ORACLE

Automating Oracle Container Engine for Kubernetes (OKE) Deployment on Oracle Compute Cloud@Customer or Private Cloud Appliance using OpenTofu

Step-by-step to automate deployments of Oracle Container Engine for Kubernetes (OKE) on Private Cloud Appliance or Compute Cloud@Customer using OpenTofu

Version [1.0]
Copyright © 2024, Oracle and/or its affiliates
Public

ORACLE

Purpose statement

This document provides the step-by-step to fully automate the deployment of Oracle Container Engine for Kubernetes (OKE) on Oracle Compute Cloud@Customer or Private Cloud Appliance using customized OpenTofu scripts.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Copyright © 2024, Oracle and/or its affiliates / Public

ORACLE

Table of contents

Copyright © 2024, Oracle and/or its affiliates / Public

ORACLE

Introduction

In today's changing world of business IT, automation and native solutions play a crucial role in ensuring flexibility, scalability and effectiveness. As more companies embrace Kubernetes for managing containers the importance of deployment processes becomes clear. This guide helps to address this need by offering a step-by-step on automating the setup of Oracle Container Engine for Kubernetes (OKE) on Compute Cloud@Customer and Private Cloud Appliance using tailored OpenTofu scripts.

OpenTofu is an open-source Infrastructure as Code (IaC) frameworks that enable users to define, provision, and manage cloud resources through declarative configuration files. OpenTofu is a community-driven alternative, built with similar principles, offering cross-cloud management while promoting open governance and flexibility.

By utilizing OpenTofu, organizations can achieve automated deployments reducing manual work and minimizing the chances of errors.

This content is provided for informational purposes and self-supported guidance only. Consultancy or other assistance related to the content is not covered under the Oracle Support contract or associated service requests. If you have questions or additional needs, then please do reach out to your Oracle Sales contact directly.

Copyright © 2024, Oracle and/or its affiliates / Public

Prerequisite

Before getting into the steps for setting up Oracle Container Engine for Kubernetes (OKE) on Oracle Compute Cloud@Customer or Private Cloud Appliance, it's important to make sure you have all the necessary prerequisites ready. This section outlines the elements needed to carry out the automation process using customized OpenTofu scripts.

The prerequisites covered in this document encompass operational aspects, including:

- **Having access to Compute Cloud@Customer or Private Cloud Appliance:** Make sure you have an active account with the appropriate permissions to create and manage resources on Oracle Compute Cloud@Customer or Private Cloud Appliance.
- **OKE Pre-requisites:** Understanding the pre-requisites needed before deploying OKE on Compute Cloud@Customer or Private Cloud Appliance. For example: Groups, policies and tags configuration which are mandatory before executing the OpenTofu scripts to deploy OKE. Refer to the Administrator Tasks section of the Container Engine for Kubernetes on Compute Cloud@Customer official documentation or Overview of Container Engine for Kubernetes on Oracle Private Cloud Appliance..

NOTE: The OpenTofu scripts listed on this solution paper includes the configuration of all pre-requisites for OKE.

> ✓ **For C3 the pre-requisite Terraform or OpenTofu must point to OCI tenancy and then other OKE terraform or OpenTofu point to C3 System.**
>
> ✓ **For PCA the pre-requisite will work directly point to PCA config.**

- **OpenTofu Setup:** Confirm that OpenTofu is installed and set up on your machine or CI/CD pipeline environment. It's also helpful to be familiar with OpenTofu concepts and syntax.
- **Networking Considerations:** Ensure that your network setup allows for communication, which includes setting up VCN (Virtual Cloud Network) configuring subnets and managing security lists.
- **Access Credentials:** You need a bastion host to connect with your Compute Cloud@Customer or Private Cloud Appliance to deploy the OKE. Collect the API keys SSH keys and Oracle Compute Cloud@Customer or Private Cloud Appliance credentials required for authentication and authorization throughout the deployment process.
- **Infrastructure Planning:** Develop a strategy outlining the resources needed capacity projections and design aspects customized to suit your organizations requirements.
- **Fundamental Understanding of Scripting:** Proficiency, in scripting languages in adapting and tailoring OpenTofu scripts for deployment situations.

By fulfilling these criteria, you will be ready to follow the instructions outlined in this paper guaranteeing a seamless and effective automation of your Oracle Container Engine for Kubernetes on Oracle Compute Cloud@Customer or Private Cloud Appliance.

Bastion Host Configuration

A bastion host, also referred to as a jump server, serves as a server created to enable access to a private network from an external network, like the internet. For the OKE deployment on Oracle Compute Cloud@Customer or Private Cloud Appliance, a bastion host is utilized to host all OpenTofu scripts utilized to deploy the OKE infrastructure. Also, it is utilized for reaching resources and instances within a subnet in your Compute Cloud@Customer or Private Cloud Appliance environments.

Copyright © 2024, Oracle and/or its affiliates / Public

Advantages of Using a Bastion Host are:

- Access Point: Acts as an entry point for administrators and users to connect to instances in the private network without exposing them directly to the internet.
- Offers a controlled gateway thereby reducing the vulnerability surface.
- Isolation of Critical Resources: Ensures that computing resources within the network are safeguarded and isolated from internet exposure. Helps in minimizing risks associated with attacks on systems.
- Monitoring: Centralizes access logging simplifying tracking and auditing access to resources. Can be integrated with security information and event management (SIEM) systems, for monitoring capabilities.

By implementing a bastion host, you can securely and efficiently manage access to your Oracle Compute Cloud@Customer or Private Cloud Appliance resources, ensuring that your private instances remain protected and accessible only through a secure, controlled entry point.

1. Install OCI CLI in your bastion host. For the step-by-step and how to install OCI CLI and properly setup the OCI CLI configuration file in your instance, refer to the following links below: https://docs.oracle.com/en-us/iaas/Content/API/SDKDocs/cliinstall.htm and https://docs.oracle.com/en/engineered-systems/private-cloud-appliance/3.0-latest/user/user-usr-ce-cli.html#usr-cli-obtain-cabundle.
2. Ensure you have OpenTofu properly installed: Refer to OpenTofu official installation documentation: https://opentofu.org/docs/intro/install/.
3. Ensure that your bastion host can access your Compute Cloud@Customer or Private Cloud Appliance and properly authenticate using OCI CLI and OpenTofu.

OpenTofu Scripts Overview

To make it easier to set up and manage Oracle Kubernetes Engine (OKE) on Oracle Compute Cloud@Customer or Private Cloud Appliance, we use a range of OpenTofu scripts. The set of scripts listed below will automate the OKE deployment (end-to-end) including all resources needed, such as: OKE Pre-requisites including groups, authorizations, and tags. VCN, subnetwork, security lists, OKE clusters, and node pool.

Listed below is the description of each OpenTofu script utilized to fully deploy OKE.

NOTE: Refer to the Appendix section of this white paper for the source code of all OpenTofu scripts. Review the variables highlighted in red and adjust to your environment accordingly.

main.tf:

- **Description**: Acts as the primary configuration file, orchestrating the deployment of all resources within the Oracle Kubernetes Engine (OKE) environment. It integrates various modules and defines the overarching structure of the OpenTofu deployment.

provider.tf:

- **Description**: Specifies provider configurations and credentials necessary for authenticating and interacting with Oracle Compute Cloud@Customer or Private Cloud Appliance.

oke_worker_subnet.tf:

- **Description**: Configures the subnet designated for OKE worker nodes, detailing the network settings and parameters to ensure proper isolation and connectivity within the OKE cluster.

Copyright © 2024, Oracle and/or its affiliates / Public

oke_worker_seclist.tf:

- **Description**: Defines the security list configurations for the OKE worker nodes subnet, specifying the inbound and outbound rules to manage traffic and ensure secure operations of worker nodes.

oke_vcn.tf:

- **Description**: Establishes the Virtual Cloud Network (VCN) setup, including the definition of subnets, route tables, and other network components necessary for the OKE cluster's infrastructure.

oke_kmi_subnet.tf:

- **Description**: Configures the subnet for Kubernetes Master Instances within the OKE cluster, ensuring that the master nodes have a dedicated and properly configured network space.

oke_kmi_seclist.tf:

- **Description**: Specifies the security list settings for the Kubernetes Master Instances subnet, detailing the rules required to control network traffic and secure the master nodes.

variables.tf:

- **Description**: Declares input variables used throughout the OpenTofu configuration, allowing for customization and reuse of the deployment parameters.

image.tf:

- **Description**: Specifies the Oracle Compute Cloud@Customer or Private Cloud Appliance images used to create the compute instances within the OKE cluster.

NOTE: There is a difference on the OKE platform image name on Private Cloud Appliance versus Oracle Compute Cloud@Customer, which need to be adjusted on the **image.tf** script before executing the scripts. This is variable is highlighted in red on the image.tf script in the appendix section of this solution paper.

To check the list of all OKE platform image available in both platforms, Compute Cloud@Customer or Private Cloud Appliance, run the **oci compute image list** command line listed below, then choose the most recent OKE platform image. The **oci compute image list** command line is the same for Compute Cloud@Customer or Private Cloud Appliance.

Example:

oci compute image list --compartment-id <your-compartment-ID> | grep -i uln-pca-Oracle-Linux.

Listed below is the output of the oci compute image list command with platform images available on this Private Cloud Appliance system. The most recent OKE platform image available on this system is the **uln-pca-Oracle-Linux8-OKE-1.28.3-20240713.oci.** For this example, we will choose this image since it is the most recent OKE platform image.

```
"display-name": "uln-pca-Oracle-Linux8-OKE-1.27.7-20240209.oci",
"display-name": "uln-pca-Oracle-Linux8-OKE-1.27.7-20240713.oci",
"display-name": "uln-pca-Oracle-Linux8-OKE-1.28.3-20240210.oci",
"display-name": "uln-pca-Oracle-Linux8-OKE-1.28.3-20240713.oci"
```


Copyright © 2024, Oracle and/or its affiliates / Public

NOTE: On Oracle Compute Cloud@Customer the same image will be the **uln-pca-Oracle-Linux8-OKE-1.28.3**, without the dash and date.

oke_cluster.tf:

- **Description**: Defines the settings and parameters for deploying the OKE cluster, including cluster size, version, and other critical configurations necessary for the cluster setup.

oke_pool.tf:

- **Description**: Configures the node pool for the OKE cluster, specifying the characteristics and sizing of the worker nodes.

terraform.tfvars:

- **Description**: Contains the variable values used to customize the OpenTofu deployment, including sensitive information such as API keys, OCID of Tenancy, and compartment details. This file centralizes configuration data, making it easier to manage and secure.

oke_dyn_grp.tf

- **Description**: Contains the code to automate the setup of a new dynamic-group on Compute Cloud@Customer or Private Cloud Appliance.

oke_policy.tf

- **Description**: Contains the code to automate the setup of policies for the dynamic-group on Compute Cloud@Customer or Private Cloud Appliance.

oke_tags_grp.tf

- **Description**: Contains the code to automate the setup Tag Namespace and key definition for OKE on Compute Cloud@Customer or Private Cloud Appliance.

NOTE: Ensure you have all OpenTofu scripts listed below together in a folder in your bastion host. For demonstration purposes, all scripts are utilizing standard network CIDR configuration, OKE cluster/node pool names, and sample names for authentication. Ensure to adjust the highlighted in red of each OpenTofu script listed on the appendix on this solution paper to best fit your environment. Below is a sample of a bastion host instance with all OpenTofu scripts needed for OKE deployment on Compute Cloud@Customer or Private Cloud Appliance.

```
-rw-r--r--. 1 root root 2077 Sep  3 18:55 main.tf
-rw-r--r--. 1 root root  112 Sep  3 18:55 provider.tf
-rw-r--r--. 1 root root  850 Sep  3 18:55 oke_worker_subnet.tf
-rw-r--r--. 1 root root 1305 Sep  3 18:55 oke_worker_seclist.tf
-rw-r--r--. 1 root root 1167 Sep  3 18:55 oke_vcn.tf
-rw-r--r--. 1 root root  912 Sep  3 18:55 oke_kmi_subnet.tf
-rw-r--r--. 1 root root 2202 Sep  3 18:55 oke_kmi_seclist.tf
-rw-r--r--. 1 root root 2676 Sep  5 19:37 variables.tf
-rw-r--r--. 1 root root  323 Sep  5 20:17 image.tf
-rw-r--r--. 1 root root  674 Sep  5 20:07 oke_cluster.tf
-rw-r--r--. 1 root root  799 Sep  5 20:05 oke_pool.tf
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
-rw-r--r--. 1 root root 1331 Sep  5 19:35 terraform.tfvars
-rw-r--r--. 1 root root  906 Sep  5 19:33 oke_policy.tf
-rw-r--r--. 1 root root  380 Sep  5 19:50 oke_dyn_grp.tf
-rw-r--r--. 1 root root  506 Sep  5 19:51 oke_tags_grp.tf
```

NOTE: Refer to the Appendix section for detailed information of all OpenTofu scripts utilized to fully deploy OKE on Compute Cloud@Customer or Private Cloud Appliance.

OKE Deployment Workflow

Setting up Oracle Kubernetes Engine (OKE) on Oracle Compute Cloud@Customer or Private Cloud Appliance involves a deployment process divided into three phases, they are:

1. **Establishing Network Infrastructure:** The initial step focuses on creating the network infrastructure, such, as configuring Virtual Cloud Networks (VCNs) subnets and security lists tailored for OKE worker and master nodes. This setup enables the network environment to support isolated communication channels required for the Kubernetes clusters operation.
2. **Deploying OKE Cluster:** After setting up the network the next phase involves deploying the OKE cluster itself. This process includes creating the Kubernetes cluster serving as the orchestration platform for applications. The deployment ensures that the cluster is correctly initialized and prepared to handle workloads.
3. **Configuring OKE Node Pools:** The final stage revolves around setting up and deploying node pools, for the OKE cluster. This step encompasses provisioning worker nodes, where containerized applications will be executed. By deploying node pools, the cluster is furnished with resources to manage application workloads effectively.

