      name         = "win-4c-16g-np1"
      min_count    = 1
      max_count    = 4
      auto_upgrade = false
      auto_repair  = true
      machine_type = "n2-standard-4"
      image_type   = "WINDOWS_LTSC"
      node_metadata = "EXPOSE"
      enable_integrity_monitoring = false
    },
    {
      name         = "win-2c-8g-np1"
      min_count    = 1
      max_count    = 4
      auto_upgrade = false
      auto_repair  = true
      machine_type = "n2-standard-2"
      image_type   = "WINDOWS_LTSC"
      node_metadata = "EXPOSE"
      enable_integrity_monitoring = false
    }
  ]
```

navnitDevOps commented on Nov 30, 2021

Hi
i am still facing issue when i am trying to create nodepools with gke-version 1.21.5-gke.1802 and using COS image

Error: error creating NodePool: googleapi: Error 500: Internal error encountered., backendError
anyone has any solution for this ???

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants