```
from ase.calculators.emt import EMT
from ase.neb import NEB
from ase.optimize import BFGS
from ase.io import write
from ase.build import fcc100, add_adsorbate
from ase.constraints import FixAtoms

# 2x2-Al(001) surface with 3 layers and an
# Au atom adsorbed in a hollow site:
initial = fcc100('Al', size=(2, 2, 3))
add_adsorbate(initial, 'Au', 1.7, 'hollow')
initial.center(axis=2, vacuum=4.0)

# Fix second and third layers:
mask = [atom.tag > 1 for atom in initial]
initial.set_constraint(FixAtoms(mask=mask))

# Initial state:
initial.calc = EMT()
qn = BFGS(initial)
qn.run(fmax=0.05)

# Final state:
final = initial.copy()
final[-1].x += final.get_cell()[0, 0] / 2
final.calc = EMT()
qn = BFGS(final)
qn.run(fmax=0.05)

images = [initial]
for i in range(3):
    image = initial.copy()
    image.calc = EMT()
    images.append(image)
images.append(final)

# NEB object with new interpolated images
neb = NEB(images)
neb.interpolate()
qn = BFGS(neb, trajectory='neb.traj')
qn.run(fmax=0.05)

write('output.traj', images)
```