Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <!-- TextView to display network status -->  
 <TextView  
 android:id="@+id/networkStatusTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Network Status: Unknown"  
 android:textSize="18sp"  
 android:textColor="#000"  
 android:layout\_centerInParent="true"/>  
  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

Mainactivity.java

package com.example.myapplication;  
  
import android.content.IntentFilter;  
import android.net.ConnectivityManager;  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 private NetworkChangeReceiver networkChangeReceiver;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Initialize the BroadcastReceiver  
 networkChangeReceiver = new NetworkChangeReceiver();  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 // Register the BroadcastReceiver dynamically  
 IntentFilter filter = new IntentFilter(ConnectivityManager.*CONNECTIVITY\_ACTION*);  
 registerReceiver(networkChangeReceiver, filter);  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 // Unregister the BroadcastReceiver to avoid memory leaks  
 unregisterReceiver(networkChangeReceiver);  
 }  
}

Manifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
<!-- <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />-->  
  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
  
</manifest>

NetworkChangeReceiver.java

package com.example.myapplication;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class NetworkChangeReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 ConnectivityManager cm = (ConnectivityManager) context.getSystemService(Context.*CONNECTIVITY\_SERVICE*);  
 NetworkInfo activeNetwork = cm.getActiveNetworkInfo();  
  
 // Get reference to the TextView in MainActivity  
 TextView networkStatusTextView = ((MainActivity) context).findViewById(R.id.*networkStatusTextView*);  
  
 if (activeNetwork != null && activeNetwork.isConnected()) {  
 // Update TextView with network status  
 networkStatusTextView.setText("Network Status: Connected");  
 Toast.*makeText*(context, "Network Connected", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 // Update TextView with network status  
 networkStatusTextView.setText("Network Status: Disconnected");  
 Toast.*makeText*(context, "Network Disconnected", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

OUTPUT:



