ANSIBLE meetup

AnsibleBenelux 2018/9/19

WiFi: visitorswifi

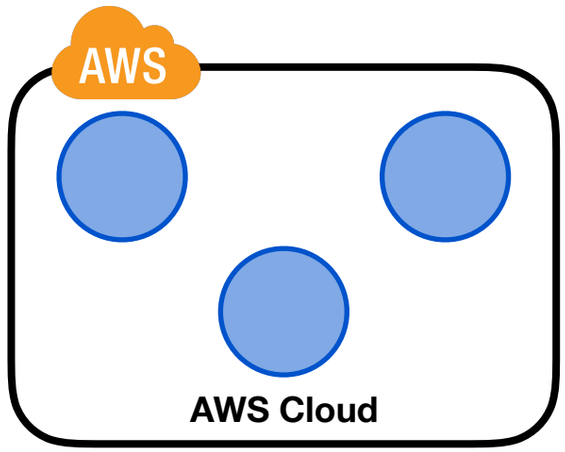
WiFi: visitorswifi

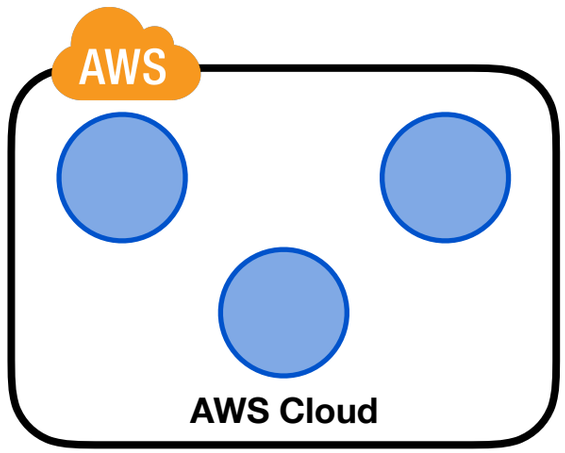
No password

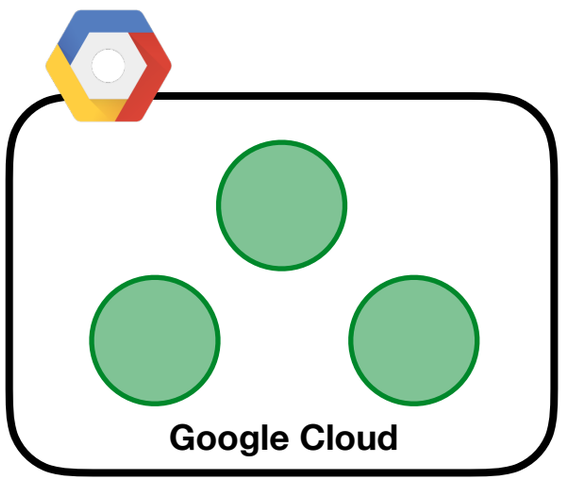
#AnsibleBenelux

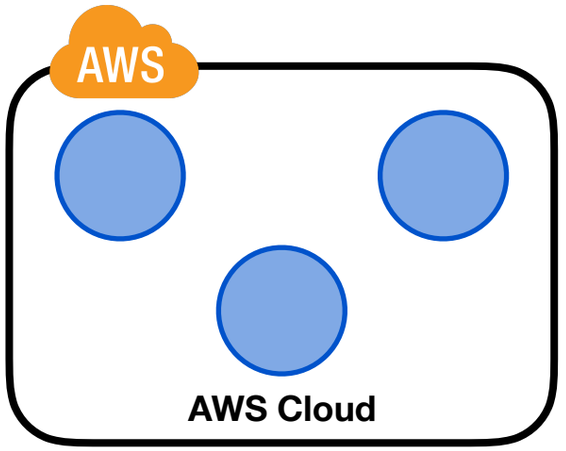
Secrets in Ansible

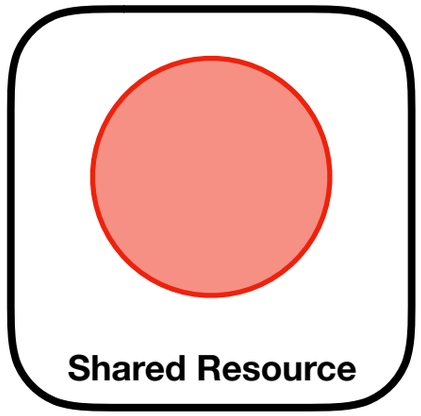
Provisioning machines with passwords, API tokens and other secrets without getting a headache

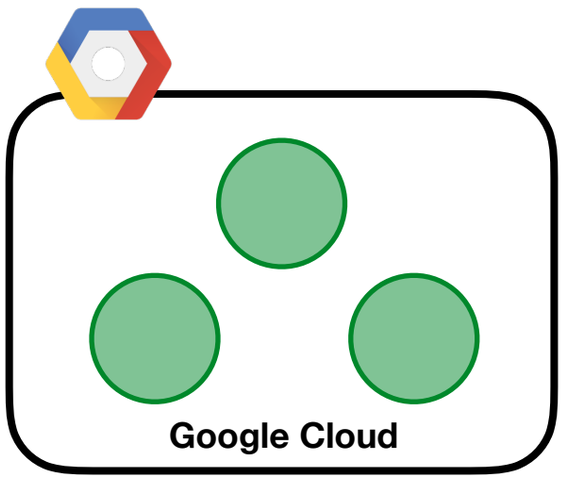

AWS
AWS Cloud

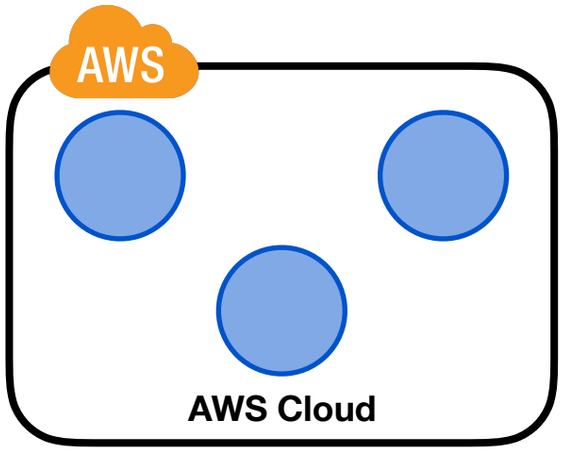

AWS
AWS Cloud

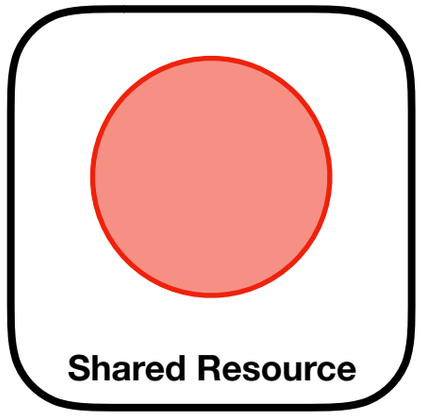

Google Cloud


AWS
AWS Cloud

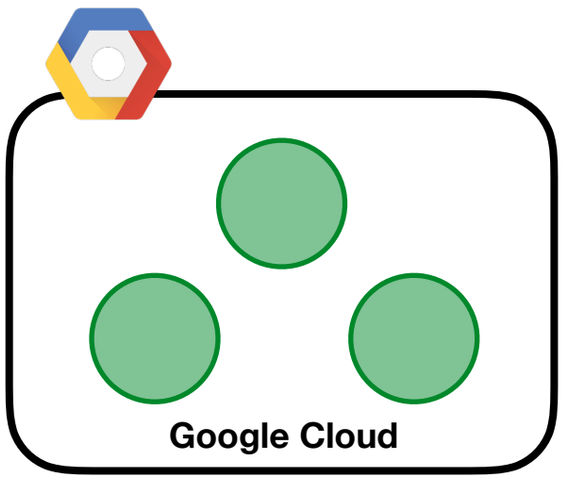

Shared Resource

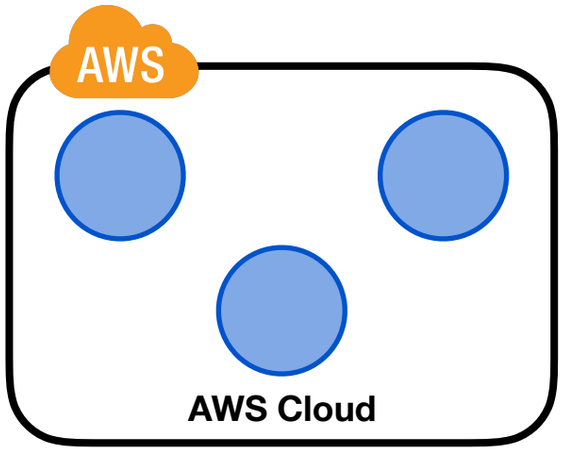

Google Cloud

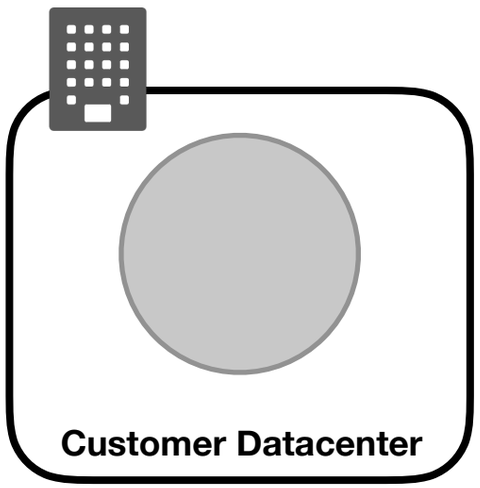

AWS
AWS Cloud

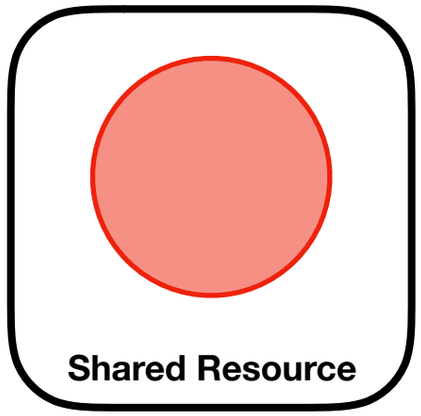

Shared Resource


DATADOG
twilio
NASA
stripe
MailChimp

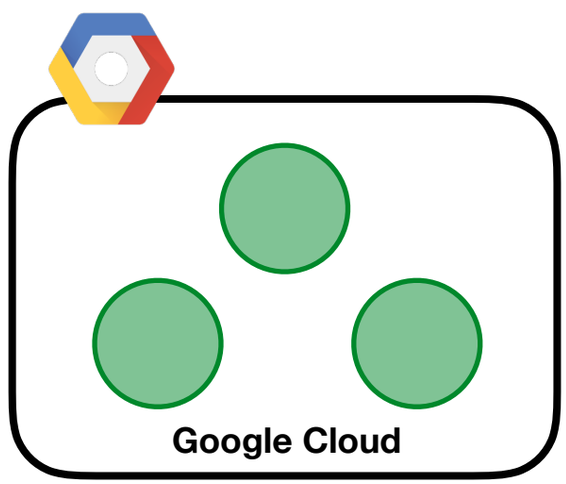

Google Cloud

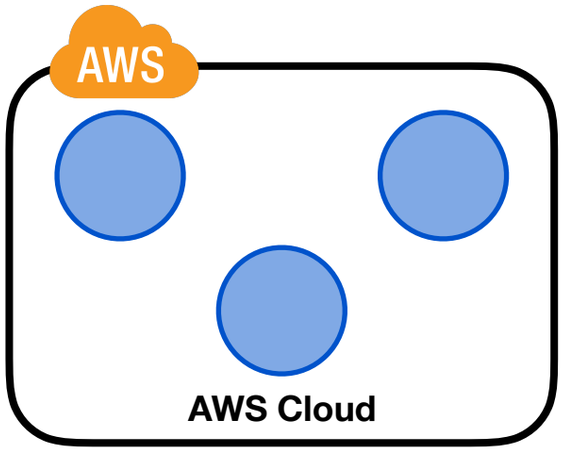

AWS
AWS Cloud

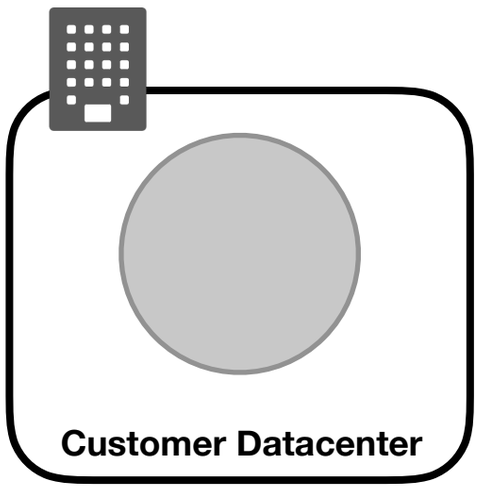

Customer Datacenter

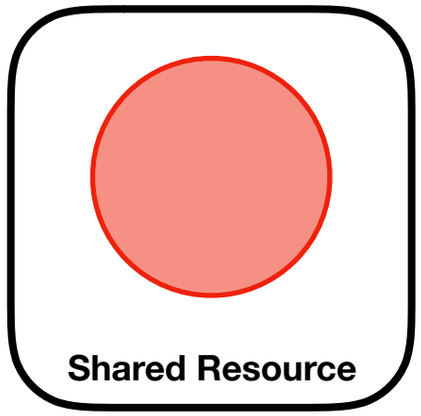

Shared Resource


DATADOG
twilio
NASA
stripe
MailChimp

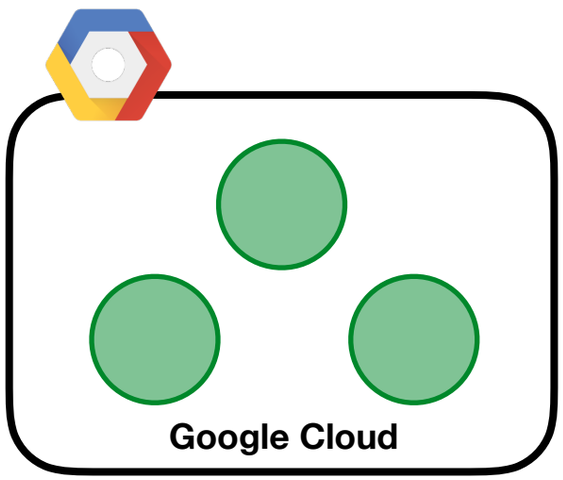

Google Cloud

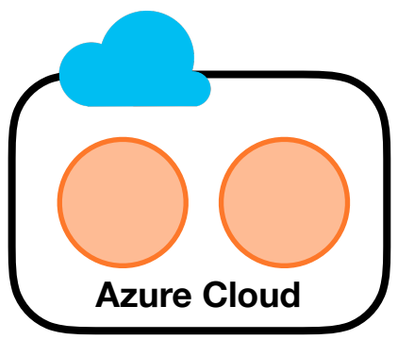

AWS
AWS Cloud

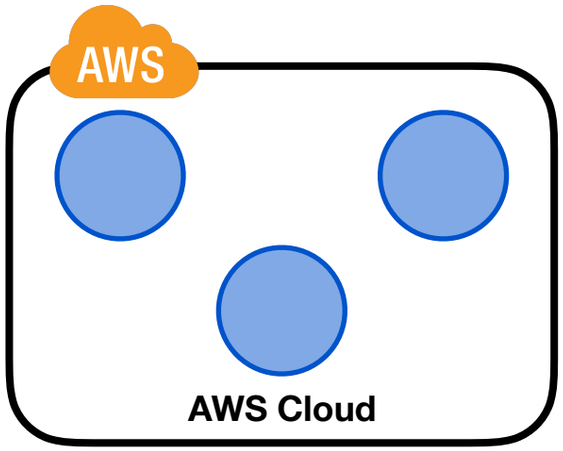

Customer Datacenter

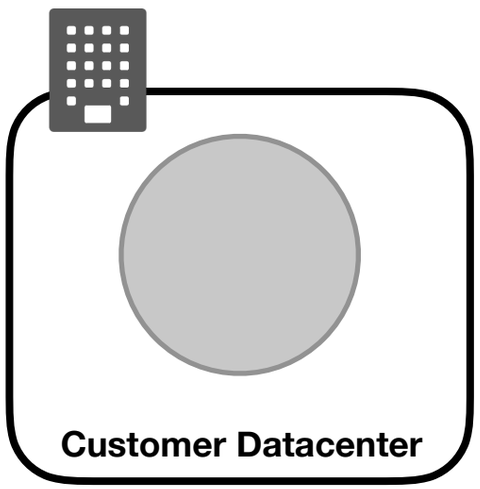

Shared Resource

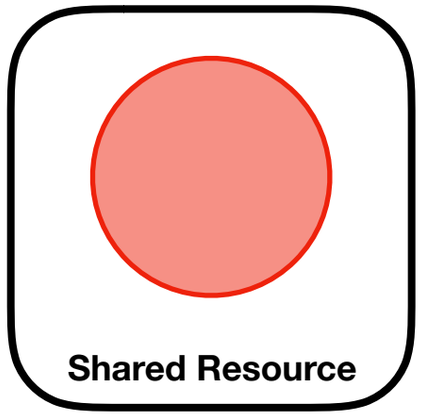

DATADOG
twilio
NASA
stripe
MailChimp


Google Cloud

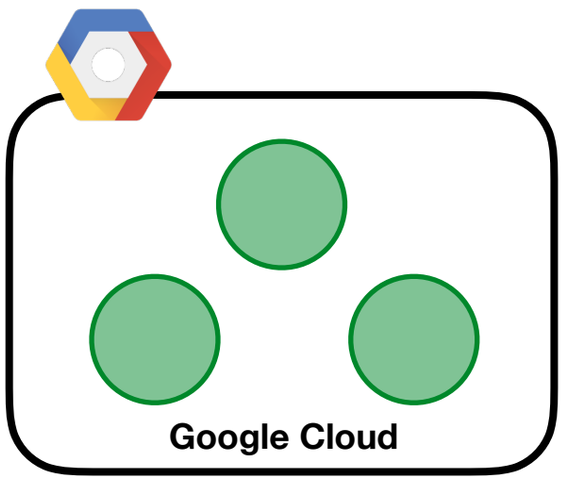

Azure Cloud

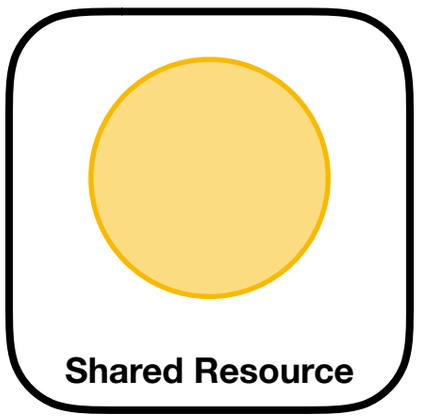

AWS
AWS Cloud

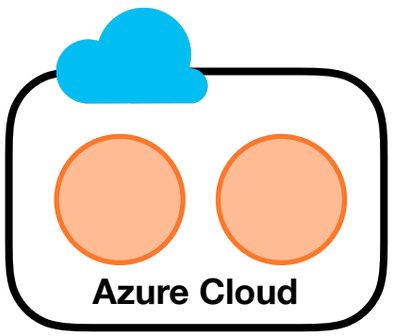

Customer Datacenter


Shared Resource


DATADOG
twilio
NASA
stripe
MailChimp


Google Cloud


Shared Resource


Azure Cloud

AWS

AWS Cloud

Customer Datacenter

Shared Resource

DATADOG

twilio

NASA

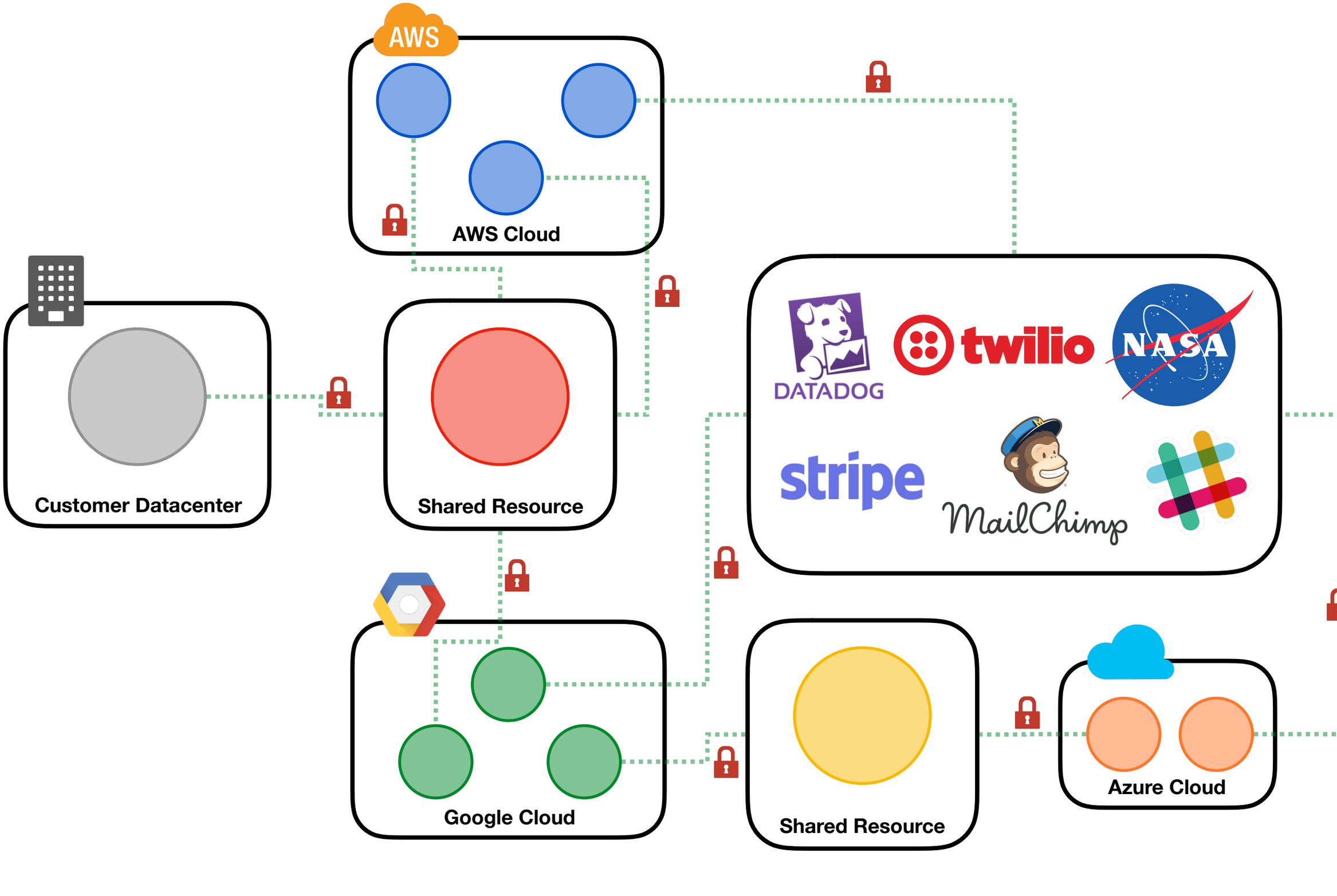
stripe

MailChimp

Google Cloud

Shared Resource

Azure Cloud


AWS
AWS Cloud
Customer Datacenter
Shared Resource
DATADOG
twilio
NASA
stripe
MailChimp
Google Cloud
Shared Resource
Azure Cloud

DevOps

AWS

AWS Cloud

Customer Datacenter

Shared Resource

DATADOG

twilio

NASA

stripe

MailChimp

Google Cloud

Shared Resource

Azure Cloud

Admins

DevOps

AWS

AWS Cloud

Customer Datacenter

Shared Resource

DATADOG

twilio

NASA

stripe

MailChimp

Google Cloud

Shared Resource

Azure Cloud

Admins

DevOps

Outside Contractors

AWS

AWS Cloud

Customer Datacenter

Shared Resource

DATADOG

twilio

NASA

stripe

MailChimp

Google Cloud

Shared Resource

Azure Cloud

Admins

DevOps

Outside Contractors

AWS

AWS Cloud

Customer Datacenter

Shared Resource

DATADOG

twilio

NASA

stripe

MailChimp

Google Cloud

Shared Resource

Azure Cloud


peteneems

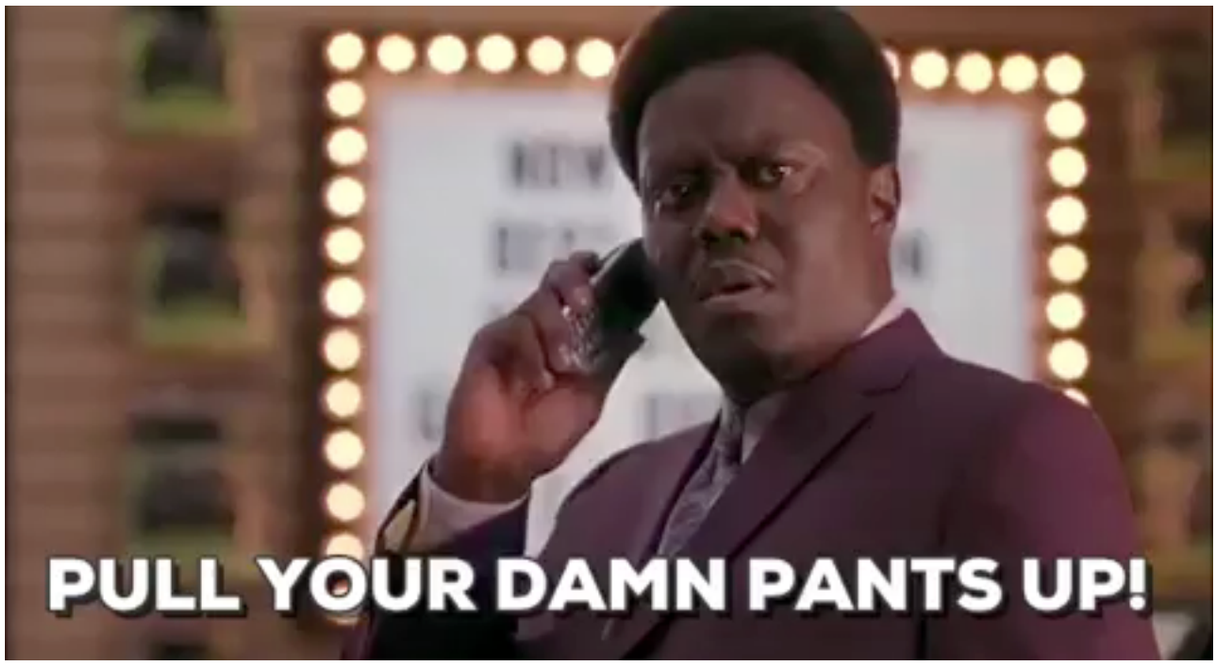
Marc Mackenbach

marc@secrethub.io

@marcmackenbach

mackenbach

SecretHub


PULL YOUR DAMN PANTS UP!

ansible-vault

ansible-vault

Vault is a built-in feature of Ansible that helps users to encrypt vars with a password.

p3pp3r0ni_pizz4

DevOps

team_vault.yml

bi9d4ddy

Admins

admins_vault.yml

p3pp3r0ni_pizz4

DevOps

team_vault.yml

```yaml
---
- hosts: db_servers
  tasks:
    - include_vars: admin_vault.yml
    - include_vars: dev_vault.yml
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user }}"
        password: "{{ db_pass }}"
        login_user: "{{ root_user }}"
        login_password: "{{ root_pass }}"
        state: present
        priv: characters:select
- hosts: web_servers
  tasks:
    - include_vars: dev_vault.yml
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user }}"
        DB_PASSWORD: "{{ db_pass }}"
        SLACK_TOKEN: "{{ slack_token }}"
```

```yaml
---
- hosts: db_servers
  tasks:
    - include_vars: admin_vault.yml
    - include_vars: dev_vault.yml
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user }}"
        password: "{{ db_pass }}"
        login_user: "{{ root_user }}"
        login_password: "{{ root_pass }}"
        state: present
        priv: characters:select
- hosts: web_servers
  tasks:
    - include_vars: dev_vault.yml
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user }}"
        DB_PASSWORD: "{{ db_pass }}"
        SLACK_TOKEN: "{{ slack_token }}"
```

admin_vault.yml

dev_vault.yml

Admin runs playbook

Dev runs playbook

bi9d4ddy

Admins

admins_vault.yml

p3pp3r0ni_pizz4

DevOps

team_vault.yml

bi9d4ddy

Admins

admins_vault.yml

p3pp3r0ni_pizz4

DevOps

team_vault.yml

f1yin9_dutchm4n

Outside Contractors

external_vault.yml

1Password

bi9d4ddy

Admins

admins_vault.yml

p3pp3r0ni_pizz4

DevOps

team_vault.yml

f1yin9_dutchm4n

Outside Contractors

external_vault.yml

Writing is manual


WHAT DO WE WANT?
WRITING ENCRYPTED SECRETS
WHEN DO WE WANT IT?
AT RUNTIME
imgflip.com

People leave

Rotate all secrets

SecretHub

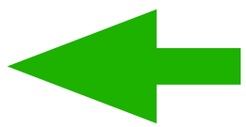
Ansible Module

```yaml
---
- hosts: db_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read database root user"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_user
      register: db_root_user
    - name: "Read database root password"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_pass
      register: db_root_pass
    - name: "Generate a database username for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Generate a database password for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user.secret }}"
        password: "{{ db_pass.secret }}"
        login_user: "{{ db_root_user.secret }}"
        login_password: "{{ db_root_pass.secret }}"
        state: present
        priv: characters:select
- hosts: web_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read the app's database user"
      secrethub_read:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Read the app's database password"
      secrethub_read:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Read the app's slack_token"
      secrethub_read:
        path: ansible-demo/infra/app/slack_token
      register: slack_token
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user.secret }}"
        DB_PASSWORD: "{{ db_pass.secret }}"
        SLACK_TOKEN: "{{ slack_token.secret }}"
```

```yaml
---
- hosts: db_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
      secrethub_read:
        path: ansible-demo/infra/postgres/root_user
      register: db_root_user
    - name: "Read database root password"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_pass
      register: db_root_pass
    - name: "Generate a database username for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Generate a database password for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user.secret }}"
        password: "{{ db_pass.secret }}"
        login_user: "{{ db_root_user.secret }}"
        login_password: "{{ db_root_pass.secret }}"
        state: present
        priv: characters:select
- hosts: web_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read the app's database user"
      secrethub_read:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Read the app's database password"
      secrethub_read:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Read the app's slack_token"
      secrethub_read:
        path: ansible-demo/infra/app/slack_token
      register: slack_token
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user.secret }}"
        DB_PASSWORD: "{{ db_pass.secret }}"
        SLACK_TOKEN: "{{ slack_token.secret }}"
```

secrethub_cli

```yaml
---
- hosts: db_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read database root user"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_user
      register: db_root_user
    - name: "Read database root password"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_pass
      register: db_root_pass
    - name: "Generate a database username for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Generate a database password for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user.secret }}"
        password: "{{ db_pass.secret }}"
        login_user: "{{ db_root_user.secret }}"
        login_password: "{{ db_root_pass.secret }}"
        state: present
        priv: characters:select
- hosts: web_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read the app's database user"
      secrethub_read:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Read the app's database password"
      secrethub_read:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Read the app's slack_token"
      secrethub_read:
        path: ansible-demo/infra/app/slack_token
      register: slack_token
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user.secret }}"
        DB_PASSWORD: "{{ db_pass.secret }}"
        SLACK_TOKEN: "{{ slack_token.secret }}"
```

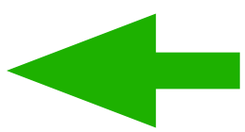


secrethub_read

```yaml
---
- hosts: db_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read database root user"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_user
      register: db_root_user
    - name: "Read database root password"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_pass
      register: db_root_pass
    - name: "Generate a database username for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Generate a database password for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user.secret }}"
        password: "{{ db_pass.secret }}"
        login_user: "{{ db_root_user.secret }}"
        login_password: "{{ db_root_pass.secret }}"
        state: present
        priv: characters:select
- hosts: web_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read the app's database user"
      secrethub_read:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Read the app's database password"
      secrethub_read:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Read the app's slack_token"
      secrethub_read:
        path: ansible-demo/infra/app/slack_token
      register: slack_token
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user.secret }}"
        DB_PASSWORD: "{{ db_pass.secret }}"
        SLACK_TOKEN: "{{ slack_token.secret }}"
```

secrethub_generate

```yaml
---
- hosts: db_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read database root user"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_user
      register: db_root_user
    - name: "Read database root password"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_pass
      register: db_root_pass
    - name: "Generate a database username for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Generate a database password for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user.secret }}"
        password: "{{ db_pass.secret }}"
        login_user: "{{ db_root_user.secret }}"
        login_password: "{{ db_root_pass.secret }}"
        state: present
        priv: characters:select
- hosts: web_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read the app's database user"
      secrethub_read:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Read the app's database password"
      secrethub_read:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Read the app's slack_token"
      secrethub_read:
        path: ansible-demo/infra/app/slack_token
      register: slack_token
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user.secret }}"
        DB_PASSWORD: "{{ db_pass.secret }}"
        SLACK_TOKEN: "{{ slack_token.secret }}"
```



<var>.secret

```yaml
---
- hosts: db_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read database root user"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_user
      register: db_root_user
    - name: "Read database root password"
      secrethub_read:
        path: ansible-demo/infra/postgres/root_pass
      register: db_root_pass
    - name: "Generate a database username for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Generate a database password for the app"
      secrethub_generate:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Create a database user for the app"
      postgresql_user:
        db: app
        name: "{{ db_user.secret }}"
        password: "{{ db_pass.secret }}"
        login_user: "{{ db_root_user.secret }}"
        login_password: "{{ db_root_pass.secret }}"
        state: present
        priv: characters:select
- hosts: web_servers
  environment:
    SECRETHUB_CREDENTIAL: "{{ lookup('env', 'SECRETHUB_CREDENTIAL') }}"
  tasks:
    - name: "Ensure the SecretHub CLI is installed"
      secrethub_cli:
    - name: "Read the app's database user"
      secrethub_read:
        path: ansible-demo/infra/app/db_user
      register: db_user
    - name: "Read the app's database password"
      secrethub_read:
        path: ansible-demo/infra/app/db_pass
      register: db_pass
    - name: "Read the app's slack_token"
      secrethub_read:
        path: ansible-demo/infra/app/slack_token
      register: slack_token
    - name: "Start the app with secrets"
      shell: "server"
      environment:
        DB_HOST: "{{ db_server }}"
        DB_NAME: app
        DB_USER: "{{ db_user.secret }}"
        DB_PASSWORD: "{{ db_pass.secret }}"
        SLACK_TOKEN: "{{ slack_token.secret }}"
```

admin access

dev access

Admin runs playbook

Dev runs playbook

Recap

SecretHub

github.com/secrethub/ansible-secrethub

QA