R. Da Costa

Kura Labs – Deployment 05 2022-11-19

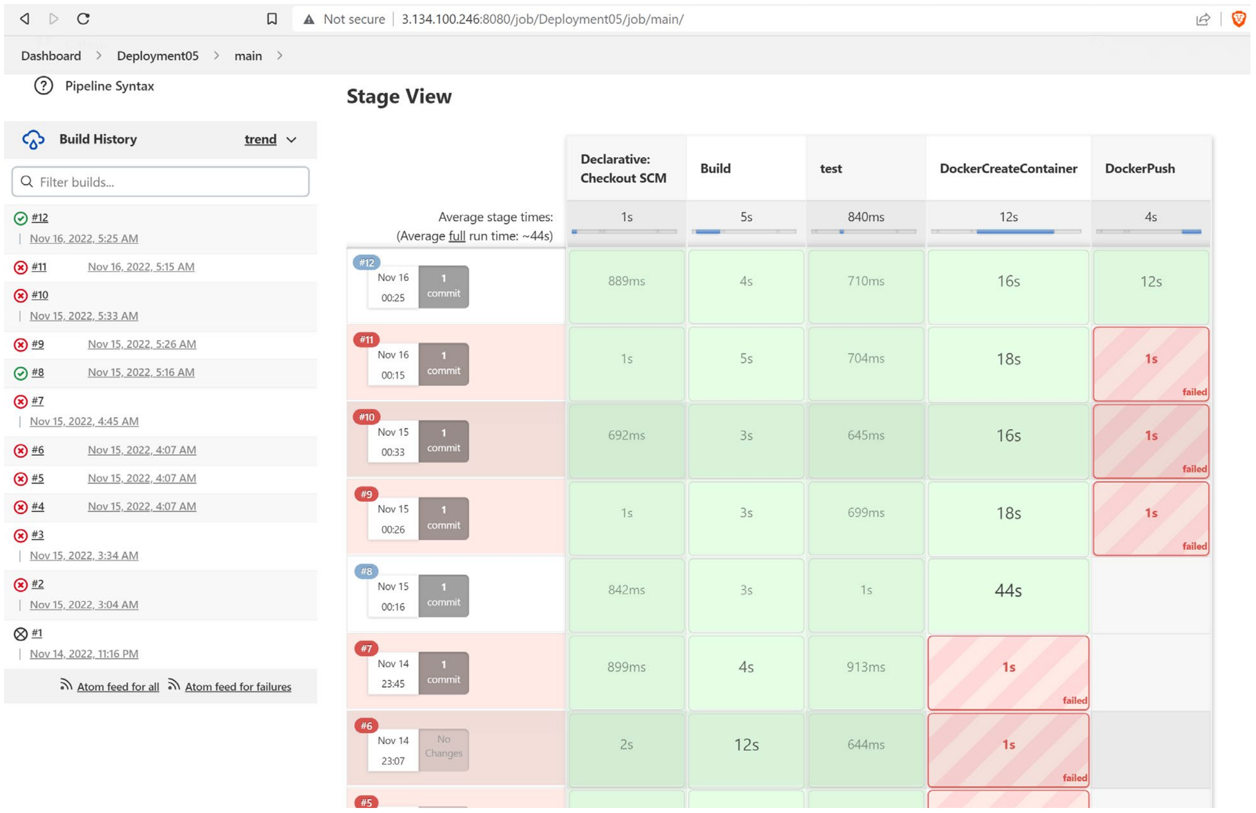
Project Outline:

Using a Jenkins CI/CD Pipeline to deploy an application that has been containerized by a Jenkins Agent that runs Docker, to infrastructure that is spun up on demand via another Jenkins Agent, hosted on a customized VPC, that runs Terraform.

To produce the final product, the Terraform agent uses application image published by the Docker Agent in conjunction with:

i. ECS cluster
ii. Task definition for container creation
iii. VPC
iv. Internet Gateway
v. Public & Private subnets
vi. Security group definitions
vii. ALB & target group definitions
viii. CloudWatch logs

Successful Pipeline BEFORE Terraform stages implemented:


Not secure | 3.134.100.246:8080/job/Deployment05/job/main/

Dashboard > Deployment05 > main >

Pipeline Syntax

Build History trend

Filter builds...

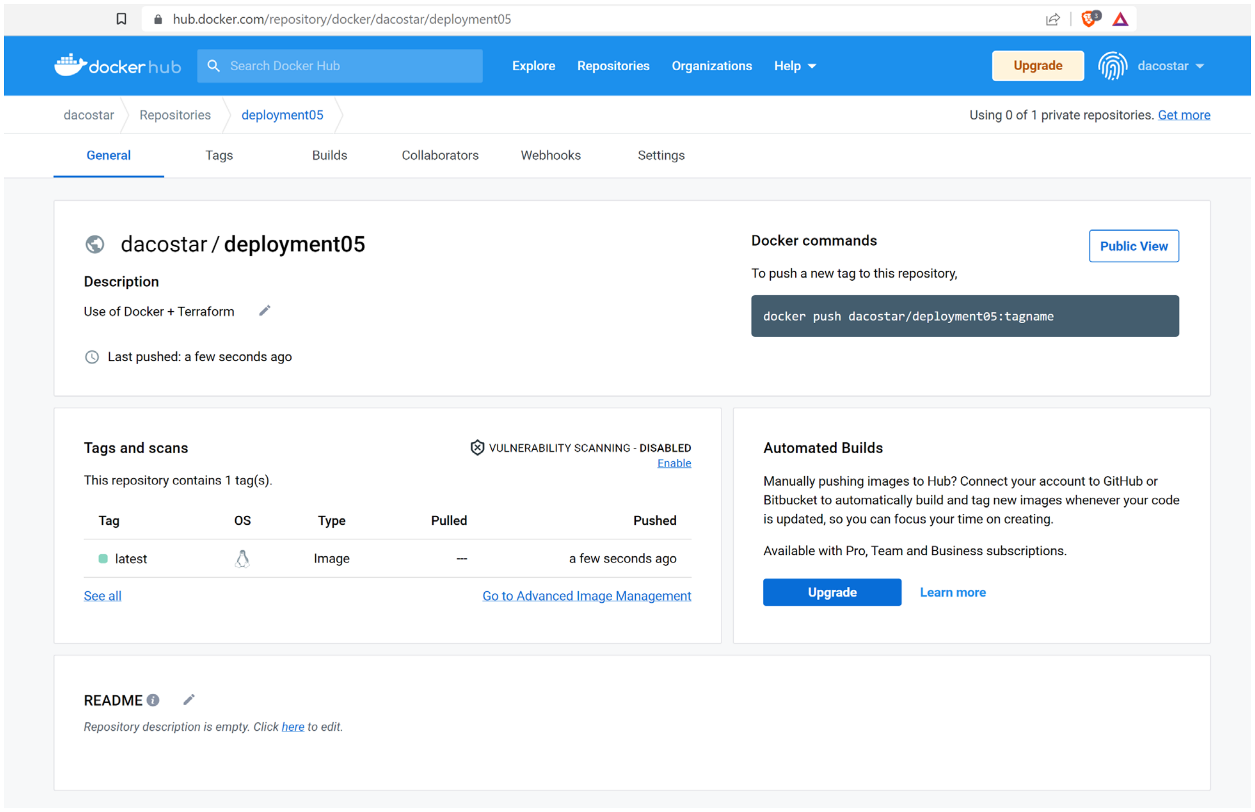
#12 Nov 16, 2022, 5:25 AM
#11 Nov 16, 2022, 5:15 AM
#10 Nov 15, 2022, 5:33 AM
#9 Nov 15, 2022, 5:26 AM
#8 Nov 15, 2022, 5:16 AM
#7 Nov 15, 2022, 4:45 AM
#6 Nov 15, 2022, 4:07 AM
#5 Nov 15, 2022, 4:07 AM
#4 Nov 15, 2022, 4:07 AM
#3 Nov 15, 2022, 3:34 AM
#2 Nov 15, 2022, 3:04 AM
#1 Nov 14, 2022, 11:16 PM

Atom feed for all Atom feed for failures

Stage View

	Declarative: Checkout SCM	Build	test	DockerCreateContainer	DockerPush
Average stage times: (Average full run time: ~44s)	1s	5s	840ms	12s	4s
#12 Nov 16 00:25 1 commit	889ms	4s	710ms	16s	12s
#11 Nov 16 00:15 1 commit	1s	5s	704ms	18s	1s failed
#10 Nov 15 00:33 1 commit	692ms	3s	645ms	16s	1s failed
#9 Nov 15 00:26 1 commit	1s	3s	699ms	18s	1s failed
#8 Nov 15 00:16 1 commit	842ms	3s	1s	44s	
#7 Nov 14 23:45 1 commit	899ms	4s	913ms	1s failed	
#6 Nov 14 23:07 No Changes	2s	12s	644ms	1s failed	
#5					

Successful push to DockerHub:


hub.docker.com/repository/docker/dacostar/deployment05

dacostar Repositories deployment05

Using 0 of 1 private repositories. Get more

dacostar / deployment05

Description

Use of Docker + Terraform

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push dacostar/deployment05:tagname
```

Tags and scans

VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	a few seconds ago

See all
Go to Advanced Image Management

Automated Builds

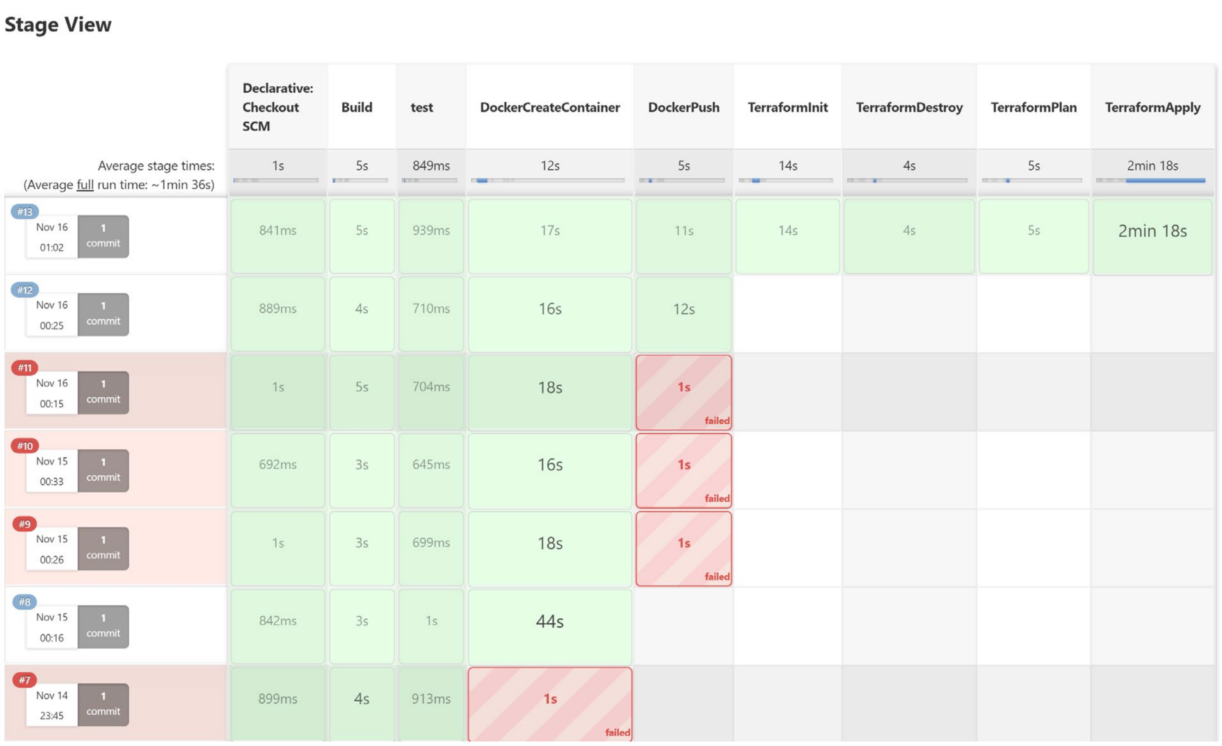
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

README

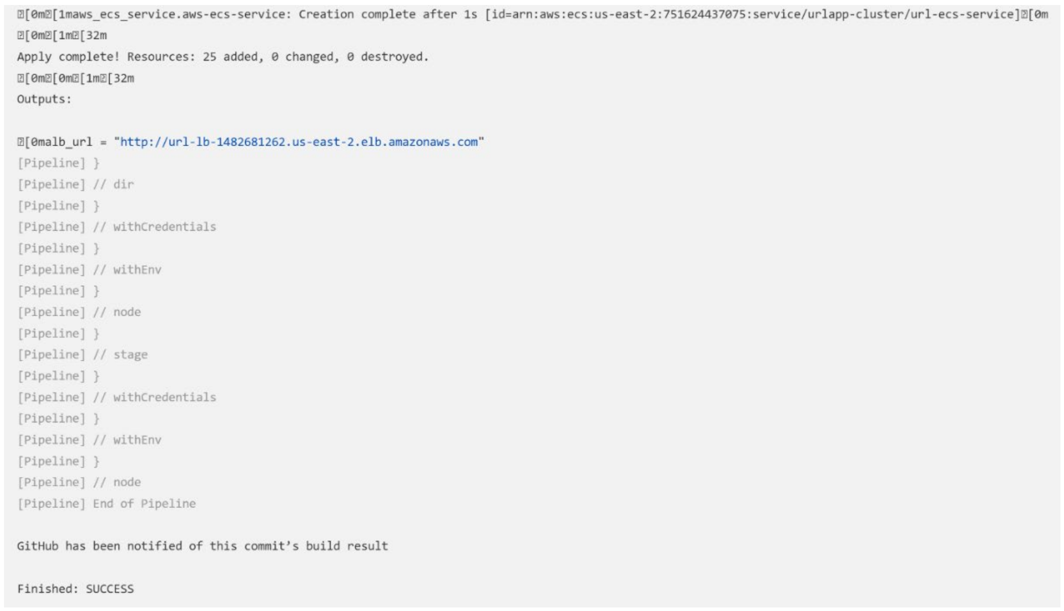
Repository description is empty. Click here to edit.

Successful Terraform Stage Executions [Init, Destroy, Plan, Apply]:


Stage View

	Declarative: Checkout SCM	Build	test	DockerCreateContainer	DockerPush	TerraformInit	TerraformDestroy	TerraformPlan	TerraformApply
Average stage times: (Average full run time: ~1min 36s)	1s	5s	849ms	12s	5s	14s	4s	5s	2min 18s
#13 Nov 16 01:02 1 commit	841ms	5s	939ms	17s	11s	14s	4s	5s	2min 18s
#12 Nov 16 00:25 1 commit	889ms	4s	710ms	16s	12s				
#11 Nov 16 00:15 1 commit	1s	5s	704ms	18s	1s failed				
#10 Nov 15 00:33 1 commit	692ms	3s	645ms	16s	1s failed				
#9 Nov 15 00:26 1 commit	1s	3s	699ms	18s	1s failed				
#8 Nov 15 00:16 1 commit	842ms	3s	1s	44s					
#7 Nov 14 23:45 1 commit	899ms	4s	913ms	1s failed					

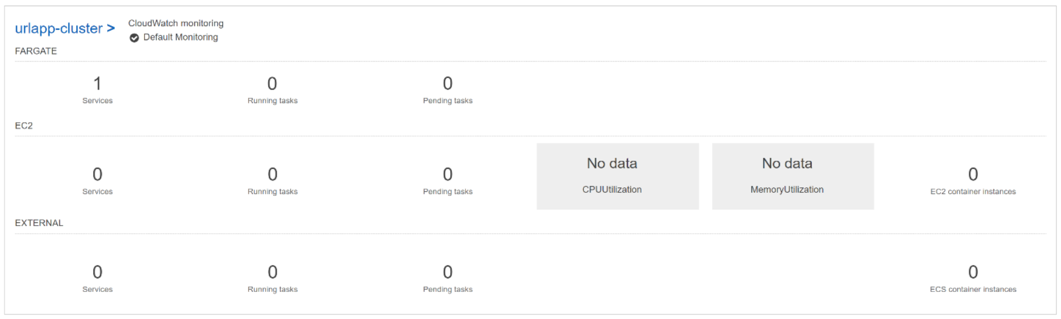
URL output [directive given to output ALB's accessible endpoint after creation in Terraform]:



```
[0m[1maws_ecs_service.aws-ecs-service: Creation complete after 1s [id=arn:aws:ecs:us-east-2:751624437075:service/urlapp-cluster/url-ecs-service][0m
[0m[1m[32m
Apply complete! Resources: 25 added, 0 changed, 0 destroyed.
[0m[0m[1m[32m
Outputs:

[0malb_url = "http://url-lb-1482681262.us-east-2.elb.amazonaws.com"
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

GitHub has been notified of this commit's build result

Finished: SUCCESS
```

Active Clusters in ECS [Elastic Container Services]:

Note: 0 Running tasks



urlapp-cluster >
CloudWatch monitoring
Default Monitoring

FARGATE

1	0	0
Services	Running tasks	Pending tasks

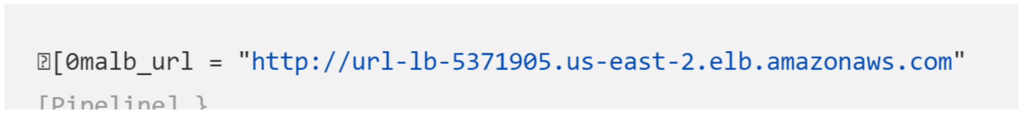
EC2

0	0	0	No data	No data	0
Services	Running tasks	Pending tasks	CPUUtilization	MemoryUtilization	EC2 container instances

EXTERNAL

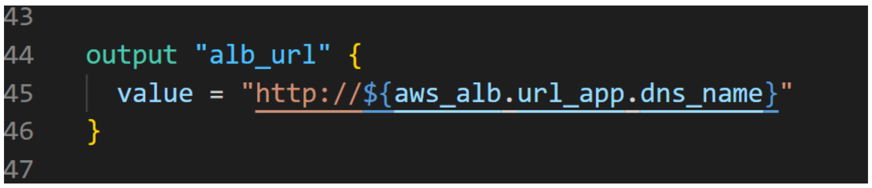
0	0	0	0
Services	Running tasks	Pending tasks	ECS container instances

Output of App url:



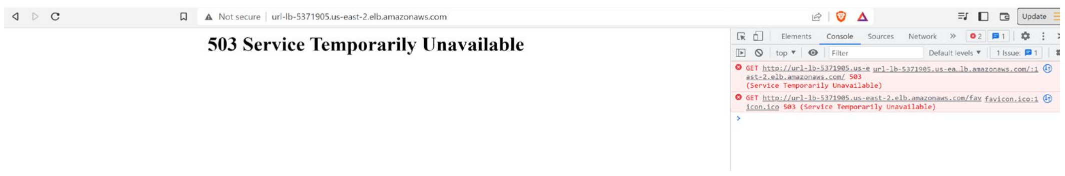
```
[0malb_url = "http://url-lb-5371905.us-east-2.elb.amazonaws.com"
[Pipeline] }
```

Defined in Terraform [ALB.tf]:



```

output "alb_url" {
  value = "http://${aws_alb.url_app.dns_name}"
}

```

503 Error:



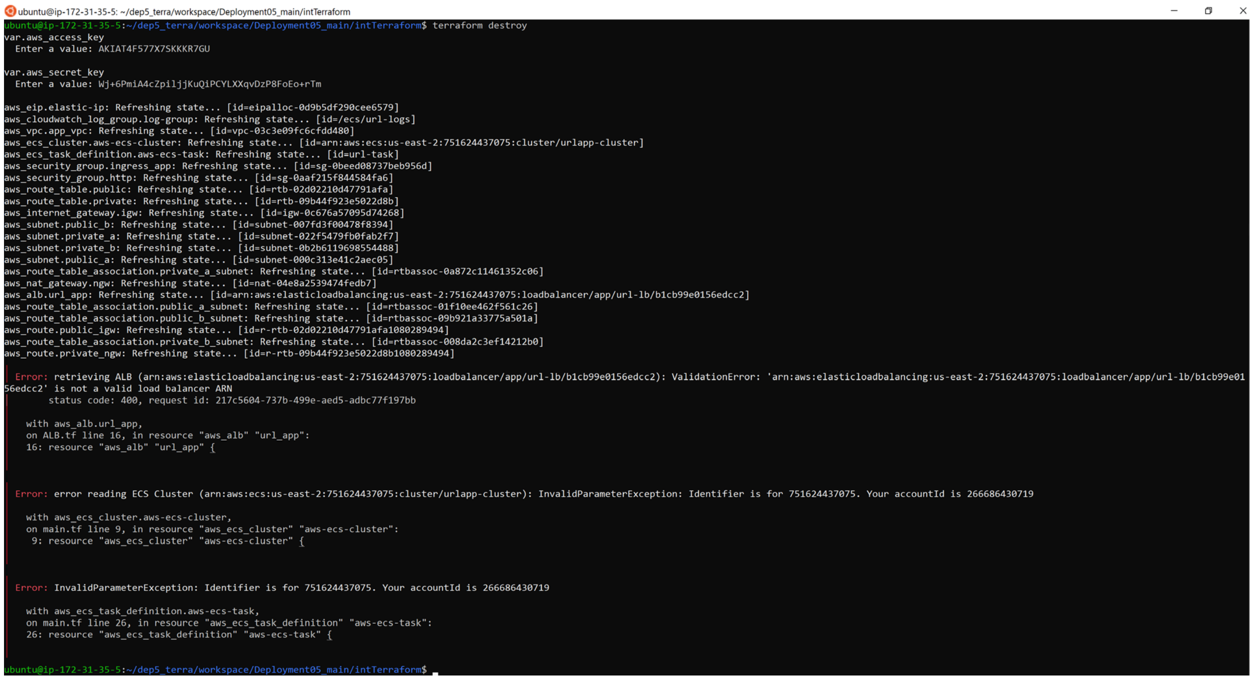
Not secure | url-lb-5371905.us-east-2.elb.amazonaws.com

503 Service Temporarily Unavailable

