Terraform

Oracle Cloud
Abstract

• Demo
  • Create instance in Oracle Public Cloud
  • Import instance from Oracle Public Cloud

• Comparison of popular infrastructure-as-code tools
Create instance in Oracle Public Cloud
A Compute Classic instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. Learn more.

No instances available matching the filters.
Terraform scripts

- `terraform.tfvars`
- `variables.tf`
- `main.tf`
- `remote-exec.sh`

- Files to upload:
  - SSH key
  - Docker image
  - `dc.yml`
terraform.tfvars

user = "username"
password = "P4$$w0rd"
identity_domain = "domain"
endpoint = "https://myendpoint.oraclecloud.com/"

https://www.terraform.io/docs/providers/opc/
variable "image" {
    default = "bitnami-lampstack-7.1.16-0-linux-ubuntu-16.04-x86_64"
}

variable "shape" {
    default = "oc3"
}

variable "ssh_key_file" {
    default = "c:\\Terraform\\demo\\ssh-keys\\terraform-key.pem"
}

variable "ssh_user_name" {
    default = "bitnami"
}
variable "remote-exec_path" {
    default = "c:\\Terraform\\demo\\oracle_ubuntu_instance_create\\remote-exec.sh"
}

variable "upload_dir" {
    default = "c:\\Terraform\\demo\\Upload"
}
main.tf

• Provider opc
• Resources
  • opc_compute_ssh_key
  • opc_compute_ip_reservation
  • opc_compute_security_list
  • opc_compute_sec_rule
  • opc_compute_instance
• Provisioner
  • file
  • remote exec
variable user {}
variable password {}
variable identity_domain {}
variable endpoint {}

provider "opc" {
    user = "${var.user}"
    password = "${var.password}"
    identity_domain = "${var.identity_domain}"
    endpoint = "${var.endpoint}"
}

https://www.terraform.io/docs/providers/opc/
resource "opc_compute_ssh_key" "sshkey" {
  name = "sshkey"
  key = "${file("${var.ssh_key_file}.pub")}"
  enabled = true
}

resource "opc_compute_ip_reservation" "ipreservation1" {
  parent_pool = "/oracle/public/ippool"
  permanent = true
}
resource "opc_compute_security_list" "terraform_sec_list" {
    name = "terraform-sec-list"
    policy = "DENY"
    outbound_cidr_policy = "PERMIT"
}
resource "opc_compute_sec_rule" "ssh_rule" {
  name = "ssh_rule"
  source_list = "seciplist:/oracle/public/public/public-internet"
  destination_list = "seclist:${opc_compute_security_list.terraform_sec_list.name}"
  action = "permit"
  application = "/oracle/public/ssh"
}

resource "opc_compute_sec_rule" "https_rule" {
  name = "https_rule"
  source_list = "seciplist:/oracle/public/public/public-internet"
  destination_list = "seclist:${opc_compute_security_list.terraform_sec_list.name}"
  action = "permit"
  application = "/oracle/public/https"
}

https://www.terraform.io/docs/providers/opc/r/opc_compute_sec_rule.html
resource "opc_compute_instance" "terraform-instance1" {
  name = "terraform-instance1"
  label = "${var.image}"
  image_list = "/Compute-${var.domain}/${var.user}/${var.image}"
  shape = "${var.shape}"
  ssh_keys = [ "${opc_compute_ssh_key.sshkey.name}" ]

  networking_info {
    index = 0
    shared_network = true
    nat = [ "${opc_compute_ip_reservation.ipreservation1.name}" ]
    sec_lists = [ "${opc_compute_security_list.terraform_sec_list.name}" ]
  }
}

https://www.terraform.io/docs/providers/opc/r/opc_compute_instance.html
connection {
  type = "ssh"
  user = "${var.ssh_user_name}"
  host = "${opc_compute_ip_reservation.ipreservation1.ip}"
  private_key = "${file(var.ssh_key_file)}"
  timeout = "10m"
  agent = false
}

