TERRAFORM TUTORIAL - MODULES

Terminology

1. **module**:
   A Terraform module is a set of Terraform configuration files in a single directory. Even a simple

Ph.D. / Golden Gate Ave, San Francisco / Seoul National Univ / Carnegie Mellon / UC Berkeley / DevOps / Deep Learning / Visualization

2. **root module**:
   When we run Terraform commands directly from a directory, it is considered the **root module**.

3. **child module**:
   A module that is called by another configuration is referred to as a **child module**.

4. **calling module**:
   Terraform commands will only directly use the configuration files in one directory, which is usually the current working directory. However, our configuration can use module blocks to call modules in other directories. When Terraform encounters a module block, it loads and processes that module's configuration files.

5. **source** argument:
   When calling a module, the source argument is required. Terraform may search for a module in the Terraform registry that matches the given string. We could also use a URL or local file path for the source of our modules.

6. `terraform init`:
   When using a new module for the first time, we must run "terraform init" to install the module. When the command is run, Terraform will install any new modules in the **.terraform/modules** directory within our configuration's working directory. For local modules, Terraform will create a symlink to the module's directory.

```
$ terraform init
Initializing modules...

Initializing the backend...

Initializing provider plugins...
```

Then, our **.terraform/modules** directory will look something like this:

```
.terraform/modules
├── ec2_instances
├── modules.json
└── vpc
```

where our root **main.f** looks like this:

```
...
module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  ...
}

module "ec2_instances" {
  source  = "terraform-aws-modules/ec2-instance/aws"
  ...
}
```

7. **terraform.tfstate/terraform.tfstate.backup**:
   These files contain our Terraform state, and are how Terraform keeps track of the relationship between our configuration and the infrastructure provisioned by it

8. **.terraform**:
   This directory contains the modules and plugins used to provision our infrastructure.

9. **provider block**:
   We don't need provider block in module configuration. When Terraform processes a module block, it will inherit the provider from the enclosing configuration. Because of this, including provider blocks in modules is not recommended.

10. **module outputs**: Which values to add as outputs?
    Because outputs are the only supported way for users to get information about resources configured by the module. We need to add outputs to our module in the **outputs.tf** file inside the module directory.
    A module's outputs can be accessed as read-only attributes on the module object, which is available within the configuration that calls the module. We can reference these outputs in expressions as module.<MODULE NAME>.<OUTPUT NAME>.

modules.php)

Terraform Tutorial - Creating AWS S3 bucket / SQS queue resources and notifying bucket event to queue (/DevOps/Terraform/Terraform-Introduction-AWS-S3-SQS.php)

Terraform Tutorial - AWS ASG and Modules (/DevOps/Terraform/Terraform-Introduction-AWS-ASG-Modules.php)

Terraform Tutorial - VPC, Subnets, RouteTable, ELB, Security Group, and Apache server I (/DevOps/Terraform/Terraform-VPC-Subnet-ELB-RouteTable-SecurityGroup-Apache-Server-1.php)

Terraform Tutorial - VPC, Subnets, RouteTable, ELB, Security Group, and Apache server II (/DevOps/Terraform/Terraform-VPC-Subnet-ELB-RouteTable-SecurityGroup-Apache-Server-2.php)

Modules

In this post, we're going to go over how to use Modules to organize Terraform-managed infrastructure. In addition to that we'll learn how a child module exposes resource to a parent module via `terraform output`.

We're using Terraform 12:

```
$ terraform -v
Terraform v0.12.28
+ provider.aws v3.34.0
```

Up to this point, we've been configuring Terraform by editing Terraform configurations directly. As our infrastructure grows, this practice has a few key problems: a lack of organization, a lack of reusability, and difficulties in management for teams.

Modules are used to create reusable components, improve organization, and to treat pieces of infrastructure as a black box.

Ref - https://www.terraform.io/intro/getting-started/modules.html (https://www.terraform.io/intro/getting-started/modules.html)

Here are the files we'll use in this post:

```
├── main.tf
├── my_modules
│   └── instance
│       ├── main.tf
│       ├── output.tf
│       └── variables.tf
├── output.tf
└── variables.tf
```

Basically, the **main.tf** file will be using another terraform file in **my-modules/instance/main.f**. For each of the **main.tf** has its own variables defined in **variables.tf**. Here are the files:

./main.tf:

```
provider "aws" {
        region = var.aws_region
}

module "my_instance_module" {
        source = "./my_modules/instance"
        ami = "ami-04169656fea786776"
        instance_type = "t2.nano"
        instance_name = "myvm01"
}
```

./variables.tf:

```
variable "aws_region" {
  description = "AWS region"
  type        = string
  default     = "us-east-1"
}
```


Terraform Tutorial - Docker nginx container with ALB and dynamic autoscaling (/DevOps/Terraform/Terraform-docker-nginx-alb-dynamic-autoscaling.php)

Terraform Tutorial - AWS ECS using Fargate : Part I (/DevOps/Terraform/Terraform-ECS-1.php)

Hashicorp Vault (/DevOps/Terraform/Hashicorp-Vault.php)

HashiCorp Vault Agent (/DevOps/Terraform/Hashicorp-Vault-agent.php)

HashiCorp Vault and Consul on AWS with Terraform (/DevOps/Terraform/Hashicorp-

```
output "instance_ip_addr" {
  value = module.my_instance_module.instance_ip_addr
  description = "The public IP address of the main instance."
}
```

Note that we're accessing output values by referring to **module**.**MODULE NAME**.**OUTPUT NAME**.

The files in the module are the following.

./my_modules/instance/main.tf:

```
resource "aws_instance" "my_instance" {
        ami = var.ami
        instance_type = var.instance_type
        key_name = var.key_name
        tags = {
                Name = var.instance_name
        }
}
```

./my_modules/instance/variables.tf:

```
variable "ami" {
  type          = string
  default       = "ami-04169656fea786776"
}

variable "instance_type" {
  type          = string
  default       = "t2-nano"
}

variable "instance_name" {
  description   = "Value of the Name tag for the EC2 instance"
  type          = string
  default       = "ExampleInstance"
}

variable "key_name" {
  type          = string
  default       = "einsteinish"
}
```

./my_modules/instance/output.tf:

Terraform-null_resource-local-exec-remote-exec-triggers.php)

AWS IAM user, group, role, and policies - part 1 (/DevOps/Terraform/Terraform_I

AWS IAM user, group, role, and policies - part 2 (/DevOps/Terraform/Terraform_I

Delegate Access Across AWS Accounts Using IAM Roles (/DevOps/Terraform/Terraform_/

AWS KMS (/DevOps/Terraform/Terraform-AWS-KMS.php)

Terraform import (/DevOps/Terraform/Terraform_I

Terraform commands cheat sheet (/DevOps/Terraform/Terraform_(

Terraform Cloud (/DevOps/Terraform/Terraform-Cloud.php)

Terraform 14 (/DevOps/Terraform/Terraform1

Creating Private TLS Certs (/DevOps/Terraform/Terraform-private-tls-certs.php)

Now that we are ready, just run terraform commands within the project root directory where the top-level **main.tf** is located:

```
$ terraform init

$ terraform validate
Success! The configuration is valid.

$ terraform plan
...
Plan: 1 to add, 0 to change, 0 to destroy.
...

$ terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # module.my_instance_module.aws_instance.my_instance will be created
  + resource "aws_instance" "my_instance" {
      + ami                          = "ami-04169656fea786776"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = (known after apply)
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.nano"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = "einsteinish"
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + secondary_private_ips        = (known after apply)
      + security_groups              = (known after apply)
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name" = "my_instance_001"
        }
      + tenancy                      = (known after apply)
      + vpc_security_group_ids       = (known after apply)

      + ebs_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
```

Linux - General, shell programming, processes & signals ... (/Linux/linux_tips1.php)

RabbitMQ... (/python/RabbitMQ_Celery/pythc

MariaDB (/DevOps/DevOps_MariaDB.php)

New Relic APM with NodeJS : simple agent setup on AWS instance (/DevOps/DevOps_NewRelic-APM-Application-Performance-Management-setup.php)

Nagios on CentOS 7 with Nagios Remote Plugin Executor (NRPE) (/DevOps/DevOps_CentOS_Nagic Remote-Plugin-Executor-NRPE.php)

Nagios - The industry standard in IT infrastructure monitoring on Ubuntu (/DevOps/DevOps_Nagios_Infrast Remote-Plugin-Executor-NRPE.php)

Zabbix 3 install on Ubuntu 14.04 & adding hosts / items / graphs (/DevOps/DevOps-Zabbix3-Server-and-Agent-Install-Ubuntu14-Adding-Hosts-Items-Graphs.php)