```
GET http://url-lb-5371905.us-east-2.elb.amazonaws.com/ 503 (Service Temporarily Unavailable)    url-lb-5371905.us-ea…lb.amazonaws.com/:1
GET http://url-lb-5371905.us-east-2.elb.amazonaws.com/favicon.ico 503 (Service Temporarily Unavailable)    favicon.ico:1
```

Attempt to run “terraform destroy” manually [to try to resolve issues]:

- Account flagged as incorrect?
- This was an indication of some level of caching of old information [remnants on the original ARN of the AWS role that was part of the project when it was downloaded]
- This also likely lead to the eventual 503 error when terraform attempted to use that ARN to create resources in my AWS account
- It was later discovered that even after this old ARN issue was resolved, the pipeline successfully ran to completion, however, deployment caches still seemed to have affected the final end point as follows:
    - The url of the ALB displayed a 503 error
    - More research would be needed to determine whether the artifact was being reproduced with a fault or the terraform stages were begin affected

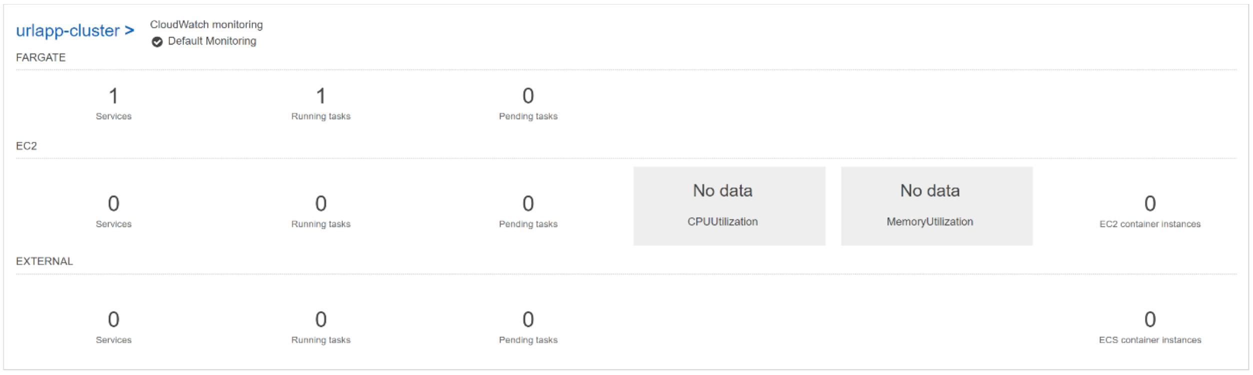


```
ubuntu@ip-172-31-35-5: ~/dep5_terra/workspace/Deployment05_main/intTerraform
ubuntu@ip-172-31-35-5:~/dep5_terra/workspace/Deployment05_main/intTerraform$ terraform destroy
var.aws_access_key
  Enter a value: AKIAT4F577X7SKKKR7GU

var.aws_secret_key
  Enter a value: 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

aws_eip.elastic-ip: Refreshing state... [id=eipalloc-0d9b5df290cee6579]
aws_cloudwatch_log_group.log-group: Refreshing state... [id=/ecs/url-logs]
aws_vpc.app_vpc: Refreshing state... [id=vpc-03c3e09fc6cfdd480]
aws_ecs_cluster.aws-ecs-cluster: Refreshing state... [id=arn:aws:ecs:us-east-2:751624437075:cluster/urlapp-cluster]
aws_ecs_task_definition.aws-ecs-task: Refreshing state... [id=url-task]
aws_security_group.ingress_app: Refreshing state... [id=sg-0beed08737beb956d]
aws_security_group.http: Refreshing state... [id=sg-0aaf215f844584fa6]
aws_route_table.public: Refreshing state... [id=rtb-02d02210d47791afa]
aws_route_table.private: Refreshing state... [id=rtb-09b44f923e5022d8b]
aws_internet_gateway.igw: Refreshing state... [id=igw-0c676a57095d74268]
aws_subnet.public_b: Refreshing state... [id=subnet-007fd3f00478f8394]
aws_subnet.private_a: Refreshing state... [id=subnet-022f5479fb0fab2f7]
aws_subnet.private_b: Refreshing state... [id=subnet-0b2b6119698554488]
aws_subnet.public_a: Refreshing state... [id=subnet-000c313e41c2aec05]
aws_route_table_association.private_a_subnet: Refreshing state... [id=rtbassoc-0a872c11461352c06]
aws_nat_gateway.ngw: Refreshing state... [id=nat-04e8a2539474fedb7]
aws_alb.url_app: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-2:751624437075:loadbalancer/app/url-lb/b1cb99e0156edcc2]
aws_route_table_association.public_a_subnet: Refreshing state... [id=rtbassoc-01f10ee462f561c26]
aws_route_table_association.public_b_subnet: Refreshing state... [id=rtbassoc-09b921a33775a501a]
aws_route.public_igw: Refreshing state... [id=r-rtb-02d02210d47791afa1080289494]
aws_route_table_association.private_b_subnet: Refreshing state... [id=rtbassoc-008da2c3ef14212b0]
aws_route.private_ngw: Refreshing state... [id=r-rtb-09b44f923e5022d8b1080289494]

