

# Infrastructure as Code или Terraform-им AWS

Alex Burym





**Hi there.  
I'm Alex Burym**

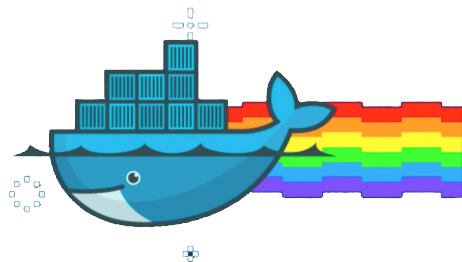


- 9+ years of diverse experience in IT
- CI/CD
- DevOps
- Cloud Computing
- Networking
- Process Automation

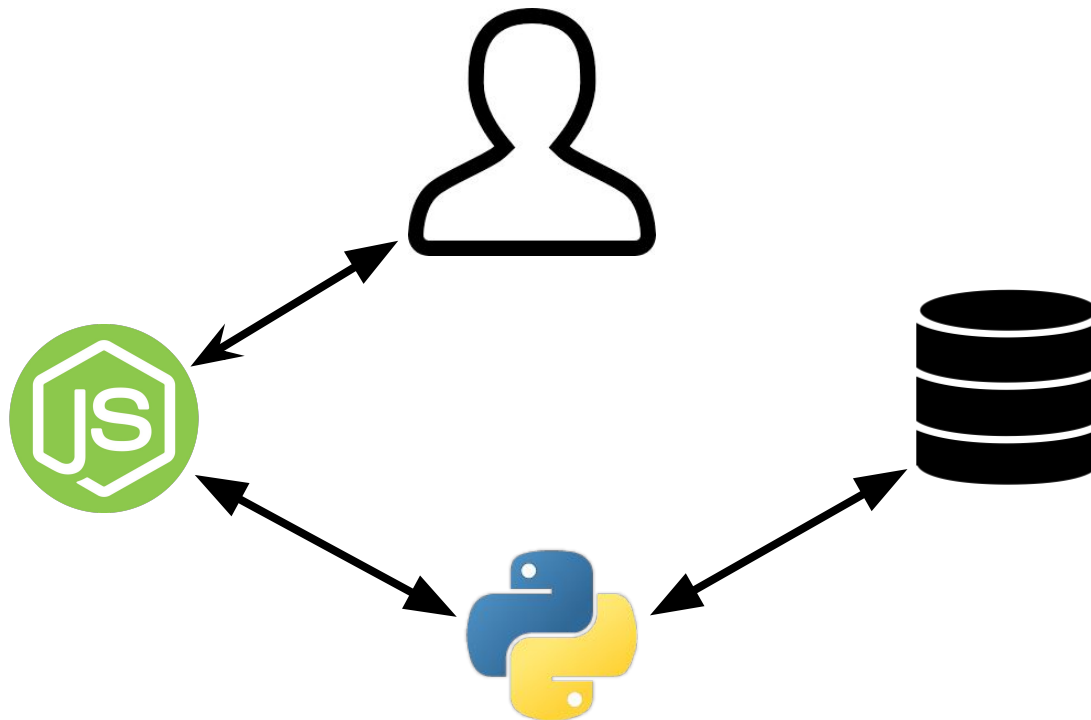
### **Contact Information:**

Email: [alex.burym@itechart-group.com](mailto:alex.burym@itechart-group.com)  
Skype: live:eng.zubr

- Typical web app?
- Why we need it?
- Just DO it!
- Something went terribly wrong...
- Infrastructure as Code?
- Tools!
- For those who play alone at home

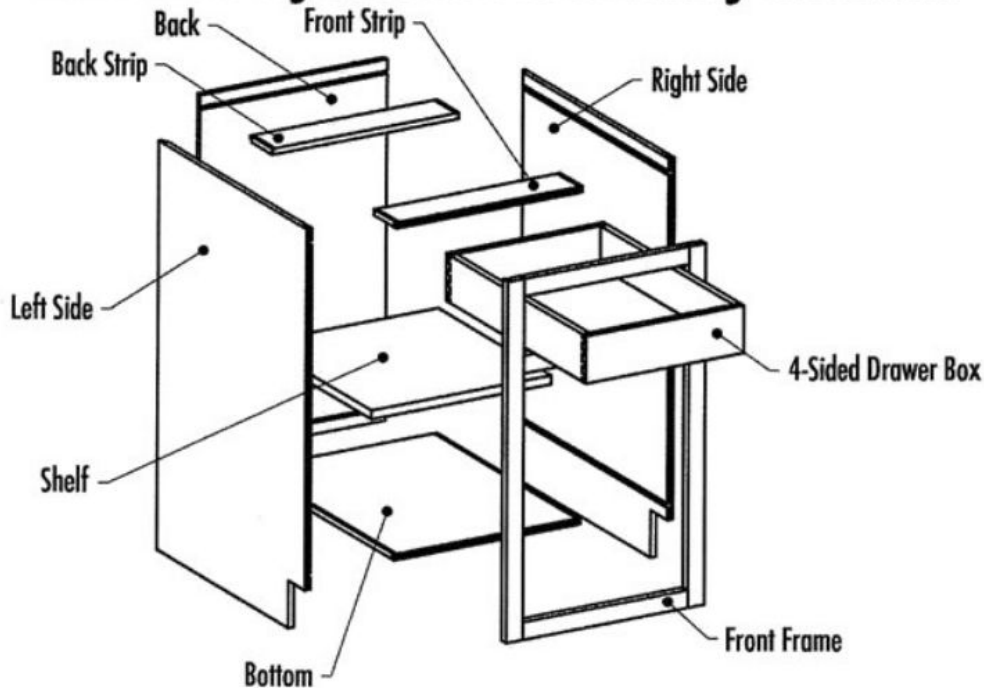


# Typical web app?

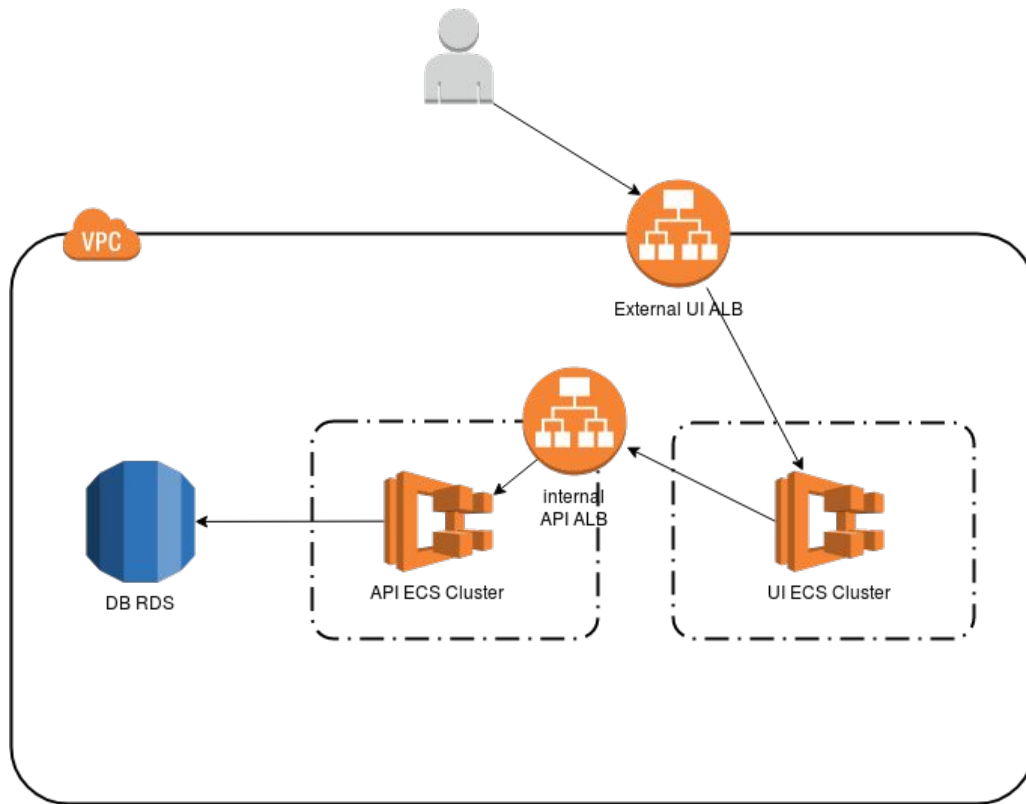


# Who really needs a manual!?

## Cabinet Systems Assembly Manual



# Typical web app?



# Once more!



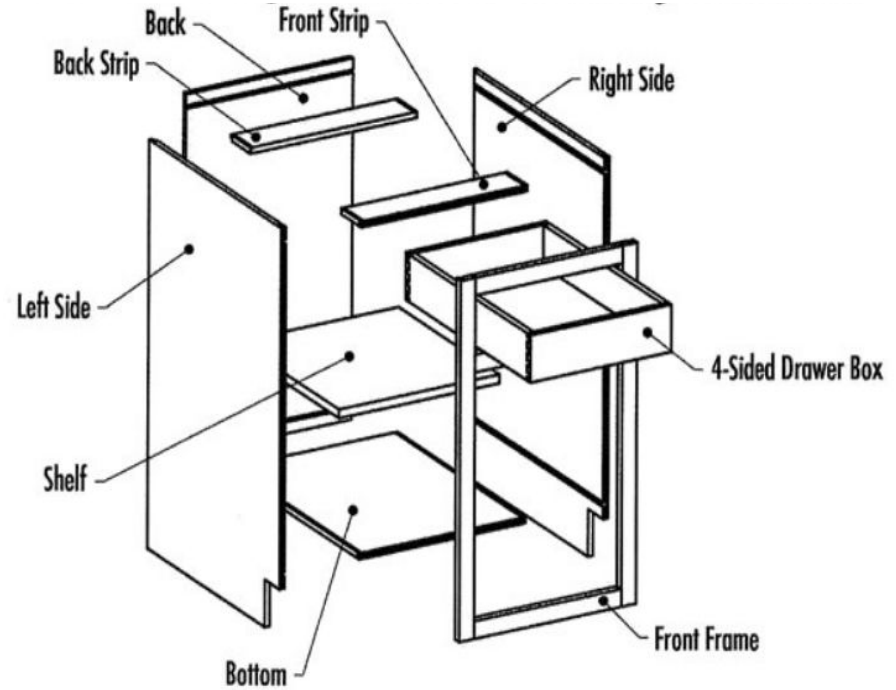
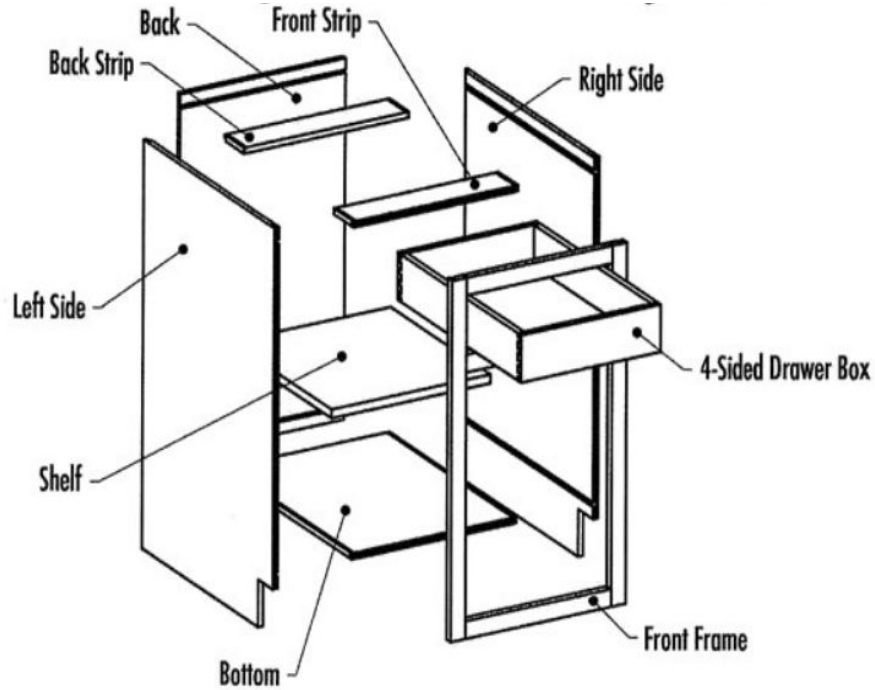
# Just DO it!

