Gerenciando ambientes PostgreSQL com o Operador CloudNativePG

Sobre mim

Gustavo Lemos

Administrador de Banco de Dados

CARGO

Administrador de banco de dados

ÁREAS DE INTERESSE

SGBDs, SOs, Monitoração, Automação

CONTATO

Linkedin: Gustavo Lemos

BLOG

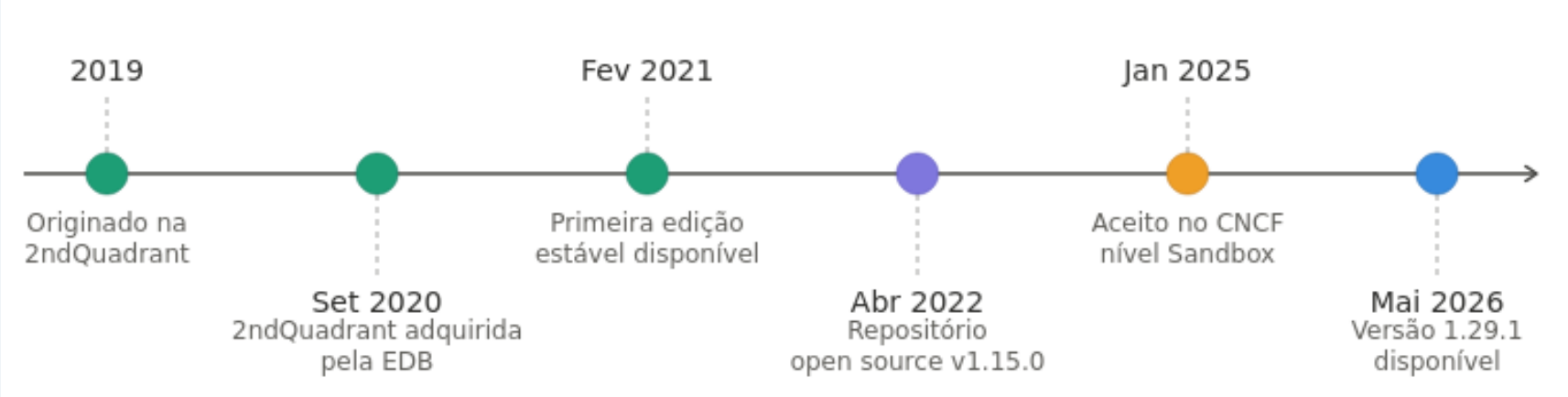
Blog: lemosdba.com.br

Agenda

01	**Introdução ao CloudNativePG** *Descrição, Histórico*	
02	**Contextualizando o CloudNativePG** *Arquitetura, operação, api, recursos e controladores customizados*	
03	**Princípios fundamentais do CloudNativePG** *Operadores, conceitos DevOps, estado de funcionamento*	
04	**Funcionalidades do CloudNativePG** *Infraestrutura e arquitetura, Provisionamento, Replicação, Backup & Recovery, Segurança, Observabilidade, Performance e responsabilidade , Upgrades*	
05	**Suporte e requisitos do CloudNativePG** *, Versões: Kubernetes e PostgreSQL container images, Barman Cloud Plugin*	
06	**Demonstração - Provisionamento e gerenciamento** *Requisitos, Provisionamento de cluster e gerenciamento*	

Introdução ao CloudNativePG

- O operador CNPG (CloudNativePG) é responsável pelo ciclo de vida de um cluster PostgreSQL.
- Utilizado para implementar e gerenciar Bancos de dados PostgreSQL em ambiente Kubernetes.
- Criado com base em conceitos DevOps + abordagem "Cloud Native" de software.



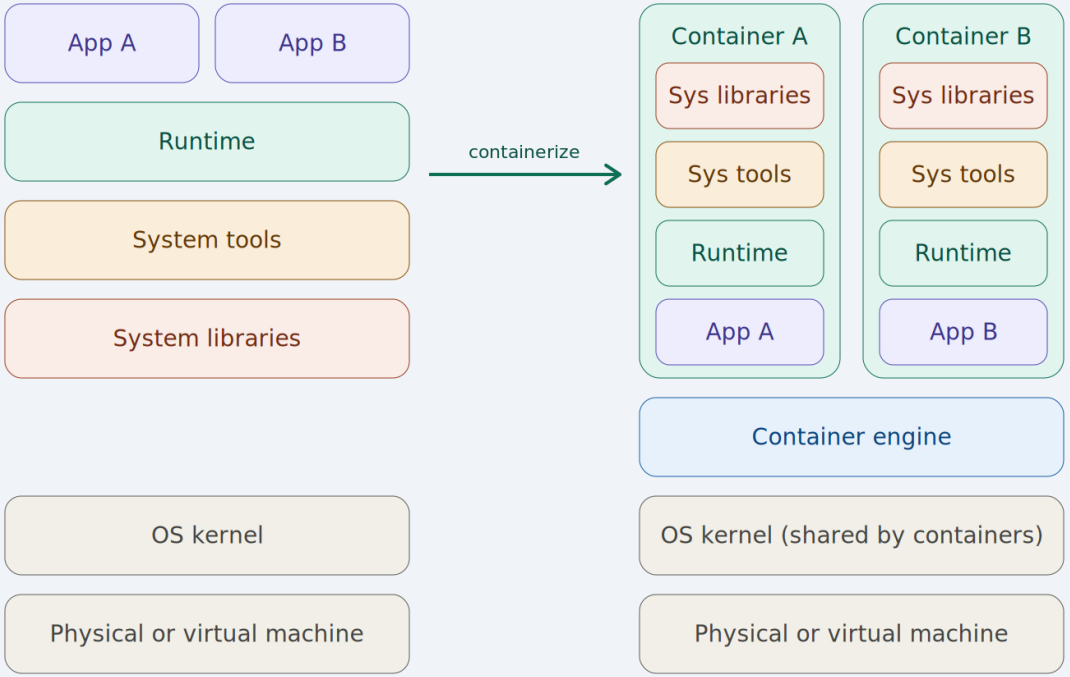
Contextualizando o CloudNativePG

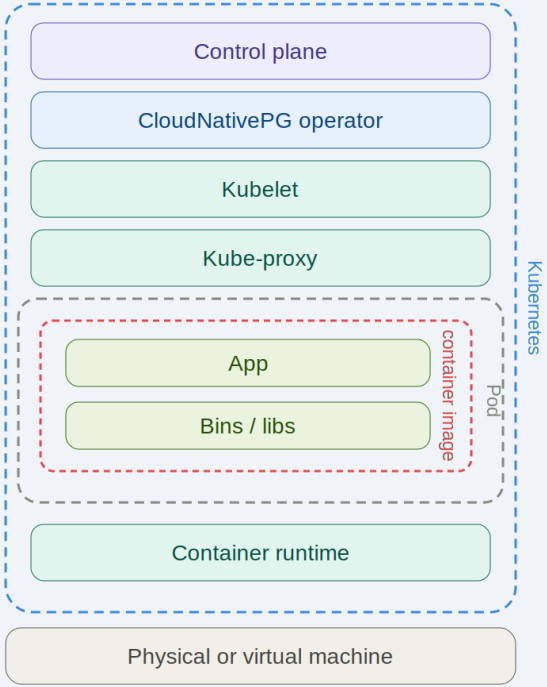
- Possivelmente a sua stack irá aumentar.

