IntelliCloud API Documentation

The IntelliCloud API provides comprehensive management and monitoring capabilities for cloud infrastructure, focusing on cost optimization and resource efficiency. It allows users to retrieve data about clusters, nodes, pods, and their associated costs, both in regular and optimized scenarios.

Base URL

```
http://localhost:8000
```

Endpoints

Clusters

Get All Clusters

- **URL**: `/clusters`
- **Method**: `GET`
- **Description**: Retrieves a list of all clusters.
- **Response**: Array of cluster objects.
- **Example Response**:

```
[
  {
    "id": 1,
    "name": "production-cluster",
    "region": "us-west-1",
    "provider": "AWS",
    "total_nodes": 5,
    "total_pods": 50,
    "total_vcpu": 40.0,
    "total_memory": 128.0,
    "total_cpu_cost": 200.0,
    "total_memory_cost": 150.0,
    "total_cost": 350.0,
    "optimized_total_cpu_cost": 180.0,
    "optimized_total_memory_cost":
120.0,
    "optimized_total_cost": 300.0,
    "compute_cost_per_month": 10500.0,
    "status": "Active",
    "created_at": "2023-10-
01T12:00:00"
  }
]
```


Get Cluster by ID

- **URL**: `/clusters/<int:cluster_id>`

- **Method**: `GET`
- **Parameters**:
  - `cluster_id` (path, integer): ID of the cluster.
- **Response**: Cluster object.
- **Error Response**: `404 Not Found` if cluster does not exist.

Nodes

Get All Nodes

- **URL**: `/nodes`
- **Method**: `GET`
- **Description**: Retrieves all nodes with nested node type details.
- **Response**: Array of node objects with `node_type` nested.
- **Example Response**:

```
[
  {
    "id": 1,
    "name": "worker-node-1",
```

```
    "cluster_id": 1,
    "cpu_cost_rate": 0.05,
    "memory_cost_rate": 0.02,
    "node_type": {
      "id": 101,
      "name": "m5.xlarge",
      "vcpu": 4.0,
      "ram_size": 16.0,
      "cpu_cost_rate_min": 0.04,
      "cpu_cost_rate_max": 0.06,
      "ram_cost_rate_min": 0.01,
      "ram_cost_rate_max": 0.03,
      "bandwidth": 1000.0,
      "power_model": "linear"
    },
    "created_at": "2023-10-
01T12:00:00"
  }
]
```


Get Node by ID

- **URL**: `/nodes/<int:node_id>`
- **Method**: `GET`
- **Parameters**:
  - `node_id` (path, integer): ID of the node.

- **Response**: Node object with nested `node_type`.

Pods

Get All Pods

- **URL**: `/pods`
- **Method**: `GET`
- **Description**: Retrieves all pods.
- **Response**: Array of pod objects.
- **Example Response**:


```
[
  {
    "id": 1,
    "name": "web-pod-1",
    "cluster_id": 1,
    "node_id": 1,
    "cpu_request": 0.5,
    "cpu_used": 0.6,
    "memory_request": 512.0,
    "memory_used": 600.0,
    "status": "Running",
    "created_at": "2023-10-
```

```
01T12:00:00"
  }
]
```

Usage & Cost Analysis

Get Node Usage

- **URL**: `/nodes/<int:node_id>/usage`
- **Method**: `GET`
- **Parameters**:
  - `node_id` (path, integer): ID of the node.
- **Response**: Array of usage entries for the node.
- **Example Response**:

```
[
  {
    "id": 1,
    "node_id": 1,
    "cpu_used": 3.2,
    "memory_used": 8.0,
    "cpu_cost": 0.16,
    "memory_cost": 0.16,
    "total_cost": 0.32,
```

```
    "date": "2023-10-01T00:00:00"
  }
]
```

Compare Node Usage (Regular vs Optimized)

- **URL**: `/nodes/<int:node_id>/usage/compare`
- **Method**: `GET`
- **Response**:

```
{
  "node_id": 1,
  "regular": {
    "avg_cpu_used": 3.2,
    "avg_memory_used": 8.0,
    "total_cost": 32.0
  },
  "optimized": {
    "avg_cpu_used": 2.8,
    "avg_memory_used": 6.5,
    "total_cost": 25.0
  },
  "savings": {
    "cpu_savings_percentage": 12.5,
    "memory_savings_percentage":
18.75,
    "cost_savings_percentage": 21.88
```

```
    }
}
```

Savings Reports

Get Monthly Cluster Savings

- **URL**: `/clusters/<int:cluster_id>/savings/monthly`
- **Method**: `GET`
- **Response**: Monthly aggregated savings data.
- **Example Response**:

```
[
  {
    "month": 10,
    "regular_cpu_cost": 200.0,
    "regular_memory_cost": 150.0,
    "regular_total_cost": 350.0,
    "optimized_cpu_cost": 180.0,
    "optimized_memory_cost": 120.0,
    "optimized_total_cost": 300.0,
    "cpu_savings_percentage": 10.0,
    "memory_savings_percentage": 20.0,
```

```
        "cost_savings_percentage": 14.29
      }
    ]
```

Data Models

Cluster

Field	Type	Description
`id`	Integer	Unique identifier
`name`	String	Cluster name
`region`	String	Cloud region (e.g., "us-west-1")
`provider`	String	Cloud provider (e.g., "AWS")
`total_nodes`	Integer	Total nodes in the cluster
`total_pods`	Integer	Total pods in the cluster
`total_vcpu`	Float	Total vCPUs

Field	Type	Description
		allocated
`total_memory`	Float	Total memory allocated (GB)
`total_cost`	Float	Total monthly cost ($)
`optimized_total_cost`	Float	Optimized total cost ($)
`status`	String	Cluster status (e.g., "Active")
`created_at`	DateTime	Creation timestamp

Node

Field	Type	Description
`id`	Integer	Unique identifier
`cluster_id`	Integer	Parent cluster ID
`node_type`	Object	NodeType details (nested)
`cpu_cost_rate`	Float	Cost per vCPU/hour ($)

Field	Type	Description
`memory_cost_rate`	Float	Cost per GB memory/hour ($)

Error Handling

- **404 Not Found**: Returned when a resource (cluster, node, pod) does not exist.

```
{ "error": "Cluster not found" }
```

Example Usage Scenario

Goal: Compare cost savings for Cluster ID 1.

1. **Get Cluster Details**:

```
GET /clusters/1
```

2. **Retrieve Monthly Savings**:

```
GET /clusters/1/savings/monthly
```

3. **Analyze Efficiency**:

```
GET /clusters/1/efficiency
```