Architecture for the Docker-Ethereum application

For this architecture requires persistent storage, and it is better to use a stateful set because it allows each Pod to have a unique, stable network identity and stable storage

ReplicaSet for Stateless Front-End**:** It is possible to assert that the front end does not necessitate persistent storage, which is why ReplicaSet is an advantageous choice.

Now, let us discuss storage. It is essential to ensure that the data persists even if the PO is eliminated. This is because the persistent volume will enable storage to be provisioned in the cluster with unique identifiers. Persistent Volume Claims enable modules to request and utilize persistent storage resources without requiring knowledge of the underlying storage implementation.

For the scaling, the horizontal Pod Autoscaler will scale the number of pod replicas by utilizing the information obtained from the CPU utilization this is the best option to control the varying loads

The application will be accessible from a single point of access using LoadBalncer, which will distribute incoming requests across the available Pods to balance the load and ensure maximum availability.

Talking about secrets they will When it comes to secrets, they will enable the secure storage and retrieval of sensitive information by Pods, when necessary, without the need to hardcode it into the application code. Kubernetes Secrets will be used to manage credentials and other sensitive information for the docker-Ethereumethereum application.

RBAC restricts access to resources based on the roles of individual users. Can regulate resource access precisely by defining roles and assigning them to users or service accounts, guaranteeing that users have the bare minimum of permissions.

Pods

	Nombre	Imágenes	Etiquetas
●	frontend-cbs2j	my-frontend-image	app: frontend
●	frontend-tprp6	my-frontend-image	app: frontend
●	frontend-zj789	my-frontend-image	app: frontend

Cluster Role Bindings

Nombre	Fecha de creación ↑
kubernetes-dashboard	3 hours ago
storage-provisioner	3 hours ago
minikube-rbac	3 hours ago
kubeadm:node-proxier	3 hours ago
system:coredns	3 hours ago
system:controller:job-controller	3 hours ago
kubeadm:node-autoapprove-bootstrap	3 hours ago
system:controller:node-controller	3 hours ago
kubeadm:kubelet-bootstrap	3 hours ago
system:controller:persistent-volume-binder	3 hours ago

Cluster Role

Cluster Roles

Nombre	Fecha de creación
kubernetes-dashboard	3 hours ago
system:coredns	3 hours ago
system:controller:namespace-controller	3 hours ago
system:controller:cronjob-controller	3 hours ago
system:kube-scheduler	3 hours ago
system:controller:validatingadmissionpolicy-status-controller	3 hours ago
system:controller:ttl-controller	3 hours ago
system:controller:persistent-volume-binder	3 hours ago
system:controller:ttl-after-finished-controller	3 hours ago
system:controller:statefulset-controller	3 hours ago

1 – 10 of 67

Services

Servicios

	Nombre	Etiquetas	Tipo	IP cluster	Endpoints Internos	Endpoints Externos	Fecha de creación
	ethereum-service	-	LoadBalancer	10.108.77.13	ethereum-service:30303 TCP ethereum-service:30954 TCP	-	6 hours ago
	kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	6 hours ago
	kubernetes-dashboard	addonmanager.kubernetes.io/mode: Reconcile kubernetes.io/metadata.name: kubernetes-dashboard kubernetes.io/minikube-addons: dashboard	Active				6 hours ago
	default	kubernetes.io/metadata.name: default	Active				6 hours ago
	kube-node-lease	kubernetes.io/metadata.name: kube-node-lease	Active				6 hours ago
	kube-public	kubernetes.io/metadata.name: kube-public	Active				6 hours ago
	kube-system	kubernetes.io/metadata.name: kube-system	Active				6 hours ago