kubectl CNPG plugin

Barman Cloud
CNPG-I plugin

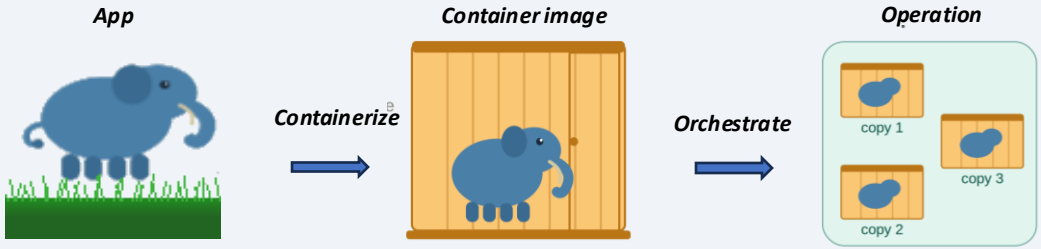
Contextualizando o CloudNativePG





Contextualizando o CloudNativePG

- Containers, isolamento de funcionalidades e gerenciamento de recursos (cpu, memória, disco)
- Operadores Kubernetes são extensões de software que fazem uso de **recursos customizados** para gerenciar aplicações e seus componentes. Quando esses recursos são combinados com **controladores customizados**, se obtém uma **API declarativa**
- **Arquitetura de microsserviços.**
- **Configuração declarativa**



Princípios fundamentais do CloudNativePG

- O operador monitora e atualiza constantemente o estado do cluster.

- Garente o correto **estado de funcionamento** do ambiente por meio de:
  - Nnível "aplicação"/banco de dados: **replicação física** para que o BD PostgreSQL mantenha o seu stateful workload no Kubernetes.
  - Nível POD: implementa o **próprio controlador customizado** para gerenciar instancias PostgreSQL.

- Definem que storage/armazenamento é o componente mais crítico para um banco de dados, gerenciando diretamente PVCs (adoção de controlador customizado em vez de controlador StatefulSet).

- **Immutable application containers**

Funcionalidades do CloudNativePG

Infrastructure & Architecture

- Bare metal
- Virtual machines
- On-premises
- Single / Multi Cloud
- Single / Multi region
- Single / Multi zone

Resources

- Clusters
- Replicas
- Connection poolers
  Native support for PgBouncer
- Load balancer
- Storage
  Declarative Tablespaces (can't be removed from the cluster)

Replication

- Physical (async e sync)
  Failover
  Switchover
- Logical
- Replica cluster
  Standalone replica
  From primary
  Cascading replication

Funcionalidades do CloudNativePG

Backup & Recovery

- CPNG-I Plugins
  - Barman cloud plugin
- Backups
  - WAL archiving + base backup
  - Object storage
- PITR
- Volume snapshot
  - Storage must support volume snapshots
- Replica Cluster
- Scheduled and on-demand Backups

Security

- TLS connections
- TLS client auth
- PGAudit
- RBAC (Operator x K8S)

Observability

- Logs to stdout
- PostgreSQL metrics exporter
- Kubernetes events

Funcionalidades do CloudNativePG

Performance & Reliability

- HA
- Self-healing
- Scale up/down de recursos
- Pod affinity/anti-affinity
- Labels, annotations, taints

Upgrades

- CloudNativePG operator
  - Release notes orientations
  - Applying the manifest or native package manager
- PostgreSQL Minor version
  - Rolling update of the cluster
- PostgreSQL Major version
  - Logical dump/restore
  - Native logical replication
  - Physical com pg_upgrade -
  - Offline in-place upgrade

Suporte e requisitos do CloudNativePG

- **Suporte das versões do CloudNativePG x Kubernetes x PostgreSQL**

Versão	Suportado atualmente?	Data de lançamento	End of Life	Versões suportadas de Kubernetes (standard k8s)	Testado, mas não suportado	Versões suportadas PostgreSQL
1.29.x	Sim	31 Mar 2026	~ Set 2026	1.33, 1.34, 1.35	1.29, 1.30, 1.31, 1.32	14 – 18

- **Kubernetes container runtime**

① **Versões 1.33, 1.34, 1.35:** Kubernetes requer um container runtime que atenda ao protocolo CRI (container runtime interface). Ex: containerd, CRI-O, Docker Engine, Mirantis Container Runtime.

- **Requisitos e recomendações Barman Cloud Plugin**

Versão	Requisito	Suporte	Recomendações	
	CloudNativePG	Armazenamento de objetos	cert-manager (for k8s)	Kubectl cng plugin
0.12.0	1.26 ou superior (recomendado 1.27 +)	Amazon S3, Microsoft Azure Blob Storage, Google Cloud Storage. * Pode utilizar implementações compatíveis com S3 ou Azure dos serviços acima.	Recomendado. Alternativamente, providenciar os próprios certificados.	Opcional, mas recomendado

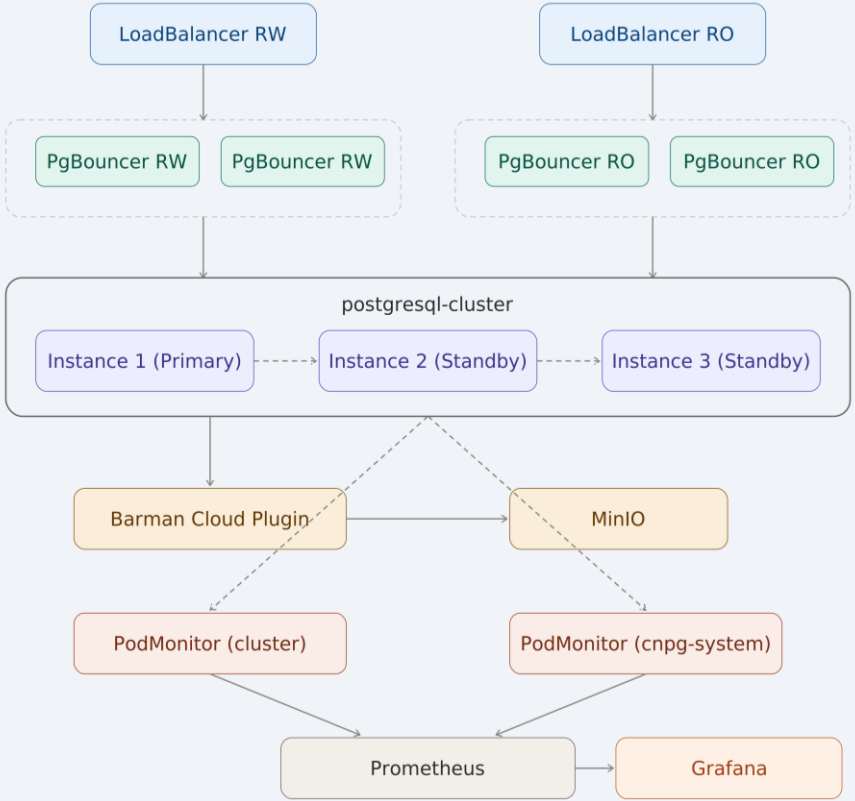
Suporte e requisitos do CloudNativePG

- **CNPG PostgreSQL Container Images**
  - Imagens são publicadas para todos os lançamentos Debian suportados e para PostgreSQL suportados pelo PGDG.
  - PostgreSQL operand container images feitas sobre a oficial Debian slim base image, para as arquiteturas linux/amd64 e linux/arm64.
  - Tipos de imagem: Minimal (lightweight), Standard (minimal extended) e System (standard + barman cloud binaries, descontinuada 09/2025)

- **Requisitos da container image do PostgreSQL no CloudNativePG**
  - É possível criar e utilizar imagens customizadas
  - Os executáveis do PostgreSQL devem estar no system path (initdb, postgres, pg_ctl, pg_controldata, pg_basebackup)
  - Componentes opcionais: PGAudit, du (usado por kubectl cnpg status), executáveis barman clound (a depender da versão do operador cnpg)

Demonstração - Ambiente a ser provisionado

- **Ambiente a ser provisionado**



Demonstração - Requisitos

- **Requisitos**
  - Ubuntu server 26.04 LTS
  - Minikube (1.38.1, kubernetes 1.35.1 compatible + Podman + Containerd)
  - Minikube addons (csi-hostpath-driver, volumesnapshots, metrics-server)
  - Podman (5.7.0)
  - Helm (3.20.0)
  - CloudNativePG (1.29.1)
  - CNPG plugin (1.29.1)
  - Barman Cloud Plugin (0.12)
  - cmctl (1.20.2)
  - MinIO
  - Kube-Prometheus-stack (Prometheus + Alert manager + Grafana)

Demonstração - Podman + Minikube install

```
lemos@ubuntuserverpg:~$ sudo apt install podman podman-compose -y
...
lemos@ubuntuserverpg:~$ podman --version
podman version 5.7.0
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0      0   0      0   0      0      0      0                              0
  0      0   0      0   0      0      0      0                              0
100 128.6M 100 128.6M   0      0 25.10M      0   00:05   00:05         29.74M
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube version
minikube version: v1.38.1
commit: c93a4cb9311efc66b90d33ea03f75f2c4120e9b0
```

Demonstração - Podman + Minikube start

```
lemos@ubuntuserverpg:~$ minikube config set rootless true
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube start --driver=podman --container-runtime=containerd --kubernetes-version=v1.35.1 --addons=csi-hostpath-driver --addons=volumesnapshots --addons=metrics-server
...
lemos@ubuntuserverpg:~$ minikube kubectl -- get pods -A
    > kubectl.sha256:  64 B / 64 B [-------------------------] 100.00% ? p/s 0s
    > kubectl:  55.88 MiB / 55.88 MiB [-----------] 100.00% 859.50 KiB p/s 1m7s
NAMESPACE     NAME                                READY   STATUS    RESTARTS   AGE
kube-system   coredns-7d764666f9-xtnh2            1/1     Running   0          4m7s
kube-system   csi-hostpath-attacher-0             1/1     Running   0          4m7s
kube-system   csi-hostpath-resizer-0              1/1     Running   0          4m7s
kube-system   csi-hostpathplugin-4qmvp            6/6     Running   0          3m43s
kube-system   etcd-minikube                       1/1     Running   0          4m12s
kube-system   kindnet-79t7j                       1/1     Running   0          4m7s
kube-system   kube-apiserver-minikube             1/1     Running   0          4m12s
kube-system   kube-controller-manager-minikube    1/1     Running   0          4m12s
kube-system   kube-proxy-lwgfc                    1/1     Running   0          4m7s
kube-system   kube-scheduler-minikube             1/1     Running   0          4m18s
kube-system   metrics-server-9d74bb658-m2qft      1/1     Running   0          4m7s
kube-system   snapshot-controller-6588d87457-5fqkm   1/1     Running   0          4m7s
kube-system   snapshot-controller-6588d87457-krcww   1/1     Running   0          4m7s
kube-system   storage-provisioner                 1/1     Running   0          4m10s
```

Demonstração - Helm install

```
lemos@ubuntuserverpg:~$ sudo apt-get install curl gpg apt-transport-https --yes
...
lemos@ubuntuserverpg:~$ curl -fsSL https://packages.buildkite.com/helm-linux/helm-debian/gpgkey | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ echo "deb [signed-by=/usr/share/keyrings/helm.gpg] https://packages.buildkite.com/helm-linux/helm-debian/any/ any main" | sudo tee /etc/apt/sources.l
ist.d/helm-stable-debian.list
deb [signed-by=/usr/share/keyrings/helm.gpg] https://packages.buildkite.com/helm-linux/helm-debian/any/ any main
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ sudo apt-get update
...
lemos@ubuntuserverpg:~$ sudo apt-get install helm
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ helm version
version.BuildInfo{Version:"v3.20.0", GitCommit:"b2e4314fa0f229a1de7b4c981273f61d69ee5a59", GitTreeState:"clean", GoVersion:"go1.25.6"}
```

Demonstração - CNPG install

```
lemos@ubuntuserverpg:~$ minikube kubectl -- apply --server-side -f \
  https://raw.githubusercontent.com/cloudnative-pg/cloudnative-pg/release-1.29/releases/cnpg-1.29.1.yaml
namespace/cnpg-system serverside-applied
customresourcedefinition.apiextensions.k8s.io/backups.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/clusterimagecatalogs.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/clusters.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/databases.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/failoverquorums.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/imagecatalogs.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/poolers.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/publications.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/scheduledbackups.postgresql.cnpg.io serverside-applied
customresourcedefinition.apiextensions.k8s.io/subscriptions.postgresql.cnpg.io serverside-applied
serviceaccount/cnpg-manager serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-database-editor-role serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-database-viewer-role serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-manager serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-publication-editor-role serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-publication-viewer-role serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-subscription-editor-role serverside-applied
clusterrole.rbac.authorization.k8s.io/cnpg-subscription-viewer-role serverside-applied
clusterrolebinding.rbac.authorization.k8s.io/cnpg-manager-rolebinding serverside-applied
configmap/cnpg-default-monitoring serverside-applied
service/cnpg-webhook-service serverside-applied
deployment.apps/cnpg-controller-manager serverside-applied
mutatingwebhookconfiguration.admissionregistration.k8s.io/cnpg-mutating-webhook-configuration serverside-applied
validatingwebhookconfiguration.admissionregistration.k8s.io/cnpg-validating-webhook-configuration serverside-applied
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- rollout status deployment -n cnpg-system cnpg-controller-manager
deployment "cnpg-controller-manager" successfully rolled out
```

Demonstração - CNPG plugin install

```
lemos@ubuntuserverpg:~$ wget https://github.com/cloudnative-pg/cloudnative-pg/releases/download/v1.29.1/kubectl-cnpg_1.29.1_linux_x86_64.deb \
  --output-document kube-plugin.deb
...
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ sudo dpkg -i kube-plugin.deb
[sudo: authenticate] Password:
Selecting previously unselected package cnpg.
(Reading database ... 140563 files and directories currently installed.)
Preparing to unpack kube-plugin.deb ...
Unpacking cnpg (1.29.1) ...
Setting up cnpg (1.29.1) ...
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg version
Build: {Version:1.29.1 Commit:a4060c152 Date:2026-05-08}
```

Demonstração - Provisionamento e gerenciamento

```
lemos@ubuntuserverpg:~$ sudo apt install golang-go
...
lemos@ubuntuserverpg:~$ OS=$(go env GOOS); ARCH=$(go env GOARCH); curl -fsSL -o cmctl https://github.com/cert-manager/cmctl/releases/latest/download/cmctl_${OS}_${ARCH}
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ chmod +x cmctl
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ sudo mv cmctl /usr/local/bin
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f \
  https://github.com/cert-manager/cert-manager/releases/latest/download/cert-manager.yaml
 ...
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f \
        https://github.com/cloudnative-pg/plugin-barman-cloud/releases/download/v0.12.0/manifest.yaml
...
lemos@ubuntuserverpg:~$ minikube kubectl -- rollout status deployment \
  -n cnpg-system barman-cloud
deployment "barman-cloud" successfully rolled out
```

Demonstração - Namespace creation

```
lemos@ubuntuserverpg:~$ minikube kubectl create namespace cloudnative-pg
namespace/cloudnative-pg created
```

Demonstração - Minio deployment + Bucket + Object store

```
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f minio.yaml
persistentvolume/minio-pv created
persistentvolumeclaim/minio-pvc created
secret/minio-secret created
deployment.apps/minio created
service/minio created
secret/minio-creds created
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- rollout status deployment/minio -n cloudnative-pg
deployment "minio" successfully rolled out
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- run minio-create-bucket \
  --image=minio/mc \
  --restart=Never \
  --namespace=cloudnative-pg \
  --quiet \
  --rm -i \
  --command \
  -- /bin/sh -c "mc alias set local http://minio.cloudnative-pg.svc.cluster.local:9000 ${MINIO_USER_NAME} ${MINIO_USER_PASSWORD} &&
                 mc mb local/pg-backups &&
                 echo 'Bucket created successfully'"
Added `local` successfully.
Bucket created successfully `local/pg-backups`.
Bucket created successfully
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f barman-minio-objectstore.yaml
objectstore.barmancloud.cnpg.io/minio-store created
```

Demonstração - Prometheus + Grafana + Zone label config

```
lemos@ubuntuserverpg:~$ helm repo add prometheus-community \
  https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ helm upgrade --install \
  -f https://raw.githubusercontent.com/cloudnative-pg/cloudnative-pg/main/docs/src/samples/monitoring/kube-stack-config.yaml \
  prometheus-community \
  prometheus-community/kube-prometheus-stack \
  --namespace cloudnative-pg
...
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- apply --namespace cloudnative-pg -f \
  https://raw.githubusercontent.com/cloudnative-pg/cloudnative-pg/main/docs/src/samples/monitoring/prometheusrule.yaml
prometheusrule.monitoring.coreos.com/cnpg-default-alerts created
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- label node minikube topology.kubernetes.io/zone=example-zone-1a
node/minikube labeled
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get nodes -L topology.kubernetes.io/zone
NAME       STATUS   ROLES           AGE   VERSION   ZONE
minikube   Ready    control-plane   43m   v1.35.1   example-zone-1a
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ helm upgrade prometheus-community prometheus-community/kube-prometheus-stack \
  -f metrics_label.yaml \
  --namespace cloudnative-pg \
  --reuse-values
...
```

Demonstração - PostgreSQL cluster creation

```
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f postgresql-cluster.yaml
secret/minio-backup-credential created
secret/postgresql-cluster-apptest-secret created
objectstore.barmancloud.cnpg.io/minio-store configured
cluster.postgresql.cnpg.io/postgresql-cluster created
database.postgresql.cnpg.io/postgresql-cluster-dbtest created
podmonitor.monitoring.coreos.com/postgresql-cluster created
podmonitor.monitoring.coreos.com/cnpg-controller-manager created
pooler.postgresql.cnpg.io/postgresql-cluster-pooler-rw created
pooler.postgresql.cnpg.io/postgresql-cluster-pooler-ro created
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get cluster postgresql-cluster -n cloudnative-pg
NAME                AGE     INSTANCES   READY   STATUS                     PRIMARY
postgresql-cluster   6m14s   3           3       Cluster in healthy state   postgresql-cluster-1
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get pods -n cloudnative-pg
NAME                                                  READY   STATUS    RESTARTS   AGE
alertmanager-prometheus-community-kube-alertmanager-0   2/2     Running   0          91m
minio-5f5f6747fc-h7zcf                                1/1     Running   0          97m
postgresql-cluster-1                                  2/2     Running   0          5m26s
postgresql-cluster-2                                  2/2     Running   0          101s
postgresql-cluster-3                                  2/2     Running   0          49s
postgresql-cluster-pooler-ro-77bcdf6bb7-2wpd9         1/1     Running   0          6m12s
postgresql-cluster-pooler-ro-77bcdf6bb7-jt525         1/1     Running   0          6m12s
postgresql-cluster-pooler-rw-9864695f-vd7hj           1/1     Running   0          6m12s
postgresql-cluster-pooler-rw-9864695f-xfx5d           1/1     Running   0          6m12s
prometheus-community-grafana-f58fbb89c-fbfb6          3/3     Running   0          92m
prometheus-community-kube-operator-659758d864-5sx2w   1/1     Running   0          92m
prometheus-community-kube-state-metrics-f958c4957-54t5h   1/1     Running   0          90m
prometheus-prometheus-community-kube-prometheus-0     2/2     Running   0          91m
```

Demonstração - PostgreSQL cluster creation

```
lemos@ubuntuserverpg:~$ minikube kubectl -- get pvc -n cloudnative-pg
NAME                       STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS      VOLUMEATTRIBUTESCLASS   AGE
minio-pvc                  Bound    pvc-ea5104fb-1a58-419c-8bf9-3324eee4970b   10Gi       RWO            standard          <unset>                 100m
postgresql-cluster-1       Bound    pvc-202c750c-415c-42a7-a358-1510d8515cc2   10Gi       RWO            csi-hostpath-sc   <unset>                 9m15s
postgresql-cluster-1-wal   Bound    pvc-8a8aa2ba-88ed-4fa4-a785-a8781be2fed6   5Gi        RWO            csi-hostpath-sc   <unset>                 9m15s
postgresql-cluster-2       Bound    pvc-6682e7bf-6a5f-4de1-8b42-5ec6d39335b3   10Gi       RWO            csi-hostpath-sc   <unset>                 7m44s
postgresql-cluster-2-wal   Bound    pvc-a8ec8f7e-a831-4aa0-890b-f0dda41cac9d   5Gi        RWO            csi-hostpath-sc   <unset>                 7m44s
postgresql-cluster-3       Bound    pvc-91619487-13f2-41db-8715-aeebe7209896   10Gi       RWO            csi-hostpath-sc   <unset>                 4m20s
postgresql-cluster-3-wal   Bound    pvc-f7728ef4-af2e-4d06-af6c-dc2c8858aa95   5Gi        RWO            csi-hostpath-sc   <unset>                 4m20s
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get services -n cloudnative-pg
NAME                                      TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)                         AGE
alertmanager-operated                     ClusterIP      None             <none>        9093/TCP,9094/TCP,9094/UDP      94m
minio                                     NodePort       10.102.7.248     <none>        9000:30900/TCP,9001:30901/TCP   100m
postgresql-cluster-pooler-ro              LoadBalancer   10.104.131.5     <pending>     5432:31968/TCP                  9m14s
postgresql-cluster-pooler-rw              LoadBalancer   10.110.23.160    <pending>     5432:32252/TCP                  9m14s
postgresql-cluster-r                      ClusterIP      10.107.215.244   <none>        5432/TCP                        9m18s
postgresql-cluster-ro                     ClusterIP      10.97.117.130    <none>        5432/TCP                        9m18s
postgresql-cluster-rw                     ClusterIP      10.111.73.215    <none>        5432/TCP                        9m18s
prometheus-community-grafana              ClusterIP      10.105.67.157    <none>        80/TCP                          95m
prometheus-community-kube-alertmanager    ClusterIP      10.99.18.55      <none>        9093/TCP,8080/TCP               95m
prometheus-community-kube-operator        ClusterIP      10.104.28.46     <none>        443/TCP                         95m
prometheus-community-kube-prometheus      ClusterIP      10.110.38.206    <none>        9090/TCP,8080/TCP               95m
prometheus-community-kube-state-metrics   ClusterIP      10.102.124.202   <none>        8080/TCP                        95m
prometheus-operated                       ClusterIP      None             <none>        9090/TCP                        94m
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get secrets -n cloudnative-pg
NAME                                                                  TYPE                       DATA   AGE
alertmanager-prometheus-community-kube-alertmanager                   Opaque                     1      95m
alertmanager-prometheus-community-kube-alertmanager-cluster-tls-config   Opaque                     1      94m
alertmanager-prometheus-community-kube-alertmanager-generated         Opaque                     1      94m
alertmanager-prometheus-community-kube-alertmanager-tls-assets-0      Opaque                     0      94m
alertmanager-prometheus-community-kube-alertmanager-web-config        Opaque                     1      94m
minio-backup-credential                                               Opaque                     2      9m30s
minio-creds                                                           Opaque                     2      100m
minio-secret                                                          Opaque                     2      100m
postgresql-cluster-app                                                kubernetes.io/basic-auth   11     9m27s
postgresql-cluster-apptest-secret                                     kubernetes.io/basic-auth   2      9m30s
postgresql-cluster-ca                                                 Opaque                     2      9m27s
postgresql-cluster-pooler                                             kubernetes.io/tls          2      9m24s
postgresql-cluster-replication                                        kubernetes.io/tls          2      9m27s
postgresql-cluster-server                                             kubernetes.io/tls          2      9m27s
prometheus-community-grafana                                          Opaque                     3      95m
prometheus-community-kube-admission                                   Opaque                     3      95m
prometheus-prometheus-community-kube-prometheus                       Opaque                     1      94m
prometheus-prometheus-community-kube-prometheus-thanos-prometheus-http-client-file   Opaque                     1      94m
prometheus-prometheus-community-kube-prometheus-tls-assets-0          Opaque                     1      94m
prometheus-prometheus-community-kube-prometheus-web-config            Opaque                     1      94m
sh.helm.release.v1.prometheus-community.v1                            helm.sh/release.v1         1      95m
sh.helm.release.v1.prometheus-community.v2                            helm.sh/release.v1         1      93m
```

Demonstração - PostgreSQL cluster creation

```
lemos@ubuntuserverpg:~$ minikube kubectl -- run minio-mc \
  -n cloudnative-pg \
  --rm -it \
  --restart=Never \
  --image=minio/mc \
  --quiet \
  --command -- /bin/sh -c "
    mc alias set local http://minio.cloudnative-pg.svc.cluster.local:9000 ${MINIO_USER_NAME} ${MINIO_USER_PASSWORD} >/dev/null 2>&1 &&
    mc ls local/pg-backups --recursive | sort -rk1,2
  "
[2026-05-28 18:26:23 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000006.gz
[2026-05-28 18:21:23 UTC]   197B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000005.00000028.backup.gz
[2026-05-28 18:21:22 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000005.gz
[2026-05-28 18:21:12 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000004.gz
[2026-05-28 18:20:25 UTC]   203B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000003.00000028.backup.gz
[2026-05-28 18:20:24 UTC] 190KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000003.gz
[2026-05-28 18:18:11 UTC] 1.4MiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000002.gz
[2026-05-28 18:17:25 UTC] 2.6MiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000001.gz
```

