



2020 POSTGRES SQL  
CONFERENCE CHINA

# 第十届PostgreSQL中国技术大会

开源 自研 新机遇

# Pigsty

 2020 POSTGRESQL  
CONFERENCE CHINA

# 第十届PostgreSQL 中国技术大会

开源 自研 新机遇

## 目录 / CONTENTS

图形界面  
监控指标  
供给方案  
开源软件



# Pigsty

Postgres in Graphic STYle

开源免费的大规模PostgreSQL集群监控系统与高可用数据库集群供给方案

冯若航



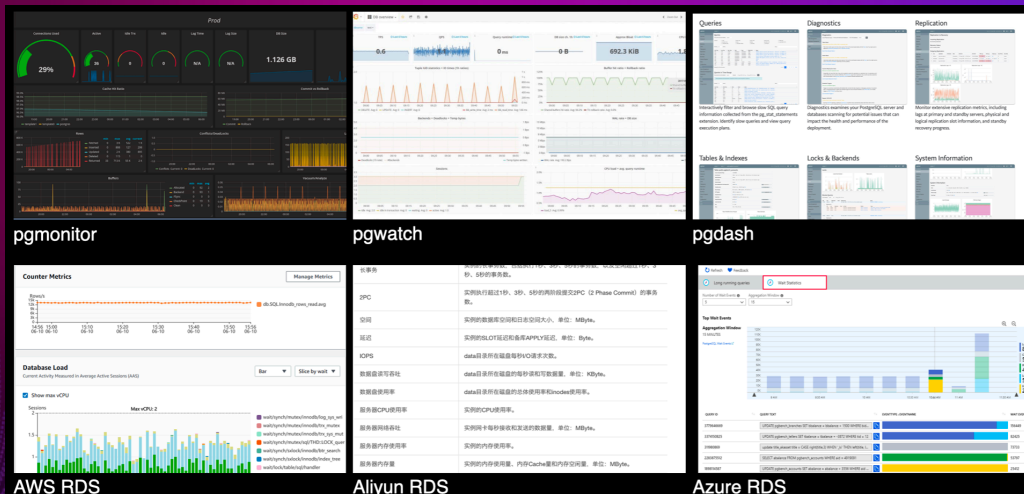
## 前言

- 监控系统 是运维管理之基石
- PostgreSQL很棒，但其生态缺少一个“足够好”的监控
- Pigsty 旨在解决这一问题



# 怎样才算“足够好”？

- GUI美观，指标全面
- 针对大规模集群管理，而非单实例监控。
- 基于开源生态



# PIGSTY

Postgres In Graphic STYle



# Pigsty

PIGSTY is a **monitoring system** for **large scale** Postgres Clusters

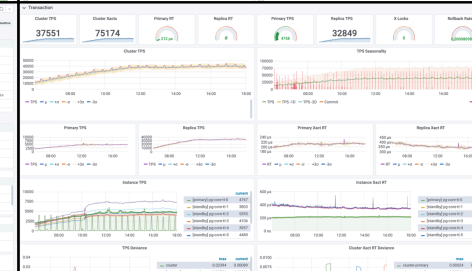
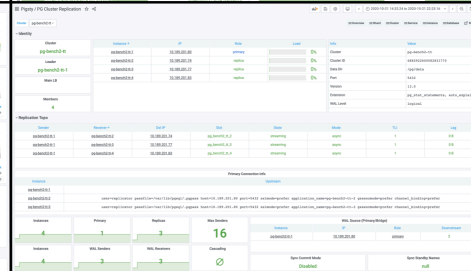
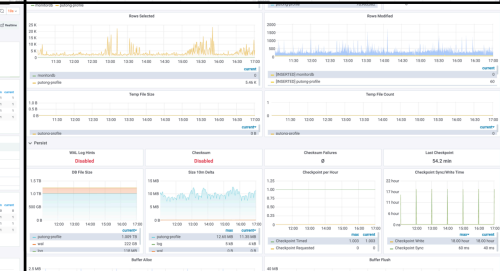
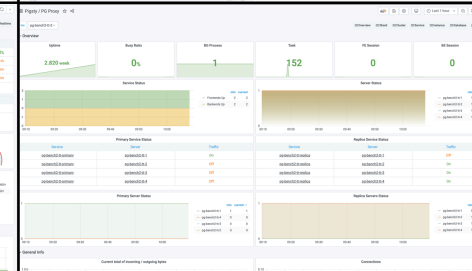
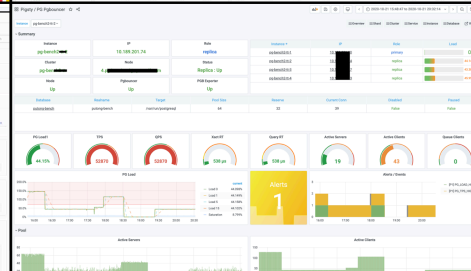
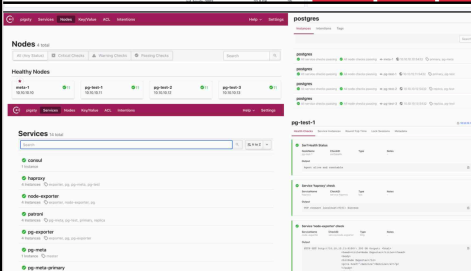
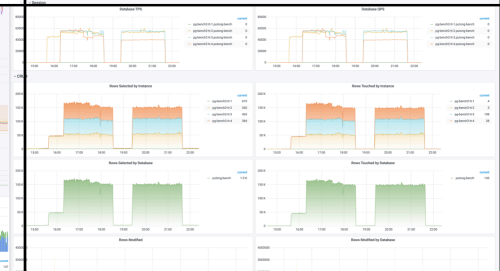
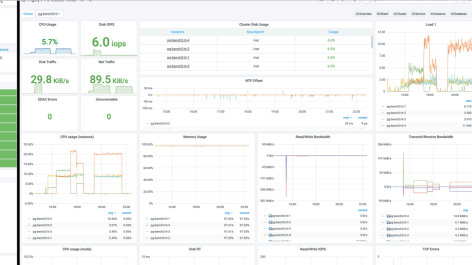
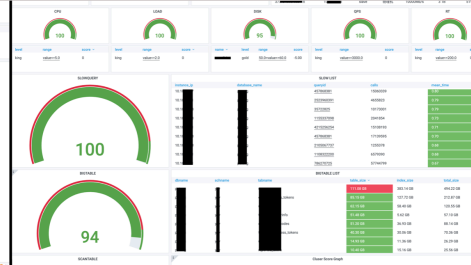
Alone with a **high-available** postgres database provision solution

- Official site: <http://pigsty.cc>
- Public demo: <http://demo.pigsty.cc>
- Local demo: <https://github.com/Vonng/pigsty>

# Graphic Interface









## 界面

了解Pigsty提供的图形化用户界面

Pigsty提供了专业且易用的PostgreSQL监控系统，浓缩了业界监控的最佳实践。您可以方便地进行修改与定制；复用监控基础设施，与其他监控系统相集成。下表是每个监控面板介绍页面的快速导航连接。

全局	集群	服务	实例	数据库
<a href="#">Home</a>	<a href="#">PG Cluster</a>	<a href="#">PG Service</a>	<a href="#">PG Instance</a>	<a href="#">PG Database</a>
<a href="#">PG Overview</a>	<a href="#">PG Cluster Replication</a>	<a href="#">PG DNS</a>	<a href="#">Node</a>	<a href="#">PG Query</a>
<a href="#">PG Shard</a>	<a href="#">PG Cluster Activity</a>		<a href="#">PG Pgbouncer</a>	<a href="#">PG Catalog</a>
<a href="#">PG Alert</a>	<a href="#">PG Cluster Session</a>		<a href="#">PG Proxy</a>	<a href="#">PG Table</a>
<a href="#">PG KPI</a>	<a href="#">PG Cluster Node</a>		<a href="#">PG Exporter</a>	<a href="#">PG Table Detail</a>
<a href="#">PG Capacity</a>	<a href="#">PG Cluster Persist</a>		<a href="#">PG Setting</a>	
<a href="#">PG Change</a>	<a href="#">PG Cluster Database</a>		<a href="#">PG Stat Activity</a>	
<a href="#">PG Monitor</a>	<a href="#">PG Cluster Stats</a>		<a href="#">PG Stat Statements</a>	
	<a href="#">PG Cluster Table</a>			
	<a href="#">PG Cluster Table Detail</a>			
	<a href="#">PG Cluster Query</a>			
	<a href="#">PG Cluster Health</a>			
	<a href="#">PG Cluster Log</a>			
	<a href="#">PG Cluster All</a>			

## 全局监控

- [Home](#)
- [PG Overview](#)
- [PG Shard](#)
- [PG Alert](#)
- [PG KPI](#)
- [PG Capacity](#)
- [PG Change](#)
- [PG Monitor](#)

