



Nodes

`Node` is an object representing a node or system component.

Basic

`Node` is an abstract concept that represents a single system component object.

A node object consists of three parts: **provider**, **resource type** and **name**. You may already have seen each part in the previous example.

```
from diagrams import Diagram
from diagrams.aws.compute import EC2

with Diagram("Simple Diagram"):
    EC2("web")
```

In the example above, the `EC2` is a node of resource type `compute` which is provided by the `aws` provider.

You can use other node objects in a similar manner:

```
# aws resources
from diagrams.aws.compute import ECS, Lambda
from diagrams.aws.database import RDS, ElastiCache
from diagrams.aws.network import ELB, Route53, VPC
...

# azure resources
from diagrams.azure.compute import FunctionApps
from diagrams.azure.storage import BlobStorage
...
```

```
# alibaba cloud resources
from diagrams.alibabacloud.compute import ECS
from diagrams.alibabacloud.storage import ObjectTableStore
...

# gcp resources
from diagrams.gcp.compute import AppEngine, GKE
from diagrams.gcp.ml import AutoML
...

# k8s resources
from diagrams.k8s.compute import Pod, StatefulSet
from diagrams.k8s.network import Service
from diagrams.k8s.storage import PV, PVC, StorageClass
...

# oracle resources
from diagrams.oci.compute import VirtualMachine, Container
from diagrams.oci.network import Firewall
from diagrams.oci.storage import FileStorage, StorageGateway
```

You can find lists of all available nodes for each provider in the sidebar on the left.

For example, [here](#) is the list of all available AWS nodes.

Data Flow

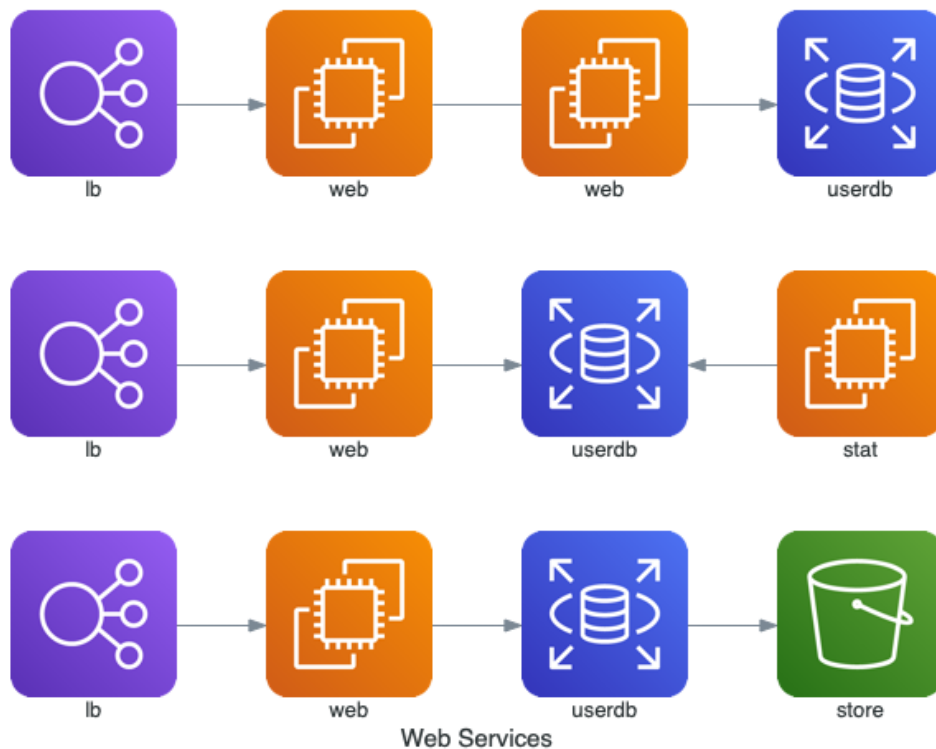
You can represent data flow by connecting the nodes with the operators `>>` , `<<` , and `-` .

- `>>` connects nodes in left to right direction.
- `<<` connects nodes in right to left direction.
- `-` connects nodes in no direction. Undirected.

```
from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.aws.database import RDS
from diagrams.aws.network import ELB
from diagrams.aws.storage import S3

with Diagram("Web Services", show=False):
    ELB("lb") >> EC2("web") >> RDS("userdb") >> S3("store")
    ELB("lb") >> EC2("web") >> RDS("userdb") << EC2("stat")
    (ELB("lb") >> EC2("web")) - EC2("web") >> RDS("userdb")
```

Be careful when using `-` and any shift operators together. It can cause unexpected results due to Python's operator precedence, so you might have to use parentheses.



The order of rendered diagrams is the reverse of the declaration order.

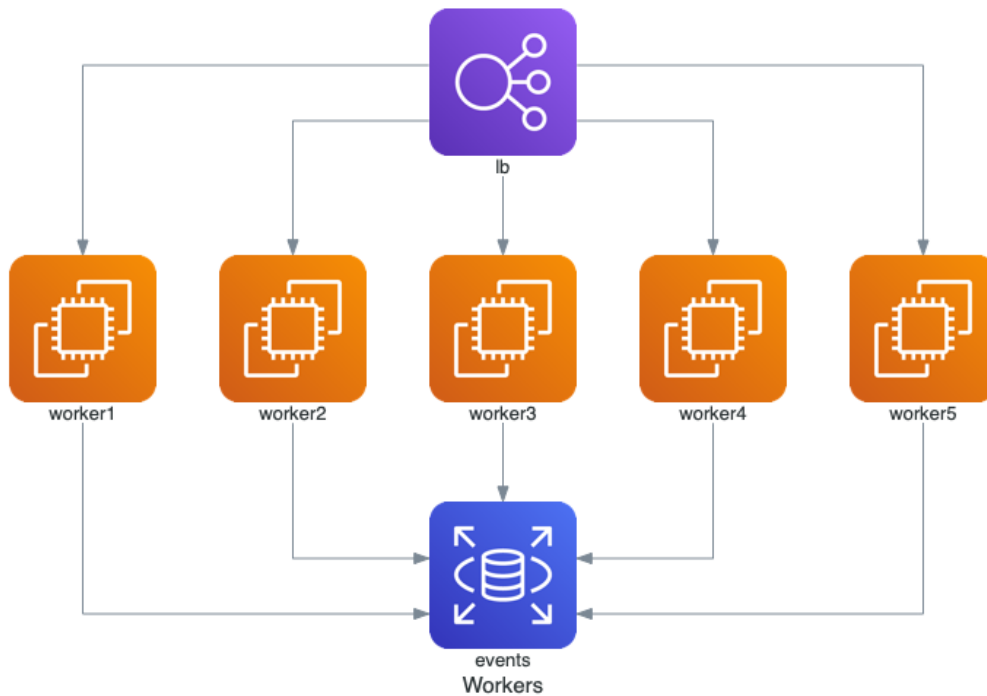
You can change the data flow direction with the `direction` parameter. The default is **LR**.

Allowed values are: TB, BT, LR, and RL

```
from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.aws.database import RDS
from diagrams.aws.network import ELB

with Diagram("Workers", show=False, direction="TB"):
    lb = ELB("lb")
    db = RDS("events")
    lb >> EC2("worker1") >> db
    lb >> EC2("worker2") >> db
    lb >> EC2("worker3") >> db
```

```
lb >> EC2("worker4") >> db
lb >> EC2("worker5") >> db
```



Group Data Flow

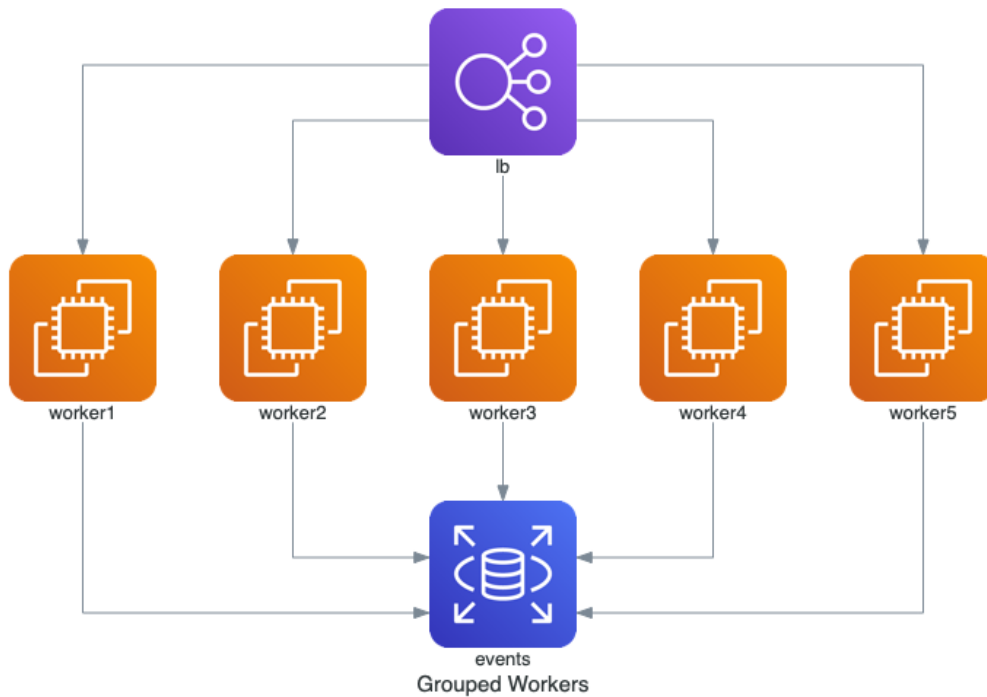
The above worker example has too many redundant flows. To avoid this, you can group nodes into a list so that all nodes are connected to other nodes at once:

```

from diagrams import Diagram
from diagrams.aws.compute import EC2
from diagrams.aws.database import RDS
from diagrams.aws.network import ELB

with Diagram("Grouped Workers", show=False, direction="TB"):
    ELB("lb") >> [EC2("worker1"),
                   EC2("worker2"),
                   EC2("worker3"),
                   EC2("worker4"),
                   EC2("worker5")] >> RDS("events")

```



You can't connect two **lists** directly because shift/arithmetic operations between lists are not allowed in Python.

Last updated on 4/13/2024

[← DIAGRAMS](#)

[CLUSTERS →](#)



Docs

Getting Started

Guides

Nodes

More

Star

40,612