OKE Deployment Using OpenTofu or Terraform

Assuming you have the bastion host properly configured, and all OpenTofu scripts adjusted to your environment, perform the following steps to deploy OKE on Compute Cloud@Customer or Private Cloud Appliance.

- Run **tofu init** command line: **tofu init** command is used to initialize a working directory containing OpenTofu configuration files. This command performs several essential tasks to set up the working directory for future OpenTofu commands.

    - **Initializes Backend Configuration**: Sets up the backend where OpenTofu state file will be stored. This could be a local file or a remote storage solution.
    - **Installs Provider Plugins**: Downloads the necessary provider plugins specified in the configuration files to manage the resources.
    - **Validates Configuration Files**: Checks the syntax and structure of the OpenTofu configuration files to ensure they are correct.

Copyright © 2024, Oracle and/or its affiliates / Public

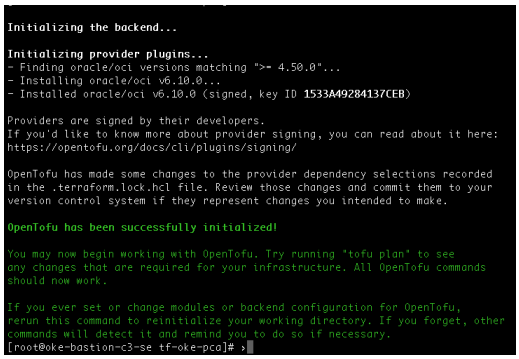

Figure 1. OpenTofu init command line output

- Adjust the script variables to best fit your environment. For example: OCIDs, public keys, VCN/subnet name, network CIDR, OKE Cluster and pool name. Refer to the appendix section for details.
- In the same folder in your bastion host, run **tofu apply** command, then when asked by OpenTofu, type **yes** to perform the actions needed to fully deploy OKE on Compute Cloud@Customer or Private Cloud Appliance.

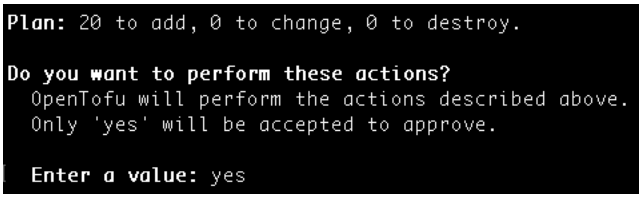

Figure 2. OpenTofu/Terraform scripts execution to deploy OKE on Compute Cloud@Customer or Private Cloud Appliance

NOTE: Usually, the full deployment of OKE on Compute Cloud@Customer or Private Cloud Appliance takes about 15 minutes. Listed below is the output log of a successful deployment.

oci_identity_tag_namespace.oracle-pca: Creating...
oci_core_vcn.oke_vcn: Creating...
oci_identity_tag_namespace.oracle-pca: Creation complete after 1s
oci_identity_tag.cluster-id: Creating...
oci_identity_tag.cluster-id: Creation complete after 2s]
oci_identity_dynamic_group.oke-dynamic-grp: Creating...
oci_identity_dynamic_group.oke-dynamic-grp: Creation complete after 2s
oci_identity_policy.oke-dyn-grp-policy: Creating...
oci_identity_policy.oke-dyn-grp-policy: Creation complete after 1s
oci_core_vcn.oke_vcn: Still creating... [10s elapsed]
oci_core_vcn.oke_vcn: Still creating... [20s elapsed]
oci_core_vcn.oke_vcn: Still creating... [30s elapsed]
oci_core_vcn.oke_vcn: Creation complete after 37s
oci_core_internet_gateway.vcn_igs: Creating...
oci_core_nat_gateway.vcn_ngs: Creating...
oci_core_security_list.workerlb: Creating...
oci_core_security_list.kmilb: Creating...
oci_core_default_security_list.oke_vcn: Creating...
oci_core_security_list.kmi: Creating...

Copyright © 2024, Oracle and/or its affiliates / Public

```
oci_core_security_list.worker: Creating...
oci_core_security_list.worker: Creation complete after 3s
oci_core_default_security_list.oke_vcn: Creation complete after 3s
oci_core_subnet.worker: Creating...
oci_core_security_list.kmi: Creation complete after 5s
oci_core_subnet.kmi: Creating...
oci_core_security_list.kmilb: Creation complete after 6s
oci_core_security_list.workerlb: Creation complete after 6s
oci_core_nat_gateway.vcn_ngs: Creation complete after 6s
oci_core_default_route_table.private: Creating...
oci_core_subnet.worker: Creation complete after 4s
oci_core_internet_gateway.vcn_igs: Creation complete after 8s
oci_core_route_table.public: Creating...
oci_core_subnet.kmi: Creation complete after 5s
oci_core_route_table.public: Creation complete after 3s
oci_core_subnet.kmi_lb: Creating...
oci_core_subnet.worker_lb: Creating...
oci_core_subnet.worker_lb: Creation complete after 3s
oci_core_subnet.kmi_lb: Creation complete after 4s
oci_containerengine_cluster.OKECluster: Creating...
oci_core_default_route_table.private: Still creating... [10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [10s elapsed]
oci_core_default_route_table.private: Still creating... [20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [20s elapsed]
oci_core_default_route_table.private: Still creating... [30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [30s elapsed]
oci_core_default_route_table.private: Still creating... [40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [40s elapsed]
oci_core_default_route_table.private: Still creating... [50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [50s elapsed]
oci_core_default_route_table.private: Still creating... [1m0s elapsed]
oci_core_default_route_table.private: Creation complete after 1m2s
oci_containerengine_cluster.OKECluster: Still creating... [1m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [1m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [1m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [1m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [1m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [1m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [2m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [2m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [2m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [2m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [2m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [2m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [3m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [3m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [3m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [3m30s elapsed]
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
oci_containerengine_cluster.OKECluster: Still creating... [3m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [3m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [4m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [4m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [4m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [4m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [4m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [4m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [5m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [5m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [5m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [5m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [5m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [5m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [6m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [6m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [6m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [6m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [6m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [6m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [7m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [7m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [7m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [7m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [7m40s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [7m50s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [8m0s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [8m10s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [8m20s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [8m30s elapsed]
oci_containerengine_cluster.OKECluster: Still creating... [8m40s elapsed]
oci_containerengine_cluster.OKECluster: Creation complete after 8m46s
oci_containerengine_node_pool.OKENodePool: Creating...
oci_containerengine_node_pool.OKENodePool: Still creating... [10s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [20s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [30s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [40s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [50s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [1m0s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [1m10s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [1m20s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [1m30s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [1m40s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [1m50s elapsed]
oci_containerengine_node_pool.OKENodePool: Still creating... [2m0s elapsed]
oci_containerengine_node_pool.OKENodePool: Creation complete after 2m8s
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
Apply complete! Resources: 20 added, 0 changed, 0 destroyed.
```

Listed below is the broken-down of the OpenTofu scripts execution.

