DevSecOps12 Presents:

HELM AND MONITORING IN DEVOPS ENVIRONMENTS

Presented by Eli Levy, Ofek Harpaz, Daniel Shahnovich, and Lior Taub

Bar-Ilan University
אוניברסיטת בר־אילן

HELM AND MONITORING: DEV-OPS MUST HAVE

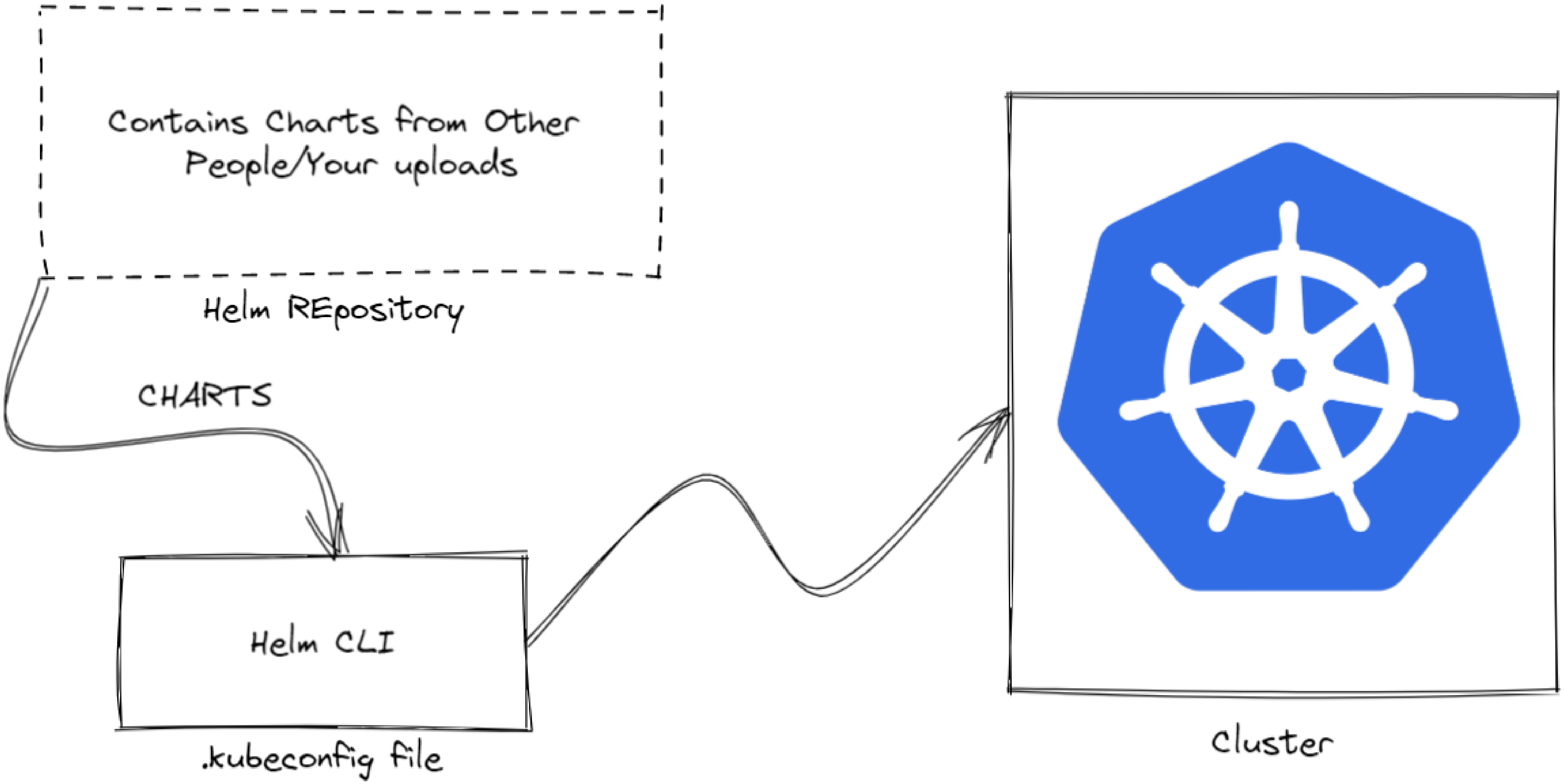
- An exploration of two critical pillars of DevOps: Helm and Monitoring.
- Discover how these tools supercharge your DevOps workflows.

HELM: THE KUBERNETES PACKAGE MANAGER

Helm is a package manager for Kubernetes applications that includes templating and lifecycle management functionality. It is essentially a package manager for Kubernetes manifests (such as Deployments, ConfigMaps, Services, etc.) that are grouped into charts.

ARCHITECTURE OF HELM

Helm Repository is a central hub where you can find charts (packages) created by either yourself or others. These charts serve as blueprints to achieve your desired application state. The Helm CLI retrieves the package, extracts its contents, and transforms the charts into valid YAML files. Subsequently, these YAML files are sent to the Kubernetes API server, resulting in the creation of a release.



Contains Charts from Other People/Your uploads

Helm REpository

CHARTS

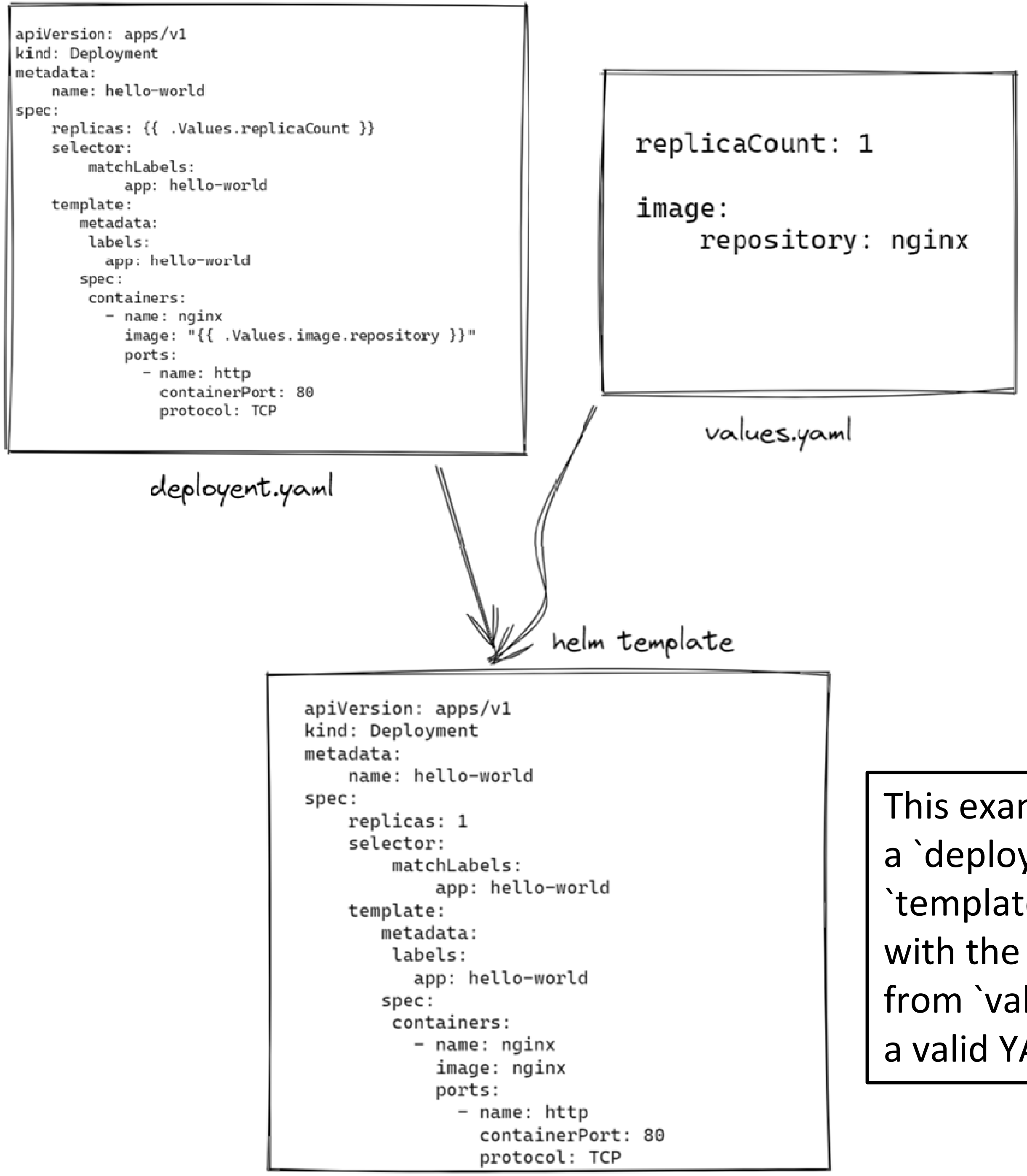
Helm CLI

.kubeconfig file

Cluster

HELM CHARTS COMPONENTS

- **`charts/`**: This directory can be used to store manually maintained chart dependencies.
  - **`templates/`:** These contain the template files which would be used to create the final manifest after combining with `values.yaml`.
  - **`Chart.yaml`:** This file contains information about the chart, such as the name and version of the chart, the maintainer, dependencies, a related website, and search terms.
  - **`values.yaml`:** This contains the default configuration for your charts. You can edit this for updating values and remove the complexity of finding specific editable items in the different manifests.