Demonstração - Scheduled backup configuration

```
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f scheduled_backup.yaml
scheduledbackup.postgresql.cnpg.io/postgresql-cluster-sched-bkp created
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get backup -n cloudnative-pg -l cnpg.io/cluster=postgresql-cluster
NAME                                          AGE   CLUSTER              METHOD   PHASE       ERROR
postgresql-cluster-sched-bkp-20260528182802   46s   postgresql-cluster   plugin   completed
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- describe scheduledbackup postgresql-cluster-sched-bkp -n cloudnative-pg
Name:         postgresql-cluster-sched-bkp
Namespace:    cloudnative-pg
Labels:       <none>
Annotations:  <none>
API Version:  postgresql.cnpg.io/v1
Kind:         ScheduledBackup
Metadata:
  Creation Timestamp:  2026-05-28T18:28:02Z
  Generation:          1
  Resource Version:    18922
  UID:                 bb9abc28-f873-471b-952c-767e341505bc
Spec:
  Backup Owner Reference:  self
  Cluster:
    Name:     postgresql-cluster
  Immediate:  true
  Method:     plugin
  Plugin Configuration:
    Name:    barman-cloud.cloudnative-pg.io
  Schedule:  0 0 0 * * *
Status:
  Last Check Time:     2026-05-28T18:28:02Z
  Last Schedule Time:  2026-05-28T18:28:02Z
  Next Schedule Time:  2026-05-29T00:00:00Z
Events:
  Type    Reason          Age   From                           Message
  ----    ------          ----  ----                           -------
  Normal  BackupSchedule  60s   cloudnative-pg-scheduledbackup  Scheduled immediate backup now: 2026-05-28 18:28:02.344564687 +0000 UTC m=+7070.041333963
  Normal  BackupSchedule  60s   cloudnative-pg-scheduledbackup  Next backup scheduled by 2026-05-29 00:00:00 +0000 UTC
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- run minio-mc \
  -n cloudnative-pg \
  --rm -it \
  --restart=Never \
  --image=minio/mc \
  --quiet \
  --command -- /bin/sh -c "
    mc alias set local http://minio.cloudnative-pg.svc.cluster.local:9000 ${MINIO_USER_NAME} ${MINIO_USER_PASSWORD} >/dev/null 2>&1 &&
    mc ls local/pg-backups --recursive | sort -rk1,2
  "
[2026-05-28 18:28:10 UTC] 5.2MiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T152804/data.tar.gz
[2026-05-28 18:28:10 UTC] 1.4KiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T152804/backup.info
[2026-05-28 18:26:23 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000006.gz
[2026-05-28 18:21:23 UTC]   197B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000005.00000028.backup.gz
[2026-05-28 18:21:22 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000005.gz
[2026-05-28 18:21:12 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000004.gz
[2026-05-28 18:20:25 UTC]   203B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000003.00000028.backup.gz
[2026-05-28 18:20:24 UTC] 190KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000003.gz
[2026-05-28 18:18:11 UTC] 1.4MiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000002.gz
[2026-05-28 18:17:25 UTC] 2.6MiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000001.gz
```

Demonstração - Backup on-demand

```
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f backup_on-demand.yaml
backup.postgresql.cnpg.io/backup-on-demand created
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- describe backup backup-on-demand -n cloudnative-pg
Name:         backup-on-demand
Namespace:    cloudnative-pg
Labels:       <none>
Annotations:  <none>
API Version:  postgresql.cnpg.io/v1
Kind:         Backup
Metadata:
  Creation Timestamp:  2026-05-28T18:31:58Z
  Generation:          1
  Resource Version:    19493
  UID:                 789ffdc6-73c0-40ed-b607-43dd7685bfe2
Spec:
  Cluster:
    Name:  postgresql-cluster
  Method:  plugin
  Plugin Configuration:
    Name:  barman-cloud.cloudnative-pg.io
Status:
  Instance Id:
    Container Id:             containerd://f7bb99c13ca48d583fb55cca188c9d6139923f6a61241c0b8afaf464b45b68ca
    Pod Name:                 postgresql-cluster-2
    Session Id:               5cec8154-b588-4e4e-838f-1451e686b1a3
  Major Version:              18
  Method:                     plugin
  Phase:                      started
  Reconciliation Started At:  2026-05-28T18:31:58Z
Events:
  Type    Reason    Age   From                   Message
  ----    ------    ----  ----                   -------
  Normal  Starting  6s    cloudnative-pg-backup  Starting backup for cluster postgresql-cluster
  Normal  Starting  6s    local-webserver        Backup started
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get backup -n cloudnative-pg
NAME                                          AGE     CLUSTER              METHOD   PHASE       ERROR
backup-on-demand                              83s     postgresql-cluster   plugin   completed
postgresql-cluster-sched-bkp-20260528182802   5m19s   postgresql-cluster   plugin   completed
lemos@ubuntuserverpg:~$ minikube kubectl -- run minio-mc \
  -n cloudnative-pg \
  --rm -it \
  --restart=Never \
  --image=minio/mc \
  --quiet \
  --command -- /bin/sh -c "
    mc alias set local http://minio.cloudnative-pg.svc.cluster.local:9000 ${MINIO_USER_NAME} ${MINIO_USER_PASSWORD} >/dev/null 2>&1 &&
    mc ls local/pg-backups --recursive | sort -rk1,2
  "
[2026-05-28 18:32:04 UTC] 5.2MiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T153159/data.tar.gz
[2026-05-28 18:32:04 UTC] 1.4KiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T153159/backup.info
[2026-05-28 18:28:10 UTC] 5.2MiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T152804/data.tar.gz
[2026-05-28 18:28:10 UTC] 1.4KiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T152804/backup.info
[2026-05-28 18:26:23 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000006.gz
[2026-05-28 18:21:23 UTC]   197B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000005.00000028.backup.gz
[2026-05-28 18:21:22 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000005.gz
[2026-05-28 18:21:12 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000004.gz
[2026-05-28 18:20:25 UTC]   203B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000003.00000028.backup.gz
[2026-05-28 18:20:24 UTC] 190KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/000000010000000000000003.gz
```

Demonstração - Manual switchover

```
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg status postgresql-cluster -n cloudnative-pg | grep -i -A5 "Instances status"
Instances status
Name                 Current LSN  Replication role  Status  QoS        Manager Version  Node
----                 -----------  ----------------  ------  ---        ---------------  ----
postgresql-cluster-1  0/7000000    Primary           OK      Burstable  1.29.1           minikube
postgresql-cluster-2  0/7000000    Standby (async)   OK      Burstable  1.29.1           minikube
postgresql-cluster-3  0/7000000    Standby (async)   OK      Burstable  1.29.1           minikube
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg promote postgresql-cluster postgresql-cluster-2 -n cloudnative-pg
{"level":"info","ts":"2026-05-28T15:41:12.924590897-03:00","msg":"Cluster has become unhealthy"}
Node postgresql-cluster-2 in cluster postgresql-cluster will be promoted
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg status postgresql-cluster -n cloudnative-pg | grep -i -A5 "Instances status"
Instances status
Name                 Current LSN  Replication role  Status  QoS        Manager Version  Node
----                 -----------  ----------------  ------  ---        ---------------  ----
postgresql-cluster-2  0/80055D8    Primary           OK      Burstable  1.29.1           minikube
postgresql-cluster-1  0/80055D8    Standby (async)   OK      Burstable  1.29.1           minikube
postgresql-cluster-3  0/80055D8    Standby (async)   OK      Burstable  1.29.1           minikube
```