Datadog - Monitoring with PagerDuty/HipChat and APM (/DevOps/DevOps-Monitoring-with-Datadog-PagerDuty-HipChat.php)

Install and Configure Mesos Cluster (/DevOps/DevOps_Mesos_Install.

```
          + volume_type           = (known after apply)
        }

      + enclave_options {
          + enabled = (known after apply)
        }

      + ephemeral_block_device {
          + device_name  = (known after apply)
          + no_device    = (known after apply)
          + virtual_name = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + http_tokens                 = (known after apply)
        }

      + network_interface {
          + delete_on_termination = (known after apply)
          + device_index          = (known after apply)
          + network_interface_id  = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

module.my_instance_module.aws_instance.my_instance: Creating...
module.my_instance_module.aws_instance.my_instance: Still creating... [10s elapsed]
module.my_instance_module.aws_instance.my_instance: Still creating... [20s elapsed]
module.my_instance_module.aws_instance.my_instance: Creation complete after 26s [id=i-062ffe

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

instance_ip_addr = [
  "54.175.174.137",
]
```

Container Orchestration : Docker Swarm vs Kubernetes vs Apache Mesos (/DevOps/DevOps-Docker-Swarm-vs-Kubernetes-vs-Apache-Mesos.php)

OpenStack install on Ubuntu 16.04 server - DevStack (/DevOps/OpenStack-Install-On-Ubuntu-16-Server.php)

AWS EC2 Container Service (ECS) & EC2 Container Registry (ECR) | Docker Registry (/DevOps/DevOps-ECS-ECR.php)

CI/CD with CircleCI - Heroku deploy (/DevOps/DevOps-CircleCI-Heroku-Deploy.php)

Introduction to Terraform with AWS elb & nginx (/DevOps/DevOps-Terraform.php)

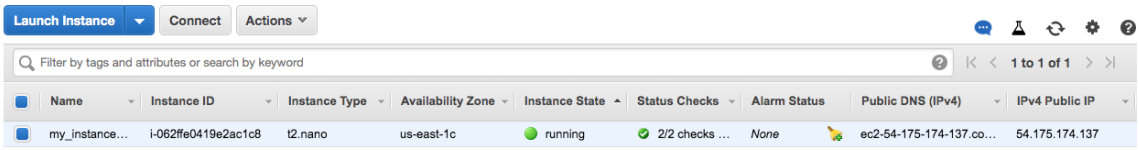
Docker & Kubernetes (/DevOps/Docker/Docker_Kuberr

Kubernetes I - Running Kubernetes Locally via Minikube (/DevOps/DevOps-Kubernetes-1-Running-Kubernetes-Locally-via-Minikube.php)

Kubernetes II - kops on AWS (/DevOps/DevOps-Kubernetes-II-kops-on-AWS.php)

Kubernetes III - kubeadm on AWS (/DevOps/DevOps-Kubernetes-III-Kubernetes-on-Linux-with-kubeadm.php)

AWS : EKS (Elastic Container Service for Kubernetes) (/DevOps/AWS/aws-EKS-Elastic-



We can confirm it from AWS cli:

```
$ aws ec2 describe-instances
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-04169656fea786776",
                    "InstanceId": "i-062ffe0419e2ac1c8",
                    "InstanceType": "t2.nano",
                    "KeyName": "einsteinish",
                    "LaunchTime": "2021-03-31T04:20:47+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1c",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-41-218.ec2.internal",
                    "PrivateIpAddress": "172.31.41.218",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-54-175-174-137.compute-1.amazonaws.com",
                    "PublicIpAddress": "54.175.174.137",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-66d25311",
                    "VpcId": "vpc-4c600529",
                    "Architecture": "x86_64",
                    ...
```

ssh into the instance:

DEVOPS / SYS ADMIN Q & A

(1A) - Linux Commands (/DevOps/DevOps-Sys-Admin-Interview-Questions-Commands.php)

(1B) - Linux Commands (/DevOps/DevOps-Sys-Admin-Interview-Questions-Commands-2.php)

(2) - Networks (/DevOps/DevOps-Sys-Admin-Interview-Questions-Networks.php)

(2B) - Networks (/DevOps/DevOps-Sys-Admin-Interview-Questions-Networks-2.php)

(3) - Linux Systems (/DevOps/DevOps-Sys-Admin-Interview-Questions-Systems.php)

(4) - Scripting (Ruby/Shell) (/DevOps/DevOps-Sys-Admin-Interview-Questions-Scripting.php)

(5) - Configuration Management (/DevOps/DevOps-Sys-Admin-Interview-Questions-Configurations.php)

(6) - AWS VPC setup (public/private subnets with NAT) (/DevOps/DevOps-Sys-Admin-Interview-Questions-AWS-VPC-Setup.php)

(6B) - AWS VPC Peering

```
$ ssh -i ~/.ssh/einsteinish.pem ubuntu@54.175.174.137
The authenticity of host '54.175.174.137 (54.175.174.137)' can't be established.
ECDSA key fingerprint is SHA256:DZ7zjNX1Ab6cZt/PpCzCYH4dLiNKNGB/To64jNkDyLM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.175.174.137' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo ".
See "man sudo_root" for details.

ubuntu@ip-172-31-41-218:~$
```

Clean up:

```
$ terraform destroy
...
Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

module.my_instance_module.aws_instance.my_instance: Destroying... [id=i-062ffe0419e2ac1c8]
module.my_instance_module.aws_instance.my_instance: Still destroying... [id=i-062ffe0419e2ac
module.my_instance_module.aws_instance.my_instance: Still destroying... [id=i-062ffe0419e2ac
module.my_instance_module.aws_instance.my_instance: Still destroying... [id=i-062ffe0419e2ac
module.my_instance_module.aws_instance.my_instance: Destruction complete after 32s

Destroy complete! Resources: 1 destroyed.
```

(/DevOps/DevOps-Sys-Admin-Interview-Questions-Web-HTTP.php)

(8) - Database (/DevOps/DevOps-Sys-Admin-Interview-Questions-Database.php)

(9) - Linux System / Application Monitoring, Performance Tuning, Profiling Methods & Tools (/DevOps/DevOps-Sys-Admin-Interview-Questions-Linux-Monitoring-System-Application-Performance-Tuning-Tools.php)

(10) - Trouble Shooting: Load, Throughput, Response time and Leaks (/DevOps/DevOps-Sys-Admin-Interview-Questions-Trouble-Shooting-Slow-Application-Performance-BottleNecks-Leaks.php)

(11) - SSH key pairs, SSL Certificate, and SSL Handshake (/DevOps/DevOps-Sys-Admin-Interview-Questions-SSH-Connection-SSL-Certificates.php)

(12) - Why is the database slow? (/DevOps/DevOps-Sys-Admin-Interview-Questions-Why-is-database-slow.php)

(13) - Is my web site down? (/DevOps/DevOps-Sys-Admin-Interview-Questions-Is-Website-down.php)

(14) - Is my server down? (/DevOps/DevOps-Sys-Admin-Interview-Questions-Is-Server-down.php)

(15) - Why is the server

Terraform output

In this post, the `terraform output` s were put intentionally to demonstrate how the child module **exposes** its resource to a parent module.

Let's look into how our module is defined in **./main.tf**:

```
module "my_instance_module" {
        source = "./my_modules/instance"
        ami = "ami-04169656fea786776"
        instance_type = "t2.nano"
        instance_name = "myvm01"
}
```

It named as **my_instance_module**.

The public ip address is defined in that module, **./my_modules/instance/output.tf**:

```
output "instance_ip_addr" {
  value = aws_instance.my_instance.*.public_ip
  description = "The public IP address of the main instance."
}
```

Here we're defining an output value named **instance_ip_addr** containing the IP address of an EC2 instance that our module created.

How can we access the resource (public_ip) from the root **output.tf**?

References:

1. Modules Overview (https://learn.hashicorp.com/tutorials/terraform/module?in=terraform/modules)

Creating a submodule - static S3 website

In this section, we will create a local submodule within our existing configuration that uses the s3 bucket resource from the AWS provider:

(24) - User account management (/DevOps/DevOps-Sys-Admin-Interview-Questions-Linux-User-Account-Management.php)

(25) - Domain Name System (DNS) (/DevOps/DevOps-Sys-Admin-Interview-Questions-DNS.php)

(26) - NGINX SSL/TLS, Caching, and Session (/DevOps/DevOps-Sys-Admin-Interview-Questions-NGINX-SSL-TLS-Caching-Session.php)

(27) - Troubleshooting 5xx server errors (/DevOps/DevOps-Sys-Admin-Interview-Questions-5xx-http-server-errors.php)

(28) - Linux Systemd: journalctl (/DevOps/DevOps-Sys-Admin-Interview-Questions-Linux-Systemd-Journalctl.php)

(29) - Linux Systemd: FirewallD (/DevOps/DevOps-Sys-Admin-Interview-Questions-Linux-Systemd-Firewalld.php)

(30) - Linux: SELinux (/DevOps/DevOps-Sys-Admin-Interview-Questions-Linux-SELinux.php)

(31) - Linux: Samba (/DevOps/DevOps-Sys-Admin-Interview-Questions-Linux-Samba.php)

(0) - Linux Sys Admin's Day to Day tasks (/DevOps/DevOps-Sys-Admin-Interview-Questions-Day-To-Day-

```
├── LICENSE
├── README.md
├── main.tf
├── modules
│   └── aws-s3-static-website-bucket
│       ├── LICENSE
│       ├── README.md
│       ├── main.tf
│       ├── outputs.tf
│       ├── variables.tf
│       └── www
│           ├── error.html
│           └── index.html
├── outputs.tf
└── variables.tf
```

Here are the files.

main.tf:

```
# Terraform configuration

provider "aws" {
  region = "us-east-1"
}

module "website_s3_bucket" {
  source = "./modules/aws-s3-static-website-bucket"

  bucket_name = "bogo-test-april-19-2021"

  tags = {
    Terraform   = "true"
    Environment = "dev"
  }
}
```

outputs.tf:

```
output "website_bucket_arn" {
  description = "ARN of the bucket"
  value       = module.website_s3_bucket.arn
}

output "website_bucket_name" {
  description = "Name (id) of the bucket"
  value       = module.website_s3_bucket.name
}

output "website_bucket_domain" {
  description = "Domain name of the bucket"
  value       = module.website_s3_bucket.domain
}
```

modules/aws-s3-static-website-bucket/main.tf:

```
resource "aws_s3_bucket" "s3_bucket" {
  bucket = var.bucket_name

  acl    = "public-read"
  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::${var.bucket_name}/*"
            ]
        }
    ]
}
EOF

  website {
    index_document = "index.html"
    error_document = "error.html"
  }

  tags = var.tags
}
```

modules/aws-s3-static-website-bucket/outputs.tf:

& adding keys to authorized_keys (/DevOps/Ansible/Ansible-aws-creating-ec2-instance.php)

AWS : Auto Scaling via AMI (/DevOps/Ansible/Ansible-aws-AutoScaling.php)

AWS : creating an ELB & registers an EC2 instance from the ELB (/DevOps/Ansible/Ansible-aws-creating-elb-and-register-ec2-instance.php)

Deploying Wordpress micro-services with Docker containers on Vagrant box via Ansible (/DevOps/Ansible/Docker-WordPress-Microservices-with-Nginx-reverse-proxy-Varnish-Mysql-Deployed-via-Ansible.php)

Setting up Apache web server (/DevOps/Ansible/Ansible_Setting

Deploying a Go app to Minikube (/DevOps/Ansible/Ansible-Deploying-a-Go-App-to-Minikube.php)

Ansible with Terraform (/DevOps/Ansible/Ansible-Terraform-null_resource-local-exec-remote-exec-triggers.php)

Jenkins

Install

```
# Output variable definitions

output "arn" {
  description = "ARN of the bucket"
  value       = aws_s3_bucket.s3_bucket.arn
}

output "name" {
  description = "Name (id) of the bucket"
  value       = aws_s3_bucket.s3_bucket.id
}

output "domain" {
  description = "Domain name of the bucket"
  value       = aws_s3_bucket.s3_bucket.website_domain
}
```

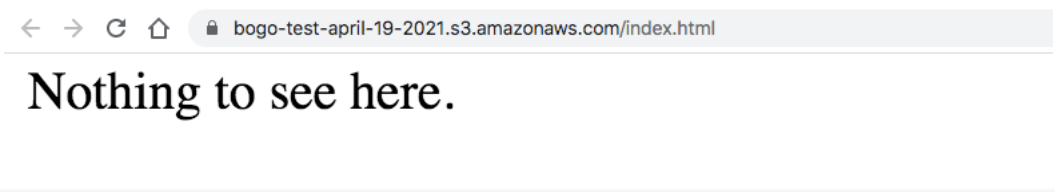
modules/aws-s3-static-website-bucket/variables.tf:

```
# Input variable definitions

variable "bucket_name" {
  description = "Name of the s3 bucket. Must be unique."
  type        = string
}

variable "tags" {
  description = "Tags to set on the bucket."
  type        = map(string)
  default     = {}
}
```

Next, run `terraform apply` and then upload the files in www/ folder to S3:

```
$ aws s3 cp modules/aws-s3-static-website-bucket/www/ s3://$(terraform output -raw website_b
```

Then, we can access the app via **https://bogo-test-april-19-2021.s3.amazonaws.com/index.html**:



Terraform Registry

The Terraform Registry (https://registry.terraform.io/) includes a directory of ready-to-use modules for various common purposes, which can serve as larger building-blocks for our infrastructure.

Here are almost identical code as in the previous section. This new code is using modules from the Terraform registry.

```
├── LICENSE
├── README.md
├── main.tf
├── modules
│   └── aws-s3-static-website-bucket
│       ├── LICENSE
│       ├── README.md
│       ├── main.tf
│       ├── outputs.tf
│       ├── variables.tf
│       └── www
│           ├── error.html
│           └── index.html
├── outputs.tf
└── variables.tf
```

Here are the files.

main.tf:

```
# Terraform configuration

provider "aws" {
  region = "us-east-1"
}

module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "2.21.0"

  name = var.vpc_name
  cidr = var.vpc_cidr

  azs             = var.vpc_azs
  private_subnets = var.vpc_private_subnets
  public_subnets  = var.vpc_public_subnets

  enable_nat_gateway = var.vpc_enable_nat_gateway

  tags = var.vpc_tags
}

module "ec2_instances" {
  source  = "terraform-aws-modules/ec2-instance/aws"
  version = "2.12.0"

  name           = "my-ec2-cluster"
  instance_count = 2

  ami                    = "ami-0c5204531f799e0c6"
  instance_type          = "t2.micro"
  vpc_security_group_ids = [module.vpc.default_security_group_id]
  subnet_id              = module.vpc.public_subnets[0]

  tags = {
    Terraform   = "true"
    Environment = "dev"
  }
}

module "website_s3_bucket" {
  source = "./modules/aws-s3-static-website-bucket"

  bucket_name = "bogo-test-april-19-2021"

  tags = {
    Terraform   = "true"
    Environment = "dev"
  }
}
```

outputs.tf:

(/DevOps/Puppet/puppet_amazo

Puppet Code Basics - Terminology (/DevOps/Puppet/puppet_basics_

Puppet with Amazon AWS on CentOS 7 (I) - Master setup on EC2 (/DevOps/Puppet/puppet_amazo

Puppet with Amazon AWS on CentOS 7 (II) - Configuring a Puppet Master Server with Passenger and Apache (/DevOps/Puppet/puppet_amazo

Puppet master /agent ubuntu 14.04 install on EC2 nodes (/DevOps/Puppet/puppet_install_

Puppet master post install tasks - master's names and certificates setup, (/DevOps/Puppet/puppet_master

Puppet agent post install tasks - configure agent, hostnames, and sign request (/DevOps/Puppet/puppet_agent_

EC2 Puppet master/agent basic tasks - main manifest with a file resource/module and immediate execution on an agent node (/DevOps/Puppet/puppet_basic_t

Setting up puppet master and agent with simple scripts on EC2 / remote install from desktop (/DevOps/Puppet/puppet_setting

EC2 Puppet - Install lamp with a manifest ('puppet apply') (/DevOps/Puppet/puppet_amazo

EC2 Puppet - Install lamp with a

```
# Output variable definitions

output "vpc_public_subnets" {
  description = "IDs of the VPC's public subnets"
  value       = module.vpc.public_subnets
}

output "ec2_instance_public_ips" {
  description = "Public IP addresses of EC2 instances"
  value       = module.ec2_instances.public_ip
}

output "website_bucket_arn" {
  description = "ARN of the bucket"
  value       = module.website_s3_bucket.arn
}

output "website_bucket_name" {
  description = "Name (id) of the bucket"
  value       = module.website_s3_bucket.name
}

output "website_bucket_domain" {
  description = "Domain name of the bucket"
  value       = module.website_s3_bucket.domain
}
```

variables.tf:

Puppet packages, services, and files (/DevOps/Puppet/puppet_packag

Puppet packages, services, and files II with nginx (/DevOps/Puppet/puppet_packag
Puppet templates (/DevOps/Puppet/puppet_templa

Puppet creating and managing user accounts with SSH access (/DevOps/Puppet/puppet_creatir

Puppet Locking user accounts & deploying sudoers file (/DevOps/Puppet/puppet_locking

Puppet exec resource (/DevOps/Puppet/puppet_exec_r

Puppet classes and modules (/DevOps/Puppet/puppet_classes

Puppet Forge modules (/DevOps/Puppet/Puppet_Forge_

Puppet Express (/DevOps/Puppet/puppet_expres

Puppet Express 2 (/DevOps/Puppet/puppet_expres

Puppet 4 : Changes (/DevOps/Puppet/puppet4_chang

Puppet --configprint (/DevOps/Puppet/puppet_configp

Puppet with Docker (/DevOps/Docker/Docker_puppet

Puppet 6.0.2 install on Ubuntu 18.04 (/DevOps/Puppet/Puppet6-Install-on-Ubuntu18.0.4.php)

```
# Input variable definitions

variable "vpc_name" {
  description = "Name of VPC"
  type        = string
  default     = "example-vpc"
}


variable "vpc_cidr" {
  description = "CIDR block for VPC"
  type        = string
  default     = "10.0.0.0/16"
}

variable "vpc_azs" {
  description = "Availability zones for VPC"
  type        = list(string)
  default     = ["us-east-1a", "us-east-1b", "us-east-1c"]
}

variable "vpc_private_subnets" {
  description = "Private subnets for VPC"
  type        = list(string)
  default     = ["10.0.1.0/24", "10.0.2.0/24"]
}

variable "vpc_public_subnets" {
  description = "Public subnets for VPC"
  type        = list(string)
  default     = ["10.0.101.0/24", "10.0.102.0/24"]
}

variable "vpc_enable_nat_gateway" {
  description = "Enable NAT gateway for VPC"
  type        = bool
  default     = true
}

variable "vpc_tags" {
  description = "Tags to apply to resources created by VPC module"
  type        = map(string)
  default = {
    Terraform   = "true"
    Environment = "dev"
  }
}
```

modules/aws-s3-static-website-bucket/main.tf:

```
resource "aws_s3_bucket" "s3_bucket" {
  bucket = var.bucket_name

  acl    = "public-read"
  policy = <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::${var.bucket_name}/*"
            ]
        }
    ]
}
EOF

  website {
    index_document = "index.html"
    error_document = "error.html"
  }

  tags = var.tags
}
```

modules/aws-s3-static-website-bucket/outputs.tf:

```
# Output variable definitions

output "arn" {
  description = "ARN of the bucket"
  value       = aws_s3_bucket.s3_bucket.arn
}

output "name" {
  description = "Name (id) of the bucket"
  value       = aws_s3_bucket.s3_bucket.id
}

output "domain" {
  description = "Domain name of the bucket"
  value       = aws_s3_bucket.s3_bucket.website_domain
}
```

Machine (/DevOps/Docker/Docker_Contai

Docker install on Ubuntu 14.04 (/DevOps/Docker/Docker_Install_

Docker Hello World Application (/DevOps/Docker/Docker_Hello_V

Nginx image - share/copy files, Dockerfile (/DevOps/Docker/Docker_Nginx_

Working with Docker images : brief introduction (/DevOps/Docker/Docker_Workir

Docker image and container via docker commands (search, pull, run, ps, restart, attach, and rm) (/DevOps/Docker/Docker_Comm

More on docker run command (docker run -it, docker run --rm, etc.) (/DevOps/Docker/Docker_Run_C

Docker Networks - Bridge Driver Network (/DevOps/Docker/Docker-Bridge-Driver-Networks.php)

Docker Persistent Storage (/DevOps/Docker/Docker_Contai

File sharing between host and container (docker run -d -p -v) (/DevOps/Docker/Docker_File_Sh

Linking containers and volume for datastore (/DevOps/Docker/Docker_Contai

Dockerfile - Build Docker images automatically I - FROM, MAINTAINER, and build context (/DevOps/Docker/Docker_Docker

Dockerfile - Build Docker

```
# Input variable definitions

variable "bucket_name" {
  description = "Name of the s3 bucket. Must be unique."
  type        = string
}

variable "tags" {
  description = "Tags to set on the bucket."
  type        = map(string)
  default     = {}
}
```

References:

1. Use Modules from the Registry (https://learn.hashicorp.com/tutorials/terraform/module-use?in=terraform/modules)

Terraform

- Introduction to Terraform with AWS elb & nginx (/DevOps/Terraform/Terraform-Introduction-AWS-elb-nginx.php)
- Terraform Tutorial - terraform format(tf) and interpolation(variables) (/DevOps/Terraform/Terraform-terraform-format-tf-and-interpolation-variables.php)
- Terraform Tutorial - user_data (/DevOps/Terraform/Terraform-terraform-userdata.php)
- Terraform Tutorial - variables (/DevOps/Terraform/Terraform-parameters-variables.php)
- Terraform Tutorial - creating multiple instances (count, list type and element() function) (/DevOps/Terraform/Terraform-creating-multiple-instances-count-list-type.php)
- Terraform 12 Tutorial - Loops with count, for_each, and for (/DevOps/Terraform/Terraform-Introduction-AWS-loops.php)
- Terraform Tutorial - State (terraform.tfstate) & terraform import (/DevOps/Terraform/Terraform-state-tfstate-import.php)
- Terraform Tutorial - Output variables (/DevOps/Terraform/Terraform-output-variables.php)
- Terraform Tutorial - Destroy (/DevOps/Terraform/Terraform-destroy.php)
- Terraform Tutorial - Modules (/DevOps/Terraform/Terraform-modules.php)
- Terraform Tutorial - Creating AWS S3 bucket / SQS queue resources and notifying bucket event to queue (/DevOps/Terraform/Terraform-Introduction-AWS-S3-SQS.php)
- Terraform Tutorial - AWS ASG and Modules (/DevOps/Terraform/Terraform-Introduction-AWS-ASG-Modules.php)
- Terraform Tutorial - VPC, Subnets, RouteTable, ELB, Security Group, and Apache server I (/DevOps/Terraform/Terraform-VPC-Subnet-ELB-RouteTable-SecurityGroup-Apache-Server-1.php)
- Terraform Tutorial - VPC, Subnets, RouteTable, ELB, Security Group, and Apache server II (/DevOps/Terraform/Terraform-VPC-Subnet-ELB-RouteTable-SecurityGroup-Apache-Server-2.php)
- Terraform Tutorial - Docker nginx container with ALB and dynamic autoscaling

Dockerfile - Build Docker images automatically III - RUN (/DevOps/Docker/Docker_Docker

Dockerfile - Build Docker images automatically IV - CMD (/DevOps/Docker/Docker_Docker

Dockerfile - Build Docker images automatically V - WORKDIR, ENV, ADD, and ENTRYPOINT (/DevOps/Docker/Docker_Docker

Docker - Apache Tomcat (/DevOps/Docker/Docker_Apache

Docker - NodeJS (/DevOps/Docker/Docker-NodeJS.php)

Docker - NodeJS with hostname (/DevOps/Docker/Docker-NodeJS-with-hostname.php)

Docker Compose - NodeJS with MongoDB (/DevOps/Docker/Docker-Compose-Node-MongoDB.php)

Docker - Prometheus and Grafana with Docker-compose (/DevOps/Docker/Docker_Prome

Docker - StatsD/Graphite/Grafana (/DevOps/Docker/Docker_StatsD_

Docker - Deploying a Java EE JBoss/WildFly Application on AWS Elastic Beanstalk Using Docker Containers (/DevOps/Docker/Docker_Contai

Docker : NodeJS with GCP Kubernetes Engine (/DevOps/Docker/Docker-NodeJS-GCP-Kubernetes-Engine.php)

- HashiCorp Vault and Consul on AWS with Terraform (/DevOps/Terraform/Hashicorp-Vault-and-Consul-on-AWS-with-Terraform.php)
- Ansible with Terraform (/DevOps/Ansible/Ansible-Terraform-null_resource-local-exec-remote-exec-triggers.php)
- AWS IAM user, group, role, and policies - part 1 (/DevOps/Terraform/Terraform_IAM_User_group_role_Policies_1.php)
- AWS IAM user, group, role, and policies - part 2 (/DevOps/Terraform/Terraform_IAM_User_group_role_Policies_2.php)
- Delegate Access Across AWS Accounts Using IAM Roles (/DevOps/Terraform/Terraform_AWS_sts_AssumeRole_Cross_Account.php)
- AWS KMS (/DevOps/Terraform/Terraform-AWS-KMS.php)
- Terraform import (/DevOps/Terraform/Terraform_Import.php)
- Terraform commands cheat sheet (/DevOps/Terraform/Terraform_commands_cheat_sheet.php)
- Terraform Cloud (/DevOps/Terraform/Terraform-Cloud.php)
- Terraform 14 (/DevOps/Terraform/Terraform14.php)
- Creating Private TLS Certs (/DevOps/Terraform/Terraform-private-tls-certs.php)