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR
<input type="checkbox"/>	New VPC for staging	vpc-05e26a8de28b76a6e	available	172.31.0.0/16
<input type="checkbox"/>	Old VPC	vpc-15130d73	available	172.31.0.0/16

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	Availability Zone
<input type="checkbox"/>	OldSubnet	subnet-b909fe95	available	vpc-15130d73   Old VPC	172.31.64.0/20	us-east-1a
<input type="checkbox"/>	Subnet4staging	subnet-03a5f2f6d...	available	vpc-05e26a8de28b76a6e   New VPC...	172.31.64.0/20	us-east-1a



# Once more!



1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of

Number of instances	<input type="text" value="1"/>	<a href="#">Launch into Auto Scaling Group</a>
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	<input type="text" value="vpc-05e26a8de28b76a6e   New VPC for staging"/>	<a href="#">Create new VPC</a>
Subnet	<input type="text" value="subnet-03a5f2f6d70783f8c   Subnet4staging   us-eas"/> <small>4091 IP Addresses available</small>	<a href="#">Create new subnet</a>
Auto-assign Public IP	<input type="text" value="Use subnet setting (Disable)"/>	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	<input type="text" value="Open"/>	<a href="#">Create new Capacity Reservation</a>
IAM role	<input type="text" value="None"/>	<a href="#">Create new IAM role</a>
Shutdown behavior	<input type="text" value="Stop"/>	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>	
Tenancy	<input type="text" value="Shared - Run a shared hardware instance"/> <small>Additional charges will apply for dedicated tenancy.</small>	

# Just DO it!

11

Name	Instance ID	Instance	Availability	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
New EC2	i-015889f2...	t2.nano	us-east-1a	running	2/2 checks ...	None	-	-
Old EC2	i-0651e73...	t2.nano	us-east-1b	running	2/2 checks ...	None	ec2-3-94-160-179.com...	3.94.160.179

Instance: **i-015889f2d7bc26cb3 (New EC2)** Private IP: 172.31.79.196

Description

Status Checks

Monitoring

Tags

Instance ID	i-015889f2d7bc26cb3
Instance state	running
Instance type	t2.nano
Elastic IPs	
Availability zone	us-east-1a
Security groups	<a href="#">launch-wizard-2</a> . <a href="#">view inbound rules</a> . <a href="#">view outbound rules</a>
Scheduled events	<a href="#">No scheduled events</a>
AMI ID	<a href="#">amzn-ami-hvm-2018.03.0.20181129-x86_64-gp2 (ami-0080e4c5bc078760e)</a>

Public DNS (IPv4)	-
IPv4 Public IP	-
IPv6 IPs	-
Private DNS	ip-172-31-79-196.ec2.internal
Private IPs	172.31.79.196
Secondary private IPs	
VPC ID	vpc-05e26a8de28b76a6e
Subnet ID	subnet-03a5f2f6d70783f8c

# ■ Something went terribly wrong...

12



А

ЧТО

НЕ

ТАК!?

# Something went terribly wrong...

13

[Subnets](#) > [Modify auto-assign IP settings](#)

## Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched

**Subnet ID** subnet-03a5f2f6d70783f8c

**Auto-assign IPv4**  Enable auto-assign public IPv4 address 

\* Required

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

## Step 3: Configure Instance Details

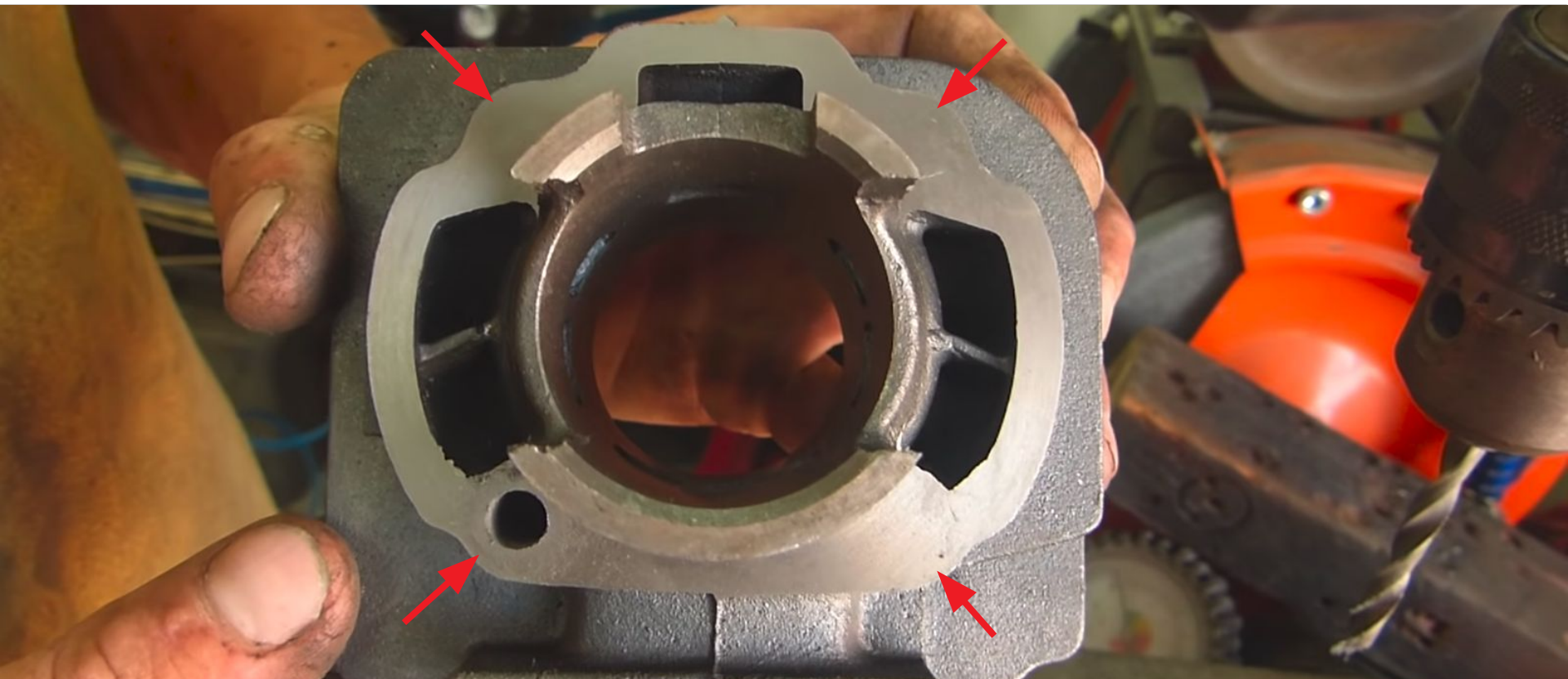
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of

Number of instances	<input type="text" value="1"/>	<a href="#">Launch into Auto Scaling Group</a>
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	<input type="text" value="vpc-05e26a8de28b76a6e   New VPC for staging"/>	<a href="#">Create new VPC</a>
Subnet	<input type="text" value="subnet-03a5f2f6d70783f8c   Subnet4staging   us-eas"/> 4091 IP Addresses available	<a href="#">Create new subnet</a>
Auto-assign Public IP	<input type="text" value="Use subnet setting (Disable)"/>	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	<input type="text" value="Open"/>	<a href="#">Create new Capacity Reservation</a>
IAM role	<input type="text" value="None"/>	<a href="#">Create new IAM role</a>
Shutdown behavior	<input type="text" value="Stop"/>	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <a href="#">Additional charges apply.</a>	
Tenancy	<input type="text" value="Shared - Run a shared hardware instance"/> <a href="#">Additional charges will apply for dedicated tenancy.</a>	



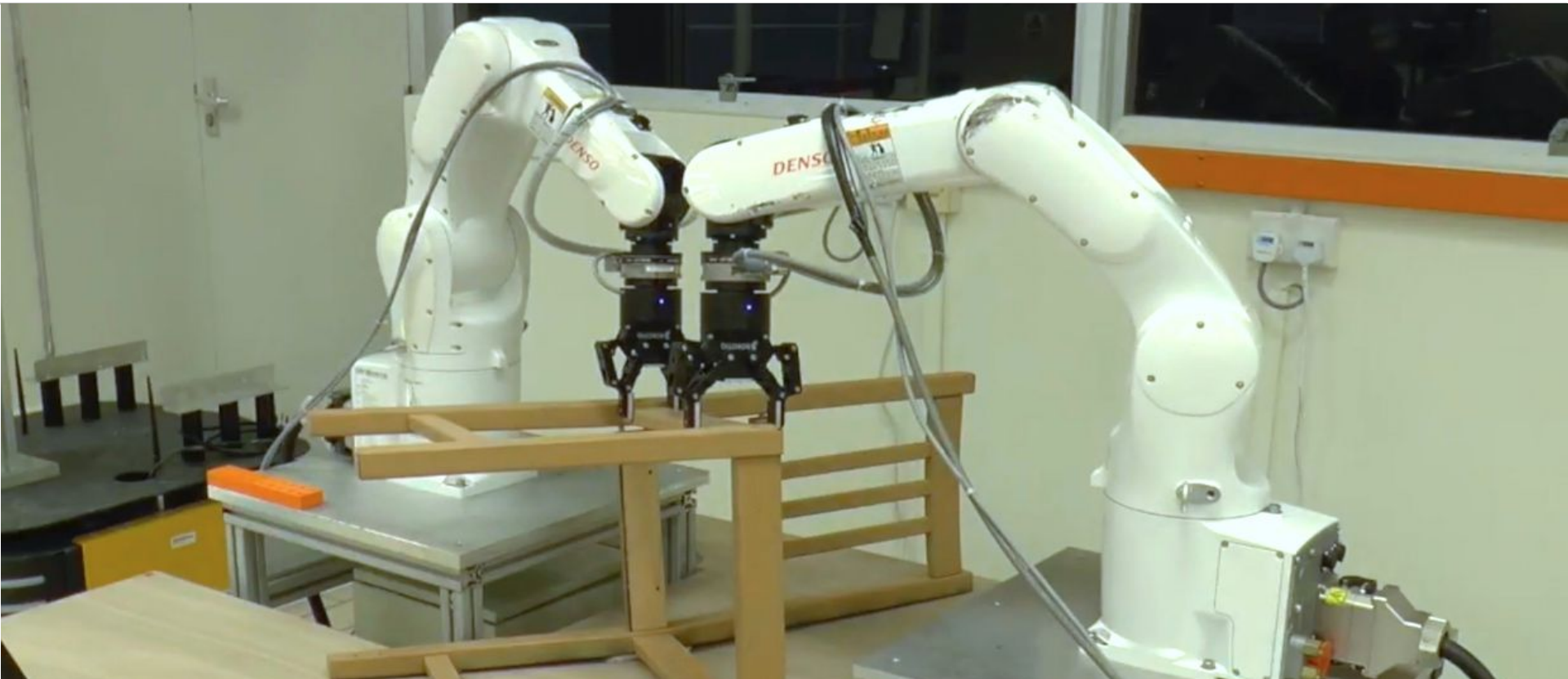
# ■ Something went terribly wrong...

