DevOps project: Implementation of CI/CD for Piggy Metrics

Team members:
- Alejandro Valderrama
- Edgar Mendoza
- Omar Santoyo

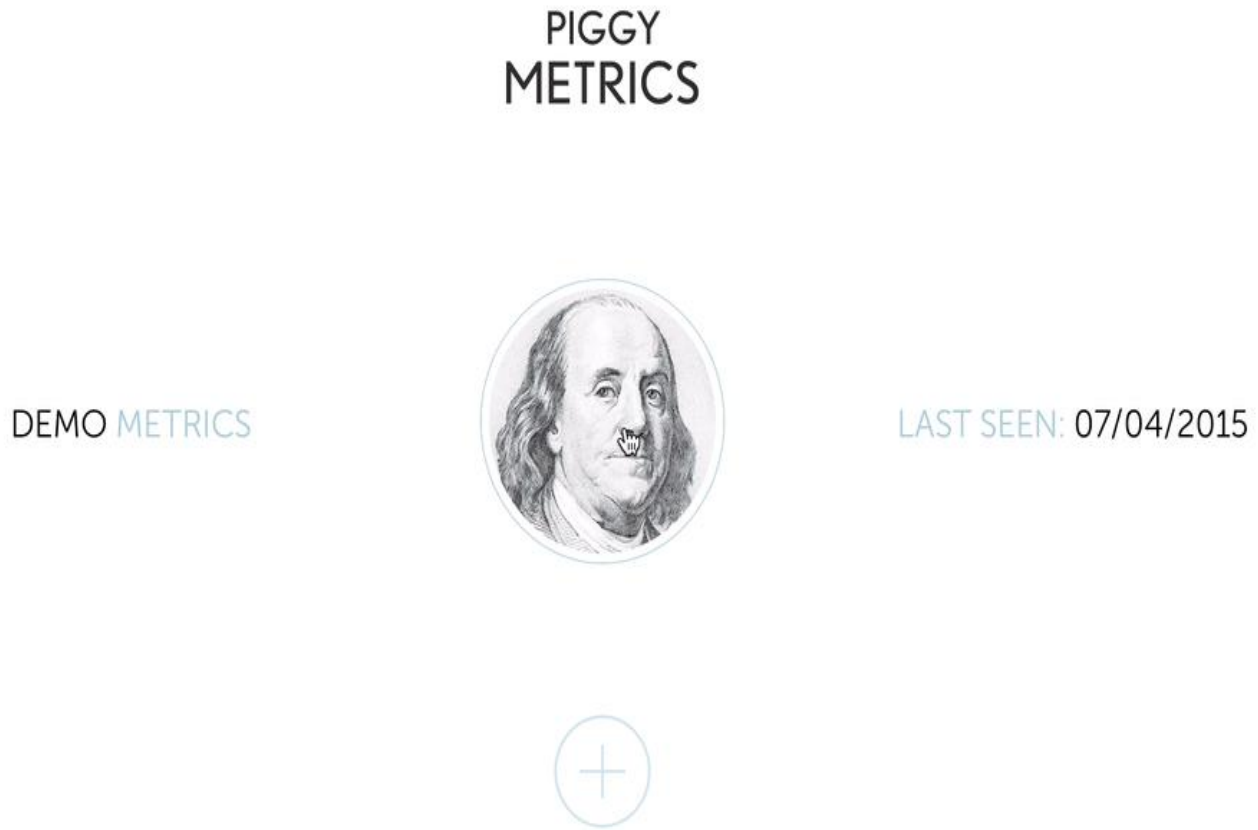
11/29/2021

Content

- Piggy Metrics app
- Why DevOps?
- DevOps strategy
- Tools & demos
- CI/CD
- Branching strategy
- Pipeline demo
- QA

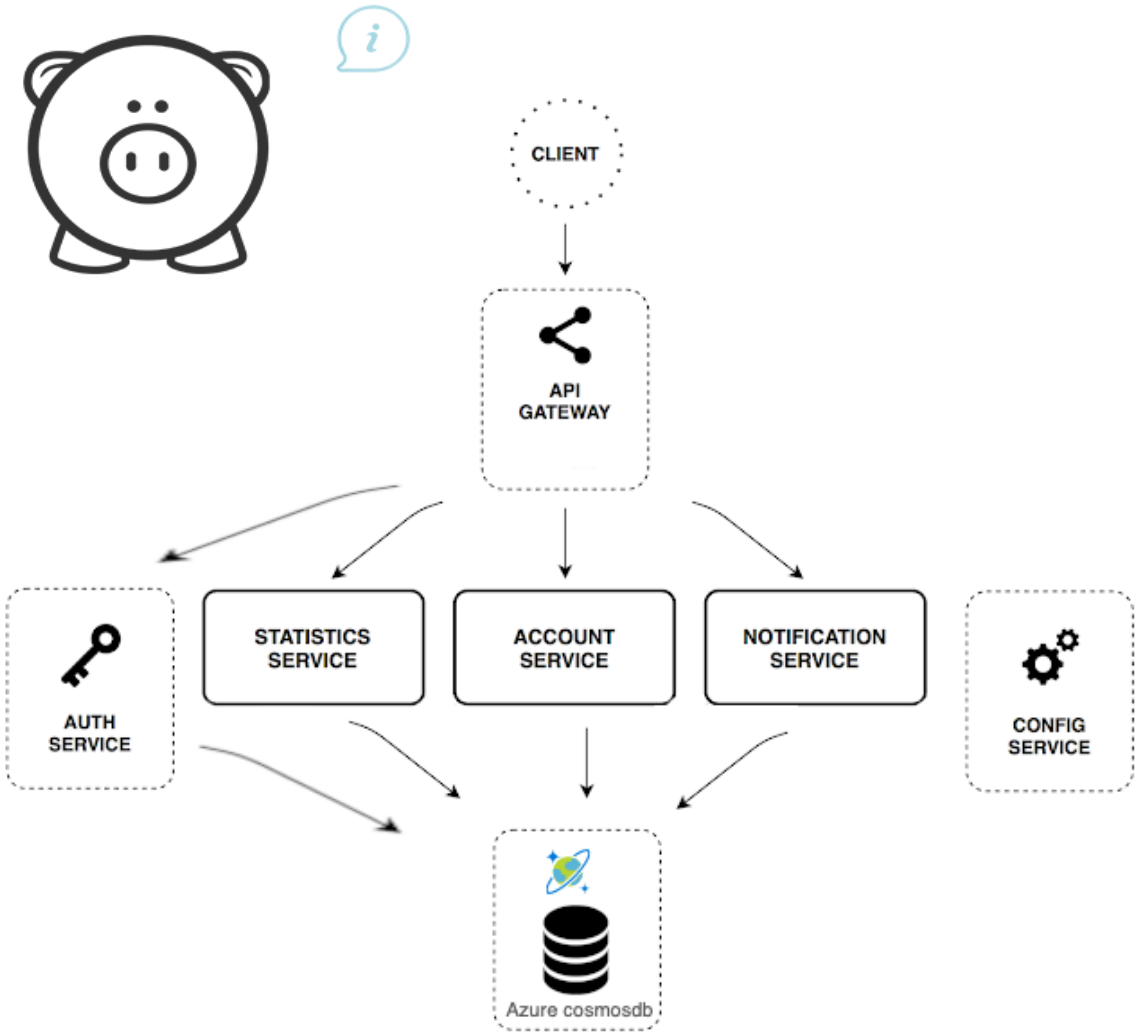
Tech Mahindra

Piggy Metrics



- Piggy Metrics is a Microservice Architecture financial advisor app
- Piggy Metrics gives us the ability to:
  - Performs calculations on major statistics parameters
  - Capture time series
  - Stores user contact information, incomes/expenses items, savings and notification setting

Piggy Metrics Architecture



- Account Service: Contains general input logic and validation: incomes/expenses items, savings and account settings

- Statistics Service: Performs calculations on major statistics parameters and captures time series for each account

- Notification Service: Stores user contact information and notification settings (reminders, backup frequency, etc.)

- Auth Service: Stores the authentication data into the database

- Config Service: Configures the spring framework used to enable the integration of all the other services

Why DevOps?

Our customers believe that it is crucial for their business to focus on rapid value creation so that they can offer users value as quickly as possible.