│ Error: retrieving ALB (arn:aws:elasticloadbalancing:us-east-2:751624437075:loadbalancer/app/url-lb/b1cb99e0156edcc2): ValidationError: 'arn:aws:elasticloadbalancing:us-east-2:751624437075:loadbalancer/app/url-lb/b1cb99e01
56edcc2' is not a valid load balancer ARN
│       status code: 400, request id: 217c5604-737b-499e-aed5-adbc77f197bb
│
│   with aws_alb.url_app,
│   on ALB.tf line 16, in resource "aws_alb" "url_app":
│   16: resource "aws_alb" "url_app" {
│
╵
╷
│ Error: error reading ECS Cluster (arn:aws:ecs:us-east-2:751624437075:cluster/urlapp-cluster): InvalidParameterException: Identifier is for 751624437075. Your accountId is 266686430719
│
│   with aws_ecs_cluster.aws-ecs-cluster,
│   on main.tf line 9, in resource "aws_ecs_cluster" "aws-ecs-cluster":
│    9: resource "aws_ecs_cluster" "aws-ecs-cluster" {
│
╵
╷
│ Error: InvalidParameterException: Identifier is for 751624437075. Your accountId is 266686430719
│
│   with aws_ecs_task_definition.aws-ecs-task,
│   on main.tf line 26, in resource "aws_ecs_task_definition" "aws-ecs-task":
│   26: resource "aws_ecs_task_definition" "aws-ecs-task" {
│
╵
ubuntu@ip-172-31-35-5:~/dep5_terra/workspace/Deployment05_main/intTerraform$
```

After debugging cached artifacts

Successful ECS cluster

Note: 1 Running Task


urlapp-cluster >
CloudWatch monitoring
Default Monitoring

FARGATE

1 Services	1 Running tasks	0 Pending tasks			

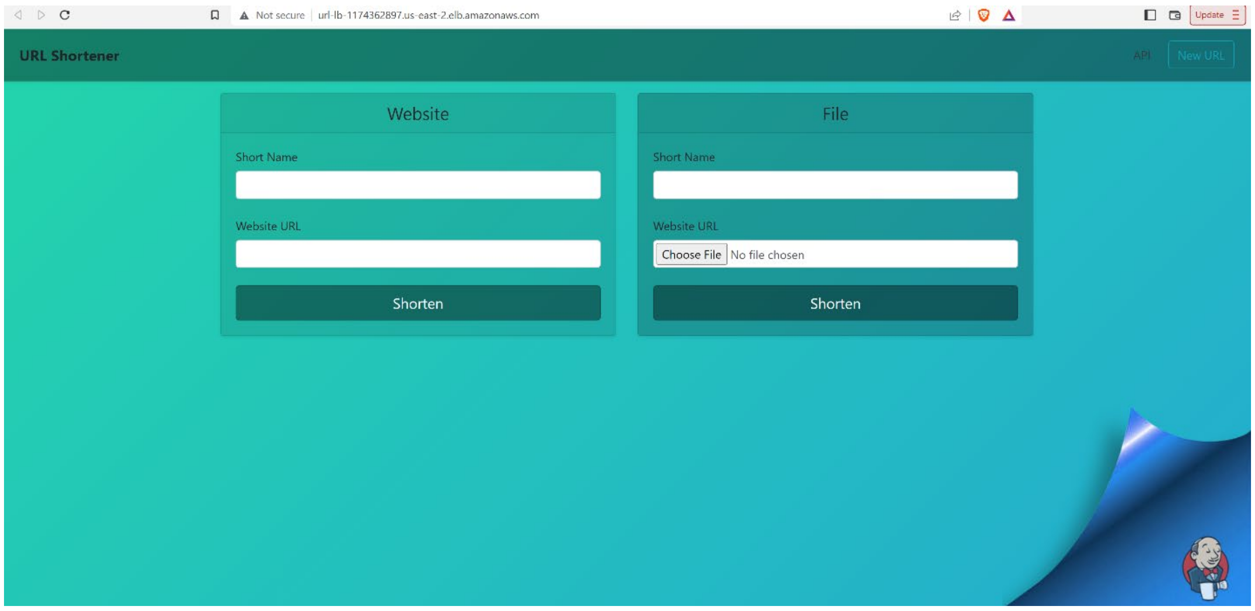
EC2

0 Services	0 Running tasks	0 Pending tasks	No data CPUUtilization	No data MemoryUtilization	0 EC2 container instances

