

## 介绍

Kubernetes 集群已成为现代云原生应用程序的首选容器编排工具。然而，随着集群规模的增长，与监控其运行状况和性能相关的挑战也随之增加。为了确保成功部署，必须能够实时监控集群状态、检测问题并采取纠正措施。如果没有足够的监控工具，集群管理员无法有效地管理其集群，从而导致中断、应用程序停机和其他严重问题。

如果您部署并管理过 Kubernetes 集群，您之前可能见过类似的内容：

```
$
$ kubectl get pods -n kube-system -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
azure-ip-masq-agent-4zhgb	1/1	Running	0	33m	10.240.1.236	aks-systempool-20474252-vmss000000
azure-ip-masq-agent-5zqqg	1/1	Running	0	33m	10.240.2.11	aks-systempool-20474252-vmss000001
azure-ip-masq-agent-6vfjd	1/1	Running	0	108m	10.240.1.174	aks-appsnodepool-20474252-vmss000003
azure-ip-masq-agent-8n6zj	1/1	Running	0	108m	10.240.1.81	aks-appsnodepool-20474252-vmss000000
azure-ip-masq-agent-b6cr2	1/1	Running	0	108m	10.240.1.112	aks-appsnodepool-20474252-vmss000001
azure-ip-masq-agent-cqg6z	1/1	Running	0	108m	10.240.1.205	aks-appsnodepool-20474252-vmss000004
azure-ip-masq-agent-gj2vw	1/1	Running	0	33m	10.240.2.42	aks-systempool-20474252-vmss000002
azure-ip-masq-agent-spqhf	1/1	Running	0	108m	10.240.1.143	aks-appsnodepool-20474252-vmss000002
azure-policy-5b6c599bc5-dfl59	1/1	Running	0	3m10s	10.240.2.43	aks-systempool-20474252-vmss000002
azure-policy-webhook-5c984bdf5-wlkxs	1/1	Running	0	3m10s	10.240.2.54	aks-systempool-20474252-vmss000002
coredns-9d6c6c99b-7cxcf	1/1	Running	0	3m4s	10.240.2.61	aks-systempool-20474252-vmss000002
coredns-9d6c6c99b-hq295	1/1	Running	0	3m10s	10.240.1.253	aks-systempool-20474252-vmss000000
coredns-9d6c6c99b-nqqbq	1/1	Running	0	3m10s	10.240.2.33	aks-systempool-20474252-vmss000001
coredns-autoscaler-599949fd86-7q9fb	1/1	Running	0	3m9s	10.240.2.53	aks-systempool-20474252-vmss000002
konnectivity-agent-fc94bf6c5-f9qsw	1/1	Running	0	3m10s	10.240.2.11	aks-systempool-20474252-vmss000001
konnectivity-agent-fc94bf6c5-kv5gf	1/1	Running	0	3m	10.240.1.236	aks-systempool-20474252-vmss000000
kube-proxy-7x2nc	1/1	Running	0	108m	10.240.1.205	aks-appsnodepool-20474252-vmss000004
kube-proxy-8fcrc	1/1	Running	0	33m	10.240.2.42	aks-systempool-20474252-vmss000002
kube-proxy-f7j5k	1/1	Running	0	33m	10.240.1.236	aks-systempool-20474252-vmss000000
