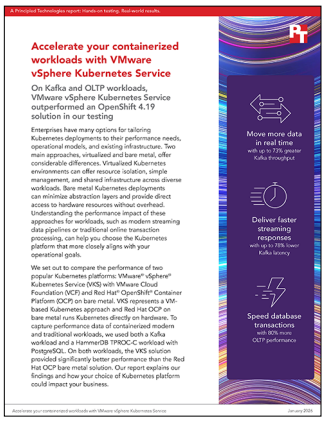

The science behind the report:

Accelerate your containerized workloads with VMware vSphere Kubernetes Service

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Accelerate your containerized workloads with VMware vSphere Kubernetes Service.

We concluded our hands-on testing on November 19, 2025. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on November 19, 2025 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Kafka producer workload results from the VMware vSphere Kubernetes Service (VKS) solution and two Red Hat OpenShift Container Platform (OCP) configurations. Higher throughput and lower latency are better Source: PT.

	VMware® vSphere® Kubernetes Service	Red Hat® OpenShift® Container Platform (virtualized on VCF)	VKS win % vs. OCP (virtualized)	Red Hat OpenShift Container Platform (bare metal)	VKS win % vs. OCP (bare metal)
One topic					
Throughput (MB/sec)	221.42	222.84	**-0.64%**	221.43	**-0.01%**
Latency (ms)	1.84	1.51	**-21.85%**	1.99	**7.54%**
Two topics					
Throughput (MB/sec)	436.56	445.62	**-2.03%**	420.10	**3.92%**
Latency (ms)	2.47	3.71	**33.33%**	6.64	**62.77%**
Four topics					
Throughput (MB/sec)	845.04	843.04	**0.24%**	723.01	**16.88%**

	VMware® vSphere® Kubernetes Service	Red Hat® OpenShift® Container Platform (virtualized on VCF)	VKS win % vs. OCP (virtualized)	Red Hat OpenShift Container Platform (bare metal)	VKS win % vs. OCP (bare metal)
Latency (ms)	8.24	14.20	**41.96%**	39.00	**78.87%**
Six topics					
Throughput (MB/sec)	1,202.66	1,146.54	**4.89%**	870.81	**38.11%**
Latency (ms)	22.51	117.91	**80.91%**	79.62	**71.73%**
Eight topics					
Throughput (MB/sec)	1,450.24	1,232.84	**17.63%**	833.72	**73.95%**
Latency (ms)	106.06	176.88	**40.04%**	156.99	**32.44%**

Table 2: HammerDB results from the VKS solution and two OCP configurations. Note that we did not gather latency data on the virtualized OCP solution, so we cannot report it here. Source: PT.

	VMware vSphere Kubernetes Service	Red Hat OpenShift Container Platform (virtualized on VCF)	VKS win % vs. OCP (virtualized)	Red Hat OpenShift Container Platform (bare metal)	VKS win % vs. OCP (bare metal)
New orders per minute (NOPM)	112,947	105,706	**6.85%**	62,459	**80.83%**
Latency (ms)	18.33	N/A	**N/A**	66.57	**72.46%**

System configuration information

Table 3: Detailed information on the systems we tested.

System configuration information	Dell™ PowerEdge™ R640	
	VCF – 4 nodes	OCP – 4 nodes
BIOS name and version	Dell 2.22.2	Dell 2.22.2
Non-default BIOS settings	None	None
Operating system name and version/build number	VMware ESXi™ 9.0.1.0, 24957456	OpenShift Container Platform 4.19
Date of last OS updates/patches applied	28 October 2025	28 October 2025
Power management policy	Performance	Performance
Processor		
Number of processors	2	2
Vendor and model	Intel® Xeon® Platinum 8260	Intel Xeon Platinum 8260
Core count (per processor)	24	24
Core frequency (GHz)	2.4	2.4
Stepping	7	7
Memory module(s)		
Total memory in system (GB)	768	768
Number of memory modules	24	24
Vendor and model	Hynix HMA84GR7CJR4N-WM	Hynix HMA84GR7CJR4N-WM
Size (GB)	32	32
Type	DDR4	DDR4
Speed (MHz)	2933	2933
Speed running in the server (MHz)	2933	2933
Storage controller 1		
Vendor and model	Dell BOSS-S1	Dell BOSS-S1
Cache size (GB)	0	0
Firmware version	2.5.13 3024	2.5.13 3024
Driver version		
Storage controller 2		
Vendor and model	Dell HBA330 Mini (Embedded)	Dell HBA330 Mini (Embedded)
Cache size (GB)	0	0
Firmware version	16.17.01 00	16.17.01 00
Local storage for OS (type A)		
Number of drives	1	1
Drive vendor and model	Intel SSDSCKKB240G8R	Intel SSDSCKKB240G8R

System configuration information	Dell™ PowerEdge™ R640	
	VCF – 4 nodes	OCP – 4 nodes
Drive size (GB)	240	240
Drive information (speed, interface, type)	6Gbps, M2, SSD	6Gbps, M2, SSD
Local storage for distributed storage (type B)		
Number of drives	4	4
Drive vendor and model	Dell Express Flash NVMe P4500	Dell Express Flash NVMe P4500
Drive size (GB)	4	4
Drive information (speed, interface, type)	8GT/s, PCIe 3.0, NVMe 1.2	8GT/s, PCIe 3.0, NVMe 1.2
Network adapter 1		
Vendor and model	Mellanox ConnectX-4 LX 25GbE SFP	Mellanox ConnectX-4 LX 25GbE SFP
Number and type of ports	2 x 25GbE	2 x 25GbE
Driver version	14.32.20.04	14.32.20.04
Network adapter 2		
Vendor and model	Mellanox ConnectX-5 Ex 100GbE QSFP	Mellanox ConnectX-5 Ex 100GbE QSFP
Number and type of ports	2 x 100GbE	2 x 100GbE
Driver version	16.35.30.06	16.35.30.06
Cooling fans		
Vendor and model	Dell	Dell
Number of cooling fans	8	8
Power supplies		
Vendor and model	Dell 0CMPGM	Dell 0CMPGM
Number of power supplies	2	2
Wattage of each (W)	1100	1100

How we tested

Initial testbed deployment and configuration

Broadcom provided two four-node Dell PowerEdge R640 clusters with identical hardware to perform these tests. Broadcom had fully set up the VCF cluster when we received it, and we validated the hardware configuration. The cluster included VMware vSAN with four P4500 disks per node and cluster networking with two 25Gb connections in a virtual distributed switch with port groups for vMotion, vSAN, VM, and Management traffic. The VCF cluster also had VKS enabled, so the only customization we did was create a VM class with 48 vCPUs and 512 GB of RAM.

We received the OCP cluster after Broadcom had already configured the cluster installation process, and we validated the hardware configuration. This cluster contained a shared storage pool across all four nodes using OpenShift Data Foundation and the Local Storage operator, so we did not need to perform any infrastructure modifications.

Testing with Kafka

Configuring VKS for Kafka testing

Before we could deploy Kafka on VKS, we needed to add worker nodes to handle the load and connect to those worker nodes. Our steps consisted of the following:

1. Create a context for the new VKS cluster, and use it:

```
vcf context create pt --endpoint [your supervisor IP address] --insecure-skip-tls-verify -u [a
vSphere username with permissions to make changes]
vcf context use pt
```

2. Create the new VKS cluster:

```
kubectl apply -f kafka-vks.yaml
```

3. Connect to the VKS cluster:

```
vcf context create vks --endpoint [your supervisor IP address] --insecure-skip-tls-verify -u [a
vSphere username with permissions to make changes] --workload-cluster-namespace=kafka --workload-
cluster-name=kafka-cluster-vks
vcf context use vks:kafka-cluster-vks
```

Deploying Kafka

1. Create the namespace for your Kafka cluster, and set it as your default:

```
kubectl create namespace strimzi
kubectl config set-context --current --namespace=strimzi
```

2. Using Helm, deploy the Strimzi Kafka operator:

```
helm install strimzi-cluster-operator oci://quay.io/strimzi-helm/strimzi-kafka-operator
```

3. Deploy the Kafka cluster:

```
kubectl apply -f kafka-cluster.yaml
```

4. Create a client pod with the benchmarking software installed:

```
kubectl apply -f client-pod.yaml
```

Running the benchmark

1. Connect to the Kafka client:

```
kubectl exec -it ubuntu -- /bin/bash
```

2. In the Kafka client, navigate to the scripts directory:

```
cd /opt/kafka_2.13-3.9.1/bin
```

3. Create the topics for the Kafka producer benchmark. If you need to create more than one topic, repeat this step, adding 1 to the topic number each time you create a topic:

```
./kafka-topics.sh --create --bootstrap-server="my-cluster-kafka-bootstrap:9092" --topic "test-
topic-1" --partitions 36 --replication-factor 3 --config retention.ms=10800000 --config min.insync.
replicas=2 --config compression.type="uncompressed"
```

4. Run the Kafka producer benchmark. If you have created more than one topic, repeat this step and immediately run it, adding 1 to the topic number, and output file topic number each time you run the test (you may choose to use a `for` loop to perform this task to ensure simultaneous operation):

```
./kafka-producer-perf-test.sh --topic "test-topic-1" --record-size=1024 --throughput=-1 --num-
records=100000000 --producer-props bootstrap.servers="my-cluster-kafka-bootstrap:9092" acks=1 batch.
size=256000 client.id="producer-1" > /tmp/producer-t1.log 2>&1 &
```

5. When the benchmark is finished, delete the database that the producer benchmark created. If you created more than one topic, repeat this step, adding 1 to the topic number each time you delete a topic:

```
./kafka-topics.sh --delete --bootstrap-server " my-cluster-kafka-bootstrap:9092" --topic
" test-topic-1"
```

6. Record the results in the `/tmp` directory before performing a new run.

Testing with HammerDB TPROC-C on PostgreSQL

Configuring VKS for HammerDB testing

Before we could deploy PostgreSQL on VKS, we needed to add worker nodes to handle the load. Our steps consisted of the following:

1. Create a context for the new VKS cluster, and use it:

```
vcf context create pt --endpoint [your supervisor IP address] --insecure-skip-tls-verify -u [a
vSphere username with permissions to make changes]
vcf context use pt
```

2. Create the new VKS cluster:

```
kubectl apply -f postgres-vks.yaml
```

3. Connect to the VKS cluster:

```
vcf context create vks --endpoint [your supervisor IP address] --insecure-skip-tls-verify -u [a
vSphere username with permissions to make changes] --workload-cluster-namespace=postgres --workload-
cluster-name=postgres-cluster-vks
vcf context use vks:postgres-cluster-vks
```

Deploying PostgreSQL

1. Create a new namespace for PostgreSQL, and set it as default:

```
kubectl create namespace cnpg-system
kubectl config set-context --current --namespace=cnpg-system
```

2. Add the PostgreSQL extensions to your Kubernetes deployment:

```
kubectl apply --server-side -f https://raw.githubusercontent.com/cloudnative-pg/cloudnative-pg/
release-1.27/releases/cnpg-1.27.1.yaml
```

3. Deploy a PostgreSQL cluster:

```
kubectl apply -f psql.yaml
```

4. Copy an optimized PostgreSQL cluster config file to the deployed cluster:

```
for i in `kubectl get pod | grep cluster | awk '{print $1}'`; do kubectl exec -it $i --cp /var/lib/
postgresql/data/pgdata/postgresql.conf /var/lib/postgresql/data/pgdata/postgresql.conf.bak; kubectl
cp postgresql.conf ${i}:/var/lib/postgresql/data/pgdata/; done
```

5. Restart the PostgreSQL cluster nodes one at a time.
6. Create the HammerDB client:

```
kubectl apply -f hammer.yaml
```

Running the benchmark

You can find the TPROC-C scripts we used in the appendix.

1. Open a connection the HammerDB client pod:

```
kubectl exec -it hammerdb-cli-pod -- /bin/bash
```

2. Create a new TPROC-C database:

```
./hammerdbcli tcl auto.create-db.tcl
```

3. Run the TPROC-C benchmark

```
./hammerdbcli tcl auto.run-tprocc.tcl
```

4. Record the results.
5. Delete the TPROC-C database:

```
./hammerdbcli tcl auto drop-db.tcl
```

Appendix: Scripts we used

Kafka

kafka-vks.yaml

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: modern-app-vks
  namespace: modern-app
spec:
  clusterNetwork:
    services:
      cidrBlocks: ["10.96.0.0/12"]
    pods:
      cidrBlocks: ["192.168.0.0/16"]
    serviceDomain: "cluster.local"
  topology:
    class: builtin-generic-v3.4.0
    version: v1.33.1---vmware.1-fips-vkr.2
    controlPlane:
      replicas: 1
      metadata:
        annotations:
          run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
    workers:
      machineDeployments:
        - class: node-pool
          name: node-pool-1
          replicas: 1
          variables:
            overrides:
            - name: vmClass
              value: kafka-48-512
            - name: node
              value:
                labels:
                  topology.kubernetes.io/region: rack1
          metadata:
            annotations:
              run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
        - class: node-pool
          name: node-pool-2
          replicas: 1
          variables:
            overrides:
            - name: vmClass
              value: kafka-48-512
            - name: node
              value:
                labels:
                  topology.kubernetes.io/region: rack2
          metadata:
            annotations:
              run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
        - class: node-pool
          name: node-pool-3
          replicas: 1
          variables:
            overrides:
            - name: vmClass
              value: kafka-48-512
            - name: node
              value:
                labels:
                  topology.kubernetes.io/region: rack3
          metadata:
            annotations:
```

```
            run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
      - class: node-pool
        name: node-pool-4
        replicas: 1
        variables:
          overrides:
          - name: vmClass
            value: vks-48-512
          - name: node
            value:
              labels:
                topology.kubernetes.io/region: rack4
        metadata:
          annotations:
            run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
  variables:
    - name: vmClass
      value: best-effort-medium
    - name: storageClass
      value: vsan-esa-default-policy-raid5
    - name: vsphereOptions
      value:
        persistentVolumes:
          defaultStorageClass: vsan-esa-default-policy-raid5
    - name: kubernetes
      value:
        security:
          podSecurityStandard:
            enforce: privileged
```

kafka-cluster.yaml

```
apiVersion: kafka.strimzi.io/v1beta2
kind: KafkaNodePool
metadata:
  name: dual-role
  labels:
    strimzi.io/cluster: my-cluster
spec:
  replicas: 12
  roles:
    - controller
    - broker
  storage:
    type: jbod
    volumes:
      - id: 0
        type: persistent-claim
        size: 200Gi
        deleteClaim: true
        kraftMetadata: shared
      - id: 1
        type: persistent-claim
        size: 200Gi
        deleteClaim: true
        kraftMetadata: shared
      - id: 2
        type: persistent-claim
        size: 200Gi
        deleteClaim: true
        kraftMetadata: shared
      - id: 3
        type: persistent-claim
        size: 200Gi
        deleteClaim: true
        kraftMetadata: shared
      - id: 4
        type: persistent-claim
        size: 200Gi
        deleteClaim: true
```

```
        kraftMetadata: shared
      - id: 5
        type: persistent-claim
        size: 200Gi
        deleteClaim: true
        kraftMetadata: shared
---

apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    rack:
      topologyKey: topology.kubernetes.io/region
    version: 4.1.0
    metadataVersion: 4.1-IV1
    listeners:
      - name: plain
        port: 9092
        type: internal
        tls: false
      - name: tls
        port: 9093
        type: internal
        tls: true
    config:
      offsets.topic.replication.factor: 3
      transaction.state.log.replication.factor: 3
      transaction.state.log.min.isr: 2
      default.replication.factor: 3
      min.insync.replicas: 2
  entityOperator:
    topicOperator: {}
    userOperator: {}
```

client-pod.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ubuntu-opt-pvc
spec:
# storageClassName: ""
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu
  labels:
    app: ubuntu
spec:
  volumes:
    - name: ubuntu-opt-vol
      persistentVolumeClaim:
        claimName: ubuntu-opt-pvc
  containers:
    - name: ubuntu
      image: ubuntu
      volumeMounts:
        - name: ubuntu-opt-vol
          mountPath: "/opt"
      resources:
        limits:
```

```
          memory: "32000Mi"
          cpu: "24000m"
        requests:
          memory: "16000Mi"
          cpu: "10000m"
      command: ["/bin/sh","-c"]
      args:
        - apt-get update -y;
          apt-get install kafkacat wget iputils-ping default-jdk -y;
          cd /opt;
          wget https://dlcdn.apache.org/kafka/3.9.1/kafka_2.13-3.9.1.tgz;
          tar -xzvf kafka_2.13-3.9.1.tgz;
          cd kafka_2.13-3.9.1/bin;
          sleep infinity;

  hostNetwork: false
  dnsPolicy: ClusterFirstWithHostNet
```

HammerDB

postgresql-vks.yaml

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: postgres-cluster-vks
  namespace: pt
spec:
  clusterNetwork:
    services:
      cidrBlocks: ["10.96.0.0/12"]
    pods:
      cidrBlocks: ["192.168.0.0/16"]
    serviceDomain: "cluster.local"
  topology:
    class: builtin-generic-v3.4.0
    version: v1.33.1---vmware.1-fips-vkr.2
    controlPlane:
      replicas: 1
      metadata:
        annotations:
          run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
    workers:
      machineDeployments:
        - class: node-pool
          name: node-pool-1
          replicas: 4
          variables:
            overrides:
            - name: vmClass
              value: vks-48-512
            - name: node
              value:
                labels:
                  topology.kubernetes.io/region: rack1
          metadata:
            annotations:
              run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
    variables:
      - name: vmClass
        value: best-effort-medium
      - name: storageClass
        value: vsan-esa-default-policy-raid5
      - name: vsphereOptions
        value:
          persistentVolumes:
            defaultStorageClass: vsan-esa-default-policy-raid5
      - name: kubernetes
        value:
          security:
            podSecurityStandard:
              enforce: privileged
```

psql.yaml

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: cnpg-postgres-cluster
  namespace: cnpg-system
spec:
  instances: 4

  # --- BOOTSTRAP WITH A SUPERUSER PASSWORD ---
  bootstrap:
    initdb:
      # this initializes the cluster and uses the superuser secret below
      database: postgres

  superuserSecret:
    name: cnpg-postgres-superuser

  # --- MAIN DATA VOLUME (PGDATA) ---
  storage:
    size: 400Gi

  # --- WAL STORAGE ---
  walStorage:
    size: 400Gi
```

hammerdb.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: hammerdb-cli-pod
  labels:
    app: hammerdb
spec:
  containers:
  - name: hammerdb-cli
    # Use the official HammerDB Docker image
    image: tpcorg/hammerdb:latest
    # Command and arguments to run your specific benchmark script
    command: ["/bin/bash","-c"]
    args:
      - sleep infinity;
```

postgresql.conf

