Requirements

How to accomplish

requirements

The correct tool of choice

Introduction

Solution to problem

Problems

- Solution has to be reproducible locally Exactly as on premises!
- Infrastructure As Code Data should be persistent!
- Time consuming custom configurations per VM!
-
- Minimal human interaction easy to use!

Solutions

- Introduce IAC elements to solve our problem!
- Fully automated procedure on every step!
- ADO pipeline either for Infrastructure or for provisioning VMs!
- Single file containing all configurations for all VMs!

Solution to problem

Problems

- Solution has to be reproducible locally Exactly as on premises!
- Infrastructure As Code Data should be persistent!
- Time consuming custom configurations per VM!
-
- Minimal human interaction easy to use!

Solutions

- Introduce IAC elements to solve our problem!
- Fully automated procedure on every step!
- ADO pipeline either for Infrastructure or for provisioning VMs!
- Single file containing all configurations for all VMs!

Solution to problem

Problems

- Solution has to be reproducible locally Exactly as on premises!
- Infrastructure As Code Data should be persistent!
- Time consuming custom configurations per VM!
- Minimal human interaction easy to use!

Solutions

- Introduce IAC elements to solve our problem!
- Fully automated procedure on every step!
- ADO pipeline either for Infrastructure or for provisioning VMs!
- Single file containing all configurations for all VMs!

Solution to problem

Problems

- Solution has to be reproducible locally Exactly as on premises!
- Infrastructure As Code Data should be persistent!
- Time consuming custom configurations per VM!
-
- Minimal human interaction easy to use!

Solutions

- Introduce IAC elements to solve our problem!
- Fully automated procedure on every step!
- ADO pipeline either for Infrastructure or for provisioning VMs!
- Single file containing all configurations for all VMs!

Requirements

How to accomplish

requirements

The correct tool of choice

Introduction

IAC

Possible Questions

- Why Ansible?
- Why Docker?
- Why Terraform?
- Why ADO pipelines?

Possible Answers

- Written in Python 2/3 developed and maintained by RedHat!
- It gives the ability to control the environment and use persistent volumes!
- We can create infinite amount of VMs and keep track of configurations through state file!
- We wanted to use a dynamic tool that is easy to use.

IAC

Possible Questions

- Why Ansible?
- Why Docker?
- Why Terraform?
- Why ADO pipelines?

Possible Answers

- Written in Python 2/3 developed and maintained by RedHat!
- It gives the ability to control the environment and use persistent volumes!
- We can create infinite amount of VMs and keep track of configurations through state file!
- We wanted to use a dynamic tool that is easy to use.

IAC

Possible Questions

- Why Ansible?
- Why Docker?
- Why Terraform?
- Why ADO pipelines?

Possible Answers

- Written in Python 2/3 developed and maintained by RedHat!
- It gives the ability to control the environment and use persistent volumes!
- We can create infinite amount of VMs and keep track of configurations through state file!
- We wanted to use a dynamic tool that is easy to use.

IAC

Possible Questions

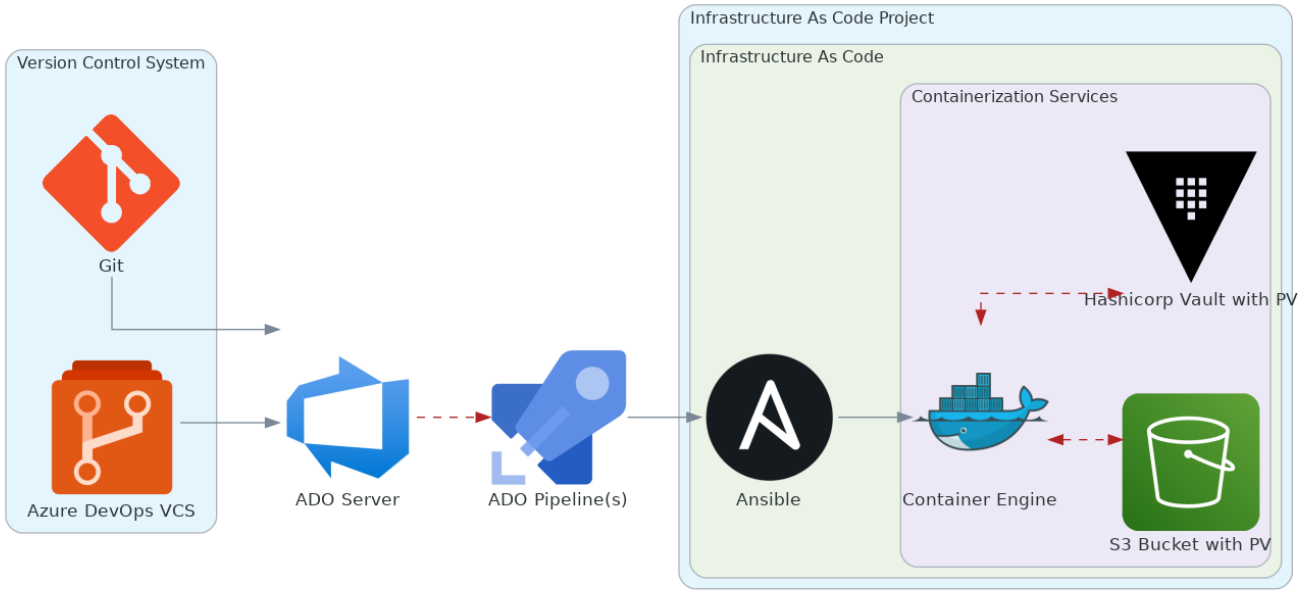
- Why Ansible?
- Why Docker?
- Why Terraform?
- Why ADO pipelines?

Possible Answers

- Written in Python 2/3 developed and maintained by RedHat!
- It gives the ability to control the environment and use persistent volumes!
- We can create infinite amount of VMs and keep track of configurations through state file!
- We wanted to use a dynamic tool that is easy to use.

High Level Desighn Infrastructure

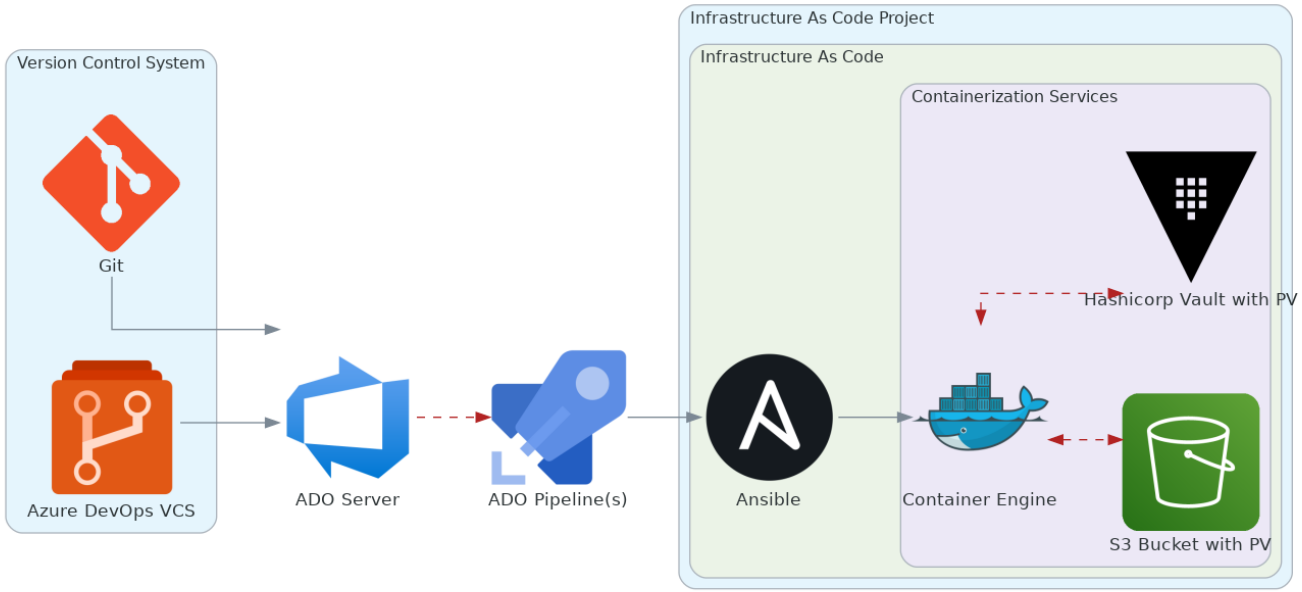
- Git / Azure DevOps VCS (where all code is stored)



High Level Design - Provisioning Infrastructure

High Level Desighn Infrastructure

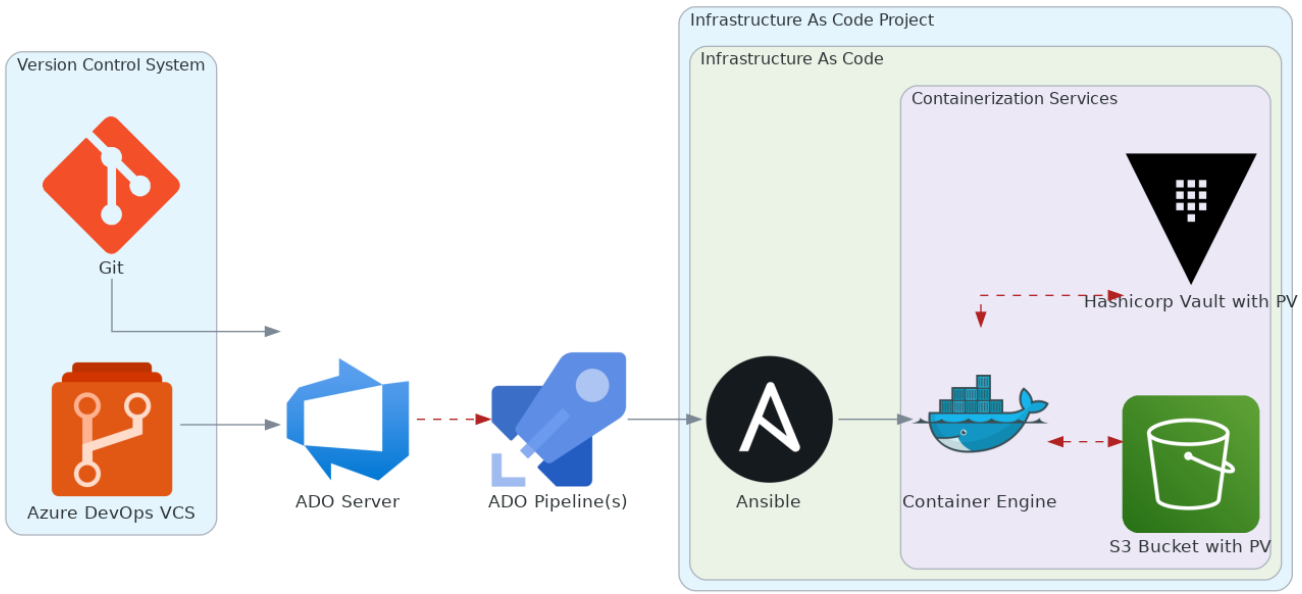
- Azure DevOps Server (triggerring infrastructure pipeline)



High Level Design - Provisioning Infrastructure

High Level Desighn Infrastructure

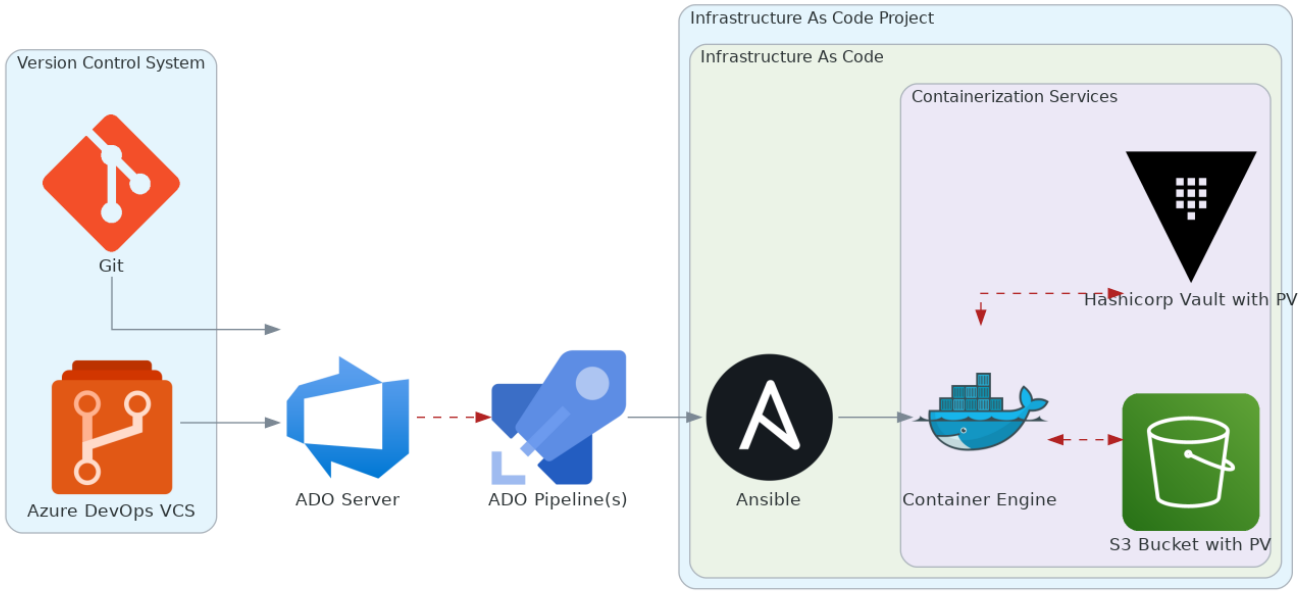
- Ansible (infrastructure provisioning tool and configurations)



High Level Design - Provisioning Infrastructure

High Level Desighn Infrastructure

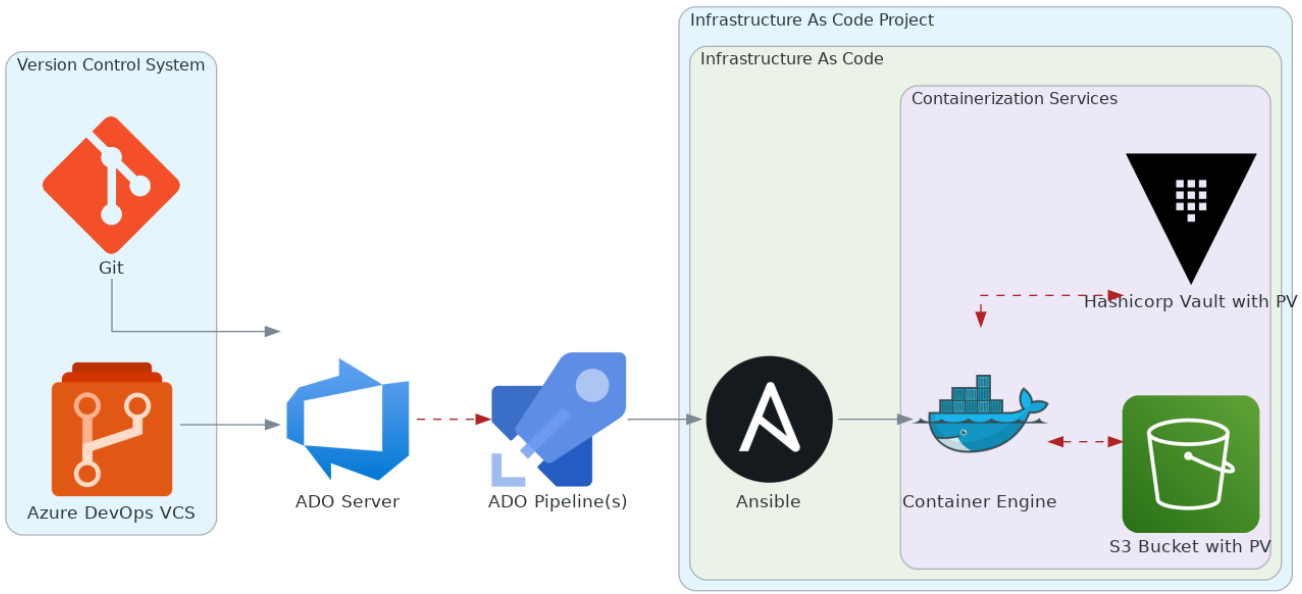
- Docker (containerization tool to apply containers)



High Level Design - Provisioning Infrastructure

High Level Desighn Infrastructure

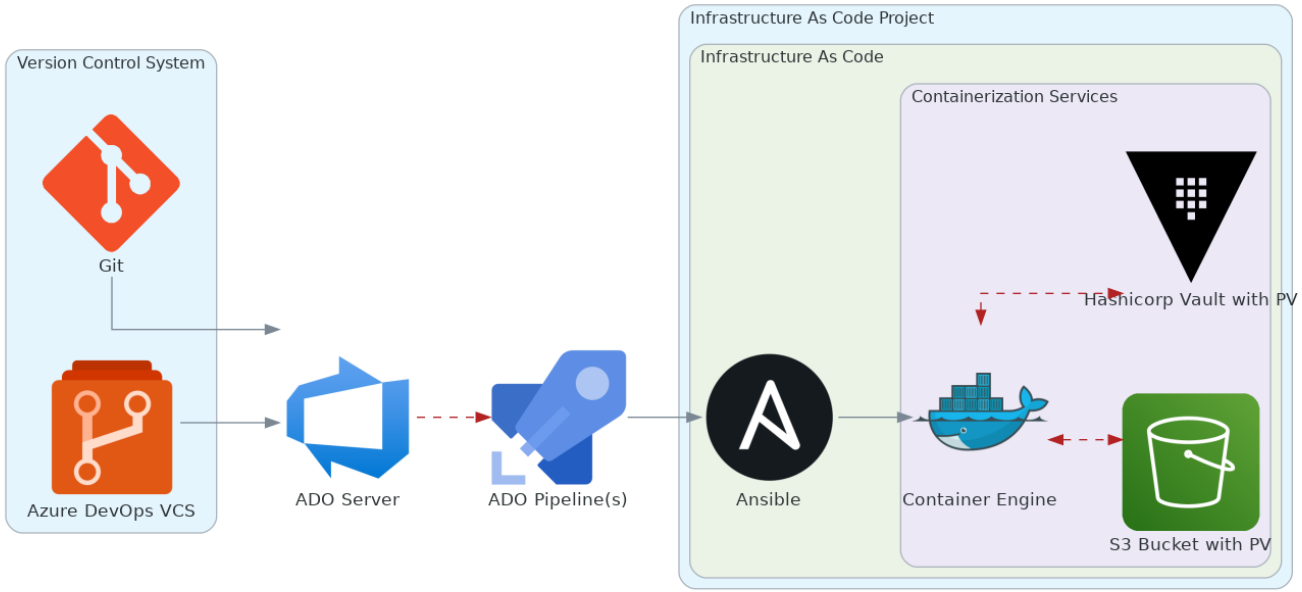
- Hashicorp Vault (storing sensitive data)



High Level Design - Provisioning Infrastructure

High Level Desighn Infrastructure

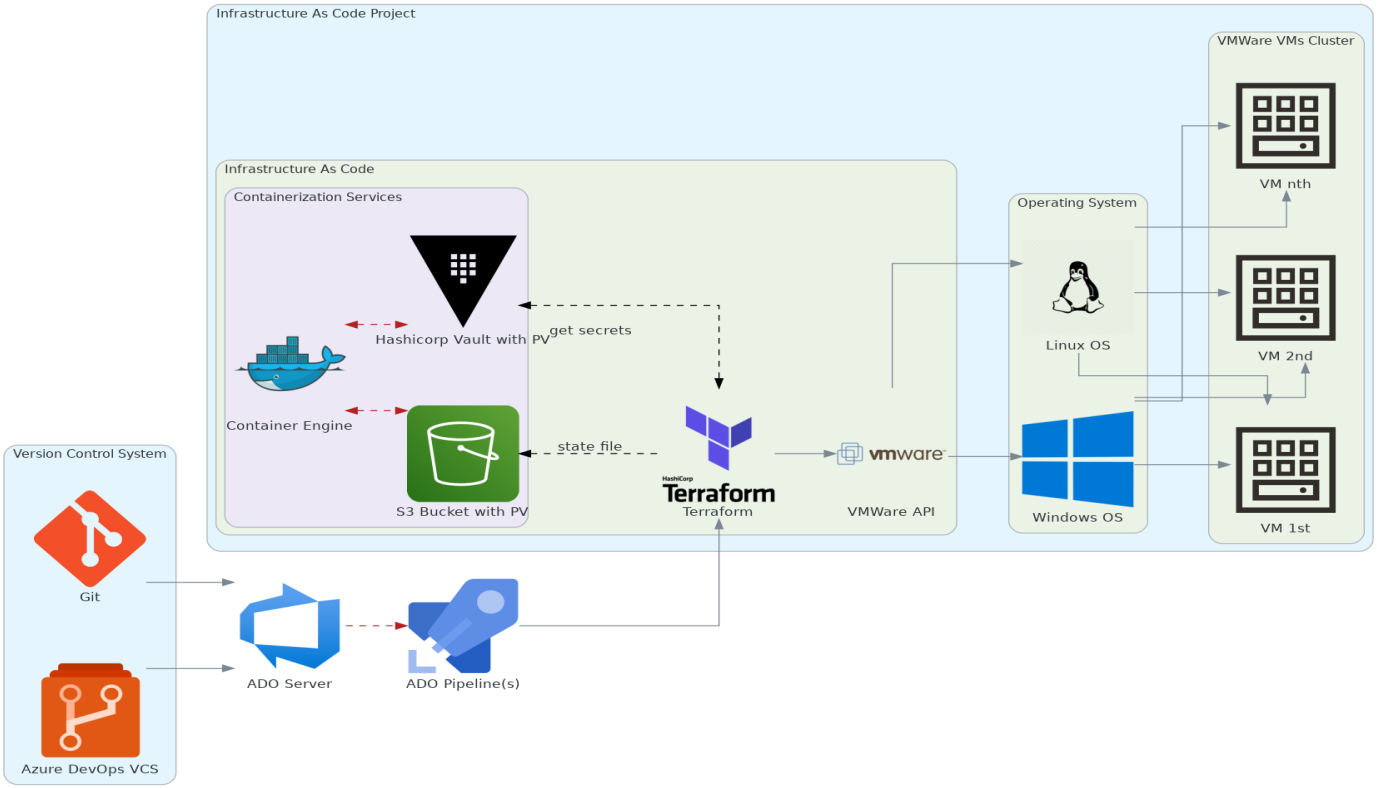
- s3 Bucket (used to store the terraform state file)



High Level Design - Provisioning Infrastructure

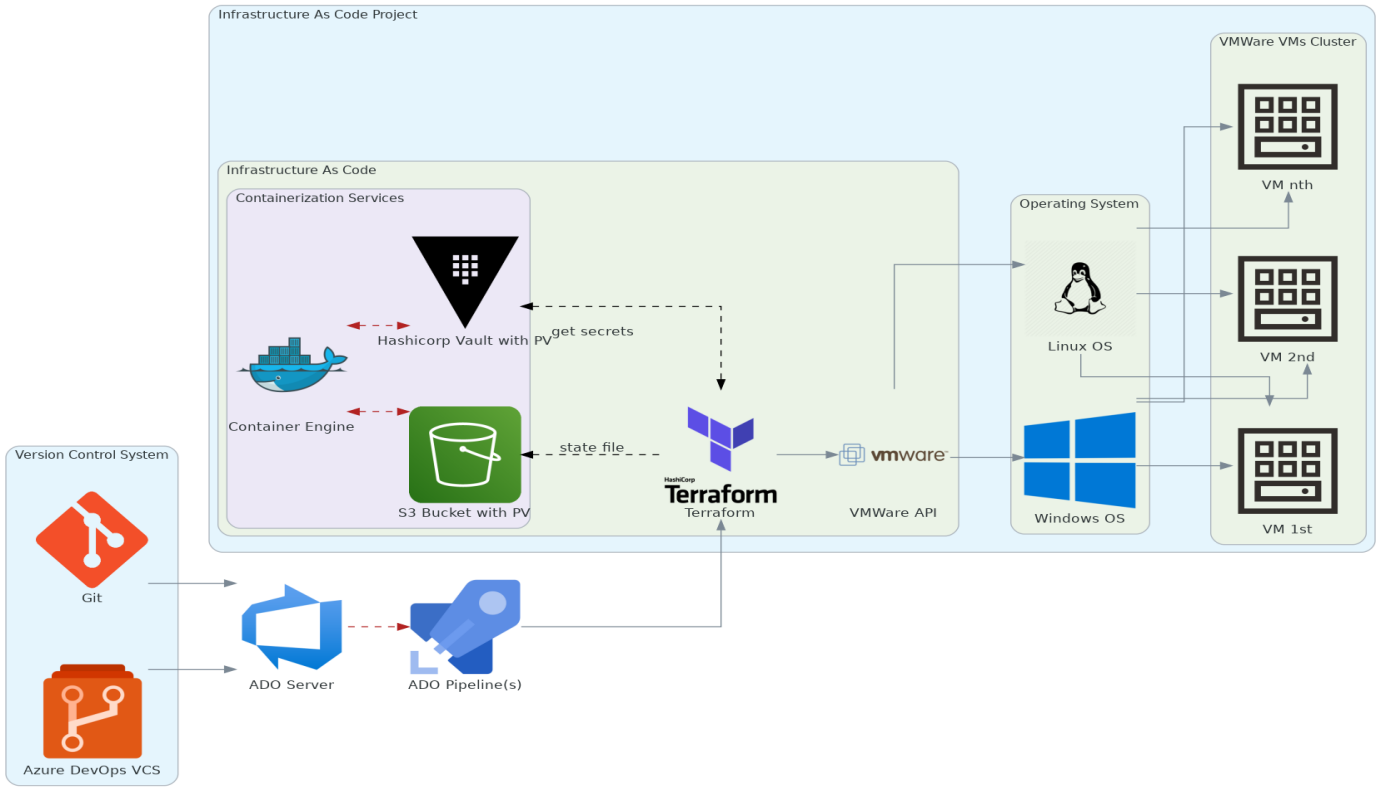
High Level Desighn Provisioning VMs

- Git / Azure DevOps VCS (where all code is stored)



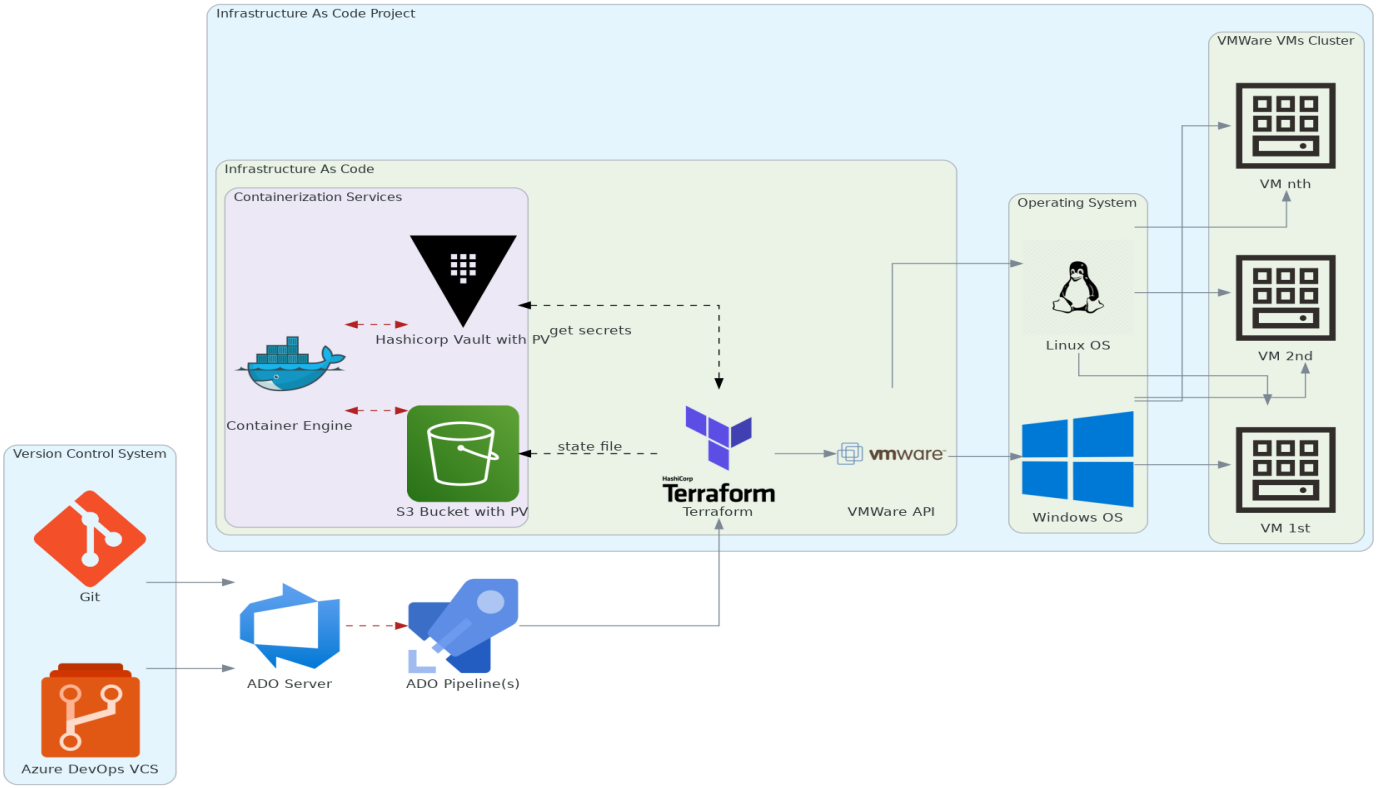
High Level Desighn Provisioning VMs

- Azure DevOps Server (triggerring infrastructure pipeline)



