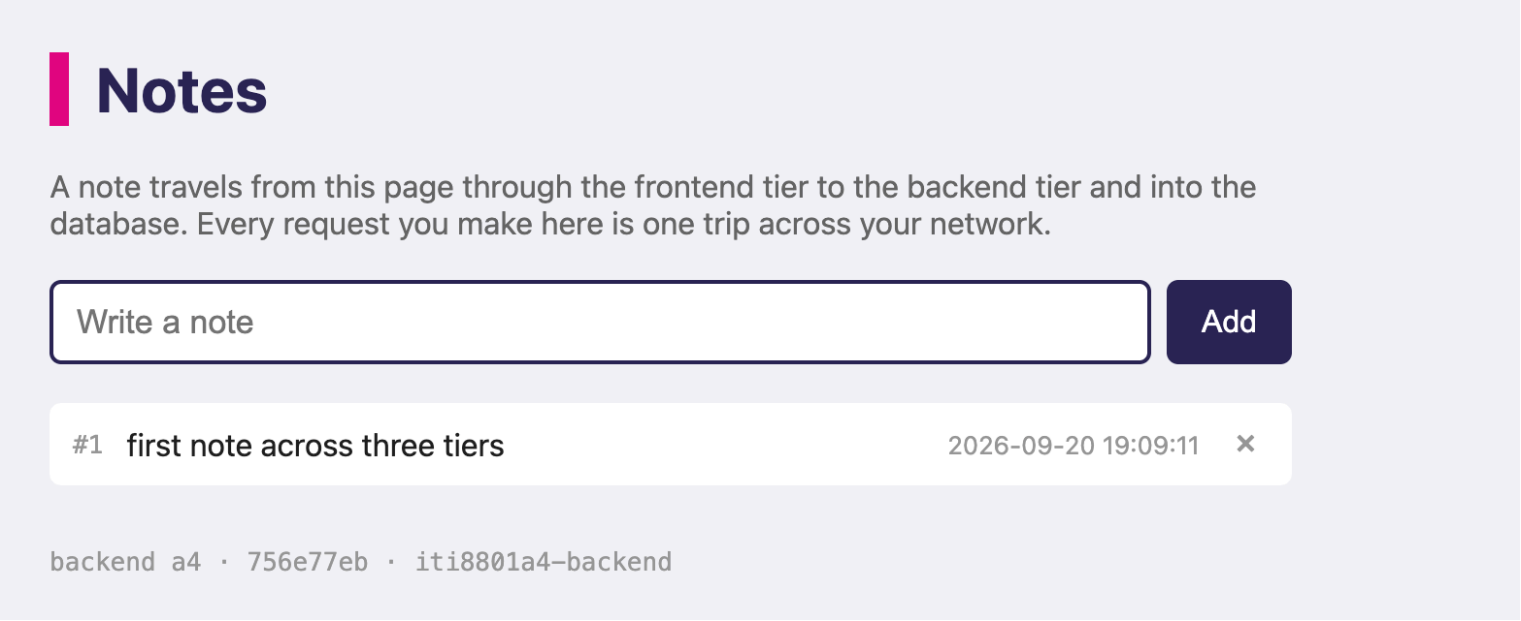
# ITI8801 · Cloud Architectures and DevOps

## Practice 4 · Assignment 4: the three-tier deployment

Siim Rebane · TalTech · Autumn 2026

# What you build

The Notes application on your cloud: two frontend machines behind a load balancer, one backend machine, one managed PostgreSQL database, each tier in its own subnet. The load balancer's address is the only public address. Written as Terraform, in modules, with the state in a remote backend.



# Machines

| Machine | Count | Subnet | Address | Runs | Port |
|---|---|---|---|---|---|
| frontend | 2 | app | `.10`, `.11` | `ghcr.io/siimrebane/iti8801-notes:a4`, `MODE=frontend` | 80 |
| backend | 1 | service | `.10` | the same image, `MODE=api` | 8080 |
| database | 1 | db | assigned by the cloud | managed PostgreSQL 16, smallest size | 5432 |

