In [43]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [44]:
```
## Azure
# vm_table_data_link = "https://azurepublicdatasettraces.blob.core.windows.r
# vm_table_data = pd.read_csv(vm_table_data_link, compression = "gzip")
vm_table_data = pd.read_csv("vm_table_data_azure.csv")
vm_table_data = vm_table_data[['vm_id', 'vm_creation_timestamp', 'vm_deletio
                               'avg_cpu', 'p95_max_cpu', 'vm_category', 'vm_
vm_table_data.sort_values(by=['vm_creation_timestamp', 'vm_deletion_timestar
vm_table_data['start_hour'] = vm_table_data['vm_creation_timestamp'] // 3600
vm_table_data['end_hour'] = vm_table_data['vm_deletion_timestamp'] // 3600
vm_table_data
```

Out[44]:

	vm_id	vm_creation_timestamp	v
0	rKggHO/04j31UFy65mDTwtjdMQL/G03xWfl3xGeiilB4/W...	424500	
1	YrR8gPtBmfNaOdnNEW5lf1SdTqQgGQHEnLHGPjySt53bKW...	1133100	
2	xzQ++JF1UAkh70CDhmzkiOo+DQn+E2TLErCFKEmSswv1pl...	0	
3	vZEivnhabRmImDr+JqKqZnpIM3WxtypwoxjfjnklR/idyR...	228300	
4	MqvcZ6Au5oul6if56MJHmoSqHtX8oRv0dPkaxCld3aUcr1...	1395600	
...	...	...	
2695542	CfZn37rcUvC4sVjWik6ylutOzNfno3c4dg6eloqpaSE8P8...	141300	
2695543	D5jsQPZSlO+KakH/yp7bPV5hrKPhyxMrh0WAzMVarUDFpf...	0	
2695544	FP9Lf4/jjWgWl9HS80x1NoeFwOPhLQo1ACPgqjtBF3+z9L...	1744800	
2695545	thW1eyboLMZJy6GgeClpLllRfqn0q7JgDYarlC9Jm5tg6i...	387000	
2695546	TqMw/UmeYGTWCvWBdL+yIw6+Pz3Vzj/OgIZNepu3scPY94...	1300500	

2695547 rows × 11 columns

In [45]:
```
hourly_data = (vm_table_data.assign(hour=lambda df: df.apply(lambda row: ran
hourly_data
```

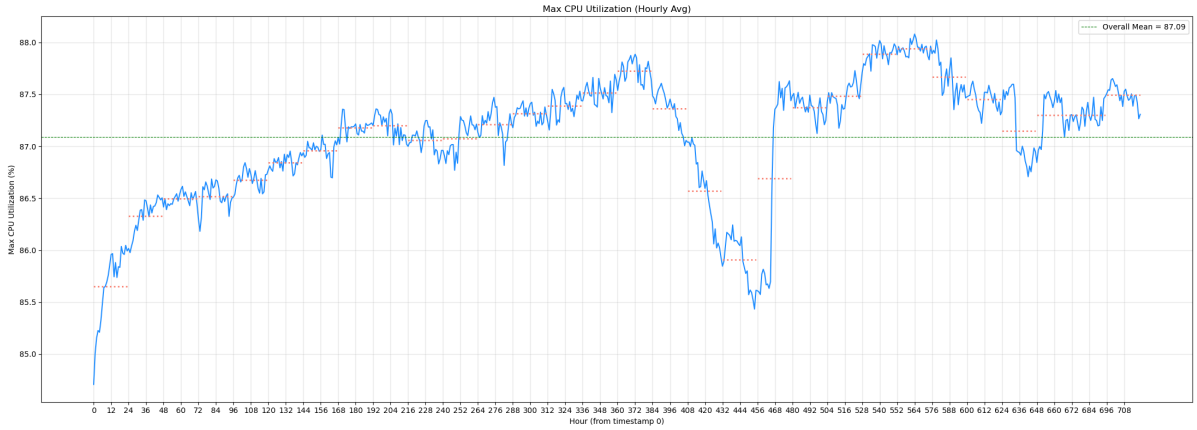
Out[45]:

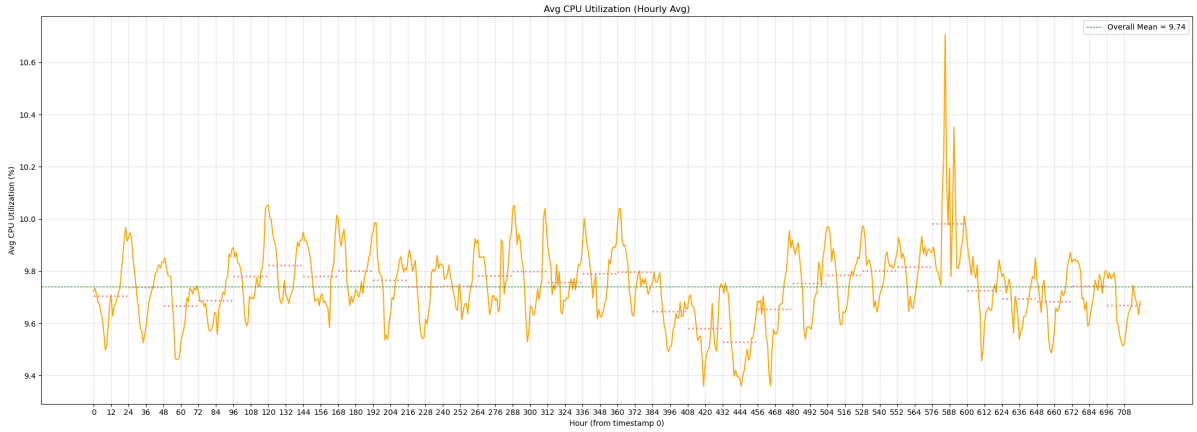
	vm_id	vm_creation_timestamp	v
0	rKggHO/04j31UFy65mDTwtjdMQL/G03xWfl3xGeiilB4/W...	424500	
0	rKggHO/04j31UFy65mDTwtjdMQL/G03xWfl3xGeiilB4/W...	424500	
1	YrR8gPtBmfNaOdnNEW5lf1SdTqQgGQHEnLHGPjySt53bKW...	1133100	
2	xzQ++JF1UAkh70CDhmzkiOo+DQn+E2TLErCFKEmSswv1pl...	0	
2	xzQ++JF1UAkh70CDhmzkiOo+DQn+E2TLErCFKEmSswv1pl...	0	
...	...	...	
2695544	FP9Lf4/jjWgWI9HS80x1NoeFwOPhLQo1ACPgqjtBF3+z9L...	1744800	
2695544	FP9Lf4/jjWgWI9HS80x1NoeFwOPhLQo1ACPgqjtBF3+z9L...	1744800	
2695544	FP9Lf4/jjWgWI9HS80x1NoeFwOPhLQo1ACPgqjtBF3+z9L...	1744800	
2695545	thW1eyboLMZJy6GgeCIpLllRfqn0q7JgDYarlC9Jm5tg6i...	387000	
2695546	TqMw/UmeYGTWCvWBdL+yIw6+Pz3Vzj/OgIZNepu3scPY94...	1300500	