kube-proxy-jwjsps	1/1	Running	0	108m	10.240.1.143	aks-appsnodepool-20474252-vmss000002
kube-proxy-lsqjg	1/1	Running	0	108m	10.240.1.81	aks-appsnodepool-20474252-vmss000000
kube-proxy-mfk75	1/1	Running	0	108m	10.240.1.112	aks-appsnodepool-20474252-vmss000001
kube-proxy-sl4vj	1/1	Running	0	33m	10.240.2.11	aks-systempool-20474252-vmss000001
kube-proxy-t8rlv	1/1	Running	0	108m	10.240.1.174	aks-appsnodepool-20474252-vmss000003
metrics-server-77c8679d7d-7pgpb	1/1	Running	0	3m10s	10.240.1.237	aks-systempool-20474252-vmss000000
omsagent-2sxjj	1/1	Running	0	108m	10.240.1.162	aks-appsnodepool-20474252-vmss000002
omsagent-5p44m	1/1	Running	0	33m	10.240.2.21	aks-systempool-20474252-vmss000001
omsagent-jchx6	1/1	Running	0	33m	10.240.1.247	aks-systempool-20474252-vmss000000
omsagent-jsjdd	1/1	Running	0	108m	10.240.1.127	aks-appsnodepool-20474252-vmss000001
omsagent-ln6jb	1/1	Running	0	33m	10.240.2.60	aks-systempool-20474252-vmss000002
omsagent-rdwpj	1/1	Running	0	108m	10.240.1.98	aks-appsnodepool-20474252-vmss000000