First, the OpenTofu scripts will create the pre-requisites and all network infrastructure, such as: Tags, authorization, dynamic-group, policies, VCNs, subnets, security lists, internet and NAT gateways needed for OKE.

oci_identity_tag_namespace.oracle-pca: Creating...
oci_core_vcn.oke_vcn: Creating...
oci_identity_tag_namespace.oracle-pca: Creation complete after 1s
oci_identity_tag.cluster-id: Creating...
oci_identity_tag.cluster-id: Creation complete after 2s]
oci_identity_dynamic_group.oke-dynamic-grp: Creating...
oci_identity_dynamic_group.oke-dynamic-grp: Creation complete after 2s
oci_identity_policy.oke-dyn-grp-policy: Creating...
oci_identity_policy.oke-dyn-grp-policy: Creation complete after 1s
oci_core_vcn.oke_vcn: Still creating... [10s elapsed]
oci_core_vcn.oke_vcn: Still creating... [20s elapsed]
oci_core_vcn.oke_vcn: Still creating... [30s elapsed]
oci_core_vcn.oke_vcn: Creation complete after 37s
oci_core_internet_gateway.vcn_igs: Creating...
oci_core_nat_gateway.vcn_ngs: Creating...
oci_core_security_list.workerlb: Creating...
oci_core_security_list.kmilb: Creating...
oci_core_default_security_list.oke_vcn: Creating...
oci_core_security_list.kmi: Creating...
oci_core_security_list.worker: Creating...
oci_core_security_list.worker: Creation complete after 3s
oci_core_default_security_list.oke_vcn: Creation complete after 3s
oci_core_subnet.worker: Creating...
oci_core_security_list.kmi: Creation complete after 5s
oci_core_subnet.kmi: Creating...
oci_core_security_list.kmilb: Creation complete after 6s
oci_core_security_list.workerlb: Creation complete after 6s
oci_core_nat_gateway.vcn_ngs: Creation complete after 6s
oci_core_default_route_table.private: Creating...
oci_core_subnet.worker: Creation complete after 4s
oci_core_internet_gateway.vcn_igs: Creation complete after 8s
oci_core_route_table.public: Creating...
oci_core_subnet.kmi: Creation complete after 5s
oci_core_route_table.public: Creation complete after 3s
oci_core_subnet.kmi_lb: Creating...
oci_core_subnet.worker_lb: Creating...
oci_core_subnet.worker_lb: Creation complete after 3s
oci_core_subnet.kmi_lb: Creation complete after 4s

- Check if all network configuration has been properly deployed in your Compute Cloud@Customer or Private Cloud Appliance. To do this step, login in the UI of your Compute Cloud@Customer or Private Cloud

Copyright © 2024, Oracle and/or its affiliates / Public

Appliance, click on Dashboard and Networking to view your Virtual Cloud Network configuration, then select your compartment which you are deploying your new OKE cluster. The new VCN will be created. The new VCN for your OKE cluster will be available.

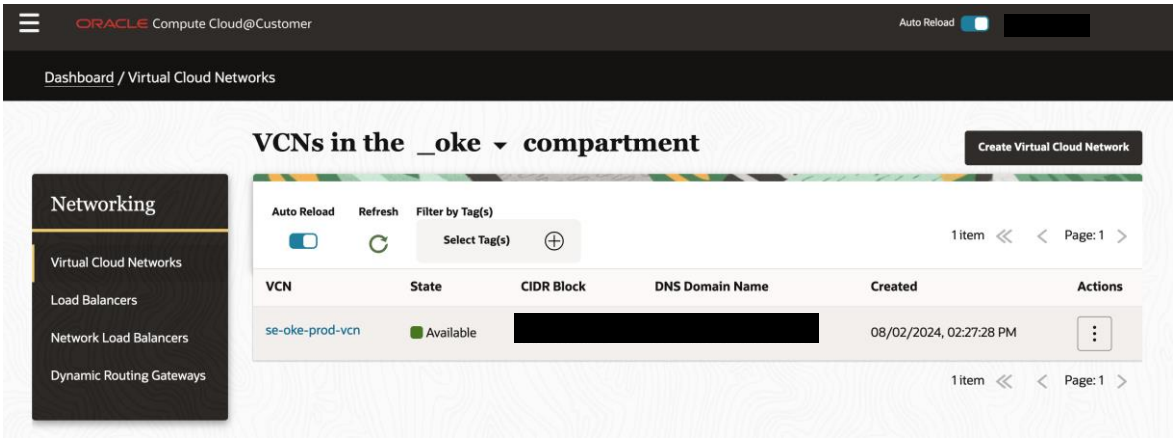

Figure 3. Compute Cloud@Customer or Private Cloud Appliance Virtual Cloud Network Configuration

- On the same place, click on your VCN name, then go to subnets to list all subnets and network resources deployed by the Terraform scripts, such as: Security lists, internet and NAT gateways, and route tables. NOTE: On this example, we are using the se-oke-prod-vcn.