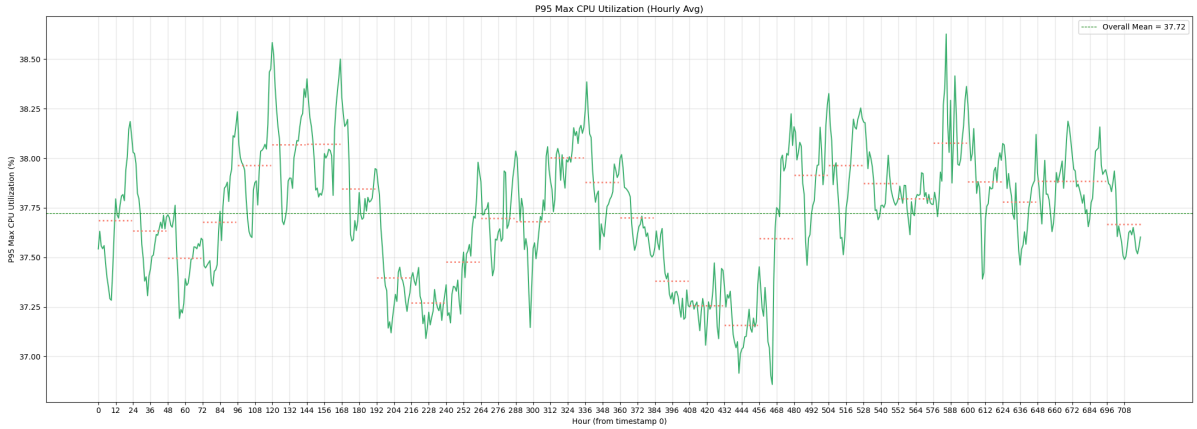
164927669 rows × 12 columns

In [46]:

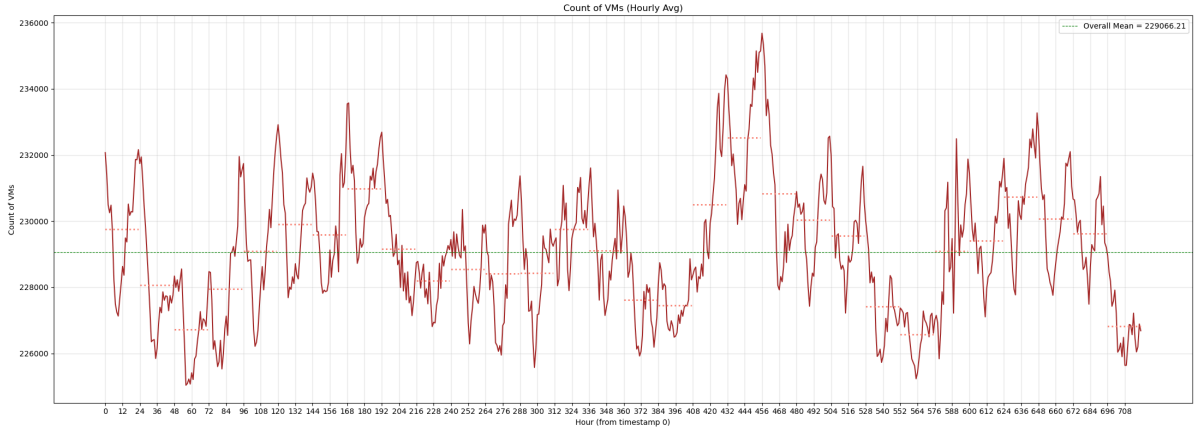
```
# Calculating hourly averages
hourly_CPU_avg = hourly_data.groupby('hour')[['max_cpu', 'avg_cpu', 'p95_max

def plot_with_24h_means(data, col_name, color, ylabel, title):
    plt.figure(figsize=(22, 8))
    plt.plot(data['hour'], data[col_name], color=color, linewidth=1.5)
    overall_mean = data[col_name].mean()
    plt.axhline(y=overall_mean, color='green', linestyle='--', linewidth=0.8

    max_hour = data['hour'].max()
    for start in range(0, max_hour + 1, 24):
        end = start + 24
        block = data[(data['hour'] >= start) & (data['hour'] < end)]
        if not block.empty:
            block_mean = block[col_name].mean()
            plt.hlines(y=block_mean, xmin=start, xmax=min(end - 1, max_hour)

    plt.title(title)
    plt.xlabel('Hour (from timestamp 0)')
    plt.ylabel(ylabel)
    plt.xticks(ticks=np.arange(0, max_hour + 1, 12))
    plt.grid(True, alpha=0.3)
    plt.legend()
    plt.tight_layout()
    plt.savefig(f'azure_plots/Azure_{col_name}_plot')
    plt.show()

# Plotting CPU Utilization
plot_with_24h_means(data=hourly_CPU_avg, col_name='max_cpu', color='dodgerb
plot_with_24h_means(data=hourly_CPU_avg, col_name='avg_cpu', color='orange',
plot_with_24h_means(data=hourly_CPU_avg, col_name='p95_max_cpu', color='medi
```





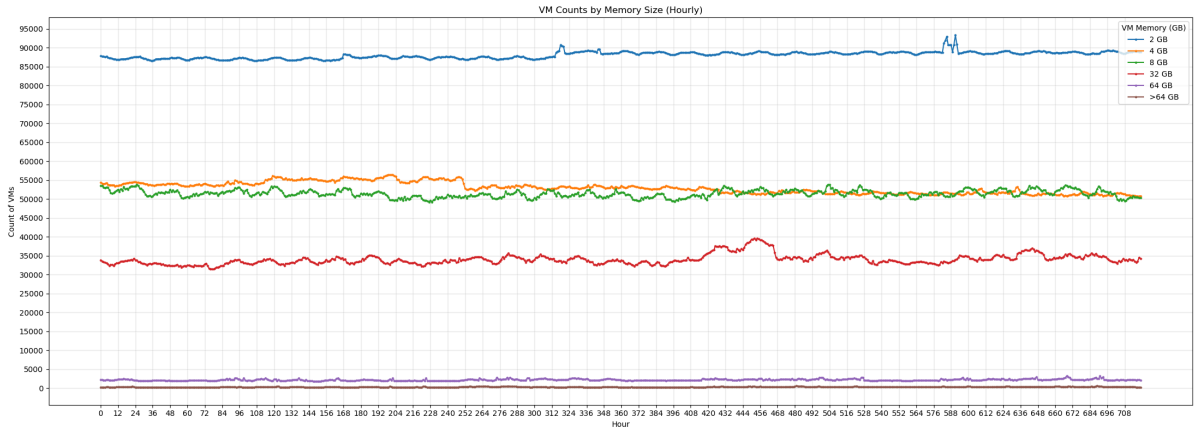


In [47]:
```
vm_counts = (hourly_data.groupby(['hour'])['vm_id'].nunique().reset_index(na
vm_counts = vm_counts.dropna(subset=['hour', 'num_vms'])
plot_with_24h_means(data=vm_counts, col_name='num_vms', color='brown', ylabe
```

