

Aug 29, 16:48

2. Understand the AWS Resources we create with Terraform (VPC, Subnet & more).mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

Dashboard [Jenkins]

eu-west-3.console.aws.amazon.com

aws Services

New VPC Experience
Tell us what you think

Egress Only Internet Gateways **New**

DHCP Options Sets **New**

Elastic IPs **New**

Managed Prefix Lists **New**

Endpoints

Endpoint Services

NAT Gateways **New**

Peering Connections

▼ SECURITY

Network ACLs

Security Groups **New**

▼ VIRTUAL PRIVATE NETWORK (VPN)

Customer Gateways

Virtual Private Gateways

Site-to-Site VPN

Security

- Configure access on **subnet level** => NACL
- Configure access on **instance level** => Security Group

Region

Your VPC

Availability Zone

Subnet

EC2 Instance **RDS Instance** ...

Network Access Control List (NACL)

machine level.

10:32 11:04

Aug 29, 16:48

2. Understand the AWS Resources we create with Terraform (VPC, Subnet & more).mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

Dashboard [Jenkins]

eu-west-3.console.aws.amazon.com

aws Services

New VPC Experience
Tell us what you think

Egress Only Internet Gateways **New**

DHCP Options Sets **New**

Elastic IPs **New**

Managed Prefix Lists **New**

Endpoints

Endpoint Services

NAT Gateways **New**

Peering Connections

SECURITY

Network ACLs

Security Groups **New**

VIRTUAL PRIVATE NETWORK (VPN)

Customer Gateways

Virtual Private Gateways

Site-to-Site VPN

Security

- Configure access on **subnet level** => NACL
- Configure access on **instance level** => Security Group

Region

Your VPC

Availability Zone

Subnet

EC2 Instance

RDS Instance

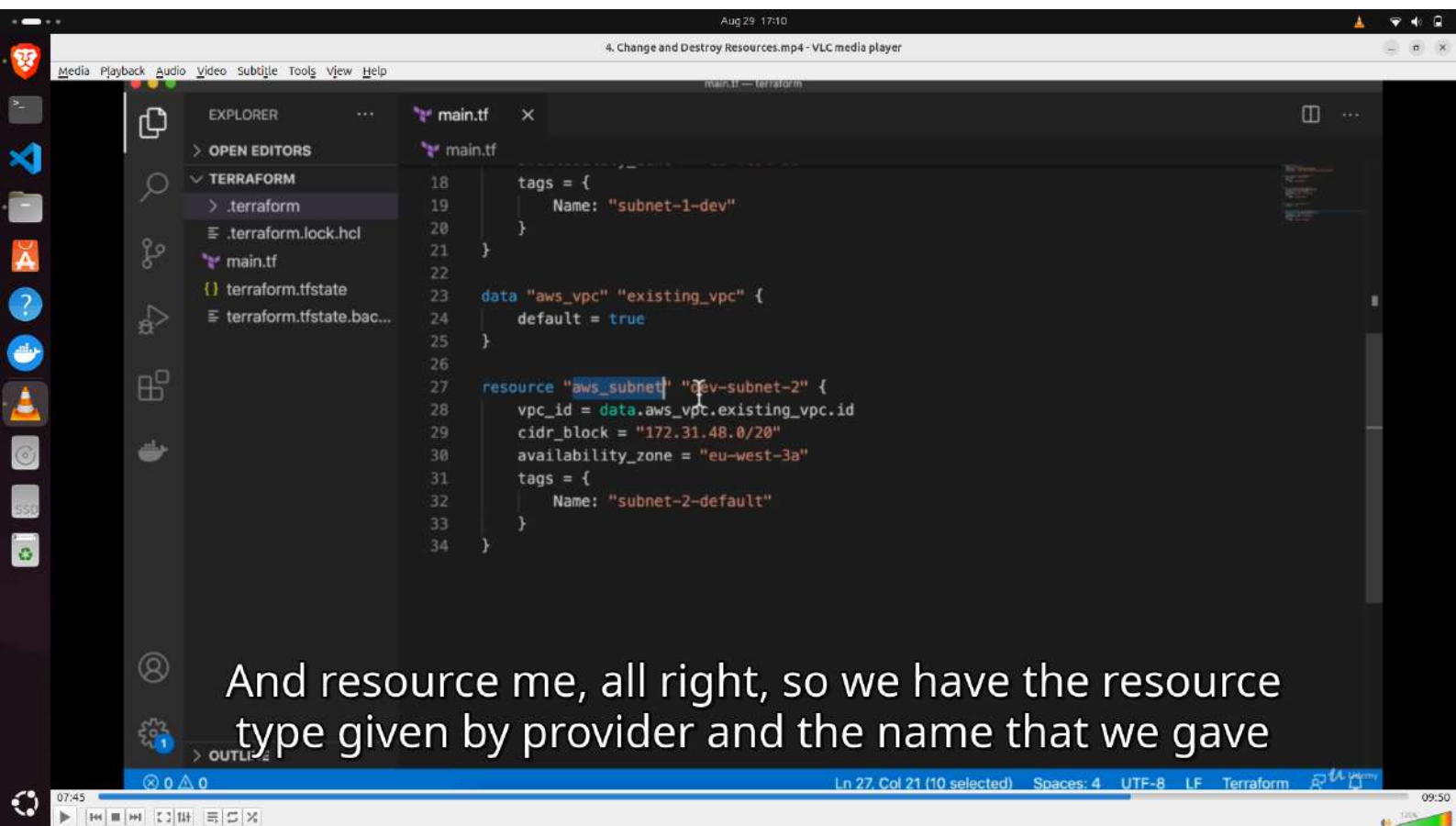
...

Network Access Control List (NACL)

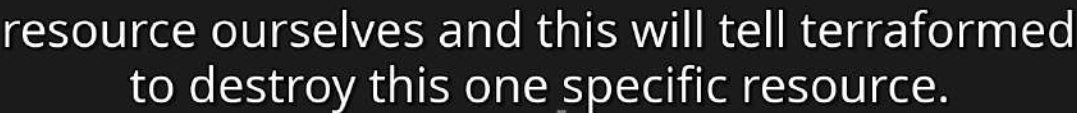
machine level.

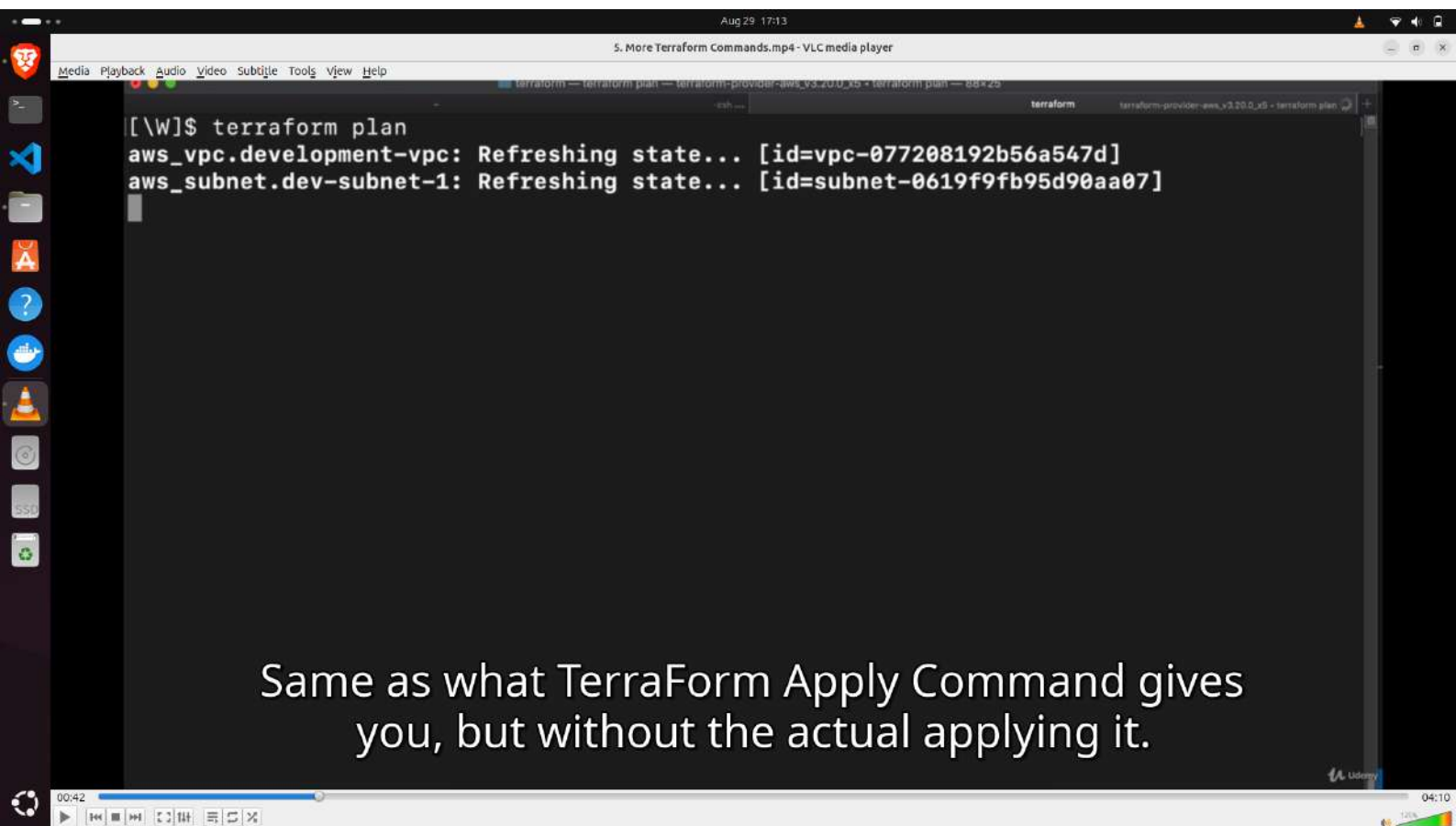
10:32

11:04

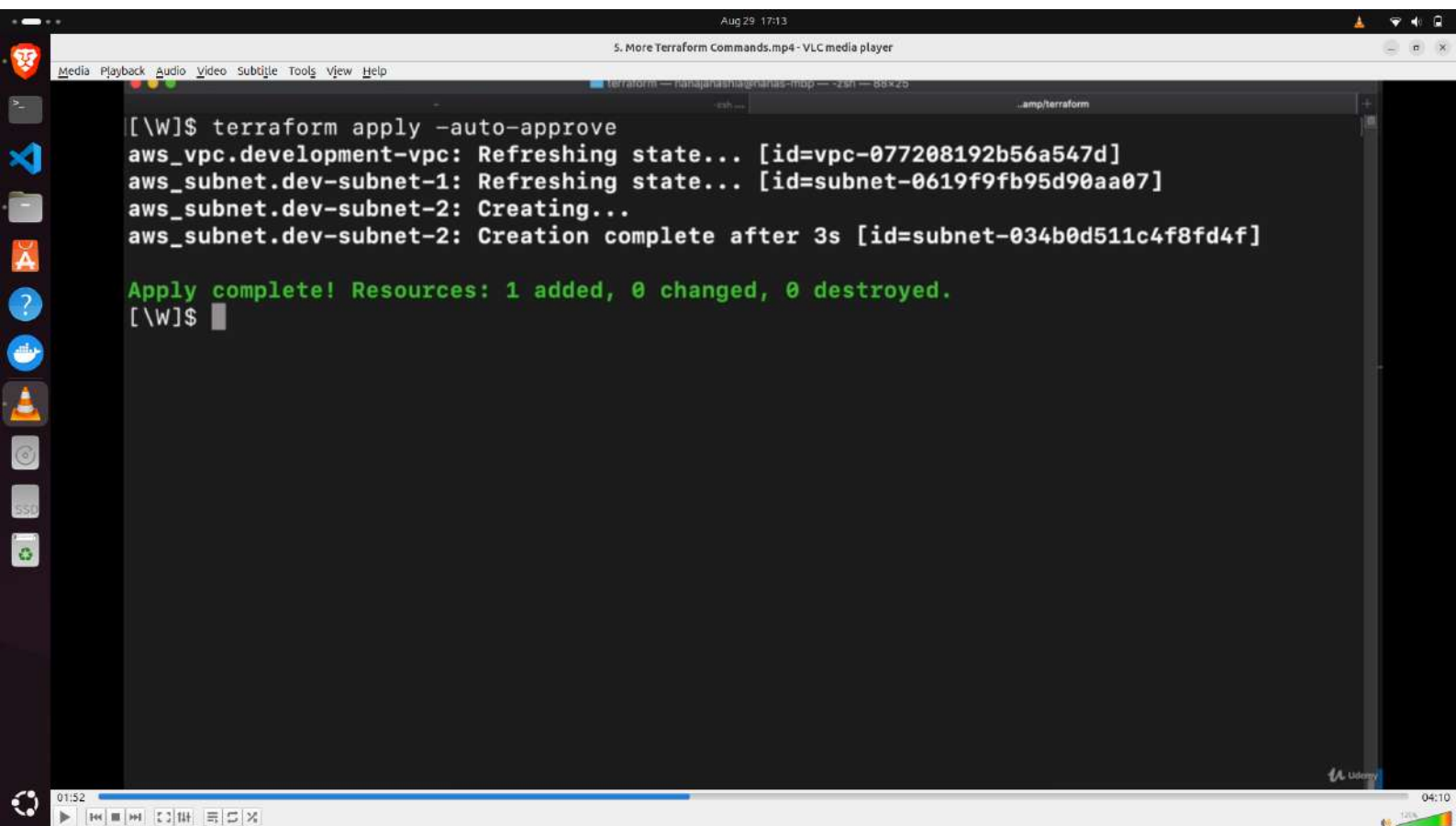


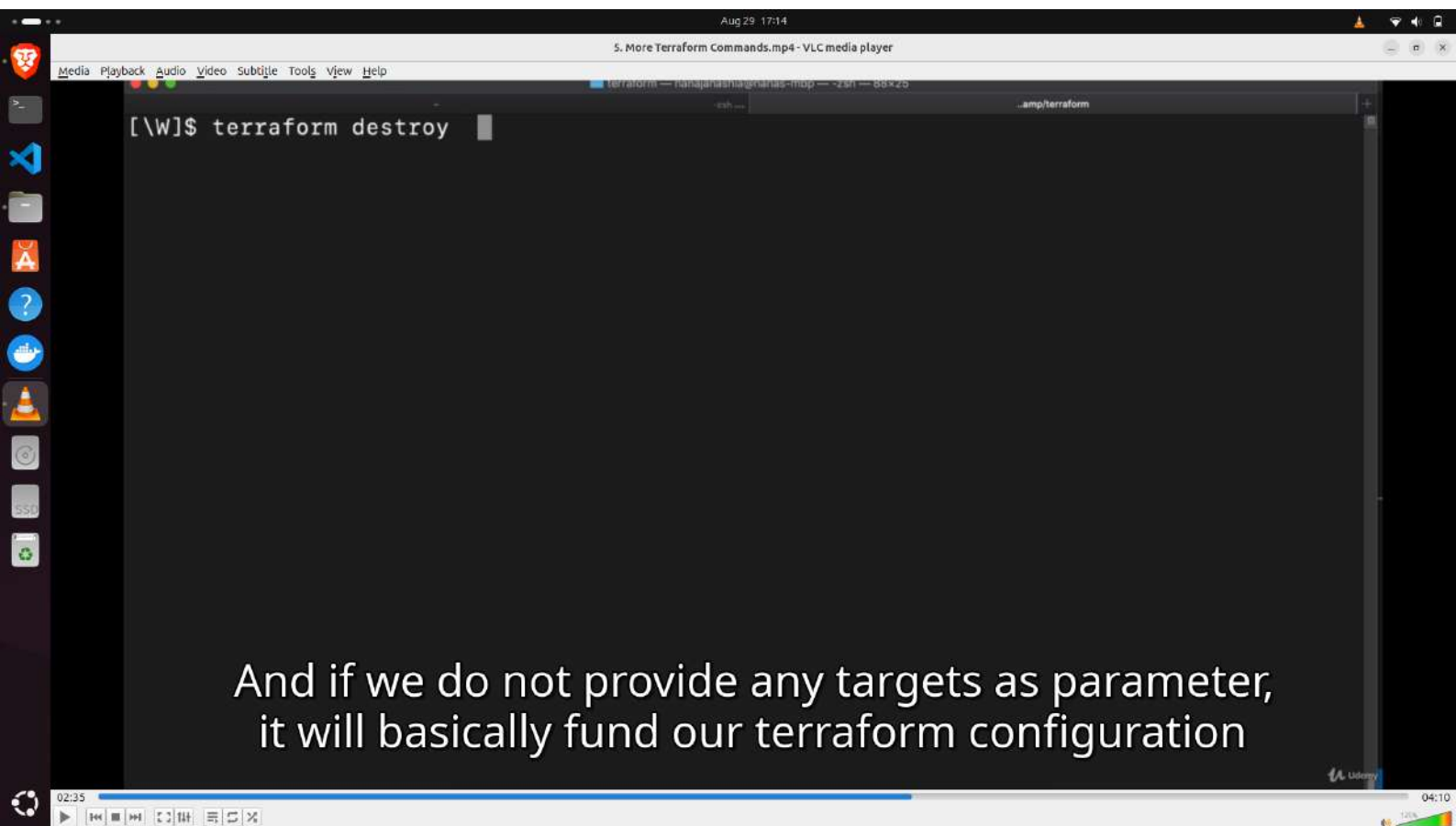
And resource me, all right, so we have the resource type given by provider and the name that we gave

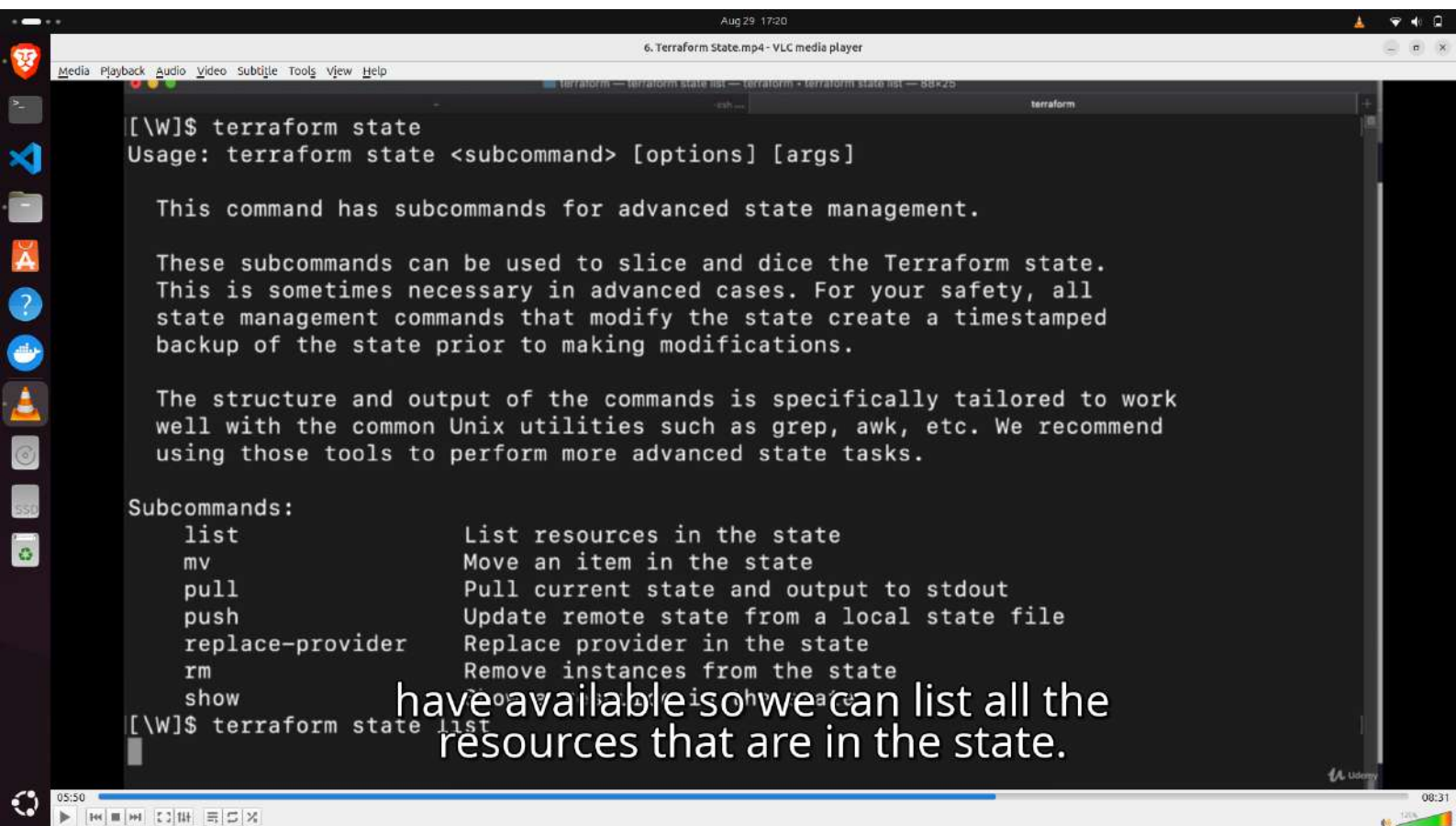


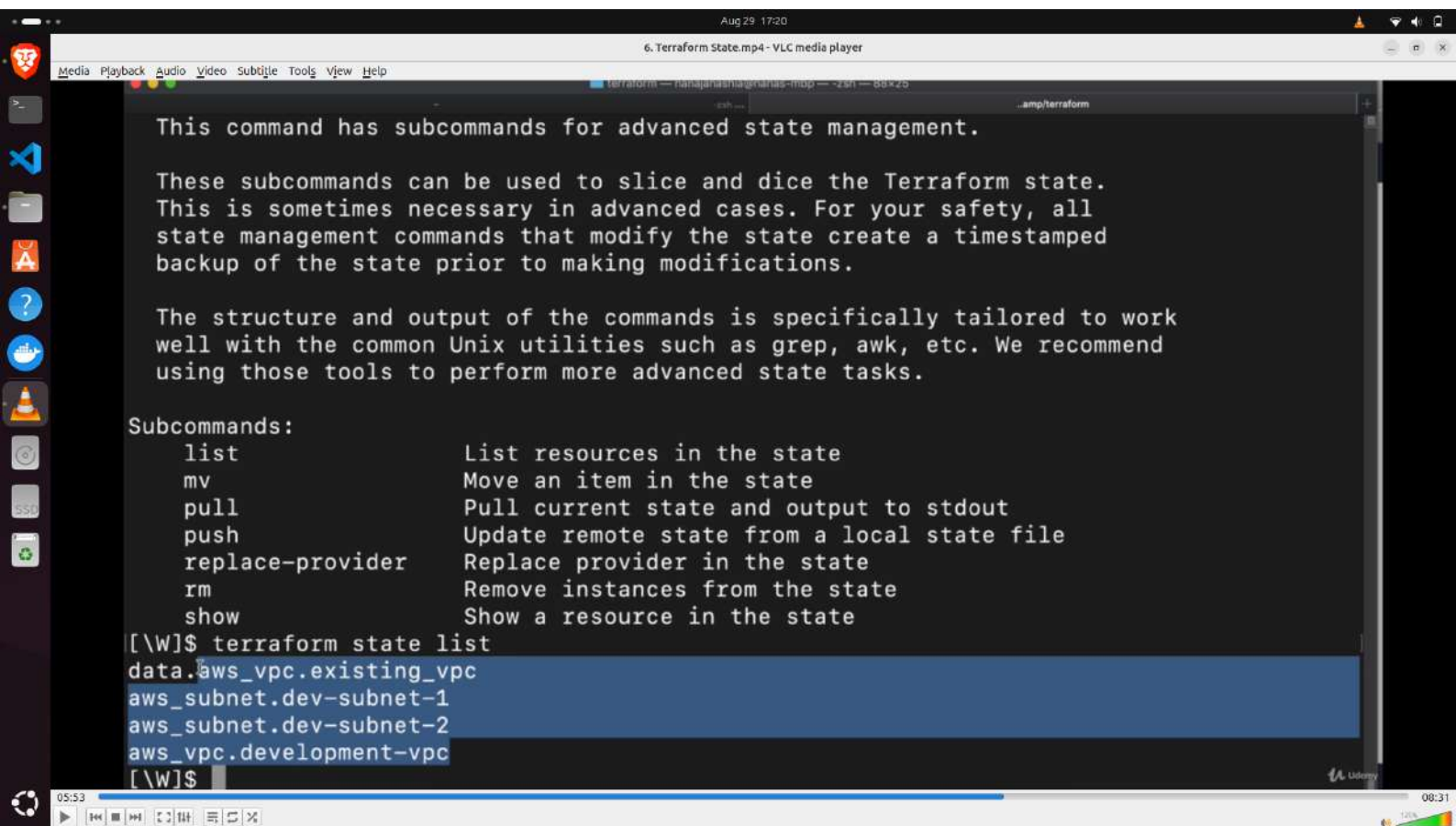


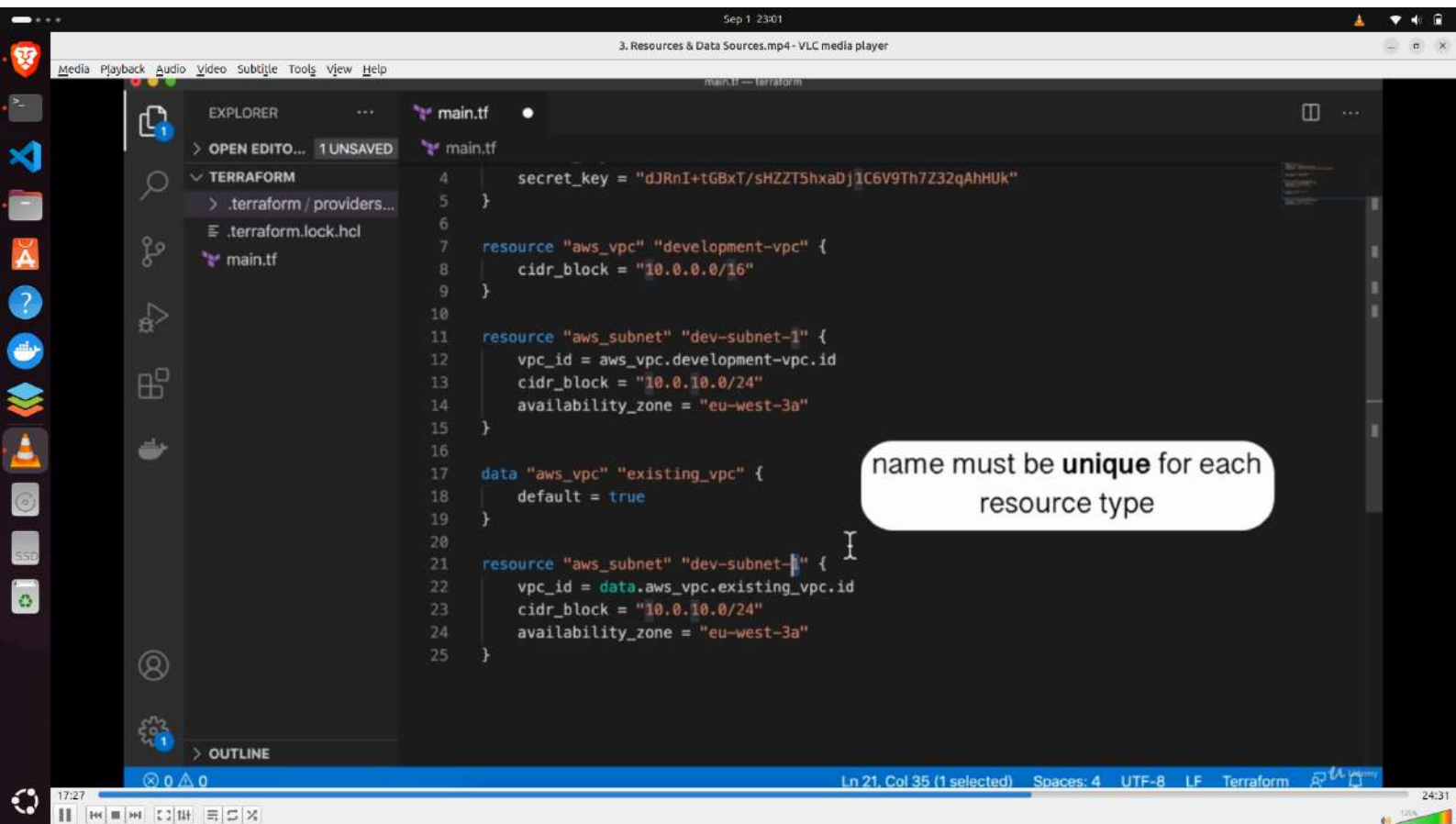
Same as what TerraForm Apply Command gives you, but without the actual applying it.











