New Relic

nerd days 1.0

Reducing toil with Terraform

Miguel Mingorance
Systems Engineer
Delivery Hero

Common challenges when building infrastructure

- Manual intervention
- Disaster recovery
- Extensibility
- Replication
- Delivery time
- Track of resources
- Changes review

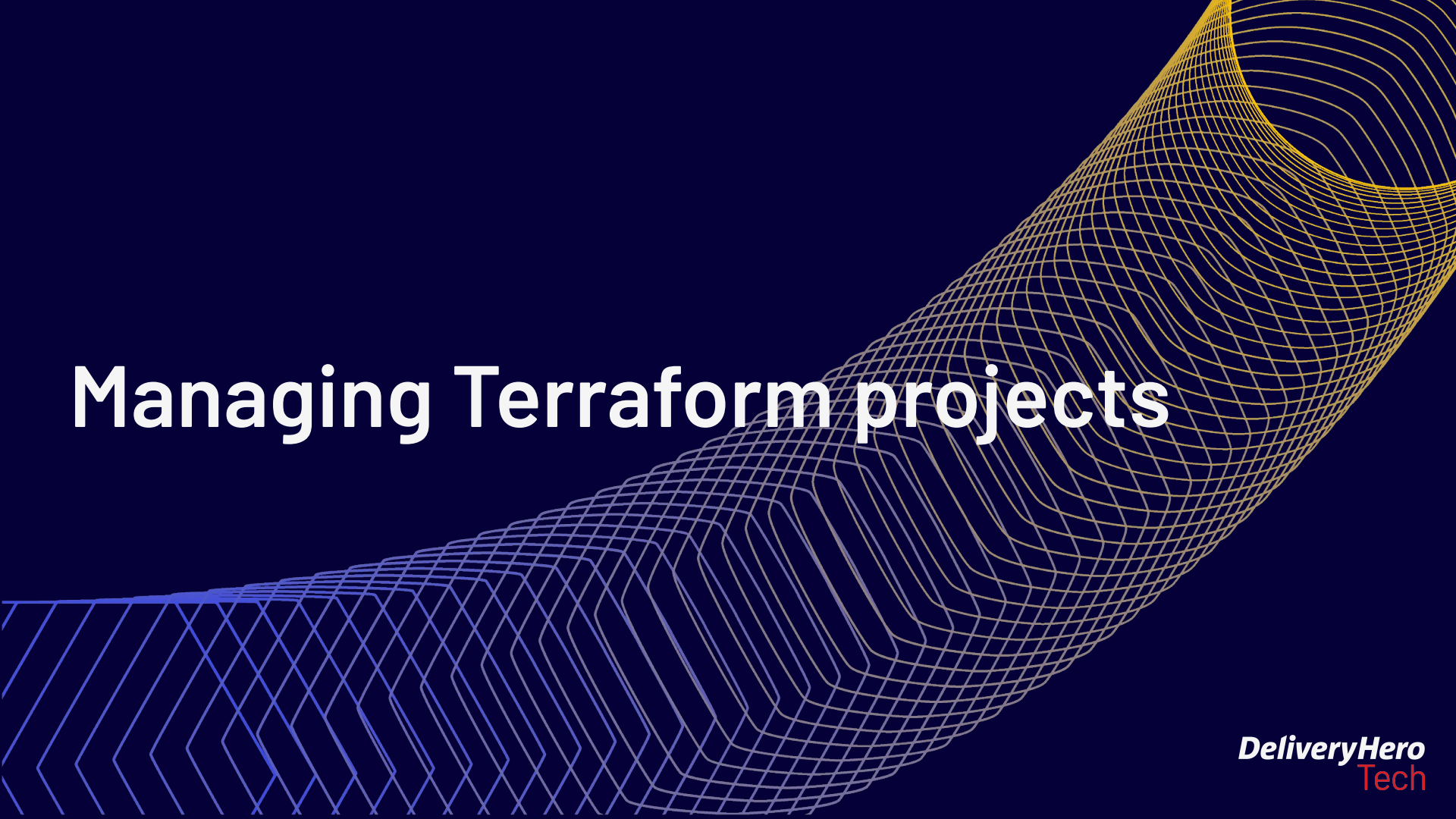
Introducing Terraform in the Organization

DeliveryHero Tech

How to adopt Terraform in the organization

1. Define clear responsibilities over the infrastructure
2. Isolate Terraform projects based on ownership
3. Choose the ***Terraform backend*** that suits you best
4. Create Terraform modules to share across the organization

Managing Terraform projects



DeliveryHero
Tech

Key points:

- Terraform state per resource type
- Ownership is over all teams or an infra team
- Standards are set in common or by an infra team
- Common repository for all the Terraform

Best option for:

- Organizations whose infrastructure and standards are managed by an infrastructure team or across all development teams
- Enabling a centralised self-service workflow
- Keeping consistency and common standards across all the infrastructure

Running Terraform on Remote

DeliveryHero Tech

Terraform on remote

Why to run on remote

- No need to keep your local machine running
- Review and approval can be set as required prior to apply
- No need to set permissions per user
- Avoid overlapping changes with “locks”
- Terraform always on latest state

How to run on remote

- Using ***Terraform Cloud*** as a managed solution
- Using ***Atlantis*** as a self-hosted solution

How to start with Atlantis

- Choose the deployment method it suits best to you:
  - *AWS Fargate*
  - *Helm Chart*
  - *Kustomize*
  - *GKE*
  - *Azure Container Instance*
  - *Roll your own*
- Configure the repos you want to use and go!