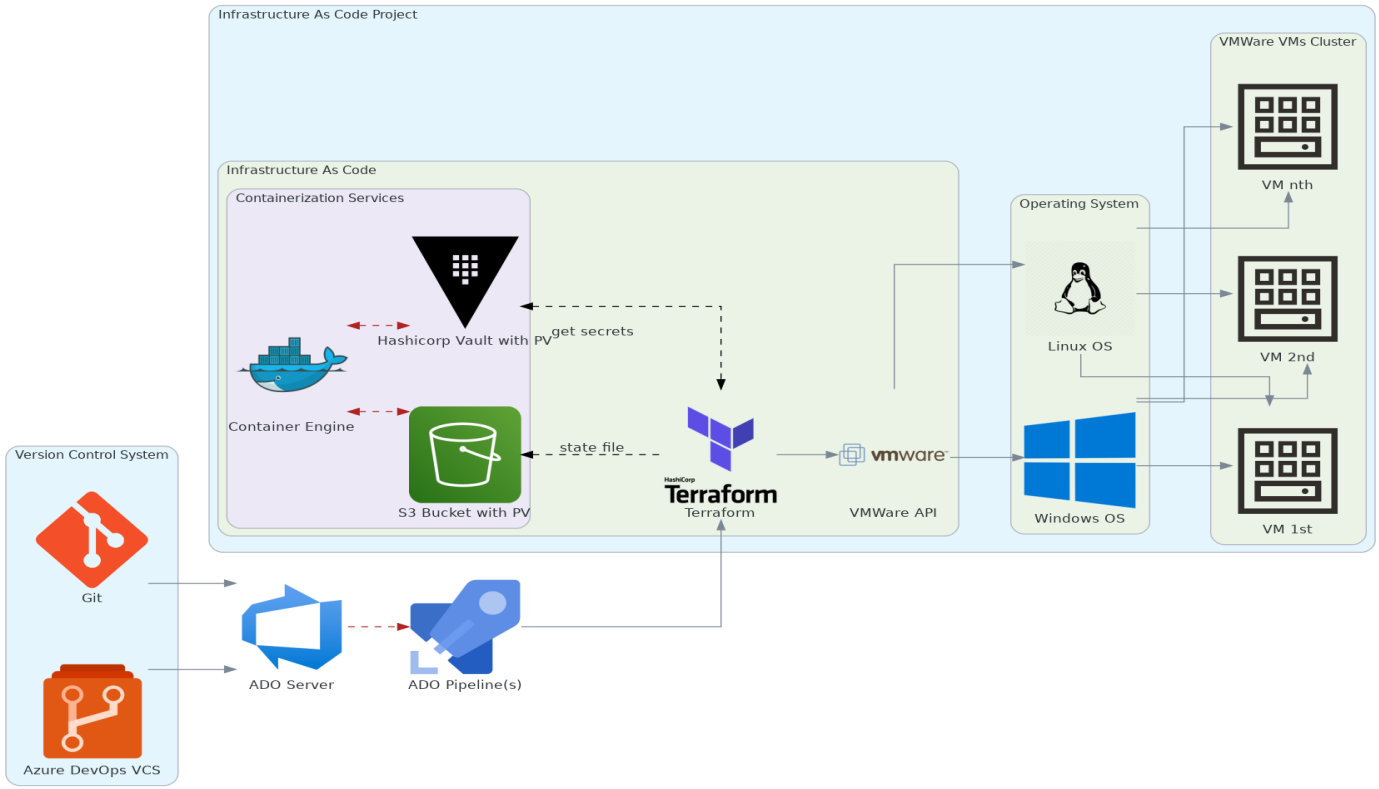
High Level Desighn Provisioning VMs

- Bash / Ansible (triggering terraform)



High Level Desighn Provisioning VMs

- Terraform (create VMs with custom configurations)



High Level Desighn Provisioning VMs

- VMware (provisioning VMware to create n number of VMs)