Data Sources

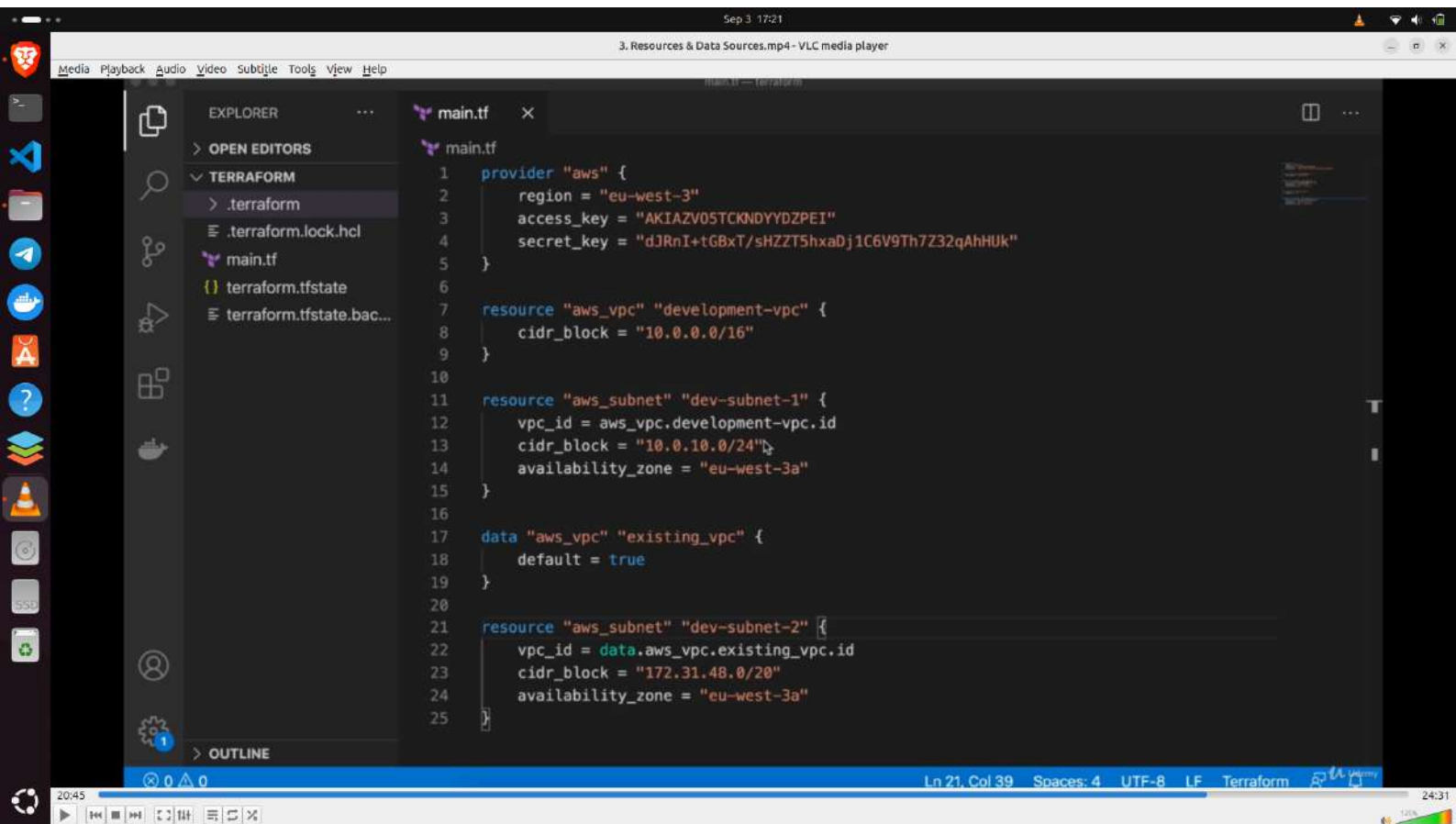
allow data to be fetched for use
in TF configuration

main.tf

main.tf

```
1  provider "aws" {
2      region = "eu-west-3"
3      access_key = "AKIAZV05TCKNDYYDZPEI"
4      secret_key = "dJRnI+tGBxT/sHZZT5hxaDj1C6V9Th7Z32qAhHUK"
5  }
6
7  resource "aws_vpc" "development-vpc" {
8      cidr_block = "10.0.0.0/16"
9  }
10
11 resource "aws_subnet" "dev-subnet-1" {
12     vpc_id = aws_vpc.development-vpc.id
13     cidr_block = "10.0.10.0/24"
14     availability_zone = "eu-west-3a"
15 }
16
```

```
16  
17 data "aws_vpc" "existing_vpc" {  
18     default = true  
19 }  
20  
21 resource "aws_subnet" "dev-subnet-1" {  
22     vpc_id = data.aws_vpc.existing_vpc.id  
23     cidr_block = "10.0.10.0/24"  
24     availability_zone = "eu-west-3a"  
25 }
```



```
[\\W]$ terraform apply
aws_vpc.development-vpc: Refreshing state... [id=vpc-077208192b56a547d]
aws_subnet.dev-subnet-2: Refreshing state... [id=subnet-0fc09b55ec46e62e7]
aws_subnet.dev-subnet-1: Refreshing state... [id=subnet-0619f9fb95d90aa07]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
[\\W]$
```


We declare the **end result!**

```
{
```



```
{
```

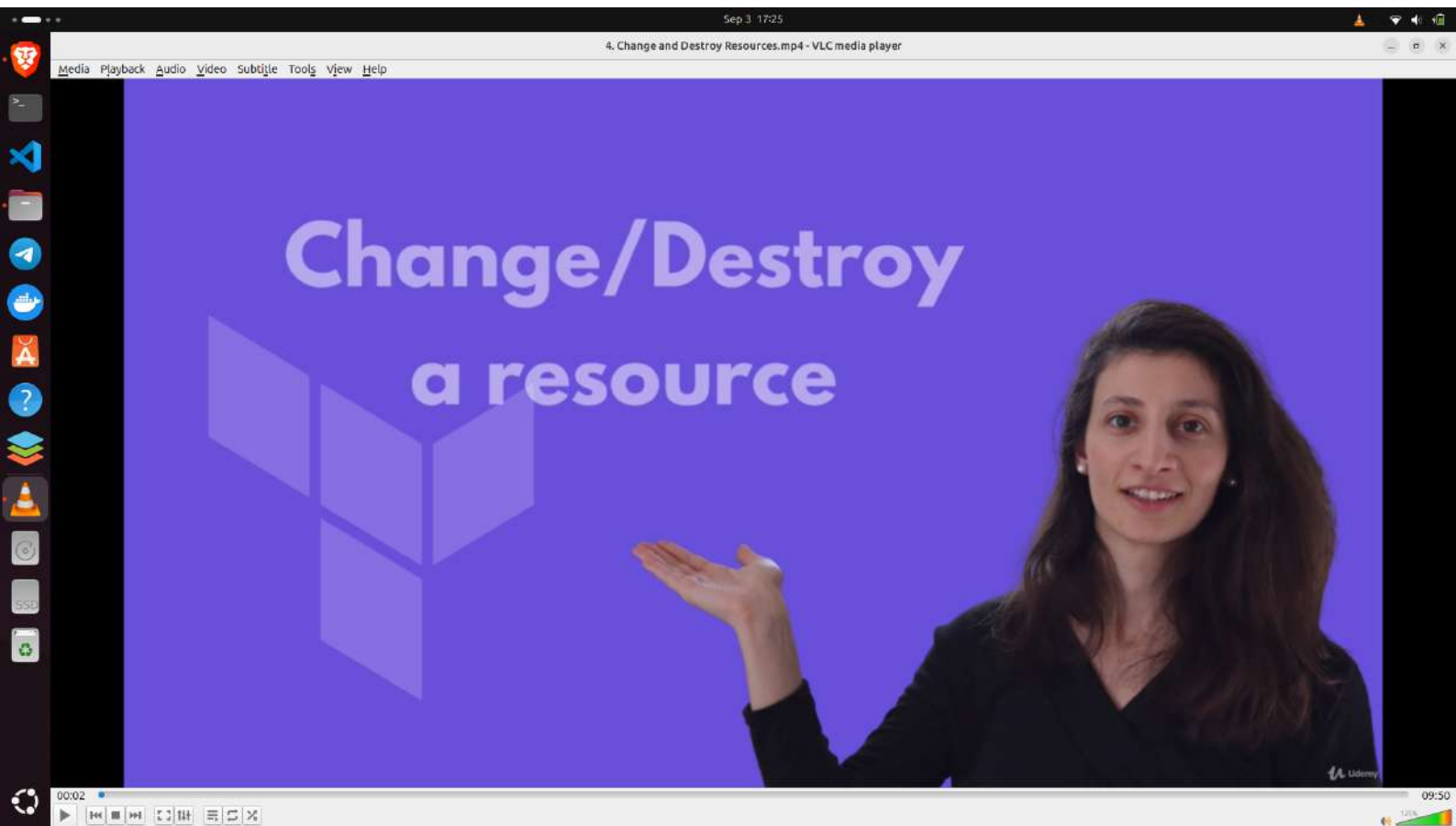


```
id
```

current AWS state

vs.

configuration (desired) state



```
8     cidr_block = "10.0.0.0/10"
9     tags = {
10         Name: "development"
11     }
12 }
13
14 resource "aws_subnet" "dev-subnet-1" {
15     vpc_id = aws_vpc.development-vpc.id
16     cidr_block = "10.0.10.0/24"
17     availability_zone = "eu-west-3a"
18     tags = {
19         Name: "development"
20     }
21 }
```

Your VPCs (1/4) [Info](#)

 *Filter VPCs*



Name



—



eks-worker-node-vpc-stack-VPC



eksctl-demo-cluster-cluster/VPC



—

ck Audio Video Subtitle Tools View Help

terraform — nanajanashia@nanas-mbp — zsh — 88x25

[\W]\$ terraform destroy -target aws_subnet.dev-subnet-2

always apply Terraform config file

especially when working in a team!

```
tags = {
```

```
  Name: "subnet-1-dev"
```

```
}
```

Difference between current and desired state?

```
data "aws_vpc" "existing_vpc" {  
  default = true
```



```
[\\W]$ terraform plan
aws_vpc.development-vpc: Refreshing state... [id=vpc-077208192b56a547d]
```

```
[\\W]$ terraform apply █
```



apply configuration without confirming

```
[\\W]$ terraform apply -auto-approve
```

```
[\\W]$ terraform destroy
```

```
et" "dev-subnet-1" {  
  vpc.development-vpc.id  
  cidr_block = "10.0.10.0/24"  
  one =  
  subnet-1-
```

you don't have to know in which **order**
you need **to delete the resources**

```
[\\W]$ terraform state
```

```
Usage: terraform state <subcommand> [options] [args]
```

This command has subcommands for advanced state management.

These subcommands can be used to slice and dice the Terraform state. This is sometimes necessary in advanced cases. For your safety, all state management commands that modify the state create a timestamped backup of the state prior to making modifications.

The structure and output of the commands is specifically tailored to work well with the common Unix utilities such as grep, awk, etc. We recommend using those tools to perform more advanced state tasks.