Demonstração - Automatic failover triggered by pod deletion

```
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg status postgresql-cluster -n cloudnative-pg | grep -i -A5 "Instances status"
Instances status
Name                 Current LSN  Replication role  Status  QoS        Manager Version  Node
----                 -----------  ----------------  ------  ---        ---------------  ----
postgresql-cluster-2  0/9000110    Primary           OK      Burstable  1.29.1           minikube
postgresql-cluster-1  0/9000110    Standby (async)   OK      Burstable  1.29.1           minikube
postgresql-cluster-3  0/9000110    Standby (async)   OK      Burstable  1.29.1           minikube
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- delete pod postgresql-cluster-2 -n cloudnative-pg
pod "postgresql-cluster-2" deleted from cloudnative-pg namespace
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg status postgresql-cluster -n cloudnative-pg | grep -i -A5 "Instances status"
Instances status
Name                 Current LSN  Replication role  Status         QoS        Manager Version  Node
----                 -----------  ----------------  ------         ---        ---------------  ----
postgresql-cluster-1  0/A005060    Primary           OK             Burstable  1.29.1           minikube
postgresql-cluster-3  0/A005060    Standby (async)   OK             Burstable  1.29.1           minikube
postgresql-cluster-2  -            -                 InternalError  Burstable  -                minikube
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- cnpg status postgresql-cluster -n cloudnative-pg | grep -i -A5 "Instances status"
Instances status
Name                 Current LSN  Replication role  Status  QoS        Manager Version  Node
----                 -----------  ----------------  ------  ---        ---------------  ----
postgresql-cluster-2  0/9000110    Primary           OK      Burstable  1.29.1           minikube
postgresql-cluster-1  0/9000110    Standby (async)   OK      Burstable  1.29.1           minikube
postgresql-cluster-3  0/9000110    Standby (async)   OK      Burstable  1.29.1           minikube
```

Demonstração - Backup recovery - Cluster bootstrap initialization

```
lemos@ubuntuserverpg:~$ ./simulate_data_loss.sh
...
==================================================
 Summary
==================================================

  [1] Created table test_table, inserted rows (1,teste1) and (2,teste2)
  [2] Pre-delete timestamp : 2026-05-29 09:19:48.250085-03
  [3] Deleted row id=2
  [4] WAL LSN              : 0/F026BC8
  [4] WAL LSN file         : 000000030000000000000000F
  [5] Post-WAL timestamp   : 2026-05-29 09:19:49.646057-03
  [6] Confirmed row id=2 deleted (idempotent)
  [7] Switched WAL         : 0/F026BE0
  [8] WAL file             : 000000030000000000000010
==================================================
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- run minio-mc   -n cloudnative-pg   --rm -it   --restart=Never   --image=minio/mc   --quiet   --command -- /bin/sh -c "
    mc alias set local http://minio.cloudnative-pg.svc.cluster.local:9000 ${MINIO_USER_NAME} ${MINIO_USER_PASSWORD} >/dev/null 2>&1 &&
    mc ls local/pg-backups --recursive | sort -rk1,2
  "
[2026-05-29 12:19:50 UTC]  55KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000300000000/000000030000000000000000F.gz
[2026-05-29 12:17:16 UTC]  18KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000300000000/000000030000000000000000E.gz
[2026-05-29 12:12:54 UTC] 5.2MiB STANDARD postgresql-cluster/postgresql-cluster/base/20260529T091248/data.tar.gz
[2026-05-29 12:12:54 UTC] 1.4KiB STANDARD postgresql-cluster/postgresql-cluster/base/20260529T091248/backup.info
[2026-05-28 19:29:29 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000300000000/000000030000000000000000D.gz
[2026-05-28 19:24:28 UTC]  22KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000300000000/000000030000000000000000C.gz
[2026-05-28 18:56:52 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000300000000/000000030000000000000000B.gz
[2026-05-28 18:51:52 UTC]  54KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000300000000/000000030000000000000000A.gz
[2026-05-28 18:48:46 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000200000000/000000020000000000000000A.partial.gz
[2026-05-28 18:48:45 UTC]    67B STANDARD postgresql-cluster/postgresql-cluster/wals/00000003.history.gz
[2026-05-28 18:48:40 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000200000000/0000000200000000000000009.gz
[2026-05-28 18:46:18 UTC]  19KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000200000000/0000000200000000000000008.gz
[2026-05-28 18:41:20 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000008.partial.gz
[2026-05-28 18:41:19 UTC]    59B STANDARD postgresql-cluster/postgresql-cluster/wals/00000002.history.gz
[2026-05-28 18:41:13 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000007.gz
[2026-05-28 18:32:04 UTC] 5.2MiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T153159/data.tar.gz
[2026-05-28 18:32:04 UTC] 1.4KiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T153159/backup.info
[2026-05-28 18:28:10 UTC] 5.2MiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T152804/data.tar.gz
[2026-05-28 18:28:10 UTC] 1.4KiB STANDARD postgresql-cluster/postgresql-cluster/base/20260528T152804/backup.info
[2026-05-28 18:26:23 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000006.gz
[2026-05-28 18:21:23 UTC]   197B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000005.00000028.backup.gz
[2026-05-28 18:21:22 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000005.gz
[2026-05-28 18:21:12 UTC]  16KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000004.gz
[2026-05-28 18:20:25 UTC]   203B STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000003.00000028.backup.gz
[2026-05-28 18:20:24 UTC] 190KiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000003.gz
[2026-05-28 18:18:11 UTC] 1.4MiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000002.gz
[2026-05-28 18:17:25 UTC] 2.6MiB STANDARD postgresql-cluster/postgresql-cluster/wals/0000000100000000/0000000100000000000000001.gz
```