15



# ■ Infrastructure as Code?

16



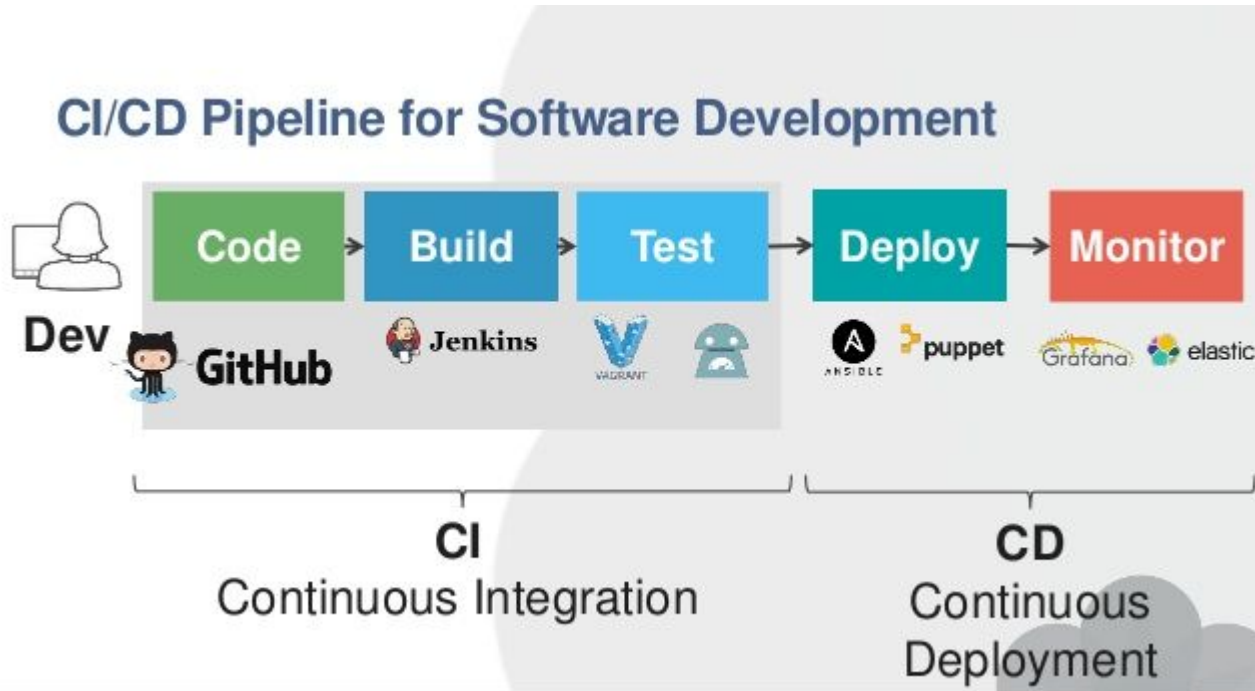


# Infrastructure as Code?

## Infrastructure as Code workflow

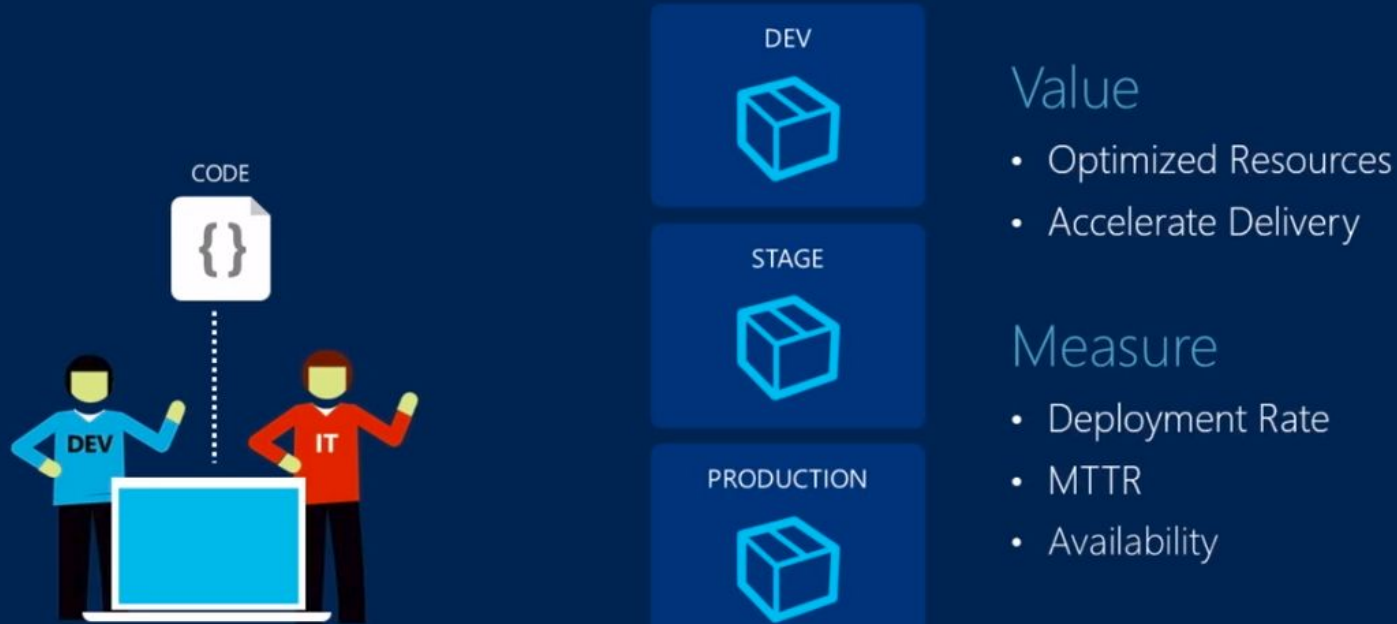


# Infrastructure as a Code?

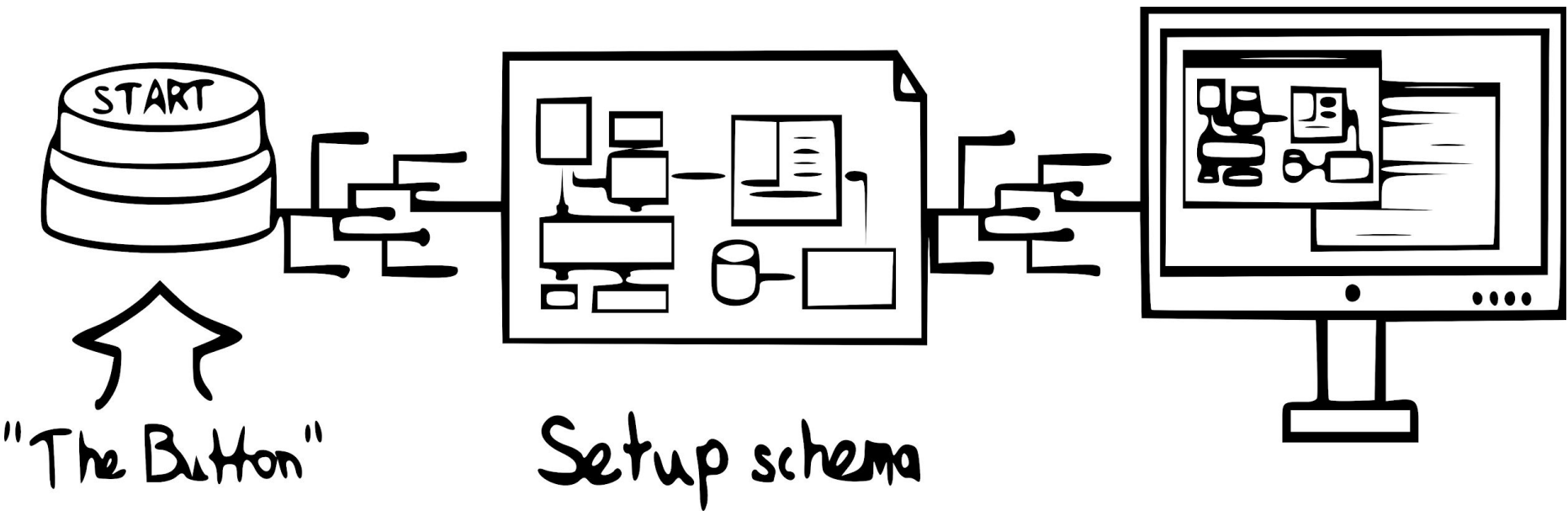


# Infrastructure as Code?

## Infrastructure & Configuration as Code



# Infrastructure as Code?





## AWS CloudFormation

Infrastructure Build Tools

[See AWS CloudFormation alternatives](#)



## Terraform

Infrastructure Build Tools

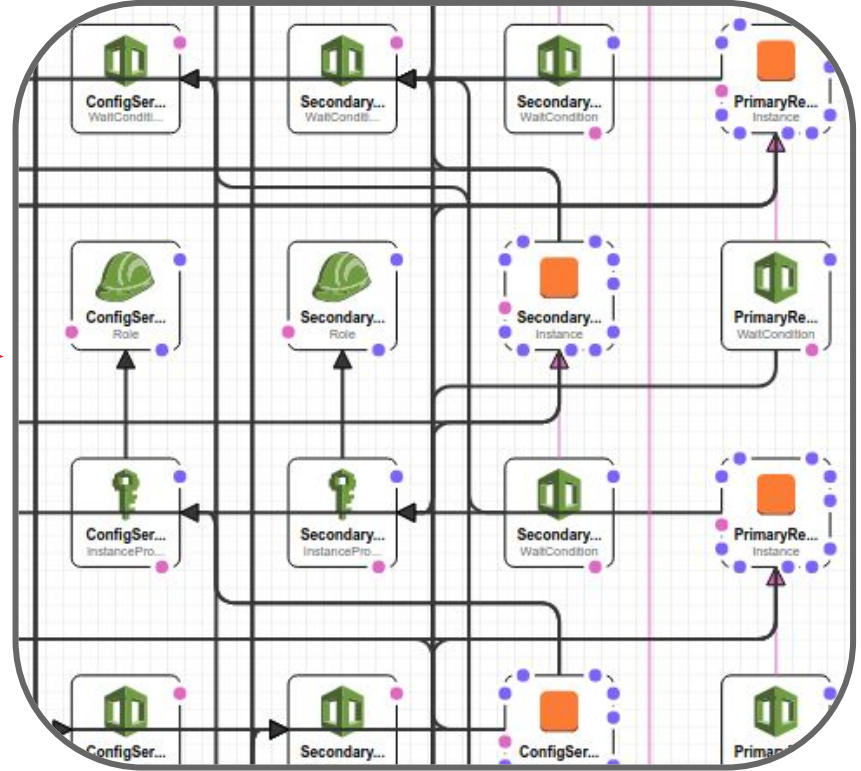
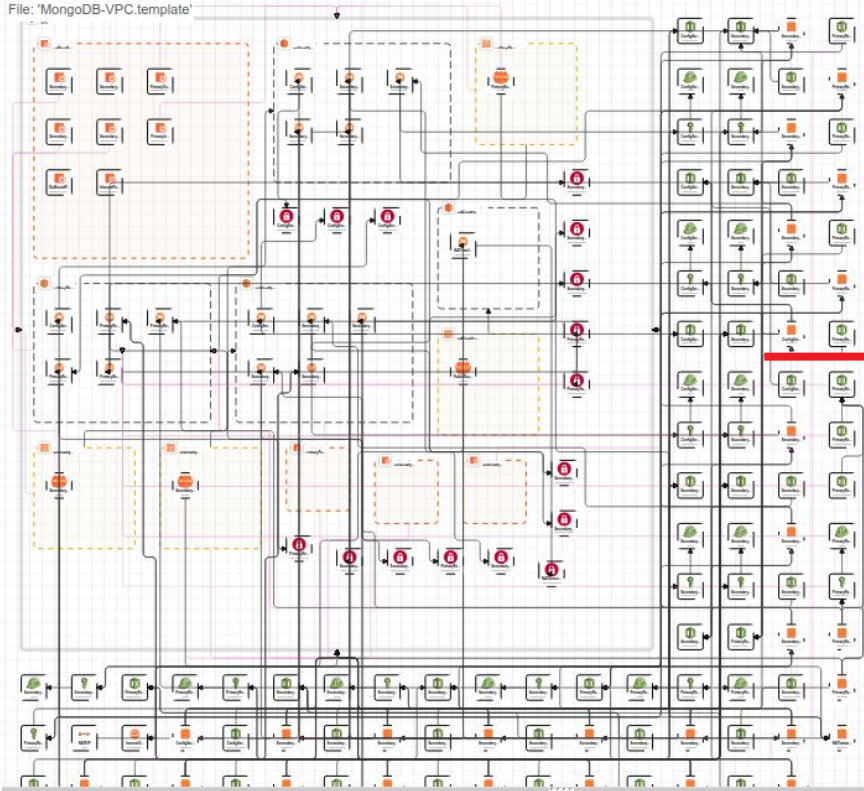
[See Terraform alternatives](#)

# CloudFormation?

```
8965     "Ref": "VPC"
8966   }
8967 ]
8968 ],
8969 },
8970 ">> config.sh\n",
8971 "echo ",
8972 {
8973   "Fn::Join": [
8974     "",
8975     [
8976       "export MongoDBVersion=",
8977       {
8978         "Ref": "MongoDBVersion"
8979       }
8980     ]
8981   ]
8982 },
8983 ">> config.sh\n",
8984 "./init.sh > install.log 2>&1 \n",
8985 "# Cleanup\n",
8986 "#rm -rf *\n",
8987 "# All is well so signal success\n",
8988 "/opt/aws/bin/cfn-signal -e 0 -r \"MongoDB
8989 {
8990   "Ref": "ConfigServer2WaitForNodeInstall
8991 },\n"
8992 ],\n"
8993 ]
8994 ],
8995 },
8996 ],
8997 "InstanceType": {
8998   "Ref": "ConfigServerInstanceType"
8999 },
9000 },
9001 "Condition": "CreateMinOneShard"
9002 }
9003 }
```



# CloudFormation?



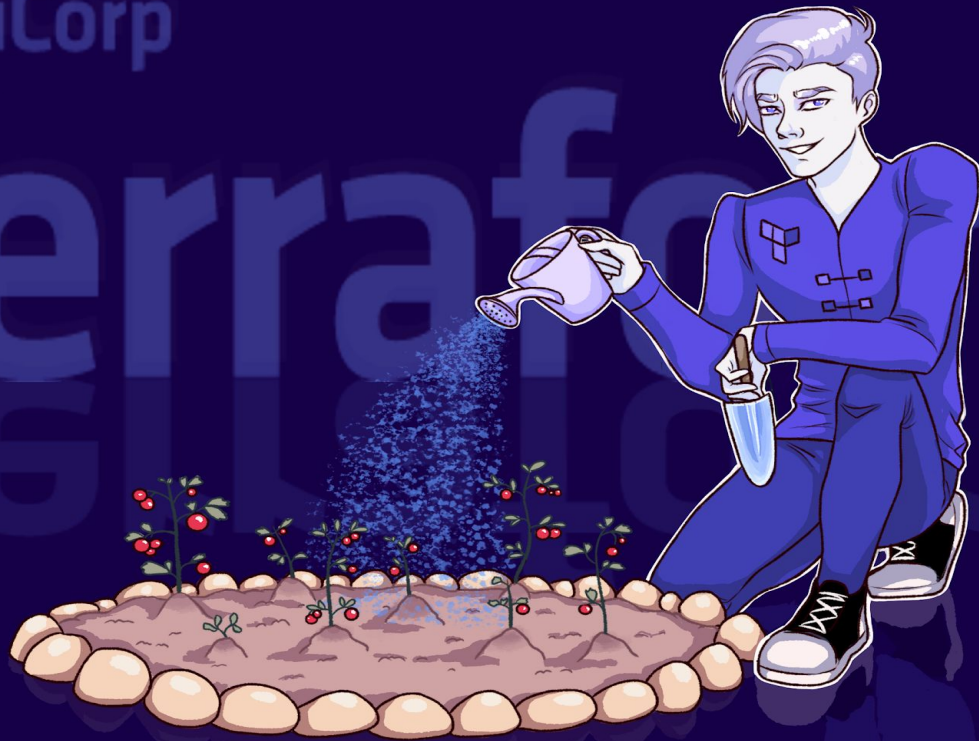


# Terraform?

HashiCorp



Terraform





# Terraform?

## Providers

- Major Cloud
- Cloud
- Infrastructure Software
- Network
- VCS
- Monitor & System Management
- Database
- Misc.
- Community
- Provisioners
- Modules
- Backends
- Plugins
- Internals

## Other Docs

- Download Terraform
- Introduction to Terraform
- Terraform Enterprise
- Guides and Whitepapers

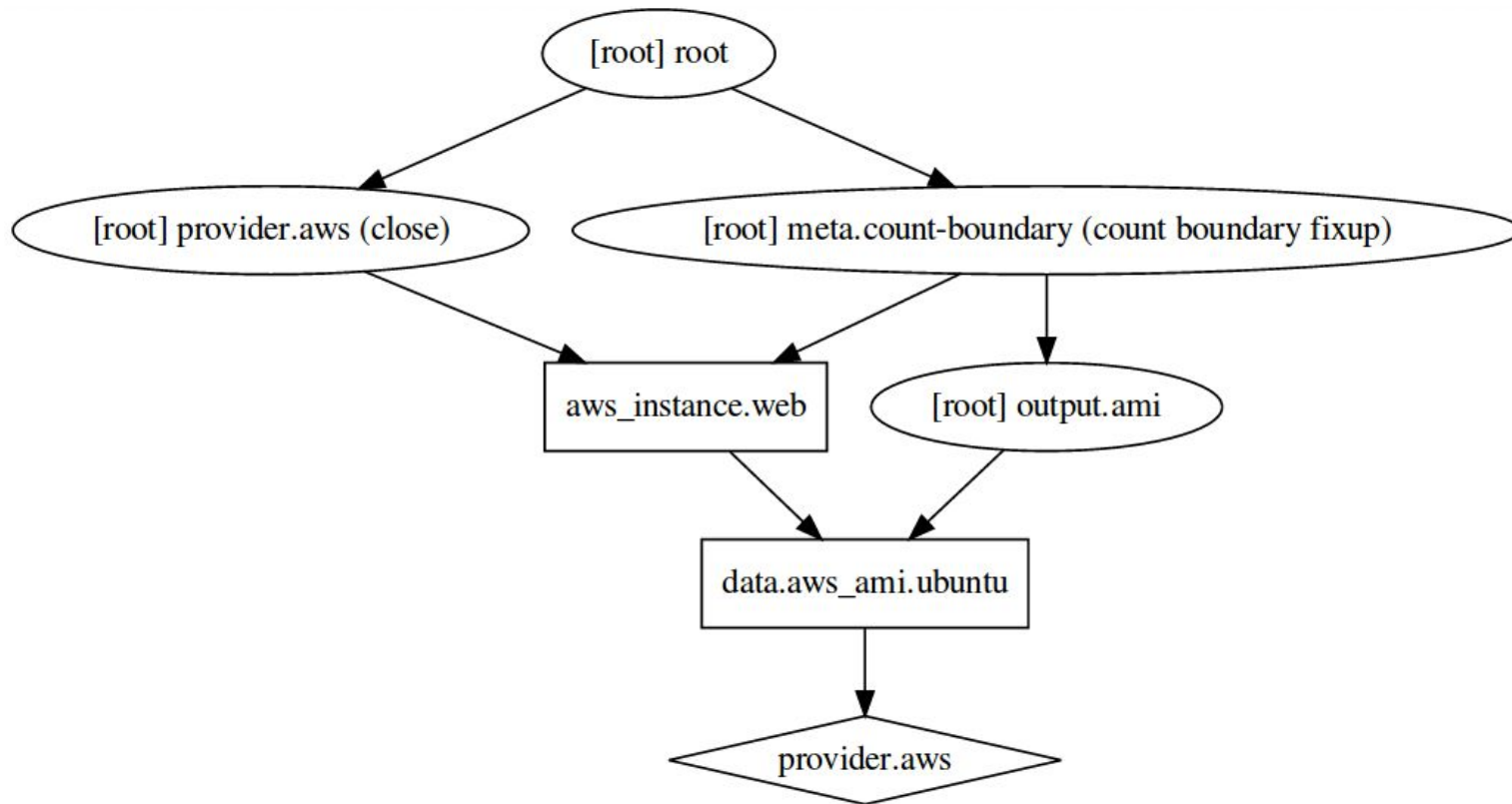
Use the navigation to the left to find available providers by type or scroll down to see all providers.

ACME	Alicloud	Archive	Naver Cloud	Netify	New Relic
Arukas	AWS	Azure Active Directory	Nomad	NS1	Null
Azure	Azure Stack	Bitbucket	Nutanix	1&1	OpenStack
Brightbox	CenturyLinkCloud	Chef	OpenTelekomCloud	OpsGenie	Oracle Cloud Infrastructure
Circonus	Cisco ASA	Cloudflare	Oracle Cloud Platform	Oracle Public Cloud	OVH
CloudScale.ch	CloudStack	Cobbler	Packet	PagerDuty	Palo Alto Networks
Consul	Datadog	DigitalOcean	PostgreSQL	PowerDNS	ProfitBricks
DNS	DNSMadeEasy	DNSimple	RabbitMQ	Rancher	Random
Docker	Dyn	External	RightScale	Rundeck	RunScope
F5 BIG-IP	Fastly	FlexibleEngine	Scaleway	Selectel	Skytap
GitHub	Gitlab	Google Cloud Platform	SoftLayer	StatusCake	Spotinst
Grafana	Hedvig	Helm	TelefonicaOpenCloud	Template	TencentCloud
Heroku	Hetzner Cloud	HTTP	Terraform	Terraform Enterprise	TLS
HuaweiCloud	Icinga2	Ignition	Triton	UCloud	UltraDNS
InfluxDB	Kubernetes	Librato	Vault	VMware vCloud Director	VMware NSX-T
Linode	Local	Logentries	VMware vSphere	Yandex	
LogicMonitor	Mailgun	MySQL			
Naver Cloud	Netlify	New Relic			

# Terraform?

```
# Create a web server  
resource "aws_instance" "web" {  
  ami           = "${data.aws_ami.ubuntu.id}"  
  instance_type = "t3.nano"  
  
  tags = {  
    Name = "My-web-server"  
  }  
}
```

# Terraform?



# Terraform?

```
+ create
```

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

```
aws_instance.web: still creating... (10s elapsed)
```

```
aws_instance.web: Creation complete after 18s (ID: i-02457a7e63d997e56)
```

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

```
- aws_instance.web
```

```
tags.Name: "My-super-web-server" => "My-MEGA-super-web-server"
```

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

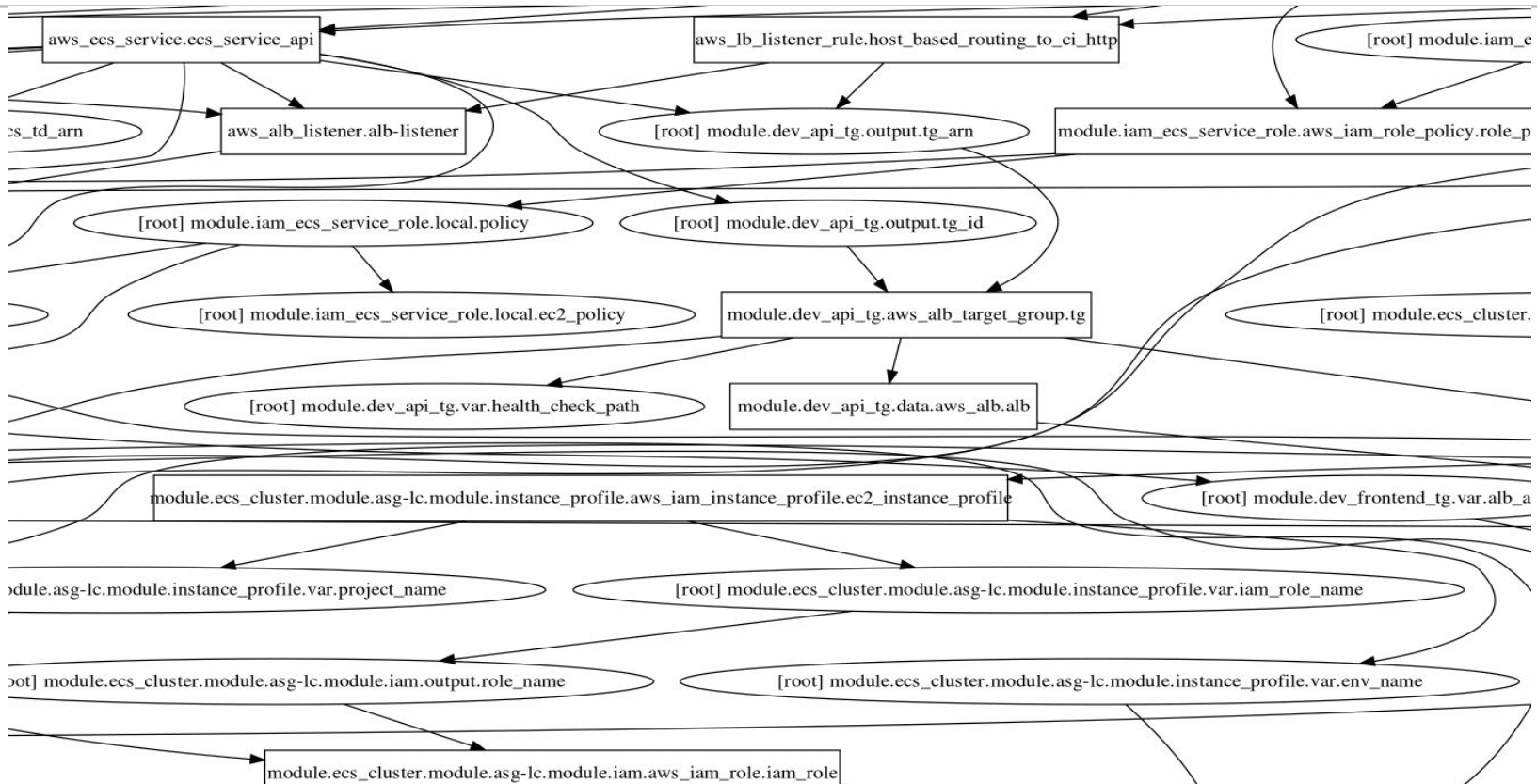
```
- aws_instance.web
```

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

```
aws_instance.web: Destruction complete after 15s
```

```
Destroy complete! Resources: 1 destroyed.
```

# Terraform?



# For those who play alone at home

30



msksborka.ru

# ■ For those who play alone at home

31

<https://aws.amazon.com/free/>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>

<https://www.terraform.io/downloads.html>

<https://learn.hashicorp.com/terraform/getting-started/install.html#installing-terraform>

# ■ For those who play alone at home

32

terraform init

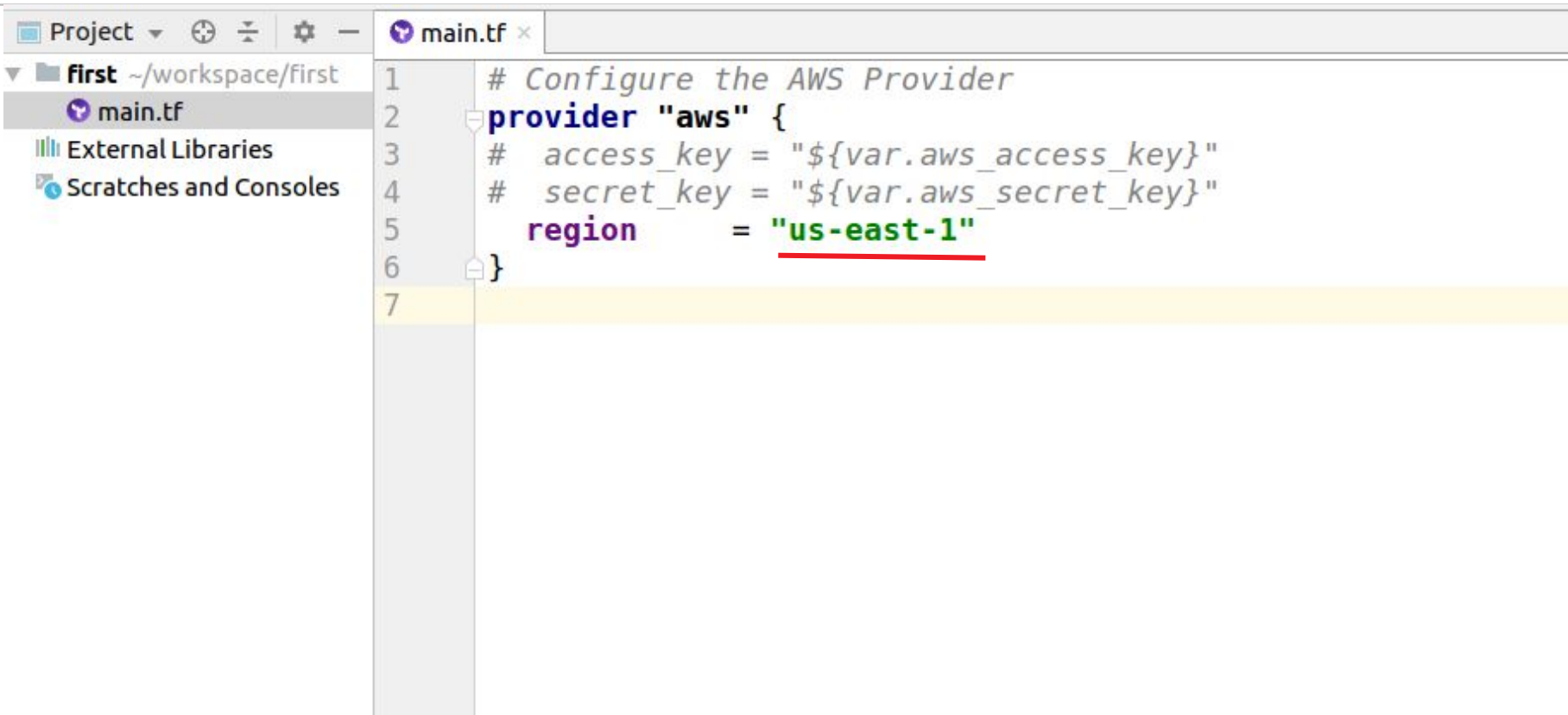
terraform plan terraform apply

terraform destroy



# For those who play alone at home

33



The image shows a screenshot of an IDE window with a file named 'main.tf' open. The left sidebar shows a project structure with a folder 'first' containing 'main.tf'. The main editor area displays the following Terraform code:

```
1 # Configure the AWS Provider
2 provider "aws" {
3   # access_key = "${var.aws_access_key}"
4   # secret_key = "${var.aws_secret_key}"
5   region      = "us-east-1"
6 }
7
```

The code is color-coded: 'provider' is blue, 'aws' is purple, 'region' is purple, and 'us-east-1' is green and underlined with a red line. Line 7 is highlighted in yellow.