- What We will be delivering to our customer:
  - *Release deliverables more frequently, with higher quality and stability*
  - *Stabilize work environment*
  - *Continuous delivery of software*
  - *Automation in repetitive tasks leaves more room for innovation*
  - *Promotes agility in the business*

DevOps strategy

- Terraform is used to provision the infrastructure needed
- Azure Cloud is our pick for cloud provider
- Docker to build containerized images
- AKS to host the Kubernetes cluster
- Extensible testing in our pipeline using GitHub actions
- Helm to deploy our application

Robust

Distributed Architecture

Adaptable

Quick and easy develop Cycle that can deploy fast

Scalable

Easy to scale using Kubernetes and AKS

Easy Maintenance

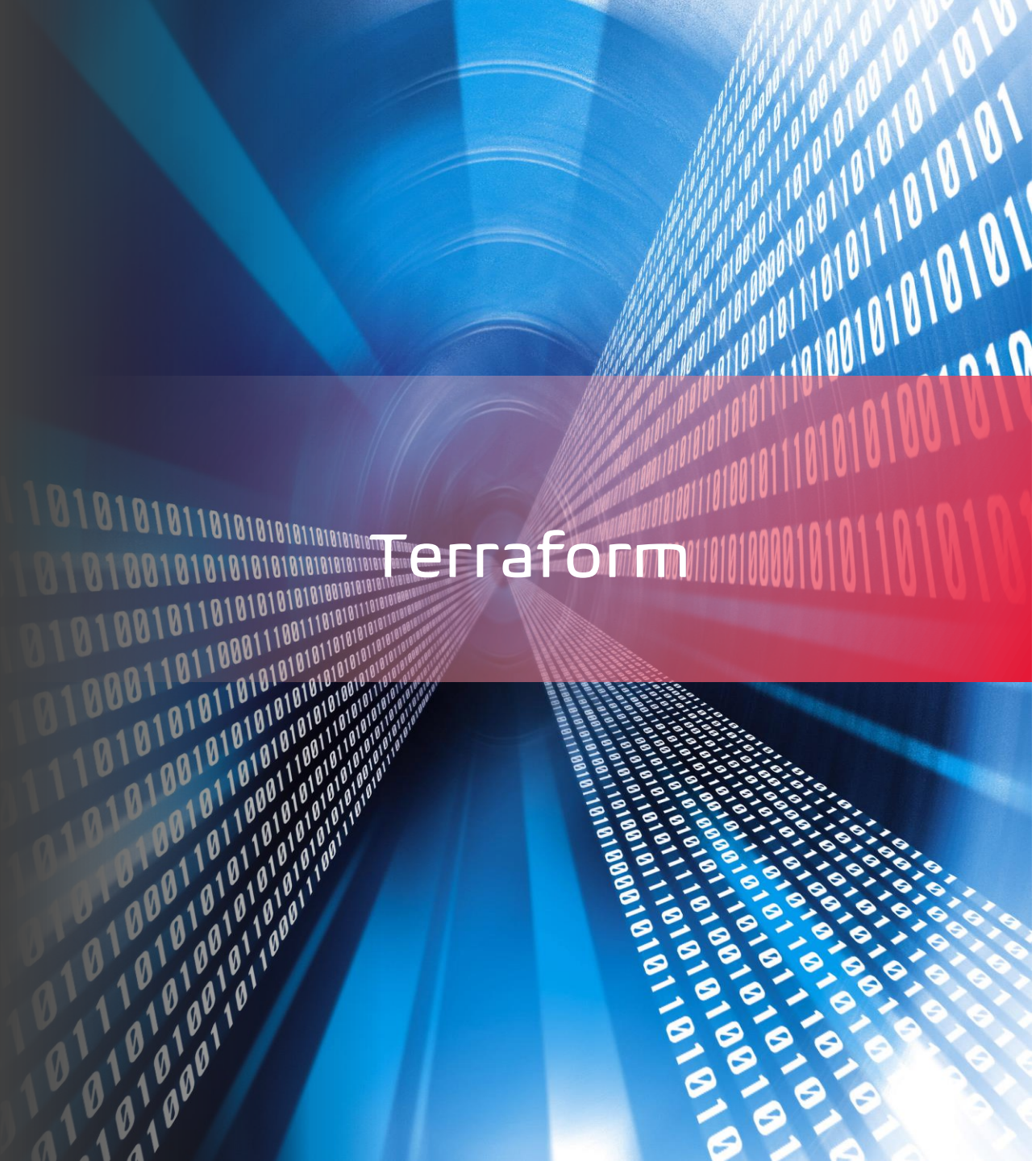
Deploy new changes Is easy and automatic

Easy provisioning

Modifications trouble-free

DevOps tools & demos

- Terraform
  - Azure infrastructure
- Docker
  - Jib
- Helm
- AKS
- Vault



Tech Mahindra

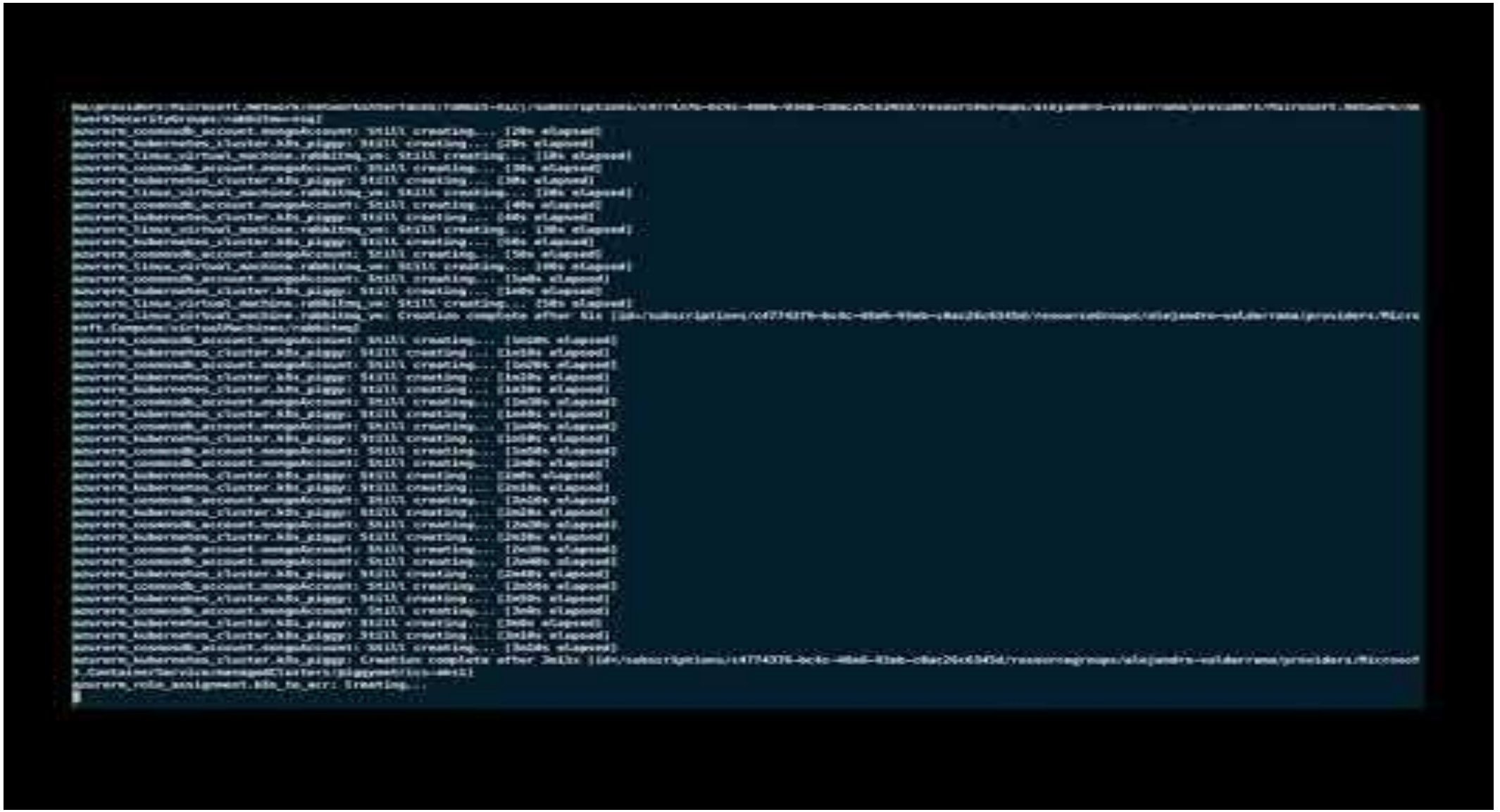
We use terraform to provision our infrastructure:

- **Terraform** is an open-source infrastructure as code software tool created by HashiCorp

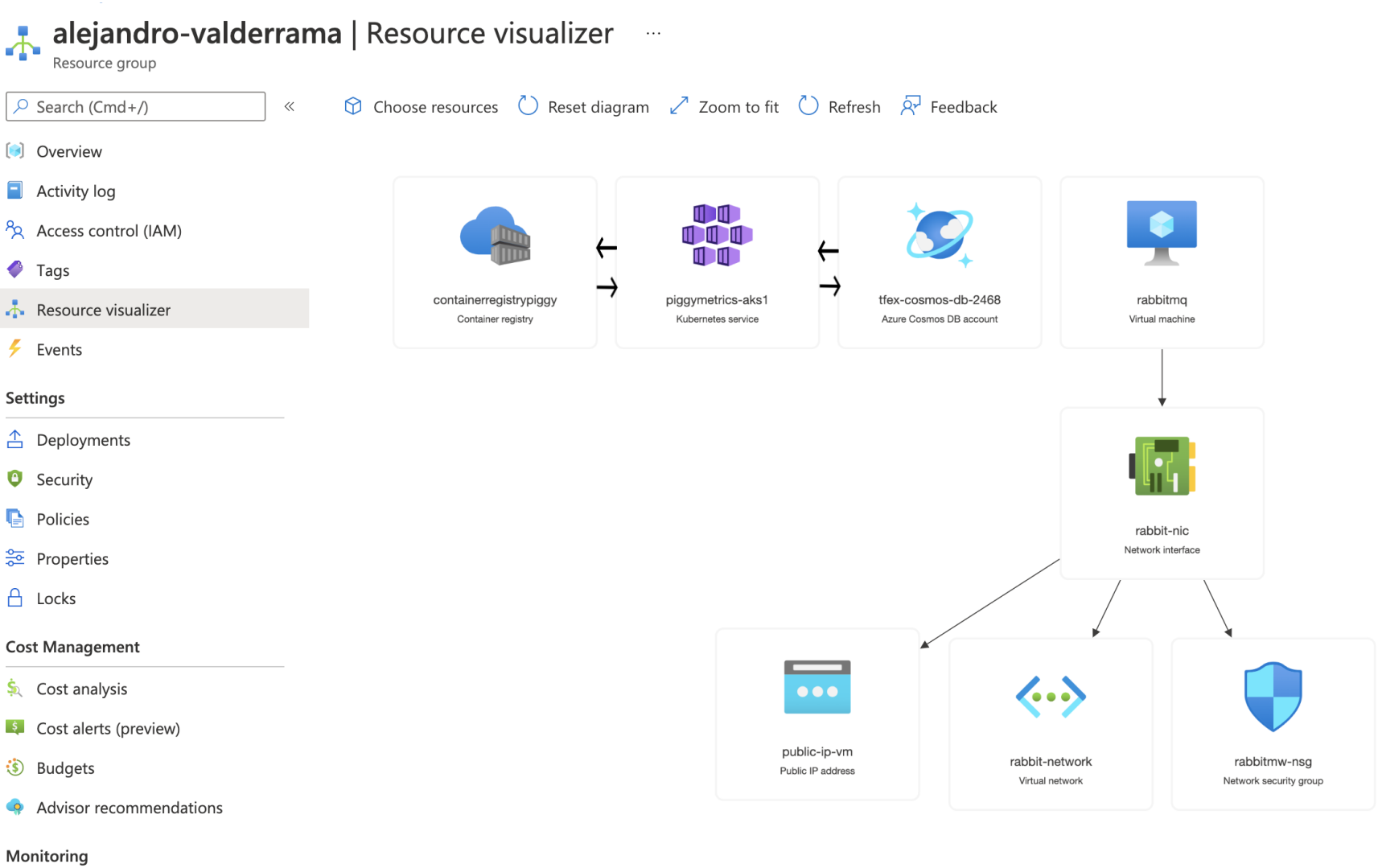
- Users define and provide data center infrastructure using a declarative configuration language known as HashiCorp Configuration Language (HCL)

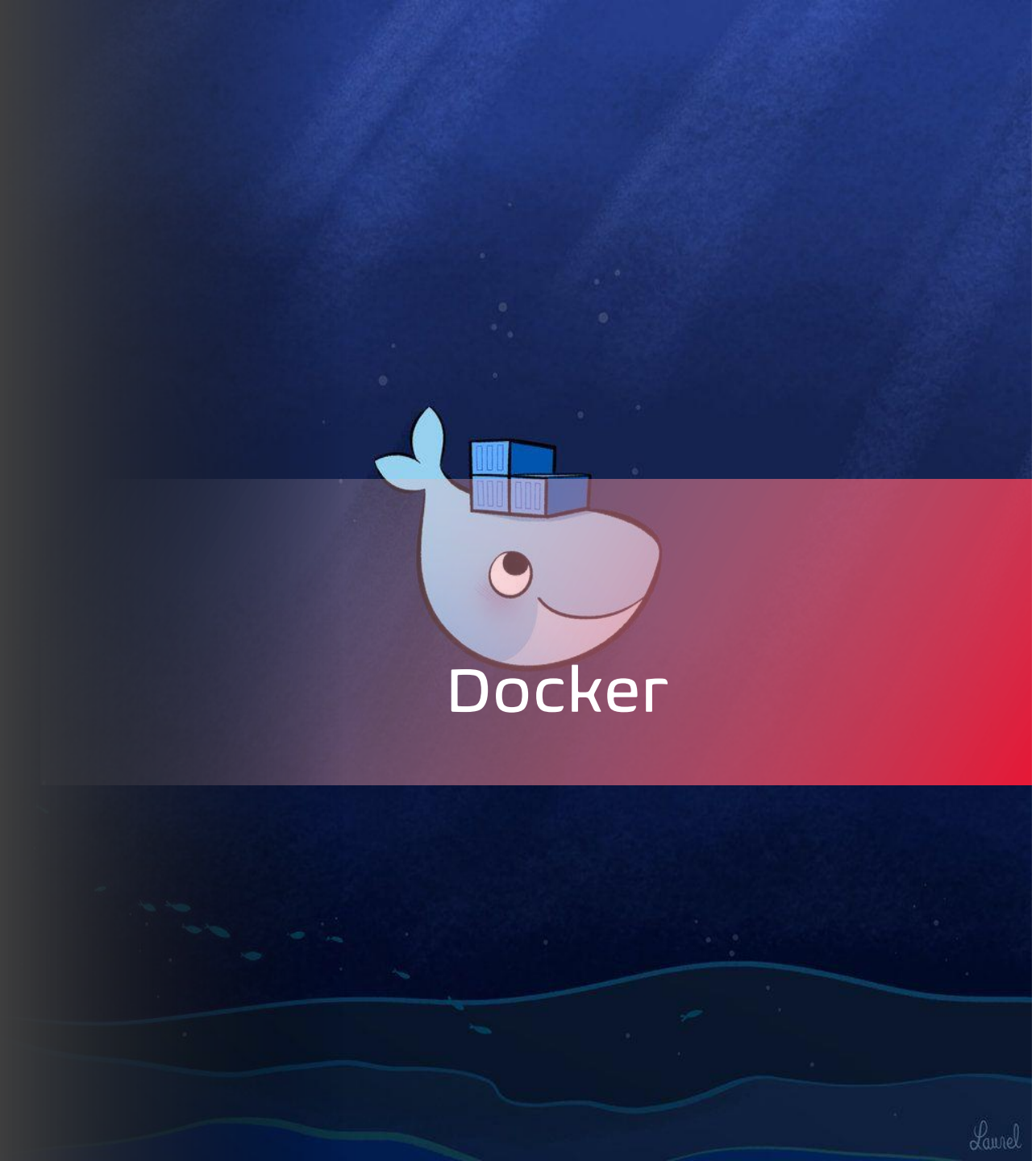
Demo terraform

Tech Mahindra



The piggymetrics infrastructure in Azure





Docker is an open source containerization platform:

- Docker provides the ability to package and run an application in a loosely isolated environment called a container
- Docker simplifies and accelerates the workflow, while giving developers the freedom to innovate with their choice of tools, application stacks, and deployment environments

