

In [1]:

```
from qiskit import *  
from qiskit.visualization import plot_histogram  
from qiskit.tools.monitor import job_monitor
```

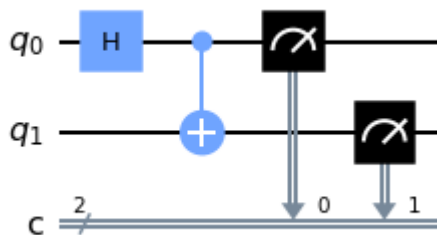
In [ ]:

```
### ----- Plot Histogram ----- ###
```

In [3]:

```
# 2-qubit Bell state  
circuit = QuantumCircuit(2, 2)  
circuit.h(0)  
circuit.cx(0, 1)  
  
meas = QuantumCircuit(2, 2)  
meas.measure(range(2), range(2))  
  
total = circuit + meas  
total.draw('mpl')
```

Out[3]:

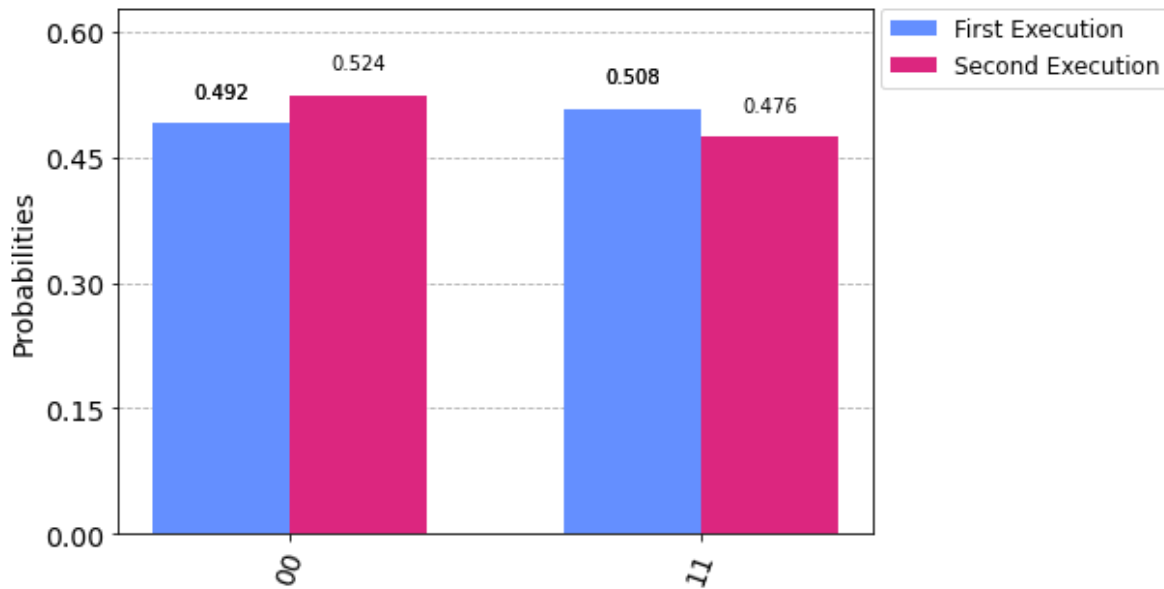


In [9]:

```
simulator = Aer.get_backend('qasm_simulator')
first_result = execute(total, simulator, shots = 1000).result()
second_result = execute(total, simulator, shots = 1000).result()
first_counts = first_result.get_counts(total)
second_counts = second_result.get_counts(total)

legend = ['First Execution', 'Second Execution']
plot_histogram([first_counts, second_counts], legend=legend)
```