# For those who play alone at home

```
eng_zubr@T520:~/workspace/first$ terraform init
```

## Initializing provider plugins...

- Checking for available provider plugins on <https://releases.hashicorp.com...>
- Downloading plugin for provider "aws" (2.1.0)...

The following providers do not have any version constraints in configuration, so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, it is recommended to add `version = "..."` constraints to the corresponding provider blocks in configuration, with the constraint strings suggested below.

```
* provider.aws: version = "~> 2.1"
```

**Terraform has been successfully initialized!**

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
eng_zubr@T520:~/workspace/first$ █
```

# For those who play alone at home

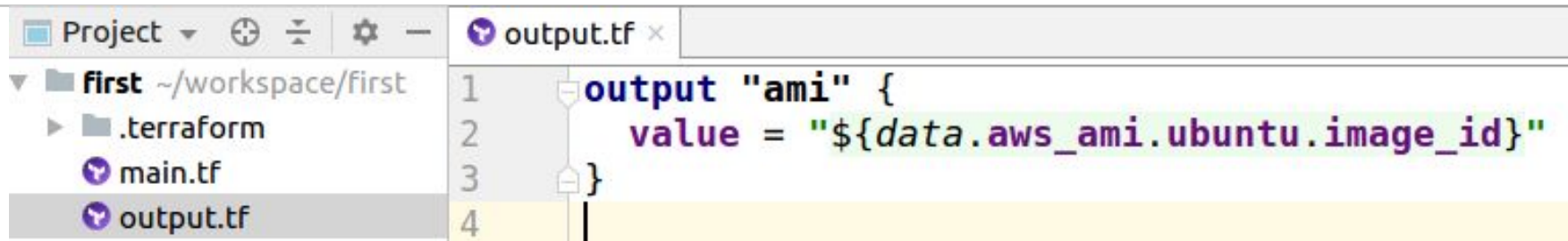
35

main.tf x

```
1  # Configure the AWS Provider
2  provider "aws" {region = "us-east-1"}
7
8  # Get ami id
9  data "aws_ami" "ubuntu" {
10     most_recent = true
11     filter {
12         name = "name"
13         values = ["ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"]
14     }
15
16     filter {
17         name = "virtualization-type"
18         values = ["hvm"]
19     }
20     owners = ["099720109477"] # Canonical
21 }
22
```

# For those who play alone at home

36



```
Project ▾ ⊕ ⊖ ⚙ — output.tf ×
├── first ~/workspace/first
│   ├── .terraform
│   ├── main.tf
│   └── output.tf
1  output "ami" {
2      value = "${data.aws_ami.ubuntu.image_id}"
3  }
4  |
```

```
eng_zubr@T520:~/workspace/first$ terraform apply
data.aws_ami.ubuntu: Refreshing state...
```

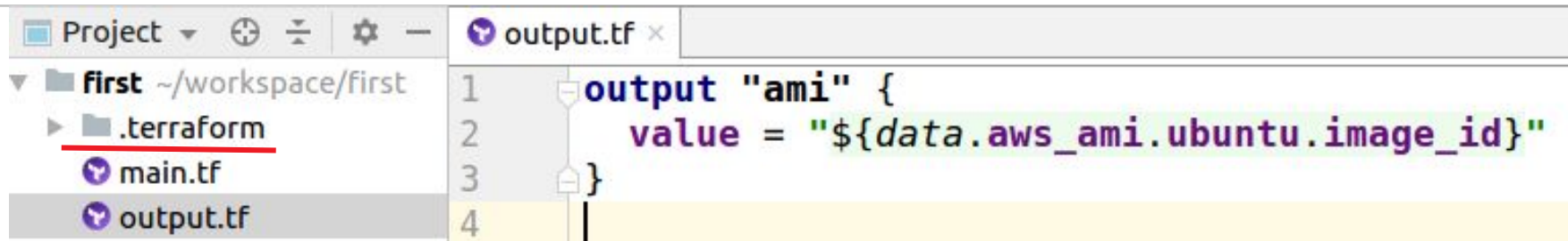
```
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

```
Outputs:
```

```
ami = ami-067d76d1273765ecf
```

# For those who play alone at home

37



```
Project ▾ ⊕ ⊖ ⚙ — output.tf ×
▼ first ~/workspace/first
  ▸ .terraform
    Ⓞ main.tf
    Ⓞ output.tf
1  output "ami" {
2      value = "${data.aws_ami.ubuntu.image_id}"
3  }
4  |
```

```
eng_zubr@T520:~/workspace/first$ terraform apply
data.aws_ami.ubuntu: Refreshing state...
```

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

```
Outputs:
```

```
ami = ami-067d76d1273765ecf
```

# For those who play alone at home

38

```
1 {
2   "version": 3,
3   "terraform_version": "0.11.7",
4   "serial": 1,
5   "lineage": "dbe9e7a1-033c-5bde-809a-376e3383765c",
6   "modules": [
7     {
8       "path": [
9         "root"
10      ],
11      "outputs": {
12        "ami": {
13          "sensitive": false,
14          "type": "string",
15          "value": "ami-067d76d1273765ecf"
16        }
17      },
18      "resources": {
19        "data.aws_ami.ubuntu": {
20          "type": "aws_ami",
21          "depends_on": [],
22          "primary": {
23            "id": "ami-067d76d1273765ecf",
```



# For those who play alone at home

39

main.tf ×

```
1  # Configure the AWS Provider
2  provider "aws" {region = "us-east-1"}
7
8  # Get ami id
9  data "aws_ami" "ubuntu" {...}
22
23 # Create a web server
24 resource "aws_instance" "web" {
25     ami = "${data.aws_ami.ubuntu.id}"
26     instance_type = "t3.nano"
27
28     tags = {
29         Name = "My-super-web-server"
30     }
31 }
32
```

# For those who play alone at home

40

```
eng_zubr@T520:~/workspace/first$ terraform plan  
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this plan, but will not be  
persisted to local or remote state storage.
```

```
data.aws_ami.ubuntu: Refreshing state...
```

-----

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:

```
+ aws_instance.web
```

```
id: <computed>  
ami: "ami-067d76d1273765ecf"  
... <computed>
```



# For those who play alone at home

41

```
subnet_id:                <computed>
tags.%:                   "1"
tags.Name:                "My-super-web-server"
tenancy:                  <computed>
volume_tags.%:           <computed>
vpc_security_group_ids.#: <computed>
```

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

-----

Note: You didn't specify an "-out" parameter to save this plan, so Terraform can't guarantee that exactly these actions will be performed if "terraform apply" is subsequently run.

# For those who play alone at home

42

```
eng_zubr@T520:~/workspace/first$ terraform apply  
data.aws_ami.ubuntu: Refreshing state...
```

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

+ create

```
tags.%:                               ↓  
tags.Name:                             "My-super-web-server"  
tenancy:                                <computed>  
volume_tags.%:                          <computed>  
vpc_security_group_ids.#:                <computed>
```

**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:

# For those who play alone at home

43

```
security_groups.#:      "" => "<computed>"
source_dest_check:     "" => "true"
subnet_id:             "" => "<computed>"
tags.%:                "" => "1"
tags.Name:             "" => "My-super-web-server"
tenancy:               "" => "<computed>"
volume_tags.%:        "" => "<computed>"
vpc_security_group_ids.#: "" => "<computed>"
aws_instance.web: Still creating... (10s elapsed)
aws_instance.web: Creation complete after 18s (ID: i-02457a7e63d997e56)
```

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

Outputs:

ami = ami-067d76d1273765ecf

# For those who play alone at home

44

Launch Instance

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance	Availability	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
My-super-web-server	i-036dbee...	t3.nano	us-east-1a	running	Initializing	None	ec2-3-92-2-196.co

Instance: **i-036dbee65a39f867a (My-super-web-server)** Public DNS: [ec2-3-92-2-196.compute-1.amazonaws.com](https://ec2-3-92-2-196.compute-1.amazonaws.com)

Description | Status Checks | Monitoring | Tags

Instance ID	i-036dbee65a39f867a	Public DNS (IPv4)	ec2-3-92-2-196.com
Instance state	running	IPv4 Public IP	3.92.2.196
Instance type	t3.nano	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-78-201.ec
Availability zone	us-east-1a	Private IPs	172.31.78.201
Security groups	<a href="#">default</a> . <a href="#">view inbound rules</a> . <a href="#">view outbound rules</a>	Secondary private IPs	
Scheduled events	<a href="#">No scheduled events</a>	VPC ID	<a href="#">vpc-15130d73</a>

# For those who play alone at home

45

```
main.tf x
1  # Configure the AWS Provider
2  provider "aws" {region = "us-east-1"}
7
8  # Get ami id
9  data "aws_ami" "ubuntu" {...}
22
23 # Create a web server
24 resource "aws_instance" "web" {
25     ami = "${data.aws_ami.ubuntu.id}"
26     instance_type = "t3.nano"
27
28     tags = {
29         Name = "My-MEGA-super-web-server"
30     }
31 }
```



# For those who play alone at home

46

```
eng_zubr@T520:~/workspace/first$ terraform apply
data.aws_ami.ubuntu: Refreshing state...
aws_instance.web: Refreshing state... (ID: i-036dbea65a39f867a)
```

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

```
~ update in-place
```

Terraform will perform the following actions:

```
~ aws_instance.web
```

```
tags.Name: "My-super-web-server" => "My-MEGA-super-web-server"
```

**Plan:** 0 to add, 1 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

# For those who play alone at home

47

```
aws_instance.web: Modifying... (ID: i-036dbea65a39f867a)
  tags.Name: "My-super-web-server" => "My-MEGA-super-web-server"
aws_instance.web: Modifications complete after 7s (ID: i-036dbea65a39f867a)
```

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

Outputs:

ami = ami-067d76d1273765ecf

<input type="checkbox"/>	Name	Instance ID	Instance	Availability	Instance State	Status Checks
<input type="checkbox"/>	My-MEGA-super-web-server	i-036dbea...	t3.nano	us-east-1a	<span style="color: green;">●</span> running	<span style="color: green;">✔</span> 2/2 checks ...

