

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

Background

When Terrateam was just starting out, like any other startup, we wanted to move fast to

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

In preparation for standing up a staging environment, we decided to map out all of the pieces of our existing AWS infrastructure in an attempt to match them to Fly.io components. We wanted to make sure we weren't missing anything.

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

```
restart_limit = 0
timeout = "3s"
tls_skip_verify = false
[services.http_checks.headers]
```

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

enforces a secure connection with SSL. We could have easily created an override, but we felt like it was good practice to configure Postgres with SSL.

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

further influenced us to pursue the not-so-fancy quick migration with minimal downtime pathway.

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

worry about CIDRs, subnets, routing, and anything else that comes along with the complexity of networking. It just works.

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

Docs >

Blog >

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

Logging

The container logging solution provided by Fly.io is basic. It's easy to view logs with the

Docs >

Blog ▾

[Atlantis to GitHub Actions](#)

[Safety First](#)

[Introducing Terrateam](#)

[Deploy Grafana to ECS with Terraform](#)

[Terraform Pre-Commit Hooks](#)

[Manage GitHub with Terraform](#)

[Terraform Registry Cache](#)

[Terraform Best Practices](#)

[Import Existing AWS Resources with Terraform](#)

[Running a GitHub Action](#)

[GitHub Action API Gotchas](#)

[Launching Without a GUI](#)

[Retrospective on Developing Terrateam in a Monorepo](#)

[Planning Doesn't Have To Lock](#)

[Terrateam Early Access](#)

[Announcing Infracost Support](#)

[What is IaC?](#)

[Terrateam vs. GitHub Actions](#)

[Flying away from AWS](#)

I'd love to see better support in the future. If I can't get a specific answer from staff, I at least would like to see a nudge in the right direction. Room for improvement.

Closing Thoughts

No platform is all rainbows and unicorns. Having said that, Fly.io gets pretty close. There are not a lot of complaints with the offering. It does help to understand what exactly the offering is though. I think Fly.io does a great job in explaining what they are and more importantly what they are not.

By design, our stack is simple, which is probably why Fly.io works so well for us. We don't require a lot of moving parts or infrastructure. We basically just need a place to run a database and a container.

Your mileage may vary. If your organization requires more comprehensive components from a cloud provider like messaging middleware, S3 buckets, IAM, etc., then Fly.io might not be for you.

In closing, we are overall very pleased with Fly.io and can recommend the platform to other teams that wish to move away from other complex and expensive cloud providers. Join the Terraform newsletter to stay updated!

root@localhost

Subscribe

< Terrateam vs. GitHub Actions

Announcing Access Control >

TERRATEAM

© Terrateam 2023