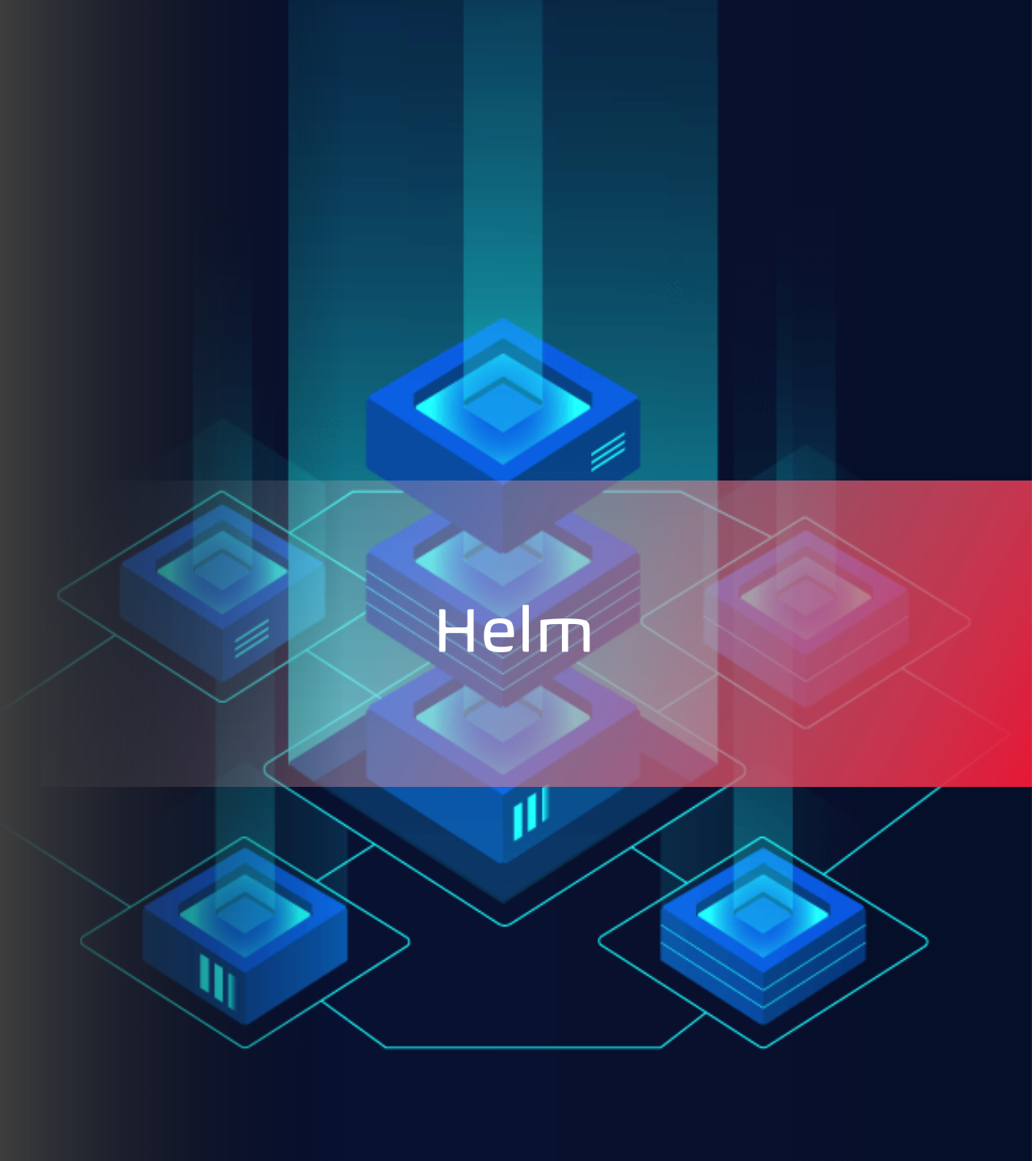
Jib is used to build our containers

• Jib builds optimized Docker and OCI images for your Java applications without a Docker daemon - and without deep mastery of Docker best-practices. It is available as plugins for Maven and Gradle and as a Java library.

- **Fast**
- **Reproducible**
- **Daemonless**

Demo docker

Tech Mahindra



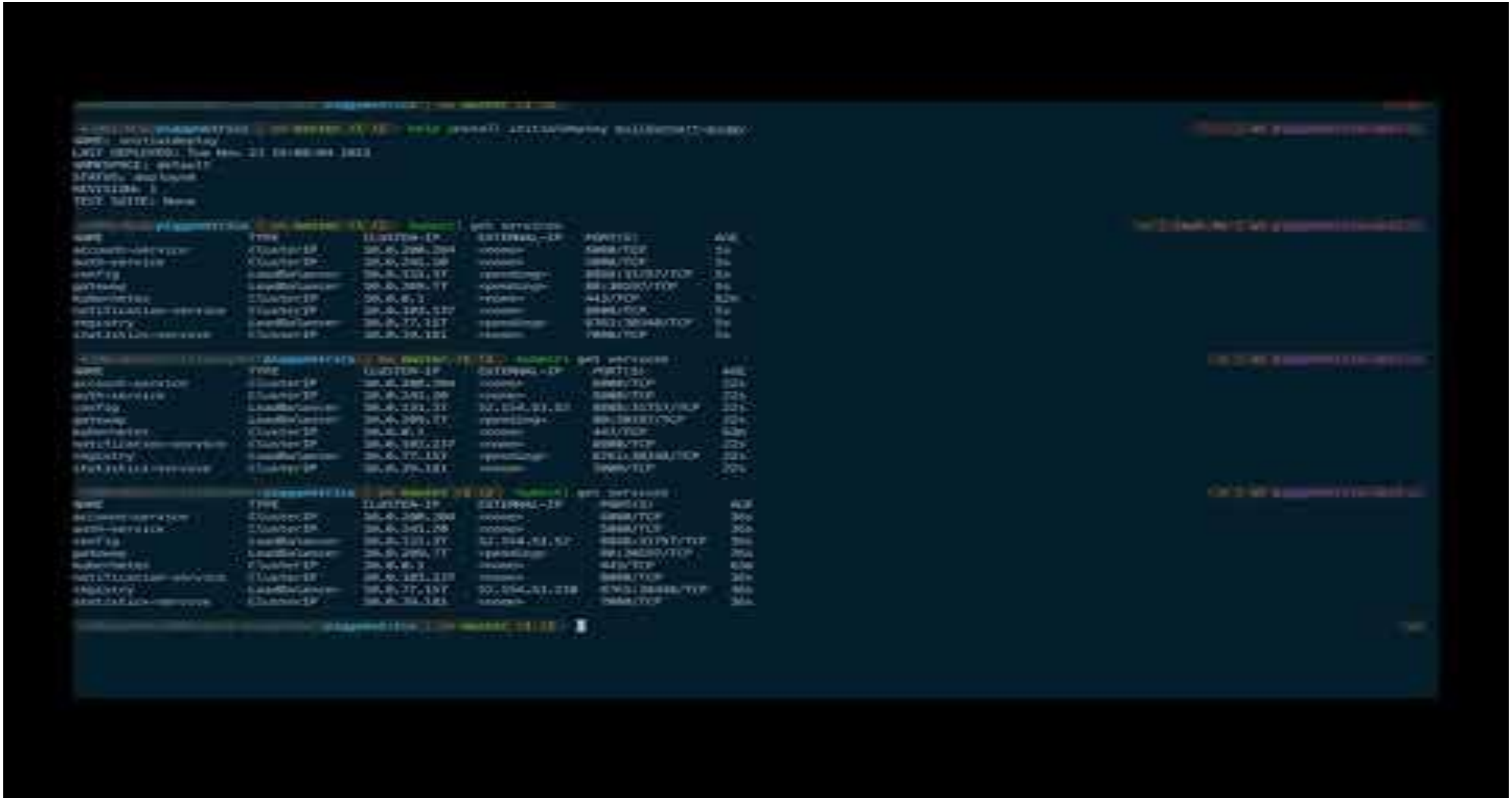
Tech Mahindra

We use Helm to quickly deploy the changes to the application:

- Helm helps combine multiple Kubernetes manifests (yaml) into a single unit called Helm Charts
- Helm gives the ability to customize application configurations during deployment

Demo helm

Tech Mahindra

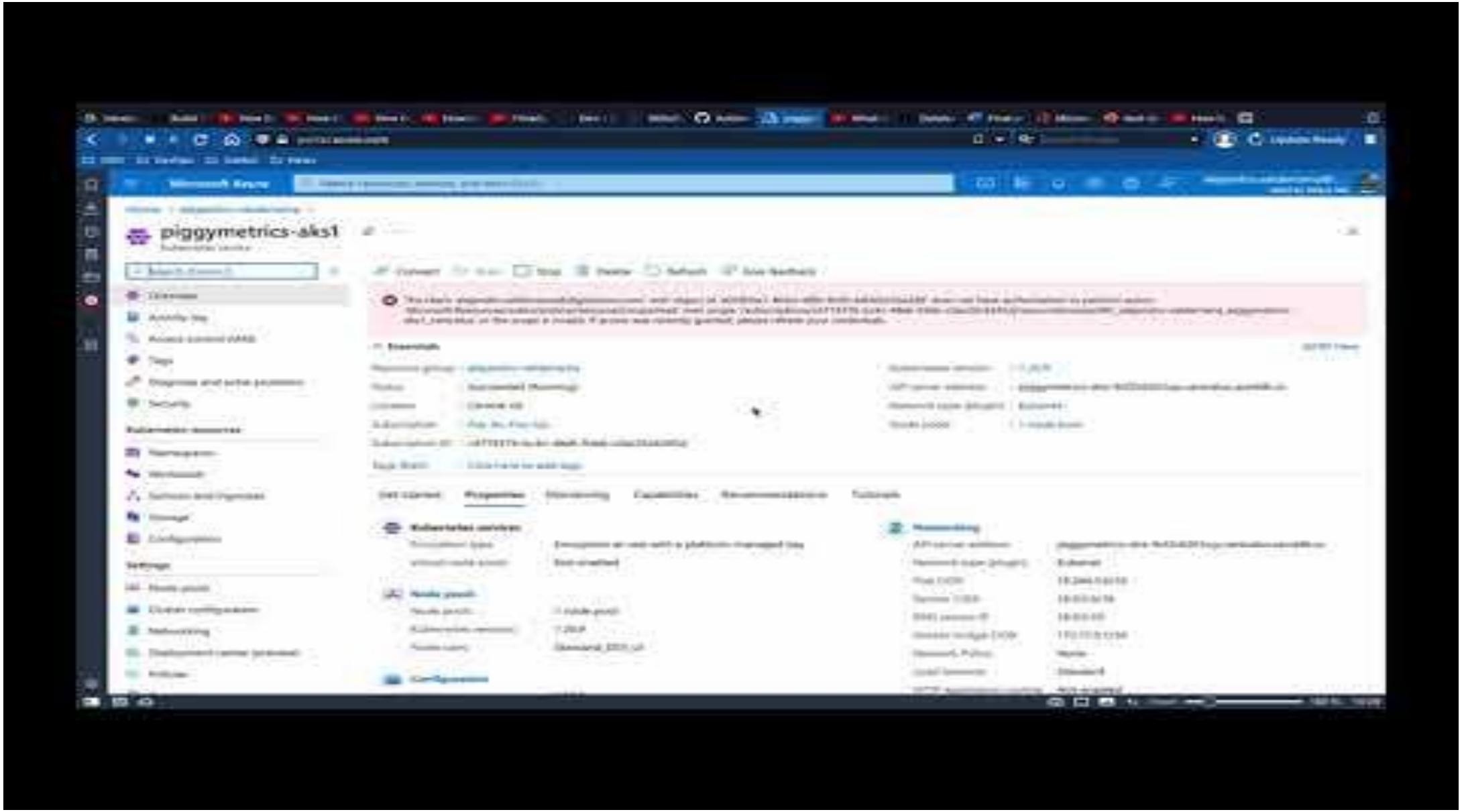


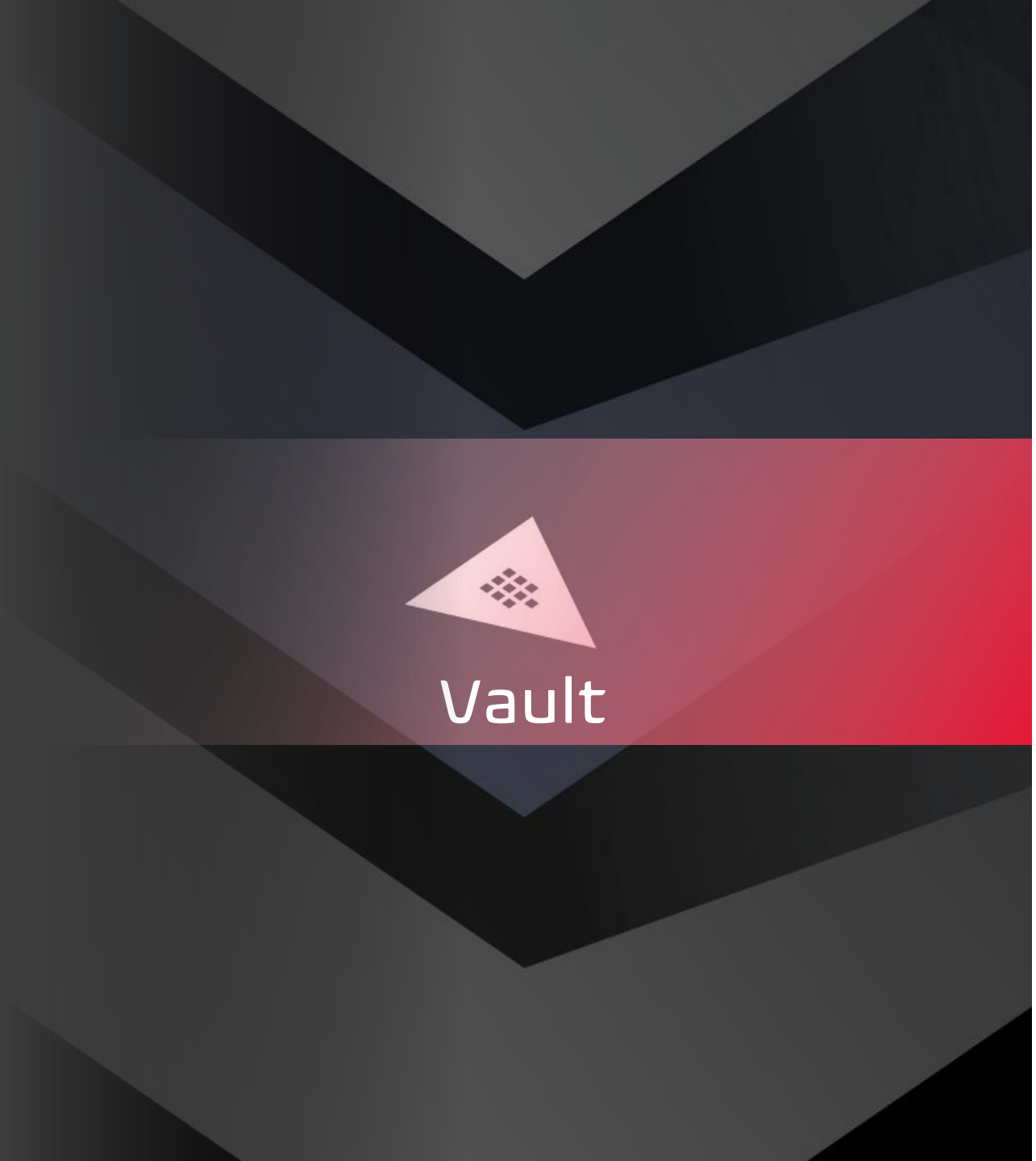
Azure Kubernetes Service (AKS)

- Azure Kubernetes Service is a managed container orchestration service based on the open-source Kubernetes system
- An organization can use AKS to handle critical functionality such as deploying, scaling and managing Docker containers and container-based applications

Demo AKS

Tech Mahindra



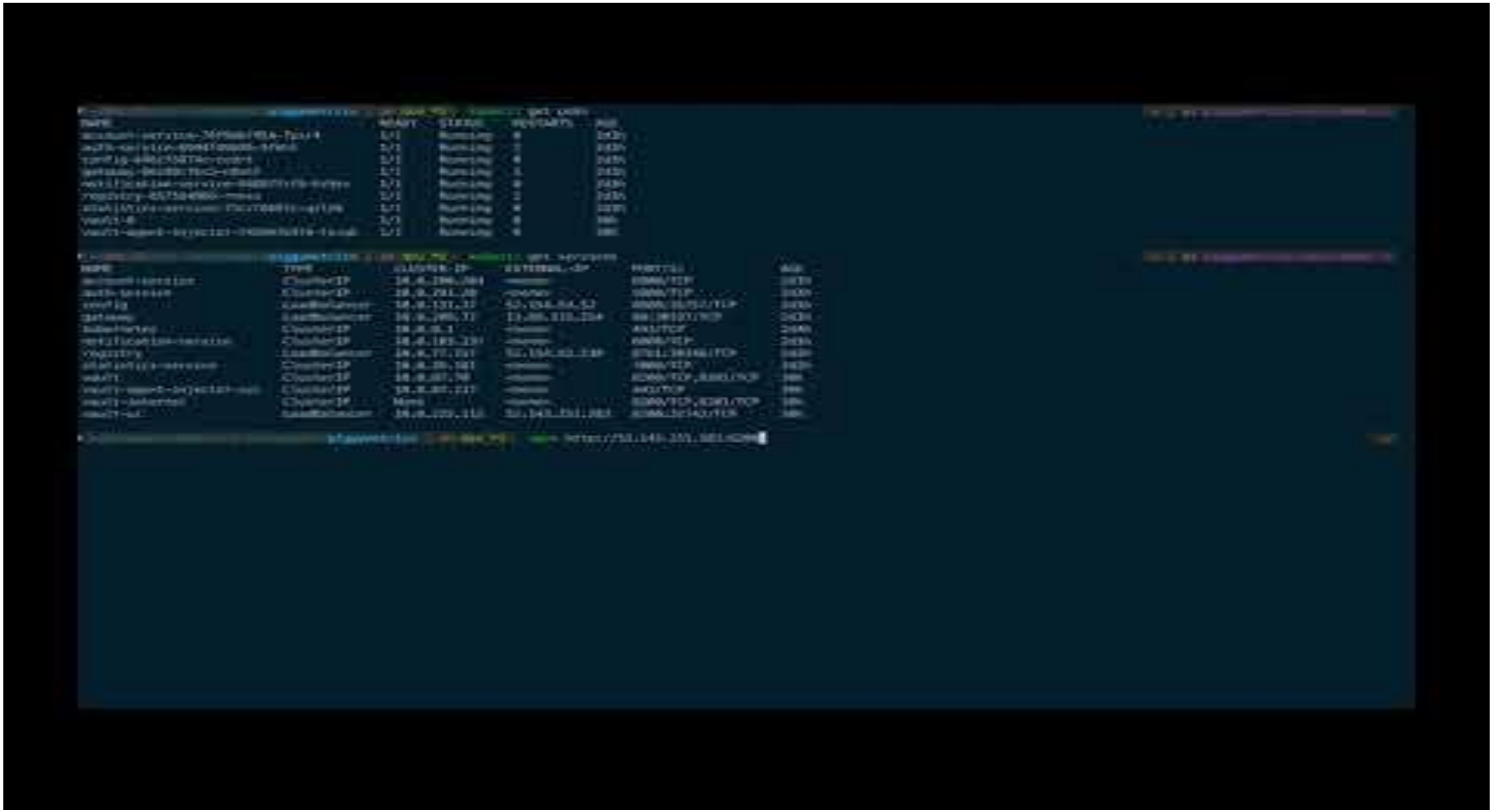


Tech Mahindra

- Vault is a secrets management tool specifically designed to control access to sensitive credentials
- It can be used to store sensitive values and at the same time dynamically generate access for specific services/applications on lease

Demo vault

Tech Mahindra



CI/CD pipeline using github actions

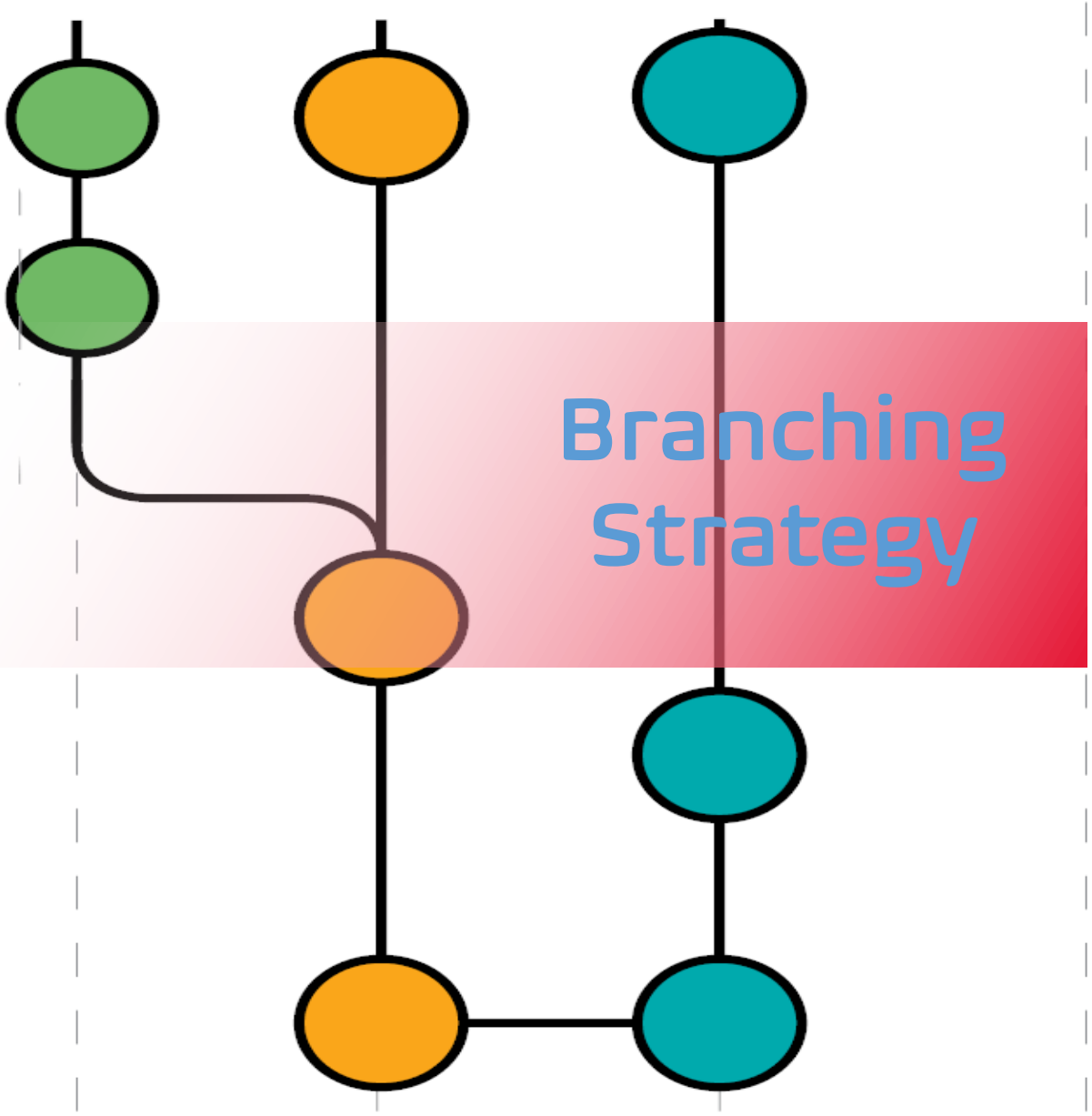
Tech Mahindra

- CI/CD
- Branching strategy
- Github actions

CI/CD

Tech Mahindra

- CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications.
- Use automation to reduce the time, effort, and risk involved in shipping a release.
- We use various pipelines that get trigger with either push or pull request in some branches.

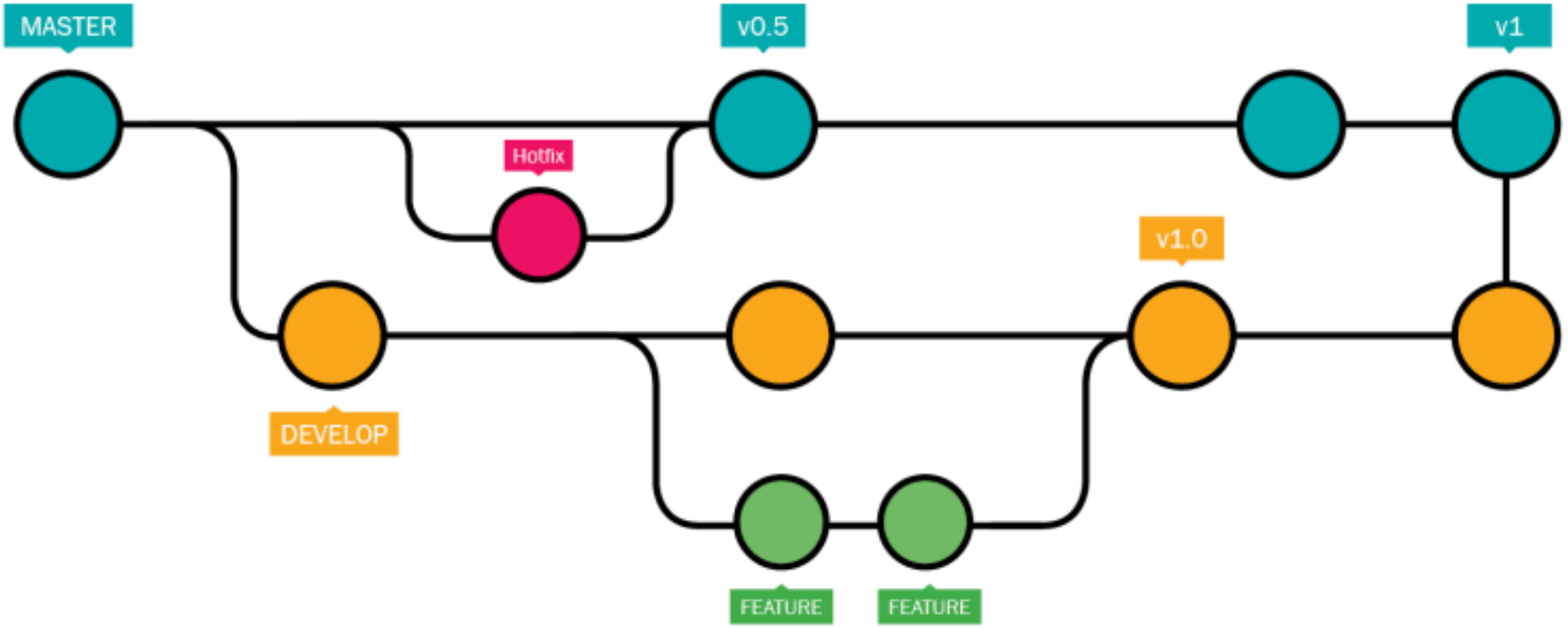


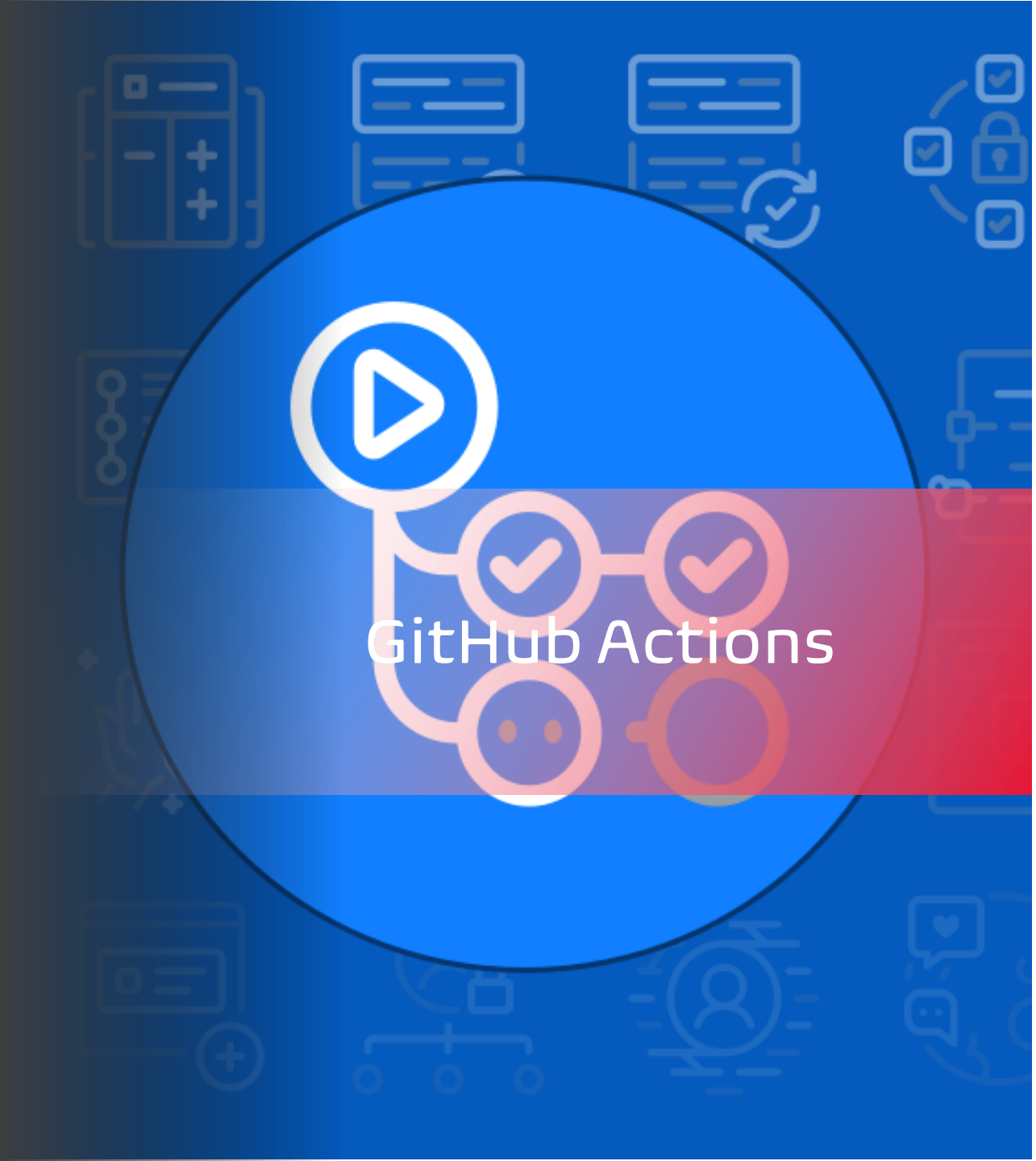
We propose a branching strategy that:

- Uses to main branches dev and master
- The master branch is always deployable.
- Push to other the dev occurs constantly

Tech Mahindra

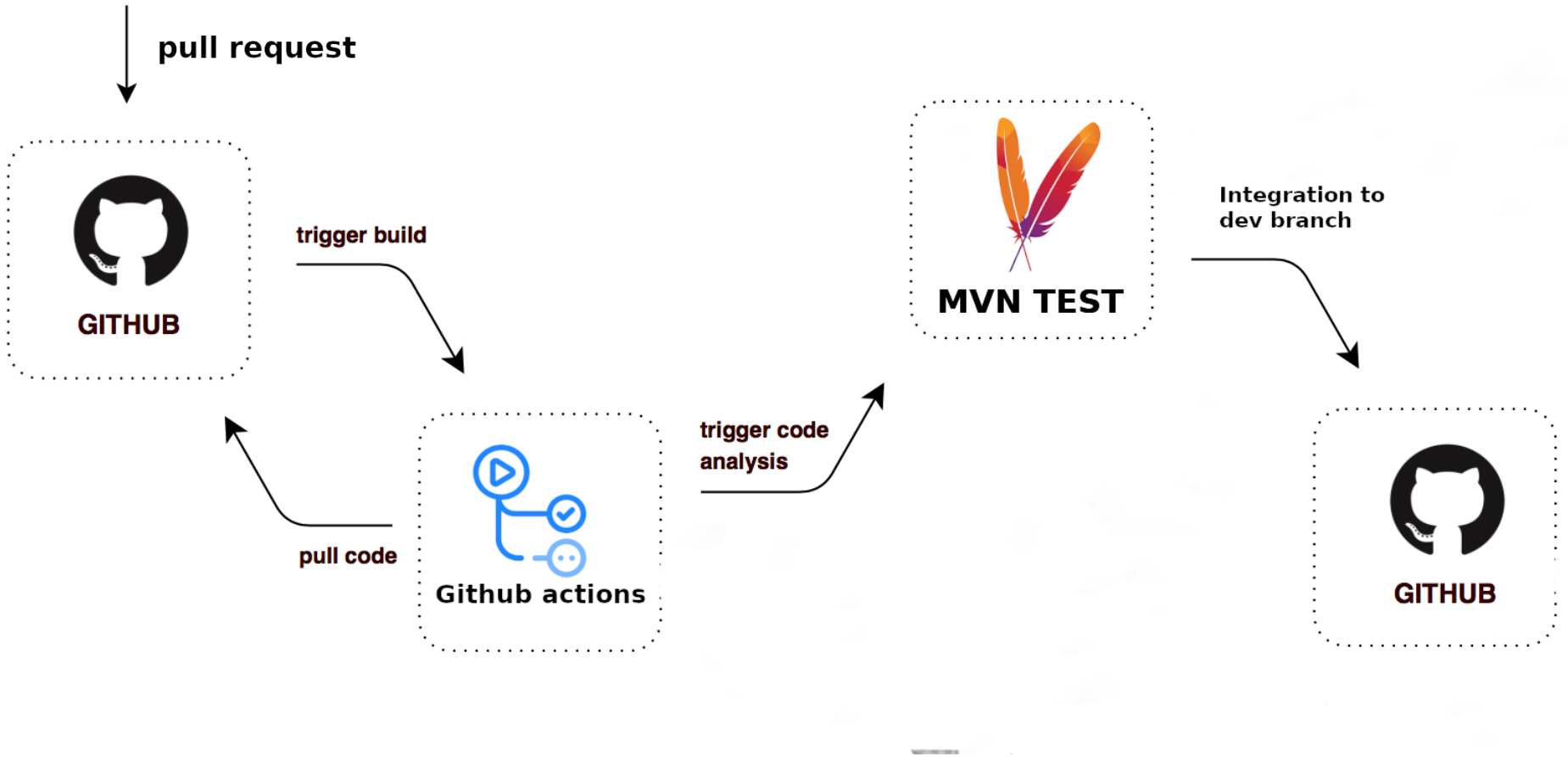
Branching strategy for Piggy Metrics.



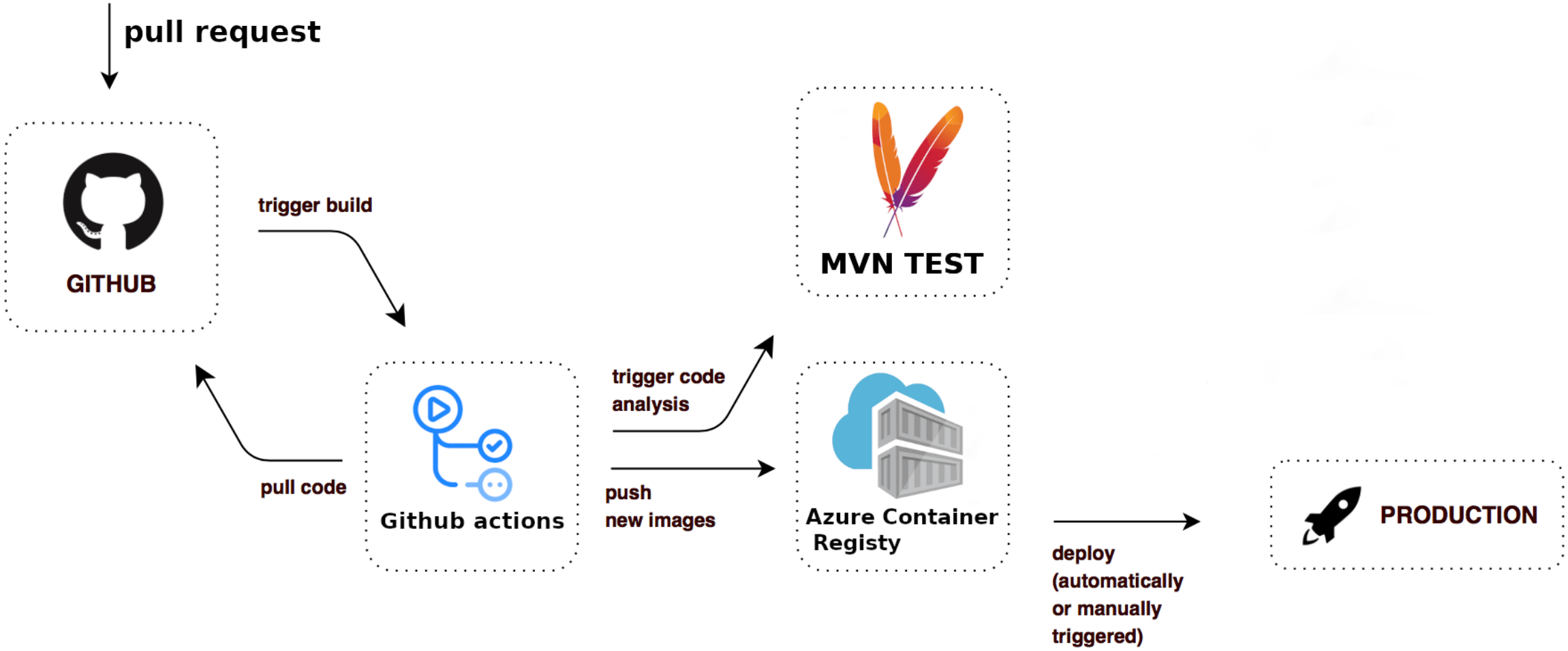


- GitHub Actions allow to create workflows that automatically build, test, publish, release, and deploy code:
- These workflows are made out of different tasks so-called actions that can be run automatically on certain events

From a new feature to dev



Pull request to master

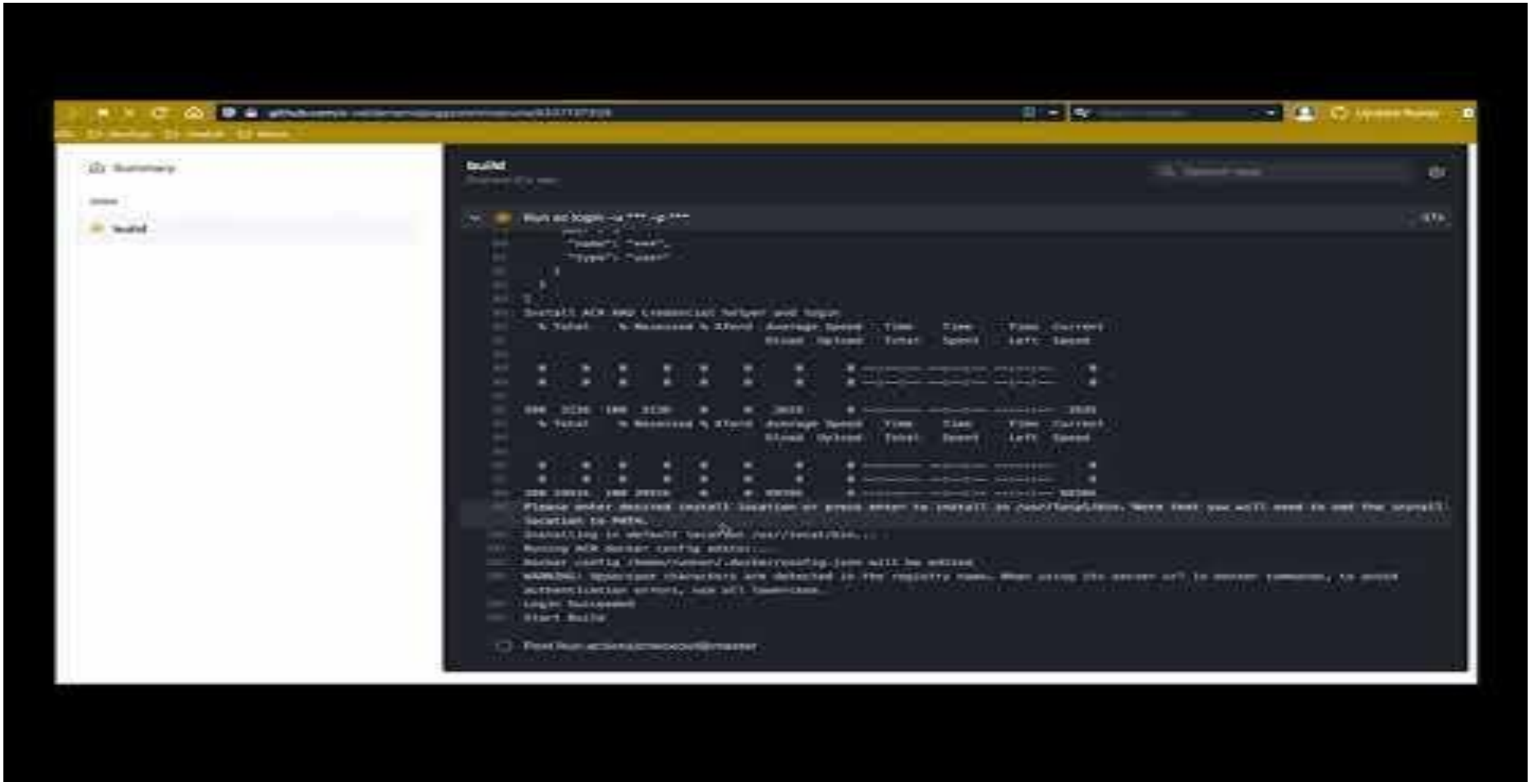


Demo feature-dev

Tech Mahindra

Demo dev-prod

Tech Mahindra



Tech Mahindra

What's next for Piggy Metrics

- Create a testing environment that deploys the changes added to the dev branch each time a pull request is accepted.
- Implement policies for enabling merging into dev and master branches.
- Add monitoring for the pods in AKS.
- Integrate vault secrete engine to interact with the CosmosDb database.

Q&A

Tech Mahindra

Thank you

Visit us at www.techmahindra.com

Disclaimer

Tech Mahindra Limited, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for information purposes and private circulation only and do not constitute an offer to buy or sell any services mentioned therein. They do not purport to be a complete description of the market conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.