omsagent-rs-5d7dd75669-ndjrt	1/1	Running	0	3m10s	10.240.2.12	aks-systempool-20474252-vmss000000
omsagent-t77rf	1/1	Running	0	108m	10.240.1.200	aks-appsnodepool-20474252-vmss000003
omsagent-zwc2t	1/1	Running	0	108m	10.240.1.208	aks-appsnodepool-20474252-vmss000004

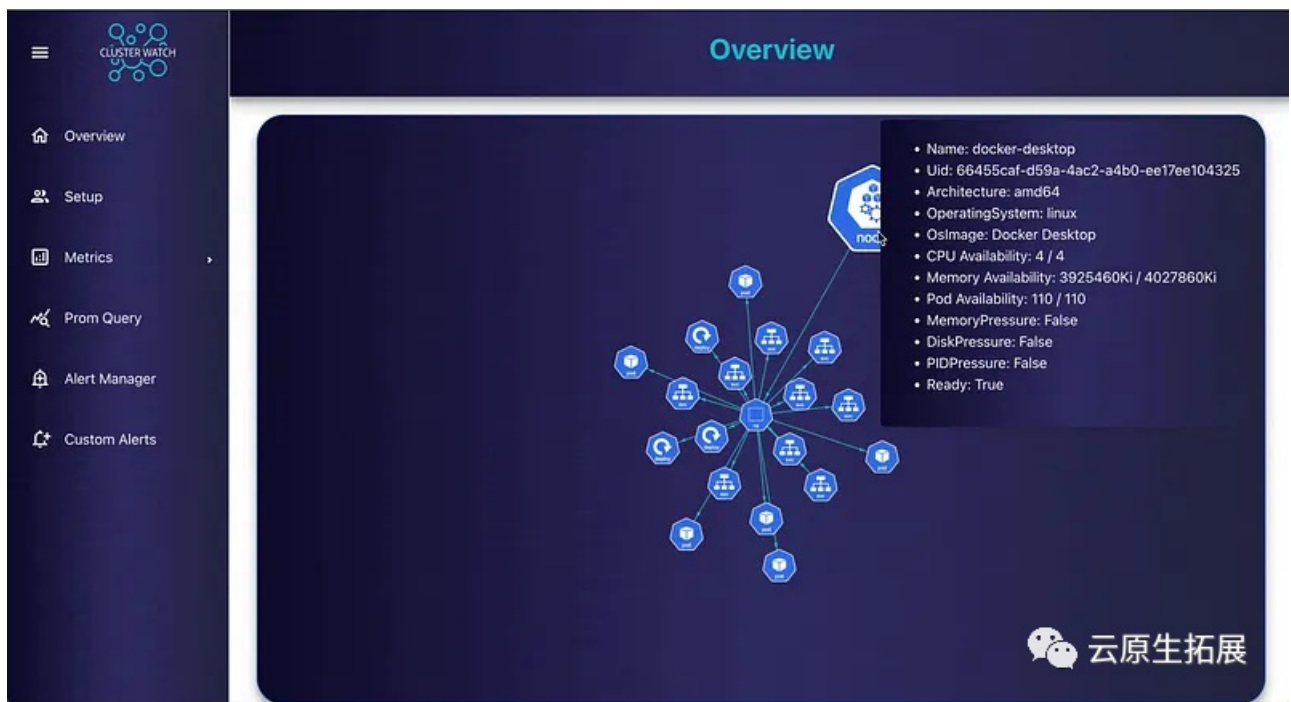
```
$
```

毫无疑问，这是实时捕获基本集群指标的最快、最简单的方法。但在容器编排可能成为整个职业生涯焦点的世界中，这些基本统计数据永远不足以满足对优化的不懈追求。

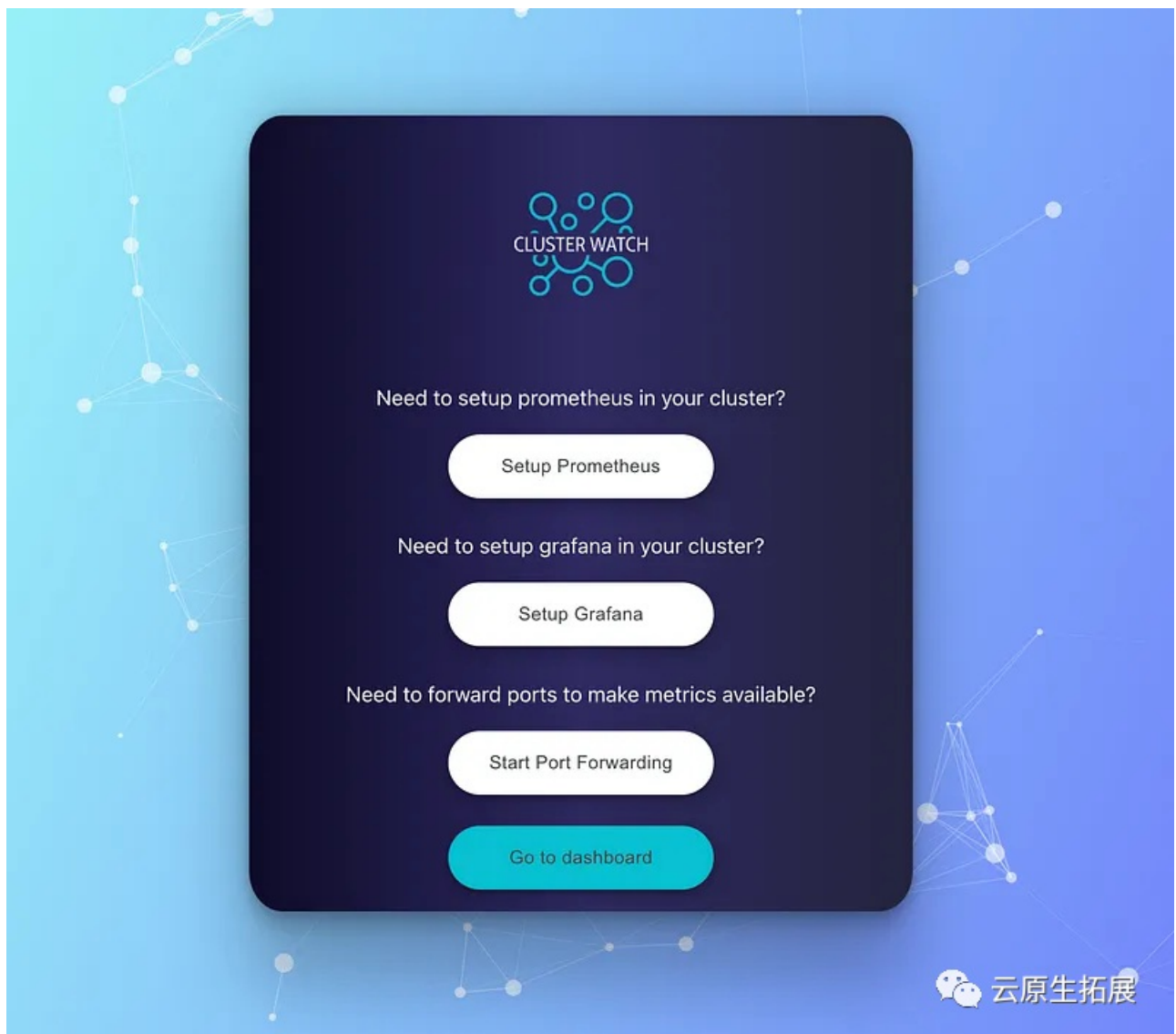
随着容器的增长，以及 Pod 和服务变得更加集群化，集群内这些单独的可部署单元的相互关系可能会很快变得非常复杂。当涉及到对应用程序进行故障排除、调试、优化和扩展时，这将不可避免地造成巨大的麻烦。为了解决这个问题，ClusterWatch 应运而生。

## 解决方案

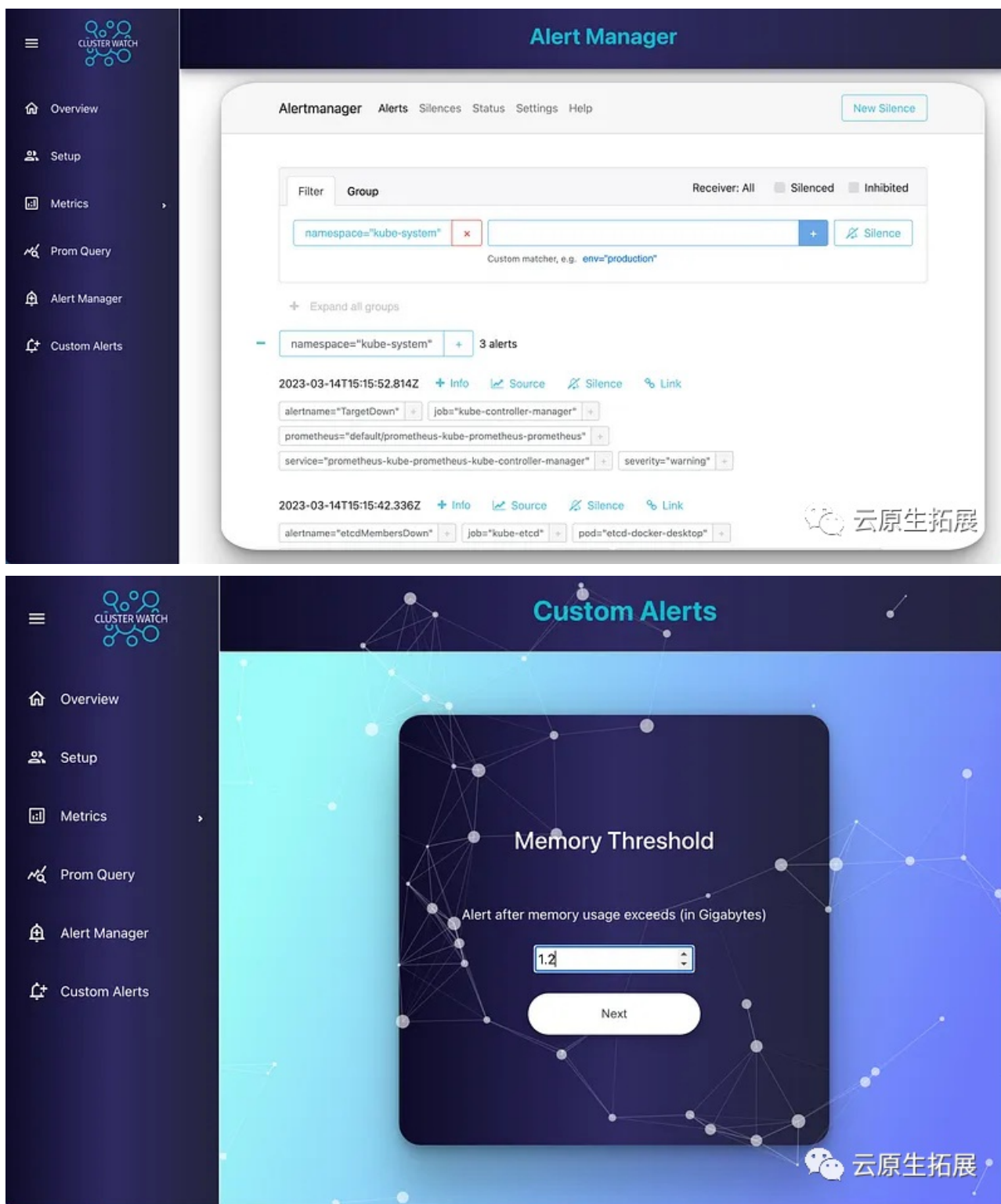
ClusterWatch 是一款开源产品，可为与 Kubernetes 集群监控相关的挑战提供全面的解决方案。通过利用 Kubernetes 编排实现最佳可扩展性和效率，ClusterWatch 能够提供对集群各个架构层的实时监控。这包括有关集群的节点、pod、服务和其其他关键组件的信息。有了这些信息，管理员可以快速检测并响应任何问题，例如节点或 Pod 故障，避免造成重大问题。



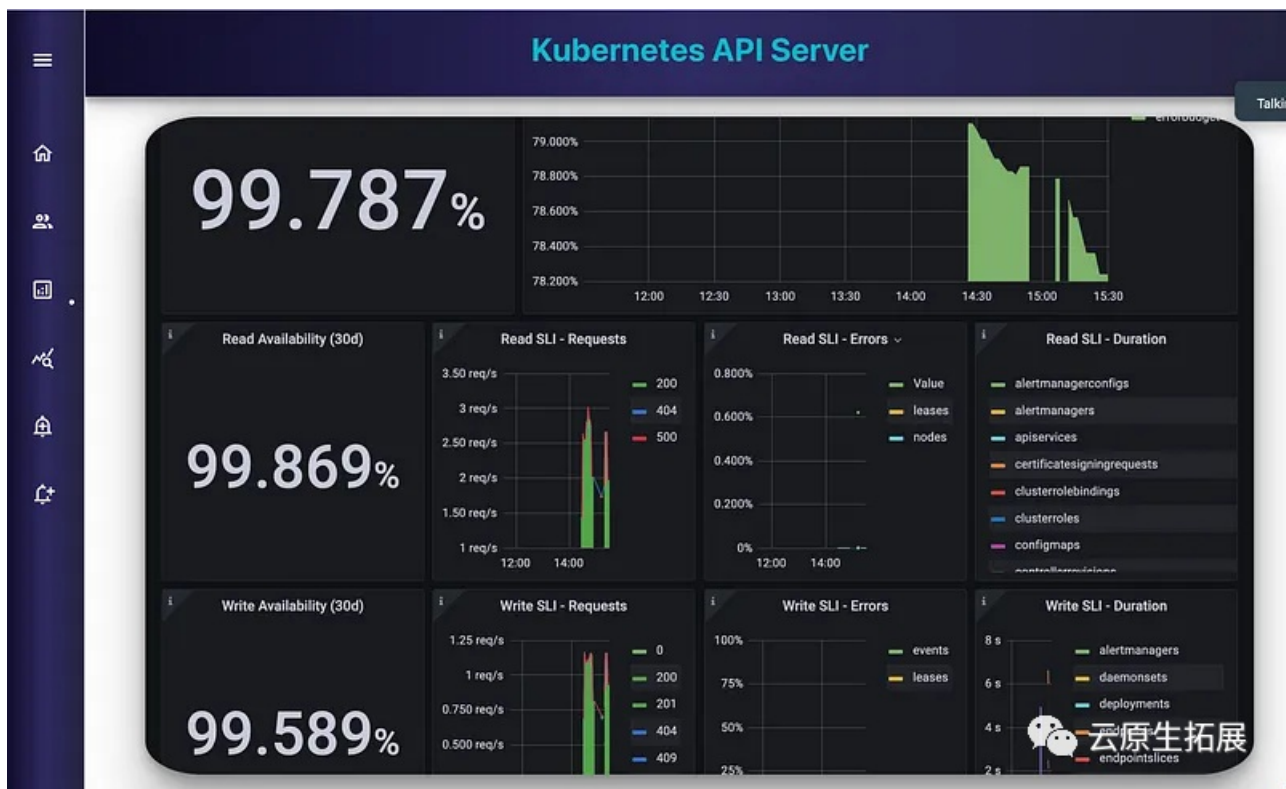
除了全面的监控功能外，ClusterWatch还为Prometheus和Grafana提供自动化配置流程，将它们合并到Kubernetes集群中。只需点击几下，用户就可以设置 Prometheus 和 Grafana 实例，并将它们无缝集成到 Kubernetes 集群中，从而轻松监控和优化应用程序的性能。



用户还可以为他们感兴趣跟踪的任何指标设置自定义警报，并且可以在警报管理器页面视图上查看这些警报。这使管理员可以轻松掌握可能出现的任何问题并实时采取纠正措施。