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
    name: hello-world
spec:
    replicas: {{ .Values.replicaCount }}
    selector:
        matchLabels:
            app: hello-world
    template:
       metadata:
        labels:
          app: hello-world
       spec:
        containers:
          - name: nginx
            image: "{{ .Values.image.repository }}"
            ports:
              - name: http
                containerPort: 80
                protocol: TCP
```

```yaml
replicaCount: 1

image:
    repository: nginx
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
    name: hello-world
spec:
    replicas: 1
    selector:
        matchLabels:
            app: hello-world
    template:
       metadata:
        labels:
          app: hello-world
       spec:
        containers:
          - name: nginx
            image: nginx
            ports:
              - name: http
                containerPort: 80
                protocol: TCP
```

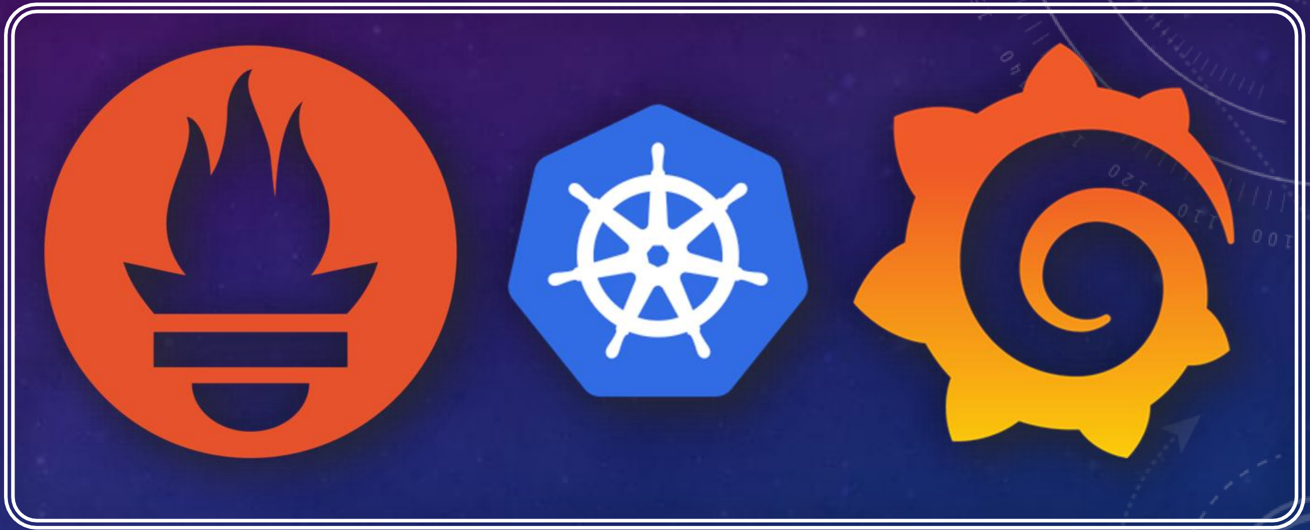
This example shows a `deployment.yaml` from `templates` being rendered with the custom values from `values.yaml` to produce a valid YAML.

BENEFITS OF HELM

- Offers a straightforward method for deploying complex applications.
- Simplifies the process of modifying specific values in your deployments.
- Provides a means to version individual packages.
- Enables the sharing of templates across organizations or on the internet.
- Streamlines the management of dependencies.