Jenkins-Multibranch-Pipeline-with-Jenkinsfile-and-Github.php)

Docker : Jenkins Master and Slave (/DevOps/Docker/Docker-Jenkins-Master-Slave-Agent-ssh.php)

Docker - ELK : ElasticSearch, Logstash, and Kibana (/DevOps/Docker/Docker_ELK_El

Docker - ELK 7.6 : Elasticsearch on Centos 7 (/DevOps/Docker/Docker_ELK_7_
Docker - ELK 7.6 : Filebeat on Centos 7 (/DevOps/Docker/Docker_ELK_7_

Docker - ELK 7.6 : Logstash on Centos 7 (/DevOps/Docker/Docker_ELK_7_

Docker - ELK 7.6 : Kibana on Centos 7 Part 1 (/DevOps/Docker/Docker_ELK_7_

Docker - ELK 7.6 : Kibana on Centos 7 Part 2 (/DevOps/Docker/Docker_ELK_7_

Docker - ELK 7.6 : Elastic Stack with Docker Compose (/DevOps/Docker/Docker_ELK_7_

Docker - Deploy Elastic Cloud on Kubernetes (ECK) via Elasticsearch operator on minikube (/DevOps/Docker/Docker_Kuberr

Docker - Deploy Elastic Stack via Helm on minikube (/DevOps/Docker/Docker_Kuberr

Docker Compose - A gentle introduction with WordPress (/DevOps/Docker/Docker-

MEAN Stack app on Docker containers : micro services (/MEAN-Stack/MEAN-Stack-NodeJS-Angular-Docker.php)

Docker Compose - Hashicorp's Vault and Consul Part A (install vault, unsealing, static secrets, and policies) (/DevOps/Docker/Docker-Vault-Consul.php)

Docker Compose - Hashicorp's Vault and Consul Part B (EaaS, dynamic secrets, leases, and revocation) (/DevOps/Docker/Docker-Vault-Consul-B.php)

Docker Compose - Hashicorp's Vault and Consul Part C (Consul) (/DevOps/Docker/Docker-Vault-Consul-C.php)

Docker Compose with two containers - Flask REST API service container and an Apache server container (/DevOps/Docker/Docker-Compose-FlaskREST-Service-Container-and-Apache-Container.php)

Docker compose : Nginx reverse proxy with multiple containers (/DevOps/Docker/Docker-Compose-Nginx-Reverse-Proxy-Multiple-Containers.php)

Docker compose : Nginx reverse proxy with multiple containers (/DevOps/Docker/Docker-Compose-Nginx-Reverse-Proxy-Multiple-Containers.php)

Docker.php)

Docker install via Puppet (/DevOps/Docker/Docker_puppet

Nginx Docker install via Ansible (/DevOps/Ansible/Ansible-Deploy-Nginx-to-Docker.php)

Apache Hadoop CDH 5.8 Install with QuickStarts Docker (/Hadoop/BigData_hadoop_CDH5

Docker - Deploying Flask app to ECS (/DevOps/Docker/Docker-Flask-ALB-ECS.php)

Docker Compose - Deploying WordPress to AWS (/DevOps/Docker/Docker-Compose-WordPress-AWS.php)

Docker - WordPress Deploy to ECS with Docker-Compose (ECS-CLI EC2 type) (/DevOps/Docker/Docker-ECS-CLI-Docker-Compose-Wordpress-EC2-Type.php)

Docker - AWS ECS service discovery with Flask and Redis (/DevOps/Docker/Docker-ALB-ECS-Fargate.php)

Docker - ECS Fargate (/DevOps/Docker/Docker-ECS-Service-Dicsovery-Redis-Flask.php)

Docker & Kubernetes 1 : minikube (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes 2 : minikube Django with Postgres - persistent volume (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes 3 :

Django with RDS via AWS Kops (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Kops on AWS (/DevOps/DevOps-Kubernetes-II-kops-on-AWS.php)

Docker & Kubernetes : Ingress controller on AWS with Kops (/DevOps/Docker/Docker-Kubernetes-kops-on-AWS-Ingress.php)

Docker & Kubernetes : HashiCorp's Vault and Consul on minikube (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : HashiCorp's Vault and Consul - Auto-unseal using Transit Secrets Engine (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Persistent Volumes & Persistent Volumes Claims - hostPath and annotations (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Persistent Volumes - Dynamic volume provisioning (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : DaemonSet (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Secrets (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : kubectl command (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Assign a Kubernetes Pod to a particular

ConfigMap (/DevOps/Docker/Docker_Kuberr

AWS : EKS (Elastic Container Service for Kubernetes) (/DevOps/AWS/aws-EKS-Elastic-Container-Service-Kubernetes.php)

Docker & Kubernetes : Run a React app in a minikube (/DevOps/Docker/Docker-Kubernetes-React-App.php)

Docker & Kubernetes : Minikube install on AWS EC2 (/DevOps/Docker/Docker-Kubernetes-Minikube-install-on-AWS-EC2.php)

Docker & Kubernetes : Cassandra with a StatefulSet (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Terraform and AWS EKS (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Pods and Service definitions (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Headless service and discovering pods (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Service IP and the Service Type (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Kubernetes DNS with Pods and Services (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes - Scaling and Updating application (/DevOps/Docker/Docker_Kuberr

Kubernetes-Horizontal-Pod-Autoscaler.php)

Docker & Kubernetes : NodePort vs LoadBalancer vs Ingress (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Load Testing with Locust on GCP Kubernetes (/DevOps/Docker/Docker-Load-Testing-with-Locust-on-GCP-Kubernetes.php)

Docker & Kubernetes : From a monolithic app to micro services on GCP Kubernetes (/DevOps/Docker/Docker-from-Monolithic-to-Micro-services-GCP-Kubernetes.php)

Docker & Kubernetes : Rolling updates (/DevOps/Docker/Docker-Kubernetes-Rolling-Updates.php)

Docker & Kubernetes : Deployments to GKE (Rolling update, Canary and Blue-green deployments) (/DevOps/Docker/Docker-Rolling-Update-Canary-Blue-Green-Deployments-to-GKE-Kubernetes.php)

Docker & Kubernetes : Slack Chat Bot with NodeJS on GCP Kubernetes (/DevOps/Docker/Docker-Slack-NodeJS-ChatBot-GCP-Kubernetes.php)

Docker & Kubernetes : Continuous Delivery with Jenkins Multibranch Pipeline for Dev, Canary, and Production Environments on GCP

Environments-GCP-Kubernetes-Engine-Namespace.php)

Docker & Kubernetes - MongoDB with StatefulSets on GCP Kubernetes Engine (/DevOps/Docker/Docker_Mongo

Docker & Kubernetes : Nginx Ingress Controller on minikube (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Nginx Ingress Controller for Dashboard service on Minikube (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Nginx Ingress Controller on GCP Kubernetes (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Kubernetes Ingress with AWS ALB Ingress Controller in EKS (/DevOps/Docker/Docker-Kubernetes-ALB-Ingress-Controller-with-EKS.php)

Docker & Kubernetes : MongoDB / MongoExpress on Minikube (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Setting up a private cluster on GCP Kubernetes (/DevOps/Docker/Docker-setting-up-private-cluster-on-GCP-Kubernetes.php)

Docker & Kubernetes : Kubernetes Namespaces (default, kube-public, kube-system) and switching namespaces (kubens) (/DevOps/Docker/Docker-Kubernetes-Namespaces.php)

StatefulSets on minikube (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : RBAC (/DevOps/Docker/Docker-Kubernetes-RBAC.php)

Docker & Kubernetes Service Account, RBAC, and IAM (/DevOps/Docker/Docker-Kubernetes-Service-Account.php)

Docker & Kubernetes - Kubernetes Service Account, RBAC, IAM with EKS ALB, Part 1 (/DevOps/Docker/Docker-Kubernetes-ALB-on-EKS-1.php)

Docker & Kubernetes : Helm Chart (/DevOps/Docker/Docker_Helm_(

Docker & Kubernetes : My first Helm deploy (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Readiness and Liveness Probes (/DevOps/Docker/Docker-Kubernetes-Readiness-Liveness-Probes.php)