Subcommands:

list	List resources in the state
mv	Move an item in the state
pull	Pull current state and output to stdout
push	Update remote state from a local state file
replace-provider	Replace provider in the state
rm	Remove instances from the state
show	Show a resource in the state

```
[\\W]$ █
```

show

Show a res

```
[\\W]$ terraform state list
```

```
[\\W]$ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev-subnet-1
aws_subnet.dev-subnet-2
aws_vpc.development-vpc
[\\W]$
```



```
output "dev-vpc-id" {  
    value = aws_vpc.development-vpc.id  
}
```

```
output "dev-subnet-id" {  
    value = aws_subnet.dev-subnet-1.id  
}
```

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

Outputs:

dev-subnet-id = "subnet-00c297a6b2c8a2788"

dev-vpc-id = "vpc-0851829c800431180"

[\w]\$

```
resource "aws_vpc" "development-vpc" {  
  cidr_block = "10.0.0.0/16"  
  tags = {  
    Name: "development"  
  }  
}
```

```
resource "aws_subnet" "dev-subnet-1" {  
  vpc_id = aws_vpc.development-vpc.id  
  cidr_block = "10.0.10.0/24"  
  availability_zone = "eu-west-3a"  
  tags = {  
    Name: "subnet-1-dev"  
  }  
}
```

DEV

**DEV
specific values**

PROD

**PROD
specific values**

main.tf

```
variable "subnet_cidr_block" {  
  description = "subnet cidr block"  
}
```

```
resource "aws_vpc" "development-vpc" {  
  cidr_block = "10.0.0.0/16"  
  tags = {  
    Name: "development"  
  }  
}
```

```
resource "aws_subnet" "dev-subnet-1" {  
  vpc_id = aws_vpc.development-vpc.id  
  cidr_block = var.subnet_cidr_block  
  availability_zone = "eu-west-3a"  
  tags = {  
    Name: "subnet-1-dev"  
  }  
}
```

**3 ways to pass value
to the input variable**

```
[\\W]$ terraform apply  
var.subnet_cidr_block  
  subnet cidr block
```

```
Enter a value: 10.0.20.0/24
```

```
aws_vpc.development-vpc: Refreshing state... [id=vpc-0851829c800431180]
```

```
[\\W]$ terraform apply -var "subnet_cidr_block=10.0.30.0/24"
```




```
9   }
10
11   variable "vpc_cidr_block" {
12     description = "vpc cidr block"
13   }
14
15   resource "aws_vpc" "development-vpc" {
16     cidr_block = var.vpc_cidr_block
17     tags = {
18       Name: "development"
19     }
20   }
21
22   resource "aws_subnet" "dev_subnet_1" {
```

Sep 4 14:20

8. Input Variables - Parameterize your Terraform Configuration to make it re-usable.mp4 - VLC media player

terraform.tfvars — terraform

 main.tf

 terraform.tfvars ●

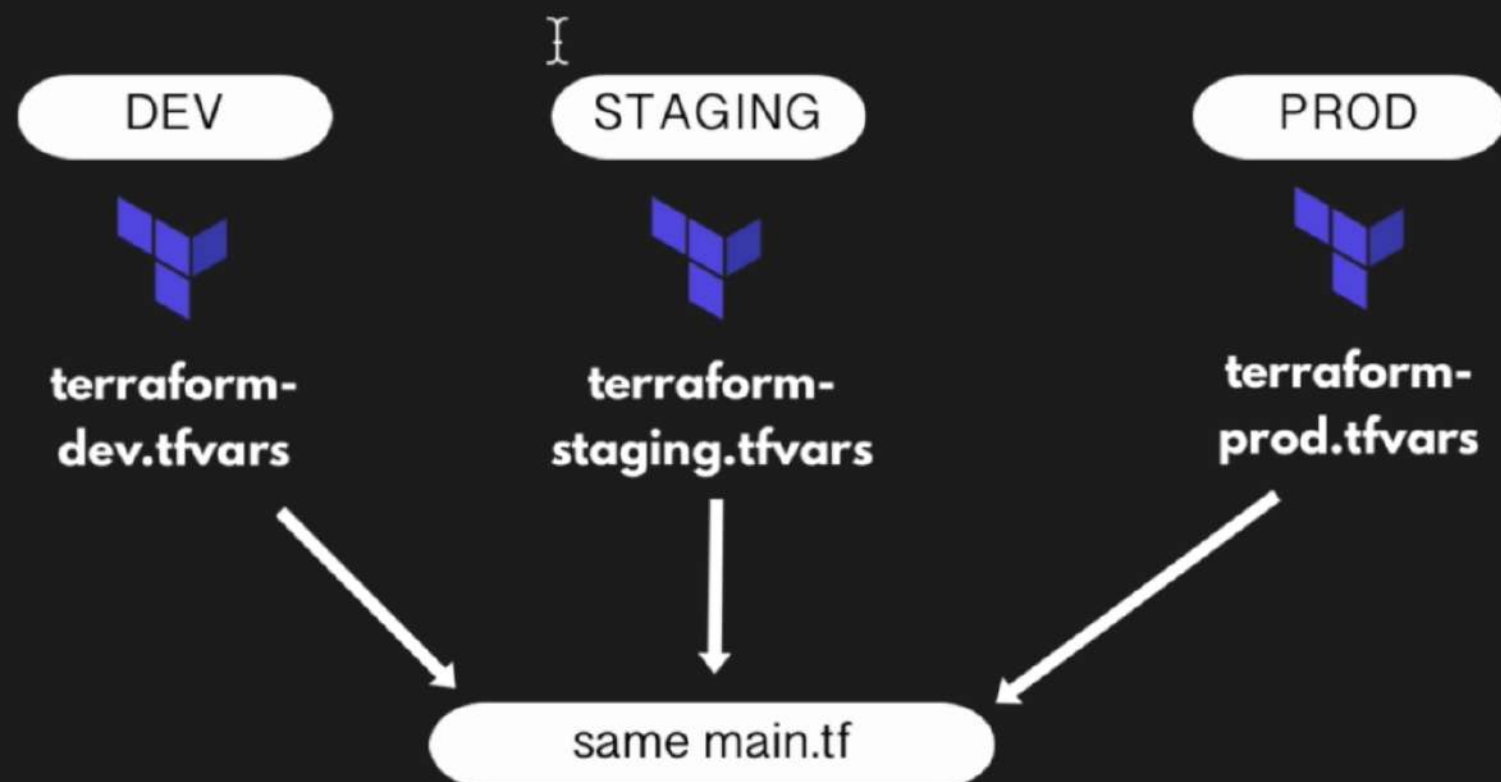
 terraform.tfstate

 terraform.tfvars

```
1 subnet_cidr_block = "10.0.40.0/24"
2 vpc_cidr_block = "10.0.0.0/16"
```

Use Case for Input Variables

```
terraform.tfvars
1  subnet_cidr_block = "10.0.40.0/24"
2  vpc_cidr_block    = "10.0.0.0/16"
3
4  environment = "development"
```





EXPLORER



> OPEN EDITORS



✓ TERRAFORM

> .terraform

≡ .terraform.lock.hcl



Y main.tf

Y terraform-dev.tfvars



{ } terraform.tfstate

≡ terraform.tfstate.backup



```
[\\W]$ terraform apply -var-file terraform-dev.tfvars
aws_vpc.development-vpc: Refreshing state... [id=vpc-0851829c800431180]
```

Default Values

```
variable "subnet_cidr_block" {  
  description = "subnet cidr block"  
  default = "10.0.10.0/24"  
}  
  
resource "aws_vpc" "development-vpc" {  
  cidr_block = "10.0.0.0/16"  
  tags = {  
    Name: "development"  
  }  
}  
  
resource "aws_subnet" "dev-subnet-1" {  
  vpc_id = aws_vpc.development-vpc.id  
  cidr_block = var.subnet_cidr_block
```



A default value makes the variable **optional**


```
variable "subnet_cidr_block" {  
  description = "subnet cidr block"  
  default = "10.0.10.0/24"  
  type = string  
}
```

```
resource "aws_vpc" "development-vpc" {  
  cidr_block = "10.0.0.0/16"  
  tags = {  
    Name: "development"  
  }  
}
```

```
resource "aws_subnet" "dev-subnet-1" {  
  vpc_id = aws_vpc.development-vpc.id
```



"type" specifies
what value types are accepted


```
variable "cidr_blocks" {  
  description = "cidr blocks for vpc and subnets"  
  type = list(string)  
}
```

```
5
6
7 variable "cidr_blocks" {
8     description = "cidr blocks for vpc and subnets"
9     type = list(string)
10 }
11
12 resource "aws_vpc" "development-vpc" {
13     cidr_block = "10.0.0.0/16"
14     tags = {
15         Name: "development"
16     }
17 }
18
19 resource "aws_subnet" "dev-subnet-1" {
20     vpc_id = aws_vpc.development-vpc.id
21     cidr_block = var.cidr_blocks[1]
22     availability_zone = "eu-west-3a"
23     tags = {
24         Name: "subnet-1-dev"
25     }
26 }
27
```

```
5  
6  
7 variable "cidr_blocks" {  
8     description = "cidr blocks for vpc and subnets"  
9     type = list(string)  
10 }  
11
```

```
12 resource "aws_vpc" "development-vpc" {  
13     cidr_block = var.cidr_blocks[0]  
14     tags = {  
15         Name: "development"  
16     }  
17 }
```

```
18 resource "aws_subnet" "dev-subnet-1" {  
19     vpc_id = aws_vpc.development-vpc.id  
20     cidr_block = var.cidr_blocks[1]  
21     availability_zone = "eu-west-3a"  
22     tags = {  
23         Name: "subnet-1-dev"  
24     }  
25 }
```



A terminal window with a dark background and light gray text. The window title bar at the top shows three colored window control buttons (red, yellow, green) on the left, followed by a series of tabs. The active tab is labeled "terraform — terraform apply -var-file terraform-dev.tfvars — terraform-provider-aws_v3.20.0_x5 • terraform apply -var-file terraform-dev.tfvars". Below the title bar, the terminal content shows a prompt "[\W]\$ " followed by the command "terraform apply -var-file terraform-dev.tfvars". A light gray cursor is positioned at the end of the command line.

```
[\W]$ terraform apply -var-file terraform-dev.tfvars
```

main.tf

terraform-dev.tfvars ×

terraform-dev.tfvars

```
1  cidr_blocks = [  
2    {cidr_block = "10.0.0.0/16", name = "dev-vpc"},  
3    {cidr_block = "10.0.10.0/24", name = "dev-subnet"}  
4  ]
```

list of objects

```
6
7 variable "cidr_blocks" {
8     description = "cidr blocks for vpc and subnets"
9     type = list(object({
10         cidr_block = string
11         name = string
12     }))
13 }
14
```

```
}  
  
resource "aws_vpc" "development-vpc" {  
  cidr_block = var.cidr_blocks[0].cidr_block  
  tags = {  
    Name: var.cidr_blocks[0].name  
  }  
}
```



```
resource "aws_subnet" "dev-subnet-1" {  
  vpc_id = aws_vpc.development-vpc.id  
  cidr_block = var.cidr_blocks[1].cidr_block  
  availability_zone = "eu-west-3a"  
  tags = {  
    Name: var.cidr_blocks[1].name  
  }  
}
```