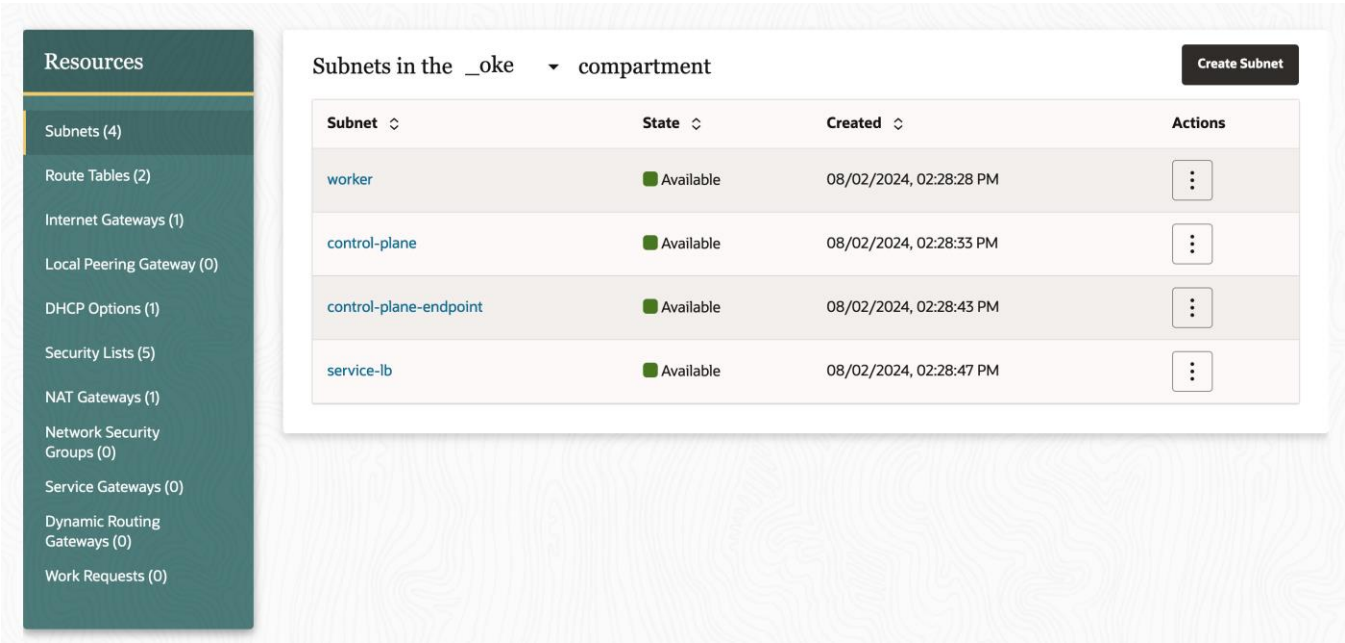

Figure 4. Compute Cloud@Customer or Private Cloud Appliance subnets and network resources overview

Copyright © 2024, Oracle and/or its affiliates / Public

- Second, the Terraform scripts will create the Kubernetes cluster. This step can take up to 10-12 minutes. During this step, the new OKE cluster will be on the “Creating” status on the Compute Cloud@Customer or Private Cloud Appliance dashboard. See output log from Terraform script below:

```
oci_containerengine_cluster.SEOKECluster: Creating...
oci_containerengine_cluster.SEOKECluster: Still creating... [10s elapsed]...[10m50s elapsed]

Apply complete! Resources: 20 added, 0 changed, 0 destroyed.
```

- Check if the new OKE cluster has been properly deployed in your Compute Cloud@Customer or Private Cloud Appliance. To do this step, login in your Compute Cloud@Customer or Private Cloud Appliance UI, click on Dashboard, then Containers. Your new OKE cluster will be available. See Figures 5 and 6 below.

NOTE: On this example, we are using the **oke_cluster-1** as the name of our new OKE cluster, and **oke_node_pool** as the name of the OKE node pool on Compute Cloud@Customer or Private Cloud Appliance.

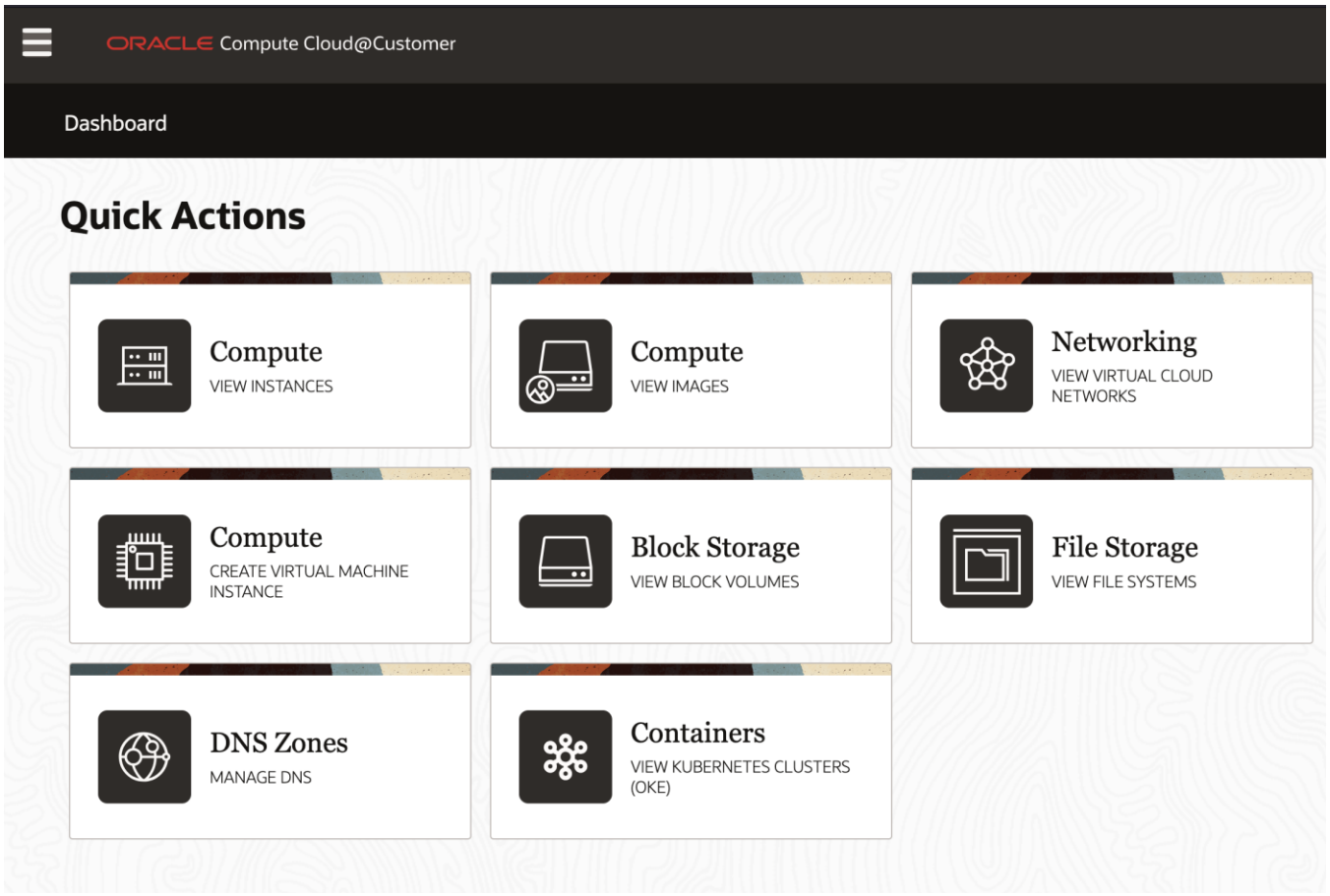

Figure 5. Compute Cloud@Customer or Private Cloud Appliance Dashboard

Copyright © 2024, Oracle and/or its affiliates / Public

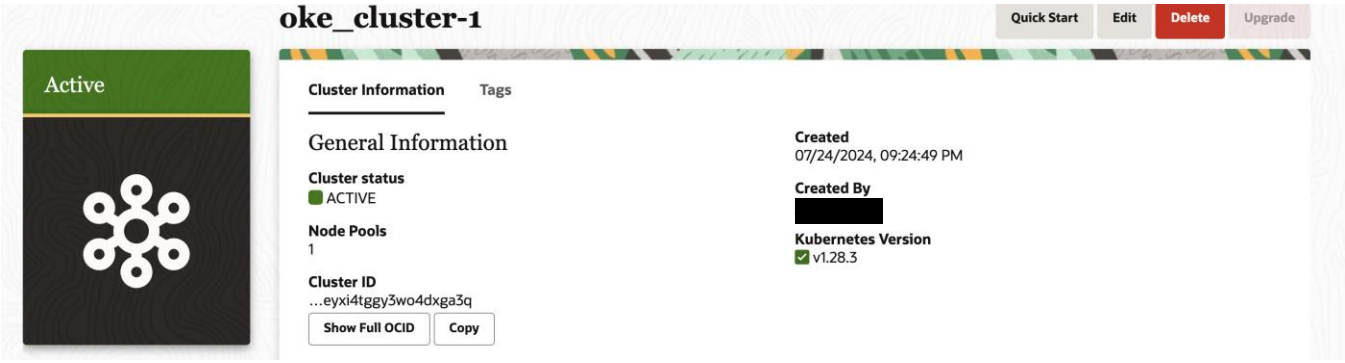

Figure 6. Compute Cloud@Customer Dashboard or Private Cloud Appliance. OKE Clusters.