- Makes the process of reverting changes easy.

MONITORING: TRANSPARENCY AND VISIBILITY

- Monitoring ensures system health and performance.
- Real-time data helps detect issues proactively.
- Central to maintaining system reliability.



POPULAR MONITORING TOOLS

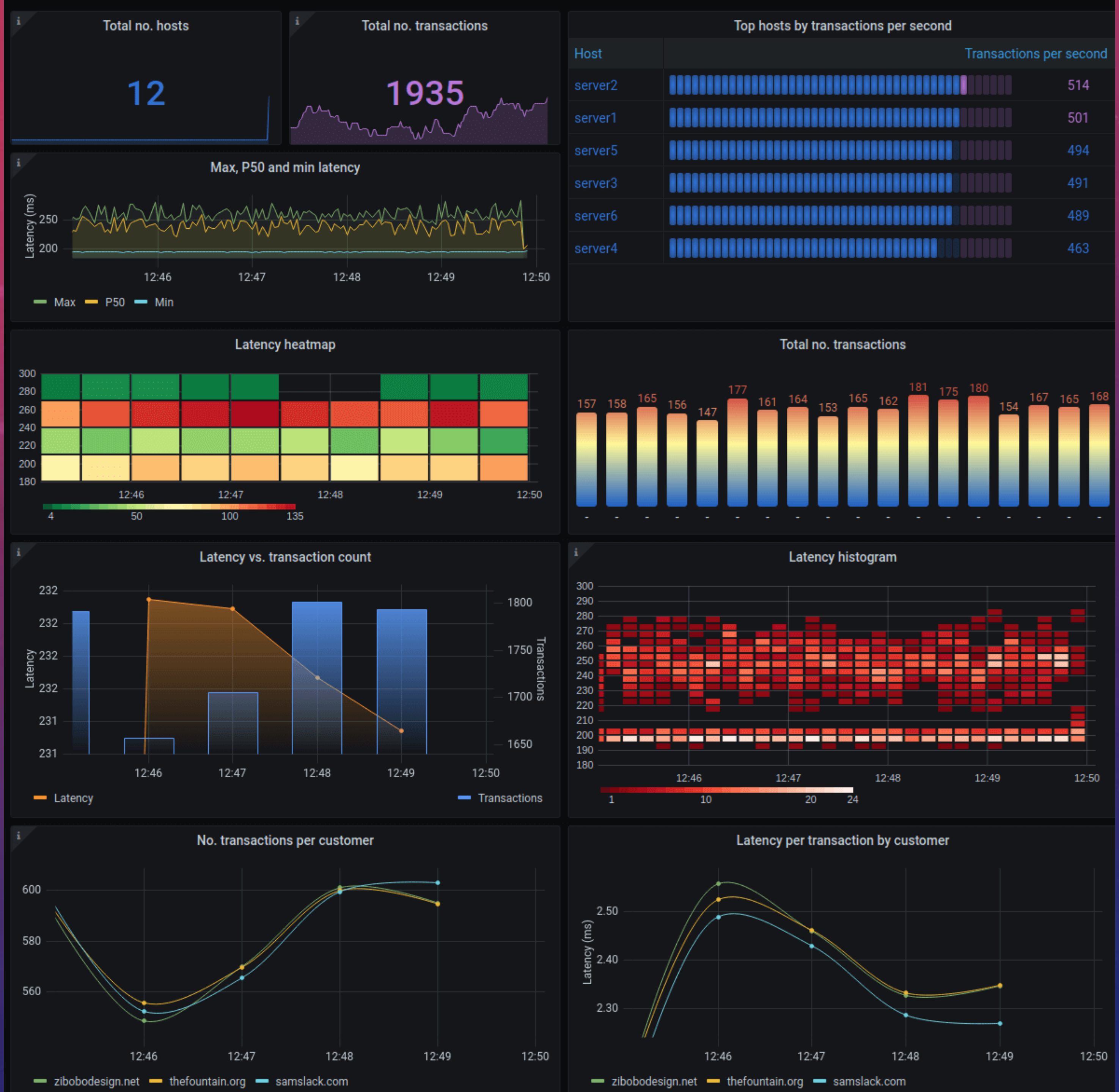
- Prometheus, Grafana, ELK Stack, etc.
- Collect, store, visualize, and analyze metrics and logs.
- Vital for understanding system behavior.

BENEFITS OF MONITORING

Early Issue Detection: Identify and resolve problems before they impact users.

Resource Optimization: Efficiently allocate resources, reducing costs.

Data-Driven Decisions: Base your actions on real-time metrics.



Top hosts by transactions per second

Host	Transactions per second
server2	514
server1	501
server5	494
server3	491
server6	489
server4	463

HELM AND MONITORING: BETTER TOGETHER

- Helm deploys, and Monitoring ensures performance.
- Together, they create a robust DevOps workflow.
- Efficiency: Streamline application deployment and monitoring.
- Visibility: Real-time insights into application performance.
- Cost Optimization: Identify resource waste and reduce expenses.
- Reliability: Proactive issue detection and zero-downtime deployments.

IN CONCLUSION

Helm simplifies Kubernetes application management.

Monitoring is essential for proactive issue detection.

Helm and Monitoring together ensure efficiency, visibility, and reliability.

Bar-Ilan University
אוניברסיטת בר־אילן

Q&A

THANK YOU!

Bar-Ilan University
אוניברסיטת בר־אילן