Observability & Monitoring

See Everything, Know Everything

Where We Are Now

- **Container Apps** running on Azure with your frontend and backend
- May have applications deployed in **Virtual Networks** with proper isolation
- **CI/CD pipelines** automatically building and deploying
- **Infrastructure as Code** with Terraform managing everything
- **Secrets** secured in Azure Key Vault

How Do You Know If It's All Working?

> *Your Container Apps are deployed. But are they healthy? Are users happy? When things break, can you figure out why? What's happening inside those containers?*

Why Observability Matters

Last week: More simple to debug, local logs and quick small changes to fix things

Now Complex to understand, cloud architecture that takes time to deploy small changes, with several moving parts, access levels, reliance on cloud providers, etc etc etc.

> ***When something breaks, where do you even start looking? How do you see inside those containers?***

Without observability:

- "Something's broken, but we don't know what"
- Hours spent guessing where the problem is
- Customers report issues before you know about them
- Fixing one thing breaks something else

The Three Pillars of Observability

Metrics

Logs

Traces

Health Checks: Is My Container App Alive?

Container Apps uses health checks to know if your app is working

Every Container App should have health check endpoints. Azure uses these to:

- Know when your app is ready to receive traffic
- Detect when your app has crashed
- Automatically restart unhealthy containers
- Route traffic away from failing replicas

Structured Logging will save you at 2AM

Bad logging:

```
console.log("User login failed");
```

Good logging

```
logger.error("User authentication failed", {
  event: "login_failed",
  user_id: userId,
  ip_address: req.ip,
  error_code: "INVALID_CREDENTIALS",
  timestamp: new Date().toISOString(),
  containerRevision: process.env.CONTAINER_APP_REVISION,
  replica: process.env.CONTAINER_APP_REPLICA_NAME
});
```

Alerting

The Golden Rule: Alert on What Matters

Good alerts:

- Tell you when customers are affected
- Give you time to fix things before they get worse
- Include enough information to start investigating

What to monitor in every application:

Signal	What it means	Container App Alert Example
Latency	How fast responses are	"Response time > 500ms"
Traffic	How many requests	"Traffic dropped 50%"
Errors	How many things are failing	"Container restart rate high"
Saturation	How full your resources are	"CPU > 80% for 5 minutes"
Availability	Is the service up?	"Zero healthy replicas"

> ***If you monitor these five things, you'll catch most problems***

Start with the basics:

- Health check endpoints in all your applications
- Structured logging with correlation IDs
- Simple metrics dashboard
- Basic alerts for service down/high error rate