## 集群监控

- [PG Cluster](#)
- [PG Cluster Replication](#)
- [PG Cluster Activity](#)
- [PG Cluster Session](#)
- [PG Cluster Node](#)
- [PG Cluster Persist](#)
- [PG Cluster Database](#)
- [PG Cluster Stat](#)
- [PG Cluster Table](#)
- [PG Cluster Table Detail](#)
- [PG Cluster Query](#)
- [PG Cluster Health](#)
- [PG Cluster Log](#)
- [PG Cluster All](#)

## 服务监控

- [PG Service](#)
- [PG DNS](#)

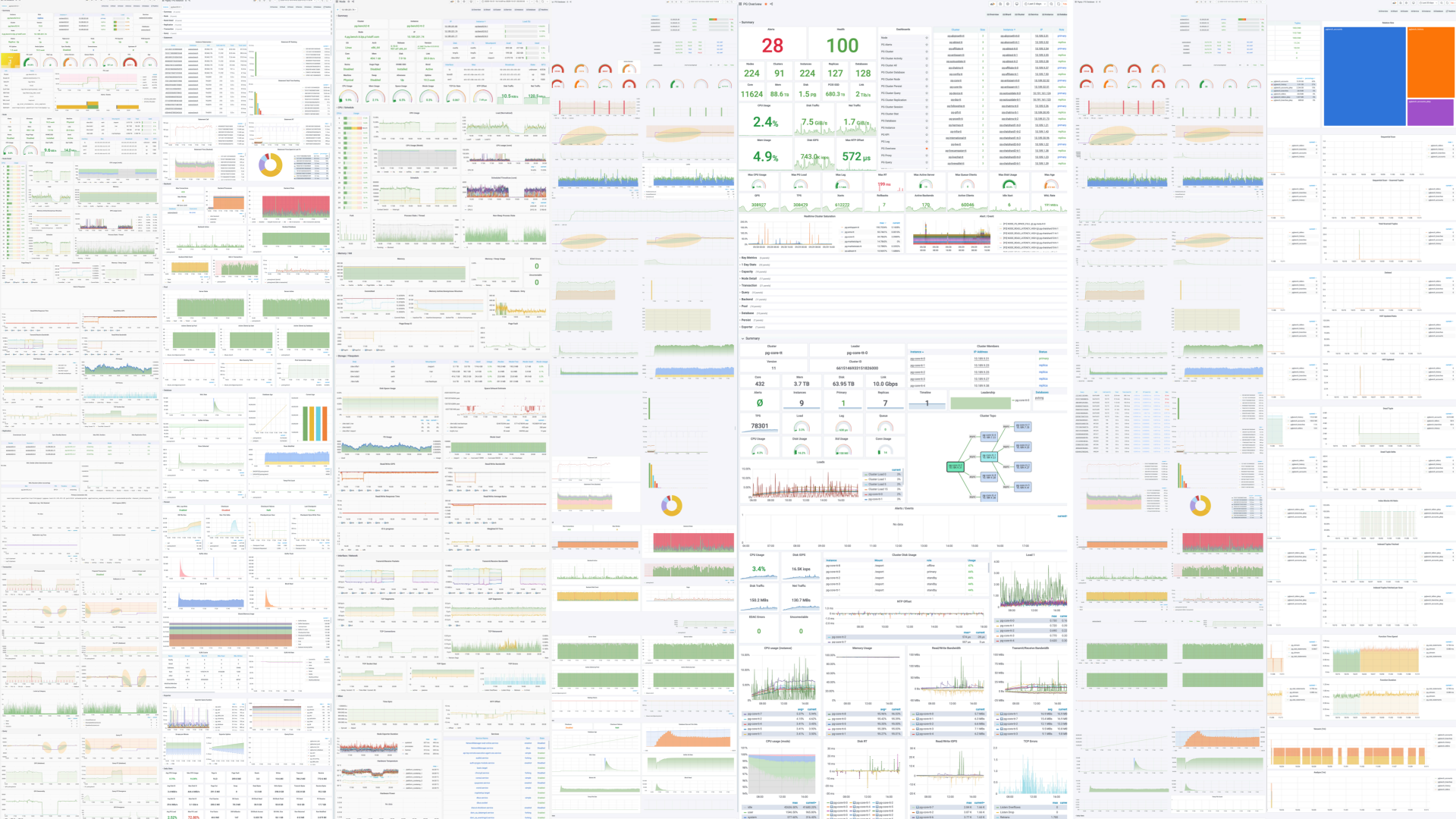
## 实例监控

- [PG Instance](#)
- [Node](#)
- [PG Pgbouncer](#)
- [PG Proxy](#)

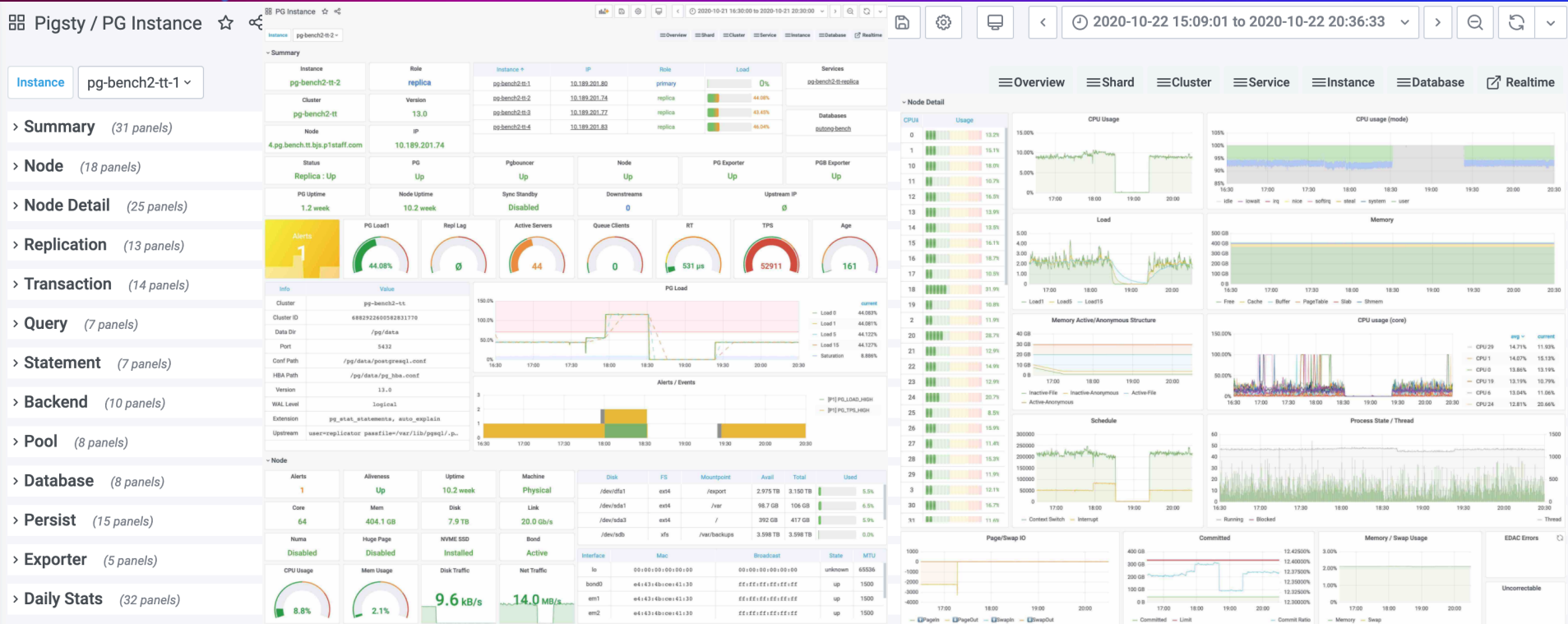
- [PG Exporter](#)
- [PG Setting](#)
- [PG Stat Activity](#)
- [PG Stat Statements](#)

## 数据库监控

- [PG Database](#)
- [PG Pool](#)
- [PG Query](#)
- [PG Table Catalog](#)
- [PG Table](#)
- [PG Table Detail](#)
- [PG Query Detail](#)

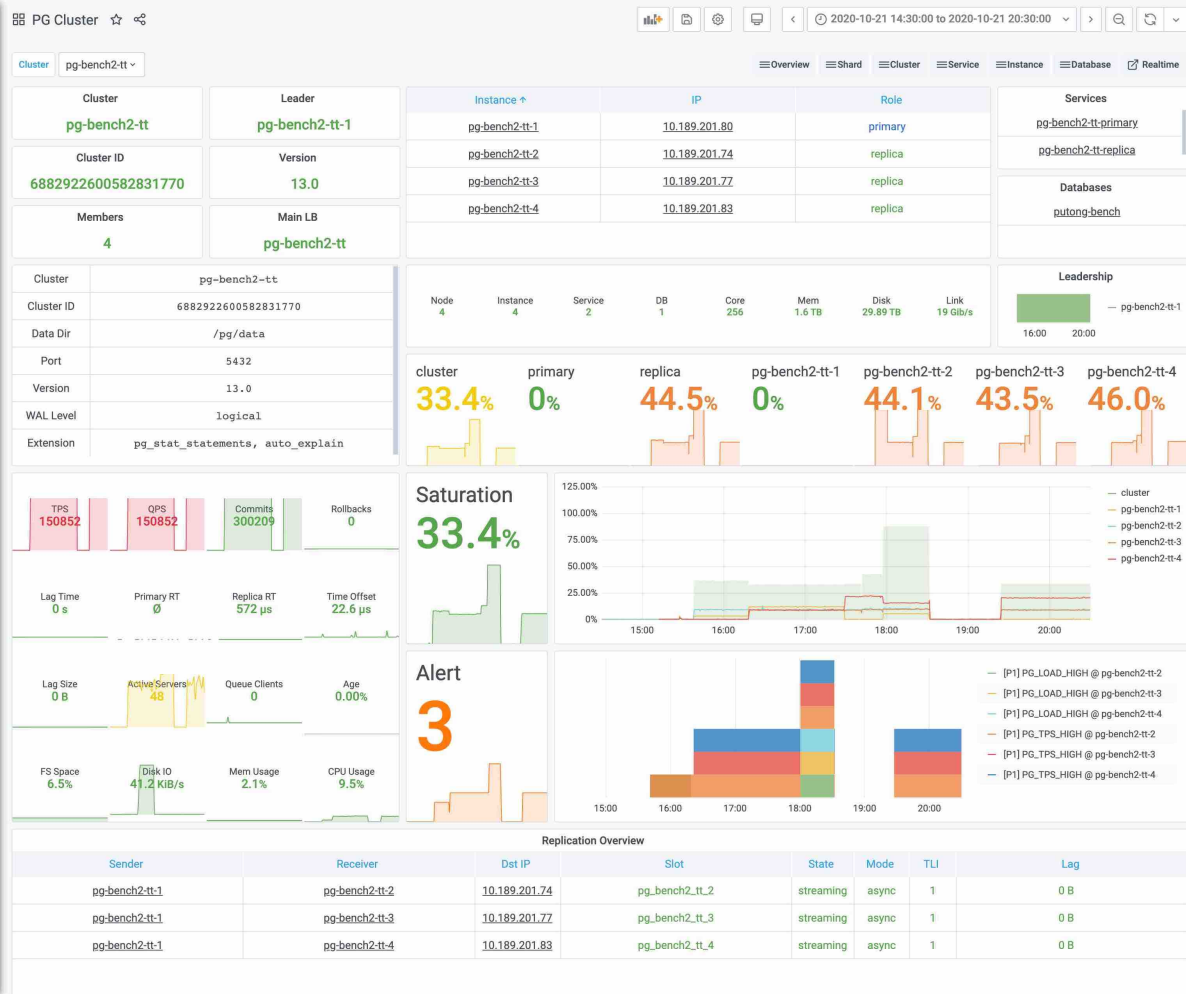


# PG Instance





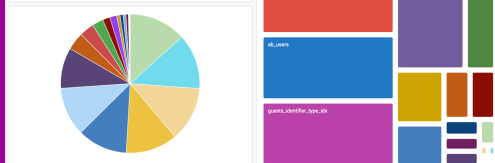
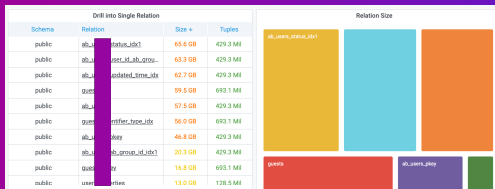
# PG Cluster



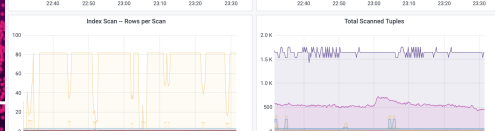
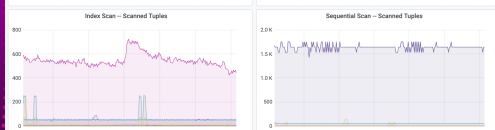
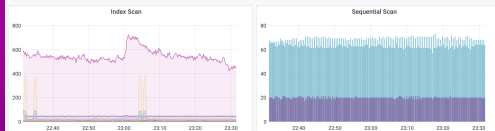


# 2020 POSTGRESQL CONFERENCE CHINA 第十届PostgreSQL中国技术大会

## PG Table / Catalog



-Table Scan



public		isd_station			(r) Table		
Size	Rows	Page	Age	Live Tuple	Dead Tuple	Dead Ratio	Block Ratio
4.538 MB	29.8 K	554	116	29760	0	0%	0%

Identifier	Value	Features	Value	Field	Value	Permissions	Value
Relation OID	19928	Has Index	TRUE	Rel Pages	554	Unlogged	FALSE
Schema OID	2202	Is Partitioned	FALSE	Rel Bytes	4338368	Temporary	FALSE
RelType OID	19940	Is Shared	FALSE	Rel Tuples	29760		
Comptype OID		Has Rule	FALSE	Rel All Visible			
Owner OID	16389	Has Trigger	FALSE	Columns	14		
Access Method ID	2	Has Subclass	FALSE	Checks	0		
Rel FileNode	19928	Row level security	FALSE	FrozenXID	636		
Tablespace OID		Force RLS	FALSE	Age	116		
Toast Table OID	19947	Is Populated	TRUE	Min XID	1		

Tuple	Value	Vacuum	Value	Analyze	Value	Blocks I/O	Value
Seq Scan	9	Vacuums	1	Analyze	1	Heap Blocks Read	538
Seq Scan Tuples	29760	Vacuum Count	0	Analyze Count	0	Heap Blocks Hit	7756
Index Scan	0	Last Vacuum		Last Analyze		Index Blocks Read	484
Index Scan Tuples	0	Since Last Vacuum		Last Analyze Elapse		Index Blocks Hit	422
Inserted Tuples	29760	AutoVacuum Count	1	AutoVacuum Count	1	Toast Blocks Read	0
Deleted Tuples	0	Last AutoVacuum	2020-10-23 15:27:13	Last AutoAnalyze	2020-10-23 15:27:13	Toast Blocks Hit	0
Updated Tuples	0	Since Last AutoVacuum	00:01:18	Last AutoAnalyze Elapse	00:01:18	Toast Index Blocks Read	0
Hot-updated Tuples	0	Inserts since Last Vacuum		Modify Since Analyze		Toast Index Blocks Hit	0

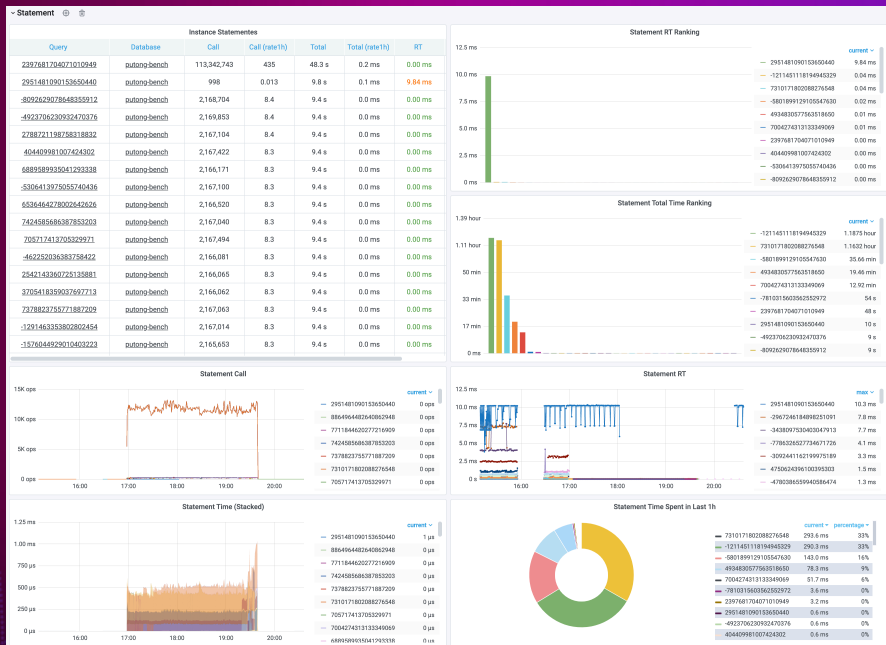
Seq #	Name	Storage	Align	NOT NULL	Null Freq	Avg Width	N Distinct	Correlation
1	station	x	i	true	0	12	-1	1.00
2	useful	x	i	false	0	7	-1	1.00
3	urban	x	i	false	0	6	0	0.48
4	name	x	i	false	0.027	14	-1	0
5	country	x	i	false	0.031	3	252	0.11
6	province	x	i	false	0.77	3	73	0.14
7	loop	x	i	false	0.63	4	0	0.19
8	location	m	d	false	0.040	32	-1	0
9	longitude	m	i	false	0.040	6	-1	0
10	latitude	m	i	false	0.040	6	0	-1
11	elevation	m	i	false	0.057	5	0	0
12	period	x	i	false	0	14	-1	0
13	begin_date	p	i	false	0	4	0	0.15
14	end_date	p	i	false	0	4	0	0.042

Schema	Index	Index ID	Table ID	Scan	Top Rel	Top Freq
public	isd_station_station	19930	19928	0	0	0
public	isd_station_begin_date_idx	19932	19928	0	0	0
public	isd_station_end_date_idx	19933	19928	0	0	0
public	isd_station_loop_idx	19934	19928	0	0	0
public	isd_station_location_idx	19935	19928	0	0	0
public	isd_station_name_idx	19936	19928	0	0	0
public	isd_station_period_idx	19937	19928	0	0	0
public	isd_station_useful_idx	19938	19928	0	0	0



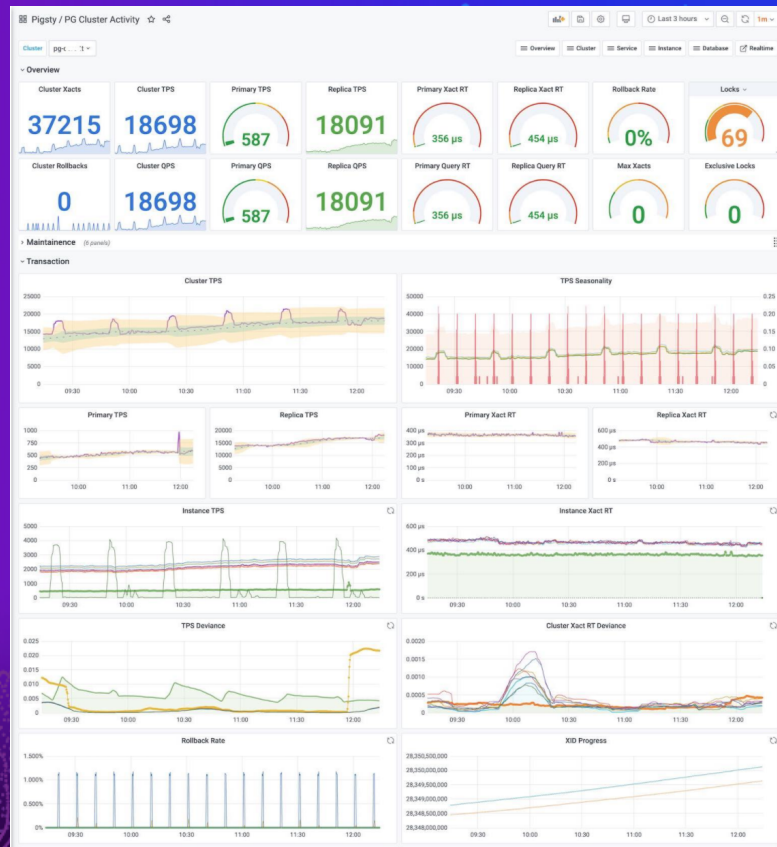


# PG Query





# PG Cluster Activity

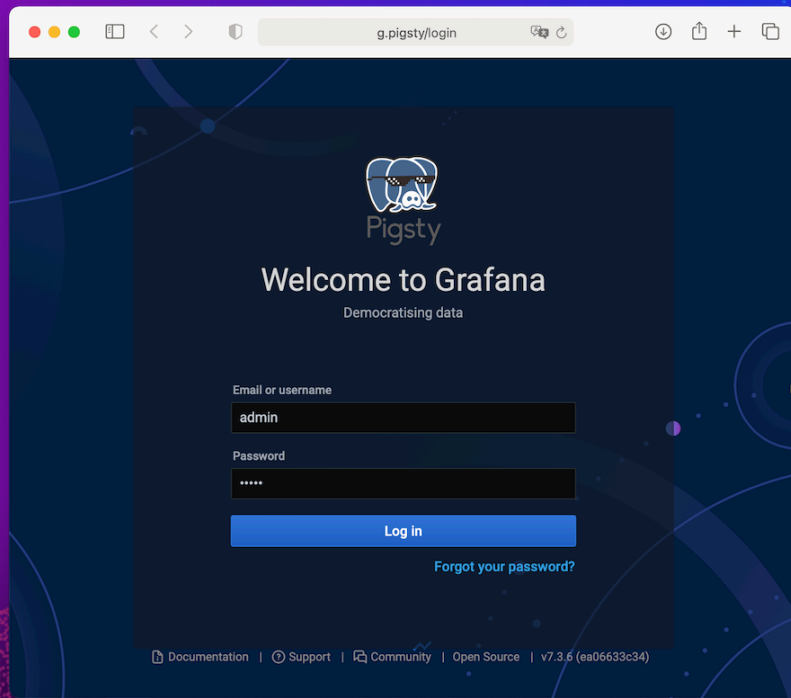






## Magic with Grafana

- 可复用的监控基础设施
- 基于变量、模板与链接的应用。
- 可定制化
- 基于开源生态



**Observability**

# Postgres Observability





# 2020 PostgreSQL Conference China 第十届中国技术大会

## Original Statements Metrics

dbname	query	calls	total_time	min_time	max_time	mean_time	stddev_time	rows	blk_io_time
putong-stats-agent	27853593815658077	897	2219791553.76822	1886110.262859	4187005.307007	2248898.4839448	4180131.879188583	0	112497970.688428
putong-stats-agent	9655319194446369798	987	2802315176.5756	1862686.802313	3779310.911204	2180741.8222695	2489331.458972214	0	562247796.084790
putong-stats-agent	286794518129221452	988	1853998835.46225	1416391.214881	2494846.22823	1888323.3625378	327034.331467058	0	958793983.642424
putong-stats-agent	778086672574368718	986	1715395201.14201	1363330.038133	2349622.065291	1739573.15125964	247663.33413721	0	955213777.949489
putong-stats-agent	728123355708239752	7767	1571991817.00080	88136.793334	7462821.1878084	282338.08353828	189395.074103674	7767	34527.46291
putong-stats-agent	14084395927304243011	986	141485419.59271	1066475.246156	194339.787151	1432540.9593073	175887.35995586	0	44523246.359176
putong-stats-agent	-3201776815681796515	985	895674905.338439	196376.488355	2933731.628959	909314.624780764	514967.071242728	914128454	876488297.628715
putong-stats-agent	761491191392588979104	986	801767117.894274	704162.766505	959125.508862	813151.23187036	35514.897331417	986	421744484.133888
putong-stats-agent	7933336414659317	986	801664711.808802	703995.95454	934999.155301	813948.8593316	38517.887221624	31362	421799358.91217
putong-stats-agent	426885380889595819	7767	798821862.15581	10733.466225	726245.437161	102848.186192328	75208.1481779832	7767	688808369.027243
putong-stats-agent	-7911671141688733992	7767	798569212.000026	10724.117559	726218.38653	102815.657525431	75205.037832809	7767	688808809.043327
putong-stats-agent	-4625841761888621259	986	537464236.658878	53851.8823281	654665.117728	958085.56461945	34147.0387714448	0	731009.133655
putong-stats-agent	-727276793895977284	286787	407833855.436463	3.448677	4086639.889012	142.13692158274	14144.7896816252	640884564	194605337.552312
putong-stats-agent	9851262676909802817	986	376436104.182905	592897.482639	381781.038725956	34433.5889048925	0	5173567.393882	
putong-stats-agent	603824713026709992	1875	364494968.648044	1746.178858	4348843.858623	139475.183274031	79399.972252818	928372	60222736.1798801
putong-stats-agent	19342542407831786	2512	616663173.815349	94336.822455	245445.383386	136277.64736880	13845.728668716	2512	1866.288004
putong-stats-agent	391674248442427011	986	249126980.892766	156042.944292	339651.396615	252664.25898343	38916.964661072	0	135008071.888451
putong-stats-agent	-1318331667078918707	286787	246683918.801301	0.828365	2483.018581	85.880384831242	137.63436694822	640884564	194651277.815887
putong-stats-agent	-3786545497358188956	7767	235137407.696444	3711.668839	213672.599911	30273.929148596	20253.964412588	7767	151457667.993189
putong-stats-agent	-1737317125746653284	7767	235088095.768852	3703.312363	213651.598897	30256.2244878186	20246.035648236	7767	151457682.684574

### Statement SQL

Query	Database	SQL	Call (avg/min)	Total (avg/min)	RT (avg/min)
20316451233323383	putong-stats-agent	select * from ...	314205	0.014	7.34 ms
42518338193838824	putong-stats-agent	select * from ...	143840	0.047	2.80 ms
2521858218882880	putong-stats-agent	select * from ...	143840	0.047	1.70 ms
27083828728812228	putong-stats-agent	select * from ...	142891	0.047	1.20 ms
884388188212228	putong-stats-agent	select * from ...	142891	0.047	1.00 ms
88129207345828393	putong-stats-agent	select * from ...	142891	0.047	0.90 ms
23188612321882882	putong-stats-agent	select * from ...	142891	0.047	0.71 ms
8888820218814650	putong-stats-agent	select * from ...	142891	0.047	0.70 ms
3223388188882128	putong-stats-agent	select * from ...	142891	0.047	0.61 ms
6448221888468664	putong-stats-agent	select * from ...	142891	0.047	0.51 ms
45312321881854542	putong-stats-agent	select * from ...	142891	0.047	15.1 ms
310462218821822739	putong-stats-agent	select * from ...	142891	0.047	15.1 ms
22018818821881888	putong-stats-agent	select * from ...	9113484	0.007	46.4 ms
88282821884538312	putong-stats-agent	select * from ...	2142504	0	9.4 ms
2051218812382927	putong-stats-agent	select * from ...	2142494	0	9.4 ms
2782711887818852	putong-stats-agent	select * from ...	2142104	0	9.4 ms
25218818828828454	putong-stats-agent	select * from ...	2142104	0	9.4 ms

### Statement RT

### Statement Total Time

### Statement RT (Detailed)

### Statement Time Spent in SQL

## Query Statements Dashboards

Instance: pg follow@pharadot1 ->
Instance: putong follow@pharadot1 ->
Query: 2048992070 ->

Home
Node
PG PG Cluster
PG PG Overview
PG PG Start

### Runtime Query

Query ID	mean_time	ms_time	max_time	stddev_time	Calls (Since Reset)	Rows (Since Reset)	Time Spent (Since Reset)	Query ID	Mean Time				
2048992070	3.94 ms	2.07 ms	98.11 ms	3.22 ms	46487937	520460163	4 day	603625462	9.96 ms				
952682592	4.26 ms	3.08 s	1.0246 ms	0.00	4112920	37381162	2 day	284929202	7.87 ms				
60328402	9.26 ms	3.07 ms	98.78 ms	3.03 ms	4452304	89466274	11 hour	140425814	27.7 ms				
166191462	0.58 ms	0.01 ms	39.98 ms	0.00 ms	74213558	74211899	58 min	445858114	166.40 ms				
1348901974	0.85 ms	0.01 ms	1.24 ms	0.00 ms	76741023	76741023	11 min	264619365	160.20 ms				
789728723	7.85 ms	3.82 ms	35.74 ms	6.815 s	48815	5482308	9 min	361672353	132.65 ms				
3201920901	0.82 ms	0.01 ms	13.05 ms	1.01 ms	371864	371864	5 min	4141547220	118.19 ms				
102749546	0.85 ms	0.01 ms	27.62 ms	2.88 ms	329776	329776	5 min	228274621	109.35 ms				
3201762999	0.78 ms	0.01 ms	28.04 ms	3.00 ms	354982	354982	4 min	248977724	97.57 ms				
95275616	0.71 ms	0.01 ms	19.70 ms	3.00 ms	339605	339605	4 min	681515720	68.64 ms				
87138079	0.64 ms	0.01 ms	44.90 ms	1.29 ms	317861	317861	3 min	603867261	65.13 ms				
					1	2	3	4	5	6	7	8	9

### Instant Summary (17 panels)

> Query Overview (4 panels)

> Query Detail

### Query

```

WITH top_monitors AS (
  select *
  from ...
  created_time > $1
  AND
  id NOT IN (SELECT UNNEST($2::bigint[]))
  AND
  user_id NOT IN (SELECT UNNEST($3::int[]))
  AND
  status = $4
  (CASE WHEN $5=5 THEN (gender=$5 OR user_id=$5) ELSE $5 END)
  ORDER BY location -> st_point(310,311) ASC
  limit $12
)
select *
from top_monitors
where user_id = $1;

```

### QPS

### RT Realtime

### Ops

### RT Mean

### RT Stddev

### IO Calls

### IO Rows

### IO Time

### Realtime Response Time

### Avg Rows Returned

### Disk IO Time

### Query RT Stats

### RT Divergence

开原 百研 新机遇



## 某云厂商数据库监控

### 提供的指标列表

17个，其中数据库相关指标约10个

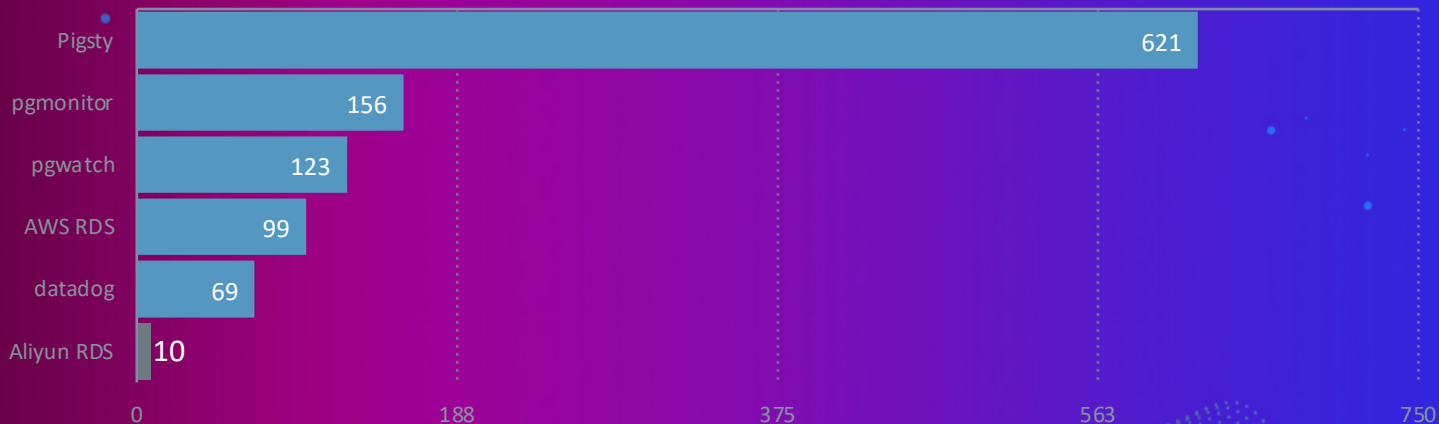
[https://help.aliyun.com/document\\_detail/102748.html](https://help.aliyun.com/document_detail/102748.html)

选择查询时间，即可查看相应的监控数据，具体监控项介绍如下。

监控项	说明
TPS	实例的每秒事务数。
操作行数	实例的每秒UPDATE、INSERT、DELETE等操作的数量。
RT	实例的响应时间，单位：秒。
连接	实例当前的连接总数、活跃连接数和空闲连接数。
膨胀点	实例进行垃圾回收的时间点。
慢查询	实例执行超过1秒、3秒、5秒的SQL数。
长事务	实例的长事务数，包括执行1秒、3秒、5秒的事务数，以及空闲超过1秒、3秒、5秒的事务数。
2PC	实例执行超过1秒、3秒、5秒的两阶段提交2PC（2 Phase Commit）的事务数。
空间	实例的数据库空间和日志空间大小，单位：MByte。
延迟	实例的SLOT延迟和备库APPLY延迟，单位：Byte。
IOPS	data目录所在磁盘每秒I/O请求次数。
数据盘读写吞吐	data目录所在磁盘的每秒读和写数据量，单位：KByte。
数据盘使用率	data目录所在磁盘的总体使用率和inodes使用率。
服务器CPU使用率	实例的CPU使用率。
服务器网络吞吐	实例网卡每秒接收和发送的数据量，单位：MByte。
服务器内存使用率	实例的内存使用率。
服务器内存量	实例的内存使用量、内存Cache量和内存空闲量，单位：MByte。

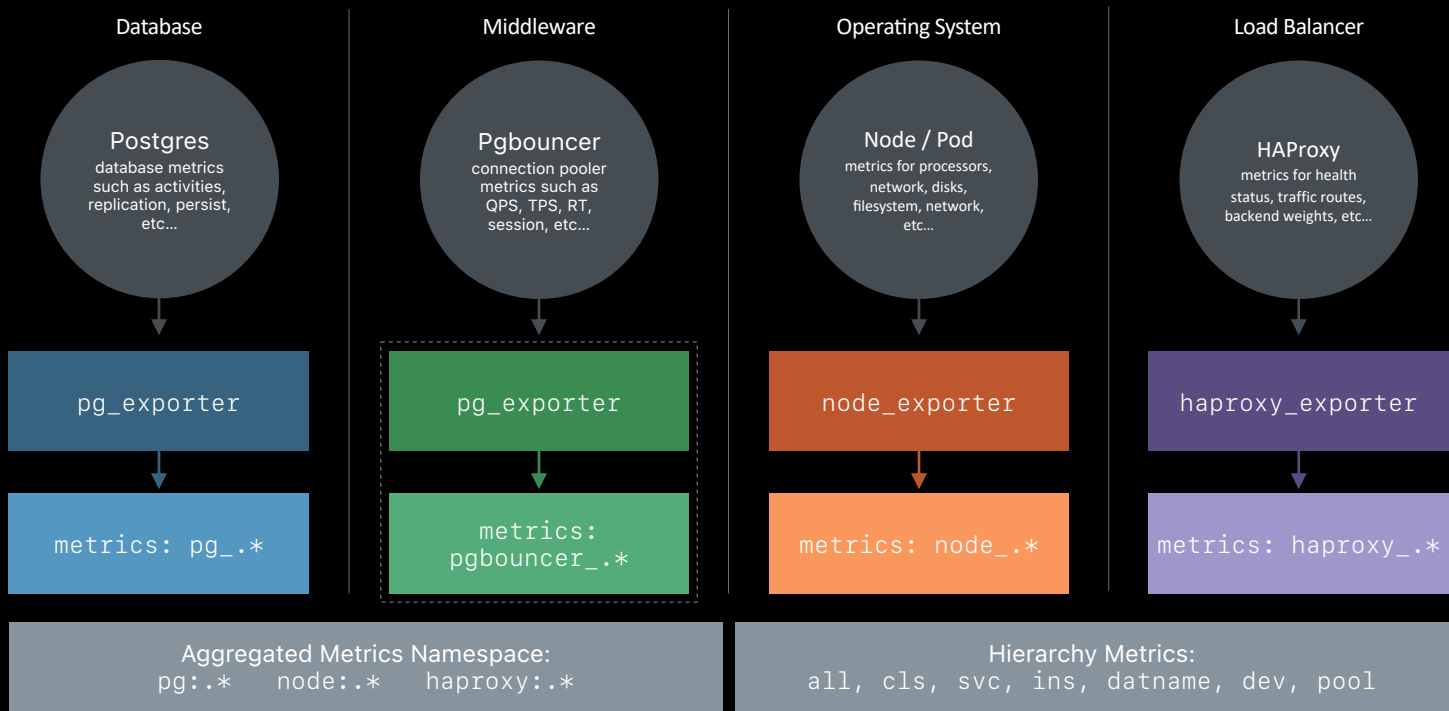


## 指标数量



只计算与数据库相关的指标（即，操作系统等指不计在哪）

# 指标来源



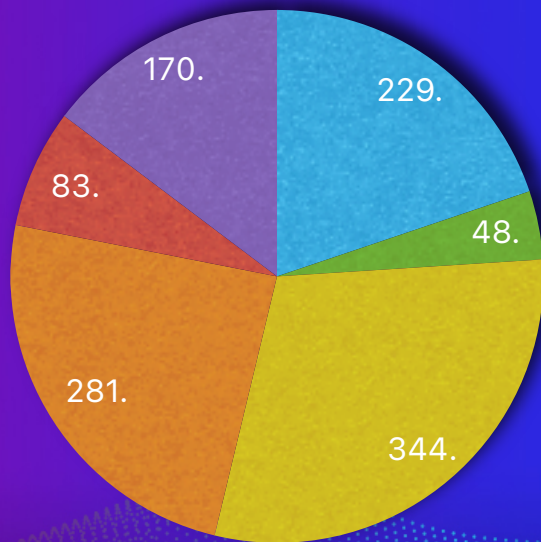


## 指标比例

1155 指标 / 集群

3k~10k+ 时间序列 / 实例

metrics type count



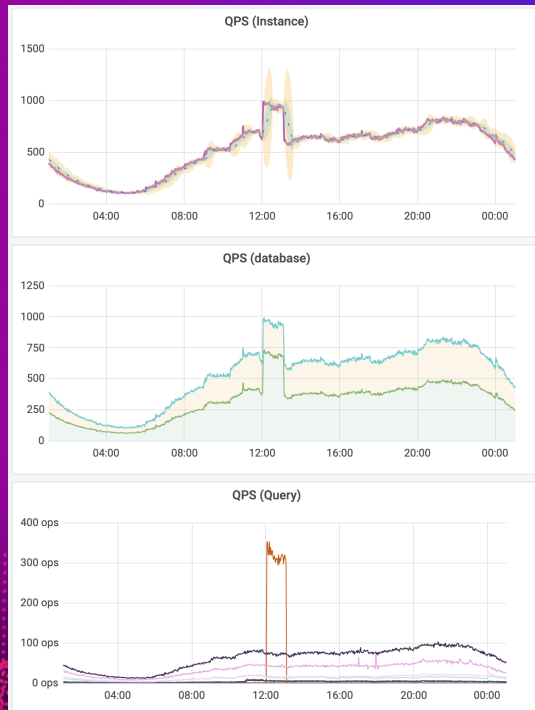
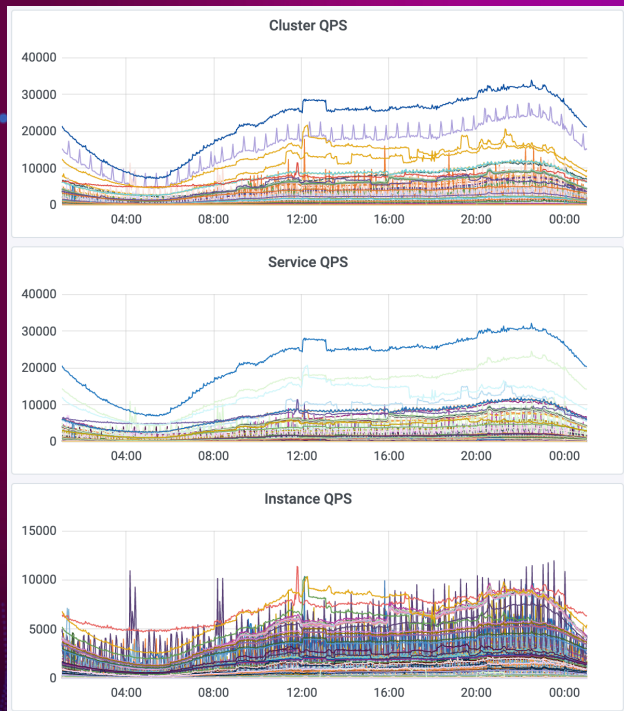
■ pg\_ ■ pgbouncer\_ ■ pg: ■ node: ■ node: ■ haproxy





# 指标层次：衍生指标

QPS：全局，集群，服务，实例，数据库，用户，池，表，查询

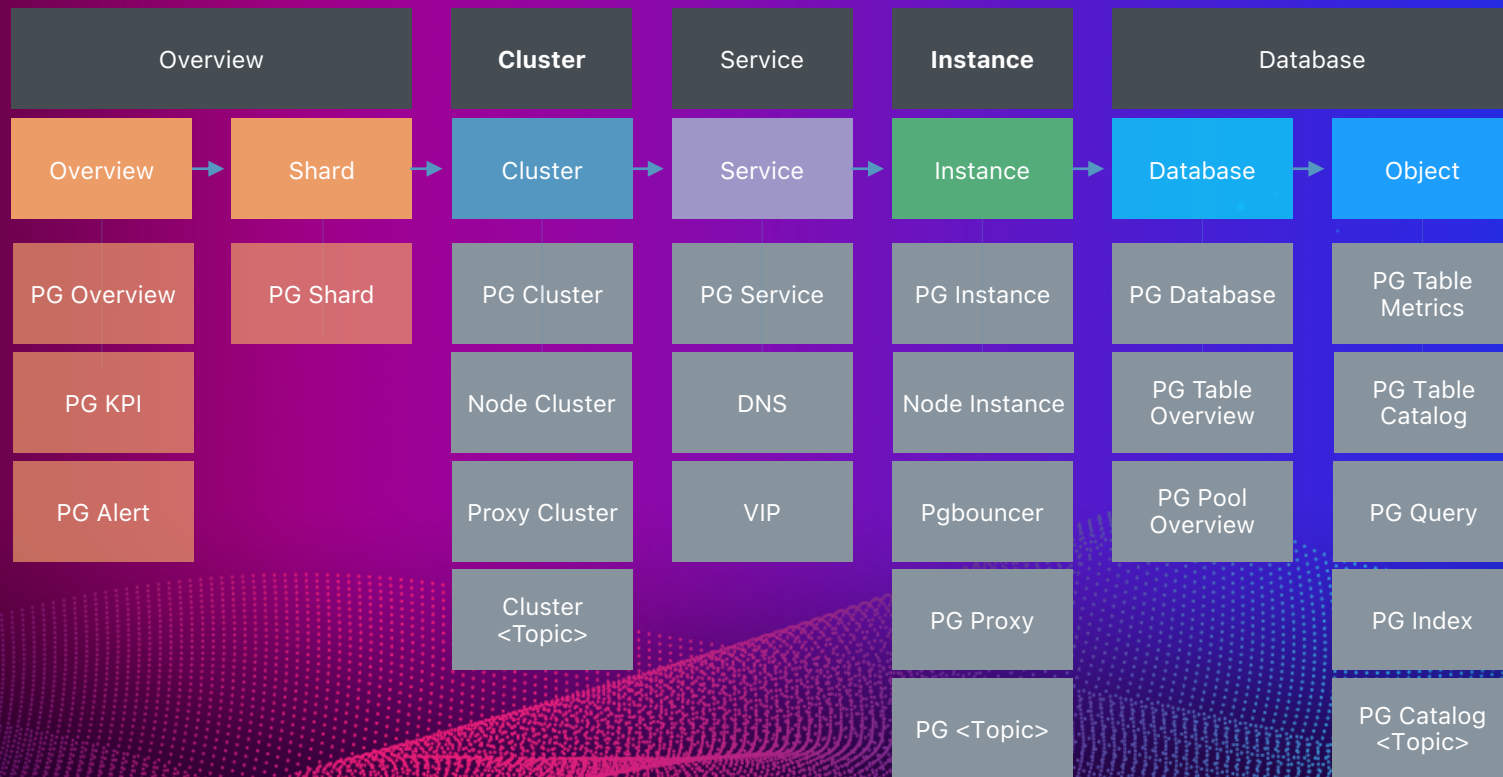


```

#=====
#                               QPS (PgBouncer)                               #
#=====
# QPS realtime (irate1m)
- record: pg:db:qps_realtime
  expr: irate(pgbouncer_stat_total_query_count){1m}
- record: pg:ins:qps_realtime
  expr: sum without(datname) (pg:db:qps_realtime{})
- record: pg:svc:qps_realtime
  expr: sum by(cls, role) (pg:ins:qps_realtime{})
- record: pg:cls:qps_realtime
  expr: sum by(cls) (pg:ins:qps_realtime{})
- record: pg:all:qps_realtime
  expr: sum(pg:cls:qps_realtime{})
# qps (rate1m)
- record: pg:db:qps
  expr: pgbouncer_stat_avg_query_count{datname="pgbouncer"}
- record: pg:ins:qps
  expr: sum without(datname) (pg:db:qps)
- record: pg:svc:qps
  expr: sum by (cls, role) (pg:ins:qps)
- record: pg:cls:qps
  expr: sum by(cls) (pg:ins:qps)
- record: pg:all:qps
  expr: sum(pg:cls:qps)

# qps avg30m
- record: pg:db:qps_avg30m
  expr: avg_over_time(pg:db:qps[30m])
- record: pg:ins:qps_avg30m
  expr: avg_over_time(pg:ins:qps[30m])
- record: pg:svc:qps_avg30m
  expr: avg_over_time(pg:svc:qps[30m])
- record: pg:cls:qps_avg30m
  expr: avg_over_time(pg:cls:qps[30m])
- record: pg:all:qps_avg30m
  expr: avg_over_time(pg:all:qps[30m])

# qps μ
- record: pg:db:qps_mu
  expr: avg_over_time(pg:db:qps_avg30m[30m])
- record: pg:ins:qps_mu
  expr: avg_over_time(pg:ins:qps_avg30m[30m])
- record: pg:svc:qps_mu
  expr: avg_over_time(pg:svc:qps_avg30m[30m])
- record: pg:cls:qps_mu
  expr: avg_over_time(pg:cls:qps_avg30m[30m])
- record: pg:all:qps_mu
  expr: avg_over_time(pg:all:qps_avg30m[30m])
  
```





Overview	→	<code>cls=~'pg-.*'</code>	<code>pg-.*</code>
Shard	→	<code>cls=~'pg-\w+shard.*'</code>	<code>pg-testshard0</code> <code>pg-testshard1</code>
Cluster	→	<code>cls=~'pg-.*'</code>	<code>pg-test</code>
Service	→	<code>cls=~'pg-.*'</code> <code>role=~'(primary replica)'</code> <code>svc='\${cls}-\${role}'</code>	<code>pg-test-primary</code> <code>pg-test-replica</code>
Instance	→	<code>cls=~'pg-.*'</code> <code>ins=~'\${cls}-\d+'</code>	<code>pg-test-0</code> <code>pg-test-1</code>
Database	→	<code>cls=~'pg-.*'</code> <code>ins=~'\${cls}-\d+'</code> <code>datname='.*'</code>	<code>datname=testdb</code>

Pigsty use independent naming system



# 身份管理与命名规则

## Service Template

```

"service": {
  "name": "postgres",
  "port": "{{ pg_port }}",
  "tags": [
    "{{ pg_role }}",
    "{{ pg_cluster }}"
  ],
  "meta": {
    "type": "postgres",
    "role": "{{ pg_role }}",
    "seq": "{{ pg_seq }}",
    "instance": "{{ pg_instance }}",
    "service": "{{ pg_service }}",
    "cluster": "{{ pg_cluster }}",
    "version": "{{ pg_version }}"
  },
  "check": {
    "tcp": "127.0.0.1:{{ pg_port }}",
    "interval": "15s",
    "timeout": "1s"
  }
}

```

## Service

pg-exporter	
Service Name	Node Name
pg-exporter	pg-bench1-tt-1
Health Checks	
Tags & Meta	
Tags	
pg_exporter, pg-exporter	
Meta	
Key	Value
cluster	pg-bench1-tt
instance	pg-bench1-tt-1
role	primary
seq	1
service	pg-bench1-tt-primary
type	exporter

## Service Discovery

### Discovered Labels

```

__address__="10.189.201.76:9099"
__meta_consul_address__="10.189.201.76"
__meta_consul_dc__="tt"
__meta_consul_health__="passing"
__meta_consul_node__="pg-bench1-tt-1"
__meta_consul_service__="node-exporter"
__meta_consul_service_id__="node-exporter"
__meta_consul_service_metadata_cluster__="pg-bench1-tt"
__meta_consul_service_metadata_instances__="pg-bench1-tt-1"
__meta_consul_service_metadata_role__="primary"
__meta_consul_service_metadata_seq__="1"
__meta_consul_service_metadata_service__="pg-bench1-tt-primary"
__meta_consul_service_metadata_type__="exporter"
__meta_consul_service_port__="9099"
__meta_consul_tagged_address_ip__="10.189.201.76"
__meta_consul_tagged_address_ip_v4__="10.189.201.76"
__meta_consul_tagged_address_wan__="10.189.201.76"
__meta_consul_tagged_address_wan_ip_v4__="10.189.201.76"
__meta_consul_tags__="pg_exporter,node-exporter,"
__metrics_path__="/debug/metrics"
__scheme__="http"
job="pg"

```

### Target Labels

```

cls="pg-bench1-tt"
ins="pg-bench1-tt-1"
instance="10.189.201.76:9099"
ip="10.189.201.76"
job="pg"
role="primary"
svc="pg-bench1-tt-primary"

```

## Labels

pg\_up{cls="pg-bench1-tt", ins="pg-bench1-tt-1", instance="10.189.201.76:9185", ip="10.189.201.76", job="pg", role="primary", svc="pg-bench1-tt-primary"}

# Provisioning

```

#-----
# cluster: pg-test
#-----
pg-test: # define cluster named 'pg-test'
# - cluster members - #
hosts:
  10.10.10.11: {pg_seq: 1, pg_role: primary, ansible_host: node-1}
  10.10.10.12: {pg_seq: 2, pg_role: replica, ansible_host: node-2}
  10.10.10.13: {pg_seq: 3, pg_role: replica, ansible_host: node-3}

# - cluster configs - #
vars:
  # basic settings
  pg_cluster: pg-test           # define actual cluster name
  pg_version: 13                # define installed pgsqsl version
  node_tune: tiny               # tune node into oltp/olap/crit/tiny mode
  pg_conf: tiny.yml            # tune pgsqsl into oltp/olap/crit/tiny mode

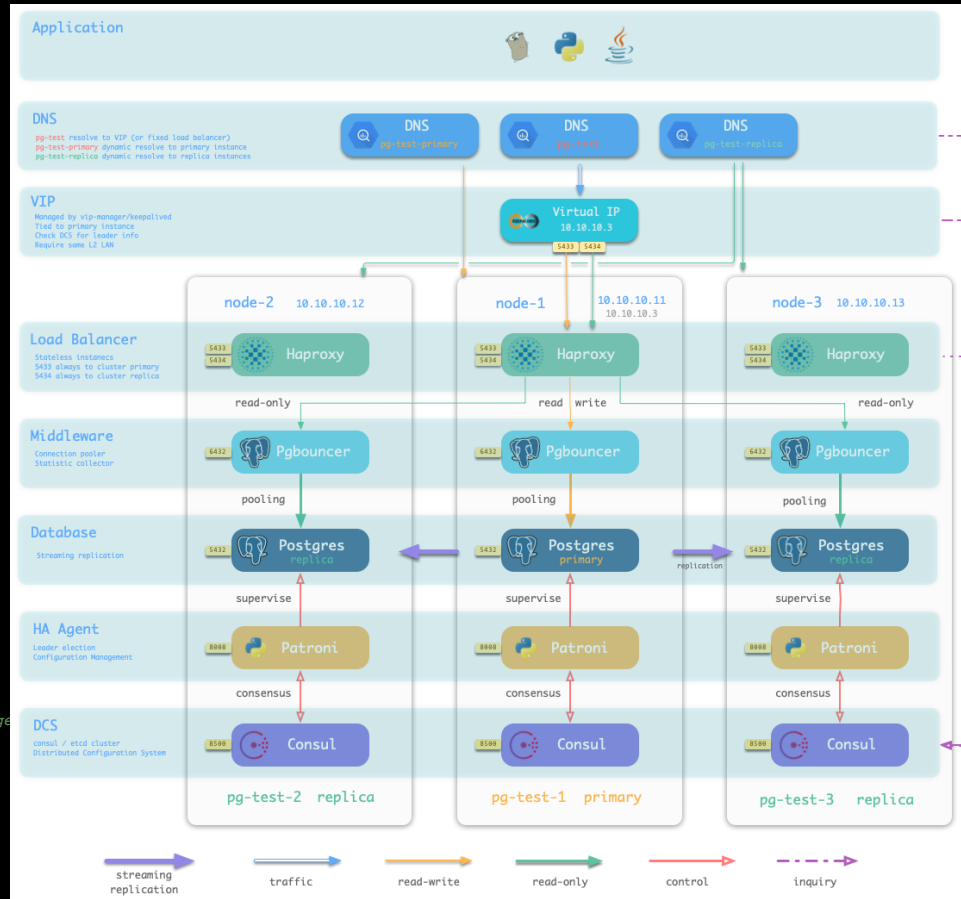
  pg_users:
    - username: dbuser_test
      password: DBUser.Test
      comment: default test user
      groups: [ dbrole_readwrite ] # dbrole_admin|dbrole_readwrite|dbrole_readonly
    - username: dbuser_dba
      password: DBUser.DBA
      comment: default DBA user
      groups: [ dbrole_admin ] # dbrole_admin|dbrole_readwrite|dbrole_readonly

  pg_databases: # create a business database 'test'
    - name: testdb
      extensions: [{name: postgis}] # create extra extension postgis
      parameters: # overwrite database meta's default search_path
        search_path: public,monitor
    - name: testdb2
      extensions: [{name: file_fdw}] # create extra extension postgis

  pg_default_database: testdb # default database will be used as primary monitor target

  # proxy settings
  vip_enabled: true # enable/disable vip (require members in same LAN)
  vip_address: 10.10.10.3 # virtual ip address
  vip_cidrmask: 8 # cidr network mask length
  vip_interface: eth1 # interface to add virtual ip