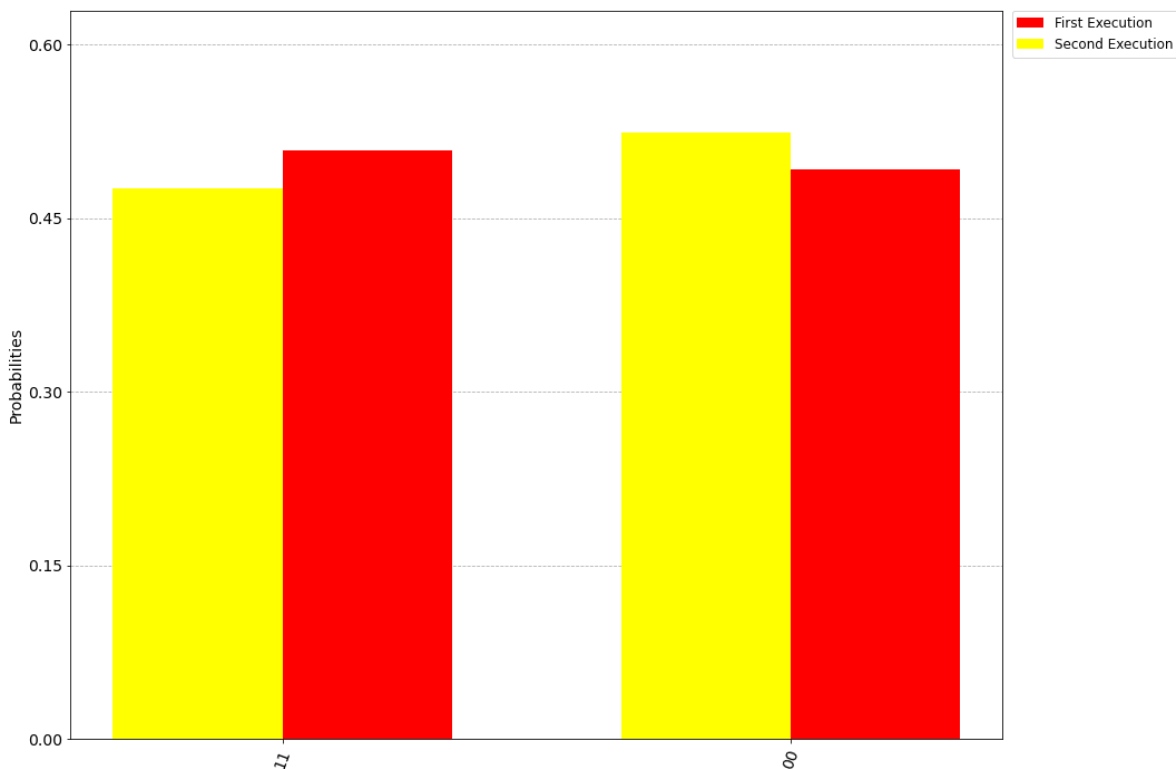
Out[9]:



In [20]:

```
plot_histogram([first_counts, second_counts], legend=legend, sort='desc', figsize=(15,12),
               color=['red', 'yellow'], bar_labels=False)
```

Out[20]:



In [21]:

```
### ----- Plot State ----- ###
```

In [22]:

```
from qiskit.visualization import plot_state_city, plot_bloch_multivector
from qiskit.visualization import plot_state_paulivec, plot_state_hinton
from qiskit.visualization import plot_state_qsphere
```

In [23]:

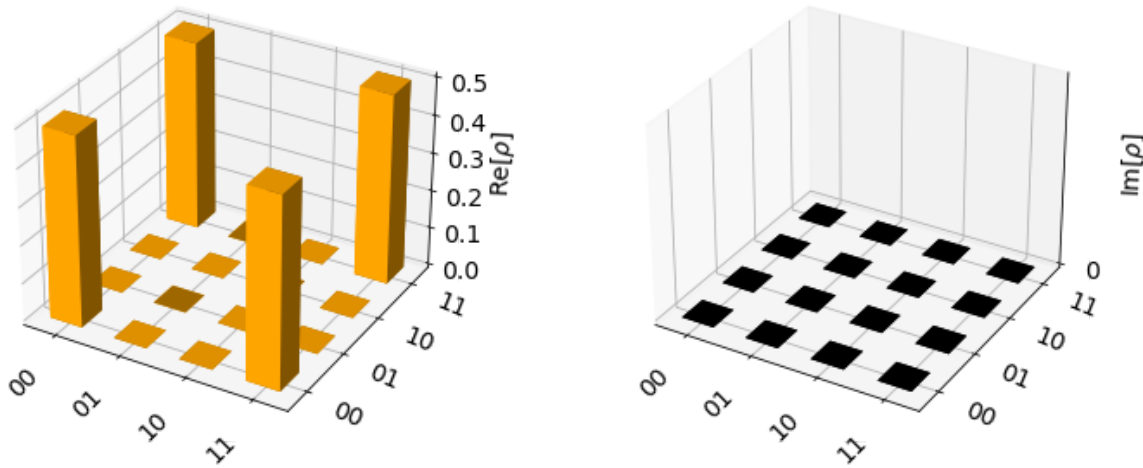
```
backend = BasicAer.get_backend('statevector_simulator')
result = execute(circuit, backend).result()
psi = result.get_statevector(circuit)
```

In [32]:

```
plot_state_city(psi, title="My State City", figsize = (12,5), color=['orange', 'black'])
```

Out[32]:

My State City

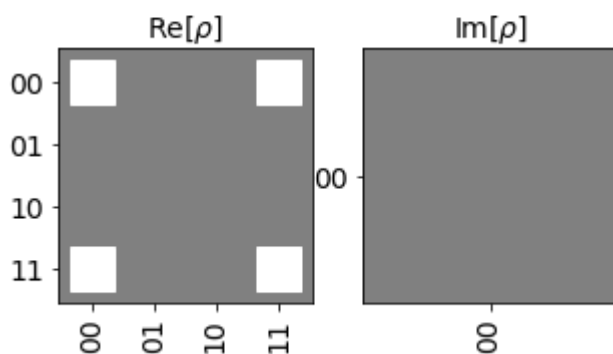


In [34]:

```
plot_state_hinton(psi, title="My State Hinton", figsize = (5,5))
```

Out[34]:

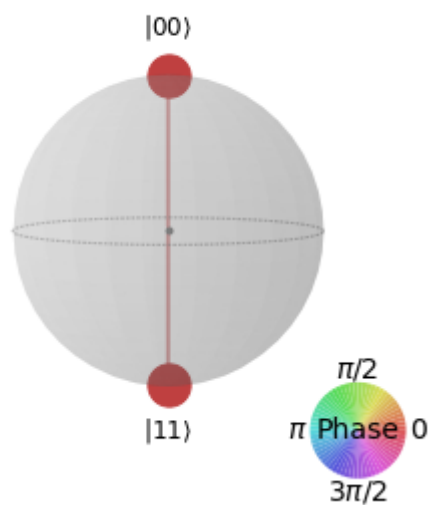
My State Hinton



In [57]:

```
plot_state_qsphere(psi, figsize = (5,5))
```

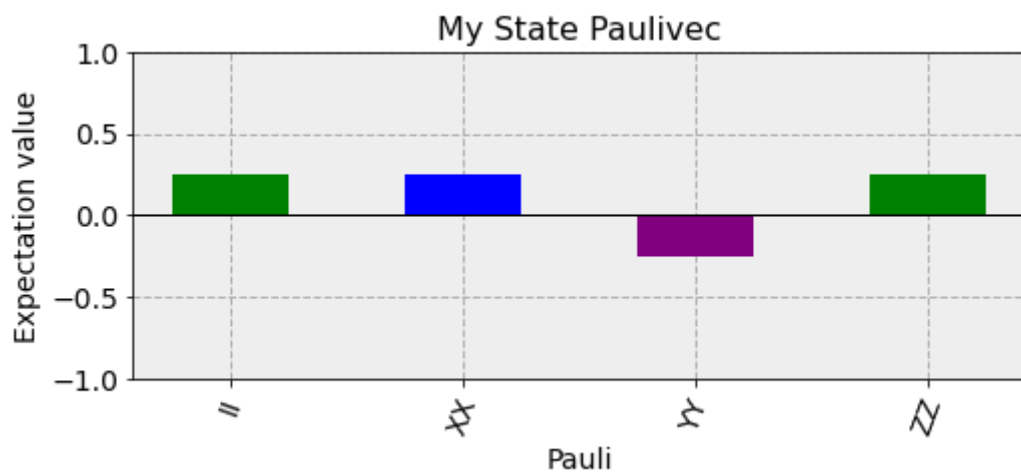
Out[57]:



In [38]:

```
plot_state_paulivec(psi, title="My State Paulivec", figsize = (8,3), color=['green', 'blue'])
```

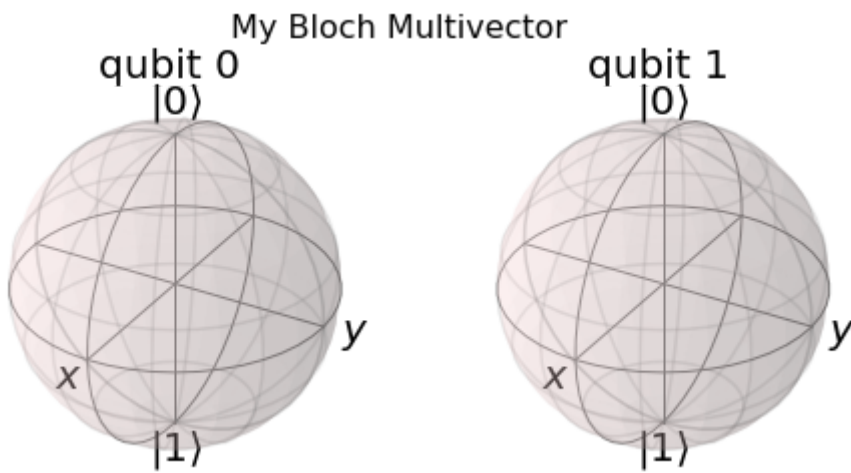
Out[38]:



In [42]:

```
plot_bloch_multivector(psi, title="My Bloch Multivector", figsize = (8,3))
```

Out[42]:



In [ ]:

```
### ----- Interactive State Plots for Jupyter Notebooks ----- ###
```

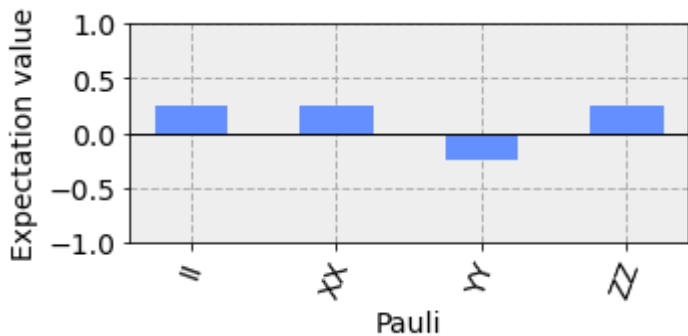
In [58]:

```
from qiskit.tools.visualization import iplot_state_paulivec
# Generate an interactive pauli vector plot
iplot_state_paulivec(psi, figsize = (5, 2))
```

<ipython-input-58-270037e15198>:3: DeprecationWarning: The iplot\_state\_pauli vec function is deprecated and will be removed in a future release. The host ed code this depended on no longer exists so this is falling back to use the matplotlib visualizations. qiskit.visualization.plot\_state\_paulivec should b e used directly moving forward.

```
iplot_state_paulivec(psi, figsize = (5, 2))
```

Out[58]:



In [ ]:

```
### ----- Plot Bloch Vector ----- ###
```

In [56]:

```
from qiskit.visualization import plot_bloch_vector
from math import sqrt
plot_bloch_vector([-0.5 / sqrt(1.5), 1.0 / sqrt(1.5), 0.5 / sqrt(1.5)], title='My Bloch Spher
```

Out[56]:

