9. Deploy user clusters

NetApp Solutions

NetApp
May 24, 2021

Table of Contents

9. Deploy user clusters

With Anthos, organizations can scale their environments to incorporate multiple user clusters and segregate workloads between teams. A single admin cluster can support up to 20 user clusters, and each user cluster can support up to 250 nodes and 7500 pods.

To configure user clusters for your deployment, complete the following steps:

1. When the anthos-admin workstation is deployed, a file called `user-cluster.yaml` is created that can be used to deploy a number of additional user clusters for running workloads. Start by copying this default file with a new name for each cluster you intend to deploy.

```
ubuntu@gke-admin-ws-200915-151421:~ $ cp config.yaml anthos-cluster01-
config.yaml
```

2. Edit the `anthos-cluster01-config.yaml` file so that it is specific for the environment that is being deployed.

   a. In a manner similar to the `admin-config.yaml` used earlier, most of the variables are already filled in or they reference the admin-cluster for the information needed to deploy. This first section confirms the information for the version of Anthos being deployed and the vCenter instance it is being deployed on.

```
apiVersion: v1
kind: UserCluster
# (Required) A unique name for this cluster
name: "anthos-cluster01"
# (Required) GKE on-prem version (example: 1.3.0-gke.16)
gkeOnPremVersion: 1.6.0-gke.7
# # (Optional) vCenter configuration (default: inherit from the admin
cluster)
# vCenter:
#   resourcePool: ""
#   datastore: ""
#   # Provide the path to vCenter CA certificate pub key for SSL
verification
#   caCertPath: ""
#   # The credentials to connect to vCenter
#   credentials:
#     username: ""
#     password: ""
```

   b. You must fill out the networking section next and select whether you are using static or DHCP mode. If you are using static addresses, you must create an IP-block file to supply addresses similar to the admin-cluster configuration.

The items under the hostConfig section are global for any time static IPs are used in a deployment. This includes both static IPs for the cluster and those used for the SeeSaw load balancers, which are configured later.

```
# (Required) Network configuration; vCenter section is optional and
inherits from
# the admin cluster if not specified
network:
# (Required) Hostconfig for static addresseses on Seesaw LB's
  hostConfig:
    dnsServers:
    - "10.61.184.251"
    - "10.61.184.252"
    ntpServers:
    - "0.pool.ntp.org"
    - "1.pool.ntp.org"
    - "2.pool.ntp.org"
    searchDomainsForDNS:
    - "cie.netapp.com"
  ipMode:
    # (Required) Define what IP mode to use ("dhcp" or "static")
    type: dhcp
    # # (Required when using "static" mode) The absolute or relative
path to the yaml file
    # # to use for static IP allocation
    # ipBlockFilePath: ""
  # (Required) The Kubernetes service CIDR range for the cluster.
Must not overlap
  # with the pod CIDR range
  serviceCIDR: 10.96.0.0/12
  # (Required) The Kubernetes pod CIDR range for the cluster. Must
not overlap with
  # the service CIDR range
  podCIDR: 192.168.0.0/16
  vCenter:
    # vSphere network name
    networkName: VM_Network
```

c. Next fill out the load balancer section. This can vary depending on the type of load balancer being deployed.

SeeSaw Example:

```
# (Required) Load balancer configuration
loadBalancer:
```

```
  # (Required) The VIPs to use for load balancing
  vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.63.172.156"
    # Shared by all services for ingress traffic
    ingressVIP: "10.63.172.157"
  # (Required) Which load balancer to use "F5BigIP" "Seesaw" or
"ManualLB". Uncomment
  # the corresponding field below to provide the detailed spec
  kind: Seesaw
  # # (Required when using "ManualLB" kind) Specify pre-defined
nodeports
  # manualLB:
  #   # NodePort for ingress service's http (only needed for user
cluster)
  #   ingressHTTPNodePort: 30243
  #   # NodePort for ingress service's https (only needed for user
cluster)
  #   ingressHTTPSNodePort: 30879
  #   # NodePort for control plane service
  #   controlPlaneNodePort: 30562
  #   # NodePort for addon service (only needed for admin cluster)
  #   addonsNodePort: 0
  # # (Required when using "F5BigIP" kind) Specify the already-
existing partition and
  # # credentials
# f5BigIP:
  #  address:
  #  credentials:
  #    username:
  #    password:
  #  partition:
  #   # # (Optional) Specify a pool name if using SNAT
  #   # snatPoolName: ""
  # (Required when using "Seesaw" kind) Specify the Seesaw configs
  seesaw:
    # (Required) The absolute or relative path to the yaml file to
use for IP allocation
    # for LB VMs. Must contain one or two IPs.
    ipBlockFilePath: "anthos-cluster01-seesaw-block.yaml"
    # (Required) The Virtual Router IDentifier of VRRP for the Seesaw
group. Must
    # be between 1-255 and unique in a VLAN.
    vrid: 101
    # (Required) The IP announced by the master of Seesaw group
    masterIP: "10.63.172.153"
```

```
    # (Required) The number CPUs per machine
    cpus: 1
    # (Required) Memory size in MB per machine
    memoryMB: 2048
    # (Optional) Network that the LB interface of Seesaw runs in
(default: cluster
    # network)
    vCenter:
    # vSphere network name
      networkName: VM_Network
    # (Optional) Run two LB VMs to achieve high availability
(default: false)
    enableHA: false
```

d. For a SeeSaw load balancer, you must create an additional external file to supply the static IP information for the load balancer. Create the file `anthos-cluster01-seesaw-block.yaml` that was referenced in this configuration section.

```
blocks:
  - netmask: "255.255.255.0"
    gateway: "10.63.172.1"
    ips:
    - ip: "10.63.172.154"
      hostname: "anthos-cluster01-seesaw-vm"
```

F5 BigIP Example:

```
loadBalancer:
  # (Required) The VIPs to use for load balancing
  vips:
    # Used to connect to the Kubernetes API
    controlPlaneVIP: "10.63.172.158"
    # Shared by all services for ingress traffic
    ingressVIP: "10.63.172.159"
  # (Required) Which load balancer to use "F5BigIP" "Seesaw" or
"ManualLB". Uncomment
  # the corresponding field below to provide the detailed spec
  kind: F5BigIP
  # # (Required when using "ManualLB" kind) Specify pre-defined
nodeports
  # manualLB:
  #   # NodePort for ingress service's http (only needed for user
cluster)
  #   ingressHTTPNodePort: 30243
  #   # NodePort for ingress service's https (only needed for user
```

```
cluster)
  #   ingressHTTPSNodePort: 30879
  #   # NodePort for control plane service
  #   controlPlaneNodePort: 30562
  #   # NodePort for addon service (only needed for admin cluster)
  #   addonsNodePort: 0
  # # (Required when using "F5BigIP" kind) Specify the already-
existing partition and
  # # credentials
  f5BigIP:
    address: "172.21.224.21"
    credentials:
      username: "admin"
      password: "admin-password"
    partition: "Anthos-Cluster-01"
  #   # # (Optional) Specify a pool name if using SNAT
  #   # snatPoolName: ""
  # (Required when using "Seesaw" kind) Specify the Seesaw configs
  # seesaw:
    # (Required) The absolute or relative path to the yaml file to
use for IP allocation
    # for LB VMs. Must contain one or two IPs.
    #  ipBlockFilePath: ""
    # (Required) The Virtual Router IDentifier of VRRP for the Seesaw
group. Must
    # be between 1-255 and unique in a VLAN.
    #  vrid: 0
    # (Required) The IP announced by the master of Seesaw group
    #  masterIP: ""
    # (Required) The number CPUs per machine
    #  cpus: 4
    # (Required) Memory size in MB per machine
    #   memoryMB: 8192
    # (Optional) Network that the LB interface of Seesaw runs in
(default: cluster
    # network)
    #   vCenter:
      # vSphere network name
      #     networkName: VM_Network
    # (Optional) Run two LB VMs to achieve high availability
(default: false)
    #   enableHA: false
```

e. The final section describes the resources for the nodes that the cluster is deploying, including creating a node pool that we can use for dynamic scaling later. This section also supplies the service account keys to register the cluster with GKE once deployed.

```
# (Optional) User cluster master nodes must have either 1 or 3
replicas (default:
# 4 CPUs; 16384 MB memory; 1 replica)
masterNode:
  cpus: 4
  memoryMB: 8192
  # How many machines of this type to deploy
  replicas: 1
# (Required) List of node pools. The total un-tainted replicas across
all node pools
# must be greater than or equal to 3
nodePools:
- name: anthos-cluster01
  # # Labels to apply to Kubernetes Node objects
  # labels: {}
  # # Taints to apply to Kubernetes Node objects
  # taints:
  # - key: ""
  #   value: ""
  #   effect: ""
  cpus: 4
  memoryMB: 8192
  # How many machines of this type to deploy
  replicas: 3
# Spread nodes across at least three physical hosts (requires at
least three hosts)
antiAffinityGroups:
  # Set to false to disable DRS rule creation
  enabled: false
# # (Optional): Configure additional authentication
# authentication:
#   # (Optional) Configure OIDC authentication
#   oidc:
#     issuerURL: ""
#     kubectlRedirectURL: ""
#     clientID: ""
#     clientSecret: ""
#     username: ""
#     usernamePrefix: ""
#     group: ""
#     groupPrefix: ""
#     scopes: ""
#     extraParams: ""
#     # Set value to string "true" or "false"
#     deployCloudConsoleProxy: ""
```

```
#     # # The absolute or relative path to the CA file (optional)
#     # caPath: ""
#   # (Optional) Provide an additional serving certificate for the
API server
#   sni:
#     certPath: ""
#     keyPath: ""
#   # (Optional) Configure LDAP authentication (preview feature)
#   ldap:
#     name: ""
#     host: ""
#     # Only support "insecure" for now (optional)
#     connectionType: insecure
#     # # The absolute or relative path to the CA file (optional)
#     # caPath: ""
#     user:
#       baseDN: ""
#       userAttribute: ""
#       memberAttribute: ""
# (Optional) Specify which GCP project to connect your logs and
metrics to
stackdriver:
  projectID: "anthos-dev"
  # A GCP region where you would like to store logs and metrics for
this cluster.
  clusterLocation: "us-east1"
  enableVPC: false
  # The absolute or relative path to the key file for a GCP service
account used to
  # send logs and metrics from the cluster
  serviceAccountKeyPath: "/home/ubuntu/logging-monitoring-key.json "
# (Optional) Specify which GCP project to connect your GKE clusters
to
gkeConnect:
  projectID: "anthos-dev"
  # The absolute or relative path to the key file for a GCP service
account used to
  # register the cluster
  registerServiceAccountKeyPath: "/home/ubuntu/connect-register-
key.json"
  # The absolute or relative path to the key file for a GCP service
account used by
  # the GKE connect agent
  agentServiceAccountKeyPath: "/home/ubuntu/component-access-
key.json"
# (Optional) Specify Cloud Run configuration
```

```
cloudRun:
  enabled: false
# # (Optional/Alpha) Configure the GKE usage metering feature
# usageMetering:
#   bigQueryProjectID: ""
#   # The ID of the BigQuery Dataset in which the usage metering data
will be stored
#   bigQueryDatasetID: ""
#   # The absolute or relative path to the key file for a GCP service
account used by
#   # gke-usage-metering to report to BigQuery
#   bigQueryServiceAccountKeyPath: ""
#   # Whether or not to enable consumption-based metering
#   enableConsumptionMetering: false
# # (Optional/Alpha) Configure kubernetes apiserver audit logging
# cloudAuditLogging:
#   projectid: ""
#   # A GCP region where you would like to store audit logs for this
cluster.
#   clusterlocation: ""
#   # The absolute or relative path to the key file for a GCP service
account used to
#   # send audit logs from the cluster
#   serviceaccountkeypath: ""
```

3. After the edits to the configuration file are complete, NetApp recommends that the file be checked for proper syntax and spacing. You can check the config file you just created. This command references the `kubeconfig` file created by the admin-cluster.

```
ubuntu@gke-admin-200915-151421:~$ gkectl check-config --kubeconfig
kubeconfig --config anthos-cluster01-config.yaml
```

4. If you are using a SeeSaw load balancer, you need to create it prior to deploying the user cluster.

```
ubuntu@gke-admin-200915-151421:~$ gkectl create loadbalancer
--kubeconfig kubeconfig --config anthos-cluster-01-config.yaml
```

5. Create the user cluster. Just as we did with the admin cluster, the process can be accelerated by skipping the additional validations because we have already run the checks in the prior step.

```
ubuntu@gke-admin-200915-151421:~$ gkectl create cluster --config anthos-
cluster-01-config.yaml --skip-validation-all
```

6. When the cluster is deployed, it creates the kubeconfig file in the local directory. This file can be used to check the status of the cluster using kubectl or for running diagnostics with gkectl.

```
ubuntu@gke-admin-ws-200915-151421:~$ kubectl get nodes --kubeconfig
anthos-cluster01-kubeconfig
NAME                    STATUS   ROLES    AGE   VERSION
anthos-cluster01-7b5995cc45-ftrdw   Ready    <none>   5m   v1.18.6-
gke.6600
anthos-cluster01-7b5995cc45-z7q9b   Ready    <none>   5m   v1.18.6-
gke.6600
anthos-cluster01-7b5995cc45-zw6sv   Ready    <none>   6m   v1.18.6-
gke.6600
ubuntu@gke-admin-ws-200915-151421:~/ $ gkectl diagnose cluster
--kubeconfig kubeconfig --cluster-name anthos-cluster01
Diagnosing user cluster "anthos-cluster01"...

- Validation Category: User Cluster VCenter
Checking Credentials...SUCCESS
Checking VSphere CSI Driver...SUCCESS
Checking Version...SUCCESS
Checking Datacenter...SUCCESS
Checking Datastore...SUCCESS
Checking Resource pool...SUCCESS
Checking Folder...SUCCESS
Checking Network...SUCCESS
Checking Datastore...SUCCESS

- Validation Category: User Cluster
Checking onpremusercluster and onpremnodepool...SUCCESS
Checking cluster object...SUCCESS
Checking machine deployment...SUCCESS
Checking machineset...SUCCESS
Checking machine objects...SUCCESS
Checking control place pods...SUCCESS
Checking gke-connect pods...SUCCESS
Checking config-management-system pods...Warning: No pod is running in
namespace "config-management-system"...SUCCESS
Checking kube-system pods...SUCCESS
Checking gke-system pods...SUCCESS
Checking storage...SUCCESS
Checking resource...System pods on UserNode cpu resource request report:
total 3059m nodeCount 3 min 637m max 1224m avg 1019m tracked amount in
bundle 4000m
System pods on UserNode memory resource request report: total 6464Mi
nodeCount 3 min 1670Mi max 2945Mi avg 2259331754 tracked amount in
bundle 8192Mi
SUCCESS
Cluster is healthy.
```

Next: Enable access to the cluster with the GKE console.