Docker & Kubernetes : Helm chart repository with Github pages (/DevOps/Docker/Docker_Helm_(

Docker & Kubernetes : Deploying WordPress and MariaDB with Ingress to Minikube using Helm Chart (/DevOps/Docker/Docker_Helm_(

Docker & Kubernetes : Deploying WordPress and MariaDB to AWS using Helm 2 Chart (/DevOps/Docker/Docker_Helm_(

(/DevOps/Docker/Docker_Helm3

Docker & Kubernetes : Helm Chart for Node/Express and MySQL with Ingress (/DevOps/Docker/Docker_Helm_(

Docker & Kubernetes : Docker_Helm_Chart_Node_Expes (/DevOps/Docker/Docker_Helm_(

Docker & Kubernetes: Deploy Prometheus and Grafana using Helm and Prometheus Operator - Monitoring Kubernetes node resources out of the box (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Istio (service mesh) sidecar proxy on GCP Kubernetes (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Istio on EKS (/DevOps/Docker/Docker-Kubernetes-EKS-with-ISTIO.php)

Docker & Kubernetes : Deploying .NET Core app to Kubernetes Engine and configuring its traffic managed by Istio (Part I) (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Deploying .NET Core app to Kubernetes Engine and configuring its traffic managed by Istio (Part II - Prometheus, Grafana, pin a service, split traffic, and inject faults) (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Helm Package Manager with MySQL on GCP Kubernetes Engine (/DevOps/Docker/Docker_Helm_I

Docker & Kubernetes : EKS Control Plane (API server) Metrics with Prometheus (/DevOps/Docker/Docker-Kubernetes-EKS-Control-Plane-API-Server-Metrics-with-Prometheus.php)

Docker & Kubernetes : Spinnaker on EKS with Halyard (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Continuous Delivery Pipelines with Spinnaker and Kubernetes Engine (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes: Multi-node Local Kubernetes cluster - Kubeadm-dind(docker-in-docker) (/DevOps/Docker/Docker-Kubernetes-Multi-Node-Local-Clusters-dind.php)

Docker & Kubernetes: Multi-node Local Kubernetes cluster - Kubeadm-kind(k8s-in-docker) (/DevOps/Docker/Docker-Kubernetes-Multi-Node-Local-Clusters-kind.php)

Docker & Kubernetes : nodeSelector, nodeAffinity, taints/tolerations, pod affinity and anti-affinity - Assigning Pods to Nodes (/DevOps/Docker/Docker_Kuberr

Docker & Kubernetes : Jenkins-X on EKS (/DevOps/Docker/Docker_Kuberr X-EKS.php)

Docker & Kubernetes : ArgoCD App of Apps with Heml on Kubernetes

Elasticsearch search engine, Logstash, and Kibana

Elasticsearch, search engine (/Hadoop/ELK/ELK_Elastic_Search

Logstash with Elasticsearch (/Hadoop/ELK/ELK_ElasticSearch_

Logstash, Elasticsearch, and Kibana 4 (/Hadoop/ELK/ELK_ElasticSearch_

Elasticsearch with Redis broker and Logstash Shipper and Indexer (/Hadoop/ELK/ELK_Logstash_Ship

Samples of ELK architecture (/Hadoop/ELK/ELK_Architecture_S

Elasticsearch indexing performance (/Hadoop/ELK/ELK_Elastic_Search

Vagrant

VirtualBox & Vagrant install on Ubuntu 14.04 (/DevOps/Vagrant/Vagrant_Virtua

Provisioning
(/DevOps/Vagrant/Vagrant_Provis

Networking - Port Forwarding
(/DevOps/Vagrant/Vagrant_Netw

Vagrant Share
(/DevOps/Vagrant/Vagrant_Share

Vagrant Rebuild & Teardown
(/DevOps/Vagrant/Vagrant_Rebu

Vagrant & Ansible
(/DevOps/Vagrant/Vagrant_Ansib

Big Data & Hadoop Tutorials

Hadoop 2.6 - Installing on Ubuntu 14.04 (Single-Node Cluster)
(/Hadoop/BigData_hadoop_Insta

Hadoop 2.6.5 - Installing on Ubuntu 16.04 (Single-Node Cluster)
(/Hadoop/BigData_hadoop_Insta

Hadoop - Running MapReduce Job
(/Hadoop/BigData_hadoop_Runn

Hadoop - Ecosystem
(/Hadoop/BigData_hadoop_Ecosy

CDH5.3 Install on four EC2 instances (1 Name node and 3 Datanodes) using Cloudera Manager 5
(/Hadoop/BigData_hadoop_CDH5

QuickStart VMs for CDH 5.3 (/Hadoop/BigData_hadoop_Quick VMs for CDH 5.3.x.php)

QuickStart VMs for CDH 5.3 II - Testing with wordcount (/Hadoop/BigData_hadoop_Quick VMs for CDH 5.3.x_II.php)

QuickStart VMs for CDH 5.3 II - Hive DB query (/Hadoop/BigData_hadoop_Quick

Scheduled start and stop CDH services (/Hadoop/BigData_hadoop_CDH5

CDH 5.8 Install with QuickStarts Docker (/Hadoop/BigData_hadoop_CDH5

Zookeeper & Kafka Install (/Hadoop/BigData_hadoop_Zook

Zookeeper & Kafka - single node single broker (/Hadoop/BigData_hadoop_Zook

Zookeeper & Kafka - Single node and multiple brokers (/Hadoop/BigData_hadoop_Zook

OLTP vs OLAP (/Hadoop/BigData_hadoop_OLTP

Apache Hadoop Tutorial I with CDH - Overview (/Hadoop/BigData_hadoop_Cloud

Apache Hadoop Tutorial II with CDH - MapReduce Word Count (/Hadoop/BigData_hadoop_Tutor

Apache Hadoop Tutorial III with CDH - MapReduce Word Count 2 (/Hadoop/BigData_hadoop_Tutor

Apache Hadoop (CDH 5) Hive

(/Hadoop/BigData_hadoop_CDH5

Apache Hive 2.1.0 install on Ubuntu 16.04 (/Hadoop/BigData_hadoop_Hive_

Apache HBase in Pseudo-Distributed mode (/Hadoop/BigData_hadoop_HBas

Creating HBase table with HBase shell and HUE (/Hadoop/BigData_hadoop_HBas

Apache Hadoop : Hue 3.11 install on Ubuntu 16.04 (/Hadoop/BigData_hadoop_Hue_

Creating HBase table with Java API (/Hadoop/BigData_hadoop_HBas

HBase - Map, Persistent, Sparse, Sorted, Distributed and Multidimensional (/Hadoop/BigData_hadoop_HBas

Flume with CDH5: a single-node Flume deployment (telnet example) (/Hadoop/BigData_hadoop_CDH5

Apache Hadoop (CDH 5) Flume with VirtualBox : syslog example via NettyAvroRpcClient (/Hadoop/BigData_hadoop_CDH5

List of Apache Hadoop hdfs commands (/Hadoop/BigData_Hadoop_fs_co

Apache Hadoop : Creating Wordcount Java Project with Eclipse Part 1 (/Hadoop/BigData_hadoop_Creat

Apache Hadoop : Creating Wordcount Java Project with

Cloudera VM UnoExample for CDH5 - local run (/Hadoop/BigData_hadoop_Creat

Apache Hadoop : Creating Wordcount Maven Project with Eclipse (/Hadoop/BigData_hadoop_Creat

Wordcount MapReduce with Oozie workflow with Hue browser - CDH 5.3 Hadoop cluster using VirtualBox and QuickStart VM (/Hadoop/BigData_hadoop_Quick

Spark 1.2 using VirtualBox and QuickStart VM - wordcount (/Hadoop/BigData_hadoop_Apac

Spark Programming Model : Resilient Distributed Dataset (RDD) with CDH (/Hadoop/BigData_hadoop_Apac

Apache Spark 2.0.2 with PySpark (Spark Python API) Shell (/Hadoop/BigData_hadoop_Apac

Apache Spark 2.0.2 tutorial with PySpark : RDD (/Hadoop/BigData_hadoop_Apac

Apache Spark 2.0.0 tutorial with PySpark : Analyzing Neuroimaging Data with Thunder (/Hadoop/BigData_hadoop_Apac

Apache Spark Streaming with Kafka and Cassandra (/Hadoop/BigData_hadoop_Apac

Apache Spark 1.2 with PySpark (Spark Python API) Wordcount using CDH5 (/Hadoop/BigData_hadoop_Apac

Embedded & Distributed
(/Drill/Drill_Tutorial_Install_on_ub

Apache Drill - Query File System, JSON, and Parquet
(/Drill/Drill_Tutorial_Query_File_S

