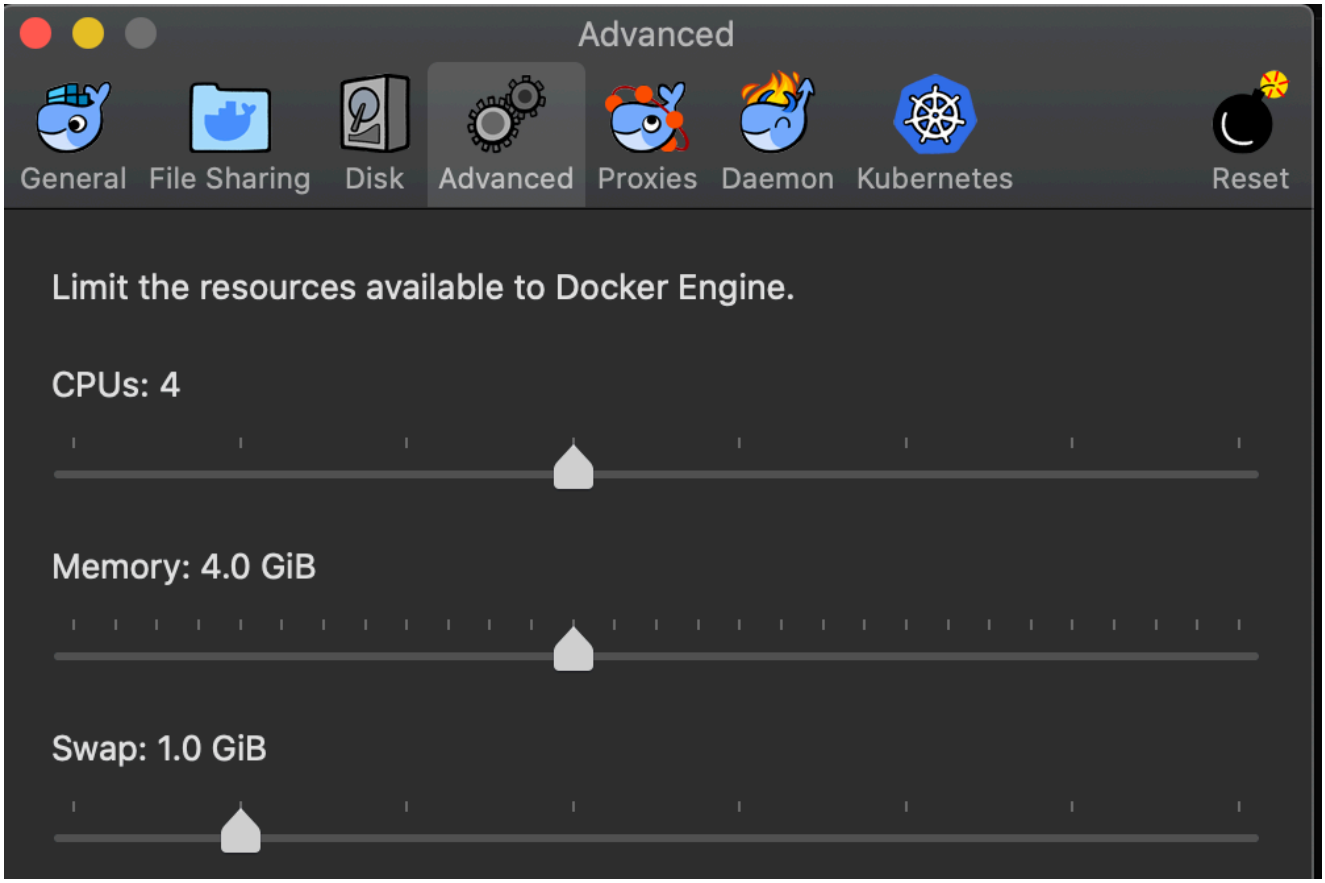
## Chapter 13 环境搭建要点

> 以下内容所涉及的基础服务，默认你已经可以自主搭建，此处仅介绍关键细节。

### 本机启动devops服务

> 在执行docker-compose指令之前，请先确保本地的docker服务的资源分配至少 2c4g




### Harbor

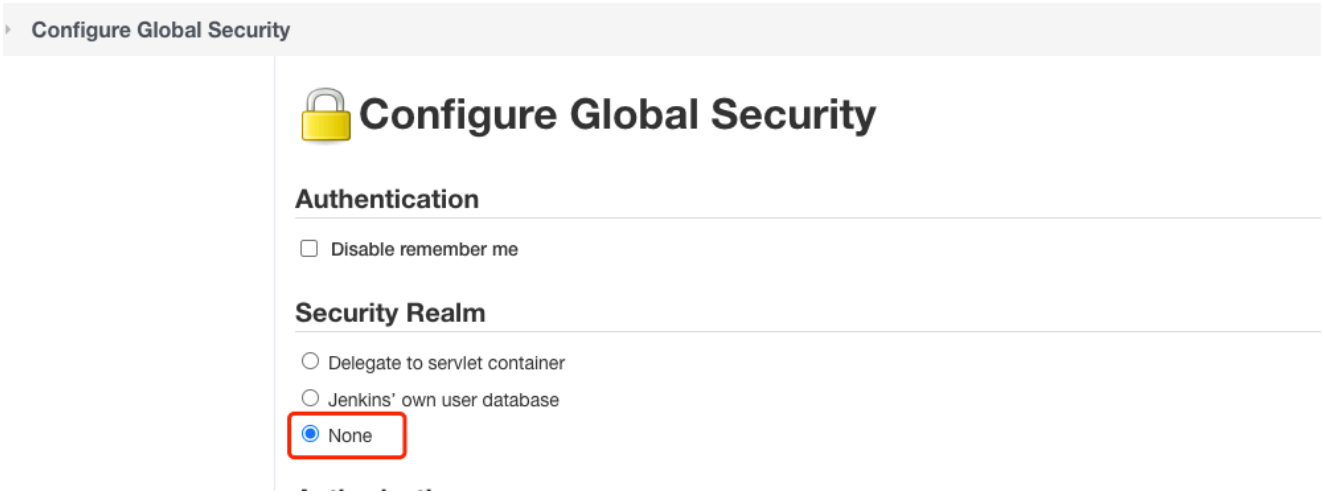
> 你有一台Harbor服务器，使用 Http 协议，默认 80 端口可以访问；
>
> 你有一个域名可以解析到此服务器上。
>
> 你的项目访问级别请勾选公开


### Gitlab

> 你有一台Gitlab服务器，使用admin账号登录，并添加 SSH Keys 。
>
> SSH Key 是你的Jenkins服务器上的 ssh公钥 【位于 .ssh/id_rsa.pub 】。

### Jenkins

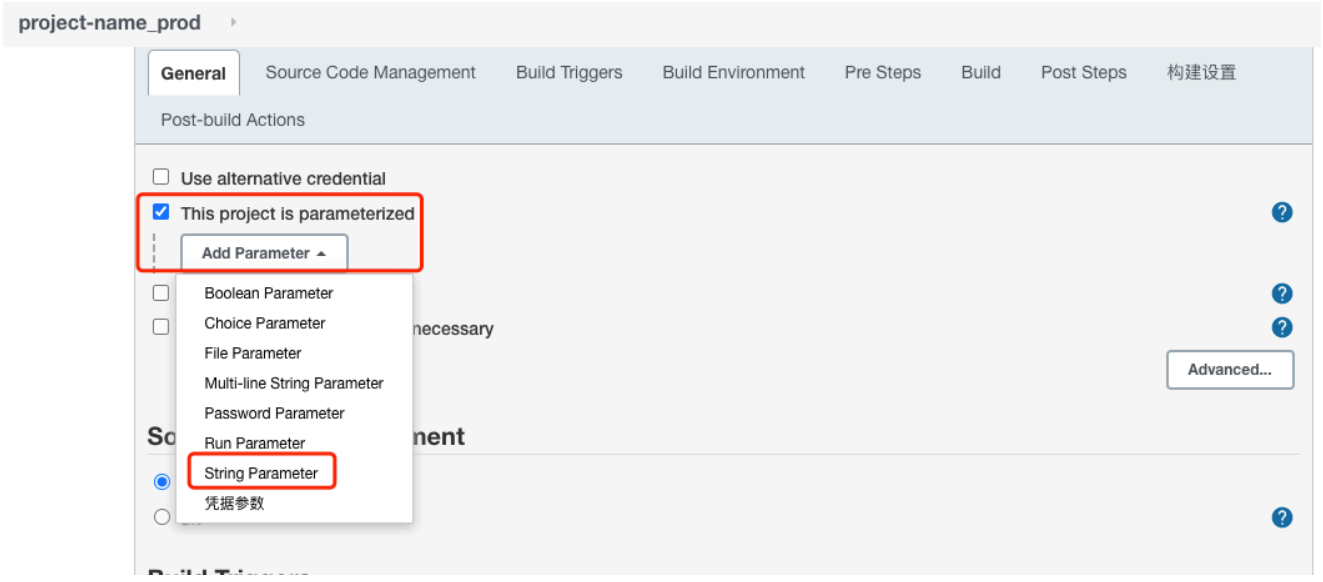
> 你有一台Jenkins服务器；
>
> 这台机器上安装了 docker， 并且使用过命令 `docker login` 登录了你的 Harbor 服务器；
>
> 这样你就可以将打包好的docker镜像推送到你的 Harbor服务器上。
>
> **配置你的Jenkins Job**
>
> 关闭跨站伪造保护，参考文章： https://blog.csdn.net/ccc_bigdata/article/details/108080084
>
> 配置全局安全：
>
> 安全领域设置为 none；



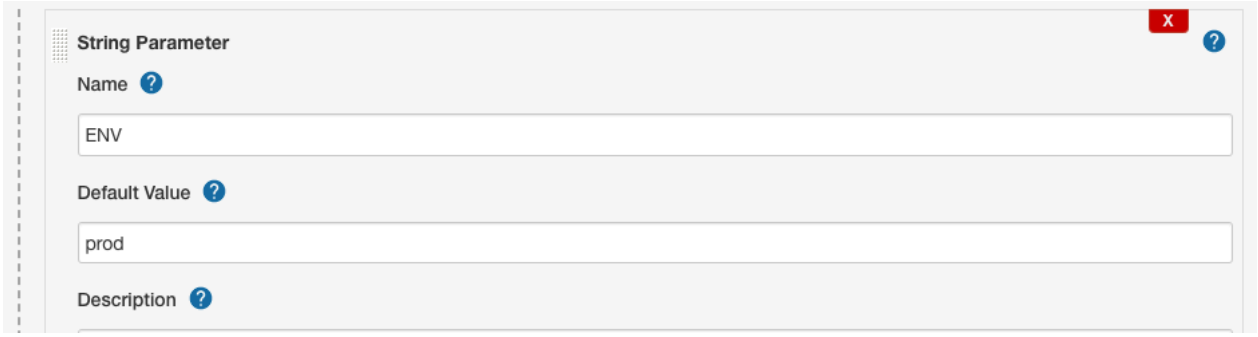

> 新建：

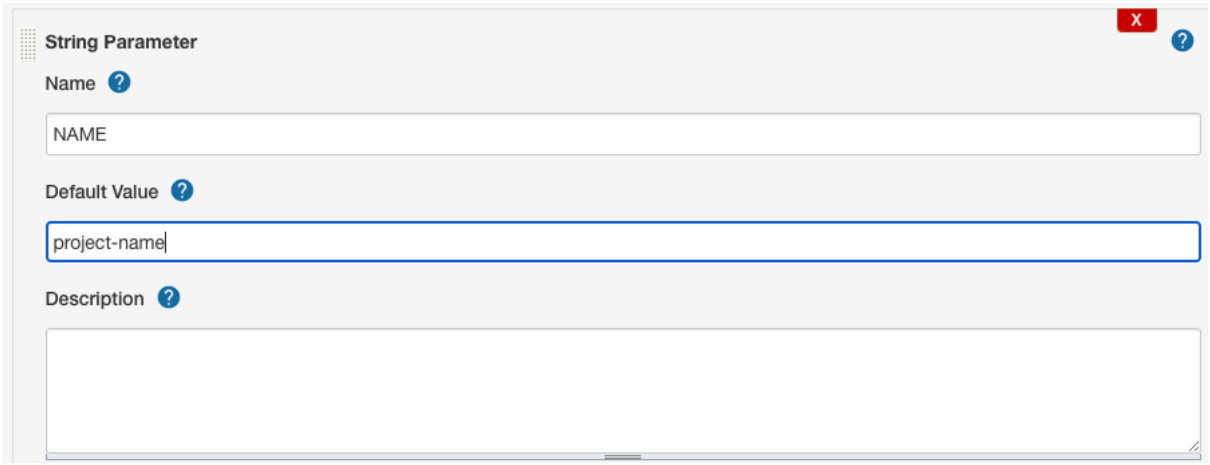

配置：

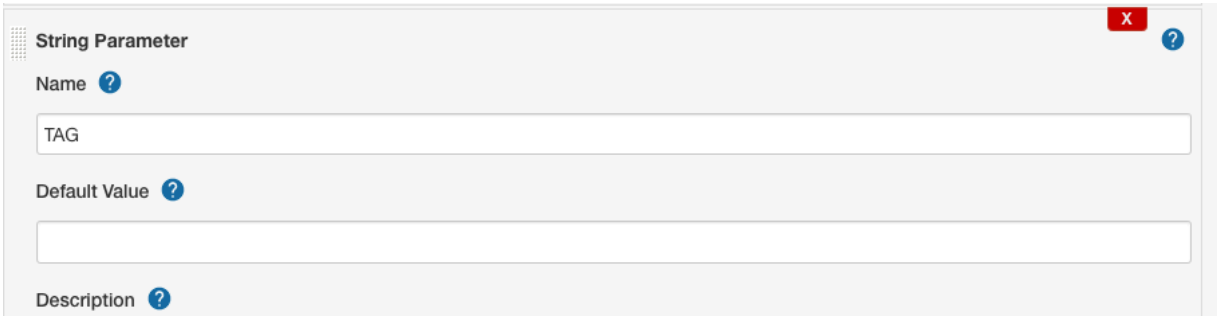
配置参数化构建过程，此处是devops代码中的入参约定。



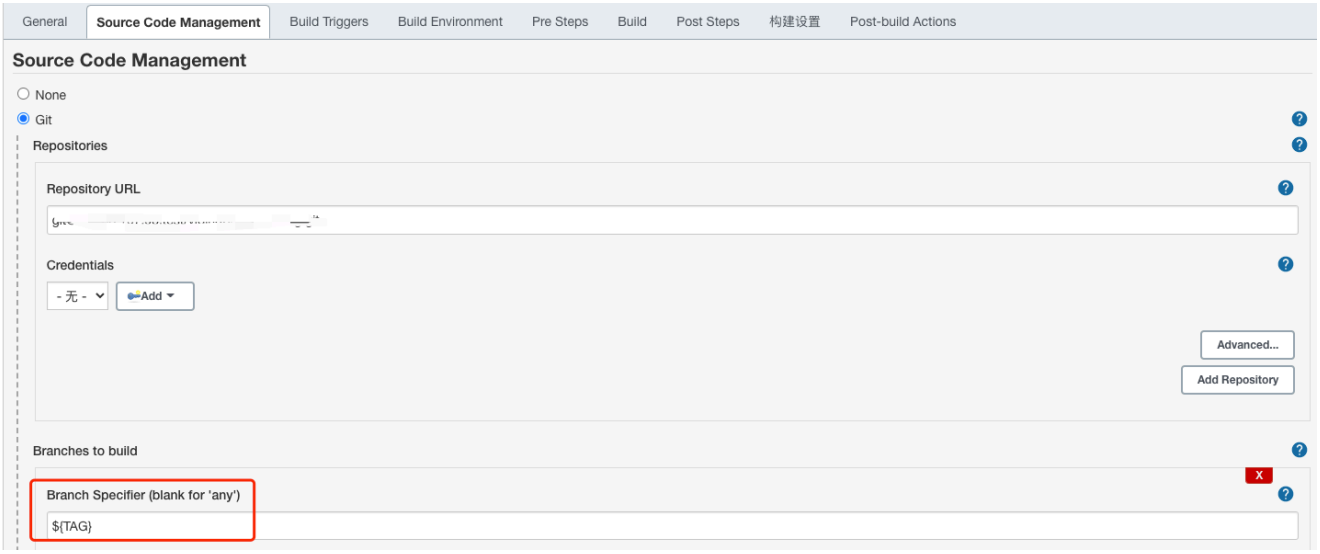

需要添加 3 个参数， ENV 、NAME、TAG






配置源代码管理，分支选择部分填写 ${TAG}，此处是devops代码中的入参约定。




以上Job配置就结束了，build过程部分，需要读者按照自己的代码编排来修改。

### 你还需要一台服务器用来部署你在devops平台新建的应用

你有一台服务器，对外开放 `TCP端口：4789` 和 `TCP端口：8080`

这台机器上安装了 docker， 配置 `TCP端口：4789` 监听 docker.socket ;

举例：

如果这台是centos系统的服务器，并且你是使用 `systemctl start doker` 来启动；

那么你可以修改docker.service文件 `/usr/lib/systemd/system/docker.service`

`ExecStart=/usr/bin/dockerd --insecure-registry harbor.xxxx.cn -H unix:///var/run/docker.sock -H tcp://0.0.0.0:4789 -H fd:// --co`

上面的 `--insecure-registry harbor.xxxx.cn` 配置了你的harbor仓库地址

`-H unix:///var/run/docker.sock -H tcp://0.0.0.0:4789 -H fd:// --containerd=/run/containerd/containerd.sock`

配置了你的unix 套接字文件/ tcp 监听端口和 fd 文件描述符

为了配合devops服务的健康检查，你的应用需要对外提供健康检查接口：

例如 springboot 应用:

application.properties

```
server.servlet.context-path=/project-name
spring.application.name=project-name
server.port=8080
```

controller

```
   @GetMapping("/actuator/health")
    public Object health() {
        Map<String, Object> rtn = new HashMap<>();
        rtn.put("status", "UP");
        return rtn;
    }
```

例如 前端应用在 `nginx.conf` 中配置：

```
    ...

     listen       80 default;
     listen       8080 default;

     ...

     location = /project-name/actuator/health {
           return 200 '{"status":"UP"}';
     }
```