基于这些强大的功能，ClusterWatch 为用户提供了单一视图，以访问他们所需的有关集群运行状况和性能的所有信息。交互式响应式仪表板提供了集群的全面概述，包括节点、Pod、服务和其他关键组件的实时指标。



总而言之，ClusterWatch是一个功能强大的开源工具，提供对Kubernetes集群的实时监控和管理。凭借其自动化配置流程、全面的监控功能和用户友好的仪表板，ClusterWatch 是希望确保基于 Kubernetes 的应用程序的运行状况和性能的组织理想解决方案。

## ClusterWatch 入门

### 第 1 步：克隆存储库：

```
git clone https://github.com/oslabs-beta/ClusterWatch.git
```

### 第 2 步：安装依赖：

```
npm install
```

### 第 3 步：启动 k8s 集群：

- 请务必在应用程序启动之前启动并运行 Kubernetes 集群。

### 第 4 步：启动应用：

```
npm run dev
```

### 第 5 步：全局安装 redis



我们使用Redis作为缓存中间件来提高数据检索速度，从而带来更好的实时监控体验。

```
brew install redis
```

## 第 6 步：导航至 <http://localhost:8080>

您现在应该被重定向到网络应用程序的设置页面。如果这是您第一次配置集群，请执行以下设置步骤：

- 如果您没有 prometheus-kube-stack helm chart（其中包括 kube-state-metrics、node-exporter 和 grafana），请使用“Setup Prometheus”按钮下载、更新和安装存储库。注意：此步骤可能需要一些时间，不用担心。
- 要在集群中部署 Grafana，请使用“设置 Grafana”按钮来应用我们的自定义 Grafana 配置。
- 如果您以前使用过我们的应用程序，请按“开始端口转发”按钮并进行监控！
- 请记住，该应用程序使用本地端口，这些端口不得在您的计算机上被占用：

— 3000: API服务器

— 8080: 客户端服务器

— 3001: Grafana

— 9090: Prometheus

— 9093: Alert Manager

— 6379: Redis