Apache Drill - HBase query
(/Drill/Drill_Tutorial_Query_HBase

Apache Drill - Hive query
(/Drill/Drill_Tutorial_Query_Hive.p

Apache Drill - MongoDB query
(/Drill/Drill_Tutorial_Query_Mong

Redis In-Memory Database

Redis vs Memcached
(/DevOps/Redis/Redis_vs_Memca

Redis 3.0.1 Install
(/DevOps/Redis/Redis_Install.php

Setting up multiple server instances on a Linux host
(/DevOps/Redis/Redis_Setting_up

Redis with Python
(/python/python_redis_with_pyth

ELK : Elasticsearch with Redis broker and Logstash Shipper and Indexer
(/Hadoop/ELK/ELK_Logstash_Ship

Platform)

GCP: Creating an Instance (/DevOps/GCP/gcp_Creating_an_I

GCP: gcloud compute command-line tool (/DevOps/GCP/gcp_gcloud_comp

GCP: Deploying Containers (/DevOps/GCP/gcp_Deploying_Cc

GCP: Kubernetes Quickstart (/DevOps/GCP/gcp_Kubernetes_C

GCP: Deploying a containerized web application via Kubernetes (/DevOps/GCP/gcp_Deploying_Cc

GCP: Django Deploy via Kubernetes I (local) (/DevOps/GCP/gcp_Django_Deplo

GCP: Django Deploy via Kubernetes II (GKE) (/DevOps/GCP/gcp_Django_Deplo

AWS (Amazon Web Services)

AWS : EKS (Elastic Container Service for Kubernetes) (/DevOps/AWS/aws-EKS-Elastic-Container-Service-Kubernetes.php)

AWS : Creating a snapshot (cloning an image) (/DevOps/AWS/aws_snapshot_an

AWS : Adding swap space to an attached volume via mkswap and swapon (/DevOps/AWS/aws_adding_swap

AWS : Creating an EC2 instance and attaching Amazon EBS volume to the instance using Python boto module with User data (/DevOps/AWS/aws_creating_an_

AWS : Creating an instance to a new region by copying an AMI (/DevOps/AWS/Launching-Instance-to-a-New-Region-from-an-AMI.php)

AWS : S3 (Simple Storage Service) 1 (/DevOps/AWS/aws_S3_Simple_St

AWS : S3 (Simple Storage Service) 2 - Creating and Deleting a Bucket (/DevOps/AWS/aws_S3_Simple_St

AWS : S3 (Simple Storage Service) 3 - Bucket Versioning (/DevOps/AWS/aws_S3_Simple_St

AWS : S3 (Simple Storage Service) 4 - Uploading a large file (/DevOps/AWS/aws_S3_uploading

AWS : S3 (Simple Storage Service) 5 - Uploading folders/files recursively (/DevOps/AWS/aws-uploading-recursive-folder-file.php)

AWS : S3 (Simple Storage Service) 6 - Bucket Policy for File/Folder View/Download (/DevOps/AWS/aws-S3-bucket-policy-view-download.php)

AWS : S3 (Simple Storage

Destination-Buckets-Owned-by-the-Same-AWS-Account-How-to-Copy-or-Move-Objects-from-one-Region-to-another.php)

AWS : S3 (Simple Storage Service) 8 - Archiving S3 Data to Glacier (/DevOps/AWS/aws-S3-Simple-Storage-Service-Archiving-Amazon-S3-Data-to-Amazon-Glacier.php)

AWS : Creating a CloudFront distribution with an Amazon S3 origin (/DevOps/AWS/aws_Creating_Clo

AWS : Creating VPC with CloudFormation (/DevOps/AWS/aws-creating-VPC-with-CloudFormation.php)

WAF (Web Application Firewall) with preconfigured CloudFormation template and Web ACL for CloudFront distribution (/DevOps/AWS/aws-WAF-Web-Application-Firewall.php)

AWS : CloudWatch & Logs with Lambda Function / S3 (/DevOps/AWS/aws-CloudWatch-logs-Lambda-S3.php)

AWS : Lambda Serverless Computing with EC2, CloudWatch Alarm, SNS (/DevOps/AWS/aws-Lambda-Serverless-S3-CloudWatch-Alarm-SNS.php)

AWS : Lambda and SNS - cross account (/DevOps/AWS/aws-Lambda-SNS.php)

AWS : CLI (Command Line Interface) (/DevOps/AWS/aws-

(/DevOps/AWS/aws-Amazon-ECS-ALB-Autoscaling-CLI.php)

AWS : ECS with cloudformation and json task definition (/DevOps/AWS/aws-ECS-with-CloudFormation-and-json-task-definition.php)

AWS : AWS Application Load Balancer (ALB) and ECS with Flask app (/DevOps/AWS/aws-ELB-ALB-Application-Load-Balancer-ECS.php)

AWS : Load Balancing with HAProxy (High Availability Proxy) (/DevOps/AWS/aws-Load-Balancing-with-HAProxy-High-Availability-Proxy.php)

AWS : VirtualBox on EC2 (/DevOps/AWS/aws_VirtualBox_C

AWS : NTP setup on EC2 (/DevOps/AWS/aws_NTP.php)

AWS: jq with AWS (/DevOps/AWS/aws-jq.php)

AWS : AWS & OpenSSL : Creating / Installing a Server SSL Certificate (/DevOps/AWS/aws-HTTPS-OpenSSL-Certificate.php)

AWS : OpenVPN Access Server 2 Install (/DevOps/AWS/aws-OpenVPN-Access-Server-Install.php)

AWS : VPC (Virtual Private Cloud) 1 - netmask, subnets, default gateway, and CIDR (/DevOps/AWS/aws-VPC-Virtual-Private-Cloud-1-netmast-subnet-default-gateway-CIDR.php)

AWS : VPC (Virtual Private Cloud) 3 - VPC Wizard with NAT (/DevOps/AWS/aws-VPC-Virtual-Private-Cloud-3-VPC-Wizard-with-NAT.php)

AWS : DevOps / Sys Admin Q & A (VI) - AWS VPC setup (public/private subnets with NAT) (/DevOps/DevOps-Sys-Admin-Interview-Questions-AWS-VPC-Setup.php)

AWS : OpenVPN Protocols : PPTP, L2TP/IPsec, and OpenVPN (/DevOps/AWS/aws-VPN-Protocols-OpenVPN-IPsec-L2TP-PPTP.php)

AWS : Autoscaling group (ASG) (/DevOps/AWS/aws-Autoscaling-Group-ASG.php)

AWS : Setting up Autoscaling Alarms and Notifications via CLI and Cloudformation (/DevOps/AWS/aws-AutoScaling-LaunchConfiguration-Notification-LoadBalancer-CloudWatch-Alert-High-Low-CPU-utilization-CLI-Cloudformation.php)

AWS : Adding a SSH User Account on Linux Instance (/DevOps/AWS/aws-adding-a-ssh-user-account-on-linux-instance.php)

AWS : Windows Servers - Remote Desktop Connections using RDP (/DevOps/AWS/aws_Windows_Se

AWS : Scheduled stopping and starting an instance - python & cron

(/DevOps/AWS/aws_detecting_st

AWS : Elastic Beanstalk with NodeJS (/DevOps/AWS/aws-Elastic-Beanstalk-with-NodeJS.php)

AWS : Elastic Beanstalk Inplace/Rolling Blue/Green Deploy (/DevOps/AWS/aws-Beanstalk-InPlaceRolling-BlueGreen-Deploy.php)

AWS : Identity and Access Management (IAM) Roles for Amazon EC2 (/DevOps/AWS/aws-IAM-Roles.php)

AWS : Identity and Access Management (IAM) Policies, sts AssumeRole, and delegate access across AWS accounts (/DevOps/AWS/aws-IAM-Policies.php)

AWS : Identity and Access Management (IAM) sts assume role via aws cli2 (/DevOps/AWS/aws-sts-assume-roles-cli.php)

AWS : Creating IAM Roles and associating them with EC2 Instances in CloudFormation (/DevOps/AWS/aws-creating-IAM-Roles-and-associating-them-with-EC2-Instances-in-CloudFormation.php)

AWS Identity and Access Management (IAM) Roles, SSO(Single Sign On), SAML(Security Assertion Markup Language), IdP(identity provider), STS(Security Token Service), and ADFS(Active Directory Federation Services) (/DevOps/AWS/aws-IAM-Roles-

AWS : Amazon Route 53 (/DevOps/AWS/aws-Route53-DNS.php)

AWS : Amazon Route 53 - DNS (Domain Name Server) setup (/DevOps/AWS/aws-Route53-DNS-Domain-Name-Server-Setup.php)

AWS : Amazon Route 53 - subdomain setup and virtual host on Nginx (/DevOps/AWS/aws-Route53-DNS-Subdomain.php)

AWS Amazon Route 53 : Private Hosted Zone (/DevOps/AWS/aws-Route53-DNS-Private-Hosted-Zone.php)

AWS : SNS (Simple Notification Service) example with ELB and CloudWatch (/DevOps/AWS/aws-Amazon-SNS-Simple-Notification-Service-example-with-ELB.php)

AWS : Lambda with AWS CloudTrail (/DevOps/AWS/aws-Lambda-with-AWS-CloudTrail.php)

AWS : SQS (Simple Queue Service) with NodeJS and AWS SDK (/DevOps/AWS/aws-Amazon-SQS-Simple-Queue-Service-with-NodeJS-AWS-SDK.php)

AWS : Redshift data warehouse (/DevOps/AWS/aws_Redshift_dat

AWS : CloudFormation - templates, change sets, and CLI (/DevOps/AWS/aws-CloudFormation-Templates.php)

UserData.php)

AWS : CloudFormation - Creating an ASG with rolling update (/DevOps/AWS/aws-CloudFormation-Autoscaling-Group-ASG-Application-Load-Balancer-ALB-with-Update-Policy-Rolling-Updates.php)

AWS : Cloudformation Cross-stack reference (/DevOps/AWS/aws-Cloudformation-CrossStck-Reference.php)

AWS : OpsWorks (/DevOps/AWS/aws-OpsWorks.php)

AWS : Network Load Balancer (NLB) with Autoscaling group (ASG) (/DevOps/AWS/aws-Autoscaling-Group-Network-Load-Balancer.php)

AWS CodeDeploy : Deploy an Application from GitHub (/DevOps/AWS/aws-CodeDeploy-Deploy-an-Application-from-GitHub.php)

AWS EC2 Container Service (ECS) (/DevOps/AWS/aws-Amazon-EC2-Container-Service-ECS.php)

AWS EC2 Container Service (ECS) II (/DevOps/AWS/aws-Amazon-EC2-Container-Service-ECS-2.php)

AWS Hello World Lambda Function (/DevOps/AWS/aws-Hello-World-Lambda-Function.php)

AWS Lambda Function Q & A (/DevOps/AWS/aws-Lambda-

Nodejs-API-Gateway.php)

AWS API Gateway endpoint invoking Lambda function (/DevOps/AWS/aws-API-Gateway.php)

AWS API Gateway invoking Lambda function with Terraform (/DevOps/AWS/aws-API-Gateway-Lambda-Terraform.php)

AWS API Gateway invoking Lambda function with Terraform - Lambda Container (/DevOps/AWS/aws-API-Gateway-Lambda-Terraform-with-ECR-Container.php)

Amazon Kinesis Streams (/DevOps/AWS/aws-Amazon-Kinesis-Streams.php)

Kinesis Data Firehose with Lambda and ElasticSearch (/DevOps/AWS/aws-Kinesis-Data-Firehose-TransformData-with-Lambda-send-Data-to-ElasticSearch.php)

Amazon DynamoDB (/DevOps/AWS/aws-Amazon-DynamoDB.php)

Amazon DynamoDB with Lambda and CloudWatch (/DevOps/AWS/aws-Amazon-DynamoDB-Lambda-CloudWatch.php)

Loading DynamoDB stream to AWS Elasticsearch service with Lambda (/DevOps/AWS/aws-Amazon-Loading-DynamoDB-Stream-to-ElasticSearch-with-Lambda.php)

Amazon ML (Machine Learning)

AWS : RDS Connecting to a DB Instance Running the SQL Server Database Engine (/DevOps/AWS/aws_Connecting_t

AWS : RDS Importing and Exporting SQL Server Data (/DevOps/AWS/aws_RDS_Importi

AWS : RDS PostgreSQL & pgAdmin III (/DevOps/AWS/aws_RDS_Postgre

AWS : RDS PostgreSQL 2 - Creating/Deleting a Table (/DevOps/AWS/aws_RDS_Postgre

AWS : MySQL Replication : Master-slave (/DevOps/AWS/aws-MySQL-Replication-Master-Slave.php)

AWS : MySQL backup & restore (/DevOps/AWS/aws-MySQL-Backup-mysqldump-Restore.php)

AWS RDS : Cross-Region Read Replicas for MySQL and Snapshots for PostgreSQL (/DevOps/AWS/aws-RDS-Cross-Region-Read-Replicas-for-MySQL-Snapshot-for-PostgreSQLs.php)

AWS : Restoring Postgres on EC2 instance from S3 backup (/DevOps/AWS/aws_S3_EC2_Post

AWS : Q & A (/DevOps/AWS/aws-Q-A.php)

AWS : Security (/DevOps/AWS/aws-Security.php)

AWS : Scaling-Up (/DevOps/AWS/aws-Scaling-

Powershell 4 Tutorial

Powersehll : Introduction (http://www.bogotobogo.com/Po

Powersehll : Help System (http://www.bogotobogo.com/Po

Powersehll : Running commands (http://www.bogotobogo.com/Po

Powersehll : Providers (http://www.bogotobogo.com/Po

Powersehll : Pipeline (http://www.bogotobogo.com/Po

Powersehll : Objects (http://www.bogotobogo.com/Po

Powershell : Remote Control (http://www.bogotobogo.com/Po

Windows Management Instrumentation (WMI) (http://www.bogotobogo.com/Po

How to Enable Multiple RDP Sessions in Windows 2012 Server (http://www.bogotobogo.com/Po

How to install and configure FTP server on IIS 8 in Windows 2012 Server (http://www.bogotobogo.com/Po

SQL Inner, Left, Right, and Outer Joins (http://www.bogotobogo.com/Po

Git/GitHub Tutorial

One page express tutorial for GIT and GitHub (/cplusplus/Git/Git_GitHub_Expre

Installation (/cplusplus/Git/Git_GitHub_Instal

add/status/log (/cplusplus/Git/Git_GitHub_status

commit and diff (/cplusplus/Git/Git_GitHub_comm

git commit --amend (/cplusplus/Git/Git_GitHub_comm

Deleting and Renaming files (/cplusplus/Git/Git_GitHub_Deleti

Undoing Things : File Checkout & Unstaging (/cplusplus/Git/Git_GitHub_Undo

Reverting commit (/cplusplus/Git/Git_GitHub_Rever

Soft Reset - (git reset --soft <SHA key>) (/cplusplus/Git/Git_GitHub_Soft_F

Mixed Reset - Default (/cplusplus/Git/Git_GitHub_Mixed

Hard Reset - (git reset --hard

Fast-forward merge (/cplusplus/Git/Git_GitHub_Fast-Forward_Merge.php)

Rebase & Three-way merge (/cplusplus/Git/Git_GitHub_Rebas

Merge conflicts with a simple example (/cplusplus/Git/Git_GitHub_Merg

GitHub Account and SSH (/cplusplus/Git/GitHub_Account_S

Uploading to GitHub (/cplusplus/Git/GitHub_Uploadin

GUI (/cplusplus/Git/GitHub_GUI.php)

Branching & Merging (/cplusplus/Git/Git_Branching_Me

Merging conflicts (/cplusplus/Git/Git_Branching_Me

GIT on Ubuntu and OS X - Focused on Branching (/cplusplus/Git/Git_Ubuntu.php)

Setting up a remote repository / pushing local project and cloning the remote repo (/cplusplus/Git/Git_Setting_Up_Re

Fork vs Clone, Origin vs Upstream (/cplusplus/Git/GitHub_Fork_Clor

Git/GitHub Terminologies (/cplusplus/Git/Git_Terminologies

Git/GitHub via SourceTree II : Branching & Merging (/cplusplus/Git/Git_GitHub_Sourc

Git/GitHub via SourceTree III : Git Work Flow

Git wiki - quick command reference (/cplusplus/Git/Git_GitHub_quick_

Subversion

Subversion Install On Ubuntu 14.04 (/cplusplus/Subversion/Subversio

Subversion creating and accessing I (/cplusplus/Subversion/Subversio

Subversion creating and accessing II (/cplusplus/Subversion/Subversio

CONTACT

BogoToBogo
contactus@bogotobogo.com (mailto:contactus@bogotobogo.com)

FOLLOW BOGOTOBOGO

(https://www.facebook.com/KHongSanFrancisco) (https://twitter.com/KHongTwit)

ABOUT US (/ABOUT_US.PHP)

contactus@bogotobogo.com (mailto:contactus@bogotobogo.com)

Golden Gate Ave, San Francisco, CA 94115