```
# -----------------------------
# PostgreSQL configuration file
# -----------------------------
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, run "pg_ctl reload", or execute
# "SELECT pg_reload_conf()".  Some parameters, which are marked below,
```

```
# require a server shutdown and restart to take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=all".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  B  = bytes            Time units:  us  = microseconds
#                kB = kilobytes                 ms  = milliseconds
#                MB = megabytes                 s   = seconds
#                GB = gigabytes                 min = minutes
#                TB = terabytes                 h   = hours
#                                               d   = days


#------------------------------------------------------------------------------
# FILE LOCATIONS
#------------------------------------------------------------------------------

# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'         # use data in another directory
              # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'    # host-based authentication file
              # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf'        # ident configuration file
              # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                       # write an extra PID file
              # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = 'localhost'              # what IP address(es) to listen on;
              # comma-separated list of addresses;
              # defaults to 'localhost'; use '*' for all
              # (change requires restart)
#port = 5432                           # (change requires restart)
max_connections = 1000                        # (change requires restart)
#reserved_connections = 0            # (change requires restart)
#superuser_reserved_connections = 3  # (change requires restart)
#unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
              # (change requires restart)
#unix_socket_group = ''                       # (change requires restart)
#unix_socket_permissions = 0777              # begin with 0 to use octal notation
              # (change requires restart)
#bonjour = off                                # advertise server via Bonjour
              # (change requires restart)
#bonjour_name = ''                   # defaults to the computer name
              # (change requires restart)

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0             # TCP_KEEPIDLE, in seconds;
              # 0 selects the system default
#tcp_keepalives_interval = 0         # TCP_KEEPINTVL, in seconds;
              # 0 selects the system default
#tcp_keepalives_count = 0            # TCP_KEEPCNT;
              # 0 selects the system default
#tcp_user_timeout = 0                # TCP_USER_TIMEOUT, in milliseconds;
              # 0 selects the system default

#client_connection_check_interval = 0        # time between checks for client
              # disconnection while running queries;
              # 0 for never
```

```
# - Authentication -

#authentication_timeout = 1min              # 1s-600s
#password_encryption = scram-sha-256 # scram-sha-256 or md5
#scram_iterations = 4096
#md5_password_warnings = on
#oauth_validator_libraries = ''      # comma-separated list of trusted validator modules

# GSSAPI using Kerberos
#krb_server_keyfile = 'FILE:${sysconfdir}/krb5.keytab'
#krb_caseins_users = off
#gss_accept_delegation = off

# - SSL -

#ssl = off
#ssl_ca_file = ''
#ssl_cert_file = 'server.crt'
#ssl_crl_file = ''
#ssl_crl_dir = ''
#ssl_key_file = 'server.key'
#ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'    # allowed TLSv1.2 ciphers
#ssl_tls13_ciphers = ''      # allowed TLSv1.3 cipher suites, blank for default
#ssl_prefer_server_ciphers = on
#ssl_groups = 'X25519:prime256v1'
#ssl_min_protocol_version = 'TLSv1.2'
#ssl_max_protocol_version = ''
#ssl_dh_params_file = ''
#ssl_passphrase_command = ''
#ssl_passphrase_command_supports_reload = off


#------------------------------------------------------------------------------
# RESOURCE USAGE (except WAL)
#------------------------------------------------------------------------------

# - Memory -

shared_buffers = 204800MB            # min 128kB
              # (change requires restart)
#huge_pages = try                    # on, off, or try
              # (change requires restart)
#huge_page_size = 0                  # zero for system default
              # (change requires restart)
temp_buffers = 4096MB                # min 800kB
#max_prepared_transactions = 0              # zero disables the feature
              # (change requires restart)
# Caution: it is not advisable to set max_prepared_transactions nonzero unless
# you actively intend to use prepared transactions.
work_mem = 4096MB                    # min 64kB
#hash_mem_multiplier = 2.0           # 1-1000.0 multiplier on hash table work_mem
maintenance_work_mem = 1024MB        # min 64kB
#autovacuum_work_mem = -1            # min 64kB, or -1 to use maintenance_work_mem
#logical_decoding_work_mem = 64MB    # min 64kB
max_stack_depth = 7MB                # min 100kB
#shared_memory_type = mmap           # the default is the first option
              # supported by the operating system:
              #   mmap
              #   sysv
              #   windows
              # (change requires restart)
dynamic_shared_memory_type = posix   # the default is usually the first option
              # supported by the operating system:
              #   posix
              #   sysv
              #   windows
              #   mmap
              # (change requires restart)
#min_dynamic_shared_memory = 0MB     # (change requires restart)
#vacuum_buffer_usage_limit = 2MB     # size of vacuum and analyze buffer access strategy ring;
              # 0 to disable vacuum buffer access strategy;
```

```
              # range 128kB to 16GB

# SLRU buffers (change requires restart)
#commit_timestamp_buffers = 0        # memory for pg_commit_ts (0 = auto)
#multixact_offset_buffers = 16               # memory for pg_multixact/offsets
#multixact_member_buffers = 32               # memory for pg_multixact/members
#notify_buffers = 16                 # memory for pg_notify
#serializable_buffers = 32           # memory for pg_serial
#subtransaction_buffers = 0          # memory for pg_subtrans (0 = auto)
#transaction_buffers = 0             # memory for pg_xact (0 = auto)

# - Disk -

#temp_file_limit = -1                # limits per-process temp file space
              # in kilobytes, or -1 for no limit

#file_copy_method = copy             # copy, clone (if supported by OS)

#max_notify_queue_pages = 1048576    # limits the number of SLRU pages allocated
              # for NOTIFY / LISTEN queue

# - Kernel Resources -

max_files_per_process = 4000          # min 64
              # (change requires restart)

# - Background Writer -

#bgwriter_delay = 200ms                      # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100         # max buffers written/round, 0 disables
#bgwriter_lru_multiplier = 2.0               # 0-10.0 multiplier on buffers scanned/round
#bgwriter_flush_after = 512kB        # measured in pages, 0 disables

# - I/O -

#backend_flush_after = 0             # measured in pages, 0 disables
effective_io_concurrency = 32        # 1-1000; 0 disables issuing multiple simultaneous IO requests
#maintenance_io_concurrency = 16     # 1-1000; same as effective_io_concurrency
#io_max_combine_limit = 128kB        # usually 1-128 blocks (depends on OS)
              # (change requires restart)
#io_combine_limit = 128kB            # usually 1-128 blocks (depends on OS)

#io_method = worker                  # worker, io_uring, sync
              # (change requires restart)
#io_max_concurrency = -1             # Max number of IOs that one process
              # can execute simultaneously
              # -1 sets based on shared_buffers
              # (change requires restart)
#io_workers = 3                              # 1-32;

# - Worker Processes -

#max_worker_processes = 8            # (change requires restart)
#max_parallel_workers_per_gather = 2 # limited by max_parallel_workers
#max_parallel_maintenance_workers = 2        # limited by max_parallel_workers
#max_parallel_workers = 8            # number of max_worker_processes that
              # can be used in parallel operations
#parallel_leader_participation = on


#------------------------------------------------------------------------------
# WRITE-AHEAD LOG
#------------------------------------------------------------------------------

# - Settings -

#wal_level = replica                 # minimal, replica, or logical
              # (change requires restart)
#fsync = on                          # flush data to disk for crash safety
              # (turning this off can cause
              # unrecoverable data corruption)
#synchronous_commit = on             # synchronization level;
```

```
              # off, local, remote_write, remote_apply, or on
#wal_sync_method = fsync             # the default is the first option
              # supported by the operating system:
              #   open_datasync
              #   fdatasync (default on Linux and FreeBSD)
              #   fsync
              #   fsync_writethrough
              #   open_sync
#full_page_writes = on                       # recover from partial page writes
#wal_log_hints = off                 # also do full page writes of non-critical updates
              # (change requires restart)
#wal_compression = off                       # enables compression of full-page writes;
              # off, pglz, lz4, zstd, or on
#wal_init_zero = on                  # zero-fill new WAL files
#wal_recycle = on                    # recycle WAL files
#wal_buffers = -1                    # min 32kB, -1 sets based on shared_buffers
              # (change requires restart)
#wal_writer_delay = 200ms            # 1-10000 milliseconds
#wal_writer_flush_after = 1MB        # measured in pages, 0 disables
#wal_skip_threshold = 2MB

#commit_delay = 0                    # range 0-100000, in microseconds
#commit_siblings = 5                 # range 1-1000

# - Checkpoints -

#checkpoint_timeout = 5min           # range 30s-1d
#checkpoint_completion_target = 0.9  # checkpoint target duration, 0.0 - 1.0
#checkpoint_flush_after = 256kB              # measured in pages, 0 disables
#checkpoint_warning = 30s            # 0 disables
max_wal_size = 5GB
min_wal_size = 80MB

# - Prefetching during recovery -

#recovery_prefetch = try     # prefetch pages referenced in the WAL?
#wal_decode_buffer_size = 512kB      # lookahead window used for prefetching
          # (change requires restart)

# - Archiving -

#archive_mode = off          # enables archiving; off, on, or always
          # (change requires restart)
#archive_library = ''        # library to use to archive a WAL file
          # (empty string indicates archive_command should
          # be used)
#archive_command = ''        # command to use to archive a WAL file
          # placeholders: %p = path of file to archive
          #               %f = file name only
          # e.g. 'test ! -f /mnt/server/archivedir/%f && cp %p /mnt/server/archivedir/%f'
#archive_timeout = 0         # force a WAL file switch after this
          # number of seconds; 0 disables

# - Archive Recovery -

# These are only used in recovery mode.

#restore_command = ''        # command to use to restore an archived WAL file
          # placeholders: %p = path of file to restore
          #               %f = file name only
          # e.g. 'cp /mnt/server/archivedir/%f %p'
#archive_cleanup_command = ''# command to execute at every restartpoint
#recovery_end_command = ''   # command to execute at completion of recovery

# - Recovery Target -

# Set these only when performing a targeted recovery.

#recovery_target = ''        # 'immediate' to end recovery as soon as a
          # consistent state is reached
          # (change requires restart)
#recovery_target_name = ''   # the named restore point to which recovery will proceed
```

```
          # (change requires restart)
#recovery_target_time = ''   # the time stamp up to which recovery will proceed
          # (change requires restart)
#recovery_target_xid = ''    # the transaction ID up to which recovery will proceed
          # (change requires restart)
#recovery_target_lsn = ''    # the WAL LSN up to which recovery will proceed
          # (change requires restart)
#recovery_target_inclusive = on      # Specifies whether to stop:
          # just after the specified recovery target (on)
          # just before the recovery target (off)
          # (change requires restart)
#recovery_target_timeline = 'latest' # 'current', 'latest', or timeline ID
              # (change requires restart)
#recovery_target_action = 'pause'    # 'pause', 'promote', 'shutdown'
              # (change requires restart)

# - WAL Summarization -

#summarize_wal = off         # run WAL summarizer process?
#wal_summary_keep_time = '10d'       # when to remove old summary files, 0 = never


#------------------------------------------------------------------------------
# REPLICATION
#------------------------------------------------------------------------------

# - Sending Servers -

# Set these on the primary and on any standby that will send replication data.

#max_wal_senders = 10        # max number of walsender processes
          # (change requires restart)
#max_replication_slots = 10  # max number of replication slots
          # (change requires restart)
#wal_keep_size = 0           # in megabytes; 0 disables
#max_slot_wal_keep_size = -1 # in megabytes; -1 disables
#idle_replication_slot_timeout = 0   # in seconds; 0 disables
#wal_sender_timeout = 60s    # in milliseconds; 0 disables
#track_commit_timestamp = off# collect timestamp of transaction commit
          # (change requires restart)

# - Primary Server -

# These settings are ignored on a standby server.

#synchronous_standby_names = ''      # standby servers that provide sync rep
          # method to choose sync standbys, number of sync standbys,
          # and comma-separated list of application_name
          # from standby(s); '*' = all
#synchronized_standby_slots = ''     # streaming replication standby server slot
          # names that logical walsender processes will wait for