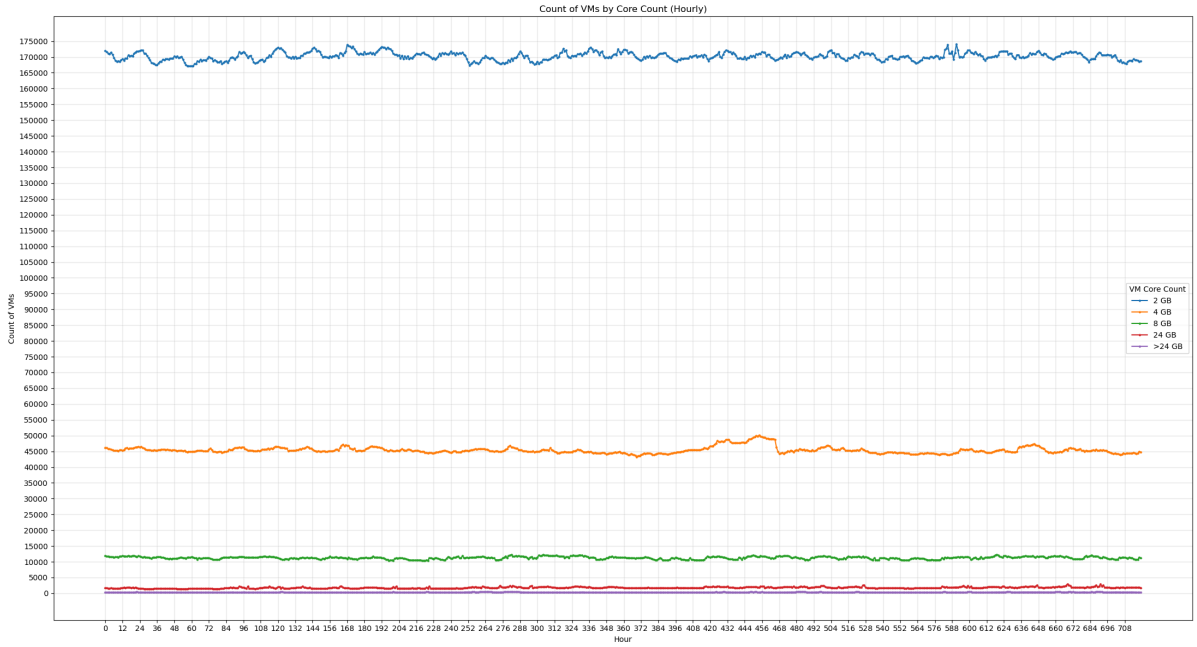


In [48]:
```
vm_order = ['2', '4', '8', '32', '64', '>64']
hourly_data['vm_memory'] = pd.Categorical(hourly_data['vm_memory'], categor
# Grouping by hour and vm_memory size to get the count of VMs
memory_counts = hourly_data.groupby(['hour', 'vm_memory']).size().unstack(f
```

```
plt.figure(figsize=(22, 8))
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b']
for vm, color in zip(vm_order, colors):
    plt.plot(memory_counts['hour'], memory_counts[vm], label=f'{vm} GB', co

plt.title('VM Counts by Memory Size (Hourly)')
plt.xlabel('Hour')
plt.ylabel('Count of VMs')
plt.xticks(ticks=np.arange(0, memory_counts['hour'].max() + 1, 12))
plt.yticks(np.arange(0, memory_counts[vm_order].values.max() + 5000, 5000))
plt.grid(True, alpha=0.3)
plt.legend(title='VM Memory (GB)')
plt.tight_layout()
plt.savefig('azure_plots/Azure_vm_memory_counts_per_hour.png')
plt.show()
```



In [49]:
```
vm_core_order = ['2', '4', '8', '24', '>24']
hourly_data['vm_core_count'] = pd.Categorical(hourly_data['vm_core_count'],
# Grouping by hour and vm_core_count size to get the count of VMs
core_counts = hourly_data.groupby(['hour', 'vm_core_count']).size().unstack(

plt.figure(figsize=(22, 12))
colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd']
for vm, color in zip(vm_core_order, colors):
    plt.plot(memory_counts['hour'], core_counts[vm], label=f'{vm} GB', colo

plt.title('Count of VMs by Core Count (Hourly)')
plt.xlabel('Hour')
plt.ylabel('Count of VMs')
plt.xticks(ticks=np.arange(0, core_counts['hour'].max() + 1, 12))
plt.yticks(np.arange(0, core_counts[vm_core_order].values.max() + 5000, 5000
plt.grid(True, alpha=0.3)
plt.legend(title='VM Core Count')
plt.tight_layout()
plt.savefig('azure_plots/Azure_vm_core_counts_per_hour.png')
plt.show()
```



In [50]:
```
vm_categories = ['Interactive', 'Unknown', 'Delay-insensitive']
hourly_data['vm_category'] = pd.Categorical(hourly_data['vm_category'], cate
# Grouping by hour and vm_category to get the count of VMs
category_counts = hourly_data.groupby(['hour', 'vm_category']).size().unsta
plt.figure(figsize=(22, 8))
colors = ['#1f77b4', '#ff7f0e', '#2ca02c']

for cat, color in zip(vm_categories, colors):
    plt.plot(category_counts['hour'], category_counts[cat], label=cat, colo

plt.title('Count of VMs by Category (Hourly)')
plt.xlabel('Hour')
plt.ylabel('Count of VMs')
plt.xticks(np.arange(0, category_counts['hour'].max() + 1, 12))
plt.grid(True, alpha=0.3)
plt.legend(title='VM Category')
plt.tight_layout()
plt.savefig('azure_plots/Azure_vm_category_counts_per_hour.png')
plt.show()
```