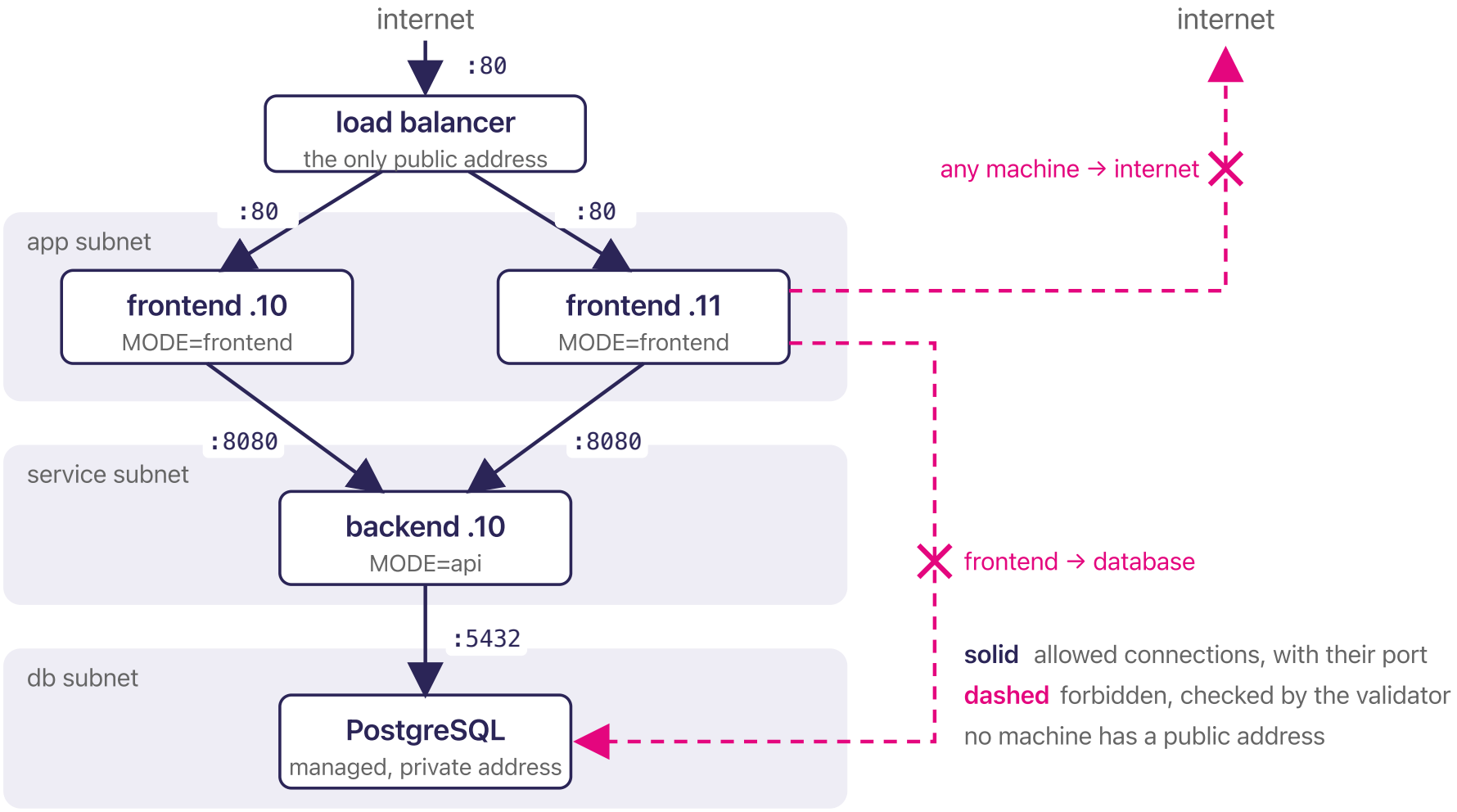
No machine has a public IP. The image's README (public repository, `app/`) lists the environment variables each mode needs; both modes run with `--network host`.

# Who may reach what

| From | To | Port |
|---|---|---|
| the internet | the load balancer | 80 |
| the load balancer | the frontends | 80 |
| the app subnet | the backend | 8080 |
| the service subnet | the database | 5432 |

Everything else is denied. In the finished deployment no machine can open a connection to the internet, and port 22 does not answer on any public address.

# The same, drawn



# Addresses

| | Azure | AWS |
|---|---|---|
| network | `10.60.0.0/16` | `10.61.0.0/16` |
| app subnet | `10.60.1.0/24` | `10.61.10.0/24` |
| service subnet | `10.60.3.0/24` | `10.61.30.0/24` |
| db subnet | `10.60.2.0/24` | `10.61.20.0/24` and `10.61.21.0/24` |
| public subnets | none | `10.61.0.0/24` and `10.61.1.0/24` |

AWS: the load balancer and a NAT gateway need public subnets; a managed database needs two db subnets in two zones.

# Limits

| | Azure | AWS |
|---|---|---|
| vCPUs per region | 6 | ample |
| machine size | the free-tier B-series size of your region, 2 vCPUs | `t3.micro` |

# The configuration

- at least two `module` blocks in the root module
- a `backend "azurerm"` or `backend "s3"` block; the storage account or bucket exists before `terraform init`
- no password, prefix or key in git: a tfvars file that is not committed

# The validator

`https://iti8801.digivader.com/lab4-validator/` : your student code and your load balancer's address, the IP on Azure, the DNS name on AWS.

What it does with that, through your load balancer: about thirty requests to see which frontends answer and whether they are the course image; one note written, read back and deleted; the backend's report read through the frontend's `/api` forwarding; a knock on port 22. Then seven boxes, each with the reason when it is red. Check as often as you like; every check has its own code.

# The seven boxes

| Box | Green when |
|---|---|
| Address is in a cloud | the load balancer's addresses belong to Azure's or AWS's published ranges |
| Frontend tier | exactly two distinct frontends answer, signed by the course image, both with an address in the app subnet |
| Notes round trip | `POST /api/notes` answers 201, the note is in `GET /api/notes`, `DELETE` returns 204 |
| Backend tier | the backend's report says SQL works, the database it dialled is in the db subnet, its own address is in the service subnet |
| Isolation | no frontend reaches the database port; no machine reaches the internet |
| SSH closed | port 22 on the load balancer's address does not answer |
| Configuration | your upload is accepted |

The first six are read live. The seventh is your `.tf` files, uploaded on the page once the six are green.

# What the upload must contain

Every `.tf` file: the root module and every module's files together, plus your boot scripts if you want. Not the state, not tfvars, not keys. A refused upload lists every reason and stores nothing; fix, and upload again on a fresh check. Accepted when:

- every `.tf` file is Terraform (a scanner checks the block structure)
- at least two `module` blocks and one `backend` block exist
- the resource types of all three tiers are present for your cloud:

| Azure | AWS |
| --- | --- |
| `azurerm_linux_virtual_machine`, `azurerm_lb`, `azurerm_postgresql_flexible_server`, `azurerm_network_security_group`, `azurerm_subnet` | `aws_instance`, `aws_lb`, `aws_db_instance`, `aws_security_group`, `aws_subnet` |

# The application's contract

Environment variables the image reads, by mode. Both modes run with `--network host`.

| Variable | Mode | Value |
|---|---|---|
| `MODE` | both | `frontend` or `api` |
| `API_URL` | frontend | `http://<backend address>:8080` |
| `DATABASE_URL` | api | `postgres://<user>:<password>@<db host>:5432/<db name>?sslmode=require` |
| `DB_TARGET` | **both** | `<db host>:5432`. The api must reach it; the frontend must not, and its report is how the validator knows |
| `PORT` | both | optional; 80 for the frontend, 8080 for the api |

The user, password and database name are whatever you define on the managed server; the api creates its own table. A managed PostgreSQL refuses connections without `sslmode=require`. Without `DB_TARGET` a frontend's report has no database check, and the isolation box is red with "no DB_TARGET given".

# Where to look things up

- The provider's documentation: `registry.terraform.io/providers/hashicorp/azurerm` and `…/hashicorp/aws`. Every resource type has a page with every argument and, at the bottom, the import id format.
- The application's README: what `MODE=frontend` and `MODE=api` need, what `/report` says.
- Lecture 3 for the language, lecture 4 for modules, backends and what the verbs mean when things disagree.
- The portal, to see what got built and to read a load balancer's health status; not to change anything Terraform owns.

# Looking inside a machine that has no way in

There is no SSH. When a box is red and the reason points at a machine, ask the platform to run a command inside it:

Azure:

```
az vm run-command invoke -g <resource group> -n <vm name> \
    --command-id RunShellScript --scripts 'docker ps -a; docker logs notes | tail -20'
```

AWS, if the instance has a role with `AmazonSSMManagedInstanceCore` and the NAT gateway exists:

```
aws ssm send-command --instance-ids <id> --document-name AWS-RunShellScript \
    --parameters commands='docker ps -a; docker logs notes | tail -20'
```

`cloud-init status` and `/var/log/cloud-init-output.log` say whether the boot script ran and what it printed. The most common finding: the container was never created because the boot script failed one line earlier.

# When a box is red: what the reason usually means

| Reason | Usual cause |
| --- | --- |
| no signed frontend answered | the load balancer's probe fails and it has no healthy backend; or the container is not running; or the wrong mode is behind the LB |
| one frontend of two | one machine's boot script failed, or the LB's pool has one member |
| address not in the app subnet | the machine is in another subnet, or reports the container's address because `--network host` is missing |
| POST answered 502 | the frontend cannot reach the backend: wrong `API_URL`, or the service subnet's rule does not allow 8080 from the app subnet |
| SQL fails | wrong `DATABASE_URL`; the database's rule does not allow the service subnet; the database is still being created |

# When a box is red, continued

| Reason | Usual cause |
| --- | --- |
| frontend reaches the database | the database rule allows the app subnet, or the whole network |
| a machine reaches the internet | a NAT gateway, a public IP or a default route still exists |
| the page never loads in the browser, but the validator's frontend box is green | the browser tried `https://` first and waited for it to time out; type `http://` in front of the address |
| port 22 answers | a machine has a public IP, or a rule opens 22 to the internet |

# What to submit, and when

- The validator's code of a check that is **PASSED**: all six live boxes green and the configuration accepted. Paste it into Moodle.
- Deadline: Sunday next week, 23:59, as for every assignment.
- Then `terraform destroy`. The state's storage can stay.



Questions