# - Standby Servers -

# These settings are ignored on a primary server.

#primary_conninfo = ''                      # connection string to sending server
#primary_slot_name = ''                     # replication slot on sending server
#hot_standby = on                   # "off" disallows queries during recovery
              # (change requires restart)
#max_standby_archive_delay = 30s     # max delay before canceling queries
              # when reading WAL from archive;
              # -1 allows indefinite delay
#max_standby_streaming_delay = 30s   # max delay before canceling queries
              # when reading streaming WAL;
              # -1 allows indefinite delay
#wal_receiver_create_temp_slot = off # create temp slot if primary_slot_name
              # is not set
#wal_receiver_status_interval = 10s  # send replies at least this often
              # 0 disables
#hot_standby_feedback = off          # send info from standby to prevent
              # query conflicts
```

```
#wal_receiver_timeout = 60s          # time that receiver waits for
              # communication from primary
              # in milliseconds; 0 disables
#wal_retrieve_retry_interval = 5s    # time to wait before retrying to
              # retrieve WAL after a failed attempt
#recovery_min_apply_delay = 0        # minimum delay for applying changes during recovery
#sync_replication_slots = off        # enables slot synchronization on the physical standby
from the primary

# - Subscribers -

# These settings are ignored on a publisher.

#max_active_replication_origins = 10 # max number of active replication origins
              # (change requires restart)
#max_logical_replication_workers = 4 # taken from max_worker_processes
              # (change requires restart)
#max_sync_workers_per_subscription = 2       # taken from max_logical_replication_workers
#max_parallel_apply_workers_per_subscription = 2     # taken from max_logical_replication_workers


#------------------------------------------------------------------------------
# QUERY TUNING
#------------------------------------------------------------------------------

# - Planner Method Configuration -

#enable_async_append = on
#enable_bitmapscan = on
#enable_gathermerge = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_incremental_sort = on
#enable_indexscan = on
#enable_indexonlyscan = on
#enable_material = on
#enable_memoize = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_parallel_append = on
#enable_parallel_hash = on
#enable_partition_pruning = on
#enable_partitionwise_join = off
#enable_partitionwise_aggregate = off
#enable_presorted_aggregate = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on
#enable_group_by_reordering = on
#enable_distinct_reordering = on
#enable_self_join_elimination = on

# - Planner Cost Constants -

#seq_page_cost = 1.0                 # measured on an arbitrary scale
#random_page_cost = 4.0                      # same scale as above
#cpu_tuple_cost = 0.01                       # same scale as above
#cpu_index_tuple_cost = 0.005        # same scale as above
#cpu_operator_cost = 0.0025          # same scale as above
#parallel_setup_cost = 1000.0        # same scale as above
#parallel_tuple_cost = 0.1           # same scale as above
#min_parallel_table_scan_size = 8MB
#min_parallel_index_scan_size = 512kB
effective_cache_size = 128GB

#jit_above_cost = 100000             # perform JIT compilation if available
              # and query more expensive than this;
              # -1 disables
#jit_inline_above_cost = 500000              # inline small functions if query is
              # more expensive than this; -1 disables
#jit_optimize_above_cost = 500000    # use expensive JIT optimizations if
              # query is more expensive than this;
```

```
              # -1 disables

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5                    # range 1-10
#geqo_pool_size = 0                 # selects default based on effort
#geqo_generations = 0               # selects default based on effort
#geqo_selection_bias = 2.0          # range 1.5-2.0
#geqo_seed = 0.0                    # range 0.0-1.0

# - Other Planner Options -

#default_statistics_target = 100     # range 1-10000
#constraint_exclusion = partition    # on, off, or partition
#cursor_tuple_fraction = 0.1         # range 0.0-1.0
#from_collapse_limit = 8
#jit = on                     # allow JIT compilation
#join_collapse_limit = 8             # 1 disables collapsing of explicit
              # JOIN clauses
#plan_cache_mode = auto                     # auto, force_generic_plan or
              # force_custom_plan
#recursive_worktable_factor = 10.0   # range 0.001-1000000


#------------------------------------------------------------------------------
# REPORTING AND LOGGING
#------------------------------------------------------------------------------

# - Where to Log -

#log_destination = 'stderr'          # Valid values are combinations of
              # stderr, csvlog, jsonlog, syslog, and
              # eventlog, depending on platform.
              # csvlog and jsonlog require
              # logging_collector to be on.

# This is used when logging to stderr:
#logging_collector = off             # Enable capturing of stderr, jsonlog,
              # and csvlog into log files. Required
              # to be on for csvlogs and jsonlogs.
              # (change requires restart)

# These are only used if logging_collector is on:
#log_directory = 'log'                       # directory where log files are written,
              # can be absolute or relative to PGDATA
#log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'     # log file name pattern,
                           # can include strftime() escapes
#log_file_mode = 0600                # creation mode for log files,
              # begin with 0 to use octal notation
#log_rotation_age = 1d                       # Automatic rotation of logfiles will
              # happen after that time.  0 disables.
#log_rotation_size = 10MB            # Automatic rotation of logfiles will
              # happen after that much log output.
              # 0 disables.
#log_truncate_on_rotation = off              # If on, an existing log file with the
              # same name as the new log file will be
              # truncated rather than appended to.
              # But such truncation only occurs on
              # time-driven rotation, not on restarts
              # or size-driven rotation.  Default is
              # off, meaning append to existing files
              # in all cases.

# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'
#syslog_sequence_numbers = on
#syslog_split_messages = on

# This is only relevant when logging to eventlog (Windows):
```

```
# (change requires restart)
#event_source = 'PostgreSQL'

# - When to Log -

#log_min_messages = warning          # values in order of decreasing detail:
               #   debug5
               #   debug4
               #   debug3
               #   debug2
               #   debug1
               #   info
               #   notice
               #   warning
               #   error
               #   log
               #   fatal
               #   panic

#log_min_error_statement = error     # values in order of decreasing detail:
               #   debug5
               #   debug4
               #   debug3
               #   debug2
               #   debug1
               #   info
               #   notice
               #   warning
               #   error
               #   log
               #   fatal
               #   panic (effectively off)

#log_min_duration_statement = -1     # -1 is disabled, 0 logs all statements
               # and their durations, > 0 logs only
               # statements running at least this number
               # of milliseconds

#log_min_duration_sample = -1        # -1 is disabled, 0 logs a sample of statements
               # and their durations, > 0 logs only a sample of
               # statements running at least this number
               # of milliseconds;
               # sample fraction is determined by log_statement_sample_rate

#log_statement_sample_rate = 1.0     # fraction of logged statements exceeding
               # log_min_duration_sample to be logged;
               # 1.0 logs all such statements, 0.0 never logs


#log_transaction_sample_rate = 0.0   # fraction of transactions whose statements
               # are logged regardless of their duration; 1.0 logs all
               # statements from all transactions, 0.0 never logs

#log_startup_progress_interval = 10s # Time between progress updates for
               # long-running startup operations.
               # 0 disables the feature, > 0 indicates
               # the interval in milliseconds.

