Diagrams Package

Draw the cloud system architecture in python

Bhavika Sewpal

August 11, 2022

Motivation

- Consistency (same icons)
- Easy to update
- Keep track of changes in the diagram

Purpose

- Diagrams let you draw the cloud system architecture in Python code
- Diagrams currently supports main major providers including:
  - AWS
  - Azure
  - GCP
  - Kubernetes
  - Alibaba

Requirements

- Diagrams require Python 3.6 or higher
- It uses Graphviz - an open source graph visualization software - to render the diagram

Objects

- Diagrams has 4 objects:
  - Diagrams
  - Nodes
  - Clusters
  - Edges

Diagrams

- Diagram represents a global diagram context
- A diagram context is created with the Diagram class

```
from diagrams import Diagram
from diagrams.k8s.compute import Pod

with Diagram("Simple Diagram", show=False):
    Pod("pod instance")
```

Diagrams (cont.)

pod instance

Simple Diagram

Nodes

- A node represents a system component
- A node consists of three parts:
  - a provider
  - a resource type
  - a name

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

In the above example, aws is the provider, compute is the resource type and EC2 is the name

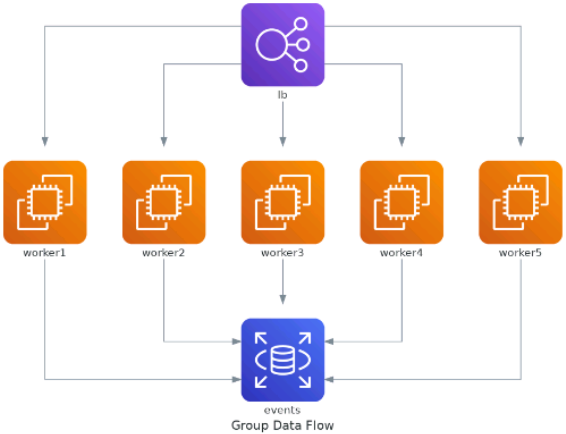
Nodes - Data Flow

- >> : Connects node in left to right direction
- << : Connects node in right to left direction
- : : Undirected

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

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

Nodes - Data Flow (cont.)

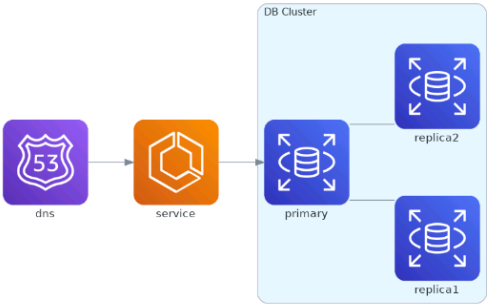


Cluster

- Cluster allows you to group nodes in an isolated group
- A cluster context is created with the Cluster Class

```
from diagrams import Cluster, Diagram
from diagrams.aws.compute import ECS
from diagrams.aws.database import RDS
from diagrams.aws.network import Route53
with Diagram("Simple Web Service with DB Cluster", show=False):
    dns = Route53("dns")
    web = ECS("service")
    with Cluster("DB Cluster"):
        db_primary = RDS("primary")
        db_primary - [RDS("replica1"),RDS("replica2")]
    dns >> web >> db_primary
```

Cluster (cont.)



Simple Web Service with DB Cluster

- Clusters can be nested as well

Edges

- An edge is a connection between nodes with some additional properties
- An edge object contains three attributes:
  - label
  - color
  - style (example: dashed, dotted, bold)

Questions?