## Fake Data for the tasks/ Hierarchy

**1. Deployment Paths**

**tasks/deployment/aws/prod/**

- **AWS Access Key**: "aws_access_key" = "PROD_AWS_ACCESS_KEY_12345"
- **AWS Secret Key**: "aws_secret_key" = "PROD_AWS_SECRET_KEY_67890"

**tasks/deployment/aws/dev/**

- **AWS Access Key**: "aws_access_key" = "DEV_AWS_ACCESS_KEY_54321"
- **AWS Secret Key**: "aws_secret_key" = "DEV_AWS_SECRET_KEY_09876"

**tasks/deployment/kubernetes/prod/**

- **Kubeconfig**: "kubeconfig" = "kube_prod_config_data"
- **Token**: "kube_token" = "PROD_KUBE_TOKEN_abcdef123456"

**tasks/deployment/kubernetes/dev/**

- **Kubeconfig**: "kubeconfig" = "kube_dev_config_data"
- **Token**: "kube_token" = "DEV_KUBE_TOKEN_zyxw98765"

**tasks/deployment/docker/prod/**

- **Docker Registry URL**: "registry_url" = "https://docker.prod.example.com"
- **Docker Credentials**: "docker_credentials" = "prod_docker_username:prod_docker_password"

**tasks/deployment/docker/dev/**

- **Docker Registry URL**: "registry_url" = "https://docker.dev.example.com"
- **Docker Credentials**: "docker_credentials" = "dev_docker_username:dev_docker_password"

**2. Config Paths**

**tasks/config/app1/credentials/**

- **Database User**: "db_user" = "app1_prod_user"
- **Database Password**: "db_password" = "app1_prod_pass_123"

**tasks/config/app1/settings/**

- **API Endpoint**: "api_endpoint" = "https://api.app1.example.com"
- **Timeout**: "timeout" = "30"

**tasks/config/app2/credentials/**

- **Database User**: "db_user" = "app2_dev_user"
- **Database Password**: "db_password" = "app2_dev_pass_456"

**tasks/config/app2/settings/**

- **API Endpoint**: "api_endpoint" = "https://api.app2.example.com"
- **Timeout**: "timeout" = "25"

**tasks/config/network/credentials/**

- **Network User**: "network_user" = "net_admin"
- **Network Password**: "network_password" = "net_secret_789"

**tasks/config/network/settings/**

- **Firewall Config**: "firewall" = "ALLOW ALL"
- **VPN Config**: "vpn" = "vpn_config_data_here"

**3. Monitoring Paths**

**tasks/monitoring/prometheus/configs/**

- **Prometheus URL**: "prometheus_url" = "https://prometheus.example.com"
- **Scrape Interval**: "scrape_interval" = "15s"

**tasks/monitoring/prometheus/tokens/**

- **Prometheus Token**: "prometheus_token" = "prometheus_token_123"

**tasks/monitoring/grafana/configs/**

- **Grafana URL**: "grafana_url" = "https://grafana.example.com"
- **Dashboard Refresh**: "refresh_rate" = "5m"

**tasks/monitoring/grafana/tokens/**

- **Grafana API Key**: "grafana_api_key" = "grafana_key_789"

**tasks/monitoring/alerts/configs/**

- **Alertmanager URL**: "alertmanager_url" = "https://alertmanager.example.com"
- **Alert Rules**: "alert_rules" = "disk_usage > 80%"

**tasks/monitoring/alerts/tokens/**

- **Alertmanager Token**: "alertmanager_token" = "alert_token_456"

## Vault Commands to Store Data

For each path, you can use a command similar to the following to store data:

```
vault kv put -mount=tasks deployment/aws/prod aws_access_key="PROD_AWS_ACCESS_KEY_12345" aws_secret_key="PROD_AWS_SECRET_KEY_67890"
```

## Fake Data for the tasks/ Hierarchy

### 1. Deployment Paths

**tasks/deployment/aws/prod/**

- **AWS Access Key**: "aws_access_key" = "PROD_AWS_ACCESS_KEY_12345"
- **AWS Secret Key**: "aws_secret_key" = "PROD_AWS_SECRET_KEY_67890"

**tasks/deployment/aws/dev/**

- **AWS Access Key**: "aws_access_key" = "DEV_AWS_ACCESS_KEY_54321"
- **AWS Secret Key**: "aws_secret_key" = "DEV_AWS_SECRET_KEY_09876"

**tasks/deployment/kubernetes/prod/**

- **Kubeconfig**: "kubeconfig" = "kube_prod_config_data"
- **Token**: "kube_token" = "PROD_KUBE_TOKEN_abcdef123456"

**tasks/deployment/kubernetes/dev/**

- **Kubeconfig**: "kubeconfig" = "kube_dev_config_data"
- **Token**: "kube_token" = "DEV_KUBE_TOKEN_zyxw98765"

**tasks/deployment/docker/prod/**

- **Docker Registry URL**: "registry_url" = "https://docker.prod.example.com"
- **Docker Credentials**: "docker_credentials" = "prod_docker_username:prod_docker_password"

**tasks/deployment/docker/dev/**

- **Docker Registry URL**: "registry_url" = "https://docker.dev.example.com"
- **Docker Credentials**: "docker_credentials" = "dev_docker_username:dev_docker_password"

**2. Config Paths**

**tasks/config/app1/credentials/**

- **Database User**: "db_user" = "app1_prod_user"
- **Database Password**: "db_password" = "app1_prod_pass_123"

**tasks/config/app1/settings/**

- **API Endpoint**: "api_endpoint" = "https://api.app1.example.com"
- **Timeout**: "timeout" = "30"

**tasks/config/app2/credentials/**

- **Database User**: "db_user" = "app2_dev_user"
- **Database Password**: "db_password" = "app2_dev_pass_456"

**tasks/config/app2/settings/**

- **API Endpoint**: "api_endpoint" = "https://api.app2.example.com"
- **Timeout**: "timeout" = "25"

**tasks/config/network/credentials/**

- **Network User**: "network_user" = "net_admin"
- **Network Password**: "network_password" = "net_secret_789"

**tasks/config/network/settings/**

- **Firewall Config**: "firewall" = "ALLOW ALL"
- **VPN Config**: "vpn" = "vpn_config_data_here"

**3. Monitoring Paths**

**tasks/monitoring/prometheus/configs/**

- **Prometheus URL**: "prometheus_url" = "https://prometheus.example.com"
- **Scrape Interval**: "scrape_interval" = "15s"

**tasks/monitoring/prometheus/tokens/**

- **Prometheus Token**: "prometheus_token" = "prometheus_token_123"

**tasks/monitoring/grafana/configs/**

- **Grafana URL**: "grafana_url" = "https://grafana.example.com"
- **Dashboard Refresh**: "refresh_rate" = "5m"

**tasks/monitoring/grafana/tokens/**

- **Grafana API Key**: "grafana_api_key" = "grafana_key_789"

**tasks/monitoring/alerts/configs/**

- **Alertmanager URL**: "alertmanager_url" = "https://alertmanager.example.com"
- **Alert Rules**: "alert_rules" = "disk_usage > 80%"

**tasks/monitoring/alerts/tokens/**

- **Alertmanager Token**: "alertmanager_token" = "alert_token_456"

## Vault Commands to Store Data

For each path, you can use a command similar to the following to store data:

bash

Copy code

```
vault kv put -mount=tasks deployment/aws/prod
aws_access_key="PROD_AWS_ACCESS_KEY_12345"
aws_secret_key="PROD_AWS_SECRET_KEY_67890"
```

You would repeat a similar command for each path with the relevant key-value pairs.

## Summary of Example Data

- **AWS credentials** are stored separately for production and development.
- **Kubernetes secrets** contain kubeconfig and tokens for each environment.
- **Docker secrets** include registry URLs and credentials.
- **App configurations** have credentials and specific settings like endpoints and timeouts.
- **Monitoring** configurations and tokens include URLs and refresh intervals.

This fake data helps illustrate the structure while demonstrating how secrets can be organized and stored in HashiCorp Vault. Let me know if you need more details or additional variations!