Instance: [i-036dbea65a39f867a](#) (My-MEGA-super-web-server)    Public DNS: ec2-3-92-2-196.compute-1.amaz



# For those who play alone at home

48

```
main.tf x
1  # Configure the AWS Provider
2  provider "aws" {region = "us-east-1"}
7
8  # Get ami id
9  data "aws_ami" "ubuntu" {
10     most_recent = true
11     filter {
12         name = "name"
13         values = ["ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-*"]
14     }
15
16     filter {
17         name = "virtualization-type"
18         values = ["hvm"]
19     }
20     owners = ["099720109477"] # Canonical
21 }
22
23 # Create a web server
24 resource "aws_instance" "web" {...}
```

# For those who play alone at home

49

```
eng_zubr@T520:~/workspace/first$ terraform apply
data.aws_ami.ubuntu: Refreshing state...
aws_instance.web: Refreshing state... (ID: i-036dbea65a39f867a)
```

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

**-/+** destroy and then create replacement

Terraform will perform the following actions:

**-/+** `aws_instance.web` (new resource required)

```
id: "i-036dbea65a39f867a" => <computed> (forces new resource)
ami: "ami-067d76d1273765ecf" => "ami-0c835d91df905128e" (forces new resource)
arn: "arn:aws:ec2:us-east-1:014173831091:instance/i-036dbea65a39f867a" => <computed>
associate_public_ip_address: "true" => <computed>
availability_zone: "us-east-1a" => <computed>
```

**Plan:** 1 to add, 0 to change, 1 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: █

# For those who play alone at home

50

```
aws_instance.web: still creating... (10s elapsed)
aws_instance.web: Creation complete after 18s (ID: i-058e79a498dce62d3)
```

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

Outputs:

ami = ami-0c835d91df905128e

<input type="checkbox"/>	My-MEGA-super-web-server	i-036ddea...	t3.nano	us-east-1a	<span style="color: red;">●</span> terminated	
<input checked="" type="checkbox"/>	My-MEGA-super-web-server	i-058e79a...	t3.nano	us-east-1a	<span style="color: green;">●</span> running	<span style="color: green;">✔</span> 2/2 checks

Instance: **i-058e79a498dce62d3 (My-MEGA-super-web-server)** Public DNS: ec2-18-234-208-64.compute-

Description

Status Checks

Monitoring

Tags

Instance ID	i-058e79a498dce62d3
Instance state	running
Instance type	t3.nano
Elastic IPs	
Availability zone	us-east-1a
Security groups	<a href="#">default</a> · <a href="#">view inbound rules</a> · <a href="#">view outbound rules</a>
Scheduled events	<a href="#">No scheduled events</a>

AMI ID	<a href="#">ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20190313 (ami-0c835d91df905128e)</a>
--------	---

# For those who play alone at home

51

```
eng_zubr@T520:~/workspace/first$ terraform destroy
data.aws_ami.ubuntu: Refreshing state...
aws_instance.web: Refreshing state... (ID: i-058e79a498dce62d3)
```

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

```
- destroy
```

Terraform will perform the following actions:

```
- aws_instance.web
```

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

**Do you really want to destroy?**

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

# For those who play alone at home

52

```
aws_instance.web: Destroying... (ID: i-058e79a498dce62d3)
aws_instance.web: Still destroying... (ID: i-058e79a498dce62d3, 10s elapsed)
aws_instance.web: Destruction complete after 15s
```

Destroy complete! Resources: 1 destroyed.

<input type="checkbox"/>	My-MEGA-super-web-server	i-036dbea...	t3.nano	us-east-1a	 terminated
<input checked="" type="checkbox"/>	My-MEGA-super-web-server	i-058e79a...	t3.nano	us-east-1a	 terminated

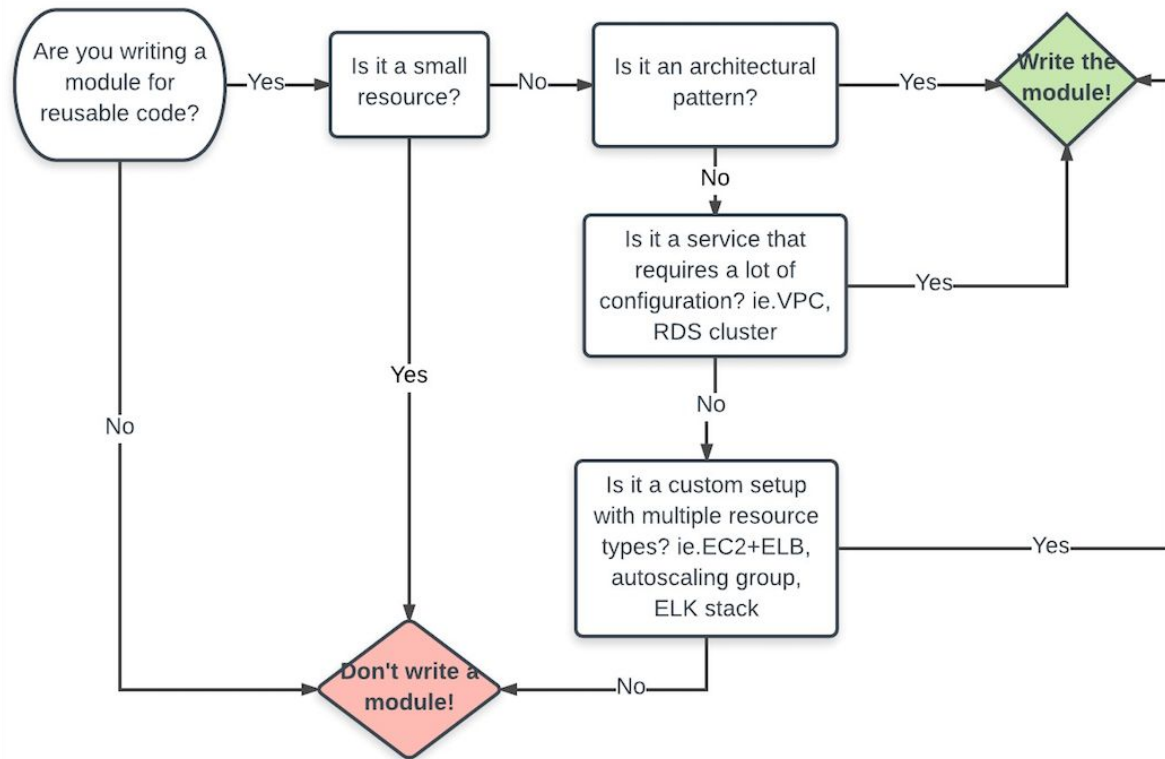
# For those who play alone at home

53

```
23 # Create a web server
24 ► resource "aws_instance" "web" {
25     count = 2
26     ami           = "${data.aws_ami.ubuntu.id}"
27     instance_type = "t3.nano"
28
29     tags = {
30         Name = "My-web-server"
31     }
32 }
```

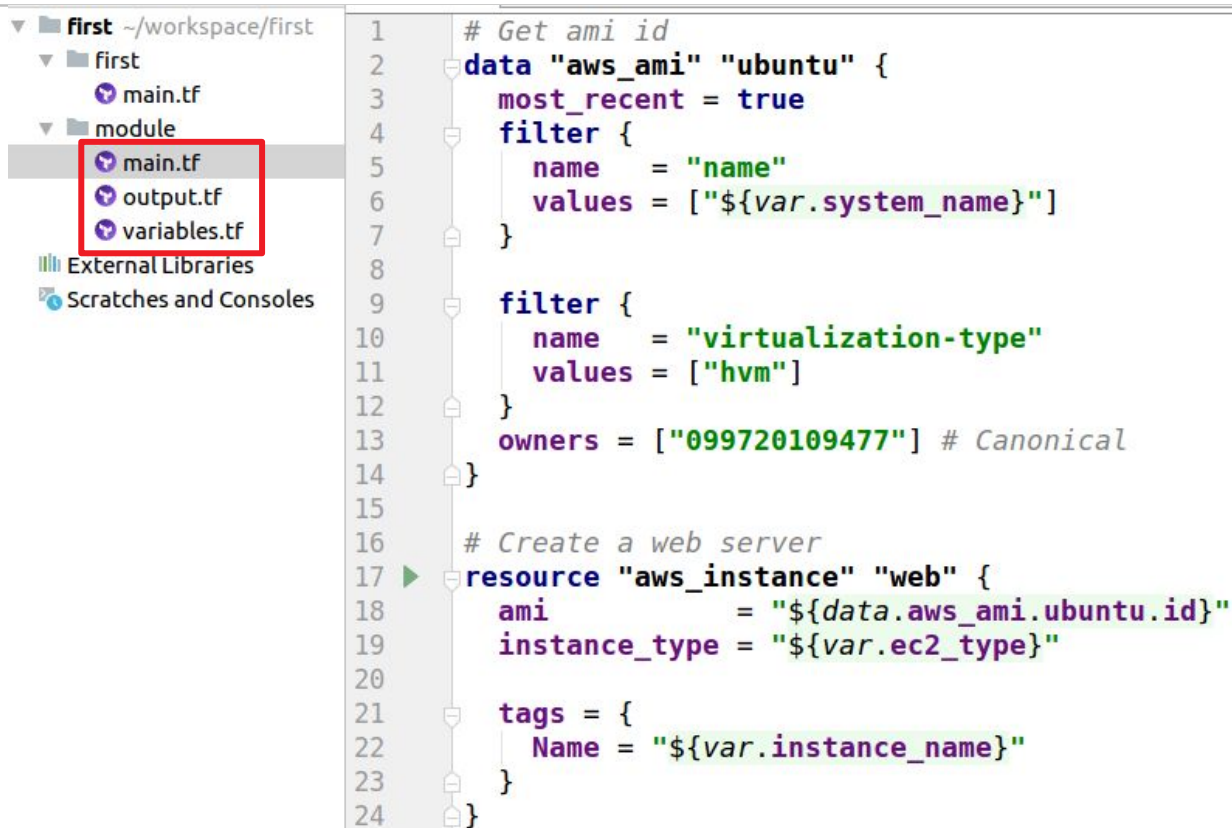


# For those who play alone at home



# For those who play alone at home

55



```
1  # Get ami id
2  data "aws_ami" "ubuntu" {
3    most_recent = true
4    filter {
5      name     = "name"
6      values  = ["${var.system_name}"]
7    }
8
9    filter {
10     name     = "virtualization-type"
11     values  = ["hvm"]
12   }
13   owners = ["099720109477"] # Canonical
14 }
15
16 # Create a web server
17 resource "aws_instance" "web" {
18   ami           = "${data.aws_ami.ubuntu.id}"
19   instance_type = "${var.ec2_type}"
20
21   tags = {
22     Name = "${var.instance_name}"
23   }
24 }
```