https://www.terraform.io/docs/provisioners/connection.html
provisioner "file" {
  source = "${var.remote-exec_path}"
  destination = "'/home/${var.ssh_user_name}/remote-exec.sh"
}

provisioner "file" {
  source = "${var.upload_dir}/energy"
  destination = "'/home/${var.ssh_user_name}/energy"
}

https://www.terraform.io/docs/provisioners/file.html
provisioner "remote-exec" {
  inline = [ "chmod 775 /home/${var.ssh_user_name}/remote-exec.sh",
            "sudo /home/${var.ssh_user_name}/remote-exec.sh",
            "mkdir deconv && mkdir deconv/appconf && mkdir deconv/conf && mkdir deconv/data && mkdir deconv/logs && mkdir deconv/tmp",
        ]
}

https://www.terraform.io/docs/provisioners/remote-exec.html
remote-exec.sh

echo "127.0.0.1 terraform-instance1" >> /etc/hosts

apt-get update
apt-get -y install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"

apt-get update
apt-get -y install docker-ce -y

apt-get -y install python-pip
pip install docker-compose
provisioner "file" {
    source = "${var.upload_dir}\dc.yml"
    destination = "/home/${var.ssh_user_name}/deconv/conf/dc.yml"
}

provisioner "remote-exec" {
    inline = [   "sudo docker load -i energy",
                  "sudo /opt/${var.ssh_user_name}/apache2/bin/apachectl stop",
                  "sudo docker-compose -p energy-deconv -f ./deconv/conf/dc.yml up -d",
    ]
}
}
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Minden jog fenntartva.
c:\Users\magyar1zsan604\Documents\Terraform\demo\OraclePublicCloud\oracle_ubuntu_instance_create>set http_proxy=mhiproxy.telekom.intra:3128
c:\Users\magyar1zsan604\Documents\Terraform\demo\OraclePublicCloud\oracle_ubuntu_instance_create>set https_proxy=mhiproxy.telekom.intra:3128
## Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>OCPUs</th>
<th>Memory</th>
<th>Volumes</th>
<th>Public IP</th>
<th>Private IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>terraform-instance1</td>
<td>Running</td>
<td>1</td>
<td>7.5 GB</td>
<td></td>
<td>144.21.73.55</td>
<td>10.29.183.14</td>
</tr>
</tbody>
</table>

A Compute Classic instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. Learn more.
Terraform – Resource Graph - Visualization

• Terraform builds dependency graphs for planning state management and more.

• [http://www.graphviz.org/Download..php](http://www.graphviz.org/Download..php)

• `$ terraform graph | dot -Tpng > tgraph1.png`
Import from Oracle Public Cloud
Terraform import scripts

• terraform.tfvars
• import.tf
• import_opc.bat
```hcl
user = "username"
password = "P4$$w0rd"
identity_domain = "domain"
endpoint = "https://myendpoint.oraclecloud.com/"
```

https://www.terraform.io/docs/providers/opc/
import.tf

variable user {}
variable password {}
variable identity_domain {}
variable endpoint {}

provider "opc" {
  user = "${var.user}"
  password = "${var.password}"
  identity_domain = "${var.identity_domain}"
  endpoint = "${var.endpoint}"
}

https://www.terraform.io/docs/providers/opc/
import.tf

resource "opc_compute_ssh_key" "sshkey" {
}
resource "opc_compute_ip_reservation" "ipreservation1" {
}
resource "opc_compute_image_list" "bitnami-lampstack" {
}
resource "opc_compute_security_list" "terraform-sec-list" {
}
resource "opc_compute_sec_rule" "ssh_rule" {
}
resource "opc_compute_sec_rule" "https_rule" {
}
resource "opc_compute_instance" "terraform-instance1" {
}
### Summary

<table>
<thead>
<tr>
<th>Instances</th>
<th>OCPUs</th>
<th>Memory</th>
<th>Volume Size in Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>7.5 GB</td>
<td>-</td>
</tr>
</tbody>
</table>

### Instances

A Compute Classic instance is a virtual machine running a specific operating system, with the CPU and memory resources that you specify. Learn more.