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_autovacuum_min_duration = 10min # log autovacuum activity;
               # -1 disables, 0 logs all actions and
               # their durations, > 0 logs only
               # actions running at least this number
               # of milliseconds.
#log_checkpoints = on
#log_connections = '' # log aspects of connection setup
          # options include receipt, authentication, authorization,
```

```
          # setup_durations, and all to log all of these aspects
#log_disconnections = off
#log_duration = off # log statement duration
#log_error_verbosity = default              # terse, default, or verbose messages
#log_hostname = off
#log_line_prefix = '%m [%p] '         # special values:
              #   %a = application name
              #   %u = user name
              #   %d = database name
              #   %r = remote host and port
              #   %h = remote host
              #   %L = local address
              #   %b = backend type
              #   %p = process ID
              #   %P = process ID of parallel group leader
              #   %t = timestamp without milliseconds
              #   %m = timestamp with milliseconds
              #   %n = timestamp with milliseconds (as a Unix epoch)
              #   %Q = query ID (0 if none or not computed)
              #   %i = command tag
              #   %e = SQL state
              #   %c = session ID
              #   %l = session line number
              #   %s = session start timestamp
              #   %v = virtual transaction ID
              #   %x = transaction ID (0 if none)
              #   %q = stop here in non-session
              #        processes
              #   %% = '%'
              # e.g. '<%u%%%d> '
#log_lock_waits = off                # log lock waits >= deadlock_timeout
#log_lock_failures = off             # log lock failures
#log_recovery_conflict_waits = off    # log standby recovery conflict waits
              # >= deadlock_timeout
#log_parameter_max_length = -1               # when logging statements, limit logged
              # bind-parameter values to N bytes;
              # -1 means print in full, 0 disables
#log_parameter_max_length_on_error = 0       # when logging an error, limit logged
              # bind-parameter values to N bytes;
              # -1 means print in full, 0 disables
#log_statement = 'none'                      # none, ddl, mod, all
#log_replication_commands = off
#log_temp_files = -1                 # log temporary files equal or larger
              # than the specified size in kilobytes;
              # -1 disables, 0 logs all temp files
log_timezone = 'Etc/UTC'

# - Process Title -

#cluster_name = ''                   # added to process titles if nonempty
              # (change requires restart)
#update_process_title = on


#------------------------------------------------------------------------------
# STATISTICS
#------------------------------------------------------------------------------

# - Cumulative Query and Index Statistics -

#track_activities = on
#track_activity_query_size = 1024    # (change requires restart)
#track_counts = on
#track_cost_delay_timing = off
#track_io_timing = off
#track_wal_io_timing = off
#track_functions = none                      # none, pl, all
#stats_fetch_consistency = cache     # cache, none, snapshot


# - Monitoring -
```

```
#compute_query_id = auto
#log_statement_stats = off
#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off


#------------------------------------------------------------------------------
# VACUUMING
#------------------------------------------------------------------------------

# - Automatic Vacuuming -

#autovacuum = on                     # Enable autovacuum subprocess?  'on'
              # requires track_counts to also be on.
autovacuum_worker_slots = 16 # autovacuum worker slots to allocate
              # (change requires restart)
#autovacuum_max_workers = 3          # max number of autovacuum subprocesses
#autovacuum_naptime = 1min           # time between autovacuum runs
#autovacuum_vacuum_threshold = 50    # min number of row updates before
              # vacuum
#autovacuum_vacuum_insert_threshold = 1000   # min number of row inserts
                     # before vacuum; -1 disables insert
                     # vacuums
#autovacuum_analyze_threshold = 50   # min number of row updates before
              # analyze
#autovacuum_vacuum_scale_factor = 0.2        # fraction of table size before vacuum
#autovacuum_vacuum_insert_scale_factor = 0.2 # fraction of unfrozen pages
          # before insert vacuum
#autovacuum_analyze_scale_factor = 0.1       # fraction of table size before analyze
#autovacuum_vacuum_max_threshold = 100000000    # max number of row updates
                     # before vacuum; -1 disables max
                     # threshold
#autovacuum_freeze_max_age = 200000000       # maximum XID age before forced vacuum
              # (change requires restart)
#autovacuum_multixact_freeze_max_age = 400000000    # maximum multixact age
                           # before forced vacuum
                           # (change requires restart)
#autovacuum_vacuum_cost_delay = 2ms  # default vacuum cost delay for
              # autovacuum, in milliseconds;
              # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1   # default vacuum cost limit for
              # autovacuum, -1 means use
              # vacuum_cost_limit

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0                       # 0-100 milliseconds (0 disables)
#vacuum_cost_page_hit = 1            # 0-10000 credits
#vacuum_cost_page_miss = 2           # 0-10000 credits
#vacuum_cost_page_dirty = 20         # 0-10000 credits
#vacuum_cost_limit = 200             # 1-10000 credits

# - Default Behavior -

#vacuum_truncate = on                # enable truncation after vacuum

# - Freezing -

#vacuum_freeze_table_age = 150000000
#vacuum_freeze_min_age = 50000000
#vacuum_failsafe_age = 1600000000
#vacuum_multixact_freeze_table_age = 150000000
#vacuum_multixact_freeze_min_age = 5000000
#vacuum_multixact_failsafe_age = 1600000000
#vacuum_max_eager_freeze_failure_rate = 0.03 # 0 disables eager scanning

#------------------------------------------------------------------------------
# CLIENT CONNECTION DEFAULTS
#------------------------------------------------------------------------------