✓
5s



```
import speedtest

# Create a Speedtest object
st = speedtest.Speedtest()

# Get the best server
st.get_best_server()

# Perform download and upload tests
download_speed = st.download() / 1_000_000 # Convert to Mbps
upload_speed = st.upload() / 1_000_000 # Convert to Mbps

# Print the results
print(f"Download speed: {download_speed:.2f} Mbps")
print(f"Upload speed: {upload_speed:.2f} Mbps")
```



```
Download speed: 2702.17 Mbps
Upload speed: 694.31 Mbps
```

```
(venv) dilip@dilip:~/Bulk_Email_sender$ python3 main.py
1 emails sent From 202052313@iiitvadodara.ac.in To harshida@ccotech.co.in
2 emails sent From 202052313@iiitvadodara.ac.in To husahu@ciphercloud.com
3 emails sent From 202052313@iiitvadodara.ac.in To harshita.rathore@moonfrogglabs.com
4 emails sent From 202052313@iiitvadodara.ac.in To imran.h@apar.com
5 emails sent From 202052313@iiitvadodara.ac.in To heena@clevertap.com
6 emails sent From 202052313@iiitvadodara.ac.in To hemant.batra@supraits.com
7 emails sent From 202052313@iiitvadodara.ac.in To hemant.pawar@ndsglobal.com
8 emails sent From 202052313@iiitvadodara.ac.in To hemendra.bist@u2opiamobile.com
9 emails sent From 202052313@iiitvadodara.ac.in To hemlata.goel@shl.com
10 emails sent From 202052313@iiitvadodara.ac.in To hemraj@cloudthat.in
11 emails sent From 202052313@iiitvadodara.ac.in To hsoni@infosenseglobal.com
12 emails sent From 202052313@iiitvadodara.ac.in To hima@systango.com
13 emails sent From 202052313@iiitvadodara.ac.in To himagauri@metamorphtech.com
14 emails sent From 202052313@iiitvadodara.ac.in To himanshub@appcino.com
15 emails sent From 202052313@iiitvadodara.ac.in To hmishra@valethi.com
16 emails sent From 202052313@iiitvadodara.ac.in To himanshu@ditserve.com
17 emails sent From 202052313@iiitvadodara.ac.in To hina.khan@acuteinformatics.in
18 emails sent From 202052313@iiitvadodara.ac.in To hsingh@hitachi-solutions.com
```

Today ▾

New Message Today



Chandrachood Chatarkar Chatarkar

@Brihaspati Kumar Pandey Sir, currently, I don't have any task. If any, pls assign.

Actions ▾

et

Here are the Good Recruitment Agencies in Bangalore IT/non-IT sectors

1.Hire Glocal : Hire Glocal is a prominent recruitment agency with a strong presence in Bangalore and all over India.one of the best recruitment agencies in India, Hire Glocal has a strong track record of success and an impressive geographic reach, serving both the IT and non-IT sectors.A wide range of services are provided by them, including turnkey recruitment, interim management, executive search, CXO hiring, staffing, and HR consultancy.

2.Humanley India : is a Human Resources and Business Services company based in Bengaluru, Karnataka. They prioritize employee dedication and satisfaction, believing that happy employees lead to increased productivity.

3.White Horse Manpower Consultancy : is a well-established recruitment agency based in Bangalore, Karnataka, India. They specialize in HR recruitment, IT services, and IT consulting, catering to both IT and non-IT sectors.

4.SutraHR : is a renowned recruitment agency in India, known for its specialized approach in IT recruitment and serving as a recruitment agency for startups. They excel in HR recruitment, IT recruitment, and offer personalized services and employer branding solutions.

631 views

Sep 5 18:00

1. Introduction to Modules.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

EXPLORER

main.tf

main.tf

```
54 protocol = "tcp"
55 cidr_blocks = [var.my_ip]
56 }
57
58 ingress {
59   from_port = 8080
60   to_port = 8080
61   protocol = "tcp"
62   cidr_blocks = [var.my_ip]
63 }
64
65 egress {
66   from_port = 0
67   to_port = 0
68   protocol = "all"
69   cidr_blocks = ["0.0.0.0/0"]
70   prefix_list_ids = []
71 }
72
73 tags = {
74   Name: "${var.env_prefix}-default-ec2"
75 }
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

Module for EC2 Instance

"webserver" module

And also a security group, because we're assigning the security group to easy to instant's, so this

01:40 09:20

The video player interface shows a presentation slide titled "Modules". The slide content includes a diagram where a "webserver module" is shown at the top, with three arrows pointing down to boxes labeled "region A", "region B", and "region C". A callout bubble next to the module box says "re-use same configuration".

In the background, a code editor displays the content of a file named "main.tf":

```

57
58 ingress {
59     from_port = 8080
60     to_port = 8080
61     protocol = "tcp"
62     cidr_blocks = ["0.0.0.0/0"]
63 }
64
65 egress {
66     from_port = 0
67     to_port = 0
68     protocol = "-1"
69     cidr_blocks = ["0.0.0.0/0"]
70     prefix_list_ids = []
71 }
72
73 tags = {
74     Name: "${var.env_prefix}-default-ec2"
75 }
76
77
78 data "aws_ami" "latest-aws2-ubuntu" {
79     most_recent = true
80     owners = ["amazon"]
81     filter {
82         name = "name"

```


Sep 5 16:01

1. Introduction to Modules.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

EXPLORER

main.tf

main.tf

93 }

94 }

95 output "ec2_public_ip" {

96 value = aws_instance.webserver.public_ip

97 }

98 }

99 }

100 }

101 }

102 }

103 }

104 }

105 }

106 }

107 }

108 }

109 }

110 }

111 }

112 }

113 }

114 }

115 }

116 }

117 }

118 }

119 }

120 }

121 }

122 }

123 }

124 }

125 }

126 }

127 }

128 }

129 }

130 }

131 }

132 }

133 }

134 }

135 }

136 }

137 }

138 }

139 }

140 }

141 }

142 }

143 }

144 }

145 }

146 }

147 }

148 }

149 }

150 }

151 }

152 }

153 }

154 }

155 }

156 }

157 }

158 }

159 }

160 }

161 }

162 }

163 }

164 }

165 }

166 }

167 }

168 }

169 }

170 }

171 }

172 }

173 }

174 }

175 }

176 }

177 }

178 }

179 }

180 }

181 }

182 }

183 }

184 }

185 }

186 }

187 }

188 }

189 }

190 }

191 }

192 }

193 }

194 }

195 }

196 }

197 }

198 }

199 }

200 }

201 }

202 }

203 }

204 }

205 }

206 }

207 }

208 }

209 }

210 }

211 }

212 }

213 }

214 }

215 }

216 }

217 }

218 }

219 }

220 }

221 }

222 }

223 }

224 }

225 }

226 }

227 }

228 }

229 }

230 }

231 }

232 }

233 }

234 }

235 }

236 }

237 }

238 }

239 }

240 }

241 }

242 }

243 }

244 }

245 }

246 }

247 }

248 }

249 }

250 }

251 }

252 }

253 }

254 }

255 }

256 }

257 }

258 }

259 }

260 }

261 }

262 }

263 }

264 }

265 }

266 }

267 }

268 }

269 }

270 }

271 }

272 }

273 }

274 }

275 }

276 }

277 }

278 }

279 }

280 }

281 }

282 }

283 }

284 }

285 }

286 }

287 }

288 }

289 }

290 }

291 }

292 }

293 }

294 }

295 }

296 }

297 }

298 }

299 }

300 }

301 }

302 }

303 }

304 }

305 }

306 }

307 }

308 }

309 }

310 }

311 }

312 }

313 }

314 }

315 }

316 }

317 }

318 }

319 }

320 }

321 }

322 }

323 }

324 }

325 }

326 }

327 }

328 }

329 }

330 }

331 }

332 }

333 }

334 }

335 }

336 }

337 }

338 }

339 }

340 }

341 }

342 }

343 }

344 }

345 }

346 }

347 }

348 }

349 }

350 }

351 }

352 }

353 }

354 }

355 }

356 }

357 }

358 }

359 }

360 }

361 }

362 }

363 }

364 }

365 }

366 }

367 }

368 }

369 }

370 }

371 }

372 }

373 }

374 }

375 }

376 }

377 }

378 }

379 }

380 }

381 }

382 }

383 }

384 }

385 }

386 }

387 }

388 }

389 }

390 }

391 }

392 }

393 }

394 }

395 }

396 }

397 }

398 }

399 }

400 }

401 }

402 }

403 }

404 }

405 }

406 }

407 }

408 }

409 }

410 }

411 }

412 }

413 }

414 }

415 }

416 }

417 }

418 }

419 }

420 }

421 }

422 }

423 }

424 }

425 }

426 }

427 }

428 }

429 }

430 }

431 }

432 }

433 }

434 }

435 }

436 }

437 }

438 }

439 }

440 }

441 }

442 }

443 }

444 }

445 }

446 }

447 }

448 }

449 }

450 }

451 }

452 }

453 }

454 }

455 }

456 }

457 }

458 }

459 }

460 }

461 }

462 }

463 }

464 }

465 }

466 }

467 }

468 }

469 }

470 }

471 }

472 }

473 }

474 }

475 }

476 }

477 }

478 }

479 }

480 }

481 }

482 }

483 }

484 }

485 }

486 }

487 }

488 }

489 }

490 }

491 }

492 }

493 }

494 }

495 }

496 }

497 }

498 }

499 }

500 }

501 }

502 }

503 }

504 }

505 }

506 }

507 }

508 }

509 }

510 }

511 }

512 }

513 }

514 }

515 }

516 }

517 }

518 }

519 }

520 }

521 }

522 }

523 }

524 }

525 }

526 }

527 }

528 }

529 }

530 }

531 }

532 }

533 }

534 }

535 }

536 }

537 }

538 }

539 }

540 }

541 }

542 }

543 }

544 }

545 }

546 }

547 }

548 }

549 }

550 }

551 }

552 }

553 }

554 }

555 }

556 }

557 }

558 }

559 }

560 }

561 }

562 }

563 }

564 }

565 }

566 }

567 }

568 }

569 }

570 }

571 }

572 }

573 }

574 }

575 }

576 }

577 }

578 }

579 }

580 }

581 }

582 }

583 }

584 }

585 }

586 }

587 }

588 }

589 }

590 }

591 }

592 }

593 }

594 }

595 }

596 }

597 }

598 }

599 }

600 }

601 }

602 }

603 }

604 }

605 }

606 }

607 }

608 }

609 }

610 }

611 }

612 }

613 }

614 }

615 }

616 }

617 }

618 }

619 }

620 }

621 }

622 }

623 }

624 }

625 }

626 }

627 }

628 }

629 }

630 }

631 }

632 }

633 }

634 }

635 }

636 }

637 }

638 }

639 }

640 }

641 }

642 }

643 }

644 }

645 }

646 }

647 }

648 }

649 }

650 }

651 }

652 }

653 }

654 }

655 }

656 }

657 }

658 }

659 }

660 }

661 }

662 }

663 }

664 }

665 }

666 }

667 }

668 }

669 }

670 }

671 }

672 }

673 }

674 }

675 }

676 }

677 }

678 }

679 }

680 }

681 }

682 }

683 }

684 }

685 }

686 }

687 }

688 }

689 }

690 }

691 }

692 }

693 }

694 }

695 }

696 }

697 }

698 }

699 }

700 }

701 }

702 }

703 }

704 }

705 }

706 }

707 }

708 }

709 }

710 }

711 }

712 }

713 }

714 }

715 }

716 }

717 }

718 }

719 }

720 }

721 }

722 }

723 }

724 }

725 }

726 }

727 }

728 }

729 }

730 }

731 }

732 }

733 }

734 }

735 }

736 }

737 }

738 }

739 }

740 }

741 }

742 }

743 }

744 }

745 }

746 }

747 }

748 }

749 }

750 }

751 }

752 }

753 }

754 }

755 }

756 }

757 }

758 }

759 }

760 }

761 }

762 }

763 }

764 }

765 }

766 }

767 }

768 }

769 }

770 }

771 }

772 }

773 }

774 }

775 }

776 }

777 }

778 }

779 }

780 }

781 }

782 }

783 }

784 }

785 }

786 }

787 }

788 }

789 }

790 }

791 }

792 }

793 }

794 }

795 }

796 }

797 }

798 }

799 }

800 }

801 }

802 }

803 }

804 }

805 }

806 }

807 }

808 }

809 }

810 }

811 }

812 }

813 }

814 }

815 }

816 }

817 }

818 }

819 }

820 }

821 }

822 }

823 }

824 }

825 }

826 }

827 }

828 }

829 }

830 }

831 }

832 }

833 }

834 }

835 }

836 }

837 }

838 }

839 }

840 }

841 }

842 }

843 }

844 }

845 }

846 }

847 }

848 }

849 }

850 }

851 }

852 }

853 }

854 }

855 }

856 }

857 }

858 }

859 }

860 }

861 }

862 }

863 }

864 }

865 }

866 }

867 }

868 }

869 }

870 }

871 }

872 }

873 }

874 }

875 }

876 }

877 }

878 }

879 }

880 }

881 }

882 }

883 }

884 }

885 }

886 }

887 }

888 }

889 }

890 }

891 }

892 }

893 }

894 }

895 }

896 }

897 }

898 }

899 }

900 }

901 }

902 }

903 }

904 }

905 }

906 }

907 }

908 }

909 }

910 }

911 }

912 }

913 }

914 }

915 }

916 }

917 }

918 }

919 }

920 }

921 }

922 }

923 }

924 }

925 }

926 }

927 }

928 }

929 }

930 }

931 }

932 }

933 }

934 }

935 }

936 }

937 }

938 }

939 }

940 }

941 }

942 }

943 }

944 }

945 }

946 }

947 }

948 }

949 }

950 }

951 }

952 }

953 }

954 }

955 }

956 }

957 }

958 }

959 }

960 }

961 }

962 }

963 }

964 }

965 }

966 }

967 }

968 }

969 }

970 }

971 }

972 }

973 }

974 }

975 }

976 }

977 }

978 }

979 }

980 }

981 }

982 }

983 }

984 }

985 }

986 }

987 }

988 }

989 }

990 }

991 }

992 }

993 }

994 }

995 }

996 }

997 }

998 }

999 }

1000 }

1001 }

1002 }

1003 }

1004 }

1005 }

1006 }

1007 }

1008 }

1009 }

1010 }

1011 }

1012 }

1013 }

1014 }

1015 }

1016 }

1017 }

1018 }

1019 }

1020 }

1021 }

1022 }

1023 }

1024 }

1025 }

1026 }

1027 }

1028 }

1029 }

1030 }

1031 }

1032 }

1033 }

1034 }

1035 }

1036 }

1037 }

1038 }

1039 }

1040 }

1041 }

1042 }

1043 }

1044 }

1045 }

1046 }

1047 }

1048 }

1049 }

1050 }

1051 }

1052 }

1053 }

1054 }

1055 }

1056 }

1057 }

1058 }

1059 }

1060 }

1061 }

1062 }

1063 }

1064 }

1065 }

1066 }

1067 }

1068 }

1069 }

1070 }

1071 }

1072 }

1073 }

1074 }

1075 }

1076 }

1077 }

1078 }

1079 }

1080 }

1081 }

1082 }

1083 }

1084 }

1085 }

1086 }

1087 }

1088 }

1089 }

1090 }

1091 }

1092 }

1093 }

1094 }

1095 }

1096 }

1097 }

1098 }

1099 }

1100 }

1101 }

1102 }

1103 }

1104 }

1105 }

1106 }

1107 }

1108 }

1109 }

1110 }

1111 }

1112 }

1113 }

1114 }

1115 }

1116 }

1117 }

1118 }

1119 }

1120 }

1121 }

1122 }

1123 }

1124 }

1125 }

1126 }

1127 }

1128 }

1129 }

1130 }

1131 }

1132 }

1133 }

1134 }

1135 }

1136 }

1137 }

1138 }

1139 }

1140 }

1141 }

1142 }

1143 }

1144 }

1145 }

1146 }

1147 }

1148 }

1149 }

1150 }

1151 }

1152 }

1153 }

1154 }

1155 }

1156 }

1157 }

1158 }

1159 }

1160 }

1161 }

1162 }

1163 }

1164 }

1165 }

1166 }

1167 }

1168 }

1169 }

1170 }

1171 }

1172 }

1173 }

1174 }

1175 }

1176 }

1177 }

1178 }

1179 }

1180 }

1181 }

1182 }

1183 }

1184 }

1185 }

1186 }

1187 }

1188 }

1189 }

1190 }

1191 }

1192 }

1193 }

1194 }

1195 }

1196 }

1197 }

1198 }

1199 }

1200 }

1201 }

1202 }

1203 }

1204 }

1205 }

1206 }

1207 }

1208 }

1209 }

1210 }

1211 }

1212 }

1213 }

1214 }

1215 }

1216 }

1217 }

1218 }

1219 }

1220 }

1221 }

1222 }

1223 }

1224 }

1225 }

1226 }

1227 }

1228 }

1229 }

1230 }

1231 }

1232 }

1233 }

1234 }

1235 }

1236 }

1237 }

1238 }

1239 }

1240 }

1241 }

1242 }

1243 }

1244 }

1245 }

1246 }

1247 }

1248 }

1249 }

1250 }

1251 }

1252 }

1253 }

1254 }

1255 }

1256 }

1257 }

1258 }

1259 }

1260 }

1261 }

1262 }

1263 }

1264 }

1265 }

1266 }

1267 }

1268 }

1269 }

1270 }

1271 }

1272 }

1273 }

1274 }

1275 }

1276 }

1277 }

1278 }

1279 }

1280 }

1281 }

1282 }

1283 }

1284 }

1285 }

1286 }

1287 }

1288 }

1289 }

1290 }

1291 }

1292 }

1293 }

1294 }

1295 }

1296 }

1297 }

1298 }

1299 }

1300 }

1301 }

1302 }

1303 }

1304 }

1305 }

1306 }

1307 }

1308 }

1309 }

1310 }

1311 }

1312 }

1313 }

1314 }

1315 }

1316 }

1317 }

1318 }

1319 }

1320 }

1321 }

1322 }

1323 }

1324 }

1325 }

1326 }

1327 }

1328 }

1329 }

1330 }

1331 }

1332 }

1333 }

1334 }

1335 }

1336 }

1337 }

1338 }

1339 }

1340 }

1341 }

1342 }

1343 }

1344 }

1345 }

1346 }

1347 }

1348 }

1349 }

1350 }

1351 }

1352 }

1353 }

1354 }

1355 }

1356 }

1357 }

1358 }

1359 }

1360 }

1361 }

1362 }

1363 }

1364 }

1365 }

1366 }

1367 }

1368 }

1369 }

1370 }

1371 }

1372 }

1373 }

1374 }

1375 }

1376 }

1377 }

1378 }

1379 }

1380 }

1381 }

1382 }

1383 }

1384 }

1385 }

1386 }

1387 }

1388 }

1389 }

1390 }

1391 }

1392 }

1393 }

1394 }

1395 }

1396 }

1397 }

1398 }

1399 }

1400 }

1401 }

1402 }

1403 }

1404 }

1405 }

1406 }

1407 }

1408 }

1409 }

1410 }

1411 }

1412 }

1413 }

1414 }

1415 }

1416 }

1417 }

1418 }

1419 }

1420 }

1421 }

1422 }

1423 }

1424 }

1425 }

1426 }

1427 }

1428 }

1429 }

1430 }

1431 }

1432 }

1433 }

1434 }

1435 }

1436 }

1437 }

1438 }

1439 }

1440 }

1441 }

1442 }

1443 }

1444 }

1445 }

1446 }

1447 }

1448 }

1449 }

1450 }

1451 }

1452 }

1453 }

1454 }

1455 }

1456 }

1457 }

1458 }

1459 }

1460 }

1461 }

1462 }

1463 }

1464 }

1465 }

1466 }

1467 }

1468 }

1469 }

1470 }

1471 }

1472 }

1473 }

1474 }

1475 }

1476 }

1477 }

1478 }

1479 }

1480 }

1481 }

1482 }

1483 }

1484 }

1485 }

1486 }

1487 }

1488 }

1489 }

1490 }

1491 }

1492 }

1493 }

1494 }

1495 }

1496 }

1497 }

1498 }

1499 }

1500 }

1501 }

1502 }

1503 }

1504 }

1505 }

1506 }

1507 }

1508 }

1509 }

1510 }

1511 }

1512 }

1513 }

1514 }

1515 }

1516 }

1517 }

1518 }

1519 }

1520 }

1521 }

1522 }

1523 }

1524 }

1525 }

1526 }

1527 }

1528 }

1529 }

1530 }

1531 }

1532 }

1533 }

1534 }

1535 }

1536 }

1537 }

1538 }

1539 }

1540 }

1541 }

1542 }

1543 }

1544 }

1545 }

1546 }

1547 }

1548 }

1549 }

1550 }

1551 }

1552 }

1553 }

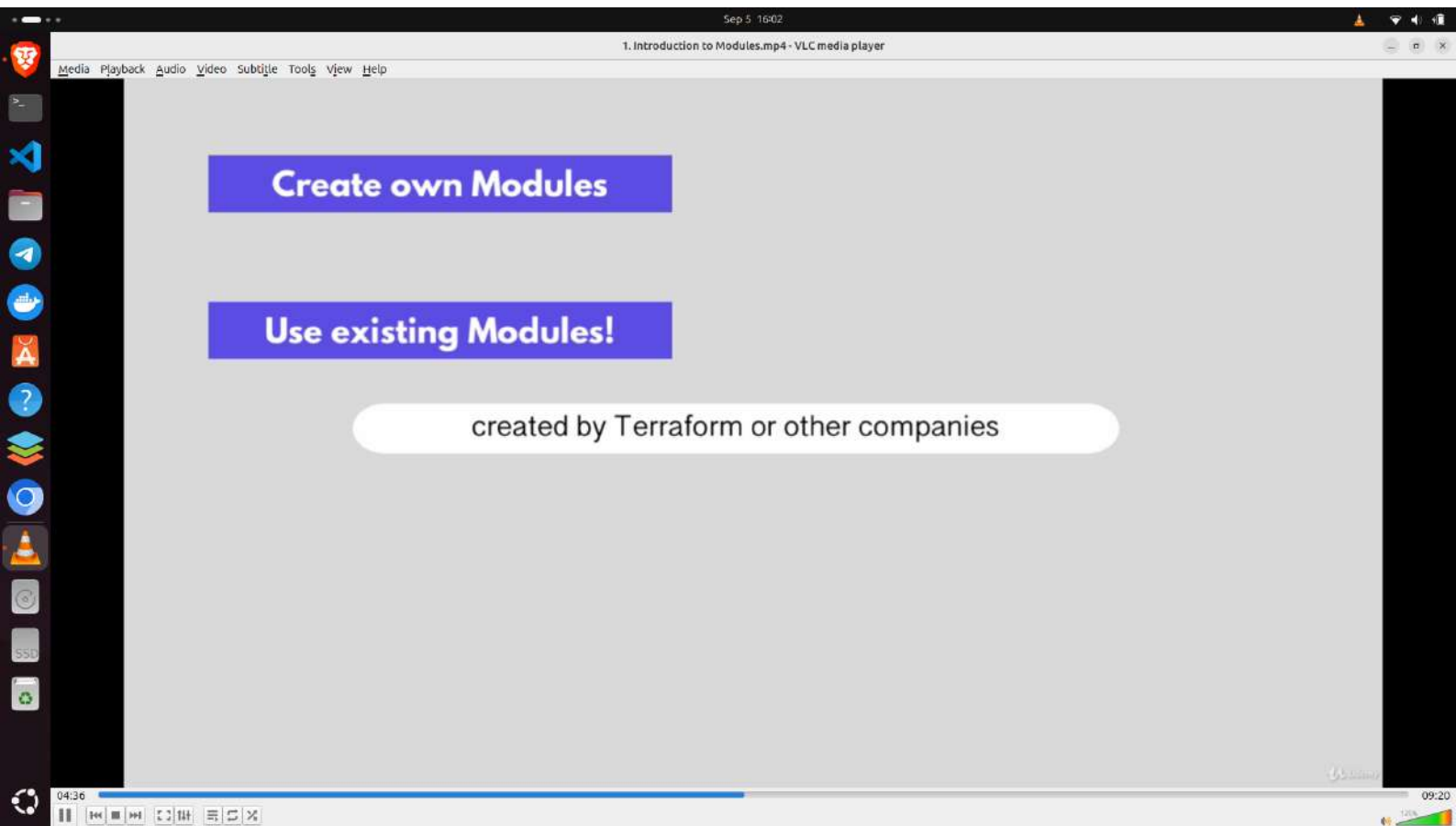
1554 }

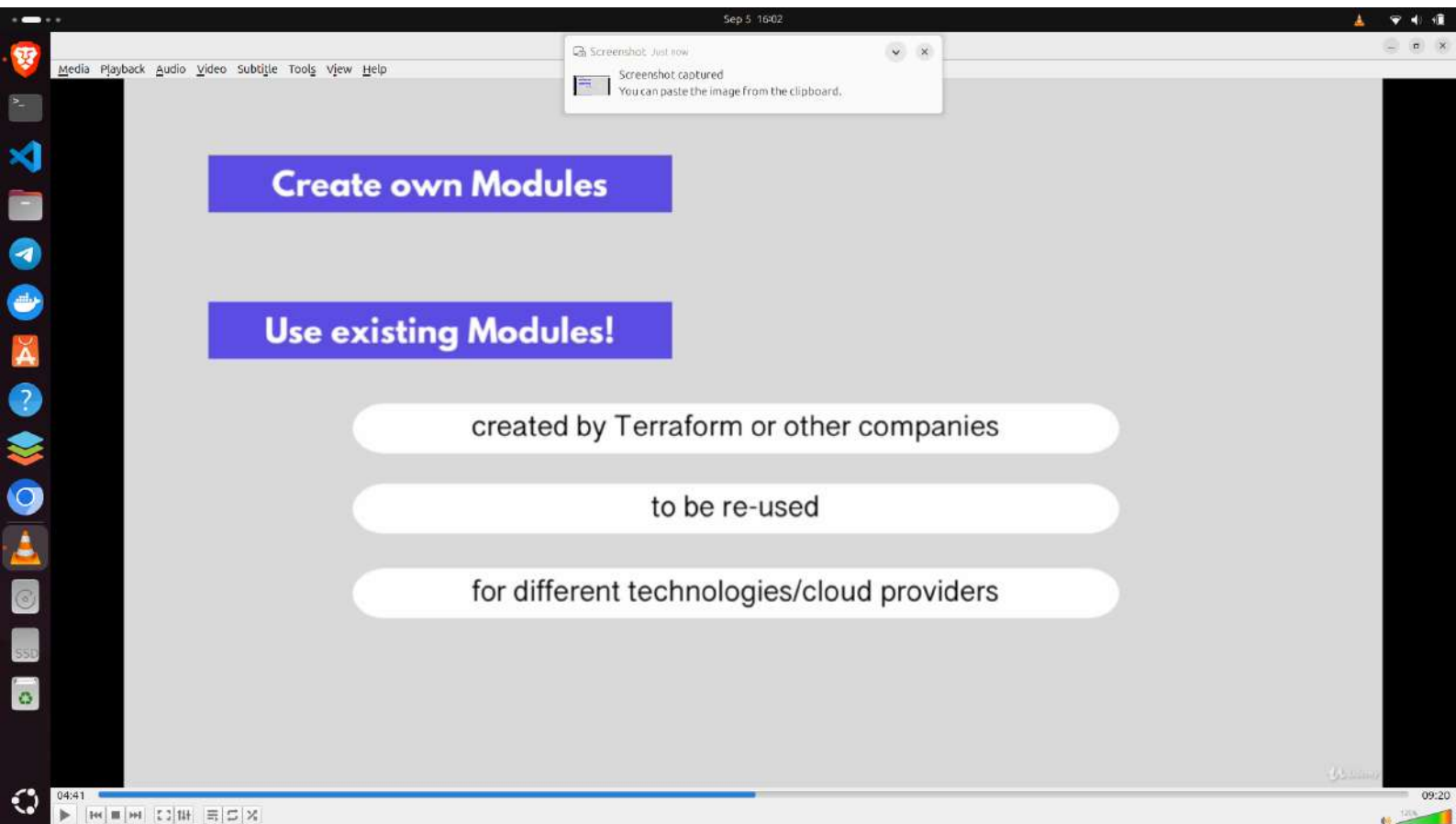
1555 }

1556 }

1557 }

1





Sep 5 16:02

1. Introduction to Modules.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

Create own Modules

Use existing Modules!

created by Terraform or other companies

to be re-used

for different technologies/cloud providers

terraform-aws-modules / vpc

Terraform module which creates VPC resources on AWS

a month ago 6.5M

aws provider

terraform-aws-modules / iam

Terraform module which creates IAM resources on AWS

15 14 days ago 1.0M

aws provider

terraform-google-modules / kubernetes-engine

A Terraform module for configuring GKE clusters

2 days ago 440.8K

google provider

04:54 09:20 120%

Sep 5 16:04

2. Terraform Project Structure - Cleanup our Main Configuration File.mp4 - VLC media player

Media Playback Audio Video Subtitle Tools View Help

EXPLORER

OPEN EDITORS

TERRAFORM

- > .terraform
- > .gitignore
- .terraform.lock.hcl
- entry-script.sh
- main.tf
- README.md
- terraform.tfstate
- terraform.tfstate.bac...
- terraform.tfvars

main.tf