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>OCPUs</th>
<th>Memory</th>
<th>Volumes</th>
<th>Public IP</th>
<th>Private IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>terraform-instance1</td>
<td>Running</td>
<td>1</td>
<td>7.5 GB</td>
<td></td>
<td>144.21.73.55</td>
<td>10.29.183.14</td>
</tr>
</tbody>
</table>
import_opc.bat

terraform init

terraform import opc_compute_ssh_key.sshkey sshkey

terraform import opc_compute_ip_reservation.ipreservation1 6f29a2d5-7858-41c7-abe5-c4efb6c53f3d

terraform import opc_compute_image_list.bitnami-lampstack bitnami-lampstack-7.0.23-0-linux-ubuntu-14.04-x86_64
import_opc.bat

```bash
terraform import opc_compute_security_list.terraform-sec-list terraform-sec-list

terraform import opc_compute_sec_rule.terraform-ssh_rule terraform-ssh_rule

terraform import opc_compute_sec_rule.terraform-https_rule terraform-https_rule

terraform import opc_compute_instance.terraform-instance1 terraform-instance1/9496061d-7279-47b4-b334-4c4f0ef0b2c3
```
Configuration Mgmt. vs. Orchestration

Chef, Puppet, Ansible, and SaltStack
• designed to install and manage software on existing servers.

CloudFormation and Terraform
• designed to provision the servers themselves, leaving the job of configuring those servers to other tools.
• With Docker and Packer, you can create images that have all the software your server needs already installed and configured.
Mutable vs. Immutable

Configuration management tools
• software update in-place.
• each server builds up a unique history of changes.
• configuration drift.

Orchestration tools
• every “change” is actually a deployment of a new server.
• reduces the likelihood of configuration drift bugs
Procedural vs. Declarative

Chef and Ansible
• encourage a procedural style where you write code that specifies, step-by-step, how to achieve some desired end state

Terraform, CloudFormation, SaltStack, and Puppet
• all encourage a more declarative style where you write code that specifies your desired end state, and the IAC tool itself is responsible for figuring out how to achieve that state
Client/Server vs. Client-Only

Chef, Puppet, and SaltStack
• the server talks to agents
• extra software instalation on every one of your servers.
• extra server just for configuration management.
• maintain it, upgrade it, make backups of it, monitor it, and restore it

CloudFormation, Ansible, and Terraform
• Terraform uses cloud provider APIs to provision infrastructure:
  • so there are no new authentication mechanisms beyond what you’re using with the cloud provider already,
  • there is no need for direct access to your servers.
<table>
<thead>
<tr>
<th>Code</th>
<th>Terraform</th>
<th>Chef</th>
<th>Ansible</th>
<th>Puppet</th>
<th>SaltStack</th>
<th>CloudFormation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All</td>
<td>All AWS only</td>
</tr>
<tr>
<td>Type</td>
<td>Orchestration</td>
<td>Config Mgmt</td>
<td>Config Mgmt</td>
<td>Config Mgmt</td>
<td>Config Mgmt</td>
<td>Orchestration</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>Immutable</td>
<td>Mutable</td>
<td>Mutable</td>
<td>Mutable</td>
<td>Mutable</td>
<td>Immutable</td>
</tr>
<tr>
<td>Language</td>
<td>Declarative</td>
<td>Procedural</td>
<td>Procedural</td>
<td>Declarative</td>
<td>Declarative</td>
<td>Declarative</td>
</tr>
<tr>
<td>Architecture</td>
<td>Client-Only</td>
<td>Client/Server</td>
<td>Client-Only</td>
<td>Client/Server</td>
<td>Client/Server</td>
<td>Client-Only</td>
</tr>
</tbody>
</table>

https://blog.gruntwork.io/why-we-use-terraform-and-not-chef-puppet-ansible-saltstack-or-cloudformation-7989dad2865c
Questions

https://144.21.72.65/DataExchangeConverter/
thank you