# - Statement Behavior -
```

```
#client_min_messages = notice        # values in order of decreasing detail:
              #   debug5
              #   debug4
              #   debug3
              #   debug2
              #   debug1
              #   log
              #   notice
              #   warning
              #   error
#search_path = '"$user", public'     # schema names
#row_security = on
#default_table_access_method = 'heap'
#default_tablespace = ''             # a tablespace name, '' uses the default
#default_toast_compression = 'pglz'  # 'pglz' or 'lz4'
#temp_tablespaces = ''                       # a list of tablespace names, '' uses
              # only default tablespace
#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#default_transaction_deferrable = off
#session_replication_role = 'origin'
#statement_timeout = 0                              # in milliseconds, 0 is disabled
#transaction_timeout = 0                    # in milliseconds, 0 is disabled
#lock_timeout = 0                           # in milliseconds, 0 is disabled
#idle_in_transaction_session_timeout = 0    # in milliseconds, 0 is disabled
#idle_session_timeout = 0                   # in milliseconds, 0 is disabled
#bytea_output = 'hex'                # hex, escape
#xmlbinary = 'base64'
#xmloption = 'content'
#gin_pending_list_limit = 4MB
#createrole_self_grant = ''          # set and/or inherit
#event_triggers = on

# - Locale and Formatting -

datestyle = 'iso, mdy'
#intervalstyle = 'postgres'
timezone = 'Etc/UTC'
#timezone_abbreviations = 'Default'  # Select the set of available time zone
              # abbreviations.  Currently, there are
              #   Default
              #   Australia (historical usage)
              #   India
              # You can create your own file in
              # share/timezonesets/.
#extra_float_digits = 1                     # min -15, max 3; any value >0 actually
              # selects precise output mode
#client_encoding = sql_ascii         # actually, defaults to database
              # encoding

# These settings are initialized by initdb, but they can be changed.
lc_messages = C                             # locale for system error message
              # strings
lc_monetary = C                             # locale for monetary formatting
lc_numeric = C                              # locale for number formatting
lc_time = C                           # locale for time formatting

#icu_validation_level = warning             # report ICU locale validation
              # errors at the given level

# default configuration for text search
default_text_search_config = 'pg_catalog.english'

# - Shared Library Preloading -

#local_preload_libraries = ''
#session_preload_libraries = ''
#shared_preload_libraries = ''               # (change requires restart)
#jit_provider = 'llvmjit'            # JIT library to use
```

```
# - Other Defaults -

#dynamic_library_path = '$libdir'
#extension_control_path = '$system'
#gin_fuzzy_search_limit = 0


#------------------------------------------------------------------------------
# LOCK MANAGEMENT
#------------------------------------------------------------------------------

#deadlock_timeout = 1s
#max_locks_per_transaction = 64              # min 10
              # (change requires restart)
#max_pred_locks_per_transaction = 64 # min 10
              # (change requires restart)
#max_pred_locks_per_relation = -2    # negative values mean
              # (max_pred_locks_per_transaction
              #  / -max_pred_locks_per_relation) - 1
#max_pred_locks_per_page = 2         # min 0


#------------------------------------------------------------------------------
# VERSION AND PLATFORM COMPATIBILITY
#------------------------------------------------------------------------------

# - Previous PostgreSQL Versions -

#array_nulls = on
#backslash_quote = safe_encoding     # on, off, or safe_encoding
#escape_string_warning = on
#lo_compat_privileges = off
#quote_all_identifiers = off
#standard_conforming_strings = on
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equals = off
#allow_alter_system = on


#------------------------------------------------------------------------------
# ERROR HANDLING
#------------------------------------------------------------------------------

#exit_on_error = off                 # terminate session on any error?
#restart_after_crash = on            # reinitialize after backend crash?
#data_sync_retry = off                       # retry or panic on failure to fsync
              # data?
              # (change requires restart)
#recovery_init_sync_method = fsync   # fsync, syncfs (Linux 5.8+)


#------------------------------------------------------------------------------
# CONFIG FILE INCLUDES
#------------------------------------------------------------------------------

# These options allow settings to be loaded from files other than the
# default postgresql.conf.  Note that these are directives, not variable
# assignments, so they can usefully be given more than once.

#include_dir = '...'                 # include files ending in '.conf' from
              # a directory, e.g., 'conf.d'
#include_if_exists = '...'           # include file only if it exists
#include = '...'                     # include file


#------------------------------------------------------------------------------
# CUSTOMIZED OPTIONS
#------------------------------------------------------------------------------
```

```
# Add settings for extensions here

# load CloudNativePG custom.conf configuration
include 'custom.conf'

# load CloudNativePG override.conf configuration
include 'override.conf'
create-db.tcl
dbset db pg
dbset bm TPC-C
diset connection pg_host cnpg-postgres-cluster-rw
diset connection pg_port 5432
diset connection pg_sslmode prefer

diset tpcc pg_count_ware 480
diset tpcc pg_num_vu 48
diset tpcc pg_superuser postgres
diset tpcc pg_defaultdbase postgres
diset tpcc pg_user tpcc
diset tpcc pg_pass tpcc
diset tpcc pg_dbase tpcc
diset tpcc pg_tspace pg_default
diset tpcc pg_partition true

buildschema
```

run-tprocc.tcl

```
set tmpdir $::env(TMP)

dbset db pg
dbset bm TPC-C
diset connection pg_host cnpg-postgres-cluster-rw
diset connection pg_port 5432
diset connection pg_sslmode prefer

diset tpcc pg_superuser postgres
diset tpcc pg_superuserpass postgres
diset tpcc pg_defaultdbase postgres
diset tpcc pg_user tpcc
diset tpcc pg_pass tpcc
diset tpcc pg_dbase tpcc
diset tpcc pg_driver timed
diset tpcc pg_total_iterations 10000000
diset tpcc pg_rampup 5
diset tpcc pg_duration 15
diset tpcc pg_allwarehouse true
diset tpcc pg_timeprofile true
diset tpcc pg_vacuum true

loadscript
puts "TEST STARTED"
vuset vu 96
vuset logtotemp 1
vucreate
tcstart
tcstatus
set jobid [ vurun ]
vudestroy
tcstop
print "TEST COMPLETE"
set of [ open $tmpdir/postgres_tprocc w ]
puts $of $jobid
close $of
```

drop-db.tcl

```
dbset db pg
dbset bm TPC-C
diset connection pg_host cnpg-postgres-cluster-rw
diset connection pg_port 5432
diset connection pg_sslmode prefer

diset tpcc pg_superuser postgres
diset tpcc pg_defaultdbase postgres
diset tpcc pg_user tpcc
diset tpcc pg_pass tpcc
diset tpcc pg_dbase tpcc
diset tpcc pg_tspace pg_default

deleteschema
```

Read the report ▶

This project was commissioned by Broadcom.