In [1]:
```
import numpy as np
import math
import matplotlib.pyplot as plt
```

In [2]:
```
X = [[1,1],[1,-1],[-1,1],[-1,-1]]
X = np.array(X)

Y = np.array([1,1,1,-1])
```

In [3]:
```
def func(X,y,w):
    n,d = X.shape
    X = np.append(X,np.ones((n,1)),axis = 1)
    f = 0.0
    for i in range(n):
        inner = 1-np.inner(X[i],w)*y[i]
        if inner > 0:
            f += inner
    return f/n
```

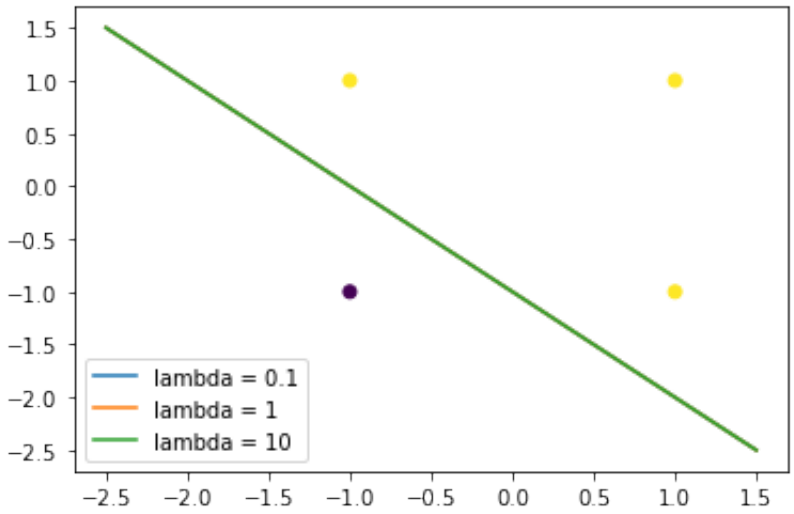
In [4]:
```
def subgrad(X,y,w):
    n,d = X.shape
    X = np.append(X,np.ones((n,1)),axis = 1)
    g = np.zeros(d+1)
    for i in range(n):
        inner = 1-np.inner(X[i],w)*y[i]
        if inner > 0:
            g +=-y[i]*X[i]
    return g/n
```

In [5]:
```
def dualavg(X,y,w,s = 0, c = 1, eta=1, maxiter=1000):
    for t in range(maxiter):
        f = func(X,y,w)
        g = subgrad(X,y,w)
        eta_t = eta/math.sqrt(t+1)
        s = s+g
        z = -eta_t*s

        if np.linalg.norm(z) == 0:
            w = 0*z
        else:
            w = min(c,np.linalg.norm(z))*z/np.linalg.norm(z)


        #w = c/np.linalg.norm(z)*z
    return w
```

In [6]:
```
X = [[1,1],[1,-1],[-1,1],[-1,-1]]
X = np.array(X)

Y = np.array([1,1,1,-1])
w = np.array([0,0,0])
```

In [7]:
```
w_01 = dualavg(X,Y,w,s = 0, c = 0.1, eta=1, maxiter=1000)
w_1 = dualavg(X,Y,w,s = 0, c = 1, eta=1, maxiter=1000)
w_10 = dualavg(X,Y,w,s = 0, c = 10, eta=1, maxiter=1000)
```

In [8]:
```
x_point = np.array([1,1,-1,-1])
y_point = np.array([1,-1,1,-1])
label = Y
plt.scatter(x_point, y_point, c = label)
x = np.linspace(-2.5,1.5,120)
y_01 = -w_01[0]/w_01[2] -w_01[1]/w_01[2] *x
y_1 = -w_1[0]/w_1[2] -w_1[1]/w_1[2] *x
y_10 = -w_10[0]/w_10[2] -w_10[1]/w_10[2] *x
plt.plot(x,y_01, label = 'lambda = 0.1')
plt.plot(x,y_1, label = 'lambda = 1')
plt.plot(x,y_10, label = 'lambda = 10')
plt.gca().legend(('lambda = 0.1','lambda = 1','lambda = 10'))
plt.show()
```



In [ ]:
```

```