- Third, the Terraform scripts will create the node pool for your Kubernetes cluster and the new OKE cluster will be available in your Compute Cloud@Customer or Private Cloud Appliance for new workload.

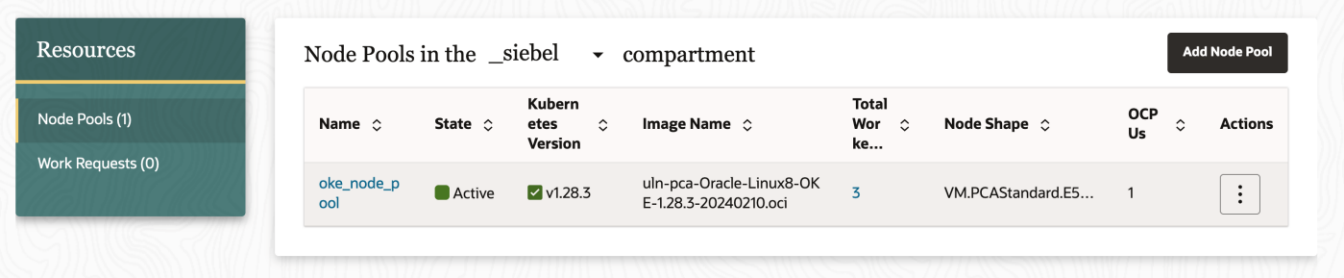

Figure 7. Compute Cloud@Customer or Private Cloud Appliance Dashboard. OKE Node Pools.

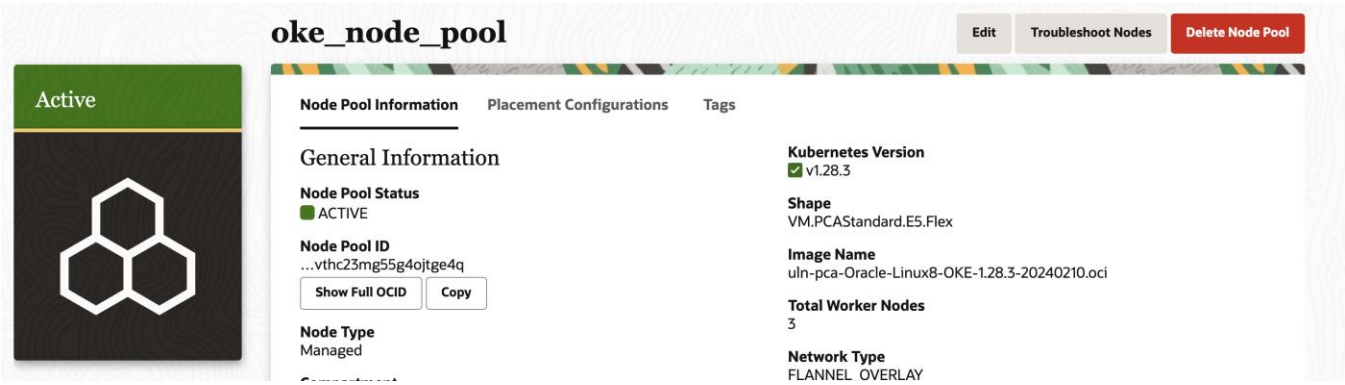

Figure 8. Compute Cloud@Customer or Private Cloud Appliance Dashboard. OKE Node Pool details

At this stage, the new OKE cluster will be fully deployed on Compute Cloud@Customer or Private Cloud Appliance and ready to receive new workloads.

Copyright © 2024, Oracle and/or its affiliates / Public

The Appendix section of this solution paper has the source code of all OpenTofu scripts utilized to fully deploy OKE on Compute Cloud@Customer or Private Cloud Appliance.

IMPORTANT: Review the variables highlighted in red below and adjust them to best fit your environment. For example, network ip addressing, cluster-name, credentials, tenancy information, all need to be adjusted to your environment.

Copyright © 2024, Oracle and/or its affiliates / Public

Appendix

main.tf

```
terraform {
  required_providers {
    oci = {
      source  = "oracle/oci"
      version = ">= 4.50.0"
    }
  }
  required_version = ">= 1.1"
}
locals {
  kube_internal_cidr = "253.255.0.0/16"
  worker_lb_ingress_rules = var.worker_lb_ingress_rules
  worker_ingress_rules = flatten([var.worker_ingress_rules, [
    {
      source   = var.vcn_cidr
      port_min = 22
      port_max = 22
    },
    {
      source   = var.workerlb_cidr
      port_min = 30000
      port_max = 32767
    },
    {
      source   = var.workerlb_cidr
      port_min = 10256
      port_max = 10256
    },
    {
      source   = var.kmi_cidr
      port_min = 22
      port_max = 65535
    },
  ]])
  worker_ingress_udp_rules = [
    {
      source   = var.worker_cidr
      port_min = 8285
      port_max = 8472
    },
    {
      source   = var.kmi_cidr
      port_min = 8285
      port_max = 8472
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
  },
]
kmi_lb_ingress_rules = [
  {
    source   = local.kube_internal_cidr
    port_min = var.kubernetes_api_port
    port_max = var.kubernetes_api_port
  },
  {
    source   = var.kube_client_cidr
    port_min = var.kubernetes_api_port
    port_max = var.kubernetes_api_port
  },
  {
    source   = var.vcn_cidr
    port_min = var.kubernetes_api_port
    port_max = var.kubernetes_api_port
  },
]
kmi_ingress_rules = [
  {
    source   = var.kube_client_cidr
    port_min = var.kubernetes_api_port
    port_max = var.kubernetes_api_port
  },
  {
    source   = var.kmilb_cidr
    port_min = var.kubernetes_api_port
    port_max = var.kubernetes_api_port
  },
  {
    source   = var.worker_cidr
    port_min = 1024
    port_max = 65535
  },
  {
    source   = var.kmi_cidr
    port_min = 1024
    port_max = 65535
  },
]
kmi_ingress_udp_rules = [
  {
    source   = var.worker_cidr
    port_min = 8285
    port_max = 8472
  },
  {
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
      source   = var.kmi_cidr
      port_min = 8285
      port_max = 8472
    },
  ]
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

terraform.tfvars

```
# Documentation:
# https://github.com/oracle-terraform-modules/terraform-oci-oke/blob/master/docs/configuration.adoc
# name of the profile to use from $HOME/.oci/config
oci_config_file_profile = "DEFAULT"

# tenancy ocid from the above profile
tenancy_ocid = "ocid1.tenancy.XXXXXXX" # Adjust to the OCID of your tenancy.

# compartment in which to build the OKE cluster
compartment_id = "ocid1.compartment.XXXXX" # Adjust to your compartment ID.

# Dynamic Group Name
oke_dyn_grp = "oke-dyn-ip-group"

# OKE Dynamic Group Policy Name
oke_policy_name = "oke-dyn-grp-policy"

# display-name for the OKE VCN
vcn_name = "secluster" # Adjust to the name of your prefrence

# CIDR of clients that are allowed to contact k8s apiserver
kube_client_cidr = "10.0.0.0/8"

# security list rules for who is allowed to contact the worker load balancer
# (adjust for your applications)
worker_lb_ingress_rules = [
  {
    source   = "10.0.0.0/8"
    port_min = 80
    port_max = 80
  },
  {
    source   = "10.0.0.0/8"
    port_min = 443
    port_max = 443
  },
]
# security list rules for who is allowed to contact worker nodes directly
# This example allows 10/8 to contact the default nodeport range
worker_ingress_rules = [
  {
    source   = "10.0.0.0/8"
    port_min = 30000
    port_max = 32767
  },
]
```


Copyright © 2024, Oracle and/or its affiliates / Public

variables.tf

```
# Documentation:
# https://github.com/oracle-terraform-modules/terraform-oci-oke/blob/master/docs/configuration.adoc
variable "ClusterName" {
default = "C1-test" # Adjust to the name of your new OKE cluster.
}
variable "public_key_oci" {
default = "/root/.oci/XXX.pub" # Adjust to the correct name of your public key.
}
variable "kubernetes_version" {
   default = "v1.28.3"
}
variable "node_pool_size" {
  default = 3 # Adjust to the size of your node pool.
}
variable "Shape" {
 default = "VM.PCAStandard1.Flex" # Adjust to the correct type of shape. NOTE: If you are deploying OKE on Private Cloud Appliance, use VM.PCAStandard1.Flex shape. If you are deploying OKE on Compute Cloud@Customer use VM.PCAStandard.E5.Flex shape.
}
variable "oci_config_file_profile" {
  type    = string
  default = "DEFAULT"
}
variable "tenancy_ocid" {
  description = "tenancy OCID"
  type        = string
  nullable    = false
}
variable "compartment_id" {
  description = "compartment OCID"
  type        = string
  nullable    = false
}
variable "oke_dyn_grp" {
    description = "Dynamic group that needs to be created for instance principal"
    default = "oke-dyn-ip-grp"
}
variable "oke_policy_name" {
    description = "Policy set name for dynamic group"
    default = "oke-instance-principal-policy"
}
variable "vcn_name" {
  description = "VCN name"
  nullable    = false
}
variable "kube_client_cidr" {
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
  description = "CIDR of Kubernetes API clients"
  type        = string
  nullable    = false
}
variable "kubernetes_api_port" {
  description = "port used for kubernetes API"
  type        = string
  default     = "6443"
}
variable "worker_lb_ingress_rules" {
  description = "traffic allowed to worker load balancer"
  type = list(object({
    source   = string
    port_min = string
    port_max = string
  }))
  nullable = false
}
variable "worker_ingress_rules" {
  description = "traffic allowed directly to workers"
  type = list(object({
    source   = string
    port_min = string
    port_max = string
  }))
  nullable = true
}

#
# IP network addressing
#
variable "vcn_cidr" {
  default = "172.31.252.0/23"
}
# Subnet for KMIs where kube-apiserver and other control
# plane applications run, max 9 nodes
variable "kmi_cidr" {
  description = "K8s control plane subnet CIDR"
  default     = "172.31.252.224/28"
}
# Subnet for KMI load balancer
variable "kmilb_cidr" {
  description = "K8s control plane LB subnet CIDR"
  default     = "172.31.252.240/28"
}
# Subnet for worker nodes, max 128 nodes
variable "worker_cidr" {
  description = "K8s worker subnet CIDR"
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
  default     = "172.31.253.0/24"
}

# Subnet for worker load balancer (for use by CCM)
variable "workerlb_cidr" {
  description = "K8s worker LB subnet CIDR"
  default     = "172.31.252.0/25"
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

provider.tf

```
provider "oci" {
  config_file_profile = var.oci_config_file_profile
  tenancy_ocid        = var.tenancy_ocid
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

image.tf

```
data "oci_core_images" "image1" {
    #Required
    compartment_id           = var.compartment_id
    #Optional
    operating_system         = "OracleLinux"
    operating_system_version = "8"
    filter {
        name   = "display_name"
         # If you are deploying OKE on Compute Cloud@Customer, use the image listed below
        #values = ["uln-pca-Oracle-Linux8-OKE-1.28.3"]
         # If you are deploying OKE on Private Cloud Appliance, use the image listed below
         values = ["uln-pca-Oracle-Linux8-OKE-1.28.3-20240713.oci"]
        regex  = true
    }
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_cluster.tf

```
resource "oci_containerengine_cluster" "OKECluster" { # Adjust to name of your preference.
        compartment_id = var.compartment_id
        kubernetes_version = var.kubernetes_version
        name =  "${var.ClusterName}-1"
        vcn_id = "${oci_core_vcn.oke_vcn.id}"
        cluster_pod_network_options {
               cni_type = "FLANNEL_OVERLAY"
        }
        endpoint_config {
                 subnet_id = "${oci_core_subnet.kmi_lb.id}"
               is_public_ip_enabled = "true"
        }
        freeform_tags = {"Department"= "Finance"} # Adjust to the your correct Department.
        options {
                 kubernetes_network_config {
                       pods_cidr = "10.96.0.0/16"
                       services_cidr = "10.244.0.0/16"
                 }
                 service_lb_subnet_ids =  ["${oci_core_subnet.worker_lb.id}"]
       }
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_dyn_grp.tf

