Managing Your YugabyteDB Cluster Using yugabyted

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB

Agenda

- yugabyted overview
- Provisioning your first YugabyteDB instance
- Support for multi-node topologies using yugabyted
- UI for monitoring the YugabyteDB cluster
- Demos
  - Getting started with YugabyteDB local cluster
  - Deploying **multi-zone** cluster using yugabyted
  - Deploying **multi-region** cluster using yugabyted

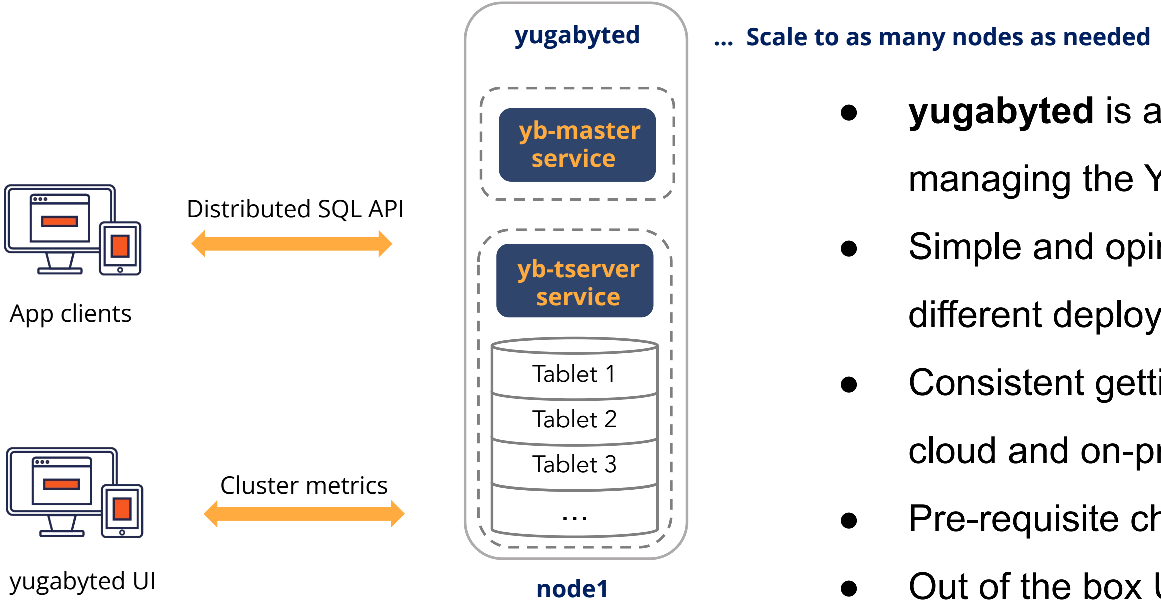
Getting started with YugabyteDB

YugabyteDB Managed *Fully Managed DBaaS*	**YugabyteDB Anywhere** *Self-Managed DBaaS*	**YugabyteDB** *Distributed SQL DB*
Database in **Yugabyte's** public cloud infrastructure	Database in **your** public, private, or hybrid infrastructure	Database in **your** public, private, or hybrid infrastructure
Fully managed infra and database operations - *handled by Yugabyte*	Automation software for infra and database day 2 operations - *self-managed*	Manual infrastructure and database day 2 operations
24/7 enterprise support and services	24/7 enterprise support and advanced services	Community support

yugabyted

Easy Way to Start, Configure and Manage Your YugabyteDB Cluster

yugabyted overview



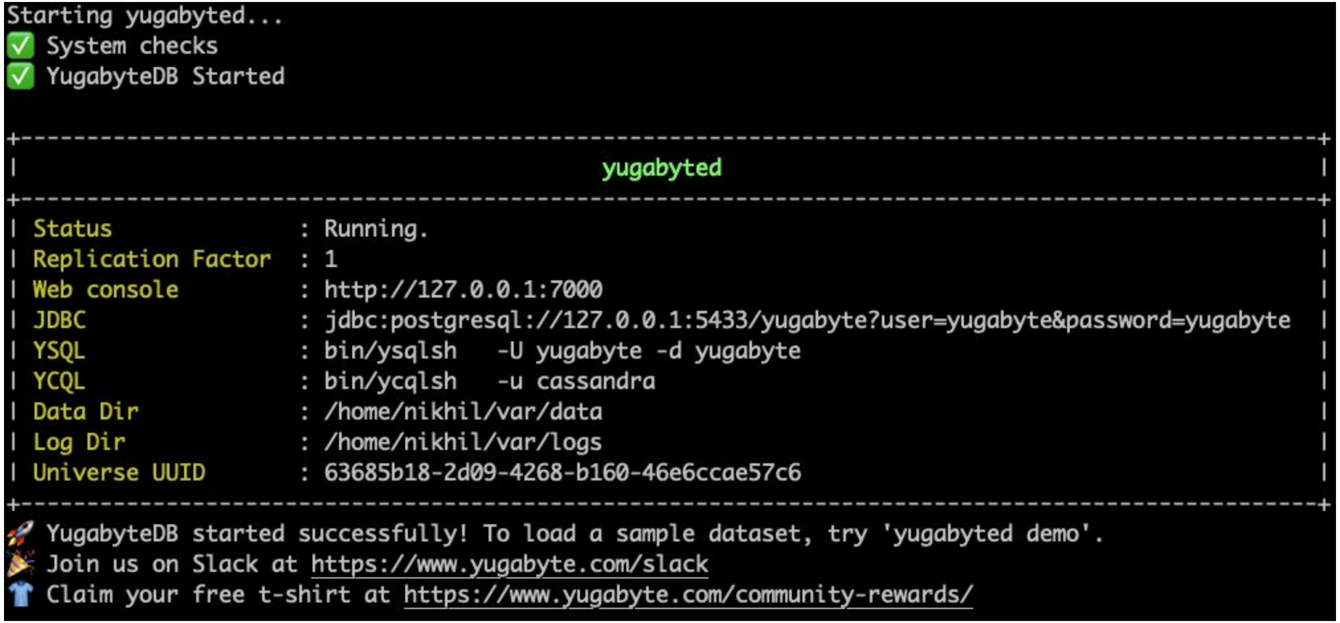
yb-tserver service Store & serve app data in/from tablets (aka shards)

yb-master service Manage tablet/shrd metadata & coordinate config changes

- **yugabyted** is a daemon service for starting and managing the YugabyteDB cluster.
- Simple and opinionated approach for launching different deployment topologies.
- Consistent getting started experience across cloud and on-prem environments.
- Pre-requisite checks run based on the OS.
- Out of the box UI for managing the cluster.
- Currently yugabyted service is in **preview**. GA soon.

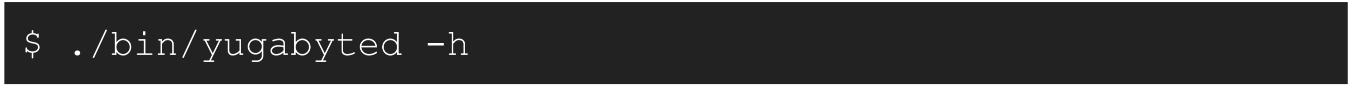
Starting your YugabyteDB local instance

```
$ ./bin/yugabyted start
```

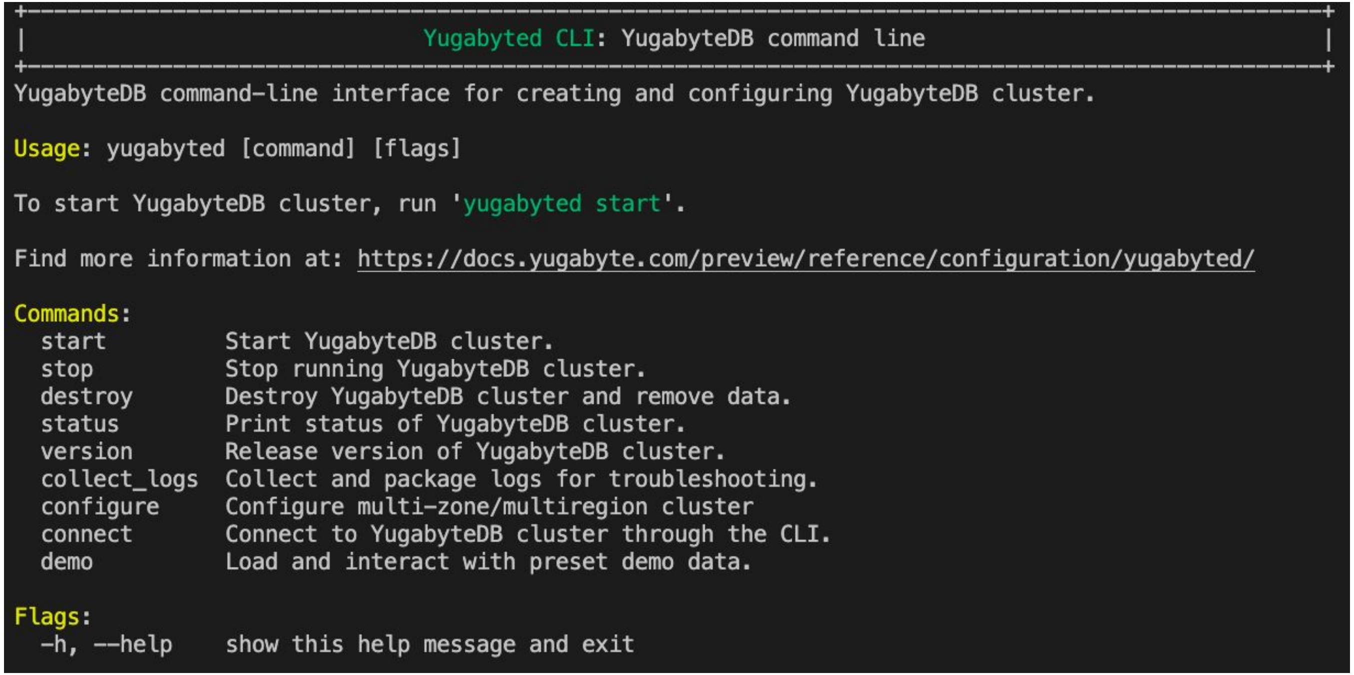


```
Starting yugabyted...
✅ System checks
✅ YugabyteDB Started

+----------------------------------------------------------------------------------------------------+
|                                             yugabyted                                              |
+----------------------------------------------------------------------------------------------------+
| Status              : Running.                                                                     |
| Replication Factor  : 1                                                                            |
| Web console         : http://127.0.0.1:7000                                                        |
| JDBC                : jdbc:postgresql://127.0.0.1:5433/yugabyte?user=yugabyte&password=yugabyte  |
| YSQL                : bin/ysqlsh   -U yugabyte -d yugabyte                                         |
| YCQL                : bin/ycqlsh   -u cassandra                                                    |
| Data Dir            : /home/nikhil/var/data                                                        |
| Log Dir             : /home/nikhil/var/logs                                                        |
| Universe UUID       : 63685b18-2d09-4268-b160-46e6ccae57c6                                         |
+----------------------------------------------------------------------------------------------------+
🚀 YugabyteDB started successfully! To load a sample dataset, try 'yugabyted demo'.
🎉 Join us on Slack at https://www.yugabyte.com/slack
👕 Claim your free t-shirt at https://www.yugabyte.com/community-rewards/
```

Commands supported by yugabyted



```
$ ./bin/yugabyted -h
```



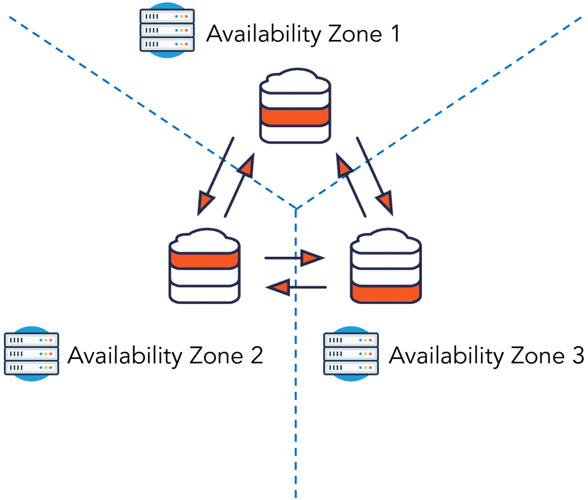
```
+-----------------------------------------------------------------------------------------------+
|                            Yugabyted CLI: YugabyteDB command line                             |
+-----------------------------------------------------------------------------------------------+
YugabyteDB command-line interface for creating and configuring YugabyteDB cluster.

Usage: yugabyted [command] [flags]

To start YugabyteDB cluster, run 'yugabyted start'.

Find more information at: https://docs.yugabyte.com/preview/reference/configuration/yugabyted/

Commands:
  start         Start YugabyteDB cluster.
  stop          Stop running YugabyteDB cluster.
  destroy       Destroy YugabyteDB cluster and remove data.
  status        Print status of YugabyteDB cluster.
  version       Release version of YugabyteDB cluster.
  collect_logs  Collect and package logs for troubleshooting.
  configure     Configure multi-zone/multiregion cluster
  connect       Connect to YugabyteDB cluster through the CLI.
  demo          Load and interact with preset demo data.

Flags:
  -h, --help    show this help message and exit
```

Demos

Deploying YugabyteDB Clusters Using yugabyted

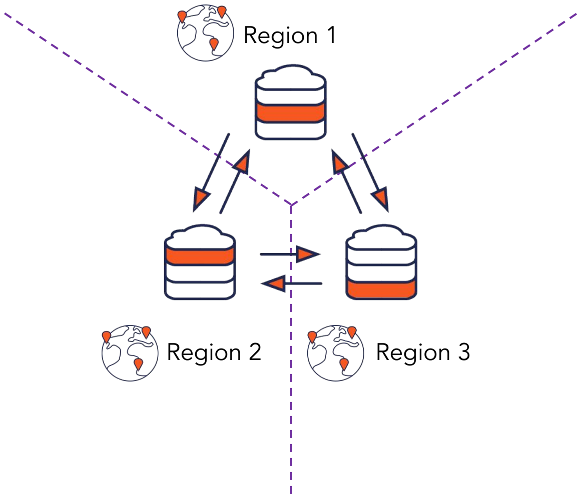
Deployment topologies supported by YugabyteDB

1. Single Region, Multi-Zone



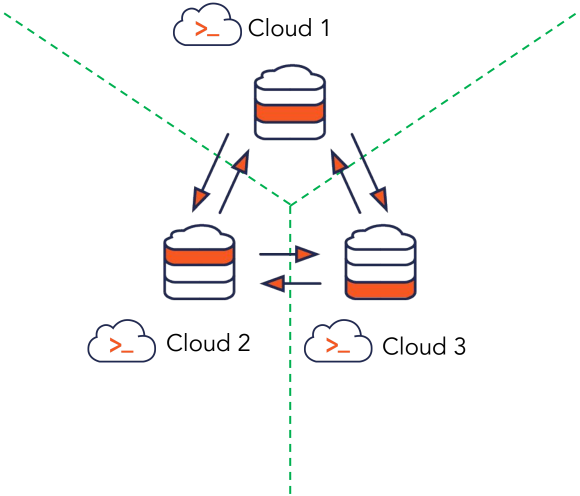
Consistent Across Zones
No WAN Latency But No
Region-Level Failover/Repair

2. Single Cloud, Multi-Region



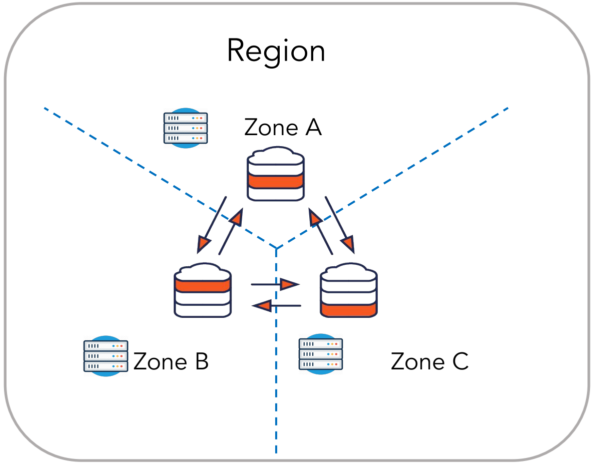
Consistent Across Regions
with Auto Region-Level
Failover/Repair

3. Multi-Cloud, Multi-Region



Consistent Across Clouds
with Auto Cloud-Level
Failover/Repair

Multi-zone topology using yugabyted



```
$ ./bin/yugabyted start
--advertise_address=<host-ip>
--cloud_location=cloud.region.zone-a
--fault_tolerance=zone
```

Step 1

Start the yugabyted with the `--cloud_location` flag

- ○ Cloud location will be `cloud.region.zone` details corresponding to VM/container
- ○ For example: Deploying to AWS EC2 instance in us-east-2a, cloud location will be `aws.us-east.us-east-2a`

- `--fault_tolerance` flag will be set to **zone**
  - ○ YugabyteDB will be configured to survive zone failure

Joining the YugabyteDB cluster

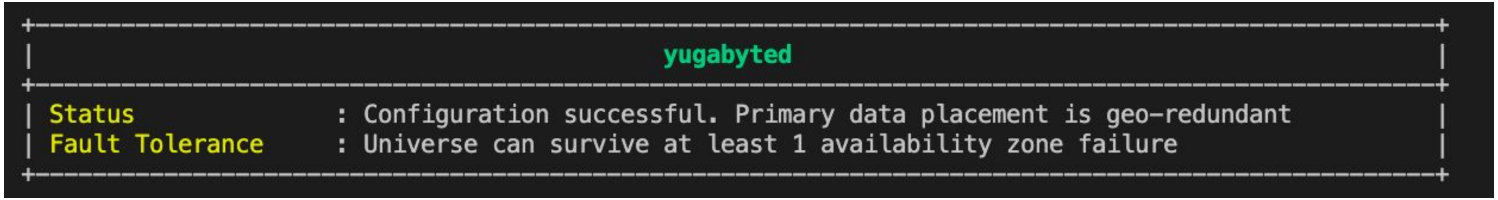
```
$ ./bin/yugabyted start --advertise_address=<host-ip>
--cloud_location=cloud.region.zone-b
--fault_tolerance=zone
--join=<ip-first-node>
```

Step 2

- After the first node is started, all the other nodes join the first node using `--join` flag
- Repeat this step on all the nodes joining the YugabyteDB cluster

Configure multi-zone configuration using yugabyted

```
$ ./bin/yugabyted configure --fault_tolerance=zone
```

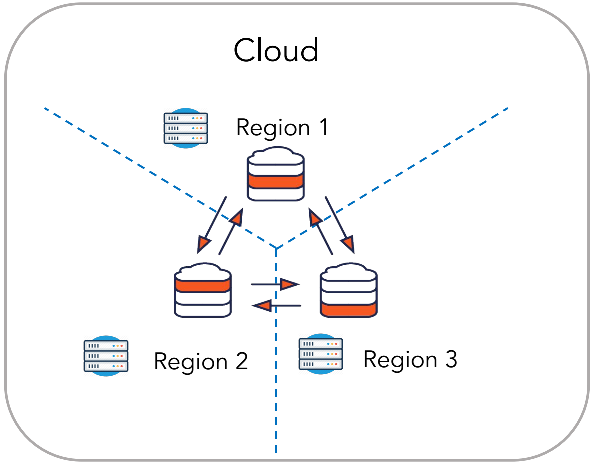


```
+--------------------------------------------------------------------------------------------------+
|                                            yugabyted                                             |
+--------------------------------------------------------------------------------------------------+
| Status                : Configuration successful. Primary data placement is geo-redundant        |
| Fault Tolerance       : Universe can survive at least 1 availability zone failure                |
+--------------------------------------------------------------------------------------------------+
```

Step 3

- `configure` command automatically applies the data placement constraint and replication factor based on the available zones.
- Data placement constraint can be overridden using the `--data_placement_constraint` flag. Replication factor can be overridden using the `--rf` flag.

Multi-region topology using yugabyted



```
$ ./bin/yugabyted start
--advertise_address=<host-ip>
--cloud_location=cloud.region-1.zone-
a --fault_tolerance=region
```

Step 1

Start the yugabyted nodes with the `--cloud_location` flag

- ○ Cloud location will be `cloud.region.zone` details corresponding to VM/container
- ○ For example: Deploying to AWS EC2 instance in us-east-2a, cloud location will be `aws.us-east.us-east-2a`

- `--fault_tolerance` flag will be set to **region**
  - ○ YugabyteDB will be configured to survive region failure