EXTERNAL

0 Services	0 Running tasks	0 Pending tasks			0 ECS container instances

Successful final deployment


Not secure | url-lb-1174362897.us-east-2.elb.amazonaws.com

URL Shortener

Potential Improvements:

1. Added cache destruction to allow image build and final terraform implementation to run properly

Helpful Links/Resources:

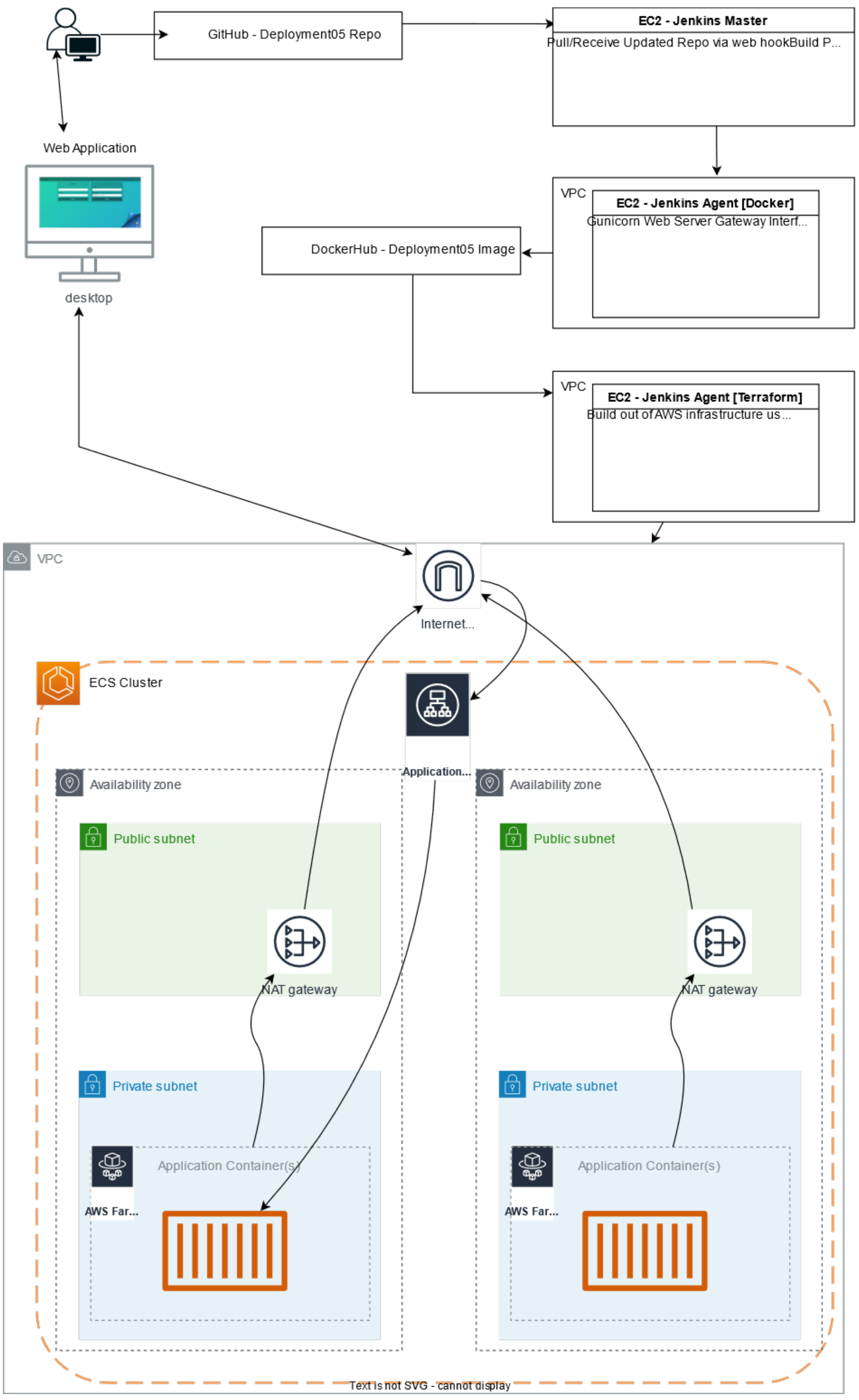
https://devopscube.com/install-configure-jenkins-2-0/

- Setting up Jenkins, Jenkins Agent directly, using docker, Kubernetes, etc.

Noteworthy Steps:

1. Create EC2 instances – Docker and Terraform Agents
    a. sudo apt install default-jre [for Agents]
2. Had to open up SSH ingress in security group of Agent servers as connection was failing even when the public IP of the Jenkins Master was specified as an authorized Source in the Inbound Rules table
3. Rather than modify PATH, etc., in order to run docker commands, "sudo" was used since ubuntu user is already part of sudoers list.

Final Layout:



GitHub - Deployment05 Repo
EC2 - Jenkins Master
Pull/Receive Updated Repo via web hookBuild P...
Web Application
VPC
EC2 - Jenkins Agent [Docker]
Gunicorn Web Server Gateway Interf...
DockerHub - Deployment05 Image
desktop
VPC
EC2 - Jenkins Agent [Terraform]
Build out of AWS infrastructure us...
VPC
Internet...
ECS Cluster
Application...
Availability zone
Availability zone
Public subnet
Public subnet
NAT gateway
NAT gateway
Private subnet
Private subnet
Application Container(s)
Application Container(s)
AWS Far...
AWS Far...
Text is not SVG - cannot display