```







# 演示沙箱

```

#####
# PostgreSQL Environment Inventory
#####
all: # top-level namespace, watch all hosts

#####
# Clusters
#####
clusters: # top-level groups, one group per database cluster (and watch! group 'meta')

# meta control
#
meta: ~2 keys

#####
# clusters: pg-meta
#
pg-meta:

# - cluster members - #
hosts:
  10.10.10.10: {pg_scsi: 1, pg_role: primary, ansible_host: meta}

# - cluster configs - #
vars:
  pg_cluster: pg-meta # define actual cluster name
  pg_version: 13 # define installed postgres version
  node_name: 01p # time node into 01p/04p/02/11/11y mode
  pg_conf: 01p.yml # time used into 01p/04p/02/11/11y mode
  patroni_scs: meta # extra instance name (default:instance)
  patroni_watchdog: off # disable watchdog (require/require/require)
  pg_users: # create a database user named 'dbuser_meta'
    - {username: dbuser_meta, password: dbuser_meta, groups: []}
  pg_databases: # create a business database 'meta'
    - meta: meta # create extra schema named 'meta'
      schemas: [1 item]
      extensions: [1 item] # create extra extension postgres
  parameters: ~1 keys
  pg_default_database: meta # default database will be used as primary monitor target

# proxy settings
via_enabled: true # enable/disable via (require members in same LAN)
via_address: 10.10.10.10 # virtual ip address
via_client_max_size: 0 # client network max length
via_interface: eth0 # interface to add virtual ip

#####
# clusters: pg-test
#
pg-test: # define cluster named 'pg-test'

# - cluster members - #
hosts:
  10.10.10.11: {pg_scsi: 1, pg_role: primary, ansible_host: node-1}
  10.10.10.12: {pg_scsi: 2, pg_role: replica, ansible_host: node-2}
  10.10.10.13: {pg_scsi: 3, pg_role: replica, ansible_host: node-3}

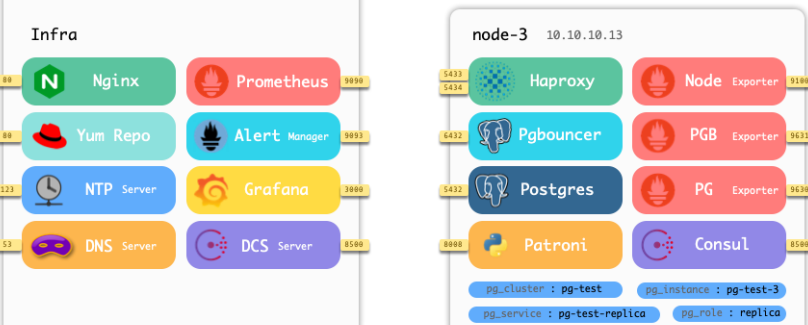
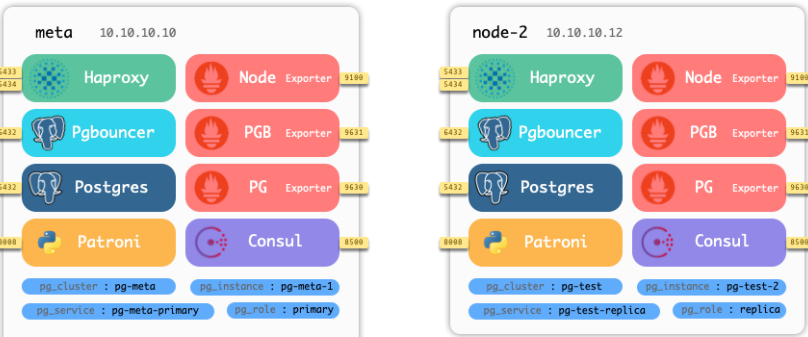
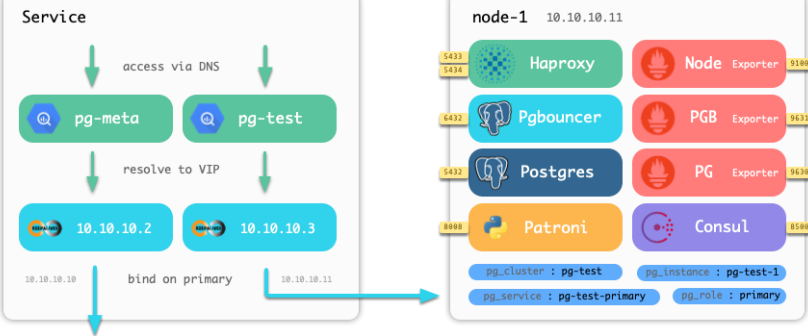
# - cluster configs - #
vars:
# basic settings
  pg_cluster: pg-test # define actual cluster name
  pg_version: 13 # define installed postgres version
  node_name: 01p # time node into 01p/04p/02/11/11y mode
  pg_conf: 01p.yml # time used into 01p/04p/02/11/11y mode

  pg_users: ~1 keys
  pg_databases: ~1 keys # default database will be used as primary monitor target

# proxy settings
via_enabled: true # enable/disable via (require members in same LAN)
via_address: 10.10.10.10 # virtual ip address
via_client_max_size: 0 # client network max length
via_interface: eth0 # interface to add virtual ip

#####
#
#
#####
vars: ~100 keys

```



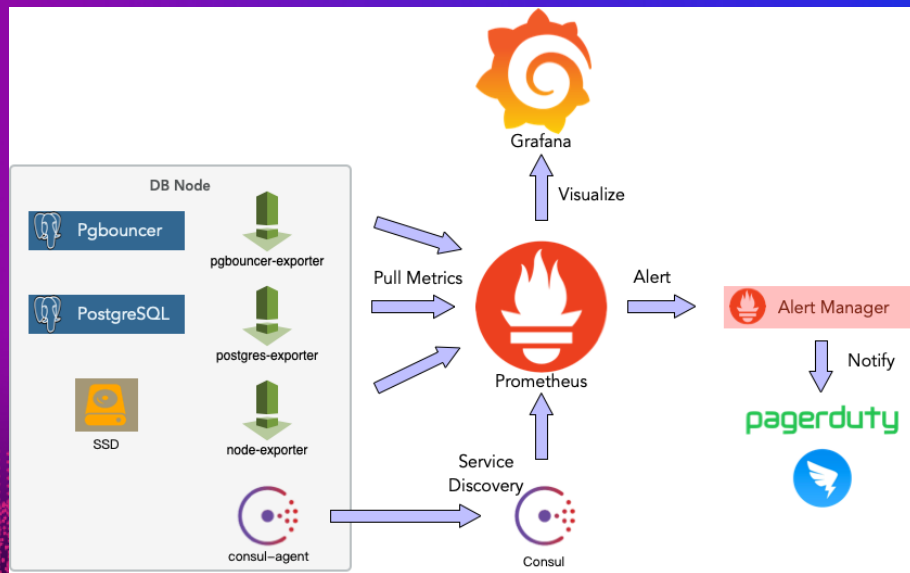


**Open Source**

# Pigsty loves open source

## embrace the open source monitoring stack

- Grafana
- Prometheus
- Alertmanager
- Consul / etcd
- PostgreSQL
- pg\_exporter





2020 POSTGRESQL CONFERENCE CHINA  
第十届PostgreSQL中国技术大会

# THANKS

谢谢观看

<http://pigsty.cc>