```
1 provider "aws" {
2     region = "eu-west-3"
3 }
4
5 variable vpc_cidr_block {}
6 variable subnet_cidr_block {}
7 variable avail_zone {}
8 variable env_prefix {}
9 variable my_ip {}
10 variable instance_type {}
11 variable public_key_location {}
12
13 resource "aws_vpc" "myapp-vpc" {
14     cidr_block = var.vpc_cidr_block
15     tags = {
16         Name: "${var.env_prefix}-vpc"
17     }
18 }
19
20 resource "aws_subnet" "myapp-subnet-1" {
21     vpc_id = aws_vpc.myapp-vpc.id
22     cidr_block = var.subnet_cidr_block
23     availability_zone = var.avail_zone
24     tags = {
25         Name: "${var.env_prefix}-subnet-1"
26     }
27 }
```

project structure

- main.tf
- variables.tf
- outputs.tf
- providers.tf

Ln 5, Col 1 (178 selected) Spaces: 4 UTF-8 LF Terraform

01:34 04:19

project structure

- root module
- /modules = "child modules"

"child module" - a module that is called by another configuration

```
me: "${var.env_prefix}-igw"
```

group multiple resources into a **logical** unit

```
aws_def...  
t_route_table_id = aws_vpc.myapp-vpc.default_route_table_id
```

6