```
# Create a dynamic group
resource "oci_identity_dynamic_group" "oke-dynamic-grp" {
    compartment_id = "${var.tenancy_ocid}"
    description    = "OKE worker dynamic group"
    matching_rule  = "tag.${oci_identity_tag_namespace.oracle-pca.name}.${oci_identity_tag.cluster-id.name}.value"
    name           = "${var.oke_dyn_grp}"
    depends_on = [oci_identity_tag.cluster-id]
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_kmi_seclist.tf

```
# NOTE: KMI Load Balancer accepts traffic on 6443 and 50051(insecure)/50052(secure).
# An end user would adjust this seclist to only accept connections from
# where they expect the network to run, although 6443 would still need to
# accept connections from the cluster KMIs and worker instances.

resource "oci_core_default_security_list" "oke_vcn" {
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_security_list_id
  egress_security_rules {
    destination      = "0.0.0.0/0"
    destination_type = "CIDR_BLOCK"
    protocol         = "all"
  }
  dynamic "ingress_security_rules" {
    iterator = icmp_type
    for_each = [3, 8, 11]
    content {
      # ping from VCN; unreachable/TTL from anywhere
      source      = (icmp_type.value == "8" ? var.vcn_cidr : "0.0.0.0/0")
      source_type = "CIDR_BLOCK"
      protocol    = "1"
      icmp_options {
        type = icmp_type.value
      }
    }
  }
}
resource "oci_core_security_list" "kmilb" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  display_name = "${var.vcn_name}-kmilb"
  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_lb_ingress_rules
    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}
resource "oci_core_security_list" "kmi" {
  compartment_id = var.compartment_id
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
  vcn_id         = oci_core_vcn.oke_vcn.id
  display_name = "${var.vcn_name}-kmi"
  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_ingress_rules
    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.kmi_ingress_udp_rules
    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "17"
      udp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_kmi_subnet.tf

```
resource "oci_core_subnet" "kmi" {
  cidr_block                 = var.kmi_cidr
  compartment_id             = var.compartment_id
  display_name               = "control-plane"
  dns_label                  = "kmi"
  vcn_id                     = oci_core_vcn.oke_vcn.id
  prohibit_public_ip_on_vnic = true
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.kmi.id
  ]
}
resource "oci_core_subnet" "kmi_lb" {
  cidr_block                 = var.kmilb_cidr
  compartment_id             = var.compartment_id
  dns_label                  = "kmilb"
  vcn_id                     = oci_core_vcn.oke_vcn.id
  display_name               = "control-plane-endpoint"
  prohibit_public_ip_on_vnic = false
  route_table_id             = oci_core_route_table.public.id
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.kmilb.id
  ]
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_policy.tf

```
#Create policies
resource "oci_identity_policy" "oke-dyn-grp-policy" {
    compartment_id = "${var.tenancy_ocid}"
    description    = "Dynamic group policies for OKE Resources"
    name           = "${var.oke_policy_name}"
    statements     = [
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-grp.name} to manage load-balancers in tenancy",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-grp.name} to manage volume-family in tenancy",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-grp.name} to manage file-family in tenancy",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-grp.name} to use instance-family in tenancy",
        "allow dynamic-group ${oci_identity_dynamic_group.oke-dynamic-grp.name} to use virtual-network-family in tenancy"
    ]
    depends_on = [oci_identity_dynamic_group.oke-dynamic-grp]
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_pool.tf

```
resource "oci_containerengine_node_pool" "OKENodePool" { # Adjust to name of your preference.
  cluster_id         = oci_containerengine_cluster.OKECluster.id # Adjust to the same name entered on oke_cluster.tf.
  compartment_id     = var.compartment_id
  kubernetes_version = var.kubernetes_version
  name               = "OKENodePool" # Adjust to the same entered on resource (1st line of this script).
  node_shape         = var.Shape
  freeform_tags = {"Department"= "Finance"} # Adjust to the same tag name entered on oke_cluster.tf script.
  node_source_details {
     image_id =  data.oci_core_images.image1.images[0]["id"]
     source_type = "IMAGE"
  }
  node_shape_config {
    ocpus = 1
    memory_in_gbs = 10
  }
  node_config_details {
    size      = var.node_pool_size
    freeform_tags = {"Department"= "Finance"} # Adjust to the same tag name entered on oke_cluster.tf script.
    placement_configs {
      availability_domain = "AD-1"
      subnet_id           =  "${oci_core_subnet.worker.id}"
     }
  }
  ssh_public_key      = file(var.public_key_oci)
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_tags_grp.tf

```
# Tag Namespace and key definition creation
resource "oci_identity_tag" "cluster-id" {
    description      = "Default tag key definition"
    name             = "cluster_id"
    tag_namespace_id = "${oci_identity_tag_namespace.oracle-pca.id}"
    depends_on = [oci_identity_tag_namespace.oracle-pca]
}
resource "oci_identity_tag_namespace" "oracle-pca" {
    compartment_id = "${var.tenancy_ocid}"
    description    = "Default Tag namespace for Oracle PCA/C3 OKE"
    name           = "OraclePCA-OKE"
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_vcn.tf

```
resource "oci_core_vcn" "oke_vcn" {
  cidr_block     = var.vcn_cidr
  dns_label      = var.vcn_name
  compartment_id = var.compartment_id
  display_name   = "${var.vcn_name}-vcn"
}
resource "oci_core_nat_gateway" "vcn_ngs" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "VCN nat g6s"
}
resource "oci_core_internet_gateway" "vcn_igs" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id

  display_name = "VCN i6t g6s"
  enabled      = true
}
resource "oci_core_default_route_table" "private" {
  manage_default_resource_id = oci_core_vcn.oke_vcn.default_route_table_id
  display_name               = "Default - private"

  route_rules {
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = oci_core_nat_gateway.vcn_ngs.id
  }
}
resource "oci_core_route_table" "public" {
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  display_name = "public"
  route_rules {
    destination       = "0.0.0.0/0"
    destination_type  = "CIDR_BLOCK"
    network_entity_id = oci_core_internet_gateway.vcn_igs.id
  }
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_worker_seclist.tf

```
resource "oci_core_security_list" "workerlb" {
  display_name   = "${var.vcn_name}-workerlb"
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_lb_ingress_rules
    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
}

resource "oci_core_security_list" "worker" {
  display_name   = "${var.vcn_name}-worker"
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_ingress_rules
    content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "6"
      tcp_options {
        min = port.value.port_min
        max = port.value.port_max
      }
    }
  }
  dynamic "ingress_security_rules" {
    iterator = port
    for_each = local.worker_ingress_udp_rules
content {
      source      = port.value.source
      source_type = "CIDR_BLOCK"
      protocol    = "17"
      udp_options {
        min = port.value.port_min
        max = port.value.port_max
```


Copyright © 2024, Oracle and/or its affiliates / Public

```
      }
    }
  }
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

oke_worker_subnet.tf

```
resource "oci_core_subnet" "worker" {
  cidr_block     = var.worker_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  display_name               = "worker"
  dns_label                  = "worker"
  prohibit_public_ip_on_vnic = true
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.worker.id
  ]
}
resource "oci_core_subnet" "worker_lb" {
  cidr_block     = var.workerlb_cidr
  compartment_id = var.compartment_id
  vcn_id         = oci_core_vcn.oke_vcn.id
  display_name               = "service-lb"
  dns_label                  = "servicelb"
  prohibit_public_ip_on_vnic = false
  route_table_id             = oci_core_route_table.public.id
  security_list_ids = [
    oci_core_default_security_list.oke_vcn.id,
    oci_core_security_list.workerlb.id
  ]
}
```


Copyright © 2024, Oracle and/or its affiliates / Public

ORACLE

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Copyright © 2024, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Author: Anderson Souza, Amardeep Dhillon