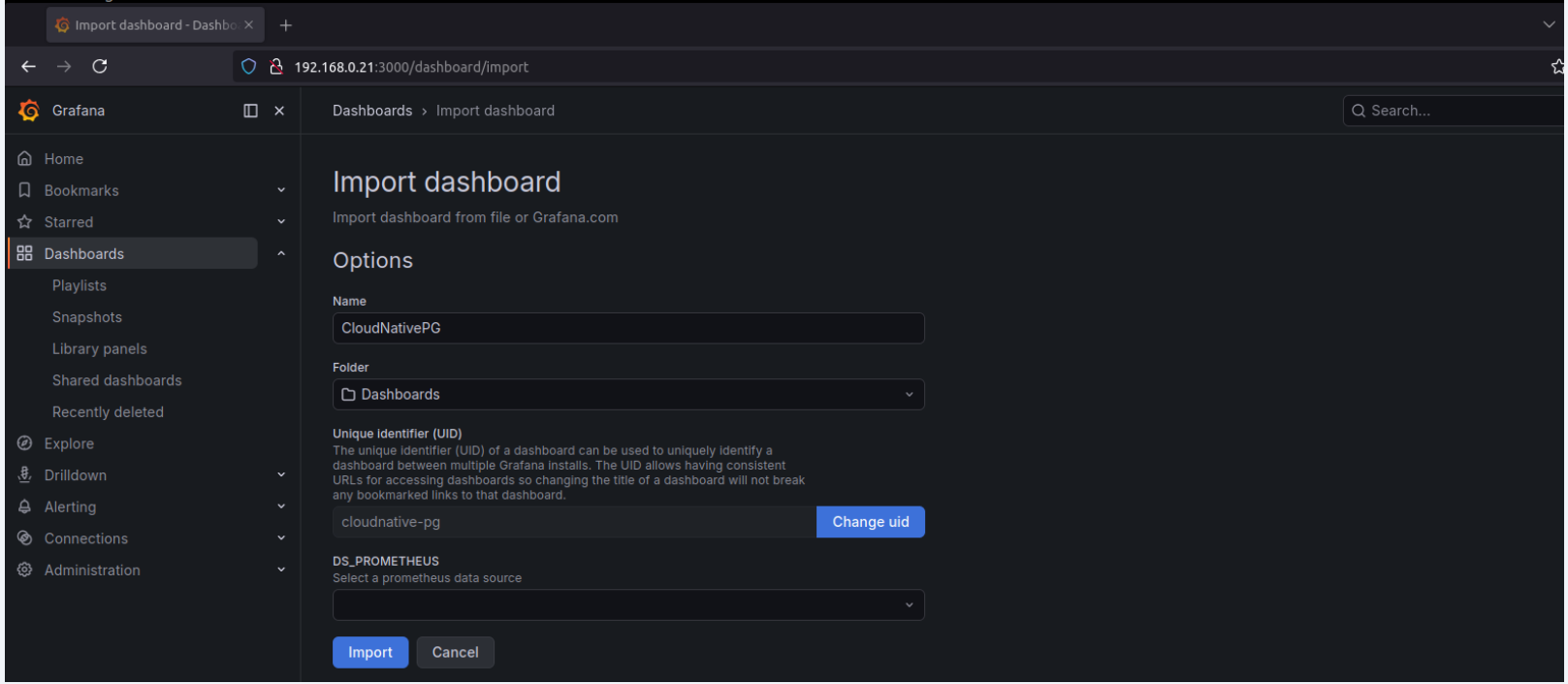
Demonstração - Backup recovery - Cluster bootstrap initialization

```
lemos@ubuntuserverpg:~$ minikube kubectl -- apply -f postgresql-cluster-recovery.yaml
cluster.postgresql.cnpg.io/postgresql-cluster-recovery created
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get cluster postgresql-cluster-recovery -n cloudnative-pg
NAME                          AGE   INSTANCES   READY   STATUS              PRIMARY
postgresql-cluster-recovery   5s    1                   Setting up primary
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get events -n cloudnative-pg --field-selector involvedObject.name=postgresql-cluster-recovery --sort-by='.lastTimestamp'
LAST SEEN   TYPE     REASON                       OBJECT                                MESSAGE
39s         Normal   CreatingPodDisruptionBudget   cluster/postgresql-cluster-recovery   Creating PodDisruptionBudget postgresql-cluster-recovery-primary
39s         Normal   CreatingServiceAccount        cluster/postgresql-cluster-recovery   Creating ServiceAccount
39s         Normal   CreatingRole                  cluster/postgresql-cluster-recovery   Creating Cluster Role
39s         Normal   CreatingInstance              cluster/postgresql-cluster-recovery   Primary instance (from backup)
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get cluster postgresql-cluster-recovery -n cloudnative-pg
NAME                          AGE   INSTANCES   READY   STATUS                                     PRIMARY
postgresql-cluster-recovery   68s   1                   Waiting for the instances to become active
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- get cluster postgresql-cluster-recovery -n cloudnative-pg
NAME                          AGE    INSTANCES   READY   STATUS                     PRIMARY
postgresql-cluster-recovery   114s   1           1       Cluster in healthy state   postgresql-cluster-recovery-1
lemos@ubuntuserverpg:~$ ./select_table.sh
==> Finding primary pod...
    Primary pod: postgresql-cluster-recovery-1

==> Selecting content from test_table...
 id |  data
----+--------
  1 | teste1
  2 | teste2
(2 rows)
```

Demonstração - Prometheus + Grafana monitoring

```
lemos@ubuntuserverpg:~$ curl -O https://raw.githubusercontent.com/cloudnative-pg/grafana-dashboards/main/charts/cluster/grafana-dashboard.json
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 247.1k 100 247.1k    0     0 320.7k      0                              0
lemos@ubuntuserverpg:~$
lemos@ubuntuserverpg:~$ minikube kubectl -- --namespace cloudnative-pg port-forward svc/prometheus-community-grafana 3000:80 --address 0.0.0.0
Forwarding from 0.0.0.0:3000 -> 3000
```



Demonstração - Prometheus + Grafana monitoring

Referências

CLOUDNATIVEPG

- **CloudNativePG Website**
  cloudnative-pg.io
- **CloudNativePG Github**
  github.com/cloudnative-pg/cloudnative-pg
- **CloudNativePG Documentation (version: 1.29)**
  cloudnative-pg.io/docs/1.29/
- **CloudNativePG postgres-containers GitHub Container Registry**
  github.com/cloudnative-pg/postgres-containers

KUBERNETES

- **Kubernetes Documentation**
  kubernetes.io/docs/home/

POSTGRESQL

- **PostgreSQL Documentation (version: current)**
  postgresql.org/docs/current
- **PostgreSQL GIT Repository Mirror**
  github.com/postgres/postgres

GENERAL

- **Recommended architectures for PostgreSQL in Kubernetes**
  cncf.io/blog/2023/09/29/recommended-architectures-for-postgresql-in-kubernetes/
- **Unleashing the Power of Postgres in Kubernetes - Articles**
  gabrielebartolini.it/articles/
- **Introducing CloudNativePG: A New Open Source Kubernetes Operator for Postgres**
  enterprisedb.com/blog/introducing-cloudnativepg-new-open-source-kubernetes-operator-postgres
- **CNCF – CloudNativePG Project**
  cncf.io/projects/cloudnativepg/

BARMAN CLOUD CNPG-I PLUGIN

- **Barman Cloud CNPG-I Plugin Documentation**
  cloudnative-pg.io/plugin-barman-cloud/docs/intro/

Obrigado!

O seu tempo e atenção são importantes.