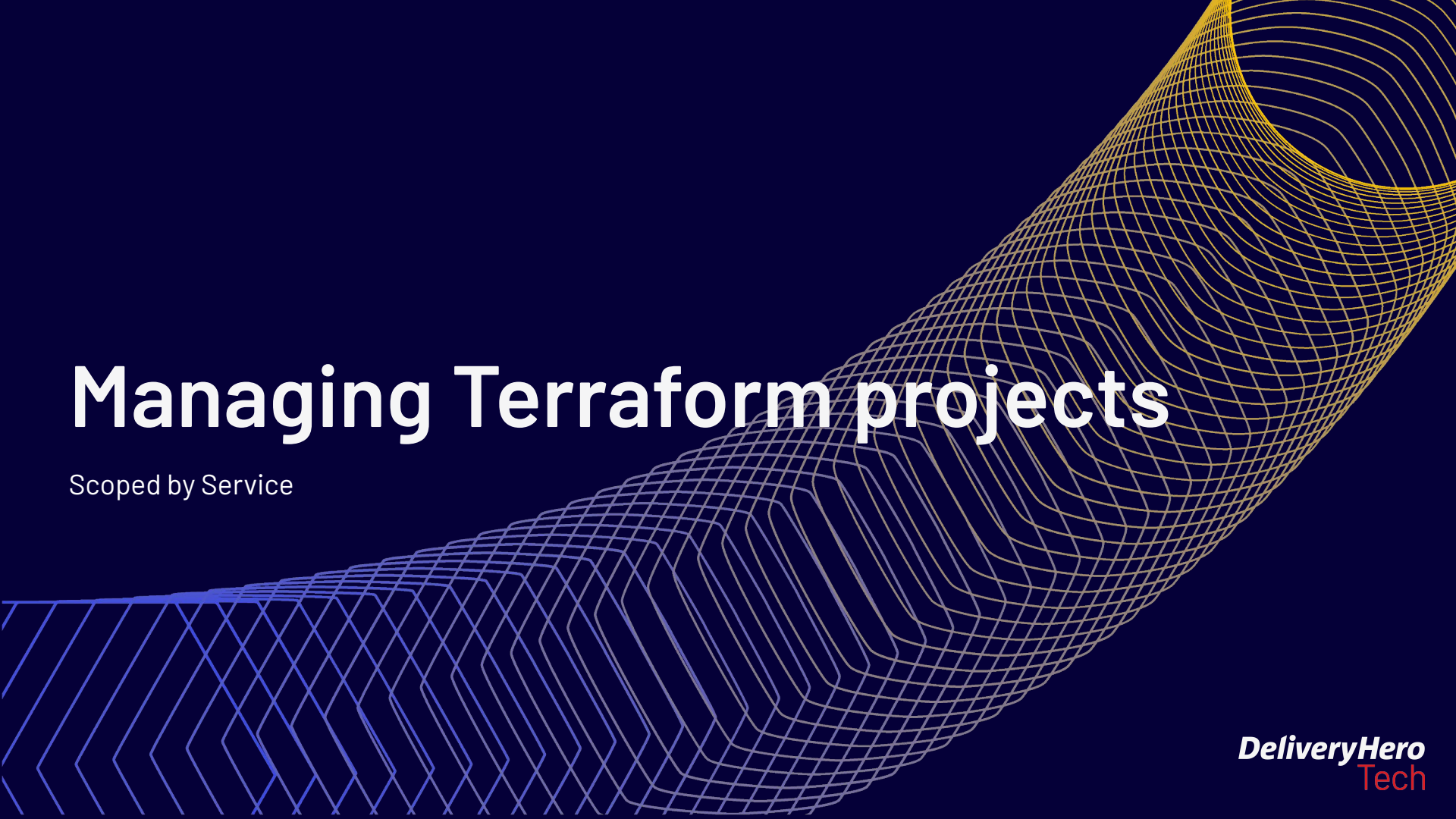
DeliveryHero Tech

DEMO TIME

DeliveryHero
Tech

Managing Terraform projects

Scoped by Service



DeliveryHero Tech

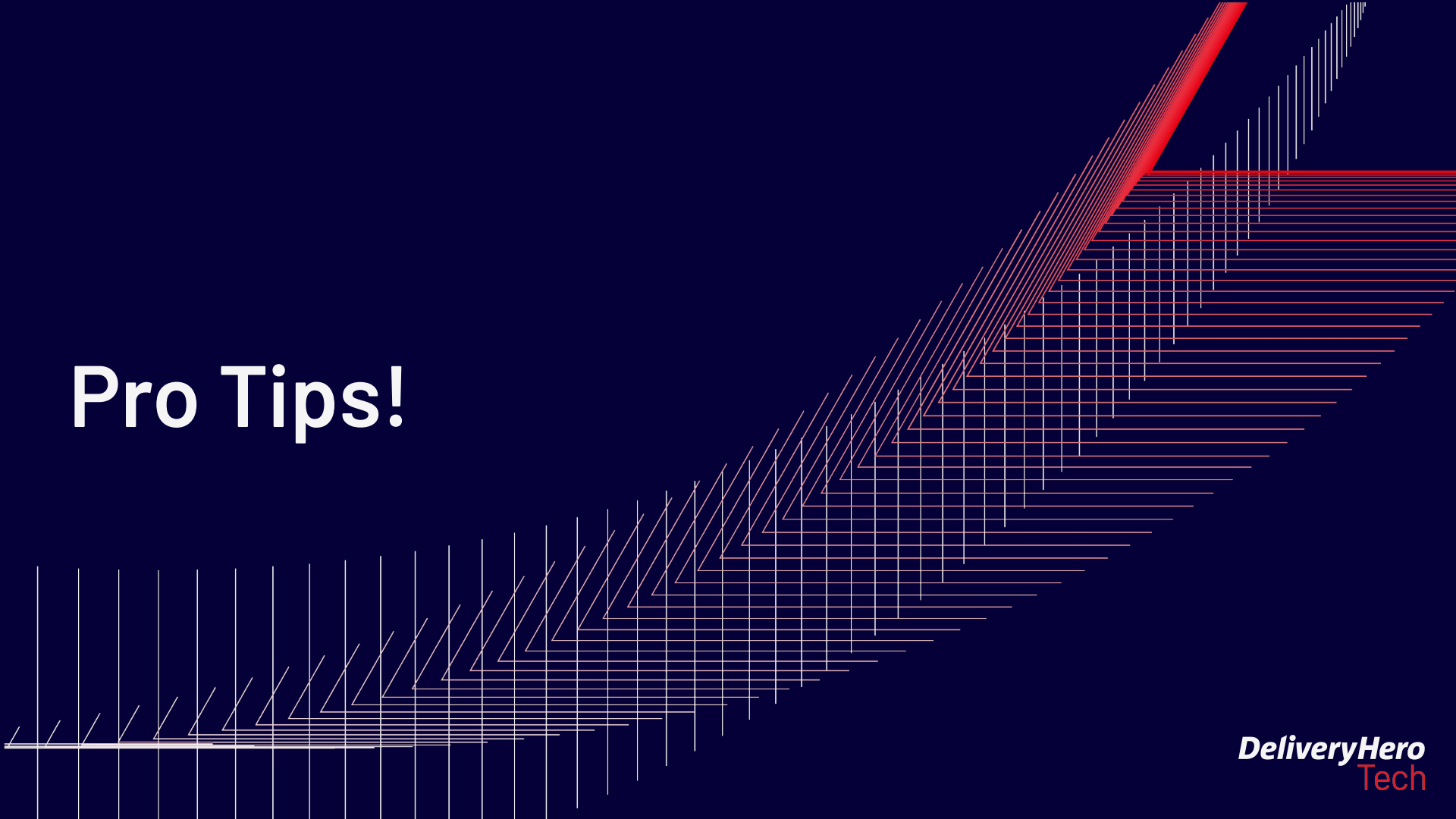
Key points:

- Terraform state per service
- Full ownership by the service owner's team
- Standards are set by the service owner's team
- Terraform can be in different repositories

Best option for:

- Organizations whose teams are fully responsible for their infrastructure and standards
- Keeping the Terraform in the same repo as the service's code
- Fast adoption of new cloud services and Terraform providers

Pro Tips!



DeliveryHero Tech

Pro Tips!

- Using variables in all of your Terraform code, allows you to quickly reuse your code within multiple environments.
- Structure your Terraform projects within different directories for each environment and/or location
- ***Workspaces*** allows you to use one single Terraform ***state file*** for multiple environments.
- Terraform has many ***providers***. Combine them together to build your entire infrastructure from a single place.
- Create your own Terraform provider and extend its flexibility.
- Use ***public*** Terraform modules
- ***tfswitch*** is a great tool to manage multiple Terraform versions in your local machine.

DeliveryHero Tech


DeliveryHero Tech
We are hiring!
careers.deliveryhero.com

https://github.com/deliveryhero/dh-nerddays-demo

DeliveryHero Tech

Q&A

Thanks For attending nerd days 1.0

Look out for your chance to request a swag pack in the follow-up email