# For those who play alone at home

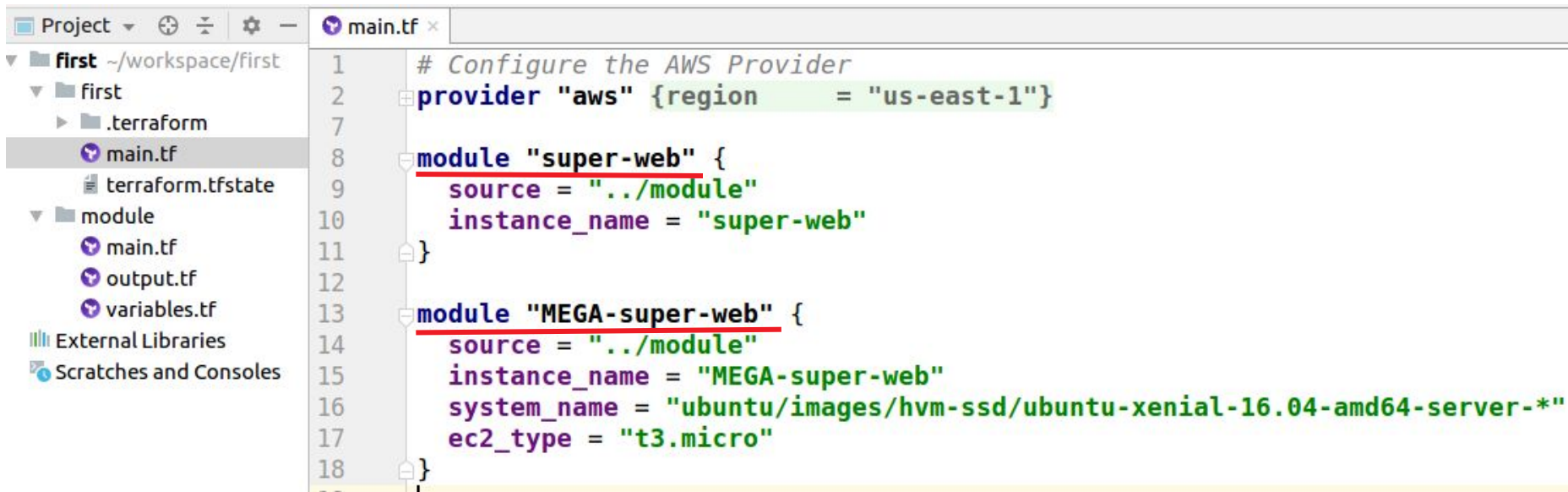
56

variables.tf x

```
1  variable "instance_name" {
2      description = "name of instance"
3      type = "string"
4  }
5
6  variable "system_name" {
7      description = "system name for filter"
8      type = "string"
9      default = "ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-*"
10 }
11
12 variable "ec2_type" {
13     description = "type of ec2 instance"
14     type = "string"
15     default = "t3.nano"
16 }
```

# For those who play alone at home

57



```
1  # Configure the AWS Provider
2  provider "aws" {region = "us-east-1"}
7
8  module "super-web" {
9      source = "../module"
10     instance_name = "super-web"
11 }
12
13 module "MEGA-super-web" {
14     source = "../module"
15     instance_name = "MEGA-super-web"
16     system_name = "ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-*"
17     ec2_type = "t3.micro"
18 }
```

# For those who play alone at home

58

```
eng_zubr@T520:~/workspace/first/first$ terraform init
```

## **Initializing modules...**

- module.super-web  
Getting source "../module"
- module.MEGA-super-web  
Getting source "../module"

## **Initializing provider plugins...**

- Checking for available provider plugins on <https://releases.hashicorp.com>...
- Downloading plugin for provider "aws" (2.1.0)...

The following providers do not have any version constraints in configuration, so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking changes, it is recommended to add version = "..." constraints to the corresponding provider blocks in configuration, with the constraint strings suggested below.

```
* provider.aws: version = "~> 2.1"
```

**Terraform has been successfully initialized!**

# For those who play alone at home

59

```
Plan: 2 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
```

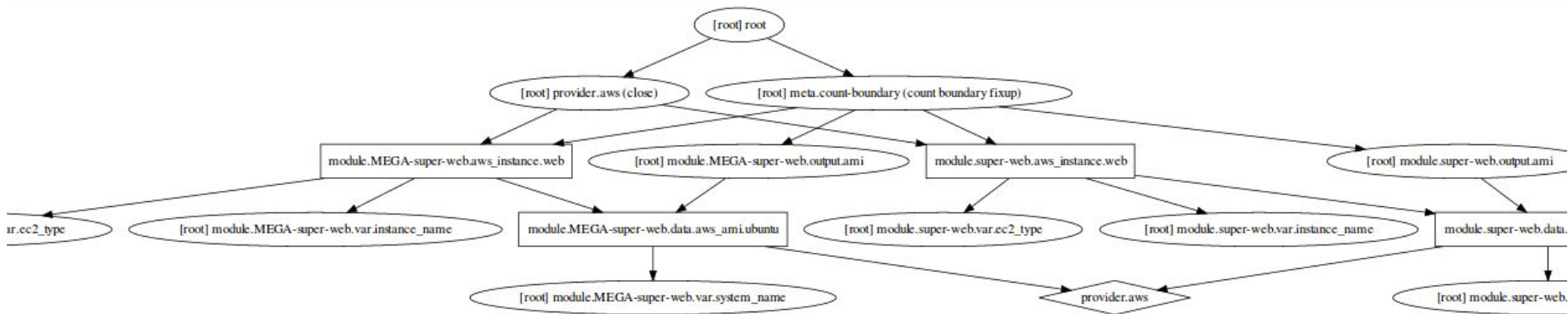
```
module.super-web.aws_instance.web: Still creating... (10s elapsed)  
module.super-web.aws_instance.web: Creation complete after 17s (ID: i-0aa3b26b6e045f99d)  
module.MEGA-super-web.aws_instance.web: Still creating... (20s elapsed)  
module.MEGA-super-web.aws_instance.web: Creation complete after 21s (ID: i-0a50bfdfbdef1370e)
```

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```



# For those who play alone at home

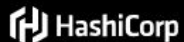
<input type="checkbox"/>	MEGA-super-web	i-0a50bdfd...	t3.micro	us-east-1a	<span style="color: green;">●</span> running
<input type="checkbox"/>	super-web	i-0aa3b26...	t3.nano	us-east-1a	<span style="color: green;">●</span> running



# For those who play alone at home

61

Secure | <https://www.terraform.io/docs/>



Learn how Terraform fits into the HashiCorp Suite >



[Intro](#) [Learn](#) [Docs](#) [Guides](#) [Extend](#) [Enterprise](#) [Download](#) [GitHub](#)

## Terraform CLI

- > Configuration Language
- > Commands (CLI)
- > Import
- > State

## Terraform Documentation

Welcome to the Terraform documentation! This documentation is more of a reference guide for all available features and options of Terraform. If you're just getting started with Terraform, please start with the [introduction and getting started guide](#) instead.

<https://www.terraform.io/docs/>



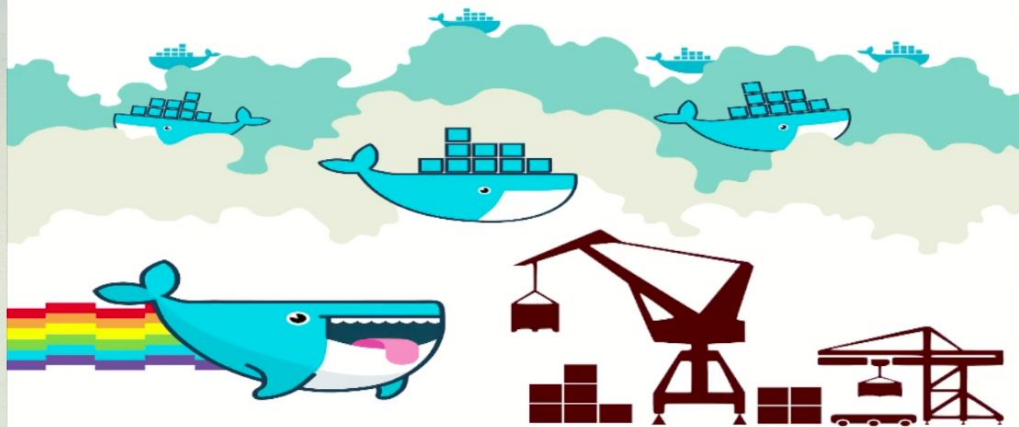
Community of Web Developers  
(January meetup #5, 2019)



<https://facebook.com/groups/websyndicate>  
<https://t.me/websyndicate>

## What about this Infra, anyhow?

4



**iTechArt** | Deployment: Fast, Fine, Affordable

**Алексей Бурим**

*iTechArt, DevOps Engineer*

Развёртывание: быстро, качественно, недорого.  
На примере AWS, ECS, Terraform

# Useful links

Show notes: <https://be34.me/show-notes>

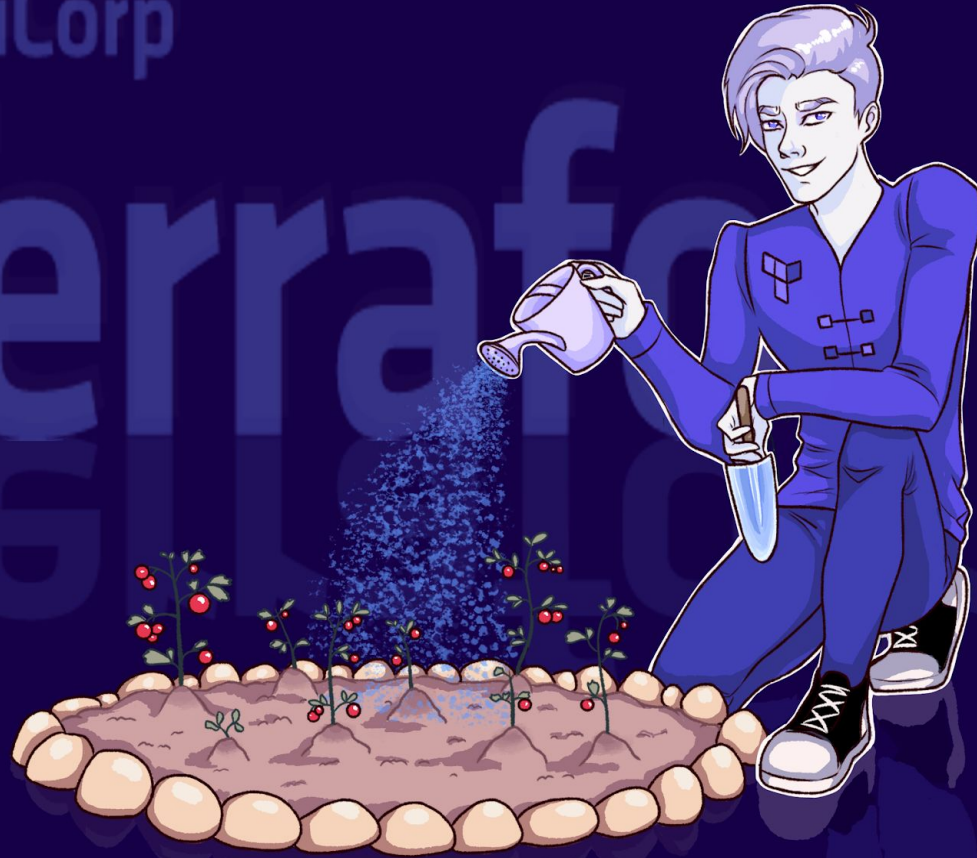


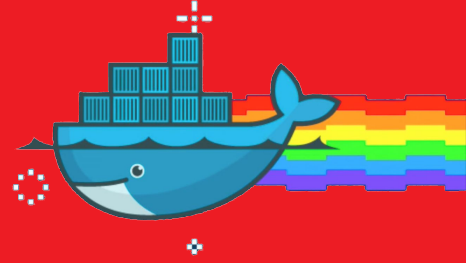
1. <https://github.com/antonbabenko/terraform-best-practices>
2. <https://stackshare.io/stackups/terraform-vs-aws-cloudformation>
3. <https://github.com/eng-Zubr/launchpad>

# Conclusion

HashiCorp

Terraform





# Thank you!

→ Questions?