Joining the YugabyteDB cluster

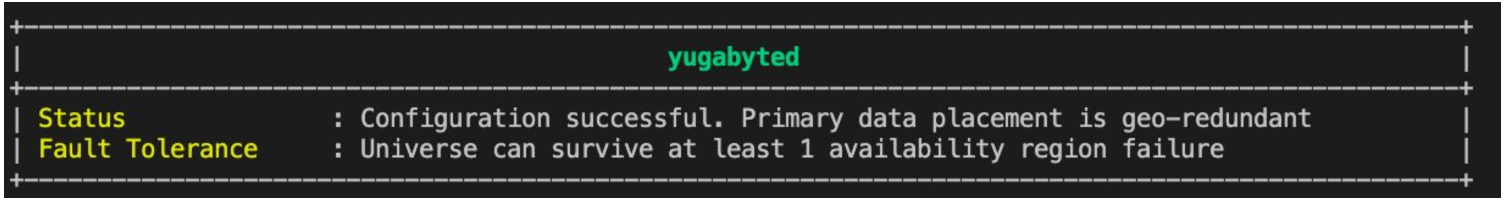
```
$ ./bin/yugabyted start --advertise_address=<host-ip>
--cloud_location=cloud.region-2.zone-a
--fault_tolerance=region
--join=<ip-first-node>
```

Step 2

- After the first node is started, all the other nodes join the first node using `--join` flag
- Repeat this step on all the nodes joining the YugabyteDB cluster

Configure multi-region configuration using yugabyted

```
$ ./bin/yugabyted configure --fault_tolerance=region
```



```
+----------------------------------------------------------------------------------------------+
|                                          yugabyted                                           |
+----------------------------------------------------------------------------------------------+
| Status              : Configuration successful. Primary data placement is geo-redundant      |
| Fault Tolerance     : Universe can survive at least 1 availability region failure            |
+----------------------------------------------------------------------------------------------+
```

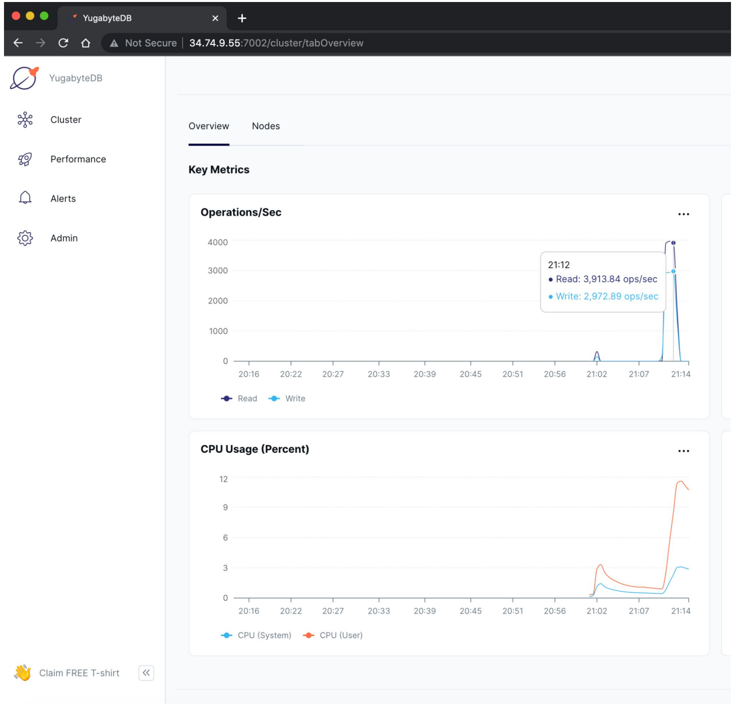
Step 2

- `configure` command automatically applies the data placement constraint and replication factor based on the available regions.
- Data placement constraint can be overridden using the `--data_placement_constraint` flag. Replication factor can be overridden using the `--rf` flag.

yugabyted UI

Managing YugabyteDB using yugabyted UI

```
$ ./yugabyted start --ui=true
```



Not Secure | 34.74.9.55:7002/cluster/tabOverview

Thank You

Join us on Slack: yugabyte.com/slack (#yftt channel)

Star us on Github: github.com/yugabyte/yugabyte-db

YFTT YUGABYTEDB FRIDAY TECH TALKS

yugabyteDB